Data-driven Stochastic Vehicle Routing Problems with Deadlines

Shanshan Wang
GERAD & Department of Decision Sciences, HEC Montréal, Canada H3T 2A7
shanshan.wang@hec.ca

Erick Delage
GERAD & Department of Decision Sciences, HEC Montréal, Canada H3T 2A7
erick.delage@hec.ca

Leandro C. Coelho
GERAD & Department of Operations and Decision Systems, Université Laval, Canada G1V 0A6
leandro.coelho@fsa.ulaval.ca

Vehicle routing problems (VRPs) with deadlines have received significant attention around the world. Motivated by a real-world food delivery problem, we assume that the travel time depends on the routing decisions, and study a data-driven stochastic VRP with deadlines and endogenous uncertainty. We use the non-parametric approaches, including k-nearest neighbor (kNN) and kernel density estimation (KDE), to estimate the decision-dependent probability distribution of travel time. To solve the resulting problem efficiently, we employ a logic-based Benders decomposition (LBBD) algorithm with several algorithmic enhancements. In particular, we propose a novel family of optimality cuts that includes the expected delay for all the sub-routes. Moreover, we solve a total travel cost minimization problem to warm-start the algorithm. We also use a local search procedure to improve the current routing decision and propose a machine learning-based lower bound heuristic to efficiently solve problems of realistic size. A practical case study for a food delivery routing problem using real-world data is conducted to show the efficiency of the proposed techniques and the advantage of the data-driven stochastic VRP in reducing the expected delay. In our case study, we show that incorporating routing decisions in a non-parametric model outperforms a state-of-the-art data-driven parametric model by 23% on average in terms of the expected delay, and the order-assignment decisions obtained from a robust model with travel-time predictors by 26% on average. Moreover, compared with the drivers' actual routes, our suggested routes can significantly improve the on-time performance of delivery services. We also quantify the value of the proposed routes with different service deadlines.

Key words: Data-driven stochastic vehicle routing, machine learning, Logic-based Benders decomposition, prescriptive analytics, deadline, endogenous uncertainty

1. Introduction

Vehicle routing problems (VRPs) define a highly studied class of combinatorial optimization problems in which products are delivered from a single depot to fulfill customers' demands using a fleet of capacitated vehicles (Toth and Vigo 2014). Deliveries usually have penalties for violated deadlines, and the goal is to minimize the total penalty cost while determining optimal vehicle routes to fulfill the demand. As a fundamental problem in the field of operations research and management science, VRPs have received significant attention and have been widely applied in transportation and logistics operations management, e.g., food delivery, school bus routing and scheduling, traveling salesman problems, waste collection, and ride-sharing services.

The classical VRPs assume that all the model parameters are known in advance. However, this assumption usually does not hold in real-world VRP applications. For example, in a food delivery problem, the driver's travel time is highly uncertain because it largely depends on many factors (e.g., driver characteristics). These might make the optimal decisions for deterministic VRPs to fail to be executed by the drivers in practice in order to deliver their targeted on-time service. To overcome these difficulties, one can employ stochastic programming (e.g., Birge and Louveaux 2011) and robust optimization (RO, e.g., Bertsimas et al. 2011) paradigms to model and address the uncertainties involved in stochastic VRPs. Depending on the method, it is assumed that either the distribution of uncertain parameters is estimated from historical observations, or that it is unknown but resides in a predefined uncertainty set. Methodologically, most of the existing studies address the stochastic VRPs using the above two approaches (Carlsson and Delage 2013, Jaillet et al. 2016, Liu and Luo 2023).

1.1. The Motivation of Endogenous Travel Time Uncertainty

Following the relevant literature (Adulyasak and Jaillet 2016, Zhang et al. 2021), we also assume that the travel time is uncertain, and its distribution can be estimated by historical observations. However, unlike these papers, we assume that the distribution of uncertain travel time depends on the routes. That is to say, travel uncertainty is endogenous and the assignment and routing decisions made in the stochastic VRPs could affect the travel time on the arc. This type of decision (or route)-dependent travel time uncertainty is motivated by real-world applications.

In food delivery services, the provider prepares the food at its central kitchen, which is referred to as the depot hereafter, and delivers it to customers within a certain radius of the depot. Customers place orders before each cutoff time and are promised delivery by a deadline. The orders with the same cutoff time are dispatched together and have the same delivery deadline. The planners determine order assignments and routes for each driver (Liu et al. 2021). If the last customer cannot be served before the deadline, a delay cost will be incurred per unit of time.

For a food delivery problem or many other last-mile delivery problems, given the assignment and routing decisions, drivers tend to decide their speed to meet the deadline based on their experience and real-time road conditions (Cui et al. 2023). For example, given an assignment and route, the driver has an estimated travel time for each arc and an estimated completion time for the last order. If this estimated completion time is close to or larger than the deadline, the driver will speed up, leading to a shorter travel time. In last-mile delivery, drivers can use electric bikes, which are allowed to take the bike lanes, as a result, the driver has more freedom to decide their speed. In fact, the intuition that travel time on an arc of the network is dependent on the total route can be confirmed empirically using a hypothesis test. In our case study, we observed that on the most traveled arcs, the null hypothesis that travel time is independent of the route could safely be rejected in favor of a dependence relation. Therefore, it is reasonable and essential for decision-makers to construct a route-dependent travel time distribution to better capture the drivers' behaviors and make the decisions for assigning orders to drivers and routing the deliveries. As indicated by the case study, compared with the drivers' real routes, our suggested routes significantly reduce the expected delays.

Recently, optimization problems under the endogenous uncertainty have been studied in the context of stochastic programming (e.g., Goel and Grossmann 2006) and DRO (e.g., Yu and Shen 2022), which assume that decisions can directly affect the distribution of uncertain parameters. Given the endogenous nature of uncertainty, they are notoriously hard both to model and solve and, not surprisingly, the relevant literature is quite sparse. To address the modeling challenge, with the help of the emerging field of prescriptive analytics (e.g., Bertsimas and Kallus 2020, Sadana et al. 2023, Jiang et al. 2024), we employ prescriptive machine learning (ML) tools to estimate a route-dependent distribution of travel time directly from historical data. To address the computational challenge, we develop an efficient logic-based Benders decomposition (LBBD) algorithm with acceleration strategies to solve real-world VRP applications of realistic size.

As the most relevant literature to our problem, Liu et al. (2021) also assumed that the travel time is dependent on the order-assignment decisions and proposed a data-driven decision-making framework that integrates mixed integer linear program (MILP) representable travel-time predictors with an order-assignment problem. Our work differs in two ways (at least). First, it extends the model beyond MILP representable travel time. Second, Liu et al. (2021) only assigns orders to drivers without detailed route planning considering that significant deviations may exist between the drivers' routes and the planned ones. In this paper, we also employ predictors that account for a routing decision, i.e., the order in which the deliveries should be made. Given that our model learns about the characteristics of the real routes to design routes. These routes could be more acceptable to drivers. As indicated by a real-world food delivery application presented in Section 5, we improve the on-time performance of delivery services by making joint decisions regarding order assignment and routing when compared with Liu et al. (2021). Moreover, by learning the deviation from the suggested and actual delivery routes, one could further capture drivers' preferences and improve the likelihood of drivers accepting the suggested routes (see, e.g., Fu et al. 2023). Therefore, from the perspective of future implementation, designing routes plays an important role in increasing driver's trust in the platform. The goal of this paper is to contribute a prescriptive ML-based decision support methodology that can use covariate data for making both the assignment and routing decisions of stochastic VRP in a data-driven context.

1.2. Contributions of This Paper

While the emerging field of prescriptive analytics has received much attention in recent years, to the best of our knowledge, this paper presents the first comprehensive study of a prescriptive ML approach for a general data-driven stochastic VRP with deadlines and endogenous travel time uncertainty. We provide novel model formulations, efficient solution algorithms, and an extensive real-world case study. Our contributions are described as follows.

- We propose a novel prescriptive ML approach for a data-driven stochastic VRP with deadlines and uncertain travel time. Motivated by real-world food delivery problems, we consider that the uncertain travel time depends on the routes, and thus incorporate endogenous travel time uncertainty in our VRP model. We then use non-parametric approaches, including k-nearest neighbor (kNN) and kernel density estimation (KDE), to estimate the route-dependent probability distribution of travel time.

- We develop an efficient LBBD method with acceleration strategies to solve the data-driven stochastic VRP, including general optimality cuts, warm start, and local search. However, given the difficulty involved with the route-dependent travel time uncertainty, we further propose an ML-based lower bound heuristic to efficiently solve instances of realistic size. Specifically, we introduce a distribution, which is first-order stochastically dominated by the distribution of travel time. Motivated by the conformal prediction, we then use a finite scenario method to model the distribution and add the corresponding linear cuts to the VRP model. To the best of our knowledge, this is the first time that an ML-based lower bound cut generation strategy is used when implementing the LBBD method in the literature.

- We conduct a practical case study for a food delivery routing problem using real-world data. We find that for the data-driven stochastic VRP with the kNN method, the acceleration strategies and lower bound improvement heuristic implemented in the LBBD decrease the average absolute gap by about 93% when compared to the default LBBD, and 90% when compared to a genetic algorithm based on Vidal et al. (2012). For KDE, the improvement is about 98% on average when compared to the default LBBD, and 91% when compared to the genetic algorithm. For the out-of-sample simulated performance, the results show that our non-parametric model with routing decisions reduces the median of the out-of-sample expected delay by about 23.1% on average when compared with an adjusted version of the assignment parametric models proposed by Liu et al. (2021), about 25.8% when compared with the order-assignment decisions obtained from the DOA-DRO model used in Liu et al. (2021). Moreover, our suggested routes significantly outperform the implemented route decisions observed in the real data set. We also quantify the value of the proposed routes with different service deadlines.

The remainder of this paper is organized as follows. Section 2 reviews the literature on stochastic VRPs with deadlines, ML-based VRPs, and the LBBD method for VRPs. Section 3 presents our data-driven stochastic VRP. We describe our LBBD solution scheme to solve the problem in Section 4. We further propose several acceleration strategies and develop a lower bound heuristic. We finally conduct an extensive case study using real-world food delivery data in terms of model training, computational efficiency of the proposed LBBD solution scheme, and out-of-sample performance of our data-driven solutions in Section 5. Section 6 concludes this paper with a summary of the important findings. The Online Appendix includes all the proofs and supporting material.

2. Literature Review

We identify three streams of relevant literature. In Section 2.1, we briefly review the studies on the stochastic VRPs with deadlines. In Section 2.2, we review the studies using ML methods to handle VRPs, and also briefly connect the general problem with the framework of integrated ML and optimization under uncertainty. Finally, Section 2.3 reviews the studies on the LBBD method for VRPs and the ML-based cut generation strategies for the decomposition method.

2.1. Stochastic VRPs with Deadlines

VRPs are well-studied problems in the field of transportation science, which have received great attention. There are several reviews and surveys for VRPs in the literature. For instance, Laporte (1992) reviewed the main exact and approximate algorithms developed for VRPs in the early days. Coelho et al. (2014) gave a comprehensive review of the literature on VRPs integrated with inventory management.

Most of the studies assume that all the parameters are deterministic (e.g., Laporte 1992). However, some parameters (e.g., customer demand, travel time) can be uncertain in practice. Thus, a number of studies on stochastic VRPs have proposed mathematical optimization frameworks for solving VRPs under different sources of uncertainties.Generally speaking, there are mainly two types of solution schemes for handling uncertainties in the literature: stochastic programming and RO. The stochastic programming approach typically assumes that the distribution of random uncertain parameters is known, which can be estimated by historical samples. In contrast, RO assumes that the distribution of random uncertain parameters is unknown, and is assumed to belong to a predefined uncertainty set.

Since our work is related to stochastic VRPs with deadlines, in the following we mainly focus on the literature on stochastic VRPs with deadlines and different sources of uncertainty. Extensive studies have explored the use of a stochastic programming approach for this type of problem. Generally, existing studies fit broadly into different categories in terms of the sources of uncertainty: stochastic demand (e.g., Hooshmand Khaligh and MirHassani 2016, Ghosal et al. 2023), uncertain travel time (e.g., Adulyasak and Jaillet 2016), and stochastic availability of vehicles (Torres et al. 2022). In particular, Hooshmand Khaligh and MirHassani (2016) proposed a multistage stochastic programming model for a single-vehicle routing problem under endogenous demand uncertainty from a dynamic viewpoint when the expected total routing cost is minimized. However, the problem size that can be addressed by their method is limited.

Our work differs from the above studies because 1) we propose a novel prescriptive ML approach for a data-driven stochastic VRP with deadlines and uncertain travel time; 2) motivated by real-world food delivery problem, we capture the feature of VRP where the uncertain travel time depends on the routes, and thus incorporate the endogenous uncertainty in the VRP; 3) we propose an efficient LBBD algorithm for solving the resulting model with instances of realistic size.

2.2. Machine Learning for VRPs

With advances in data science techniques and the rapid availability of data in real life, there has been rapidly increasing interest in the applications of ML tools to address the challenges (e.g., uncertainty and dynamics) that are involved in real-world VRP applications (Soeffker et al. 2022). These ML tools have been integrated with analytical/optimization approaches to strengthen problem formulations or improve algorithmic performance under different problem settings.

One of the main streams of research is to use reinforcement learning (RL) to address the dynamics in real-time vehicle routing and relocation problems. Its idea is to employ RL techniques to improve the VRP algorithms and construct solutions (e.g., Nazari et al. 2018). It trains a state-based decision policy by interacting with the environment and observing the rewards. For example, Nazari et al. (2018) proposed an end-to-end framework for solving VRPs using RL, where they trained a single policy model that finds near-optimal solutions only by observing the reward signals and following feasibility rules. We refer interested readers to surveys by Hildebrandt et al. (2023) for more recent advances in using RL for dynamic VRPs in logistics and supply chain management. Although RL-based methods provide an efficient way to handle the dynamics for real-world VRPs, they require a tremendous amount of data to explore high-dimensional state-action spaces and simplify the problem by ignoring uncertainties and related covariates to ensure efficient training.

Recently, the method of prescriptive analytics has been rapidly developed for data-driven decision-making problems. The main idea is to first predict or estimate the uncertainty behavior using data and then find optimal decisions by solving an optimization model that depends on the output of prediction or estimation (e.g., Nguyen et al. 2020, Sadana et al. 2023). This type of method has also been widely used in various applications (Mišić and Perakis 2020), e.g., portfolio selection, inventory management, and last-mile delivery.

Our work is more closely related to this stream of research, which mainly relies on this prescriptive ML method to address a general data-driven stochastic VRP. We employ ML techniques to handle the travel time uncertainty and estimate the distribution of travel time, which is dependent on the routing decisions. To the best of our knowledge, our work is the first to propose a prescriptive ML approach to address a general data-driven stochastic VRP with deadlines and endogenous travel time in the literature. Since related studies for applying the discussed prescriptive ML method to address a general stochastic VRP are rare, here we also include studies that are close to VRPs and their variants, e.g., meal delivery with fixed routes (Hildebrandt and Ulmer 2022), order assignment (Liu et al. 2021), and deterministic last-mile delivery problem (Serkan Özarık et al. 2022).

Perhaps, the closest related study consists of Liu et al. (2021), who proposed a data-driven decision-making framework that integrates MILP representable travel-time predictors with an order assignment problem. Leveraging recent advances in stochastic and DRO, they discussed tractable predictors and linear prediction models and derived the reformulations of integrated models, where they assumed that the distribution of service time is unknown but resides in a moment-based ambiguity set. The resulting model can be efficiently solved by the proposed branch-and-price algorithm. Using data for a real-world food delivery problem, they demonstrated the superior performance of their proposed order-assignment models over the SAA approach. Our work differs from theirs with respect to the following aspects: 1) instead of only deciding the order dispatch decision, we further incorporate the routing decisions in the data-driven decision-making model and address a general VRP model with deadlines; 2) our model accounts for arbitrary ML travel-time predictors while Liu et al. (2021) needed MILP representable ones; 3) we consider travel time uncertainty and use non-parametric ML approaches to estimate its distribution, which is dependent on the routing decisions. In contrast, Liu et al. (2021) provided a point estimate on the average total travel time; 4) we also propose an efficient LBBD algorithm with acceleration strategies to solve the resulting VRP model of realistic size; 5) using the same set of real-world data, we show that our proposed model outperforms their model in terms of the expected delay.

2.3. Logic-based Benders Decomposition for VRPs

Another stream of related research consists of using LBBD for solving VRPs and ML for improving the decomposition schemes. The LBBD method is a generalized Benders decomposition, which allows subproblems to be mixed-integer programs (instead of linear programs in classical Benders decomposition). The LBBD method has been proven to efficiently solve a large number of combinatorial optimization problems, such as bin-packing problems, VRPs, home healthcare delivery, and location-transportation problems. We refer interested readers to a survey by Rahmaniani et al. (2017) for more recent advances in LBBD.

Specifically, several studies employ the LBBD method to solve VRP variants in the literature (e.g., Li et al. 2022). Most of these studies implement LBBD with enhancements, including optimization cut, branch-and-check, and subproblem relaxation.

In this paper, we also propose an efficient LBBD method with acceleration strategies to solve stochastic VRP, including warm start, local search, and optimality cuts. However, given the difficulty of our model involved with route-dependent travel time uncertainty, we propose an ML-based lower-bound heuristic to solve the problems of realistic size efficiently. To the best of our knowledge, we are the first to propose an ML-based lower bound cut generation strategy in the implementation of the LBBD method in the literature.

In this regard, our work is also related to the literature on the use of ML tools to generate or learn the cuts for improving the algorithmic efficiency of decomposition approaches when solving stochastic programs. However, the related literature is very limited. More recently, Larsen et al. (2022b) proposed a method to quickly predict expected tactical descriptions of operational solutions in the context of two-stage stochastic programming when the second stage is demanding computationally. Larsen et al. (2022a) studied mixed-integer linear two-stage stochastic programs where the second stage is computationally demanding. Their idea is to gain large reductions in online solution time while incurring small reductions in first-stage solution accuracy by substituting the exact second-stage solutions of Benders decomposition with fast and accurate supervised ML predictions. Unlike the aforementioned studies, we use an ML-based method to model a distribution, which is approximately first-order stochastically dominated by the distribution of travel time, and add the corresponding linear constraints to the master problem of LBBD method.

3. Problem Description and Formulation

In Section 3.1, we introduce the proposed routing set (see, e.g., Adulyasak and Jaillet 2016, Zhang et al. 2021) used in our model. Then, we present a data-driven stochastic model formulation for VRP with deadlines and endogenous travel time uncertainty in Section 3.2.

3.1. Description of the Routing Set

The routing set is to design at most L a priori routes to service all customers. It requires that each customer location is serviced exactly once and by only one vehicle; each vehicle, if used, starts from the depot and ends at the last visited customer location, without exceeding its capacity. The problem is defined on a complete and directed graph $\hat{\mathcal{G}} = (\hat{\mathcal{N}}, \hat{\mathcal{A}})$, where $\hat{\mathcal{N}} = \{0\} \cup \mathcal{N}_c$ and $\hat{\mathcal{A}} = \{(i,j) \,|\, i,j \in \hat{\mathcal{N}}, i \neq j\}$ represent a set of nodes and arcs, respectively. Node 0 is the depot, which is equipped with L homogeneous vehicles of maximum capacity Q, and nodes in set $\mathcal{N}_c = \{1,\dots,I\}$ represent a set of I customer locations. Each node $i \in \mathcal{N}_c$ has a demand d_i. The latest time for starting the service (i.e., deadline) is denoted by τ, and all customer locations have the same τ. For each arc $(i,j) \in \hat{\mathcal{A}}$, the random travel time is denoted by $\tilde{t}_{ij}$. Without loss of generality, we assume that for $(i,j) \in \hat{\mathcal{A}}$, $\tilde{t}_{ij}$ includes the service time at node $i \in \mathcal{N}_c$.

We add L dummy nodes for vehicles: $\mathcal{N}_d = \{I+1,\dots,I+L\}$ to denote the destinations (see, e.g., Adulyasak and Jaillet 2016). We then transform the original graph to an extended one $\mathcal{G} = \{\mathcal{N}, \mathcal{A}\}$, where $\mathcal{N} = \{0\} \cup \mathcal{N}_c \cup \mathcal{N}_d$, and $\mathcal{A} = \mathcal{A}_o \cup \mathcal{A}_c \cup \mathcal{A}_d$ with $\mathcal{A}_o = \{(0,j) \,|\, j \in \mathcal{N}_c\}$, $\mathcal{A}_c = \{(i,j) \,|\, i,j \in \mathcal{N}_c, i \neq j\}$, $\mathcal{A}_d = \{(i,j) \,|\, i \in \mathcal{N}_c, j \in \mathcal{N}_d\}$. In the extended graph, for the node $i \in \mathcal{N}_d$, the demand $d_i = 0$, and the travel time of the arcs in $\mathcal{A}_d$ is set to zero. Given any subset $\mathcal{H} \subseteq \mathcal{N}$ of nodes, the sets of their outgoing and incoming arcs are defined as

$$E^+(\mathcal{H}) = \{(i,j) \in \mathcal{A} \,|\, i \in \mathcal{H}, j \in \mathcal{N} \backslash \mathcal{H}\},$$
$$E^-(\mathcal{H}) = \{(j,i) \in \mathcal{A} \,|\, i \in \mathcal{H}, j \in \mathcal{N} \backslash \mathcal{H}\}.$$

For simplicity, we use (i,j) and a interchangeably to represent an arc in $\mathcal{A}$, and $E^+(i)$ and $E^+(\{i\})$ interchangeably for $i \in \mathcal{N}$.

The following decision variables are used in the routing set. The binary variable $x_a = 1$ if a is part of some vehicle's route, and 0 otherwise. The binary variable s_a^ℓ is equal to one if arc a is part of the route to node $\ell \in \mathcal{N}_d$. We denote $\boldsymbol{s} = (\boldsymbol{s}^\ell)_{\ell \in \mathcal{N}_d}$ and $\boldsymbol{s}^\ell \in \{0,1\}^{|\mathcal{A}|}$, where $|\cdot|$ be the cardinality of a set. A feasible routing set $\mathcal{S}$ is defined as

$$\mathcal{S} := \left\{ \boldsymbol{x} \in \{0,1\}^{|\mathcal{A}|}, \boldsymbol{s} \in \{0,1\}^{|\mathcal{A}| \times |\mathcal{N}_d|} \,|\, (1a) - (1g) \right\},$$

namely that there exist an $\boldsymbol{x} \in \{0,1\}^{|\mathcal{A}|}$ such that

$$\sum_{a \in E^+(i)} x_a = \sum_{a \in E^-(i)} x_a = 1, \qquad \forall i \in \mathcal{N}_c, \tag{1a}$$

$$\sum_{a \in E^+(0)} x_a = \sum_{\ell \in \mathcal{N}_d} \sum_{a \in E^-(\ell)} x_a, \tag{1b}$$

$$\sum_{a \in E^-(\ell)} x_a \leq 1, \qquad \forall \ell \in \mathcal{N}_d, \tag{1c}$$

$$\sum_{a \in E^+(0)} s_a^\ell = \sum_{a \in E^-(\ell)} s_a^\ell = \sum_{a \in E^-(\ell)} x_a, \qquad \forall \ell \in \mathcal{N}_d, \tag{1d}$$

$$\sum_{a \in E^-(i)} s_a^\ell - \sum_{a \in E^+(i)} s_a^\ell = 0, \qquad \forall \ell \in \mathcal{N}_d, i \in \mathcal{N}_c, \tag{1e}$$

$$s_a^\ell \le x_a, \qquad \forall \ell \in \mathcal{N}_d, a \in \mathcal{A}, \tag{1f}$$

$$\sum_{a \in E^+(\mathcal{H})} x_a \ge \left\lceil \frac{\sum_{i \in \mathcal{H}} d_i}{Q} \right\rceil, \qquad \forall \mathcal{H} \subseteq \mathcal{N}_c, |\mathcal{H}| \ge 2. \tag{1g}$$

Constraints (1a) state that all customers must be visited. Constraints (1b) ensure that the number of vehicles leaving the depot and arriving at the destination equals each other. Constraints (1c) denote that one can choose whether or not to visit the nodes in set $\mathcal{N}_d$. Constraints (1d) are used to guarantee that the route to node ℓ should start at the depot and arrive at node ℓ. Constraints (1e) denote the flow balance should hold at the intermediate nodes. Constraints (1f) link the arc variables x_a and the route variables s_a^ℓ. Constraints (1g) are subtour elimination and vehicle capacity constraints, which are also referred to as rounded capacity inequalities (Lysgaard et al. 2004). One can also add the following set of symmetry-breaking constraints:

$$\sum_{a \in \mathcal{A}} s_a^\ell \le \sum_{a \in \mathcal{A}} s_a^{\ell-1}, \quad \forall \ell \in \mathcal{N}_d \backslash \{I+1\}. \tag{2}$$

Constraints (2) impose that the number of customers served by vehicle ℓ is larger than the number served by $\ell - 1$, which implicitly ensure that if vehicle ℓ is used to serve the customers, vehicle $\ell - 1$ must be used.

For our multiple capacitated vehicles problem, as stated by Adulyasak and Jaillet (2016), one could use the original graph and incorporate the index $\ell \in \{1, \ldots, L\}$ into variables $\boldsymbol{x}$ and $\boldsymbol{s}$. Then one could impose vehicle capacity constraints on variable $\boldsymbol{x}$ for each vehicle separately. However, the problem size would be significantly increased. Therefore, in this paper, we use the extended graph to formulate the feasible routing set.

3.2. Data-Driven Stochastic VRP Formulation

Motivated by a real-world food delivery problem, we construct a route-dependent travel time distribution in order to better capture the drivers' behaviors and make the routing decisions for the drivers. We consider the VRP with the deadlines and endogenous uncertainty, where the distribution of the random travel time for routes depends on the decision $\boldsymbol{s}$. The objective is to minimize the total expected delay for each route, which is widely used in the literature about VRP with deadlines (e.g., Adulyasak and Jaillet 2016). More specifically, the VRP model is given as follows:

$$\underset{(\boldsymbol{x},\boldsymbol{s}) \in \mathcal{S}}{\text{minimize}} \sum_{\ell \in \mathcal{N}_d} \mathbb{E}_{\mathbb{P}_{(\boldsymbol{s}^\ell)}} \left(\tilde{\xi} - \tau\right)^+, \tag{3a}$$

where $\tilde{\xi}$ is the random travel time for route $\boldsymbol{s}^\ell$ with the distribution $\mathbb{P}_{(\boldsymbol{s}^\ell)}$. If the travel time uncertainty $\tilde{\boldsymbol{t}}$ for each arc follows a known probability distribution $\mathbb{P}$ independent of $\boldsymbol{s}^\ell$, then $\tilde{\xi} = \tilde{\boldsymbol{t}}^\top \boldsymbol{s}^\ell$, and the distribution $\mathbb{P}_{(\boldsymbol{s}^\ell)}$ in problem (3) can be replaced by the distribution $\mathbb{P}$.

In a realistic situation, the decision maker can collect data on the travel time and side information about the routes, providing a more accurate description of the uncertain parameters that can be used to make the present decision. Given a set of historical data $\{\hat{\xi}^1,\dots,\hat{\xi}^N\}$, where N is the number of scenarios, and $\hat{\xi}^i \in \mathbb{R}_+$ represents the total travel time about the route $i=\{0,n^i_{(1)},\dots,n^i_{(v_i)}\}$ for $i=1,\dots,N$, and $n^i_{(j)} \in \mathcal{N}_c$ is the jth customer location in the route i, v_i is the number of customer locations serviced by the route i. In this paper, we exploit real data to capture the drivers' behavior and describe the uncertainty with

$$\mathbb{P}_{(\boldsymbol{s}^\ell)} = \sum_{i=1}^{N(\boldsymbol{s}^\ell)} p_i(\boldsymbol{s}^\ell)\delta_{\xi^i(\boldsymbol{s}^\ell)}, \tag{4}$$

where $N(\boldsymbol{s}^\ell)$, $p_i(\boldsymbol{s}^\ell)$ and $\xi^i(\boldsymbol{s}^\ell)$ are the number of scenarios, the probability (or weight) and the total travel time for scenario $i \in \{1,\dots,N(\boldsymbol{s}^\ell)\}$, and are all represented as functions of $\boldsymbol{s}^\ell$. $\delta_{\xi^i(\boldsymbol{s}^\ell)}$ is a Dirac distribution concentrating its unit mass at $\xi^i(\boldsymbol{s}^\ell)$.Then, the stochastic data-driven VRP (3) can be rewritten as follows:

$$\underset{(\boldsymbol{x},\boldsymbol{s})\in\mathcal{S}}{\text{minimize}} \sum_{\ell\in\mathcal{N}_d}\sum_{i=1}^{N(\boldsymbol{s}^\ell)} p_i(\boldsymbol{s}^\ell)\left(\xi^i(\boldsymbol{s}^\ell)-\tau\right)^+. \tag{5a}$$

The objective function (5a) can be seen as a weighted average of historical delay observations.

There are several ways for ML algorithms to construct such probability functions. In this paper, we focus on kNN and KDE methods, which are popular nonparametric ML methods with great prediction power. For historical route $i \in \{1,\cdots,N\}$, let $\hat{\gamma}^i \in \mathbb{R}_+$ be the feature (or the side information) of the route, and $\bar{\gamma}^\ell$ be the feature of the route $\boldsymbol{s}^\ell$. We describe the following probability functions:

Definition 1 *The kNN for constructing a probability function is defined as follows:*

$$p_i(\boldsymbol{s}^\ell)=\begin{cases}1/k, & \textit{if } \hat{\gamma}^i \textit{ is a k-nearest neighbor of } \bar{\gamma}^\ell,\\ 0, & \textit{otherwise.}\end{cases}$$

$\hat{\gamma}^i$ is a k-nearest neighbor of $\bar{\gamma}^\ell$ if $\left|\{j\in\{1,\dots,n\}\backslash i : \|\hat{\gamma}^j-\bar{\gamma}^\ell\| < \|\hat{\gamma}^i-\bar{\gamma}^\ell\|\}\right| < k$, where it is assumed for simplicity that there are no ties.

Definition 2 *The KDE for constructing a probability function is defined as follows:*

$$p_i(\boldsymbol{s}^\ell)=\frac{K_{\boldsymbol{H}}\left(\bar{\gamma}^\ell-\hat{\gamma}^i\right)}{\sum_{j=1}^n K_{\boldsymbol{H}}\left(\bar{\gamma}^\ell-\hat{\gamma}^j\right))},$$

where $K(\cdot)$ is a kernel function, $K_{\boldsymbol{H}}(\boldsymbol{x}) = |\boldsymbol{H}|^{-1/2}K(\boldsymbol{H}^{-1/2}\boldsymbol{x})$, and $\boldsymbol{H}$ is bandwidth diagonal matrix. In this paper, we consider the Gaussian kernel function given by

$$K(\boldsymbol{x}) = (2\pi)^{-d/2}\exp(-\frac{1}{2}\boldsymbol{x}^\top\boldsymbol{x}).$$

Nonparametric methods favor generality and are effective since they do not require parametric assumptions. For the kNN and KDE methods, extremely large values of the parameter k and elements in bandwidth matrix $\boldsymbol{H}$ imply that the VRP model (5) reduces to a problem with an equal probability over all observations for all routes. Conversely, by using small values, most of the probability mass is assigned to the samples closest to route $\boldsymbol{s}^\ell$. For more details on the above methods, see García-Portugués (2023).

For more complex conditional generative models with complicated nonlinear structures, such as the conditional Gaussian mixture model (CGMM, see, e.g., Gilardi et al. (2002)), they are also compatible with our VRP model. It is well known that the conditional generative models can easily generate numerical characteristics of the underlying distributions or other performance estimations through Monte Carlo methods (Huang et al. 2022). Therefore, compared to the kNN and KDE methods, generative models are able to generate new data, which have no obvious connection to probability distributions over samples of input variables, and could be essential for evaluating the out-of-sample performance of routes obtained from problem (5) and demonstrating the computational efficiency of our solution scheme that is proposed in Section 4.

In terms of the features selection, we use the features proposed in the literature, which are based on the number of locations, visiting area, distance, dispersion, and their interactive terms (Liu et al. 2021). Çavdar and Sokol (2015) used $\sqrt{I(cstedv_a cstedv_b)}$ to address the dispersion, where $cstedv_a$ and $cstedv_b$ are the standard deviations of the latitudinal and longitudinal differences between the depot and customer locations, respectively. Chien (1992) and Kwon et al. (1995) suggested $\sqrt{SI}$, $I\sqrt{S}$, $\sqrt{S'(I+1)}$, $\sqrt{S'(I+1)^3}$, $\sqrt{\frac{S'}{I+1}}$ to account for the visiting area, where S and S' are the area of the smallest rectangle covering the customer locations visited by a driver and covering both the depot and the customer locations visited by a driver, respectively. As a result, the above features as well as the number of customer locations I and the length of the routes are specified as the feature vector. Because of the availability of data, we focus our attention on the characteristics of the route itself in the empirical application. Although driver characteristics are not considered in this paper, these and other features can be easily incorporated into our model.

4. A Logic-based Benders Decomposition Solution Scheme

In this section, we first propose an LBBD solution procedure for the data-driven stochastic VRP with deadlines in Section 4.1. Section 4.2 presents our proposed acceleration strategies. In Section 4.3, we develop an ML-based lower bound heuristic.

4.1. A Logic-based Benders Decomposition Algorithm

Although some ML models can be reformulated as convex representations under some assumptions, it is hard to extend them to more general problems and this poses a major challenge in solving larger and more realistic instances. For example, Liu et al. (2021) used tractable feature vectors and linear ML models, which are compatible with their order-assignment optimization, to predict travel time. Grimstad and Andersson (2019) and Anderson et al. (2020) assumed that the nonlinear operator in deep neural networks is a rectified linear unit. They formulated the deep neural networks as a 0-1 mixed integer linear program with the big-M method. When it comes to larger and more realistic deep neural networks, the computing time can become too large.

To overcome these issues, in this section, we use an LBBD to solve our proposed VRP with the features proposed in Section 3.2. One advantage of the LBBD algorithm is that it allows us to solve the data-driven VRP without any assumptions about the problem structure. Moreover, as indicated by our numerical study, using the strategies proposed later in this paper can greatly improve the performance of the LBBD algorithm. In particular, we decompose the problem as a master problem and a set of subproblems. In the master problem, we determine the routes without considering the expected delay, i.e., the master problem is a relaxation of the data-driven VRP where the expected delay is ignored. Hence, the master problem is a combinatorial optimization problem. For each route obtained from the master problem, we compute the probabilities based on the nonparametric ML methods proposed in Section 3.2 and the expected delay in the subproblems.

Since the subproblems are always feasible, when we obtain a feasible integer solution from the master problem, we only generate optimality cuts for each subproblem and add them to the master problem. Unlike the traditional LBBD that solves the master problem and a set of subproblem iteratively and introduce optimality cuts in the master problem, we add the optimality cuts in a branch-and-cut framework, then the master problem is solved only once.

The master problem is defined as follows:

$$\text{(MP)} \qquad \underset{(\boldsymbol{x},\boldsymbol{s})\in\mathcal{S},\boldsymbol{\eta}\geq\mathbf{0}}{\text{minimize}} \sum_{\ell\in\mathcal{N}_d} \eta_\ell. \tag{6a}$$

Let $(\bar{\boldsymbol{x}},\bar{\boldsymbol{s}})$ be a feasible solution of (MP). LBBD relies on adding valid optimality cuts for each η_ℓ that will be added to the MP. While one could simply employ the simple optimality cut:

$$\eta^\ell \geq obj^\ell - obj^\ell \sum_{a:\bar{s}^\ell_a=1} (1-s^\ell_a), \quad \forall \ell\in\mathcal{N}_d, \tag{7}$$

where $obj^\ell := \sum_{i=1}^{N(\bar{\boldsymbol{s}}^\ell)} p_i(\bar{\boldsymbol{s}}^\ell)\left(\xi^i(\bar{\boldsymbol{s}}^\ell)-\tau\right)^+$, one can obtain richer optimality cuts by assuming that the travel time distribution of a route is independent of which vehicle is servicing it. To formalize this assumption, we first define for each vehicle $\ell\in\mathcal{N}_d$, the set of arcs of the route that it follows, i.e.:

$$\mathcal{R}(\bar{\boldsymbol{s}}^\ell) := \left\{(0,n^\ell_{(1)}),\ldots,(n^\ell_{(v_\ell-1)},n^\ell_{(v_\ell)})\right\},$$

where $n^\ell_{(i)}\in\mathcal{N}_c$ is the ith customer location in the route, and v_ℓ is the number of customer locations serviced by the route.

Assumption 1. *[Vehicle insensitive travel time] For all $\ell_1, \ell_2 \in \mathcal{N}_d$ and all $(\boldsymbol{x}_1, \boldsymbol{s}_1), (\boldsymbol{x}_2, \boldsymbol{s}_2) \in \mathcal{S}$:*

$$\mathcal{R}(\bar{\boldsymbol{s}}_1^{\ell_1}) = \mathcal{R}(\bar{\boldsymbol{s}}_2^{\ell_2}) \Rightarrow \mathbb{P}_{(\bar{\boldsymbol{s}}_1^{\ell_1})} = \mathbb{P}_{(\bar{\boldsymbol{s}}_2^{\ell_2})}.$$

Assumption 1 is a natural one to employ when the fleet of vehicles is homogeneous and historical data cannot be used to infer the route-dependent travel time distribution of each specific driver.

Assumption 1 allows to replace the optimality cuts described in (7) with:

$$\eta^j \geq obj^\ell - obj^\ell \sum_{a \in \mathcal{A}^j(\bar{\boldsymbol{s}}^\ell)} (1 - s_a^j), \quad \forall j, \ell \in \mathcal{N}_d, \tag{8}$$

where

$$\mathcal{A}^j(\bar{\boldsymbol{s}}^\ell) = \left\{(0, n_{(1)}^\ell), \ldots, (n_{(v_\ell)}^\ell, j)\right\}.$$

Optimality cuts (8) indicate that if $s_a^j = 1$ for all $a \in \mathcal{A}^j(\bar{\boldsymbol{s}}^\ell)$, then the expected delay is at least obj^ℓ, while otherwise at least 0. This cut is valid since the expected delay is always non-negative, and according to Assumption 1 is equal to obj^ℓ when its route is the same at the route of $\bar{\boldsymbol{s}}^\ell$.

Algorithm 1 in Appendix A.1 describes the detailed procedure of the LBBD algorithm. It is well-known that Algorithm 1 terminates in finitely many iterations as it is based on branching on a finite number of binary variables. In Algorithm 1, a new feasible solution obtained from the LP relaxation of (MP) always has zero objective value, hence the LB is equal to zero except that the algorithm could enumerate all the possible feasible routes or obtain a very small UB value. Therefore, the LBBD algorithm generally yields a large gap, and the computational cost might be prohibitive if the historical travel times are significantly larger than the service deadline τ.

4.2. Acceleration Strategies

In Section 4.2.1, we provide a class of general optimality cuts extended from the traditional optimality cuts (8) for the LBBD algorithm. We present a warm starting strategy in Section 4.2.2, and local search procedures in Section 4.2.3.

4.2.1. A General Optimality Cut Optimality cuts (8) can be weak, which are useful only when $s_a^j = 1$ for all $a \in \mathcal{A}^j(\bar{\boldsymbol{s}}^\ell)$ and $j \in \mathcal{N}_d$, i.e., it only includes the expected delay for one route. Moreover, in the worst case, it needs to enumerate all possible feasible routes to get an optimal solution. Thus requiring long computing time, especially for large-scale instances. Herein, we extend cuts (8) to new general optimality cuts, which contain cuts (8) and the expected delay for all the subroutes. Thus, within a tight time limit, (MP) could include more information about the routes by adding the general cuts when compared with cuts (8). In particular, for the general optimality cuts, we start with the first customer location visited by a vehicle, then we sequentially add other locations to the cuts such that any subroute has the value of expected delay.

Given a feasible route $\bar{\boldsymbol{s}}^\ell$ obtained from (MP), for $i = 1, \dots, v_\ell$, let the expected delay for the partial route $\left\{(0, n^\ell_{(1)}), \dots, (n^\ell_{(i-1)}, n^\ell_{(i)})\right\}$ be denoted by $obj^\ell_{(i)}$, where $n^\ell_{(0)}$ is node 0. Then, the general optimality cuts take the following form:

$$\begin{aligned}\eta^j \geq\ & obj^\ell_{(1)} s^j_{(0,n_{(1)})} - M \sum_{i\in\mathcal{N}_c\setminus\{n_{(1)}\}} s^j_{(0,i)} + (obj^\ell_{(2)} - obj^\ell_{(1)}) s^j_{(n_{(1)},n_{(2)})} - M \sum_{i\in\mathcal{N}_c\setminus\{n_{(2)}\}} s^j_{(n_{(1)},i)} + \\ & \dots \\ & + (obj^\ell_{(v_\ell)} - obj^\ell_{(v_\ell-1)}) s^j_{(n_{(v_\ell-1)},n_{(v_\ell)})} - M \sum_{i\in\mathcal{N}_c\setminus\{n_{(v_\ell)}\}} s^j_{(n_{(v_\ell-1)},i)} - M \sum_{i\in\mathcal{N}_c} s^j_{(v_\ell,i)}, \qquad \forall j \in \mathcal{N}_d,\end{aligned} \tag{9}$$

where M is a positive constant guaranteeing that the general cuts are inactive when the vehicle does not visit these subroutes. We can simply set M equal to the sum of positive coefficients in the cut. The following proposition gives the validity of the general optimality cuts (9).

PROPOSITION 1. *Given Assumption 1, the general optimality cuts* (9) *are valid for the data-driven VRP* (5).

4.2.2. Warm Start LBBD algorithms are well-known to suffer from ineffective initial iterations due to low-quality solutions generated at the beginning of the algorithm. In this section, we use a warm starting procedure to generate a good initial solution for (MP). The underlying idea of the warm starting procedure is to use the hybrid genetic search proposed by Vidal et al. (2012) to solve a total travel cost minimization problem, then a feasible solution of the problem with the smallest expected delay is used to warm start the LBBD algorithm. Specifically, the total travel cost minimization problem is

$$\underset{(\boldsymbol{x},\boldsymbol{s})\in\mathcal{S}}{\text{minimize}} \sum_{a\in\mathcal{A}} c_a x_a, \tag{10a}$$

where c_a is transportation cost of arc $a \in \mathcal{A}$. In this paper, we take c_a as the distance associated with arc a. To equally use all the vehicles, one can limit the number of customer locations visited by one vehicle to no more than $\left\lceil \frac{I}{L} \right\rceil + 1$.

The hybrid genetic search proposed by Vidal et al. (2012) can be used to solve problem (10). Vidal et al. (2012) proposed an algorithm framework to address the capacitated VRP with constrained route duration and the goal is to minimize the total travel cost. The general scheme of the heuristic includes a population of individuals with feasible and infeasible subpopulations, then selects two parent individuals to produce a new individual, which is enhanced by using local search procedures. It inserts the new enhanced individual into the appropriate subpopulation in relation to its feasibility. In order to use the hybrid genetic search method, we change the constrained route duration to address the number of customer location limit and obtain a population of individuals with feasible subpopulation $\{\boldsymbol{x}^m, \boldsymbol{s}^m\}_{m=1,\ldots,M'}$. Then we compute the expected delay obj^m of $(\boldsymbol{x}^m, \boldsymbol{s}^m)$ for $m = 1, \ldots, M'$, and let $m^* = \arg \min_{m=1,\ldots,M'} obj^m$. Finally, we reorder the vehicle indices for $(\boldsymbol{x}^{m^*}, \boldsymbol{s}^{m^*})$ to satisfy the symmetry-breaking constraints (2), and let $\hat{\eta} = obj^{m^*}$. Thus, $(\boldsymbol{x}^{m^*}, \boldsymbol{s}^{m^*}, \hat{\eta})$ is a feasible solution of problem (5) and can be used as a warm start for the LBBD algorithm. Algorithm 2 in Appendix A.2 outlines the warm starting procedure. As stated in the case study, solving the hybrid genetic search algorithm with the objective of minimizing the travel cost is computed within seconds. Thus, it doesn't increase the computational burden.

4.2.3. Local Search To speed up the computation, particularly for large-scale instances, we introduce local search procedures into our LBBD algorithm. The local search procedures have been used in many heuristic frameworks, such as the hybrid genetic search (Vidal et al. 2012) and the variable neighborhood search. In Section 5, we numerically show that the LBBD algorithm can be significantly improved by incorporating the local search procedures and outperforms a heuristic based on the hybrid genetic search with the local search procedures.

The local search procedures include relocations, which insert one or two consecutive customers to another position; swaps, which exchange one or two consecutive customers with another position; 2-opt, an intraroute move where a selected sequence of customers is reversed; and 2-opt*, where the end parts of two different routes are interchanged, or the start and end parts of two different routes are connected, respectively. The local search phase stops when all possible procedures have been successively tried and failed to produce any improvement. For further details about the local search procedures, interested readers are referred to Vidal et al. (2012) and Vidal (2022).

To incorporate the local search procedures, we start from an integer solution $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{s}})$ obtained from (MP), and calculate the corresponding expected delay $o\bar{b}j$. $\bar{\boldsymbol{s}}$ is then improved by the local search procedures. If a new solution $\bar{\boldsymbol{s}}'$ is better than the incumbent $\bar{\boldsymbol{s}}$ in terms of the expected delay, then we construct $\bar{\boldsymbol{x}}'$ based on $\bar{\boldsymbol{s}}'$, after reordering the vehicle indices to satisfy the symmetry-breaking constraints, we set the new solution $(\bar{\boldsymbol{x}}', \bar{\boldsymbol{s}}')$ as the incumbent solution, and update the UB; otherwise, it continues with the next iteration. The incorporation of local search procedures is formalized as Algorithm 3 in Appendix A.3.

4.3. Lower Bound Heuristic

It is well-known that the capacitated VRP is NP-hard. As a generalization of capacitated VRP, our data-driven stochastic VRP (5) is also NP-hard. Although the numerical results show that the proposed acceleration strategies could effectively reduce the absolute gap, large-sized instances are still difficult to solve. Moreover, the lower bound obtained from the LBBD algorithm is usually equal to zero within a tight time limit. Thus, we introduce a novel lower bound heuristic to reduce the computational cost in proving optimality.

We first show how to use an $\boldsymbol{s}$ dependent distribution $\bar{\mathbb{P}}_{(\boldsymbol{s})}$, which is first-order stochastically dominated by the distribution under $\mathbb{P}_{(\boldsymbol{s})}$, for all $\boldsymbol{s}$, to obtain an equivalent formulation of problem (3) in Lemma 1.

LEMMA 1. *For all* $(\boldsymbol{x},\boldsymbol{s})\in\mathcal{S}$, *let the distribution of* $\tilde{\xi}$ *under some* $\bar{\mathbb{P}}_{(\cdot)}$ *be uniformly first-order stochastically dominated by its distribution under* $\mathbb{P}_{(\cdot)}$, *i.e.*

$$\bar{\mathbb{P}}_{(\boldsymbol{s}^\ell)}(\tilde{\xi}\le z)\ge\mathbb{P}_{(\boldsymbol{s}^\ell)}(\tilde{\xi}\le z)\,,\ \forall z\in\mathbb{R},\ \forall\ell\in\mathcal{N}_d,\ \forall(\boldsymbol{x},\boldsymbol{s})\in\mathcal{S},$$

then problem (3) *is equivalent to the following problem:*

$$\begin{aligned}
&\underset{(\boldsymbol{x},\boldsymbol{s})\in\mathcal{S},\boldsymbol{\eta}}{\text{minimize}}\ \sum_{\ell\in\mathcal{N}_d}\eta_\ell && && \text{(11a)}\\
&\text{subject to } \eta_\ell\ge\mathbb{E}_{\mathbb{P}_{(\boldsymbol{s}^\ell)}}\left[\left(\tilde{\xi}-\tau\right)^+\right], && \forall\ell\in\mathcal{N}_d, && \text{(11b)}\\
&\qquad\qquad\ \eta_\ell\ge\mathbb{E}_{\bar{\mathbb{P}}_{(\boldsymbol{s}^\ell)}}\left[\left(\tilde{\xi}-\tau\right)^+\right], && \forall\ell\in\mathcal{N}_d. && \text{(11c)}
\end{aligned}$$

To model $\bar{\mathbb{P}}_{(\cdot)}$, we can use a discrete distribution model $\bar{\mathbb{P}}_{(\boldsymbol{s}^\ell)}(\tilde{\xi}\le y):=(1/(T+1))\mathbf{1}\{0\le y\}+(1/(T+1))\sum_{t=1}^{T}\mathbf{1}\{\sum_{a\in\mathcal{A}}\theta_a^t s_a^\ell+\bar{\theta}^t\le y\}$, for some $\theta_a^t,\bar{\theta}^t\in\mathbb{R}$ and $T\in\mathbb{Z}_+$. Here $\mathbf{1}\{\cdot\}$ is an indicator function, which returns 1 if the expression in $\{\cdot\}$ is true. Then constraints (11c) can be expressed as follows:

$$\eta_\ell\ge(1/(T+1))\sum_{t=1}^{T}(\sum_{a\in\mathcal{A}}\theta_a^t s_a^\ell+\bar{\theta}^t-\tau)^+,\qquad\forall\ell\in\mathcal{N}_d.\qquad(12)$$

Motivated by conformal prediction, which uses past experience to determine precise levels of confidence in new predictions (Shafer and Vovk 2008), we assemble a data set $\{(\hat{\boldsymbol{s}}^j,\boldsymbol{q}_j)\}_{j=1}^J$ where each $\boldsymbol{q}_j\in\mathbb{R}^T$ such that $q_j^t:=Q_{\mathbb{P}_{(\hat{\boldsymbol{s}}_j)}}^{t/(T+1)}(\tilde{\xi})$, and $Q_{\mathbb{P}}^{\alpha}(Y)$ is the α quantile of Y under the distribution $\mathbb{P}$. One then solves the following linear program:

$$\begin{aligned}
&\underset{\boldsymbol{\theta}\ge0}{\text{maximize}}\ \sum_{t=1}^{T}\sum_{j=1}^{J}(\sum_{a\in\mathcal{A}}\theta_a^t\hat{s}_a^j+\bar{\theta}^t) && && \text{(13a)}\\
&\text{subject to } q_j^t\ge\sum_{a\in\mathcal{A}}\theta_a^t\hat{s}_a^j+\bar{\theta}^t, && \forall t=1,\ldots,T,\ j=1,\ldots,J, && \text{(13b)}\\
&\qquad\qquad\ \boldsymbol{\theta}^{t+1}\ge\boldsymbol{\theta}^t, && \forall t=1,\ldots,T-1. && \text{(13c)}
\end{aligned}$$

To evaluate $\boldsymbol{\theta}^t$ and $\bar{\theta}^t$ obtained from problem (13) for $t = 1, \ldots, T$, we randomly generate a set of routes $\{\hat{\boldsymbol{s}}'^j\}_{j=1,\ldots,J'}$ and compute the following probability that the expected delay obtained by the finite scenario model is smaller than the one from the distribution $\mathbb{P}_{(\hat{\boldsymbol{s}}'^j)}$:

$$\frac{\sum_{j=1}^{J'} \mathbf{1}\left\{(1/(T+1))\sum_{t=1}^{T}(\sum_{a\in\mathcal{A}}\theta_a^t \hat{s}_a'^j + \bar{\theta}^t - \tau)^+ \leq \mathbb{E}_{\mathbb{P}_{(\hat{\boldsymbol{s}}'^j)}}\left[\left(\tilde{\xi}-\tau\right)^+\right]\right\}}{J'},$$

The probability might be small, to correct the value of $\boldsymbol{\theta}^t$ and $\bar{\theta}^t$, we can replace constraints (12) with the following constraints by using a hyperparameter $\rho \in \mathbb{R}_+$:

$$\eta_\ell \geq \rho(1/(T+1))\sum_{t=1}^{T}(\sum_{a\in\mathcal{A}}\theta_a^t s_a^\ell + \bar{\theta}^t - \tau)^+, \quad \forall \ell \in \mathcal{N}_d. \tag{14}$$

Here ρ is used to reduce the likelihood of introducing incorrect cuts (12). This could occur when the expected delay obtained from $\boldsymbol{\theta}$ is larger than the exact one with a high probability.

Given a threshold ϵ', if the probability is less than ϵ', we decrease ρ, otherwise, we increase it. We can use a bisection method to find the proper ρ such that the probability can meet the threshold. More specifically, the bisection method is given in Algorithm 4 in Appendix A.4.

In order to use the LBBD algorithm with the acceleration strategies to solve problem (11), we first add constraints (14) to the master problem (6), leading to a new master problem of the following form:

$$\underset{(\boldsymbol{x},\boldsymbol{s})\in\mathcal{S},\boldsymbol{\eta}}{\text{minimize}} \sum_{\ell\in\mathcal{N}_d} \eta_\ell \tag{15a}$$

$$\text{subject to (14).} \tag{15b}$$

As indicated by our numerical study, when τ is small, the lower bound of the LBBD algorithm can be positive. The warm starting procedure proposed in Section 4.2.2 can also provide an initial solution to the master problem (15). If a solution obtained from problem (15) is integer, we apply the local search procedure to improve the solution and set the new solution as an incumbent solution if the new solution has a smaller expected delay. Then, we add the general optimality cuts based on the incumbent solution to the master problem (15). Otherwise, we continue branching. The algorithm stops when the node list is empty, or the lower bound is larger than or close to the upper bound. Algorithm 5 in Appendix A.5 describes the procedure of our implementation of the LBBD algorithm with the acceleration strategies and the lower bound heuristic.

5. A Real-World Food Delivery Case Study

In Section 5.1, we describe the real-world food delivery data set. Section 5.2 presents a Hoeffding's D independence test. In Section 5.3, we show how to estimate the ground-truth distribution of the delivery travel time for the routes, which is done by designing a hypothesis test. Section 5.4 describes the procedures for training the ML models used in this paper. In Section 5.5, we present the computational performance of our LBBD algorithms with enhancements. Finally, we provide the out-of-sample performance of our data-driven VRP solutions with benchmark models and policies in Section 5.6.

5.1. Data Description

We use real-world data from a food delivery service provider in Shanghai, China, to show the advantage of the data-driven VRP model and the performance of the proposed algorithm. The data set has 839 customer locations and 4076 routes for two months in 2015. Specifically, the data set records the longitude and latitude, the number of items ordered for the customer locations, the actual travel time, the travel sequences from the depot to the last visited customer location, and a batch identifier for the routes, where all orders with the same service deadline are referred to as one batch. Thus, a batch represents an instance, where the optimal vehicle routes are determined to fulfill the demands of customer locations with the same service deadline.We split the real data as a training set and a test set, of which 3268 routes in the training set and the reminding 808 routes serve as the test set (i.e., 80% training data and 20% test data). See Liu et al. (2021) for more details about the data set.

5.2. Hypothesis Test to Validate on Endogenous Travel Time

In order to validate that arcs's travel times are dependent on the choice of route, we apply a Hoeffding's D independence test (see Hoeffding (1994)) between arc travel time and length of the route on arcs with more than 10 visits (i.e. part of more than 10 batches) in our dataset. More specifically, for each arc a among these arcs, we gather the dataset $\{t_i^a, \bar{d}_i)\}_{i\in\mathcal{N}(a)}$, where t_i^a is the travel time of arc a in the route i, $\bar{d}_i$ is the length of the route i, and $\mathcal{N}(a)$ is the set of routes containing the arc a. The Hoeffding's D independence test compares the following hypotheses

- **Null hypothesis**: "The travel time on the arc is independent of the length of route."
- **Alternative hypothesis**: "The travel time on the arc is dependent on the length of route."

We set the significance threshold of the test to 5%. Among the 173 arcs with more than 10 visits, we observed that the null hypothesis is rejected for 75 of them (43% of arcs with more than 10 visits). Focusing on the arcs that have more than 60 visits, we observed a rejection rate of nearly 93% (14 out of 15 arcs). We conclude from this evidence that provided enough data, one can confirm that arc travel time is dependent on the route length for most arcs of the network in this case study.

5.3. Ground-Truth Distribution Estimation

To assess the out-of-sample performance of implementing the solutions obtained from the data-driven VRP, we need to estimate the ground-truth distribution of the delivery travel time for the routes proposed by the different approaches. Then, the delivery travel time is sampled from the estimated ground-truth distribution to evaluate the out-of-sample delivery delays. In particular, we use CGMM as our ground-truth distribution, which could capture any distribution as long as the number of Gaussians in the mixture is large enough (Gilardi et al. 2002).

To verify the statistical quality of an estimated distribution of the delivery travel time for the routes, we design a hypothesis test that verifies the null hypothesis. Specifically, let $\{x_i, y_i\}_{i=1}^N$ be the feature and total travel time for each observed route, which are sampled from the ground-truth distribution $\mathbb{P}$ on the outcome space $(\mathcal{X} \times \mathbb{R})$. The role of a conditional distribution estimator $\hat{F}_{Y|X}$ such that $\hat{F}_{Y|X}(y;x) \approx \mathbb{P}(Y \leq y|X=x)$ for all $y \in \mathbb{R}$ almost surely when $x \sim \mathbb{P}$. Then, the null hypothesis and alternative hypothesis are defined as

- **Null hypothesis**: "$\mathbb{P}(Y \leq y|X=x) = \hat{F}_{Y|X}(y;x)$"
- **Alternative hypothesis**: "$\mathbb{P}(Y \leq y|X=x) \neq \hat{F}_{Y|X}(y;x)$"

To test the null hypothesis, we first augment the data set with $Z := 1 + \sum_{k=1}^K \mathbf{1}\{Y > \hat{Y}_k(X)\}$, where each $\hat{Y}_k(X)$ is drawn i.i.d. from $\hat{F}_{Y|X}(y;X)$, for some $K>1$. The test statistic, parameterized by $K>1$, takes the form of:

$$\chi_K^2 := \sum_{k=1}^{K+1} \frac{(\sum_{i=1}^N \mathbf{1}\{z_i = k\})^2}{N/(K+1)} - N\,,$$

where $z_i := 1 + \sum_{k=1}^K \mathbf{1}\{y_i > \hat{Y}_k(x_i)\}$ for all $i = 1, \ldots, N$.

Lemma 2. *Assuming that $\hat{F}_{Y|X}(y;x)$ is continuous for all x, then under the null hypothesis, we have that χ_K^2 converges to a chi-square distribution with K degrees of freedom.*

Lemma 2 gives rise to the following statistical testing procedure:

- For all $i = 1, \ldots, N$, let $z_i := 1 + \sum_{k=1}^K \mathbf{1}\{y_i > \hat{Y}_k(x_i)\}$.
- Calculate the sample statistic $\chi_K^2 := \sum_{k=1}^{K+1} \frac{(\sum_{i=1}^N \mathbf{1}\{z_i = k\})^2}{N/(K+1)} - N$.
- Compute the p-value of the test statistic using an inverse χ^2 distribution with K degrees of freedom.
- If p-value is below the desired confidence level then reject the null hypothesis.

p-value can be used to compare the goodness-of-fit of the different proposals for $\hat{F}_{Y|X}$.

CGMM is trained using the training data, and evaluated in the test set. To train CGMM, we first divide our data set into training (3268 points), validation (3268 points), and test set (808 points). Here, the training and validation sets use the same dataset. We set the number of

Gaussians in the mixture from 5 to 20, and the maximum number of iterations to 2,000. The mixture distribution parameters are output by a neural network and trained to maximize overall log-likelihood. The whole training procedure is implemented in PyTorch. In addition, at the training phase, for each number of mixtures and each iteration, we train the model using the training set. Then we generate $K = 1633$ samples for each data in the validation set from the current model such that $N_{\text{valid}}/(K+1) = 2$ and compute the overall p-value, where N_{valid} is the number of data in the validation set. At the end of the training procedure, we use the one with a minimum p-value as our generative model. To evaluate the generative model, we generate $K = 1633$ samples for each data in the test set and compute the p-value, which is 0.26. So the hyper-parameters are chosen from the training and validation sets and evaluated using the test set. If we choose a significance level of 5%, then we cannot reject the null hypothesis for CGMM, that is, we can accept the hypothesis that CGMM is the same as the ground-truth distribution. Therefore, in the remainder of this section, we use the CGMM as the ground-truth model to demonstrate the computational efficiency and evaluate the out-of-sample performance.

5.4. Model Training

All the ML models in this paper are also trained using the training data. We use tenfold cross-validation to select the best hyper-parameters, i.e., the k for kNN and bandwidth for KDE, and the shrinkage parameters in the least absolute shrinkage and selection operator regression (LASSO) used in Liu et al. (2021). The training and validation procedures are implemented in Python for kNN and KDE, and in R for LASSO. Specifically, we randomly split the training dataset into ten equal-sized groups. One of the groups is used as the validation set for testing the model, and the remaining groups are used as the training set. The cross-validation process is repeated ten times, at each time, the model is trained on the training set and scored on the validation set by computing the mean squared error (MSE). At the end of the process, the hyper-parameter with the minimum average cross-validation MSE over 10 times is used. The average square root of the cross-validation MSE for the kNN and KDE methods are equal to 9.4 and 9.6, respectively. compared with the KDE method, the kNN method achieves lower error. Therefore, in the remainder of this section, the kNN method will be among the models used by the optimization model.

5.5. Computational Performance

We study the computational performance of our proposed algorithm. We start in Section 5.5.1 by presenting some details regarding the implementation of the LBBD algorithm, followed with a description of how the lower bound heuristic was calibrated in Section 5.5.2. Section 5.5.3 presents the computational results under the endogenous travel time model with kNN method identified in Section 5.4, while the computational performance for KDE method is presented in Appendix C.

5.5.1. Implementation Details: To evaluate the numerical performance of using the LBBD algorithm with the proposed acceleration strategies and the lower bound heuristic to solve the data-driven VRP, in our test examples, we vary the service deadline τ from 25 minutes to 65 minutes, which could adjust different food preparation time. To set a reasonable number of vehicles and fairly compare the performance of different variants, following Liu et al. (2021), we let the maximum capacity $Q := 30$, and the number of customer locations visited by one vehicle is no more than 8. The number of drivers L used for each batch is set to be $\min\{I_c * 1.5, I_c + 4\}$, where I_c is the minimum number of drivers needed according to the capacity restriction and the number of customer locations limit. Based on the training data from the real data, ten new training dataset were generated by using CGMM for the VRP to obtain statistics about the performance of the different algorithms. The kNN and KDE methods are trained using the training data. Then, the decisions from the data-driven VRP are derived and evaluated for each batch in the test set. For the test data, only the batches that contain more than 10 customer locations are used, so there are 44 batches in total, where the number of customer locations varies from 10 to 58.

All experiments are conducted on the Cedar cluster of the Digital Alliance of Canada with a single CPU core and 32G memory. The algorithm was implemented with the C++ programming language using IBM's CPLEX solver, version 12.10. For all computations, we set a maximum runtime of 20 minutes for all algorithms, which corresponds to 20 minutes for practical food preparation. The CPLEX default settings are used for all algorithms, and we apply method LBBD—with optimality cuts added as lazy constraints, we set the number of threads as one. The warm start and the $\boldsymbol{\theta}$ in the lower bound heuristic are generally computed within seconds, so we did not present these runtimes in the tables.

5.5.2. Computational Results on the Lower Bound Heuristic: In this section, we provide details on training $\boldsymbol{\theta}$, and choosing ρ for the lower bound heuristic with kNN and KDE methods, which are defined in Section 4.3. Given a set of customers $\mathcal{N}_c = \{1, \ldots, I\}$, considering the computational cost, we let $J = I \times I$, and the set of routes for training $\boldsymbol{\theta}$ be $\{\{0, i\}, \{0, i, j\}\}_{i,j \in \mathcal{N}_c, i \neq j}$. We let $T = 9$ and calculate the 10th, 20th, $\ldots$, and 90th quantiles based on the discrete distribution constructed by the kNN and KDE methods. We then solve problem (13) to obtain an optimal solution. To choose the value of ρ, we set the threshold $\varepsilon' = 0.85$ and randomly generate 8,000 routes with different numbers of customer locations. Then we use the bisection Algorithm 3 to choose ρ. Table 1 presents the lower bound probability, and the average absolute gap (in minutes) when the lower bound value is larger than the expected delay for the kNN method based on a newly generated set of routes with different numbers of locations.

From Table 1 we can see that the out-of-sample lower bound probability is around the threshold ε', and the average absolute gap is small. Although parameter $\boldsymbol{\theta}$ is trained with at most two

Table 1 The lower bound probability and the average absolute gap (in minutes) are reported.

τ	25	35	45	55	65
LBProb	0.85	0.85	0.85	0.85	0.85
Avg_AGap	0.3	0.22	0.16	0.11	0.06

customer locations, we use the different numbers of locations to choose ρ and evaluate the heuristic. The results in Table 1 indicate that the lower bound value could meet the threshold value. When the lower bound value is larger than the expected delay, this gap is less than one minute on average. Therefore, for the data-driven VRP with the kNN method, we conclude that the lower bound heuristic could provide a valid lower bound value. Similar observations can be found for the KDE method (see Appendix C.1).

5.5.3. Computational Results under kNN Model: We now present our results on the acceleration strategies and lower bound improvement heuristic implemented in the LBBD algorithm 5 for the data-driven VRP with the kNN method. In this section, we consider the service deadline $\tau \in \{25, 35, 45, 55, 65\}$.We illustrate the performance of the following six different variants:

- BDWLG_LH: refers to using the default LBBD algorithm with the warm starting, local search procedures, the general optimality cuts, and the lower bound improvement heuristic on the data-driven VRP model (5).
- BDWLG: refers to using the default LBBD algorithm with the warm starting, local search procedures, and the general optimality cuts.
- BDWL: refers to using the default LBBD algorithm with the warm starting and local search procedures.
- BDW: refers to using the default LBBD algorithm only with the warm starting procedure.
- BD: refers to using the default LBBD method with neither acceleration strategies nor the lower bound improvement heuristic.
- Genetic: refers to using a genetic heuristic, where the objective function of the total travel cost minimization in the hybrid genetic search heuristic (Vidal et al. 2012) is adjusted to the data-driven VRP model (5).

Table 2 displays the average upper bound and estimated lower bound, the average absolute gap, the solution time for the instances that can be solved to optimality, the average time spent for the subproblem and local search procedures, the average number of cuts and nodes, and the optimal ratio for the kNN method, where the optimal ratio is computed as the number of solved instances over the total 440 instances. The results for the KDE method are presented in Appendix C.2.

From Table 2 we can observe that BDWLG_LH achieves the smallest average absolute gap and largest optimal ratio among all the six methods for smaller service deadlines, while BDWLG

Table 2 The average upper bound (UB) and estimated lower bound (Est_LB), the average absolute gap (Gap), the average solution time (in seconds) for the instances solved optimally (AvT), and subproblem and local search procedures (AvT_SP), the average number of cuts (# of cuts) and nodes (# of nodes), and the optimal ratio (Ratio) are reported.

τ	Method	UB	Est_LB	Gap	AvT	AvT_SP	# of cuts	# of nodes	Ratio
25	BDWLG_LH	48.67	35.58	13.09	129.8	147.7	318	70,664	0.17
	BDWLG	47.67	0.00	47.67	N/A	675.5	1,716	2,667	0.00
	BDWL	47.71	0.00	47.71	N/A	695.1	1,727	2,270	0.00
	BDW	67.76	0.00	67.76	N/A	10.3	24,972	19,496	0.00
	BD	101.29	0.00	101.29	N/A	10.3	24,163	18,185	0.00
	Genetic	84.78	0.00	84.78	N/A	N/A	N/A	N/A	0.00
35	BDWLG_LH	18.70	12.58	6.12	156.3	122.1	277	74,460	0.21
	BDWLG	17.84	0.00	17.84	N/A	648.3	1,904	3,348	0.00
	BDWL	17.96	0.00	17.96	N/A	680.8	1,977	2,968	0.00
	BDW	32.60	0.00	32.60	N/A	9.4	22,598	20,848	0.00
	BD	61.74	0.00	61.74	N/A	9.8	23,096	19,278	0.00
	Genetic	46.00	0.00	46.00	N/A	N/A	N/A	N/A	0.00
45	BDWLG_LH	5.35	3.11	2.24	183.5	122.1	395	78,170	0.20
	BDWLG	5.05	0.00	5.05	2.7	595.5	2,012	3,900	0.02
	BDWL	5.05	0.00	5.05	5.9	636.1	2,138	3,616	0.02
	BDW	12.53	0.00	12.53	3.5	9.3	22,470	20,292	.005
	BD	33.29	0.00	33.29	8.2	9.9	22,962	19,346	.005
	Genetic	22.42	0.00	22.42	N/A	N/A	N/A	N/A	0.01
55	BDWLG_LH	1.00	0.44	0.57	152.1	97.7	430	48,045	0.22
	BDWLG	0.91	0.00	0.91	86.5	475.9	1,819	3,895	0.12
	BDWL	0.91	0.00	0.91	95.4	505.8	1,963	3,802	0.12
	BDW	3.68	0.00	3.68	43.3	8.1	19,379	20,553	0.03
	BD	15.97	0.00	15.97	3.2	8.7	20,373	18,565	0.03
	Genetic	8.55	0.00	8.55	N/A	N/A	N/A	N/A	0.05
65	BDWLG_LH	0.10	0.01	0.09	170.2	59.7	293	14,553	0.45
	BDWLG	0.09	0.00	0.09	111.6	221.3	974	2,516	0.49
	BDWL	0.09	0.00	0.09	107.5	238.2	1,050	2,603	0.48
	BDW	0.87	0.00	0.87	15.2	5.8	12,695	16,685	0.11
	BD	6.68	0.00	6.68	25.2	6.7	15,088	15,153	0.03
	Genetic	2.63	0.00	2.6	N/A	N/A	N/A	N/A	0.22

has a better average upper bound. Only when the service deadline $\tau = 65$, BDWLG_LH gives a comparable performance to BDWLG and BDWL in terms of the average absolute gap and optimal ratio. In particular, when $\tau = 55$, compared with BDWLG, BDWLG_LH reduces the average absolute gap by more than 38%, and improves the optimal ratio by about 46%. However, the average upper bound obtained by BDWLG_LH is increased modestly. This is mainly because BDWLG_LH has a larger problem size than BDWLG, BDWLG_LH could explore a larger number of nodes and return fewer integer solutions. This translates to a larger average upper bound. Moreover, when $\tau = 25$, a significantly smaller average absolute gap is obtained from BDWLG_LH, although it is still larger than 10. In such cases, the average estimated lower bound of the expected delay is larger than 35 minutes, possibly not meeting the decision-makers' objectives. Thus, one needs to increase the number of vehicles to achieve the desired goal. In addition, BDWL yields a slightly poor performance when compared with BDWLG. By using the general optimality cuts, either the average absolute gap or the solution time for the solved instances is reduced. Furthermore, BDWL has a significantly better performance than BDW, indicating that the local search procedures could provide improved integer solutions with reasonably good quality. Although it does require some time to implement the local search procedures. This increase in the local search time can be related to the reduction in the number of optimality cuts and nodes explored in the algorithm. Moreover, BDW significantly outperforms BD. This confirms that the acceleration strategies and the lower bound heuristic proposed in Section 4 are crucial to improving the efficiency of the LBBD algorithm. We also notice that when compared with the genetic heuristic, the LBBD algorithm with acceleration strategies delivers significant improvement in terms of the average absolute gap and the optimal ratio. Only BD gives a worse performance than the genetic heuristic.

Insight 1. *BDWLG has the best integer solutions for the data-driven VRP with the kNN method. In the meantime, BDWLG_LH could decrease the average absolute gap and improve the optimal ratio.*

5.6. Out-of-Sample Performance of the Data-Driven VRP Solutions

To evaluate the out-of-sample performance of the routes obtained from the data-driven VRP with the kNN and KDE methods, in this section, we use the real data as the training dataset, and generate $\{\bar{\xi}_\ell^1, \ldots, \bar{\xi}_\ell^{N'}\}_{N'=10{,}000}$ samples from the CGMM for the optimal route $\boldsymbol{s}^{*\ell}$. We then calculate the following out-of-sample expected delay for each batch:

$$\text{out-of-sample expected delay} := \frac{1}{N'} \sum_{i=1}^{N'} \sum_{\ell \in \mathcal{N}_d} \left(\bar{\xi}_\ell^i - \tau\right)^+$$

We first show the performance of the data-driven VRP with kNN and KDE methods, and an adjusted version of the assignment model proposed by Liu et al. (2021) in Section 5.6.1. We then compare the kNN method to the order-assignment decisions obtained from the DOA-DRO model used in Liu et al. (2021) in Section 5.6.2 and the implemented route decisions observed in the real data set in Section 5.6.3.

5.6.1. Comparisons with Benchmark Models In this section, we show the out-of-sample performance of the data-driven VRP with the kNN and KDE methods using the feature vector proposed in Section 3.2 and the following three methods:

- ALiu$_{\text{opt}}$: refers to adjusting the assignment model proposed by Liu et al. (2021) to our problem by letting the travel time include the service time, where the routing decisions are optimal (provided by an "oracle") given the assignment decisions. Please refer to the detailed formulations and solution schemes of ALiu$_{\text{opt}}$ in Appendix D.
- ALiu$_{\text{kNN}}$: refers to the use of ALiu$_{\text{opt}}$ but using the kNN method to optimize the routing decisions. See Appendix D for more details.
- kNNT: refers to the data-driven VRP with the kNN method using the tractable feature vectors used in Liu et al. (2021).

The experiment setting is analogous to the setting used in Section 5.5.1 with the service deadline $\tau \in \{45, 55, 65\}$. For ALiu$_{\text{opt}}$, it can be seen as the best version on the out-of-sample performance achieved by the assignment model based on Liu et al. (2021). Figure 1 presents the statistics of the out-of-sample expected delay for the kNN and KDE methods and the assignment models based on Liu et al. (2021) for three different service deadlines.

Figure 1 shows that the kNN method with the features used in this paper performs better than other methods for the service deadline $\tau = 55$ and 65. The kNN method significantly outperforms the KDE method. Meanwhile, when compared with Aliu$_{\text{kNN}}$, the median of the out-of-sample expected delay obtained by Aliu$_{\text{opt}}$, is reduced by an average of about 25.7%. Moreover, compared with Aliu$_{\text{opt}}$, the kNN method using the tractable features in Liu et al. (2021) improves the median of the out-of-sample expected delay by about 20.9% on average, the kNN method with the features used in this paper further reduces the median by 23.1%. This indicates that both the kNN method and the features used in this paper could yield an improvement in the out-of-sample performance over the assignment model based on Liu et al. (2021) for $\tau = 55$ and 65. For smaller service deadline ($\tau = 45$), we notice that kNN, kNNT, and Aliu$_{\text{opt}}$ have comparable out-of-sample performance, and better performance when compared with Aliu$_{\text{kNN}}$ and KDE.

Insight 2. *Using the tractable features proposed by Liu et al. (2021), our non-parametric model with routing decisions significantly outperforms the parametric assignment model Aliu$_{\text{opt}}$. This performance is further improved by using the features in this paper. Therefore, the kNN method with the features used in this paper appears to be the preferred method for such a data-driven problem.*

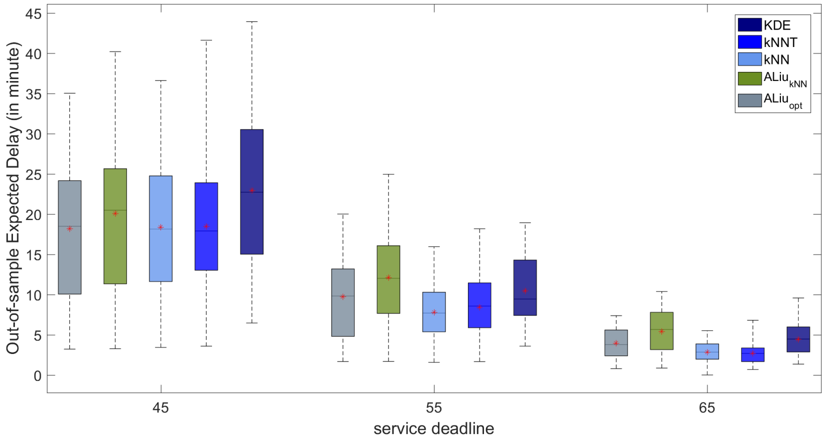

Figure 1 Statistics of the average out-of-sample delay for the assignment model based on Liu et al. (2021) with optimal routing decisions (ALiu$_{\text{opt}}$), and using kNN method to optimize the routing decisions (ALiu$_{\text{kNN}}$), the data-driven VRP with kNN (kNN) and KDE (KDE) using the features in this paper and the kNN model using the tractable features in Liu et al. (2021) (kNNT). The red star identifies the mean of the out-of-sample expected delay.

5.6.2. Comparisons with the Assignment Decisions from Liu et al. (2021) We compare the data-driven VRP with the kNN method using the features in this paper to the order-assignment decisions obtained from the DOA-DRO model proposed by Liu et al. (2021). For a fair comparison, the parameter setting is set to be the same as the setting used in Liu et al. (2021), and the kNN model is trained using the training dataset used in Liu et al. (2021). Since Liu et al. (2021) used a clustering algorithm to the customer locations in each instance based on geographical proximity such that the customer locations with slight differences in latitude and longitude are considered as one location. Given the assignment decisions from Liu et al. (2021), to obtain the routing decisions, we separate the clusters into the original customer locations. We let the number of vehicles equal to the one used in Liu et al. (2021). Figure 2 presents the statistics of the out-of-sample expected delay for the kNN method and the assignment decisions from Liu et al. (2021) for three different service deadlines.

We observe from Figure 2 that the data-driven VRP with the kNN method achieves better performance than the assignment decisions obtained from Liu et al. (2021). Specifically, when compared with Liu$_{\text{opt}}$, kNN reduces the median of out-of-sample expected delay by about 25.8% on average. For the maximal out-of-sample expected delay, a significant improvement is obtained by about 62.8% on average. From Figure 1 and 2, we see that the performance gap between kNN and ALiu$_{\text{opt}}$ is smaller than the gap between kNN and Liu$_{\text{opt}}$. This might be because a certain number of customer locations are considered as one customer location by using the clustering algorithm in Liu et al. (2021), then one driver might serve a large number of customer locations (we found that

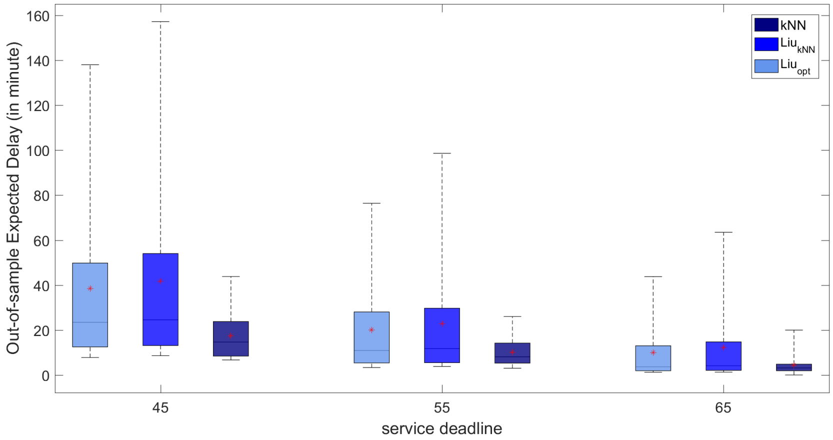

Figure 2 Statistics of the average out-of-sample delay for the assignment decisions obtained from Liu et al. (2021) with optimal routing decisions (Liu$_{\text{opt}}$), and using kNN method to optimize the routing decisions (Liu$_{\text{kNN}}$), the data-driven VRP with kNN using the features in this paper (kNN). The red star identifies the mean of the out-of-sample expected delay.

one driver could serve up to 16 customer locations). This might lead to a larger performance gap between kNN and Liu$_{\text{opt}}$.

Insight 3. *Compared to the order-assignment decisions obtained from DOA-DRO model proposed by Liu et al. (2021), our method yields 25.8% improvement in out-of-sample expected delay.*

5.6.3. Comparison with the Implemented Routing Decisions We compare our models to the routing decisions observed in the test set (i.e., the implemented routing decisions). We set the number of drivers to be the same as the actual number of dispatched drivers. Figure 3 presents the statistics of the out-of-sample expected delay for the implemented routes (Real), the implemented assignment decisions with the routes from the kNN method (RkNN), and the data-driven VRP with the kNN method (kNN) for three service deadlines. Since the implemented service deadline is 55 minutes, we vary the service deadline around the implemented one by letting $\tau \in \{45, 55, 65\}$.

From Figure 3 we can observe that compared to the actual routes chosen by drivers, RkNN improves the maximum value of the out-of-sample expected delay from 11.3 to 10.7 when $\tau = 65$, from 24.2 to 21.5 when $\tau = 55$, and from 42.7 to 38.8 when $\tau = 45$. A similar observation can be found for other statistics. Moreover, by using the data-driven VRP with the kNN method, the maximum value of the out-of-sample expected delay is significantly improved by about 62% on average, and about 16% for the mean of the out-of-sample expected delay when compared with other methods. While for the median, 25th, and minimum values of the average delay, the kNN method has a comparable performance to the other methods.

Insight 4. *Given the implemented assignment decisions, the routes from the kNN method have*

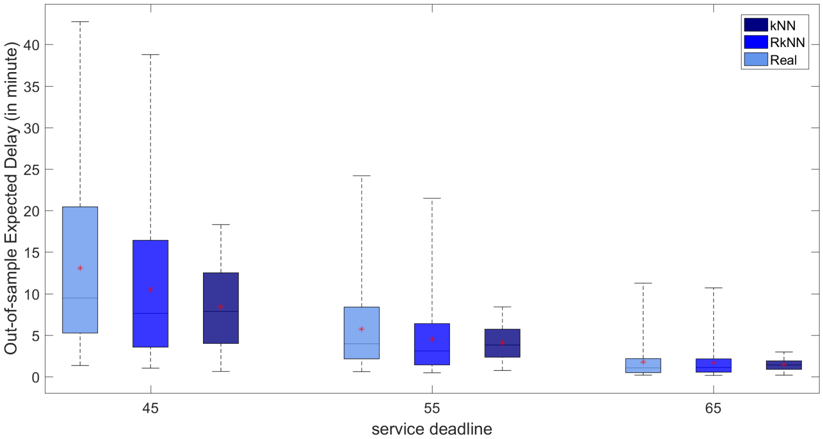

Figure 3 Statistics of the out-of-sample expected delay for the implemented routes (Real), the implemented assignment decision with the routes from the kNN methods (RkNN), and the data-driven VRP with the kNN method (kNN) for three service deadlines. The red star identifies the mean of the out-of-sample expected delay.

a better performance than the ones chosen by drivers. A more stable out-of-sample performance is further achieved by our model. This indicates integrating the assignment and routing decision processes plays an important role in reducing the expected delay.

6. Concluding Remarks

In this paper, we propose a novel prescriptive ML approach for a data-driven stochastic VRP with deadlines and endogenous travel time. We then use non-parametric approaches including kNN and KDE to estimate the route-dependent travel time distribution. An LBBD scheme with some acceleration strategies and an ML-based lower-bound heuristic was proposed to solve the problem. We conduct a practical case study with real-world data for a food delivery routing problem.

The computational results show that the proposed algorithm can significantly decrease the average absolute gap when compared to the default LBBD and a genetic heuristic based on Vidal et al. (2012). In terms of the out-of-sample performance, our data-driven stochastic VRP with the kNN method successfully reduces the out-of-sample expected delay when compared with an adjusted version of the assignment parametric models proposed by Liu et al. (2021), the order-assignment decisions obtained from the DOA-DRO model used in Liu et al. (2021), and the implemented route decisions observed in the real data set. We also quantify the value of the proposed routes with different service deadlines. This indicates that integrating the assignment and routing decision processes within a non-parametric framework plays an important role in reducing the expected delay.

We believe that the models and real-world case study for the VRP should motivate its use in other types of applications, e.g. network design, and facility location, etc. It would also be interesting to extend our proposed VRP to a multiperiod VRP in the context of contextual optimization.

References

Adulyasak Y, Jaillet P (2016) Models and algorithms for stochastic and robust vehicle routing with deadlines. *Transportation Science* 50(2):608–626.

Anderson R, Huchette J, Ma W, Tjandraatmadja C, Vielma JP (2020) Strong mixed-integer programming formulations for trained neural networks. *Mathematical Programming* 183(1):3–39.

Bertsimas D, Brown DB, Caramanis C (2011) Theory and applications of robust optimization. *SIAM Review* 53(3):464–501.

Bertsimas D, Kallus N (2020) From predictive to prescriptive analytics. *Management Science* 66(3):1025–1044.

Birge JR, Louveaux F (2011) *Introduction to Stochastic Programming* (Springer Science & Business Media).

Carlsson JG, Delage E (2013) Robust partitioning for stochastic multivehicle routing. *Operations Research* 61(3):727–744.

Çavdar B, Sokol J (2015) A distribution-free TSP tour length estimation model for random graphs. *European Journal of Operational Research* 243(2):588–598.

Chien TW (1992) Operational estimators for the length of a traveling salesman tour. *Computers & Operations Research* 19(6):469–478.

Coelho LC, Cordeau JF, Laporte G (2014) Thirty years of inventory routing. *Transportation Science* 48(1):1–19.

Cui R, Lu Z, Sun T, Golden JM (2023) Sooner or later? promising delivery speed in online retail. *Manufacturing & Service Operations Management*. https://doi.org/10.1287/msom.2021.0174.

Fu G, Zhang P, Lei D, Qi W, Shen ZJM (2023) Learning for guiding: A framework for unlocking trust and improving performance in last-mile delivery. Available at SSRN: https://ssrn.com/abstract=4639706.

García-Portugués E (2023) *Notes for nonparametric statistics*. https://bookdown.org/egarpor/NP-UC3M/.

Ghosal S, Ho CP, Wiesemann W (2023) A unifying framework for the capacitated vehicle routing problem under risk and ambiguity. *Operations Research*. https://doi.org/10.1287/opre.2021.0669.

Gilardi N, Bengio S, Kanevski M (2002) Conditional gaussian mixture models for environmental risk mapping. *Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing*, 777–786.

Goel V, Grossmann IE (2006) A class of stochastic programs with decision dependent uncertainty. *Mathematical Programming* 108(2):355–394.

Grimstad B, Andersson H (2019) ReLU networks as surrogate models in mixed-integer linear programs. *Computers & Chemical Engineering* 131:106580.

Hildebrandt FD, Thomas BW, Ulmer MW (2023) Opportunities for reinforcement learning in stochastic dynamic vehicle routing. *Computers & Operations Research* 150:106071.

Hildebrandt FD, Ulmer MW (2022) Supervised learning for arrival time estimations in restaurant meal delivery. *Transportation Science* 56(4):1058–1084.

Hoeffding W (1994) A non-parametric test of independence. *The Collected Works of Wassily Hoeffding* 214–226.

Hooshmand Khaligh F, MirHassani S (2016) A mathematical model for vehicle routing problem under endogenous uncertainty. *International Journal of Production Research* 54(2):579–590.

Huang Z, Lam H, Zhang H (2022) Evaluating aleatoric uncertainty via conditional generative models. ArXiv preprint arXiv:2206.04287.

Jaillet P, Qi J, Sim M (2016) Routing optimization under uncertainty. *Operations Research* 64(1):186–200.

Jiang H, Cao J, Shen ZJM (2024) Intertemporal pricing via nonparametric estimation: Integrating reference effects and consumer heterogeneity. *Manufacturing & Service Operations Management* 26(1):28–46.

Kwon O, Golden B, Wasil E (1995) Estimating the length of the optimal TSP tour: An empirical study using regression and neural networks. *Computers & Operations Research* 22(10):1039–1046.

Laporte G (1992) The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research* 59(3):345–358.

Larsen E, Frejinger E, Gendron B, Lodi A (2022a) Fast continuous and integer L-shaped heuristics through supervised learning. ArXiv preprint arXiv:2205.00897.

Larsen E, Lachapelle S, Bengio Y, Frejinger E, Lacoste-Julien S, Lodi A (2022b) Predicting tactical solutions to operational planning problems under imperfect information. *INFORMS Journal on Computing* 34(1):227–242.

Li Y, Côté JF, Coelho LC, Wu P (2022) Novel formulations and logic-based Benders decomposition for the integrated parallel machine scheduling and location problem. *INFORMS Journal on Computing* 34(2):1048–1069.

Liu S, He L, Max Shen ZJ (2021) On-time last-mile delivery: Order assignment with travel-time predictors. *Management Science* 67(7):4095–4119.

Liu S, Luo Z (2023) On-demand delivery from stores: Dynamic dispatching and routing with random demand. *Manufacturing & Service Operations Management* 25(2):595–612.

Lysgaard J, Letchford AN, Eglese RW (2004) A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming* 100(2):423–445.

Mišić VV, Perakis G (2020) Data analytics in operations management: A review. *Manufacturing & Service Operations Management* 22(1):158–169.

Nazari M, Oroojlooy A, Snyder L, Takác M (2018) Reinforcement learning for solving the vehicle routing problem. *Advances in Neural Information Processing Systems* 31.

Nguyen VA, Zhang F, Blanchet J, Delage E, Ye Y (2020) Distributionally robust local non-parametric conditional estimation. *Advances in Neural Information Processing Systems* 33:15232–15242.

Pearson K (1900) X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 50(302):157–175.

Rahmaniani R, Crainic TG, Gendreau M, Rei W (2017) The Benders decomposition algorithm: A literature review. *European Journal of Operational Research* 259(3):801–817.

Sadana U, Chenreddy A, Delage E, Forel A, Frejinger E, Vidal T (2023) A survey of contextual optimization methods for decision making under uncertainty. *arXiv preprint arXiv:2306.10374* .

Serkan Özarık S, da Costa P, Florio AM (2022) Machine learning for data-driven last-mile delivery optimization. Available at SSRN: https://ssrn.com/abstract=4012376.

Shafer G, Vovk V (2008) A tutorial on conformal prediction. *Journal of Machine Learning Research* 9(3):371–421.

Shaked M, Shanthikumar JG (2007) Stochastic Orders. *New York, NY: Springer New York*.

Soeffker N, Ulmer MW, Mattfeld DC (2022) Stochastic dynamic vehicle routing in the light of prescriptive analytics: A review. *European Journal of Operational Research* 298(3):801–820.

Torres F, Gendreau M, Rei W (2022) Vehicle routing with stochastic supply of crowd vehicles and time windows. *Transportation Science* 56(3):631–653.

Toth P, Vigo D (2014) *Vehicle Routing: Problems, Methods, and Applications* (Society for industrial and applied mathematics).

Vidal T (2022) Hybrid genetic search for the CVRP: Open-source implementation and SWAP neighborhood. *Computers & Operations Research* 140:105643.

Vidal T, Crainic TG, Gendreau M, Lahrichi N, Rei W (2012) A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Operations Research* 60(3):611–624.

Yu X, Shen S (2022) Multistage distributionally robust mixed-integer programming with decision-dependent moment-based ambiguity sets. *Mathematical Programming* 196(1-2):1025–1064.

Zhang Y, Zhang Z, Lim A, Sim M (2021) Robust data-driven vehicle routing with time windows. *Operations Research* 69(2):469–485.

Appendix A: Algorithm Details

A.1. Implementation of the LBBD Algorithm

Algorithm 1 describes the detailed procedure of the LBBD algorithm.

Algorithm 1: Logic-based Benders Decomposition Algorithm

```
Input A tolerance $\epsilon > 0$.
Initialize $\nu = 0$, $UB = +\infty$, $LB = -\infty$, $\mathcal{N} = \{o\}$, $o$ has no branching constraints.
while ($\mathcal{N}$ is nonempty and $UB - LB > \epsilon$ ) do
    $\nu = \nu + 1$.
    Select a node $o \in \mathcal{N}$, $\mathcal{N} \leftarrow \mathcal{N}/\{o\}$.
    Solve the LP relaxation of (MP) at the node $o$, obtain optimal solution $(\bar{\boldsymbol{x}}^\nu, \bar{\boldsymbol{s}}^\nu)$ and optimal objective
     $lobj^\nu$.
    if $lobj^\nu < UB$ then
        if $(\bar{\boldsymbol{x}}^\nu, \bar{\boldsymbol{s}}^\nu)$ is integer then
            For $\ell \in \mathcal{N}_d$, compute the expected delay $obj^\ell$ for $\bar{\boldsymbol{s}}^{\nu\ell}$.
            Add optimality cuts (8) to (MP).
            if $\sum_{\ell \in \mathcal{N}_d} obj^\ell \leq UB$ then
                $UB = \sum_{\ell \in \mathcal{N}_d} obj^\ell$, and $(\boldsymbol{x}^*, \boldsymbol{s}^*) = (\bar{\boldsymbol{x}}^\nu, \bar{\boldsymbol{s}}^\nu)$.
            end
        end
        if $(\bar{\boldsymbol{x}}^\nu, \bar{\boldsymbol{s}}^\nu)$ is fractional then
            Update $LB := \max\{LB, lobj^\nu\}$.
            Branch, resulting in nodes $o^*$ and $o^{**}$, $\mathcal{N} \leftarrow \mathcal{N} \cup \{o^*, o^{**}\}$.
        end
    end
end
return UB and its corresponding optimal solution $(\boldsymbol{x}^*, \boldsymbol{s}^*)$.
```

A.2. Warm Start

Algorithm 2 gives the outline of the warm starting procedure.

Algorithm 2: Warm Start

```
Use the hybrid genetic search method to solve problem (10), and obtain feasible subpopulation
 $\{\boldsymbol{x}^m, \boldsymbol{s}^m\}_{m=1,\ldots,M'}$.
Compute the expected delay $obj^m$ of $(\boldsymbol{x}^m, \boldsymbol{s}^m)$ for $m = 1, \ldots, M'$.
Let $m^* = \arg\min_{m=1,\ldots,M'} obj^m$, and $\hat{\eta} = obj^{m^*}$.
Reorder the vehicle indices for $(\boldsymbol{x}^{m^*}, \boldsymbol{s}^{m^*})$ to satisfy the symmetry-breaking constraints.
return $(\boldsymbol{x}^{m^*}, \boldsymbol{s}^{m^*}, \hat{\eta})$.
```

Algorithm 3: Local Search Procedures

```
Input $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{s}})$ and expected delay $o\bar{b}j$.
Output $(\bar{\boldsymbol{x}}'', \bar{\boldsymbol{s}}'')$ and expected delay $o\bar{b}j''$.
Find a solution $\bar{\boldsymbol{s}}'$ by improving $\bar{\boldsymbol{s}}$ with local search.
Let $o\bar{b}j'$ be the expected delay based on $\bar{\boldsymbol{s}}'$.
if $o\bar{b}j' < o\bar{b}j$ then
    Construct $\bar{\boldsymbol{x}}'$ based on $\bar{\boldsymbol{s}}'$.
    Reorder the vehicle indices of $(\bar{\boldsymbol{x}}', \bar{\boldsymbol{s}}')$ to satisfy the symmetry-breaking constraints.
    Set $(\boldsymbol{x}'', \boldsymbol{s}'', obj'') := (\bar{\boldsymbol{x}}', \bar{\boldsymbol{s}}', o\bar{b}j')$.
else
    Set $(\boldsymbol{x}'', \boldsymbol{s}'', obj'') := (\bar{\boldsymbol{x}}, \bar{\boldsymbol{s}}, o\bar{b}j)$.
end
```

A.3. Local Search

The local search procedures are provided in Algorithm 3.

A.4. Determining the Value of ρ

Algorithm 4 gives an overview of the bisection method to determine the value of ρ.

Algorithm 4: Determining the value of ρ

```
Input A tolerance $\epsilon' > 0$.
Initialize Upper bound $\bar{\rho}$ and lower bound $\underline{\rho}$.
$\rho = (\bar{\rho} + \underline{\rho})/2$.
Compute the probability:
    $$\frac{\sum_{j=1}^{J'} \mathbf{1}\left\{(1/(T+1))\sum_{t=1}^{T}(\sum_{a\in\mathcal{A}} \theta_a^t \hat{s}_a^{\prime j} + \bar{\theta}^t - \tau)^+ \le \mathbb{E}_{\mathbb{P}_{(\hat{s}'^j)}}\left[\left(\tilde{\xi} - \tau\right)^+\right]\right\}}{J'},$$
while $\bar{\rho} - \underline{\rho} > \epsilon'$ do
    if the probability is larger than $\epsilon'$ then
        $\underline{\rho} = \rho$ and $\rho = (\bar{\rho} + \underline{\rho})/2$.
    end
    else
        $\bar{\rho} = \rho$ and $\rho = (\bar{\rho} + \underline{\rho})/2$.
    end
end
```

A.5. LBBD Algorithm with the Acceleration Strategies and Heuristic

Algorithm 5 presents the procedure of our implementation of the LBBD algorithm with the acceleration strategies and the lower bound heuristic.

Algorithm 5: Logic-based Benders Decomposition Algorithm

```
Input A tolerance $\epsilon > 0$.
Initialize $\nu = 0$, $UB = +\infty$, $LB = -\infty$, $\mathcal{N} = \{o\}$, $o$ has no branching constraints.
$(\boldsymbol{x}^{m^*}, \boldsymbol{s}^{m^*}, \hat{\eta}) \Leftarrow$ Warm Start.
UB=$\hat{\eta}$, and set $\boldsymbol{x}^{m^*}, \boldsymbol{s}^{m^*}$ to be an initial solution for (MP).
while ($\mathcal{N}$ is nonempty and $UB - LB > \epsilon$ ) do
    $\nu = \nu + 1$.
    Select a node $o \in \mathcal{N}$, $\mathcal{N} \leftarrow \mathcal{N}/\{o\}$.
    Solve the LP relaxation of (MP) at the node $o$, and obtain the optimal solution $(\bar{\boldsymbol{x}}^{\nu}, \bar{\boldsymbol{s}}^{\nu})$ and the optimal
      objective $lobj^{\nu}$.
    if $lobj^{\nu} < UB$ then
        if $(\bar{\boldsymbol{x}}^{\nu}, \bar{\boldsymbol{s}}^{\nu})$ is an integer then
            For $\ell \in \mathcal{N}_d$, we compute the expected delay $obj^{\nu\ell}$ for $\bar{\boldsymbol{s}}^{\nu\ell}$.
            $(\boldsymbol{x}'', \boldsymbol{s}'', obj'') \Leftarrow$ Local Search Procedure.
            Add the general optimality cut based on the solution $\boldsymbol{s}''$ to (MP).
            if $\sum_{\ell \in \mathcal{N}_d} obj''^{\ell} \leq UB$ then
                $UB = \sum_{\ell \in \mathcal{N}_d} obj''^{\ell}$, and $(\boldsymbol{x}^*, \boldsymbol{s}^*) = (\bar{\boldsymbol{x}}'', \bar{\boldsymbol{s}}'')$.
            end
        end
        if $(\bar{\boldsymbol{x}}^{\nu}, \bar{\boldsymbol{s}}^{\nu})$ is fractional then
            Update $LB := \max\{LB, lobj^{\nu}\}$.
            Branch, resulting in nodes $o^*$ and $o^{**}$, $\mathcal{N} \leftarrow \mathcal{N} \cup \{o^*, o^{**}\}$.
        end
    end
end
return UB and its corresponding optimal solution $(\boldsymbol{x}^*, \boldsymbol{s}^*)$.
```

Appendix B: Proof of Propositions and Lemmas

B.1. Proof of Proposition 1

Let $\bar{\boldsymbol{s}}^{\nu}$ be a routing solution generated in the νth iteration for one of the vehicles. On a later iteration, if one vehicle's route does not start by visiting the first customer location of $\bar{\boldsymbol{s}}^{\nu}$, then the right hand of the cut (9) will be disabled. If it does start with the same location, then either the route ends there and thus the cut returns the right expected delay, or it pursues to a second destination. In the latter case, the second destination could be different then the route in $\bar{\boldsymbol{s}}^{\nu}$ thus disabling the cut, or pass by the same second stop. If the route is complete, then the cut returns the expected delays of a route that only goes to these two destinations. Otherwise, the equation pursues either canceling the cut or adding the right difference to recuperate the expected delay obtained if this was the last destination. Hence, the general optimality cuts (9) contain both the traditional optimality cut (8) together with all the optimality cuts for subroutes of $\bar{\boldsymbol{s}}^{\nu}$. They are therefore valid. □

B.2. Proof of Lemma 1

Given a $\boldsymbol{s}^{\ell}$ and $\ell \in \mathcal{N}_d$, from Shaked and Shanthikumar (2007), we have that first-order stochastic dominance implies

$$\mathbb{E}_{\mathbb{P}_{(\boldsymbol{s}^{\ell})}}\left[\phi\left(\xi\right)\right] \geq \mathbb{E}_{\bar{\mathbb{P}}_{(\boldsymbol{s}^{\ell})}}\left[\phi\left(\xi\right)\right], \text{ for all non-decreasing functions } \phi.$$

Let $\phi\left(\xi\right) = \left(\xi - \tau\right)^{+}$, which is non-decreasing, then

$$\mathbb{E}_{\mathbb{P}_{(\boldsymbol{s}^{\ell})}}\left[\left(\xi - \tau\right)^{+}\right] \geq \mathbb{E}_{\bar{\mathbb{P}}_{(\boldsymbol{s}^{\ell})}}\left[\left(\xi - \tau\right)^{+}\right].$$

Hence, (11c) is a relaxation of (11b). Therefore, problem (3) is equivalent to problem (11). □

B.3. Proof of Lemma 2

Under the null hypothesis, one can first establish that since $\hat{F}_{Y|X}(y;x)$ is continuous we have that Z is uniformly distributed among $\{1,\ldots,K+1\}$. Namely, for all $k > 1$:

$$\mathbb{P}(Z = k) = \mathbb{P}(\hat{Y}_{(k-1)}(X) < Y < \hat{Y}_{(k)}(X)) = 1/(K+1),$$

where $\hat{Y}_{(k)}(X)$ is the k-th smallest realization in the set $\{\hat{Y}(X)\}$, because all $\hat{Y}_k(X)$ and Y are i.i.d. given X. It therefore follows straightforwardly (see Pearson (1900)) that χ^2_K is a Pearson's cumulative test statistic for the multinomial distribution of $\sum_{i=1}^{N} \mathbf{1}\{z_i = k\}$. □

Appendix C: Computational Results for KDE Method

We present the lower bound heuristic and the computational results for KDE method in this section.

C.1. Computational Results on Lower Bound Heuristic for KDE Method

Table 3 presents the lower bound probability, and the average absolute gap (in minutes) when the lower bound value is larger than the expected delay for the KDE method based on a newly generated set of routes with different numbers of locations.

Table 3 The lower bound probability and the average absolute gap (in minutes) are reported.

τ	25	35	45	55	65
LBProb	0.85	0.85	0.85	0.85	0.85
Avg_AGap	0.45	0.25	0.11	0.04	0.01

C.2. Computational Results under KDE Method:

We present our results on the acceleration strategies and lower bound improvement heuristic implemented in the LBBD algorithm 5 for the data-driven VRP with the KDE method. In this section, we set the service deadline to be 55 minutes, which corresponds to the practical scenario (Liu et al. 2021). We illustrate the performance of the six different variants presented in Section 5.5.3. Table 4 shows the average upper bound and the estimated lower bound, the average absolute gap, the solution time for the instances that can be solved to optimality, the average time spent for the subproblem and local search procedures, the average number of cuts and nodes, and the optimal ratio for the KDE method, where the optimal ratio is computed as the number of solved instances over the total 440 instances.

Table 4 The average upper bound (UB) and estimated lower bound (Est_LB), the average absolute gap (Gap), the average solution time (in seconds) for the instances solved optimally (AvT), and subproblem and local search procedures (AvT_SP), the average number of cuts (# of cuts) and nodes (# of nodes), and the optimal ratio (Ratio) are reported.

Model	Method	UB	Est_LB	Gap	AvT	AvT_SP	# of cuts	# of nodes	Ratio
KDE	BDWLG_LH	0.17	0.10	0.07	104.0	72.0	782	29,847	0.39
	BDWLG	0.11	0.00	0.11	117.4	253.4	3,285	8,883	0.12
	BDWL	0.12	0.00	0.12	96.3	323.1	4,512	8,687	0.11
	BDW	2.60	0.00	2.60	N/A	3.8	20,278	21,774	0.00
	BD	2.87	0.00	2.87	N/A	3.8	20,388	22,282	0.00
	Genetic	0.78	0.00	0.78	N/A	N/A	N/A	N/A	0.04

Similar to Table 2, we can observe from Table 4 that BDWLG_LH achieves the smallest average absolute gap and largest optimal ratio among all the six methods, while BDWLG has a better average upper bound. For KDE, the improvement in the average absolute gap and the optimal ratio is about 40% and 69%, respectively. In addition, BDWL yields a slightly poor performance when compared with BDWLG, and BDWL has a significantly better performance than BDW. Furthermore, BDW outperforms BD. For KDE, BD and BDW perform worse than the genetic heuristic. The results from Table 4 further confirm the efficiency of the acceleration strategies and the lower bound heuristic proposed in Section 4.

Appendix D: The Formulations and Solution Schemes of ALiu$_{\text{opt}}$ and ALiu$_{\text{kNN}}$

Without loss of generality, in this paper, we assume that the travel time of arcs includes the service time at nodes. Then the assignment model proposed by Liu et al. (2021) is reduced to the following problem:

$$\underset{\boldsymbol{y}}{\text{minimize}} \sum_{\ell \in \mathcal{N}_d} (q_\ell - \tau)^+ \tag{16a}$$

$$
\begin{aligned}
\text{subject to } & \sum_{\ell \in \mathcal{N}_d} y_{i\ell} = 1, && \forall i \in \mathcal{N}_c, && \text{(16b)}\\
& \sum_{i \in \mathcal{N}_c} d_i y_{i\ell} \leq Q, && \forall \ell \in \mathcal{N}_d, && \text{(16c)}\\
& q_\ell = q(\boldsymbol{y}_\ell), && \forall \ell \in \mathcal{N}_d, && \text{(16d)}\\
& y_{i\ell} \in \{0,1\}, && \forall i \in \mathcal{N}_c, \ell \in \mathcal{N}_d, && \text{(16e)}
\end{aligned}
$$

where binary decision $y_{i\ell}$ is equal to one if customer location i is assigned to vehicle ℓ, zero otherwise. $q(\boldsymbol{y}_\ell)$ denotes a function of the customer locations served by driver ℓ that predict the total travel time by driver ℓ.

Following Liu et al. (2021), we also use LASSO as the prediction model, and the tractable predictors proposed by Liu et al. (2021) for the travel time. After linearizing each class of tractable predictors (see Liu et al. (2021) for more details), we can solve linear problem (16).

Given the optimal assignment decision $\hat{y}_{i\ell}$ for customer $i \in \mathcal{N}_c$ and vehicle $\ell \in \mathcal{N}_d$, we want to optimize the route for each vehicle ℓ. Let set $\mathcal{I}_\ell := \{i \in \mathcal{N}_c : \hat{y}_{i\ell} = 1\}$ for all $\ell \in \mathcal{N}_d$, i.e., the set of customer locations that are assigned to vehicle ℓ, and set $\mathcal{J}_\ell := \{1, \ldots, |\mathcal{I}_\ell|\}$ denotes the set of positions for vehicle ℓ. Binary decision $z^\ell_{ij} = 1$ if customer i is assigned to position j for vehicle ℓ.

In order to use the LBBD algorithm to solve the routing model for $\ell \in \mathcal{N}_d$, we first define the following master problem:

$$
\begin{aligned}
\underset{\boldsymbol{\eta}, \boldsymbol{z}}{\text{minimize}}\ & \eta_\ell, && && \text{(17a)}\\
\text{subject to } & \sum_{i \in \mathcal{I}_\ell} z^\ell_{ij} = 1, && \forall j \in \mathcal{J}_\ell, && \text{(17b)}\\
& \sum_{j \in \mathcal{J}_\ell} z^\ell_{ij} = 1, && \forall i \in \mathcal{I}_\ell, && \text{(17c)}\\
& z^\ell_{ij} \in \{0,1\}, && \forall i \in \mathcal{I}_\ell, j \in \mathcal{J}_\ell. && \text{(17d)}\\
& \eta_\ell \geq 0. && && \text{(17e)}
\end{aligned}
$$

The constraints guarantee that each customer is assigned to one position, and each position is allotted to only one customer, and customer i can be assigned to a position belonging to ℓ only if i is assigned to vehicle ℓ. Given $\hat{z}^\ell_{ij}$ from the master problem (17), let

$$\bar{\mathcal{J}}^i_\ell = \{j \in \mathcal{J}_\ell | \hat{z}^\ell_{ij} = 1\}.$$

We can also extract the corresponding route $\hat{\boldsymbol{s}}^\ell$. For ALiu$_{\text{opt}}$, we use CGMM to generate $N' = 10,000$ samples $\bar{\xi}^i_\ell$ and let the expected delay for vehicle $\ell \in \mathcal{N}_d$ be $lobj_\ell = \frac{1}{N'} \sum_{i=1}^{N'} \left(\bar{\xi}^i_\ell - \tau\right)^+$. We then add the following optimality cut to the master problem:

$$\eta_\ell \geq lobj_\ell - lobj_\ell \sum_{i \in \mathcal{I}_\ell} \sum_{j \in \bar{\mathcal{J}}^i_\ell} (1 - z^\ell_{ij}) \tag{18}$$

We can also use the kNN method to optimize the routing decision, i.e., ALiu$_{\text{kNN}}$. We use the kNN method to define the discrete probability $p_i(\hat{\boldsymbol{s}}^\ell)$, then $lobj_\ell = \sum_{i=1}^{N(\hat{\boldsymbol{s}}^\ell)} p_i(\hat{\boldsymbol{s}}^\ell)\left(\xi^i(\hat{\boldsymbol{s}}^\ell) - \tau\right)^+$.