



# Data-driven optimization for last-mile delivery

Hongrui Chu<sup>1</sup> · Wensi Zhang<sup>2</sup> · Pengfei Bai<sup>1</sup> · Yahong Chen<sup>3</sup>

Received: 9 October 2020 / Accepted: 1 February 2021 / Published online: 22 February 2021  
© The Author(s) 2021

## Abstract

This paper considers how an online food delivery platform can improve last-mile delivery services' performance using multi-source data. The delivery time is one critical but uncertain factor for online platforms that also regarded as the main challenges in order assignment and routing service. To tackle this challenge, we propose a data-driven optimization approach that combines machine learning techniques with capacitated vehicle routing optimization. Machine learning methods can provide more accurate predictions and have received increasing attention in the operations research field. However, different from the traditional predict-then-optimize paradigm, we use a new smart predict-then-optimize framework, whose prediction objective is constructed by decision error instead of prediction error when implementing machine learning. Using this type of prediction, we can obtain a more accurate decision in the following optimization step. Efficient mini-batching gradient and heuristic algorithms are designed to solve the joint order assignment and routing problem of last-mile delivery service. Besides, this paper considers the mutual effect between routing decision and deliverytime, and provides the corresponding solution algorithm. In addition, this paper conducts a computational study and finds that the proposed method's performance has an approximate 5% improvement compared with other methods.

**Keywords** Data-driven optimization · Last-mile delivery · Machine learning · Prediction · Routing behavior

## Introduction

The last 20 years have witnessed an explosion of e-commerce, which has prompted customers to purchase online instead of visiting physical stores. Platforms like MeiTuan and UberEATS are selling many foods every day and have become more and more popular because of their convenient online catering service. These platforms provide end-to-end services for customers and restaurants. This means that after customers make requests on platforms, restaurants prepare food in the central kitchen once they receive orders from platforms, find drivers to deliver the food, and finally deliver the food from restaurants to customers according to the arrangement of platforms [1]. There is no doubt that

platforms play an essential role in this food delivery service; however, one key challenge they have to face is how quickly and efficiently they deliver food from restaurants to customers with limited drivers. This problem is consistent with the well-known Last-Mile Problem (LMP) [2], which transports goods from a public node to customers and can be formulated by a modified VRP (Vehicle Routing Problem) model. In our case, the public node could be the nearest local logistics service center or central kitchen, and the transport means of food LMP include walking, taking a taxi, or driving a private vehicle.

LMP is potentially very expensive and accounts for a large part expenditure of the whole business activity of food platforms [3]. With the increasing business competition and fast delivery requirement, online platforms have to improve the performance of Last-Mile Transportation Systems (LMTS), to gain a large market share, control cost and ensure delivery timeliness. So, it is a vital task for online food platforms to solve LMP effectively. What's more, the last-mile delivery problem is a computational challenge because VRP is an NP-hard problem in theory, and there are many uncertainty factors in practice that affect the parameter values [4]. Using optimization theories and methods, related

---

✉ Hongrui Chu  
chuhongrui@cueb.edu.cn

<sup>1</sup> School of Management and Engineering, Capital University of Economics and Business, Beijing 100070, China

<sup>2</sup> School of Economics, Ocean University of China, Qingdao 266100, Shandong, China

<sup>3</sup> Beijing Intelligent Logistics System Collaborative Innovation Center, Beijing Wuzi University, Beijing 101149, China

managers can make order assignment and routing decisions for LMP by solving an improved VRP, which has been solved effectively by exact or heuristic algorithms. However, realistic uncertain factors in last-mile delivery, such as drivers' behavior characteristics and traffic conditions, are difficult to be predicted exactly using general statistical methods for their inherent complexities. This leads to the inaccurate estimation of optimization parameters, such as service time or service cost. Consequently, the uncertain factors will result in an unreasonable optimization formulation and ineffective management decision.

Early related studies usually assume that optimization models' parameters are constant and estimate those parameters by sample statistics or specialist experience. In those works, researchers focus on how to design efficient algorithms to derive exact or satisfactory solutions. Considering the existence of uncertainties in practice, optimization problems involving uncertain factors began to receive more attention. In the beginning, stochastic programming theory is proposed to handle the randomness of parameters [5]. For example, as one of the widely used stochastic programming measures, chance constraint requires the probability of reference target to satisfy a given threshold. In the past decade, robust optimization (RO) and distributionally robust optimization (DRO) have been drawing more and more attention for their advantages in using real data by combining statistical techniques [6–8]. In the robust optimization framework, it is crucial to construct the uncertain set of parameters based on the sample data, which also has a significant impact on the optimization results and decision.

Nowadays, with the development of information technology, massive data have been accumulated in many fields. In the meanwhile, AI technologies include machine learning, deep learning, and reinforcement learning have exhibited high effectiveness in prediction, classification, and other problems for many practical applications. Sequentially, data-driven optimization, which combines traditional optimization methods with AI prediction tools, has become a new research frontier in the operations research field. Typically, the optimization decision can be improved dramatically due to the accessibility and excellence in parameter estimation of the machine learning method. The data-driven combination form has been applied in many real analytics fields [9–11]. However, most of the present studies execute the procedures of machine learning and optimization separately. More specifically, the machine learning method is only used to generate parameters' prediction, and the optimization model is only used to generate an optimal decision based on the estimated parameters given by machine learning. This procedure takes the same paradigm as stochastic programming, robust optimization, and distributionally robust optimization. The paradigm is defined as *predict-then-optimize* [12]. In essence, the predict-then-optimize paradigm does

not account for how the predictions would be used in the downstream optimization model, no matter how the prediction methods are used. This asynchronous procedure leads to an inconsistency phenomenon in the predict-then-optimize paradigm, that is, the quality of a prediction is not equivalent to the quality of a decision.

To overcome the shortcomings of the predict-then-optimize paradigm, Elmachtoub and Grigas [12] propose a smart predict-then-optimize (SPO) framework. Although the SPO approach also maintains the sequence that first prediction then optimization, but the quality of parameter prediction is measured by decision error rather than prediction error. The authors also present some simple examples to demonstrate the SPO strongly outperforms the ordinary optimization models with regular machine learning prediction.

In this paper, we aim to improve online platforms' last-mile delivery performance using the SPO framework. We consider a simplified scenario: an online delivery service provider first prepares packages at his central depot and then deliver all packages to the customers by drivers within a certain radius from the depot. Each driver has a limited capacity, and the service provider needs to ensure that each driver's total travel time is less than a given threshold. This setting is a typical and straightforward LMP case for online food platforms or logistics companies, and the problem can be formulated by a Capacitated Vehicle Routing Problem (CVRP) model. However, different from the previous conventional models, we assume the travel time between any two customers is uncertain and can be predicted by the large volume and multi-source data. The data may incorporate the distance, weather, season, driver's profile, and real-time traffic data from mobile applications. The booming online platforms provide a convenient shopping experience for customers and accumulate enough relevant data for smart decisions. Therefore, adequate data guarantee the basic requirements of our methods.

What's more, this paper assumes that the driver's behavior impacts his travel time, which leads to a mutual effect between parameters and decision variables. The problem setting is similar to [1], which integrates order assignment optimization with a travel time prediction model. However, they do not take the decision error caused by inaccurate prediction into consideration, while our work involving this point by using a different mathematic formulation and SPO data-driven optimization implementation framework. Furthermore, this work provides a specific solution algorithm to handle the coupled problem of parameters and decision variables.

The remainder of this paper is organized as follows: we review the relevant studies about data-driven optimization and LMP in Sect. 2. Then Sect. 3 provides the corresponding model and the solving algorithms. In Sect. 4, we compare the proposed model with other methods to indicate our

model's advantage by numerical studies. Finally, the conclusion and future research directions are presented in Sect. 5.

## Related literature

There are several literature streams relevant to our work. This section presents a brief overview of those researches from two domains: LMP and data-driven optimization.

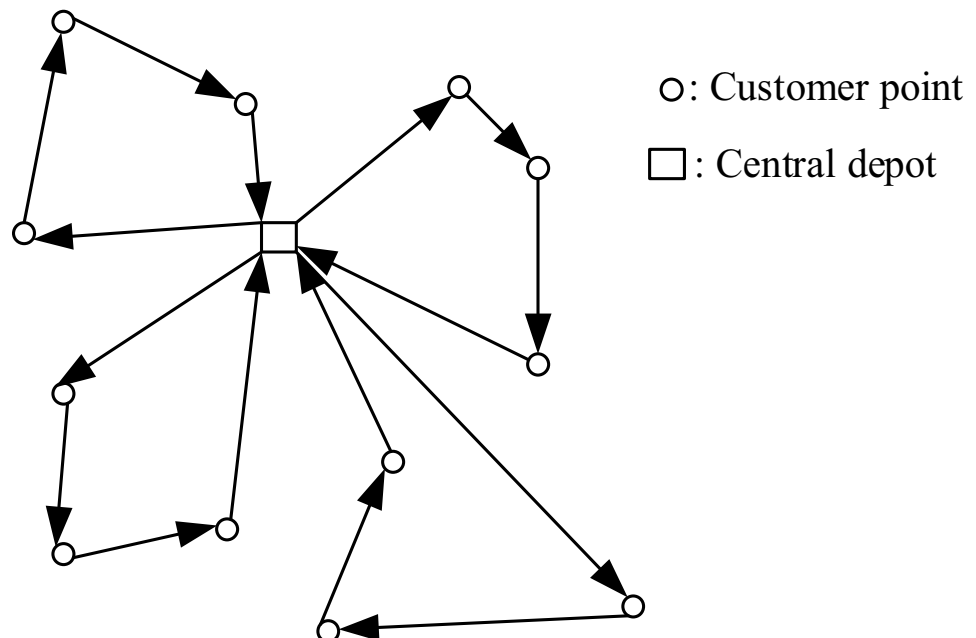
### Last-mile problem

The classical LMP can be described schematically as Fig. 1, which contains a central depot that prepares packages and several customers distributed around the central depot in a fixed area. A fleet of vehicles visit customers and return to the central depot after all the packages are delivered. There are also some variants of the classical LMP. For example, Wang and Odoni [2] study the last-mile problem for passenger transportation in a stochastic setting. This paper studies the last-mile problem following a classical VRP framework, which is a packages' assignment and delivery problem and has been widely studied in the operations research literature [13]. Jiang et al. [14] consider a similar last-mile delivery problem and aim to reduce the total costs and carbon emissions, their model is a variant of the traveling salesman problem. Zhou et al. [15] study a green VRP considering dual last-mile delivery services with stochastic travel times. However, both of them do not consider the prediction problem with multivariate data. VRP is first introduced by Dantzig and Ramser [16], they concern the gasoline delivery problem for a gas station by truck. After their seminal paper, a

vast number of various extensional works of VRP have been published, range from stochastic context [17], dynamic environment [18], delivery time constraint [19] to time window constraint [20]. For a broad overview of VRP variants, the readers are recommended to refer to Toth and Vigo [13].

Due to the complexity and NP-hard of VRP, designing an effective algorithm is always a critical job in this field. Various heuristic or exact algorithms have been proposed and verified. The history of VRP heuristic algorithms is as old as the problem itself. Dantzig and Ramser [16] sketch a simple heuristic based on successive matchings of vertices through the solution of relaxed linear programs. Clarke and Wright [21] propose a useful greedy heuristic to derive the approximate solution of VRP. Since then, a wide variety of constructive and improvement heuristics have been proposed. Such as constructive heuristics provide a starting solution to improve algorithms [22, 23], classical improvement heuristics perform intra-route and inter-route moves [24, 25], metaheuristics include local search methods [26] and population-based heuristics [27]. In essence, VRP is a combinatorial optimization which can also be solved by classical exact algorithms, such as branch-and-bound algorithms with assignment problem relaxation [28, 29], branch-and-bound algorithms with shortest spanning tree [30], branch-and-cut algorithms [31] and column generation algorithms [32]. This paper's proposed LMP model is a kind of capacitated VRP, but it has a prominent characteristic that there are mutual effects between parameters and decision variable. In this paper, a data-driven SPO framework and design-related algorithm is used for the proposed complex model.

**Fig. 1** Schematic of the last-mile transportation



## Data-driven optimization

The main purpose of this study is to improve the optimal vehicle routing decision for last-mile delivery using real data. Therefore, this paper is also closely related to the stream of data-driven optimization. Generally speaking, data-driven optimization approaches find suboptimal solutions from data and highlight the importance of real data on the decision. Previous data-driven optimization approaches can be roughly classified as operational statistics, sample average approximation, and robust optimization. Operational statistics are recognized as the beginning work of data-driven optimization in operational research, this approach integrates parameter estimation and optimization to obtain better solutions compared with the traditional approach. Liyanage and Shanthikumar [33] apply it to a single period newsvendor problem, and Chu et al. [34] investigate how to find the optimal operational statistic with a similar problem setting. The sample average approximation's (SAA) basic idea is using sample average function to approximate expected value function, and then solving sample average optimization problem to derive an optimal solution. Kleywegt et al. [35] introduce this approach and present the convergence rates, stopping rules, and computational complexity of this procedure. After then, the SAA approach is widely used in different problems. For example, Schütz et al. [36] use SAA to solve a two-stage supply chain design problem for a Norwegian meat industry, Chang et al. [37] apply SAA scheme to solve a flood emergency logistics problem. Mathematical programming's solution is usually incorrect when there are random parameters. The reason is that we cannot use a specific distribution to describe the ambiguity of uncertainty parameters. Robust optimization, and its extension distributionally robust optimization are proposed to overcome this shortcoming. Those approaches represent random parameters by some uncertainty sets which do not have a specific distribution expression but can be constructed by historical data. Delage and Ye [8] propose a distributionally robust model that describes the uncertainty by mean and covariance matrix constraints, then provide probabilistic arguments to design optimization model. Bertsimas et al. [38] develop a tractable framework for solving an adaptive distributionally robust linear optimization problem, they minimize the worst-case objective over a class of second-order conic representable ambiguity set.

Although the above data-driven approaches perform well with real data, they do not have a clear way to use feature data. The machine learning model has a prominent advantage in exploiting multi-source data, so recent advanced researches attempt to integrate the machine learning model with the optimization model. Most of this integration follows a predict-then-optimize framework mentioned in the introduction, such as Ferreira et al. [9] first adopt regression

tree to estimate lost sales and predict future demand using historical sale data, then develop an algorithm to solve multiproduct price optimization for the online retailer. Liu et al. [1] use a class of machine learning models, such as LASSO regression, ridge regression, and support vector machine (SVM), to predict travel time, then integrate the predictors within the optimization model to solve the last-mile delivery problem. However, in those papers, the machine learning method only accounts for exact prediction, and does not guarantee optimal decisions in uncertain environments. Therefore, researchers begin to consider improving the prediction model by the decision error instead of prediction error. Kao et al. [39] seek to train a machine learning model that minimizes loss for a nominal unconstrained optimization problem. Considering a similar problem with [39], Donti et al. [40] develop a heuristic algorithm to address a more general setting. Unfortunately, both Kao et al. [39] and Donti et al. [40] focus on the unconstrained optimization models. More recently, Elmachtoub and Grigas [12] design a new predictive framework to incorporate the general optimization problem structure, which enables robust performance against model Misspecification. This paper investigates the LMDP using the SPO framework presented in [12], and proposes an algorithm to deal with the mutual effect between decision variables and uncertainty parameters.

## Problem description and assumptions

We first provide an overview of the concerned online platforms in the last-mile delivery problem, then introduce the SPO framework to integrate prediction model and optimization formulation. Finally, we present relevant algorithms to solve the proposed SPO-LMP model.

### A. LMP modeling

In the beginning, we describe the LMP of the online platforms shown in Fig. 1 in detail. One single provider (platform or logistics company) takes charge of LMP service for a given area. The delivered item can be food or package which has been prepared in the central depot. The provider has known the location and demand quantity of each custom before the delivery service. Furthermore, the provider's LMP consists of two procedures: first, the provider should assign the delivery task to a fleet of vehicles and then decides the delivery sequence for each vehicle. The vehicles are assumed to be heterogeneous with different capacities. The drivers have different driving habits and preferences, which will lead to different travel durations in the same service arc. Once the provider schedules an arrangement plan, the driver will execute a given delivery route, i.e., visiting each customer's destination, and then return to central depot

to pick up the items for the next delivery tour. Delays are inevitable, primarily when the orders are poorly assigned to drivers. For this reason, the provider’s primary focus on the timeliness of delivery due to the intense market competition and customer faster delivery expectation. According to the above explanation, the concerned last-mile problem could be formulated as a capacitated vehicle routing problem with the on-time delivery objective function, whose notations and formulation are presented in the following.

We denote the central depot as node  $0$ , and all customers are given by a set  $N = \{1, 2, \dots, n\}$ . The demand of customer  $i (i \in N)$  is given by a scalar  $d_i \geq 0$ , e.g., the weight or number of the goods needed to deliver. Suppose there are  $k$  drivers who have diverse vehicles with different capacity  $Q_k$  and fixed identical operating cost  $C$ , and the set of drivers is defined as  $k \in \{1, 2, \dots, K\}$ . The drivers make deliveries for customers’ pooling  $N$  starting from central depot node  $0$ , the traveling time from customer node  $i$  to node  $j$  is denoted by  $t_{ij}$ . We need to find the optimal routing for each vehicle to minimize the total delivery time and total operating cost under vehicle capacity constraints.

With the above-given notations, we can construct a delivery model by network optimization framework. Let  $V = \{0\} \cup N = \{0, 1, \dots, n\}$  be the set of nodes and  $E = \{(ij) : i, j \in V, i \neq j \text{ and } (ij) \neq (ji)\}$  be the set of edges, then we get a directed graph  $G = (V, E)$ . Notice that  $|E| = n(n + 1)$  since  $t_{ij} \neq t_{ji}$ . It is reasonable because of the complex road conditions in reality, which make the trip between two customers has different travel time in a different direction. In sum, the last-mile delivery problem is defined by a directed and weighted graph  $G = (V, E, t_{ij}, d_i)$  together with a driver set  $K$ , each vehicle has a delivery capacity  $Q$ .

To present the corresponding formulation concisely, some condensed notations will be further defined. Suppose  $S$  is an arbitrary non-empty subset of  $V, S \subseteq V$ , let the in-arcs and out-arcs of  $S$  be  $\delta^-(S) = \{(i, j) \in E : i \notin S, j \in S\}$  and  $\delta^+(S) = \{(i, j) \in E : i \in S, j \notin S\}$ . the in-arcs and out-arcs of a node  $i$  are calculated by setting  $S = \{i\}$ . For a customer subset  $S$ , it is assumed that the minimum number of drivers used to serve  $S$  is  $r(S)$ . It is clear that  $r(S)$  is dependent on customer demand  $d_i, i \in S$  and vehicle capacity  $Q$ , and  $r(S)$  can be computed by solving a bin packing problem. A simple formulation without considering the heterogeneity of drivers can be presented by two-index decision variable  $x_{ij}$  with  $(i, j) \in E$ ,  $x_{ij}$  is binary variable indicating a driver first deliver  $i$  then  $j$  if  $x_{ij} = 1$ , no driver moves from  $i$  to  $j$  if  $x_{ij} = 0$ . Another decision variable is the exact number of used drivers,  $m$ . With the above notations, the last-mile delivery model can be given as follows:

$$\min_{m \leq K, x} \sum_{(i,j) \in E} t_{ij}x_{ij} + \beta Cm, \tag{1}$$

$$\text{s.t. } \sum_{j \in \delta^+(i)} x_{ij} = 1, \forall i \in N, \tag{2}$$

$$\sum_{i \in \delta^-(j)} x_{ij} = 1, \forall j \in N, \tag{3}$$

$$\sum_{j \in \delta^+(0)} x_{0j} = m, \tag{4}$$

$$\sum_{i,j \in \delta^+(S)} x_{ij} \geq r(S), \forall S \subseteq V, \tag{5}$$

$$x_{ij} \in \{0, 1\}, \forall (i, j) \in E. \tag{6}$$

In this model,  $\beta$  is a constant parameter that adjusts service quality and operating cost. The objective function (1) contains the total travel time cost and total operating cost. Constraints (2) and (3) ensure that each customer can be visited precisely once in a route. Constraint (4) ensures that the number of drivers at work is no more than  $k$ . Constraint (5) works for eliminating illegal routes and serves as capacity constraints.

Following the assumptions commonly used in previous researches, travel time  $t$  is assumed to be uncertain. However, unlike previous studies that define travel time as continuous or discrete random variables, this study tends to emphasize the predictability of random travel time. In reality, travel time is highly dependent on several relevant factors, such as road conditions, weather, transportation distance, and driver characteristics. One of the difficulties in our LMP is that the travel time is dependent on the driver’s behavior. Therefore, we introduce a three-index decision 0–1 variable  $x = (x_{ijk})$  to describe the driver’s behavior.  $x_{ijk} = 1$  if driver  $k$  services customer  $i$  and customer  $j$  successively; otherwise,  $x_{ijk} = 0$ . Let  $y_{ik} = 1$  if driver  $k$  services customer  $i$ , else  $y_{ik} = 0$  for  $i \in \{1, \dots, n\}$ , and  $y_{0k} = m$  represents that every driver must start from a central depot. For simplification, let  $t = (t_{ij})$  and the proposed LMP can be formulated as:

$$\min_{m \leq K, x} \sum_i \sum_j \sum_k t_{ij}x_{ijk} + \beta Cm, \tag{7}$$

$$\text{s.t. } \sum_j x_{ijk} = 1, \forall i \in N, k = 1, \dots, m, \tag{8}$$

$$\sum_i x_{ijk} = 1, \forall j \in N, k = 1, \dots, m, \tag{9}$$

$$\sum_i d_i y_{ik} \leq Q_k, k = 1, \dots, m, \tag{10}$$

$$\sum_k y_{ik} = \begin{cases} 1, & i = 1, \dots, n \\ m, & i = 0 \end{cases} \quad (11)$$

$$x_{ijk} \in \{0, 1\}, \forall (i, j) \in E, k = 1, \dots, m. \quad (12)$$

As mentioned above, online platforms have abundant available data for prediction to make more accurate decisions. Therefore, how to generate a smart decision by exploiting real data is a crucial problem. In this paper, an SPO framework is adopted to solve the LMDP of online platforms.

## B. SPO paradigm

The SPO framework trains prediction models based on a new loss function represented by decision error instead of decision error. In our setting, actual travel time  $t$  is not known when the service provider makes a delivery decision, but some associated features can be observed. We assume  $p$  available features are denoted by vector  $f \in R^p$ , such as the above-mentioned driver's profile, routing length, weather, season or traffic information. Let  $F(t, x, m)$  be the objective function represented by Eq. (7), i.e.,  $F(t, x, m) = \sum_{(i,j) \in E,k} t_{ij} x_{ijk} + \beta m C$ . For simplicity, we condense the decision variable  $x := (x, m)$ , then the objective function is simplified to  $F(t, x)$ . Traditional optimization frameworks usually figure out the conditional distribution of  $t$  with given feature data  $f$ , and then solve the corresponding model with the expected objective function. For example, stochastic programming uses chance constraints or a two-stage model to handle the random variables. However, the SPO framework we adopted integrates the machine learning method with optimization programming. Next, some notations are replinshed and the key ingredients of the SPO paradigm are introduced.

Assume that  $(f_1, t_1), (f_2, t_2), \dots, (f_s, t_s)$  denotes the training data for machine learning, where  $f_i \in \mathcal{X} \subset R^p$  is a feature vector representing contextual information associated with travel time  $t_i$ . A hypothesis class of travel time vector prediction models is  $H : \mathcal{X} \rightarrow R^q$ , which predicts travel time  $\hat{t}$  associated with feature  $f$  by  $\hat{t} := H(f)$ . A loss function  $l(\hat{t}, t)$  quantifies the error in predicting  $\hat{t}$  when the realized true travel time is  $t$ . In general, given the training data  $(f_1, t_1), (f_2, t_2), \dots, (f_s, t_s)$ , an optimal prediction model  $H^*$  is derived to minimize the empirical loss function:

$$\min_H \frac{1}{n} \sum_{i=1}^n l(H(f_i), t_i). \quad (13)$$

After deriving the prediction model  $H^*$ , we can solve the LMP model (7–12) and induce an optimal decision  $X^*(f)$  according to an observed feature vector  $f$ .

In the traditional predict-then-optimize framework, the loss function is usually wholly independent of the LMP

optimization problem. For example, the mean squared error loss function  $l(\hat{t}, t) = \frac{\hat{t}-t}{2}$  is widely used in the linear regression models of machine learning. The corresponding prediction problem is to find the best linear model  $H^*$  to minimize  $\frac{1}{n} \sum_{i=1}^n H^* f_i - t_{i2}$  without considering the underlying structure of the optimization problem.

The SPO framework takes advantage of problem structure to train the prediction model, predict travel time, and then construct the loss function intelligently. According to the defined notations,  $p$  is the dimension of the feature vector,  $s$  is the number of training samples, and  $q$  is the dimension of the decision matrix, i.e.,  $q = i * j$ ,  $i, j \in V$ . The nominal LMP model (1–5) can be simplified as

$$\min_x F(t, x), \quad (14)$$

$$s.t. x \in \Gamma. \quad (15)$$

In this model,  $\Gamma$  is a non-empty feasible region generated by constraints (8–12).  $F(t, x)$  is a linear function concerning  $x$  with redefining  $t := (t, \beta C)$ , then the optimal solution set can be expressed as  $X^*(t) := \arg \min_{x \in \Gamma} t^T x$ .

The SPO loss function is  $l_{SPO}^{X^*}(\hat{t}, t) := t^T X^*(\hat{t}) - F^*(t)$ , in which  $\hat{t}$  is a travel time prediction and  $X^*(\hat{t})$  is a decision obtained by solving (11–12). After implementing the routing decision  $X^*(\hat{t})$ , the actual travel time  $t$  and objective cost  $F^*(t)$  is realized, then excess cost  $t^T X^*(\hat{t}) - F^*(t)$  can be derived. Since  $X^*(\hat{t})$  may contain more than one solution, another definition that does not depend on the particular choice of the optimization oracle  $X^*(\hat{t})$  is provided by

$$l_{SPO}(\hat{t}, t) = \max_{x \in X^*(\hat{t})} t^T x - F^*(t). \quad (16)$$

Given the training data and definition of SPO loss function, we need to determine a travel time prediction model with minimal SPO loss function  $l_{SPO}(\hat{t}, t)$ . The prediction model would be determined by empirical risk minimization principle similar to Eq. (13), which leads to the following optimization problem:

$$\min_H \frac{1}{n} \sum_{i=1}^n l_{SPO}(H(f_i), t_i). \quad (17)$$

To overcome the difficulty in solving the above optimization problem, Elmachtoub and Grigas [12] provide a useful approximate loss function. Indeed, the SPO loss function  $l_{SPO}(\hat{t}, t)$  may not be continuous with  $\hat{t}$  because  $X^*(\hat{t})$  cannot be guaranteed to be continuous with respect to  $\hat{t}$ . However, the approximate surrogate loss function is convex and can be defined as

$$l_{SPO+}(\hat{t}, t) = \xi_{\Gamma}(t - 2\hat{t}) + 2\hat{t}^T X^*(t) - F^*(t). \quad (18)$$

In Eq. (18), the support function of feasible region  $\Gamma$ ,  $\xi_{\Gamma}(t) = \max_{x \in S} \{t^T x\}$  is convex in  $t$ .  $\xi_{\Gamma}(\cdot)$  is finite and for all  $t \in R^q$ , there is  $\xi_{\Gamma}(t) = -F^*(-t) = t^T x^*(-t)$ . Consequently, the optimization problem (17) has the following empirical risk counterpart:

$$\min_H \frac{1}{n} \sum_{i=1}^n l_{SPO+}(H(f_i), t). \tag{19}$$

This paper uses the linear prediction function  $H(f) = B * f, B \in R^{q \times p}$  to predict travel time. To improve accuracy and preserve convexity of the prediction model, we incorporate ridge penalty  $\Omega(B) = B_F^2/2$  into Eq. (19), where  $B_F$  denotes the Frobenius norm of matrix  $B$ . These presumptions lead to the following version of the risk model:

$$\min_{B \in R^{q \times p}} \frac{1}{n} \sum_{i=1}^n l_{SPO+}(Bf_i, t_i) + \lambda \Omega(B), \tag{20}$$

where  $\lambda \geq 0$  is a regularization parameter and selected by adjusting. The above optimization problem (20) is a convex optimization since both  $\Omega(B)$  and SPO + loss function  $l_{SPO+}$  are convex.

The classical gradient approach can be used to solve the SPO+ problem due to the convexity of (20). For convenience, let  $\phi_i(B) = l_{SPO+}(Bf_i, t_i) + \lambda \Omega(B)$ , then the optimization objective function is reformulated as  $\sum_{i=1}^n \phi_i(B)$ . As mentioned above,  $l_{SPO+}(\cdot, t)$  is convex for a given  $t$ , and  $\Omega(B)$  is a convex ridge penalty function, then the sum of  $\phi_i(B)$  is also convex. Furthermore, the sub-gradients of  $\phi_i(B)$  is  $2(X^*(t) - X^*(2\hat{t} - t))x_i^T + \lambda \nabla \Omega(B)$ , and according to Nemirovski et al. [41], we set the  $i$ -th update step-size from the reverse gradient direction to be  $\Upsilon_i = 2/\lambda(i + 2)$ . Given the travel information data  $(f_1, t_1), (f_2, t_2), \dots, (f_s, t_s)$ , problem (20) can be solved by the mini-batching gradient descent approach. The algorithm is as follows:

Algorithm 1:

Step 1. Initialize prediction matrix  $B_0 \in R^{q \times p}$  and the integer mini-batch size  $M, i = 0$ .

Step 2. Pick up samples  $(f_{1+iM}, t_{1+iM}), (f_{2+iM}, t_{2+iM}), \dots, (f_{(1+i)M}, t_{(1+i)M})$  and compute  $B_{i+1}$  as follows:

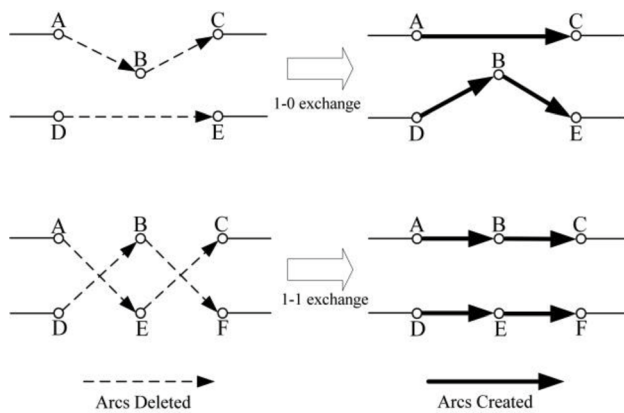
For  $j = 1 + iM, \dots, (1 + i)M$ :

$$\begin{aligned} \tilde{X}_j^i &= X^*(2B_i f_j - t_j) \\ \tilde{G}_j^i &= 2(X^*(t_j) - \tilde{X}_j^i) f_j^T + \lambda \nabla \Omega(B_i) \end{aligned}$$

Update  $B_{i+1}$  with  $\gamma_i = \frac{2}{\lambda(i+2)}$ :

$$\begin{aligned} G^i &= \frac{1}{M} \sum_{j=1+iM}^{(1+i)M} \tilde{G}_j^i \\ B_{i+1} &= B_i - \gamma_i G^i \\ i &= i + 1 \end{aligned}$$

Step 3. Stop if accuracy requirement is satisfied or all the samples are used.



**Fig. 2** exchange local search operators for the VRP

In Algorithm 1, we use the sample data successively, resulting in early stopping if there is large-scale sample data. Therefore, the batch data can also be constructed uniformly at random from a given sample set a standard application of the stochastic gradient descent approach. It is important to emphasize that the optimal solution of the LMP problem is required several times in each iteration of Algorithm 1. To guarantee efficiency, we need to choose a suitable algorithm for LMP to accelerate the convergent speed. This paper adopts a simulated annealing (SA) algorithm to compute  $X^*(\cdot)$  in each iteration step of Algorithm 1. Because the travel time and optimal routing have a mutual effect on each other, after implementing the prediction Algorithm 1, we leverage the idea of classical improvement heuristics of VRP to retrieve the final optimal routing.

Classical local search descent methods do not accept non-improvement moves at each iteration, whereas the SA algorithm does with specific probabilities. Therefore, the SA method can reach a satisfactory suboptimal solution much faster than local search descent methods. However, the SA algorithm is still based on a local search mechanism for iterating. We first present two commonly used exchange operators and the classical local search descent method of VRP.

Suppose a feasible routing solution for LMP have been already obtained,  $X = \{R_1, \dots, R_i, \dots, R_j, \dots, R_k\}$ , where  $R_i$

is the ordered set of customers serviced by driver  $i$ . The following procedure easily derives the initial feasible solution: let each customer being served by a driver, and then compute travel time-saving  $t_{0i} + t_{0j} - t_{ij}$  about customers  $i$  and  $j$  which are served by one vehicle from the central depot. These savings are sorted in decreasing order. Finally, this procedure merges customers corresponding to the highest saving without violating the capacity restriction until no further merges are possible.

Define a non-empty subset of  $R_i$  and  $R_j$  as  $S_1$  and  $S_2$ , respectively. An interchange between  $R_i$  and  $R_j$  is a replacement by  $R'_i = (R_i - S_1) \cup S_2$  and  $R'_j = (R_j - S_2) \cup S_1$ . Then we get a new neighboring solution  $X' = \{R_1, \dots, R'_i, \dots, R'_j, \dots, R_k\}$ . In general, the interchange is defined according to the threshold size of the subset. For example, if the size of  $S_1$  and  $S_2$  is less than  $\lambda$ , i.e.,  $|S_1| \leq \lambda$  and  $|S_2| \leq \lambda$ , the interchange is called  $\lambda$ -exchange. In this paper, we set  $\lambda = 1$ . Correspondingly, the search order of a given pair  $(R_i, R_j)$  has two standard operators described in Fig. 2. The 1–0 exchange wipes off a customer in a route and relocates the customer to another route. The 1–1 exchange swaps two customers in a routing pair.

For a given routing pair, the above exchange operators will generate a series of neighbors. In practical calculation, we can specify the searching order of neighbors following the permutation of routing pairs, which has a total number of  $k(k-1)/2$  different pairs:

$$(R_1, R_2), \dots, (R_1, R_k), (R_2, R_3), \dots, (R_{k-1}, R_k). \quad (21)$$

Following the above searching order of neighborhood, local search algorithms improve solution performance from the current  $X_t$  to another  $X_{t+1}$  in its neighborhood  $N(X_t)$  if the objective value  $F(X_{t+1})$  is less than  $F(X_t)$ . However, the SA algorithm accepts the improvement according to probabilities that are determined by a temperature parameter. The temperature parameter tends to zero following a deterministic cooling schedule. We adopt SA with a 1-interchange mechanism to solve  $X^*(\cdot)$  in Algorithm 1, and the SA is introduced as Algorithm 2:



Algorithm 2

Step 1: Initial a feasible solution  $X_1$ , search over the neighborhood  $N(X_1)$ , record the largest change  $\Delta F_{max}$  and smallest change  $\Delta F_{min}$  in objective function values. Let the starting temperatures  $T_s = \Delta F_{max}$  and final temperatures  $T_f = \Delta F_{min}$ . Define the total number of feasible exchanges in cycling is  $N_f$ . Set iteration number  $i = 1$ , the iteration temperature  $T_i = T_s$ , optimal solution  $X^* = X_1$ , and optimal temperature  $T^* = T_s$ .

Step 2: Select a solution  $X'$  from  $N(X_i)$  to search and compute difference  $\Delta = F(X_i) - F(X')$ .

If  $\Delta \geq 0$ , then set  $X_{i+1} = X'$ . If  $\Delta < 0$ , compute acceptable probability  $P_i = \exp(-\frac{\Delta}{T_i})$  and sample  $\theta$  randomly from  $[0,1]$ , set  $X_{i+1} = X'$  if  $P_i \geq \theta$ . If  $F(X_i) \leq F(X^*)$ , then  $X^* = X_{i+1}$  and  $T^* = T_i$ .

Otherwise,  $X_{i+1} = X_i$ .

Step 3: Update the temperatures according to

$$T_{i+1} = \frac{T_i}{1 + \beta_i T_i}, \quad \beta_i = \frac{T_s - T_f}{(N_f + \sqrt{i})nT_sT_f}$$

and set  $i = 1 + i$ .

Step 4: Stop if the optimal temperature  $T^*$  is close enough to  $T_f$ .

Algorithms 1 and 2 provide a detailed procedure for linear regression prediction and optimal routing of LMP. However, as we have said previously, the travel time is affected by the driver’s behavior, which is not involved in Algorithms 1 and 2. Therefore, we should make some adjustments to derive the optimal routing solution after getting the best linear regression  $Bf$  by Algorithms 1 and 2. The optimal routing solution of LMP is generated by Algorithm 3.

Algorithm 3

Step 1. Initial the regression  $H(f) = Bf$  by sample training as described in algorithm 1 and 2.

Step 2. Initial a feasible routing solution  $X_1$ , combine with other feature data to form  $f$ .

Step 3. Predict travel time by  $H(f) = Bf$ , then solve the LMP by algorithm 2. The objective value  $F(\cdot)$  of algorithm 2 is calculated according to a new travel time vector for the feature  $f$  is changed in each interchange search.

Step 4. Stop if the stopping criterion is met.

### Experimental design

This section presents numerical experiments to show the performance of the data-driven SPO+ framework for solving LMP. Specifically, we use Matlab R2017b to implement the programming of Algorithms 1–3. To provide suitable training samples, we generate experimental data according to Elmachtoub and Grigas [12]. Notice that there are  $n$  customers and one central depot, the dimension of the travel time vector is  $q = n \times (n - 1)$ . We firstly determine a matrix  $B^* \in R^{q \times p}$  whose entries is generated by Bernoulli distribution with probability 0.5, and then generate the training data as follows:

1. Generate the feature vector  $\hat{f}_i \in R^{p-1}$  by a multivariate Gaussian distribution with i.i.d. standard regular entries, i.e.,  $\hat{f}_i \sim N(0, I_{p-1})$ , where  $I_{p-1}$  is an identity matrix. Notice that

$\hat{f}_i$  is a part of the total feature vector  $f_i = (\hat{f}_i, f_i^k)$ ,  $f_i^k$  represents the driver’s feature.

2. The travel time vector  $t_{ij}$  is generated by employing a polynomial kernel function and is according to  $t_{ij} = \left( \left[ \left( \frac{1}{\sqrt{p}}(B^*f) + 3 \right)^{deg} + 1 \right] * \epsilon \right)_{(i-1)j+j}$ , where  $\epsilon$  is a multiplicative noise matrix whose entries are generated independently from the uniform distribution on  $[1 - \bar{\epsilon}, 1 + \bar{\epsilon}]$  for some given parameter  $\bar{\epsilon} \geq 0$ .

The parameter  $deg$  controls the amount of model misspecification. When  $deg = 1$ , the expected value of travel time  $t$  is indeed a linear function. When  $deg$  takes a large value, the traditional regression methods will be sensitive to outliers and have poor performance. In this experiment, we set  $deg = 2$  which will generate enough outliers to compare SPO+ framework against the normal loss approach.

Furthermore, we set  $f_i^k = \sqrt{|i+j-k|}$  which construct a larger expectation value  $t_{ij}$  if  $i+j$  is far away from  $k$ , and  $t_{ij}$  is small if  $i+j$  closes to  $k$ .

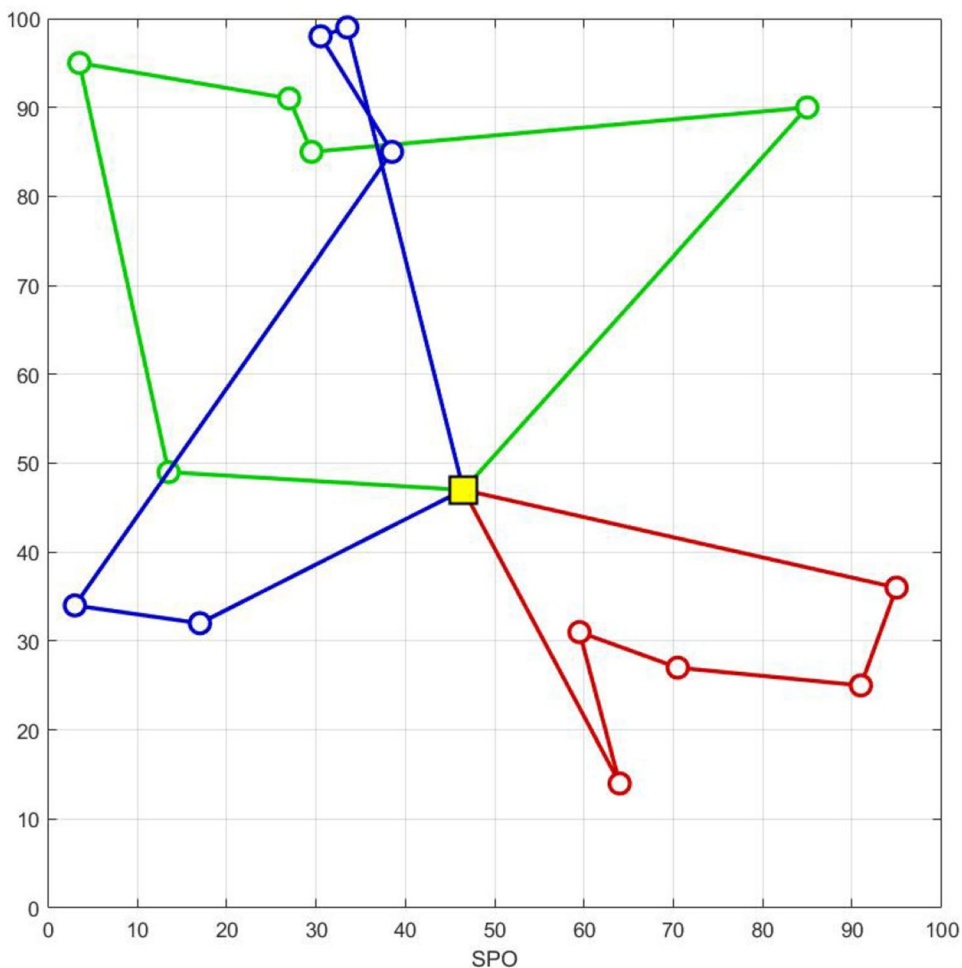
Following the generating process above, experiments are implemented whose specific procedure and parameters setting are described as follows. Assuming there are  $p = 5$  features, and the nodes-vehicles pair is (15, 3) in which further entry is a total number of customers and the latter is the number of total drivers. The customer demand is random generated as  $d_i = [24, 20, 20, 25, 24, 13, 16, 20, 25, 25, 16, 17, 22, 19, 15]$  and vehicle capacity vector is  $Q = [94, 108, 100]$ . We generate customers' positions in a 100\*100 region and suppose the initial depot is situated in the central position. In case (15, 5), the travel time dimension is  $q = 15 \times 14$ , and  $B$  has  $q \times p$  parameters to estimate. For the sample size that has a heavy impact on the practical performance, we choose the training set size  $S = 10000$ , i.e., for a driver, we generate  $s/k$  training data and let mini-batching be 100. The other parameters are set as follows: discounted operating cost  $\beta C = 10$ ,  $\bar{\epsilon} = 0.5$ , and the regularization parameter  $\lambda$  is selected by

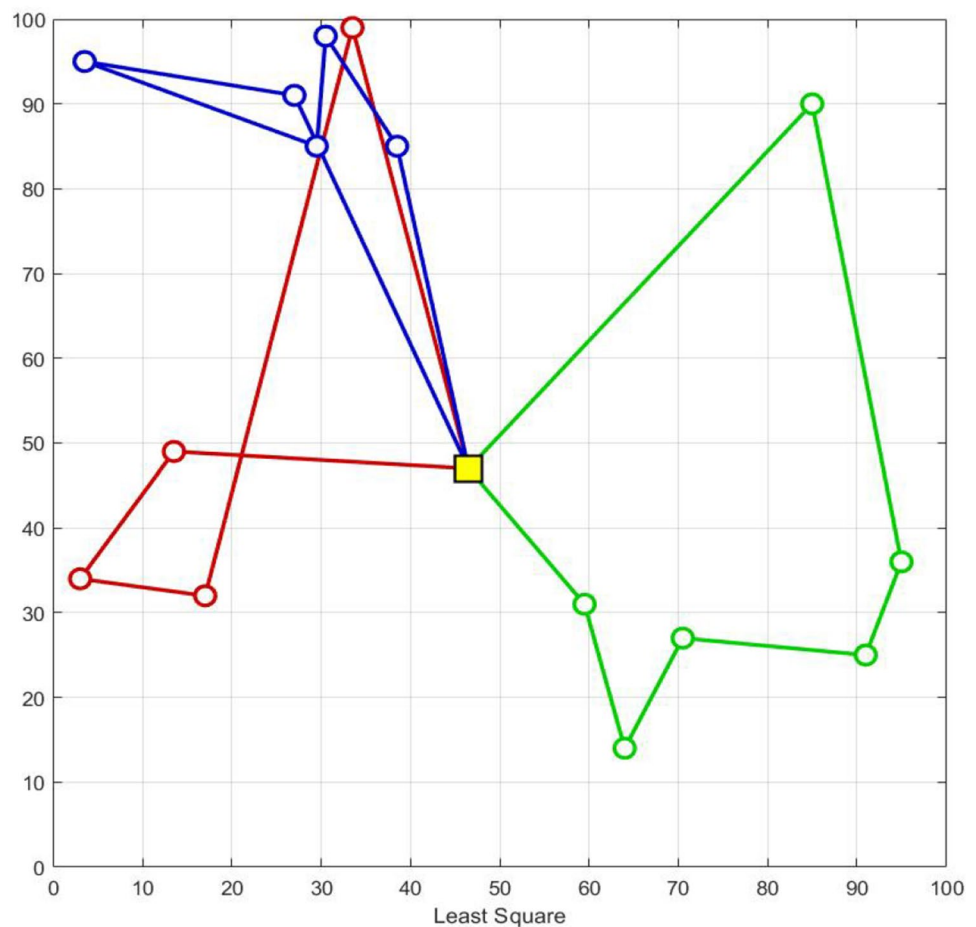
validating the ten different values evenly spaced between 0 and 1.

After training the prediction model by algorithms 1 and 2, we generate random testing data  $(f, t)$  and compute the optimal routing solution  $X^*$ . We compare our model with the general least square predict and optimization model, the expected optimization model. The results are summarized in Figs. 3, 4, 5, 6.

Figures 3, 4, 5 present the online platforms' detailed routing decisions by SPO, Least Square and expectation methods, respectively. In these figures, the customer depots represented by the circles, the square is the central service depot, and each polyline with the same color is the routing path of a given driver. We can see that the routing results of the three models have significant differences; even the driver visits the same customer set. For example, the top right corner customer point is connected with a different customer set. The SPO model assigns a driver to take charge of the bottom right five customers, which is in line with the results in the expectation model; however, comparing Fig. 3

Fig. 3 The LMP routing of SPO



**Fig. 4** The LMP routing of least square

and Fig. 5, it is clear that the service sequence of the SPO solution is quite different from that of the expected solution.

Furthermore, we make a thousand simulation that contains 1000 feature vectors for prediction. After predicting and optimizing, we compute the travel costs of three routing solutions according to actual travel time. Box plots in Fig. 6 draw the results. It is clear that the SPO model has a lower travel cost compare with the other two methods; the gap is about 5%. From the box plots, we can also find that the least square model has a similar performance as the expectation model, which indicates the traditional predict-then-optimize paradigm has a bottleneck, although the prediction model is effective in our numerical setting.

## Conclusion

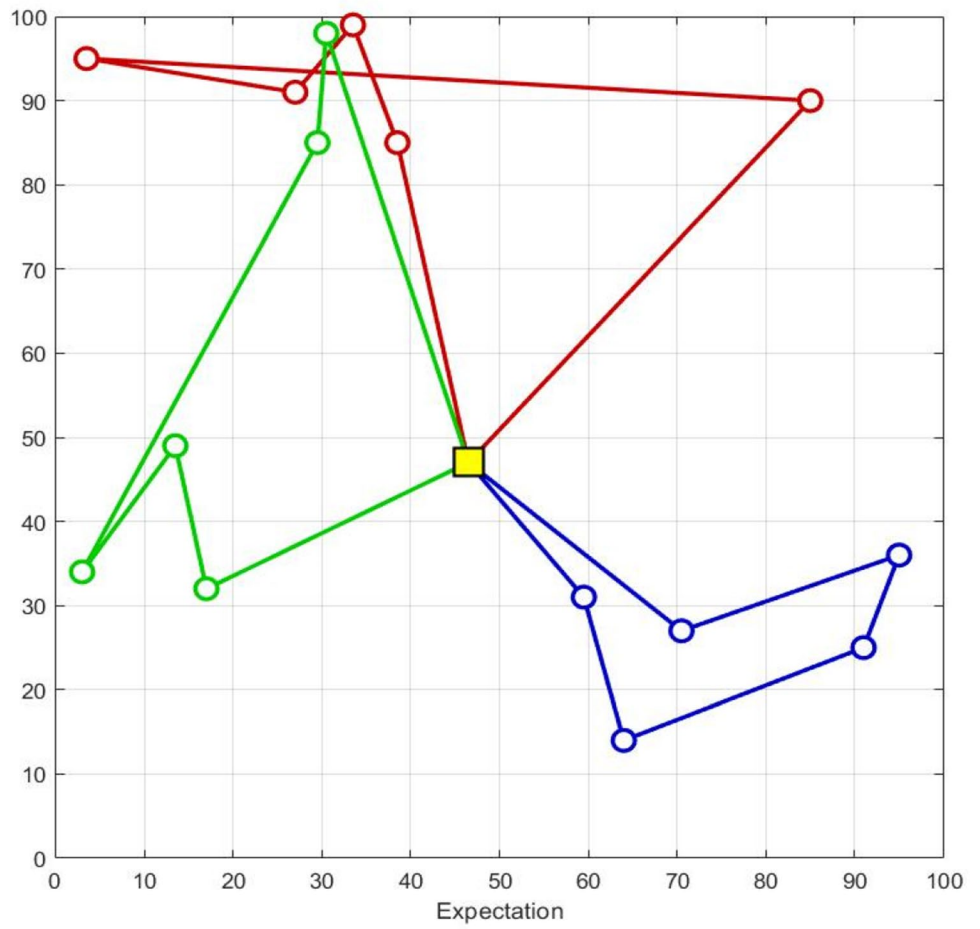
In this paper, a novel SPO framework is adopted to solve the emerging LMP of online food platforms. Our study mainly contains two innovations, the first one is that we provide a

completed data-driven framework to solve the LMP more effectively, the second one is that we consider the drivers' behavior in the model and propose the corresponding solving algorithms. The experimental results indicate that the proposed SPO framework provides a more effective delivery routing than the other methods.

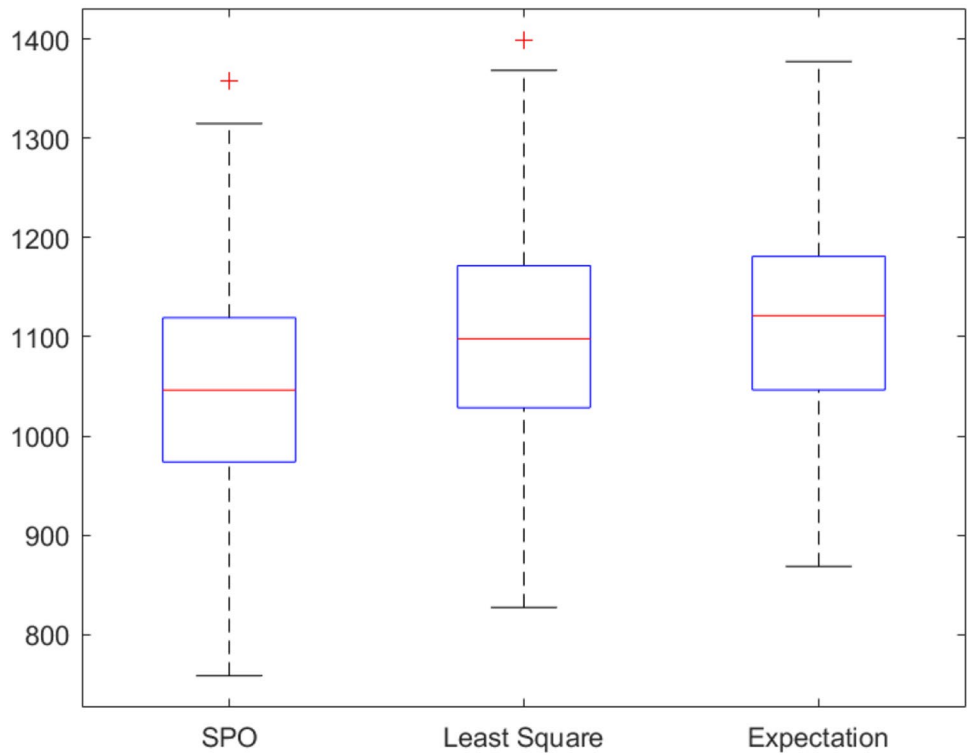
Prediction and optimization are two significant challenges in analyzing real world problems. In the era of big data, a new regulatory principle to decide is combing data from Multi-Dimension. Data-driven optimization methods, especially the combination of optimization models and machine learning methods, are causing more and more attention. This work provides an effective model for LMP, and the implementation framework can be embedded to the management system for the daily operation of food delivery platforms.

Naturally, this work has many further improvement directions for future studies. First, this paper does not consider the pickup problem, which will make the LMP more difficult to solve. Second, a faster and more effective algorithm should be proposed for other more complex conditions, such as

**Fig. 5** The LMP routing of expectation



**Fig. 6** The LMP routing results of SPO, least-square and expectation methods



unknown parameters in the constraints, nonlinear objective function, and so on.

**Acknowledgement** This work was supported by Beijing Intelligent Logistics System Collaborative Innovation Center Foundation under Grant BILSCIEC-2019KF-17, Fundamental Research Funds for Universities Affiliated to Beijing of Capital University of Economics and Business under Grant XRZ2020015, Humanities and Social Science Fund of Ministry of Education of China under Grant 18YJCZH247 and Shandong Social Science Planning Funds under Grant 18DGLJ01.

## Compliance with ethical standards

**Conflict of interest** On behalf of all authors, the corresponding author states that there is no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Liu S, He L, and Z.-J, Shen M (2020) On-time last mile delivery: order assignment with travel time predictors. *Manag Sci.* <https://doi.org/10.1287/mnsc.2020.3741>
- Wang H, Odoni A (2014) Approximating the performance of a “last mile” transportation system. *Transport Sci* 50(2):659–675
- Boyer KK, Tomas Hult G, Frohlich M (2003) An exploratory analysis of extended grocery supply chain operations and home delivery. *Integrat Manuf Syst* 14(8):652–663
- Boyer KK, Prud'homme AM, Chung W (2009) The last mile challenge: evaluating the effects of customer density and delivery window patterns. *J Bus Logist* 30(1):185–201
- Shapiro A, Dentcheva D, and Ruszczyński A (2014) Lectures on stochastic programming: modeling and theory. SIAM, Philadelphia, p 511. <https://doi.org/10.1137/1.9781611973433>
- Ben-Tal A, El Ghaoui L, and Nemirovski A (2009) Robust optimization. Princeton University Press, New Jersey, p 576. <https://press.princeton.edu/books/hardcover/9780691143682/robust-optimization>
- Bertsimas D, Gupta V, Kallus N (2018) Data-driven robust optimization. *Math Program* 167(2):235–292
- Delage E, Ye Y (2010) Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operat Res* 58(3):595–612
- Ferreira KJ, Lee BHA, Simchi-Levi D (2016) Analytics for an online retailer: demand forecasting and price optimization. *Manuf Serv Operat Manag* 18(1):69–88
- Ban G-Y, Karoui NE, Lim AEB (2018) Machine learning and portfolio optimization. *Manage Sci* 64(3):1136–1154
- Ban G-Y, Rudin C (2019) The big data newsvendor: practical insights from machine learning. *Operat Res* 67(1):90–108
- Elmachtoub AN, Grigas P (2020) Smart predict, then optimize. [arXiv:1710.08005v5](https://arxiv.org/abs/1710.08005v5)
- Toth P, Vigo D (2014) Vehicle routing: problems, methods, and applications. SIAM, Philadelphia, p 467. <https://doi.org/10.1137/1.9781611973594>
- Jiang L, Chang H, Zhao S, Dong J, Lu W (2019) A travelling salesman problem with carbon emission reduction in the last mile delivery. *IEEE Access* 7:61620–61627
- Zhou F, He Y, Zhou L (2019) Last mile delivery with stochastic travel times considering dual services. *IEEE Access* 7:159013–159021
- Dantzig GB, Ramser JH (1959) The truck dispatching problem. *Manage Sci* 6(1):80–91
- Bertsimas DJ, Ryzin GV (1991) A stochastic and dynamic vehicle routing problem in the euclidean plane. *Operat Res* 39(4):601–615
- Bertsimas DJ, Ryzin GV (1993) Stochastic and dynamic vehicle routing in the euclidean plane with multiple capacitated vehicles. *Operat Res* 41(1):60–76
- Campbell AM, Thomas BW (2008) Probabilistic traveling salesman problem with deadlines. *Transport Sci* 42(1):1–21
- Jaillet P, Qi J, Sim M (2016) Routing optimization under uncertainty. *Operat Res* 64(1):186–200
- Clarke G, Wright JW (1964) Scheduling of vehicles from a central depot to a number of delivery points. *Operat Res* 12(4):568–581
- Foster BA, Ryan DM (1976) An integer programming approach to the vehicle scheduling problem. *J Operat Res Soc* 27(2):367–384
- Renaud J, Boctor FF, Laporte G (1996) An improved petal heuristic for the vehicle routing problem. *J Oper Res Soc* 47(2):329–336
- Pisinger D, Ropke S (2007) A general heuristic for vehicle routing problems. *Comput Oper Res* 34(8):2403–2435
- Thompson PM, Psaraftis HN (1993) Cyclic transfer algorithm for multivehicle routing and scheduling problems. *Oper Res* 41(5):935–946
- Mladenović N, Hansen P (1997) Variable neighborhood search. *Comput Oper Res* 24(11):1097–1100
- Prins C (2004) A simple and effective evolutionary algorithm for the vehicle routing problem. *Comput Oper Res* 31(12):1985–2002
- Lysgaard J, Letchford AN, Eglese RW (2004) A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Math Program* 100(2):423–445
- Laporte G, Mercure H, Nobert Y (1986) An exact algorithm for the asymmetrical capacitated vehicle routing problem. *Networks* 16(1):33–46
- Fisher ML (1994) Optimal solution of vehicle routing problems using minimum K-trees. *Oper Res* 42(4):626–642
- Baldacci R, Christofides N, Mingozzi A (2008) An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Math Program* 115(2):351–385
- Baldacci R, Mingozzi A, Roberti R (2011) New route relaxation and pricing strategies for the vehicle routing problem. *Oper Res* 59(5):1269–1283
- Liyanage LH, Shanthikumar JG (2005) A practical inventory control policy using operational statistics. *Oper Res Lett* 33(4):341–348
- Chu LY, Shanthikumar JG, Shen Z-JM (2008) Solving operational statistics via a Bayesian analysis. *Oper Res Lett* 36(1):110–116
- Kleywegt AJ, Shapiro A, Homem-de-Mello T (2002) The sample average approximation method for stochastic discrete optimization. *SIAM J Optim* 12(2):479–502
- Schütz P, Tomaskard A, Ahmed S (2009) Supply chain design under uncertainty using sample average approximation and dual decomposition. *Eur J Oper Res* 199(2):409–419
- Chang M-S, Tseng Y-L, Chen J-W (2007) Transportation research part E. *Logist Transport Rev* 43(6):737–754

38. Bertsimas D, Sim M, Zhang M (2019) Adaptive distributionally robust optimization. *Manage Sci* 65(2):604–618
39. Yi-Hao K, Benjamin VR, Xiang Y (2009) Directed regression. In: *Advances in neural information processing systems*, 22 pp 889–897. <https://proceedings.neurips.cc/paper/2009/file/0c74b7f78409a4022a2c4c5a5ca3ee19-Paper.pdf>
40. Donti P, Amos B, Kolter JZ (2017) Task-based end-to-end model learning in stochastic optimization. In: *Advances in neural information processing systems*, 30 pp 5484–5494. <https://papers.nips.cc/paper/2017/file/3fc2c60b5782f641f76bcefc39fb2392-Paper.pdf>
41. Nemirovski A, Juditsky A, Lan G, Shapiro A (2009) Robust stochastic approximation approach to stochastic programming. *SIAM J Optim* 19(4):1574–1609