Tackling Decision Dependency in Contextual Stochastic Optimization

Abstract: We provide a novel approach to solving the contextual stochastic optimization problem with the decision-dependent effect. Our algorithm overcomes the model dependency of the parameters on the decision variables, whereas traditional contextual optimization algorithms generally fail to tackle this decision dependency embedded in the model. To solve this problem, we introduce the contextual gradient concept that prescribes the true objective's first-order derivative and develop the contextual gradient descent (CGD) algorithm. We prove that the proposed CGD method achieves a bounded error to the global optimality under the convexity condition, and converges to a stationary point of expected gradient otherwise. Using real-world datasets, extensive numerical experiments demonstrate the superior numerical performance of our proposed CGD algorithm compared to existing methods that can be applied to contextual optimization problems with the decision-dependency effect.

Key words: contextual optimization, prescriptive analytics, decision-dependency

1 Introduction

In many real-world management processes, decision makers (DM) regularly confront the following stochastic optimization model,

$$\min_{x\in\mathcal{X}} g(x) = \mathbb{E}_{y\sim f(y)}[l(x,y)], \tag{1}$$

where $l(x,y)$ is the objective function to be minimized, and x is the decision variable in a feasible region $\mathcal{X}\in\mathbb{R}^d$; y is some stochastic model parameters, whose distribution density function is $f(y)$. Note that $f(y)$ is unknown to the DM. There are two complex tasks in this stochastic optimization problem: prediction and optimization. The prediction task requires DM to predict stochastic parameters in models, such as demand and lead time. The optimization task usually involves optimizing decisions with the aim of maximizing profits or minimizing costs based on the prediction result. When some side information z is provided to estimate y, *i.e.*, $f(y;z)$ is dependent on z, the problem is also called Contextual Stochastic Programming (CSO). We refer to Sadana et al. (2023) for a comprehensive review of recent contextual optimization works.

The CSO problem is widely applied in practice. A notable example is the contextual newsvendor problem. In this problem, the DM should optimize the order quantity x with unknown demand y. Some features z are provided for the DM to estimate the distribution of the demand. The DM also has historical demand data $\{z^{(i)}, D^{(i)}\}_{i=1}^N$. And the objective is to minimize the minus revenue $f(x,y) = -p(y\wedge x)+cx-s(x-y)^+$, where p,c,s are the per-unit price, inventory cost and salvage value respectively. And $\wedge$ denotes component-wise

minimum, $(\cdot)^+ = \max\{\cdot, 0\}$. The contextual newsvendor problem has been widely studied, and methods like sample average approximation (SAA), empirical risk minimization (ERM), and deep learning are applied to solve such problem (see, e.g., Ban and Rudin (2018), Lin et al. (2022), Levi et al. (2015), Godfrey and Powell (2001)).

For these applications, an intrinsic challenge is that the stochastic parameter depends on decision variables. In other words, we need to know x to estimate the distribution $f(y;z,x)$ in (1). For instance, customer demand is usually affected by the pricing decision made in a revenue management context. However, except for a few options (Bertsimas and Kallus 2019, Bertsimas and Koduri 2022), the contextual optimization model with endogenous uncertainty is rarely studied. Although Bertsimas and Kallus (2019) provided a weighted SAA approach and built an approximated model to estimate the expectation of objective function conditioned on x and z in (1), they just proved the tractability when using discretization, and a special solution approach applicable for the tree-weight case. Therefore, how to efficiently solve the general contextual optimization problem (1) under the decision-dependent effect still remains unclear. The aim of this research is to provide a general approach to solve the decision-dependent contextual optimization models.

Note that most of the existing solution approaches for CSO are not directly applicable to CSO with decision-dependent uncertainty. Most traditional approaches (e.g., SAA and ERM) estimate the distribution of stochastic parameters conditioned on contextual information $y|z$, and the estimated conditional distribution is fixed in the downstream optimization step since the contextual information z does not change. However, the conditional distribution $y|z,x$ changes continuously during the optimization step because of the iteration of x. Hence, it is hard to perform optimization unless we have the estimation of y conditioned on every possible combination of z and x.

To address these issues, a natural idea is to incorporate the decision variable x into the estimation of the stochastic parameter (Bertsimas and Kallus 2019). That is, to add decision variables as an input of the estimation model to estimate the conditional distribution. Although this estimation approach also achieves asymptotic optimality (see Theorem 10 in Bertsimas and Kallus (2019)), its downstream optimization task is difficult to perform. This is because the decision variables appear in both the estimation model and objective function, causing the non-convexity of the optimization model. Furthermore, for some discrete estimation models (e.g., kNN, tree models, and random forest), the approximate objective function is discontinuous to the decision variables even though $l(x,y)$ is continuous.

Another idea to handle the decision-dependency is to use the local linear method to estimate the distribution within a neighborhood of the current decision (see, e.g., Liu et al. (2021)). However, the convergence of the local linear regression approach requires that the stochastic parameter follows a regression model of the decision variable, while this assumption may not hold in practice since the variance of the stochastic parameter can also change with the decision. Moreover, they did not consider the impact of contextual information. Therefore, to the best of our knowledge, no algorithm has been developed to directly solve a

contextual stochastic optimization problem under the decision-dependent effect, which we address in this paper.

Instead, we consider algorithms that directly approximate the derivative of the true objective, *i.e.*, the true expectation of the stochastic objective function. Specifically, we propose the *Contextual Gradient* in (7). Note that the concept of contextual gradient also appears in Lee et al. (2022). But in that work it denotes a scaled gradient to suit meta-learning task, while in our work it is an approximation to the unknown stochastic gradient. Under mild conditions, we show that the contextual gradient is an unbiased estimation of the expected derivative of the objective function (Proposition 1). Thus the contextual gradient can be a proxy for the gradient and can be embedded into some gradient-based algorithms, for instance, gradient descent (GD) and stochastic gradient descent (SGD).

We establish a global convergence guarantee of the contextual gradient descent (CGD) algorithm when the objective function is convex. We prove the error upper bounds of CGD under convex and strong convex conditions. Specifically, we show that the error of CGD can be separated into two parts: decision-dependent error and initial value error. Furthermore, under the strongly-convex condition, the CGD algorithm achieves a converging upper bound of the distance between the iterative solutions and the global optimal solution.

We also extend the convergence results to the general non-convex case. Under the non-convex condition, although we cannot completely characterize global optimality guarantees, we find that the algorithm can converge to the stationary point of the original prescriptive optimization problem, in which the true expected gradient is zero. We further show that this condition is necessary for global optimality. Our convergence results in both convex and non-convex conditions are similar to those of gradient descent (see, e.g., Bertsekas (1999)), which indicates that the contextual gradient is a reasonable estimation of the true gradient and can suitably be a proxy for the gradient in CSO problems.

We further conduct extensive numerical experiments to demonstrate the effectiveness of the proposed CGD algorithm compared to several different policies, including the discretization method and the estimate-then-optimize framework based on linear decision rule assumption. Our numerical experiments are conducted on both a simulated newsvendor pricing problem and a real-world electricity pricing problem.

1.1 Contributions

Our contributions are mainly three-fold:

Concept of contextual gradient. We propose the concept of contextual gradient, which is a direct estimation of the gradient of the true objective expectation. The contextual gradient is derived by applying a weighted-SAA approach to the true expected gradient. We prove that the contextual gradient is an unbiased estimation of the expected gradient of the objective function $l(x,y)$ (Proposition 1). The contextual gradient thereby enables the design of gradient-based algorithms and thus solves the challenge arising from the inaccessibility of gradient information for contextual optimization models with endogenous uncertainty.

Algorithm design and convergence guarantee. We develop the Contextual Gradient Descent (CGD) algorithm by embedding the contextual gradient into the gradient descent algorithm. As far as we know, this is the first formal approach to solve the contextual stochastic optimization model under decision dependency. Our algorithm enjoys convergence guarantees under both convex and non-convex cases. Technically, we extend the performative prediction analysis in Mendler-Dünner et al. (2020a) to a contextual and prescriptive case, where the dependency on contextual information is involved and the expected gradient can only be estimated by historical data but not sampling.

Numerical results. Our numerical results show the superior performance of the proposed CGD algorithm. We conduct numerical experiments based on data that comes from both the simulation and practice. Compared to the discretization approach, our algorithm obtains a smaller gap in a significantly shorter time. Moreover, compared to the parametric estimate-then-optimize (ETO) solution approach that takes linear decision assumption to the stochastic parameter, our algorithm is more generalizable by its distribution-free and nonparametric setting, while the parametric ETO method need distributional assumptions on the decision dependency. Our algorithm outperforms by 33% under complex demand distribution. Even when the true demand is exactly a linear regression model, our algorithm shows $< 5\%$ gap to the parametric PTO approach that owns a correct prior knowledge to the problem.

1.2 Applications

Price-setting newsvendor. Although the application of end-to-end model to newsvendor problems is widely studied (Ban and Rudin 2018, Lin et al. 2022), the end-to-end price-setting newsvendor problem is seldom studied except for some options like quantile regression (Harsha et al. 2021). In the newsvendor pricing problem, the DM should also make decision on pricing, while the stochastic demand is dependent on the pricing decision, hence shows the decision-dependent property. In this case, the PTO framework does not work since we cannot optimize the pricing decision with fixed demand. Our framework can directly optimize the pricing decision based on the historical pricing and demand data.

Assortment problem. In assortment problems, the customer demands need to be somehow estimated from data can depend on many contextual features, especially the customer type. Furthermore, they also depend on the assortment decision made by the DM. Specifically, the assortment decision affects the probability that an arriving customer chooses the product. Many existing work adopts an online learning framework that gradually learns the demand conditioned on the assortment decision (Li et al. 2022, Jasin et al. 2024, Kallus and Udell 2020). But if we already have offline data that contains historical decision, features and demands, this decision dependency property is convenient for our prescriptive framework to make a single period assortment decision. An online approach first predict the demands for a given type of customer then optimize the assortment decision, while our framework optimize the assortment decision directly.

1.3 Literature Review

The fusion of data-driven contextual optimization has become more and more popular in recent years. In this section, we provide a review on contextual optimization and stochastic models with decision dependency and compare them with our approach.

Estimate-then-optimize models. The most relevant stream of research is the estimate-then-optimize (ETO) paradigm. ETO first gives an estimation of the conditional distribution given the context feature, before optimizing the decision under the conditional distribution. The most related work is that of Bertsimas and Kallus (2019), who proposed a general nonparametric approach to solve the prescribe optimization problem. They estimated the conditional distribution using ML methods – for example, nearest neighbor and decision tree. The input of such ML methods were features, and decision variables if the decision can affect uncertainty. The output of the ML methods were treated as the sample weight, which was used to directly estimate the true objective. Bertsimas and McCord (2019) further extended the work by adding the penalize term into the objective. Compared to our work, they bypassed the difficulty of computation under decision-dependency. In fact, Bertsimas and Kallus (2019) only gave the solution approach when tree regression method is adopted in the estimation step, otherwise the decision-dependent model can only be solved by discretization (see Theorem 4 in Bertsimas and Kallus (2019)). Srivastava et al. (2021) further designed a regularized approximation to guarantee the out-of-sample performance based on the Nadaraya-Watson kernel regression. And Lin et al. (2022) adopted this ETO framework to a risk-averse newsvendor problem. However, these works assumed that the stochastic parameters are independent of decision variable. It is still not clear how to handle the decision-dependent property, which must be addressed in our work.

Apart from nonparametric regression by ML methods, another way to approximate the objective is to use the sample average approximation (SAA). The SAA method is to approximate the conditional expectation by averaging historical samples (Kleywegt et al. 2002, Homem-de Mello 2001). Feng and Shanthikumar (2022) pointed out that a pure SAA approach will cause overfitting. Therefore, common approaches including adding a regularizer or constraints that can control the predicted profit variability (Levi et al. 2015, 2007, Cheung and Simchi-Levi 2019, Qin et al. 2022). Kannan et al. (2022) proposed an SAA framework with covariate data. However, SAA still assumes thee independence of stochastic parameters to decisions and thus cannot be directly applied to the decision dependent model.

Another stream of works adopt the policy-based paradigm and directly optimize parameterized policies. Ban and Rudin (2018) adopted the linear decision rule and solved the newsvendor model by optimizing the parameters in the linear decision. They proved its superiority to SAA approach. Apart from the linear decision, Zhang and Gao (2017) and Huber et al. (2019) adopted the neural network to parameterize the decision. However, these approaches still imposed the independence of stochastic parameters to decisions. In contrast, our framework can estimate the expectation conditioned on both side information and decisions, and thus provide a clear way to deal with the decision-dependent property.

Integrate optimization models. Another stream of work on contextual optimization is the integrate optimization, which is an end-to-end framework that integrates the prediction and the optimization step. One type of integrate optimization model is to learn the conditional distribution. Donti et al. (2017) modeled the conditional distribution in a parametric manner. They showed that their task-based learning model outperforms MLE estimation approach in most cases. Grigas Paul (2023) proposed an estimate-optimize framework that, where the conditional distribution was estimated by a hypothesis class and the decision was optimized by the ERM principle with regularized oracle. Compared to their research, our framework does not need to presume the hypothesis class, thus can better deal with situations with a lack of distribution information. Kallus and Mao (2022) estimated the conditional distribution by a random forest that can directly optimize the downstream decisions. Though the estimation of the distribution conditioned on side information has been well-established, existing research does not provide a clear way to estimate the distribution of decision.

The other model focuses on designing the loss function in the prediction step to meet the subsequent optimization goal. Elmachtoub and Grigas (2022) designed a new SPO+ loss function for SPO paradigm to address the intractability for the traditional SPO loss function. They proved the consistency result for the SPO+ loss function and provided practical computational approaches for linear predictor cases, and the generalization bounds of SPO was provided in El Balghiti et al. (2022). Compared with their work, our framework can solve both convex and non-convex cases. The SPO paradigm has been extended to several contextual optimization problems. In Butler and Kwon (2023), they extended the SPO framework to the two-stage linear programming model and designed a convex approximation for the loss function. And the SPO framework was also adopted in combinatorial optimization problems and mixed integer linear programming (MILP) problems (Mandi et al. 2020, Jeong et al. 2022). The SPO framework provides a new prediction paradigm that can minimize the decision error, but it still cannot handle the condition when the decision can affect the distribution of unknown parameters.

Other existing studies investigated the task-based end-to-end optimization models. For the linear programming problem, Cristian et al. (2022) proposed a neural network framework that can learn to solve the downstream linear programming problems. Their end-to-end model could be solved with exact derivatives. For the non-differentiable case, Wilder et al. (2018) proposed a decision-focused learning framework for combinatorial problems. They overcame the non-differentiating property by considering the KKT conditions of the continuous relaxation of the combinatorial problem. And Mandi and Guns (2020) further investigated the homogeneous self-dual embedding of the relaxed MILP problem. Similar to their works, our objective function is also non-differentiable, and we also overcome the challenge of computing the gradient for the end-to-end optimization model, but we overcome this by approximating the gradient of true expectation, rather than differentiating the continuous relaxation of the objective function.

To summary, most CSO frameworks assumed the independence of stochastic parameters on decision variables. It is hard to extend existing CSO to the decision-dependent context since the estimation of distribution conditioned on decision variables is seldom discussed. There are some works related to CSO problems with decision-dependent uncertainty (Bertsimas and Kallus 2019, Harsha et al. 2021), but they either focused on a specific problem (*e.g.* newsvendor pricing problem), or did not provide a way for optimization. To fix this gap, our framework provides a computable optimization approach by the proposed CGD algorithm.

Models with decision dependency. Though the decision-dependent property has not been fully studied in the field of contextual optimization, it has been widely studied in operations management (OM) models, especially revenue management problems such as dynamic pricing. We refer to den Boer (2015) for a comprehensive review on the dynamic pricing problems. We find that most dynamic pricing models are parametric or impose some assumptions on the distribution form, for example, the additive demand (Biswas and Avittathur 2018, Wang and Chen 2015) and multiplication demand (Kazaz and Webster 2011, Salinger and Ampudia 2011). For the nonparametric and distribution-free models, Besbes and Zeevi (2009) studied an online learning dynamic pricing policy, and Chen et al. (2019) further extended to the replenishment policy. Similar to their work, we also adopt a nonparametric setting without the knowledge of either the distribution or dependency function. But we focus on a single-period problem rather than a multi-period online learning setting. Liu et al. (2021) proposed a coupled learning enabled optimization (CLEO) algorithm to solve the stochastic programming models with decision-dependent uncertainty. The CLEO algorithm estimated the conditional distribution by local linear regression within a delicately designed trust region. But the convergence of CLEO algorithm relies on the regression assumption of the decision variable, that is, $y = \phi(x) + \varepsilon$ in (1), while it is still not clear how to estimate the conditional distribution when the stochastic parameters y cannot be constructed as a regression model of the decision variable x, which is commonly seen in practice. For example, when the variance is also dependent on the decision variable, the error item ε also changes with decision x. Furthermore, compared with the CLEO algorithm for SP problems without contextual information, we introduce the impact of contextual information into our framework.

There are other stochastic programming (SP) models that investigated the decision-dependent property (Goel and Grossmann 2006, Larson et al. 2019). Dupačová (2006) solved the SP model when the distribution is dependent on the decision. Mendler-Dünner et al. (2020a) proposed a repeated gradient descent and provided an iterative approach to solve the decision-dependent SP problem. These SP works bypassed issues of the estimation step. They either assumed that the distribution is known or estimated the distribution by sampling, which cannot be realized in our setting. Moreover, while Mendler-Dünner et al. (2020a) only investigated the performance under strongly convex conditions, we extend the performance analysis of the algorithm to general cases. Liu et al. (2024) gave a gradient approximation approach through an bayesian approach, while their distribution function $f(y;x,\theta)$ needs to be parameterized by θ and the gradient estimator that contains $\mathbb{E}_{\pi_t}[f(\cdot;x,\theta)]$ is hard to compute when the specific form of $f(y;x)$ is unknown.

Another stream of related work investigated decision-dependent property from the perspective of robust optimization (RO) and distributionally robust optimization (DRO). Luo and Mehrotra (2020) constructed a decision-dependent robust ambiguity set to solve a stochastic resource allocation problem. And Noyan et al. (2021) constructed the ambiguity set by Wasserstein distance. These approaches typically relied on some priori knowledge about the distribution family or distribution map, while our framework does not rely on the knowledge about the distribution family and the dependency form.

To summarize, most existing decision-dependent models adopted a parametric way to estimate the conditional distribution or assumed the regression relationship between decision variables and stochastic parameters. In contrast, we consider a nonparametric and distribution-free approach and utilize the context information to solve the problem. Hence, our algorithm can be used for more complex distribution and decision-dependency cases.

Organizations

The rest of the paper is organized as follows. In Section 2, we formally state the model and assumptions, as well as the intuition of the contextual gradient, we further investigate the convergence property of the contextual gradient. In Section 3, we propose the CGD algorithm and provide its convergence under convex and non-convex cases. In Section 4, we conduct the numerical experiment on the CGD algorithm and compare it with other existing solution approaches. Finally, Section 5 concludes the paper.

Preliminaries

For simplicity of notation, we use $x \wedge y$ to denote $\min\{x, y\}$, and $(x)^+$ to denote $\max\{x, 0\}$. We let ∇ be the gradient denotation, and ∂ denote the subgradient set. Let $x = (x_1, ..., x_d)^T$. The subscript denotes the corresponding coordinate of a vector, and $\mathbf{1}$ denotes the all-one vector. We use $\|\cdot\|$ to denote l_2 norm for vectors and matrix. A function f is $L-$Lipschitz continuous on $x \in \mathcal{X}$ if $\|f(x_1) - f(x_2)\| \leq L\|x_1 - x_2\|$ for any $x_1, x_2 \in \mathcal{X}$. A function f is $\gamma-$strongly convex if $f(x_1) - f(x_2) - \nabla f(x_1)^T(x_1 - x_2) \geq \gamma\|x_1 - x_2\|^2$. We use $[N] := \{1, ..., N\}$ to denote the subscript set.

2 Problem Setting and Contextual Gradient Formulation

In this section, we formally introduce the decision-dependent contextual optimization problem, as well as the concept of the contextual gradient.

2.1 Problem Setting and Assumptions

Throughout the paper, we consider a contextual optimization model with decision dependency, where the distribution of model parameters depends on both the contextual features and decision variables. Specifi-

cally, we focus on the task of minimizing the expected objective function given the contextual features and history datasets:

$$\begin{aligned} &\min_x g(x) \triangleq \mathbb{E}_{y\sim f(y;x,z)}[l(x,y)] \\ &\text{s.t. } x \in X, \end{aligned} \tag{2}$$

Here, x is the decision variable, z is the observed contextual information, and y is the random model parameter. Moreover, $f(y;x,z)$ denotes the probability density function of y, indicating that the uncertainty of y is dependent on x and z. The objective function is denoted by $l(\cdot,\cdot)$. We assume the feasible region X is known with certainty. Apart from the parameters in (2), the DM also has historical decision samples $\mathcal{D}_n = \{x^i, z^i, y^i\}_{i=1}^n$.

We make several assumptions about the optimization problem (2).

ASSUMPTION 1 **(Same bounded range)**. *The value range of a random parameter y remains the same under any x, and the value range of y is bounded.*

In practice, Assumption 1 is usually satisfied since the stochastic model parameters are usually bounded. For example, the stochastic demand should be positive and there shall typically be an upper bound on it. We can take the union set of the value ranges of y under different x and assign the probability outside of the distribution $y|x$ as 0. We denote the range and its volume as Ω and S_Ω.

ASSUMPTION 2 **(Differentiation–integration exchange)**. *We assume that $l(x,y)$ is differentiable in X, and there exists $L^1(y)$ function $g(y)$, $|g(y) \geq n|l(x+\frac{1}{n},y) - l(x,y)|$ for all $x \in X$ and $y \in \Omega$.*

Assumption 2 is the assumption of Lebesgue Dominated Convergence Theorem, which implies that we can change the order of integration and differentiation when calculating the derivative of the integral of $l(x,y)$ (see Theorem 2.27 in B.Folland (1999)). That is,

$$\nabla_x \int l(x,y) f(y;x,z) dy = \int \nabla_x (l(x,y) f(y;x,z)) dy,$$

which enables us to access the derivative of the objective function. This assumption is reasonable because most objective functions are integrable and bounded at almost everywhere in practice. In the following, we assume that the objective function satisfies Assumption 2.

We also assume that the distance between the decision-dependent distributions under different decisions can be bounded by the distance between the two decisions.

ASSUMPTION 3 **(ε-sensitivity)**. *We assume that the distribution map $f(y;\cdot,z)$ is ε-sensitive to decision variable. That is, for all x_1, x_2,*

$$W_1(f(y;x_1,z), f(y;x_2,z)) \leq \varepsilon \|x_1 - x_2\|_2, \tag{3}$$

where W_1 denotes the earth mover's distance (Rubner et al. 2000).

Intuitively, Assumption 3 ensures that the difference between decision-dependent distributions is not too large under different decisions x_1, x_2. Therefore, when analyzing the gap between the approximate solution and the optimal solution, we can convert the difference between expectations under different distributions into the distance between decision variables.

ASSUMPTION 4 **(Lipschitz continuous and bounded gradient)**. *$l(x,y)$ is lipschitz continuous in both x and y and its gradient in x in bounded. That is,*

$$(a)|l(x_1,y)-l(x_2,y)| \le L_1\|x_1-x_2\|, \quad \forall x_1,x_2 \in \mathcal{X}, y \in \Omega,$$
$$(b)|l(x,y_1)-l(x,y_2)| \le L_2\|y_1-y_2\|, \quad \forall x \in \mathcal{X}, y_1,y_2 \in \Omega,$$
$$(c)\|\nabla_x l(x,y)\| \le L_3^c, \quad \forall x \in \mathcal{X}, y \in \Omega.$$

ASSUMPTION 5 **(Lipschitz gradient and density)**. *The cost function $l(x,y)$ is smooth and Lipschitz continuous with a Lipschitz gradient. The probability density function of y has a Lipschitz gradient. That is,*

$$(a)\frac{\|\nabla_x l(x_1,y)-\nabla_x l(x_2,y)\|}{\|x_1-x_2\|} \le L_{1g}, \quad \forall x_1,x_2 \in \mathcal{X}, y \in \Omega,$$
$$(b)\frac{\|\nabla_x l(x,y_1)-\nabla_x l(x,y_2)\|}{\|y_1-y_2\|} \le L_{2g}, \quad \forall x \in \mathcal{X}, y_1,y_2 \in \Omega,$$
$$(c)\frac{\|\nabla_x f(y;x_1,z)-\nabla_x f(y;x_2,z)\|}{\|x_1-x_2\|} \le L_{3g}, \quad \forall x_1,x_2 \in \mathcal{X}, y \in \Omega.$$

Assumptions 4 and 5 are mild because the value ranges of decision variables are usually bounded in practice. Moreover, they are only required in some of the convergence results. We assume the Lipschitz property of the objective function so that we can guarantee the approximate error of the contextual gradient will cause a bounded error in our algorithm. We also assume that the distribution function of random parameters has a limited change rate when the decision x changes.

2.2 Weighted-SAA Estimation

Now, we discuss the weighted-SAA approach that the contextual gradient concept centers around. Specifically, Bertsimas and Kallus (2019) and Lin et al. (2022) proposed a weighted-SAA approach to estimate the expectation conditioned on the contextual information. They adopted the weighted-SAA approach to approximate the expected objective function:

$$\min_x \hat{g}(x) \stackrel{\Delta}{=} \sum_{i=1}^{N} w^i(x,z) l(x,y^i), \tag{4}$$

where $w^i(x,z)$ is a weight function derived from the data by ML methods. The input of the ML model are the contextual features, and decision variables if the model is decision-dependent. We refer the reader to Bertsimas and Kallus (2019) and Lin et al. (2022) for the choices of the particular ML model, including kNN, kernel regression, tree, and random forest (RF). The approximate model (4) does not restrict the form of the weight function. For brevity, we only present the definition of the kNN weight. Other definitions of weight functions (e.g., kernel regression, CART, random forest) can be found in Section EC.1.

DEFINITION 1 (KNN WEIGHT). The weight function can be derived from the definition of kNN:

$$w^{\text{kNN},i}(x,z) = \frac{1}{k}\mathbb{I}\{(x^i,z^i) \text{ is a kNN of } (x,z)\}, \quad \forall i \in [N], \tag{5}$$

where $\mathbb{I}\{\cdot\}$ is the indicator function, $[N] = \{1,2,...,N\}$ denotes the index set, and x^i is a kNN of x if and only if $|\{j \in \{1,...,N\}\backslash i : \|x^j - x\| < \|x^i, x\|\}| < k$.

The weighted-SAA estimation incorporates the similarity between the previous scenario and the current scenario, where a scenario refers to variables that can affect random parameters in the model, including the contextual features and decision variables. For instance, when the current scenario (x,z) and previous scenario (x^i,z^i) are dissimilar, the kernel weight will be small, meaning that sample i is not similar to the current condition and plays a minor role in the decision-making process.

However, we note that the approximate model (4) is hard to solve when the model is decision-dependent. First, there are cross-product terms since the decision variable exists in both the weight $w^i(x,z)$ and objective $l(x,y)$, thus the model can be non-convex even when $l(x,y)$ is convex on x. Second, the approximate model can be non-smooth and discontinuous if the discrete ML estimation models are adopted (e.g., kNN and tree models), even when $l(x,y)$ is smooth. In this case, traditional optimization approaches cannot be directly used for optimizing the weighted-SAA problem, and new methods are required for the estimation with the decision dependency taken into account.

2.3 Contextual Gradient

We formally describe the concept of contextual gradient. We construct a prescriptive model to approximate the gradient of objective in (2) using contextual information. The contextual gradient can be derived from the true gradient of the objective function. Since we have assumed that $l(x,y)$ is differentiable in Assumption 2, the gradient of objective function is

$$G(x';z') = \nabla_x \mathbb{E}[l(x,y)|z = z', x = x']. \tag{6}$$

And the contextual gradient is given by performing the weighted-SAA prescription to $G(x)$ directly.

DEFINITION 2 (CONTEXTUAL GRADIENT). Given a contextual variable z' and a decision x', the contextual gradient at x' given dataset $\{z^i,x^i,y^i\}$ is defined as

$$\hat{G}_N(x';z') = \sum_{i=1}^{N} w^{(i)}(x',z')\nabla_x l(x',y^i), \tag{7}$$

where $w^{(i)}(\cdot,\cdot)$ is the weight function calculated by historical data.

The contextual gradient overcomes the barrier of the loss of first-order information of the objective function in (4). In fact, the true gradient of the objective function (6) under decision dependency is hard to calculate since the DM needs to know the correlation between current and past decisions to calculate the cross-product terms. We can prove, in fact, that the contextual gradient is an unbiased estimation of the true gradient, *i.e.*, it converges to the expectation of the gradient of the objective function.

PROPOSITION 1 **(Convergence of Contextual Gradient)**. *Suppose that the joint distribution of* (x,z) *is absolutely continuous in the dataset, and its density function is bounded away from* 0 *to* $+\infty$ *on the support of* x,z*, and is twice continuously differentiable. Then the following uniform convergence over the inputs to the weights* (x',z')*, for some* $c_N \to \infty$*, almost surely,*

$$\lim_{N\to\infty} \sup_{\|x'\|+\|z'\|\le c_N} \left|\hat{G}_N(x';z') - \mathbb{E}[\nabla_x l(x,y)|x=x',z=z']\right| = 0. \tag{8}$$

Proposition 1 shows the convergence of the contextual gradient to the expectation of the gradient of the objective function. Nevertheless, we point out that converging to the *expectation of the objective gradient* is not equivalent to converging to the *gradient of expected objective function*. These two concepts are different because of the existence of the decision-dependent effect.

PROPOSITION 2. *Suppose the distribution of y is dependent on x, the expectation of the objective function* $\mathbb{E}[\nabla_x l(x,y)|X=x',Z=z']$ *is not equal to the gradient of the objective expectation* $G(x';z')$ *defined in (6) when the decision-dependency exists.*

Proposition 2 indicates that the contextual gradient cannot converge to the true gradient if the contextual model is decision-dependent because it converges to the expectation of the objective gradient instead of the gradient of the expected objective function. However we can still embed the contextual gradient into the gradient-based algorithms. In fact, the contextual gradient enjoys similar converging properties to the true gradient, that is, 1) the stationary point of the expected gradient is also a necessary condition for the global optimality, and 2) if the cost function is convex or strongly convex, the expected gradient can also guarantee a bounded error. These two properties of the contextual gradient are in accordance with the converging performance of true gradient, thus making the convergence of our proposed contextual gradient algorithm possible.

3 Algorithm Design and Convergence Analysis

In this section, we present ideas behind the design of the CGD algorithm and derive convergence guarantees for it. Herein, we focus on the case of non-constrained condition (*i.e.*, $\mathcal{X}=\mathbb{R}^d$). We first show the convergence result of the CGD algorithm under the convex case and prove its error bound $|g(x_N^k)-g(x^*)|$ of to the optimal solution in Theorem 1. And in the strong convex case, we show a stronger convergence of the distance $\|x_N^k-x^*\|$ in Theorem 2. Furthermore, we extend our results to the general non-convex setting and analyze the convergence property of the CGD algorithm under two types of step size choices. We demonstrate that the CGD algorithm also converges to some stationary point with bounded expected gradient in a rate of $O(\varepsilon^{-2})$. Despite the gap between the expected gradient and the true gradient of expectation, we demonstrate that converging to some stationary point with a bounded expected gradient is a necessary condition of optimality in Theorem 3.

3.1 Contextual Gradient Descent

Since Proposition 1 provides a converging result of the contextual gradient to the true expected gradient, intuitively, we can embed this contextual gradient into some gradient-based optimization algorithms, such as the gradient descent algorithm and stochastic gradient algorithm. Based on this idea, we design the CGD algorithm in Algorithm 1.

Algorithm 1 The Contextual Gradient Descent Algorithm

Input: initial solution x^0, contextual information z, dataset $\{x^i, z^i, y^i\}_{i=1}^N$.

Output: solution $\hat{x}^*$.

1: $r = 0$
2: **while** Stop criteria not satisfied **do**
3: Calculate contextual gradient $\hat{G}_N(x^r; z)$ by (7)
4: Select step size η^r
5: Let $x^{r+1} = x^r - \eta^r \hat{G}_N(x^r; z)$
6: $r = r + 1$
7: **end while**
8: $x^* = x^r$

The contextual gradient descent follows a similar descending paradigm to the gradient descent algorithm. We note that the step size must be suitably determined to implement the CGD algorithm. We adopt the diminishing step size and the Armijo step size in our work. The selection of the step size will also affect the convergence result, which is to be analyzed in the next section. The detailed comparison between two choices of step size can be seen in Appendix EC.4.2.

EXAMPLE 1 (EXAMPLE OF CGD ON NEWSVENDOR PRICING PROBLEM). In the newsvendor pricing problem with contextual information, the DM need to jointly make the pricing and ordering decision to maximize its profit. In this case, the decision variable is $x = (p, q)^T$ and the objective function is $l(x, y) = l(p, q, y) = -p(y \wedge q) + cq - s(q - y)^+$. Since $l(p, q, y)$ is not smooth when $q = y$, we study its contextual subgradient instead.

$$\partial_{p,q} l(p, q, y) = \left\{ \begin{bmatrix} -(y \wedge q) \\ -(p - c) + (p - s)e \end{bmatrix} : e \in [\mathbb{I}\{q > y\}, \mathbb{I}\{q \geq y\}] \right\}. \tag{9}$$

The subgradient set only contains one element almost everywhere, therefore we can still embed the contextual subgradient into the gradient descent algorithm in Algorithm 1. Specifically, in each iteration, we first calculate the $w^(i)(p^r, q^r, z)$ by the weighted-SAA approach, then calculate the contextual gradient $\hat{G}_N(p^r, q^r; z) = \sum_{i=1}^N w^{(i)}(p^r, q^r, z) \partial_{p,q} l(p^r, q^r, y^i)$. When $q^r = y^i$, we select any element from the subgradient set $\partial_{p,q} l(p^r, q^r, y^i)$.

As a result, the CGD algorithm can retain both the first-order information of the original objective function and the contextual information by directly prescribing the gradient. (8) implies that the contextual gradient is an unbiased approximation to the expected gradient. Therefore, the convergence of CGD algorithm and typical gradient descent may share some commonalities. That is, we can expect the convergence of CGD algorithm to global optimality under convex case, and to a stationary point under general non-convex cases.

3.2 Convergence Analysis

In this subsection, we analyze the convergence of the CGD algorithm under both the convex and non-convex case. Despite the inconsistency of expected gradient and true gradient of expectation in Proposition 2, we still prove the error bound in the convex setting and an asymptotic convergence to a stationary point in the non-convex setting. Intuitively, we overcome this barrier by showing that $\hat{G}_N(x,z)$ is also a reasonable descent direction in our proof.

3.2.1 Convergence under Convex Case

In this subsection, we focus on the convergence guarantee of the CGD algorithm when $l(x,y)$ is convex on x. We first give the error bound under general convex case in Theorem 1.

THEOREM 1 **(Error Bound in Convex Case)**. *Suppose that $l(x,y)$ is convex on x and Assumptions 2, 3, 4(b) and 4(c) are satisfied. Denote x_N^r as the rth iteration of CGD algorithm based on an N-samples dataset. Then for any small $\zeta > 0$, there exists N_0, when the sample size $N > N_0$, after k iterations,*

$$\min_{0\le r\le k}\{\mathbb{E}_{f(y;x_N^r,z)}[l(x_N^r,y)] - \mathbb{E}_{f(y;x^*,z)}[l(x^*,y)]\} \le \frac{\varepsilon L_2 \sum_{r=0}^k \eta^r \|x^* - x_N^r\|}{\sum_{r=0}^k \eta^r} + \frac{\|x_N^0 - x^*\|^2 + (L_3^c)^2 \sum_{r=0}^k (\eta^r)^2}{2\sum_{r=0}^k \eta^r} + \frac{3}{2}\zeta.$$

Proof sketch: the main difficulty in this proof is to handle the decision dependent effect: since the distribution of y changes while x moves. We solve this problem by separating out the decision dependent error by Lemma EC.2. Specifically, we convert the expectation gap under different x and distribution $f(y;x,z)$ into the expectation gap under different different x but the same distribution and we get

$$g(x_N^k) - g(x^*) \le \varepsilon L_2 \|x^* - x_N^k\| + \mathbb{E}_{f(y;x_N^k,z)}[l(x_N^k,y) - l(x^*,y)].$$

Note that this conversion causes additional error. And this additional error goes to the first term in the right side and finally become the decision dependent error. For the second term we follow the standard analyze of gradient descent: we first study the recursive relationship of the sequence $\|x_N^k - x^*\|$. Then we substitute the second term into the recursive formula by the convexity of $l(x,y)$, thereby obtaining the relationship

between the second term and $\|x_N^{k+1} - x^*\| - \|x_N^k - x^*\|$. *Finally, we obtain the upper bound of the second term by adding the recursive equation from* $r = 0$ *to k.*

Theorem 1 is a statement about the minimum functional error between the generated sequence x_N^r and the optimal solution x^* after k iterations. Except for the $\frac{3}{2}\zeta$ term which is related to the sample size and approximation error, the remaining error bound can be divided into two parts. The first term on the right-hand side is the bound on the decision-dependent error, which comes from the decision-dependent characteristic (see proof of Theorem 1). We observe that the decision-dependent error depends on the distribution distance ε and the Lipschitz constant L_2, which indicates that the decision-dependent error is influenced by the sensitivity of distribution shift to the variation of decision variable x and random parameter y. We also note that the decision-dependent error will decrease as the solution x^r gets closer to the optimal solution x^*. When the decision space is bounded, the decision-dependent error can be bounded by $\varepsilon L_2 D_X$, where D_X denotes the maximum distance in the decision space (for example, the gap between the maximal and minimum prices in a dynamic pricing problem).

The second term on the right-hand side comes from the internal error of the typical gradient descent algorithm. The only way to decrease the internal error is to continue the iterative process. As the step size η^r is often decreasing and less than 1, the internal error term will converge to zero as k increases.

Although Theorem 1 gives the error bound of the objective function, it is still not clear how the solution sequence converges to the optimal point. Therefore, we investigate the distance to the optimal solution under the strongly convex condition. Mendler-Dünner et al. (2020a) proves the convergence result when the conditional distribution $y|x$ is known in advance. Similar to their work, we prove by bridging the CGD solution and optimal solution by a intermediate stable point.

DEFINITION 3 (STABLE POINT). Under contextual information z, The stable point is the fix point of the following iteration principle:

$$x = \arg\min_x \mathbb{E}_{f(y;x_{PS},z)}[l(x,y)]. \tag{10}$$

We then state the distance bound between the solution sequence of the CGD algorithm and the stable point x_{PS} in Theorem 3.

PROPOSITION 3 **(Distance to stable points)**. *Suppose that Assumptions 2, 3, 5(a) and 5(b) are satisfied,* $l(x,y)$ *is* γ*-strongly convex in x, and at least one stable point* x_{PS} *exists. We denote* $A = \gamma - \varepsilon L_1^c$ *and* $B = L_1^c\sqrt{1+\varepsilon^2}$. *For the case* $A \geq 2B \geq 0$ *we take a constant step size* η *that satisfies*

$$4B^2\eta^2 - 2A\eta + 1 = 0.$$

Then, for any small $\xi > 0$*, there exists a sample size* N_0 *such that, for all* $N > N_0$*, we have the following conclusion after* $k+1$ *iterations:*

Case 1. If $1-2\eta A+2\eta^2B^2>0$*, then*

$$\|x_N^{k+1}-x_{PS}\| \le C^{k+1}\|x_N^1-x^*\| + \xi\eta\frac{1-C^{k+1}}{1-C}, \tag{11}$$

where $C=\sqrt{1-2\eta A+2\eta^2B^2}<1$.

Case 2. If $1-2\eta A+2\eta^2B^2\le 0$*, then for any* $K>0$*, there exists* $k>K$ *such that*

$$\|x_N^{k+1}-x_{PS}\| \le (1+\sqrt{2})\xi\eta. \tag{12}$$

Proof sketch: The main difficulty of this proof is also to handle the error caused by the approximation. Since

$$\|x_N^{k+1}-x_{PS}\|^2 \le \|x_N^k-x_{PS}\|^2 - 2\eta\hat{G}_N(x_N^k)^T(x_N^k-x_{PS}) + (\eta^2)\|\hat{G}_N(x_N^k)\|^2,$$

we can observe that different from the analysis in Mendler-Dünner et al. (2020a), the third term is the contextual gradient rather than the expected gradient. We bound the third term by the unbiased property of contextual gradient combined with Assumption 4(c). For the second term, we can use existing conclusion to bound $\mathbb{E}_{f(y;x_N^k,z)}[\nabla_x l(x_N^k,y)]^T(x_N^k-x_{PS})$*. Thus, we write* $\|x_N^{k+1}-x_{PS}\|^2$ *in the form of* x_N^k *and then we can find the recurrence relation of* $\|x_N^{k+1}-x_{PS}\|^2$*.*

Proposition 3 specifies the distance to the stationary point under the strongly convex condition. In Case 1, the bound can also be divided into two components: the first term comes from the initial distance between x_N^1 and x_{PS} and the parameter C, which is related to the strongly convex parameter γ, the Lipschitz continuous parameter L_1^c, and the distribution distance ε. When the convexity is very strong, and the objective function and decision-dependent distribution do not react sensitively to the decision variables, C is small and the first term decreases rapidly. This result is reasonable because strong convexity increases the converge speed and the decision-dependent effect diminishes as L_1^c and ε decrease. The second term relates to the approximate error. As shown in Proposition 1, ξ becomes sufficiently small when the sample size is large. Therefore, we can reduce the second term by increasing the sample size.

In Case 2, we prove that the distance will decrease when it exceeds the bound $(1+\sqrt{2})\xi\eta$ until it reaches the bound again. Therefore, we can prove that the distance will either decrease or fluctuate around $(1+\sqrt{2})\xi\eta$.

We now focus on the distance to the optimal solution. We have investigated the distance bound between the solution sequence and the stable point x_{PS} in Proposition 3. In the following, we show the relationship between the stable point and the optimal solution.

LEMMA 1 **(Theorem 4.3 in (Mendler-Dünner et al. 2020a))**. *Suppose that* $l(x,y)$ *is* L_y*-Lipschitz in y and strongly convex, and that Assumption 3 is satisfied. Then, for every stable point* x_{PS}*, we have*

$$\|x^*-x_{PS}\| \le \frac{2L_y\varepsilon}{\gamma}.$$

Lemma 1 gives the distance bound between the stable point and the optimal solution in the strongly convex case. Therefore, we can now state the upper bound of the solution error of CGD algorithm.

THEOREM 2 **(Error bound under strongly convex case)**. *Denote x^* as the optimal solution of $g(x) = \mathbb{E}_{f(y;x,z)}[l(x,y)]$ and $l(x,y)$ is $\gamma-$strongly convex. Assume that $\gamma - \varepsilon L_1^c \geq 2L_1^c\sqrt{1+\varepsilon^2}$, and set the step size η according to Proposition 3. After k iterations, for any $\xi > 0$, there exists a sample size N_0 such that, if $N > N_0$, the solution gap is bounded by*

$$|x_N^k - x^*| \leq \frac{2L_2\varepsilon}{\gamma} + \max\left\{C^{k+1}|x_N^1 - x^*| + \xi\eta\frac{1-C^k}{1-C}, (1+\sqrt{2}\xi\eta)\right\},$$

where C is defined in Proposition 3.

In summary, when the objective function is strongly convex, we first prove the distance bound to an intermediate stable point. Since the stable point is close to the optimal solution, we can then prove the distance bound to the optimal solution of the CGD algorithm under the strongly convex case. The error bound in Theorem 2 can also be divided into two parts. The first term is related to the decision-dependent parameter ε, implying that it stems from the decision-dependent effect of the contextual model. The second part is the inertial error of the CGD algorithm, which is decreasing to the iteration number k and related to the choice of step size η.

3.2.2 Extension to Non-convex Case

Up to now, we have assumed the convexity of the objective function. However, in most practical settings the objective function non-convex to the decision variables. In the dynamic pricing models, the revenue may have a complex functional relationship to the pricing decision. To investigate the performance of the CGD algorithm under more general cases, we extend the convergence analysis to this non-convex case. Similar to the convergence result of typical gradient descent, we study the convergence of CGD algorithm to a stationary point, that is, the point x^* where $\mathbb{E}_{f(y;x^*,z)}[\nabla_x l(x^*,y)]$ is small.

PROPOSITION 4 **(Convergence under diminishing step size)**. *Suppose Assumptions 2, 3, 1, 5 and 4(a) hold, that the objective function $l(x,y)$ is twice differentiable in x and its absolute value is bounded by a constant L_4. If the gradient of the distribution density is also bounded by a constant L_5, and the step size η^r is diminishing with $\sum_{r=0}^{\infty}\eta^r = \infty$, there exists a sample size N_0, when $N > N_0$, any limit point of the sequence generated by the CGD algorithm is a stationary point of the cost gradient expectation.*

$$\text{if } \lim_{N\to\infty}\lim_{r(\in\mathcal{K})\to\infty} x_N^r = \bar{x}, \text{ then } \mathbb{E}_{f(y;\bar{x},z)}[\nabla_x l(\bar{x},y)] = 0. \tag{13}$$

PROPOSITION 5 **(Convergence under Armijo step size)**. *Under Assumptions 2, 3, and 4(a), suppose that the CGD algorithm adopts the Armijo step size with σ, and that the sample size N is sufficiently large. Then, any limit point $\bar{x}$ of the sequence generated by the CGD algorithm has a bounded expected gradient.*

$$\text{if } \lim_{N\to\infty}\lim_{r(\in\mathcal{K})\to\infty} x_N^r = \bar{x}, \text{ then } \|\mathbb{E}_{f(y;\bar{x},z)}[\nabla_x l(\bar{x},y)]\| \leq \frac{\varepsilon L_1}{1-\sigma}. \tag{14}$$

Propositions 4 and 5 state that the CGD algorithm converges to the stationary point of the expected gradient. Note that this conclusion relies on Assumptions 1 and 5, which are strong conditions and may not be generally satisfied. Furthermore, when the constants S_Ω, L_4, L_5 are large, this convergence result may have a poor performance in practice. Compared with the diminishing step size, the expected gradient for CGD with Armijo step size is not guaranteed to converge to 0, but this convergence result holds under a milder condition where Assumptions 1 and 5 may not hold. And we can then derive from Theorem 3 that (14) is a necessary condition of optimality.

Then we investigate the convergence rate of the CGD algorithm. Compared with typical gradient descent, which only requires $O(1/\varepsilon^2)$ iterations to obtain an $\varepsilon-$stationary solution, the CGD algorithm also requires $O(1/\varepsilon^2)$ steps to converge to a range with expected gradient upper bound.

PROPOSITION 6. *Suppose Assumptions 2, 3, 4 and 5(a) hold, when $N \to \infty$, with fix step size $\eta \leq \min\{\frac{1}{L_{1g}}, \frac{1}{L_3^c}\}$, we have*

$$\min_{r=0,\ldots,k} \|\mathbb{E}_{f(y;x_N^r,z)}[\nabla_x l(x_N^r, y)]\|^2 \leq \frac{2(\mathbb{E}_{f(y;x_N^0,z)}[l(x_N^0, y)] - \mathbb{E}_{f(y;x^*,z)}[l(x^*, y)])}{\eta(k+1)} + \frac{L_3^c L_1 \varepsilon}{2}. \tag{15}$$

Like the convergence result in Proposition 5, Proposition 6 shows that CGD with constant step size will converge to a point with limited expected gradient, and the convergence rate is $O(\varepsilon^{-2})$, which corresponds to the $O(\varepsilon^{-2})$ lower bound of Agarwal et al. (2012). The bias term $\frac{L_3^c L_1 \varepsilon}{2}$ to stationary point implies the error caused by decision dependency. In specific, it rises from the heterogeneous distribution under different decisions.

The above convergence results are relevant to the expected gradient. However, according to Proposition 2, the expected gradient is not equal to the gradient of expected objective function. In other words, the station point of expected gradient is not the stationary point of the true expected objective function $g(x)$ in (1) if we analogize the CGD algorithm to the typical GD algorithm. Though this inconsistency exists, we prove that similar to the GD algorithm, the convergence to the expected gradient is a necessary condition of optimality.

THEOREM 3 **(Necessary condition of optimality)**. *If x^* is the optimal solution for a decision-dependent problem $\min_x \mathbb{E}_{f(y;x,z)}[l(x, y)]$, where l is an L_1 Lipschitz function and Assumptions 2 and 3 are satisfied, then $\|\mathbb{E}_{f(y;x^*,z)}[\nabla_x l(x^*, y)]\| \leq L_1 \varepsilon$.*

Theorem 3 builds a connection between the CGD algorithm and optimality condition. It indicates that one necessary condition for optimality is that the norm of the expected gradient should not be too large. For the diminishing step size, the expected gradient will be sufficiently small, thus satisfying the necessary condition. For the Armijo step size, the necessary bound in Theorem 3 is actually the upper bound when $\sigma = 0$ in Proposition 5. Therefore, although the converging point of CGD algorithm is not the zero point of the expected objective function, it still satisfies the necessary condition of optimality.

4 Numerical Results

In this section, we validate the convergence performance of the CGD algorithm and compare its performance against other methods. In our experiments, we use both simulated data and real-world data from the electricity industry to validate the effectiveness of the proposed CGD algorithm. Specifically, we compare the performance of the proposed CGD algorithm with the prescriptive approach in Bertsimas and Kallus (2019) and the estimate-then-optimize approach under the linear decision assumption. All computations were carried out in Python 3.10 on an Intel i7-9750H processor with 32.0 GB of RAM.

4.1 Data Description and Experiment Setup

We conduct numerical experiments on two datasets. The first dataset comes from a real-world power plant pricing scenario. This dataset describes the electricity demand and price situation in an electricity plant. The factors affecting the electricity demand include temperature, solar exposure, etc. On a daily basis, the manager needs to decide on the electricity price to maximize the revenue. The main challenge is that the manager does not know the functional relationship between price and demand under the current features, and can only estimate the demand based on historical pricing and current feature. Note that the intuitive inverse relationship between electricity consumption and price may not be clear in the dataset, since a lower demand may lead to a lower pricing decision in practice. This also reflects the importance of decision-dependency in our model. The source of the real-world dataset is given in Section EC.4.1. We denote this dataset as D_{real}.

The second dataset contains simulated data. We generated demand, price, and feature data from a known distribution and functional relationship. The aim was to maximize the revenue through optimal pricing and order quantity decisions. The underlying demand distribution and parameter values of this dataset are given in Section EC.4.1. We denote this dataset as D_{simu}.

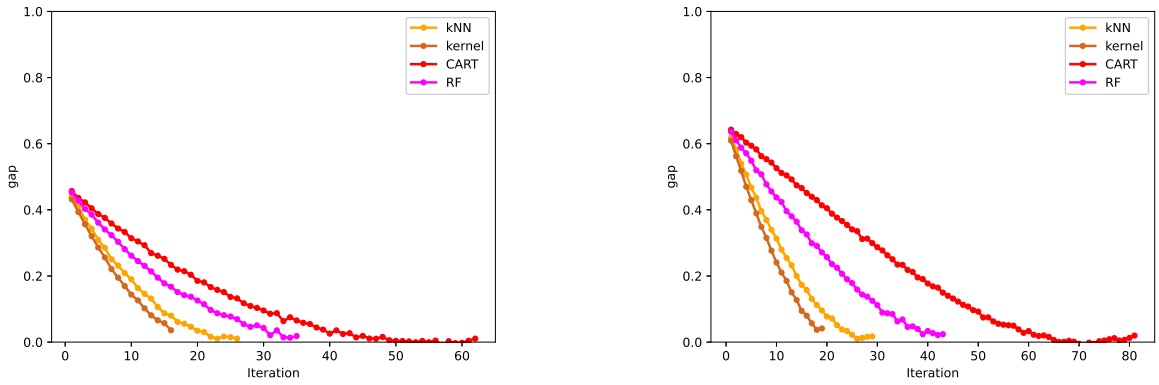
In our experiment, we first calculate the contextual gradient by (7). The hyperparameters of each ML method (e.g., kNN, kernel regression, CART, and RF weighting) are tuned through a grid search. The initial values are $(p_0, q_0) = (15, 30)$ in D_{simu} and $p_0 = \text{pmid}(D_{real})$ in D_{real}, where $\text{pmid}(D_{real})$ denotes the median of historical pricing decision. In the simulated dataset D_{simu}, we evaluate the performance by the *optimality gap*, *i.e.*, $(\mathbb{E}_{f(y;x^*,z)}[l(x^*,y)] - \mathbb{E}_{f(y;x,z)}[l(x,y)])/\mathbb{E}_{f(y;x^*,z)}[l(x^*,y)]$. Note that the true distribution $f(y;x,z)$ is known while evaluation, while it is unknown when we are solving the problem by CGD algorithm. In the real dataset D_{real}, however, we do not know the true distribution, hence we cannot calculate the true revenue of the output decision. In this case, we evaluate the solution quality by comparing the output decision to the practical pricing decision in the test dataset. For any (x^j, z^j, y^j) in the test set, the CGD algorithm outputs a solution $\hat{x}$ by z_j, and the performance is evaluated by the deviation $|\hat{x} - x^j|$. Since the demand of electricity companies is similar at the same time of year, we can assume that their pricing decisions are relatively reasonable in practice. Therefore, it can be a benchmark to evaluate the solution quality of CGD algorithm.

4.2 Convergence Performance

In this section, we validate the convergence performance of the CGD algorithm from different aspects.

Convergence under convex case.

We first perform the convergence of CGD algorithm under different weighting approaches under the convex price-only newsvendor pricing problem (*i.e.*, fixing the ordering decision) in the simulated dataset D_{simu}. The iteration stops when the step size is below 10^{-5} or the solution exceeds the upper or lower bound, the convergence results are shown in Figure 1(a). We also extend to a strongly convex case by adding a quadratic penalize term into the objective function $l(x,y)$. the convergence results under the strongly convex case are shown in Figure 1(b). We observe that in both convex and strongly convex cases, the optimality gaps are less than 5%. We also observe that each algorithm has a slight deviation from the optimal solution before the iterations stop. This deviation reflects the approximation error of the weight SAA approximation method. This illustrates why we use the Armijo rule to select the step size: the Armijo rule ensures that the estimated function value decreases monotonically, that is, the true function value will not deviate significantly from the local minimum. A detailed performance comparison between two step sizes is provided in Section EC.4.2.



(a) Convergence under convex case

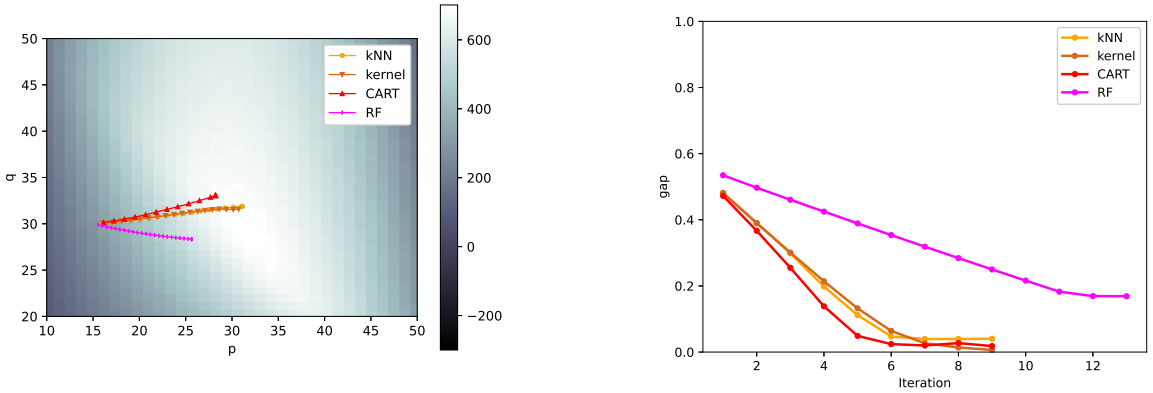
(b) Convergence under strongly-convex case

Figure 1 Comparison CGD algorithm with different weight under convex and non-convex conditions on D_{simu}.

Convergence under non-convex case.

We then investigate the convergence of the algorithm under the non-convex setting. Specifically, we jointly optimize the pricing and ordering decision in the newsvendor pricing problem on the simulated dataset D_{simu}, where $l(x,y)$ is non-convex to the decisions $x=(p,q)$. The performance of our algorithm is evaluated by the true expected objective function, which is calculated by the predetermined demand distribution in the simulation setting. We can observe from Figure 2(a) that all four models descend toward the local optimum.

Kernel regression and CART exhibit the best performance, indicating that they have more accurate estimates of profit and gradient. Note that the priority of these two weighting methods does not always hold true, but depends on whether the weight method gives an accurate prescription to the conditional distribution of the demand. The optimality gaps to the local optimum are shown in Figure 2(b), the CGD algorithm has a relatively high convergence rate. Among the four weight functions, the optimality gaps of kNN, kernel regression, and CART are less than 5%.



(a) Process snapshot (b) Optimality gap

Figure 2 Convergence of CGD algorithm with different weight under non-convex condition on D_{simu}.

Sample Efficiency.

To verify the performance of the model under small sample conditions, we study the performance of the CGD algorithm under different sample sizes on D_{simu}. We use the kNN weight method with $k = 20$ and generate five stochastic features for each sample size. The average, maximum, and minimum optimality gaps are shown in Figure 3. As expected, the CGD algorithm converges to a stationary point of the true objective with a small sample size. When the sample size is large, the optimality gap becomes more stable and decreases correspondingly, because a larger sample size provides more information about the true distribution.

Performance on Practical Contextual Pricing Problem

We then test the performance of the CGD algorithm under the practical pricing problem in the electricity industry. We divided D_{real} into two parts, a training set (n = 1895, 90%) and a test set (n = 211, 10%). We learn the contextual gradient based on the training set before optimizing the pricing decision based on current contextual information in the test set. We evaluate the performance of CGD algorithm by comparing to the practical decision in the test set. We can observe from Figure 4 that the difference between most

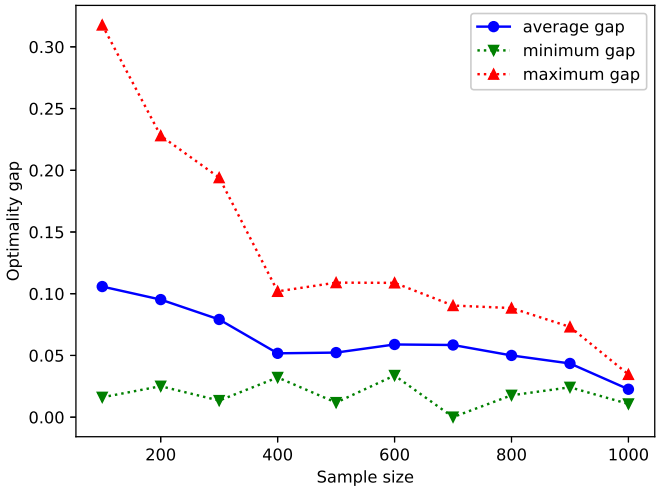


Figure 3 Optimality gap under different sample sizes on D_{simu}.

pricing decisions given by the CGD algorithm and the actual pricing is less than 5%. In summary, the CGD algorithm can effectively learn the contextual information and make a reasonable pricing decision that is close to the practice.

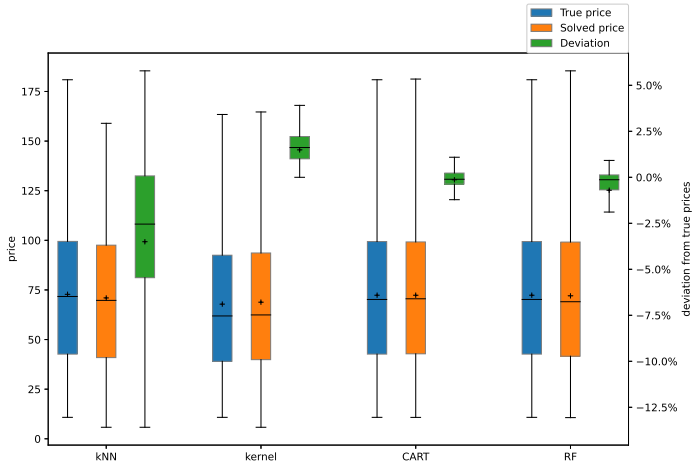


Figure 4 Comparison between real pricing decision and output pricing decision on D_{real}.

4.3 Comparison to Other Policies

In this section, we compare the proposed CGD algorithm with other approaches that can potentially solve the contextual optimization problem under decision-dependent effect. The first benchmark is proposed by Bertsimas and Kallus (2019), which is to solve the weighted-SAA prescriptive model in (4) directly by discretization (PRE+DIS). Another benchmark is an estimate-then-optimize framework that adopts the linear

decision rule and builds a linear regression model of y to the decision variable x and contextual information z before optimizing the decision under the regression model (LR+OPT). In the PRE+DIS approach, we first build the prescriptive model according to (4) before finding the best solution by exploring all potential pricing and ordering decisions. The LR+OPT approach imposes a linear context assumption to the distribution $y|x,z$. The linear assumption is representative since it is widely used in estimate-then-optimize models and one can improve its generalization ability by transforming the covariate variables z (Ban and Rudin 2018, Demirovic et al. 2019). Specifically, the LR+OPT framework first estimates a linear regression model $\hat{y}(x,z) = \alpha_0 + \alpha^T(x,z)$, where the $alpha_0$ and α are the coefficients of the linear model. Then it substitutes the model parameter y by $\hat{y}(x,z)$ before optimizing x directly.

Comparison to PRE+DIS

In the following, we compare the CGD algorithm with the discretization solution of the corresponding weighted-SAA prescriptive model on D_{simu}. In each pair of comparisons, we adopt the same ML estimate model with the same hyperparameter setting. We compare both the optimality gap and running time. Table 1 documents the complete numerical results for the comparison between the CGD and PRE+DIS under each ML estimate model. Results show that the CGD algorithm gains higher revenue than the discretization method. And the time consumption of CGD is at least ten times less than PRE+DIS. We can also observe that the optimality gap of PRE+DIS strategy under random forest estimation is extremely large. This indicates that the performance of discretization relies highly on the precision of estimation, while the CGD algorithm can still descend towards a correct direction even though the estimation is not so accurate. Therefore, compared with the prescription then discretization method, the CGD algorithm is more robust to the estimation quality.

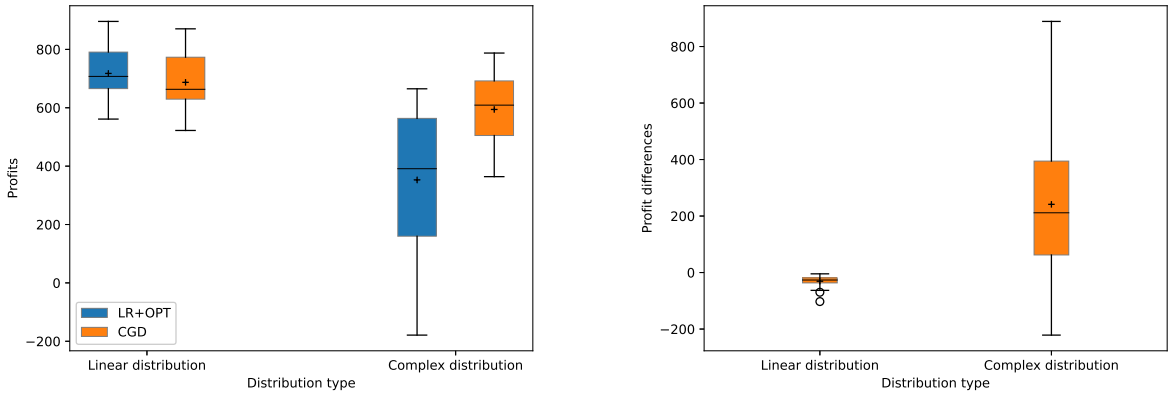
Table 1 Optimality gap and running time comparison between CGD and PRE+DIS.

Strategies	kNN		kernel		CART		RF	
	gap	time (sec)	gap	time (sec)	gap	time (sec)	gap	time (sec)
PRE+DIS	10.27%	60.37	0.94%	118.14	0.96%	14.11	114.44%	91.75
CGD	1.48%	1.92	0.56%	2.04	1.99%	1.07	4.97%	5.87

Comparison to LR+OPT

Another potential solution to the contextual optimization problem under decision-dependency in existing literature is to adopt the LR+OPT framework. To illustrate the generalization ability of both the CGD algorithm and LR+OPT, we generated two sets of demand samples, one with a simple linear decision rule, where the demand y is exactly linear in p,q and z ($y = 60 - p + \mathbf{1}^T z + \varepsilon$). This dataset is denoted as D_{lin}. Another group of demand samples follows a complex multiplicative relation to the decision variable x and

we denote this dataset as D_{mul}, and the parameter setting is provided in EC.4.1. We evaluate the profit performance on both two datasets of the CGD algorithm and LR+OPT framework respectively.



(a) Profits comparison between CGD and LR+OPT under two demand models

(b) Profit differences of CGD and LR+OPT under two demand models

Figure 5 Comparison between CGD and LR+OPT approaches under two demand models

The results of our experiments are shown in Figure 5, where Figure 5(a) denotes the profit performance on the test dataset, and Figure 5(b) shows the difference of CGD profit minus the LR+OPT profit on the test dataset. We can observe that on the linear demand dataset D_{lin}, the LR+OPT outperforms the CGD method. This result is not surprising because the true demand model satisfies the demand prediction assumptions exactly. We can observe that the gap between these two methods is not significant and CGD method still performs well in this case. In contrast, in the complex demand distribution scenario D_{mul}, our CGD method significantly outperforms the LR+OPT framework. Moreover, when the demand prediction assumption deviates from the true distribution, the LR+OPT strategy sometimes generates lower profits and may fail to converge. This result illustrates the generalization ability of the CGD algorithm. Compared with the LR+OPT framework, the CGD algorithm adopts a distribution-free and nonparametric setting, making it generalizable to complex distribution cases.

In summary, when there is little information about the distribution of stochastic parameters, adopting the distribution-free method results in better adaptation to a range of real-world scenarios, leading to more robust solutions than assuming a specific distribution and decision-dependency rule for the stochastic parameters.

5 Conclusion and Future Directions

In this paper, we propose a novel approach to solve the contextual optimization problem under decision dependency. Compared with existing policies, the contextual gradient retains the first-order information of

the objective function, and thus is more efficient than the discretization approach. Our CGD algorithm also has a strong theoretical convergence guarantee under both the convex and non-convex cases and has a great generalization ability because the method is fully nonparametric.

Much remains open and requires further investigation. First, there may exist other algorithm designs based on contextual gradient. For example, one can embed the contextual gradient into the proximal gradient descent algorithm and stochastic gradient descent algorithm. One may characterize distance convergence properties. Second, In this paper, the convergence result is limited to the unconstrained setting. Efficient algorithms are still absent for solving the constraint contextual optimization problem under decision dependency.

References

Agarwal, Alekh, Peter L. Bartlett, Pradeep Ravikumar, Martin J. Wainwright. 2012. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *IEEE Transactions on Information Theory*, 58 (5), 3235-3249. doi:10.1109/TIT.2011.2182178.

Ban, Gah-Yi, Cynthia Rudin. 2018. The big data newsvendor: Practical insights from machine learning. *Operations Research*, 67 (1), 90-108.

Bertsekas, Dimitri P. 1999. *Nonlinear Programming*. Athena Scientific.

Bertsimas, Dimitris, Nathan Kallus. 2019. From predictive to prescriptive analytics. *Management Science*, 66 (3), 1025-1044.

Bertsimas, Dimitris, Nihal Koduri. 2022. Data-driven optimization: A reproducing kernel hilbert space approach. *Operations Research*, 70 (1), 454-471. doi:10.1287/opre.2020.2069.

Bertsimas, Dimitris, Christopher McCord. 2019. Optimization over continuous and multi-dimensional decisions with observational data. *Neural Information Processing Systems*.

Besbes, Omar, Assaf Zeevi. 2009. Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. *Operations Research*, 57 (6), 1407-1420. doi:10.1287/opre.1080.0640. URL `https://doi.org/10.1287/opre.1080.0640`.

B.Folland, Gerald. 1999. *Real Analysis Modern Techniques and Their Applications 2nd Edition*.

Biswas, Indranil, Balram Avittathur. 2018. The price-setting limited clearance sale inventory model. *Annals of Operations Research*, .

Butler, Andrew, Roy H. Kwon. 2023. Gradient boosting for convex cone predict and optimize problems. *Operations Research Letters*, 51 (1), 79-83.

Chen, Boxiao, Xiuli Chao, Hyun-Soo Ahn. 2019. Coordinating pricing and inventory replenishment with nonparametric demand learning. *Operations Research*, 67 (4), 1035-1052. doi:10.1287/opre.2018.1808. URL `https://doi.org/10.1287/opre.2018.1808`.

Cheung, Wang Chi, David Simchi-Levi. 2019. Sampling-based approximation schemes for capacitated stochastic inventory control models. *Mathematics of Operations Research*, 44 (2), 668-692.

Cristian, Rares, Pavithra Harsha, Georgia Perakis, Brian Quanz, Ioannis Spantidakis. 2022. End-to-end learning via constraint-enforcing approximators for linear programs with applications to supply chains. URL https://api.semanticscholar.org/CorpusID:259953167.

Demirovic, Emir, Peter James Stuckey, James Bailey, Jeffrey Chan, Christopher Leckie, Kotagiri Ramamohanarao, Tias Guns. 2019. Predict+optimise with ranking objectives: Exhaustively learning linear functions. *International Joint Conference on Artificial Intelligence*. URL https://api.semanticscholar.org/CorpusID:199465744.

den Boer, Arnoud V. 2015. Dynamic pricing and learning: Historical origins, current research, and new directions. *Surveys in Operations Research and Management Science*, 20 (1), 1-18. doi:https://doi.org/10.1016/j.sorms.2015.03.001. URL https://www.sciencedirect.com/science/article/pii/S1876735415000021.

Donti, Priya L., Brandon Amos, J. Zico Kolter. 2017. Task-based end-to-end model learning in stochastic optimization. *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17, Curran Associates Inc., Red Hook, NY, USA, 5490–5500.

Dupačová, Jitka. 2006. Optimization under exogenous and endogenous uncertainty. doi:10.13140/2.1.2682.2089.

El Balghiti, Othman, Adam N. Elmachtoub, Paul Grigas, Ambuj Tewari. 2022. Generalization bounds in the predict-then-optimize framework. *Mathematics of Operations Research*, 48 (4), 2043-2065. doi:10.1287/moor.2022.1330. URL https://pubsonline.informs.org/doi/abs/10.1287/moor.2022.1330.

Elmachtoub, Adam N., Paul Grigas. 2022. Smart "predict, then optimize". *Management Science*, 68 (1), 9-26.

Feng, Qi, J. George Shanthikumar. 2022. Developing operations management data analytics. *Production and Operations Management*, 31 (12), 4544-4557.

Godfrey, Gregory A., Warren B. Powell. 2001. An adaptive, distribution-free algorithm for the newsvendor problem with censored demands, with applications to inventory and distribution. *Management Science*, 47 (8), 1101-1112.

Goel, Vikash, Ignacio E. Mathematical Programming Grossmann. 2006. A class of stochastic programs with decision dependent uncertainty. *Mathematical Programming*, 108 355-394.

Grigas Paul, Zuo Jun Shen, Qi Meng. 2023. Integrated conditional estimation-optimization.

Harsha, Pavithra, Ramesh Natarajan, Dharmashankar Subramanian. 2021. A prescriptive machine-learning framework to the price-setting newsvendor problem. *INFORMS Journal on Optimization*, 3 (3), 227-253.

Homem-de Mello, Tito. 2001. *Monte Carlo Methods for Discrete Stochastic Optimization*. Springer US, Boston, MA, 97-119.

Huber, Jakob, Sebastian Müller, Moritz Fleischmann, Heiner Stuckenschmidt. 2019. A data-driven newsvendor problem: From data to decision. *European Journal of Operational Research*, 278 (3), 904-915. doi:https://doi.org/10.1016/j.ejor.2019.04.043. URL https://www.sciencedirect.com/science/article/pii/S0377221719303807.

Jasin, Stefanus, Chengyi Lyu, Sajjad Najafi, Huanan Zhang. 2024. Assortment optimization with multi-item basket purchase under multivariate mnl model. *Manufacturing & Service Operations Management*, 26 (1), 215-232. doi:10.1287/msom.2021.0526. URL https://doi.org/10.1287/msom.2021.0526.

Jeong, Jihwan, Parth Jaggi, Andrew Butler, Scott Sanner. 2022. An exact symbolic reduction of linear smart predict+optimize to mixed integer linear programming. *International Conference on Machine Learning*. URL https://api.semanticscholar.org/CorpusID:250340791.

Kallus, Nathan, Xiaojie Mao. 2022. Stochastic optimization forests. *Management Science*, 69 (4), 1975-1994. doi:10.1287/mnsc.2022.4458. URL https://doi.org/10.1287/mnsc.2022.4458.

Kallus, Nathan, Madeleine Udell. 2020. Dynamic assortment personalization in high dimensions. *Operations Research*, 68 (4), 1020-1037. doi:10.1287/opre.2019.1948. URL https://doi.org/10.1287/opre.2019.1948.

Kannan, Rohit, Guzin Bayraksan, James R. Luedtke. 2022. Data-driven sample average approximation with covariate information.

Kazaz, Burak, Scott Webster. 2011. The impact of yield-dependent trading costs on pricing and production planning under supply uncertainty. *Manufacturing & Service Operations Management*, 13 (3), 404-417.

Kleywegt, Anton J., Alexander Shapiro, Tito Homem-de Mello. 2002. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12 (2), 479-502.

Larson, Jeffrey, Matt Menickelly, Stefan M. Wild. 2019. Derivative-free optimization methods. *Acta Numerica*, 28 287 - 404.

Lee, Sanghyuk, Seunghyun Lee, Byung Song. 2022. Contextual gradient scaling for few-shot learning. 3503-3512. doi:10.1109/WACV51458.2022.00356.

Levi, Retsef, Georgia Perakis, Joline Uichanco. 2015. The data-driven newsvendor problem: New bounds and insights. *Operations Research*, 63 (6), 1294-1306.

Levi, Retsef, Robin O. Roundy, David B. Shmoys. 2007. Provably near-optimal sampling-based policies for stochastic inventory control models. *Mathematics of Operations Research*, 32 (4), 821-839.

Li, Shukai, Qi Luo, Zhiyu Huang, Cong Shi. 2022. Online learning for constrained assortment optimization under markov chain choice model. *SSRN Electronic Journal*, URL https://api.semanticscholar.org/CorpusID:248333022.

Lin, Shaochong, Youhua Chen, Yanzhi Li, Zuo-Jun Max Shen. 2022. Data-driven newsvendor problems regularized by a profit risk constraint. *Production and Operations Management*, 31 (4), 1630-1644.

Liu, Junyi, Guangyu Li, Suvrajeet Sen. 2021. Coupled learning enabled stochastic programming with endogenous uncertainty. *Mathematics of Operations Research*, 47 (2), 1681-1705.

Liu, Tianyi, Yifan Lin, Enlu Zhou. 2024. Bayesian stochastic gradient descent for stochastic optimization with streaming input data. *SIAM Journal on Optimization*, 34 (1), 389-418. doi:10.1137/22M1478951.

Luo, Fengqiao, Sanjay Mehrotra. 2020. Distributionally robust optimization with decision dependent ambiguity sets. *Optimization Letters*, 14 (8), 2565-2594. doi:10.1007/s11590-020-01574-3. URL https://doi.org/10.1007/s11590-020-01574-3https://link.springer.com/content/pdf/10.1007/s11590-020-01574-3.pdf.

Mandi, Jayanta, Emir Demirovi, Peter Stuckey, Tias Guns. 2020. Smart predict-and-optimize for hard combinatorial optimization problems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34 1603-1610. doi:10.1609/aaai.v34i02.5521.

Mandi, Jayanta, Tias Guns. 2020. Interior point solving for lp-based prediction+optimisation. *ArXiv*, abs/2010.13943. URL https://api.semanticscholar.org/CorpusID:225076353.

Mendler-Dünner, Celestine, Juan Perdomo, Tijana Zrnic, Moritz Hardt. 2020a. Stochastic optimization for performative prediction. *International Conference on Machine Learning*, 7599-7609.

Mendler-Dünner, Celestine, Juan C. Perdomo, Tijana Zrnic, Moritz Hardt. 2020b. Performative prediction. *arXiv:2006.06887*, .

Noyan, Nilay, Gábor Rudolf, Miguel Lejeune. 2021. Distributionally robust optimization under a decision-dependent ambiguity set with applications to machine scheduling and humanitarian logistics. *INFORMS Journal on Computing*, 34 (2), 729-751. doi:10.1287/ijoc.2021.1096. URL https://doi.org/10.1287/ijoc.2021.1096.

Qin, Hanzhang, David Simchi-Levi, Li Wang. 2022. Data-driven approximation schemes for joint pricing and inventory control models. *Management Science*, 68 (9), 6591-6609.

Rubner, Yossi, Carlo Tomasi, Leonidas J. Guibas. 2000. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40 (2), 99-121.

Sadana, Utsav, Abhilash Chenreddy, Erick Delage, Alexandre Forel, Emma Frejinger, Thibaut Vidal. 2023. A survey of contextual optimization methods for decision making under uncertainty.

Salinger, Michael, Miguel Ampudia. 2011. Simple economics of the price-setting newsvendor problem. *Management Science*, 57 (11), 1996-1998.

Srivastava, P.R., Yijie Wang, Grani Adiwena Hanasusanto, Chin Pang Ho. 2021. On data-driven prescriptive analytics with side information: A regularized nadaraya-watson approach.

Wang, Chong, Xu Chen. 2015. Optimal ordering policy for a price-setting newsvendor with option contracts under demand uncertainty. *International Journal of Production Research*, 53 (20), 6279-6293.

Wilder, Bryan, Bistra N. Dilkina, Milind Tambe. 2018. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. *ArXiv*, abs/1809.05504. URL https://api.semanticscholar.org/CorpusID:52281733.

Zhang, Yanfei, Junbin Gao. 2017. Assessing the performance of deep learning algorithms for newsvendor problem. Derong Liu, Shengli Xie, Yuanqing Li, Dongbin Zhao, El-Sayed M. El-Alfy, eds., *Neural Information Processing*. Springer International Publishing, Cham, 912-921.

E-Companion for Tackling Decision Dependency in Contextual Stochastic Optimization

EC.1 Definition of Weight Functions

In this section, we present some definitions of the weight functions that can be used to construct the approximate model (4).

DEFINITION EC.1 (KERNEL REGRESSION WEIGHT). We can use the kernel function that measures the distances in (x,z) to construct the weight function:

$$w^{\mathrm{KR},i}(x,z) = \frac{K_h\left((x,z)-(x^i,z^i)\right)}{\sum_{j=1}^{n} K_h((x,z)-(x^j,z^j))}, \tag{EC.1}$$

where $K_h : \mathbb{R}^{dim(z)+1} \to \mathbb{R}$ is the kernel function with bandwidth h. Common kernel functions include the uniform kernel, triangular kernel and Gaussian kernel. If not noted, the kernel functions below refer to the Guassian kernel function:

$$K(z) = \frac{1}{\sqrt{2\pi}}\exp^{-\|z\|_2^2/2}. \tag{EC.2}$$

DEFINITION EC.2 (CART WEIGHT). The CART weight functions are given by:

$$w^{\mathrm{CART},i}(x,z) = \frac{\mathbb{I}\{R(x,z)=R(x^i,z^i)\}}{|\{j : R(x^j,z^j)=R(x,z)\}|}, \tag{EC.3}$$

where $R : \mathcal{X}\times Z \to \{1,...,r\}$ is the function that maps features to the r leaves on the CART. In the CART, a leaf is a collection of sample points that are classified to the same group.

DEFINITION EC.3 (RANDOM FOREST WEIGHT). The random forest weight functions are given by:

$$w^{\mathrm{RF},i}(x,z) = \frac{1}{N_E}\sum_{e=1}^{N_E} w^{\mathrm{CART},i,e}(x,z), \tag{EC.4}$$

where N_E is the number of estimators in the random forest, and $w^{\mathrm{CART},i,e}(x,z)$ is the CART weight of the eth estimator in the random forest.

One of the advantage of random forest weight is that the variance will not get large as N_E increases, while the estimation will be more accurate. The only cost is that it will consume more time to calculate the random forest weight if N_E become larger.

EC.2 Description of Solution Methods

In this section, we explain the solution methods adopted in the numerical experiment section.

EC.2.1 Diminishing Step

The diminishing step adopt the step size η^r such that $\eta^r > \eta^{r+1}$ and $\sum_{r=0}^{\infty}\eta^r = \infty$. A typical choice is $\eta^r = C/(r+1)$, where C is a constant that can be adjusted to suit different problems.

EC.2.2 Armijo Step

Let $f(\cdot)$ denote the objective function we want to minimize. The Armijo principle chooses the step size η^r by the following steps (we denote the ascent direction as d^r) in algorithm 2

Algorithm 2 Armijo step size

Input: iteration solution x^r, contextual information z, α_0, $\beta \in (0,1)$, $\sigma \in [0,1)$, tolerance ε.

Output: step size η^r.

1: $\eta^r = \alpha_0$

2: $x^{r+1} = x^r + \eta^r d^r$;

3: **while** $\eta^r \geq \varepsilon$ and $f(x^r) - f(x^{r+1}) < \sigma\eta^r(\hat{G}_N(x^r;z))^T d^r$ **do**

4: $\quad \eta^r = \eta^r * \beta$;

5: $\quad x^{r+1} = x^r + \eta^r d^r$;

6: **end while**

7: **return** η^r

Note that the hyperparameter σ can be 0 in our problem. When $\sigma = 0$, the armijo step size ensure that the objective function descent in an approximate context. We also show the special meaning when $\sigma = 0$ in Proposition 5.

EC.3 Proofs

Proof of Proposition 1

The proof Proposition 1 roughly follows the proof of Theorem EC.9 in Bertsimas and Kallus (2019). However, there are difference between them since Proposition 1 is about the convergence of derivative function rather than the objective function.

Specifically, for every x, the marginal distribution of $y \sim f(y;x,z)$ is independent of y conditioned on z, the ignorability assumption satisfies. Furthermore, The feasible region for x is nonempty, and we only restrict the up and down limit of the two decisions.

Therefore, we need to prove that the expected gradient $\mathbb{E}[\nabla_x l(x,y)|x = x', z = z']$ is bounded and equicontinuous on x. First, from Assumption 4(c) we have $|\nabla_x l(x,y)| < \infty$ for every $x \in X$ and $y \in Y$, thus $\liminf_{x\in X, \|x\|\to\infty} \inf_{y\in Y} |\nabla_x l(x,y)| < \infty$. Then from Assumption 5, for any $x \in X, \varepsilon > 0, x's.t. \|x - x'\| \leq \varepsilon/L_{1g}$,

$$\begin{aligned} \|\nabla_x l(x,y) - \nabla_x l(x',y')\| &\leq L_{1g}\|x' - x\| \\ &\leq \varepsilon. \end{aligned}$$

Thus $\nabla_x l(x,y)$ is equicontinuous. Then the proof is completed by Theorem EC.9 in Bertsimas and Kallus (2019).

Proof of Proposition 2

Assume that Assumption 1 holds. We rewrite the objective expectation to the integrate form:

$$\nabla_x \mathbb{E}_{f(y;x,z)}[l(x,y)] = \nabla_x \int_{y\in\Omega} l(x,y) f(y;x,z) dy.$$

Suppose that the derivative of $l(x,y), f(y;x,z)$ can be bounded by an L^1 function g for all x,y, then the derivative and integration operator can be switched.

$$\begin{aligned}
\nabla_x \mathbb{E}_{f(y;x,z)}[l(x,y)] &= \int_{y\in\Omega} \nabla_x [l(x,y) f(y;x,z)] dy \\
&= \int_{y\in\Omega} (\nabla_x l(x,y))\, f(y;x,z) dy \\
&\quad + \int_{D\in\Omega} (\nabla_x f(y;x,z))\, l(x,y) dy \\
&= \mathbb{E}_{D\sim f_D(p,z)}[\partial_{p,q} l(x,y)] \\
&\quad + \int_{D\in\Omega} (\nabla_x f(y;x,z))\, l(x,y) dy.
\end{aligned}$$

Therefore, the equality holds only when the second term of the last equation equals to 0, which is not guaranteed. So the expectation of cost gradient do not equal to the gradient of objective expectation and thus the convergence of approximate gradient fails.

Before we begin to proof the convergence results, we first state some important results. The following lemmas show how Assumption 3 affects the distance between expectations of different distributions.

LEMMA EC.1. *Kantorovich-Rubinstein For all function f that is $1-$Lipschitz*

$$\|\mathbb{E}_{d\sim D(p)}\mathbb{E}[f(d)] - \mathbb{E}_{d\sim D(p')}\mathbb{E}[f(d)]\| \le W_1(D(p), D(p')).$$

LEMMA EC.2. *Suppose Assumption 3 holds. Let $f: R^n \to R^d$ be an $L-$Lipschitz function, and let $X, X' \in \mathbb{R}^n$ be random variables such that $W_1(X,X') \le C$. Then*

$$\|\mathbb{E}[f(X)] - \mathbb{E}[f(X')]\|_2 \le LC. \tag{EC.5}$$

Proof of Lemma EC.2

Since

$$\begin{aligned}
\|\mathbb{E}[f(X)] - \mathbb{E}[f(X')]\|_2^2 &= (\mathbb{E}[f(X)] - \mathbb{E}[f(X')])^T (\mathbb{E}[f(X)] - \mathbb{E}[f(X')]) \\
&= \|\mathbb{E}[f(X)] - \mathbb{E}[f(X')]\|_2 \frac{(\mathbb{E}[f(X)] - \mathbb{E}[f(X')])^T}{\|\mathbb{E}[f(X)] - \mathbb{E}[f(X')]\|_2} (\mathbb{E}[f(X)] - \mathbb{E}[f(X')]),
\end{aligned}$$

we define the unit vector $e := \frac{(\mathbb{E}[f(X)] - \mathbb{E}[f(X')])^T}{\|\mathbb{E}[f(X)] - \mathbb{E}[f(X')]\|_2}$, we can get:

$$\|\mathbb{E}[f(X)] - \mathbb{E}[f(X')]\|_2^2 = \|\mathbb{E}[f(X)] - \mathbb{E}[f(X')]\|_2 (\mathbb{E}[e^T f(X)] - \mathbb{E}[f(X')]).$$

Since f is a one-dimensional $L-$lipschitz function, we can apply Lemma EC.1 and Assumption 3 to obtain that for all e,

$$\|\mathbb{E}[f(X)] - \mathbb{E}[f(X')]\|_2^2 \le \|\mathbb{E}[f(X)] - \mathbb{E}[f(X')]\|_2 LC.$$

Thus completing the proof

Proof of Theorem 1

We analyze the error of x_N^{k+1}:

$$\begin{aligned}\|x_N^{k+1}-x^*\|^2 =&\|x_N^k-\eta^k\hat{G}_N(x_N^k,z)-x^*\|^2\\ =&\|x_N^k-x^*\|^2-2\eta^k\hat{G}_N(x_N^k,z)^T(x_N^k-x^*)+(\eta^k)^2\|\hat{G}_N(x_N^k,z)\|^2.\end{aligned}$$

let $N\to+\infty$ for both sides, denote $\lim_{N\to\infty}x_N^k$ as x^k for simplicity. From Proposition 1 we have $\lim_{N\to\infty}\hat{G}_N(x,z)=\mathbb{E}_{f(y;x,z)}[\nabla_x l(x,y)]$ for any x. Therefore, for any $\zeta>0,x,z$ and vector v, $\exists N_0$, $\forall N>N_0$,

$$\|\hat{G}_N(x,z)\|\le\|\mathbb{E}_{f(y;x,z)}[\nabla_x l(x,y)]\|+\zeta,$$

and

$$\hat{G}_N(x,z)^Tv\le\mathbb{E}_{f(y;x,z)}[\nabla_x l(x,y)]^Tv+\zeta\|v\|.$$

Thus, for any $\zeta>0$, we let $\zeta_1=\frac{\zeta}{\|x_N^k-x^*\|}$ and $\zeta_2=\sqrt{\zeta/\eta^k}$, $\exists N_0$, for any fix $N>N_0$ we have:

$$\begin{aligned}\|x_N^{k+1}-x^*\|^2 =&\|x_N^k-x^*\|^2-2\eta^k\mathbb{E}_{f(y;x_N^k,z)}[\nabla_x l(x_N^k,y)]^T(x_N^k-x^*)\\ &+(\eta^k)^2\|\mathbb{E}_{f(y;x_N^k,z)}[\nabla_x l(x_N^k,y)]\|^2+3\eta^k\zeta.\end{aligned}$$

We bound the second term by convexity of the cost function

$$\begin{aligned}\mathbb{E}_{f(y;x_N^k,z)}[\nabla_x l(x_N^k,y)]^T(x_N^k-x^*) =&\mathbb{E}_{f(y;x_N^k,z)}[\nabla l(x_N^k,y)^T(x_N^k-x^*)]\\ \ge&\mathbb{E}_{f(y;x_N^k,z)}[l(x_N^k,y)-l(x^*,y)].\end{aligned}$$

For the third term, we bound by Assumption 4.

$$\|\mathbb{E}_{f(y;x_N^k,z)}[\nabla_x l(x_N^k,y)]\|^2\le L_3^c.$$

Thus

$$2\eta^k\mathbb{E}_{f(y;x_N^k,z)}[l(x_N^k,y)-l(x^*,y)]\le-\|x_N^{k+1}-x^*\|^2+\|x_N^k-x^*\|^2+(\eta^k)^2(L_3^c)^2+3\eta^k\zeta.$$

We further investigate the right side. We have

$$\begin{aligned}\mathbb{E}_{f(y;x_N^k,z)}[l(x_N^k,y)-l(x^*,y)] =&-\mathbb{E}_{f(y;x_N^k,z)}[l(x^*,y)]+\mathbb{E}_{f(y;x^*,z)}[l(x^*,y)]\\ &-\mathbb{E}_{f(y;x^*,z)}[l(x^*,y)]+\mathbb{E}_{f(y;x_N^k,z)}[l(x_N^k,y)]\\ \ge&-|\mathbb{E}_{f(y;x_N^k,z)}[l(x^*,y)]-\mathbb{E}_{f(y;x^*,z)}[l(x^*,y)]|\\ &-\mathbb{E}_{f(y;x^*,z)}[l(x^*,y)]+\mathbb{E}_{f(y;x_N^k,z)}[l(x_N^k,y)]\\ \ge&-\varepsilon L_2\|x^*-x_N^k\|-\mathbb{E}_{f(y;x^*,z)}[l(x^*,y)]+\mathbb{E}_{f(y;x_N^k,z)}[l(x_N^k,y)].\end{aligned}$$

This inequality reflects the main difficulty of our proof: to construct the gap of $g(x)=\mathbb{E}_{f(y;x,z)}[l(x,y)]$ between x^* and x_N^k. Then we substitute the inequality and have

$$\begin{aligned}2\eta^k(\mathbb{E}_{f(y;x_N^k,z)}[l(x_N^k,y)]-\mathbb{E}_{f(y;x^*,z)}[l(x^*,y)])\le&2\eta^k\varepsilon L_2\|x^*-x_N^k\|-\|x_N^{k+1}-x^*\|^2\\ &+\|x_N^k-x^*\|^2+(\eta^kL_3^c)^2+3\eta^k\zeta.\end{aligned}$$

Take summation from $r=0$ to k and take the minimum of the left side, we obtain

$$(2\sum_{r=0}^{k}\eta^r)\min_{0\le r\le k}\{\mathbb{E}_{f(y;x^r,z)}[l(x^r,y)]-\mathbb{E}_{f(y;x^*,z)}[l(x^*,y)]\} \le 2\varepsilon L_2\sum_{r=0}^{k}\eta^r\|x^*-x^k\| + \|x^0-x^k\|^2+(L_3^c)^2\sum_{r=0}^{k}(\eta^r)^2+3\sum_{r=0}^{k}\eta^r\zeta.$$

Hence we complete the proof by dividing $2\sum_{r=0}^{k}\eta^r$ on both sides.

Proof of Proposition 3

We investigate the distance between x_N^k and a stable point x_{PS}.

$$\begin{aligned}\|x_N^{k+1}-x_{PS}\|^2 &= \|x_N^k-\eta\hat{G}_N(x_N^k;z)-x_{PS}\|^2\\ &= \|x_N^k-x_{PS}\|^2-2\eta\hat{G}_N(x_N^k)^T(x_N^k-x_{PS})+(\eta^2)\|\hat{G}_N(x_N^k)\|^2.\end{aligned}$$

We begin by upper bounding the second term. From Proposition 1, we know that for any $\xi>0$, there exists a sample size N_0 such that $\sup_x\|\hat{G}_N(x)-\mathbb{E}_{f(y;x,z)}[\nabla_x l(x,y)]\|\le\xi$ for all $N>N_0$. Thus we have

$$\hat{G}_N(x_N^k)^T(x_N^k-x_{PS}) \ge \mathbb{E}_{f(y;x_N^k,z)}[\nabla_x l(x_N^k,y)]^T(x_N^k-x_{PS})-\xi\|x_N^k-x_{PS}\|.$$

We can further bound the second term using the same approach as the proof of proposition 2.3 in Mendler-Dünner et al. (2020b)'s work. They give that

$$\mathbb{E}_{f(y;x_N^k,z)}[\nabla_x l(x_N^k,y)]^T(x_N^k-x_{PS})\ge B\|x_N^k-x_{PS}\|^2.$$

We then bound the third term:

$$\|\hat{G}_N(x_N^k)\|^2\le\xi^2+\|\mathbb{E}_{f(y;x_N^k,z)}[\nabla_x l(x_N^k,y)]\|^2.$$

We can also adopt the same approach in the proof of proposition 2.3 in Mendler-Dünner et al. (2020b). They give that under Assumptions 4 and 3,

$$\|\mathbb{E}_{f(y;x_N^k,z)}[\nabla_x l(x_N^k,y)]\|^2\le 2B^2\|x_N^k-x_{PS}\|^2.$$

Therefore, we obtain

$$\|x_N^{k+1}-x_{PS}\|^2\le(1-2\eta A+2\eta^2B^2)\|x_N^k-x_{PS}\|^2+2\eta\xi\|x_N^k-x_{PS}\|+\xi^2\eta^2. \tag{EC.6}$$

In case 1, to give a reasonable distance bound, we need to choose η such that the right-hand side is a perfect quadratic polynomial. Thus we choose η such that

$$4B^2\eta^2-2A\eta+1=0.$$

Note that from the Viete's theorem, the two solutions are both positive since we assume $A > 0$. Thus we only need to ensure that the equation have real solution, that is

$$4A^2 \geq 16B^2, \quad A \geq 2B.$$

And we take the square root two both sides of equation (EC.6)

$$\|x_N^{k+1} - x_{PS}\| \leq \sqrt{1 - 2\eta A + 2\eta^2 B^2}\|x_N^k - x_{PS}\| + \xi\eta.$$

We denote $C = \sqrt{1 - 2\eta A + 2\eta^2 B}$ and divide both sides by C^{k+1}

$$\frac{\|x_N^{k+1} - x_{PS}\|}{C^{k+1}} \leq \frac{\|x_N^k - x_{PS}\|}{C^k} + \frac{\xi\eta}{C^{k+1}}.$$

Take the summation on both sides from 0 to $k+1$ and we obtain

$$\|x_N^{k+1} - x_{PS}\| \leq C^{k+1}\|x_N^1 - x^*\| + \xi\eta\frac{1 - C^{k+1}}{1 - C}.$$

Note that $\eta A - \eta^2 B^2 = \frac{2\eta A + 1}{4} > 0$, thus $C < 1$ and the distance is decreasing.

Now we focus on case 2. Since the quadratic term on the right-hand side of (EC.6) is less than zero, we obtain

$$\|x_N^{k+1} - x_{PS}\|^2 \leq 2\eta\xi\|x_N^k - x_{PS}\| + \xi^2\eta^2. \tag{EC.7}$$

Thus

$$\|x_N^{k+1} - x_{PS}\|^2 - \|x_N^k - x_{PS}\|^2 \leq -\|x_N^k - x_{PS}\|^2 + 2\eta\xi\|x_N^k - x_{PS}\| + \xi^2\eta^2.$$

If $\|x_N^k - x_{PS}\| \geq (1 + \sqrt{2})\xi\eta$, we can derive that $\|x_N^{k+1} - x_{PS}\|^2 - \|x_N^k - x_{PS}\|^2 \leq 0$, which indicates that although the distance may exceed the bound $(1 + \sqrt{2})\xi\eta$ some time, it will decrease immediately until it reach the bound, hence complete the proof.

Proof of Theorem 2

Since $l(x, y)$ is strongly convex in x and L_2-Lipschitz continuous in y, the proof is then complete by imposing the triangular inequality to Lemma 1 and Proposition 3.

Proof of Proposition 4

The proof is divided into two steps. In the first step, we prove that the objective function $\mathbb{E}_{f(y;x,z)}[l(x, y)]$ has Lipschitz gradient in x. Then we prove that under diminishing step, any converging subsequence converge to the stationary point.

We denote $g(x) = \mathbb{E}_{f(y;x,z)}[l(x, y)]$, then for any $x_1, x_2 \in \mathcal{X}$

$$\|\nabla_x g(x_1) - \nabla_x g(x_2)\| = \|\nabla_x \mathbb{E}_{f(y;x_1,z)}[l(x_1, y)] - \nabla_x \mathbb{E}_{f(y;x_2,z)}[l(x_2, y)]\|.$$

According to Assumption 2, we can write the expectation to integrate form and change the integrate operator and derivative operator.

$$\begin{aligned}\|\nabla_x g(x_1)-\nabla_x g(x_2)\| =& \|\int_{y\in\Omega}\nabla_x(l(x_1,y)f(y;x_1,z))dy-\int_{y\in\Omega}\nabla_x(l(x_2,y)f(y;x_2,z))dy\|\\ \le& \|\int_y l(x_1,y)(\nabla_x f(y;x_1,z))-l(x_2,y)(\nabla_x f(y;x_2,z))dy\|\\ &+\|\int_y(\nabla_x l(x_1,y))f(y;x_1,z)-(\nabla_x l(x_2,y))f(y;x_2,z)dy\|\\ =& I+II.\end{aligned}$$

The second inequality follows by the multiplication rule of derivative. We then analyze I and II respectively.

$$\begin{aligned}I\le& \|\int_y l(x_1,y)\nabla_x f(y;x_1,z)dy-\int_y l(x_1,y)\nabla_x f(y;x_2,z)dy\|\\ &+\|\int_y l(x_1,y)\nabla_x f(y;x_2,z)dy-\int_y l(x_2,y)\nabla_x f(y;x_2,z)dy\|\\ \le& \int_y |l(x_1,y)|\|\nabla_x f(y;x_1,z)-\nabla_x f(y;x_2,z)\|dy\\ &+\int_y |l(x_1,y)-l(x_2,y)|\|\nabla_x f(y;x_2,z)\|dy\\ \le& S_\Omega L_4 L_{3g}\|x_1-x_2\|+S_\Omega L_5 L_1\|x_1-x_2\|.\end{aligned}$$

The first inequality holds from the triangular inequality. The second inequality holds by the Cauchy-Schwarz inequality. The third inequality holds by the Lipschitz continuous characteristic and intermediate value theorem, where S_Ω denotes of the measurement of the set Ω.

We can also bound the second term by the following steps:

$$\begin{aligned}II\le& \|\int_y \nabla_x l(x_1,y)f(y;x_1,z)dy-\int_y\nabla_x l(x_1,y)f(y;x_2,z)dy\|\\ &+\|\int_y\nabla_x l(x_1,y)f(y;x_2,z)dy-\int_y\nabla_x l(x_2,y)f(y;x_2,z)dy\|\\ =&\|\mathbb{E}_{f(y;x_1,z)}[\nabla_x l(x_1,y)]-\mathbb{E}_{f(y;x_2,z)}\nabla_x l(x_1,y)\|+\|\mathbb{E}_{f(y;x_2,z)}[\nabla_x l(x_1,y)-\nabla_x l(x_1,y)]\|\\ \le&\varepsilon L_{2g}\|x_1-x_2\|+L_{2g}\|x_1-x_2\|.\end{aligned}$$

The first inequality holds by the triangular inequality. The first equality holds by the definition of expectation. The second inequality holds by Lemma EC.2 and the definition of Lipschitz gradient.

Thus, $\|\nabla_x g(x_1)-\nabla_x g(x_2)\|\le[(\varepsilon+1)L_{2g}+S_\Omega(L_1L_5+L_4L_{3g})]\|x_1-x_2\|$. Hence the objective function has Lipschitz gradient and $L=(\varepsilon+1)L_{2g}+S_\Omega(L_1L_5+L_4L_{3g})$.

recall that the update rule is given by

$$x_N^{r+1}=x_N^r+\eta^r\hat{G}_N(x;z).$$

From descent lemma, we have

$$g(x_N^{r+1})\le g(x_N^r)+\eta^r\hat{G}_N(x_N^r;z)^T\nabla g(x_N^r)+\frac{L(\eta^r)^2}{2}\|\hat{G}_N(x_N^r;z)\|^2.$$

Taking $N \to \infty$ on both sides, since $g(x)$ is continuous and $\lim_{N\to\infty} \hat{G}_N(x;z) = \mathbb{E}_{f(y;x,z)}[\nabla_x l(x,y)]$ from Proposition 1.

$$g(x^{r+1}) - g(x^r) \leq \eta^r \mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)]^T \nabla g(x^r) + \frac{L(\eta^r)^2}{2} \|\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)]\|^2.$$

where $x^r = \lim_{N\to\infty} x_N^r$.

Since

$$\begin{aligned}\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)]^T g(x^r) =& \|\nabla_x \mathbb{E}_{f(y;x^r,z)}[l(x^r,y)]\|^2 + \|\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)]\|^2 \\ &- \|\int_y l(x^r,y)\nabla_x f(y;x^r,z)dy\|^2 \\ \geq& (\|\nabla_x \mathbb{E}_{f(y;x^r,z)}[l(x^r,y)]\|^2 - L_4^2 L_5^2 S_\Omega^2) + \|\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)]\|^2.\end{aligned}$$

Note that since the range of y is limited, we can scale the random parameters y so that $S_\Omega^2 \leq \frac{\|\nabla_x \mathbb{E}_{f(y;x^r,z)}[l(x^r,y)]\|}{L_4 L_5}$. Thus,

$$\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)]^T g(x^r) \geq \|\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)]\|^2.$$

Therefore,

$$g(x^{r+1}) - g(x^r) \leq -\eta^r(1 - \frac{L\eta^r}{2})\|\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)]\|^2.$$

Since η^r is diminishing, for any $\xi \in (0,1)$, there exists $\bar{r}$ such that for any $r \geq \bar{r}$, we have

$$g(x^{r+1}) - g(x^r) \leq -\eta^r \xi \|\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)]\|^2.$$

Since $\lim_{r\in\mathcal{K}\to\infty} x^r = \bar{x}$ and $g(x)$ is continuous, we have $\lim_{r\to\infty} g(x^r) = g(\bar{x})$. Taking summation on both sides from $r = \bar{r}$ to ∞, we can obtain that

$$\sum_{r=\bar{r}}^{\infty} \eta^r \xi \|\mathbb{E}_{f(D;x^r,z)}[\nabla_x l(x^r,D)]\|^2 \leq g(x^{\bar{r}}) - \lim_{r\to\infty} g(x^r).$$

Since $\sum_{r=\bar{r}}^{\infty} \alpha^r = +\infty$, we have $\lim_{r\in\mathcal{K}\to\infty} \|\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)]\|^2 = 0$, hence $\mathbb{E}_{f(y;\bar{x},z)}[\nabla_x l(\bar{x},y)] = 0$ and the proof is completed.

Proof of Proposition 5

To simplify the denotation, we omit the limitation of $N \to \infty$. Therefore the descent direction is $d^r = -\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)]$, where $x^r = \lim_{N\to\infty} x_N^r$. According to the armijo principle:

$$g(x^r) - g(x^{r+1}) \geq -\eta^r \sigma \|\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)]\|^T d^r.$$

Since $\lim_{r(\in\mathcal{K})\to\infty} \sup_r \|\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)]\| \geq 0$. The sequence $\mathbb{E}_{f(y;x^r,z)}[l(x^r,y)]$ decreases monotonically and have a lower bound. Thus

$$\lim_{r(\in\mathcal{K})\to\infty} g(x^r) - g(x^{r+1}) = 0,$$

which is followed by

$$\lim_{r(\in\mathcal{K})\to\infty}\eta^r=0.$$

Hence, by the definition of the armijo rule, we must have for some index $\bar{r}\geq 0$

$$g(x^r)-g(x^r+\frac{\eta^r}{\beta}d^r)<-\sigma\frac{\eta^r}{\beta}\|\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)]\|^T d^r,\forall r\in\mathcal{K},r\geq\bar{r}.$$

We denote

$$p^r:=\frac{d^r}{\|d^r\|},\quad \bar{\eta}^r:=\frac{\eta^r\|d^r\|}{\beta}.$$

Since $\|p^r\|=1$, there exists a subsequence $\{p^r\}_{\bar{\mathcal{K}}}$ of $\{p^r\}_{\mathcal{K}}$ such that $\{p^r\}_{\bar{\mathcal{K}}}\to\bar{p}$, where $\bar{p}$ is a unit vector. Then

$$\frac{g(x^r)-g(x^{r+1})}{\bar{\eta}^r}<-\sigma(\mathbb{E}_{f(x^r;y,z)}[\nabla_x l(y,x^r)])^T p^r.$$

Hence,

$$\frac{g(x^r)-\mathbb{E}_{f(y;x^r,z)}[l(x^{r+1},y)]+\mathbb{E}_{f(y;x^r,z)}[l(x^{r+1},y)]-g(x^{r+1})}{\bar{\eta}^r}<-\sigma(\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)])^T p^r. \tag{EC.8}$$

By Lemma EC.2, $g(x^r)-g(x^{r+1})\geq-\varepsilon L_1\|\bar{\eta}^r p^r\|$.

By using the mean value theorem,

$$\begin{aligned}\frac{-\varepsilon L_1\|\bar{\eta}^r p^r\|}{\bar{\eta}^r}&+\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r+\tilde{\alpha}^r p^r,y)]^T p^r\\&<-\sigma(\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)])^T p^r.\end{aligned} \tag{EC.9}$$

Let $r(\in\bar{\mathcal{K}})\to\infty$,

$$-\varepsilon L_1-(\mathbb{E}_{f(y;\bar{x},z)}[\nabla_x l(\bar{x},y)])^T\bar{p}<-\sigma\mathbb{E}_{f(y;\bar{x},z)}[\nabla_x l(\bar{x},y)]^T\bar{p}.$$

Substituting $d^r=-\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)]$, we have

$$-\varepsilon L_1<-(1-\sigma)\|\mathbb{E}_{f(y;\bar{x},z)}[\nabla_x l(\bar{x},y)]\|.$$

which completes the proof.

Proof of Proposition 6

For any given sequence $\{x_N^r\}_{r=1}^k$, denote $g_r(x)=\mathbb{E}_{f(y;x^r,z)}[l(x,y)]$. Then $\nabla_x g_r(x)=\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x,y)]$.

Since $l(x,y)$ has $L_{1g}-$Lipschitz gradient, by the descent lemma, when $N\to\infty$,

$$\begin{aligned}g_r(x_N^{r+1})-g_r(x_N^r)=&g_r(x_N^r-\eta\mathbb{E}_{f(y;x_N^r,z)}[\nabla_x l(x_N^r,y)])-g_r(x_N^r)\\\leq&-(1-\frac{L_{1g}\eta}{2})\eta\|\mathbb{E}_{f(y;x_N^r,z)}[\nabla_x l(x_N^r,y)]\|^2.\end{aligned}$$

Thus,

$$\sum_{r=0}^{k}\|\mathbb{E}_{f(y;x_N^r,z)}[\nabla_x l(x_N^r,y)]\|^2\leq\frac{2}{\eta}\sum_{r=0}^{k}\left(g_r(x_N^{r+1})-g_r(x_N^r)\right). \tag{EC.10}$$

And,

$$\begin{aligned}\sum_{r=0}^{k} g_r(x_N^{r+1}) - g_r(x_N^r) =& \sum_{r=0}^{k} [g_{r+1}(x_N^{r+1}) - g_r(x_N^r)] + [g_r(x_N^{r+1}) - g_{r+1}(x_N^{r+1})] \\ \leq& g_0(x_N^0) - g_*(x^*) + \sum_{r=0}^{k} g_r(x_N^{r+1}) - g_{r+1}(x_N^{r+1}) \\ \leq& g_0(x_N^0) - g_*(x^*) + L_1\varepsilon \sum_{r=1}^{k} \|x_N^r - x_N^{r-1}\| \\ =& g_0(x_N^0) - g_*(x^*) + L_1\varepsilon \sum_{r=1}^{k} \|\eta \mathbb{E}_{f(y;x_N^r,z)}[\nabla_x l(x_N^r, y)]\| \\ \leq& g_0(x_N^0) - g_*(x^*) + L_1\varepsilon \sum_{r=1}^{k} \eta L_3^c,\end{aligned}$$

where the first inequality holds because $g_*(x^*) \leq g_r(x_N^r)$ for any x^r, the second inequality holds by Lemma EC.2 and Assumption 3, and the third inequality holds by Assumption 4(c).

Then the proof completes by taking the union inequality on the left side of (EC.10) and dividing both sides by $k+1$.

Proof of Theorem 3

We proof the theorem by contradiction. Suppose x^* maximize $\max_x g(x) = \mathbb{E}_{f(y;x,z)}[l(x,y)]$, and $\|\mathbb{E}_{f(y;x^*,z)}[\nabla_x l(x^*,y)]\| > L_1\varepsilon$. Then for any $x_1 \in X$,

$$\begin{aligned}g(x_1) - g(x^*) =& \left(\mathbb{E}_{f(y;x_1,z)}[l(x_1,y)] - \mathbb{E}_{f(y;x,z)}[l(x,y)]\right) \\ =& \left(\mathbb{E}_{f(y;x_1,z)}[l(x_1,y)] - \mathbb{E}_{f(y;x^*,z)}[l(x_1,y)]\right) \\ &+ \left(\mathbb{E}_{f(y;x^*,z)}[l(x_1,y)] - \mathbb{E}_{f(y;x,z)}[l(x,y)]\right).\end{aligned}$$

From Lemma EC.2, we have

$$|\mathbb{E}_{f(y;x_1,z)}[l(x_1,y)] - \mathbb{E}_{f(y;x^*,z)}[l(x_1,y)]| \leq L_1\varepsilon\|x_1 - x^*\|.$$

For the second term, we expand $l(x_1,y)$ at x^* and obtain

$$\mathbb{E}_{f(y;x^*,z)}[l(x_1,y)] - \mathbb{E}_{f(y;x,z)}[l(x,y)] = \mathbb{E}_{f(y;x^*,z)}[\nabla_x l(x^*,y)]^T(x_1 - x^*) + o(\|x_1 - x^*\|),$$

where $o(\|x_1 - x^*\|)$ denotes the first-order infinitesimals to $\|x_1 - x^*\|$. By substituting the two terms above and divide both sides by $\|x_1 - x^*\|$, we obtain

$$\frac{g(x_1) - g(x^*)}{\|x_1 - x^*\|} \geq -L_1\varepsilon + \mathbb{E}_{f(y;x^*,z)}[\nabla_x l(x^*,y)]^T \frac{(x_1 - x^*)}{\|x_1 - x^*\|} + \frac{o(\|x_1 - x^*\|)}{\|x_1 - x^*\|}.$$

We let $x_1 - x^*$ take the same direction of $\mathbb{E}_{f(y;x^*,z)}[\nabla_x l(x^*,y)]$, hence the second term on the right side becomes $\|\mathbb{E}_{f(y;x^*,z)}[\nabla_x l(x^*,y)]\|$. Therefore, for any $\xi > 0$, there exists x_1 that is sufficiently close to x^* such that

$$\frac{g(x_1) - g(x^*)}{\|x_1 - x^*\|} \geq -L_1\varepsilon + \|\mathbb{E}_{f(y;x^*,z)}[\nabla_x l(x^*,y)]\| - \xi.$$

Since $\|\mathbb{E}_{f(y;x^*,z)}[\nabla_x l(x^*,y)]\| > L_1\varepsilon$ and ξ can be sufficiently small, we have $g(x_1) - g(x^*) > 0$, which contradicts with the condition that $g(x^*)$ is the optimal solution.

EC.4 Experiment Supplements

EC.4.1 Description of data

Table EC.1 Description of real electricity pricing data

Variable	Type	Description	Statistics
Date	datetime	the date of the recording	min 1Jan15, max 6Oct20
Demand	float	a total daily electricity demand in MWh	min 85.1k, median 120k, max 171k
RRP	float	a recommended retail price in AUD$/MWh	min 0, median 66.7, max 300
min temperature	float	minimum temperature during the day in Celsius	min 0.6, median 11.3, max 28
max temperature	float	maximum temperature during the day in Celsius	min 9, median 19.1, max 43.5
solar exposure	float	total daily sunlight energy in MJ/m$\hat{2}$	min 0.7, median 12.7, max 33.3
rainfall	float	daily rainfall in mm	min 0, median 0, max 54.6
school day	boolean	if students were at school on that day	True 69%, False 31%
holiday	boolean	if the day was a state or national holiday	True 4%, False 96%

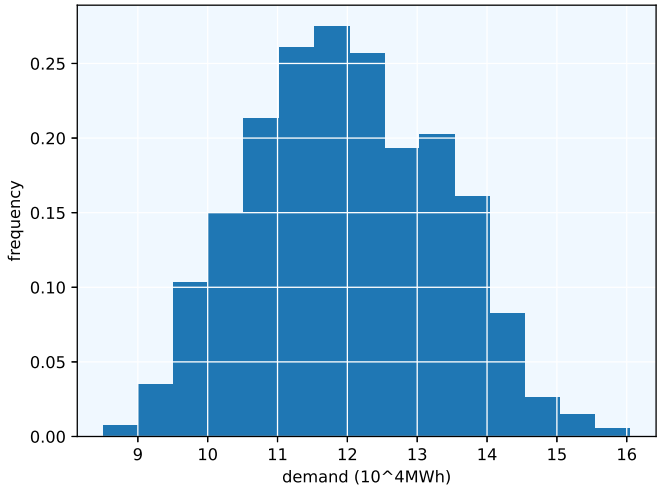
Real data The real dataset comes from a real-world power plant pricing scenario (https://www.kaggle.com/datasets/aramacus/electricity-demand-in-victoria-australia). This dataset describes the electricity demand and price situation in Victoria, Australia from 2015 to 2020. The distribution of demand can be seen in Figure EC.1. The descriptive information of the real data is shown in Table EC.1. The factors that influence daily demand are *price, temperature, solar exposure, school day* and *holiday*. Note that we perform an artificial transformation on the temperature. We define heating degree day (HDD) as $HDD = (T_{min} - 18)^+$, and cooling heating degree day (CDD) as $CDD = (15 - T_{max})^+$, where T_{max} and T_{min} are the highest and lowest centigrade temperatures in one day. This transformation can better reflect the relationship between temperature and electricity demand. The demand is sensitive to price, but it also depend on other features such as temperature and holiday. In our work, we consider the temperature, solar, rainfall, school_day and holiday factors. Note that the scales of features are different, so we standardize the feature to $[0,1]$ when processing the data. We use Euclidean metric to measure the distance between samples.

Simulation data In terms of simulation data, we generate the demand by the following model.

$$D = \max\{0, 60 - p + 12a^T(X + 0.25\phi) + 5b^T X\theta\}, \tag{EC.11}$$

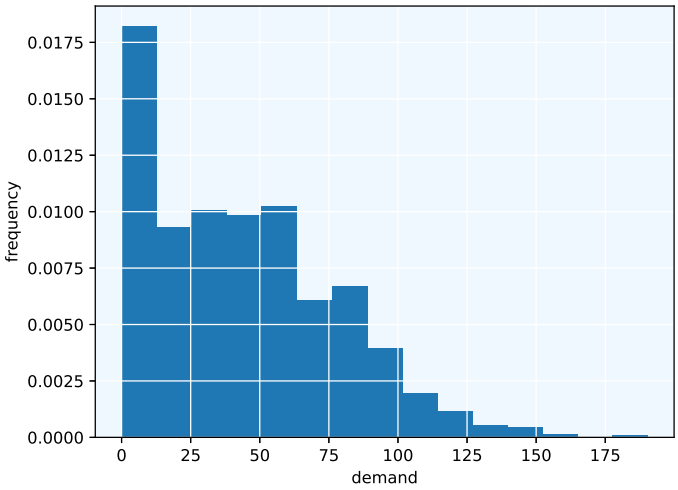
where $\phi \sim N(0, I_4)$ is a 4-dimensional vector, $\theta \sim N(0,1)$ is also a Guassian parameter. Both θ and ϕ are the stochastic factors that cause demand fluctuation. The constant vector $a = (0,8,1,1,1)^T$ and $b = (-1,1,0,0)^T$. Note that we refer the demand model to Lin et al. (2022). The demand distribution under

Figure EC.1 Demand distribution for real data



$p = 20$ is shown in Figure EC.2. We observe that the distribution is skew and long tail, thus hard to predict by simple models such as linear regression.

Figure EC.2 Demand distribution for simulated data



The second demand model is simply a linear regression model to suit the PTO assumption, where $D = 60 - p + (1,1,1,1)^T z + \phi$ and $\phi \sim N(0,1)$. Thus the demand follows a normal distribution under any price and feature.

EC.4.2 Step size comparison

The step size of CGD algorithm adopted in the numerical experiment part is the Armijo step size with $\sigma = 0$. In Section 3.2.2, we have analyzed the difference on convergence between the diminishing step size and Armijo step size. In this section, we will evaluate the difference by experiment.

We first compare two kinds of step size in the simulated dataset. We set the step size constant $C = 0.05$ in diminishing step size approach. The realized profit and optimality gap are shown in Table EC.2. We can find that the diminishing step size performs worse than Armijo step size in this case. We believe the reason is that the assumptions for the convergence under diminishing step size are usually too strong. The value range Ω in Assumption 1 and L_4, L_5 constant in Proposition 4 maybe large in practice, causing a bad convergence performance. Moreover, we find that although any convergent subsequence converge to the local maximum according to Proposition 4, the diminishing step size cannot stop at the local maximum automatically, which indicates that the diminishing step size may not lead to any convergence subsequence. Therefore, the diminishing step size need a careful selection on the step size constant C and stop criteria.

Table EC.2 Realized profit of Armijo step size and Diminishing step size

Method	kNN	kernel	CART	RF
Armijo	674.37	698.20	689.94	584.07
Diminishing	524.05	512.858	560.0039	388.8681

We also evaluate the effect of hyperparameter σ on CGD algorithm. Table EC.3 reports the optimality gap and iteration number for different constant σ under kernel regression . Figure EC.3 plots the supplementary result in terms of σ and optimality gap. We observe the performance is stable when $\alpha_0 \in (0.01, 0.1)$, and when $\sigma \leq 0.2$. The increment of both α_0 and σ can reduce the iteration numbers, thus accelerate the solution. But when α_0 is larger than 0.5, the optimality gap may become larger. Larger σ can block the update of solution and may cause the algorithm to stop before reaching the convergence.

Table EC.3 Performance comparison among different initial step sizes and σ of Armijo step size

(α_0, σ)	profit	optimality gap	iterations	(α_0, σ)	profit	optimality gap	iterations
(0.01, 0)	695.89	0.97%	90	(0.5, 0)	672.02	4.37%	2
(0.01, 0.1)	688.28	2.05%	74	(0.5, 0.1)	691.25	1.63%	2
(0.01, 0.2)	659.48	6.15%	62	(0.5, 0.2)	667.36	5.03%	2
(0.01, 0.5)	475.37	32.35%	28	(0.5, 0.5)	562.69	19.92%	1
(0.01, 0.9)	300.00	57.31%	0	(0.5, 0.9)	300.00	57.31%	0
(0.05, 0)	698.20	0.64%	18	(1, 0)	653.18	7.05%	1
(0.05, 0.1)	688.84	1.97%	15	(1, 0.1)	653.18	7.05%	1
(0.05, 0.2)	662.67	5.70%	13	(1, 0.2)	653.18	7.05%	1
(0.05, 0.5)	478.73	31.87%	6	(1, 0.5)	569.02	19.02%	1
(0.05, 0.9)	300.00	57.31%	0	(1, 0.9)	300.00	57.31%	0
(0.1, 0)	696.72	0.85%	9				
(0.1, 0.1)	689.42	1.89%	8				
(0.1, 0.2)	659.40	6.16%	7				
(0.1, 0.5)	491.16	30.10%	3				
(0.1, 0.9)	300.00	57.31%	0				

Figure EC.3 Performance comparison among different initial step sizes and σ of Armijo step size