Online Hedge Reservation for Diverse Plans and Competitive Analysis

Binghan Wu, Wei Bao, Dong Yuan, Bing Zhou
Faculty of Engineering, The University of Sydney
biwu6051@uni.sydney.edu.au, wei.bao@sydney.edu.au, dong.yuan@sydney.edu.au, bing.zhou@sydney.edu.au

Abstract

In this paper, we investigate the plan reservation problem with diverse plans in mobile networks. The pricing scheme includes: 1) Pay-as-you-go (PAYG) payment; 2) All-in-one plan: an upfront fee is charged to cover data volume of a period of time; and 3) Directional plan: an upfront fee is charged to cover data volume of a specific app for a period of time. We investigate online plan reservation with competitive analysis, as the data volume is not known until an app is used. The problem is challenging as there are multiple directional plans and one all-in-one plan, creating a large decision space and complicated correlations among the decisions. We propose the Online Hedge Reservation (OHR) Algorithm to solve the problem and prove that it achieves $e^{\beta}/(e^{\beta}-1)$ competitive ratio when each plan is valid till the end of each calendar month and $2e^{\beta}/(e^{\beta}-1)$ competitive ratio when each plan is valid for a full month, where β is the ratio of prices of the directional plans and the all-in-one plan. This is an exciting neat extension of the competitive ratio $e/(e-1)$ of the classic ski-rental problem. Finally, trace-driven simulation is conducted to further verify the advantages of the OHR Algorithm.

Introduction

With the fast advances of mobile applications, sponsored data plan (SDP) was recently adopted as a win-win-win solution to Internet Service Providers (ISPs), content providers (CPs), and end users. CPs provide users with network traffic subsidies, which can attract more users for CPs, reduce users' costs, improve ISPs' revenue, and motivate them to increase investment in the infrastructure (Zhang and Wang 2014; Zhang, Wu, and Wang 2015). As a typical and mature form of SDP, directional plan is widely adopted. A directional plan will charge an upfront fee. Then, during the active period of the plan, there is no cost for the traffic generated by apps of one CP, as long as the traffic does not reach a cap. Table 1[1] shows an example of directional plans in the real world.

Although the benefits of directional plans to a population of users have been studied and verified in the perspective of network economy, as a *single* user, it is still unclear how to purchase them in the *temporal dimension*. The challenge is that the user's future data traffic is usually non-predictable, but the plans should be reserved in advance. Especially, ISPs will also provide pay-as-you-go (PAYG) and all-in-one plan, as shown in Table 1, which greatly challenge users' decision on plan reservation. For example, a user consumes 450 MB on TikTok watching short videos and 50 MB on other apps (without directional plans) in the first month. The best payment method is to order ByteDance Plan and cover the rest 50 MB with PAYG. The total cost is CNY 23.5 (USD 3.35). In contrast, covering all data volume by the all-in-one plan is a much worse decision which will cost CNY 128 (USD 18.29), i.e., about 540% of the optimal cost. However, as the users' interests may drastically change (Rosenfelder 2018), the optimal reservation may not work in the next month. For example, the user starts to play a popular mobile game supported by Baidu in the following month, which can be covered by Baidu Plan. Only 100 MB is used by TikTok, but 500 MB is consumed by the game. If the user keeps using the old best payment method in the last month, the cost will be CNY 154 (USD 22) while using the ByteDance plan and Baidu plan saves about 88% in this time.

In this paper, we are motivated to investigate the online plan reservation scheme. Given the available directional plans and the all-in-one plan, the objective is how to reserve them to minimize the overall cost. However, the user's future data usage is not known in advance (Rosenfelder 2018; Zang et al. 2019; Unterbrunner et al. 2009), so that we focus on the online solution with competitive analysis (Komm 2016). Nevertheless, the challenge is three-fold: (1) There are M directional plans and an all-in-one plan to choose, which create a large decision space. (2) Some apps have directional plans but others do not, causing unbalanced decisions among different apps. (3) There is a complicated correlation between the all-in-one plan and directional plans. The reservation of a directional plan affects the decision on the all-in-one plan, and vice versa. However, it is difficult to quantify how they will affect each other.

To address the aforementioned issues, we propose the Online Hedge Reservation (OHR) Algorithm with two randomized hedge values. When PAYG cost in the current billing cycle reaches the thresholds, the directional plan and the all-in-one plan will be reserved respectively. The hedge values are randomized, following our designed probability distribu-

[1]https://shop.10086.cn/list/146%5f100%5f100%5f2%5f1%5f0%5f0.html

Payment method	Covered apps	Fee
Pay-as-you-go	N/A	CNY 0.29 (USD 0.04)/MB
5G Smart Enjoy Plan	All Apps	CNY 128 (USD 18.4)$/Month
ByteDance Plan	Today's Headlines, TikTok, Volcano Video	CNY 9 (USD 1.3)/Month
Alibaba Plan	Taobao, Tmall, Alipay, YouKu, Koubei, Xianyu, MayiWealth	CNY 9 (USD 1.3)/Month
Baidu Plan	Baidu, Baidu Map, Mobile Assistant, Baidu Keyboard	CNY 9 (USD 1.3) /Month

Table 1: China Mobile Plans

tions, and thus risks caused by the uncertain future are counteracted, leading to a quantified performance bound (i.e., competitive ratio). We consider two realistic plan settlement modes: (1) non-extension plans, each plan is valid till the end of each calendar month; and (2) extension plans, each plan is valid for a full month since it is reserved. Our competitive analysis firstly shows that the competitive ratio of the OHR Algorithm is $\frac{e^\beta}{e^\beta - 1}$ for the non-extension mode, where β is the ratio of prices between the directional plan and the all-in-one plan. This is an exciting neat extension of the competitive ratio $\frac{e}{e-1}$ of the classic ski-rental problem (Karlin, Kenyon, and Randall 2003). [2] For the extension mode, it additionally introduces complicated temporal correlations, as the plans are no longer independent in each calendar month. We prove that the OHR Algorithm still works well in this mode with a competitive ratio of $\frac{2e^\beta}{e^\beta - 1}$.

Finally, an intensive trace-driven simulation is conducted to verify the theoretical results. We compare the performance of OHR with benchmarks. The results show that OHR outperforms all of them in the series of simulation.

Related Work

Sponsored Data Plan

Sponsored data plans (SDPs) in two-sided markets (Armstrong 2006) have attracted attention in academia and industry over recent years. Several telecom companies have already provided sponsored data plans. The motivation of the sponsored data plan is to introduce more profits to both ISPs and CPs by attracting more users, to finally reach a win-win-win situation.

[2] The classic ski-rental problem studies whether to pay by PAYG (rent) or upfront fee (buy) for one commodity. The non-extension mode studied in this paper is equivalent to the ski-rental problem if there is one app only and one directional plan is available for this app.

Different from the scope of this work, one typical class of research investigates SDPs as games among users, ISPs, and CPs. For example (Hande et al. 2009; Andrews et al. 2014) studied the impact of SDPs on market competition. (Njoroge et al. 2014) studied the impact of neutral and non-neutral networks, and it showed that the neutral networks motivate CPs' participation. (Ma 2016) showed that CPs' competitive subsidy to users will increase the ISP's investment in network infrastructure. (Zhang, Wu, and Wang 2015) employed the two-stage Stackelberg Game to study the impact of sponsored data plan on competition among CPs suggesting that small CPs are benefited in the short term and large CPs are benefited in the long term. Another work (Zhao et al. 2020) additionally considered user diversity preferences. It is believed that the competitive advantage of large CPs will be weakened after considering diversified demand.

The above category of studies addressed the question: why does SDP exist and how is the price determined as a consequence of games among ISPs, CPs, and (a large number of) users. Different from them, we focus on the problem that given that SDPs are set, how to reserve them in an online fashion with unknown future, in the perspective of a single user.

Ski-Rental and Applications

Online algorithms are widely used to solve problems where the future is unknown or adversary. The ski-rental problem (Karlin, Kenyon, and Randall 2003), which is to decide whether to "buy" (upfront cost) or "rent" (PAYG cost) without the knowledge of future, is a typical one to be addressed by online algorithms and competitive analysis. This problem is also known as the Bahncard problem in a public transportation system (Fleischer 2001). Variants of ski-rental problems are investigated in recent years. (Shi et al. 2018) studied the ski-rental problem with nonlinear cost function. (Khanafer, Kodialam, and Puttaswamy 2013) allowed constraints (first or second moment) on the stochastic adversary. (Feldkord, Markarian, and Der Heide 2017) generalized the problem by considering fluctuations in rental prices. (Meyerson 2005) considered that the choices of purchase are limited in time. (Wang, Li, and Wang 2020) considered multiple shops, and the customer has to stick with one shop to rent or buy. (Zhang and Conitzer 2020) considers the ski-rental problem with multiple desired resources. The cost of requiring a new resource is a submodular function of the set of the resources that have not been purchased. (Wang, Li, and Wang 2020) considered the predictions by machine-learning approaches.

Variants of ski-rental problems are also investigated in real-world Information and Communications Technology (ICT) systems. (Yang, Pan, and Liu 2019; Wang, Liang, and Li 2015) focused on the cloud instance reservation problem (e.g. whether to reserve with upfront or rent it on demand), and gave the online solutions with competitive ratio. (Dinh et al. 2020) studied how edge nodes rent or buy remote resources to meet the capacity requirement. (Zang et al. 2019) developed an online ski-rental algorithm considering both computation instance reservation and communication channel reservation. A range of caching problems can also be formulated as ski-rental problems. (Bansal, Buchbinder, and Naor 2012) studied the generalized caching problem with cache size K and developed a randomized online algorithm. In (Rosenthal and Veloso 2012), a robot is waiting for human assistance. It uses the ski-rental algorithm to decide whether to wait in place or actively go to the office to find someone for help.

However, the above studies are substantially different from our study in terms of dimensions, hierarchy of choices, and temporal correlations: in this work, (1) the number of apps is an arbitrary integer, instead of 1 or 2; (2) the plans are hierarchical, where the all-in-one plan can further cover directional plans; (3) some apps do not have directional plans and they can only be covered by PAYG or the all-in-one plan; (4) we further consider the extension mode, which adds more complicated temporal correlations among the plans. To the best of our knowledge, this is the first study investigating two-level (all-in-one plan plus directional plan) and multiple-dimension (multiple directional plans) online data plan reservation problem, based on the ski-rental approach.

Problem Formulation

System Overview

The system is operated in continuous time $t \in [0, T_{\max})$, where $T_{\max}$ remains unknown till the end. K apps are installed on a user's smart device. At t, the user may use one app, which generates data traffic, to be charged by its service provider. Please note that one app may also represent a cluster of apps operated by a same content provider. For example, Today's Headlines and TikTok are regarded as one app offered by ByteDance. For presentation convenience, we stick to the term "app" throughout the rest of this paper.

There are three charge options: (1) Pay-as-you-go (PAYG), (2) Directional plan, and (3) All-in-one-plan. (2) and (3) can be further divided into two modes detailed as follows.

(1) **PAYG**: Each 1 unit of data volume will incur a cost of Δp. Data volume of V will incur a PAYG cost of $V\Delta p$. We assume that data volume is an integer value.

(2a) **Directional plan** (non-extension mode): At time t, the user can subscript to a directional plan for one app k. It will incur an immediate upfront fee $C_d\Delta p$. In $[t, \lceil t/T \rceil \cdot T)$, the data volume generated by app k will not incur any further cost. This means that the plan is valid from t to the next multiple of T (e.g., a plan is valid till the end of each month).

(2b) **Directional plan** (extension mode): At time t, the user can subscript to a directional plan for one app k. It will incur an immediate upfront fee $C_d\Delta p$. In $[t, t+T)$, the data volume generated by app k will not incur any further cost. This means that the plan is valid for a complete T (e.g., a plan is valid for one month).

(3a) **All-in-one plan** (non-extension mode): At time t, the user can subscript an all-in-one plan for all apps. It will incur an immediate upfront fee $C_o\Delta p$. Then, from $[t, \lceil t/T \rceil \cdot T)$, the data volume generated by all apps will not incur any further cost.

(3b) **All-in-one plan** (extension mode): At time t, the user can subscript an all-in-one plan for all apps. It will incur an immediate upfront fee $C_o\Delta p$. Then, from $[t, t+T)$, the data volume generated by all apps will not incur any further cost.

In this paper, we consider two modes: ① Non-extension mode: (1), (2a), and (3a) are available; ② Extension mode: (1), (2b), and (3b) are available. Please note that we consider two modes since both of them are possible in reality.

Among the K apps, only M apps, $M \leq K$, will offer the three payment options. These apps are called *directional feasible* apps. For the rest of apps, directional plan is not offered, only PAYG and the all-in-one plan are offered. These apps are called *directional infeasible* apps. Please note that both directional feasible and directional infeasible apps are possible in reality. An example is given in Table 1, where three directional feasible apps are available, and all the rest of apps do not have directional plans.

We assume that C_d is uniform among different apps. This is because CPs should be deemed as fair competitors of the market to ensure network neutrality (Janevski 2019). Indeed, in the real example in Table 1, the same monthly fee of different CPs is applied.

At one time, the user uses one app and generate data traffic. For presentation convenience, such app usage is called a *task*. The task is labeled by $i = 1, 2, \ldots, I$ in chronological order. Task i can be presented by a 3-tuple: $(d(i), a(i), t(i))$, where $d(i)$ represents the data volume generated by task i. $a(i)$ represents the app index $a(i) \in \{1, 2, \ldots, K\}$. $t(i)$ represents the time when task i arrives. For any task i and task j, $i < j$ indicates $t(i) < t(j)$.

Due to the uncertainty of data traffic usage, we assume that users do not have any future information about data usage. For task i, the user only knows $(d(i), a(i), t(i))$ at time $t(i)$. This model does not assume that the sequence of the tasks respects any prior distribution, so this is an online system.

Since Δp remains the same for all costs, it is regarded as 1 without loss of generality in the rest of the paper, unless otherwise specified. We assume $C_d \gg 1$, as the price of a plan is much higher than that of a unit of data volume. We assume $C_o \geq 2C_d$ and $K \geq 2$ as the user uses at least two apps.

Cost Minimization

We aim to minimize the overall cost, such that each task is covered by PAYG, directional plan, or all-in-one plan.

Let τ_{kl} denote the time we purchase a directional plan of app k, where $l = 1, 2, \ldots, L_k$ denotes the lth purchase, and L_k denotes the total number of purchases.

Let τ_{0l} denote the time we purchase an all-in-one plan, where $l = 1, 2, \ldots, L_0$ denotes the lth purchase, and L_0 denotes the total number of purchases. $\mathbf{L}$ denotes $\{L_k\}, k = 0, 1, \ldots, M$ and $\boldsymbol{\tau}$ denotes $\{\tau_{kl}\}, k = 0, 1, \ldots, M, l = 1, 2, \ldots, L_k$.

We aim to minimize the overall cost $\min_{\boldsymbol{\tau}, \mathbf{L}} \sum_{i=1}^{I} \mathrm{PAYG}(i) + C_d \sum_{i=1}^{M} L_i + C_o L_0$, where $\mathrm{PAYG}(i)$ is the PAYG cost of the task i. $\mathrm{PAYG}(i) = 0$ if it is covered by a plan. Otherwise, it is equal to $d(i)$. For the non-extension mode, task i is covered by a directional plan if $\exists l$ such that $t(i) \in [\tau_{a(i)l}, \lceil \tau_{a(i)l}/T \rceil \cdot T)$; it is covered by an all-in-one plan if $\exists l$ such that $t(i) \in [\tau_{0l}, \lceil \tau_{0l}/T \rceil \cdot T)$. For the extension mode, task i is covered by a directional plan if $\exists l$ such that $t(i) \in [\tau_{a(i)l}, \tau_{a(i)l} + T)$; it is covered by an all-in-one plan if $\exists l$ such that $t(i) \in [\tau_{0l}, \tau_{0l} + T)$.

Online Algorithm and Performance Evaluation

Due to the online setting, we do not know the future. Hence, it is not possible to derive the minimum cost as an offline optimization problem. In this paper, we focus on online plan reservation, and evaluate the proposed algorithm by competitive analysis, which is commonly adopted for online algorithms. Let $s = \{(d(i), a(i), t(i))\}, i = 1, 2, \ldots, I$, be any input sequence of tasks. A randomized online algorithm ALG is called c-competitive if it satisfies the following inequality for all possible input sequences: $\mathbb{E}(\text{ALG}(s)) \leq c \cdot \text{OPT}(s)$, where $\text{ALG}(s)$ is the cost of algorithm ALG and we consider its expectation. $\text{OPT}(s)$ is the cost of the optimal offline algorithm when the input s is known.

A randomized online algorithm makes randomized decisions according to designed random distributions, and thus we need to consider its mean performance. Our proposed algorithm is a randomized online algorithm.

Online Hedge Reservation Algorithm

In this section, we introduce the Online Hedge Reservation (OHR) Algorithm. We first introduce two important concepts: typical cost and hedge values. Then, we formally propose the OHR Algorithm. Finally, we list the probability distribution of the hedge values.

Typical Cost

Before presenting the OHR Algorithm, we first define the *typical costs*. Suppose current time (now) is t, the *typical cost of app k at t*, $\theta_k(t)$, is defined as the PAYG costs of app k generated from the last multiple of T till now, i.e., in $[\lfloor t/T \rfloor \cdot T, t]$, if we do not reserve any plan now. The *overall typical cost at* t, $\theta_o(t)$, is defined as the overall PAYG cost of all apps generated since the last multiple of T till now, i.e., in $[\lfloor t/T \rfloor \cdot T, t]$, if we do not reserve any plan now.

$$\theta_k(t) = \sum_{\substack{i: \lfloor t/T \rfloor \cdot T \leq t(i) \leq t \text{ and } a(i)=k \\ \text{task } i \text{ is not covered by plan reserved before } t}} d(i), \qquad (1)$$

$$\theta_o(t) = \sum_{\substack{i: \lfloor t/T \rfloor \cdot T \leq t(i) \leq t \\ \text{task } i \text{ is not covered by plan reserved before } t}} d(i). \qquad (2)$$

Please note that we may reserve a plan at t when a task arrives, but its PAYG fee $d(i)$ can be counted in the typical cost.

Hedge Values

We define γ_d and γ_o as two hedge values. If the typical cost of (directional feasible) app k reaches γ_d, we reserve a directional plan of app k. If the overall typical cost reaches γ_o, we reserve an all-in-one plan.

At time t, the typical cost reflects the user's recent PAYG cost, implying the user's tendency to use app k (or all apps).

Algorithm 1: Online Hedge Reservation (OHR)

1 **if** *current time is a multiple of* T **then**
2   (Re)generate γ_d and γ_o according to PMFs in (3)–(4).
3 **end**
4 **if** *new arrival of task i:* $(d(i), a(i), t(i))$ **then**
5   Set θ_o be the overall typical cost calculated in (2)
6   **if** *There is no active all-in-one plan* **then**
7   **if** $\theta_o \geq \gamma_o$ **then**
8   Reserve all-in-one plan.
9   **else**
10   **if** *If* $a(i)$ *is directional infeasible* **then**
11   Pay the task i's data by PAYG.
12   **else**
13   **if** *There is no active directional plan for app* $a(i)$ **then**
14   Set $\theta_{a(i)}$ be the typical cost by (1).
15   **if** $\theta_{a(i)} \geq \gamma_d$ **then**
16   Reserve directional plan of app $a(i)$.
17   **else**
18   Pay the task i's data by PAYG.
19   **end**
20   **end**
21   **end**
22   **end**
23   **end**
24 **end**
25 Repeat from Line 1

We reserve plans if the typical costs reach the hedge values. However, since the future is completely unknown and could be adversarial, the worst case is that the app is never used after the reservation. To this end, values of γ_d and γ_o are designed as random values. Large values γ_d and γ_o will counteract the risk of small values, and vice versa. Therefore, they are called hedge values.

Online Hedge Reservation Algorithm

Algorithm 1 shows the OHR Algorithm. Whenever a task arrives, if is already covered by an active plan (Lines 6 and 13), no action is needed. Otherwise, the typical costs will be calculated (Lines 5 and 14). If they reach the hedge values, then a plan will be reserved (Lines 8 and 16). A directional infeasible app will always be covered by PAYG (Line 18). Also, the hedge values will be regenerated every T to avoid adaptive adversary sequence to "learn" the hedge values, causing worse performance (Komm 2016).

Please also note that Algorithm 1 is the same for both non-extension and extension modes. The duration of the plans will differ for the two modes, so that the typical costs are calculated differently and thus plans are reserved differently.

Distributions of γ_d and γ_o

The probability mass functions (PMFs) of γ_d and γ_o are

$$\mathbb{P}(\gamma_d = i) \triangleq P_i^{(d)} = \begin{cases} 1, & \text{if } i = C_d, \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

$$\mathbb{P}(\gamma_o = i) \triangleq P_i^{(o)} = \begin{cases} aq^{i-1}, \text{ if } i \in [1, C_d - 1], \\ \frac{C_o}{2C_o - 1} aq^{C_d - 2}, \text{ if } i = C_o, \\ 0, \text{ otherwise}, \end{cases} \tag{4}$$

where $q \triangleq \frac{C_o}{C_o - 1}$ and $a \triangleq (\frac{1-q^{C_d-1}}{1-q} + \frac{C_o}{2C_o-1} q^{C_d-2})^{-1}$.

Please note that this distributions will lead to $\frac{e^\beta}{e^\beta - 1}$ competitive ratio for the non-extension mode, and $\frac{2e^\beta}{e^\beta - 1}$ for the extension mode. Detailed derivations will be given in Appendix (Proof of Theorem 1).

Another observation is that the γ_d is equal to C_d with probability 1. This is derived by the analysis in Appendix (Proof of Theorem 1). We do not assume γ_d is deterministic at the beginning, but deterministic γ_d will lead to the nice form of bound, and will give a local minimum competitive ratio compared with other PMFs. This means the random distribution of γ_o is already good enough to counteract risks. There is no need to further randomize γ_d.

Useful Definitions for Further Analysis

In what follows, we introduce some important definitions to analyze the competitive ratio. Especially, the analysis will need to consider the realizations when γ_d and γ_0 are given. Then, we can quantify the mean performance by averaging through the PMFs of γ_d and γ_0.

Definition 1 (Algorithm output cost)**.** *For a sequence s, and given γ_d and γ_o, let $ALG^{(n)}(s, \gamma_d, \gamma_o)$ (resp. $ALG^{(e)}(s, \gamma_d, \gamma_o)$) denote the overall cost generated by the OHR Algorithm if γ_d and γ_o are fixed for non-extension plans (resp. for extension plans).*

Definition 2 (Optimal cost)**.** *For a sequence s, let $OPT^{(n)}(s)$ ($OPT^{(e)}(s)$) denote the minimum cost to cover all the tasks if s is known in advance, for non-extension plans (resp. for extension plans).*

Definition 3 (Deterministic competitive ratio)**.** *For a sequence s, and given γ_d and γ_o, the deterministic competitive ratio of the OHR Algorithm with non-extension and extension plans are $R^{(n)}(s, \gamma_d, \gamma_o) \triangleq \frac{ALG^{(n)}(s,\gamma_d,\gamma_o)}{OPT^{(n)}(s)}$, and $R^{(e)}(s, \gamma_d, \gamma_o) \triangleq \frac{ALG^{(e)}(s,\gamma_d,\gamma_o)}{OPT^{(e)}(s)}$, respectively.*

Analysis for Non-Extension Plans

In this section, we focus on the non-extension mode. Plans will terminate at $iT, i = 1, 2, \ldots$. As a consequence, the plan reservation and cost analysis in each $[iT, (i+1)T), i = 0, 1, \ldots$ is independent. Without loss of generality, we study the time range $[0, T)$ in this section, which is representative to sequence with any length. A sequence from 0 to arbitrary $T_{\max}$ can be divided in to sub-sequences in $[iT, (i+1)T)$ and each sub-sequence follows the same analysis. Please note that such setting follows a range of ski-rental problems, such as (Wang, Liang, and Li 2015; Zang et al. 2019). Therefore, we focus on typical sequences, which is defined as follows.

Definition 4 (Typical sequence)**.** *If a sequence only has data volume in $[0, T)$, it is called a typical sequence.*

Pruning the Typical Sequence

For a typical sequence, we focus on its performance operated by Algorithm 1. Let s_0 denote a typical sequence. $V(k)$ is defined as its total data volume of app k. We establish "worse cases" to bound the performance.

First, we find that by moving data volume from one app to another, the cost of Algorithm 1 may increase but the optimal cost does not, so that we can derive upper bounds. We have the following two Lemmas.

Lemma 1. *Given any γ_d and γ_o, for any typical sequence s_0, if there exist two apps, say apps k and l, such that $V(k) < \gamma_d$ and $V(l) < \gamma_d$, then the deterministic competitive ratio will not decrease if we move $\min(\gamma_d - V(k), V(l))$ data volume from app l to app k. In other words, let s_0' be a sequence as we move $\min(\gamma_d - V(k), V(l))$ data volume from app l to app k. We have*

$$\frac{ALG^{(n)}(s_0, \gamma_d, \gamma_o)}{OPT^{(n)}(s_0)} \leq \frac{ALG^{(n)}(s_0', \gamma_d, \gamma_o)}{OPT^{(n)}(s_0')}. \tag{5}$$

The proof is shown in Appendix.

Lemma 2. *Given any γ_d and γ_o, for any typical sequence s_0, if there exists two apps, say apps k and l, such that $V(k) < \gamma_d$ and $C_d > V(l) > \gamma_d$, then the deterministic competitive ratio will not decrease if we move $\min(\gamma_d - V(k), V(l) - \gamma_d)$ data volume from app l to k. In other words, let s_0' be a sequence as we move $\min(\gamma_d - V(k), V(l) - \gamma_d)$ data volume from app l to app k. We have*

$$\frac{ALG^{(n)}(s_0, \gamma_d, \gamma_o)}{OPT^{(n)}(s_0)} \leq \frac{ALG^{(n)}(s_0', \gamma_d, \gamma_o)}{OPT^{(n)}(s_0')}. \tag{6}$$

The proof is shown in Appendix.

By applying the above two Lemmas, we can construct a pruned sequence by moving data volume among the apps, and the deterministic competitive ratio is not reduced. Then, we can focus on analyzing the pruned sequence, which provides performance bound of any typical sequence.

For any given typical sequence s_0, we can apply the following procedure:

1. We create B directional infeasible apps with 0 data volume, as they will not influence the performance. B is sufficiently large.
2. While there exist two apps k and l such that $\gamma_d < V(l) < C_d$ and $V(k) < \gamma_d$, we move $\min(\gamma_d - V(k), V(l) - \gamma_d)$ data volume form l to k. $V(k) < \gamma_d$ can always be found as we create a large number of apps with 0 volume in Step 1).
3. While there exist two apps k and l s.t. $V(k) < \gamma_d$ and $V(l) < \gamma_d$, then move $\min(V(l), \gamma_d - V(k))$ data volume form l to k.

4. The outcome sequence is s_1, which is called a *pruned typical sequence*.

We further define $s_1 = \mathcal{P}(s_0)$ to denote s_1 is a pruned sequence of s_0, and $\mathcal{S}_1$ denote the set of any pruned sequence.

It is straightforward to show that a pruned typical sequence satisfies the following properties

- m directional feasible apps have data volume no less than C_d, where $m \geq 0$.
- n directional feasible apps have data volume equal to γ_d, where $n \geq 0$.
- a directional infeasible apps have data volume no less than C_d, where $a \geq 0$.
- b directional infeasible apps have data volume equal to γ_d, where $b \geq 0$.
- At most 1 app has data volume x in $(0, \gamma_d)$.
- All other apps have 0 data volume.

From Lemmas 1 and 2, and the above procedure to generate the pruned typical sequence, we know Steps 2) and 3) will not reduce the deterministic competitive ratio. Thus the deterministic competitive ratio of the pruned sequence will not be reduced compared to the original typical sequence. Therefore, it is straightforward to reach the following Lemma.

Lemma 3. *Given γ_d and γ_o, s_0 is any typical sequence, and s_1 is the pruned sequence generated by s_0. $s_1 = \mathcal{P}(s_0)$. Then, we have*

$$\frac{ALG^{(n)}(s_0, \gamma_d, \gamma_o)}{OPT^{(n)}(s_0)} \leq \frac{ALG^{(n)}(s_1, \gamma_d, \gamma_o)}{OPT^{(n)}(s_1)}. \tag{7}$$

In what follows, we can focus to bound the performance of a pruned typical sequence instead of an arbitrary sequence.

Bounding Pruned Typical Sequence

For a pruned typical sequence, we first investigate the deterministic competitive ratio when $\gamma_d \leq N$ and $\gamma_d \leq \gamma_o$. N is the total data volume.

Lemma 4. *Given γ_d and γ_o, s_1 is any pruned sequence with total data volume N. When $\gamma_d \leq N$ and $\gamma_d \leq \gamma_o$, the upper bound of the deterministic competitive ratio is $\frac{2C_o + C_d - 1}{\gamma_d}$.*

$$\frac{ALG^{(n)}(s_1, \gamma_d, \gamma_o)}{OPT^{(n)}(s_1)} \leq \frac{2C_o + C_d - 1}{\gamma_d}. \tag{8}$$

The proof is shown in Appendix.

By combining the above Lemmas, we can derive the competitive ratio.

Theorem 1. *The competitive ratio of Algorithm 1 in the non-extension mode is*

$$\frac{\mathbb{E}[ALG^{(n)}(s)]}{OPT^{(n)}(s)} \leq \frac{e^{\beta}}{e^{\beta} - 1}, \tag{9}$$

for any sequence s, where $\beta = \frac{C_d - 1}{C_o - 1} \simeq \frac{C_d}{C_o}$.

The proof is shown in Appendix.

Theorem 1 finally concludes the competitive ratio of Algorithm 1. This is a neat extension of the competitive ratio $\frac{e}{e-1}$ of the classic ski-rental problem (Karlin, Kenyon, and Randall 2003). Our problem is equivalent to the ski-rental problem if there is one app only and one directional plan is available for this app.

Analysis for Extension Plans

In this section, we focus on the extension mode. An extension plan is valid for a complete T since it is reserved. Different from the non-extension case, it will introduce temporal correlations among durations $[iT, (i+1)T), i = 1, 2, \ldots$. Therefore, the competitive analysis will be more challenging compared with the non-extension case. Additional efforts are required to bridge typical sequences and arbitrary sequences.

One observation is that if the input sequence is a typical sequence with only data volume in $[0, T)$, then the extension case is equivalent to non-extension case, and we have

$$\text{ALG}^{(e)}(s_0, \gamma_0, \gamma_d) = \text{ALG}^{(n)}(s_0, \gamma_0, \gamma_d), \tag{10}$$

$$\text{OPT}^{(e)}(s_0, \gamma_0, \gamma_d) = \text{OPT}^{(n)}(s_0, \gamma_0, \gamma_d), \tag{11}$$

for any typical sequence s_0. Through applying this property, in what follows, we prove that the deterministic competitive ratio of any sequence is bounded by that of typical sequences.

Lemma 5. *For any given γ_d and γ_o, and for any input sequence s, we can find a typical sequence s_0 satisfying the following bound*

$$\frac{ALG^{(e)}(s, \gamma_d, \gamma_o)}{OPT^{(e)}(s)} \leq 2\frac{ALG^{(e)}(s_0, \gamma_d, \gamma_o)}{OPT^{(e)}(s_0)} \tag{12}$$

$$= 2\frac{ALG^{(n)}(s_0, \gamma_d, \gamma_o)}{OPT^{(n)}(s_0)}. \tag{13}$$

The proof is shown in Appendix.

Therefore, we can see that the performance bound of any sequence can be bounded by twice of that of a typical sequence. Then by further combining the approach to reach Theorem 1, we can straightforwardly reach the following Theorem.

Theorem 2. *The competitive ratio of the Algorithm 1 in the extension mode is*

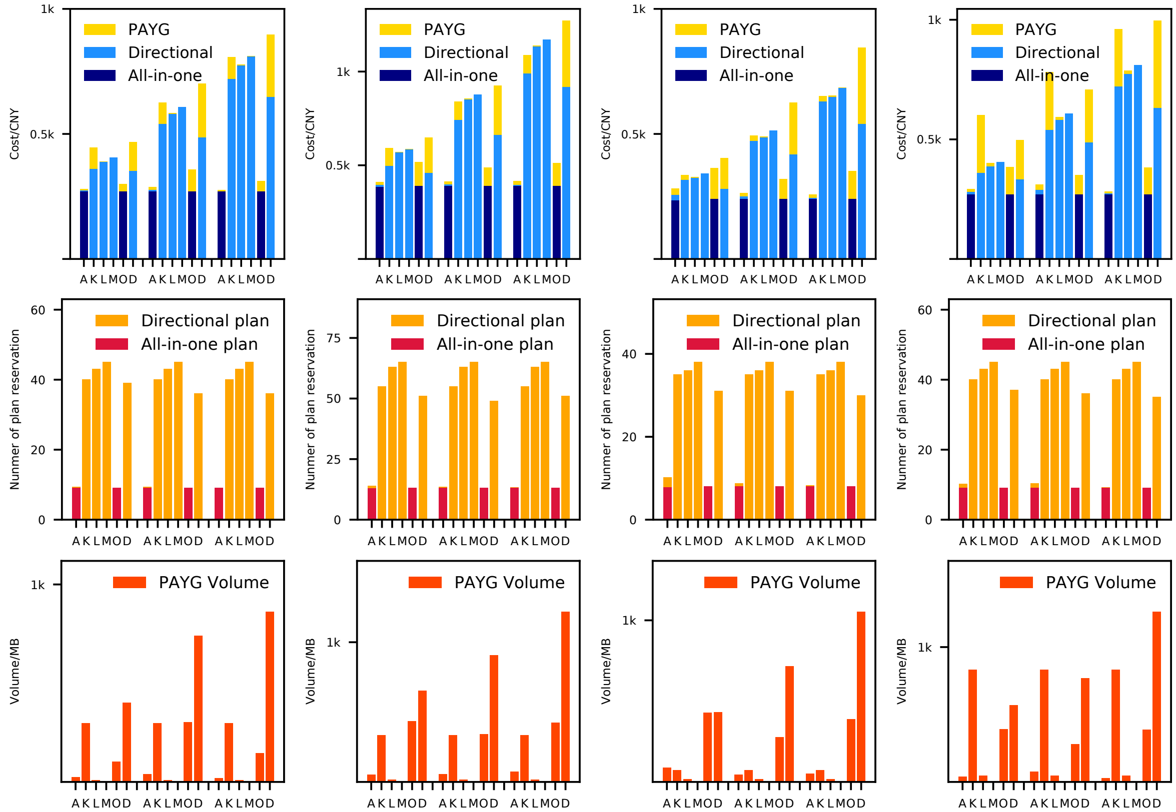
$$\frac{\mathbb{E}[ALG^{(e)}(s)]}{OPT^{(e)}(s)} \leq \frac{2e^{\beta}}{e^{\beta} - 1}, \tag{14}$$

for any sequence s.

Performance Evaluation

Trace-Driven Simulation

In this section, we use real-world trace data (Rojas 2018) to evaluate the performance of the OHR Algorithm. The data set contains 6-day network traffic, collected by packet sniffer tools. To the best of our knowledge, there is no publicly



(a) Non-Extension, $T = 1$ week. (b) Non-Extension, $T = 0.5$ week. (c) Extension, $T = 1$ week. (d) Extension, $T = 0.5$ week.

Figure 1: Performance comparison.

available data set to trace individual users' data usage (with labels of different types of application-layer traffic) for a sufficiently long time. (Rojas 2018) is the most similar one and we pre-process it to emulate one user's data traffic. The simulation is run on MacBook Air, 1.6GHz i5 Core, 8GB LPDDR3, a laptop-level computer.

Data Pre-processing: In the data set, each line of data is labeled with multiple features. We utilize timestamp, data volume, and L7Protocol. L7Protocol is to distinguish the upper layer apps. In the trace, there are 227 different L7protocols, and we randomly category them into 5 apps (5 groups of apps), where 4 of them have directional plans and the rest one does not. The collected data is from multiple users within a short period of time, and we scale the time from 6 days to 144 weeks to emulate the traffic of one user. The traffic density is reduced 168 times to represent a single user.

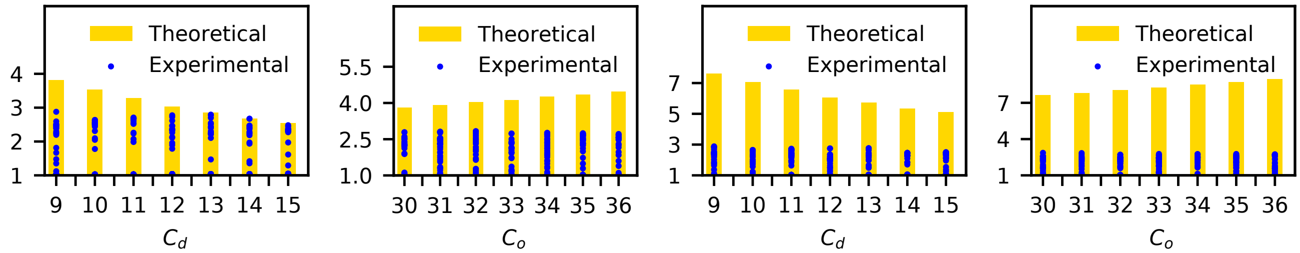
Plan Pricing: Other simulation parameters used in this section are obtained from real-world examples. According to the package tariff standard of China Mobile in 2020 [3], we set the pricing granularity as 1MB and PAYG price as Δp = CNY $0.29/\text{MB}$, all-in-one plan upfront fee as $C_o\Delta p$ = CNY 30, and directional plan upfront fee as $C_d\Delta p$ = CNY 9. $T = 1$ week. These values are regarded as default values. Without otherwise specified, these default values are employed throughout this section.

[3] https://service.bj.10086.cn/poffice/package/showpackage.action?PACKAGECODE=GPRSYW&productShowCode=sjllkxb

Benchmarks

We consider the following benchmarks:

- Directional Plan for Top 1 [K]: At $iT, i = 1, 2, \ldots$, the algorithm calculates the PAYG costs of different apps in the past T, and reserves the directional plan for the app with the highest PAYG cost. Others use PAYG.
- Directional Plan for Top 2 [L]: Similar to Algorithm [K], but it reserves directional plans for the two apps with the highest PAYG costs.
- Directional Plan for Top 3 [M]: Similar to Algorithm [K], but it reserves directional plan for the three apps with the highest PAYG costs.
- Ski-rental All-in-one Only [O]: Using the ski-rental algorithm(Karlin, Kenyon, and Randall 2003) by only allowing all-in-one plan.
- Ski-rental Directional Only [D]: Using the ski-rental algorithm by only allowing directional plans.



(a) Non-Extension with different C_d. (b) Non-Extension with different C_o. (c) Extension with different C_d. (d) Extension with different C_o.

Figure 2: Competitive ratio verification.

Performance

The experimental results are shown in Fig. 1. Our proposed the OHR Algorithm is marked as [A] and the rest are benchmarks. Since the OHR Algorithm is a randomized algorithm, each data point is averaged over 10 rounds of simulation. Figs. 1 (a)–(d) include three sub-graphs from top to bottom. The top graph shows the total cost (with breakdowns) of different algorithms in the experiment. The middle graph shows the number of reserved plans for different algorithms. The bottom graph shows the traffic covered by PAYG. Moreover, each sub-graph has three groups of experiment outcomes with directional plan upfront fee CNY 9 (left), CNY 13.5 (middle), and CNY 18 (right). Accordingly, β is 0.3, 0.45, and 0.6 respectively.

In all results, OHR has the best performance. Figs. 1(a)–(b) show the results for the non-extension mode. In Fig. 1(a), OHR performs the best. Compared to [O], the number of reserved all-in-one plans is almost same, but OHR could strategically use directional plans to further save cost. Algorithms [K], [L], and [M] try to use historical experiences to make the current decision. However, as the users may use very different apps in different periods, once the algorithm reserves a directional plan, another app's traffic may become very large, causing delayed reservation. [D] performs worst as it reserves too many directional plans, which could be merged as an all-in-one plan.

Fig. 1(b) shows the simulation result when T is reduced to 0.5 week. Shorter T indicates inflation of plans. The results in Fig. 1(b) are roughly similar to those in 1(a), indicating that the advantages of the OHR still hold for different T values.

Figs. 1(c)–(d) show the simulation result of the extension mode. Compared to the non-extension mode, the overall costs are slightly smaller as each plan is valid for full T no matter when it is reserved. Still, the results show a similar trend, demonstrating that OHR also works well for the non-extension mode.

Competitive Ratio Verification

In this subsection, the competitive ratios are verified. We simulate the competitive ratio by running Algorithm 1 on randomly selected 30-week pieces in the trace. We have 10 pieces with 10 output data points and the outcome in each piece is averaged over 10 rounds of simulation. The optimal cost can be derived by exhausted search.

Figs. 2(a) and (b) show the results in the non-extension mode. Fig. 2(a) shows the theoretical and experimental values of competitive ratios with different C_d. It illustrates that the theoretical value decreases with the increase of C_d due to (9). All the experimental results are bounded by the theoretical ones. Fig. 2(b) shows results with different C_o. Still, all the experimental results are bounded by the theoretical ones, verifying the competitive ratio in Theorem 1.

Figs. 2(c) and (d) show the results in the extension mode. All the experimental results are bounded by the theoretical ones, verifying the competitive ration in Theorem 2. Note that the theoretical competitive ratio in the extension mode is twice as much as that in the non-extension mode. However, almost all the experimental results are less than half of the theoretical results, demonstrating that the real-world performance of the OHR Algorithm in the extension mode is almost as good as that in the non-extension mode.

Conclusion and Future Work

In this paper, we investigate the online plan reservation scheme where three payment methods are considered: (1) PAYG, (2) All-in-one plan, and (3) Directional plan. We proposed the OHR Algorithm to address the problem and evaluate the competitive ratios considering two modes: non-extension mode and extension mode. For the non-extension mode, we conclude that the competitive ratio is $\frac{e^\beta}{e^\beta-1}$. It is a neat extension to the classic ski-rental problem with competitive ratio $\frac{e}{e-1}$. For the extension mode, we prove that the competitive ratio is $\frac{2e^\beta}{e^\beta-1}$. Finally, the theoretical results are verified by trace-driven simulation. We compare the performance of the OHR Algorithm with benchmarks, showing that OHR performs the best.

In some real-world instances, a data cap may be employed to limit the overall data volume in a period of time. In this case, our model still works for a wide range of situations when the data cap is sufficiently large, and users do not exceed the cap. However, rigorous characterization of data cap substantially increases the difficulty as it leads to coupled decisions in the temporal dimension and volume dimension. How to quantify competitive ratios is thus drastically more challenging. Therefore, we leave it for future work.

References

Andrews, M.; Ozen, U.; Reiman, M. I.; and Wang, Q. 2014. Economic Models of Sponsored Content in Wireless Networks with Uncertain Demand. In *Smart Data Pricing*, 267–288. Hoboken, NJ, USA: John Wiley & Sons, Inc. ISBN 9781118611661.

Armstrong, M. 2006. Competition in Two-Sided Markets. *The RAND Journal of Economics* 37(3): 668–691. ISSN 07416261.

Bansal, N.; Buchbinder, N.; and Naor, J. S. 2012. Randomized Competitive Algorithms for Generalized Caching. *SIAM Journal on Computing* 41(2): 391–414. ISSN 0097-5397.

Dinh, T. Q.; Liang, B.; Quek, T. Q.; and Shin, H. 2020. Online Resource Procurement and Allocation in a Hybrid Edge-Cloud Computing System. *IEEE transactions on wireless communications* 19(3): 2137–2149. ISSN 1536-1276.

Feldkord, B.; Markarian, C.; and Der Heide, F. M. A. 2017. Price Fluctuation in Online Leasing 17–31.

Fleischer, R. 2001. On the Bahncard problem. *Theoretical Computer Science* 268(1): 161–174.

Hande, P.; Chiang, M.; Calderbank, R.; and Rangan, S. 2009. Network Pricing and Rate Allocation with Content Provider Participation. In *IEEE INFOCOM 2009*, 990–998. IEEE. ISBN 9781424435128. ISSN 0743166X.

Janevski, T. 2019. Network Neutrality. In *QoS for Fixed and Mobile Ultra-Broadband*, 261–282. Chichester, UK: John Wiley & Sons, Ltd. ISBN 9781119470502.

Karlin; Kenyon; and Randall. 2003. Dynamic TCP Acknowledgment and Other Stories about e /(e - 1). *Algorithmica* 36(3): 209–224. ISSN 0178-4617.

Khanafer, A.; Kodialam, M.; and Puttaswamy, K. P. N. 2013. The constrained Ski-Rental problem and its application to online cloud cost optimization. In *2013 Proceedings IEEE INFOCOM*, 1492–1500. IEEE. ISBN 9781467359443. ISSN 0743166X.

Komm, D. 2016. *An Introduction to Online Computation: Determinism, Randomization, Advice*. Texts in Theoretical Computer Science. An EATCS Series. Cham: Springer International Publishing. ISBN 9783319427478.

Ma, R. 2016. Subsidization Competition: Vitalizing the Neutral Internet. *IEEE/ACM Transactions on Networking (TON)* 24(4): 2563–2576. ISSN 1063-6692.

Meyerson, A. 2005. The parking permit problem. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, volume 2005, 274–282. IEEE. ISBN 0769524680. ISSN 02725428.

Njoroge, P.; Ozdaglar, A.; Stier-Moses, N. E.; and Weintraub, G. Y. 2014. Investment in Two-Sided Markets and the Net Neutrality Debate. *Review of Network Economics* 12(4): 355–402. ISSN 2194-5993.

Rojas, J. S. 2018. IP Network Traffic Flows Labeled with 75 Apps. Https://www.kaggle.com/jsrojas/ip-network-traffic-flows-labeled-with-87-apps.

Rosenfelder, S. 2018. New Data Shows Uninstalls Remain A Significant Pain for Apps. Technical report, AppsFlyer.

Rosenthal, S.; and Veloso, M. 2012. Mobile Robot Planning to Seek Help with Spatially-Situated Tasks. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, 2067–2073. AAAI Press.

Shi, M.; Lin, X.; Fahmy, S.; and Shin, D.-H. 2018. Competitive Online Convex Optimization with Switching Costs and Ramp Constraints. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, volume 2018-, 1835–1843. IEEE. ISBN 9781538641286. ISSN 0743166X.

Unterbrunner, P.; Giannikis, G.; Alonso, G.; Fauser, D.; and Kossmann, D. 2009. Predictable performance for unpredictable workloads. *Proceedings of the VLDB Endowment* 2(1): 706–717. ISSN 21508097.

Wang, S.; Li, J.; and Wang, S. 2020. Online Algorithms for Multi-shop Ski Rental with Machine Learned Predictions. *arXiv.org* ISSN 2331-8422.

Wang, W.; Liang, B.; and Li, B. 2015. Optimal Online Multi-Instance Acquisition in IaaS Clouds. *IEEE Transactions on Parallel and Distributed Systems* 26(12): 3407–3419. ISSN 1045-9219.

Yang, S.; Pan, L.; and Liu, S. 2019. An Online Algorithm for Selling Your Reserved IaaS Instances in Amazon EC2 Marketplace. In *2019 IEEE International Conference on Web Services (ICWS)*, 296–303. IEEE. ISBN 9781728127170.

Zang, S.; Bao, W.; Yeoh, P. L.; Vucetic, B.; and Li, Y. 2019. Filling Two Needs With One Deed: Combo Pricing Plans for Computing-Intensive Multimedia Applications. *IEEE Journal on Selected Areas in Communications* 37(7): 1518–1533. ISSN 0733-8716.

Zhang, H.; and Conitzer, V. 2020. Combinatorial Ski Rental and Online Bipartite Matching. In *Proceedings of the 21st ACM Conference on economics and computation*, EC '20, 879–910. ACM. ISBN 1450379753.

Zhang, L.; and Wang, D. 2014. Sponsoring content: Motivation and pitfalls for content service providers. In *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 577–582. IEEE. ISBN 9781479930883. ISSN 0743166X.

Zhang, L.; Wu, W.; and Wang, D. 2015. Sponsored Data Plan: A Two-Class Service Model in Wireless Data Networks. *ACM SIGMETRICS Performance Evaluation Review* 43(1): 85–96. ISSN 01635999.

Zhao, Y.; Wang, H.; Su, H.; Zhang, L.; Zhang, R.; Wang, D.; and Xu, K. 2020. Understand Love of Variety in Wireless Data Market Under Sponsored Data Plans. *IEEE journal on selected areas in communications* 38(4): 766–781. ISSN 0733-8716.