Online Planning with Offline Simulation

Wang Chi Cheung
Department of Industrial Systems Engineering and Management, National University of Singapore
Guodong Lyu, Chung-Piaw Teo
NUS Business School & Institute of Operations Research and Analytics, National University of Singapore
Hai Wang
School of Information Systems, Singapore Management University

One of the central issues in (finite horizon) online planning problems is to synthesize the impact of real time decisions on the subsequent states of the system, and the performance in the remaining time horizon (cost-to-go function). A complete resolution often leads to intractable dynamic programming problems. In this paper, we propose a computationally efficient approach to this problem that attains near-optimal performance in non-stationary environments. More specifically, we study a general class of online planning problems with concave objective functions and (global) feasibility constraints. A wide range of problems in supply chain management, online advertising, and network revenue management etc., can be appropriately modelled using this online planning framework. Leveraging on the value of the "gradient" information obtained from offline simulation (generated from the distributional information), we develop a generic approach to facilitate online planning for this class of problems. Furthermore, our proposed approach produces near optimal solution with sublinear regret and satisfies the feasibility constraints with high probability. We present extensive numerical evidence to validate the performance of this approach, and discuss its improvement over existing techniques that assume the underlying environment is stationary.

Key words: Online Planning; Non-Stationary Environment; Distributional Information; Offline Simulation

1. Introduction

In recent years, we have witnessed an explosion of interest in the use of online optimization techniques on various operational problems (e.g., Agrawal et al. 2014, Asadpour et al. 2020). Under the prototypical framework of online optimization, the decision maker interacts with the dynamic environment and makes resource allocation decisions without visibility of the evolution of future scenarios, i.e., the decision making scenarios unfold sequentially and the online actions adapt a strategy to the dynamic environment purely based on historical information. Its performance is often benchmarked against an offline optimal solution, computed with perfect hindsight knowledge of the scenarios. Although future information is not integrated in the design of online algorithms, this paradigm can achieve near-optimal performance guarantee, vis-a-viz the offline optimal solution, in many stochastic convex

programming problems. This happens for many classes of operations problems when the environments are stationary (all scenarios are i.i.d.) or under the random permutation models (Shalev-Shwartz et al. 2009, Agrawal and Devanur 2015). For example, Agrawal et al. (2014) demonstrated that a dual-price based algorithm, which is calibrated based on partially revealed objective coefficients and constraint matrix information, could achieve near-optimal performance on online linear programming problems with random permutation inputs. Zhong et al. (2018) developed a debt-based allocation policy (via gradient descent on the realized cost functions with stationary inputs) to facilitate online resource allocation and to serve heterogeneous customer segments with differentiated fill rate targets.

The operating environments in many practical operations management problems are however often non-stationary in nature. In these cases, traditional online algorithms may fail to provide satisfactory performance guarantee, since the dual-price or gradient information[1] extracted from historical observations might not provide useful information to guide actions and prepare for the future. To address this problem, a recent stream of literature used the concept of *variation budget* to account for the change in dynamic environments/cost functions (Besbes et al. 2015, Chen et al. 2019). Assuming a bounded variational budget, Besbes et al. (2015) demonstrated that near-optimal performance can be achieved by a class of online policy, giving higher weightage to historical observations closer to the decision epochs, while the "outdated" information is abandoned.

However, in the field of operations management, the general preference is to use the information about the future, i.e., distributional information gleaned from predictive analytics and forecasting tools, to adjust the planning strategy in a dynamic fashion, to take into account the "cost-to-go functions" in performance evaluation (Alaei et al. 2012, Esfandiari et al. 2015, Ma et al. 2020). This has become more common in recent time as firms invest in sophisticated digital technologies to obtain better forecasts to facilitate planning. For instance, Uber forecasts the future demand scenarios for real-time driver repositioning in order to provide better ride-sharing service.[2] In another instance, Ma et al. (2020) constructed an approximate solution to solve network revenue management problems leveraging on the distributional information (i.e., customer arrival rates in their setting).

Indeed, it is not surprising that more information about the future could potentially enhance the capability of online algorithms to handle the non-stationary random inputs. The central question here is: **how could we exploit future information in the design of online planning strategy**? To this end, we utilize a sequence of offline samples obtained from simulation to incorporate the distributional information of the future into online planning. In short, we exploits knowledge of the probability distributions of the stochastic inputs to design an Offline-to-Online approach with provable performance guarantee.

[1] An illustrative resource allocation example under a non-stationary environment is provided in Section 5.1.

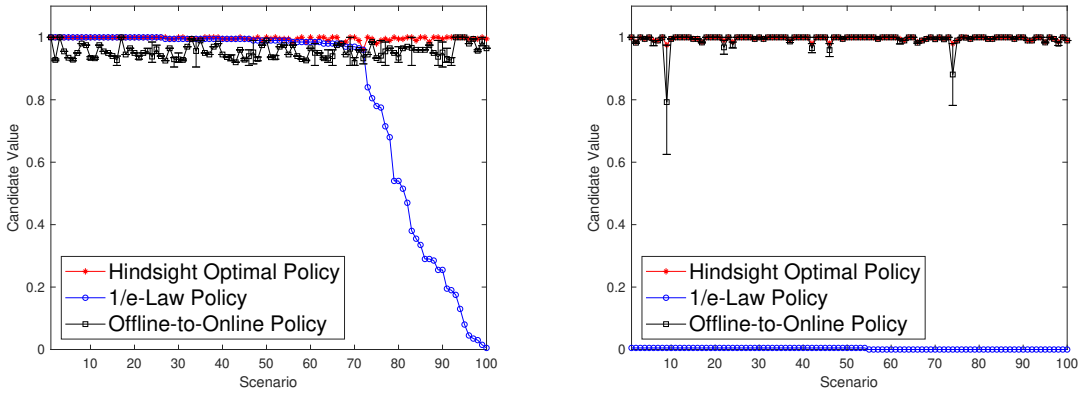
[2] Forecasting at Uber: An Introduction. Retrieved from `https://eng.uber.com/forecasting-introduction/`

There is a growing stream of literature in the simulation community on the use of offline simulation to facilitate online planning. Here, simulation is used as data generator, so that state-of-the-art analytics tools can be used to build predictive models for real time applications. Hong and Jiang (2019) provided an overview of this framework (called simulation analytics) and discussed applications in estimation, ranking and selection, and simulation optimization problems. This framework has been successfully used to predict portfolio risk in real time (cf. Jiang et al. 2020). Our approach goes one step further, and exploits the gradient information obtained from the simulations to provide near optimal performance guarantee.

To better understand the value of offline simulation, we use the secretary problem for illustration.

EXAMPLE 1 (SECRETARY PROBLEM). Consider $\Gamma = 200$ candidates to be interviewed in a sequential manner. Each candidate is associated with a score, and the decision maker aims to recruit the one with the highest score. The score of each candidate is not known a priori, and the score is only revealed after the candidate has been interviewed. For each candidate, the decision to recruit or not has to be made on the spot after the interview, and the decision is irreversible.

Figure 1 Comparison of different policies for the secretary problem.



(a) Stationary Case (b) Non-Stationary Case

Note. [1] In the stationary case, the candidate values are i.i.d. generated from the set $\{\frac{1}{\Gamma}, \frac{2}{\Gamma}, \ldots, \frac{\Gamma}{\Gamma}\}$ with replacement. In the non-stationary case, the value of γ^{th} candidate is sampled from $\{0, \frac{\Gamma-\gamma+1}{\Gamma}\}$, each with probability 0.5.
[2] Under the offline-to-online scheme, we randomly generate the "bid-price" for 1000 times in each scenario and plot the average performance. The error bar captures the 25^{th} to 75^{th} quantile.
[3] With a slight modification, if no one is selected, the last candidate would be chosen under all policies. For ease of exposition, we arrange the scenarios in the descending order of the performance of the $(1/e)$-law policy.

For this problem it is well known that the $(1/e)$-Law, which selects the first candidate whose score is larger than all the first Γ/e ones, guarantees the best candidate with probability at least $1/e$ if the candidate values are i.i.d. generated (Bruss 1984). However, as shown in Figure 1(a), the $(1/e)$-Law suffers in the scenarios when the best candidate comes from the last $\Gamma(1-1/e)$ group. Figure 1(b) shows that this policy, without using the knowledge of the distribution of the scores, performs poorly

in the non-stationary case, when the score of γ^{th} candidate is sampled from $\{0, \frac{\Gamma-\gamma+1}{\Gamma}\}$, each with probability 0.5. The scores of the candidates are therefore non-stationary.

The hindsight optimal policy, which knows all the candidates' scores in advance, always finds the best candidate. The $(1/e)$-Law almost surely finds an inferior candidate in all simulations in this environment. Interestingly, if the score distribution of each candidate is given, we can exploit this information to generate better policy. Using our offline-to-online approach, we can guarantee near-optimal performance as detailed in Section 3. Here, a candidate is selected if his/her score exceeds a certain randomized "bid-price" (obtained from offline simulation). This policy performs almost as well as the hindsight policy, in both stationary and non-stationary environments! ■

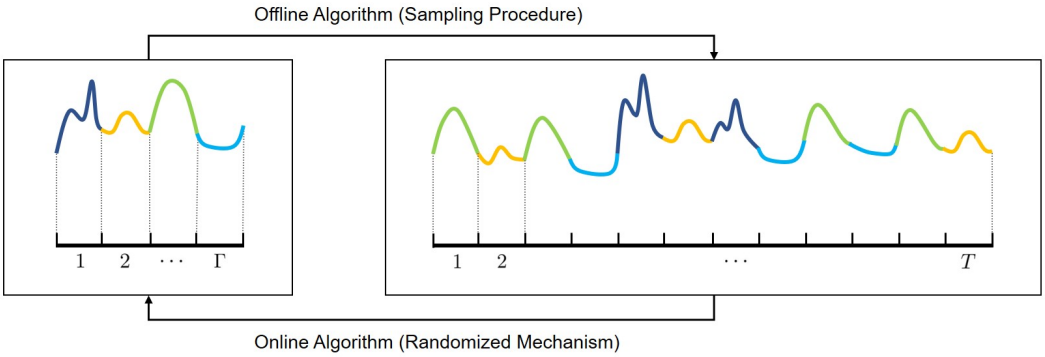
More generally, we explore the value of offline simulation in online planning under the framework of *online stochastic optimization* problem with general concave objective functions, convex global feasibility constraints, and non-stationary stochastic inputs. A wide range of operations management applications belongs to this framework, including online packing (Agrawal et al. 2014), online advertising (Agrawal and Devanur 2015, Esfandiari et al. 2015), online assortment (Agrawal et al. 2016, Li et al. 2020), network revenue management (Bumpensanti and Wang 2020, Ma et al. 2020), and supply chain management (Lyu et al. 2019, Asadpour et al. 2020, Xu et al. 2020). Our approach is also conceptually simple – we sample a finite number of scenarios randomly, based on the distributional information known, and solve a related deterministic "multi-sample" optimization problem (cf. Figure 2) using an online algorithm to obtain "gradient" information on the trade-offs between objective function and global constraints. This will be used to guide the online allocation decision for our original problem. In this way, we utilize the same distributional information as the conventional approach to stochastic dynamic programming, but our online algorithm is able to produce near optimal solutions to these large-scale stochastic resource allocation problems efficiently, beating the curse of dimensionality that plagues many dynamic programming algorithms.

The **main contributions** of this work are summarized as follows.

- We propose a class of randomized algorithms to facilitate online planning in non-stationary stochastic environments. The algorithm consists of an offline phase, which involves sampling a number of scenarios, as well as an online phase, which involves deploying a randomly chosen policy constructed with the sampled scenarios. The framework is generic enough to capture general concave objective functions and global feasibility constraints. Therefore, our algorithms can be adapted to a wide range of operations management applications.

- We establish the value of offline simulation in the design of online planning policy. With the distribution information on the stochastic environment, we demonstrate that the randomized algorithm's optimality gap, which is also known as *regret* in the online optimization literature, converges to zero as the number of time periods (Γ in Figure 2) and the number of scenarios sampled (T

Figure 2 **Schematic drawing for the randomized algorithm.**



Note. Under the sampling scheme, the scenario at each sample $t = 1, \ldots, T$ follows the same distribution as the scenario at period γ^t in the original stochastic problem, where γ^t is uniformly generated at random from $\{1, \ldots, \Gamma\}$.

in Figure 2) increase. In addition, the algorithm produces solution that satisfies the feasibility constraints with high probability. Notably, the algorithm can also be adapted to address the case when the distributions across different planning periods are correlated. Our technical contributions involve a novel transformation from a set of offline samples to a collection of scenarios, which encapsulates critical information about the non-stationary stochastic environment, the concave objective, and the feasibility constraints for optimization.

- Under the solution scheme, we decompose the multi-period online planing problem into multiple single-period convex optimization problems. The decomposition presents great computational advantages on solving large scale stochastic optimization problems, circumventing the curse of dimensionality in the traditional stochastic dynamic programming approach.

The rest of this paper is structured as follows. Relevant literature is reviewed in Section 2. We formally formulate the online planning problem and develop the offline-to-online algorithm in Section 3. We address the case of correlated distributions in Section 4. In Section 5, we use this approach to solve several pertinent supply chain management problems. Section 6 concludes this paper. The proofs of our technical results are relegated to Appendix A. We provide more elaboration of the algorithm and further numerical experiments on an online advertising problem in Appendix B, to demonstrate the general applicability of this framework.

Notation. Let $\|\cdot\|$ denote a norm function on $\mathbb{R}^K$. Let $B(R) := \{\boldsymbol{w} : \|\boldsymbol{w}\| \leq R\}$ denote the closed ball of radius R under $\|\cdot\|$. The dual norm $\|\cdot\|_*$ of $\|\cdot\|$ is defined as $\|\boldsymbol{\theta}\|_* := \max_{\boldsymbol{x} \in B(1)} \boldsymbol{\theta}^\top \boldsymbol{x}$. The closed ball of radius L under the dual norm $\|\cdot\|_*$ is denoted as $B_*(L) := \{\boldsymbol{\theta} : \|\boldsymbol{\theta}\|_* \leq L\}$. For a convex function $\Lambda : D \to \mathbb{R}$, its Fenchel dual is defined as $\Lambda^*(\boldsymbol{\theta}) := \max_{\boldsymbol{w} \in D}\{\boldsymbol{\theta}^\top \boldsymbol{w} - \Lambda(\boldsymbol{w})\}$. When Λ is L-Lipschitz continuous with respect to norm $\|\cdot\|$ and is closed,[3] we have $\Lambda(\boldsymbol{w}) = \max_{\boldsymbol{\theta} \in B_*(L)}\{\boldsymbol{w}^\top \boldsymbol{\theta} - \Lambda^*(\boldsymbol{\theta})\}$.

[3] That is, when the set $\{(\boldsymbol{w}, y) : y \geq \phi(\boldsymbol{w})\}$ is closed with respect to the Euclidean metric topology on $\mathbb{R}^{K+1}$.

We denote $\underline{\partial}\Lambda(\boldsymbol{w})$ as the set of sub-gradient of Λ at $\boldsymbol{w}$. Similarly, for a concave function $\phi : D \to \mathbb{R}$, we define $\phi^*(\boldsymbol{\theta}) := \max_{\boldsymbol{w}\in D} \left\{\boldsymbol{w}^\top \boldsymbol{\theta} + \phi(\boldsymbol{w})\right\}$, which is the Fenchel dual of convex function $-\phi$. Consequently, when ϕ is L-Lipschitz continuous with respect to norm $\|\cdot\|$ and is closed, we have $\phi(\boldsymbol{w}) = \min_{\boldsymbol{\theta}\in B_*(L)} \left\{\phi^*(\boldsymbol{\theta}) - \boldsymbol{\theta}^\top \boldsymbol{w}\right\}$. We denote $\overline{\partial}\phi(\boldsymbol{w})$ as the set of super-gradient of ϕ at $\boldsymbol{w}$.

2. Literature Review

Our online planning framework is related to the online convex optimization (OCO) problem, which has been extensively studied in operations management, operations research, and computer science communities, due to its diverse applications (Shalev-Shwartz et al. 2009, Bubeck et al. 2015).

In the classical OCO setting, in each round the (convex) cost function reveals itself in an adversarial manner after the decision maker selects the action. Zinkevich (2003) introduced the online gradient descent (OGD) algorithm, and demonstrated that this algorithm achieves a sub-linear *static regret* performance bound in comparison with the static hindsight optimal solution. Furthermore, the OCO techniques have been applied to address stochastic convex optimization problems, in which the underlying (stationary) stochastic process is unknown to the decision maker (Huh and Rusmevichientong 2014). Notably, the decision made at each period only depends on past observations in the stationary environment, and hence it is not necessary to specify the stochastic process information a priori or to make any forecast on future scenarios. Motivated by this nonparametric learning scheme, the framework of OCO has been used to address various operations management problems. For example, Zhang et al. (2018) exploited the OCO techniques to address the perishable inventory control problem given that the demands are stationary across time.

In the non-stationary environment, a sequence of dynamic hindsight optimal solutions, which may change over the planning horizon, serves as a more natural benchmark to evaluate the performance of online decisions via a so-called *dynamic regret* metric. Besbes et al. (2015) introduced the concept of variation budget to control the change of dynamic environments/cost functions, and demonstrated that a restarting algorithm based online policy suffices to achieve sub-linear dynamic regret under a sub-linear variance budget. The "price" of non-stationary has been further explored by Chen et al. (2019) and Cheung et al. (2020). Along this direction, the decision made at each period only depends on the historical observations close to this period while the "outdated" information is abandoned. However, there is no room for the "distributional" information coming into the design of online policy.

Another approach to deal with non-stationary stochastic environments in operations management literatures is to specify the stochastic process over the entire planning horizon in advance (Alaei et al. 2012, Wang et al. 2018, Ma et al. 2020). Along this direction, the stochastic information is given in the form of probability distributions. Similar to the aforementioned works, we also assume that the decision maker has a direct access to the stochastic distributional information, but does not know the

exact future scenarios. However, in comparison to the existing works such as Alaei et al. (2012), Wang et al. (2018), Ma et al. (2020), we consider a more general online planning framework which allows for general concave objective function and convex feasibility constraints. To this end, we note that this online planning framework is analogous to the online stochastic convex optimization problem studied by Agrawal and Devanur (2015), who considered both stationary and random permutation inputs. To the best of our knowledge, this is the first paper that uses OCO to address the online planning problem with non-stationary stochastic inputs.

In the end, we note that the accurate distributional information of the environments might not be readily available in practice. An alternate way is to assume that the DM only has access to a collection of scenarios/samples. For example, Vee et al. (2010) studied the online assignment problem based on a set of random samples from future arriving users. They showed that a sampled problem instance suffices to construct near-optimal solution to the full problem. Hardt et al. (2016) studied the performance of stochastic gradient method (SGM) over a finite set of training (offline) samples. They demonstrated the SGM is stable in the sense that the performance under SGM varies slightly if a single point in the training samples is replaced by a new sample generated from the same distribution. Indeed, our O2O algorithm can also provide the same sub-linear regret guarantee given a finite training samples, as long as the scenarios are sampled from the actual process. However, since the base optimization model studied in the present work is different from the models in Vee et al. (2010) and Hardt et al. (2016), our technical results are established based on novel analyses.

3. Models and Analysis

In this section, we formally describe the online planning problem and the main results obtained.

3.1. Problem Description

We consider a general framework of online planning problem, in which the decision maker (DM) interacts with a heterogeneous stochastic environment in Γ time periods. During each time period $\gamma \in \{1, \ldots, \Gamma\}$, the DM interacts with the environment in the following three steps.

1. The environment reveals a scenario $\boldsymbol{\omega}_\gamma$ to the DM. The scenario $\boldsymbol{\omega}_\gamma$ represents the contextual information during period γ. For example, in a two sided platform setting, $\boldsymbol{\omega}_\gamma$ encodes the supply (agents) and demand (customers) avaialble for service during the period. The scenario $\boldsymbol{\omega}_\gamma$ is sampled from a possibly infinite scenario set Ω, according to a distribution Ξ_γ. To facilitate the analysis, we first assume that the distributions $\{\Xi_1, \ldots, \Xi_\Gamma\}$ are independent, *but not identical in general*. The case of correlated distributions will be formally discussed in Section 4. The variations in the distributions $\{\Xi_1, \ldots, \Xi_\Gamma\}$ model changing environments, which are ubiquitous in operational settings.

2. Contingent on the scenario $\boldsymbol{\omega}_\gamma$, the DM selects a decision $\boldsymbol{x}_\gamma$ from a possibly infinite set $\mathcal{X}(\boldsymbol{\omega}_\gamma)$, i.e., the feasible region associated with $\boldsymbol{\omega}_\gamma$. For example, a decision $\boldsymbol{x}_\gamma \in \mathcal{X}(\boldsymbol{\omega}_\gamma)$ could represent a

feasible matching of supply and demand for the scenario $\boldsymbol{\omega}_\gamma$. The decision $\boldsymbol{x}_\gamma$ is in general a random variable with support $\mathcal{X}(\boldsymbol{\omega}_\gamma)$.

3. The quality of the decisions is calibrated based on the outcome of K metrics $\boldsymbol{f}(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma) = (f_k(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma))_{k=1}^K \in [0,1]^K$ to the DM. Note the outcomes $\boldsymbol{f}(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma)$ could depend on γ, and this dependence can be modeled by incorporating the time γ in $\boldsymbol{\omega}_\gamma$. These outcomes could signify the reward earned by the DM for each and every objective.

The DM has limited information about the heterogeneous stochastic environment. In the same line as solving stochastic dynamic programming problems (e.g., Ma et al. 2020), we assume that the DM has a direct access to the scenario distributions $\{\Xi_1, \ldots \Xi_\Gamma\}$. For any $\boldsymbol{\omega} \in \Omega$, the DM knows the feasible decision set $\mathcal{X}(\boldsymbol{\omega})$, and also the outcomes $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\omega})$ for each $\boldsymbol{x} \in \mathcal{X}(\boldsymbol{\omega})$. In the second step, the DM is required to select a decision in a *non-anticipatory* manner. The choice of decision $\boldsymbol{x}_\gamma$ only depends on the observations $\{(\boldsymbol{\omega}_s, \boldsymbol{x}_s, \boldsymbol{f}(\boldsymbol{x}_s, \boldsymbol{\omega}_s))\}_{s=1}^{\gamma-1} \cup \{\boldsymbol{\omega}_\gamma\}$ available at the start of period γ and the DM's knowledge on the probability distributions associated with the environment. Mathematically, by saying that $\{\boldsymbol{x}_\gamma\}_{\gamma=1}^\Gamma$ is non-anticipatory, the decision $\boldsymbol{x}_\gamma$ is required to be $\sigma(\{(\boldsymbol{\omega}_s, \boldsymbol{x}_s, \boldsymbol{f}(\boldsymbol{x}_s, \boldsymbol{\omega}_s))\}_{s=1}^{\gamma-1} \cup \{\boldsymbol{\omega}_\gamma\})$-measurable for each $\gamma \in \{1, \ldots, \Gamma\}$. Note that we do not encode the system status to track how the system changes, and this is different from the Markov decision process model.

The DM wishes to maximize the reward function $\phi(\cdot)$ over the vector of average reward $\sum_{\gamma=1}^\Gamma \boldsymbol{f}(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma)/\Gamma$, while satisfying the constraint that $\sum_{\gamma=1}^\Gamma \boldsymbol{f}(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma)/\Gamma$ must lie in a convex compact set $S \subseteq [0,1]^K$. By appropriately choosing S, the constraint $\sum_{\gamma=1}^\Gamma \boldsymbol{f}(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma)/\Gamma \in S$ captures the operational constraints faced by the DM. For example, when the DM has to ensure that each objective $\sum_{\gamma=1}^\Gamma f_k(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma)/\Gamma$ is at least a certain target KPI $\tau_k \in [0,1]$, these operational constraints can be modelled by

$$d\left(\sum_{\gamma=1}^\Gamma \boldsymbol{f}(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma)/\Gamma, S\right) = 0,$$

with $S = \prod_{k=1}^K [\tau_k, 1]$, and $d(\cdot,\cdot)$ denotes a distance function. If $f_1(\boldsymbol{x}, \boldsymbol{w})$ denotes the profit earned by choosing solution $\boldsymbol{x}$ in scenario $\boldsymbol{w}$, and our objective is to maximize aggregate profit, we could choose

$$\phi\left(\sum_{\gamma=1}^\Gamma \boldsymbol{f}(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma)/\Gamma\right) := \sum_{\gamma=1}^\Gamma f_1(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma)/\Gamma.$$

In this way, this online planning framework can be used to model a variety of operational problems.

In summary, the DM is faced with the following online planning problem $\mathrm{DP}_1(\phi, S)$. In addition to the feasible region S, the new problem involves a concave reward function $\phi : \mathbb{R}^K \to \mathbb{R}$. The function ϕ is L-Lipschitz continuous with respect to a fixed norm $\|\cdot\|$. Define $d(\boldsymbol{w}, S) := \min_{\boldsymbol{u} \in S} \|\boldsymbol{w} - \boldsymbol{u}\|$. The online planning problem can be expressed as

$$\mathrm{DP}_1(\phi, S) : \max\ \phi\left(\frac{1}{\Gamma}\sum_{\gamma=1}^\Gamma \boldsymbol{f}(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma)\right)$$

$$\text{s.t. } d\left(\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}(\boldsymbol{x}_\gamma,\boldsymbol{\omega}_\gamma),S\right)=0,\ \text{almost surely,} \tag{1}$$
$$\boldsymbol{x}_\gamma \text{ non-anticipatory},\ \gamma=1,\ldots,\Gamma,$$
$$\boldsymbol{x}_\gamma\in\mathcal{X}(\boldsymbol{\omega}_\gamma),\ \gamma=1,\ldots,\Gamma.$$

The feasibility constraints (1) require $\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}(\boldsymbol{x}_\gamma,\boldsymbol{\omega}_\gamma)$ to lie in S. In particular, the constraints are independent of underlying norm $\|\cdot\|$ that defines the distance function. The reason of expressing the constraint (1) in terms of $d(\cdot,\cdot)$ is that, in the forthcoming section, we propose non-anticipatory policies that are nearly feasible, in the sense that they satisfy $d(\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}(\boldsymbol{x}_\gamma,\boldsymbol{\omega}_\gamma)/\Gamma,S)\le\epsilon$ for some $\epsilon>0$ with high probability. In such a guarantee, the identity of the underlying norm is consequential. For the ease of exposition, we make the following assumption to make sure that the problem $\mathrm{DP}_1(\phi,S)$ is well-posed by ensuring its feasibility.

Assumption 1. *The instance of problem $DP_1(\phi,S)$ is always feasible, i.e., there exists $\boldsymbol{x}(\omega_\gamma)\in\mathcal{X}(\boldsymbol{\omega}_\gamma)$ for $\gamma=1,\ldots,\Gamma$ such that $\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}(\boldsymbol{x}(\omega_\gamma),\boldsymbol{\omega}_\gamma)/\Gamma\in S$.*

Next, as stated in Assumption 2, we assume that the forecast model $\{\Xi_\gamma\}_{\gamma=1}^{\Gamma}$ is accessible to the DM as a data generating oracle.

Assumption 2. *For any $\gamma\in\{1,\ldots,\Gamma\}$ and any given positive integer T, the DM can draw T i.i.d. samples that are distributed as Ξ_γ.*

3.2. Upper Bound on the Expected Optimum of $\mathrm{DP}_1(\phi,S)$

We denote the optimal value of $\mathrm{DP}_1(\phi,S)$ as $\mathrm{opt}(\mathrm{DP}_1(\phi,S))$, and consider the expected optimum $\mathbb{E}[\mathrm{opt}(\mathrm{DP}_1(\phi,S))]$, where the expectation is taken over the random realization of the scenarios $\{\boldsymbol{\omega}_\gamma\}_{\gamma=1}^{\Gamma}$ and the inherent randomness in the optimal non-anticipatory policy. Achieving $\mathbb{E}[\mathrm{opt}(\mathrm{DP}_1(\phi,S))]$ is a daunting task. Even when the joint distribution $\Xi_{1:\Gamma}$ of the scenarios is provided to the DM, it is still computationally intractable to compute a non-anticipatory policy that achieves the expected optimum $\mathbb{E}[\mathrm{opt}(\mathrm{DP}_1(\phi,S))]$. Consequently, the benchmark $\mathrm{opt}(\mathrm{DP}_1(\phi,S))$ still appears rather opaque. Instead, we introduce an "upper bound" benchmark problem by removing the non-anticipatory constraints in $\mathrm{DP}_1(\phi,S)$, which yields a more tractable benchmark. For a scenario $\boldsymbol{\omega}_\gamma$ at period γ, we denote $\mathcal{F}(\boldsymbol{\omega}_\gamma):=\{\boldsymbol{f}(\boldsymbol{x}_\gamma,\boldsymbol{\omega}_\gamma):\boldsymbol{x}_\gamma\in\mathcal{X}(\boldsymbol{\omega}_\gamma)\}$ as the collection of possible outcomes, and denote $\mathrm{Conv}(\mathcal{F}(\boldsymbol{\omega}_\gamma))$ as the convex hull of the outcome set $\mathcal{F}(\boldsymbol{\omega}_\gamma)$. Now, for a realization of scenarios $\boldsymbol{\omega}_1,\ldots,\boldsymbol{\omega}_\Gamma$, we define

$$\begin{aligned}\mathrm{UB}(\phi,S):=\max_{\boldsymbol{x}}\ &\phi\left(\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}_\gamma\right)\\ \text{s.t. }\ &d\left(\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}_\gamma,S\right)\le 0,\\ &\boldsymbol{f}_\gamma\in\mathrm{Conv}(\mathcal{F}(\boldsymbol{\omega}_\gamma)),\ \gamma=1,\ldots,\Gamma.\end{aligned}$$

The convex hull $\text{Conv}(\mathcal{F}(\boldsymbol{\omega}_\gamma))$ incorporates the randomization of the optimal policy for selecting the decision at period γ. Similar to $\text{DP}_1(\phi, S)$, we define $\text{opt}(\text{UB}(\phi, S))$ as the optimal solution to $\text{UB}(\phi, S)$, and denote $\mathbb{E}[\text{opt}(\text{UB}(\phi, S))]$ as its expected optimum. Different from $\mathbb{E}[\text{opt}(\text{DP}_1(\phi, S))]$, the expectation in $\mathbb{E}[\text{opt}(\text{UB}(\phi, S))]$ is taken solely over the random realization of $\boldsymbol{\omega}_{1:\Gamma}$. The following proposition justifies the choice of the benchmark $\mathbb{E}[\text{opt}(\text{UB}(\phi, S))]$.

PROPOSITION 1. *Suppose Assumption 1 holds. For any realization of $\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_\Gamma$, the upper bound problem $UB(\phi, S)$ is feasible. In addition, it holds that $\mathbb{E}[opt(UB(\phi, S))] \geq \mathbb{E}[opt(DP_1(\phi, S))]$.*

The proposition is proved in Appendix A.1, which further explains the role of $\text{Conv}(\mathcal{F}(\boldsymbol{\omega}_\gamma))$.

3.3. The Online Planning Problem with No Constraint ($S = \mathbb{R}^K$): $DP_1(\phi, \mathbb{R}^K)$

To gain intuitions and insights, we first study the non-constrained problem $\text{DP}_1(\phi, \mathbb{R}^K)$, by replacing the feasible region S with $\mathbb{R}^K$. Clearly, the constraint (1) is always satisfied in this specialized problem so that we can focus on maximizing the global concave function $\phi(\cdot)$. Notably, by specifying $\phi(\boldsymbol{w}) = -d(\boldsymbol{w}, S)$ for a convex set $S \subseteq [0,1]^K$, the objective is equivalent to minimizing the distance from the vector $\boldsymbol{w}$ to the feasible region S. Therefore, the no-constraint problem $\text{DP}_1(-d(\boldsymbol{w}, S), \mathbb{R}^K)$ also captures the feasibility problem $\text{DP}_1(0, S)$, where the DM aims to have the vector of average reward $\sum_{\gamma=1}^{\Gamma} \boldsymbol{f}(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma)/\Gamma$ as close to the feasible region S as possible.

In this section, we develop a novel *Offline-to-Online* (O2O) algorithm to solve this problem.

O2O Algorithm. Our algorithmic framework consists of the offline and the online algorithms, which are respectively displayed in Algorithms 1 and 2. Both offline and online algorithms assume the access to the following optimization oracle $\mathcal{O}$:

ASSUMPTION 3. *The DM has the access to an optimization oracle $\mathcal{O}$. For each $\boldsymbol{\theta} \in \mathbb{R}^K$, $\boldsymbol{\omega} \in \Omega$, the oracle call $\mathcal{O}(\boldsymbol{\theta}, \boldsymbol{\omega})$ returns an optimal solution to the optimization problem:*

$$\max_{\boldsymbol{x} \in \mathcal{X}(\boldsymbol{\omega})} \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\omega}).$$

When there are multiple optimal solutions, the oracle selects an optimal solution consistently, so that the oracle defines a well-defined function that maps each $(\boldsymbol{\theta}, \boldsymbol{\omega})$ to an optimal solution.

The offline algorithm is run before the DM encounters the actual scenarios $\{\boldsymbol{\omega}_\gamma\}_{\gamma=1}^{\Gamma}$ in online problem instance $\text{DP}_1(\phi, \mathbb{R}^K)$. While DM has no access to the real time information $\{\boldsymbol{\omega}_\gamma\}_{\gamma=1}^{\Gamma}$ when he runs the offline algorithm, the DM does have access to samples of the distributions $\Xi_{\gamma}{}_{\gamma=1}^{\Gamma}$, which characterize implicitly the stochastic environment the DM is operating in.

As described in Algorithm 1, the offline algorithm involves a sampling procedure (Line 3-10) to capture the future information during the entire planning horizon and produces a collection of gradients which are crucial for solving $\text{DP}_1(\phi, \mathbb{R}^K)$. Through the sampling procedure, we translate

Algorithm 1 Offline algorithm, to be run before the online process.

1: **Input:** Objective function ϕ, number of iterations T, learning rate $\eta > 0$, regularizer Λ, and domain $D \subseteq B_*(L)$.
2: **Initialize:** Initial weight vector $\boldsymbol{\theta}^1 = \min_{\boldsymbol{\theta} \in D} \Lambda(\boldsymbol{\theta})$.
3: **for** $t = 1, \ldots, T$ **do**
4: Sample a period index γ^t uniformly at random from $\{1, \ldots, \Gamma\}$.
5: Sample a random scenario $\boldsymbol{\omega}^t$ according to the distribution Ξ_{γ^t}.
6: Compute decision $\boldsymbol{x}^t = \mathcal{O}(-\boldsymbol{\theta}^t, \boldsymbol{\omega}^t)$, by calling the optimization oracle $\mathcal{O}$.
7: Compute gradient $\boldsymbol{z}^t = \nabla_{\boldsymbol{\theta}}[g^t(\boldsymbol{\theta}^t)]$, where $g^t(\boldsymbol{\theta}) := \phi^*(\boldsymbol{\theta}) + (-\boldsymbol{\theta})^\top \boldsymbol{f}(\boldsymbol{x}^t, \boldsymbol{\omega}^t)$.
8: ▷ By duality, $\boldsymbol{z}^t$ can be computed by solving the concave maximization problem $\boldsymbol{z}^t \in \operatorname{argmax}_{\boldsymbol{w} \in [0,1]^K} \{(\boldsymbol{\theta}^t)^\top \boldsymbol{w} + \phi(\boldsymbol{w}) - \boldsymbol{f}(\boldsymbol{x}^t, \boldsymbol{\omega}^t)\}$.
9: Compute weight vector $\boldsymbol{\theta}^{t+1} = \operatorname{argmin}_{\boldsymbol{\theta} \in D} \left\{ \eta \cdot \left[\sum_{q=1}^t \boldsymbol{z}^{q\top} \boldsymbol{\theta} \right] + \Lambda(\boldsymbol{\theta}) \right\}$.
10: **end for**
11: **Output:** The collection of weight vectors $\Theta = \{\boldsymbol{\theta}^t\}_{t=1}^T$.

the non-stationary stochastic problem into a "multi-sample" optimization problem (cf. Figure 2). In Line 9, we develop an online mirror descent (OMD) algorithm to solve this multi-sample problem. In this way, the offline algorithm combines OMD with oracle $\mathcal{O}$ by solving the linearized problem from a randomly drawn period and a randomly drawn scenario. The linearization is adaptively updated, by virtue of OMD, to ensure that the output Θ is useful for the online planning problem. To make the discussion clearer, we use superscript t to indicate the t^{th} sample in the offline problem, which is different from the subscript γ used for the γ period scenario in the online instance. We remark that Algorithm 1 requires the information of all periods, i.e. $\Xi_1, \ldots, \Xi_\Gamma$. Indeed, in the problem $\mathrm{DP}_1(\phi, \mathbb{R}^K)$, the optimal decision in each period is in general dependent on all the scenarios.

Algorithm 2 Online algorithm, to be run during the online process.

1: **Input:** A finite collection of weight vectors Θ from Algorithm 1.
2: **for** $\gamma = 1, \ldots, \Gamma$ **do**
3: Sample a weight vector $\boldsymbol{\theta}_\gamma$ uniformly at random from Θ.
4: Observe the γ-period scenario $\boldsymbol{\omega}_\gamma$.
5: Select decision $\boldsymbol{x}_\gamma$, where $\boldsymbol{x}_\gamma = \mathcal{O}(-\boldsymbol{\theta}_\gamma, \boldsymbol{\omega}_\gamma)$, by calling the optimization oracle $\mathcal{O}$.
6: **end for**

The online algorithm (Algorithm 2) is run when the DM encounters the online problem $\mathrm{DP}_1(\phi, \mathbb{R}^K)$. As stated, the DM needs to make decisions $\boldsymbol{x}_\gamma$'s based on the observed scenarios $\boldsymbol{\omega}_\gamma \sim \Xi_\gamma$ on the fly. In Line 3, the online algorithm involves a randomization algorithm in the sense that the weight vector $\boldsymbol{\theta}_\gamma$ at each period γ is randomly generated from Θ. In this way, the online algorithm essentially tries

to mimic the offline algorithm, by solving linearized problems with weight vectors drawn u.a.r. from Θ. We hightlight that $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_\Gamma$ are i.i.d., and they are drawn u.a.r. with replacement from Θ.

To be more concrete, the offline algorithm is designed based on the online mirror descent (OMD) algorithm, which is based on the Mirror Descent algorithm proposed by Nemirovsky and Yudin (1983). OMD has been employed to solve online optimization problems in the i.i.d. and random permutation settings by Agrawal and Devanur (2015). A survey on OMD is provided in Shalev-Shwartz et al. (2012). The OMD algorithm requires a *mirror map* $\Lambda(\cdot)$, which is typically a strongly convex function. The notion of strong convexity is defined below:

DEFINITION 1. Let $\alpha \geq 0$. A function $\Lambda : D \to \mathbb{R}$ is α-strongly convex over the domain D with respect to the norm $\|\cdot\|_*$, if we have

$$\Lambda(\boldsymbol{u}) \geq \Lambda(\boldsymbol{w}) + \boldsymbol{z}^\top(\boldsymbol{u} - \boldsymbol{w}) + \frac{\alpha}{2}\|\boldsymbol{u} - \boldsymbol{w}\|_*^2,$$

for any $\boldsymbol{u}, \boldsymbol{w} \in D$ and $\boldsymbol{z} \in \underline{\partial}\Lambda(\boldsymbol{w})$.

In the following proposition, we recall the algorithm details of OMD and its performance guarantee.

PROPOSITION 2 **(Nemirovsky and Yudin (1983), Shalev-Shwartz et al. (2012))**. *Let $g_1, \ldots, g_T$ be a sequence of convex and R-Lipschitz function with respect to the norm $\|\cdot\|_*$ on domain D. In addition, let Λ be a 1-strongly convex function over the domain D with respect to $\|\cdot\|_*$. Consider the OMD algorithm with learning rate $\eta = \lambda/(R\sqrt{T})$, where $\lambda^2 := \max_{\boldsymbol{\theta}\in D}\{\Lambda(\boldsymbol{\theta})\} - \min_{\boldsymbol{\theta}\in D}\{\Lambda(\boldsymbol{\theta})\}$. For each $t = 1, \ldots, T$, perform*

$$\boldsymbol{\theta}_t \leftarrow \mathit{argmin}_{\boldsymbol{\theta}\in D}\left\{\eta \cdot \left[\sum_{q-1}^{t-1} \boldsymbol{\theta}^\top \boldsymbol{z}_q\right] + \Lambda(\boldsymbol{\theta})\right\}, \tag{2}$$

where $\boldsymbol{z}_q \in \underline{\partial} g_q(\boldsymbol{\theta}_q)$. The following inequality holds:

$$\frac{1}{T}\sum_{t=1}^{T} g_t(\boldsymbol{\theta}_t) - \min_{\boldsymbol{\theta}\in D}\left\{\frac{1}{T}\sum_{t=1}^{T} g_t(\boldsymbol{\theta})\right\} \leq \frac{2\lambda R}{\sqrt{T}}.$$

Note that the OMD algorithm provides versatile tools for online optimization in our settings, since it can be applied to problem settings with different underlying norms, assuming the corresponding mirror map. As Algorithm 1 could appear rather abstract (though general), we extract from the online optimization literature on the incarnation of the offline algorithm in special cases. We highlight two examples for the cases when $\|\cdot\| = \|\cdot\|_2$ and $\|\cdot\| = \|\cdot\|_\infty$. The descriptions of the two examples follow the exposition in Shalev-Shwartz et al. (2012).

EXAMPLE 2. [Gradient Descent under $\|\cdot\|_2$-Lipschitz Continuity] Consider the case when the underlying norm is Euclidean, that is, ϕ is L-Lipschitz continuous w.r.t. $\|\cdot\| = \|\cdot\|_2$. The execution of the offline algorithm requires a mirror map Λ to be 1-strongly convex w.r.t. the dual norm $\|\cdot\|_* = \|\cdot\|_2$

over the domain $D = B_*(L)$, which is the Euclidean ball of radius L. An eligible candidate is $\Lambda_2(\boldsymbol{\theta}) := \boldsymbol{\theta}^\top\boldsymbol{\theta}/2 + I_{B(L,\|\cdot\|_*)}(\boldsymbol{\theta})$.[4] Next, in order to achieve the convergence postulated in Proposition 2, we set the learning rate $\eta_2 = \lambda_2/(R_2\sqrt{T})$, where $R_2 = 2\|\mathbf{1}_K\|_2 = 2\sqrt{K}$, and $\lambda_2 = L/\sqrt{2}$. Consequently, we have $\eta_2 = L/(\sqrt{8KT})$.

Altogether, under the specified learning rate η_2, mirror map Λ_2 and domain $D = B_*(L)$, the gradient update rule in Line 9 of Algorithm 1 is:

$$\begin{aligned}\boldsymbol{\theta}^{t+1} &\leftarrow \operatorname{argmin}_{\boldsymbol{\theta}:\|\boldsymbol{\theta}\|_2\le L}\left\{\frac{L}{\sqrt{8KT}}\left[\sum_{q=1}^{t}(\boldsymbol{z}^q)^\top\boldsymbol{\theta}\right] + \frac{1}{2}\boldsymbol{\theta}^\top\boldsymbol{\theta}\right\}\\ &= \operatorname{argmin}_{\boldsymbol{\theta}:\|\boldsymbol{\theta}\|_2\le L}\left\{\left\|\boldsymbol{\theta} + \frac{L}{\sqrt{8KT}}\sum_{q=1}^{t}\boldsymbol{z}^q\right\|_2^2\right\}\\ &= -\frac{L\sum_{q=1}^{t}\boldsymbol{z}^q}{\max\left\{\sqrt{8KT}, \|\sum_{q=1}^{t}\boldsymbol{z}^q\|_2\right\}}. \end{aligned} \tag{3}$$

■

Example 3. [Multiplicative Weight Update under $\|\cdot\|_\infty$-Lipschitz Continuity] In certain applications, such as resource constrained problems with Lagrangian relaxations, the reward function ϕ is L-Lipschitz w.r.t. $\|\cdot\| = \|\cdot\|_\infty$. Additionally, for all $\boldsymbol{w} \in [0,1]^K$, we have $\overline{\partial}\phi(\boldsymbol{w}) \subseteq S_{\ge 0}(L, \|\cdot\|_*) := \{\boldsymbol{\theta} : \|\boldsymbol{\theta}\|_* = L, \boldsymbol{\theta} \ge 0\}$, where $\|\cdot\|_* = \|\cdot\|_1$. In such applications, it suffices to define a mirror map Λ_∞ that is 1-strongly convex w.r.t. to $\|\cdot\|_1$ over the domain $D = S_{\ge 0}(L, \|\cdot\|_1)$. An eligible candidate is the negative entropy function

$$\Lambda_\infty(\boldsymbol{\theta}) = L\sum_{k=1}^{K}\theta_k\log\theta_k + I_{S_{\ge 0}(L,\|\cdot\|_1)}(\boldsymbol{\theta}), \tag{4}$$

where $0\log 0 := 0$. To achieve the convergence rate postulated in Proposition 2, we set the learning rate $\eta_\infty = \lambda_\infty/(R_\infty\sqrt{T})$, where $\lambda_\infty = L\sqrt{\log K}$ and $R = 2$. Consequently, we have $\eta_\infty = L\sqrt{\log K}/(2\sqrt{T})$.

Altogether, the gradient update rule in Line 9 of Algorithm 1 has the following incarnation:

$$\boldsymbol{\theta}^{t+1} = \frac{(Le^{w_{t,1}}, \ldots, Le^{w_{t,K}})}{\sum_{k=1}^{K} e^{w_{t,k}}}, \text{ where } w_{t,k} := -\sqrt{\frac{\log K}{4T}}\sum_{q=1}^{t} z_k^q.$$

■

We highlight that both the offline and online algorithms could be implemented efficiently. Indeed, given Assumption 3, we can solve for the decision in the scalarized problem (cf. Line 6 of Algorithm 1 and Line 5 of Algorithm 2) in each iteration efficiently. Next, to update the weight vector $\boldsymbol{\theta}_t$ (cf. Line 9 of Algorithm 1), Example 2 and 3 have depicted two closed-form polices. The calculation

[4] For a set $U \subseteq \mathbb{R}^K$, the function I_U is defined as : $I_U(\boldsymbol{\theta}) = 0$ if $\boldsymbol{\theta} \in U$, and $I_U(\boldsymbol{\theta}) = \infty$ if $\boldsymbol{\theta} \notin U$.

of gradient $\boldsymbol{z}^t$ (cf. Line 7 of Algorithm 1) involves a concave maximization problem, which can also be solved efficiently. For example, when $\phi(\boldsymbol{w}) = -d(\boldsymbol{w}, S)$ for some convex set $S \subseteq [0,1]^K$. The objective is to minimize the distance from the vector $\boldsymbol{w}$ to the target set S. In this case, we have $\boldsymbol{z}^t \in \operatorname{argmax}_{\boldsymbol{w}\in S}(\boldsymbol{\theta}^t)^\top \boldsymbol{w}$. Furthermore, when the set S is a polyhedral, this is indeed a linear program. To make the discussion concrete, we describe a specific implementation of the O2O algorithm for the resource allocation problem in Section 5.1.

Now, we are ready to establish the main results developed based on the O2O algorithm.

Main Results. We demonstrate the performance guarantee of our O2O algorithm on $\mathrm{DP}_1(\phi, \mathbb{R}^K)$ in the form of regret bound. The regret of an algorithm is defined as

$$\mathrm{Reg}_1(\Gamma, T) := \mathbb{E}[\mathrm{opt}(\mathrm{UB}(\phi, S))] - \phi\left(\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma} \boldsymbol{f}(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma)\right),$$

which is the difference between the offline benchmark and the algorithm's objective. The offline benchmark is an upper bound on the optimal expected reward collected by an oracle who knows the true distribution $\Xi_{1:\Gamma}$. For the statement of our regret bound, we employ the $\tilde{O}(\cdot)$ notation, which conceals multiplicative logarithmic factors of order $\log(K\max\{T,\Gamma\}/\delta)$ and additive terms of the order $\log(K\max\{T,\Gamma\}/\delta)/\min\{\Gamma, T\}$, where δ is the order of the probabilistic guarantee.

Theorem 1. *Suppose that Assumptions 2, 3 hold. Consider the application of Algorithm 2 to the online problem ($DP_1(\phi, \mathbb{R}^K)$), where the input Θ is generated by Algorithm 1. With probability at least $1 - O(\delta)$, we have*

$$Reg_1(\Gamma, T) = \tilde{O}\left(L\,\|\mathbf{1}_K\|\left(\frac{1}{\sqrt{\Gamma}} + \frac{\max\{\lambda, 1\}}{\sqrt{T}}\right)\right).$$

In the regret bound, the term associated with T comes from two parts – the performance loss due to the sampling procedure in Lines 3-10 of Algorithm 1, and the performance loss due to the implementation of OMD algorithm. This term reflects the non-asymptotic performance guarantee of our algorithm. Therefore, the performance of O2O algorithm improves with the sample size T. The term associated with Γ captures the impact of planning horizon on the performance of our O2O algorithm. Intuitively, the algorithm tolerates more "errors" over a longer planning horizon. With a longer planning horizon, we would expect better performance by implementing the O2O algorithm. In addition, the constant λ is defined in Proposition 2. The probability guarantee term δ is suppressed in the notation $\tilde{O}(\cdot)$. In this way, our O2O algorithm rigorously incorporates the forecast information and crystallizes the delicate balance between historical performance and future forecast in the non-stationary environment.

To make the discussion clearer, we sketch the proof of Theorem 1 below. We first bound the generalization error incurred by employing the sampling procedure in the offline algorithm and randomization algorithm in the online algorithm. Under the O2O algorithm, we assert that

$$\phi\left(\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}(\boldsymbol{x}_\gamma,\boldsymbol{\omega}_\gamma)\right)\geq\phi\left(\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{f}(\boldsymbol{x}^t,\boldsymbol{\omega}^t)\right)-\underbrace{L\,\|\mathbf{1}_K\|\sqrt{\frac{2\log(6K/\delta)}{\Gamma}}}_{(\ddagger)}-\underbrace{L\,\|\mathbf{1}_K\|\sqrt{\frac{2\log(6K/\delta)}{T}}}_{(\dagger)},\tag{5}$$

(w.p. $1-2\delta/3$)

where the term (†) concerns the online algorithm, and the term (‡) concerns the offline algorithm. Conditioned on the output $\Theta=\{\boldsymbol{\theta}^t\}_{t=1}^T$ generated in the offline algorithm are i.i.d., and then they are uniformly drawn in the online algorithm, we derive the terms (†, ‡) using Azuma-Hoeffding inequality (cf. Proposition 3 in Appendix A).

We continue to bound $\phi(\sum_{t=1}^T\boldsymbol{f}(\boldsymbol{x}^t,\boldsymbol{\omega}^t)/T)$. By considering the OMD procedure on the dual of the reward function, we claim that

$$\phi\left(\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{f}(\boldsymbol{x}^t,\boldsymbol{\omega}^t)\right)\geq\mathbb{E}[\mathrm{opt}(\mathrm{UB}(\phi,\mathbb{R}^K))]-\underbrace{\frac{4\lambda L\|\mathbf{1}_K\|}{\sqrt{T}}}_{(\$)}-\underbrace{\frac{L\,\|\mathbf{1}_K\|\sqrt{2\log(6K/\delta)}}{\sqrt{T}}}_{(\pounds)}\tag{6}$$

(w.p. $1-\delta/3$)

The term (\$) is derived by applying Proposition 2 on the series of functions $\{g_t\}_{t=1}^T$, defined as $g_t(\boldsymbol{\theta})=\phi^*(\boldsymbol{\theta})-\boldsymbol{\theta}^\top\boldsymbol{f}(\boldsymbol{x}^t,\boldsymbol{\omega}^t)$. The term (£) is by applying the Hoeffding inequality on the random sequence associated with $\{\boldsymbol{\omega}^1,\ldots,\boldsymbol{\omega}^T\}$ in the offline algorithm.

Combining Equation (5) and (6), we demonstrate that the O2O algorithm achieves near-optimal performance guarantee for problem $\mathrm{DP}_1(\phi,\mathbb{R}^K)$. ■

Finally, we note that our analysis is tight. The second term $\tilde{O}(L\|\mathbf{1}_K\|\max\{\lambda,1\}/\sqrt{T})$, which accounts for the optimality gap from the offline algorithm, matches the regret bound for OMD in Proposition 2. In the first term $\tilde{O}(L\|\mathbf{1}_K\|/\sqrt{\Gamma})$, which accounts for the optimality gap from the online algorithm, the dependence $\tilde{O}(1/\sqrt{\Gamma})$ matches the stochastic error bound by the Hoeffiding's inequality. The multiplicative factor $L\|\mathbf{1}_K\|$ naturally arises from the L-Lipschitz continuity of ϕ.

3.4. Solving the General Problem $\mathbf{DP}_1(\phi,S)$

In this section, we solve the general problem $\mathrm{DP}_1(\phi,S)$. Note that we need to maximize the objective and guarantee that the constraints are satisfied. To facilitate the analysis, we consider two types of regrets corresponding respectively to regret in the objective function and feasibility constraint:

$$\mathrm{Reg}_1(\Gamma,T)=\mathbb{E}[\mathrm{opt}(\mathrm{UB}(\phi,S))]-\phi\left(\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}(\boldsymbol{x}_\gamma,\boldsymbol{\omega}_\gamma)\right),$$

$$\mathrm{Reg}_2(\Gamma, T) = d\left(\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma} \boldsymbol{f}_\gamma(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma), S\right).$$

Algorithmic Framework. Leveraging on the O2O algorithm introduced in Section 3.3, we develop here an approach (Algorithm 3) to solve the general problem. Similar to before, Algorithm 3 also assumes the access to the forecast oracle (as per Assumption 2), and the access to the optimization oracle (as per Assumption 3), for running the offline algorithm. We first obtain an estimate of the optimal objective value ($\hat{Z}$) through sampling average approximation, and then use this as a new constraint to guide the online planning decisions using the previous approach. The details are described in Algorithm 3.

Algorithm 3 Algorithm for the general problem $\mathrm{DP}_1(\phi, S)$.

1: **Input:** Reward function ϕ, constraint set S. Input parameters T, η, Λ, D for the offline algorithm, and confidence parameter δ for objective value estimation.
2: **for** $\tau = 1, \ldots, T$ **do**
3: Draw a sequence of samples $\boldsymbol{\omega}^\tau_{1:\Gamma} \sim \Xi_{1:\Gamma}$ based on the forecast model.
4: Solve for the optimal value Z_τ of $\mathrm{UB}(\phi, S)$ under the scenarios $\boldsymbol{\omega}^\tau_{1:\Gamma}$.
5: **end for**
6: Compute the estimated optimum $\hat{Z} = \frac{1}{T}\sum_{\tau=1}^{T} Z_\tau$.
7: Define the following concave objective function $\breve{\phi} : [0,1]^K \to \mathbb{R}^K$:

$$\breve{\phi}(\boldsymbol{w}) = -d(\boldsymbol{w}, \breve{S}) \text{ , where } \breve{S} = S \cap \left\{\boldsymbol{w} \in [0,1]^K : \phi(w) \geq \hat{Z} - 2L\|\mathbf{1}_K\|\sqrt{\frac{2\log(2/\delta)}{T}}\right\}.$$

8: Apply the offline algorithm (Algorithm 1) with reward function $\breve{\phi}$, number of iterations T, as well as input η, Λ, D. The offline algorithm returns a collection of weight vectors Θ.
9: The DM applies the online algorithm (Algorithm 2) with the collection Θ, which returns a sequence of decisions $\{\boldsymbol{x}_\gamma\}_{\gamma=1}^{\Gamma}$ contingent upon the scenarios $\{\boldsymbol{\omega}_\gamma\}_{\gamma=1}^{\Gamma}$ revealed online.

We note that Lines 2 – 8 are run *before* the DM starts the online process that interacts with the sequential scenarios $\{\boldsymbol{\omega}_\gamma\}_{\gamma=1}^{\Gamma}$. Lines 2 – 8 result in the set of gradients Θ, which are used to make online decisions in Line 9 by applying Algorithm 2. In Line 7, we refine the constraint set to $\breve{S}$, and require that the attained objective value $\phi(\cdot)$ should be close to the estimated optimal value $\hat{Z}$, with a small enough estimation error in the order of $\sqrt{T}$. In this way, we extend the O2O algorithm to solve problem $\mathrm{DP}_1(\phi, S)$ by translating the planning problem into a no-constraint problem with objective function $\breve{\phi}(\cdot) = -d(\cdot, \breve{S})$.

Main Results. We establish the performance guarantee of Algorithm 3 for the general problem $\mathrm{DP}_1(\phi, S)$ in terms of regret bounds on both $\mathrm{Reg}_1(\Gamma, T)$ and $\mathrm{Reg}_2(\Gamma, T)$. The O2O algorithm is shown to be near-optimal, and satisfies the feasibility constraints with high probability.

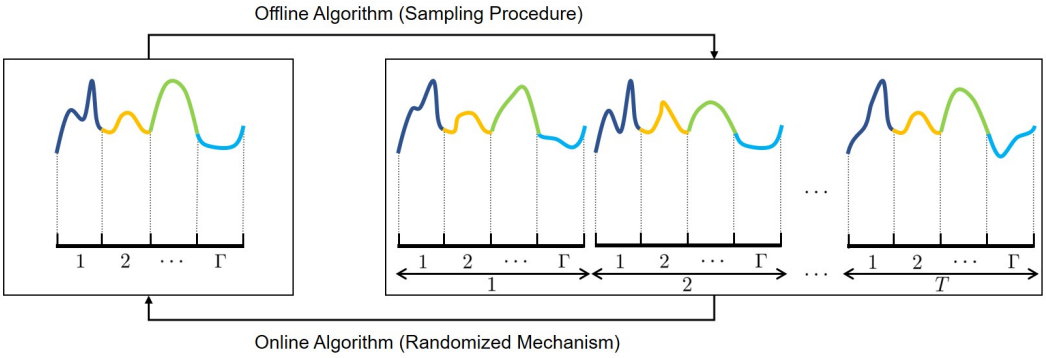
THEOREM 2. *Consider the problem $DP_1(\phi, S)$, and suppose that Assumptions 1, 2, and 3 are satisfied. With probability $1 - O(\delta)$, Algorithm 3 satisfies the following regret bounds for $DP_1(\phi, S)$:*

$$Reg_1(\Gamma, T) = \tilde{O}\left(L \|\mathbf{1}_K\| \left(\frac{1}{\sqrt{\Gamma}} + \frac{\max\{\lambda, 1\}}{\sqrt{T}}\right)\right),$$
$$Reg_2(\Gamma, T) = \tilde{O}\left(\|\mathbf{1}_K\| \left(\frac{1}{\sqrt{\Gamma}} + \frac{\max\{\lambda, 1\}}{\sqrt{T}}\right)\right).$$

4. Extension: Correlated Distributions

In this section, we consider the case when the distributions $\{\Xi_1, \ldots, \Xi_\Gamma\}$ could be arbitrary correlated over the entire planning horizon. To address this case, we refine the O2O framework, as described in Figure 3. For ease of exposition, we focus on the non-constrained problem $\mathrm{DP}_1(\phi, \mathbb{R}^K)$, while the general problem $\mathrm{DP}_1(\phi, S)$ could be solved using the similar algorithmic framework developed in section 3.4. The following Algorithm 4 and 5 illustrate the logic of this refined O2O policy.

Figure 3 Schematic drawing for the (refined) randomized algorithm.



Note. Under the sampling scheme, the stochastic processes over Γ periods are i.i.d. re-generated for T samples in the offline problem.

With a slight abuse of notation, we also denote $\boldsymbol{\omega}_\gamma^t \sim \Xi_\gamma$ as the generated scenario for period γ at sample t, and $\boldsymbol{x}_\gamma^t$ as the corresponding solution. The offline algorithm is also run before the DM encounters the online problem $\mathrm{DP}_1(\phi, \mathbb{R}^K)$, and the DM has no access to the real time information $\{\boldsymbol{\omega}_\gamma\}_{\gamma=1}^\Gamma$ when he runs the offline algorithm. As described in Algorithm 4, the offline algorithm involves a refined sampling procedure (Line 3-10) to capture the information of distributions that might be correlated during the entire planning horizon and produces a collection of gradients which are crucial for solving $\mathrm{DP}_1(\phi, \mathbb{R}^K)$. Through the refined sampling procedure, we also translate the non-stationary stochastic problem into a "multi-sample" optimization problem (cf. Figure 3). Notably, since the scenarios could be correlated, we cannot sample the scenario for each single period as described in Algorithm 1. Instead, we sample the scenarios over the entire Γ planning horizon at each sample

$t = 1, \ldots, T$ (Line 4). Furthermore, from Line 5 to 7, we call the oracle $\mathcal{O}(-\boldsymbol{\theta}^t, \cdot)$ to solve the planning problem using the same $\boldsymbol{\theta}_t$ at each period $\gamma = 1, \ldots, \Gamma$. In Line 9, we also apply the OMD algorithm to solve this multi-sample problem. Although we change the way to sample the scenarios, we highlight that the output $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_\Gamma$ are i.i.d., and they will be drawn u.a.r. with replacement from Θ to solve the online planning problem.

Algorithm 4 (Refined) Offline algorithm, to be run before the online process.

1: **Input:** Objective function ϕ, number of iterations T, learning rate $\eta > 0$, regularizer Λ, and domain $D \subseteq B_*(L)$.
2: **Initialize:** Initial weight vector $\boldsymbol{\theta}^1 = \min_{\boldsymbol{\theta} \in D} \Lambda(\boldsymbol{\theta})$.
3: **for** $t = 1, \ldots, T$ **do**
4: Sample a sequence of random scenarios $\boldsymbol{\omega}_\gamma^t \sim \Xi_\gamma$ for $\gamma = 1, \ldots, \Gamma$.
5: **for** $\gamma = 1, \ldots, \Gamma$ **do**
6: Compute decision $\boldsymbol{x}_\gamma^t = \mathcal{O}(-\boldsymbol{\theta}^t, \boldsymbol{\omega}_\gamma^t)$, by calling the oracle $\mathcal{O}$ at each period.
7: **end for**
8: Compute gradient $\boldsymbol{z}^t = \nabla_{\boldsymbol{\theta}}[g^t(\boldsymbol{\theta}^t)]$, where $g^t(\boldsymbol{\theta}) := \phi^*(\boldsymbol{\theta}) + (-\boldsymbol{\theta})^\top \left[\frac{1}{\Gamma} \sum_{\gamma=1}^{\Gamma} \boldsymbol{f}(\boldsymbol{x}_\gamma^t, \boldsymbol{\omega}_\gamma^t)\right]$.
9: Compute weight vector $\boldsymbol{\theta}^{t+1} = \operatorname{argmin}_{\boldsymbol{\theta} \in D} \left\{\eta \cdot \left[\sum_{q=1}^{t} \boldsymbol{z}^{q\top} \boldsymbol{\theta}\right] + \Lambda(\boldsymbol{\theta})\right\}$.
10: **end for**
11: **Output:** The collection of weight vectors $\Theta = \{\boldsymbol{\theta}^t\}_{t=1}^T$.

As stated, the DM needs to make decisions $\boldsymbol{x}_\gamma$'s based on the observed scenarios $\boldsymbol{\omega}_\gamma \sim \Xi_\gamma$ on the fly. Same as Algorithm 2, the refined online algorithm (Algorithm 5) also involves the randomization algorithm to sample the weight vector $\boldsymbol{\theta}_\gamma$ from Θ and call the oracle $\mathcal{O}(-\boldsymbol{\theta}_\gamma, \cdot)$ to make the decision at each period γ.

Algorithm 5 (Refined) Online algorithm, to be run during the online process.

1: **Input:** A finite collection of weight vectors Θ from Algorithm 4.
2: **for** $\gamma = 1, \ldots, \Gamma$ **do**
3: Sample a weight vector $\boldsymbol{\theta}_\gamma$ uniformly at random from Θ.
4: Observe the γ-period scenario $\boldsymbol{\omega}_\gamma$.
5: Select decision $\boldsymbol{x}_\gamma$, where $\boldsymbol{x}_\gamma = \mathcal{O}(-\boldsymbol{\theta}_\gamma, \boldsymbol{\omega}_\gamma)$, by calling the optimization oracle $\mathcal{O}$.
6: **end for**

Next, we demonstrate the performance guarantee of this refined O2O algorithm on $\mathrm{DP}_1(\phi, \mathbb{R}^K)$ in the form of regret bound. Different from previous analysis, we investigate the performance of our

revised O2O algorithm in terms of the expected average value over the entire planning horizon. In this way, we revise the regret of an algorithm as

$$\mathrm{Reg}_3(\Gamma, T) := \mathbb{E}[\mathrm{opt}(\mathrm{UB}(\phi, S))] - \phi\left(\mathbb{E}\left[\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma} \boldsymbol{f}(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma)\right]\right),$$

which captures the difference between the offline benchmark and the algorithm's objective. Given the same algorithm, we note that $\phi\left(\mathbb{E}\left[\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma} \boldsymbol{f}(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma)\right]\right) \geq \mathbb{E}\left[\phi\left(\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma} \boldsymbol{f}(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma)\right)\right]$, which is clearly true by Jensen's inequality. Therefore, we have $\mathrm{Reg}_3(\Gamma, T) \leq \mathbb{E}\left[\mathrm{Reg}_1(\Gamma, T)\right]$. In this sense, we remark that the performance guarantee using $\mathrm{Reg}_3(\Gamma, T)$ is relatively "weaker" than using $\mathrm{Reg}_1(\Gamma, T)$.

While the scenarios across different periods (within each sample t) could be correlated, the sequences $\boldsymbol{\omega}^t_{1:\Gamma}$ are independently generated for different samples. Therefore, we apply the concentration equality on the average reward vector $\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma} \boldsymbol{f}(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma)$, instead of the vector $\boldsymbol{f}(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma)$ at a single period. This modification is crucial to derive the performance guarantee as follows.

THEOREM 3. *Assume the access to a forecast model and an optimization oracle, according to Assumptions 2, 3. Consider the application of Algorithm 5 to the online problem* $(DP_1(\phi, \mathbb{R}^K))$*, where the input* Θ *is generated by Algorithm 4. With probability at least* $1 - O(\delta)$*, we have*

$$\mathit{Reg}_3(\Gamma, T) = \tilde{O}\left(L\,\|\mathbf{1}_K\|\left(\frac{\max\{\lambda, 1\}}{\sqrt{T}}\right)\right).$$

The proof of Theorem 3 is similar to Theorem 1. We also bound the generalization error incurred by employing the refined sampling procedure in the offline algorithm and randomization algorithm in the online algorithm. Under the refined O2O algorithm, we assert that

$$\phi\left(\mathbb{E}\left[\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma} \boldsymbol{f}(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma)\right]\right) \geq \phi\left(\frac{1}{T}\sum_{t=1}^{T}\left[\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma} \boldsymbol{f}(\boldsymbol{x}^t_\gamma, \boldsymbol{\omega}^t_\gamma)\right]\right) - \underbrace{L\,\|\mathbf{1}_K\|\sqrt{\frac{2\log(4K/\delta)}{T}}}_{\S}, \tag{7}$$

(w.p. $1 - \delta/2$)

where the term (§) concerns an execution of the online algorithm, conditioned on the output $\Theta = \{\boldsymbol{\theta}^t\}_{t=1}^T$ by the offline algorithm are i.i.d., and then they are uniformly drawn in the online algorithm. Therefore, we have

$$\mathbb{E}\left[\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma} \boldsymbol{f}(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma) \;\middle|\; \Theta\right] = \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma} \boldsymbol{f}(\boldsymbol{x}^t_\gamma, \boldsymbol{\omega}^t_\gamma)\right].$$

In this way, we prove the desired bound in (§) using Azuma-Hoeffding inequality.

Next, similar to Equation (6), we claim that

$$\phi\left(\frac{1}{T}\sum_{t=1}^{T}\left[\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma} \boldsymbol{f}(\boldsymbol{x}^t_\gamma, \boldsymbol{\omega}^t_\gamma)\right]\right) \geq \mathbb{E}[\mathrm{opt}(\mathrm{UB}(\phi, \mathbb{R}^K))] - L\|\mathbf{1}_K\|\left[\frac{4\lambda + \sqrt{2\log(4K/\delta)}}{\sqrt{T}}\right]. \tag{8}$$

(w.p. $1 - \delta/2$)

Note that the sequences $\boldsymbol{\omega}_{1:\Gamma}^t$ are independently generated for different samples $t = 1, \ldots, T$, we refine the series of functions $\{g_t\}_{t=1}^T$ as $g_t(\boldsymbol{\theta}) = \phi^*(\boldsymbol{\theta}) - \boldsymbol{\theta}^\top \left[\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma} \boldsymbol{f}(\boldsymbol{x}_\gamma^t, \boldsymbol{\omega}_\gamma^t)\right]$ to utilize the OMD method in the offline algorithm.

Altogether, combining Equation (7) and (8), Theorem 3 is proved. ■

5. Applications in Supply Chain Management

The O2O algorithm can be used to address a wide spectrum of operations management problems. In the rest of this paper, we apply this framework to deal with several fundamental resource allocation challenges in supply chain management. In Section 5.1, we examine the performance of our online policy in a capacity pooling system, and demonstrate the value of incorporating distributional information in the design of allocation policy to deal with non-stationary demand arrivals. We describe a specific implementation of the O2O algorithm here to illustrate the practicability of our algorithmic framework. In Section 5.2, we study the online order fulfillment problem in a distribution network, and show how our approach can incorporate the trade-offs of both inventory holding and transportation cost in the planning process. In addition to studying the order fulfillment policy, we show that this O2O algorithm could be used to guide the design of flexible networks to balance the trade-offs between demand fulfillment and shipping cost. Using a case study on Amazon China (cf. Xu et al. 2020), we demonstrate that our algorithm could Pareto outperform the state-of-the-art solutions.

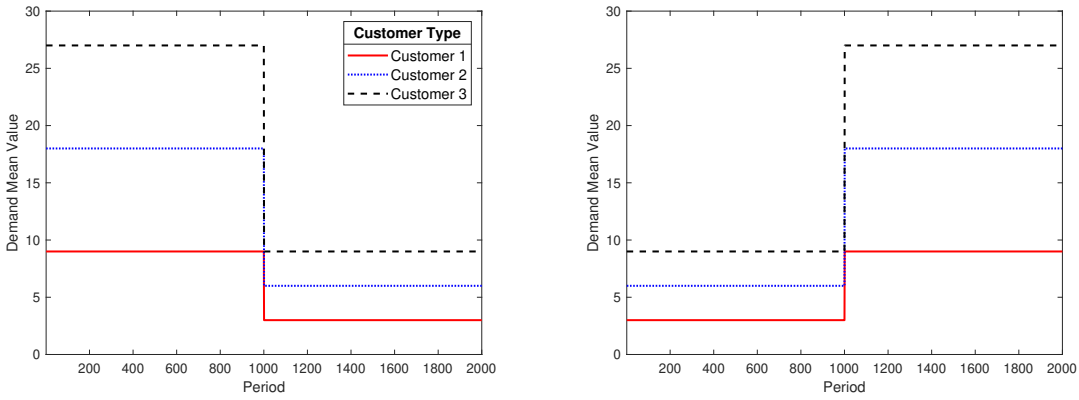
5.1. Resource Allocation in Capacity Pooling System

Capacity pooling is a common strategy used in practice to serve multiple demand segments with a common pool of resource (e.g., Alptekindğlu et al. 2013, Zhong et al. 2018, Jiang et al. 2019). For instance, a supplier stocking goods for delivery to multiple retailers may face the challenge of meeting the KPIs promised in its service-level agreements with retailers. In this context, the service-level agreement is a commitment by a supplier to achieve a minimum fill rate over a specified time horizon. This problem is often studied in the literature assuming that the demands faced are iid, to facilitate the analysis of the fill rates attained. This ignores the more practical challenge in the problem when the demands faced by the retailers are non-stationary across time. We show that our O2O algorithm is able to deal with this challenge, while the state-of-art resource allocation algorithms developed under the stationary assumption fail to deliver satisfactory performance in the non-stationary environment.

To make the discussion concrete, we consider a capacity pooling system in which a retailer supplies a common product to $K = 3$ customers over $\Gamma = 2000$ periods. The retailer serves these customers using a fixed amount of resources c at each period. Customers are faced with non-stationary demands and we assume that the demand $\boldsymbol{d}_\gamma$ follows heterogeneous Poisson distributions with mean values $\boldsymbol{\lambda}_\gamma := \boldsymbol{\lambda}_0 \times e_\gamma$, where the parameter e_γ represents the demand seasonality at period $\gamma = \{1, \ldots, \Gamma\}$ and $\boldsymbol{\lambda}_0 = (3, 6, 9)$ denotes the baseline of demand mean values. In the numerical experiments, we

consider two cases: (1) we let $e_\gamma = 3$ for $\gamma \in [1, \Gamma/2]$, and $e_\gamma = 1$ for $\gamma \in [\Gamma/2+1, \Gamma]$; (2) we let $e_\gamma = 1$ for $\gamma \in [1, \Gamma/2]$, and $e_\gamma = 3$ for $\gamma \in [\Gamma/2+1, \Gamma]$. The demand mean values in the two cases are plotted in Figure 4(a) and (b), respectively. In addition, customers require differentiated fill rate targets $\boldsymbol{\beta} = (0.85, 0.90, 0.95)$, i.e., the expected amount of resource allocated to customer k over the whole horizon should be at least $\beta_k \mu_k$, where $\mu_k := \frac{1}{\Gamma} \sum_{\gamma=1}^{\Gamma} \lambda_{k,\gamma}$ represents the demand mean value across the entire planning horizon.

Figure 4 Demand mean values in the non-stationary environments.



(a) Case 1 (b) Case 2

Following the description in Section 3.3, we formulate the fill-rate constrained resource allocation problem in the capacity pooling system (CPS) as a feasibility problem:

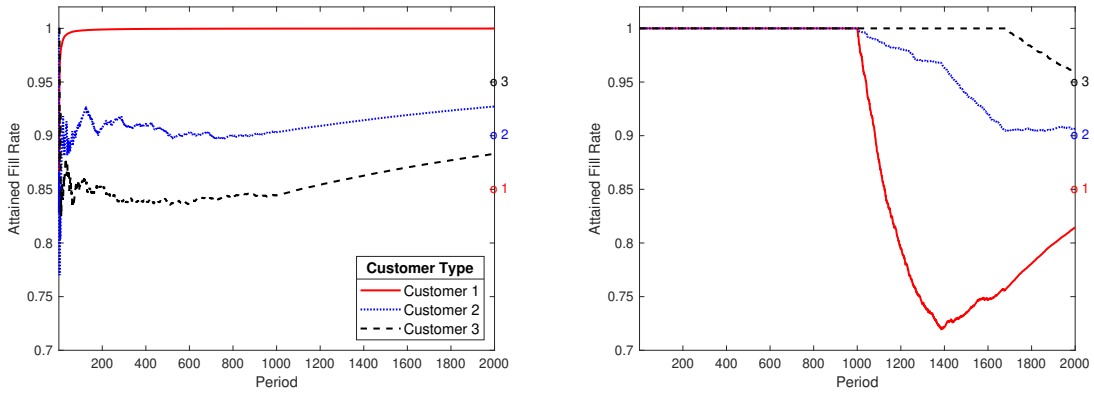
$$
\begin{aligned}
\text{(CPS)} \quad \max \; & 0 \\
\text{s.t.} \; & \frac{1}{\Gamma} \sum_{\gamma=1}^{\Gamma} x_{k,\gamma} \geq \beta_k \mu_k, \; k = 1, \ldots, K, \text{ almost surely}, \\
& \sum_{k=1}^{K} x_{k,\gamma} \leq c, \; \gamma = 1, \ldots, \Gamma, \\
& 0 \leq x_{k,\gamma} \leq d_{k,\gamma}, \; k = 1, \ldots, K, \; \gamma = 1, \ldots, \Gamma.
\end{aligned}
$$

where $x_{k,\gamma}$ denotes the amount of resource allocated to customer k at period γ. The first set of constraints forces the fill rate requirement for each and every customer to be met almost surely at the end of planning horizon. The second and third sets of constraints require respectively that the non-negative allocation quantity to customer k cannot exceed the realized demand $d_{k,\gamma}$ and the total allocation quantity at each period γ cannot exceed the capacity c.

Analogous to the setting in Zhong et al. (2018), we consider an identical capacity profile at each period and use sampling average approximation (SAA) method to obtain the minimal capacity level

c^* such that all the fill rate targets are attainable in the offline setting, i.e., to ensure the feasibility of problem (CPS). Note that Zhong et al. (2018) demonstrated that their debt-based online allocation policy, together with the minimal capacity level c^*, are able to attain all the required fill rate targets in a stationary stochastic environment. Nevertheless, Figure 5 plots the glide path of attained fill rate (i.e., $\sum_{\gamma=1}^{s} x_{k,\gamma} / \sum_{\gamma=1}^{s} d_{k,\gamma}$ for $s = 1, \ldots, \Gamma$) over the entire planning horizon, and it shows clearly that the debt-based policy cannot meet the fill rate requirements in both cases.

Figure 5 Attained fill rate over time under the online policy by Zhong et al. (2018).



(a) Case 1 (b) Case 2

Note. The fill rate targets are marked on the right-hand-side of each figure.

In fact, problem (CPS) could be re-formulated as the unconstrained problem $\mathrm{DP}_1(\phi, \mathbb{R}^K)$, which minimizes the Euclidean distance from the attained fill rate vector to the corresponding target. To see this, we let $f_k(\boldsymbol{x}_\gamma, \boldsymbol{d}_\gamma) := \beta_k \mu_k - x_k^t$ denote the k^{th} outcome function, and $\phi(\boldsymbol{w}) := -\frac{1}{2}\|\boldsymbol{w}\|_2^2$ the global reward function. To make the discussion clearer, we specialize the O2O algorithm for this specific application, using the gradient descent method under $\|\cdot\|_2$-Lipschitz Continuity as provided in Example 2. In Algorithm 6 (offline algorithm), the weight vector θ_k^{t+1} could be normalized by the time-dependent constant Δ_t, without changing the decision $\boldsymbol{x}^{t+1} = \mathcal{O}^{\mathrm{CPS}}(-\boldsymbol{\theta}^{t+1}, \boldsymbol{d}^{t+1})$. Therefore, the decision maker only needs to specify the number of iterations T to run this offline algorithm, while other parameters (e.g., learning rate η) do not affect the allocation quality.

Figure 6 depicts the attained fill rate under our O2O algorithm. It is straightforward to see that all the fill rate requirements are met at the end of planning horizon in both cases. Compared with Figure 5(a), in which customer 1 is served with higher priority due to the (relatively) higher debt, Figure 6(a) shows that customer 1 should be served with lower priority due to the smaller fill rate requirement if future demand samples are incorporated into the allocation policy. In case 2, almost all the demands could be satisfied at the first half of the planning horizon. With the demands increasing

Algorithm 6 Offline algorithm for problem (CPS).

1: INPUT: Objective function $\phi(\cdot) = -\|\cdot\|_2^2$, Number of iterations $T = 10^4$.

2: INITIALIZE: Initial weight vector $\boldsymbol{\theta}^1 = (1, 1, 1)$.

3: **for** $t = 1, \ldots, T$ **do**

4: Sample a period index γ^t uniformly at random from $\{1, \ldots, \Gamma\}$.

5: Sample a random demand $\boldsymbol{d}^t$ according to the distribution $\boldsymbol{\lambda}_{\gamma^t}$.

6: Compute decision $\boldsymbol{x}^t = \mathcal{O}^{\text{CPS}}(-\boldsymbol{\theta}^t, \boldsymbol{d}^t)$, by calling the optimization oracle $\mathcal{O}^{\text{CPS}}$:

$$
\begin{aligned}
(\mathcal{O}^{\text{CPS}}) \max_{\boldsymbol{x}^t} \ & \sum_{k=1}^{K} -\theta_k^t(\beta_k \mu_k - x_k^t) \\
\text{s.t.} \ & \sum_{k=1}^{K} x_k^t \le c^*, \\
& 0 \le x_k^t \le d_k^t, \ k = 1, \ldots, K.
\end{aligned}
$$

7: Compute the gradient $z_k^t := -(\beta_k \mu_k - x_k^t)$ for $k = 1, \ldots, K$.

8: Following Equation (3), update the weight $\theta_k^{t+1} := -\sum_{q=1}^{t} (\beta_k \mu_k - x_k^q) \times \Delta_t$, for $k = 1, \ldots, K$, where $\Delta_t := -\frac{L}{\max\left\{\sqrt{8KT}, \|-(\sum_{q=1}^{t} \beta_k \mu_k - x_k^q)\|_2\right\}}$.

9: **end for**

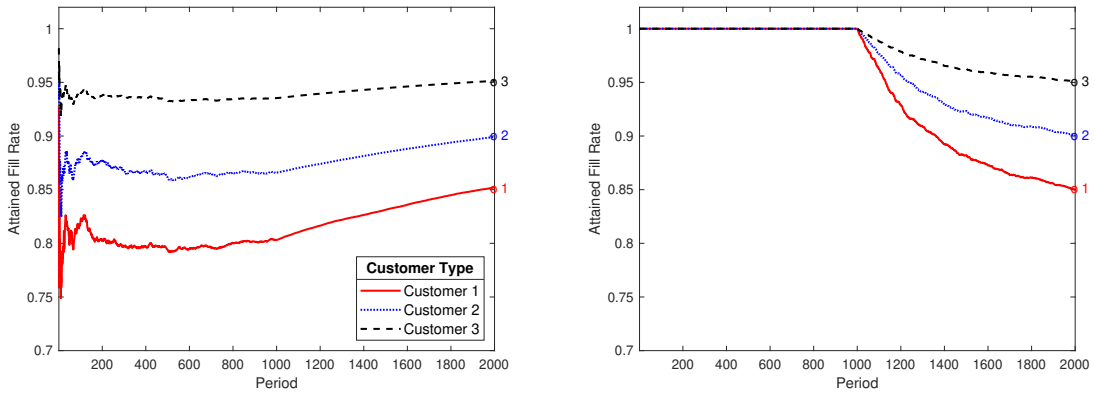
10: OUTPUT: The collection of weight vectors $\Theta = \{\boldsymbol{\theta}^t\}_{t=1}^T$.

above the capacity level, the attained fill rates decline at the second half of the planning horizon. Notably, the fill rate of customer 1 declines more sharply in Figure 5(b), compared with the pattern in Figure 6(b). The reason is that customer 1 accumulates consecutively smaller debt under the priority policy by Zhong et al. (2018) at the first half of the planning horizon, and hence gains lower priority to be served during the remaining time. Differently, the increasing demand trend "guides" the O2O algorithm to balance the resource allocation among three customers under limited capacity so as to fulfill the fill rate requirements. This validates the value of incorporating the offline demand samples into the design of online allocation strategy in non-stationary environments.

5.2. Resource Allocation in Order Fulfillment Network

The boom in E-commerce has given rise to the need for efficient and smart order fulfillment systems. To mitigate the impact of demand uncertainty, multi-sourcing strategy is often used in the order fulfillment network, so that each supply could be used to serve the demand requirements from multiple geographical regions (Jordan and Graves 1995). While this practice can improve the quality of resource allocation (i.e., reduce the mis-match between supply and demand), it may inadevertently lead to increase in operational cost and shipping cost, since goods are no longer sourced from the nearest fulfillment centers. The performance of network design with limited flexibility has been theoretically justified when the resource allocation decisions are made either offline (e.g., Chou et al. 2010, Wang and Zhang 2015) or online (e.g., Lyu et al. 2019, Asadpour et al. 2020, Xu et al. 2020).

Figure 6 Attained fill rate over time under our O2O algorithm.



(a) Case 1 (b) Case 2

Note. The fill rate targets are marked on the right-hand-side of each figure.

In general, the performance of resource allocation is examined based on the expected volume of fulfilled demand, and this is also the primary principle to design the network structure with limited flexibility. We note that, if the network design is constrained by KPI in total shipping cost, then the planner needs to carefully balance the trade-off between the two objectives – maximizing the amount of demand fulfilled and minimizing the total shipping cost – in the design of the network. In this section, we show that our O2O planning framework can be used to address this complicated challenge. We explicitly incorporate the shipping cost into the network design and resource allocation problem. In our numerical study on a case reproduced from Xu et al. (2020), our network design solution is shown to outperform the state-of-the-art solution from Xu et al. (2020) in terms of both demand fulfillled and shipping cost *simultaneously*.

We consider a bipartite order fulfillment network $\mathcal{G}$, in which on one side is a set $\mathcal{J}$ of demand regions, whereas on the other side is a set $\mathcal{I}$ of supply nodes. We also use $\mathcal{G}$ to represent the set of links in the network. The shipping cost associated with the link between demand region j and supply node i is denoted as $l_{i,j}$. For the ease of exposition, we formulate the resource allocation problem in a single-period setting and assume the random demands are stationary across time periods. We discussed the computational performance on the non-stationary case at the end of this section.

The demand at region j is denoted as d_j, for $\forall j \in \mathcal{J}$, and the capacity level at supply node i is denoted as c_i, for $\forall i \in \mathcal{I}$. The support of all the realized demand is denoted as Ω. We represent the resource allocation quantity from supply node i to demand region j as $x_{i,j}(\boldsymbol{d})$, which depends on the demand realizations $\boldsymbol{d} \in \Omega$. Furthermore, the feasible region for the resource allocation problem is

characterized by $\mathcal{X}(\boldsymbol{d})$:

$$\mathcal{X}(\boldsymbol{d}) := \left\{ \boldsymbol{x}(\boldsymbol{d}) \in \mathbb{R}^{|\mathcal{I}|\times|\mathcal{J}|} \;\middle|\; \begin{array}{ll} \sum_{i\in\mathcal{I}} x_{i,j}(\boldsymbol{d}) \le d_j, & \forall j \in \mathcal{J} \\ \sum_{j\in\mathcal{J}} x_{i,j}(\boldsymbol{d}) \le c_i, & \forall i \in \mathcal{I} \\ x_{i,j}(\boldsymbol{d}) \ge 0, & \forall (i,j) \in \mathcal{G} \\ x_{i,j}(\boldsymbol{d}) = 0, & \forall (i,j) \notin \mathcal{G} \end{array} \right\},$$

where the first and second sets of constraints require that the total amount allocated to demand region j should not exceed the required amount d_j, and the total allocation quantity from supply node i should be smaller than the available capacity level c_i. The third and forth set of constraints indicate the non-negative resource allocation requirement.

To maximize the demand fulfillment, we introduce a fill rate target $\beta_j \in (0,1)$ to ensure that the expected amount of resource distributed to region j should be at least $\beta_j\mu_j$, where μ_j denotes the expected demand at region j. The fill rate profile $(\beta_1, \beta_2, \ldots, \beta_N)$ is determined (e.g., using the SAA approach) such that the total demand fulfillment $\sum_{j=1}^{N} \beta_j\mu_j$ is maximized. In this way, the demand fulfillment maximization objective can be guaranteed as long as the fill rate targets are met (Lyu et al. 2019). Combining with the objective to minimize the total shipping cost, we formulate the supply chain management problem as follows:

$$\begin{aligned} \text{(OFN)} \quad \min_{\boldsymbol{x}(\boldsymbol{d})} \; & \mathbf{E}\left[\sum_{(i,j)\in\mathcal{G}} l_{i,j} x_{i,j}(\boldsymbol{d})\right] \\ \text{s.t.} \; & \mathbf{E}\left[\sum_{i\in\mathcal{I}} x_{i,j}(\boldsymbol{d})\right] \ge \beta_j\mu_j, \; \forall j \in \mathcal{J} \\ & \boldsymbol{x}(\boldsymbol{d}) \in \mathcal{X}(\boldsymbol{d}), \; \forall \boldsymbol{d} \in \Omega \end{aligned}$$

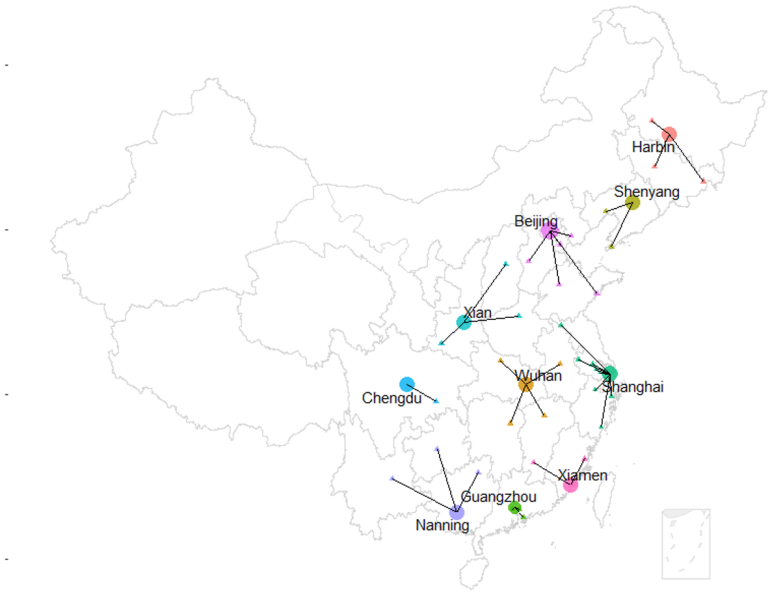
where the first set of constraints forces the fill-rate requirement to be met for each and every demand region. Clearly, this problem can be modelled by our online planning framework. By implementing our O2O algorithm, the allocation solution, denoted by $x_{i,j}^{O2O}(\boldsymbol{d})$, is associated with the minimal shipping cost and maximal demand fulfillment. With a slight abuse of notation, we denote the expected allocation quantity from supply i to demand region j as $X_{i,j}^{O2O} := \mathbf{E}\left[x_{i,j}^{O2O}(\boldsymbol{d})\right]$. We rank $\{X_{i,j}^{O2O}\}$ in the descending order so that we can select the *Top M* links to control the operational cost if it is costly to configure more than M links. We denote the network structure under this simple principle as *O2O Network*.

To this end, we note that Shi et al. (2019) introduced the notion of Generalized Chaining Gap (GCG), and showed that GCG could be used to measure the effective flexibility structures in a make-to-order system. Along this direction, Xu et al. (2020) showed that GCG could also indicate

the system performance (in terms of expected number of lost sales) in an online order fulfillment setting. They demonstrated that GCG network with $|\mathcal{I}|+|\mathcal{J}|-1$ links could perform very well, and developed an effective method to design such GCG networks. Furthermore, they revised the design of GCG network to incorporate the shipping cost.[5] We note that the performance of their proposed network structures depends crucially on a well calibrated capacity profiles, and this in turn limits its use in practice. Using the Amazon China case reproduced from Xu et al. (2020), we numerically show that our O2O network structures can overcome this limitation, and its performance are more "robust" no matter how the capacity are configured.

Analogous to the description of Amazon China case in Xu et al. (2020), we also consider 10 order fulfillment centers (supply nodes) and 44 main demand centers (demand regions). As shown in Figure 7, Amazon China constructed a dedicated order fulfillment network, i.e., each demand center is served by one primary fulfillment center. In total, there are 44 links in this network.

Figure 7 Dedicated order fulfillment network of Amazon China.



Note. Order fulfillment centers are represented by (big) dots, and demand centers are (small) triangles. The links indicate the fulfillment network structure.

Table 1 summarizes the fulfillment centers, demand centers, and demand parameters. We label each fulfillment center by a unique ID. The demand centers connected to the same fulfillment center are clustered into the same region. This is indeed an unbalance network, and the number of demand

[5] We remark that the revised GCG network could be obtained by solving a minimal spanning tree problem in the case of $|\mathcal{I}|+|\mathcal{J}|-1$ links. However, this revised GCG approach is computationally non-trivial for general cases.

centers to serve by each fulfillment center varies from 2 to 9. Furthermore, we consider a stochastic batch demand setting, and the random demand of center j follows a (truncated) normal distribution $d_j \sim \max(0, \text{Normal}(\mu_j, (\mu_j/3)^2))$, where μ_j is proportional to the arrival rate p_j (scaled by a factor 100) as described in Xu et al. (2020). We note that this modification does not change the GCG network structures derived from Xu et al. (2020), and hence we can use directly their network solutions for comparison.

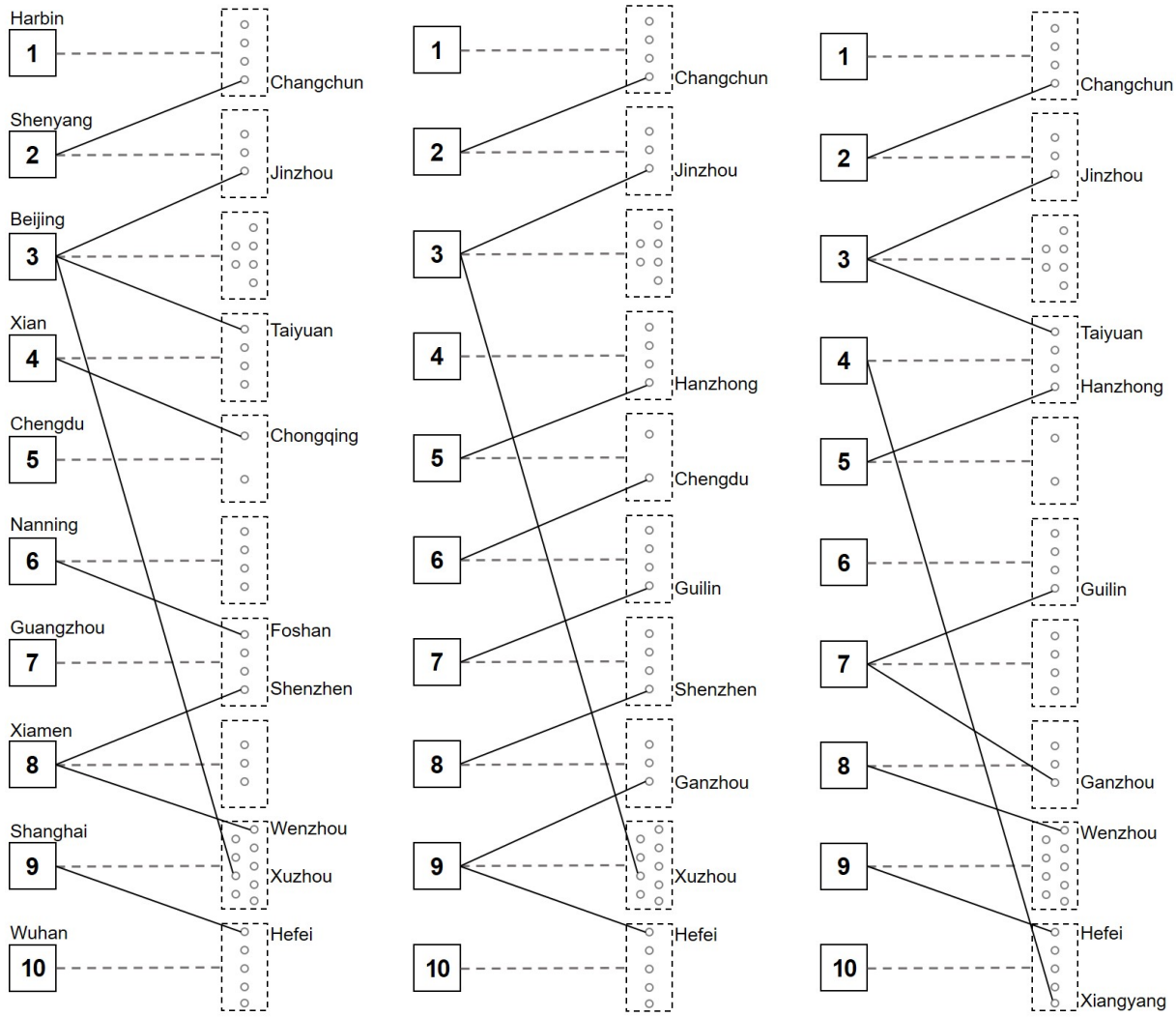
Table 1 Summary of the order fulfillment network.

ID	Fulfillment Center	Demand Center	Demand Mean ($\boldsymbol{\mu}$)
1	Harbin	Harbin, Daqing, Changchun, Yanbian	$(1.9, 1.9, 1.3, 1.3)$
2	Shenyang	Shenyang, Jinzhou, Dalian	$(1.7, 2.2, 2.2)$
3	Beijing	Beijing, Tianjin, Tangshan, Shijiazhuang, Jinan, Qingdao	$(5.7, 1.5, 1.5, 2.9, 2.0, 1.9)$
4	Xian	Xian, Hanzhong, Zhengzhou, Taiyuan	$(0.9, 1.8, 1.9, 1.9)$
5	Chengdu	Chengdu, Chongqing	$(2.8, 3.6)$
6	Nanning	Nanning, Kunming, Guilin, Guiyang	$(0.6, 1.0, 1.0, 1.1)$
7	Guangzhou	Guangzhou, Foshan, Shenzhen, Dongguan	$(4.5, 4.5, 4.5, 2.8)$
8	Xiamen	Xiamen, Fuzhou, Ganzhou	$(0.9, 0.7, 1.9)$
9	Shanghai	Shanghai, Suzhou, Hangzhou, Ningbo, Wenzhou, Changzhou, Wuxi, Nanjing, Xuzhou	$(2.6, 2.6, 2.6, 2.6, 2.6, 1.9, 2.6, 2.6, 5.7)$
10	Wuhan	Wuhan, Nanchang, Changsha, Hefei, Xiangyang	$(2.9, 2.9, 2.1, 0.9, 1.2)$

W.L.O.G., we consider a limited flexibility system with $|\mathcal{I}| + |\mathcal{J}| - 1$ links. In this setting, Xu et al. (2020) constructed a GCG network to maximize the demand fulfillment (i.e., minimize the lost sales). The GCG network structure is depicted in Figure 8(b). Following the same manner in Xu et al. (2020), we allocate μ_j amount of resources evenly to the fulfillment centers that are connected to demand center j. This gives rise to the capacity configuration $\boldsymbol{c}^{CGC}$ that will be used in the following numerical experiments. In order to incorporate the objective of shipping distance minimization, Xu et al. (2020) revised the GCG network structure. We reproduce the revised GCG network structure in Figure 8(c). Intuitively, the GCG network provides higher priority (by adding additional links) to serve the demand centers with larger demands, while the revised GCG network prioritizes the links with shorter shipping distance. Next, we solve problem (OFN) in a fully-flexible order fulfillment network using the same capacity $\boldsymbol{c}^{CGC}$, and select the top $|\mathcal{I}| + |\mathcal{J}| - 1$ links to construct our O2O network structure, which is described in Figure 8(a). Interestingly, the O2O network also presents a chaining structure even though such constraints are not enforced into our approach.

For the ease of exposition, we examine the performance of different network structures under the same capacity profile $\boldsymbol{c}^{CGC}$ and the same O2O allocation policy. More concretely, we solve problem (OFN) in a fully-flexible system to obtain a collection of weight vectors at the offline stage. Next, we solve the online problem (with the same weight vectors) under different network structures. The proportion of lost sales and average shipping distance under the three structures are provided in

Figure 8 **Different order fulfillment network structures.**



(a) O2O Network (b) GCG Network (c) Revised GCG Network

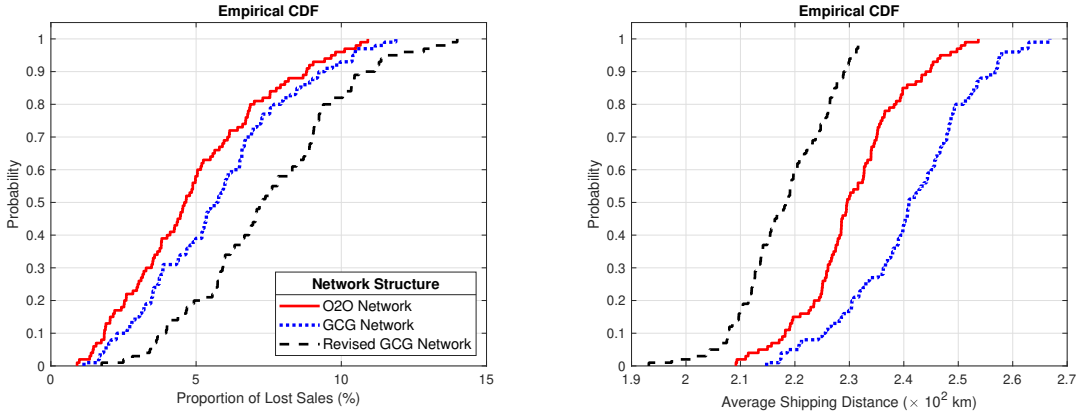
Table 2. The GCG network performs the best performance in terms of the lost sales, but results in the highest shipping distance. On the contrary, the revised GCG network attains the lowest shipping distance, but suffers from the largest proportion of lost sales. Our O2O network works well for both objectives. Compared to the GCG network, the O2O network suffers from an additional 0.07% drop in lost sales, but the O2O network reduces the average shipping distance by 16 km.

Table 2 **Performance comparison between different network structures.**

Performance	O2O network	GCG network	Revised GCG network
Proportion of Lost Sales (%)	3.92	3.85	5.46
Average Shipping Distance ($\times 10^2$ km)	2.28	2.44	2.23

Next, we compare the performance of different network structures when the capacity profiles are configured differently. We scale each capacity level c_i^{GCG} by a random factor, which is uniformly distributed in the interval $[0.75, 1.25]$. In this way, we generate 100 capacity profiles and evaluate the average performance. We re-solve problem (OFN) and obtain the O2O network structure in correspondence to each capacity profile. Since the demand profile does not change, the structures of both GCG network and revised GCG network remain the same under different capacity configurations. As shown in Figure 9, we plot the empirical CDF of lost sales and shipping distance under the three networks. Notably, our O2O network simultaneously outperforms the GCG network in terms of both demand fulfillment maximization and shipping distance minimization. This implies the robustness of our O2O solution. Although the revised GCG network achieves the lowest shipping distance, the proportion of lost sales is the highest, and in particularly it is 51.01% higher than our O2O solution.

Figure 9 **Performance comparison under different capacity configurations.**

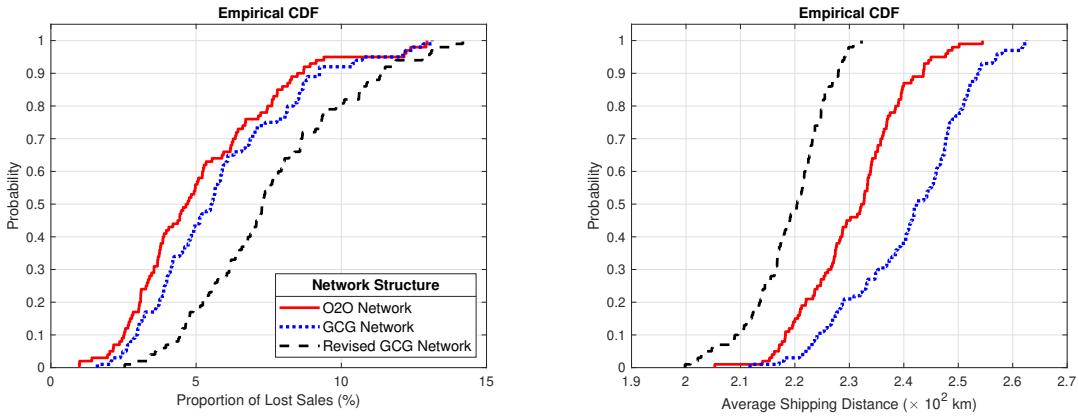


(a) Lost Sales

(b) Shipping Distance

In the end, we examine the performance of different network structures in a non-stationary stochastic environment. We consider a multi-period problem with planning horizon $\Gamma = 2000$. At each period $\gamma = 1, \ldots, \Gamma$, we set the demand mean as $\mu_{\gamma,j} = \mu_j \times \epsilon$ for demand center j, where ϵ is uniformly distributed in the interval $[0.75, 1.25]$. We also generate 100 capacity profiles to evaluate the average performance in the same way above. As shown in Figure 10, our O2O network also consistently outperforms the GCG network in terms of the lost sales and the average shipping distance the non-stationary setting.

Figure 10 **Performance comparison in non-stationary stochastic environments.**



(a) Lost Sales (b) Shipping Distance

6. Concluding Remarks

Leveraging on big data and analytics, machines and operations have become much smarter than before. The insights derived from data help the operations to better interact with the dynamic environments. In this paper, we study the interactions between models and data from the angle of online planning, and exploit the offline simulation in the design of online planning strategy.

We address a general class of online planning problems with concave objective functions and global feasibility constraints under non-stationary environments. Leveraging on the access to the probability distributions of the stochastic inputs to generate offline samples, we develop a generic solution oracle with near-optimal performance guarantee. This online planning framework can be appropriately used to solve a wide range of operations management problems, including supply chain management, urban logistics, revenue management, and ride-sharing problems. Interested readers may refer to Lyu (2019) for more applications of this approach. Furthermore, this and variant of the ideas presented here may lead to practical heuristic to solve large scale stochastic optimization problems, circumventing the curse of dimensionality in the traditional stochastic DP approach.

This work can be extended in many ways. We highlight one interesting direction to be further explored. Similar to most stochstic dynamic programing literatures, we assume the perfect knowledge on the distributional information $\{\Xi_\gamma\}_{\gamma=1}^{\Gamma}$ so that we can generate sufficient samples at the offline stage. A follow-up question is, *what if the offline samples are generated from an imperfect forecast model, instead of the actual distribution model*? This in turn depends on how we measure the forecast accuracy mathematically. To our knowledge, there is no universal way to examine the impact of forecast error on the performance of online planning policies. Here, we provide one possible way to recast the forecast model. With a slight abuse of notation, we denote the forecast model as $\{\widetilde{\Xi}_\gamma\}_{\gamma=1}^{\Gamma}$, where each $\widetilde{\Xi}_\gamma$ is the *scenario forecast probability distribution* for time period γ. In this way, the

forecast distribution $\widetilde{\Xi}_\gamma$ serves to approximate the actual stochastic process Ξ_γ. While $\Xi_\gamma, \widetilde{\Xi}_\gamma$ are generally different for each γ, we assume that they share the same support Ω: $\Pr[\Xi_\gamma \in \Omega] = \Pr[\widetilde{\Xi}_\gamma \in \Omega] = 1$. Furthermore, we denote $\tilde{\boldsymbol{\omega}}_\gamma$ as a sample from $\widetilde{\Xi}_\gamma$, and $\tilde{\boldsymbol{\omega}}_{1:\Gamma} := \{\tilde{\boldsymbol{\omega}}_\gamma\}_{\gamma=1}^{\Gamma}$ as the sequence of random scenarios. Consequently, we have $\tilde{\boldsymbol{\omega}}_{1:\Gamma} \sim \widetilde{\Xi}_{1:\Gamma} := \prod_{\gamma=1}^{\Gamma} \widetilde{\Xi}_\gamma$. We assume that the forecast model $\{\widetilde{\Xi}_\gamma\}_{\gamma=1}^{\Gamma}$ is accessible to the DM as the sample generating oracle, while the DM cannot approach to the actual model $\{\Xi_\gamma\}_{\gamma=1}^{\Gamma}$.

We propose to quantify the forecast error of the model $\{\widetilde{\Xi}_\gamma\}_{\gamma=1}^{\Gamma}$ as

$$\Psi := \sup_{\boldsymbol{x}} \frac{1}{\|\mathbf{1}_K\|} \mathrm{SF}(\boldsymbol{x}),$$

where $\mathrm{SF}(\boldsymbol{x}) := \left\| \frac{1}{\Gamma} \sum_{\gamma=1}^{\Gamma} \left\{ \mathbf{E}_{\boldsymbol{\omega}_{1:\Gamma} \sim \Xi_{1:\Gamma}} [\boldsymbol{f}(\boldsymbol{x}(\boldsymbol{\omega}_\gamma), \boldsymbol{\omega}_\gamma)] - \mathbf{E}_{\tilde{\boldsymbol{\omega}}_{1:\Gamma} \sim \widetilde{\Xi}_{1:\Gamma}} [\boldsymbol{f}(\boldsymbol{x}(\tilde{\boldsymbol{\omega}}_\gamma), \tilde{\boldsymbol{\omega}}_\gamma)] \right\} \right\|$. The supremum in the above equation is taken over all *decision rules*, where a decision rule $\boldsymbol{x} = \{\boldsymbol{x}(\boldsymbol{\omega})\}_{\boldsymbol{\omega} \in \Omega}$ maps a scenario $\boldsymbol{\omega}$ to a feasible decision $\boldsymbol{x} \in \mathcal{X}(\boldsymbol{\omega})$. In this way, the quantity $\mathrm{SF}(\boldsymbol{x})$ represents the shortfall associated with the decision rule $\boldsymbol{x}$, and the shortfall with $\boldsymbol{x}$ measures the distance between the average outcome with forecast model $\widetilde{\Xi}_{1:\Gamma}$ and actual model $\Xi_{1:\Gamma}$. Consequently, the forecast error Ψ naturally measures the discrepancy between the forecast and actual model. Note that Ψ is not known to the DM.

Next, we illustrate how to solve the problem $(\mathrm{DP}_1(\phi, \mathbb{R}^K))$ based on the revised forecast model, and the general problem $(\mathrm{DP}_1(\phi, S))$ can be also solved in a similar manner. Under the revised forecast model, we only need to revise the offline sampling procedure in the O2O algorithm, i.e., in Line 5 of Algorithm 1, we sample a random scenario $\tilde{\boldsymbol{\omega}}^t$ according to the distribution $\widetilde{\Xi}_{\gamma^t}$. It can be shown that, with this type of imperfect forecast, the optimality gap of the algorithm scales naturally with the discrepancy between actual and forecast models. More concretely, with probability at least $1 - O(\delta)$, we have

$$\mathrm{Reg}_1(\Gamma, T) = \tilde{O}\left(L \|\mathbf{1}_K\| \left(\frac{1}{\sqrt{\Gamma}} + \Psi + \frac{\max\{\lambda, 1\}}{\sqrt{T}} \right) \right).$$

Since this imperfect forecast model is not the focus of this paper, we refer interested readers to the proof in Cheung et al. (2019). It largely follows the analysis in Appendices A.2, A.3. Notably, in the case when the DM only knows that the distributions $\{\Xi_\gamma\}_{\gamma=1}^{\Gamma}$ are randomly permuted over the planning horizon, but does not know the specific order at each period (the forecast error $\Psi = 0$ in this case), our proposed algorithm can also achieve the same performance guarantee.

References

Agrawal, Shipra, Vashist Avadhanula, Vineet Goyal, Assaf Zeevi. 2016. A near-optimal exploration-exploitation approach for assortment selection. *Proceedings of the 2016 ACM Conference on Economics and Computation*. 599–600.

Agrawal, Shipra, Nikhil R Devanur. 2015. Fast algorithms for online stochastic convex programming. *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 1405–1424.

Agrawal, Shipra, Zizhuo Wang, Yinyu Ye. 2014. A dynamic near-optimal algorithm for online linear programming. *Operations Research* **62**(4) 876–890.

Alaei, Saeed, MohammadTaghi Hajiaghayi, Vahid Liaghat. 2012. Online prophet-inequality matching with applications to ad allocation. *Proceedings of the 13th ACM Conference on Electronic Commerce*. 18–35.

Alptekindğlu, Aydın, Arunava Banerjee, Mabel Anand Paul, Nikhil Jain. 2013. Inventory pooling to deliver differentiated service. *Manufacturing & Service Operations Management* **15**(1) 33–44.

Asadpour, Arash, Xuan Wang, Jiawei Zhang. 2020. Online resource allocation with limited flexibility. *Management Science* **66**(2) 642–666.

Balseiro, Santiago R, Jon Feldman, Vahab Mirrokni, Shan Muthukrishnan. 2014. Yield optimization of display advertising with ad exchange. *Management Science* **60**(12) 2886–2907.

Besbes, Omar, Yonatan Gur, Assaf Zeevi. 2015. Non-stationary stochastic optimization. *Operations research* **63**(5) 1227–1244.

Bruss, F Thomas. 1984. A unified approach to a class of best choice problems with an unknown number of options. *The Annals of Probability* 882–889.

Bubeck, Sébastien, et al. 2015. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning* **8**(3-4) 231–357.

Bumpensanti, Pornpawee, He Wang. 2020. A re-solving heuristic with uniformly bounded loss for network revenue management. *Management Science* **66**(7) 2993–3009.

Chen, Xi, Yining Wang, Yuxiang Wang. 2019. Non-stationary stochastic optimization under l_ {p, q}-variation measures. *Operations Research* **67**(6) 1752–1765.

Cheung, Wang Chi, Guodong Lyu, Chung Piaw Teo, Hai Wang. 2019. Online planning with offline forecasting. *SSRN* URL `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3395781`.

Cheung, Wang Chi, David Simchi-Levi, Ruihao Zhu. 2020. Hedging the drift: Learning to optimize under non-stationarity. *Available at arXiv:1903.01461* .

Chou, Mabel C, Geoffrey A Chua, Chung-Piaw Teo, Huan Zheng. 2010. Design for process flexibility: Efficiency of the long chain and sparse structure. *Operations research* **58**(1) 43–58.

Esfandiari, Hossein, Nitish Korula, Vahab Mirrokni. 2015. Online allocation with traffic spikes: Mixing adversarial and stochastic models. *Proceedings of the Sixteenth ACM Conference on Economics and Computation*. 169–186.

Hardt, Moritz, Ben Recht, Yoram Singer. 2016. Train faster, generalize better: Stability of stochastic gradient descent. *Proceedings of the 33rd International Conference on Machine Learning*. 1225–1234.

Hong, Jeff L., Guangxin Jiang. 2019. Offline simulation online application: A new framework of simulation-based decision making. *Asia-Pacific Journal of Operational Research* **36**(6).

Huh, Woonghee Tim, Paat Rusmevichientong. 2014. Online sequential optimization with biased gradients: theory and applications to censored demand. *INFORMS Journal on Computing* **26**(1) 150–159.

Jiang, Guangxin, Jeff L. Hong, Barry L. Nelson. 2020. Online risk monitoring using offline simulation. *INFORMS Journal on Computing* **32**(2) 356–375.

Jiang, Jiashuo, Shixin Wang, Jiawei Zhang. 2019. Achieving high individual service-levels without safety stock? optimal rationing policy of pooled resources. *Available at SSRN 3385089* 1–38.

Jordan, William C, Stephen C Graves. 1995. Principles on the benefits of manufacturing process flexibility. *Management Science* **41**(4) 577–594.

Li, Xiaolong, Ying Rong, Renyu Philip Zhang, Huan Zheng. 2020. Personalized sales targets with customer choices. *Available at SSRN 3538755* .

Lyu, Guodong. 2019. Online resource allocation: Theory and applications. Ph.D. thesis.

Lyu, Guodong, Wang-Chi Cheung, Mabel C Chou, Chung-Piaw Teo, Zhichao Zheng, Yuanguang Zhong. 2019. Capacity allocation in flexible production networks: Theory and applications. *Management Science* **65**(11) 5091–5109.

Ma, Yuhang, Paat Rusmevichientong, Mika Sumida, Huseyin Topaloglu. 2020. An approximation algorithm for network revenue management under nonstationary arrivals. *Operations Research* **68**(3) 834–855.

Nemirovsky, A. S., D. B. Yudin. 1983. Problem complexity and method efficiency in optimization. *Wiley Interscience Series in discrete mathematics* .

Shalev-Shwartz, Shai, Ohad Shamir, Nathan Srebro, Karthik Sridharan. 2009. Stochastic convex optimization. *Proceedings of the 22nd Annual Conference on Learning Theory (COLT)*.

Shalev-Shwartz, Shai, et al. 2012. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning* **4**(2) 107–194.

Shi, Cong, Yehua Wei, Yuan Zhong. 2019. Process flexibility for multiperiod production systems. operations research. *Operations Research* **67**(5) 1300–1320.

Steuer, Ralph E. 1986. Multiple criteria optimization. *Theory, computation and applications* .

Vee, Erik, Sergei Vassilvitskii, Jayavel Shanmugasundaram. 2010. Optimal online assignment with forecasts. *Proceedings of the 11th ACM conference on Electronic commerce*. 109–118.

Wang, Xinshang, Van-Anh Truong, David Bank. 2018. Online advance admission scheduling for services with customer preferences. *Available at arXiv:1805.10412* .

Wang, Xuan, Jiawei Zhang. 2015. Process flexibility: A distribution-free bound on the performance of k-chain. *Operations Research* **63**(3) 555–571.

Xu, Zhen, Hailun Zhang, Jiheng Zhang, Rachel Zhang. 2020. Online demand fulfillment under limited flexibility. *Management Science* Forthcoming.

Yang, Jian, Erik Vee, Sergei Vassilvitskii, John Tomlin, Jayavel Shanmugasundaram, Tasos Anastasakos, Oliver Kennedy. 2010. Inventory allocation for online graphical display advertising. *Available at arXiv:1008.3551* .

Zhang, Huanan, Xiuli Chao, Cong Shi. 2018. Perishable inventory systems: Convexity results for base-stock policies and learning algorithms under censored demand. *Operations Research* **66**(5) 1276–1286.

Zhong, Yuanguang, Zhichao Zheng, Mabel C Chou, Chung-Piaw Teo. 2018. Resource pooling and allocation policies to deliver differentiated service. *Management Science* **64**(4) 1555–1573.

Zinkevich, Martin. 2003. Online convex programming and generalized infinitesimal gradient ascent. *Proceedings of the 20th international conference on machine learning (icml-03)*. 928–936.

Appendix

A. Proofs

The three key theorems in this work are proved in Appendix A.2, A.3, and A.4, respectively.

A.1. Proof of Proposition 1

PROPOSITION 1. *Suppose Assumption 1 holds. For any realization of $\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_\Gamma$, the upper bound problem UB(ϕ, S) is feasible. In addition, it holds that $\mathbb{E}[opt(UB(\phi, S))] \geq \mathbb{E}[opt(DP_1(\phi, S))]$.*

PROOF. We first demonstrate the feasibility. Let's condition on a sequence $\boldsymbol{\omega}_{1:\Gamma}$ of scenarios from period 1 to γ, and let $\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_\Gamma^*$ as the sequence of decisions under the optimal policies. Even with our conditioning on $\boldsymbol{\omega}_{1:\Gamma}$, the decisions $\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_\Gamma^*$ are random variables if the optimal policy is a randomized policy[6]. Now, for each γ, let $\boldsymbol{f}_\gamma^* = \mathbb{E}[\boldsymbol{f}(\boldsymbol{x}_\gamma^*, \boldsymbol{\omega}_\gamma) | \boldsymbol{\omega}_{1:\gamma}]$, where the expectation is taken over $\boldsymbol{x}_\gamma^*$ on the internal randomness (if any) of the optimal algorithm.

We claim that $\boldsymbol{f}_1^*, \ldots, \boldsymbol{f}_\Gamma^*$ are feasible to UB(ϕ, S). Indeed, by the convexity of $d(\cdot, S)$, we have

$$d\left(\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma} \boldsymbol{f}_\gamma^*, S\right) \leq \mathbb{E}\left[d\left(\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma} \boldsymbol{f}(\boldsymbol{x}_\gamma^*, \boldsymbol{\omega}_\gamma), S\right) \mid \boldsymbol{\omega}_{1:\Gamma}\right] \leq 0,$$

and by the definition of convex hull we also have $\boldsymbol{f}_\gamma^* \in \text{Conv}(\mathcal{F}(\boldsymbol{\omega}))$ for all γ.

Finally, the required inequality follows from the feasibility of $\boldsymbol{f}_1^*, \ldots, \boldsymbol{f}_\Gamma^*$ to UB(ϕ, S) and

$$\begin{aligned}
\mathbb{E}[\text{opt}(\text{DP}(S, \phi)) | \boldsymbol{\omega}_{1:\Gamma}] &= \mathbb{E}\left[\phi\left(\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma} \boldsymbol{f}(\boldsymbol{x}_\gamma^*, \boldsymbol{\omega}_\gamma)\right) \mid \boldsymbol{\omega}_{1:\Gamma}\right] \\
&\leq \phi\left(\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma} \mathbb{E}\left[\boldsymbol{f}(\boldsymbol{x}_\gamma^*, \boldsymbol{\omega}_\gamma) \mid \boldsymbol{\omega}_{1:\Gamma}\right]\right) \qquad (9) \\
&= \phi\left(\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma} \boldsymbol{f}_\gamma^*\right) \leq \text{opt}(\text{UB}(\phi, S)).
\end{aligned}$$

Step (9) is by the concavity of ϕ. Note that conditioning on $\boldsymbol{\omega}_{1:\Gamma}$ is the same as conditioning on $\boldsymbol{\omega}_{1:\gamma}$, since the optimal policy is required to be non-anticipatory. Taking expectation on both sides over the randomness of $\boldsymbol{\omega}_{1:\Gamma}$ yields the required inequality. ■

A.2. Proof of Theorem 1

Before we state the proof of Theorem 1, we need to introduce the Azuma-Hoeffding inequality to enable our analysis. The Azuma-Hoeffding inequality is summarized as follows:

PROPOSITION 3. *Let $\{X_t\}_{t=1}^T \in [0,1]^T$ be a martingale difference sequence adapted to a filtration $\{\mathcal{F}_t\}_{t=0}^T$. That is, X_t is $\mathcal{F}_t$-measurable, and $\mathbf{E}[X_t | \mathcal{F}_{t-1}] = 0$. For any $\delta \in (0,1)$, we have*

$$\mathbb{P}\left[\left|\frac{1}{T}\sum_{t=1}^{T} X_t\right| \leq \sqrt{\frac{2\log(2/\delta)}{T}}\right] \geq 1 - \delta.$$

[6] By a randomized policy, it means that at a period γ, contingent upon the realized scenarios $\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_{\gamma-1}$ so far, the policy chooses the action x_γ according to a probability distribution on $\mathcal{X}_\gamma$. If the optimal policy turns out to be deterministic, then $\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_\Gamma^*$ are deterministic conditioned on $\boldsymbol{\omega}_{1:\Gamma}$.

Now, we are ready to present the proof of Theorem 1.

THEOREM 1. *Suppose that Assumptions 2, 3 hold. Consider the application of Algorithm 2 to the online problem* $(DP_1(\phi, \mathbb{R}^K))$*, where the input* Θ *is generated by Algorithm 1. With probability at least* $1 - O(\delta)$*, we have*

$$Reg_1(\Gamma, T) = \tilde{O}\left(L \|\mathbf{1}_K\| \left(\frac{1}{\sqrt{\Gamma}} + \frac{\max\{\lambda, 1\}}{\sqrt{T}}\right)\right).$$

PROOF. To start the analysis, for each $1 \le \gamma \le \Gamma$, we define

$$\boldsymbol{F}_\gamma := \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{\omega}_\gamma \sim \Xi_\gamma} \left[\boldsymbol{f}(\mathcal{O}(-\boldsymbol{\theta}^t, \boldsymbol{\omega}_\gamma), \boldsymbol{\omega}_\gamma) \middle| \boldsymbol{\theta}^t \right].$$

We bound the reward gained in the online problem $\mathrm{DP}_1(\phi, \mathbb{R}^K))$ as follows:

$$\begin{aligned}
&\phi\left(\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma} \boldsymbol{f}(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma)\right) && (10)\\
\ge &\phi\left(\frac{1}{T}\sum_{t=1}^{T} \boldsymbol{f}(\boldsymbol{x}^t, \boldsymbol{\omega}^t)\right) - L\left\|\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma} \boldsymbol{f}(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma) - \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{f}(\boldsymbol{x}^t, \boldsymbol{\omega}^t)\right\| \\
\ge &\phi\left(\frac{1}{T}\sum_{t=1}^{T} \boldsymbol{f}(\boldsymbol{x}^t, \boldsymbol{\omega}^t)\right) - \underbrace{L\left\|\frac{1}{T}\sum_{t=1}^{T} \boldsymbol{f}(\boldsymbol{x}^t, \boldsymbol{\omega}^t) - \frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma} \boldsymbol{F}_\gamma\right\|}_{(\ddagger)} - \underbrace{L\left\|\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma} [\boldsymbol{F}_\gamma - \boldsymbol{f}(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma)]\right\|}_{(\dagger)} \\
\ge &\phi\left(\frac{1}{T}\sum_{t=1}^{T} \boldsymbol{f}(\boldsymbol{x}^t, \boldsymbol{\omega}^t)\right) - L\|\mathbf{1}_K\| \left[\sqrt{\frac{2\log(6K/\delta)}{\Gamma}} + \sqrt{\frac{2\log(6K/\delta)}{T}}\right] \text{ w.p. } 1 - 2\delta/3. && (11)
\end{aligned}$$

Note that step (11) is the essential step that helps us bound the generalization error incurred by employing the forecast model in the offline step. We bound the terms $(\dagger, \ddagger)$ by applying the Hoeffding inequality provided in Proposition 3. The term $(\dagger)$ concerns the online algorithm, and the term $(\ddagger)$ concerns the offline algorithm. Next, we show step (11) by bounding $(\dagger, \ddagger)$ individually.

We first prove that $\Pr\left[(\dagger) \le \|\mathbf{1}_K\| \sqrt{2\log(6K/\delta)/\Gamma}\right] \ge 1 - \delta/3$. Consider an execution of the online algorithm, conditioned on the output $\Theta = \{\boldsymbol{\theta}^t\}_{t=1}^T$ by the offline algorithm, where Θ is fed to the online algorithm as the input. We claim that $\mathbb{E}[\boldsymbol{f}(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma)|\Theta] = \boldsymbol{F}_\gamma$, where the expectation is taken over $\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma$. The claim is readily justified by Line 5 in Algorithm 2, which asserts that $\boldsymbol{x}_\gamma = \mathcal{O}(-\boldsymbol{\theta}_\gamma, \boldsymbol{\omega}_\gamma)$, where $\Pr[\boldsymbol{\theta}_\gamma = \boldsymbol{\theta}^t|\Theta] = 1/T$. Consequently,

$$\begin{aligned}
&\Pr\left[(\dagger) \le \|\mathbf{1}_K\| \sqrt{\frac{2\log(6K/\delta)}{\Gamma}} \middle| \Theta\right] \\
\ge &\Pr\left[\left|\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma} f_k(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma) - \frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma} F_{\gamma,k}\right| \le \sqrt{\frac{2\log(6K/\delta)}{\Gamma}} \text{for each } 1 \le k \le K \;\middle|\; \Theta\right] && (12)\\
\ge &1 - \delta/3. && (13)
\end{aligned}$$

Step (12) is by the fact that, for each k, we have $f_k(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma) - F_{\gamma,k} \in [-1, 1]$. Step (13) is by Proposition 3 and a union bound over $k \in \{1, \ldots, K\}$. Finally, by taking the expectation over Θ, we establish the bound for $(\dagger)$.

In a similar vein, we argue that $\Pr\left[(\ddagger) \le \|\mathbf{1}_K\|\sqrt{2\log(6K/\delta)/T}\right] \ge 1-\delta/3$. Let's consider an execution of the offline algorithm. For each $t \in \{1,\dots,T\}$, consider

$$\begin{aligned}\boldsymbol{F}^t &:= \frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma} \mathbb{E}_{\boldsymbol{\omega}_\gamma^t \sim \Xi_\gamma}\left[\boldsymbol{f}(\mathcal{O}(-\boldsymbol{\theta}^t, \boldsymbol{\omega}_\gamma^t), \boldsymbol{\omega}_\gamma^t)\right],\\ \boldsymbol{Y}^t &:= \boldsymbol{f}(\boldsymbol{x}^t, \boldsymbol{\omega}^t) - \boldsymbol{F}^t, \text{ and}\\ \mathcal{F}^{t-1} &:= \sigma(\{\boldsymbol{\omega}^s, \boldsymbol{x}^s, \boldsymbol{f}(\boldsymbol{x}^s, \boldsymbol{\omega}^s), \boldsymbol{\theta}^s\}_{s=1}^{t-1} \cup \{\boldsymbol{\theta}^t\}).\end{aligned}$$

The filtration $\mathcal{F}^{t-1}$ represents the information available to the DM at the end of time step $t-1$. Evidently, for each t, the random variable $\boldsymbol{Y}^t$ is $\mathcal{F}^t$-measurable, and $\mathbb{E}[\boldsymbol{Y}_t|\mathcal{F}^{t-1}] = \mathbf{0}$. Consequently, by applying the Hoeffding inequality with $X_t = Y_k^t$ for every $k \in \{1,\dots,K\}$ and the filtration process $\{\mathcal{F}^t\}_{t=1}^T$, we know that $\Pr\left[|\sum_{t=1}^T Y_k^t/T| \le \sqrt{2\log(2K/\delta)/\Gamma}\right] \ge 1-\delta/K$. Applying a union over $k \in \{1,\dots,K\}$, we obtain our claimed inequality.

We continue by focusing on $\phi(\sum_{t=1}^T \boldsymbol{f}(\boldsymbol{x}^t, \boldsymbol{\omega}^t)/T)$. To proceed, we follow the style of Agrawal and Devanur (2015), who compared online solutions and certain offline benchmarks in certain stationary settings by considering the OMD procedure on the dual of the reward function.

To this end, recall the notation that $\lambda^2 = \max_{\boldsymbol{\theta}\in B_*(L)}\{\Lambda(\boldsymbol{\theta})\} - \min_{\boldsymbol{\theta}\in B_*(L)}\{\Lambda(\boldsymbol{\theta})\}$. We have

$$\begin{aligned}&\phi\left(\frac{1}{T}\sum_{t=1}^T \boldsymbol{f}(\boldsymbol{x}^t, \boldsymbol{\omega}^t)\right)\\ =&\min_{\boldsymbol{\theta}\in B_*(L)}\left\{\frac{1}{T}\sum_{t=1}^T \phi^*(\boldsymbol{\theta}) - \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}^t, \boldsymbol{\omega}^t)\right\}\\ \ge&\frac{1}{T}\sum_{t=1}^T\left\{\phi^*(\boldsymbol{\theta}^t) - \boldsymbol{\theta}^{t\top}\boldsymbol{f}(\boldsymbol{x}^t, \boldsymbol{\omega}^t)\right\} - \frac{4\lambda L\|\mathbf{1}_K\|}{\sqrt{T}} \qquad (14)\\ =&\frac{1}{T}\sum_{t=1}^T\left\{\phi^*(\boldsymbol{\theta}^t) - \boldsymbol{\theta}^{t\top}\boldsymbol{f}(\mathcal{O}(-\boldsymbol{\theta}^t, \boldsymbol{\omega}^t), \boldsymbol{\omega}^t)\right\} - \frac{4\lambda L\|\mathbf{1}_K\|}{\sqrt{T}}\\ \ge&\frac{1}{T}\sum_{t=1}^T\left\{\phi^*(\boldsymbol{\theta}^t) + \mathbb{E}_{\boldsymbol{\omega}_{1:\Gamma}^t \sim \Xi_{1:\Gamma}}\left[\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}(-\boldsymbol{\theta}^t)^\top \boldsymbol{f}(\mathcal{O}(-\boldsymbol{\theta}^t, \boldsymbol{\omega}_\gamma^t), \boldsymbol{\omega}_\gamma^t)\right]\right\} - \frac{\left(4\lambda + \sqrt{2\log(6K/\delta)}\right)L\|\mathbf{1}_K\|}{\sqrt{T}}\\ &\text{(w.p. } \ge 1-\delta/3) \qquad (15)\\ \ge&\frac{1}{T}\sum_{t=1}^T\left\{\phi^*(\boldsymbol{\theta}^t) + \mathbb{E}_{\boldsymbol{\omega}_{1:\Gamma}^t \sim \Xi_{1:\Gamma}, \boldsymbol{x}_{1:\Gamma}^*}\left[\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}(-\boldsymbol{\theta}^t)^\top \boldsymbol{f}(\boldsymbol{x}_\gamma^*, \boldsymbol{\omega}_\gamma^t)\right]\right\} - L\|\mathbf{1}_K\|\left[\frac{4\lambda + \sqrt{2\log(6K/\delta)}}{\sqrt{T}}\right] \qquad (16)\\ \ge&\frac{1}{T}\sum_{t=1}^T \mathbb{E}_{\boldsymbol{\omega}_{1:\Gamma}^t \sim \Xi_{1:\Gamma}, \boldsymbol{x}_{1:\Gamma}^*}\left[\min_{\boldsymbol{\theta}^*\in B_*(L)}\left\{\phi^*(\boldsymbol{\theta}^*) - \boldsymbol{\theta}^{*\top}\left(\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}(\boldsymbol{x}_\gamma^*, \boldsymbol{\omega}_\gamma^t)\right)\right\}\right] - L\|\mathbf{1}_K\|\left[\frac{4\lambda + \sqrt{2\log(6K/\delta)}}{\sqrt{T}}\right]\\ \le&\mathbb{E}[\mathrm{opt}(\mathrm{UB}(\phi, \mathbb{R}^K))] - L\|\mathbf{1}_K\|\left[\frac{4\lambda + \sqrt{2\log(6K/\delta)}}{\sqrt{T}}\right]. \qquad (17)\end{aligned}$$

Step (14) is by applying Proposition 2 on the series of functions $\{g_t\}_{t=1}^T$, defined as $g_t(\boldsymbol{\theta}) = \phi^*(\boldsymbol{\theta}) - \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}^t, \boldsymbol{\omega}^t)$, with the mirror map Λ as stated in Algorithm 1. For every t, the function g_t is $2\|\mathbf{1}_K\|$-Lipschitz continuous with respect to $\|\cdot\|_*$. Step (15) is by applying the Hoeffding inequality on the random variables, similar to the analysis of $(\ddagger)$. Step (16) is by the assumption of the optimization oracle. In this step, we denote x_γ^* as the decision optimal to $\mathrm{DP}(\phi, \mathbb{R}^K)$ at period γ, and x_γ^* is a random variable in general as previously discussed. Recall that $\boldsymbol{\omega}^1, \dots, \boldsymbol{\omega}^T$ in the offline algorithm are i.i.d. with common distribution Ξ^{off}.

Altogether, combining (11) with (17), Theorem 1 is proved. ■

A.3. Proof of Theorem 2

THEOREM 2. *Consider the problem $DP_1(\phi, S)$, and suppose that Assumptions 1, 2, and 3 are satisfied. With probability $1-O(\delta)$, Algorithm 3 satisfies the following regret bounds for $DP_1(\phi, S)$:*

$$Reg_1(\Gamma, T) = \tilde{O}\left(L\|\mathbf{1}_K\|\left(\frac{1}{\sqrt{\Gamma}} + \frac{\max\{\lambda, 1\}}{\sqrt{T}}\right)\right),$$
$$Reg_2(\Gamma, T) = \tilde{O}\left(\|\mathbf{1}_K\|\left(\frac{1}{\sqrt{\Gamma}} + \frac{\max\{\lambda, 1\}}{\sqrt{T}}\right)\right).$$

PROOF. We first show that $\hat{Z}$ (introduced in Line 6 of Algorithm 3) estimates the upper bound value $\mathbb{E}[\mathrm{opt}(\mathrm{UB}(\phi, S))]$. For each τ, consider the centered random variable $X_\tau := Z_\tau - \mathbb{E}[\mathrm{opt}(\mathrm{UB}(\phi, S))]$. Clearly, $\mathbb{E}[X_\tau] = 0$, and with certainty we have $|X_\tau| \leq 2L\|\mathbf{1}_K\|$. By the Hoeffding inequality (see Proposition 3), we know that

$$\Pr\left[\left|\hat{Z} - \mathbb{E}[\mathrm{opt}(\mathrm{UB}(\phi, S))]\right| \leq 2L\|\mathbf{1}_K\|\sqrt{\frac{2\log(2/\delta)}{T}}\right] \geq 1-\delta. \tag{18}$$

Next, we apply the analysis in the proof of Theorem 1 to evaluate the solution returned by Lines 8, 9 in Algorithm 3. Observe that the function $\breve{\phi}$ is 1-Lipschitz w.r.t. the norm $\|\cdot\|$. By extracting the reasoning from inequality (10) to inequality (15) in the proof of Theorem 1 to reward function $\breve{\phi}$, we see that, with probability $1-\delta$, we have

$$\begin{aligned}\breve{\phi}\left(\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma)\right) \geq & \frac{1}{T}\sum_{t=1}^{T}\left\{\breve{\phi}^*(\boldsymbol{\theta}^t) + \mathbb{E}_{\boldsymbol{\omega}^t_{1:\Gamma}\sim\Xi_{1:\Gamma}}\left[\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}(-\boldsymbol{\theta}^t)^\top \boldsymbol{f}(\mathcal{O}(-\boldsymbol{\theta}^t, \boldsymbol{\omega}^t_\gamma), \boldsymbol{\omega}^t_\gamma)\right]\right\} \\ & - \|\mathbf{1}_K\|\left[\sqrt{\frac{2\log(6K/\delta)}{\Gamma}} + \frac{4\lambda}{\sqrt{T}} + 2\sqrt{\frac{2\log(6K/\delta)}{T}}\right].\end{aligned} \tag{19}$$

Next, we proceed differently from the proof of Theorem 1. For the sequence of scenarios $\boldsymbol{\omega}^t_{1:\Gamma} = \{\boldsymbol{\omega}^t_\gamma\}_{\gamma=1}^{\Gamma}$ in the tth iteration of the offline Algorithm, let $(\boldsymbol{f}^{t,*}_\gamma)_{\gamma=1}^{\Gamma}$ be the optimal solution to $\mathrm{UB}(\phi, S)$ under $\boldsymbol{\omega}^t_{1:\Gamma}$. While $W^t = (\boldsymbol{f}^{t,*}_1, \ldots, \boldsymbol{f}^{t,*}_\Gamma)$ is a vector of random variables (which are $\sigma(\boldsymbol{\omega}^t_{1:\Gamma})$-measurable), the random vectors $W^1, \ldots, W^T$ are i.i.d. since $\boldsymbol{\omega}^1_{1:\Gamma}, \ldots, \boldsymbol{\omega}^T_{1:\Gamma}$ are i.i.d.. We now have

$$\begin{aligned}& \breve{\phi}^*(\boldsymbol{\theta}^t) + \mathbb{E}_{\boldsymbol{\omega}^t_{1:\Gamma}\sim\Xi_{1:\Gamma}}\left[\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}(-\boldsymbol{\theta}^t)^\top \boldsymbol{f}(\mathcal{O}(-\boldsymbol{\theta}^t, \boldsymbol{\omega}^t_\gamma), \boldsymbol{\omega}^t_\gamma)\right] \\ \geq & \breve{\phi}^*(\boldsymbol{\theta}^t) + \mathbb{E}_{\boldsymbol{\omega}^t_{1:\Gamma}\sim\Xi_{1:\Gamma}}\left[\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}(-\boldsymbol{\theta}^t)^\top \boldsymbol{f}^{t,*}_\gamma\right] \\ \geq & \min_{\boldsymbol{\theta}\in B_*(1)}\left\{\breve{\phi}^*(\boldsymbol{\theta}) + (-\boldsymbol{\theta})^\top \mathbb{E}_{\boldsymbol{\omega}^t_{1:\Gamma}\sim\Xi_{1:\Gamma}}\left[\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}^{t,*}_\gamma\right]\right\} \\ = & -d\left(\mathbb{E}_{\boldsymbol{\omega}^t_{1:\Gamma}\sim\Xi_{1:\Gamma}}\left[\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}^{t,*}_\gamma\right], \breve{S}\right).\end{aligned} \tag{20}$$

Now, we claim that (20) $= 0$ with probability at least $1-\delta$. To see the claim, firstly with certainty we have:

$$d\left(\mathbb{E}_{\boldsymbol{\omega}^t_{1:\Gamma}\sim\Xi_{1:\Gamma}}\left[\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}^{t,*}_\gamma\right], S\right) \leq \mathbb{E}_{\boldsymbol{\omega}^t_{1:\Gamma}\sim\Xi_{1:\Gamma}}\left[d\left(\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}^{t,*}_\gamma, S\right)\right] = 0.$$

Next, we have

$$\begin{aligned}\phi\left(\mathbb{E}_{\boldsymbol{\omega}_{1:\Gamma}^t\sim\Xi_{1:\Gamma}}\left[\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}_\gamma^{t,*}\right]\right) &\geq \mathbb{E}_{\boldsymbol{\omega}_{1:\Gamma}^t\sim\Xi_{1:\Gamma}}\left[\phi\left(\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}_\gamma^{t,*}\right)\right]\\ &= \mathbb{E}[\mathrm{opt}(\mathrm{UB}(\phi,S))]\\ &\geq \hat{Z} - 2L\|\mathbf{1}_K\|\sqrt{\frac{2\log(2/\delta)}{T}} \text{ w.p. } \geq 1-\delta. \end{aligned} \tag{21}$$

Step (21) is by the application of Hoeffding inequality in (18).

Altogether, we have established (20) $= 0$ with probability $\geq 1-\delta$, and (19) reduces to the following inequality that holds with probability $\geq 1-2\delta$:

$$\begin{aligned}&d\left(\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}(\boldsymbol{x}_\gamma,\boldsymbol{\omega}_\gamma), S\cap\left\{\boldsymbol{w}\in[0,1]^K : \phi(\boldsymbol{w})\geq \mathbb{E}[\mathrm{opt}(\mathrm{UB}(\phi,S))] - 4L\|\mathbf{1}_K\|\sqrt{\frac{2\log(2/\delta)}{T}}\right\}\right)\\ &\leq \|\mathbf{1}_K\|\left[\sqrt{\frac{2\log(6K/\delta)}{\Gamma}} + \frac{4\lambda}{\sqrt{T}} + 2\sqrt{\frac{2\log(6K/\delta)}{T}}\right], \end{aligned} \tag{22}$$

where we apply the Hoeffding inequality in (18) to replace the quantity $\hat{Z}$ by its lower bound. Now, inequality (22) certainly implies that, with probability at least $1-\delta$, we have

$$d\left(\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}(\boldsymbol{x}_\gamma,\boldsymbol{\omega}_\gamma), S\right) \leq \|\mathbf{1}_K\|\left[\sqrt{\frac{2\log(6K/\delta)}{\Gamma}} + \frac{4\lambda}{\sqrt{T}} + 2\sqrt{\frac{2\log(6K/\delta)}{T}}\right], \tag{23}$$

$$\phi\left(\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}(\boldsymbol{x}_\gamma,\boldsymbol{\omega}_\gamma)\right) \geq \mathbb{E}[\mathrm{opt}(\mathrm{UB}(\phi,S))] - L\|\mathbf{1}_K\|\left[\sqrt{\frac{2\log(6K/\delta)}{\Gamma}} + \frac{4\lambda}{\sqrt{T}} + 6\sqrt{\frac{2\log(6K/\delta)}{T}}\right]. \tag{24}$$

Therefore, we derive the regret bound for $\mathrm{Reg}_1(T,\Gamma)$. ■

A.4. Proof of Theorem 3

THEOREM 3. *Assume the access to a forecast model and an optimization oracle, according to Assumptions 2, 3. Consider the application of Algorithm 5 to the online problem $(DP_1(\phi,\mathbb{R}^K))$, where the input Θ is generated by Algorithm 4. With probability at least $1-O(\delta)$, we have*

$$Reg_3(\Gamma,T) = \tilde{O}\left(L\|\mathbf{1}_K\|\left(\frac{\max\{\lambda,1\}}{\sqrt{T}}\right)\right).$$

PROOF. We bound the expected average reward gained in the online problem $\mathrm{DP}_1(\phi,\mathbb{R}^K))$ as follows:

$$\begin{aligned}&\phi\left(\mathbb{E}\left[\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}(\boldsymbol{x}_\gamma,\boldsymbol{\omega}_\gamma)\right]\right)\\ \geq&\phi\left(\frac{1}{T}\sum_{t=1}^{T}\left[\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}(\boldsymbol{x}_\gamma^t,\boldsymbol{\omega}_\gamma^t)\right]\right) - L\underbrace{\left\|\mathbb{E}\left[\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}(\boldsymbol{x}_\gamma,\boldsymbol{\omega}_\gamma)\right] - \frac{1}{T}\sum_{t=1}^{T}\left[\frac{1}{\Gamma}\sum_{t=1}^{\Gamma}\boldsymbol{f}(\boldsymbol{x}_\gamma^t,\boldsymbol{\omega}_\gamma^t)\right]\right\|}_{(\S)}\\ \geq&\phi\left(\frac{1}{T}\sum_{t=1}^{T}\left[\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}(\boldsymbol{x}_\gamma^t,\boldsymbol{\omega}_\gamma^t)\right]\right) - L\|\mathbf{1}_K\|\left[\sqrt{\frac{2\log(4K/\delta)}{T}}\right] \text{ w.p. } 1-\delta/2. \end{aligned} \tag{25}$$

We prove that $\Pr\left[(\S)\leq\|\mathbf{1}_K\|\sqrt{2\log(4K/\delta)/\Gamma}\right]\geq 1-\delta/2$. Consider an execution of the online algorithm, conditioned on the output $\Theta=\{\boldsymbol{\theta}^t\}_{t=1}^T$ by the offline algorithm, where Θ is fed to the online algorithm as the input. We claim that

$$\mathbb{E}\left[\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}(\boldsymbol{x}_\gamma,\boldsymbol{\omega}_\gamma)\;\middle|\;\Theta\right] = \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}(\boldsymbol{x}_\gamma^t,\boldsymbol{\omega}_\gamma^t)\right],$$

where the expectation on the left hand-side is taken over all the $\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma$. The claim is readily justified by Line 5 in Algorithm 5, which asserts that $\boldsymbol{x}_\gamma = \mathcal{O}(-\boldsymbol{\theta}_\gamma, \boldsymbol{\omega}_\gamma)$ with $\Pr[\boldsymbol{\theta}_\gamma = \boldsymbol{\theta}^t | \Theta] = 1/T$. Using a similar argument in the proof of Theorem 1, we have

$$
\begin{aligned}
&\Pr\left[(\S) \leq \|\mathbf{1}_K\| \sqrt{\frac{2\log(4K/\delta)}{T}} \;\middle|\; \Theta\right] \\
\geq &\Pr\left[\left|\frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma} f_k(\boldsymbol{x}_\gamma^t, \boldsymbol{\omega}_\gamma^t) - \frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\mathbb{E}\left[f_k(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma)\right]\right)\right| \leq \sqrt{\frac{2\log(4K/\delta)}{T}} \text{for each } 1 \leq k \leq K \;\middle|\; \Theta\right] \\
\geq &1 - \delta/2. \qquad (26)
\end{aligned}
$$

Step (26) is by Proposition 3 and a union bound over $k \in \{1, \ldots, K\}$. Finally, by taking the expectation over Θ, we establish the bound for (§).

We continue by focusing on $\phi(\frac{1}{T}\sum_{t=1}^{T}(\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}(\boldsymbol{x}_\gamma^t, \boldsymbol{\omega}_\gamma^t)))$. To proceed, we use the same technique in the proof of Theorem 1 to compare the online solutions and certain offline benchmarks by considering the OMD procedure on the dual of the reward function. Recall the notation that $\lambda^2 = \max_{\boldsymbol{\theta} \in B_*(L)}\{\Lambda(\boldsymbol{\theta})\} - \min_{\boldsymbol{\theta} \in B_*(L)}\{\Lambda(\boldsymbol{\theta})\}$. We have

$$
\begin{aligned}
&\phi\left(\frac{1}{T}\sum_{t=1}^{T}\left[\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}(\boldsymbol{x}_\gamma^t, \boldsymbol{\omega}_\gamma^t)\right]\right) \\
= &\min_{\boldsymbol{\theta} \in B_*(L)}\left\{\frac{1}{T}\sum_{t=1}^{T}\phi^*(\boldsymbol{\theta}) - \boldsymbol{\theta}^\top\left[\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}(\boldsymbol{x}_\gamma^t, \boldsymbol{\omega}_\gamma^t)\right]\right\} \\
\geq &\frac{1}{T}\sum_{t=1}^{T}\left\{\phi^*(\boldsymbol{\theta}^t) - \boldsymbol{\theta}^{t\top}\left[\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}(\boldsymbol{x}_\gamma^t, \boldsymbol{\omega}_\gamma^t)\right]\right\} - \frac{4\lambda L\|\mathbf{1}_K\|}{\sqrt{T}} \qquad (27) \\
= &\frac{1}{T}\sum_{t=1}^{T}\left\{\phi^*(\boldsymbol{\theta}^t) - \boldsymbol{\theta}^{t\top}\left[\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}(\mathcal{O}(-\boldsymbol{\theta}^t, \boldsymbol{\omega}_\gamma^t), \boldsymbol{\omega}_\gamma^t)\right]\right\} - \frac{4\lambda L\|\mathbf{1}_K\|}{\sqrt{T}} \\
\geq &\frac{1}{T}\sum_{t=1}^{T}\left\{\phi^*(\boldsymbol{\theta}^t) + \mathbb{E}_{\boldsymbol{\omega}_{1:\Gamma}^t \sim \Xi_{1:\Gamma}}\left[\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}(-\boldsymbol{\theta}^t)^\top\boldsymbol{f}(\mathcal{O}(-\boldsymbol{\theta}^t, \boldsymbol{\omega}_\gamma^t), \boldsymbol{\omega}_\gamma^t)\right]\right\} - \frac{\left(4\lambda + \sqrt{2\log(4K/\delta)}\right)L\|\mathbf{1}_K\|}{\sqrt{T}} \\
&(\text{w.p.} \geq 1 - \delta/2) \qquad (28) \\
\geq &\frac{1}{T}\sum_{t=1}^{T}\left\{\phi^*(\boldsymbol{\theta}^t) + \mathbb{E}_{\boldsymbol{\omega}_{1:\Gamma}^t \sim \Xi_{1:\Gamma}}\left[\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}(-\boldsymbol{\theta}^t)^\top\boldsymbol{f}_\gamma^{t,*}\right]\right\} - L\|\mathbf{1}_K\|\left[\frac{4\lambda + \sqrt{2\log(4K/\delta)}}{\sqrt{T}}\right] \qquad (29) \\
\geq &\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{\boldsymbol{\omega}_{1:\Gamma}^t \sim \Xi_{1:\Gamma}}\left[\min_{\boldsymbol{\theta}^* \in B_*(L)}\left\{\phi^*(\boldsymbol{\theta}^*) - \boldsymbol{\theta}^{*\top}\left(\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}_\gamma^{t,*}\right)\right\}\right] - L\|\mathbf{1}_K\|\left[\frac{4\lambda + \sqrt{2\log(4K/\delta)}}{\sqrt{T}}\right] \\
= &\mathbb{E}[\text{opt}(\text{UB}(\phi, \mathbb{R}^K))] - L\|\mathbf{1}_K\|\left[\frac{4\lambda + \sqrt{2\log(4K/\delta)}}{\sqrt{T}}\right]. \qquad (30)
\end{aligned}
$$

Step (27) is by applying Proposition 2 on the series of functions $\{g_t\}_{t=1}^T$, defined as $g_t(\boldsymbol{\theta}) = \phi^*(\boldsymbol{\theta}) - \boldsymbol{\theta}^\top\left[\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\boldsymbol{f}(\boldsymbol{x}_\gamma^t, \boldsymbol{\omega}_\gamma^t)\right]$, with the mirror map Λ as stated in Algorithm 4. For every t, the function g_t is $2\|\mathbf{1}_K\|$-Lipschitz continuous with respect to $\|\cdot\|_*$. Step 28 is by an application of the Hoeffding inequality. Step (29) is by the assumption of the optimization oracle, and the last step (30) is by the definition of $\mathbb{E}[\text{opt}(\text{UB}(\phi, \mathbb{R}^K))]$.

Altogether, Theorem 3 is proved. ■

B. Application in Online Graphical Display Advertising

To further illustrate the practicability of our algorithm framework, we apply the O2O algorithm to address another complicated application – online graphical display advertising (OGDA) – in operations management. OGDA aims at displaying graphical messages (e.g., video, image) to targeted users (Yang et al. 2010). Similar to other types of online advertising, one of the crucial issues in OGDA is how to appropriately allocate the resources (user visits/impressions) to serve the demands (advertiser campaigns). This indeeds hinges on the objectives specified by the web publisher and advertisers. In general, the publisher can make different contracts with advertisers. On one hand, multiple advertisers can bid in real-time for one ad slot when a user visits the web page. Based on the (predicted) probability that a user click on the ad (usually called click-through rate in the literatures), advertisers may bid at different prices. Intuitively, advertisers would bid higher if the user is more likely to click on the ad, and the objective for the advertiser is to obtain clicks on the ad. The publisher would then choose the highest bidder so as to maximize the total revenue. On the other hand, in order to reach a diverse pool of audience for a long-term benefit, the advertisers can buy a fixed number of ad impressions in advance, and then the publisher has to guarantee the required amount of impressions over a particular period. We call this contract as *Delivery Guarantee* throughout this section. Such a delivery guarantee has been widely used in practice, such as Yahoo's RightMedia and Google's DoubleClick (Yang et al. 2010, Balseiro et al. 2014). In presence of delivery guarantee, the publisher is faced with the dilemma of reserving impressions for different advertisers, while maximizing the total revenue obtained from the bidding algorithm in real-time. This leads to the central question addressed in this section: *How to allocate the impressions to different advertisers, so as to meet the delivery guarantee, while to maximize the total revenue?*

This is indeed a multi-objective problem faced by many web page publishers. Yang et al. (2010) studied a deterministic version of this problem (i.e., assuming that all the user visits are given) by taking a sequence of weight functions on different objectives, and then optimized the (single) weighted-sum objective. Under this scheme, it could be laborious and challenging to choose the right weight functions, in particular when the relative importance associated with each objective function can not be directly quantified. In the present work, we provide an alternative approach to address this class of multi-objective optimization problems. We formulate the delivery guarantee requirements as a set of feasibility constraints, with the objective to maximize total revenue. Furthermore, we employ a stochastic model and allocate the impressions to different advertisers in real-time upon the arrival of each user visit.

More concretely, we use a bipartite graph to represent the ads allocation network. On one side is a set of $\mathcal{I} = \{1, \ldots, |\mathcal{I}|\}$ users, whereas on the other side is a set of $\mathcal{J} = \{1, \ldots, |\mathcal{J}|\}$ advertisers. The publisher allocates the arriving user $i \in \mathcal{I}$ to at most one advertiser $j \in \mathcal{J}$ at each period $\gamma = 1, \ldots, \Gamma$. While not necessary, we assume that each period corresponds to a small enough interval of time and there is at most one user visit at each period. Furthermore, the arrival of type-i user, denoted by a random variable $\omega_{\gamma,i}$, follows a non-stationary stochastic process over the planning horizon. $\omega_{\gamma,i} = 1$ if type-i user arrives at period γ; and $\omega_{\gamma,i} = 0$ otherwise. Upon the arrival of type-i user at period γ, advertiser-j bids for the user visit at price $b_{i,j}$, which depends on the user type. In fact, our modeling framework also allows the advertiser-j to

bid for the type-i user at period γ with a randomized price $b_{i,j} + \epsilon_\gamma$, where ϵ_γ represents a random noise term. In addition, we allow each advertiser-j to specify an initial budget B_j over the planning horizon. At the beginning of the planning horizon, advertisers sign a delivery contract with the publisher. For ease of exposition, we denote the required proportion of delivery to advertiser-j as β_j, and hence the publisher needs to allocate $\beta_j \Gamma$ impressions to advertiser-j over the entire planning horizon under the delivery guarantee contract. Overall, the publisher then has two choices at each period: (1) allocating the impression to the highest bidder while not violating the total budget of each advertiser; (2) allocating the impression to serve the delivery guarantee constraint of each advertiser. Given a sequence of user visits $\{\boldsymbol{\omega}_\gamma\}_{\gamma=1}^{\Gamma}$, we can formulate the deterministic version of online graphical display advertising problem as:

$$\begin{aligned} \text{(OGDA)} \quad \max \; & \sum_{\gamma=1}^{\Gamma} \sum_{i=1}^{|\mathcal{I}|} \sum_{j=1}^{|\mathcal{J}|} b_{i,j} x_{\gamma,i,j} \\ \text{s.t.} \; & \sum_{\gamma=1}^{\Gamma} \sum_{i=1}^{|\mathcal{I}|} x_{\gamma,i,j} \geq \beta_j \Gamma, \; \forall j \in \mathcal{J} \\ & \sum_{\gamma=1}^{\Gamma} \sum_{i=1}^{|\mathcal{I}|} b_{i,j} x_{\gamma,i,j} \leq B_j, \; \forall j \in \mathcal{J} \\ & \boldsymbol{x}_\gamma \in \mathcal{X}(\boldsymbol{\omega}_\gamma) \end{aligned}$$

where

$$\mathcal{X}(\boldsymbol{\omega}_\gamma) := \left\{ \boldsymbol{x}_\gamma \in \mathbb{R}^{|\mathcal{I}| \times |\mathcal{J}|} \;\middle|\; \begin{array}{ll} \displaystyle\sum_{j=1}^{|\mathcal{J}|} x_{\gamma,i,j} \leq w_{\gamma,i}, & \forall i \in \mathcal{I} \\ x_{\gamma,i,j} \in \{0,1\}, & \forall i \in \mathcal{I}, j \in \mathcal{J} \end{array} \right\}.$$

In problem (OGDA), the objective is to maximize the total revenue. The first set of constraints ensures the delivery guarantee to each advertiser, and the second set of constraints forces the budget of each advertiser not to be violated. In the feasible region $\mathcal{X}(\boldsymbol{\omega}_\gamma)$, the first set of constraints forces that each user-j, if comes at period γ, can be allocated to at most one advertiser j; the second set of constraints indicates that the allocation decision is a binary variable.

While the deterministic OGDA problem can be directly solved using standard optimization techniques, it is by no means to be trivial to solve the stochastic OGDA problem in an online manner. In what follows, we show that the O2O algorithm can be applied here to solve this complicated problem. To see this, we introduce $2|\mathcal{J}|+1$ performance metrics:

$$f_0(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma) := \sum_{i=1}^{|\mathcal{I}|} \sum_{j=1}^{|\mathcal{J}|} b_{i,j} x_{\gamma,i,j} - 0; \; f_j(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma) := \sum_{i=1}^{|\mathcal{I}|} x_{\gamma,i,j} - \beta_j, \; \forall j \in \mathcal{J}; \text{ and}$$

$$f_{|J|+j}(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma) := -\sum_{i=1}^{|\mathcal{I}|} b_{i,j} x_{\gamma,i,j} + B_j/\Gamma, \; \forall j \in \mathcal{J}.$$

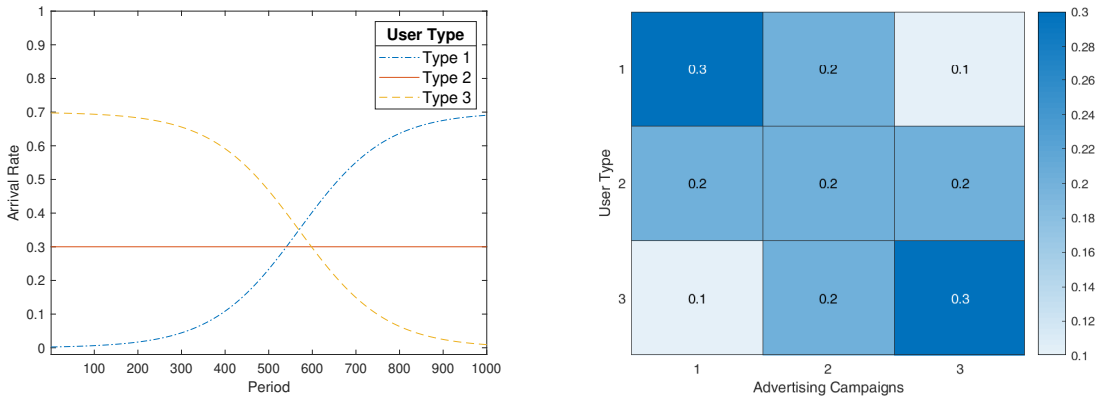
Furthermore, we represent the global objective function as $\phi(\boldsymbol{w}) := w_0 + \sum_{j=1}^{2|\mathcal{J}|}(0 \times w_j)$ for $\boldsymbol{w} \in \mathbb{R}^{2|\mathcal{J}|+1}$. In this way, problem (OGDA) can be equivalently reformulated as:

$$\text{(OGDA-O2O)} \quad \max \; \phi\left(\frac{1}{\Gamma} \sum_{\gamma=1}^{\Gamma} \boldsymbol{f}(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma) \right)$$

$$\text{s.t.} \quad d\left(\frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma} \boldsymbol{f}(\boldsymbol{x}_\gamma, \boldsymbol{\omega}_\gamma), \mathbb{R}_+^{2|\mathcal{J}|+1}\right) \leq 0$$
$$\boldsymbol{x}_\gamma \in \mathcal{X}(\boldsymbol{\omega}_\gamma)$$

Next, we numerically demonstrate that our O2O algorithm achieves near-optimal performance in terms of total revenue obtained, and ensures both the delivery guarantee and budget satisfaction to all advertisers almost surely. We consider a concrete case with three types of users and three types of advertisers in the ads allocation network. The non-stationary user arrival rates are depicted in Figure 11(a). We normalize the summation of total arrival rate to be 1 so that there is only one user arrival at each period. The value of bidding price $b_{j,i}$ is provided in Figure 11(b). For example, advertiser-3 bids for user-1 at price 0.1.

Figure 11 Heterogeneous user arrivals and advertiser bidding strategies.



(a) User Arrival Rates

(b) User-Dependent Bidding Prices

We consider three policies for comparison. The hindsight policy without delivery guarantee is set as the benchmark policy, and we evaluate the relative revenue loss of the remaining two policies. More concretely, the relative revenue loss is defined as the revenue gap to the benchmark policy and then divided by the total revenue under the benchmark policy. We introduce the penalty function $\|\max\{\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}, \mathbf{0}\}\|_2$ to measure the delivery guarantee gap between the proportion of delivered impressions $\hat{\boldsymbol{\beta}}$ and the delivery target $\boldsymbol{\beta}$. Note that the user arrivals are randomly generated, we simulate 10^3 sample paths and compare the average performance.

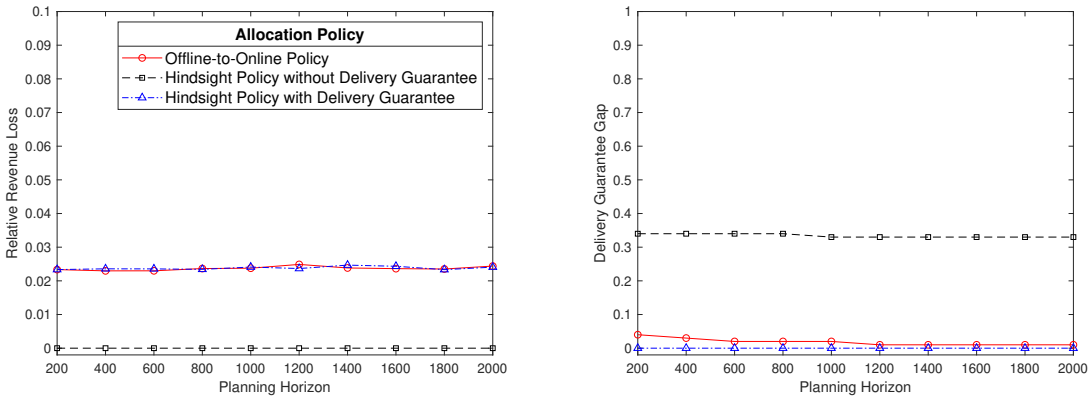
1. Hindsight Policy with Delivery Guarantee: This policy knows all the information (i.e., user arrival sequence) in advance, and we solve a deterministic version of problem (OGDA).

2. Hindsight Policy without Delivery Guarantee: This policy also knows all the information in advance. We remove the delivery guarantee constraint from problem (OGDA) and solve a deterministic version of this revised problem.

3. Offline-to-Online Policy: The O2O algorithm proposed in this paper. At the offline stage, we set the number of iterations $T = 10^5$, and use the multiplicative weight policy to update the weight vector $\boldsymbol{\theta}_t$ at each iteration.

To highlight the trade-offs between revenue maximization and delivery guarantee, we suppress the discussion on the impact of budget constraints. The total budget specified by each advertiser-j is set to be $B_j = 0.3 \times \Gamma$ so that all the budget constraints can be completely satisfied. Furthermore, we assume that all the advertisers sign the same delivery guarantee contract with $\beta_j = 1/3$ for all $j \in \mathcal{J}$. In this setting, the publisher equally allocates the impressions to three advertisers during the planning horizon. By varying the planning horizon Γ from 200 to 2000, we observe that, as shown in Figure 12, the O2O policy achieves near-optimal performance in terms of both revenue maximization and delivery guarantee in all cases, compared with the hindsight policy with delivery guarantee. The delivery guarantee gap under the O2O policy slightly shrinks with the increase of Γ, which implies that the O2O policy can tolerate more errors over a longer planning horizon. This is consistent with the performance guarantee result in Theorem 2. Notably, the publisher suffers from a significant volume of revenue loss in order to fulfill the delivery guarantee contract, as shown in Figure 12(a). On the other hand, Figure 12(b) confirms that the revenue maximization approach cannot ensure feasible delivery guarantee.

Figure 12 Comparison of different policies for the OGDA problem.



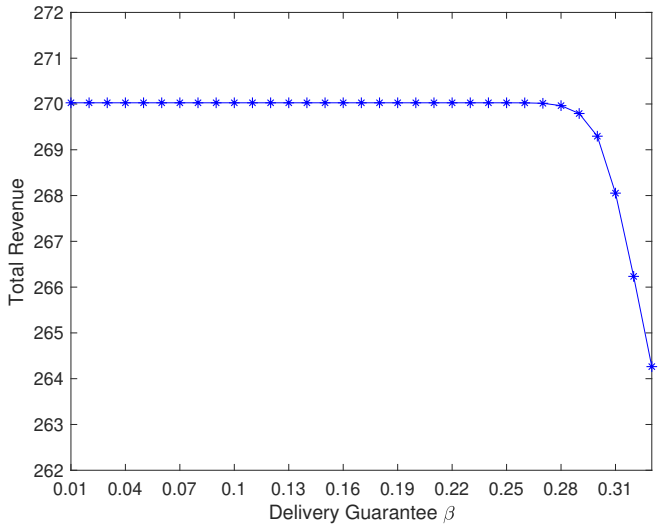
(a) Relative Revenue Loss

(b) Delivery Guarantee Gap

A natural follow-up question faced by the publisher is how to design the delivery guarantee contract to balance the objective of maximizing total revenue and the objective of reaching more audiences for different advertisers? To answer this question, the next experiment explores the impact of delivery contract parameter on the total revenue. For ease of exposition, we consider the same contract signed with different advertisers, i.e., we set $\beta_j = \beta$ for $\forall j \in \mathcal{J}$. We consider the case with $\Gamma = 10^3$. By varying the parameter β from 0.01 to 0.33, we plot the revenue vs. delivery guarantee graph, together with the efficient frontier in Figure 13.

We observe that the total revenue is not affected when the delivery guarantee parameter β is small. Once the parameter β reaches a threshold, the total revenue reveals a decreasing pattern. In this way, we use the delivery contract to balance the trade-offs between different conflicting objectives, which is the core issue in multi-objective optimization. Following the classic approach in the multi-objective optimization literatures

Figure 13 Efficient frontier in the OGDA problem.



(e.g., Steuer 1986, Yang et al. 2010), we can restrict the search of a "satisfactory" solution on the efficient frontier, and in turn design an attractive delivery contract.

At a high level, our O2O framework facilitates an alternative option to solve the stochastic multi-objective optimization problem with non-stationary inputs in an online fashion. The decision maker can select the most important objective to optimize, whereas specifies an attainable target set for the remaining objectives and put them together into the feasibility constraint.