Electronic Research Archive

http://www.aimspress.com/journal/era

ERA, 31(4): 2265–2285.
DOI: 10.3934/era.2023116
Received: 04 October 2022
Revised: 17 January 2023
Accepted: 18 January 2023
Published: 24 February 2023

Research article

Tutorial on prescriptive analytics for logistics: What to predict and how to predict

Xuecheng Tian[1,†], Ran Yan[2,†,*], Shuaian Wang[3,†], Yannick Liu[3,†] and Lu Zhen[4,†]

[1] Department of Logistics & Maritime Studies, The Hong Kong Polytechnic University, Hung Hom, Hong Kong

[2] School of Civil & Environmental Engineering, Nanyang Technological University, Singapore

[3] Faculty of Business, The Hong Kong Polytechnic University, Hung Hom, Hong Kong

[4] School of Management, Shanghai University, Shanghai, China

† The authors contributed equally to this work.

* **Correspondence:** Email: ran-angela.yan@connect.polyu.hk.

Abstract: The development of the Internet of things (IoT) and online platforms enables companies and governments to collect data from a much broader spatial and temporal area in the logistics industry. The huge amount of data provides new opportunities to handle uncertainty in optimization problems within the logistics system. Accordingly, various prescriptive analytics frameworks have been developed to predict different parts of uncertain optimization problems, including the uncertain parameter, the combined coefficient consisting of the uncertain parameter, the objective function, and the optimal solution. This tutorial serves as the pioneer to introduce existing literature on state-of-the-art prescriptive analytics methods, such as the predict-then-optimize framework, the smart predict-then-optimize framework, the weighted sample average approximation framework, the empirical risk minimization framework, and the kernel optimization framework. Based on these frameworks, this tutorial further proposes possible improvements and practical tips to be considered when we use these methods. We hope that this tutorial will serve as a reference for future prescriptive analytics research on the logistics system in the era of big data.

Keywords: machine learning; predictive analytics; prescriptive analytics; optimization; logistics

1. Introduction

1.1. Background and motivating examples

Uncertainty is ubiquitous in logistics, such as uncertain travel time due to unexpected weather and traffic conditions, the fluctuating prices of delivery services due to the varying supply-demand relationship, and the random transportation demand due to the changes in economy and society [1]. Uncertainty is generally perceived as having negative effects on the logistics system, which increases running cost, decreases resource usage, and reduces customer satisfaction [1]. Therefore, an increasing number of logistics studies consider the uncertainty, aiming to mitigate the adverse effects of uncertainty on operations. For uncertain optimization problems in the logistics industry, we notice that the uncertainty can exist in different parts of the optimization problems. Some optimization problems have uncertainty in their objective functions, such as the routing problem (see Example 1 where the travel time in each arc is uncertain), and the energy-cost aware scheduling problem (see Example 2 where the energy price during each time period is uncertain). Other optimization problems have uncertainty in their constraints, such as problems with constraints established to fulfill a given level of service or uncertain demand (see Example 3 where the demand of each booking class is uncertain). Furthermore, it is also possible that uncertainty exists in both objective functions and constraints of optimization problems. To illustrate these observations, we show three examples in the logistics system which have uncertainty in different parts of their optimization problems. The first two examples have uncertainty in their objective functions, and the third example has uncertainty in its constraints.

Example 1. *Routing problem.*

Assume that there is a transport network denoted by $\mathcal{G} = (\mathcal{N}, \mathcal{S})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{S}$ is the set of arcs. Each arc $s \in \mathcal{S}$ has an uncertain travel time, denoted by c_s, and we define $\boldsymbol{c} := (c_1, ..., c_{|\mathcal{S}|})$. The objective is to decide a path on which to drive from origin $o \in \mathcal{N}$ to destination $d \in \mathcal{N}$ with the minimum travel time. Define $\boldsymbol{x} := (x_1, ..., x_{|\mathcal{S}|})$ as a binary decision vector, where x_s represents the decision variable that takes the value of one if arc s is traversed and zero otherwise. The mathematical model is as follows:

$$\min_{x \in \mathcal{X}} Z_{routing}(\boldsymbol{c}, \boldsymbol{x}) = \min_{x \in \mathcal{X}} \sum_{s \in \mathcal{S}} c_s x_s, \tag{1.1}$$

where $\mathcal{X}$ is a given set that describes the network constraints.

Example 2. *Energy-cost aware scheduling problem.*

Assume that $\mathcal{J}$ is the set of tasks, $\mathcal{R}$ is the set of available resources, and $\mathcal{T}$ is the set of time periods in equal length. Each task $j \in \mathcal{J}$ is specified by its duration d_j (an integer multiple of a time period), earliest starting time at the beginning of period e_j, latest ending time at the beginning of period l_j, and power usage p_j. Denote u_{jr} as the resource usage of task j for resource r, q_r as the available capacity of resource r, v_{jt} as the binary variable that takes the value of 1, only if task j starts at the beginning of time period t and zero otherwise. Furthermore, we require that each task is only scheduled once, and the machine can be scheduled to finish more than one task simultaneously. Assuming that y_t is the uncertain energy price during time period t, the objective is to minimize the total energy cost. Thus, we define $\boldsymbol{v}$ as a $|\mathcal{J}| \times |\mathcal{T}|$ matrix with elements $v_{jt}, j \in \mathcal{J}, t \in \mathcal{T}$ and $\boldsymbol{y} := (y_1, ..., y_{|\mathcal{T}|})$. The mathematical model is as follows:

$$\min_{\boldsymbol{v}} Z_{ener}(\boldsymbol{y}, \boldsymbol{v}) = \min_{\boldsymbol{v}} \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} v_{jt} (\sum_{t \leq t' < t + d_j} p_j y_{t'}) \tag{1.2}$$

subject to

$$\sum_{e_j \leq t \leq l_j - d_j} v_{jt} = 1 \; j \in \mathcal{J} \tag{1.3}$$

$$\sum_{j \in \mathcal{J}} \sum_{\max\{0, t-d_j\} < t' \leq t} u_{jr} v_{jt'} \leq q_r \; r \in \mathcal{R}, t \in \mathcal{T} \tag{1.4}$$

$$v_{jt} \in \{0, 1\} \; j \in \mathcal{J}, t \in \mathcal{T}. \tag{1.5}$$

Constraints (1.3) ensure that each task is scheduled only once from the earliest starting time to the latest ending time. Constraints (1.4) meet the resource requirement of the machine.

Although these two examples both have uncertainty in their objective functions, a noticeable difference between the formulation of objective functions in Examples 1 and 2 is that a coefficient p_j exists in objective function (1.2), in addition to decision variables and uncertain parameters. We finally show another example with uncertainty in its constraints.

Example 3. *Static network revenue management.*

Denote $\mathcal{K}$ as the set of booking classes, g_k as the decision variable representing the available capacity that the freight company intends to reserve for bookings of class k over the finite planning horizon, c_k as the operating cost of reserving a booking of class k, f_k as the revenue of completing a booking of class k, h_k as the amount of capacity used by a booking of class k, Q as the amount of available capacity, D_k as the uncertain demand for bookings of class k, γ_k as the penalty cost if the real demand of class k cannot be met because of the shortage in allocated capacity, and ξ_k as the recourse variable that represents the shortage amount of capacity for bookings of class k. We define $\boldsymbol{g} := (g_1, ..., g_{|\mathcal{K}|})$, $\boldsymbol{D} := (D_1, ..., D_{|\mathcal{K}|})$, and $\boldsymbol{\xi} := (\xi_1, ..., \xi_{|\mathcal{K}|})$. The objective is to determine the optimal reserved capacities for bookings of different classes to maximize the expected profit, i.e., the difference between expected revenue and the expected penalty cost, over the finite planning horizon. The two-stage mathematical model is as follows:

[Stage 1]

$$\max_{\boldsymbol{g}} Z_{static}(\boldsymbol{g}, \boldsymbol{D}) = \max_{\boldsymbol{g}} \Big\{ \mathbb{E}[\pi(\boldsymbol{g}, \boldsymbol{D})] - \sum_{k \in \mathcal{K}} c_k g_k \Big\} \tag{1.6}$$

subject to

$$\sum_{k \in \mathcal{K}} h_k g_k \leq Q \tag{1.7}$$

$$g_k \geq 0 \; k \in \mathcal{K}. \tag{1.8}$$

[Stage 2]

$$\pi(\boldsymbol{g}, \boldsymbol{D}) = \max_{\boldsymbol{\xi}} \sum_{k \in \mathcal{K}} \Big[\min(g_k, D_k) f_k - \gamma_k \xi_k \Big] \tag{1.9}$$

subject to

$$g_k + \xi_k \geq D_k \; k \in \mathcal{K} \tag{1.10}$$

$$\xi_k \geq 0 \; k \in \mathcal{K}. \tag{1.11}$$

Constraints (1.7) ensure that the accepted bookings do not exceed the available capacity. Constraints (1.10) ensure that the sum of the capacity allocated to bookings and the unsatisfied demand should be no smaller than the uncertain demand.

1.2. Literature review

To model and solve optimization problems with uncertainty, different frameworks have been developed. Bertsimas and Koduri [2] divided these frameworks into two main categories, according to whether they take data as a primitive or not. The first category contains relevant literature in stochastic programming [3] and robust programming [4, 5] that does not take data as a primitive. These methods generally preset distributions for uncertain parameters without using any real data. However, it is unrealistic for decision-makers to know the ground-truth distributions of uncertain parameters.

Instead, because we are able to collect and store huge amounts of data, thanks to the development of internet technologies, frameworks in the second category emerge, which contain relevant studies taking data as a primitive to characterize uncertainty. These frameworks can further be classified into two subcategories according to what kind of data they use, including historical data of uncertain parameters themselves and other auxiliary data that can be used to predict uncertain parameters. Frameworks in the first subcategory only use historical data to approximate the scenarios or distributions of uncertain parameters, but do not consider other auxiliary data that might be useful to predict the uncertain parameters, such as the sample average approximation (SAA) framework [6] and the data-driven distributionally robust optimization framework [7, 8]. Frameworks in the second subcategory apply various machine learning (ML) techniques to predict uncertain parameters by leveraging, not only their historical data, but also other related auxiliary data. This paper focuses on introducing the state-of-the-art frameworks in the second subcategory.

The advancement of business analytics techniques in the second subcategory is attributed to the development of the Internet of things (IoT) and online platforms, enabling companies and governments to collect data from a much broader spatial and temporal area [9]. For example, on-demand ride-hailing companies, such as Uber and Lyft, have stored millions of records taken by passengers around the globe since their establishment, which can help them develop smarter dispatch and pricing algorithms to achieve a more cost-effective match between supply and demand in a dynamic environment [9]. More specifically, according to the classification by He et al. [9], data for logistics studies generally comes from the private sector, the public sector, and other sources. Private sector data comes from private transportation and logistics service providers [10, 11], social media and map service platforms [12, 13], and emerging micromobility service providers [14]. Public sector data mainly comes from government agencies [15] and public transit system operators [16, 17]. Other sources include nongovernmental organizations [18], field research [19, 20], and third-party platforms. The most common type of data used in logistics studies include origin-destination demand of passengers and customers [21–23], retailer sales data across different outlets [24, 25], and real-world road network data [26–28].

Figure 1 depicts the workflow of business analytics [9]. The workflow is motivated by the business problem, which consists of the collection, preprocessing, and interpretation of the data, the selection and refinement of predictive analytics methods, and the modeling for decision making in prescriptive analytics. Common business analytics scenarios of applying big data techniques to the logistics industry include, but are not limited to, driving and commuting [22,29–31], freight transport [32–35], last-mile delivery [23, 36–38], manufacturing [39], and public services (e.g., healthcare service delivery, efficient distribution of food, water, and humanitarian aid, and military industrial logistics) [19,40,41]. In the end, the business analytics is aimed at prescribing sound decisions; that is, we generally focus on the paths called prescriptive analytics. In order to derive decisions from data, there are two different paths for prescriptive analytics, namely indirect path and direct path.

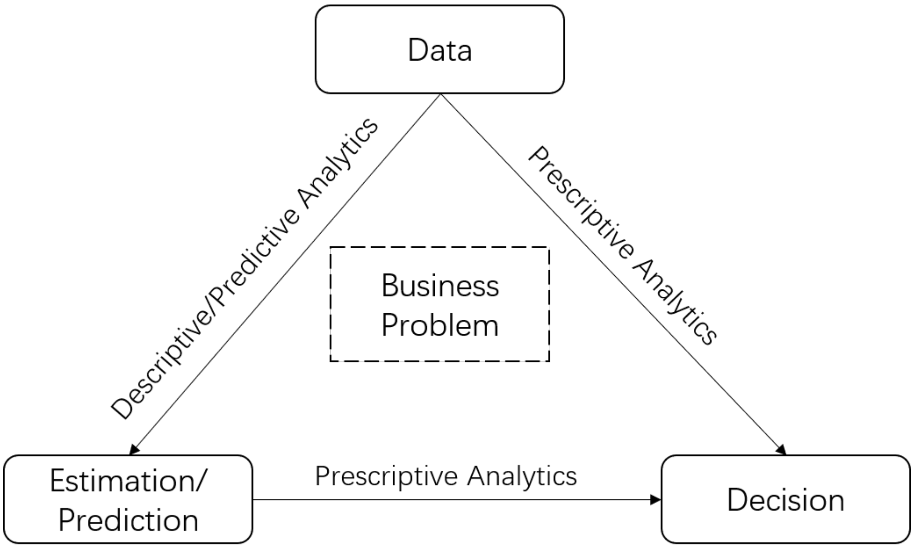

Figure 1. A general workflow of business analytics [9].

The indirect path is to first derive estimations or predictions by predictive analytics, and estimations and predictions are then served as inputs to the downstream decision process. The indirect path involves data, prediction, and decision, and is generally termed predict-then-optimize (PO) framework or estimate-then-optimize framework. Currently, many applications of smart technologies and big data analytics methods have demonstrated potential promise in enhancing the efficiency and effectiveness in various logistics operations and transportation systems [42]. During the estimation and prediction stage, statistical analysis, such as Poisson process [43, 44], kernel density models [45, 46], and continuous approximation [27, 47], is often used to characterize the demand process of the logistics system. We have also witnessed an increased use of econometric and statistical learning tools in logistics studies to explore the relationship between demand and various covariates [17, 22]. Furthermore, a wide range of predictive models, ranging from classical statistical methods (e.g., the popular autoregressive integrated moving average (ARIMA) [24]) to novel ML methods (e.g., decision trees, support vector machines, random forests, and neural networks) [15,48,49], have been used in logistics studies. Empirically, Gunasekaran et al. [50] have analyzed how big data and predictive analytics assimilation affects supply chain and organizational performance. Their findings suggested that connectivity and information sharing under the mediation effect of top management commitment are positively related to big data predictive analytics acceptance. At last, during the optimization stage, the predicted values or distributions of the unknown parameters will be plugged into the downstream optimization problems. Corresponding literature has been thoroughly reviewed by Chung [42], Nguyen et al. [51], and Wang et al. [52].

Although the PO framework is easy to understand and implement, there is always a mismatch between the objectives of the predictive model and the optimization model. Sometimes, a good prediction may not lead to a good decision [2]. As an alternative to indirect path, direct path is a recent trend in prescriptive analytics, which goes directly from data to decision and contains many advanced frameworks, such as the smart predict-then-optimize (SPO) framework [53], the weighted sample average approximation (w-SAA) framework [54, 55], the empirical risk minimization (ERM) framework [55, 56], and the kernel optimization (KO) framework [2, 46, 54, 55]. These frameworks are rarely reviewed and compared in the existing literature.

1.3. Summary

Whichever path these prescriptive analytics frameworks take, their ultimate goal is to prescribe optimal decisions through predicting one of the three parts of the uncertain optimization problem by ML methods, namely, the uncertain parameter, the objective function, or the optimal solution. Yan et al. [57] further proposed that the combined coefficient consisting of uncertain parameters, such as the $\sum_{t\le t'<t+d_j} p_j y_{t'}$ part in Example 2, can also be predicted. This kind of prediction is attributed to the structural feature of the optimization problem, and can take any form, such as polynomial or exponential expressions. Therefore, we summarize that there are four parts that can be predicted by prescriptive analytics frameworks, including the uncertain parameter, the combined coefficient, the objective function, and the optimal solution. Accordingly, regarding the three examples shown above, the parts that can be predicted for each example are shown in Table 1.

Table 1. The parts that can be predicted for each example.

Example/Part to be predicted	Uncertain parameter	Combined coefficient	Objective function	Optimal solution
Example 1	Yes	No	Yes	Yes
Example 2	Yes	Yes	Yes	Yes
Example 3	Yes	No	Yes	Yes

Remark 1. *For Example 1, there are three parts that can be predicted, including the uncertain parameter* $\boldsymbol{c}$*, the objective function* $Z_{routing}(\boldsymbol{c}, \boldsymbol{x})$*, and the optimal solution* $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} Z_{routing}(\boldsymbol{c}, \boldsymbol{x})$*. For Example 2, there are four parts that can be predicted, including the uncertain parameter* y_t*, the combined coefficient* $\sum_{t\le t'<t+d_j} p_j y_{t'}$*, the objective function* $Z_{ener}(\boldsymbol{y}, \boldsymbol{v})$*, and the optimal solution* $\boldsymbol{v}^* = \arg\min_{\boldsymbol{v}} Z_{ener}(\boldsymbol{y}, \boldsymbol{v})$*. For Example 3, there are three parts that can be predicted, including the uncertain parameter* D_k*, the objective function* $Z_{static}(\boldsymbol{g}, \boldsymbol{D})$*, and the optimal solution* $\boldsymbol{g}^* = \arg\min_{\boldsymbol{g}} Z_{static}(\boldsymbol{g}, \boldsymbol{D})$*. Therefore, the difference lies in that the coefficient prediction can only be used in uncertain models with structure like Example 2. The common thing is that parameter prediction, objective prediction, and optimizer prediction can all be applied to uncertain problems regardless of where the uncertainty in the optimization model is.*

The huge amount of data acts as catalysts for the development of prescriptive analytics, giving rise to various methods of predicting different parts of uncertain optimization problems. This tutorial makes the following contributions: First, we classify the prediction targets in prescriptive analytics into four categories, including the uncertain parameter, the combined coefficient, the objective function, and the optimal solution. Second, regarding different prediction targets, we review the corresponding state-of-the-art prescriptive analytics frameworks, which are rarely summarised and compared in the existing literature. Third, regarding different prescriptive analytics frameworks, we further propose possible improvements and practical tips to be considered when these frameworks are used in practice. Accordingly, we use the three examples to show how these methods can be used in real applications when and where appropriate.

2. Uncertain parameter and coefficient prediction methods

2.1. The PO frameworks

If we have access to auxiliary data related to the uncertain parameters in optimization problems, the most common method for solving uncertain problems is to predict uncertain parameters using ML models, which turns uncertain problems into easy-to-solve deterministic problems. If combined coefficients exist in the model, i.e., the polynomial $\sum_{t\le t'<t+d_j} p_j y_{t'}$ in Example 2, we can also predict the combined coefficients directly.

Take Example 1, for instance. Assume that we have collected the travel time on each arc of the past n days, denoted by $c_s^i, i \in \{1, ..., n\}, s \in \mathcal{S}$, where we define $\boldsymbol{c}^i := (c_1^i, ..., c_{|S|}^i)$, as well as the auxiliary feature vector associated with the travel time, including features such as whether it is a working day, rainfall, temperature, and wind, amongst others, denoted by $\boldsymbol{a}^i \in \mathcal{A} \subset \mathbb{R}^{d_a}, i \in \{1, ..., n\}$. Given the new feature vector of today based on weather forecast, denoted by $\boldsymbol{a}^0$, our goal is to find a good path; that is, a path with minimum travel time. If we randomly pick a day, the features (i.e., auxiliary data) and travel times are random, denoted by $(\tilde{\boldsymbol{a}}, \tilde{\boldsymbol{c}})$, and their joint distribution is denoted by $\mathbb{D}$. Given the new feature vector $\tilde{\boldsymbol{a}} = \boldsymbol{a}^0$, $\tilde{\boldsymbol{c}}$ is still a random variable, whose distribution is drawn from $\mathbb{D}$, denoted by $\mathbb{D}_{\boldsymbol{a}^0}$. Consequently, given the new feature vector $\boldsymbol{a}^0$, we should solve the following model for Example 1:

$$\min_{\boldsymbol{x}\in\mathcal{X}} \mathbb{E}_{(\tilde{\boldsymbol{a}},\tilde{\boldsymbol{c}})\sim\mathbb{D}}[Z_{routing}(\tilde{\boldsymbol{c}}, \boldsymbol{x})|\tilde{\boldsymbol{a}} = \boldsymbol{a}^0] = \min_{\boldsymbol{x}\in\mathcal{X}} \mathbb{E}_{\tilde{\boldsymbol{c}}\sim\mathbb{D}_{\boldsymbol{a}^0}}[Z_{routing}(\tilde{\boldsymbol{c}}, \boldsymbol{x})]. \tag{2.1}$$

Because the objective function is linear in the uncertain parameter of Example 1, we can further obtain that

$$\min_{\boldsymbol{x}\in\mathcal{X}} \mathbb{E}_{\tilde{\boldsymbol{c}}\sim\mathbb{D}_{\boldsymbol{a}^0}}[Z_{routing}(\tilde{\boldsymbol{c}}, \boldsymbol{x})] = \min_{\boldsymbol{x}\in\mathcal{X}} Z_{routing}(\mathbb{E}_{\tilde{\boldsymbol{c}}\sim\mathbb{D}_{\boldsymbol{a}^0}}[\tilde{\boldsymbol{c}}], \boldsymbol{x}). \tag{2.2}$$

In order to solve $\min_{\boldsymbol{x}\in\mathcal{X}} Z_{routing}(\mathbb{E}_{\tilde{\boldsymbol{c}}\sim\mathbb{D}_{\boldsymbol{a}^0}}[\tilde{\boldsymbol{c}}], \boldsymbol{x})$ with the conditional expectation $\mathbb{E}_{\tilde{\boldsymbol{c}}\sim\mathbb{D}_{\boldsymbol{a}^0}}[\tilde{\boldsymbol{c}}]$, the PO framework is a typical method, which firstly predicts the uncertain parameter $\tilde{\boldsymbol{c}}$ given the new observation $\boldsymbol{a}^0$ by developing an ML model F^* based on the dataset $\{(\boldsymbol{a}^i, \boldsymbol{c}^i)\}_{i=1}^n$, and then plugs the prediction $\hat{\boldsymbol{c}} = F^*(\boldsymbol{a}^0)$ into the optimization problem to derive decisions. Considering that the cost is a continuous prediction target, we can use mean squared error (MSE) loss to train F^*, which is expressed as follows:

$$L_{MSE} = \frac{1}{n} \sum_{i=1}^{n} \left\|\boldsymbol{c}^i - F^*(\boldsymbol{a}^i)\right\|_2^2. \tag{2.3}$$

Assuming that we have infinitely many data, under mild conditions, we can obtain the best estimate

$$\hat{\boldsymbol{c}} = F^*(\boldsymbol{a}^0) = \mathbb{E}(\tilde{\boldsymbol{c}}|\tilde{\boldsymbol{a}} = \boldsymbol{a}^0) = \mathbb{E}_{\tilde{\boldsymbol{c}}\sim\mathbb{D}_{\boldsymbol{a}^0}}[\tilde{\boldsymbol{c}}]. \tag{2.4}$$

The conditional expectation $\mathbb{E}_{\tilde{\boldsymbol{c}}\sim\mathbb{D}_{\boldsymbol{a}^0}}[\tilde{\boldsymbol{c}}]$ is approximated by the estimated $\hat{\boldsymbol{c}}$, and the optimization problem of Example 1 is successfully solved by using the conditional mean $\hat{\boldsymbol{c}}$.

2.1.1. The w-SAA method

A general assumption underlying the PO framework is that the objective function is linear in the uncertain parameter. However, if the objective function (1.1) is not linear in the uncertain parameter, the PO framework is not able to solve the original problem. For example, considering that a student is going to take an exam that starts in 60 minutes, meaning that a route is good only when its overall travel time is less or equal than 60 minutes, the original problem (1.1) should be

$$\min_{\boldsymbol{x}\in\mathcal{X}} \mathbb{E}_{\tilde{\boldsymbol{c}}\sim\mathbb{D}_{\boldsymbol{a}^0}} \mathbb{I}(\sum_{s\in\mathcal{S}} \tilde{c}_s x_s > 60), \tag{2.5}$$

where $\mathbb{I}(\cdot)$ is an indicator function which takes the value of one if the condition is true and zero otherwise, and the objective is to minimize the probability that the chosen route is not good given the new observation $\boldsymbol{a}^0$. In this case, the objective function is not linear in the uncertain parameter, so $\min_{\boldsymbol{x}\in\mathcal{X}} \mathbb{E}_{\tilde{\boldsymbol{c}}\sim\mathbb{D}_{\boldsymbol{a}^0}} \mathbb{I}(\sum_{s\in\mathcal{S}} \tilde{c}_s x_s > 60) \neq \min_{\boldsymbol{x}\in\mathcal{X}} \mathbb{I}(\sum_{s\in\mathcal{S}} \mathbb{E}_{\tilde{\boldsymbol{c}}\sim\mathbb{D}_{\boldsymbol{a}^0}}[\tilde{c}_s] x_s > 60)$. To be more specific, assume that there are two paths A and B. Path A's travel time is 60 minutes, and path B's travel time is 59 minutes with 50% chance or 61 minutes with 50% chance. If a student goes to take an exam that starts in one hour, these two paths are very different. If we solve $\min_{\boldsymbol{x}\in\mathcal{X}} \mathbb{E}_{\tilde{\boldsymbol{c}}\sim\mathbb{D}_{\boldsymbol{a}^0}} \mathbb{I}(\sum_{s\in\mathcal{S}} \tilde{c}_s x_s > 60)$, the optimal solution should select path A only. However, if we solve $\min_{\boldsymbol{x}\in\mathcal{X}} \mathbb{I}(\sum_{s\in\mathcal{S}} \mathbb{E}_{\tilde{\boldsymbol{c}}\sim\mathbb{D}_{\boldsymbol{a}^0}}[\tilde{c}_s] x_s > 60)$, the optimal solution would be both path A and path B. Therefore, the PO framework cannot solve the original problem when the original objective function is not linear in the uncertain parameter. In order to remedy this issue, we can take the following methods: the w-SAA method and the quantile-regression based method.

Instead of predicting certain values of uncertain parameters, Bertsimas and Kallus [54] proposed a w-SAA framework to predict the conditional distribution of the uncertain parameter, given the new observation. Under this framework, take objective function (2.5) as an example, given a new observation $\boldsymbol{a}^0$, the conditional distribution of $\tilde{\boldsymbol{c}}$ is approximated empirically as $w(\boldsymbol{a}^i, \boldsymbol{a}^0), i \in \{1, .., n\}$, where $w(\boldsymbol{a}^i, \boldsymbol{a}^0)$ measures the similarity between the historical example $\boldsymbol{a}^i$ and the new observation $\boldsymbol{a}^0$, where its format depends on the ML model we use. If we use a k-nearest neighbor (kNN) model, $w(\boldsymbol{a}^i, \boldsymbol{a}^0) = 1/k$ if $\boldsymbol{a}^i$ is a kNN of $\boldsymbol{a}^0$ and zero otherwise, and $w(\boldsymbol{a}^i, \boldsymbol{a}^0)$ can be seen as an approximation of the conditional distribution of $\tilde{\boldsymbol{c}}$ given $\boldsymbol{a} = \boldsymbol{a}_0$, namely, $\mathbb{D}_{\boldsymbol{a}^0}$; that is, the approximate distribution of $\tilde{\boldsymbol{c}}$ has n scenarios $\boldsymbol{c}^1, \boldsymbol{c}^2, ..., \boldsymbol{c}^n$ with probabilities $w(\boldsymbol{a}^1, \boldsymbol{a}^0), w(\boldsymbol{a}^2, \boldsymbol{a}^0), ..., w(\boldsymbol{a}^n, \boldsymbol{a}^0)$ (and it is possible that some probabilities $w(\boldsymbol{a}^i, \boldsymbol{a}^0)$ are 0, meaning that the approximate distribution of $\tilde{\boldsymbol{c}}$ has less than n scenarios). After obtaining the conditional marginal distribution $\mathbb{D}_{\boldsymbol{a}^0}$, the approximation of objective function (2.5) is as follows:

$$\min_{\boldsymbol{x}\in\mathcal{X}} \sum_{i=1}^{n} w(\boldsymbol{a}^i, \boldsymbol{a}^0) \mathbb{I}(\boldsymbol{c}^i \boldsymbol{x} > 60). \tag{2.6}$$

2.1.2. The quantile-regression based global method

If we aim to predict the conditional distribution of an uncertain parameter, the w-SAA method is a local ML method, which predicts the conditional distribution by measuring closeness to existing data [2, 58]. This method in some sense throws away some data that is not close to the observation, and so it needs a lot of data to work well [2]. As an alternative, Wang and Yan [59] proposed a quantile-regression based global method to take all data into account when estimating a single-dimensional parameter. Because the quantile-regression based global method is stemmed from the traditional regression model, for a particular arc $s \in \mathcal{S}$ in Example 1, if we first assume using the linear regression model $F_s(\boldsymbol{a}) = \boldsymbol{w}_s^\top \boldsymbol{a}$ as the predictive model, where $\boldsymbol{w}_s$ is a $d_a \times 1$ vector (recall that d_a is the dimension of the feature vector), we have

$$\boldsymbol{w}_s^* \in \arg\min_{\boldsymbol{w}_s} \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{w}_s^\top \boldsymbol{a}^i - c_s^i)^2 \tag{2.7}$$

and $F_s^*(\boldsymbol{a}) = \boldsymbol{w}_s^{*\top} \boldsymbol{a}$. Next, given the new observation $\boldsymbol{a}^0$, we can obtain $\hat{c}_s = F_s^*(\boldsymbol{a}^0) = \boldsymbol{w}_s^{*\top} \boldsymbol{a}^0$ and $\hat{\boldsymbol{c}} := (\hat{c}_1, ..., \hat{c}_{|\mathcal{S}|})$. However, by minimizing the sum of squared errors using the traditional regression model, we are estimating the conditional mean $\mathbb{E}_{\tilde{\boldsymbol{c}} \sim \mathbb{D}_{\boldsymbol{a}^0}}[\tilde{\boldsymbol{c}}]$ instead of the conditional distribution $\tilde{\boldsymbol{c}} \sim \mathbb{D}_{\boldsymbol{a}^0}$, which may not work well when the objective function is not linear in the uncertain parameters. Alternatively, we can introduce a parameter $\alpha \in [0, 1]$ and obtain $\boldsymbol{w}_s^{\alpha *}$ by solving

$$\min_{\boldsymbol{w}_s^\alpha} \frac{1}{n} \sum_{i=1}^{n} \Big[(1 - \alpha) \max\left((\boldsymbol{w}_s^\alpha)^\top \boldsymbol{a}^i - c_s^i, 0\right) + \alpha \max\left(c_s^i - (\boldsymbol{w}_s^\alpha)^\top \boldsymbol{a}^i, 0\right)\Big]. \tag{2.8}$$

By minimizing the above weighted sum of over- and under-estimation errors, we are estimating the 100αth percentile of the uncertain parameter. For Example 1, we can estimate the 5th, 15th, ..., 95th percentile of $\tilde{c}_s$. The distribution of $\tilde{c}_s | \boldsymbol{a}^0$ is thus approximately $\Pr(\tilde{c}_s = (\boldsymbol{w}_s^{\alpha *})^\top \boldsymbol{a}^0) = \frac{1}{10}$, $\alpha = 0.05, 0.15, ..., 0.95$. Next, there are two cases to consider to solve objective function (2.5) after $\tilde{c}_s | \boldsymbol{a}^0$ is obtained under each percentile. First, if we assume that the travel times of different arcs are highly correlated, we should solve

$$\min_{\boldsymbol{x} \in \mathcal{X}} \frac{1}{10} \sum_{\alpha = 0.05, 0.15, ..., 0.95} \Big[\mathbb{I}((\boldsymbol{a}^0)^\top \boldsymbol{w}^{\alpha *} \boldsymbol{x}) > 60\Big], \tag{2.9}$$

where $\boldsymbol{w}^{\alpha *} := (\boldsymbol{w}_1^{\alpha *}, ..., \boldsymbol{w}_{|\mathcal{S}|}^{\alpha *})^\top$ is a $d_a \times |\mathcal{S}|$ matrix. Otherwise, if we assume that the travel times of different arcs follow independent distributions, we need to define $\boldsymbol{w}_{sample}^* := (\boldsymbol{w}_1^{\alpha' *}, ..., \boldsymbol{w}_{|\mathcal{S}|}^{\alpha' *})^\top$ as a $d_a \times |\mathcal{S}|$ matrix, where α' is randomly sampled from {0.05, 0.15, ..., 0.95} for each arc. Considering that there are $|\mathcal{S}|$ arcs and each arc has 10 percentile values, there would be $10^{|\mathcal{S}|}$ possible combinations for $\boldsymbol{w}_{sample}^*$. Because it is time-consuming to find all possible combinations in a large network, we resample λ times from all combinations and denote each combination as $\boldsymbol{w}_{sample}^{\epsilon *}$. We should then solve

$$\min_{\boldsymbol{x} \in \mathcal{X}} \frac{1}{\lambda} \sum_{\epsilon = 1}^{\lambda} \Big[\mathbb{I}((\boldsymbol{a}^0)^\top \boldsymbol{w}_{sample}^{\epsilon *} \boldsymbol{x}) > 60\Big] \tag{2.10}$$

to prescribe final decisions.

2.2. The SPO frameworks

For the frameworks mentioned above, the loss function used to train the ML models generally only focuses on minimizing the prediction error, such as the MSE loss function (2.3), which does not consider the impact of the predictions on the downstream optimization problems, leading to suboptimal solutions. Therefore, a more natural and appropriate method is to plug the optimization problem into the training process of the ML models, which is generally termed SPO framework. The commonly used loss function, designed for measuring decision error, under this framework for parameter prediction of Example 1, namely SPO loss, is expressed as follows:

$$L_{SPO} = \frac{1}{n}\sum_{i=1}^{n}\left[Z_{routing}(\boldsymbol{c}^i, \boldsymbol{x}^*(\hat{\boldsymbol{c}}^i)) - Z_{routing}(\boldsymbol{c}^i, \boldsymbol{x}^*(\boldsymbol{c}^i))\right], \tag{2.11}$$

where $\boldsymbol{x}^*(\hat{\boldsymbol{c}}^i) = \arg\min_{x\in X} Z_{routing}(\hat{\boldsymbol{c}}^i, \boldsymbol{x})$ and $\boldsymbol{x}^*(\boldsymbol{c}^i) = \arg\min_{x\in X} Z_{routing}(\boldsymbol{c}^i, \boldsymbol{x})$.

In order to synthesize predictive and prescriptive techniques to create ML systems that learn to make decisions based on empirical data, the resulting composite models often employ constrained optimization as a neural network layer, and are trained in an end-to-end method. Therefore, most SPO-related studies use feed-forward neural networks (NNs) with deep learning architectures composed of a sequence of layers [60]. However, training ML models using SPO loss might be computationally difficult because of the nonconvex and discontinuous characteristics of the SPO loss function for combinatorial optimization problems. This is because the discrete and discontinuous solution space prevents the learning problem from easily differentiating the decision loss over the predicted values. Consequently, it is infeasible to pass back the gradients to inform the predictive model regarding how it should adjust its weights to improve the decision quality of the prescribed solutions [61]. To overcome this problem, Wilder et al. [62] added a quadratic regularization term to the objective function of the relaxed form of the combinatorial problem, but this method can only be applied to combinatorial problems with a totally unimodular matrix. Ferber et al. [61] strengthened this method by employing a cutting-plane solution approach, which tightened the continuous relaxation by adding constraints removing fractional solutions. Furthermore, instead of computing the real decision loss by directly solving the combinatorial problem during the training process, some studies have designed a class of surrogate loss functions based on a sub-gradient, such as Elmachtoub and Grigas [53] and Mandi et al. [63]. For these discussed approaches, a common issue is that these methods all need to repeatedly solve the (possibly relaxed) optimization problem, bringing a huge burden on the computational efficiency. In contrast, Mulamba et al. [64] used a noise contrastive approach by viewing sub-optimal solutions as noise examples and caching them, which replaced optimization calls with a look-up table in the solution cache, so as to improve the training efficiency.

Furthermore, some studies begin to train decision trees to obtain personalized decision from a finite set of possible options instead of focusing only on the prediction error. Kallus [65] trained trees with a loss function, maximizing the effectiveness of the predictions rather than minimizing the prediction errors. Bertsimas et al. [66] studied a similar treatment recommendation problem, but adopted a weighted loss function to combine prediction and decision error. Elmachtoub et al. [67] considered a more general class of decision-making problems that could involve a large number of decisions represented by a general feasible region. To train decision trees under SPO loss, they proposed a tractable methodology called SPOTs. They claimed that SPOTs could benefit from the interpretability of decision trees, allowing for an interpretable segmentation of a set of contextual features with different optimal solutions to the optimization problem of interest. In a recent study, Kallus and Mao [68] also studied how to fit the node splitting policies in contextual stochastic optimization problems to directly minimize the optimization costs.

No matter what method we use, i.e., the PO frameworks and the SPO frameworks, or local ML methods and global ML methods, our goal is to predict a perfect value or a perfect distribution of the uncertain parameter to help us prescribe an optimal solution that is near to the full-information perfect solution. Recalling that we can also predict the combined coefficient in the objective function, these frameworks can be further applied to the combined coefficient prediction. The only difference between the prediction of a parameter and the prediction of a combined coefficient is that the output value of the ML model for the combined coefficient prediction should be computed beforehand, according to the structure of the expression. Take Example 2, for instance, where the prediction target is changed from the unit energy price during time period t, y_t, to the total energy cost of task j starting from time period t, $\sum_{t\leq t'<t+d_j} p_j y_{t'}$. Because there are J tasks, and considering that each task has its own earliest starting time e_j, latest ending time l_j, and working duration d_j, we thus need to train $\sum_{j\in\mathcal{J}}(d_j-e_j-l_j)$ ML models (since we assume for each task and for each feasible starting time period, there is a corresponding predictor) or a multi-output regression model if we are going to predict $\sum_{t\leq t'<t+d_j} p_j y_{t'}$. This indicates that the combined coefficient prediction may lead to more computational burdens.

In summary, parameter and coefficient prediction methods are the most popular in prescriptive analytics. Among different prescriptive analytics frameworks, the predict-then-optimize framework is easy to implement, whereas we can use various ML models to predict the conditional mean of uncertain parameters or coefficients in linear objective functions, or adopt the local w-SAA method or global quantile-regression model to predict the conditional distribution of uncertain parameters or coefficients in non-linear objective functions. However, because those frameworks mentioned above neglect the impact of predictions on downstream decisions, the SPO frameworks are thus proposed, whose tractability and scalability are two major obstacles to be solved for non-convex and discontinuous combinatorial problems.

3. Objective function prediction methods

As mentioned above, when we predict the uncertain parameter, we can use the PO or SPO frameworks to predict a conditional expectation if the objective function is linear in the uncertain parameter, or we can use the w-SAA framework or the quantile-regression based global method to predict a conditional distribution of the uncertain parameter if the objective function is non-linear in the uncertain parameter. Furthermore, it is worth noting that the w-SAA and the quantile-regression based global method can also be seen as methods to approximate the objective function although they do not take the perspective as predicting the objective [54]. The objective prediction is a recent trend in prescriptive analytics [54], which generally takes local learning methods. However, because local learning methods predict by measuring the closeness to existing data, whereas global learning methods predict by choosing a functional form of the prediction that minimizes some loss functions on existing data, the latter methods perform better with less data, extrapolate better to outliers, and perform better in higher dimensions [2]. Bertsimas and Koduri [2], thus, proposed a global ML method to predict the objective function, which has never been done before in the literature. This section focuses on their method of objective function prediction, and proposes a more general method based on their method.

3.1. The kernelized method

Recall that we can use the linear regression to predict the conditional expectation of $\tilde{\boldsymbol{c}}$, and $\boldsymbol{A}$ is the matrix with rows $\boldsymbol{a}^i$ for Example 1. Assume that $\boldsymbol{A}^\top \boldsymbol{A}$ is invertible, the optimal solution of optimization problem (2.7) takes the following form:

$$\boldsymbol{w}_s^* = (\boldsymbol{A}^\top \boldsymbol{A})^{-1} \boldsymbol{A}^\top \boldsymbol{c}_s, \tag{3.1}$$

where $\boldsymbol{c}_s := (c_s^1, ..., c_s^n)$ is the vector of historical target values. Therefore, given the new observation $\boldsymbol{a}^0$, the prediction of c_s (note that c_s is not an element in $\boldsymbol{c}_s$) for arc $s \in \mathcal{S}$ is

$$\mathbb{E}[c_s | \boldsymbol{a} = \boldsymbol{a}^0] \approx \boldsymbol{w}_s^{*\top} \boldsymbol{a}^0 = (\boldsymbol{c}_s)^\top \boldsymbol{A} (\boldsymbol{A}^\top \boldsymbol{A})^{-1} \boldsymbol{a}^0. \tag{3.2}$$

Alternatively, if we aim not to predict c_s, but the objective function $Z_{routing}(\boldsymbol{c}, \boldsymbol{x}) = \sum_{s \in \mathcal{S}} c_s x_s$ by finding some functions $\boldsymbol{w}(\boldsymbol{x})$ such that $\mathbb{E}[Z_{routing}(\boldsymbol{c}, \boldsymbol{x}) | \boldsymbol{a}] \approx \boldsymbol{w}(\boldsymbol{x})^\top \boldsymbol{a}$, we should compute

$$\min_{\boldsymbol{w}(\boldsymbol{x})} \frac{1}{n} \sum_{i=1}^{n} (Z_{routing}(\boldsymbol{c}^i, \boldsymbol{x}) - \boldsymbol{w}(\boldsymbol{x})^\top \boldsymbol{a}^i)^2. \tag{3.3}$$

The optimal solution of optimization problem (3.3) would be

$$\boldsymbol{w}^*(\boldsymbol{x}) = (\boldsymbol{A}^\top \boldsymbol{A})^{-1} \boldsymbol{A}^\top Z_{routing}(\boldsymbol{C}, \boldsymbol{x}), \tag{3.4}$$

and the approximation of $\mathbb{E}[Z_{routing}(\boldsymbol{c}, \boldsymbol{x}) | \boldsymbol{a} = \boldsymbol{a}^0]$ would be

$$\mathbb{E}[Z_{routing}(\boldsymbol{c}, \boldsymbol{x}) | \boldsymbol{a} = \boldsymbol{a}^0] \approx \boldsymbol{w}^*(\boldsymbol{x})^\top \boldsymbol{a}^0 = Z_{routing}(\boldsymbol{C}, \boldsymbol{x})^\top \boldsymbol{A} (\boldsymbol{A}^\top \boldsymbol{A})^{-1} \boldsymbol{a}^0, \tag{3.5}$$

where $\boldsymbol{C}$ is a matrix with rows $\boldsymbol{c}^i$, and $Z_{routing}(\boldsymbol{C}, \boldsymbol{x})$ is the vector $(Z_{routing}(\boldsymbol{c}^1, \boldsymbol{x}), ..., Z_{routing}(\boldsymbol{c}^n, \boldsymbol{x}))$. Following the method of using regression to predict the objective function and to generalize the approach to non-linear predictions, Bertsimas and Koduri [2] further used kernel tricks to predict the objective function. For Example 1, they denote the approximate objective function by $h(\boldsymbol{a}^i, \boldsymbol{x}) \in \mathcal{H}$, where $\mathcal{H}$ is the Hilbert space defined by a positive-definite kernel function $K(\boldsymbol{a}^i, \boldsymbol{a}^j)$ (see Definition 1 in Bertsimas and Koduri [2]), and the function (3.3) should be computed as:

$$\min_{h(\cdot, \boldsymbol{x}) \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (Z_{routing}(\boldsymbol{c}^i, \boldsymbol{x}) - h(\boldsymbol{a}^i, \boldsymbol{x}))^2 + \sigma \sum_{i=1}^{n} (h(\boldsymbol{a}^i, \boldsymbol{x}))^2, \tag{3.6}$$

which is computationally tractable, thanks to the representer theorem (see Proposition 1 in Bertsimas and Koduri [2]); here $\sigma \sum_{i=1}^{n} (h(\boldsymbol{a}^i, \boldsymbol{x}))^2$ denotes the regularization term used to prevent overfitting. According to the representer theorem, the optimal solution of optimization problem (3.6) must take the form

$$h(\boldsymbol{a}^i, \boldsymbol{x}) = \sum_{j=1}^{n} \mu^j(\boldsymbol{x}) K(\boldsymbol{a}^j, \boldsymbol{a}^i), \tag{3.7}$$

where $\mu^j(\boldsymbol{x}) \in \mathbb{R}$ is a function with respect to the decision vector $\boldsymbol{x}$, and $K(\boldsymbol{a}^i, \boldsymbol{a}^j)$ is the kernel function. Plugging (3.7) into (3.6) and following the same procedures as using regression to predict the objective function, and given the new observation $\boldsymbol{a}^0$, the objective function of Example 1 $Z_{routing}(\boldsymbol{c}, \boldsymbol{x})$ can be approximated by

$$[Z_{routing}(\boldsymbol{c}, \boldsymbol{x})|\boldsymbol{a} = \boldsymbol{a}^0] \approx h(\boldsymbol{a}^0, \boldsymbol{x}) = K(\boldsymbol{A}, \boldsymbol{a}^0)^\top(\hat{\boldsymbol{K}} + \alpha n I)^{-1} Z_{routing}(\boldsymbol{C}, \boldsymbol{x}), \tag{3.8}$$

where $K(\boldsymbol{A}, \boldsymbol{a}^0)$ is the vector $(K(\boldsymbol{a}^1, \boldsymbol{a}^0), ..., K(\boldsymbol{a}^n, \boldsymbol{a}^0))$, $\hat{\boldsymbol{K}}$ is the $n \times n$ kernel matrix with components $\hat{K}_{ij} = K(\boldsymbol{a}^i, \boldsymbol{a}^j)$, and I is the $n \times n$ identity matrix. After we obtain the predicted objective function, as shown in (3.8) given a new observation $\boldsymbol{a}^0$, we then minimize it following constraints $\boldsymbol{x} \in \mathcal{X}$ to obtain decisions.

3.2. The global objective function prediction method using a general function form

Because the kernel function $K(\boldsymbol{a}^i, \boldsymbol{a}^j)$ only considers the auxiliary features $\boldsymbol{a}$, we propose a more general case, which assumes that the predicted objective function for Example 1, denoted by $g(\boldsymbol{a}, \boldsymbol{x})$, maintains the original structure, and is as follows:

$$g(\boldsymbol{a}, \boldsymbol{x}) = \sum_{l=1}^{n} \theta_l(\boldsymbol{a}) Z_{routing}(\boldsymbol{c}^l, \boldsymbol{x}), \tag{3.9}$$

where $\theta_l(\boldsymbol{a}) \in \mathbb{R}$. Under this prediction, the decision loss function over the solution space should be

$$\sum_{i=1}^{n} \int [g(\boldsymbol{a}^i, \boldsymbol{x}) - Z_{routing}(\boldsymbol{c}^i, \boldsymbol{x})] d\boldsymbol{x}, \tag{3.10}$$

whose minimization is computationally intractable because we do not know the ground-truth distributions of $\boldsymbol{x}$. Practically, we only have empirical data points and their optimal solutions are $\boldsymbol{x}^{*j} = \arg\min_{\boldsymbol{x} \in \mathcal{X}} Z_{routing}(\boldsymbol{c}^j, \boldsymbol{x})$, $j = 1, ..., n$. Then, loss function (3.10) can be empirically approximated as:

$$\begin{aligned} L_{obj} &= \sum_{i=1}^{n} \sum_{j=1}^{n} (g(\boldsymbol{a}^i, \boldsymbol{x}^{*j}) - Z_{routing}(\boldsymbol{c}^i, \boldsymbol{x}^{*j})) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} (\sum_{l=1}^{n} \theta_l(\boldsymbol{a}^i) Z_{routing}(\boldsymbol{c}^l, \boldsymbol{x}^{*j}) - Z_{routing}(\boldsymbol{c}^i, \boldsymbol{x}^{*j})). \end{aligned} \tag{3.11}$$

Furthermore, we should add a regularization term to prevent overfitting:

$$\begin{aligned} L_{obj} = &\sum_{i=1}^{n} \sum_{j=1}^{n} (\sum_{l=1}^{n} \theta_l(\boldsymbol{a}^i) Z_{routing}(\boldsymbol{c}^l, \boldsymbol{x}^{*j}) - Z_{routing}(\boldsymbol{c}^i, \boldsymbol{x}^{*j})) \\ &+ \sigma \sum_{i=1}^{n} \sum_{l=1}^{n} \left(\theta_l(\boldsymbol{a}^i)\right)^2. \end{aligned} \tag{3.12}$$

Then, our goal is to determine $\theta_l^*(\boldsymbol{a})$ $(l = 1, ..., n)$ by solving $\min_{\boldsymbol{\theta}} L_{obj}$. Though $\theta_l(\boldsymbol{a})$ can take any form, for simplicity, we assume that $\theta_l(\boldsymbol{a}) = \boldsymbol{\theta}_l^\top \boldsymbol{a}$, where $\boldsymbol{\theta}$ is a vector with the same dimension of $\boldsymbol{a}$.

Now, the minimization of loss function (3.12) is a regression problem, where there are $n \times n$ records, indexed by $\{(1,1), ..., (i,j), ..., (n,n)\}$. For record (i,j), the target is denoted by $Z_{routing}(\boldsymbol{c}^i, \boldsymbol{x}^{*j})$, and it has $n \times d_a$ features (recall that $\boldsymbol{a}$ has d_a features), indexed by $\{(1,1), ..., (l,d'), ..., (n,d_a)\}$, where the value of feature (l,d') is denoted by $a^i_{d'} \times Z_{routing}(\boldsymbol{c}^l, \boldsymbol{x}^{*j})$. Therefore, we can use Ridge regression, whose regularization term is also in quadratic form, to minimize loss function (3.12).

In summary, regarding the methods for predicting the objective function, Bertsimas and Koduri [2] proposed the first global method that predicts the objective function in a functional form using kernel tricks. Considering that the kernel method does not maintain the original structure of the objective function, we propose a more general method, which deserves investigation and comparison in future studies.

4. Optimal solution prediction methods

4.1. The ERM method

For prescriptive analytics frameworks, in addition to predicting the uncertain parameter, the combined coefficient, and the objective function, a more direct way is to predict the optimal solution, as our ultimate goal is to prescribe a solution that is near the perfect solution under the condition that the uncertain information is known. Ban and Rudin [56] proposed two ERM algorithms to predict the optimal solution. For example, the ERM approach to solving Example 1 with auxiliary data is as follows:

$$\min_{\boldsymbol{x}(\cdot)\in\mathcal{L},\{\boldsymbol{x}:\mathcal{A}\to\mathbb{R}^{|\mathcal{S}|}\}} \hat{R}(\boldsymbol{x}(\boldsymbol{a}); (\boldsymbol{a}^i, \boldsymbol{c}^i)_{i=1}^n) = \min_{\substack{\boldsymbol{x}(\cdot)\in\mathcal{L} \\ \boldsymbol{x}(\boldsymbol{a})\in\mathcal{X}}} \frac{1}{n}\sum_{i=1}^{n} Z_{routing}(\boldsymbol{c}^i, \boldsymbol{x}(\boldsymbol{a}^i)), \tag{4.1}$$

where the decision now is a function $\boldsymbol{x}(\cdot)$ that maps the feature space $\mathcal{A}$ to reals, $\hat{R}$ is called the empirical risk of function $\boldsymbol{x}(\cdot)$ with respect to the dataset $\{(\boldsymbol{a}^i, \boldsymbol{c}^i)\}_{i=1}^n$, and we need to specify the function class $\mathcal{L}$ and enforce $\boldsymbol{x}(\boldsymbol{a}) \in \mathcal{X}$ to ensure that each training data record meets the network constraints. We note that it is possible that, for a given new observation $\boldsymbol{a}^0$, the prescribed decision may not follow the network constraints, namely $\boldsymbol{x}(\boldsymbol{a}^0) \notin \mathcal{X}$; therefore, we can set the prescribed decision as the nearest neighbour of $\boldsymbol{x}(\boldsymbol{a}^0)$ in $\mathcal{X}$, where we may need to solve a programming model $\min_{\boldsymbol{x}'\in\mathcal{X}} \|\boldsymbol{\epsilon}\|$, where $\boldsymbol{x}' = \boldsymbol{x}(\boldsymbol{a}^0) + \boldsymbol{\epsilon}$, and $\boldsymbol{\epsilon} \in \mathbb{R}^{|\mathcal{S}|}$ is the decision variable. Consider that we apply linear decision rules to predict the optimal solution of the form

$$\mathcal{L} = \{\boldsymbol{x} : \mathcal{A} \to \mathbb{R}^{|\mathcal{S}|} : \boldsymbol{x}(\boldsymbol{a}) = \boldsymbol{X}\boldsymbol{a}\}, \tag{4.2}$$

where $\boldsymbol{X}$ is a $|\mathcal{S}| \times d_a$ matrix with rows $\boldsymbol{x}^s = (x_1^s, .., x_{d_a}^s)$, $s = 1, ..., |\mathcal{S}|$. By using this linear form, the ERM problem (4.1) is as follows:

$$\min_{\boldsymbol{x}(\boldsymbol{a})=\boldsymbol{X}\boldsymbol{a}} \hat{R}(\boldsymbol{x}(\boldsymbol{a}); (\boldsymbol{a}^i, \boldsymbol{c}^i)_{i=1}^n) = \min_{\boldsymbol{X}\boldsymbol{a}\in\mathcal{X}} \frac{1}{n}\sum_{i=1}^{n} Z_{routing}(\boldsymbol{c}^i, \boldsymbol{X}\boldsymbol{a}^i). \tag{4.3}$$

To prevent overfitting, we can add a regularization term to (4.3) as follows:

$$\min_{\boldsymbol{X}\boldsymbol{a}\in\mathcal{X}} \Big[\frac{1}{n}\sum_{i=1}^{n} Z_{routing}(\boldsymbol{c}^i, \boldsymbol{X}\boldsymbol{a}^i) + \sigma \sum_{j=1}^{d_a}\sum_{s\in\mathcal{S}} (x_j^s)^2\Big]. \tag{4.4}$$

Therefore, when we use the linear form to estimate the optimal solution, the learning task is to find the best $x_j^s,\ s = 1, ..., |\mathcal{S}|,\ j = 1, ..., d_a$, by solving (4.4). After we obtain $\boldsymbol{X}^*$, given a new observation $\boldsymbol{a}^0$, the prescribed decision is thus $\boldsymbol{X}^*\boldsymbol{a}^0$.

4.2. The kernelized method

Furthermore, following the general formulation of the ERM approach, some studies have used kernel tricks to estimate the function that prescribes the optimal solution. Ban and Rudin [56] proposed an approach to predict the optimal solution by using the kernel optimization method, but it can only be applied to the newsvendor problem. Notz and Pibernik [55] proposed a kernelized ERM approach for the flexible capacity management problem, and proved its performance guarantees. Bertsimas and Koduri [2] proposed a general method to use kernel functions to predict the optimal solution. Taking Example 1, for instance, when using kernel tricks to predict the optimal solution, we restrict each $x^s(\boldsymbol{a}) \in \boldsymbol{x}(\boldsymbol{a}), s \in \mathcal{S}$, in optimization problem (4.1) to be in a reproducing kernel Hilbert space $\mathcal{H}$, which is associated with a kernel K. Then, the empirical regularized kernelized version of (4.1) is as follows:

$$\min_{\substack{x^1(\cdot),...,x^{|\mathcal{S}|}(\cdot)\in\mathcal{H}\\ x^1(\boldsymbol{a}),...,x^{|\mathcal{S}|}(\boldsymbol{a})\in\mathcal{X}}} \Big[\frac{1}{n}\sum_{i=1}^{n} Z_{routing}\big(\boldsymbol{c}^i; x^1(\boldsymbol{a}^i), ..., x^{|\mathcal{S}|}(\boldsymbol{a}^i)) + \sigma\sum_{s\in\mathcal{S}}\sum_{i=1}^{n}(x^s(\boldsymbol{a}^i))^2\Big]. \tag{4.5}$$

According to the conclusion of Bertsimas and Koduri [2], the optimal solution to the optimization problem (4.5) takes the form

$$x^s(\boldsymbol{a}) = \sum_{i=1}^{n}\mu_i^s K(\boldsymbol{a}^i, \boldsymbol{a}), s \in \mathcal{S}, \tag{4.6}$$

where μ is the solution to

$$\min_{\substack{\mu^1,...,\mu^{|\mathcal{S}|}\in\mathbb{R}^n\\ \hat{\boldsymbol{K}}\mu^1,...,\hat{\boldsymbol{K}}\mu^{|\mathcal{S}|}\in\mathcal{X}}} \Big[\frac{1}{n}\sum_{i=1}^{n} Z_{routing}\Big(\boldsymbol{c}^i; (\hat{\boldsymbol{K}}\boldsymbol{\mu}^1)_i, ..., (\hat{\boldsymbol{K}}\boldsymbol{\mu}^{|\mathcal{S}|})_i\Big) + \sigma\sum_{s\in\mathcal{S}}(\boldsymbol{\mu}^s)^\top\hat{\boldsymbol{K}}\boldsymbol{\mu}^s\Big], \tag{4.7}$$

where $\boldsymbol{\mu}^s = (\mu_1^s, ..., \mu_n^s)$, and $\hat{\boldsymbol{K}}$ is the $n \times n$ kernel matrix with components $\hat{K}_{ij} = K(\boldsymbol{a}^i, \boldsymbol{a}^j)$. Now, after specifying kernel functions, the decision variables of (4.7) are all μ_i^ss. After we obtain $\mu_i^{s*}, i \in \{1, ..., n\}, s \in \mathcal{S}$, given a new observation $\boldsymbol{a}^0$, the prescribed decision for arc $s \in \mathcal{S}$ is calculated as $x^s(\boldsymbol{a}^0) = \sum_{i=1}^n \mu_i^{s*} K(\boldsymbol{a}^i, \boldsymbol{a}^0)$.

In summary, ERM models and kernelized methods are all possible approaches to optimal solution prediction, whose performance guarantees have been shown in existing literature, such as Ban and Rudin [55], Bertsimas and Kallus [54], and Bertsimas and Koduri [2]. For all these methods, we need to note that the prescribed solution may violate the problem constraints, so post adjustments of prescribed solutions may be needed.

5. Conclusions and future research directions

This study summarizes existing literature on prescriptive analytics methods in the logistics system. We first point out that four parts in the optimization problems can be predicted, namely, the uncertain parameter, the combined coefficient, the objective function, and the optimal solution. The predictions of the uncertain parameter and the combined coefficient are the most common topics in existing literature on prescriptive analytics in the logistics system, which takes the indirect path from data to decision via prediction. Among these methods for parameter and coefficient prediction, the PO framework is the easiest to implement, whereas the SPO framework focuses more on the decision quality, but may be computationally intractable. It is worth noting that, if the objective function is not linear in the uncertain parameter, the w-SAA method is an alternative to predict the conditional distribution by using local learning methods, and the quantile-regression based method is another alternative that takes global data into account. Furthermore, the prediction of objective function and optimal solution takes the direct path, which goes from data to decision by choosing a functional form of the prediction that minimizes some loss functions on existing data. The methods for predicting the objective function and the optimal solutions are rooted in ERM algorithms, where the most commonly used functional form is the kernel function for its good applicability in handling nonlinearities. Uncertainties are ubiquitous in logistics problems, where these prescriptive analytics frameworks may work well in providing sound decisions. For different uncertain optimization problems, we do not know the best method; instead, this paper discusses a few existing alternatives, and proposes possible improvements, which constitute an arsenal of prescriptive analytics frameworks to be considered when and where appropriate.

Apart from improvements regarding different prescriptive analytics frameworks, we further propose the following future research directions. First, as stated in this tutorial, integrating learning and optimization in prescriptive analytics for logistics needs tailored learning algorithms that consider the structural characteristics of downstream optimization problems. In order to achieve optimal prescriptive targets, we may need to propose new methodologies and tools for both ML and optimization parts. Second, starting from proposing new methodologies of prescriptive analytics frameworks, we wish to validate their values by applying them to real industrial problems. In this way, when and how prescriptive analytics can improve decision-making can be empirically investigated. At last, the development of prescriptive analytics frameworks can also stimulate the collection of data in the industry. Investigating what kind of data we need, and examining the influence of data quality, can further promote better decision-making.

Acknowledgments

The authors thank the three reviewers for their valuable comments.

Conflict of interest

The authors declare there is no conflicts of interest.

References

1. W. Wang, Y. Wu, Is uncertainty always bad for the performance of transportation systems, *Commun. Transp. Res.*, **1** (2021), 100021. https://doi.org/10.1016/j.commtr.2021.100021
2. D. Bertsimas, N. Koduri, Data-driven optimization: A Reproducing Kernel Hilbert Space approach, *Oper. Res.*, **70** (2021), 454–471. https://doi.org/10.1287/opre.2020.2069

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

4. A. Ben-Tal, L. E. Ghaoui, A. Nemirovski, *Robust Programming*, Princeton University Press, Princeton, 2009.

5. D. Bertsimas, D. B. Brown, C. Caramanis, Theory and applications of robust optimization, *SIAM Rev.*, **53** (2011), 464–501. https://doi.org/10.1137/080734510

6. A. J. Kleywegt, A. Shapiro, T. Homem-de Mello, The sample average approximation for stochastic discrete optimization, *SIAM J. Optim.*, **12** (2002), 479–502. https://doi.org/10.1137/S1052623499363220

7. D. Bertsimas, V. Gupta, N. Kallus, Data-driven robust optimization, *Math. Program.*, **167** (2018), 235–292. https://doi.org/10.1007/s10107-017-1125-8

8. E. Delage, Y. Ye, Distributionally robust optimization under moment uncertainty with application to data-driven problems, *Oper. Res.*, **58** (2010), 595–612. https://doi.org/10.1287/opre.1090.0741

9. L. He, S. Liu, Z. J. M. Shen, Smart urban transport and logistics: A business analytics perspective, *Prod. Oper. Manag.*, **31** (2022), 3771–3787. https://doi.org/10.1111/poms.13775

10. L. He, H. Y. Mak, Y. Rong, Z. J. M. Shen, Service region design for urban electric vehicle sharing systems, *Manuf. Serv. Oper. Manag.*, **19** (2017), 309–327. https://doi.org/10.1287/msom.2016.0611

11. M. Lu, Z. Chen, S. Shen, Optimizing the profitability and quality of service in carshare systems under demand uncertainty, *Manuf. Serv. Oper. Manag.*, **20** (2018), 162–180. https://doi.org/10.1287/msom.2017.0644

12. R. Cui, S. Gallino, A. Moreno, D. J. Zhang, The operational value of social media information, *Prod. Oper. Manag.*, **27** (2018), 1749–1769. https://doi.org/10.1111/poms.12707

13. J. Carlsson, S. Song, Coordinated logistics with a truck and a drone, *Manag. Sci.*, **64** (2018), 4052–4069. https://doi.org/10.1287/mnsc.2017.2824

14. Z. Zou, H. Younes, S. Erdoğan, J. Wu, Exploratory analysis of real-time e-scooter trip data in Washington, DC, *Transp. Res. Rec.*, **2674** (2020), 285–299. https://doi.org/10.1177/0361198120919760

15. C. Glaeser, M. Fisher, X. Su, Optimal retail location: Empirical methodology and application to practice: Finalist–2017 M&SOM practice-based research competition, *Manuf. Serv. Oper. Manag.*, **21** (2019), 86–102. https://doi.org/10.1287/msom.2018.0759

16. D. Bertsimas, Y. Sian Ng, J. Yan, Joint frequency-setting and pricing optimization on multimodal transit networks at scale, *Transp. Sci.*, **54** (2020), 839–853. https://doi.org/10.1287/trsc.2019.0959

17. D. Bertsimas, A. Delarue, P. Jaillet, S. Martin, Travel time estimation in the age of big data, *Oper. Res.*, **67** (2019), 498–515. https://doi.org/10.1287/opre.2018.1784

18. H. de Vries, J. van de Klundert, A. Wagelmans, The roadside healthcare facility location problem a managerial network design challenge, *Prod. Oper. Manag.*, **29** (2020), 1165–1187. https://doi.org/10.1111/poms.13152

19. J. Boutilier, T. Chan, Ambulance emergency response optimization in developing countries, *Oper. Res.*, **68** (2020), 1315–1334. https://doi.org/10.1287/opre.2019.1969

20. E. Gralla, J. Goentzel, C. Fine, Problem formulation and solution mechanisms: A behavioral study of humanitarian transportation planning, *Prod. Oper. Manag.*, **25** (2016), 22–35. https://doi.org/10.1111/poms.12496

21. Z. Hao, L. He, Z. Hu, J. Jiang, Robust vehicle pre-allocation with uncertain covariates, *Prod. Oper. Manag.*, **29** (2020), 955–972. https://doi.org/10.1111/poms.13143

22. A. Kabra, E. Belavina, K. Girotra, Bike-share systems: Accessibility and availability, *Manag. Sci.*, **66** (2020), 3803–3824. https://doi.org/10.1287/mnsc.2019.3407

23. S. Liu, L. He, Z. J. M. Shen, On-time last-mile delivery: Order assignment with travel-time predictors, *Manag. Sci.*, **67** (2021), 4095–4119. https://doi.org/10.1287/mnsc.2020.3741

24. S. Steinker, K. Hoberg, U. Thonemann, The value of weather information for e-commerce operations, *Prod. Oper. Manag.*, **26** (2017), 1854–1874. https://doi.org/10.1111/poms.12721

25. M. Ang, Y. Lim, M. Sim, Robust storage assignment in unit-load warehouses, *Manag. Sci.*, **58** (2012), 2114–2130. https://doi.org/10.1287/mnsc.1120.1543

26. M. Lim, H. Mak, Y. Rong, Toward mass adoption of electric vehicles: Impact of the range and resale anxieties, *Manuf. Serv. Oper. Manag.*, **17** (2015), 101–119. https://doi.org/10.1287/msom.2014.0504

27. J. Carlsson, M. Behroozi, K. Mihic, Wasserstein distance and the distributionally robust TSP, *Oper. Res.*, **66** (2018), 1603–1624. https://doi.org/10.1287/opre.2018.1746

28. G. Baloch, F. Gzara, Strategic network design for parcel delivery with drones under competition, *Transp. Sci.*, **54** (2020), 204–228. https://doi.org/10.1287/trsc.2019.0928

29. J. Shu, M. Chou, Q. Liu, C. Teo, I. Wang, Models for effective deployment and redistribution of bicycles within public bicycle-sharing systems, *Oper. Res.*, **61** (2013), 1346–1359. https://doi.org/10.1287/opre.2013.1215

30. G. Cachon, K. Daniels, R. Lobel, The role of surge pricing on a service platform with self-scheduling capacity, *Manuf. Serv. Oper. Manag.*, **19** (2017), 368–384. https://doi.org/10.1287/msom.2017.0618

31. S. Datner, T. Raviv, M. Tzur, D. Chemla, Setting inventory levels in a bike sharing network, *Transp. Sci.*, **53** (2019), 62–76. https://doi.org/10.1287/trsc.2017.0790

32. H. Abouee-Mehrizi, O. Berman, S. Sharma, Optimal joint replenishment and transshipment policies in a multi-period inventory system with lost sales, *Oper. Res.*, **63** (2015), 342–350. https://doi.org/10.1287/opre.2015.1358

33. R. Yuan, S. Graves, T. Cezik, Velocity-based storage assignment in semi-automated storage systems, *Prod. Oper. Manag.*, **28** (2019), 354–373. https://doi.org/10.1111/poms.12925

34. Q. Deng, X. Fang, Y. Lim, Urban consolidation center or peer-to-peer platform? The solution to urban last-mile delivery, *Prod. Oper. Manag.*, **30** (2021), 997–1013. https://doi.org/10.1111/poms.13289

35. Z. Wang, J. Sheu, C. Teo, G. Xue, Robot scheduling for mobile-rack warehouses: Human–robot coordinated order picking systems, *Prod. Oper. Manag.*, **31** (2022), 98–116. https://doi.org/10.1111/poms.13406

36. W. Qi, L. Li, S. Liu, Z. J. M. Shen, Shared mobility for last-mile delivery: Design, operational prescriptions, and environmental impact, *Manuf. Serv. Oper. Manag.*, **20** (2018), 737–751. https://doi.org/10.1287/msom.2017.0683

37. B. Yildiz, M. Savelsbergh, Provably high-quality solutions for the meal delivery routing problem, *Transp. Sci.*, **53** (2019), 1372–1388. https://doi.org/10.1287/trsc.2018.0887

38. M. Ulmer, B. Thomas, A. Campbell, N. Woyak, The restaurant meal delivery problem: Dynamic pickup and delivery with deadlines and random ready times, *Transp. Sci.*, **55** (2021), 75–100. https://doi.org/10.1287/trsc.2020.1000

39. S. Jain, G. Shao, S. J. Shin, Manufacturing data analytics using a virtual factory representation, *Int. J. Prod. Res.*, **55** (2017), 5450–5464. https://doi.org/10.1080/00207543.2017.1321799

40. A. Nasrollahzadeh, A. Khademi, M. Mayorga, Real-time ambulance dispatching and relocation, *Manuf. Serv. Oper. Manag.*, **20** (2018), 467–480. https://doi.org/10.1287/msom.2017.0649

41. X. Li, X. Zhao, W. Pu, P. Chen, F. Liu, Z. He, Optimal decisions for operations management of BDAR: A military industrial logistics data analytics perspective, *Comput. Ind. Eng.*, **137** (2019), 106100. https://doi.org/10.1016/j.cie.2019.106100

42. S. Chung, Applications of smart technologies in logistics and transport: A review, *Transp. Res. Part E Logist. Transp. Rev.*, **153** (2021), 102455. https://doi.org/10.1016/j.tre.2021.102455

43. H. Mak, Y. Rong, Z. J. M. Shen, Infrastructure planning for electric vehicles with battery swapping, *Manag. Sci.*, **59** (2013), 1557–1575. https://doi.org/10.1287/mnsc.1120.1672

44. L. He, G. Ma, W. Qi, X. Wang, Charging an electric vehicle-sharing fleet, *Manuf. Serv. Oper. Manag.*, **23** (2021), 471–487. https://doi.org/10.1287/msom.2019.0851

45. T. Chan, D. Demirtas, R. Kwon, Optimizing the deployment of public access defibrillators, *Manag. Sci.*, **62** (2016), 3617–3635. https://doi.org/10.1287/mnsc.2015.2312

46. T. Chan, Z. J. M. Shen, A. Siddiq, Robust defibrillator deployment under cardiac arrest location uncertainty via row-and-column generation, *Oper. Res.*, **66** (2018), 358–379. https://doi.org/10.1287/opre.2017.1660

47. J. Carlsson, M. Behroozi, R. Devulapalli, X. Meng, Household-level economies of scale in transportation, *Oper. Res.*, **64** (2016), 1372–1387. https://doi.org/10.1287/opre.2016.1533

48. T. Huang, D. Bergman, R. Gopal, Predictive and prescriptive analytics for location selection of add-on retail products, *Prod. Oper. Manag.*, **28** (2019), 1858–1877. https://doi.org/10.1111/poms.13018

49. N. Salari, S. Liu, Z. J. M. Shen, Real-time delivery time forecasting and promising in online retailing: When will your package arrive, *Manuf. Serv. Oper. Manag.*, **24** (2022), 1421–1436. https://doi.org/10.1287/msom.2022.1081

50. A. Gunasekaran, T. Papadopoulos, R. Dubey, S. Wamba, S. Childe, B. Hazen, et al., Big data and predictive analytics for supply chain and organizational performance, *J. Bus. Res.*, **70** (2017), 308–317. https://doi.org/10.1016/j.jbusres.2016.08.004

51. A. Nguyen, L. Zhou, V. Spiegler, P. Ieromonachou, Y. Lin, Big data analytics in supply chain management: A state-of-the-art literature review, *Comput. Oper. Res.*, **98** (2018), 254–264. https://doi.org/10.1016/j.cor.2017.07.004

52. G. Wang, A. Gunasekaran, E. Ngai, T. Papadopoulos, Big data analytics in logistics and supply chain management: Certain investigations for research and applications, *Int. J. Prod. Res.*, **176** (2016), 98–110. https://doi.org/10.1016/j.ijpe.2016.03.014

53. A. Elmachtoub, P. Grigas, Smart "predict, then optimize", *Manag. Sci.*, **68** (2022), 9–26. https://doi.org/10.1287/mnsc.2020.3922

54. D. Bertsimas, N. Kallus, From predictive to prescriptive analytics, *Manag. Sci.*, **66** (2020), 1025–1044. https://doi.org/10.1287/mnsc.2018.3253

55. P. Notz, R. Pibernik, Prescriptive analytics for flexible capacity management, *Manag. Sci.*, **68** (2022), 1756–1775. https://doi.org/10.1287/mnsc.2020.3867

56. G. Ban, C. Rudin, The big data newsvendor: Practical insights from machine learning, *Oper. Res.*, **67** (2019), 90–108. https://doi.org/10.1287/opre.2018.1757

57. Y. Ran, S. Wang, K. Fagerholt, A semi-"smart predict then optimize" (semi-SPO) method for efficient ship inspection, *Transp. Res. Part B Methodol.*, **142** (2020), 100–125. https://doi.org/10.1016/j.trb.2020.09.014

58. S. Wang, X. Tian, R. Yan, Y Liu, A deficiency of prescriptive analytics—No perfect predicted value or predicted distribution exists, *Electron. Res. Arch.*, **30** (2022), 3586–3594. https://doi.org/10.3934/era.2022183

59. S. Wang, R. Yan, "Predict, then optimize" with quantile regression: A global method from predictive to prescriptive analytics and applications to multimodal transportation, *Multimodal Transp.*, **1** (2022), 100035. http://doi.org/10.1016/j.multra.2022.100035

60. J. Kotary, F. Fioretto, P. Van Hentenryck, B. Wilder, End-to-end constrained optimization learning: A survey, in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, (2021), 4475–4482. https://doi.org/10.24963/ijcai.2021/610

61. A. Ferber, B. Wilder, B. Dilkina, M. Tambe, MIPaaL: Mixed integer program as a layer, in *Proceedings of the AAAI Conference on Artificial Intelligence*, (2020), 1504–1511. https://doi.org/10.1609/aaai.v34i02.5509

62. B. Wilder, B. Dilkina, M. Tambe, Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization, in *Proceedings of the AAAI Conference on Artificial Intelligence*, (2019), 1658–1665. https://doi.org/10.1609/aaai.v33i01.33011658

63. J. Mandi, E. Demirovi, P. Stuckey, T. Guns, Smart predict-and-optimize for hard combinatorial optimization problems, in *Proceedings of the AAAI Conference on Artificial Intelligence*, (2020), 1603–1610. https://doi.org/10.1609/aaai.v34i02.5521

64. M. Mulamba, J. Mandi, M. Diligenti, M. Lombardi, V. Bucarey, T. Guns, Contrastive losses and solution caching for predict-and-optimize, in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, (2021), 2833–2840. https://doi.org/10.24963/ijcai.2021/390

65. N. Kallus, Recursive partitioning for personalization using observational data, in *Proceedings of the 34th International Conference on Machine Learning*, (2017), 1789–1798.

66. D. Bertsimas, J. Dunn, N. Mundru, Optimal prescriptive trees, *INFORMS J. Optim.*, **1** (2019), 164–183. https://doi.org/10.1287/ijoo.2018.0005

67. A. Elmachtoub, J. Liang, R. Mcnellis, Decision trees for decision-making under the predict-then-optimize framework, in *Proceedings of the 37th International Conference on Machine Learning*, (2020), 2858–2867.

68. N. Kallus, X. Mao, Stochastic optimization forests, *Manag. Sci.*, **2022** (2022). https://doi.org/10.1287/mnsc.2022.4458