3-2024

Multiobjective Stochastic Optimization: A Case of Real-Time Matching in Ride-Sourcing Markets

Guodong LYU
Hong Kong University of Science and Technology

Wang Chi CHEUNG
National University of Singapore

Chung-Piaw TEO
National University of Singapore

Hai WANG
Singapore Management University, haiwang@smu.edu.sg

Multi-Objective Stochastic Optimization

A Case of Real-Time Matching in Ride-Sourcing Markets

Guodong Lyu
School of Business and Management, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong

Wang Chi Cheung
Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore, SG 117576

Chung-Piaw Teo
Institute of Operations Research and Analytics, National University of Singapore, Singapore, SG 117602
NUS Business School, National University of Singapore, Singapore, SG 117592

Hai Wang*
School of Computing and Information Systems, Singapore Management University, Singapore, SG 178902

Problem Definition: The job of any marketplace is to facilitate the matching of supply with demand in real-time. Success is often measured using various metrics. The challenge is to design matching algorithms to balance the trade-offs among multiple objectives in a stochastic environment, to arrive at a "compromise" solution, which minimizes say the ℓ_p-norm-based distance function (for some $1 \leq p \leq \infty$) between the attained performance metrics and the target performances.

Methodology/Results: We observe that the sample-average-approximation formulation of this multi-objective stochastic optimization problem can be solved by an online algorithm that uses only gradient information from "historical" (i.e., past) sample information, and not on the current state of the system. The online algorithm relies on a set of weight functions, which are updated adaptively over time, based on real-time tracking of the gaps in attained performance and the performance target. This allows us to recast the online algorithm as a randomized algorithm to solve the original stochastic problem. When the pre-determined performance targets are attainable, our randomized policy achieves the targets with a near-optimal performance guarantee (measured by regret, or deviation away from the optimal performance). When the targets are not attainable, our policy generates a compromise solution to the multi-objective stochastic optimization problem, even when the efficient frontier for this stochastic optimization problem cannot be explicitly characterized a-priori. We implement our model to address a challenge faced by a ride-sourcing platform, that matches passengers and drivers in real-time. Four performance metrics—platform revenue, driver service score, pick-up distance, and number of matched pairs—are simultaneously considered in the design of ride-matching algorithm, without pre-specifying the weight on each performance metric. This mechanism has been extensively tested using synthetic and real data.

Managerial Implications: We show that under appropriate conditions, all parties in the ride-sourcing ecosystem, from drivers, passengers, to the platform, can be better off under our compromise matching policy, compared to other popular policies currently in use. In particular, the platform can obtain higher revenue, ensure better drivers (with higher service scores) are assigned more orders, and passengers are more likely to be matched to better drivers (albeit with a slight increase in the waiting time), compared to existing policies that focus on pick-up distance minimization. The ability to balance the conflicting goals in multiple objectives in a stochastic operating environment, has the potential to contribute to the long-term sustainable growth of ride-sourcing platforms.

Key words: Multi-objective Optimization; Compromise Solution; Online Algorithms; Ride-sourcing

1. Introduction

In many cities in the world, ride-sourcing companies such as Uber, Lyft, Grab, and Didi Chuxing, have been able to leverage on Internet-based platforms to facilitate the integration of passengers' and drivers' mobility data on smart phones in real-time, to match passengers to drivers nearby. This allows the online platforms to dispense with the need for drivers to cruise around to search for passengers on the streets. These clear operational advantages have motivated many shared service business models

* Corresponding Author

in public transportation, and have been a disruptive force to the traditional taxi industry. Uber, for instance, has grown from a ride-sourcing service provider to a one-stop mobile transportation platform, offering a variety of services in more than 700 metropolitan areas in 65 countries. It offers a menu of services, including UberX, Uber Black, Uber Pool, Taxi, SUV, etc., to more than 91 million users with 15 million daily riders (Data as of July, 2019). Didi Chuxing, in China, offers a variety of services from Taxi, Express, Premier, Luxe, Hitch, Bus, Minibus, Designated Driving, Car Rental, Enterprise Solutions and Bike-Sharing etc., to more than 550 million users (Data as of February, 2019). These companies are now evaluating electric vehicles and self-driving technologies, to address the transportation needs of the citizens in a sustainable urban ecology.

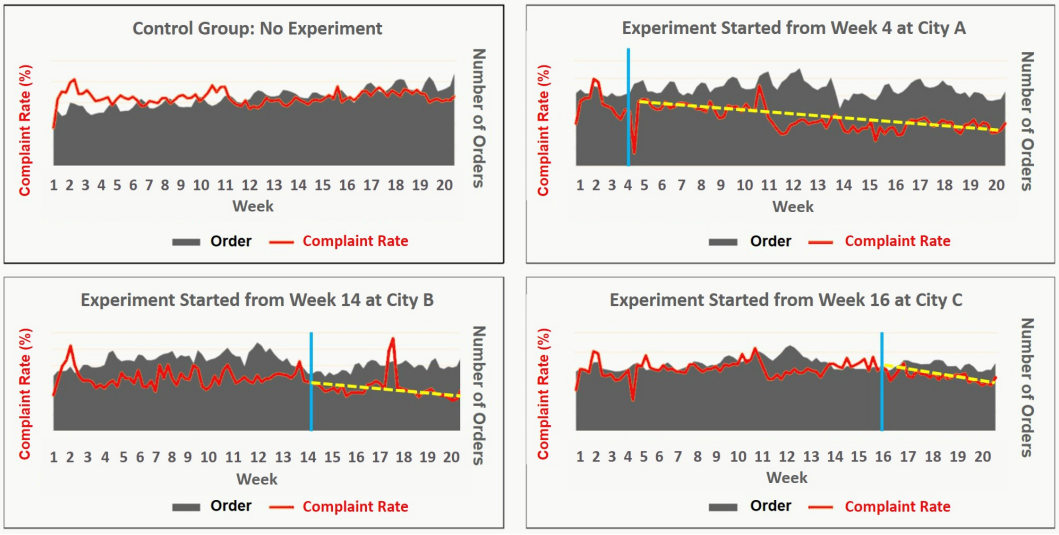
While the technology is now ripe for large scale deployment, matching passengers (demand) with drivers (supply) in real-time is a challenging problem for the ride-sourcing platforms due to multiple objectives to be considered. The most straightforward short-term objective for the platform is to maximize the total revenue, which depends on the served order value and the total number of served or canceled passengers. Another short-term objective is to minimize the pick-up distance for passengers. In practice, the greedy policy, which assigns passenger to the "nearest" available driver, is widely used so that passengers can be picked-up as quick as possible. However, this ignores the fundamental challenge to build a strong community of service providers (part- and full-time drivers) and to deliver high quality service experience to the passengers. Clearly, the latter objective is also crucial for a long-term sustainable growth for the platforms. Uber, for instance, uses ratings provided by passengers to rate the drivers, and boots those drivers whose ratings fall below a threshold in their system, in order to improve the quality of ride service provided to the passengers.

How should we incorporate multiple objectives, including platform revenue, pick-up time, and ride service quality, in the design of the dispatch algorithm? A standard solution is to find an appropriate way to combine the various objectives into a single "merged" objective. One industry player uses the service score, which is used to quantify driver's service quality, to distort the distance function so that drivers with higher service scores have lower distortion compared to those with lower service scores, so that the chances of jobs being assigned to a better driver are higher under the greedy policy using the distorted distance function. The A/B tests (by setting a large weight on the service score in the matching policy) performed by our industry collaborator seem to indicate a systematic decrease in the number of complaints against the drivers with this change in implementation (cf. Figure 1), possibly due to an improvement in the quality of ride service. Unfortunately, the total revenue earned also dropped in the field experiment.[1] While it is necessary to carefully tune the distortion function so that the performances on service quality and total revenue are balanced, finding and calibrating

[1] We use simulations to recover the A/B test result in Section 5. We demonstrate that a standard service quality maximization policy achieves good performance on service score, but at the sacrifice of the total revenue.

the right distortion function is laborious and non-trivial, and there is a lack of theory to guide the selection of such functions.

Figure 1 **A/B Test on assigning more jobs to drivers with higher service scores. The Yellow 'Dashed-line' indicates a decreasing trend in the passenger complaint rate after the experiment.**



This challenge motivates us to study a class of multi-objective decision making problems, and design an efficient algorithm to balance these objectives in a calibrated manner. In multi-objective optimization, it is natural to restrict our search of a solution on the Pareto frontier (Steuer 1986). Note that this traditional approach entails a delicate selection of weighting function to aggregate the multiple objectives into a single one, and the central issue there is also the choice of the weighting function to be used for aggregation (Marler and Arora 2010). To this end, Yu (1973) introduced the concept of a "compromise solution"—a feasible solution whose performance is closest (in terms of the Euclidean metric) to an ideal point (or a predetermined Key Performance Indicator (KPI) target) in the K-dimensional performance space, when we measure the quality of a solution using K different objective functions. The ideal point is called a utopia point, which can be constructed by optimizing each objective function separately, but is otherwise non-attainable at the same time.

Given the utopia target, it is straightforward to obtain the Euclidean (ℓ_2-norm) compromise solution by solving a related quadratic programming problem, if we have a good characterization of the efficient frontier. However, in a stochastic setting with dynamic decision scenarios, e.g., dynamic arrival of passengers and drivers on the ride-sourcing platforms, it remains a daunting challenge to find the compromise solution.

In this paper, our contributions are as follows:

1. We first solve a sample-average-approximate (SAA) version of the multi-objective stochastic optimization problem, and develop an online algorithm (associated with a set of weight functions) to obtain the "compromise" solution to the SAA problem. The weight function in the SAA problem is adaptive, and uses only historical sample information generated.[2] This allows us to recast the online algorithm as a randomized algorithm for this class of stochastic problems, and provides explicit allocation rules to guide deployment of resources in real-time. The methods and analyses are developed for problems without recourse (single-period) and with multiple stages of recourse (multi-period) settings.
2. We determine whether the aspirational KPI targets are attainable for a given problem instance. To be concrete, we obtain a set of necessary and sufficient conditions to characterize the space of attainable KPI metrics. Interestingly, if the KPI targets (e.g., the utopia target) are not attainable, our algorithm generates the compromise solution to the multi-objective stochastic optimization problem, even when the efficient frontier cannot be easily characterized. This result holds for any ℓ_p-norm induced penalty functions, where $1 \leq p \leq \infty$.

This multi-objective framework is developed to address a crucial question faced by the ride-sourcing platform: How should multiple objectives be addressed in the real-time matching decisions? We apply our framework to incorporate four objectives—platform revenue maximization, service quality maximization, pick-up distance minimization, and number of matched pair maximization—in the dispatch algorithm, and drive the compromise matching policy. Compared to solutions that focus on optimizing only a single metric, we have the following observations:

3. Compared to the pick-up distance minimization policy, the compromise matching solution dispatches more orders to drivers with higher service scores, sacrificing only a little on pick-up time. This has long term strategic consequences—as better drivers (higher service scores) are assigned more jobs, their income will increase, and this could motivate them to work longer on the platform. Consequently, the overall service quality of the platform will improve.
4. Compared to the service quality maximization policy, the compromise matching solution serves higher value passengers (those with longer commute distances) with higher priority, and provides them with faster pick-up service. As a result, the revenue accrued is higher, despite that a small proportion of lower value passengers are underserved. We note that some ride-sourcing platforms are already investing in bike-sharing and scooter-sharing markets, which aim to serve passengers with shorter commute distances. In this way, our matching policy could help balance their business strategies in both ride-sourcing and bike-/scooter-sharing markets, to provide

[2] Note that the traditional SAA approach uses the information on all the samples to determine the optimal solution. Even though the SAA approach may return an approximation to the optimal value when the problem scale is not too large, this method does not provide a means to adapt to a scenario different from its sampled scenarios.

more comfortable service for passengers with differentiated travel requests for both long and short distances.

The rest of the paper is structured as follows. Relevant literature is reviewed in Section 2. We formulate the multi-objective optimization problem and develop the compromise solution in Section 3. In Section 4, we extend our analytical results from a single-period setting to a multi-period setting. In Section 5, we implement the multi-objective approach in a ride-sourcing market, and Section 6 concludes. All proofs are relegated to Appendix A. In Appendix B, we provide more details on the ride-matching benchmark policies. As further checks on the performance of the proposed policies, we also describe an implementation of our compromise matching solution using an in-house simulator developed by our industry collaborator, in Appendix C.

2. Literature Review

Multi-objective optimization has been extensively studied in both academic and industrial areas, where the decision making process is concerned with trade-offs under multiple criteria. We refer the readers to Marler and Arora (2004), Ehrgott (2005), Yu (2013), Serafini (2014) and the references therein for a comprehensive review of the fundamentals of this problem.

A classical approach to address the multi-objective optimization problems is to characterize a set of Pareto optimal solutions so that the decision maker can choose the most "preferred" one. The space composed of these Pareto optimal solutions is also called "efficient frontier" in the literature (Steuer 1986). Among all the Pareto solutions, Yu (1973) introduced the concept of compromise solution and focused on the ℓ_p norms functions to study the distance from the compromise solution to the utopia point. Gearhart (1979) and White (1984) extended this topic by considering different distance functions. Notably, when the distance function is given, it is straightforward to derive the compromise solution in the setting where both objective function and feasible domain are well characterized. However, when the feasible domain is presented in a stochastic setting, characterizing the compromise solution becomes extremely challenging. In this paper, we propose an online mechanism to derive the compromise solution to the multi-objective optimization problem with stochastic inputs. Next, we restrict our focus on the related multi-objective online optimization literature.

Mahdavi et al. (2013) studied a multi-objective optimization problem under the online convex optimization (OCO) framework. They casted the multi-objective problem into a stochastic constrained problem by forcing different thresholds on the original objective functions and selected a new objective function to optimize. A primal-dual online algorithm was developed to minimize the selected single objective and to meet these pre-determined thresholds. We note that the feasible domain for the decision variables is fixed in this paper, and the decision maker was required to make the decision prior to the scenario realization. In our paper we study a different multi-objective problem in which

the feasible domain depends on the realized stochastic information and the decision is made after observing the realized scenario. Uziel and El-Yaniv (2017) extended the multi-objective work in Mahdavi et al. (2013) to the case where the underlying unknown process is stationary and ergodic, and proposed a minimax histogram aggregation algorithm to solve the multi-objective online optimization problem. By introducing a threshold on each objective function, the multi-objective problem is converted to a stochastic constrained OCO problem, which was also studied by Yu et al. (2017). Busa-Fekete et al. (2017) studied the multi-objective online bandits problem. Different from the classic single-objective multi-armed bandit problem (Auer et al. 2002), the obtained cost after each round is no longer a scalar but a vector. They reformulated the multi-objective problem by using the Generalized Gini Index aggregation function and developed an online gradient descent algorithm.

The multi-objective problem studied in the present paper is also related to the online stochastic optimization problem with feasibility constraints. In this vein, Zhong et al. (2018) studied a capacity-pooling problem with fill-rate constraints. Leveraging on Blackwell's approachability theorem, Zhong et al. (2018) developed an online allocation policy to guarantee the fill-rate requirement asymptotically. Liang et al. (2023) applied Blackwell's approachability theorem in a passenger-flow control problem with the goal of providing egalitarian transport service for passengers on different origin-destination pairs. Lyu et al. (2019) extended the results in Zhong et al. (2018) to a production network setting and used the OCO technique to establish a non-asymptotic performance guarantee for their online allocation policy. Along this direction, Jiang et al. (2023) provided an integrated framework to address the resource allocation problem with different types of service-level constraints. They showed that a (randomized) max-weighted-service policy is optimal asymptotically. Furthermore, we note that Agrawal and Devanur (2015) studied a generic online stochastic optimization problem with concave objective function and feasibility constraints. They designed an efficient algorithm to solve the problem and demonstrated a near-optimal performance guarantee in regret when the stochastic models are either stationary or random permutation. One of the key differences between our work and the aforementioned studies is that we focus on exploring the properties of online solutions in the case in which KPI targets (e.g., the fill-rate target in Zhong et al. (2018)) are not attainable. We link this problem to the stochastic multi-objective optimization problem and demonstrate that our policy generates the compromise solution to this multi-objective problem. In addition, we derive closed-form policies when the penalty functions are ℓ_1, ℓ_2, and ℓ_∞ norms, and hence the corresponding policies can be efficiently applied to solve large-scale problems. Finally, we generalize our results to a multi-period stochastic setting with multiple stages of recourse in which the stochastic environments are allowed to be non-stationary and endogenous across different periods.

Multi-objective optimization has an important application in the ride-sourcing markets. In practice, the matching decisions between drivers and passengers are supposed to take the trade-offs among

different objectives (or interests of managers from different departments) into account. Although much effort has been devoted to the design of diverse matching policies for the two-sided market, the majority of these works focused on single-objective optimization problems. For example, Zhang et al. (2017) developed a batch matching system, with the objective to maximize the driver acceptance rate for each order. Different from the traditional one-order-to-one-driver matching mechanism, they dispatched each order to multiple drivers and let drivers compete for the order. Leveraging on the forecast of demand and supply patterns, Xu et al. (2018) developed an MDP-based order dispatch algorithm to maximize certain reward functions. Wang et al. (2018) introduced the concept of stability in dynamic ride-sharing and provided mathematical programming approaches to solve stable and nearly stable ride-share matching problems, with the objective of minimizing the pick-up detour distance. Braverman et al. (2019) studied the empty-car routing problem, with the objective to maximize the availability of empty cars when passengers request ride-sourcing service. They considered a closed queuing network model and studied a fluid-based optimization problem. In a large market regime, they demonstrated a stationary distribution for the closed queuing network. Özkan and Ward (2020) proposed a linear programming based matching policy that accounts for temporal changing demand and supply and customer patience, with the objective of maximizing the overall number of passengers being served. Hu and Zhou (2022) studied the dynamic matching control of a two-sided, discrete-time matching system in which both the supply and demand may leave the platform if the wait time before getting passengers or drivers is too long, with the objective to maximize the expected total discounted profit. Along this direction, Baccara et al. (2020) proposed a dynamic matching mechanism to balance the trade-off between increasing matching qualities (by waiting for a thicker market) and reducing waiting costs (by performing a quicker matching). Furthermore, Johari et al. (2021) addressed the exploration-exploitation trade-off between a myopic matching to maximize total revenue and learning the type of the suppliers to improve efficiency. Besides these specific matching polices, Feng et al. (2021) examined the impact of matching mechanisms on the efficiency of the ride-hailing system (e.g., reducing the waiting time). Jiang et al. (2021) investigated how behavioral biases affect works' relocation decisions in on-demand platforms. Other relevant studies include Bai et al. (2019), Castro et al. (2020), Ke et al. (2020), Yan et al. (2020), Li et al. (2021b), Zhu et al. (2021), Zhou et al. (2022). One can refer to Wang and Yang (2019), Benjaafar and Hu (2020), Hu (2021), Wang (2022) for a review of related problems in ride-sourcing and transportation-enabled services. However, to our knowledge, few studies shed light on the multi-objective matching policy.

3. Models and Analysis: Single Period Problem

For ease of understanding, we first describe the framework and the main results for a single-period multi-objective optimization problem. We extend the analysis later to a multi-period optimization

problem with recourse, and show how this can be used to model the matching problem in a ride-sourcing platform.

3.1. Problem Description

The decision maker faces the following multi-objective optimization problem in a single-period stochastic environment: Let $\boldsymbol{\omega}$ denote a supply-demand scenario, and $\boldsymbol{x}(\boldsymbol{\omega})$ the matching solution in scenario $\boldsymbol{\omega}$. The multiple performance metrics are denoted by $f_k(\cdot)$, for $k = 1, 2, \ldots, K$. For ease of exposition, we assume that the non-dominated solutions are obtained by solving the following:

$$
\begin{aligned}
(\mathcal{S}_0) \quad & \max_{\boldsymbol{x}(\boldsymbol{\omega})} \left\{ \mathbf{E}_{\boldsymbol{\omega}\sim\Psi}\left[f_1(\boldsymbol{x}(\boldsymbol{\omega}),\boldsymbol{\omega})\right], \mathbf{E}_{\boldsymbol{\omega}\sim\Psi}\left[f_2(\boldsymbol{x}(\boldsymbol{\omega}),\boldsymbol{\omega})\right], \ldots, \mathbf{E}_{\boldsymbol{\omega}\sim\Psi}\left[f_K(\boldsymbol{x}(\boldsymbol{\omega}),\boldsymbol{\omega})\right] \right\} \\
& \text{s.t.} \quad \boldsymbol{x}(\boldsymbol{\omega}) \in \mathcal{X}(\boldsymbol{\omega}), \ \forall \boldsymbol{\omega} \in \boldsymbol{\Omega},
\end{aligned}
$$

where the set $\mathcal{X}(\boldsymbol{\omega})$ is a convex feasible region associated with the scenario realization $\boldsymbol{\omega}$, and the maximization is over a vector of performance metrics.

The scenario $\boldsymbol{\omega}$ is randomly generated from distribution Ψ, which has a potentially infinite support set $\boldsymbol{\Omega}$. Moreover, the decision rule $\boldsymbol{x}(\boldsymbol{\omega})$ is made upon the realization of $\boldsymbol{\omega}$. For example, the scenario realization $\boldsymbol{\omega}$ could represent the driver-passenger (supply-demand) information revealed in the ride-sourcing market, and the set $\mathcal{X}(\boldsymbol{\omega})$ contains all the feasible matching solutions.

There are K ($K \geq 2$) objectives to be maximized. For each $\boldsymbol{\omega} \in \boldsymbol{\Omega}$, the function $f_k(\cdot, \boldsymbol{\omega}) : \mathcal{X}(\boldsymbol{\omega}) \to \mathbb{R}$ represents the k^{th} KPI objective function under the scenario $\boldsymbol{\omega}$. We write $\boldsymbol{f} := (f_1, f_2, \ldots, f_K)$ in short. Furthermore, we do not restrict any formal conditions on the objective functions, but assume that the optimization problem $\max_{\boldsymbol{x}\in\mathcal{X}(\boldsymbol{\omega})} \boldsymbol{c}^\top \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\omega})$ has a global optimal solution for any $\boldsymbol{c} \in \mathbb{R}^K_{\geq 0}$, $\boldsymbol{\omega} \in \Omega$, with tie-breaking arbitrarily whenever there are multiple optimal solutions. Clearly, this assumption holds for any concave objective function $f_k(\cdot, \boldsymbol{\omega})$.

To synthesize and integrate the performance over multiple dimensions, we formulate a new problem ($\mathcal{S}$-ϕ_p), which seeks a solution $\boldsymbol{x}(\boldsymbol{\omega})$ that achieves a KPI target $\boldsymbol{\tau} = (\tau_1, \tau_2, \ldots, \tau_K) \in \mathbb{R}^K$ closest (in a sense defined by the ℓ_p norm induced penalty function ϕ_p) to the *utopia point* $\boldsymbol{U} \in \mathbb{R}^K$:

$$
\begin{aligned}
(\mathcal{S}\text{-}\phi_p) \quad & \min_{\boldsymbol{x}(\boldsymbol{\omega}),\boldsymbol{\tau}} \ \phi_p(\boldsymbol{U} - \boldsymbol{\tau}) \\
& \text{s.t.} \quad \mathbf{E}_{\boldsymbol{\omega}\sim\Psi}\left[f_k(\boldsymbol{x}(\boldsymbol{\omega}),\boldsymbol{\omega})\right] \geq \tau_k, \ k = 1, 2, \ldots, K, \qquad (1) \\
& \qquad \boldsymbol{x}(\boldsymbol{\omega}) \in \mathcal{X}(\boldsymbol{\omega}), \ \forall \boldsymbol{\omega} \in \boldsymbol{\Omega}.
\end{aligned}
$$

The utopia point $\boldsymbol{U} = (U_1, U_2, \ldots, U_K)$ is defined as follows. For each k, the term U_k is the expected optimal value of problem $(\mathcal{S}_0)$ when $\mathbf{E}[f_k(\boldsymbol{x}(\boldsymbol{\omega}),\boldsymbol{\omega})]$ is the only objective function to be maximized.

The penalty function $\phi_p : \mathbb{R}^K \to \mathbb{R}_{\geq 0}$, which is induced by the ℓ_p norm, is defined as $\phi_p(\boldsymbol{w}) := \|\boldsymbol{w}^+\|_p = \left(\sum_{k=1}^K |w_k^+|^p\right)^{1/p}$ for $p \in [1, \infty]$. It is clearly that ϕ_p is a convex function with the properties

that $\phi_p(\boldsymbol{w}) = 0$ for all $\boldsymbol{w} \in \mathbb{R}^K_{\leq 0}$, and ϕ_p is non-decreasing in each of the K dimensions. Consequently, ϕ_p imposes a penalization whenever the attained KPI target τ_k is less than the corresponding utopia point value U_k for some $k \in \{1, 2, \ldots, K\}$.

A natural question is whether there exists a set of *static* weight functions such that the optimal targets τ^* to ($\mathcal{S}$-ϕ_p) can be satisfied for all k using the traditional weighted-sum policy by merging the multiple objectives into a single one. Unfortunately, the following Example 1 provides a negative answer.

EXAMPLE 1. We consider a ride-matching problem with $C = 4$ drivers, to be allocated to $K = 2$ locations with stochastic demand for rides denoted by $\boldsymbol{\omega} = (\omega_1, \omega_2)$. The demands could be $(3,3)$, $(1,3)$, or $(3,1)$, with equal probability. Here $f_k(\boldsymbol{x}(\boldsymbol{\omega}), \boldsymbol{\omega}) = \mathbb{I}(x_k(\boldsymbol{\omega}) \geq \omega_k)$, where $x_k(\boldsymbol{\omega})$ represents the number of rides served in location k, and $\mathbb{I}(\cdot)$ indicates whether the demand ω_k has been fully served. The utopia point $\boldsymbol{U} = (1,1)$. Given a set of weight functions $\bar{\boldsymbol{z}} = (\bar{z}_1, \bar{z}_2)$, the traditional weighted sum policy can be obtained by solving:

$$\begin{aligned} \max_{(x_1(\boldsymbol{\omega}), x_2(\boldsymbol{\omega}))} \quad & \bar{z}_1 \mathbb{I}(x_1(\boldsymbol{\omega}) \geq \omega_1) + \bar{z}_2 \mathbb{I}(x_2(\boldsymbol{\omega}) \geq \omega_2) \\ \text{s.t.} \quad & x_1(\boldsymbol{\omega}) + x_2(\boldsymbol{\omega}) \leq 4,\ 0 \leq x_1(\boldsymbol{\omega}) \leq \omega_1,\ 0 \leq x_2(\boldsymbol{\omega}) \leq \omega_2, \\ & \boldsymbol{\omega} = (\omega_1, \omega_2) \in \{(3,3), (1,3), (3,1)\}. \end{aligned}$$

The optimal solution to the problem above depends on both the weight vector $(\bar{z}_1, \bar{z}_2)$ and the demand realization (ω_1, ω_2). Next, we consider two static weighted-sum policies:

- Location 1 is always served with a higher priority ($\bar{z}_1 = 1, \bar{z}_2 = 0$). In this case, the served demand is $\boldsymbol{x}(3,3) = (3,1)$, $\boldsymbol{x}(1,3) = (1,3)$, $\boldsymbol{x}(3,1) = (3,1)$. The service levels attained for locations 1 and 2 are $(1, 2/3)$, respectively.
- Location 2 is always served with a higher priority ($\bar{z}_1 = 0, \bar{z}_2 = 1$). In this case, the served demand is $\boldsymbol{x}(3,3) = (1,3)$, $\boldsymbol{x}(1,3) = (1,3)$, $\boldsymbol{x}(3,1) = (3,1)$. The service levels attained for locations 1 and 2 are $(2/3, 1)$, respectively.

Even if both locations are served with equal priority, i.e., $\bar{z}_1 = 1, \bar{z}_2 = 1$, the deterministic solution to the problem ensures that one of the two service levels attained above must hold. So the service level target $(5/6, 5/6)$ cannot be achieved simultaneously. However, if we prioritize location 1 and location 2 with equal probability, i.e., a randomized policy, we can easily verify that the target $(5/6, 5/6)$ can be achieved simultaneously. This is closer to the utopia point $\boldsymbol{U}$. ■

Therefore, it is necessary to go beyond the static weight functions to guarantee that the attained KPI targets stay close to the utopia point in the stochastic environment. To do so, We generate a sequence of i.i.d. scenarios $\{\boldsymbol{\omega}_m\}_{m=1}^M \sim \Psi$, for M sufficiently large, and solve the problem using SAA instead. Let $\boldsymbol{x}_m$ denote the matching decision in sample m. We say that $\boldsymbol{x}_m$ is *non-anticipative* or $\boldsymbol{x}_m$

is *online*, if $\boldsymbol{x}_m$ is determined using only the information $\{\boldsymbol{x}_s, \boldsymbol{\omega}_s\}_{s=1}^{m-1} \cup \{\boldsymbol{\omega}_m\}$ available up to sample m. Equivalently, the decision variable $\boldsymbol{x}_m$ is $\sigma(\{\boldsymbol{x}_s, \boldsymbol{\omega}_s\}_{s=1}^{m-1} \cup \{\boldsymbol{\omega}_m\})$-measurable for all $1 \leq m \leq M$.

In this way, we can reformulate the single-period problem ($\mathcal{S}$-ϕ_p) to the following SAA problem:

$$
\begin{aligned}
(\mathcal{M}\text{-}\phi_p) \min_{\boldsymbol{x}_m, \boldsymbol{\tau}} \; & \phi_p(\boldsymbol{U} - \boldsymbol{\tau}) \\
\text{s.t.} \; & \liminf_{M \to \infty} \frac{1}{M} \left(\sum_{m=1}^{M} f_k(\boldsymbol{x}_m, \boldsymbol{\omega}_m) \right) \geq \tau_k, \; k = 1, 2, \ldots, K, \\
& \boldsymbol{x}_m \in \mathcal{X}(\boldsymbol{\omega}_m), \; m = 1, 2, \ldots \\
& \boldsymbol{x}_m \text{ non-anticipative}, \; m = 1, 2, \ldots
\end{aligned} \tag{2}
$$

In this way, the allocation policy for $\boldsymbol{\omega}_m$ depends on the history $\{\boldsymbol{x}_s, \boldsymbol{\omega}_s\}_{s=1}^{m-1} \cup \{\boldsymbol{\omega}_m\}$ and is therefore a randomized policy, depending on the samples drawn.

3.2. Policy and Compromise Solutions for Problem ($\mathcal{M}$-ϕ_p)

Online Compromise Policy DAO$_p$ for Problem ($\mathcal{M}$-ϕ_p):

We develop a class of online compromise policies—*Debt Associated Online* (DAO$_p$) policy—for the multi-sample problem ($\mathcal{M}$-ϕ_p). At sample $(m+1)$, for each $k \in \{1, 2, \ldots, K\}$ we can track the average debt, $w_k(m+1)$, owe to KPI objective f_k from sample 1 to m:

$$
w_k(m+1) := \frac{1}{m} \sum_{s=1}^{m} D_k(s), \tag{3}
$$

where

$$
D_k(s) := U_k - f_k(\boldsymbol{x}_s^{\text{DAO}_p}, \boldsymbol{\omega}_s)
$$

represents the debt at sample s. The debt vector $\boldsymbol{w}(m+1)$ serves to compare the utopia point with the vector of average objective values obtained by the decision maker from sample 1 to m. Intuitively, when we have $w_k(m+1) > 0$, it means that the previous decisions have not achieved the k^{th} required KPI target U_k over first m samples. Hence, the decision at sample $(m+1)$ should prioritize the maximization of the k^{th} objective $f_k(\cdot, \boldsymbol{\omega}_{m+1})$, in order to stay close to the k^{th} KPI target U_k at the end of sample M. Otherwise, when we have $w_k(m+1) \leq 0$, it suggests a lower priority on the maximization of $f_k(\cdot, \boldsymbol{\omega}_{m+1})$, since the previous decisions have already met the required KPI target U_k. Altogether, debt vector $\boldsymbol{w}(m+1)$ quantifies the relative priority associated with each of the K objectives $f_1(\cdot, \boldsymbol{\omega}_{m+1}), f_2(\cdot, \boldsymbol{\omega}_{m+1}), \ldots, f_K(\cdot, \boldsymbol{\omega}_{m+1})$ at sample $(m+1)$.

Furthermore, in order to accommodate the penalty function ϕ_p, we consider $\boldsymbol{z}(m+1)$, which is the projection of a scaled version of $\boldsymbol{w}^+(m+1)$ onto the unit ball induced by the dual norm of ℓ_p. The projection is motivated by the Nesterov's Smoothening Technique (Nesterov 2005), where $\boldsymbol{z}(m+1)$ is in fact the gradient of a certain smoothed surrogate function of ϕ_p, as detailed in Appendix A.1.

After that, we solve convex program (DAO$_p(m+1)$) involving $\boldsymbol{z}(m+1)$ for the realized scenario in sample $(m+1)$. We summarize the idea in the following description of policy DAO$_p$:

Debt Associated Online (DAO$_p$) policy for penalty function ϕ_p:

For $m=0,1,\ldots$, do:

1. At the start of sample $(m+1)$, compute the debt vector $\boldsymbol{w}(m+1)=(w_k(m+1))_{k=1}^K$ averaged across sample 1 to m, where
$$w_k(m+1)=U_k-\frac{1}{m}\sum_{s=1}^{m} f_k(\boldsymbol{x}_s^{\mathrm{DAO}_p},\boldsymbol{\omega}_s),$$
and $\boldsymbol{U}=(U_k)_{k=1}^K$ is the utopia point for ($\mathcal{M}$-ϕ_p). We also set $\boldsymbol{w}(1)=(1,1,\ldots,1)$ as a starting point for $m=0$.
2. Define $\eta(m+1):=\sqrt{\frac{2\Delta^2}{K^{(q-2)^+/q}(m+1)}}$, where q satisfies $1/p+1/q=1$.
3. Compute the Euclidean projection $\boldsymbol{z}(m+1)$ of $\boldsymbol{w}^+(m+1)$ onto the unit ball defined by $\|\cdot\|_q$:
$$\boldsymbol{z}(m+1)\in \mathrm{argmin}_{\|\boldsymbol{z}\|_q\le 1}\left\|\boldsymbol{z}-\frac{2}{\eta(m+1)}\boldsymbol{w}^+(m+1)\right\|_2^2. \tag{4}$$
4. After observing the realized scenario $\boldsymbol{\omega}_{m+1}$, compute the decision $\boldsymbol{x}_{m+1}^{\mathrm{DAO}_p}$, which is an optimal solution to the following convex program:
$$(\mathrm{DAO}_p(m+1))\ \max_{\boldsymbol{x}_{m+1}}\ \sum\nolimits_{k=1}^{K} z_k(m+1)f_k(\boldsymbol{x}_{m+1},\boldsymbol{\omega}_{m+1})$$
$$\text{s.t.}\quad \boldsymbol{x}_{m+1}\in\mathcal{X}(\boldsymbol{\omega}_{m+1}),$$
where the Euclidean projection $\boldsymbol{z}(\cdot)$ denotes the weight function/vector for different objectives.

In step 2, we define constant $\Delta_k:=\max_{\boldsymbol{\omega}\in\boldsymbol{\Omega},\boldsymbol{x}(\boldsymbol{\omega})\in\mathcal{X}(\boldsymbol{\omega})} f_k(\boldsymbol{x}(\boldsymbol{\omega}),\boldsymbol{\omega})$ for each $k\in\{1,2,\ldots,K\}$, and define constant
$$\Delta:=\sqrt{\sum_{k=1}^{K}\Delta_k^2}. \tag{5}$$
Note that Δ depends only on the ranges of objective functions $f_1,f_2,\ldots,f_K$, but does not depend on the sample size M. Furthermore, the scale factor $\eta(m+1)$ also does not depend on M.

Example 2. To make the discussion clearer, we elaborate on the policy DAO$_p$ when $p\in\{1,2,\infty\}$. For these values, the adaptive weight vector $\boldsymbol{z}(m+1)$ can be computed explicitly.

- When $p=1$, $q=\infty$. In this case, the objective function in step 3 in policy DAO$_1$ can be decoupled dimension-wise. Indeed, it is straightforward to verify that, for each $1\le k\le K$, we have
$$\boldsymbol{z}_k(m+1)=\begin{cases}\frac{2}{\eta(m+1)}\boldsymbol{w}_k^+(m+1) & \text{if } \boldsymbol{w}_k^+(m+1)\le\eta(m+1)/2\\ 1 & \text{if } \boldsymbol{w}_k^+(m+1)>\eta(m+1)/2\end{cases}.$$

- When $p=2$, $q=2$. In this case, $\boldsymbol{z}(m+1)$ is a scalar multiple of $\boldsymbol{w}^+(m+1)$ by (4). Therefore, we can just set $\boldsymbol{z}(m+1)$ to be $\boldsymbol{w}^+(m+1)$, and solve problem ($\text{DAO}_2(m+1)$) to obtain the solution $\boldsymbol{x}_{m+1}^{\text{DAO}_2}$. In addition to a higher efficiency, it is easier to implement policy DAO_2, since it does not require the knowledge of Δ.
- When $p=\infty$, $q=1$. In this case, $\boldsymbol{z}(m+1)$ is constrained to the K-simplex. The required solution $\boldsymbol{z}(m+1)$, a point in $\{\boldsymbol{z}:\sum_{k=1}^K \boldsymbol{z}_k \le 1, \boldsymbol{z}_k \ge 0\}$ closest to $2\boldsymbol{w}^+(m+1)/\eta(m+1)$ in squared Euclidean distance, has the following expression for its k^{th} coordinate, by an argument similar to water-filling:

$$z_k(m+1) = \left[w_k^+(m+1) - c\right]^+,$$

where $c \in [0, \max_{1\le k\le K} w_k^+(m+1)]$ is the smallest number such that $\sum_{k=1}^K \left[w_k^+(m+1)-c\right]^+ \le 1$. The value of c can be found by conducting a binary search on $[0, \max_{1\le k\le K} w_k^+(m+1)]$. The expression for $z_k(m+1)$ signifies that, under the debt penalty function ϕ_∞, the decision maker should focus on the few KPIs with highest debt accumulated between time steps 1 and m, and ignore other KPIs. Such a strategy is in agreement with the nature of penalty function ϕ_∞, which penalizes for the KPI with the most debt. ■

Optimality of Policy DAO_p for Problem ($\mathcal{M}$-ϕ_p):

Let $\boldsymbol{w}^* = (w_1^*, w_2^*, \ldots, w_K^*)$, where $w_k^* = U_k - \mathbf{E}_{\boldsymbol{\omega}\sim\Psi}[f_k(\boldsymbol{x}^*(\boldsymbol{\omega}), \boldsymbol{\omega})]$, and $\{\boldsymbol{x}^*(\boldsymbol{\omega})\}_{\boldsymbol{\omega}\in\boldsymbol{\Omega}}$ is an optimal compromise solution to problem ($\mathcal{S}$-ϕ_p). The following proposition states that our online policy DAO_p attains the compromise solution with near-optimal performance in regret, which is defined as the additive optimality gap from achieved performance to the hindsight optimal performance.

PROPOSITION 1. *Consider the sample-average-approximate problem ($\mathcal{M}$-ϕ_p), and let $q \in [1,\infty]$ satisfy $1/p + 1/q = 1$. The average debt $\boldsymbol{w}(M+1)$ under policy DAO_p satisfies*

$$\mathbf{E}\left[\phi_p(\boldsymbol{w}(M+1))\right] - \phi_p(\boldsymbol{w}^*) \le \sqrt{\frac{8K^{\frac{(q-2)^+}{q}}\Delta^2}{M}}. \tag{6}$$

The expectation in (6) is taken over $\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \ldots, \boldsymbol{\omega}_m$. Proposition 1 implies that the gap from achieved performance under policy DAO_p to the hindsight optimal performance converges to 0 when the sample size M becomes sufficiently large. In fact, Proposition 1 also holds when we replace the utopia point $\boldsymbol{U}$ with any other KPI target, and our policy returns the solution with the minimal ℓ_p-distance to the corresponding target.

We make the following two remarks about policy DAO_p. First, policy DAO_p is indeed non-anticipatory, since the debt vector $\boldsymbol{w}(m+1)$ is calculated based on the debt information from sample 1 to m. Second, while the decision in sample $(m+1)$ requires the knowledge of scenario realization $\boldsymbol{\omega}_{m+1}$, the debt vector $\boldsymbol{w}(m+1)$ is *independent* of $\boldsymbol{\omega}_{m+1}$. Such an independence turns out to facilitate the solution of problem ($\mathcal{S}$-ϕ_p), as we proceed to discuss next.

3.3. Policy and Compromise Solutions for ($\mathcal{S}$-ϕ_p)

In this subsection, we demonstrate that policy DAO$_p$, which is designed for the SAA problem ($\mathcal{M}$-ϕ_p), can be used to compute a succinct and near optimal solution oracle to solve the single-period stochastic problem ($\mathcal{S}$-ϕ_p) for any KPI target $\boldsymbol{U}$. More precisely, we construct a solution $\{\boldsymbol{x}^{\mathcal{S}\text{-}\phi_p}(\boldsymbol{\omega})\}_{\boldsymbol{\omega}\in\boldsymbol{\Omega}}$ which nearly satisfies the first set of attainability constraints and completely satisfies the second set of constraints in ($\mathcal{S}$-ϕ_p).

An immediate challenge in solving problem ($\mathcal{S}$-ϕ_p) is that a feasible solution $\{\boldsymbol{x}(\boldsymbol{\omega})\}_{\boldsymbol{\omega}\in\boldsymbol{\Omega}}$ is a menu of decisions, and the length of the menu increases linearly with $|\boldsymbol{\Omega}|$. Worse still, when $\boldsymbol{\Omega}$ is infinite, such a menu of decisions appears impossible to be specified. We use the property that debt vector $\boldsymbol{w}(m)$ in policy DAO$_p$ is independent of $\boldsymbol{\omega}_m$ to construct a succinct representation of a menu of decisions for ($\mathcal{S}$-ϕ_p), even when $\boldsymbol{\Omega}$ is infinite. The succinct representation is in the form of an oracle, which generates an allocation $\boldsymbol{x}^{\mathcal{S}\text{-}\phi_p}(\boldsymbol{\omega})\in\mathcal{X}(\boldsymbol{\omega})$ for any given scenario realization $\boldsymbol{\omega}\in\boldsymbol{\Omega}$. We provide the construction of such an oracle in Algorithm 1.

Algorithm 1 Randomized Mechanism for Problem ($\mathcal{S}$-ϕ_p).

1: INPUT: A sufficiently large M.
2: Generate i.i.d. scenario realizations $\boldsymbol{\omega}_1,\boldsymbol{\omega}_2,\ldots,\boldsymbol{\omega}_M\sim\Psi$.
3: Perform policy (DAO$_p$) progressively on $\boldsymbol{\omega}_1,\boldsymbol{\omega}_2,\ldots,\boldsymbol{\omega}_M$, which generates weight vectors $\{\boldsymbol{z}(m)\}_{m=1}^{M}$.
4: Sample a period index $\bar{m}$ uniformly at random from $\{1,2,\ldots,M\}$, and let $\bar{\boldsymbol{z}}=\boldsymbol{z}(\bar{m})$.
5: Construct the following oracle: For any given $\boldsymbol{\omega}\in\boldsymbol{\Omega}$, the oracle generates allocation $\boldsymbol{x}^{\mathcal{S}\text{-}\phi_p}(\boldsymbol{\omega})$, which is an optimal solution to the following convex program:

$$\max \sum\nolimits_{k=1}^{K} \bar{z}_k f_k(\boldsymbol{x}(\boldsymbol{\omega}),\boldsymbol{\omega})$$

$$\text{s.t.}\quad \boldsymbol{x}(\boldsymbol{\omega})\in\mathcal{X}(\boldsymbol{\omega}).$$

THEOREM 1. *Consider the single-period stochastic problem ($\mathcal{S}$-ϕ_p). By performing Algorithm 1 with $M=\lceil 8\Delta^2/\epsilon^2\rceil$, the output oracle $\{\boldsymbol{x}^{\mathcal{S}\text{-}\phi_p}(\boldsymbol{\omega})\}_{\boldsymbol{\omega}\in\boldsymbol{\Omega}}$ is feasible for problem ($\mathcal{S}$-ϕ_p), and has an objective value within an additive error $\sqrt{\frac{8K^{\frac{(q-2)^+}{q}}\Delta^2}{M}}$ away from the optimum of problem ($\mathcal{S}$-ϕ_p), where q satisfies $1/p+1/q=1$.*

The proof for Theorem 1 is deferred to Appendix A.2. To gain more insight into our approach, we provide next a characterization of the set of attainable KPI targets $\boldsymbol{\tau}\in\mathbb{R}^K$ such that there exists a decision rule $\{\boldsymbol{x}(\boldsymbol{\omega})\}_{\boldsymbol{\omega}\in\boldsymbol{\Omega}}$ that ensures the feasibility of constraints (1). In other words, we derive a

set of necessary and sufficient conditions for a KPI target $\boldsymbol{\tau}$ to be attainable by a certain decision rule $\{\boldsymbol{x}(\boldsymbol{\omega})\}_{\boldsymbol{\omega}\in\Omega}$. We provide the main result in Theorem 2 and lay out the proof in Appendix A.3.

THEOREM 2. *Consider the single-period stochastic problem* ($\mathcal{S}$-ϕ_p). *A KPI target* $\boldsymbol{\tau}$ *is attainable under a certain decision rule if and only if conditions (7) hold:*

$$\mathbf{E}_{\boldsymbol{\omega}\sim\Psi}\left[\max_{\boldsymbol{x}(\boldsymbol{\omega})}\left\{\sum_{k=1}^{K}\theta_k f_k(\boldsymbol{x}(\boldsymbol{\omega}),\boldsymbol{\omega}) : \boldsymbol{x}(\boldsymbol{\omega})\in\mathcal{X}(\boldsymbol{\omega})\right\}\right] \geq \sum_{k=1}^{K}\theta_k\tau_k,\ \forall\boldsymbol{\theta}\in\mathbb{R}^K_{\geq 0}. \tag{7}$$

In general, it is not easy to obtain closed-form expressions for the set of conditions in (7). However, our randomized decision rule can be used as a heuristic to examine the attainability of any KPI target $\boldsymbol{\tau}$—If the Euclidean norm of the corresponding debt vector under solution $\{\boldsymbol{x}^{\mathcal{S}\text{-}\phi_p}(\boldsymbol{\omega})\}_{\boldsymbol{\omega}\in\boldsymbol{\Omega}}$ would not converge to 0 as M increases, it means that the KPI target cannot be attained by any decision rule.

4. Multi-Period Stochastic Environment

We further generalize our analysis to a multi-period setting with multiple stages of recourse, in which the decision maker needs to determine resource allocation in each time period $t\in\{1,2,\ldots,T\}$. With a slight abuse of notation, we denote the scenario realized at period t as $\boldsymbol{\omega}^t$, which follows the probability distribution Ψ^t and belongs to a potentially infinite scenario set $\boldsymbol{\Omega}^t$. For notational convenience, we represent $\boldsymbol{\omega}^{1:T}=(\boldsymbol{\omega}^1,\boldsymbol{\omega}^2,\ldots,\boldsymbol{\omega}^T)$ as the realized scenarios during the entire planning horizon. We let $\boldsymbol{x}^t$ and $\boldsymbol{y}^t$ denote, respectively, the decision variable and state variable at time period t. The state space is denoted by $\mathbb{Y}$; i.e., $\boldsymbol{y}^t\in\mathbb{Y}$, $\forall t=1,2,\ldots,T$. In the stochastic environment, the state variable $\boldsymbol{y}^t$ randomly evolves based on $\boldsymbol{x}^{t-1}$, $\boldsymbol{\omega}^{t-1}$, and $\boldsymbol{y}^{t-1}$. We write

$$\boldsymbol{y}^t := \mathcal{Y}(\boldsymbol{x}^{t-1},\boldsymbol{\omega}^{t-1},\boldsymbol{y}^{t-1}),\ t=2,3,\ldots,T \tag{8}$$

to encode the evolution of the stochastic system, where the function $\mathcal{Y}(\cdot)$ maps the state and decision at time period $t-1$ to period t. For convenience, we denote $\boldsymbol{y}^1:=\mathcal{Y}(\boldsymbol{x}^0,\boldsymbol{\omega}^0,\boldsymbol{y}^0)$ as the initial state revealed at period 1. Analogous to the single-period problem ($\mathcal{S}$-ϕ_p), we formulate a multi-period stochastic problem that seeks a decision rule $\boldsymbol{x}^t:=\boldsymbol{x}^t(\boldsymbol{y}^t,\boldsymbol{\omega}^t)\in\mathcal{X}(\boldsymbol{y}^t,\boldsymbol{\omega}^t)$ that achieves the KPI target $\boldsymbol{\tau}=(\tau_1,\tau_2,\ldots,\tau_K)\in\mathbb{R}^K$ closest to the utopia point $\boldsymbol{U}\in\mathbb{R}^K$.

$$\begin{aligned}
(\mathcal{G}\text{-}\mathcal{S}\text{-}\phi_p)\quad \min_{\boldsymbol{x}^t,\boldsymbol{y}^t,\boldsymbol{\tau}}\ & \phi_p(\boldsymbol{U}-\boldsymbol{\tau})\\
\text{s.t.}\quad & \mathbf{E}\left[\sum_{t=1}^{T} f_k(\boldsymbol{x}^t,\boldsymbol{y}^t,\boldsymbol{\omega}^t)\right]\geq\tau_k,\ k=1,2,\ldots,K,\\
& \boldsymbol{y}^t=\mathcal{Y}(\boldsymbol{x}^{t-1},\boldsymbol{\omega}^{t-1},\boldsymbol{y}^{t-1}),\ \forall\boldsymbol{\omega}^{t-1}\in\boldsymbol{\Omega}^{t-1},\ t=1,2,\ldots,T,\\
& \boldsymbol{x}^t\in\mathcal{X}(\boldsymbol{y}^t,\boldsymbol{\omega}^t),\ \forall\boldsymbol{\omega}^t\in\boldsymbol{\Omega}^t,\ t=1,2,\ldots,T,
\end{aligned} \tag{9}$$

where $f_k(\boldsymbol{x}^t, \boldsymbol{y}^t, \boldsymbol{\omega}^t)$ denotes the objective value for each objective k at time period t. The expectation in Equation (9) is taken over $\boldsymbol{\omega}^{1:T}$. The utopia target U_k is defined as the expected optimal value when $\mathbf{E}\left[\sum_{t=1}^{T} f_k(\boldsymbol{x}^t, \boldsymbol{\omega}^t, \boldsymbol{y}^t)\right]$ is the only objective function to maximize. Clearly, this multi-period problem ($\mathcal{G}$-$\mathcal{S}$-ϕ_p) is a stochastic dynamic programming problem. It is challenging to solve to the optimum even when there is only one objective.

To solve the model, similar to Section 3, we also reformulate problem ($\mathcal{G}$-$\mathcal{S}$-ϕ_p) to a related SAA problem by generating a branch of scenarios $\{\boldsymbol{\omega}_m^t\}_{m=1}^M \sim \boldsymbol{\omega}^t$ (for each and every time period $t = 1, 2, \ldots, T$) over M samples. We use the superscript t to indicate the period index, while subscript m to indicate the sample epoch throughout the paper. Note that the scenarios $\{\boldsymbol{\omega}_m^1, \boldsymbol{\omega}_m^2, \ldots, \boldsymbol{\omega}_m^T\}$ are allowed to be correlated on the same sample path m, while the scenarios $\{\boldsymbol{\omega}_1^t, \boldsymbol{\omega}_2^t, \ldots, \boldsymbol{\omega}_M^t\}$ for the same time index t are i.i.d. We denote by $\{\boldsymbol{x}_m^t, \boldsymbol{y}_m^t\}$ the decision variable and state variable respectively at period t on sample path m, and we force $\{\boldsymbol{x}_m^t, \boldsymbol{y}_m^t\}$ to be non-anticipative so that $\{\boldsymbol{x}_m^t, \boldsymbol{y}_m^t\}$ can only use the information available up to period t on sample path m as well as the information from sample 1 to $m-1$. In this way, we can formulate:

$$
\begin{aligned}
(\mathcal{G}\text{-}\mathcal{M}\text{-}\phi_p) \quad \min_{\boldsymbol{x}_m^t, \boldsymbol{y}_m^t, \boldsymbol{\tau}} \quad & \phi_p(\boldsymbol{U} - \boldsymbol{\tau}) \\
\text{s.t.} \quad & \liminf_{M\to\infty} \frac{1}{M} \sum_{m=1}^{M} \left(\sum_{t=1}^{T} f_k(\boldsymbol{x}_m^t, \boldsymbol{y}_m^t, \boldsymbol{\omega}_m^t) \right) \geq \tau_k, \ k = 1, 2, \ldots, K, \\
& \boldsymbol{y}_m^t = \mathcal{Y}(\boldsymbol{x}_m^{t-1}, \boldsymbol{\omega}_m^{t-1}, \boldsymbol{y}_m^{t-1}), \ t = 1, 2, \ldots, T, \ m = 1, 2, \ldots \\
& \boldsymbol{x}_m^t \in \mathcal{X}(\boldsymbol{y}_m^t, \boldsymbol{\omega}_m^t), \ t = 1, 2, \ldots, T, \ m = 1, 2, \ldots \\
& \boldsymbol{x}_m^t \text{ non-anticipative}, \ t = 1, 2, \ldots, T, \ m = 1, 2, \ldots
\end{aligned}
$$

Similarly, define constant $\Delta_k^T := \max_{\boldsymbol{x}^t, \boldsymbol{y}^t} \sum_{t=1}^{T} f_k(\boldsymbol{x}^t, \boldsymbol{y}^t, \boldsymbol{\omega}^t)$ for any $\boldsymbol{\omega}^t \in \boldsymbol{\Omega}^t, \ t = 1, 2, \ldots, T$, and define constant

$$
\Delta^T := \sqrt{\sum_{k=1}^{K} (\Delta_k^T)^2}. \tag{10}
$$

Here we use the superscript T to indicate the dependence of Δ^T on the planning horizon of problem ($\mathcal{G}$-$\mathcal{M}$-ϕ_p), but Δ^T is a constant independent of sampling size M. We extend the class of online policies $\{\text{DAO}_p\}_{p\in[1,\infty]}$ to solve problem ($\mathcal{G}$-$\mathcal{M}$-ϕ_p), as detailed below.

Generalized Debt Associated Online policy ($\mathcal{G}$-DAO$_p$):

For $m = 0, 1, \ldots$, do the following:

1. At the start of sample $(m+1)$, compute the debt vector $\boldsymbol{w}(m+1) = (w_k(m+1))_{k=1}^K$ averaged across sample 1 to m, where

$$
w_k(m+1) = U_k - \frac{1}{m} \sum_{s=1}^{m} \left\{ \sum_{t=1}^{T} f_k\big(\boldsymbol{x}_s^{\mathcal{G}\text{-DAO}_p, t}, \boldsymbol{y}_s^{\mathcal{G}\text{-DAO}_p, t}, \boldsymbol{\omega}_s^t\big) \right\},
$$

and $\boldsymbol{U} = (U_k)_{k=1}^K$ is the utopia point for problem ($\mathcal{G}$-$\mathcal{M}$-ϕ_p).

2. Define $\eta(m+1) := \sqrt{\frac{2(\Delta^T)^2}{K^{(q-2)^+/q}(m+1)}}$, where q satisfies $1/p+1/q=1$.
3. Compute the Euclidean projection $\boldsymbol{z}(m+1)$ of $\boldsymbol{w}^+(m+1)$ onto the unit ball defined by $\|\cdot\|_q$:

$$\boldsymbol{z}(m+1) \in \operatorname{argmin}_{\|\boldsymbol{z}\|_q \le 1} \left\| \boldsymbol{z} - \frac{2}{\eta(m+1)} \boldsymbol{w}^+(m+1) \right\|_2^2. \tag{11}$$

4. Compute the decision $\{\boldsymbol{x}_{m+1}^{\mathcal{G}\text{-DAO}_p,t}, \boldsymbol{y}_{m+1}^{\mathcal{G}\text{-DAO}_p,t}\}_{t=1}^T$ by solving the stochastic dynamic programming:

$$\begin{aligned}
(\mathcal{G}\text{-DAO}_p(m+1)) \ \max \ & \sum_{k=1}^K z_k(m+1) \mathbf{E}\left[\sum_{t=1}^T f_k\big(\boldsymbol{x}_{m+1}^t, \boldsymbol{y}_{m+1}^t, \boldsymbol{\omega}_{m+1}^t\big)\right] \\
\text{s.t.} \ \ & \boldsymbol{y}_{m+1}^t = \mathcal{Y}(\boldsymbol{x}_{m+1}^{t-1}, \boldsymbol{y}_{m+1}^{t-1}, \boldsymbol{\omega}_{m+1}^{t-1}), \ \forall \boldsymbol{\omega}_{m+1}^{t-1} \in \boldsymbol{\Omega}^{t-1}, \ t=1,2,\ldots,T, \\
& \boldsymbol{x}_{m+1}^t \in \mathcal{X}(\boldsymbol{y}_{m+1}^t, \boldsymbol{\omega}_{m+1}^t), \ \forall \boldsymbol{\omega}_{m+1}^t \in \boldsymbol{\Omega}^t, \ t=1,2,\ldots,T.
\end{aligned}$$

   The k^{th} objective value at period t is denoted by $f_k\big(\boldsymbol{x}_{m+1}^{\mathcal{G}\text{-DAO}_p,t}, \boldsymbol{y}_{m+1}^{\mathcal{G}\text{-DAO}_p,t}, \boldsymbol{\omega}_{m+1}^t\big)$ under the $\mathcal{G}$-DAO$_p$ policy for $t=1,2,\ldots,T$.

At Step 4 we need to solve a stochastic dynamic programming ($\mathcal{G}$-DAO$_p(m+1)$); i.e., the decision $(\boldsymbol{x}_{m+1}^{\mathcal{G}\text{-DAO}_p,t}, \boldsymbol{y}_{m+1}^{\mathcal{G}\text{-DAO}_p,t})$ at period t is made upon realization of scenario $\boldsymbol{\omega}_{m+1}^t$, while the "future" information $(\boldsymbol{\omega}_{m+1}^{t+1}, \boldsymbol{\omega}_{m+1}^{t+2}, \ldots, \boldsymbol{\omega}_{m+1}^T)$ is not revealed to the decision maker. Indeed, this is a dynamic programming problem with a weighted-sum objective function. For ease of exposition, we suppress discussion of how to solve problem ($\mathcal{G}$-DAO$_p(m+1)$) to optimality, but focus on providing transparency into the rules used to solve this multi-objective stochastic problem.

Next, we assert the near optimality of policy $\mathcal{G}$-DAO$_p$ for problem ($\mathcal{G}$-$\mathcal{M}$-ϕ_p), by comparing the average debt vector $\boldsymbol{w}(m+1)$ with an debt vector $\boldsymbol{w}^*$ optimal for the multi-period stochastic problem ($\mathcal{G}$-$\mathcal{S}$-ϕ_p). Vector $\boldsymbol{w}^* = (w_1^*, w_2^*, \ldots, w_K^*)$ is defined as $w_k^* = U_k - \sum_{t=1}^T \mathbf{E}[f_k(\boldsymbol{x}^{*,t}, \boldsymbol{y}^{*,t}, \boldsymbol{\omega}^t)]$, where $\{\boldsymbol{x}^{*,t}, \boldsymbol{y}^{*,t}\}_{t=1}^T$ represents the optimal solution to problem ($\mathcal{G}$-$\mathcal{S}$-ϕ_p). In addition, recall the definition of constant Δ^T, which is independent of sample size M in Equation (10). We demonstrate the sublinear regret bound for the online policy $\mathcal{G}$-DAO$_p$ in Proposition 2.

Proposition 2. *Consider the sample-average-approximate problem ($\mathcal{G}$-$\mathcal{M}$-ϕ_p) and let $q \in [1,\infty]$ satisfy $1/p+1/q=1$. The average debt $\boldsymbol{w}(M+1)$ under policy $\mathcal{G}$-DAO$_p$ satisfies*

$$\mathbf{E}\left[\phi_p(\boldsymbol{w}(M+1))\right] - \phi_p(\boldsymbol{w}^*) \le \sqrt{\frac{8K^{\frac{(q-2)^+}{q}}(\Delta^T)^2}{M}}. \tag{12}$$

To prove Proposition 2, we treat the multi-period stochastic problem as a "single" problem and suppress discussion of its dynamic structure. Since the scenarios $\{\boldsymbol{\omega}_m^t\}_{m=1}^M$ are i.i.d. generated across different samples, we apply online convex optimization techniques to address the aggregated problem $\sum_{t=1}^T f_k\big(\boldsymbol{x}_m^t, \boldsymbol{y}_m^t, \boldsymbol{\omega}_m^t\big)$ on each sample path m, instead of a single-period problem $f_k\big(\boldsymbol{x}_m^t, \boldsymbol{y}_m^t, \boldsymbol{\omega}_m^t\big)$. We claim that the solution under policy $\mathcal{G}$-DAO$_p$ converges to the hindsight compromise solution as

long as the optimal solution to problem ($\mathcal{G}$-DAO$_p(m)$) could be obtained. The remaining proof is similar to the proof for Proposition 1 and is relegated to Appendix A.4.

Furthermore, similar to Section 3.3, we cast online policy $\mathcal{G}$-DAO$_p$ into a randomized mechanism to solve the stochastic problem ($\mathcal{G}$-$\mathcal{S}$-ϕ_p). The randomized mechanism is described in Algorithm 2.

Algorithm 2 Randomized Mechanism for the Multi-period Stochastic Problem ($\mathcal{G}$-$\mathcal{S}$-ϕ_p).

1: INPUT: The utopia KPI target $\boldsymbol{U} \in \mathbf{R}^K$ and a sufficiently large M.

2: Generate i.i.d. scenario realizations $\{\boldsymbol{\omega}_1^t, \boldsymbol{\omega}_2^t, \ldots, \boldsymbol{\omega}_M^t\} \sim \Psi^t$ for all $t = 1, 2, \ldots, T$.

3: Perform policy ($\mathcal{G}$-DAO$_p$) progressively on $\{\boldsymbol{\omega}_m^1, \boldsymbol{\omega}_m^2, \ldots, \boldsymbol{\omega}_m^T\}$, which generates weight vectors $\boldsymbol{z}(m)$, for all samples $m = 1, 2, \ldots, M$.

4: Construct the following oracle: Sample a period index $\bar{m}$ uniformly at random from $\{1, 2, \ldots, M\}$ with replacement, and let $\bar{\boldsymbol{z}} = \boldsymbol{z}(\bar{m})$. The oracle generates a sequence of decisions $\{\boldsymbol{x}^{\mathcal{G}\text{-}\mathcal{S}\text{-}\phi_p,t}, \boldsymbol{y}^{\mathcal{G}\text{-}\mathcal{S}\text{-}\phi_p,t}\}_{t=1}^T$, which is an optimal solution to the following stochastic dynamic programming:

$$
\begin{aligned}
\max \ & \sum_{k=1}^{K} \bar{z}_k \mathbf{E}\left[\sum_{t=1}^{T} f_k\big(\boldsymbol{x}^t, \boldsymbol{y}^t, \boldsymbol{\omega}^t\big)\right] \\
\text{s.t.} \ & \boldsymbol{y}^t = \mathcal{Y}(\boldsymbol{x}^{t-1}, \boldsymbol{\omega}^{t-1}, \boldsymbol{y}^{t-1}), \ \forall \boldsymbol{\omega}^t \in \boldsymbol{\Omega}^t, \ t = 1, 2, \ldots, T, \\
& \boldsymbol{x}^t \in \mathcal{X}(\boldsymbol{y}^t, \boldsymbol{\omega}^t), \ \forall \boldsymbol{\omega}^t \in \boldsymbol{\Omega}^t, \ t = 1, 2, \ldots, T.
\end{aligned}
$$

Finally, we demonstrate the desired performance guarantee for the randomized mechanism ($\mathcal{G}$-$\mathcal{S}$-ϕ_p), as summarized in Theorem 3. The proof for Theorem 3 is relegated to Appendix A.5.

THEOREM 3. *Consider the multi-period stochastic problem ($\mathcal{G}$-$\mathcal{S}$-ϕ_p). By performing Algorithm 2 with $M = \lceil 8(\Delta^T)^2/\epsilon^2 \rceil$, the output oracle $\{\boldsymbol{x}^{\mathcal{G}\text{-}\mathcal{S}\text{-}\phi_p,t}, \boldsymbol{y}^{\mathcal{G}\text{-}\mathcal{S}\text{-}\phi_p,t}\}_{t=1}^T$ is feasible for problem ($\mathcal{G}$-$\mathcal{S}$-ϕ_p), and has an objective value within an additive error $\sqrt{\frac{8K\frac{(q-2)+}{q}(\Delta^T)^2}{M}}$ away from the optimum of ($\mathcal{G}$-$\mathcal{S}$-ϕ_p), where q satisfies $1/p + 1/q = 1$.*

5. Application: Ride-Sourcing Markets

In this section, we apply the multi-objective stochastic optimization approach to derive what we call a compromise matching policy for the ride-sourcing markets, and investigate the impact of incorporating service quality in the dispatch algorithm. Our computational results confirm that the proposed approach is able to balance the conflicting goals in the multiple targets to obtain a good solution that come close to the ideal performance target for each KPI. More importantly, it allows the platform to earn more revenue, assigns more jobs to better drivers and better drivers to the passengers, for minimial reduction in waiting time performance.

5.1. Problem Setup

We study a multi-period matching problem, in which drivers are dispatched to serve nearby passengers in a batch-matching environment. In addition, the batch-matching interval is predetermined, which is consistent with the industrial practice. At each period $t = 1, 2, \ldots, T$, the matching scenario $(\boldsymbol{y}^t, \boldsymbol{\omega}^t)$ is realized. Here $\boldsymbol{y}^t$ denotes the set of left-over passengers and drivers from period $t-1$, and $\boldsymbol{\omega}^t$ the set of new passengers and idle drivers who join the market at period t. With a slight abuse of notation, $N^p(\boldsymbol{y}^t, \boldsymbol{\omega}^t)$ and $N^d(\boldsymbol{y}^t, \boldsymbol{\omega}^t)$ denote respectively the number of passengers to be served and the number of available drivers at period t. Let $\mathcal{N}(\boldsymbol{y}^t, \boldsymbol{\omega}^t)$ denote the bipartite graph for this two sided market. We allow passenger $i \in \{1, 2, \ldots, N^p(\boldsymbol{y}^t, \boldsymbol{\omega}^t)\}$ to be matched to driver $j \in \{1, 2, \ldots, N^d(\boldsymbol{y}^t, \boldsymbol{\omega}^t)\}$ if the distance $d_{i,j}$ between them is shorter than a tolerance level (e.g., 3 km); i.e., $(i, j) \in \mathcal{N}(\boldsymbol{y}^t, \boldsymbol{\omega}^t)$ if $d_{i,j} \leq 3$ km. Furthermore, we let $\mathcal{X}(\boldsymbol{y}^t, \boldsymbol{\omega}^t)$ denote the feasible region for the matching solutions at period t:

$$\mathcal{X}(\boldsymbol{y}^t, \boldsymbol{\omega}^t) := \left\{ \boldsymbol{x} \in \mathbb{R}^{|N^p(\boldsymbol{y}^t, \boldsymbol{\omega}^t)| \times |N^d(\boldsymbol{y}^t, \boldsymbol{\omega}^t)|} \;\middle|\; \begin{array}{rl} \sum_{i=1}^{N^p(\boldsymbol{y}^t, \boldsymbol{\omega}^t)} x_{i,j} \leq & 1,\ \forall j = 1, 2, \ldots, N^d(\boldsymbol{y}^t, \boldsymbol{\omega}^t) \\ \sum_{j=1}^{N^d(\boldsymbol{y}^t, \boldsymbol{\omega}^t)} x_{i,j} \leq & 1,\ \forall i = 1, 2, \ldots, N^p(\boldsymbol{y}^t, \boldsymbol{\omega}^t) \\ x_{i,j} \in & \{0, 1\},\ \forall (i, j) \in \mathcal{N}(\boldsymbol{y}^t, \boldsymbol{\omega}^t) \\ x_{i,j} = & 0,\ \forall (i, j) \notin \mathcal{N}(\boldsymbol{y}^t, \boldsymbol{\omega}^t) \end{array} \right\},$$

where the first set of constraints implies that one driver can be matched to at most one passenger, and the second set requires that one passenger can be picked up by at most one driver.

In the ride-sourcing market, the platform operator would want to dispatch more jobs to drivers with higher service score. This would help to retain the better drivers in the system, and provide better service experience to the passengers. However, without an appropriate matching policy, this could not come without sacrificing the total revenue obtained, as well as the average pick-up distance between dispatched drivers and passengers. In this section, we formally answer the question: *How to incorporate multiple objectives into the matching algorithm? Specifically, how to incorporate the driver's service quality without sacrificing too much performance on the other key objectives?* To balance the different KPIs, four key considerations need to be taken into account to design the matching policies in these markets:

- **Platform revenue**: Passengers with higher order revenue should be served with higher priority. We denote $r_i(\boldsymbol{y}^t, \boldsymbol{\omega}^t)$ as the revenue earned by serving passenger i at period t.
- **Service quality**: Drivers with better service (measured by service score) are required to be matched with higher priority. We denote $s_j(\boldsymbol{y}^t, \boldsymbol{\omega}^t)$ as the score earned by dispatching available driver

j at period t. We highlight the fact that the service score is a driver-related property and determined by historical trip ratings and driving behavior. The service score is fixed for each driver and does not vary by trip in the case study.

- **Pick-up distance**: Passengers and nearby drivers with shorter pick-up distance should be matched with higher priority. We denote $d_{i,j}(\boldsymbol{y}^t, \boldsymbol{\omega}^t)$ as the pick-up distance between passenger i and driver j at period t. Note that the underlying concern is to maximize total number of matched pairs, and hence we introduce a big number $\hat{d}$, which should be larger than the pick-up distance tolerance benchmark (i.e., $\hat{d} > 3$ km). We denote $\hat{d} - d_{i,j}(\boldsymbol{y}^t, \boldsymbol{\omega}^t)$ as a so-called "saved pick-up distance". In this way, we transform the real pick-up distance minimization problem into a saved pick-up distance maximization problem.
- **Matched pair**: The platform also aims to maximize the number of total matched pairs, i.e., to serve as many passengers as possible.

Before describing specific matching polices, we unify the following notation and terminology. Given any matching scenario $(\boldsymbol{y}, \boldsymbol{\omega})$ and feasible solution $\boldsymbol{x} \in \mathcal{X}(\boldsymbol{y}, \boldsymbol{\omega})$, we let

$$f_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\omega}) := \alpha_1 \left[\sum_{(i,j) \in \mathcal{N}(\boldsymbol{y}, \boldsymbol{\omega})} r_i(\boldsymbol{y}, \boldsymbol{\omega}) x_{i,j} \right], \; f_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\omega}) := \alpha_2 \left[\sum_{(i,j) \in \mathcal{N}(\boldsymbol{y}, \boldsymbol{\omega})} s_j(\boldsymbol{y}, \boldsymbol{\omega}) x_{i,j} \right],$$

$$f_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\omega}) := \alpha_3 \left[\sum_{(i,j) \in \mathcal{N}(\boldsymbol{y}, \boldsymbol{\omega})} \left(\hat{d} - d_{i,j}(\boldsymbol{y}, \boldsymbol{\omega}) \right) x_{i,j} \right], \; f_4(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\omega}) := \alpha_4 \left[\sum_{(i,j) \in \mathcal{N}(\boldsymbol{y}, \boldsymbol{\omega})} x_{i,j} \right]$$

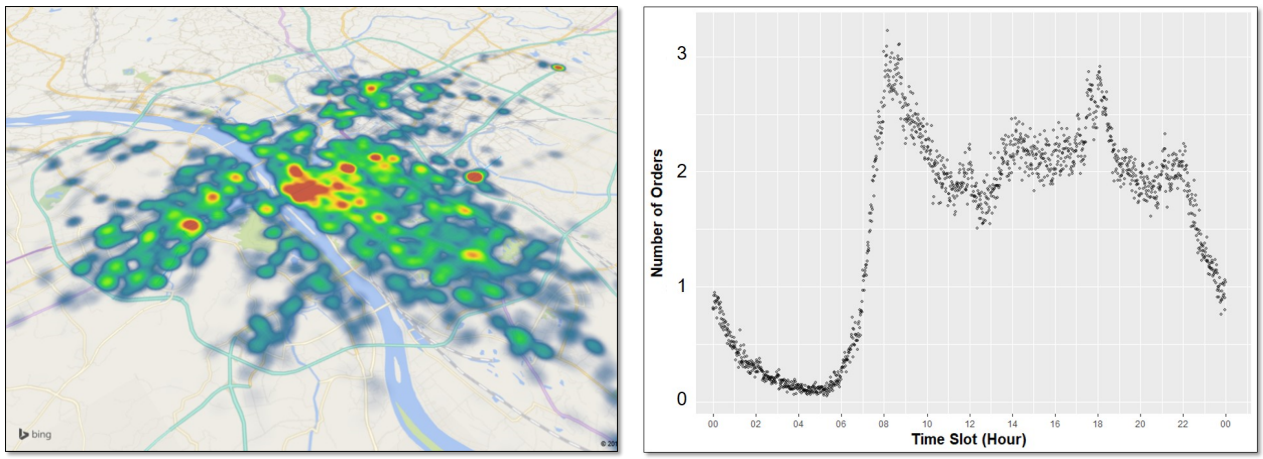
denote total revenue obtained, service score, saved pick-up distance, and the number of matched pairs, respectively. Since the magnitude of the four KPIs are different, we set revenue as the baseline (i.e., we let $\alpha_1 = 1.00$) and normalize the other three KPIs to similar magnitudes using parameters $\{\alpha_2, \alpha_3, \alpha_4\}$. As the mean value of revenue obtained by serving each passenger is 19.79 and the mean value of driver service score is 78.34, we rescale the driver service score by multiplying a factor $\alpha_2 = 19.79/78.34 = 0.25$. We set $\hat{d} = 6$ km and $\alpha_3 = 4.00$ so that the magnitude of saved pick-up distance driver and passenger pair is around 20 on average. Furthermore, we let $\alpha_4 = 20$ to ensure that the normalized number of matched pairs is also around 20. Note that the respective targets for the KPIs will be scaled accordingly based on the choice of α's.

5.2. Dataset Description

We extract real-world data from Didi Chuxing, the largest on-demand ride-sourcing platform in China, which was founded in June 2012. Our dataset contains one week of ride-sourcing records for a mega city in China (abbreviated City C). On the demand side, the dataset for orders contains detailed information on each travel request (by the passenger), including order booking time, latitude and longitude of the trip's origin and destination, revenue, etc. On the supply side, the dataset on

driver-trace contains minute-level tracking information on all active drivers on the platform, including latitude and longitude of the driver's location, his/her service status,[3] etc. The driver service scores are contained in a separate dataset, in which the driver information is hidden due to confidentiality issues. This dataset allows us to calibrate the distribution of the service scores. Overall, the ratio of served passengers is around 85% in the historical records, which implies relatively balanced supply and demand in the ride-sourcing market in City C.

Figure 2 Spatial and temporal distribution of orders. The absolute number of orders is normalized.



(a) Daily Aggregated Spatial Distribution

(b) Daily Temporal Distribution

Figure 2 depicts the spatial and temporal distribution of ride-sourcing orders on the demand side. We observe that the central regions of the city are crowded, with more orders at the daily aggregated level. The region, with peak demand on the eastern side of the city is located at the international airport. In terms of temporal distribution, we find that the morning peak hour, from 8:00 a.m. to 9:00 a.m., has the highest demand. Demand then fluctuates between 10:00 a.m. and 17:00 p.m. During the evening peak hour, demand again climbs until 18:00 p.m., then declines in the late evening and after midnight and hits bottom between 4:00 a.m. and 5:00 a.m. Typically, during off-peak hours (e.g., 15:00 p.m.-16:59 p.m.), the number of orders is relatively stationary. However, during peak hours (e.g., 17:00 p.m.-18:59 p.m.), the number of orders fluctuates significantly.

We develop a ride-sourcing simulator, using a set of real-life order records and driver routing records, to implement different matching policies. More concretely:

[3] The dataset records three service statuses of active drivers: (1) Busy, for drivers who are delivering passengers; (2) Occupied, for drivers who are on their way to pick up passengers; and (3) Idle, for drivers who are not assigned to serve any passenger. If a driver logs out from the platform, his/her status is recorded as inactive, and no passenger requests will be assigned to him/her.

- Passenger order information in each scenario is sampled from the order set. Passengers' patience levels (tolerance for waiting time before being assigned to a driver) are generated uniformly between 60 seconds and 300 seconds.
- Driver status (inactive, idle active, and busy active), routing behavior, and travel time between each specific pair of origins and destinations have been considered in our simulator. For idle active drivers, we perform a random walk to simulate their cruising routes. For busy active drivers, we generate the shortest path to convey passengers from their origin to their destination.
- Driver service scores are randomly assigned from the anonymous service score set. This random assignment captures the actual distribution of the service scores.
- The matching interval is set to be 12 seconds in the simulator.[4] Unmatched drivers and passengers are delayed to the next interval until they leave the market.
- Driver information and passenger order information are bootstrapped from the real dataset for 300 days—i.e., the sample size is set as $M = 300$ in the numerical experiments to examine the performance of different matching policies.
- The ride-sourcing environment is simulated using Java programming language, and the related optimization problems are solved by Gurobi (9.0.2) solver. All agent-based simulations are performed on a 2.70 GHz i7-6820HQ CPU Windows PC with 16GB RAM.

5.3. Ride-Matching Policies

To implement our multi-objective framework, we first obtain an implementable target $\boldsymbol{U} := (U_1, U_2, U_3, U_4)$. Ideally, we can develop a stochastic dynamic programming model for each single objective maximization problem and record the best KPI value as a component of the target vector. However, even such stochastic dynamic programming models for a single objective are intractable due to the curse of dimensionality in the large-scale ride-matching problem. To overcome the computational obstacles, we first apply myopic policies to solve single-objective maximization problems. In fact, our framework also allows the decision maker to estimate the KPI target from historical data for the ease of implementation. Furthermore, we observe that the temporal and spatial matching patterns across different days (samples) are similar. The order time and destination of regular passengers are predictable to some extent (cf. Xu et al. 2018). This allows us to assume that the matching profiles $\{\boldsymbol{\omega}_1^t, \boldsymbol{\omega}_2^t, \ldots, \boldsymbol{\omega}_M^t\}$ across M days are i.i.d. generated from the distribution Ψ^t for each time stamp t, while the matching profiless $\{\boldsymbol{\omega}_m^1, \boldsymbol{\omega}_m^2, \ldots, \boldsymbol{\omega}_m^T\}$ within each day m could be arbitrarily correlated. We set the penalty function as ϕ_2 in model ($\mathcal{G}$-$\mathcal{M}$-ϕ_2) and apply policy $\mathcal{G}$-DAO$_2$ for Algorithm 2 under a multi-period setting (in Section 4) in the ride-matching context, which provides a specific matching algorithm—denoted as the *Compromise Matching* (CM) policy—for the ride-matching problem. The

[4] We conducted experiments with different matching intervals, which present similar performance and insights.

CM policy minimizes the Euclidean distance from the attained performance to the predetermined KPI target.

In the numerical experiments, we compare the performance of the proposed CM policy with two classes of benchmark policies and one simplified version of the CM policy.

1. *Single-objective Maximization* policies: we select revenue maximization (RM) policy, service maximization (SM) policy, saved pick-up time maximization (TM) policy, and pair maximization (PM) policy. Note that the utopia target comprises of the objective values of these four single-objective policies.
2. *Weighted-sum Matching* (WM) policy: we assign a deterministic weight on each objective function. This policy is popular in practice due to its simplicity. Although the weights can be adjusted when the manager observes poor performance on a KPI, it is more like adhoc adjustment without scientific guidance. We test different sets of weights and report the results with identical weights for illustrative purpose.
3. *Simplified Compromise Matching* (S-CM) policy: we apply Algorithm 1 under a single-period setting (in Section 3) in the ride-matching context assuming i.i.d. decision scenarios to derive this policy. It is a purely online policy without using any forecast information of the future. We note that online policies similar to S-CM have been implemented in online advertising (e.g., Agrawal and Devanur 2015, Li et al. 2021a) and supply chain management (e.g., Lyu et al. 2019, 2022, Jiang et al. 2023) with satisfactory numerical performance.

More details about the implementation of the CM policy and benchmark WM and S-CM polices are provided in Appendix B. For both CM policy and single-objective policies, we first evaluate their performance in a myopic setting,[5] in which the matching problem is solved in an online fashion, and then in a forward-looking setting in which demand forecast is incorporated into matching policies. We remark that the multi-period stochastic ride-matching problems are not solved to optimality in both settings due to the curse of dimensionality.

5.4. Numerical Experiments

We examine the performance of different matching policies over $M = 300$ days in the same simulated ride-sourcing environment. In Section 5.4.1, we validate that the CM policy attains the smallest Euclidean distance from the attained KPIs to the performance targets, compared to the benchmark policies. The benefits of the CM policy for different stakeholders are discussed in Section 5.4.2. We perform further tests on the CM policy in Section 5.4.3 to ensure that the performance observed is robust. Finally, we study the value of incorporating demand forecast into ride-matching policies in Section 5.4.4.

[5] We note that myopic matching polices have been widely used in ride-sourcing markets (e.g., Zhang et al. 2017, Wang and Yang 2019) due to the computational challenges in solving large-scale problems in real-time.

5.4.1. Validation of the CM Policy We first solve four single-objective maximization models in a myopic setting, and calculate the daily average KPI performance. As shown in Table 1, we observe that all four policies achieve similar performance regarding the number of matched pairs. More concretely, the ratios of matched passengers under the RM, SM, TM, and PM policies are 84.92%, 84.24%, 85.47%, and 84.92%, respectively. Since we simulate passenger patience level to consider passenger cancellation behavior, passengers are more likely to remain in the market if the pick-up distance is shorter. Therefore, although the PM policy aims to to purely maximize the number of matched pairs, the TM policy instead achieves the highest ratio of matched pairs. The differences in total revenue obtained under the four policies are also marginal, since the supply is sufficient to serve the majority of passengers under all four policies. However, the matching quality will differ significantly on the supply side, since passengers could be served by drivers with different service scores and pick-up distances in a supply-abundant market. Figure 3(a) and 3(b) also clearly show that the SM policy achieves much higher service scores than the other three policies, while the TM policy achieves much better performance on saved pick-up distance. Overall, the single-objective maximization policy performs well in terms of the KPI that is targeted to be maximize, but it suffers from significant performance loss on at least one other KPI.

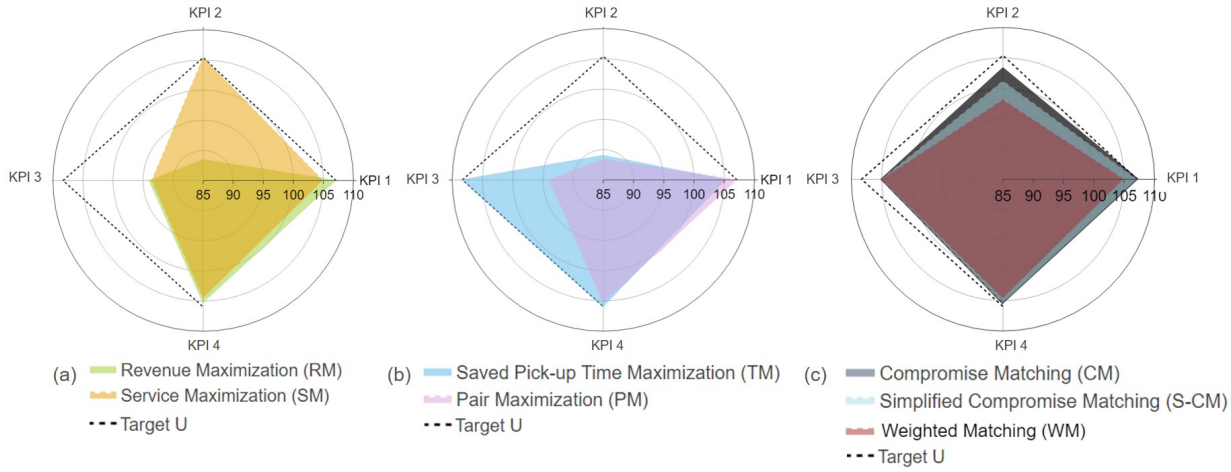
Table 1 Average performances of different ride-matching policies over $M = 300$ days.

KPIs	RM	SM	TM	PM	Target $\boldsymbol{U}$	WM	S-CM	CM
Revenue	107.12	104.89	105.27	107.11	107.12	104.84	107.01	107.29
Service Score	88.41	105.40	89.01	88.42	105.40	98.03	101.14	103.32
Saved Pick-up Distance	93.94	93.51	108.32	93.93	108.32	104.88	104.95	105.08
Matched Pair	105.28	104.44	105.97	105.28	105.97	104.57	105.33	105.51
Euclidean Distance	22.26	15.05	16.50	22.26	0.00	8.56	5.47	3.87

We select the best KPI value achieved by the single-objective maximization models and piece them together to get the target $\boldsymbol{U} = \{107.12, 105.40, 108.32, 105.97\}$. Leveraging on the same target, we examine the performance of the S-CM policy and CM policy. Table 1 shows the Euclidean distances to the KPI targets $\boldsymbol{U}$ under different ride-matching policies. The Euclidean distance under CM policy is clearly much smaller than that under the single-objective maximization policies.

As shown in Figure 3, the performance of the S-CM policy becomes much more balanced on different KPIs. Compared with the TM policy, the S-CM policy loses a little bit on the pick-up distance and the number of matched pairs, but improves the matching quality significantly. On the other hand, the S-CM policy is outperformed by the SM policy in terms of the service score, but it improves the performance on both pick-up distance and total revenue. Furthermore, it shows that the S-CM policy is Pareto dominated by the CM policy. This verifies the value of incorporating full

Figure 3 **Average performances of different ride-matching policies over $M = 300$ days.**



KPI 1: Revenue; **KPI 2:** Service Score; **KPI 3:** Saved Pick-up Distance; **KPI4:** Matched Pair

stochastic information to design the "right" weight function, as well as a "better" matching policy, particularly when the matching scenarios are not stationary across different periods.

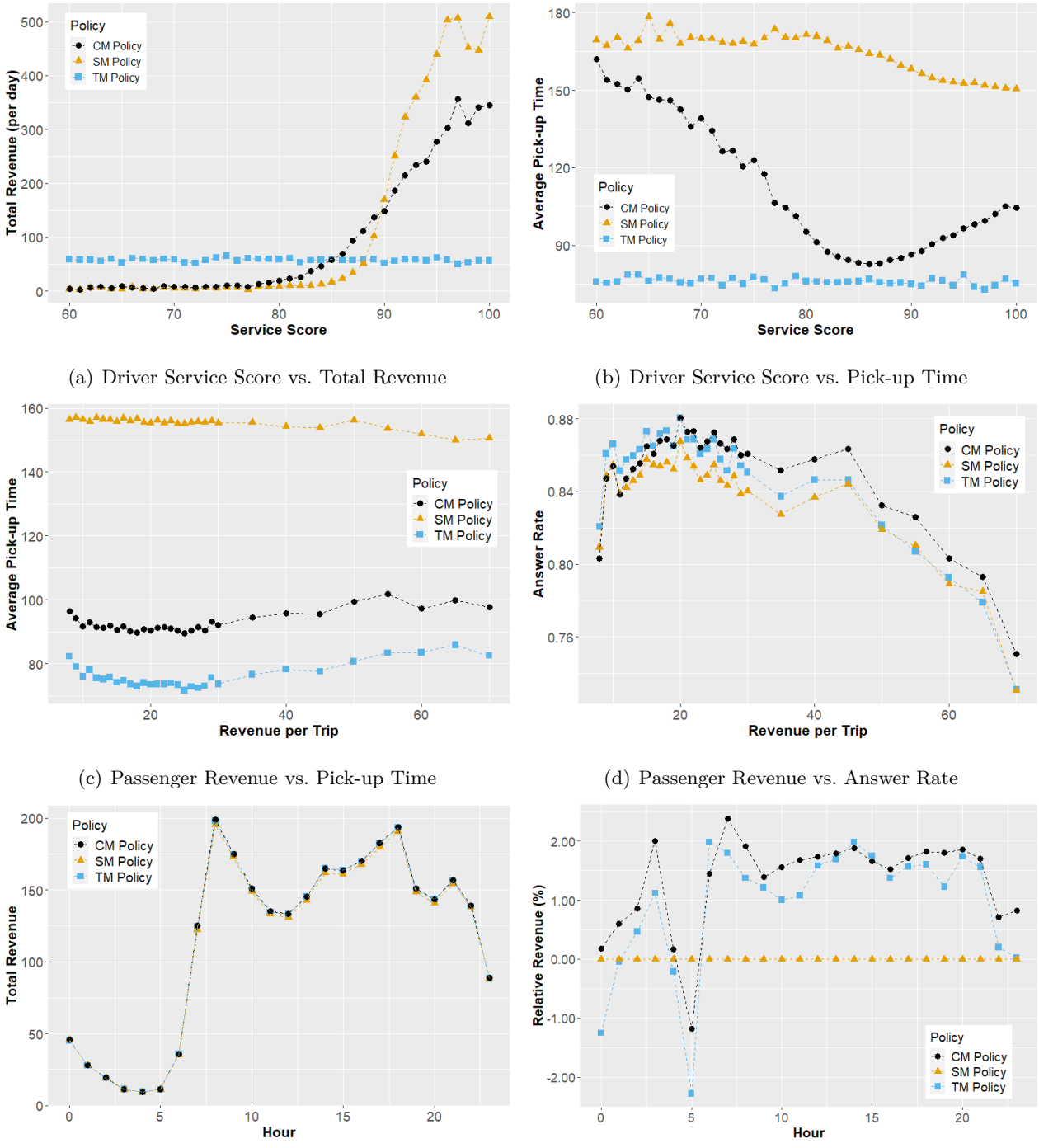
It is also worth noting that the S-CM policy operates completely online, and the weight function is adaptive to the historical scenarios, thereby resulting in enhanced computational efficiency compared to our CM policy. The S-CM policy can achieve near-optimal performance in stationary ride-matching scenarios, while the CM policy is suitable for both stationary and non-stationary settings.

5.4.2. Benefits of the CM Policy for Different Stakeholders Next, we investigate the benefits of the CM policy for different stakeholders—passengers, drivers, and the platform—in the ride-sourcing ecosystem. To make the discussion clearer, we introduce four indicators to evaluate the performance of different matching policies—revenue (per trip), pick-up time (per trip), service quality (per trip), and answer rate (proportion of passengers being served)—on the individual trip level. As shown in Table 2, the CM policy maintains a delicate balance between multiple objectives and brings value to all stakeholders. Since the performance of different polices regarding revenue and matched pairs differ only marginally, we use the SM and TM policies as benchmark policies for comparison to examine the benefit of the CM policy for different stakeholders. The detailed results are as shown in Figure 4.

Table 2 **Average performance of different ride-matching policies on individual trips.**

Performance (per trip)	RM	SM	TM	PM	WM	S-CM	CM
Revenue	19.758	19.657	19.624	19.756	19.743	19.746	19.757
Service Quality	77.461	92.656	77.464	77.504	77.495	88.282	90.005
Pick-up Time	157.783	155.911	75.970	157.830	75.421	92.229	92.404
Answer Rate (%)	84.921	84.244	85.473	84.916	84.387	84.962	85.114

Figure 4 Benefits of CM policy for drivers, passengers, and the platform.



(a) Driver Service Score vs. Total Revenue

(b) Driver Service Score vs. Pick-up Time

(c) Passenger Revenue vs. Pick-up Time

(d) Passenger Revenue vs. Answer Rate

(e) Total Revenue for the Platform

(f) Relative Revenue for the Platform

Note. (a) and (b) illustrate the benefits of CM policy for drivers; (c) and (d) for passengers; while (e) and (f) for the platform.

For Drivers: Compared with the TM policy, the CM policy improves service quality per trip by 16.19%. This implies that more jobs are assigned to drivers with higher service quality under the CM policy. Figure 4(a) shows that the total revenue earned by drivers with higher service score (e.g., higher than 85) increases under the CM policy. This outcome would motivate drivers to increase their service score by providing better ride-sourcing service. We also observe that drivers with service scores higher than 90 earn higher revenue under the SM policy. However, Figure 4(b) implies that the pick-up time under the SM policy is much higher for drivers, compared with the CM policy. Therefore, the CM policy helps all drivers save the pick-up time on the road, and this would possibly reduce the trip cancellation rate compared with the SM policy. Interestingly, we observe that the average pick-up time increases for the drivers service score higher than 90 under the CM policy. This is partially because our CM policy needs to balance the performance on different objectives: (*i*) high-service (above 90) drivers are more likely to be assigned to serve the passengers with high-revenue (e.g., long traveling distance to rural areas or airport), and they are less likely to be matched with passengers nearby immediately after serving the high-revenue passengers; (*ii*) medium-service (between 80 and 90) drivers have high priority to be matched with the nearest passengers so that the performance on saved pick-up distance can also be improved.

We remark that the drivers with lower service scores (e.g., below 70) are underserved under both SM policy and our CM policy. In the long run, these drivers might leave the platform if they are not assigned to any passenger. To address this issue, we can divide the drivers into different groups based on their service scores, and incorporate the "fairness" concern into the dispatch policy to guarantee a minimal number of passengers to be served for the drivers from each group. In this way, we add additional "objective functions" to the multi-objective ride-matching model.

For Passengers: The CM policy reveals a delicate trade-off between pick-up time and service quality for passengers. For example, compared with the TM policy, the average pick-up time slightly increases under the CM policy (cf. Figure 4(c)), but passengers are compensated with significantly higher service quality. In contrast with the SM policy, the average pick-up time is much shorter under the CM policy. Furthermore, we observe that the revenue per trip increases by 0.68% and 0.51%, respectively, compared with the TM and SM policy. In other words, more drivers are dispatched to serve these passengers with higher revenue per trip under the CM policy. Figure 4(d) demonstrates that passengers with higher revenue are served with relatively higher priority under the CM policy, compared with both TM and SM policy. As a result, passengers with lower revenue per trip (e.g., below 11) suffer from slightly lower answer rates under the CM policy. Note that some ride-sourcing platforms (e.g., Didi Chuxing[6]) are also investing in the bike-sharing and scooter-sharing markets,

[6] China's Didi Chuxing to launch bike-sharing platform and brand. Retrieved from `http://www.businesstimes.com.sg/transport/chinas-didi-chuxing-to-launch-bike-sharing-platform-and-brand`

which aim at serving those passengers with shorter travel distance. This compromise matching policy would balance their business strategies in both ride-sourcing and bike/scooter-sharing markets, i.e., to provide more comfortable service for passengers with travel request of both long and short distance.

For the Platform: As a side effect, although the answer rate reduces by 0.42%, total revenue for the whole day under the CM policy still increases by 0.25%, compared with the TM policy. Total revenue under the CM policy increases significantly, by 1.55%, compared with the SM policy. More precisely, Figure 4(e) plots the trajectory of total revenue across different time periods. It shows that total revenue is closely dependent on the demand pattern (cf. Figure 2(b)). To make the comparison clearer, we set the SM policy as the baseline and plot the relative increment for the TM and CM policy in Figure 4(f). We observe that the benefit of the CM policy is more significant during those periods when drivers start their work shifts—i.e., when drivers switch from inactive status to active. Note that our simulation starts from 0:00 a.m. and the initial active driver pool is identical for all matching policies. Compared with the benchmark policies, the CM policy serves more passengers before 3:00 a.m. As a result, the number of idle active drivers decreases in subsequent periods, and hence the total revenue earned between 4:00 a.m. and 5:00 a.m. drops. From 6:00 a.m. to 9:00 a.m. many drivers start their work shift, and hence the revenue increment increases again. Since many drivers work part-time in the ride-sourcing market and must work full-time jobs during the daytime, many will leave the platform during the daytime and become active again after 17:00 p.m. Therefore, the revenue increment becomes more significant between 17:00 p.m. and 17:59 p.m.

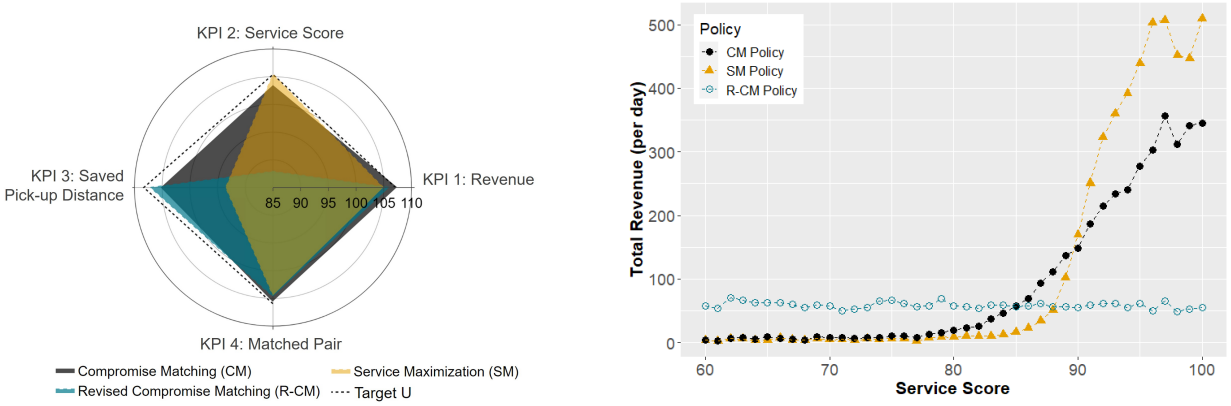
In the long term, as noted in the introduction (cf. Figure 1), if the ride-sourcing platform gives priority to drivers with high service quality, the complaint rate from passengers would decrease, and this positive signal could in turn attract more passengers. On the other hand, the income for drivers with higher quality will increase and motivate them to work for a longer time on the platform, due to positive labor supply elasticity (cf. Chen and Sheldon 2016, Sun et al. 2019, Angrist et al. 2021). In turn, the overall service quality of the platform will be improved. Long-term effects on both passengers and drivers can contribute to building a better brand reputation for the platform. However, if we only focus on service maximization, passengers would suffer from much longer pick-up time and the trip cancellation rate would increase. This would definitely hurt the core operations of the ride-sourcing platform. Instead, our CM policy provides a delicate ride-matching solution to improve service quality and also guarantees satisfactory performance on pick-up time.

5.4.3. Robustness Checks of the CM Policy We perform a robustness check to validate that the CM policy consistently dispatches more jobs to the drivers with higher service quality. For this purpose, we use the SM policy and a revised compromise matching (R-CM) policy as the benchmark policy for comparison. Note that the SM policy is similar to the policy implemented in Figure 1 by

prioritizing the drivers with higher service scores. By comparing with the SM policy, we expect to recover the result in the field experiment as depicted in Figure 1, and show that such a policy will improve the ride-matching service quality, but at the expense of both revenue and pick-up distance. For the R-CM policy, we consider a three-component KPI target, which only includes the platform revenue term, pick-up distance, and matched pair term, but the service score term is excluded. By comparing with the R-CM policy, we can directly illustrate the value of incorporating service score into the matching decision. Furthermore, we implement the matching policies in an online manner, and calculate the daily average KPI performances.

As shown in Figure 5(a), the SM policy indeed achieves the highest service score (KPI 2), while the performances on total revenue (KPI 1), pick-up distance (KPI 3), and matched pair (KPI 4) are the worst. This result is consistent with the observation in the field experiment by our industry collaborator (cf. Figure 1). The R-CM policy achieves the shortest pick-up distance, while the service score obtained is the lowest as we expect. This is intuitive since the service score is completely ignored in this policy. By incorporating the service score, the CM policy significantly increases the average service score, and also attains higher total revenue than the R-CM policy. This result comes at the expense of increasing the pick-up distance within a tolerated range.

Figure 5 **The value of incorporating the service score.**



(a) Overall Perforamnce

(b) Driver Service Score vs. Total Revenue

In Figure 5(b), we divide the drivers into different groups based on their services scores, and calculate the average total revenue in each group. Clearly, the average revenue obtained under the R-CM policy is independent of the driver service score, while the CM policy dispatches more jobs to the drivers with higher service scores.

As further checks on the performance of the proposed policies, we have performed an implementation of our compromise matching solution using an in-house simulator developed by our industry

collaborator. Due to the scale of the problem, we tested only the performance of S-CM against various benchmark policies. The results obtained are similar, showing that the insights obtained are generally reproducible. The details are in Appendix C.

5.4.4. Value of Demand Forecast in Ride-Matching Policies We remark that myopic policies are applied to solve the stochastic dynamic matching problems in the numerical experiments above. Motivated by Xu et al. (2018), we leverage on demand forecast to ameliorate the performance of myopic ride-matching policies and evaluate the value of demand forecast in ride-matching policies. More concretely, we consider a *one-period forward-looking mechanism* when we solve the ride-matching problem in a sequential manner. At each matching interval t, we assume that the decision maker can observe the new requests in the interval $t+1$, and dispatch drivers to serve passengers from both intervals t and $t+1$. Passengers from interval t are served immediately upon being matched; however, we only block drivers who are matched to passengers from $t+1$, and execute a second-round matching at interval $t+1$. In this way, the ride-sourcing market becomes "richer" and we expect that the matching quality would improve.

We design three policies—the Saved Pick-up Time Maximization with Forward-Looking (TMwFL) policy, Service Maximization with Forward-Looking (SMwFL) policy, and Compromise Matching with Forward-Looking (CMwFL) policy—to examine the value of "demand forecast" against myopic policies (e.g., the TM, SM, and CM policies). As shown in Table 3, the class of FL policies serves more passengers (i.e., increases the number of matched pairs), compared with the myopic policies. This observation is consistent with the result from Xu et al. (2018). Since the number of matched pairs is also related to other KPIs, it shows clearly that the three FL policies Pareto dominate their myopic counterparts. However, the FL policy cannot balance the trade-offs between different objectives without the assistance of the CM policy. Therefore, it is valuable to implement CM policy to serve KPIs from different parties, and use a machine learning technique to improve the quality of matching in real-world implementations.

Table 3 The value of demand forecast in the design of ride-matching policies.

KPIs	Target	TM	TMwFL	SM	SMwFL	CM	CMwFL
Revenue	107.12	105.27	105.56	104.89	105.21	107.29	107.48
Service Score	105.40	89.01	89.38	105.40	106.03	103.32	103.92
Saved Pick-up Distance	108.32	108.32	109.32	93.51	94.50	105.08	106.15
Matched Pair	105.97	105.97	106.42	104.44	104.89	105.51	105.92
Euclidean Distance	0.00	16.50	16.09	15.05	13.99	3.87	2.63

6. Conclusion

Motivated by the challenge of matching decisions under multi-objective requirements in ride-sourcing markets, we develop an efficient algorithm to solve the multi-objective optimization problem in which the decision makers act in a stochastic environment. We propose a set of necessary and sufficient conditions to characterize the feasible domain for attainable KPI targets. We demonstrate that the randomized algorithm, which is recast from an online solution, induces the compromise solution when we use the utopia point as the target. In this way, we can extend the classical notion of a compromise solution to multi-objective stochastic optimization problems. Notably, the methods and analysis are developed for both single-period and multi-period settings; in the latter case, the stochastic environments are allowed to be non-stationary with carry-over across different periods.

The novel multi-objective optimization framework has great potential in solving practical problems. We apply this framework to develop a compromise matching policy for the ride-sourcing markets and our extensive simulations on the driver dispatching problem show that this framework can lead to a favourable outcome. Interestingly, we demonstrate that the compromise matching policy is able to dispatch more orders to the drivers with higher service scores with minimal impact for the other key objectives. This positive effect could contribute to building a better brand reputation for the platform. In the numerical experiments, we incorporate platform revenue, driver service score, pick-up distance, and the number of matched pairs, to design the matching policy. In the end, we remark that it would be interesting to implement this compromise matching policy in the industry and study the long-term effect. We leave these and other issues for future research.

Acknowledgment

The authors are grateful to the Matching Group from Didi Chuxing for their help in our numerical study. Their feedbacks have helped the authors to clarify many practical issues pertaining to the multi-objective matching model.

References

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.

Angrist, Joshua D, Sydnee Caldwell, Jonathan V Hall. 2021. Uber versus taxi: A driver's eye view. *American Economic Journal: Applied Economics* **13**(3) 272–308.

Auer, Peter, Nicolo Cesa-Bianchi, Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine Learning* **47**(2) 235–256.

Baccara, Mariagiovanna, SangMok Lee, Leeat Yariv. 2020. Optimal dynamic matching. *Theoretical Economics* **15**(3) 1221–1278.

Bai, Jiaru, Kut C So, Christopher S Tang, Xiqun Chen, Hai Wang. 2019. Coordinating supply and demand on an on-demand service platform with impatient customers. *Manufacturing & Service Operations Management* **21**(3) 556–570.

Benjaafar, Saif, Ming Hu. 2020. Operations management in the age of the sharing economy: what is old and what is new? *Manufacturing & Service Operations Management* **22**(1) 93–101.

Braverman, Anton, Jim G Dai, Xin Liu, Lei Ying. 2019. Empty-car routing in ridesharing systems. *Operations Research* **67**(5) 1437–1452.

Busa-Fekete, Róbert, Balázs Szörényi, Paul Weng, Shie Mannor. 2017. Multi-objective bandits: Optimizing the generalized gini index. *International Conference on Machine Learning*. PMLR, 625–634.

Castro, Francisco, Hamid Nazerzadeh, Chiwei Yan. 2020. Matching queues with reneging: a product form solution. *Queueing Systems* **96**(3-4) 359–385.

Chen, M Keith, Michael Sheldon. 2016. Dynamic pricing in a labor market: Surge pricing and flexible work on the uber platform. *Ec* **16** 455.

Ehrgott, Matthias. 2005. *Multicriteria optimization*, vol. 491. Springer Science & Business Media.

Feng, Guiyun, Guangwen Kong, Zizhuo Wang. 2021. We are on the way: Analysis of on-demand ride-hailing systems. *Manufacturing & Service Operations Management* **23**(5) 1237–1256.

Gearhart, WB. 1979. Compromise solutions and estimation of the noninferior set. *Journal of Optimization Theory and Applications* **28**(1) 29–47.

Hu, Ming. 2021. From the classics to new tunes: A neoclassical view on sharing economy and innovative marketplaces. *Production and Operations Management* **30**(6) 1668–1685.

Hu, Ming, Yun Zhou. 2022. Dynamic type matching. *Manufacturing & Service Operations Management* **24**(1) 125–142.

Jiang, Jiashuo, Shixin Wang, Jiawei Zhang. 2023. Achieving high individual service levels without safety stock? optimal rationing policy of pooled resources. *Operations Research* **71**(1) 358–377.

Jiang, Zhong-Zhong, Guangwen Kong, Yinghao Zhang. 2021. Making the most of your regret: Workers' relocation decisions in on-demand platforms. *Manufacturing & Service Operations Management* **23**(3) 695–713.

Johari, Ramesh, Vijay Kamble, Yash Kanoria. 2021. Matching while learning. *Operations Research* **69**(2) 655–681.

Ke, Jintao, Hai Yang, Xinwei Li, Hai Wang, Jieping Ye. 2020. Pricing and equilibrium in on-demand ride-pooling markets. *Transportation Research Part B: Methodological* **139** 411–431.

Li, Xiaolong, Ying Rong, Renyu Philip Zhang, Huan Zheng. 2021a. Online advertisement allocation in the presence of customer choices. *Available at SSRN 3538755* .

Li, Xinwei, Jintao Ke, Hai Yang, Hai Wang, Yaqian Zhou. 2021b. A general matching model for on-demand mobility services. *Available at SSRN 3915450* .

Liang, Jinpeng, Guodong Lyu, Chung-Piaw Teo, Ziyou Gao. 2023. Online passenger flow control in metro lines. *Operations Research* **71**(2) 768–775.

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.

Lyu, Guodong, Mabel C Chou, Chung-Piaw Teo, Zhichao Zheng, Yuanguang Zhong. 2022. Stochastic knapsack revisited: The service level perspective. *Operations Research* **70**(2) 729–747.

Mahdavi, Mehrdad, Tianbao Yang, Rong Jin. 2013. Stochastic convex optimization with multiple objectives. *Advances in Neural Information Processing Systems* **26** 1115–1123.

Marler, R Timothy, Jasbir S Arora. 2004. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization* **26**(6) 369–395.

Marler, R Timothy, Jasbir S Arora. 2010. The weighted sum method for multi-objective optimization: new insights. *Structural and Multidisciplinary Optimization* **41**(6) 853–862.

Nesterov, Yu. 2005. Smooth minimization of non-smooth functions. *Mathematical Programming* **103**(1) 127–152.

Özkan, Erhun, Amy R Ward. 2020. Dynamic matching for real-time ride sharing. *Stochastic Systems* **10**(1) 29–70.

Serafini, Paolo. 2014. *Mathematics of multi objective optimization*, vol. 289. Springer.

Steuer, Ralph E. 1986. *Multiple criteria optimization.* Wiley.

Sun, Hao, Hai Wang, Zhixi Wan. 2019. Model and analysis of labor supply for ride-sharing platforms in the presence of sample self-selection and endogeneity. *Transportation Research Part B: Methodological* **125** 76–93.

Uziel, Guy, Ran El-Yaniv. 2017. Multi-objective non-parametric sequential prediction. *Advances in Neural Information Processing Systems* **30**.

Wang, Hai. 2022. Transportation-enabled urban services: A brief discussion. *Multimodal Transportation* **1**(2) 100007.

Wang, Hai, Hai Yang. 2019. Ridesourcing systems: A framework and review. *Transportation Research Part B: Methodological* **129** 122–155.

Wang, Xing, Niels Agatz, Alan Erera. 2018. Stable matching for dynamic ride-sharing systems. *Transportation Science* **52**(4) 850–867.

White, DJ. 1984. Multiobjective programming and penalty functions. *Journal of Optimization Theory and Applications* **43**(4) 583–599.

Xu, Zhe, Zhixin Li, Qingwen Guan, Dingshui Zhang, Qiang Li, Junxiao Nan, Chunyang Liu, Wei Bian, Jieping Ye. 2018. Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 905–913.

Yan, Chiwei, Helin Zhu, Nikita Korolko, Dawn Woodard. 2020. Dynamic pricing and matching in ride-hailing platforms. *Naval Research Logistics (NRL)* **67**(8) 705–724.

Yu, Hao, Michael J Neely, Xiaohan Wei. 2017. Online convex optimization with stochastic constraints. *Advances in Neural Information Processing Systems* **30**.

Yu, Po-Lung. 1973. A class of solutions for group decision problems. *Management Science* **19**(8) 936–946.

Yu, Po-Lung. 2013. *Multiple-criteria decision making: concepts, techniques, and extensions*, vol. 30. Springer Science & Business Media.

Zhang, Lingyu, Tao Hu, Yue Min, Guobin Wu, Junying Zhang, Pengcheng Feng, Pinghua Gong, Jieping Ye. 2017. A taxi order dispatch model based on combinatorial optimization. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2151–2159.

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.

Zhou, Yaqian, Hai Yang, Jintao Ke, Hai Wang, Xinwei Li. 2022. Competition and third-party platform-integration in ride-sourcing markets. *Transportation Research Part B: Methodological* **159** 76–103.

Zhu, Zheng, Jintao Ke, Hai Wang. 2021. A mean-field Markov decision process model for spatial-temporal subsidies in ride-sourcing markets. *Transportation Research Part B: Methodological* **150** 540–565.

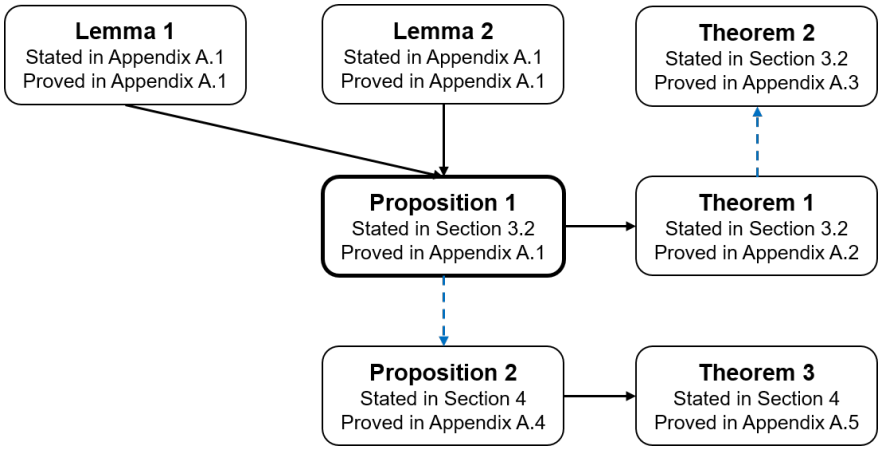
Appendix

A. Proofs of Main Results

The proof of Proposition 1 requires that we artificially construct a surrogate function (summarized in in Lemma 1) and hinges on the properties of the policy DAO$_p$ (summarized in Lemma 2). Leveraging on Proposition 1 and the randomization mechanism, we prove the main Theorem 1. Theorem 2 is a direct implication of Theorem 1. We highlight the fact that Proposition 1 is the key that connects all of the main results.

Because Proposition 2 generalizes the results in Proposition 1 from a single-period setting to a multiple-period setting, we only highlight the main connection but omit the details in the proof of Proposition 2. The proof of Theorem 3, which is derived from Proposition 2, is similar to the proof of Theorem 1. To make the discussion clearer, we provide the following Figure 6 to show the connections between the technical results.

Figure 6 Connections between the technical results.



Note. Proposition 1, Theorem 1, and Theorem 2 are related to the single-period stochastic problem; Proposition 2 and Theorem 3 are related to the multi-period stochastic problem.

A.1. Proof Framework for Proposition 1

For a fixed p, Proposition 1 is proved by analyzing a sequence of *surrogate functions* $\{\hat{\phi}_{\eta(m),p}\}_{m=1}^{\infty}$, where each function $\hat{\phi}_{\eta(m),p}$ serves as a smoothened approximation to the original penalty function ϕ_p. The proof for Proposition 1 adheres to the following outline.

- First, we recall basic facts about dual norms and state Lemma 1 based on Nesterov (2005). This Lemma demonstrates that, via *Nesterov's smoothening technique*, our surrogate functions indeed satisfy the desired approximation and smoothness properties.
- Second, we provide another Lemma 2, which relates the performance of the policy DAO$_p$ in each period to the performance of the optimal policy for the single-period stochastic problem ($\mathcal{S}$-ϕ_p). The Lemma is vital for proving the convergence of the policy DAO$_p$ to near-optimum.
- Third, we apply these two Lemmas and prove Proposition 1 by tracking the average debt vector across samples.

Step 1: Surrogate function construction and its key properties. For this step, it is useful to recall elementary facts about dual norms. First, for a given norm $\|\cdot\|$ on $\mathbb{R}^K$, its *dual norm* $\|\cdot\|^*$ is defined as $\|\boldsymbol{y}\|^* := \max_{\boldsymbol{z}\in\mathbb{R}^K:\|\boldsymbol{z}\|\leq 1} \boldsymbol{y}^\top \boldsymbol{z}$. Second, for $p,q\in[1,\infty]$ which satisfies $1/p+1/q=1$, we have $\|\cdot\|_p^* = \|\cdot\|_q$. We now define $\hat{\phi}_{p,\eta}$ for given $p\in[1,\infty], \eta\geq 0$. Here, we denote q as the number which satisfies $1/p+1/q=1$:

$$\hat{\phi}_{p,\eta}(\boldsymbol{w}) := \max_{\|\boldsymbol{z}\|_q\leq 1, \boldsymbol{z}\geq 0} \boldsymbol{z}^\top \boldsymbol{w} - \frac{\eta}{2}\boldsymbol{z}^\top \boldsymbol{z}.$$

Surrogate function $\hat{\phi}_{p,\eta}$ can be understood as a perturbed version of ϕ_p by an additive term $-\frac{\eta}{2}\boldsymbol{z}^\top\boldsymbol{z}$. Indeed, penalty function ϕ_p can be rewritten as $\phi_p(x) = \max_{\|\boldsymbol{z}\|_q\leq 1,\boldsymbol{z}\geq 0} \boldsymbol{z}^\top \boldsymbol{x}$, which in particular implies that $\hat{\phi}_{p,0}=\phi_p$. The construction of $\hat{\phi}_{\eta,p}$ stemmed from Nesterov's smoothening technique (Nesterov 2005), which results in the following desirable properties:

LEMMA 1 **(Nesterov (2005))**. *Consider arbitrary but fixed $p\in[1,\infty], \eta\geq 0$, and let $q\in[1,\infty]$ satisfy $1/p+1/q=1$. Surrogate function $\hat{\phi}_{p,\eta}$ is convex and differentiable, with*

$$\nabla\hat{\phi}_{p,\eta}(\boldsymbol{w}) \in \underset{\|\boldsymbol{z}\|_q\leq 1,\boldsymbol{z}\geq 0}{argmax} \left\{\boldsymbol{z}^\top\boldsymbol{w} - \frac{\eta}{2}\boldsymbol{z}^\top\boldsymbol{z}\right\} = \underset{\|\boldsymbol{z}\|_q\leq 1}{argmin} \left\|\boldsymbol{z} - \frac{2}{\eta}\boldsymbol{w}^+\right\|_2^2.$$

In addition, surrogate function $\hat{\phi}_{p,\eta}$ is $(1/\eta)$-smooth. That is, for any $\boldsymbol{w}^1,\boldsymbol{w}^2\in\mathbb{R}^K$ we have

$$\nabla\hat{\phi}_{p,\eta}(\boldsymbol{w}^1)^\top(\boldsymbol{w}^2-\boldsymbol{w}^1) \leq \hat{\phi}_{p,\eta}(\boldsymbol{w}^2) - \hat{\phi}_{p,\eta}(\boldsymbol{w}^1) \leq \nabla\hat{\phi}_{p,\eta}(\boldsymbol{w}^1)^\top(\boldsymbol{w}^2-\boldsymbol{w}^1) + \frac{\|\boldsymbol{w}^2-\boldsymbol{w}^1\|_2^2}{\eta}. \tag{13}$$

Lastly, surrogate function $\hat{\phi}_{p,\eta}$ serves as an approximation to the original debt penalty function ϕ_p, in the sense that the following inequalities hold for any $\boldsymbol{w}\in\mathbb{R}^K$, and any $0\leq\eta_1<\eta_2$:

$$\hat{\phi}_{p,\eta_2}(\boldsymbol{w}) \leq \hat{\phi}_{p,\eta_1}(\boldsymbol{w}) \leq \hat{\phi}_{p,\eta_2}(\boldsymbol{w}) + \frac{\eta_2-\eta_1}{2}K^{(q-2)^+/q}. \tag{14}$$

We remark that a high value of η results in a high degree of smoothness for $\hat{\phi}_{\eta,p}$, while a low value of η gives rise to good approximation for $\hat{\phi}_{\eta,p}$. Lemma 1 implies that vector $\boldsymbol{z}(m+1)$, computed in step 3, is in fact the gradient of surrogate function $\hat{\phi}_{p,\eta(m+1)}$ at $\boldsymbol{w}(m+1)$, that is,

$$\boldsymbol{z}(m+1) = \nabla\hat{\phi}_{p,\eta(m+1)}(\boldsymbol{w}(m+1)). \tag{15}$$

Consequently, our policy is based on implementing gradient descent algorithm on surrogate functions $\hat{\phi}_{p,\eta(1)}, \hat{\phi}_{p,\eta(2)}, \ldots$, where $\eta(1)\geq\eta(2)\geq\ldots$ is a suitably chosen decreasing sequence, and $\lim_{m\to\infty}\eta(m)=0$.

We reproduce the proof by (Nesterov (2005)) for completeness sake.

PROOF OF LEMMA 1. The convexity of surrogate function $\hat{\phi}_{p,\eta}$ and the expression for its derivative are evident, thus we focus on establishing the $(1/\eta)$-smoothness of $\hat{\phi}_{p,\eta}$. The lower bound in (13) follows from the convexity and differentiability of surrogate function $\hat{\phi}_{p,\eta}$; thus, in the following we focus on the upper bound in (13).

Upper bound in (13). First, we denote $\boldsymbol{\zeta}^i = \nabla\hat{\phi}_{p,\eta}(\boldsymbol{w}^i)$ for $i=1,2$. We demonstrate that

$$\|\nabla\hat{\phi}_{p,\eta}(\boldsymbol{w}^1) - \nabla\hat{\phi}_{p,\eta}(\boldsymbol{w}^2)\|_2 = \|\boldsymbol{\zeta}^1-\boldsymbol{\zeta}^2\|_2 \leq \frac{1}{\eta}\|\boldsymbol{w}^1-\boldsymbol{w}^2\|_2. \tag{16}$$

Now, consider concave function $g(\boldsymbol{z}) = \boldsymbol{z}^\top\boldsymbol{w}^1 - \frac{\eta}{2}\boldsymbol{z}^\top\boldsymbol{z}$ over convex domain $D=\{\boldsymbol{z}\in\mathbb{R}^K : \|\boldsymbol{z}\|_q\leq 1, \boldsymbol{z}\geq 0\}$. Since $\boldsymbol{\zeta}^1$ maximizes g over D, we have $\nabla g(\boldsymbol{\zeta}^1)^\top(\boldsymbol{z}-\boldsymbol{\zeta}^1)\leq 0$ for any $\boldsymbol{z}\in D$. In particular, $\nabla g(\boldsymbol{\zeta}^1)^\top(\boldsymbol{\zeta}^2 -$

$\boldsymbol{\zeta}^1) \leq 0$, which is equivalent to $(\boldsymbol{w}^1 - \eta\boldsymbol{\zeta}^1)^\top(\boldsymbol{\zeta}^2 - \boldsymbol{\zeta}^1) \leq 0$. By swapping the roles of $\boldsymbol{w}^1, \boldsymbol{w}^2$, we also have $(\boldsymbol{w}^2 - \eta\boldsymbol{\zeta}^2)^\top(\boldsymbol{\zeta}^1 - \boldsymbol{\zeta}^2) \leq 0$. Together, adding these two resulting inequalities yield

$$\eta\|\boldsymbol{\zeta}^1 - \boldsymbol{\zeta}^2\|_2^2 \leq (\boldsymbol{w}^1 - \boldsymbol{w}^2)^\top(\boldsymbol{\zeta}^1 - \boldsymbol{\zeta}^2).$$

By Cauchy-Schwartz inequality, we have $(\boldsymbol{w}^1 - \boldsymbol{w}^2)^\top(\boldsymbol{\zeta}^1 - \boldsymbol{\zeta}^2) \leq \|\boldsymbol{w}^1 - \boldsymbol{w}^2\|_2\|\boldsymbol{\zeta}^1 - \boldsymbol{\zeta}^2\|_2$. By putting back the definitions of $\boldsymbol{\zeta}^1, \boldsymbol{\zeta}^2$, inequality (16) is proved.

Finally, to prove the upper bound for (13), consider the following single variate function $h(r) = \hat{\phi}_{p,\eta}(\boldsymbol{w}^1 + r(\boldsymbol{w}^2 - \boldsymbol{w}^1))$. By the Mean Value Theorem, there exists $\bar{r} \in [0, 1]$ such that

$$\hat{\phi}_{p,\eta}(\boldsymbol{w}^2) - \hat{\phi}_{p,\eta}(\boldsymbol{w}^1) = h(1) - h(0) = h'(\bar{r}) = \nabla\hat{\phi}_{p,\eta}(\boldsymbol{w}^1 + \bar{r}(\boldsymbol{w}^2 - \boldsymbol{w}^1))^\top(\boldsymbol{w}^2 - \boldsymbol{w}^1).$$

Therefore, we have

$$\begin{aligned}\hat{\phi}_{p,\eta}(\boldsymbol{w}^2) - \hat{\phi}_{p,\eta}(\boldsymbol{w}^1) - \nabla\hat{\phi}_{p,\eta}(\boldsymbol{w}^1)^\top(\boldsymbol{w}^2 - \boldsymbol{w}^1) &= \left[\nabla\hat{\phi}_{p,\eta}(\boldsymbol{w}^1 + \bar{r}(\boldsymbol{w}^2 - \boldsymbol{w}^1)) - \nabla\hat{\phi}_{p,\eta}(\boldsymbol{w}^1)\right]^\top(\boldsymbol{w}^2 - \boldsymbol{w}^1) \\ &\leq \frac{\bar{r}}{\eta}(\boldsymbol{w}^2 - \boldsymbol{w}^1)^\top(\boldsymbol{w}^2 - \boldsymbol{w}^1) \leq \frac{1}{\eta}\|\boldsymbol{w}^2 - \boldsymbol{w}^1\|_2^2,\end{aligned}$$

which establishes the upper bound in (13).

Approximation guarantee (14). The lower bound in (14) is clearly true, so we focus on the upper bound in (14). Now, let $\boldsymbol{\zeta}$ maximizes function $g(\boldsymbol{z}) = \boldsymbol{z}^\top\boldsymbol{w} - \frac{\eta_1}{2}\boldsymbol{z}^\top\boldsymbol{z}$ over domain $D = \{\boldsymbol{z} \in \mathbb{R}^K : \|\boldsymbol{z}\|_q \leq 1, \boldsymbol{z} \geq 0\}$. Then we know that $\hat{\phi}_{p,\eta_1}(\boldsymbol{w}) \leq \hat{\phi}_{p,\eta_2}(\boldsymbol{w}) + \frac{\eta_2 - \eta_1}{2}\|\boldsymbol{\zeta}\|_2^2$. To prove the upper bound, it now suffices to compute the maximum value of $\|\boldsymbol{z}\|_2^2$ over D, which upper bounds $\|\boldsymbol{\zeta}\|_2^2$. Now, for $1 \leq q \leq 2$, the maximum value is 1, achieved by any standard basis vector in $\mathbb{R}^K$. For $2 \leq q \leq \infty$, the maximum value is $K^{(q-2)/q}$, achieve by $(\underbrace{K^{-1/q}, \ldots, K^{-1/q}}_{K \text{ times}})$. ■

Step 2: Comparison of policy DAO$_p$ and the optimal policy. In this step, we provide a Lemma that facilitates the comparison between policy DAO$_p$ and the optimal policy for ($\mathcal{S}$-ϕ_p). To state the Lemma, we define $y_k^* := \mathbf{E}_{\boldsymbol{\omega}\sim\Psi}[f_k(\boldsymbol{x}^*(\boldsymbol{\omega}), \boldsymbol{\omega})]$ for each $k \in \{1, 2, \ldots, K\}$, where $\{\boldsymbol{x}^*(\boldsymbol{\omega})\}_{\boldsymbol{\omega}\in\boldsymbol{\Omega}}$ is the optimal solution to the single-period problem ($\mathcal{S}$-ϕ_p). For a given sample index m and a realization of average debt $\boldsymbol{w}(m)$ from period 1 to $(m-1)$, we denote

$$y_k^{\text{DAO}_p}(m) := \mathbf{E}_{\boldsymbol{\omega}_m\sim\Psi}\left[f_k(\boldsymbol{x}_m^{\text{DAO}_p}, \boldsymbol{\omega}_m)\middle|\boldsymbol{w}(m)\right] \tag{17}$$

for each $k \in \{1, 2, \ldots, K\}$. The conditional expectation in (17) is taken over only on the randomness of the period t scenario $\boldsymbol{\omega}_m$, which is distributed as Ψ, while the average debt $\boldsymbol{w}(m)$ is held deterministic.

Lemma 2. *For any given sample m and any realization of average debt $\boldsymbol{w}(m)$, the following inequality holds with certainty:*

$$\sum_{k=1}^K z_k(m)y_k^{DAO_p}(m) \geq \sum_{k=1}^K z_k(m)y_k^*. \tag{18}$$

Proof of Lemma 2. Now, let's condition on a realization of $\boldsymbol{z}(m+1)$. By Step 4 in policy DAO$_p$, we know that

$$\sum_{k=1}^K z_k(m+1)f_k(\boldsymbol{x}_{m+1}^{\text{DAO}_p}, \boldsymbol{\omega}_{m+1}) \geq \sum_{k=1}^K z_k(m+1)f_k(\boldsymbol{x}(\boldsymbol{\omega}_{m+1}), \boldsymbol{\omega}_{m+1})$$

for any $\boldsymbol{x}(\boldsymbol{\omega}_{m+1}) \in \mathcal{X}(\boldsymbol{\omega}_{m+1})$. In particular, for an optimal solution $\{\boldsymbol{x}(\boldsymbol{\omega})\}_{\boldsymbol{\omega}\in\boldsymbol{\Omega}}$ of problem ($\mathcal{S}$-ϕ_p), we have

$$\sum_{k=1}^{K} z_k(m+1) f_k(\boldsymbol{x}_{m+1}^{\mathrm{DAO}_p}, \boldsymbol{\omega}_{m+1}) \geq \sum_{k=1}^{K} z_k(m+1) f_k(\boldsymbol{x}^*(\boldsymbol{\omega}_{m+1}), \boldsymbol{\omega}_{m+1}).$$

Take expectation over $\boldsymbol{\omega}_{m+1} \sim \Psi$ yields the required inequality (18). ■

Step 3: Evaluation of the debt under policy $\mathbf{DAO}_p$. In the final step, we apply Lemmas 1, 2 to prove Proposition 1:

Proposition 1. *Consider the sample-average-approximate problem ($\mathcal{M}$-ϕ_p), and let $q \in [1,\infty]$ satisfy $1/p + 1/q = 1$. The average debt $\boldsymbol{w}(M+1)$ under policy DAO$_p$ satisfies*

$$\mathbf{E}\left[\phi_p(\boldsymbol{w}(M+1))\right] - \phi_p(\boldsymbol{w}^*) \leq \sqrt{\frac{8K^{\frac{(q-2)^+}{q}}\Delta^2}{M}}. \tag{6}$$

Proof of Theorem 1. Recall that the debt $\boldsymbol{D}(s)$ at sample s (as compared to the utopia point $\boldsymbol{U}$) is defined as $D_k(s) = U_k - f_k(\boldsymbol{x}_s^{\mathrm{DAO}_p}, \boldsymbol{\omega}_s)$ for $k \in \{1,2,\ldots,K\}$, and also recall that $\boldsymbol{w}(m+1) = \frac{1}{m}\sum_{s=1}^{m} \boldsymbol{D}(s)$ is the average debt vector for the first m samples. Denote $\boldsymbol{w}(0) = 0$. For any $0 < m \leq M-1$, we have

$$\begin{aligned}
&\hat{\phi}_{p,\eta(m)}(\boldsymbol{w}(m+1)) \\
\leq &\hat{\phi}_{p,\eta(m)}(\boldsymbol{w}(m)) + \nabla\hat{\phi}_{p,\eta(m)}(\boldsymbol{w}(m))^\top [\boldsymbol{w}(m+1) - \boldsymbol{w}(m)] + \frac{\|\boldsymbol{w}(m+1) - \boldsymbol{w}(m)\|_2^2}{\eta(m)} && (19)\\
= &\hat{\phi}_{p,\eta(m)}(\boldsymbol{w}(m)) + \frac{\nabla\hat{\phi}_{p,\eta(m)}(\boldsymbol{w}(m))^\top [\boldsymbol{D}(m) - \boldsymbol{w}(m)]}{m} + \frac{\|\boldsymbol{D}(m) - \boldsymbol{w}(m)\|_2^2}{m^2\eta(m)} && (20)\\
\leq &\phi_{p,\eta(m)}(\boldsymbol{w}(m)) + \frac{\nabla\hat{\phi}_{p,\eta(m)}(\boldsymbol{w}(m))^\top [\boldsymbol{D}(m) - \boldsymbol{w}(m)]}{m} + \frac{\Delta^2}{m^2\eta(m)} && (21)\\
= &\hat{\phi}_{p,\eta(m)}(\boldsymbol{w}(m)) + \frac{\nabla\hat{\phi}_{p,\eta(m)}(\boldsymbol{w}(m))^\top \{\mathbf{E}[\boldsymbol{D}(m) \mid \boldsymbol{w}(m)] - \boldsymbol{w}(m)\}}{m} + \frac{\Delta^2}{m^2\eta(m)} \\
&\qquad + \frac{\nabla\hat{\phi}_{p,\eta(m)}(\boldsymbol{w}(m))^\top \{\boldsymbol{D}(m) - \mathbf{E}[\boldsymbol{D}(m) \mid \boldsymbol{w}(m)]\}}{m}. && (22)
\end{aligned}$$

Step (19) is justified by applying the upper bound in Lemma 1, with $\boldsymbol{w}^2 \to \boldsymbol{w}(m+1)$, $\boldsymbol{w}^1 \to \boldsymbol{w}(m)$. Step (20) is by the definitions of $\boldsymbol{w}(m), \boldsymbol{D}(m)$:

$$\boldsymbol{w}(m+1) = \frac{m-1}{m}\boldsymbol{w}(m) + \frac{1}{m}\boldsymbol{D}(m) \quad \Leftrightarrow \quad \boldsymbol{w}(m+1) - \boldsymbol{w}(m) = \frac{\boldsymbol{D}(m) - \boldsymbol{w}(m)}{m}.$$

In step (21), we bound the term $\|\boldsymbol{D}(m) - \boldsymbol{w}(m)\|_2^2$ with certainty from above based on the definition of constant Δ (c.f. (5)):

$$\|\boldsymbol{D}(m) - \boldsymbol{w}(m)\|_2 \leq \Delta. \tag{23}$$

Finally, in step (22), we decompose the difference term $\boldsymbol{D}(m) - \boldsymbol{w}(m)$ by considering the conditional expectation $\mathbf{E}[\boldsymbol{D}(m) \mid \boldsymbol{w}(m)]$; the conditional expectation is taken over $\boldsymbol{\omega}(m)$, while the debt $\boldsymbol{w}(m)$ is deterministic under the conditional expectation.

Lemma 2 and the fact that $\nabla\hat{\phi}_{p,\eta(m)}(\boldsymbol{w}(m)) = \boldsymbol{z}(m)$ imply that

$$\nabla\hat{\phi}_{p,\eta(m)}(\boldsymbol{w}(m))^\top \mathbf{E}[\boldsymbol{D}(m) \mid \boldsymbol{w}(m)] \leq \nabla\hat{\phi}_{p,\eta(m)}(\boldsymbol{w}(m))^\top \boldsymbol{w}^*.$$

Thus, we can proceed with bounding (22) in the following:

$$\begin{aligned}\hat{\phi}_{p,\eta(m)}(\boldsymbol{w}(m+1)) &\leq \hat{\phi}_{p,\eta(m)}(\boldsymbol{w}(m)) + \frac{\nabla\hat{\phi}_{p,\eta(m)}(\boldsymbol{w}(m))^\top[\boldsymbol{w}^*-\boldsymbol{w}(m)]}{m} + \frac{\Delta^2}{m^2\eta(m)} \\ &\quad + \frac{\nabla\hat{\phi}_{p,\eta(m)}(\boldsymbol{w}(m))^\top\{\boldsymbol{D}(m)-\mathbf{E}[\boldsymbol{D}(m)\,|\,\boldsymbol{w}(m)]\}}{m} \\ &\leq \hat{\phi}_{p,\eta(m)}(\boldsymbol{w}(m)) + \frac{\hat{\phi}_{p,\eta(m)}(\boldsymbol{w}^*)-\hat{\phi}_{p,\eta(m)}(\boldsymbol{w}(m))}{m} + \frac{\Delta^2}{m^2\eta(m)} \\ &\quad + \frac{\nabla\hat{\phi}_{p,\eta(m)}(\boldsymbol{w}(m))^\top\{\boldsymbol{D}(m)-\mathbf{E}[\boldsymbol{D}(m)\,|\,\boldsymbol{w}(m)]\}}{m}. \end{aligned} \tag{24}$$

where step (24) is by applying the lower bound in inequality (13) with $\boldsymbol{w}^1 \leftarrow \boldsymbol{w}(m)$ and $\boldsymbol{w}^2 \leftarrow \boldsymbol{w}^*$. Rearranging (24) and recalling the approximation of $\hat{\phi}_{p,\eta(m+1)}$ to ϕ_p stated in Lemma 1 give

$$\begin{aligned}&\hat{\phi}_{p,\eta(m+1)}(\boldsymbol{w}(m+1)) - \hat{\phi}_{p,\eta(m+1)}(\boldsymbol{w}^*) \\ &\quad\leq \left(1-\frac{1}{m}\right)\left[\hat{\phi}_{p,\eta(m)}(\boldsymbol{w}(m)) - \hat{\phi}_{p,\eta(m)}(\boldsymbol{w}^*)\right] + \frac{K^{\frac{(q-2)^+}{q}}(\eta(m)-\eta(m+1))}{2} + \frac{\Delta^2}{m^2\eta(m)} \\ &\qquad + \frac{\nabla\hat{\phi}_{p,\eta(m)}(\boldsymbol{w}(m))^\top\{\boldsymbol{R}(m)-\mathbf{E}[\boldsymbol{R}(m)\,|\,\boldsymbol{w}(m)]\}}{m}. \end{aligned} \tag{25}$$

Let $\Phi_m := m(\hat{\phi}_{p,\eta(m+1)}(\boldsymbol{w}(m+1)) - \hat{\phi}_{p,\eta(m+1)}(\boldsymbol{w}^*))$. Then inequality (25) can be rewritten as

$$\Phi_m \leq \Phi_{m-1} + \frac{K^{\frac{(q-2)^+}{q}} m(\eta(m)-\eta(m+1))}{2} + \frac{\Delta^2}{m\eta(m)} + \nabla\hat{\phi}_{p,\eta(m)}(\boldsymbol{w}(m))^\top\{\boldsymbol{D}(m)-\mathbf{E}[\boldsymbol{D}(m)\,|\,\boldsymbol{w}(m)]\}. \tag{26}$$

Now, we apply inequality (26) recursively from $m=M$ to $m=1$, which results in

$$\Phi_M \leq \left[\sum_{m=1}^{M}\frac{K^{\frac{(q-2)^+}{q}}}{2}\eta(m) + \frac{\Delta^2}{m\eta(m)}\right] - \frac{K^{\frac{(q-2)^+}{q}} M\eta(m+1)}{2} + \sum_{m=1}^{M}\nabla\hat{\phi}_{p,\eta(m)}(\boldsymbol{w}(m))^\top\{\boldsymbol{D}(m)-\mathbf{E}[\boldsymbol{D}(m)\,|\,\boldsymbol{w}(m)]\}. \tag{27}$$

Applying our definition of $\eta(m) = \sqrt{2\Delta^2 \Big/ \left(K^{\frac{(q-2)^+}{q}} m\right)}$ to (27), we then have

$$\begin{aligned}&\hat{\phi}_{p,\eta(m+1)}(\boldsymbol{w}(M+1)) - \hat{\phi}_{p,\eta(m+1)}(\boldsymbol{w}^*) \\ &\leq \frac{1}{M}\sqrt{2K^{\frac{(q-2)^+}{q}}\Delta^2}\left[\sum_{m=1}^{M}\frac{1}{\sqrt{m}}\right] - \frac{K^{\frac{(q-2)^+}{q}}\eta(m+1)}{2} + \frac{1}{M}\sum_{m=1}^{M}\nabla\hat{\phi}_{p,\eta(m)}(\boldsymbol{w}(m))^\top\{\boldsymbol{D}(m)-\mathbf{E}[\boldsymbol{D}(m)\,|\,\boldsymbol{w}(m)]\} \\ &\leq \sqrt{\frac{8K^{\frac{(q-2)^+}{q}}\Delta^2}{M}} - \frac{K^{\frac{(q-2)^+}{q}}\eta(m+1)}{2} + \frac{1}{M}\sum_{m=1}^{M}\nabla\hat{\phi}_{p,\eta(m)}(\boldsymbol{w}(m))^\top\{\boldsymbol{D}(m)-\mathbf{E}[\boldsymbol{D}(m)\,|\,\boldsymbol{w}(m)]\}. \end{aligned}$$

By Lemma 1, we have inequality $0 \leq \hat{\phi}_{p,\eta(m+1)}(\boldsymbol{w}) - \phi_p(\boldsymbol{w}) \leq K^{(q-2)^+/q}\eta(m+1)/2$ for all $\boldsymbol{w}$. Thus, we recover the error bound for ℓ_p-norm penalty function ϕ_p:

$$\phi_p(\boldsymbol{w}(M+1)) - \phi_p(\boldsymbol{w}^*) \leq \sqrt{\frac{8K^{\frac{(q-2)^+}{q}}\Delta^2}{M}} + \frac{1}{M}\sum_{m=1}^{M}\nabla\hat{\phi}_{p,\eta(m)}(\boldsymbol{w}(m))^\top\{\boldsymbol{D}(m)-\mathbf{E}[\boldsymbol{D}(m)\,|\,\boldsymbol{w}(m)]\}. \tag{28}$$

Finally, taking expectation on $\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \ldots, \boldsymbol{\omega}_M \sim \Psi$ completes the proof. ■

A.2. Proof of Theorem 1

THEOREM 1. *Consider the single-period stochastic problem ($\mathcal{S}$-ϕ_p). By performing Algorithm 1 with $M = \lceil 8\Delta^2/\epsilon^2 \rceil$, the output oracle $\{\boldsymbol{x}^{\mathcal{S}\text{-}\phi_p}(\boldsymbol{\omega})\}_{\boldsymbol{\omega}\in\boldsymbol{\Omega}}$ is feasible for problem ($\mathcal{S}$-ϕ_p), and has an objective value within an additive error $\sqrt{\frac{8K\frac{(q-2)^+}{q}\Delta^2}{M}}$ away from the optimum of problem ($\mathcal{S}$-ϕ_p), where q satisfies $1/p+1/q=1$.*

PROOF OF THEOREM 1. Consider the randomized mechanism displayed in Algorithm 1. Clearly, we have $x^{\mathcal{S}\text{-}\phi_p}(\boldsymbol{\omega}) \in \mathcal{X}(\boldsymbol{\omega})$ for all $\boldsymbol{\omega} \in \boldsymbol{\Omega}$. Thus, we focus on verifying the bound on the expectation for each $k \in \{1,2,\ldots,K\}$. We claim that

$$\mathbf{E}_{\bar{m},\boldsymbol{\omega}}[f_k(\boldsymbol{x}^{\mathcal{S}\text{-}\phi_p}(\boldsymbol{\omega}),\boldsymbol{\omega})] = \frac{1}{M}\mathbf{E}\left[\sum_{m=1}^{M} f_k(\boldsymbol{x}_m^{\mathrm{DAO}_p},\boldsymbol{\omega}_m)\right]$$

holds for each $k \in \{1,2,\ldots,K\}$. The expectation on the right hand side is taken over $\boldsymbol{\omega}_1,\boldsymbol{\omega}_2,\ldots,\boldsymbol{\omega}_M$, which are i.i.d. as Ψ, and on the left hand side we also have $\boldsymbol{\omega}\sim\Psi$. Consequently, conditional on $\bar{m}=m$, we know that $\boldsymbol{x}^{\mathcal{S}\text{-}\phi_p}(\boldsymbol{\omega})$ and $\boldsymbol{x}_m^{\mathrm{DAO}_p}$ are identically distributed, thus the equality follows.

By applying the performance guarantee for policy DAO_p in Proposition 1, we arrive at our desired performance guarantee for the oracle $\{\boldsymbol{x}^{\mathcal{S}\text{-}\phi_p}(\boldsymbol{\omega})\}_{\boldsymbol{\omega}\in\boldsymbol{\Omega}}$ in the Theorem. ■

A.3. Proof of Theorem 2

THEOREM 2. *Consider the single-period stochastic problem ($\mathcal{S}$-ϕ_p). A KPI target $\boldsymbol{\tau}$ is attainable under a certain decision rule if and only if conditions (7) hold:*

$$\mathbf{E}_{\boldsymbol{\omega}\sim\Psi}\left[\max_{\boldsymbol{x}(\boldsymbol{\omega})}\left\{\sum_{k=1}^{K}\theta_k f_k(\boldsymbol{x}(\boldsymbol{\omega}),\boldsymbol{\omega}) : \boldsymbol{x}(\boldsymbol{\omega})\in\mathcal{X}(\boldsymbol{\omega})\right\}\right] \geq \sum_{k=1}^{K}\theta_k\tau_k,\ \forall\boldsymbol{\theta}\in\mathbb{R}^K_{\geq 0}. \tag{7}$$

PROOF OF THEOREM 2. We first prove the necessity part, and then use Theorem 1 to prove the sufficiency.

Necessity for Conditions (7). Suppose that a KPI target $\boldsymbol{\tau}$ can be achieved by a certain decision rule $\{\boldsymbol{x}^{\mathcal{A}}(\boldsymbol{\omega})\}_{\boldsymbol{\omega}\in\Omega}$, then we have

$$\mathbf{E}_{\boldsymbol{\omega}\sim\Psi}\left[f_k(\boldsymbol{x}^{\mathcal{A}}(\boldsymbol{\omega}),\boldsymbol{\omega})\right] \geq \tau_k,\ \forall k\in\{1,2,\ldots,K\}. \tag{29}$$

Alternatively, we can construct a non-anticipatory policy $\mathcal{A}$ such that

$$\liminf_{M\to\infty}\frac{1}{M}\left(\sum_{m=1}^{M} f_k(\boldsymbol{x}_m^{\mathcal{A}},\boldsymbol{\omega}_m)\right) \geq \tau_k,\ \forall k\in\{1,2,\ldots,K\}, \tag{30}$$

where $\boldsymbol{x}_m^{\mathcal{A}}$ represents the decision made at sample m under policy $\mathcal{A}$. By taking a linear combination with any $\boldsymbol{\theta}\in\mathbb{R}^K_{\geq 0}$, conditions (30) imply that

$$\liminf_{M\to\infty}\frac{1}{M}\left(\sum_{k=1}^{K}\sum_{m=1}^{M}\theta_k f_k(\boldsymbol{x}_m^{\mathcal{A}},\boldsymbol{\omega}_m)\right) \geq \sum_{k=1}^{K}\theta_k\tau_k,\ \forall\boldsymbol{\theta}\in\mathbb{R}^K_{\geq 0}. \tag{31}$$

By definition, we have $\boldsymbol{x}_m^{\mathcal{A}}\in\mathcal{X}(\boldsymbol{\omega}_m)$, which yields the following inequality for any $\boldsymbol{\theta}\in\mathbb{R}^K_{\geq 0}$:

$$\max_{\boldsymbol{x}(\boldsymbol{\omega}_m)}\left\{\sum_{k=1}^{K}\theta_k f_k(\boldsymbol{x}(\boldsymbol{\omega}_m),\boldsymbol{\omega}_m) : \boldsymbol{x}(\boldsymbol{\omega}_m)\in\mathcal{X}(\boldsymbol{\omega}_m)\right\} \geq \sum_{k=1}^{K}\theta_k f_k(\boldsymbol{x}_m^{\mathcal{A}},\boldsymbol{\omega}_m). \tag{32}$$

Combining Equations (31) and (32), it is straightforward to conclude the following. For a KPI target $\boldsymbol{\tau}$ to be attainable under some policy, it is necessary to have

$$\liminf_{M\to\infty}\frac{1}{M}\sum_{m=1}^{M}\left[\max_{\boldsymbol{x}(\boldsymbol{\omega}_m)}\left\{\sum_{k=1}^{K}\theta_k f_k(\boldsymbol{x}(\boldsymbol{\omega}_m),\boldsymbol{\omega}_m) : \boldsymbol{x}(\boldsymbol{\omega}_m)\in\mathcal{X}(\boldsymbol{\omega}_m)\right\}\right] \geq \sum_{k=1}^{K}\theta_k\tau_k,\ \forall\boldsymbol{\theta}\in\mathbb{R}^K_{\geq 0}, \tag{33}$$

which is equivalent to the set of conditions (7) by the Strong Law of Large Number. This proves the necessity part of Theorem 2.

Sufficiency for Conditions (7). The sufficiency part is indeed a direct implementation of Theorem 1. To see this, consider the penalty function ϕ_2 and define $w_k^* = \tau_k - \mathbf{E}_{\boldsymbol{\omega}\sim\Psi}[f_k(\boldsymbol{x}^*(\boldsymbol{\omega}),\boldsymbol{\omega})]$, where $\{\boldsymbol{x}^*(\boldsymbol{\omega})\}_{\boldsymbol{\omega}\in\boldsymbol{\Omega}}$ is an optimal compromise solution to problem ($\mathcal{S}$-ϕ_2) and τ_k denotes the attainable target along the k^{th} objective. If conditions (7) hold, we have $\phi_2(\boldsymbol{w}^*)=0$. Therefore, by performing Algorithm 1 (for $p=2$) with $M=\lceil 8\Delta^2/\epsilon^2\rceil$ to solve problem ($\mathcal{S}$-ϕ_2) under the attainable target $\boldsymbol{\tau}$, the output oracle $\{\boldsymbol{x}^{\mathcal{S}\text{-}\phi_2}(\boldsymbol{\omega})\}_{\boldsymbol{\omega}\in\boldsymbol{\Omega}}$ ensures that

$$\tau_k - \mathbf{E}_{\bar{m},\boldsymbol{\omega}}[f_k(\boldsymbol{x}^{\mathcal{S}\text{-}\phi_2}(\boldsymbol{\omega}),\boldsymbol{\omega})] = \tau_k - \frac{1}{M}\mathbf{E}\left[\sum_{m=1}^{M} f_k(\boldsymbol{x}_m^{\mathrm{DAO}_2},\boldsymbol{\omega}_m)\right] \tag{34}$$

$$\leq \mathbf{E}\left[\max\left\{\tau_k - \frac{1}{M}\sum_{m=1}^{M} f_k(\boldsymbol{x}_m^{\mathrm{DAO}_2},\boldsymbol{\omega}_m), 0\right\}\right] \tag{35}$$

$$\leq \mathbf{E}\left[\phi_2(\boldsymbol{w}(M+1))\right] \leq \sqrt{\frac{8\Delta^2}{M}} \leq \epsilon \tag{36}$$

holds for each and every $k\in\{1,2,\ldots,K\}$. With a slight abuse of notation, we let $\boldsymbol{x}_m^{\mathrm{DAO}_2}$ denote the decision under policy DAO_2 that corresponds to realized scenarios $\boldsymbol{\omega}_m$, which is randomly generated at step 2 of Algorithm 1. The expectation on the right hand side of (34) is taken over $\boldsymbol{\omega}_1,\boldsymbol{\omega}_2,\ldots,\boldsymbol{\omega}_M$, which are i.i.d. as Ψ, and on the left hand side, we also have $\boldsymbol{\omega}\sim\Psi$. Consequently, conditional on $\bar{m}=m$ (see step 4 of Algorithm 1), we know that $\boldsymbol{x}^{\mathcal{S}\text{-}\phi_2}(\boldsymbol{\omega})$ and $\boldsymbol{x}_m^{\mathrm{DAO}_2}$ are identically distributed, and thus equality (34) follows. The inequality in (35) is derived by definition and the inequalities in (36) are based on Proposition 1.

Altogether, the sufficiency of conditions (7) is established. ■

A.4. Proof of Proposition 2

Proposition 2. *Consider the sample-average-approximate problem ($\mathcal{G}$-$\mathcal{M}$-ϕ_p) and let $q\in[1,\infty]$ satisfy $1/p+1/q=1$. The average debt $\boldsymbol{w}(M+1)$ under policy $\mathcal{G}$-DAO$_p$ satisfies*

$$\mathbf{E}\left[\phi_p(\boldsymbol{w}(M+1))\right] - \phi_p(\boldsymbol{w}^*) \leq \sqrt{\frac{8K^{\frac{(q-2)^+}{q}}(\Delta^T)^2}{M}}. \tag{12}$$

Proof of Proposition 2. We treat the multi-period stochastic problem as a "single" problem. To see this, we define $F_k(\boldsymbol{x}^{1:T},\boldsymbol{y}^{1:T},\boldsymbol{\omega}^{1:T}) := \sum_{t=1}^{T} f_k\big(\boldsymbol{x}^t,\boldsymbol{y}^t,\boldsymbol{\omega}^t\big)$, where $\boldsymbol{x}^{1:T}=\{\boldsymbol{x}^1,\boldsymbol{x}^2,\ldots,\boldsymbol{x}^T\}$, $\boldsymbol{y}^{1:T}=\{\boldsymbol{y}^1,\boldsymbol{y}^2,\ldots,\boldsymbol{y}^T\}$, and $\boldsymbol{\omega}^{1:T}=\{\boldsymbol{\omega}^1,\boldsymbol{\omega}^2,\ldots,\boldsymbol{\omega}^T\}$ represent the decision matrice and scenario matrix during the planning horizon, respectively. Analogous to the proof of Proposition 1, we can apply a similar argument to the aggregated function $F_k(\boldsymbol{x}^{1:T},\boldsymbol{y}^{1:T},\boldsymbol{\omega}^{1:T})$ to derive the convergence result in Proposition 2. ■

A.5. Proof of Theorem 3

Theorem 3. *Consider the multi-period stochastic problem ($\mathcal{G}$-$\mathcal{S}$-ϕ_p). By performing Algorithm 2 with $M=\lceil 8(\Delta^T)^2/\epsilon^2\rceil$, the output oracle $\{\boldsymbol{x}^{\mathcal{G}\text{-}\mathcal{S}\text{-}\phi_p,t},\boldsymbol{y}^{\mathcal{G}\text{-}\mathcal{S}\text{-}\phi_p,t}\}_{t=1}^T$ is feasible for problem ($\mathcal{G}$-$\mathcal{S}$-ϕ_p), and has an objective value within an additive error $\sqrt{\frac{8K^{\frac{(q-2)^+}{q}}(\Delta^T)^2}{M}}$ away from the optimum of ($\mathcal{G}$-$\mathcal{S}$-ϕ_p), where q satisfies $1/p+1/q=1$.*

Proof of Theorem 3. Consider the randomized mechanism displayed in Algorithm 2. We claim that

$$\mathbf{E}_{\bar{m},\boldsymbol{\omega}^{1:T}}[F_k(\boldsymbol{x}^{\mathcal{G}\text{-}\mathcal{S}\text{-}\phi_p,1:T},\boldsymbol{y}^{\mathcal{G}\text{-}\mathcal{S}\text{-}\phi_p,1:T},\boldsymbol{\omega}^{1:T})]=\frac{1}{M}\mathbf{E}\left[\sum_{m=1}^{M}F_k(\boldsymbol{x}_m^{\mathcal{G}\text{-DAO}_p,1:T},\boldsymbol{y}_m^{\mathcal{G}\text{-DAO}_p,1:T},\boldsymbol{\omega}_m^{1:T})\right]$$

holds for each $k\in\{1,2,\ldots,K\}$. The expectation on the right hand side is taken over $(\boldsymbol{\omega}_m^1,\boldsymbol{\omega}_m^2,\ldots,\boldsymbol{\omega}_m^T)$, which are i.i.d. as $(\Psi^1,\Psi^2,\ldots,\Psi^T)$, respectively, and on the left hand side we also have $\boldsymbol{\omega}^t\sim\Psi^t$ for $t=1,2,\ldots,T$. Consequently, conditional on $\bar{m}=m$, the equality holds. The remaining proof is similar to the argument in Theorem 1. ■

B. Details of Different Matching Policies

As detailed in Algorithm 3, we set the penalty function as ϕ_2 in model ($\mathcal{G}$-$\mathcal{M}$-ϕ_2) and apply policy $\mathcal{G}$-DAO$_2$ to develop the compromise matching (CM) policy in a multi-period ride-matching context.

Algorithm 3 Compromise Matching (CM) Policy

1: Initialize: KPI target $\boldsymbol{U}=(U_1,U_2,\ldots,U_K)$; Bootstrap $\hat{M}$ ride-matching samples $\{\boldsymbol{\omega}_m^{1:T}\}_{m=1}^{\hat{M}}$.

2: For each sample $m=1,2,\ldots,\hat{M}$:

a. Adaptively compute weights for the objective function for $m>1$:

$$w_k(m)=U_k-\frac{1}{m-1}\sum_{s=1}^{m-1}\left\{\sum_{t=1}^{T}f_k\left(\boldsymbol{x}_s^{\hat{\text{CM}},t},\boldsymbol{y}_s^{\hat{\text{CM}},t},\boldsymbol{\omega}_s^t\right)\right\}.$$

We set $\boldsymbol{w}(1)=(1,1,\ldots,1)$ as the starting point for $m=1$.

b. Obtain the optimal solution $\{\boldsymbol{x}_m^{\hat{\text{CM}},t},\boldsymbol{y}_m^{\hat{\text{CM}},t}\}_{t=1}^{T}$ to the following stochastic dynamic programming problem:

$$\begin{aligned}\max\ &\sum_{k=1}^{K}w_k^+(m)\mathbf{E}\left[\sum_{t=1}^{T}f_k\left(\boldsymbol{x}_m^t,\boldsymbol{y}_m^t,\boldsymbol{\omega}_m^t\right)\right]\\ \text{s.t.}\ &\boldsymbol{y}_m^t=\mathcal{Y}(\boldsymbol{x}_m^{t-1},\boldsymbol{y}_m^{t-1},\boldsymbol{\omega}_m^{t-1}),\ \forall\boldsymbol{\omega}_m^{t-1}\in\boldsymbol{\Omega}^{t-1},\ t=1,2,\ldots,T,\\ &\boldsymbol{x}_m^t\in\mathcal{X}(\boldsymbol{y}_m^t,\boldsymbol{\omega}_m^t),\ \forall\boldsymbol{\omega}_m^t\in\boldsymbol{\Omega}^t,\ t=1,2,\ldots,T\end{aligned}$$

where $w_k^+(m):=\max\{w_k(m),\epsilon\}$ and $\epsilon=10^{-4}$.

3: Collect a set of weight vectors $\{\boldsymbol{w}^+(m)\}_{m=1}^{\hat{M}}$.

4: For each day $m=1,2,\ldots,M$:

a. Sample an index $\bar{m}$ uniformly at random from $\{1,2,\ldots,\hat{M}\}$ with replacement.

b. Solve the stochastic dynamic programming problem:

$$\begin{aligned}(\text{CM}(m))\ \max\ &\sum_{k=1}^{K}w_k^+(\bar{m})\mathbf{E}\left[\sum_{t=1}^{T}f_k\left(\boldsymbol{x}_m^t,\boldsymbol{y}_m^t,\boldsymbol{\omega}_m^t\right)\right]\\ \text{s.t.}\ &\boldsymbol{y}_m^t=\mathcal{Y}(\boldsymbol{x}_m^{t-1},\boldsymbol{y}_m^{t-1},\boldsymbol{\omega}_m^{t-1}),\ \forall\boldsymbol{\omega}_m^{t-1}\in\boldsymbol{\Omega}^{t-1},\ t=1,2,\ldots,T,\\ &\boldsymbol{x}_t^m\in\mathcal{X}(\boldsymbol{y}_t^m,\boldsymbol{\omega}_t^m),\ \forall\boldsymbol{\omega}_t^m\in\boldsymbol{\Omega}^t,\ t=1,2,\ldots,T\end{aligned}$$

We denote $\{\boldsymbol{x}_m^{\text{CM},t},\boldsymbol{y}_m^{\text{CM},t}\}_{t=1}^{T}$ as the optimal solution to problem CM(m). The attained KPI along the k^{th} direction is denoted as $\sum_{t=1}^{T}f_k\left(\boldsymbol{x}_m^{\text{CM},t},\boldsymbol{y}_m^{\text{CM},t},\boldsymbol{\omega}_m^t\right)$ at day m.

5: Evaluate the average KPI: $\tau_k^{\text{CM}}:=\dfrac{1}{M}\displaystyle\sum_{m=1}^{M}\sum_{t=1}^{T}f_k\left(\boldsymbol{x}_m^{\text{CM},t},\boldsymbol{y}_m^{\text{CM},t},\boldsymbol{\omega}_m^t\right),\ k=1,2,\ldots,K.$

Notably, the weighted-sum matching problems at Step (2b) and (4b) can be treated as a "single"-objective stochastic dynamic matching problem, but it is in general challenging to obtain the Bellman optimal solution to the stochastic dynamic program due to the curse of dimensionality in the ride-matching problem. Therefore, we propose two heuristic policies for the ease of implementation:

- Myopic policy: At each matching interval t, this policy simply maximizes the immediate objective value at current interval t, while ignores the future dynamics.
- One-period Forward-looking policy: At each matching interval t, we incorporate the demand forecast for period $t+1$ to solve the matching problem at current interval t. In this way, we dispatch drivers to serve passengers from both interval t and $t+1$. Passengers from interval t are served immediately upon being matched; however, we only block drivers who are matched to passengers from $t+1$, and execute a second-round matching at interval $t+1$.

Compared with the myopic policy, one-period forward-looking policy utilizes more information to solve the multi-period dynamic matching problem. Table 3 in Section 5.4 shows that the forward-looking policy could possible Pareto dominate the myopic policy. To further improve the matching quality, we can also leverage on advanced MDP techniques (e.g., Xu et al. 2018) to solve the matching problem (CM(m)). We highlight that our CM policy provides a scheme to deal with the trade-offs between different objectives.

Next, we describe the benchmark policy—simplified compromise matching (S-CM) policy—in Algorithm 4. This policy assumes that the matching scenarios within each day m are i.i.d. generated, and updates the weight vector $\boldsymbol{w}(m,t+1)$ period by period within each day m.

Algorithm 4 Simplified Compromise Matching (S-CM) Policy

1: INITIALIZE: KPI target $\boldsymbol{U}' = \frac{1}{T} \times (U_1, U_2, \ldots, U_K)$.

2: For each day $m = 1, 2, \ldots, M$:

- For the bootstrapped matching scenario at each interval $t = 1, 2, \ldots, T$:

— Adaptively compute weights for the objective function for $t > 1$:

$$w_k(m,t) = U'_k - \frac{1}{t-1} \sum_{\iota=1}^{t-1} \left\{ f_k\left(\boldsymbol{x}_m^{\text{S-CM},\iota}, \boldsymbol{y}_m^{\text{S-CM},\iota}, \boldsymbol{\omega}_m^{\iota}\right) \right\}$$

We set $\boldsymbol{w}(m,1) = (1,1,\ldots,1)$ as the starting point for $t = 1$.

— Solve the following weighted matching problem:

$$\begin{aligned} (\text{S-CM}(m,t)) \ \max \ & \sum_{k=1}^{K} w_k^+(m,t) f_k\left(\boldsymbol{x}_m^t, \boldsymbol{y}_m^t, \boldsymbol{\omega}_m^t\right) \\ \text{s.t.} \ \ & \boldsymbol{y}_m^t = \mathcal{Y}(\boldsymbol{x}_m^{t-1}, \boldsymbol{y}_m^{t-1}, \boldsymbol{\omega}_m^{t-1}), \\ & \boldsymbol{x}_m^t \in \mathcal{X}(\boldsymbol{y}_m^t, \boldsymbol{\omega}_m^t) \end{aligned}$$

where $w_k^+(m,t) := \max\{w_k(m,t), \epsilon\}$ and $\epsilon = 10^{-4}$. We denote $\{\boldsymbol{x}_m^{\text{S-CM},t}, \boldsymbol{y}_m^{\text{S-CM},t}\}$ as the optimal solution to problem S-CM(m,t). The k^{th} KPI attained is denoted as $f_k\left(\boldsymbol{x}_m^{\text{S-CM},t}, \boldsymbol{y}_m^{\text{S-CM},t}, \boldsymbol{\omega}_m^t\right)$ at period t of day m.

3: Calculate the average KPI: $\tau_k^{\text{S-CM}} := \dfrac{1}{M} \displaystyle\sum_{m=1}^{M} \sum_{t=1}^{T} f_k\left(\boldsymbol{x}_m^{\text{S-CM},t}, \boldsymbol{y}_m^{\text{S-CM},t}, \boldsymbol{\omega}_m^t\right)$, $k = 1, 2, \ldots, K$.

Finally, we provide the details of the weighted-sum matching policy and the single-objective maximization policy in Algorithm 5. W.L.O.G., we use the weighted-sum matching policy for illustration, since the four single-objective policies are special cases with different weight functions $\{w_1, w_2, \ldots, w_K\}$.

Algorithm 5 Weighted-sum Matching (WM) Policy

1: For each day $m = 1, 2, \ldots, M$:

- Solve the stochastic dynamic programming problem:

$$\begin{aligned} (\mathrm{WM}(m)) \ \max \ & \mathbf{E}\left[\sum_{t=1}^{T}\sum_{k=1}^{K} w_k f_k\left(\boldsymbol{x}_m^t, \boldsymbol{y}_m^t, \boldsymbol{\omega}_m^t\right)\right] \\ \text{s.t.} \ \ & \boldsymbol{y}_m^t = \mathcal{Y}(\boldsymbol{x}_m^{t-1}, \boldsymbol{y}_m^{t-1}, \boldsymbol{\omega}_m^{t-1}),\ \forall \boldsymbol{\omega}_m^{t-1} \in \boldsymbol{\Omega}^{t-1},\ t = 1, 2, \ldots, T, \\ & \boldsymbol{x}_m^t \in \mathcal{X}(\boldsymbol{y}_m^t, \boldsymbol{\omega}_m^t),\ \forall \boldsymbol{\omega}_m^t \in \boldsymbol{\Omega}^t,\ t = 1, 2, \ldots, T \end{aligned}$$

  We denote $\{\boldsymbol{x}_m^{\mathrm{WM},t}, \boldsymbol{y}_m^{\mathrm{WM},t}\}_{t=1}^T$ as the optimal solution to problem WM(m). The k^{th} KPI attained is denoted as $f_k\left(\boldsymbol{x}_m^{\mathrm{SM},t}, \boldsymbol{y}_m^{\mathrm{SM},t}, \boldsymbol{\omega}_m^t\right)$ at period t of day m.

2: Calculate the average KPI: $\tau_k^{\mathrm{WM}} := \dfrac{1}{M}\displaystyle\sum_{m=1}^{M}\sum_{t=1}^{T} f_k\left(\boldsymbol{x}_m^{\mathrm{WM},t}, \boldsymbol{y}_m^{\mathrm{WM},t}, \boldsymbol{\omega}_m^t\right),\ k = 1, 2, \ldots, K.$

Similar to Algorithm 3, we can also apply myopic policy and one-period forward-looking policy to solve the problem (WM(m)) in Algorithm 5.

C. Robustness Check using an Industry Ride-Matching Simulator

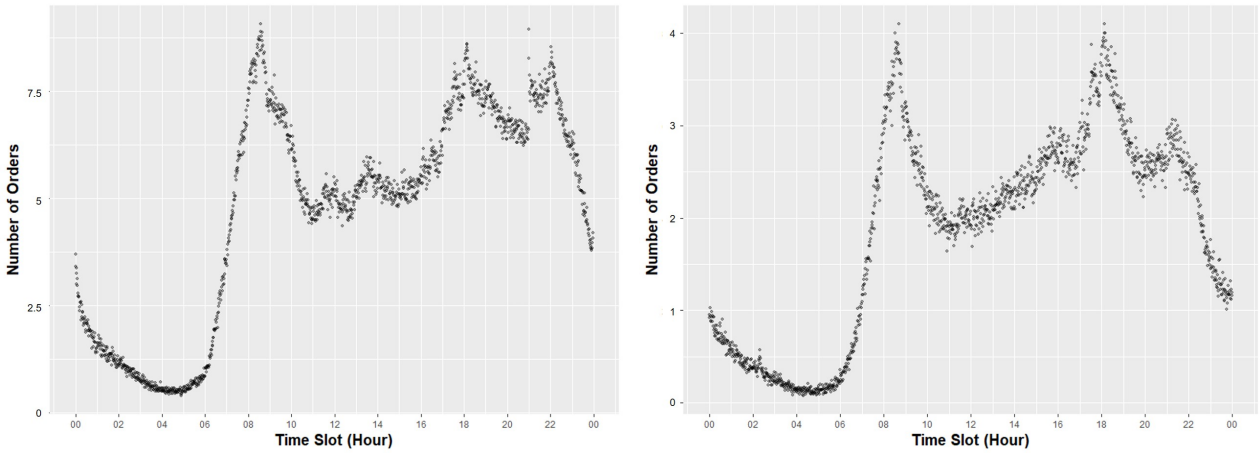
In this section, we investigate the performance of our compromise matching policy using a ride-matching simulator developed by our industry collaborator, with detailed simulation of the rides on the city streets. The ride-matching scenarios are simulated using the data from City B and City H. On the demand side, the matching simulator contains detailed order information on each travel request (by the passenger), including order booking time, latitude and longitude of the trip's origin and destination, revenue, etc. On the supply side, the matching simulator contains minute-level tracking information on all active drivers on the platform, including latitude and longitude of the driver's location, his/her service status, etc. Different from the case of City C, we have the exact service score information of each driver in City B and H.

As shown in Figure 7, the number of total orders in City B is more than twice of that in City H. However, the number of total registered drivers in City B is only 1.08 times that of City H. In other words, City B suffers from more serious driver shortage and hence drivers from City B have more choices to be dispatched under different policies. In addition, City B is larger than City H in terms of geographical size, and hence drivers in City B would take longer time to serve a passenger.

Compared to the matching simulator introduced in Section 5.2, this industrial simulator captures passengers' behavior on cancellation, driver status, and trip routing. More concretely:

- Passenger order information in each scenario is sampled from the order set. Passengers' patience levels (tolerance for waiting time before being assigned to a driver) are calibrated from historical data.

Figure 7 Temporal order distributions of City B and H. The absolute number of orders is normalized.



(a) City B (b) City H

- Driver status (inactive, idle active, and busy active), routing behavior, and travel time between each specific pair of origins and destinations have been considered in our simulator. For idle active drivers, a random walk is performed to simulate their cruising routes. For busy active drivers, their traveling paths are simulated using a virtual GPS routing system.
- The (weighted) matching problem is solved in a batch matching manner. The matching interval is set to be 2 seconds (because the demand scales in both City B and City H are much larger than the scale in City C). Unmatched drivers and passengers are delayed to the next period until they leave the market.
- Drivers and orders are generated from the same dataset so that the ride-sourcing environments can be controlled to be similar for different matching policies.

Note that our S-CM policy can also balance the trade-off between different objectives (cf. Table 1 and Figure 3), and hence we implement the S-CM policy for the ease of implementation in the industrial simulator. We highlight that the S-CM policy is a purely online policy and can be embedded into the industrial simulator directly. Furthermore, we take three KPI components—the revenue, pick-up distance, and service quality—into the design of the S-CM policy. Since the revenue maximization policy and pair maximization policy reveal similar performance, we do not consider the matched pair component in this section. This simplification also provides a new angle of robustness check for the S-CM policy.

We compare the performance of our S-CM policy with the one used in the industrial simulator (denote as 'Legacy Policy'), as well as the popular "closest distance" policy (CD Policy) that is widely studied in academic literatures (e.g., Özkan and Ward 2020). The legacy policy is a MDP-based policy and the objective is to optimize a weighted-sum reward function to maximize revenue. Our collaborated matching team trained a proper set of hyperparameters to characterize the reward function by aggregating multiple components (each component is assigned with a fixed weight). Indeed, this industrial policy is the most natural benchmark for our comparison since it was carefully designed to address the multi-objective ride-matching problems. We remark that the matching team provides us the legacy policy with some confidential components removed,

and then we incorporate this policy and develop numerical experiments in the industrial simulator. Aside from the confidential components removed from the legacy policy, we also need to clarify that there still exist some gaps between the legacy policy (with the pre-modified version embedded in the industrial simulator) and the "exact" policy implemented in the platform (App). In fact, the matching team has done numerous experiments to validate that the performance of this legacy policy (in the industrial simulator) is very close to the performance of the exact policy (in the real-life platform with numerous factors, uncertainty, and noise). Therefore, when the company plans to roll out any new matching policies, the team will first test the performance of this new idea in the industrial simulator and set the legacy policy as benchmark for comparison. In a similar way, we tested our S-CM policy using this industrial simulator, and our numerical experiments were conducted in collaboration with the matching team. We also tested other policies such as the RM Policy, SM Policy, and WM Policy (with identical weight), but their performances are far from satisfactory in the industrial simulator.

In the following implementations, the legacy policy is set as the benchmark for the other two policies. Based on numerical comparisons, we observe that the performances of legacy policy and CD policy are similar. Namely, the pick-up distance is on the top priority of the legacy policy even though the set of hyperparameters has been carefully calibrated so as to maximize the total revenue. We follow the suggestions from our collaborator to report five indicators to evaluate the performance of different matching policies, including Revenue (per trip), Pick-up Distance (per trip), Service Quality (per trip), Total Answer Count, and Total Revenue. In particular, we also report the relative difference, i.e., improvement or deterioration, for each KPI for other two policies. For Revenue (per trip), Service Quality (per trip), Total Answer Count, and Total Revenue, we report the difference in terms of percentage since the absolute value has been normalized to avoid some confidential issues. For Pick-up distance, we report the difference in terms of absolute value.

We implement and compare the performances of three policies over a whole day for both cities (i.e., simulations start at 00:00 a.m. and end at 23:59 p.m.). For ease of implementation, we estimate a static KPI target for the whole day. For example, when we implement the S-CM policy on Friday, we estimate the KPI target based on the historical records of previous Fridays with similar weather conditions, given that the ride-sourcing records persist a significant day-of-week pattern. Results are summarized in Table 4. Consistent with the results in Section 5.4.2, the S-CM policy obtains a delicate balance between multiple objectives and brings value to all the stakeholders in the ride-sourcing ecosystem: passengers, drivers, and the platform.

For Drivers: Compared to the benchmark policy, the S-CM policy improves service quality per trip by 0.92% and 0.29% for City B and H, respectively. This implies more jobs are assigned to drivers with higher service quality. Figure 8 demonstrates that expected total revenue earned by drivers with higher service scores (e.g., higher than 101) increases under the S-CM policy. We also find that the revenue increment for these drivers is indeed due to more orders being dispatched to them. This outcome would motivate drivers to increase their service score by providing better ride-sourcing service to passengers. In addition, we observe a decreasing trend in total revenue for these drivers with extreme high service scores. One possible explanation is that a large proportion of drivers are part-time and their revenue also depends on their total business

Table 4 **Implementation of three matching policies in the ride-sourcing simulator: (+) indicates improvement while (-) for decline, compared with the benchmark Policy.**

City	Policy	Revenue	Pick-up Dis. (km)	Service	Answer Count	Total Revenue
City B	Legacy	10.00[a]	0.65	100.00[b]	10000[c]	100,000.00
	S-CM	10.19	0.70	100.92	9842	100,264.40
	(%)	1.88% (+)	0.05 (-)	0.92% (+)	-1.58% (-)	0.26% (+)
	CD	10.00	0.65	100.00	10002	99,997.56
	(%)	-0.02% (-)	0.00 (-)	-0.00% (-)	0.02% (+)	-0.00% (-)
City H	Legacy	7.03	0.80	102.59	3750	26,377.90
	S-CM	7.13	0.83	102.89	3722	26,525.41
	(%)	1.33% (+)	0.03 (-)	0.29% (+)	-0.76% (-)	0.56% (+)
	CD	7.04	0.81	102.60	3753	26,403.13
	(%)	0.03% (+)	0.01 (-)	0.00% (+)	-0.06% (-)	0.10% (+)

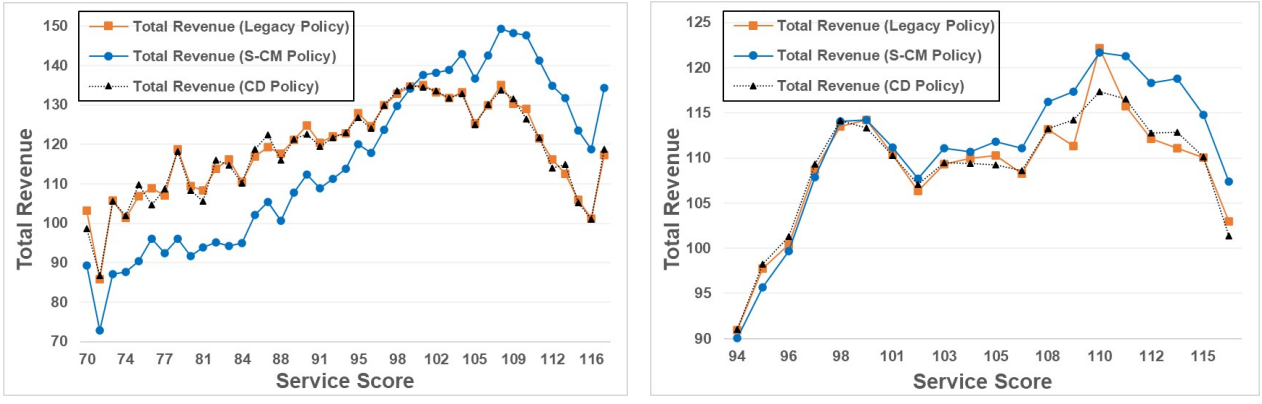
[*a*] The revenue under legacy policy in City B is normalized to be 10.00. We normalize the remaining parameters associated with revenue in relative to this basis.

[*b*] The service score under legacy policy in City B is normalized to be 100.00. We normalize the remaining parameters associated with service score in relative to this basis.

[*c*] The answer count under legacy policy in City B is normalized to be 10000. We also normalize other answer counts and total revenue amounts in relative to this basis.

hours (i.e., active time as a driver on the platform). The dataset reveals this pattern: these drivers with service scores in the interval [98, 108] are more active than the ones with scores in the interval [109,116]. Even so, our S-CM policy dispatches more orders to these drivers with higher service scores consistently. We remark that this result is slightly different from the one in Figure 4(a) as the service scores are randomly generated for the drivers in City C.

Figure 8 **Total revenue per day earned by drivers with different service scores: total revenue for each driver is calculated by aggregating her/his income over the whole day. We plot total revenue for drivers belonging to the same service score interval. The absolute total revenue and service score are normalized according to the baselines [*a*] and [*b*], respectively (cf. Table 4).**

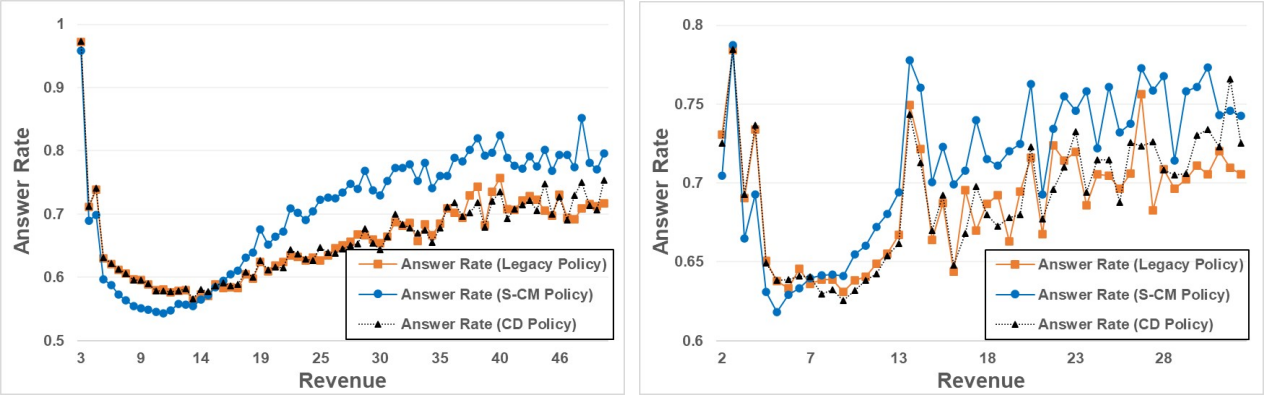


(a) City B (b) City H

Note. We highlight that the ranges of the service scores in City B and H are different from the range in City C due to different normalization rules.

For Passengers: We observe that the revenue per trip increases by 1.88% and 1.33% in City B and H, respectively. In other words, more drivers are dispatched to serve these passengers with higher revenue (i.e., longer travel distance), by performing the S-CM policy. Figure 9 demonstrates that passengers with higher revenue are served with higher priority under the S-CM policy (when the order's revenue exceeds the cross point), and their answer rates increase significantly. We observe that passengers with shorter travel distances suffer from decreasing answer rates under three policies. In fact, more short-travel orders are displayed during peak hours when the number of available drivers is insufficient to serve all these passengers. Therefore, their answer rates become lower.

Figure 9 **Answer rates of passengers with different revenue: The answer rate is calculated by (number of orders being matched)/(number of total orders) for passengers belonging to the same fare interval. The answer rate measures the probability of being served. The revenue is normalized according to the baseline $[a]$ in Table 4.**

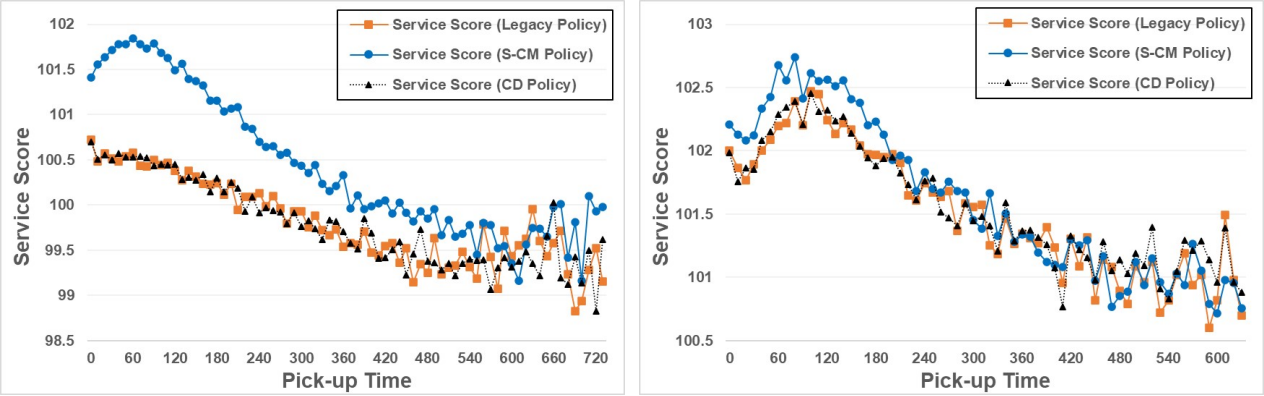


(a) City B (b) City H

In addition, on the passenger side, although the average pick-up distance (pick-up time) increases under the S-CM policy, the service quality increases for these passengers being matched, as shown in Figure 10. Therefore, passengers are also better off as they can enjoy their trips with higher service quality.

For the platform: As a side effect, although the pick-up distance per trip increases by 55.20 meters and 33.51 meters, and the answer count reduces by 1.58% and 0.76%, the total revenue for the whole day under the CM policy still increases by 0.26% in City B and 0.56% in City H. More precisely, Figure 11 plots the total revenue increment over the whole day. We observe that the increment is more significant during those periods when drivers start their working shifts, i.e., when drivers switch their inactive status to active. Note that our simulation starts from 0:00 a.m., and the initial active driver pool is identical for all matching policies. Compared to the benchmark policy, the S-CM policy tends to match idle active drivers to those passengers with higher revenue, and hence drivers must take longer to serve such orders. As a result, the number of idle active drivers decreases in subsequent periods. The total revenue increment reaches around 1.00% during the first three hours (0:00 a.m. to 2:59 a.m.), but decreases between 3:00 a.m. and 5:59 a.m. From 6:00 a.m. to 9:00 a.m., many drivers start their working shifts, and hence the revenue increment increases again. Since

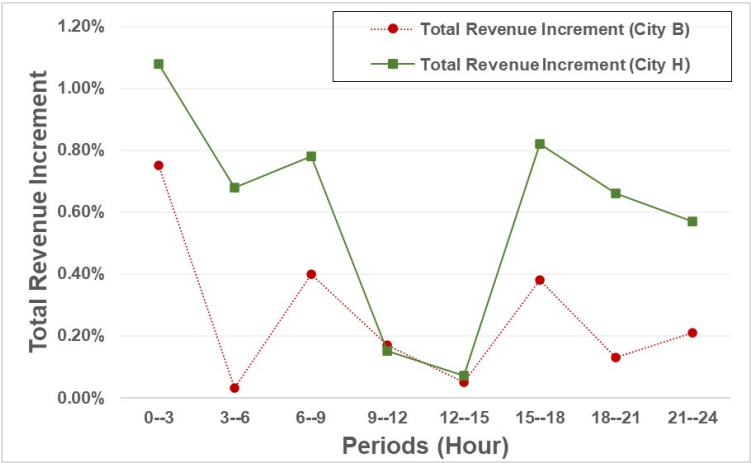
Figure 10 **Service quality per trip with different pick-up time: The service quality per trip is calculated by the average service scores (from driver side) attached to these trips belonging to the same pick-up time interval, which measures the pick-up distance. The service score is normalized according to the baseline [b] in Table 4.**



(a) City B (b) City H

many drivers work part-time in the ride-sourcing market and must work full-time jobs during the daytime, many will leave the platform during the daytime and become active again after 17:00 p.m. Therefore, the revenue increment becomes much more significant between 15:00 p.m. and 17:59 p.m.

Figure 11 **Total revenue increment under the CM Policy in City B and H, compared to the benchmark policy.**



Overall, using the industrial ride-matching simulator, we validate that all parties in the ride-sourcing ecosystem, from drivers, passengers, to the platform, are better off under our proposed matching policy. This is consistent with the numerical results in Section 5.4.