Learning-augmented Online Algorithm for Two-level Ski-rental Problem

Keyuan Zhang[1], Zhongdong Liu[1], Nakjung Choi[2], Bo Ji[1]

[1]Virginia Tech, Blacksburg, VA, USA
[2]Nokia Bell Labs
keyuanz@vt.edu, zhongdong@vt.edu, nakjung.choi@nokia-bell-labs.com, boji@vt.edu

Abstract

In this paper, we study the two-level ski-rental problem, where a user needs to fulfill a sequence of demands for multiple items by choosing one of the three payment options: paying for the on-demand usage (i.e., rent), buying individual items (i.e., single purchase), and buying all the items (i.e., combo purchase). Without knowing future demands, the user aims to minimize the total cost (i.e., the sum of the rental, single purchase, and combo purchase costs) by balancing the trade-off between the expensive upfront costs (for purchase) and the potential future expenses (for rent). We first design a robust online algorithm (RDTSR) that offers a worst-case performance guarantee. While online algorithms are robust against the worst-case scenarios, they are often overly cautious and thus suffer a poor average performance in typical scenarios. On the other hand, Machine Learning (ML) algorithms typically show promising average performance in various applications but lack worst-case performance guarantees. To harness the benefits of both methods, we develop a learning-augmented algorithm (LADTSR) by integrating ML predictions into the robust online algorithm, which outperforms the robust online algorithm under accurate predictions while ensuring worst-case performance guarantees even when predictions are inaccurate. Finally, we conduct numerical experiments on both synthetic and real-world trace data to corroborate the effectiveness of our approach.

1 Introduction

Decision-making under uncertainty is crucial in many real-world scenarios, such as buying or leasing a car, purchasing a daily or annual parking permit, and so on. Such problems are often modeled as online rent-or-buy problems. One classic example is the *ski-rental* problem, where a skier goes skiing for an unknown number of days and can choose to rent skis at a lower daily cost or buy them at a higher price to ski freely thereafter. The ski-rental problem and its variants have been extensively studied and found applications in cloud computing (Khanafer, Kodialam, and Puttaswamy 2013), power management (Antoniadis et al. 2021), serverless edge computing (Pan et al. 2022), etc.

However, the ski-rental problem only involves making a binary decision (i.e., rent or buy for one item only), which makes it inadequate to model more sophisticated scenarios where multiple levels of decisions are involved. Consider cloud computing as an example. An organization may request multiple types of server instances (e.g., HPC instances for complex system simulations, Memory instances for storing data, and GPU instances for AI tasks) from the cloud vendors such as Amazon EC2, Microsoft Azure, and Google Cloud (Miguel 2023). The cloud vendors usually offer multiple pricing options: i) pay-as-you-go: the cost only depends on the demand; ii) instance reservation: paying an upfront cost to cover the demand for that type of instance within a certain period; iii) combo offers: offering increased discounts for reserving multiple types of instances. Hence, making choices wisely among these plans is crucial to reducing expenses on cloud resources. Additionally, similar scenarios are common for individual users. For example, when subscribing to various digital services (e.g., Amazon offers video, music, and games), users often face choices among multiple plans: renting specific content, subscribing to one single service, or subscribing to a combo membership that grants full access to multiple services. Other applications, such as mobile data plans or Bahncard reservations, also offer similar payment options (Wu, Bao, and Yuan 2021).

To model multiple levels of decisions, we consider the two-level ski-rental problem introduced in Wu, Bao, and Yuan (2021). Specifically, it considers scenarios where a user needs to fulfill the demands for multiple items in a given but *unknown* time horizon. Upon the arrival of one demand for some item, the user can cover the demand by choosing one of the three payment options: i) rent, ii) single purchase, and iii) combo purchase. Rent is an on-demand payment with no upfront cost; a single purchase incurs an upfront cost (denoted by C_s) but covers the demands for that specific item since then and onward; a combo purchase has a higher upfront cost (denoted by $C_c \geq C_s$) while covering the demands for all the items till the end. While the ski-rental problem only considers rent and single purchases, a unique challenge in the two-level ski-rental problem lies in the intricate decision-making between single purchases and a combo purchase covering multiple items. To minimize the total cost (i.e., the sum of the rental, single purchase, and combo purchase costs), Wu, Bao, and Yuan (2021) propose online algorithms that make decisions without knowing any future information. *Competitive ratio (CR)* is employed to evaluate the algorithms, which is defined as the ratio between the cost

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

of the online algorithm and that of the optimal offline algorithm, over all the feasible inputs (see the formal definition of CR in Section 2). While they prove a CR for their deterministic algorithm, the CR can be unbounded when multiple units of demand arrive at the same time (see Remark 3.1). We propose our robust online algorithm to bridge this gap.

Although online algorithms are robust against the worst-case situation, they often exhibit excessive caution towards uncertainty, resulting in poor average performance in many real-world scenarios. In contrast, machine learning (ML) algorithms show promising average performance in many applications by using historical data to construct useful prediction models. However, they lack robustness guarantees as the predictions can be highly inaccurate due to distribution drift (Lu et al. 2018) or adversarially chosen test data (Szegedy et al. 2014).

In the second part of this work, we first show that a simple algorithm that blindly follows ML predictions cannot guarantee robustness in worst-case scenarios. Then, we design a novel learning-augmented online algorithm by integrating ML predictions into our online algorithm. Specifically, we show that our algorithm achieves two desired properties: i) with an accurate prediction, it offers a better CR than our robust online algorithm *(consistency)* and ii) even if the prediction is inaccurate, it still retains a worst-case guarantee *(robustness)*. Our learning-augmented online algorithm takes the predictions as the black-box oracles without prior knowledge of the prediction quality.

We summarize our main contributions in the following.

First, we propose an online algorithm for the two-level ski-rental problem. We prove that it achieves a CR of $3 - \frac{1}{C_s} - \frac{1}{C_c}(2 - \frac{1}{C_s})$, providing a stronger performance guarantee compared to the State-of-the-Art algorithm introduced in Wu, Bao, and Yuan (2021) (see Remark 3.1).

Second, we design a novel learning-augmented algorithm that integrates ML predictions into robust online decision-making. We prove that the proposed algorithm can guarantee both consistency and robustness. To the best of our knowledge, this is the first work that designs learning-augmented algorithms for the two-level ski-rental problem.

Finally, we conduct numerical experiments to evaluate the performance of our online algorithm and learning-augmented algorithm using both synthetic data and real-world trace data. The results verify our theoretical analyses and corroborate that our learning-augmented algorithm guarantees both consistency and robustness.

Related Work. The ski-rental problem is introduced in Karlin et al. (1988), which gives the best deterministic online algorithm with a CR of 2. Then, an optimal randomized online algorithm achieving $e/(e-1)$-competitiveness is proposed in Karlin et al. (1994) (e is the Euler's number). Several variants of the ski-rental problem are later studied. In the following, we briefly discuss the variants that are most relevant to ours. Fleischer (2001) studies the *Bahncard problem*, where buying a Bahncard provides discounts on subsequent railway tickets for a certain duration. Ai et al. (2014) study the *multi-shop ski-rental problem*, where a user needs to choose when and where to buy from multiple shops with different prices for renting and purchasing. Recently, Wu et al. (2022) study the *multi-commodity ski-rental problem*, where a user requires a bundle of commodities altogether and needs to choose payment options for each commodity.

On the other hand, researchers have designed learning-augmented algorithms for a variety of online optimization problems. In the seminal work (Lykouris and Vassilvtiskii 2018), they show that by incorporating ML predictions, the classic online Marker algorithm can achieve both consistency and robustness for the caching problem. Learning-augmented algorithm design is also studied for the ski-rental problem and its variants (Purohit, Svitkina, and Kumar 2018; Bamas, Maggiori, and Svensson 2020; Wang, Li, and Wang 2020). For example, both single and multiple ML predictions are incorporated into the online algorithm for the multi-shop ski-rental problem (Wang, Li, and Wang 2020). We refer interested readers to the surveys (Boyar et al. 2017; Mitzenmacher and Vassilvitskii 2022) for comprehensive coverage. In our work, however, the combo purchase can cover all the single purchases and thus result in coupled decisions. This "two-level" payment option is more practical in quite a few real-world applications but introduces unique challenges in the algorithm design.

2 System Model and Problem Formulation

System Model. We consider a time-slotted model with time index $t = 1, 2, \dots, T$, where T is the length of a finite time horizon *unknown* to the user (i.e., the decision maker). The user needs to fulfill demands for K items arriving over time. Upon the arrival of demands in each time-slot, the user can fulfill them by choosing from different options (rent or purchase). Without loss of generality, we assume that the demands arriving in the same time-slot are for one item only.

Let $d(t) := (i(t), a(t))$ denote the demand arriving in time-slot t, where $i(t) \in \{1, 2, \dots, K\}$ is the index of the item and $a(t)$ is a non-negative integer representing the amount of the demand. In particular, $d(t) = (0, 0)$ indicates no demand in time-slot t. Let $\mathbf{D} := \{d(t)\}_{t=1}^{T}$ denote the sequence of demands over the entire time horizon T.

To fulfill the demand in each time-slot, the user can choose one of the three payment options: i) rent, ii) single purchase, and iii) combo purchase.

i) **Rent**: The user pays a unit rental cost to cover one unit of demand in time-slot t.

ii) **Single purchase**: The user pays an upfront fee $C_s > 1$, which covers all the demands for item $i(t)$ from time t to T. For ease of analysis, we assume the same single purchase cost C_s for all the items.

iii) **Combo purchase**: The user pays a higher upfront fee $C_c \in (C_s, K \cdot C_s)$, which covers the demands for all the items from time t to T.

We assume that both C_s and C_c are integers and that once the payment is made in some time-slot t, the associated demands can be fulfilled immediately. While choosing to rent may offer a lower immediate cost, it may result in a higher future cost when confronted with increased future demands. In contrast, choosing to purchase (either single or combo) incurs a higher upfront cost, but it covers the potential future demands.

Problem Formulation. Let $r(t)$ be the binary rental decision in time-slot t, where $r(t)=1$ if the user decides to rent, otherwise $r(t)=0$. Let t_k denote the time when the user makes a single purchase for item k. We set $t_k=\infty$ when a single purchase is never made. Similarly, let t_c denote the time when the user makes a combo purchase, and we set $t_c=\infty$ when a combo purchase is never made. We consider *online* algorithm π that determines $\{r^\pi(t)$ for $t=1,2,\ldots,T,\{t_k^\pi\}_{k=1}^K,t_c^\pi\}$ only using information available in time-slot t: the rental decision history $\{r^\pi(\tau)\}_{\tau=1}^{t-1}$, single purchase history $\{t_k^\pi\}_{k=1}^K$, combo purchase history t_c^π, demand history $\{d(\tau)\}_{\tau=1}^t$, number of instances K, single purchase cost C_s, and combo purchase cost C_c; no future information is available to the algorithm. For notational simplicity, we drop the superscript π in the rest of the paper whenever the context is clear.

Let $\mathcal{C}(\mathbf{D},\pi)$ be the total cost under Algorithm π:

$$\mathcal{C}(\mathbf{D},\pi):=\sum_{t=1}^{T}a(t)r(t)+\sum_{k=1}^{K}C_s\mathbb{1}_{\{t_k\le T\}}+C_c\mathbb{1}_{\{t_c\le T\}},\tag{1}$$

where $\mathbb{1}_{\{\cdot\}}$ is the indicator function and the three terms correspond to the total rental cost, total single purchase cost, and combo purchase cost, respectively. Given a demand sequence $\mathbf{D}$, the objective is to minimize the total cost:

$$\min_{r(t),t_k,t_c} \quad \mathcal{C}(\mathbf{D},\pi) \tag{2a}$$

$$\text{subject to} \quad r(t)+\mathbb{1}_{\{t\ge t_{i(t)}\}}+\mathbb{1}_{\{t\ge t_c\}}\ge 1, \quad t\in\{1,2,\ldots,T\} \tag{2b}$$

$$r(t)\in\{0,1\},\ t\in\{1,2,\ldots,T\} \tag{2c}$$

$$t_k>0, \quad k\in\{1,2,\ldots,K\} \tag{2d}$$

$$t_c>0, \tag{2e}$$

where Contraint (2b) requires that each demand must be fulfilled via either renting or purchasing (single or combo).

Competitive Ratio. Without knowledge of the future demand, it is usually difficult for an online algorithm to attain the same minimum total cost achieved by an optimal offline algorithm, which requires knowledge of the entire demand sequence $\mathbf{D}$ beforehand. To measure the performance of an online algorithm, we consider a widely adopted metric called *competitive ratio* (CR), defined as the ratio between the cost of the online algorithm and that of the optimal offline algorithm over all possible inputs. Formally, an online algorithm π is called c-competitive if there exists a constant $c\ge 1$ such that for any demand sequence $\mathbf{D}$, there is

$$\mathcal{C}(\mathbf{D},\pi)\le c\cdot OPT(\mathbf{D}),\tag{3}$$

where $OPT(\mathbf{D})$ is the cost of the optimal offline algorithm under the demand sequence $\mathbf{D}$.

3 Robust Online Algorithm

In this section, we introduce our robust online algorithm and prove its competitive ratio. To begin with, we first present two important notions that will be used in our algorithm: indicative cost and purchase threshold.

Indicative Cost and Purchase Threshold. We define the indicative cost $\psi_k(t)$ for item k in time-slot t as the total demand for item k during the interval $[1,t]$ that is not covered by any purchase. Then, indicative cost $\psi_k(t)$ evolves as

$$\psi_k(t):=\begin{cases}\psi_k(t-1)+a(t), & i(t)=k \text{ and } \\ & t<\min\{t_k,t_c\},\\ \psi_k(t-1), & \text{otherwise,}\end{cases}\tag{4}$$

where $\psi_k(t)$ increases by $a(t)$ if the demand in time-slot t is for item k (i.e., $i(t)=k$) and is not covered by (either single or combo) purchases (i.e., $t<\min\{t_k,t_c\}$); otherwise, it remains unchanged. Note that $\psi_k(0)=0$ for all k. Let $\lambda_s\in(1,C_s]$ be the threshold for making single purchases. That is, if the single purchase indicator $\psi_k(t)$ reaches λ_s (i.e., $\psi_k(t)\ge\lambda_s$), we will make a single purchase for item k. We assume the same threshold λ_s for all items because they have the same single purchase price C_s. This is also an assumption made in Wu, Bao, and Yuan (2021). While our algorithm can potentially be adapted to address item-specific purchase costs and thresholds, a more sophisticated competitive analysis is expected.

Similarly, let $\psi_c(t)$ and $\lambda_c\in(1,C_c]$ be the overall indicative cost $\psi_c(t)$ and the combo purchase threshold, respectively. We define the overall indicative cost $\psi_c(t)$ as

$$\psi_c(t):=\sum_{k=1}^{K}\min\{\psi_k(t),\lambda_s\}.\tag{5}$$

When $\psi_c(t)$ reaches the combo purchase threshold λ_c (i.e., $\psi_c(t)\ge\lambda_c$), a combo purchase will be made. We assume that thresholds λ_s and λ_c are integers due to the integral demand we consider.

Intuitively, when indicative cost $\psi_k(t)$ reaches the single purchase threshold λ_s, it reflects a substantial demand for item k, leading to a single purchase of item k. Similarly, when the overall indicative cost $\psi_c(t)$ reaches λ_c, it indicates substantial demand for multiple items, leading to a combo purchase rather than single purchases or rent.

Algorithm Description. We now present our *Robust Deterministic Two-level Ski-rental (RDTSR)* Algorithm in Algorithm 1 and explain how it works in the following. When a new demand $d(t)$ arrives in time-slot t, it first updates the indicative costs $\psi_k(t)$ and $\psi_c(t)$ according to Eqs. (4) and (5), respectively (Lines 2-3). If the demand is already covered by a previous (single or combo) purchase (Line 4), it does nothing. Otherwise, if the indicative cost exceeds the threshold, it will purchase by prioritizing the combo purchase (Lines 7-10); if not, it will cover the demand via renting (Line 12).

Competitive Analysis. We derive the CR of RDTSR as a function of thresholds λ_s and λ_c. By choosing proper values of λ_s and λ_c, we show that the CR is upper bounded by 3.

Theorem 3.1. *The CR of RDTSR is upper bounded by*

$$3-\frac{1}{C_s}-\frac{1}{C_c}(2-\frac{1}{C_s}).\tag{6}$$

Algorithm 1: Robust Deterministic Two-level Ski-rental (RDTSR) Algorithm

```
Input  : $K$, $C_s$, $C_c$, $\lambda_s$, $\lambda_c$, $\mathbf{D}$ (revealed in an online
         manner)
Output: $r(t), t_k, t_c$
Init.  : $\psi_c(t), \psi_k(t), r(t) \leftarrow 0, t_k, t_c \leftarrow \infty$
while new demand $d(t)$ arrives do
    Update $\psi_k(t)$ according to Eq. (4);
    Update $\psi_c(t)$ according to Eq. (5);
    if $d(t)$ is covered by a previous purchase then
        Do nothing and wait till next time-slot;
    else
        if $\psi_c(t) \geq \lambda_c$ then
            $t_c \leftarrow t$; // combo purchase
        else if $\psi_{i(t)}(t) \geq \lambda_s$ then
            $t_{i(t)} \leftarrow t$; // single purchase
        else
            $r(t) \leftarrow 1$; // rent
        end
    end
end
```

In particular, this upper bound is achieved by choosing thresholds $\lambda_s = C_s$ and $\lambda_c = C_c$.

Proof sketch. The detailed proof is provided in Appendix B and we give a proof sketch in the following. The proof has two key steps: 1) given any λ_s and λ_c, we derive an upper bound of the CR, which is a function of λ_s and λ_c; 2) we show that the CR is at most $3 - \frac{1}{C_s} - \frac{1}{C_c}(2 - \frac{1}{C_s})$ when $\lambda_s = C_s$ and $\lambda_c = C_c$.

Note that it is typically challenging to directly analyze the performance for an arbitrary demand sequence. Therefore, we break Step 1) into three substeps: 1a) We derive an upper bound of CR over all the demand sequences with the same total demand. The total demand is a vector consisting of K elements where each element represents the total demand for the associated item. 1b) Following a similar line of analysis in Wu, Bao, and Yuan (2021), we define a standard total demand to simplify the upper bound we obtained in Step 1a). 1c) We obtain an upper bound of CR over all the standard total demand, which is $\max\{\frac{\lambda_s - 1 + C_s}{\lambda_s}, \frac{\lambda_c - 1 + C_c + (\lambda_c - 1)\lambda_s^{-1}(C_s - 1)}{\min\{C_c, \lambda_c\}}\}$. In Step 2), we minimize the function of λ_s and λ_c derived in Step 1c). □

Remark 3.1. *Wu, Bao, and Yuan (2021) proposed a similar deterministic online algorithm and showed that their algorithm is $(3 - 1/C_s)$-competitive. However, we construct a simple example to show that the CR of their algorithm can be unbounded when allowing multiple units of demand to arrive in a time-slot (i.e., $a(t) > 1$). Specifically, consider a demand sequence $\tilde{\mathbf{D}} := \{(1, C_s), (2, C_s), \dots, (K, C_s)\}$, i.e., C_s units of demand for item k arrive in time-slot k for all k. In their algorithm, single purchases will be made for all the items, while the combo purchase will never be made as their algorithm skips the updates of the overall indicative cost $\psi_c(t)$ after a single purchase. With the fact that the cost of the optimal offline algorithm is at most C_c, their algorithm's competitive ratio is at least KC_s/C_c, which can be arbitrarily large when $KC_s \gg C_c$. We address this issue by redefining the indicative cost and assigning the highest priority to the combo purchase. It is also noteworthy that even for the setting with unit demand arrival (i.e., $a(t) = 1$), our algorithm still has a slightly improved upper bound ($3 - \frac{1}{C_s} - \frac{1}{C_c}(2 - \frac{1}{C_s})$ vs. $(3 - \frac{1}{C_s})$).*

4 Learning-augmented Online Algorithm

In the previous section, we introduce a robust online algorithm, RDTSR, with a worst-case guarantee. However, online algorithms are often overly conservative and may have a poor average performance in real-world scenarios. In contrast, ML algorithms have the advantage of utilizing extensive historical data and building well-trained models, which enables them to achieve a promising average performance. Nonetheless, ML algorithms lack a worst-case guarantee when facing adversaries, outliers, distribution shifts, etc. To harness the benefits of both methods, we design a learning-augmented online algorithm that achieves both consistency and robustness, two commonly used notions in the literature.

4.1 Machine Learning Predictions

We consider the case where an ML algorithm provides a prediction $y := (y_1, \dots, y_K)$ that represents the predicted total demand for each item. Predictions of the total demand are commonly used in real-world applications. For example, in cloud computing, various resources such as CPU, memory, and network bandwidth can be fed into a predictor to forecast future workload (Masdari and Khoshnevis 2020).

We assume that the ML prediction y is available in the very beginning (i.e., $t = 0$) and is provided in full (i.e., all y_k's are available). The actual total demand can be represented by a vector $z := (z_1, \dots, z_K)$, where z_k denotes the total demand for item k, i.e., $z_k = \sum_{t=1}^{T} \mathbb{1}_{\{i(t)=k\}} a(t)$. The prediction error is defined as the $\ell 1$-norm distance from the actual total demand to the predicted ones, and we use $\eta_k := |y_k - z_k|$ and $\eta := \sum_{k=1}^{K} \eta_k$ to denote the prediction error for item k and the total prediction error, respectively.

Consider a learning-augmented online algorithm $\pi^\dagger$ that generates a solution with cost $\mathcal{C}(\mathbf{D}, \pi^\dagger)$ using information available in time-slot t and a prediction y. Algorithm $\pi^\dagger$ is called *α-consistent* if $\mathcal{C}(\mathbf{D}, \pi^\dagger) \leq \alpha \cdot OPT(\mathbf{D})$ for any sequence $\mathbf{D}$ when the prediction is perfect (i.e., $\eta = 0$), and it is called *β-robust* if $\mathcal{C}(\mathbf{D}, \pi^\dagger) \leq \beta \cdot OPT(\mathbf{D})$ for any sequence $\mathbf{D}$, regardless of the prediction error η. Here, consistency measures the performance in the best-case scenario with perfect predictions, while robustness measures the performance of a learning-augmented algorithm in the worst-case scenario regardless of the prediction quality.

4.2 A Simple Algorithm: Follow the Prediction

First, we show that a simple algorithm called *Follow the Prediction (FTP)*, cannot guarantee robustness. FTP operates in the following way: it compares the combo purchase cost C_c with the total cost without the combo purchase option (i.e.,

$\sum_{k=1}^{K}\min\{C_s, y_k\}$), and makes a combo purchase if it is lower (i.e., $\sum_{k=1}^{K}\min\{C_s, y_k\} \geq C_c$); otherwise, it makes a single purchase for item k if $y_k \geq C_s$ or rents for item k when needed. Lemma 4.1 gives an upper bound on the total cost under FTP.

Lemma 4.1. *FTP satisfies $\mathcal{C}(\mathbf{D}, \mathrm{FTP}) \leq OPT(\mathbf{D}) + \eta$.*

Proof sketch. The detailed proof is provided in Appendix E and we give a proof sketch in the following. We first show that the cost of both FTP and the optimal offline algorithm only depends on total demand. Then, we only need to analyze the CR based on the total demand. We use $ALG(z)$ and $OPT(z)$ to denote the cost of FTP and the optimal offline algorithm under a total demand z, respectively. Depending on whether FTP and the optimal offline algorithm each make a combo purchase or not, we consider four cases. Finally, we show $ALG(z) \leq OPT(z) + \eta$ for each case. □

Remark 4.1. *When the prediction is perfect (i.e., $\eta = 0$), FTP achieves the optimal cost. However, the performance of FTP can be very poor when the prediction error η is large. For example, when $\sum_{k=1}^{K}\min\{C_s, y_k\} < C_c$ and $y_k < C_s$, but there is a very large total demand for item k, FTP would choose to rent for item k at all times, resulting in an unbounded cost and thus an unbounded CR as the optimal cost is bounded by C_c. Therefore, although FTP can achieve consistency, it lacks a robustness guarantee.*

4.3 Design of Learning-augmented Algorithm

As following the prediction in a straightforward way (e.g., FTP) cannot guarantee the worst-case performance, we aim to wisely integrate the ML predictions with RDTSR and design a learning-augmented online algorithm that can achieve both consistency and robustness.

The main idea of the design is to adjust the threshold of RDTSR based on the prediction: if a prediction suggests a (single or combo) purchase, then we should decrease the associated threshold to encourage the purchase (recall that the threshold determines when to make a purchase); if a prediction suggests not making a purchase, we should increase the associated threshold. The extent to which we adjust the threshold relies on our confidence in the prediction.

Based on the above intuition, we design the *Learning-augmented Deterministic Two-level Ski-rental (LADTSR)* Algorithm, presented in Algorithm 2. It is almost the same as RDTSR, except for the choice of thresholds. First, we specify a trust parameter $\theta \in [0, 1]$, reflecting the confidence in the prediction: a smaller θ indicates higher confidence. Then, we adjust the thresholds as follows: for each single purchase threshold $\lambda_{s,k}$ for item k, we reduce it to θC_s if the prediction suggests a single purchase for this item (i.e., $y_k \geq C_s$); otherwise, we increase $\lambda_{s,k}$ to C_s/θ. For the combo purchase threshold λ_c, we reduce it to $\theta^2 C_c$ (see Remark 4.4 for an explanation of this choice instead of θC_c) if the prediction suggests a combo purchase (i.e., $\sum_{k=1}^{K}\min\{C_s, y_k\} \geq C_c$); otherwise, we increase λ_c to C_c/θ. We set the threshold to infinity in case of division by 0. Note that when running RDTSR (Line 13), the definition of $\psi_c(t)$ (i.e., Eq. (5)) needs to be modified slightly by replacing λ_s with $\lambda_{s,k}$.

Algorithm 2: Learning-augmented Deterministic Two-level Ski-rental (LADTSR) Algorithm

```
   Input : $y, \theta, K, C_s, C_c$, $\mathbf{D}$ (revealed in an online
           manner)
   Output: $r(t), t_k, t_c$
   Init.  : $\psi_c(t), \psi_k(t), r(t) \leftarrow 0, t_k, t_c \leftarrow \infty$
1  for item $k = \{1, 2, \ldots, K\}$ do
2      if $y_k \geq C_s$ then
3          $\lambda_{s,k} \leftarrow \theta C_s$;
4      else
5          $\lambda_{s,k} \leftarrow C_s/\theta$;
6      end
7  end
8  if $\sum_{k=1}^{K}\min\{C_s, y_k\} \geq C_c$ then
9      $\lambda_c \leftarrow \theta^2 C_c$;
10 else
11     $\lambda_c \leftarrow C_c/\theta$;
12 end
13 Run RDTSR with thresholds $\lambda_{s,k}$ and $\lambda_c$;
```

4.4 Analysis of Learning-augmented Algorithm

In this subsection, we focus on the consistency and robustness analysis of LADTSR with $\theta \in (0, 1)$. The special cases of LADTSR with $\theta = 0$ and $\theta = 1$ correspond to FTP and RDTSR, respectively. It is noteworthy that by choosing different values of θ, LADTSR exhibits a crucial trade-off between consistency and robustness.

Theorem 4.1. *The CR of LADTSR is upper bounded by*

$$\min\left\{1 + \theta + \theta^2 + \frac{1 + 2\theta}{1 - \theta} \cdot \frac{\eta}{OPT(\mathbf{D})}, 1 + \theta^{-1} + \theta^{-3}\right\}, \tag{7}$$

for $\theta \in (0, 1)$. Futhermore, LADTSR is $(1 + \theta + \theta^2)$-consistent and $(1 + \theta^{-1} + \theta^{-3})$-robust.

Proof sketch. The detailed proof is provided in Appendices F and H, and we give a proof sketch in the following.

We first prove the first bound $1 + \theta + \theta^2 + \frac{1+2\theta}{1-\theta} \cdot \frac{\eta}{OPT(\mathbf{D})}$. Our goal is to show that this upper bound holds for any demand sequence and any prediction. First, we consider two cases for the choice of λ_c (i.e., $\lambda_c = \theta^2 C_c$ and $\lambda_c = C_c/\theta$). In each case, we consider four subcases, depending on whether LADTSR and the optimal offline algorithm each make a combo purchase or not. In each subcase, we derive an upper bound of CR, which is a function of η and $OPT(\mathbf{D})$. Combining all eight subcases yields the first bound.

Next, we prove the second bound $1+\theta^{-1}+\theta^{-3}$. The analysis is similar to Step 1) in the proof of Theorem 3.1 and also has three steps: 1a) Given any prediction, we derive an upper bound of CR over all the demand sequences with the same total demand. 1b) We simplify the upper bound obtained in Step 1a) through standard total demand. 1c) We show that the upper bound derived in 1b) is at most $1 + \theta^{-1} + \theta^{-3}$. Note that here we need a generalized standard total demand compared to the proof of Theorem 3.1 because LADTSR has two different values of single purchase threshold (i.e., θC_s and C_s/θ), while RDTSR only has one threshold value (i.e., λ_s). □

Remark 4.2. *With any $\theta \in (0, 1)$, the CR of LADTSR is at most $1+\theta^{-1}+\theta^{-3}$, regardless of the prediction error η, implying the worst-case performance guarantees (i.e., robustness). In particular, if LADTSR does not trust the prediction at all (i.e., $\theta \to 1$), then we have $1+\theta^{-1}+\theta^{-3} \to 3$, achieving a similar performance guarantee to that of RDTSR.*

Remark 4.3. *When the prediction is perfect (i.e., $\eta = 0$), the CR of LADTSR is at most $1 + \theta + \theta^2$ as $1 + \theta + \theta^2 < 1 + \theta^{-1} + \theta^{-3}$ for $\theta \in (0, 1)$, implying $(1 + \theta + \theta^2)$-consistency. In particular, if LADTSR fully trusts the prediction (i.e., $\theta \to 0$), then we have $1 + \theta + \theta^2 \to 1$, achieving a similar performance to the optimal offline algorithm.*

Remark 4.4. *We now explain the reason why we set the combo purchase threshold λ_c as $\theta^2 C_c$ instead of θC_c. Specifically, we present an example to show that LADTSR fails to achieve 1-consistency when $\theta \to 0$ if we choose $\lambda_c = \theta C_c$. Consider $C_c = (K - 1)C_s + 1$ and a demand sequence $\tilde{\mathbf{D}} := \{(1, C_s), (2, C_s), \ldots, (K, C_s)\}$. If the prediction is perfect (i.e., $y = (C_s, \ldots, C_s)$), then LADTSR sets $\lambda_{s,k} = \theta C_s$ for all the items since $y_k \geq C_s$ (in fact, $y_k = C_s$) and $\lambda_c = \theta C_c$ since $\sum_{k=1}^{K} \min\{C_s, y_k\} = KC_s > C_c$. When the demand (k, C_s) arrives, according to Eqs.* (4) *and* (5)*, $\psi_k(k)$ increases by C_s, and $\psi_c(k)$ increases by $\min\{\psi_k(k), \lambda_{s,k}\} = \theta C_s$ (recall that we modify Eq.* (5) *slightly for LADTSR). So in each time-slot t, the overall indicative cost is $\psi_c(t) = \sum_{k=1}^{K} \min\{\psi_k(t), \lambda_{s,k}\} = t\theta C_s$, and it reaches λ_c only when $t = K$ because $\psi_c(K - 1) = (K - 1)\theta C_s < \theta(K - 1)C_s + \theta = \theta C_c = \lambda_c < K\theta C_s = \psi_c(K)$. That is, LADTSR will make a combo purchase when the last demand arrives; for the previous $K - 1$ demands, LADTSR will make single purchases for these items as their demands exceed the single purchase threshold. As a result, the total cost is $(K - 1)C_s + C_c$. Since the optimal offline cost is $C_c = (K - 1)C_s + 1$ (by assumption), the CR of LADTSR is at least $\frac{(K-1)C_s+C_c}{C_c} = \frac{C_c-1+C_c}{C_c} = 2 - \frac{1}{C_c}$, which is larger than 1 for any θ.*

5 Experimental Results

In this section, we conduct numerical experiments using both synthetic and real-world trace data to evaluate the performance of our proposed algorithms (RDTSR and LADTSR).[1] The results demonstrate the robustness of RDTSR and corroborate that LADTSR offers the desired trade-off between consistency and robustness.

5.1 Datasets

Synthetic Dataset. We consider the same settings adopted in Wu, Bao, and Yuan (2021). Specifically, the demand sequence $\mathbf{D}$ is generated in the following way: First, the time horizon T follows a uniform distribution ranging from 1 to 60. The demand in each time-slot is randomly assigned to one item, following two standard distributions: i) **Uniform:** the demand is uniformly distributed among all the items; ii) **Long-tailed:** the demand distribution follows the Pareto law, i.e., 80% of demands are assigned to 20% of items. The theoretical results in Wu, Bao, and Yuan (2021) hold only for unit-demands (i.e., $a(t) = 1$ in all time-slots). In our experiments, we also consider the case where multiple units of demand may arrive in each time-slot (i.e., $a(t) \geq 1$). We set the number of items $K = 6$.

Cloud Cost Management Dataset. We consider the application of cloud service cost management, where a user needs to fulfill the demands from multiple servers via either rental or purchase (single or combo) options. To simulate the demands from multiple servers, we use the traces of Virtual Machine (VM) workloads from Microsoft Azure (Cortez et al. 2017). These traces include the average CPU workload every 5 minutes for 30 consecutive days. We randomly shuffle the workloads from multiple servers to generate the demand sequence $\mathbf{D}$. The number of VMs is 9 (i.e., $K = 9$).

Mobile App Usage Dataset. We consider the application where a user needs to pay for the data usage from multiple mobile Apps. The data usage traces are provided by Yu et al. (2018), which records the user ID, App ID, timestamp, and data traffic. We use the trace for multiple Apps from one user in one day as the demand sequence for one problem instance. The number of Apps is 4 (i.e., $K = 4$).

Unless otherwise specified, we set the single purchase price $C_s = 9$ and the combo purchase price $C_c = 30$ for all the three datasets we consider.

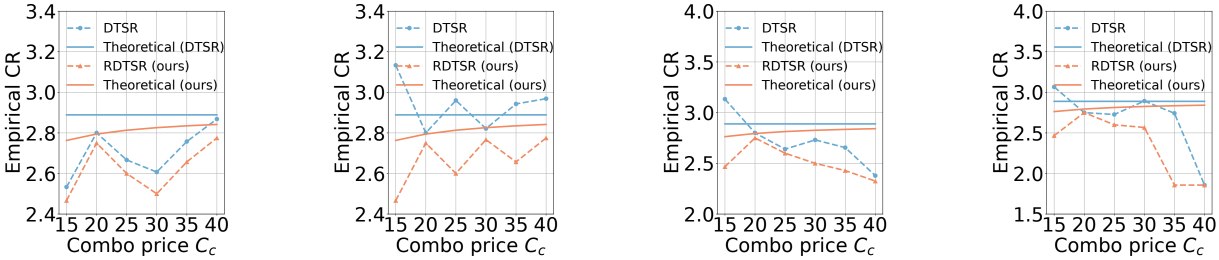
5.2 Robust Online Algorithms

In this subsection, we compare RDTSR with the *Deterministic Two-level Ski-Rental (DTSR)* algorithm proposed in Wu, Bao, and Yuan (2021).

Experimental Setup. We vary the value of the combo purchase price C_c from 15 to 40. For each value of C_c, we run our experiments for 10^4 independent demand sequences, with 40% being the uniform sequences and 60% being the long-tailed sequences. All the experiments are implemented in Python and are conducted on a laptop with a 12th Gen Intel(R) i5-12500H processor and 16GB memory.

Results. Fig. 1 shows the empirical and theoretical CR of RDTSR and DTSR. Here, the empirical CR represents the worst cost ratio (under the online algorithm and the optimal offline algorithm) over 10^4 independent demand sequences. We can observe that our algorithm (RDTSR) exhibits a lower empirical CR compared to the existing algorithm (DTSR). Furthermore, the empirical CR of RDTSR consistently remains below its theoretical bound, verifying our theoretical results. In the setting of multi-unit demands, the empirical CR of DTSR exceeds the theoretical bound (e.g., when $C_c = 15, 25, 35, 40$ in Fig. 1b). This suggests that the CR of DTSR obtained in Wu, Bao, and Yuan (2021) does not hold when allowing multiple units of demand to

[1] Source code: https://github.com/nauyek/LADTSR.



(a) Unit demands (b) Multi-unit demands (c) Azure dataset (d) AppUsage dataset

Figure 1: Empirical CR on the synthetic dataset.

arrive in a time-slot (see Remark 3.1 for a detailed discussion). Similar results can be observed from the two real-world datasets (see Fig. 1c and Fig. 1d). Note that for the AppUsage dataset, when $C_c = 40$, the combo purchase option is redundant and the problem boils down to the ski-rental problem (because $K \cdot C_s = 36 < 40$).

Furthermore, the simulation results also show that RDTSR and DTSR exhibit similar average performance, which is provided in Appendix J.

5.3 Learning-augmented Online Algorithms

In this subsection, we study the performance of LADTSR under varying prediction errors. We first explain how to generate ML predictions in our experiments.

ML Prediction Generation. Recall that LADTSR receives the total demand for each item as a prediction. We construct an ML task that uses historical total demand information to predict the next total demand for one item. Suppose that there are N rounds of two-level ski-rental problem instances, and for each round n, we have a total demand z_k^n for item k. Then, we can use a sequence of historical total demand with length $l-1$ (i.e., $z_k^{n-1}, z_k^{n-2}, \ldots, z_k^{n-l+1}$) to predict z_k^n. To generate a sequence of total demand for each item, for the Synthetic dataset, we generate multiple problem instances (40% uniform sequences followed by 60% long-tailed sequences) and then concatenate them together; such a sequence of total demands for each item can be directly obtained in the two real-world datasets. Then, we use a sliding window to generate multiple sequences of total demand for training and testing. The window size l is 10, 513, and 7 for the Synthetic, Azure, and AppUsage datasets, respectively. The last total demand is the ground truth while the previous sequence of total demand is the historical data. The window size is specifically chosen for each dataset due to the performance of the models. Finally, we use 80% of the sequences for training. The Mean Average Percentage Error (MAPE) of our ML models during the test is at most 5%.

To generate predictions with varying qualities, we first train the ML models without any perturbation. Then during the test, we perturb the input of the network by adding a bias μ that is uniform across the whole sequence of input. This represents the scenario where the prediction is biased due to the distribution shift between training and testing data.

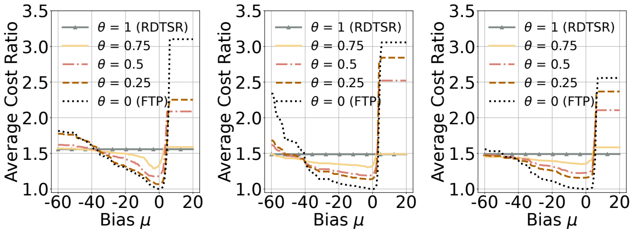
Experimental Setup. We train a *Long Short-term Memory (LSTM)* network to predict the total demand for each item. The network has two LSTM layers followed by one fully connected layer. The hidden states of the two LSTM layers are both 10, 256, and 10 for the Synthetic, Azure, and AppUsage datasets, respectively. We use the mean absolute error as the loss function and employ the Adam optimizer to train the weights. The model is implemented in TensorFlow. For each model, the training process takes about 2 minutes on an NVIDIA GeForce RTX 3060 Laptop GPU.

To study the impact of trust parameter θ, we evaluate the performance of LADTSR with different values of $\theta \in \{0, 0.25, 0.5, 0.75, 1\}$. Recall that LADTSR with $\theta = 0$ and $\theta = 1$ corresponds to FTP and RDTSR, respectively.

Results. Fig. 2 shows the average cost ratio of LADTSR with different values of θ under varying prediction errors (corresponding to varying values of bias μ). Here the average cost ratio represents the average ratio of cost (under the online algorithm and the optimal offline algorithm) over all the problem instances. Fig. 2a shows that while FTP (i.e., LADTSR with $\theta = 0$) performs quite well when the prediction is accurate (i.e., $\mu = 0$), its performance degrades significantly when the bias increases. For LADTSR with $\theta \in \{0.25, 0.5, 0.75\}$, when the predictions are accurate, they all outperform RDTSR (i.e., LADTSR with $\theta = 1$), achieving better average performance compared to the robust online algorithm without using any predictions. When the prediction is perturbed by a large bias μ ($\mu \leq -40$ or $\mu \geq 10$), the performance degrades, but not significantly compared to FTP. Furthermore, with different values of θ, LADTSR provides different trade-off curves for consistency and robustness. It is noteworthy that the performance of our learning-augmented algorithms degrades abruptly when the actual demand is overestimated ($\mu > 0$), while at a slower rate when it is underestimated. This happens because overestimation leads to unnecessary (single or combo) purchases, resulting in an immediate rise in the cost ratio. On the other hand, underestimation incurs additional rental costs before the algorithm makes a purchase, rising gradually as the bias level $|\mu|$ increases.

6 Conclusion

Motivated by real-world applications, we investigated the two-level ski-rental problem with multiple payment options



(a) Synthetic dataset (b) Azure dataset (c) AppUsage dataset

Figure 2: Average cost ratio under varying prediction errors.

and designed a 3-competitive deterministic online algorithm called RDTSR. By integrating useful ML predictions with RDTSR, we proposed a learning-augmented online algorithm called LADTSR, which achieves both consistency and robustness. As for future work, one important direction is to obtain a non-trivial competitive ratio lower bound for the studied problem as well as the optimal trade-off for the learning-augmented algorithm design; another interesting direction is to design learning-augmented algorithms that can achieve better performance by adaptively choosing the trust parameter θ; it would also be nice to design algorithms that account for the asymmetric impact of overestimation and underestimation in predictions.

Acknowledgments

This work is supported in part by the Commonwealth Cyber Initiative (CCI).

References

Ai, L.; Wu, X.; Huang, L.; Huang, L.; Tang, P.; and Li, J. 2014. The multi-shop ski rental problem. In *The 2014 ACM international conference on Measurement and modeling of computer systems*, 463–475. Association for Computing Machinery.

Antoniadis, A.; Coester, C.; Eliás, M.; Polak, A.; and Simon, B. 2021. Learning-augmented dynamic power management with multiple states via new ski rental bounds. *Advances in Neural Information Processing Systems*, 34: 16714–16726.

Bamas, É.; Maggiori, A.; and Svensson, O. 2020. The primal-dual method for learning augmented algorithms. *Advances in Neural Information Processing Systems*, 33: 20083–20094.

Boyar, J.; Favrholdt, L. M.; Kudahl, C.; Larsen, K. S.; and Mikkelsen, J. W. 2017. Online algorithms with advice: A survey. *ACM Computing Surveys (CSUR)*, 50(2): 1–34.

Cortez, E.; Bonde, A.; Muzio, A.; Russinovich, M.; Fontoura, M.; and Bianchini, R. 2017. Resource central: Understanding and predicting workloads for improved resource management in large cloud platforms. In *Proceedings of the 26th Symposium on Operating Systems Principles*, 153–167.

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

Karlin, A. R.; Manasse, M. S.; McGeoch, L. A.; and Owicki, S. 1994. Competitive randomized algorithms for nonuniform problems. *Algorithmica*, 11(6): 542–571.

Karlin, A. R.; Manasse, M. S.; Rudolph, L.; and Sleator, D. D. 1988. Competitive snoopy caching. *Algorithmica*, 3: 79–119.

Khanafer, A.; Kodialam, M.; and Puttaswamy, K. P. 2013. The constrained ski-rental problem and its application to online cloud cost optimization. In *2013 Proceedings IEEE INFOCOM*, 1492–1500. IEEE.

Lu, J.; Liu, A.; Dong, F.; Gu, F.; Gama, J.; and Zhang, G. 2018. Learning under concept drift: A review. *IEEE transactions on knowledge and data engineering*, 31(12): 2346–2363.

Lykouris, T.; and Vassilvtiskii, S. 2018. Competitive Caching with Machine Learned Advice. In *Proceedings of the 35th International Conference on Machine Learning*, 3296–3305. PMLR.

Masdari, M.; and Khoshnevis, A. 2020. A survey and classification of the workload forecasting methods in cloud computing. *Cluster Computing*, 23(4): 2399–2424.

Miguel, P. G. 2023. Guide to the 23 Best Cloud Service Providers in 2023. https://thectoclub.com/tools/best-cloud-service-providers. Accessed: 2023-12-19.

Mitzenmacher, M.; and Vassilvitskii, S. 2022. Algorithms with predictions. *Communications of the ACM*, 65(7): 33–35.

Pan, L.; Wang, L.; Chen, S.; and Liu, F. 2022. Retention-aware container caching for serverless edge computing. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*, 1069–1078. IEEE.

Purohit, M.; Svitkina, Z.; and Kumar, R. 2018. Improving online algorithms via ML predictions. *Advances in Neural Information Processing Systems*, 31.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I. J.; and Fergus, R. 2014. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR*.

Wang, S.; Li, J.; and Wang, S. 2020. Online algorithms for multi-shop ski rental with machine learned advice. *Advances in Neural Information Processing Systems*, 33: 8150–8160.

Wu, B.; Bao, W.; and Yuan, D. 2021. Competitive Analysis for Two-Level Ski-Rental Problem. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13): 12034–12041.

Wu, B.; Bao, W.; Yuan, D.; and Zhou, B. 2022. Competitive Analysis for Multi-Commodity Ski-Rental Problem. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, 4672–4678.

Yu, D.; Li, Y.; Xu, F.; Zhang, P.; and Kostakos, V. 2018. Smartphone app usage prediction using points of interest. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(4): 174.

A Preliminary Results

Before we delve into the proof of our main results, we begin by presenting several useful lemmas. These lemmas will be crucial in our analysis of the competitive ratio (CR) of our robust online algorithm (RDTSR) and learning-augmented online algorithm (LADTSR).

First, recall that in Eq. (4), the indicative cost $\psi_k(t)$ is defined as

$$\psi_k(t) = \begin{cases} \psi_k(t-1) + a(t), & i(t) = k \text{ and } \\ & t < \min\{t_k, t_c\}, \\ \psi_k(t-1), & \text{otherwise}, \end{cases} \tag{8}$$

where $\psi_k(t)$ increases by $a(t)$ if the demand in time-slot t is for item k (i.e., $i(t) = k$) and is not covered by (either single or combo) purchases (i.e., $t < \min\{t_k, t_c\}$); otherwise, it remains unchanged. We initialize $\psi_k(t)$ as 0 at the beginning, i.e., $\psi_k(0) = 0$. Instead of using recursion, Lemma A.1 shows that the indicative cost $\psi_k(t)$ can also be defined iteratively: intuitively, it is the sum of demands for item k during the interval $[1, t]$ when neither a single purchase for item k nor a combo purchase is made.

Lemma A.1. *The indicative cost $\psi_k(t)$ in Eq. (8) can be equivalently written as*

$$\psi_k(t) := \sum_{\tau=1}^{t} \mathbb{1}_{\{i(\tau)=k\}} \cdot \mathbb{1}_{\{\tau<t_k\}} \cdot \mathbb{1}_{(\tau<t_c)} \cdot a(\tau). \tag{9}$$

Proof. The proof is straightforward. Let $\psi'_k(t)$ evolves as Eq. (9), we have

$$\psi'_k(t) - \psi'_k(t-1) = \mathbb{1}_{\{i(t)=k\}} \cdot \mathbb{1}_{\{t<t_k\}} \cdot \mathbb{1}_{\{t<t_c\}} \cdot a(t). \tag{10}$$

That is, we have $\psi'_k(t) - \psi'_k(t-1) = a(t)$ if three conditions are met: i) $i(t) = k$, ii) $t < t_k$, iii) $t < t_c$; otherwise $\psi'_k(t) - \psi'_k(t-1) = 0$. Clearly, this is the same as Eq. (8). □

The following two lemmas state that the combo purchase decision made by our algorithms only depends on the total demand, rather than the specific order of demand arrival. Recall that the total demand of each demand sequence is denoted by $z = \{z_1, \dots, z_K\}$, where z_k denotes the total demand of item k, i.e.,

$$z_k = \sum_{t=1}^{T} \mathbb{1}_{\{i(t)=k\}} \cdot a(t). \tag{11}$$

Lemma A.2. *Given any total demand z, LADTSR will make the combo purchase iff $\sum_{k=1}^{K} \min\{z_k, \lambda_{s,k}\} \geq \lambda_c$.*

Proof. First, we prove the sufficient condition "⇒": it turns out that

$$\sum_{k=1}^{K} \min\{z_k, \lambda_{s,k}\} \overset{(a)}{\geq} \psi_c(t_c) \overset{(b)}{\geq} \lambda_c. \tag{12}$$

To see this, for inequality (a), we have

$$\begin{aligned} z_k &\overset{(c)}{=} \sum_{t=1}^{T} \mathbb{1}_{\{i(t)=k\}} \cdot a(t) \\ &\geq \sum_{t=1}^{t_c} \mathbb{1}_{\{i(t)=k\}} \cdot a(t) \\ &\geq \sum_{t=1}^{t_c} \mathbb{1}_{\{i(t)=k\}} \cdot \mathbb{1}_{\{t<t_k\}} \cdot \mathbb{1}_{\{t<t_c\}} \cdot a(t) \\ &\overset{(d)}{=} \psi_k(t_c), \end{aligned} \tag{13}$$

where (c) is due to the definition of z_k in Eq. (11) and (d) is due to the equivalent definition of $\psi_k(t)$ in Eq. (9). Furthermore, recall that the definition of the overall indicative cost $\psi_c(t)$ in time-slot t for LADTSR is

$$\psi_c(t) = \sum_{k=1}^{K} \min\{\psi_k(t), \lambda_{s,k}\}. \tag{14}$$

Therefore, we have

$$\sum_{k=1}^{K} \min\{z_k, \lambda_{s,k}\} \geq \sum_{k=1}^{K} \min\{\psi_k(t_c), \lambda_{s,k}\} = \psi_c(t_c).$$

The first inequality (a) has been proved.

The inequality (b) holds because LADTSR makes the combo purchase, which happens only when $\psi_c(t_c) \geq \lambda_c$. Combining the inequalities (a) and (b), we conclude the proof for the sufficient condition.

Next, we prove the necessary condition "⇐": when $\sum_{k=1}^{K} \min\{z_k, \lambda_{s,k}\} \geq \lambda_c$, we prove that LADTSR will make the combo purchase by contradiction. Suppose LADTSR does not make the combo purchase (i.e., $t_c > T$), which implies that we have $\psi_c(T) < \lambda_c$ at the end of the demand sequence, according to Line 7 of RDTSR (holds for LADTSR as well). However, we can show that $\psi_c(T) \geq \lambda_c$, which leads to the contradiction.

To see this, we first show that for each item k, we have

$$\min\{\psi_k(T), \lambda_{s,k}\} \geq \min\{z_k, \lambda_{s,k}\}. \tag{15}$$

We consider two cases: whether LADTSR makes the single purchase for item k or not.

i) If LADTSR makes the single purchase for k, we have $\psi_k(T) \geq \lambda_{s,k}$ according to Line 9 of RDTSR (holds for LADTSR as well). Therefore, we have $\min\{\psi_k(T), \lambda_{s,k}\} = \lambda_{s,k} \geq \min\{z_k, \lambda_{s,k}\}$.

ii) Otherwise, if it does not make the single purchase for k (i.e., $t_k > T$), we have

$$\begin{aligned} \psi_k(T) &\overset{(a)}{=} \sum_{t=1}^{T} \mathbb{1}_{\{i(t)=k\}} \cdot \mathbb{1}_{\{t<t_k\}} \cdot \mathbb{1}_{\{t<t_c\}} \cdot a(t) \\ &\overset{(b)}{=} \sum_{t=1}^{T} \mathbb{1}_{\{i(t)=k\}} \cdot a(t) \overset{(c)}{=} z_k, \end{aligned} \tag{16}$$

where (a) is due to the definition of $\psi_k(t)$ in Eq. (9), (b) is due to the combo purchase and single purchase for k is never made according to the assumption (i.e., $t_k, t_c > T$), and (c) is due to the definition of z_k in Eq. (11). In addition, we have $\psi_k(T) \leq \lambda_{s,k}$ because the algorithm does not make the combo purchase and the single purchase for any item k. Therefore, we have $\min\{\psi_k(T), \lambda_{s,k}\} = \psi_k(T) = z_k \geq \min\{z_k, \lambda_{s,k}\}$.

Now, we have $\min\{\psi_k(T), \lambda_{s,k}\} \geq \min\{z_k, \lambda_{s,k}\}$ for any item k by combining above two cases. Finally, we have

$$\begin{aligned}\psi_c(T) &\overset{(a)}{=} \sum_{k=1}^{K} \min\{\psi_k(T), \lambda_{s,k}\} \\ &\overset{(b)}{\geq} \sum_{k=1}^{K} \min\{z_k, \lambda_{s,k}\} \overset{(c)}{\geq} \lambda_c,\end{aligned} \tag{17}$$

where (a) is due to the definition of $\psi_c(t)$ in Eq. (14), (b) is due to $\min\{\psi_k(T), \lambda_{s,k}\} \geq \min\{z_k, \lambda_{s,k}\}$ for each item k, and (c) is due to the necessary condition. Eq. (17) leads to the contradiction that $\psi_c(T) < \lambda_c$. We complete the proof for the necessary condition. □

Lemma A.3. *Given any total demand z, RDTSR will make the combo purchase iff $\sum_{k=1}^{K} \min\{z_k, \lambda_s\} \geq \lambda_c$.*

Proof. Recall that in RDTSR, the single purchase for each item is the same (i.e., λ_s), which is a specific instance of that of LADTSR by letting $\lambda_{s,1} = \cdots = \lambda_{s,K} = \lambda_s$. Thus, the proof of Lemma A.2 also holds in this case. □

The last lemma in this section is to analyze the cost of the optimal offline algorithm. In Lemma A.4, we will show that the cost of the optimal offline algorithm remains the same over all the demand sequences with the same total demand.

Lemma A.4. *Given any total demand z, the cost of the optimal offline algorithm over all the demand sequences with the same total demand z is the same. Let $OPT(z)$ denote the cost of the optimal offline algorithm on total demand z and we have*

$$OPT(z) = \min\{C_c, \sum_{k=1}^{K} \min\{z_k, C_s\}\}. \tag{18}$$

Proof. First, we consider the offline optimal cost without the combo purchase option. In this case, there are two options to fulfill the demand for each item: renting or purchasing. Clearly, the optimal choice for each item k is as follows: when $z_k < C_s$, always rent; otherwise, when $z_k \geq C_s$, purchase at the beginning. Therefore, the total optimal cost for all the items is $\sum_{k=1}^{K} \min\{z_k, C_s\}$.

Next, we consider the offline optimal cost with the combo purchase option. As the combo purchase can fulfill demands for all the items, the optimal choice can only be one of the two cases: either making the combo purchase at the beginning or never making the combo purchase at all. In the first case, the cost is C_c. In the second case, as we previously analyzed, the optimal cost without the combo purchase is $\sum_{k=1}^{K} \min\{z_k, C_s\}$. Therefore, the cost of the optimal offline algorithm is one of the minimum values among the two cases, i.e., $\min\{C_c, \sum_{k=1}^{K} \min\{z_k, C_s\}\}$. □

B Proof of Theorem 3.1

Proof. Our goal is to show that by choosing the thresholds λ_s and λ_c properly, the CR of RDTSR is at most $3 - \frac{1}{C_s} - \frac{1}{C_c}(2 - \frac{1}{C_s})$. In particular, when $\lambda_s = C_s$ and $\lambda_c = C_c$, RDTSR can achieve a CR of $3 - \frac{1}{C_s} - \frac{1}{C_c}(2 - \frac{1}{C_s})$.

The outline of the proof is as follows. Generally, our analysis has two steps:

1) We derive the upper bound of CR over all the demand sequences, which is a function of two thresholds: λ_s and λ_c (see Eq. (42));

2) To minimze the CR by optimizing λ_s and λ_c, we show that when $\lambda_s = C_s$ and $\lambda_c = C_c$, the CR is at most $3 - \frac{1}{C_s} - \frac{1}{C_c}(2 - \frac{1}{C_s})$.

Step 1): Since directly analyzing the CR on demand sequences is hard, we break it down into three substeps:

1a) We derive the upper bound of CR over all the demand sequences with the same total demand z, which is a function of total demand z and two thresholds λ_s, λ_c (see Lemma B.1);

1b) We define a standard total demand (see Definition B.1) to simplify the upper bound we obtained in Step 1a) (see Lemma B.2).

1c) We analyze the upper bound of CR that holds for any standard total demand, which is also the upper bound of CR over all the possible demand sequences. This upper bound is a function of λ_s and λ_c (i.e., Eq. (42)).

Step 1a): We present the definition of CR on a total demand z as follows. Let $\mathcal{D}_z$ denote the set of all the demand sequences that have the same total demand z, i.e., $\mathcal{D}_z = \{\mathbf{D} \mid \mathbf{D} \text{ has a total demand } z\}$. Then the CR on a total demand z, denoted by $CR(z)$, can be defined as the worst ratio of the cost of the online algorithm to that of the optimal offline algorithm over all the demand sequences with the same total demand z, i.e.,

$$CR(z) := \max_{\mathbf{D} \in \mathcal{D}_z} \frac{ALG(\mathbf{D})}{OPT(\mathbf{D})}, \tag{19}$$

where $ALG(\mathbf{D})$ represents the cost of RDTSR and $OPT(\mathbf{D})$ represents the cost of the optimal offline algorithm, both on the demand sequence $\mathbf{D}$.

Furthermore, according to Lemma A.2, the cost of the optimal offline algorithm is the same over all the demand sequences in $\mathcal{D}_z$ (denoted as $OPT(z)$ in Eq. (18)). Therefore, we can rewrite $CR(z)$ as

$$CR(z) = \frac{\max_{\mathbf{D} \in \mathcal{D}_z} ALG(\mathbf{D})}{OPT(z)}. \tag{20}$$

To find the upper bound for $CR(z)$, we can first focus on the upper bound for the cost of RDTSR on the total demand z, denoted by $U_{\text{ALG}}(z)$, i.e.,

$$\max_{\mathbf{D}\in\mathcal{D}_z} ALG(\mathbf{D}) \leq U_{\text{ALG}}(z). \tag{21}$$

With $U_{\text{ALG}}(z)$, we can define the upper bound for the CR on total demand z (denoted by $U_{\text{CR}}(z)$) as

$$U_{\text{CR}}(z) \coloneqq \frac{U_{\text{ALG}}(z)}{OPT(z)}, \tag{22}$$

which is greater or equal to $CR(z)$. To derive the value of $U_{CR}(z)$, we only need to analyze $U_{\text{ALG}}(z)$ because $OPT(z)$ is already obtained in Lemma A.4. Given a total demand z, although the cost of RDTSR can vary across different demand sequences, from Lemma A.3, we know that the combo purchase decision only depends on the total demand z. Therefore, we can consider two cases: if RDTSR makes the combo purchase or not. We present the result of $U_{\text{ALG}}(z)$ in Lemma B.1.

Lemma B.1. *Given a total demand z, the upper bound for RDTSR's cost $U_{\text{ALG}}(z)$ is as follows:*

i) If RDTSR does not make the combo purchase, we have

$$U_{\text{ALG}}(z) = \sum_{k=1}^{K} U_{\text{ALG}}^{k}(z), \tag{23}$$

where $U_{\text{ALG}}^{k}(z)$ denotes the upper bound of the cost of RDTSR for item k on total demand z and we have

$$U_{\text{ALG}}^{k}(z) = \begin{cases} z_k, & z_k < \lambda_s, \\ \lambda_s - 1 + C_s, & \textit{otherwise}; \end{cases} \tag{24}$$

ii) If RDTSR makes the combo purchase, we have

$$\begin{aligned} U_{\text{ALG}}(z) =& \lambda_c - 1 + C_c \\ &+ \min\{\sum_{k=1}^{K} \mathbb{1}_{\{z_k \geq \lambda_s\}}, \frac{\lambda_c - 1}{\lambda_s}\}(C_s - 1). \end{aligned} \tag{25}$$

We prove Lemma B.1 in Appendix C.

Step 1b): From Eqs. (24) and (25), we can see that $U_{\text{CR}}(z)$ depends on the total demand z, which has K variables (i.e., $z_1, \ldots, z_K$). When K is large, it is still challenging for us to analyze the CR on the total demand z. Hence, in the following, we establish a special total demand called *standard total demand* z_{std} to analyze $U_{\text{CR}}(z)$. The standard total demand has two properties: 1) all the standard total demand can be expressed by three variables (see Definition B.1), and 2) for any total demand z, there exists a standard total demand z_{std}, which satisfies $U_{\text{CR}}(z) \leq U_{\text{CR}}(z_{\text{std}})$.

Next, we give the definition of the standard total demand.

Definition B.1. *A total demand z is said to be the standard total demand $z_{\text{std}}(m, n, x)$ if it satisfies:*

i) m items have a total demand of at least C_s;
ii) n items have a total demand equal to λ_s;
iii) among all the remaining items, each item has a total demand within the interval $[0, \lambda_s)$. Let x represent the sum of their total demands;

here m, n, x can be any non-negative integers.

Algorithm 3: Construct $z_{\text{std}}(m, n, x)$ from z

```
Input  : $z$
Output: $z'$
$z' \leftarrow z$;
while there exists item $k$ s.t. $\lambda_s \leq z'_k < C_s$ do
    $z'_k \leftarrow \lambda_s$;
end
```

Next, we show a specific way to construct a standard total demand given any total demand z. The construction approach is presented in Algorithm 3. First, we start with a total demand $z' = \{z'_1, \ldots, z'_K\}$, which equals to z (i.e., $z'_k = z_k$ for item $k = 1, \ldots, K$). Then, for each item k, we modify z' as follows:

i) If $\lambda_s \leq z'_k < C_s$, set z'_k as λ_s.
ii) All the remaining items remain the same.

By Lemma B.2, we claim that for any total demand z, following the above construction approach, we can find a standard total demand to bound $U_{\text{CR}}(z)$.

Lemma B.2. *Given any total demand z, the total demand z′ constructed by Algorithm 3 is a standard total demand. Furthermore, we have*

$$U_{\text{CR}}(z) \leq U_{\text{CR}}(z'). \tag{26}$$

We prove Lemma B.2 in Appendix D.

Step 1c): Given any total demand z, we use $z_{\text{std}}(m, n, x)$ to denote the constructed standard total demand. To obtain $U_{\text{CR}}(z_{\text{std}}(m, n, x))$, we first derive $OPT(z_{\text{std}}(m, n, x))$ and $U_{\text{ALG}}(z_{\text{std}}(m, n, x))$, respectively. For $OPT(z_{\text{std}}(m, n, x))$, we can rewrite Eq. (18) as

$$\begin{aligned} OPT(z_{\text{std}}(m, n, x)) &= \min\{C_c, \sum_{k=1}^{K} \min\{z_k, C_s\}\} \\ &= \min\{C_c, mC_s + n\lambda_s + x\}, \end{aligned} \tag{27}$$

where the second term $mC_s + n\lambda_s + x$ is because there are m items with a total demand of at least C_s and the remaining items have a total demand no greater than C_s and the sum of their total demands is $n\lambda_s + x$ (in fact, n items with a total demand equal to $\lambda_s \leq C_s$ and the other items with a total demand less than λ_s).

For $U_{\text{ALG}}(z_{\text{std}}(m, n, x))$, if RDTSR does not make the combo purchase, according to Lemma B.1, we have

$$U_{\text{ALG}}(z_{\text{std}}(m, n, x)) = (m + n)(\lambda_s - 1 + C_s) + x, \tag{28}$$

where the first item is because there are $m + n$ items with a total demand of at least λ_s, and the second item is due to for the remaining items, the total demand for each item is less than λ_s ($\lambda_s \leq C_s$) and the sum of their total demands is x. On the other hand, if RDTSR makes the combo purchase, according to Lemma B.1, we have

$$U_{\text{ALG}}(z_{\text{std}}(m,n,x)) = \lambda_c - 1 + C_c + \min\{m+n, \frac{\lambda_c - 1}{\lambda_s}\}(C_s - 1), \tag{29}$$

where the term $m+n$ is due to there are $m+n$ items having a total demand greater or equal to λ_s.

Next, we show how to find an upper bound of CR over all the standard total demand. For any standard total demand $z_{\text{std}}(m,n,x)$, we consider two cases: A) RDTSR makes the combo purchase on it, and B) RDTSR does not make the combo purchase on it.

Case A): In this case, according to Eqs. (27) and (29), U_{CR} becomes

$$U_{\text{CR}}(z_{\text{std}}(m,n,x)) = \frac{\lambda_c - 1 + C_c + \min\left\{m+n, \frac{\lambda_c-1}{\lambda_s}\right\}(C_s-1)}{\min\{C_c, mC_s + n\lambda_s + x\}}. \tag{30}$$

Because the algorithm makes the combo purchase, according to Lemma A.2, we have $\sum_{k=1}^{K} \min\{z_k, \lambda_s\} \geq \lambda_c$. Then for the standard total demand, we have

$$\sum_{k=1}^{K} \min\{z_k, \lambda_s\} = (m+n)\lambda_s + x \geq \lambda_c. \tag{31}$$

Observe that we can set $m = 0$ without reducing U_{CR}. To see this, for any $z_{\text{std}}(m,n,x)$ with $m \neq 0$, there exists another standard total demand $z'_{\text{std}}(m',n',x')$ such that $m' = 0$, $n' = m+n$, and $x' = x$. First, we have $(m'+n')\lambda_s + x' = (m+n)\lambda_s + x \geq \lambda_c$, which implies that RDTSR also makes the combo purchase on $z'_{\text{std}}(m',n',x')$ according to Lemma A.2. Hence, we have

$$\begin{aligned} &U_{\text{CR}}(z'_{\text{std}}(m',n',x')) \\ &= \frac{\lambda_c - 1 + C_c + \min\left\{m'+n', \frac{\lambda_c-1}{\lambda_s}\right\}(C_s-1)}{\min\{C_c, m'C_s + n'\lambda_s + x\}} \\ &= \frac{\lambda_c - 1 + C_c + \min\left\{m+n, \frac{\lambda_c-1}{\lambda_s}\right\}(C_s-1)}{\min\{C_c, (m+n)\lambda_s + x\}} \\ &\overset{(a)}{\geq} \frac{\lambda_c - 1 + C_c + \min\left\{m+n, \frac{\lambda_c-1}{\lambda_s}\right\}(C_s-1)}{\min\{C_c, mC_s + n\lambda_s + x\}} \\ &= U_{\text{CR}}(z_{\text{std}}(m,n,x)), \end{aligned} \tag{32}$$

where (a) is due to $(m+n)\lambda_s + x \leq mC_s + n\lambda_s + x$ because we assume $\lambda_s \leq C_s$. Therefore, for any $z_{\text{std}}(m,n,x)$ in Case A with $m > 0$, there exists another standard total demand $z'_{\text{std}}(m',n',x')$ in Case A with $m' = 0$ and $U_{\text{CR}}(z'_{\text{std}}(m',n',x')) \geq U_{\text{CR}}(z_{\text{std}}(m,n,x))$.

Hence, Eq. (30) is upper bounded by

$$\begin{aligned} &U_{\text{CR}}(z_{\text{std}}(m,n,x)) \leq U_{\text{CR}}(z'_{\text{std}}(m',n',x')) \\ &= \frac{\lambda_c - 1 + C_c + \min\left\{n', \frac{\lambda_c-1}{\lambda_s}\right\}(C_s-1)}{\min\{C_c, n'\lambda_s + x\}}, \end{aligned} \tag{33}$$

and Eq. (31) becomes

$$n'\lambda_s + x \geq \lambda_c. \tag{34}$$

According to Eq. (34), the denominator of Eq. (33) is at least $\min\{C_c, \lambda_c\}$. We can rewrite Eq. (33) as

$$\begin{aligned} &U_{\text{CR}}(z_{\text{std}}(m,n,x)) \\ &\leq \frac{\lambda_c - 1 + C_c + \min\left\{n', \frac{\lambda_c-1}{\lambda_s}\right\}(C_s-1)}{\min\{C_c, \lambda_c\}}. \end{aligned} \tag{35}$$

As $\min\{n', \frac{\lambda_c-1}{\lambda_s}\} \leq \frac{\lambda_c-1}{\lambda_s}$, the numerator is at most $\lambda_c - 1 + C_c + \frac{\lambda_c-1}{\lambda_s}(C_s-1)$. Therefore, we have

$$U_{\text{CR}}(z_{\text{std}}(m,n,x)) \leq \frac{\lambda_c - 1 + C_c + \frac{\lambda_c-1}{\lambda_s}(C_s-1)}{\min\{C_c, \lambda_c\}}. \tag{36}$$

Case B): In this case, we consider two subcases: I) The optimal offline algorithm does not make the combo purchase. II) It makes the combo purchase.

Case B-I): According to Eqs. (27) and (28), the upper bound $U_{\text{CR}}(z_{\text{std}})$ becomes

$$\begin{aligned} U_{\text{CR}}(z_{\text{std}}(m,n,x)) &= \frac{(m+n)(\lambda_s - 1 + C_s) + x}{mC_s + n\lambda_s + x} \\ &\overset{(a)}{\leq} \frac{(m+n)(\lambda_s - 1 + C_s) + x}{(m+n)\lambda_s + x} \\ &\leq \frac{(m+n)(\lambda_s - 1 + C_s)}{(m+n)\lambda_s} \\ &= \frac{\lambda_s - 1 + C_s}{\lambda_s}, \end{aligned} \tag{37}$$

where (a) is due to $\lambda_s \leq C_s$.

Case B-II): According to Eqs. (27) and (28), the upper bound $U_{\text{CR}}(z_{\text{std}})$ becomes

$$U_{\text{CR}}(z_{\text{std}}(m,n,x)) = \frac{(m+n)(\lambda_s - 1 + C_s) + x}{C_c}. \tag{38}$$

Because RDTSR does not make the combo purchase, according to Lemma A.2, we have

$$\sum_{k=1}^{K} \min\{z_k, \lambda_s\} = (m+n)\lambda_s + x \leq \lambda_c - 1. \tag{39}$$

Substitute Eq. (39) into (38), we have

$$U_{\text{CR}}(z_{\text{std}}(m,n,x)) \leq \frac{\lambda_c - 1 + (m+n)(C_s-1)}{C_c}. \tag{40}$$

It turns out that we can increase m and n to find another standard total demand z'_{std} in Case A and have $U_{\text{CR}}(z'_{\text{std}}) \geq U_{\text{CR}}(z_{\text{std}})$. Suppose that z'_{std} can be expressed by m', n', and x', where $m' \geq m$, $n' \geq n$ and $x' = x$. The upper bound $U_{\text{CR}}(z'_{\text{std}}(m',n',x'))$ can be expressed as

$$\begin{aligned} &U_{\text{CR}}(z'_{\text{std}}(m,n,x)) \\ &= \frac{\lambda_c - 1 + C_c + \min\{m'+n', \frac{\lambda_c-1}{\lambda_s}\}(C_s-1)}{C_c}. \end{aligned} \tag{41}$$

Clearly, we have $m + n \le m' + n'$; according to Eq. (39), we have $m + n \le \frac{\lambda_c - 1}{\lambda_s}$. Therefore, we have $m + n \le \min\{m' + n', \frac{\lambda_c - 1}{\lambda_s}\}$. It directly implies that $U_{\mathrm{CR}}(z'_{\mathrm{std}}(m', n', x')) \ge U_{\mathrm{CR}}(z_{\mathrm{std}}(m, n, x))$. Furthermore, we can conclude that the upper bound of $U_{\mathrm{CR}}(z_{\mathrm{std}})$ we find for Case A is also the upper bound for Case B-II.

In summary, in Case A and Case B-II the CR is upper bounded by Eq. (36), and in Case B-I it is upper bounded by Eq. (37). Combining two upper bounds, we have the CR which is upper bounded by

$$\max\{\frac{\lambda_s - 1 + C_s}{\lambda_s}, \frac{\lambda_c - 1 + C_c + \frac{\lambda_c - 1}{\lambda_s}(C_s - 1)}{\min\{C_c, \lambda_c\}}\}. \tag{42}$$

Step 2): The next step is to minimize Eq. (42) by optimizing the thresholds λ_s and λ_c. Firstly, observe that when λ_s increases, both terms in the Eq. (42) decrease. Therefore, the function takes the minimum when $\lambda_s = C_s$ since we assume $\lambda_s \le C_s$. Therefore, Eq. (42) becomes

$$\max\{2 - \frac{1}{C_s}, \frac{\lambda_c - 1 + C_c + \frac{\lambda_c - 1}{C_s}(C_s - 1)}{\min\{C_c, \lambda_c\}}\}. \tag{43}$$

Next, we analyze the second item in Eq. (43). We have

$$\begin{aligned}
&\frac{\lambda_c - 1 + C_c + \frac{\lambda_c - 1}{C_s}(C_s - 1)}{\min\{C_c, \lambda_c\}} \\
&\overset{(a)}{=} \frac{\lambda_c - 1 + C_c + \frac{\lambda_c - 1}{C_s}(C_s - 1)}{\lambda_c} \\
&= 1 + \frac{C_c - 1}{\lambda_c} + \frac{\lambda_c - 1}{\lambda_c}\frac{C_s - 1}{C_s} \\
&= 2 - \frac{1}{C_s} + \frac{1}{\lambda_c}(C_c + \frac{1}{C_s} - 2),
\end{aligned} \tag{44}$$

where (a) is due to we assume $\lambda_c \le C_c$. Because $C_c + C_s^{-1} \ge C_s + C_s^{-1} \ge 2$, Eq. (44) decreases monotonically as λ_c increases. When λ_c takes the largest (i.e., $\lambda_c = C_c$), Eq. (44) takes the minimum. Therefore, we have

$$2 - \frac{1}{C_s} + \frac{1}{\lambda_c}(C_c + \frac{1}{C_s} - 2) = 3 - \frac{1}{C_s} - \frac{1}{C_c}(2 - \frac{1}{C_s}). \tag{45}$$

Comparing $3 - \frac{1}{C_s} - \frac{1}{C_c}(2 - \frac{1}{C_s})$ with $2 - \frac{1}{C_s}$, we have

$$\begin{aligned}
&3 - \frac{1}{C_s} - \frac{1}{C_c}(2 - \frac{1}{C_s}) - (2 - \frac{1}{C_s}) \\
&= 1 - \frac{1}{C_c}(2 - \frac{1}{C_s}) \\
&= \frac{1}{C_c}(C_c + \frac{1}{C_s} - 2) \ge 0,
\end{aligned} \tag{46}$$

where the last inequality is due to $C_c + C_s^{-1} \ge 2$. Finally, we conclude that when $\lambda_s = C_s$ and $\lambda_c = C_c$, the CR of RDTSR is at most

$$3 - \frac{1}{C_s} - \frac{1}{C_c}(2 - \frac{1}{C_s}). \tag{47}$$

□

C Proof of Lemma B.1.

Proof. We analyze $U_{\mathrm{ALG}}(z)$ in two cases: A) RDTSR does not make the combo purchase and B) RDTSR makes the combo purchase.

Case A): In this case, we want to show that if RDTSR does not make the combo purchase (i.e., $t_c > T$), then for any item k, we have

$$U^k_{\mathrm{ALG}}(z) = \begin{cases} z_k, & z_k < \lambda_s, \\ \lambda_s - 1 + C_s, & \text{otherwise.} \end{cases}$$

We consider two subcases: I) the total demand of item k is less than the single purchase threshold (i.e., $z_k < \lambda_s$) and II) the total demand of item k is no less than the single purchase threshold (i.e., $z_k \ge \lambda_s$).

Case A-I): If $z_k < \lambda_s$, RDTSR will never make the single purchase for item k. This is because RDTSR makes the single purchase for item k in some time-slot t only when $\psi_k(t) \ge \lambda_s$ $(t \le T)$. However, we have

$$\begin{aligned}
\psi_k(T) &\overset{(a)}{=} \sum_{t=1}^{T} \mathbb{1}_{\{i(t)=k\}} \cdot \mathbb{1}_{\{t<t_k\}} \cdot \mathbb{1}_{\{t<t_c\}} \cdot a(t) \\
&\le \sum_{t=1}^{T} \mathbb{1}_{\{i(t)=k\}} \cdot a(t) \\
&\overset{(b)}{=} z_k < \lambda_s,
\end{aligned} \tag{48}$$

where (a) is due to the equivalent definition of $\psi_k(t)$ (i.e., Eq. (9)) and (b) is due to the definition of z_k (i.e., Eq. (11)). Consequently, RDTSR will not make the single purchase for item k and covers all the demand for item k by rental with a cost of z_k. Moreover, we have $\psi_k(T) = z_k$ because neither the single purchase nor the combo purchase is made.

Case A-II): If $z_k \ge \lambda_s$, then we can prove that RDTSR will make the single purchase for item k by contradiction. Suppose that RDTSR does not make the single purchase for k, then we have $\mathbb{1}_{\{t<t_k\}} = 1$ for any time-slot t. Similarly, as we assumed that RDTSR does not make the combo purchase, we have $\mathbb{1}_{\{t<t_c\}} = 1$ for any time-slot t. Therefore, according to Lemma A.1, we have $\psi_k(T) = \sum_{t=1}^{T} \mathbb{1}_{\{i(t)=k\}} \cdot a(t) = z_k$. Since $z_k \ge \lambda_s$, we know that RDTSR will make the single purchase for item k, which is constracting with our assumption. Therefore, we conclude that RDTSR will make the single purchase for item k, and the corresponding single purchase cost for item k is C_s. Next, we analyze the rental cost of RDTSR for item k before it makes the single purchase. It turns out that the rental cost is at most $\lambda_s - 1$ because if the total rental cost for item k is greater than $\lambda_s - 1$ (i.e., greater or equal to λ_s) in some time-slot t, then according to Lemma A.1, we have $\psi_k(t) \ge \lambda_s$, which means that RDTSR will make the single purchase for item k in time-slot t rather than opting for renting. In summary, the total cost for item k is $\lambda_s - 1 + C_s$.

Now we derive $U^k_{\mathrm{ALG}}(z)$ as

$$U^k_{\mathrm{ALG}}(z) = \begin{cases} z_k, & z_k < \lambda_s, \\ \lambda_s - 1 + C_s, & \text{otherwise,} \end{cases} \tag{49}$$

and the corresponding $U_{\text{ALG}}(z)$ is

$$U_{\text{ALG}}(z) = \sum_{k=1}^{K} U_{\text{ALG}}^{k}(z). \tag{50}$$

Case B): We want to show that if RDTSR makes the combo purchase, we have

$$\begin{aligned} U_{\text{ALG}}(z) =& \lambda_c - 1 + C_c \\ &+ \min\{\sum_{k=1}^{K} \mathbb{1}_{\{z_k \geq \lambda_s\}}, \frac{\lambda_c - 1}{\lambda_s}\}(C_s - 1). \end{aligned} \tag{51}$$

In time-slot t_c, we know that RDTSR makes the combo purchase with a cost of C_c. No cost is incurred after the time-slot t_c because all the following demands are covered by the combo purchase. Next, we analyze the single purchase cost and the rental cost before t_c.

For the single purchase cost, let n_s denote the number of single purchases made by RDTSR, and thus the total single purchase cost is $n_s C_s$.

For the total rental cost, it turns out that the rental cost of RDTSR during the interval $[1, t_c - 1]$ is at most $\lambda_c - 1 - n_s$. To see this, in the following, we first show that the total rental cost until time-slot $(t_c - 1)$ is at most $\psi_c(t_c - 1) - n_s$, and then we show that $\psi_c(t_c - 1) \leq \lambda_c - 1$.

First, we show that the total rental cost until time-slot $(t_c - 1)$ is at most $\psi_c(t_c - 1) - n_s$. As RDTSR does not make the combo purchase during the interval $[1, t_c - 1]$, we can directly apply the result in Case A) for the demand sequence during the interval $[1, t_c - 1]$. Specifically, for any demand sequence $\mathbf{D} = \{d(t)\}_{t=1}^{T} \in \mathcal{D}_z$ with the total demand z, let $\mathbf{D}'$ denote the subsequence during $[1, t_c - 1]$, i.e, $\mathbf{D}' = \{d(t)\}_{t=1}^{t_c-1}$. Let z' denote the total demand of $\mathbf{D}'$. For each item k, we consider two cases: $z'_k < \lambda_s$ and $z'_k \geq \lambda_s$.

i) If $z'_k < \lambda_s$, according to the analysis in Case A-I), we have the rental cost is z'_k and $\psi_k(t_c - 1) = z'_k$. Since $z'_k < \lambda_s$, the rental cost for item k is $z'_k = \psi_k(t_c - 1) < \min\{\psi_k(t_c - 1), \lambda_s\}$.
ii) If $z'_k \geq \lambda_s$, according to the analysis in Case A-II), RDTSR will make the single purchase for item k (i.e., $\psi_k(t_c - 1) \geq \lambda_s$) and the rental cost is at most $\lambda_s - 1$. Therefore, the rental cost is at most $\lambda_s - 1 = \min\{\psi_k(t_c - 1), \lambda_s\} - 1$.

Because we assume that the number of single purchases is n_s, therefore the total rental cost is at most $\sum_{k=1}^{K} \min\{\psi_k(t_c - 1), \lambda_s\} - n_s$. According to the definition of $\psi_c(t)$ (i.e., Eq. (14)), we have $\psi_c(t_c - 1) = \sum_{k=1}^{K} \min\{\psi_k(t_c - 1), \lambda_s\}$. Therefore, the total rental cost for all the items is at most $\psi_c(t_c - 1) - n_s$.

Then we show that

$$\psi_c(t_c - 1) \leq \lambda_c - 1. \tag{52}$$

This is because if $\psi_c(t_c - 1) > \lambda_c - 1$ (i.e., $\psi_c(t_c - 1) \geq \lambda_c$), then according to Line 7 of RDTSR, it will make the combo purchase in time-slot $t_c - 1$ rather than t_c. This leads to the contradiction. Therefore, we conclude that the total rental cost is at most $\lambda_c - 1 - n_s$.

In summary, if RDTSR makes the combo purchase and n_s single purchases, its total cost is at most

$$\begin{aligned} &\lambda_c - 1 - n_s + C_c + n_s C_s \\ =& \lambda_c - 1 + C_s + n_s(C_s - 1). \end{aligned} \tag{53}$$

Next, we identify the value of n_s. Clearly, $U_{\text{ALG}}(z)$ increases monotonically as n_s increases. Therefore, we need to identify the possible maximum value of n_s. First, the count of single purchases cannot exceed the number of items for which its total demand exceeds λ_s. Then, we have

$$n_s \leq \sum_{k=1}^{K} \mathbb{1}_{\{z_k \geq \lambda_s\}}. \tag{54}$$

On the other hand, we can show that $n_s \leq (\lambda_c - 1)/\lambda_s$. To see this, we only need to show that $n_s \lambda_s \leq \psi_c(t_c - 1) \leq \lambda_c - 1$. The first inequality holds because if RDTSR makes the single purchase for item k, we have $\psi_k(t_c - 1) \geq \lambda_s$, which means $\lambda_s = \min\{\psi_k(t_c - 1), \lambda_s\}$ Therefore, for these n_s items, we have $n_s \lambda_s \leq \sum_{k=1}^{K} \min\{\psi_k(t_c - 1), \lambda_s\} = \psi_c(t_c - 1)$. We have shown that $\psi_c(t_c - 1) \leq \lambda_c - 1$ in Eq. (52), so the second inequality holds.

Now, we obtain the upper bound of n_s, which is $n_s \leq \min\{\sum_{k=1}^{K} \mathbb{1}_{\{z_k \geq \lambda_s\}}, (\lambda_c - 1)/\lambda_s\}$. The upper bound of $ALG(z)$ is then

$$\begin{aligned} U_{\text{ALG}}(z) =& \lambda_c - 1 + C_c \\ &+ \min\{\sum_{k=1}^{K} \mathbb{1}_{\{z_k \geq \lambda_s\}}, \frac{\lambda_c - 1}{\lambda_s}\}(C_s - 1). \end{aligned} \tag{55}$$

□

D Proof of Lemma B.2

Proof. First, we show that the output total demand z' is a standard total demand. According to Algorithm 3, there is one loop, i.e., Lines 2-3. After this loop, no item has a total demand within the interval (λ_s, C_s). In other words, the total demand can fall into one of three categories: i) at least C_s, ii) equal to λ_s, or iii) within the interval $[0, \lambda_s)$. Clearly, the constructed total demand z' satisfies the three conditions in Definition B.1 at the same time.

Second, we show that $U_{\text{CR}}(z') \geq U_{\text{CR}}(z)$. Initially, we have $z' = z$. Then we only need to show that after the loop, $U_{\text{CR}}(z')$ does not decrease. As we will see in Lemma D.1, for each iteration in this loop, $U_{\text{CR}}(z')$ does not decrease. □

Lemma D.1. *Given any total demand z, if there exists an item j such that $\lambda_s \leq z_j < C_s$, then there exists another total demand $z' = \{z'_1, \ldots, z'_K\}$ such that*

$$z'_k = \begin{cases} \lambda_s, & k = j, \\ z_k, & k \neq j, \end{cases} \tag{56}$$

and we have $U_{\text{CR}}(z) \leq U_{\text{CR}}(z')$.

Proof. By the definition of $U_{\text{CR}}(z)$, we can proceed the proof in the following two steps: 1) we show that $OPT(z) \geq OPT(z')$, and 2) we show that $U_{\text{ALG}}(z) = U_{\text{ALG}}(z')$.

Step 1): For $OPT(z)$, we have

$$\begin{aligned}OPT(z) &\overset{(a)}{=} \min\{C_c, \sum_{k=1}^{K} \min\{C_s, z_k\}\} \\ &\overset{(b)}{=} \min\{C_c, z_j + \sum_{k=1,k\neq j}^{K} \min\{C_s, z_k\}\} \\ &\overset{(c)}{\geq} \min\{C_c, z'_j + \sum_{k=1,k\neq j}^{K} \min\{C_s, z'_k\}\} \\ &\overset{(d)}{=} \min\{C_c, \sum_{k=1}^{K} \min\{C_s, z'_k\}\} \overset{(e)}{=} OPT(z'),\end{aligned} \tag{57}$$

where (a) comes from the definition of $OPT(z)$ in Eq. (18), (b) is due to $\min\{C_s, z_j\} = z_j$ because $z_j \leq C_s$, (c) is due to $z'_k \leq z_k$ for item $k = 1, \ldots, K$ according to Eq. (56), (d) is due to $\min\{C_s, z'_j\} = z'_j$ because $z'_j = \lambda_s \leq C_s$, and (e) comes from Eq. (18).

Step 2): Then, we show that $U_{\text{ALG}}(z) = U_{\text{ALG}}(z')$. To see this, we first show that $\sum_{k=1}^{K} \min\{z_k, \lambda_s\} = \sum_{k=1}^{K} \min\{z'_k, \lambda_s\}$, which implies that RDTSR's combo purchase decisions are the same for z and z' according to Lemma A.2. This is due to two facts: 1) For item j, we have $\min\{z_j, \lambda_s\} = \min\{z'_j, \lambda_s\} = \lambda_s$ because $z_j, z'_j \geq \lambda_s$. 2) For item k that $k \neq j$, we have $z_k = z'_k$ according to Eq. (56). Therefore, we conclude that $\sum_{k=1}^{K} \min\{z_k, \lambda_s\} = \sum_{k=1}^{K} \min\{z'_k, \lambda_s\}$ because $\min\{z_k, \lambda_s\} = \min\{z'_k, \lambda_s\}$ for any item k. Next, to compare $U_{\text{ALG}}(z)$ and $U_{\text{ALG}}(z')$, we consider two cases: whether RDTSR makes the combo purchase on z and z' or not.

If RDTSR makes the combo purchase, recall that according to Eq. (25) we have $U_{\text{ALG}}(z) = \lambda_c - 1 + C_c + \min\{\sum_{k=1}^{K} \mathbb{1}_{\{z_k \geq \lambda_s\}}, \frac{\lambda_c - 1}{\lambda_s}\}(C_s - 1)$. We only need to compare the term related to z and z' (i.e., $\sum_{k=1}^{K} \mathbb{1}_{\{z_k \geq \lambda_s\}}$ and $\sum_{k=1}^{K} \mathbb{1}_{\{z'_k \geq \lambda_s\}}$). It turns out that $\sum_{k=1}^{K} \mathbb{1}_{\{z_k \geq \lambda_s\}} = \sum_{k=1}^{K} \mathbb{1}_{\{z'_k \geq \lambda_s\}}$ because for item j, we have $z_j, z'_j \geq \lambda_s$ (i.e., $\mathbb{1}_{\{z_j \geq \lambda_s\}} = \mathbb{1}_{\{z'_j \geq \lambda_s\}} = 1$), and for item k that $k \neq j$, we have $z_k = z'_k$ according to Eq. (56) (i.e., $\mathbb{1}_{\{z_k \geq \lambda_s\}} = \mathbb{1}_{\{z'_k \geq \lambda_s\}}$). Therefore, in this case, we have $U_{\text{ALG}}(z) = U_{\text{ALG}}(z')$.

If RDTSR does not make the combo purchase, according to Eq. (24), we have

$$U^k_{\text{ALG}}(z) = \begin{cases} z_k, & z_k < \lambda_s, \\ \lambda_s - 1 + C_s, & \text{otherwise.} \end{cases}$$

We claim that $U^k_{\text{ALG}}(z) = U^k_{\text{ALG}}(z')$ for $k = 1, \ldots, K$. For item j, we have $z_j, z'_j \geq \lambda_s$ and $U^j_{\text{ALG}}(z) = U^j_{\text{ALG}}(z') = \lambda_s - 1 + C_s$; for item k such that $k \neq j$, we have $z_k = z'_k$ according to Eq. (56) and hence $U^k_{\text{ALG}}(z) = U^k_{\text{ALG}}(z')$. Therefore, we have $U_{\text{ALG}}(z) = \sum_{k=1}^{K} U^k_{\text{ALG}}(z) = U^k_{\text{ALG}}(z')$.

Combining two cases, we have $U_{\text{ALG}}(z) = U_{\text{ALG}}(z')$. □

E Proof of Lemma 4.1

Proof. First, we claim that given any total demand z and a prediction y, the cost of FTP over all the demand sequences $\mathbf{D} \in \mathcal{D}_z$ is the same. We use $ALG(z)$ to denote the cost of FTP on a total demand z. There are two cases for FTP: If the prediction suggests the combo purchase, i.e., $\sum_{k=1}^{K} \min\{C_s, y_k\} \geq C_c$, FTP will make the combo purchase in $t = 1$. Otherwise, FTP will never make the combo purchase. For each item k, FTP will make the single purchase for k if the prediction suggests so (i.e., $y_k \geq C_s$), otherwise FTP always pays the demand for k by rental. Therefore, we have

$$ALG(z) = \begin{cases} C_c, & \sum_{k=1}^{K} \min(C_s, y_k) \geq C_c, \\ \sum_{k=1}^{K} ALG_k(z), & \text{otherwise,} \end{cases} \tag{58}$$

where $ALG_k(z)$ denotes the cost of FTP for item k when it does not make the combo purchase, i.e.,

$$ALG_k(z) = \begin{cases} C_s, & y_k \geq C_s, \\ z_k, & \text{otherwise.} \end{cases} \tag{59}$$

We can see that the cost of FTP only depends on the actual total demand z and the predicted total demand y. For the optimal offline algorithm, according to Lemma A.4, we have

$$OPT(z) = \min\{C_c, \sum_{k=1}^{K} \min\{z_k, C_s\}\}. \tag{60}$$

Therefore, we can analyze the cost of FTP and the cost of the optimal offline algorithm based on the total demand. Specifically, given a total demand of z, we consider two cases: A) FTP makes the combo purchase on z, and B) FTP does not make the combo purchase on z. Similarly, we consider two cases for the optimal offline algorithm: I) the optimal offline algorithm makes the combo purchase, and II) the optimal offline algorithm does not make the combo purchase. Therefore, we consider $2 \times 2 = 4$ cases and we show that the result holds for each case.

Case A-I): In this case, we have $ALG(z) = C_c$ and $OPT(z) = C_c$ from Eqs. (58) and (60), respectively. As $\eta \geq 0$, we have $ALG(z) \leq OPT(z) + \eta$.

Case A-II): In this case, we have $ALG(z) = C_c$ and $OPT(z) = \sum_{k=1}^{K} \min\{z_k, C_s\}$ from Eqs. (58) and (60), respectively. Next, we show that for each item k, we have

$$\min\{z_k, C_s\} + \eta_k \geq \min\{y_k, C_s\}. \tag{61}$$

We show this for $z_k \geq C_s$ and $z_k < C_s$, respectively:

i) If $z_k \geq C_s$, we have $\min\{z_k, C_s\} + \eta_k = C_s + \eta_k \geq C_s \geq \min\{y_k, C_s\}$.

ii) If $z_k < C_s$, we have $\min\{z_k, C_s\} + \eta_k = z_k + \eta_k \geq y_k \geq \min\{y_k, C_s\}$.

Then we have

$$\begin{aligned} ALG(z) &= C_c \\ &\overset{(a)}{\leq} \sum_{k=1}^{K} \min\{y_k, C_s\} \\ &\overset{(b)}{\leq} \sum_{k=1}^{K} (\min\{z_k, C_s\} + \eta_k) \\ &\leq OPT(z) + \eta, \end{aligned} \tag{62}$$

where (a) is due to the condition that the prediction suggests the combo purchase (i.e., Eq. (58)) and (b) is due to Eq. (61).

Case B-I): In this case, we have $ALG(z) = \sum_{k=1}^{K} ALG_k(z)$ and $OPT(z) = C_c$ from Eqs. (58) and (60), respectively. We show that for each item k, we have

$$ALG_k(z) \leq \min\{C_s, y_k\} + \eta_k. \tag{63}$$

We show this for $y_k \geq C_s$ and $y_k < C_s$, respectively:

i) If $y_k \geq C_s$, we have $ALG_k(z) = C_s = \min\{C_s, y_k\} \leq \min\{C_s, y_k\} + \eta_k$,
ii) If $y_k < C_s$, we have $ALG_k(z) = z_k \leq y_k + \eta_k = \min\{C_s, y_k\} + \eta_k$.

Then we have

$$\begin{aligned} ALG(z) &= \sum_{k=1}^{K} ALG_k(z) \\ &\overset{(a)}{\leq} \sum_{k=1}^{K} (\min\{C_s, y_k\} + \eta_k) \\ &\overset{(b)}{<} C_c + \eta = OPT + \eta, \end{aligned} \tag{64}$$

where (a) comes from the Eq. (63) and (b) comes from the condition that the prediction does not suggest the combo purchase (i.e., $\sum_{k=1} \min\{C_s, y_k\} < C_c$).

Case B-II): In this case, we have $ALG(z) = \sum_{k=1}^{K} ALG_k(z)$ and $OPT(z) = \sum_{k=1}^{K} \min\{z_k, C_s\}$ from Eqs. (58) and (60), respectively. Without the combo purchase, for each item k, this problem degenerates into a ski-rental problem (i.e., z_k is the ski days and C_s is the purchase price). In this case, FTP will make the single purchase for item k if the predicted total demand $y_k \geq C_s$ or keep renting for all the demand for item k if the predicted total demand $y_k < C_s$. In fact, FTP is exactly Algorithm 1 in Purohit, Svitkina, and Kumar (2018). Therefore, we can apply the Lemma 2.1 proved in Purohit, Svitkina, and Kumar (2018), which states that the cost of the algorithm is less than the sum of the offline optimal and the prediction error, i.e.,

$$ALG_k(z) \leq \min\{z_k, C_s\} + \eta_k. \tag{65}$$

Finally, by summing up all the items, we obtain

$$\begin{aligned} ALG(z) = \sum_{k=1}^{K} ALG_k(z) \leq \sum_{k=1}^{K} \min\{z_k, C_s\} + \eta_k \\ = OPT(z) + \eta. \end{aligned} \tag{66}$$

Combining all the four subcases, we complete the proof. □

F Proof of Consistency

Proof. In this section, we prove the first upper bound $(1 + \theta + \theta^2) + \frac{1+2\theta}{1-\theta} \frac{\eta}{OPT(\mathbf{D})}$. The outline of the proof is as follows. Our goal is to show that this upper bound holds for any demand sequence under any predictions. First, given any predictions, we consider two cases: if $\lambda_c = \theta^2 C_c$ and if $\lambda_c = C_c/\theta$. For each case, given any demand sequence, similar to the proof of Lemma 4.1, we consider four subcases. That is, we consider two subcases for LADTSR: A) LADTSR makes the combo purchase. B) LADTSR does not make the combo purchase. For each subcase A and B, we consider two subcases for the optimal offline algorithm: I) The optimal offline algorithm makes the combo purchase. II) The optimal offline algorithm does not make the combo purchase. Therefore, we consider 8 subcases. For each subcase, we will derive the upper bound of CR, which is a function of the trust parameter θ and the prediction error η.

Before we delve into each subcase, similar to the proof for Theorem 3.1 and Lemma 4.1, we need to analyze the cost of LADTSR on a total demand z. We still use $CR(z)$ to denote the CR of LADTSR, $U_{\text{ALG}}(z)$ to denote the upper bound of the cost of LADTSR, and $U_{\text{CR}}(z)$ to denote the upper bound of CR of LADTSR, respectively. Therefore, we have

$$CR(z) \leq U_{\text{CR}}(z) = \frac{U_{\text{ALG}}(z)}{OPT(z)}. \tag{67}$$

Next, we analyze $U_{\text{ALG}}(z)$. Similar to Lemma B.1, we present the result of $U_{\text{ALG}}(z)$ for LADTSR in Lemma F.1.

Lemma F.1. *Given a total demand z, the upper bound for LADTSR's cost $U_{\text{ALG}}(z)$ is as follows:*

i) If LADTSR does not make the combo purchase,

$$U_{\text{ALG}}(z) = \sum_{k=1}^{K} U_{\text{ALG}}^k(z), \tag{68}$$

where $U_{\text{ALG}}^k(z)$ denotes the upper bound for LADTSR's cost on item k and we have

$$U_{\text{ALG}}^k(z) = \begin{cases} z_k, & \text{if } z_k < \lambda_{s,k}, \\ \lambda_{s,k} + C_s, & \text{otherwise.} \end{cases} \tag{69}$$

ii) If LADTSR makes the combo purchase, let s_1 denote the number of items whose single purchase is C_s/θ and its total demand for this item is no less than the single purchase C_s/θ (i.e., those items k such that $\lambda_{s,k} = C_s/\theta$ and $z_k \geq \lambda_{s,k} = C_s/\theta$), and let s_2 denote the number of items whose single purchase is θC_s and its total demand for this item is no less than the single purchase θC_s (i.e., those items k such that $\lambda_{s,k} = \theta C_s$ and $z_k \geq \lambda_{s,k} = \theta C_s$), then we have

$$U_{\text{ALG}}(z) = \lambda_c + C_c + f(s_1, s_2) C_s, \tag{70}$$

where $f(s_1, s_2)$ is a function that calculates the maximum number of single purchases and is computed by

$$f(s_1, s_2) = \min\{\frac{\lambda_c}{\theta C_s}, s_2 + \frac{\lambda_c - \theta C_s s_2}{C_s/\theta}, s_1 + s_2\}. \tag{71}$$

We prove Lemma F.1 in Appendix G.

Next, we consider the case when $\lambda_c = \theta^2 C_c$, which means the prediction suggests a combo purchase. According to Line 8 in LADTSR, we have

$$\sum_{k=1}^{K} \min\{y_k, C_s\} \geq C_c. \tag{72}$$

Case A-I): In this case, we have $OPT(z) = C_c$. For LADTSR, according to Lemma F.1, we have

$$\begin{aligned} U_{\mathrm{ALG}}(z) &= \lambda_c + C_c + f(s_1, s_2)C_s \\ &\overset{(a)}{\leq} \lambda_c + C_c + \frac{\lambda_c}{\theta C_s} C_s \\ &= \theta^2 C_c + C_c + \theta C_c \\ &= (1+\theta+\theta^2)C_c \\ &\leq (1+\theta+\theta^2)(OPT(z)+\eta), \end{aligned} \tag{73}$$

where (a) is due to $f(s_1, s_2) \leq \frac{\lambda_c}{\theta C_s}$. Therefore, we have

$$CR(z) \leq \frac{U_{\mathrm{ALG}}(z)}{OPT(z)} \leq (1+\theta+\theta^2)(1+\frac{\eta}{OPT(z)}). \tag{74}$$

Case A-II): In this case, from Lemma A.4 we have $OPT(z) = \sum_{k=1}^{K} \min\{C_s, z_k\}$. For LADTSR, from Eq. (73), we have

$$U_{\mathrm{ALG}}(z) \leq (1+\theta+\theta^2)C_c. \tag{75}$$

Next, we will show that $C_c \leq OPT(z) + \eta$. This is because

$$\begin{aligned} C_c &\overset{(a)}{\leq} \sum_{k=1}^{K} \min\{y_k, C_s\} \\ &\overset{(b)}{\leq} \sum_{k=1}^{K} (\min\{z_k, C_s\} + \eta_k) \\ &= OPT + \eta, \end{aligned} \tag{76}$$

where (a) is due to Eq. (72) and (b) is shown as follows:

- If $z_k \geq C_s$, we have $\min\{z_k, C_s\} + \eta_k = C_s + \eta_k \geq C_s \geq \min\{y_k, C_s\}$.
- If $z_k < C_s$, we have $\min\{z_k, C_s\} + \eta_k = z_k + \eta_k \geq y_k \geq \min\{y_k, C_s\}$.

Therefore we have,

$$CR(z) \leq \frac{U_{\mathrm{ALG}}(z)}{OPT(z)} = (1+\theta+\theta^2)(1+\frac{\eta}{OPT(z)}). \tag{77}$$

Case B-I): In this case, we will show that such a total demand does not exist. First, we have

$$\begin{aligned} \sum_{k=1}^{K} \min\{z_k, \lambda_{s,k}\} &\overset{(a)}{\geq} \sum_{k=1}^{K} \min\{z_k, \theta C_s\} \\ &\geq \sum_{k=1}^{K} \min\{\theta z_k, \theta C_s\} \\ &= \theta \sum_{k=1}^{K} \min\{z_k, C_s\} \\ &\overset{(b)}{\geq} \theta C_c \geq \theta^2 C_c = \lambda_c, \end{aligned} \tag{78}$$

where (a) is due to $\lambda_{s,k} \geq \theta C_s$ for any item k and (b) is due to the optimal offline algorithm making the combo purchase. Based on Lemma A.2, Eq. (78) implies that LADTSR will make the combo purchase, leading to the contradiction that it does not make the combo purchase in this case.

Case B-II): In this subcase, the problem degenerates into the ski-rental problem. Furthermore, our LADTSR is equivalent to the LADTSR proposed in Purohit, Svitkina, and Kumar (2018). That is, buy on day $\lceil \theta C_s \rceil$ if $y_k \geq C_s$, otherwise buy on day $\lceil C_s/\theta \rceil$. Therefore, we can directly apply the Theorem 2.2 in their work, i.e., for each item k, the cost of LADTSR for item k (denoted by $ALG_k(z)$) is at most $(1+\theta)OPT_k(z) + \frac{1}{1-\theta}\eta_k$, where $OPT_k(z)$ is the cost of the optimal offline algorithm for item k. As the total cost of both algorithms in this case is the sum of their cost for each item, we have

$$\begin{aligned} CR(z) &= \frac{\sum_{k=1}^{K} ALG_k(z)}{\sum_{k=1}^{K} OPT_k(z)} \\ &\leq \frac{(1+\theta)\sum_{k=1}^{K} OPT_k(z) + \frac{1}{1-\theta}\sum_{k=1}^{K}\eta_k}{\sum_{k=1}^{K} OPT_k(z)} \\ &= 1+\theta+\frac{1}{1-\theta}\frac{\eta}{OPT(z)}. \end{aligned} \tag{79}$$

Combining all the four subcases, we have

$$\begin{aligned} CR(z) \leq& \max\{1+\theta+\theta^2, 1+\theta\} \\ &+ \max\{1+\theta+\theta^2, \frac{1}{1-\theta}\}\frac{\eta}{OPT(z)}. \end{aligned} \tag{80}$$

Clearly, we have $1+\theta \leq 1+\theta+\theta^2$. Comparing $1+\theta+\theta^2$ and $\frac{1}{1-\theta}$, we have

$$\begin{aligned} &\frac{1}{1-\theta} - (1+\theta+\theta^2) \\ &= \frac{1-(1+\theta+\theta^2)(1-\theta)}{1-\theta} \\ &= \frac{1-(1-\theta^3)}{1-\theta} \\ &= \frac{\theta^3}{1-\theta} > 0. \end{aligned} \tag{81}$$

Therefore, for the case when $\lambda_c = \theta^2 C_c$, we have

$$CR(z) \leq 1+\theta+\theta^2 + \frac{1}{1-\theta}\frac{\eta}{OPT(z)}. \tag{82}$$

Now we consider another case when $\lambda_c = C_c/\theta$. According to Lines 8-12 in LADTSR, we have

$$\sum_{k=1}^{K} \min\{C_s, y_k\} < C_c. \tag{83}$$

We follow the above proof and consider four subcases.

Case A-I): In this case, we have $OPT(z) = C_c$ and LADTSR makes the combo purchase. First, we need to find the connection between the $OPT(z)$ and the prediction error η. It turns out that we can show that

$$C_c \leq \frac{\theta}{1-\theta}\eta. \tag{84}$$

Suppose we have n items (index from 1 to n) with single purchase threshold θC_s and m items (index from $n+1$ to $n+m$) with threshold C_s/θ. Clearly, we have $n+m=K$.

Because LADTSR makes the combo purchase, according to Lemma A.2, we have $\sum_{k=1}^{K}\min\{z_k,\lambda_{s,k}\}\geq\lambda_c$. Therefore, we have

$$\sum_{k=1}^{n}\min\{z_k,\theta C_s\}+\sum_{k=n+1}^{n+m}\min\{z_k,\frac{C_s}{\theta}\}\geq\frac{C_c}{\theta}. \tag{85}$$

Move $\sum_{k=1}^{n}\min\{z_k,\theta C_s\}$ to the RHS and we have,

$$\begin{aligned}\sum_{k=n+1}^{n+m}\min\{z_k,\frac{C_s}{\theta}\}&\geq\frac{C_c}{\theta}-\sum_{k=1}^{n}\min\{z_k,\theta C_s\}\\&\overset{(a)}{\geq}\frac{C_c}{\theta}-n\theta C_s\\&\geq\frac{1-\theta}{\theta}C_c+(C_c-n\theta C_s)\\&\geq\frac{1-\theta}{\theta}C_c+(C_c-nC_s),\end{aligned} \tag{86}$$

where (a) is due to $\min\{z_k,\theta C_s\}\leq\theta C_s$.

On the other hand, according to Lines 1-7 in LADTSR, for any item k, if $\lambda_{s,k}=C_s/\theta$, we have $y_k<C_s$; if $\lambda_{s,k}=\theta C_s$, we have $y_k\geq C_s$. Then, Eq. (83) becomes

$$\sum_{k=1}^{K}\min\{C_s,y_k\}=nC_s+\sum_{k=n+1}^{n+m}y_k<C_c. \tag{87}$$

Move nC_s to the RHS, we have

$$\sum_{k=n+1}^{n+m}y_k<C_c-nC_s. \tag{88}$$

Substitute Eq. (88) into (86) we have,

$$\sum_{k=n+1}^{n+m}\min\{z_k,\frac{C_s}{\theta}\}\geq\frac{1-\theta}{\theta}C_c+\sum_{k=n+1}^{n+m}y_k. \tag{89}$$

On the other hand, we have

$$\sum_{k=n+1}^{n+m}\min\{z_k,\frac{C_s}{\theta}\}\leq\sum_{k=n+1}^{n+m}z_k\leq\sum_{k=n+1}^{n+m}(y_k+\eta_k), \tag{90}$$

Combining Eqs. (89) and (90) we obtain,

$$\begin{aligned}\frac{1-\theta}{\theta}C_c+\sum_{k=n+1}^{n+m}y_k&\leq\sum_{k=n+1}^{n+m}(y_k+\eta_k),\\\frac{1-\theta}{\theta}C_c&\leq\sum_{k=n+1}^{n+m}\eta_k\leq\eta.\end{aligned} \tag{91}$$

Therefore, we show that $C_c\leq\frac{\theta}{1-\theta}\eta$.

Next, we analyze $U_{\text{ALG}}(z)$. According to Lemma F.1, as $f(s_1,s_2)\leq s_2+\frac{\lambda_c-\theta C_s s_2}{C_s/\theta}$, we have

$$\begin{aligned}U_{\text{ALG}}(z)&\leq\lambda_c+C_c+(s_2+\frac{\lambda_c-\theta C_s s_2}{C_s/\theta})C_s\\&=\frac{C_c}{\theta}+C_c+(s_2+\frac{C_c}{C_s}-\theta^2 s_2)C_s\\&\leq\frac{C_c}{\theta}+C_c+(s_2+\frac{C_c}{C_s})C_s.\end{aligned} \tag{92}$$

Next, we show that $s_2\leq C_c/C_s$. From Eq. (87), as $\sum_{k=n+1}^{n+m}y_k\geq 0$, we have $n<\frac{C_c}{C_s}$. Clearly, we have $s_2\leq n<C_c/C_s$ because n is the total number of items with a threshold equal to θC_s. Therefore, we have

$$\begin{aligned}U_{\text{ALG}}(z)&\leq\frac{C_c}{\theta}+C_c+\frac{2C_c}{C_s}C_s\\&=C_c+(2+\frac{1}{\theta})C_c.\end{aligned} \tag{93}$$

Substitute Eq. (84) into Eq. (93), we have

$$\begin{aligned}U_{\text{ALG}}(z)&\leq C_c+\frac{\theta}{1-\theta}\eta(2+\frac{1}{\theta})\\&=OPT(z)+\frac{1+2\theta}{1-\theta}\eta.\end{aligned} \tag{94}$$

Therefore, we have

$$CR(z)\leq\frac{U_{\text{ALG}}(z)}{OPT(z)}=1+\frac{1+2\theta}{1-\theta}\frac{\eta}{OPT(z)}. \tag{95}$$

Case A-II): In this case, we show that such a total demand does not exist. First, we have

$$\begin{aligned}\sum_{k=1}^{K}\min\{z_k,\lambda_{s,k}\}&\overset{(a)}{\leq}\sum_{k=1}^{K}\min\{z_k,\frac{C_s}{\theta}\}\\&\leq\sum_{k=1}^{K}\min\{\frac{z_k}{\theta},\frac{C_s}{\theta}\}\\&=\frac{1}{\theta}\sum_{k=1}^{K}\min\{z_k,C_s\}\overset{(b)}{<}\frac{C_c}{\theta},\end{aligned} \tag{96}$$

where (a) is due to $\lambda_{s,k}\leq C_s/\theta$ for any item k and (b) is due to the condition that the optimal offline algorithm does not make the combo purchase. According to Lemma A.2, it implies that LADTSR does not make the combo purchase, leading to the contradiction.

Case B-I): In this case, LADTSR does not make the combo purchase. According to Lemma F.1, we have

$$U_{\text{ALG}}^{k}(z)=\begin{cases}z_k, & z_k<\lambda_{s,k},\\\lambda_{s,k}+C_s, & \text{otherwise.}\end{cases} \tag{97}$$

Next, we show that for each item k, we have

$$U_{\text{ALG}}^{k}(z)\leq(1+\theta)(\min\{C_s,y_k\}+\eta_k). \tag{98}$$

We consider two cases: $y_k\geq C_s$ and $y_k<C_s$.

- If $y_k\geq C_s$, we have $\lambda_{s,k}=\theta C_s$ and $\min\{y_k,C_s\}=C_s$. We consider two subcases: $z_k<\theta C_s$ and $z_k\geq\theta C_s$.
  - If $z_k<\theta C_s$, we have $U_{\text{ALG}}^{k}(z)=z_k<\theta C_s\leq C_s$. As we have $C_s=\min\{y_k,C_s\}\leq\min\{y_k,C_s\}+\eta_k\leq(1+\theta)(\min\{y_k,C_s\}+\eta_k)$, we obtain $U_{\text{ALG}}^{k}(z)\leq(1+\theta)(\min\{C_s,y_k\}+\eta_k)$.
  - If $z_k\geq\theta C_s$, we have $U_{\text{ALG}}^{k}(z)=\theta C_s+C_s\leq(1+\theta)C_s\leq(1+\theta)(\min\{y_k,C_s\}+\eta_k)$.
- If $y_k<C_s$, we have $\lambda_{s,k}=C_s/\theta$ and $\min\{y_k,C_s\}=y_k$. Similarly, we consider two subcases: $z_k<C_s/\theta$ and $z_k\geq C_s/\theta$.

- If $z_k < C_s/\theta$, we have $U^k_{\text{ALG}}(z) = z_k$. As we have $z_k \le y_k + \eta_k = \min\{y_k, C_s\} + \eta_k \le (1+\theta)(\min\{y_k, C_s\} + \eta_k)$, we obtain $U^k_{\text{ALG}}(z) \le (1+\theta)(\min\{C_s, y_k\} + \eta_k)$.
- If $z_k \ge C_s/\theta$, we have $U^k_{\text{ALG}}(z) = C_s/\theta + C_s = (1+\theta)(C_s/\theta) \le (1+\theta)z_k \le (1+\theta)(\min\{y_k, C_s\} + \eta_k)$.

Combining all the cases, we show that Eq. (98) holds. As $OPT(z) = C_c$, we have

$$\begin{aligned}
CR(z) &\le \frac{\sum_{k=1}^{K} U^k_{\text{ALG}}(z)}{OPT(z)} \\
&\le \frac{(1+\theta)(\sum_{k=1}^{K}(\min\{y_k, C_s\} + \eta_k))}{C_c} \\
&\overset{(a)}{\le} \frac{(1+\theta)(C_c + \eta)}{C_c} \\
&= 1 + \theta + (1+\theta)\frac{\eta}{OPT(z)},
\end{aligned} \tag{99}$$

where (a) is due to Eq. (83).

Case B-II): In this case, same as Case B-II when $\lambda_c = \theta^2 C_c$ we have

$$CR(z) \le 1 + \theta + \frac{1}{1-\theta}\frac{\eta}{OPT(z)}. \tag{100}$$

Combining all the four subcases, we derive the upper bound for CR when $\lambda_c = C_c/\theta$, i.e.,

$$\begin{aligned}
CR(z) \le &\max\{1, 1+\theta\} \\
&+ \max\{\frac{1+2\theta}{1-\theta}, 1+\theta, \frac{1}{1-\theta}\}\frac{\eta}{OPT(z)}.
\end{aligned} \tag{101}$$

Clearly, we have $\frac{1+2\theta}{1-\theta} \ge \frac{1}{1-\theta} \ge 1+\theta$ for $\theta \in (0,1)$. Therefore, we have

$$CR(z) \le 1 + \theta + \frac{1+2\theta}{1-\theta}\frac{\eta}{OPT(z)}. \tag{102}$$

Finally, combing the upper bound we derive when $\lambda_c = \theta^2 C_c$ (i.e., Eq. (82)), we have

$$\begin{aligned}
CR(z) \le &\max\{1+\theta, 1+\theta+\theta^2\} \\
&+ \max\{\frac{1}{1-\theta}, \frac{1+2\theta}{1-\theta}\}\frac{\eta}{OPT(z)}.
\end{aligned} \tag{103}$$

Clearly, we have $1+\theta \le 1+\theta+\theta^2$ and $\frac{1}{1-\theta} \le \frac{1+2\theta}{1-\theta}$. Therefore, we obtain the upper bound as

$$CR(z) \le (1+\theta+\theta^2) + \frac{1+2\theta}{1-\theta}\frac{\eta}{OPT(z)}. \tag{104}$$

According to Lemma A.4, the optimal offline cost only depends on the total demand vector regardless of specific demand sequence $\mathbf{D}$. Therefore, we have

$$CR(z) \le (1+\theta+\theta^2) + \frac{1+2\theta}{1-\theta}\frac{\eta}{OPT(\mathbf{D})}. \tag{105}$$

□

G Proof of Lemma F.1

Proof. We prove the result of $U_{\text{ALG}}(z)$ in two cases respectively, i.e., A) LADTSR does not make the combo purchase and B) LADTSR makes the combo purchase.

Case A): In this case, we want to show that

$$U^k_{\text{ALG}}(z) = \begin{cases} z_k, & z_k < \lambda_{s,k}, \\ \lambda_{s,k} + C_s, & \text{otherwise.} \end{cases} \tag{106}$$

Because all the demands are integers, therefore $\psi_k(t)$ is always an integer. Therefore, $\psi_k(t) \ge \lambda_{s,k}$ is equivalent to $\psi_k(t) \ge \lceil \lambda_{s,k} \rceil$. Furthermore, we can directly apply the result in Lemma B.1. That is, the cost of LADTSR for item k is at most z_k if $z_k < \lceil \lambda_{s,k} \rceil$, otherwise it is at most $\lceil \lambda_{s,k} \rceil - 1 + C_s \le \lambda_{s,k} + C_s$. Because $z_k < \lceil \lambda_{s,k} \rceil$ is equivalent to $z_k < \lambda_{s,k}$ as z_k are integers, we have

$$U^k_{\text{ALG}}(z) = \begin{cases} z_k, & z_k < \lambda_{s,k}, \\ \lambda_s + C_s, & \text{otherwise.} \end{cases} \tag{107}$$

Case B): In this case, we want to show that

$$U_{\text{ALG}}(z) = \lambda_c + C_c + f(s_1, s_2)C_s, \tag{108}$$

where

$$f(s_1, s_2) = \min\{\frac{\lambda_c}{\theta C_s}, s_2 + \frac{\lambda_c - \theta C_s s_2}{C_s/\theta}, s_1 + s_2\}. \tag{109}$$

Similar to the proof for Lemma B.1, suppose LADTSR makes n_s single purchases before t_c (recall that t_c is the time-slot when the combo purchase is made), and we will discuss the value of n_s later. Next, we will show that

$$U_{\text{ALG}}(z) = \lambda_c + C_c + n_s C_s. \tag{110}$$

To see this, as the combo purchase cost is C_c and the single purchase cost is $n_s C_s$, we only need to show that the total rental cost is λ_c. To see this, we first show that the rental cost for each item k is at most $\min\{\psi_k(t_c - 1), \lambda_{s,k}\}$.

On the one hand, in any time-slot t, if LADTSR chooses to cover the demand by rental (with a rental cost of $a(t)$), we have $\psi_{i(t)}(t) = \psi_{i(t)}(t-1) + a(t)$, which means the associated indicative cost will also increase by $a(t)$. It directly implies that the rental cost for each item k until time-slot $t_c - 1$ cannot exceed $\psi_k(t_c - 1)$.

On the other hand, the rental cost for each item k cannot exceed $\lambda_{s,k}$ as well. We can prove this by contradiction. Soppuse in some time-slot t, LADTSR chooses to rent for item $i(t)$ and its rental cost exceeds $\lambda_{s,i(t)}$. As the rental cost is at most $\psi_{i(t)}(t)$, we have $\psi_{i(t)}(t) \ge \lambda_{s,i(t)}$, which directly implies that LADTSR will make the single purchase for $i(t)$, leading to the contradiction.

Therefore, the total rental cost is at most

$$\sum_{k=1}^{K} \min\{\psi_k(t_c - 1), \lambda_{s,k}\} \overset{(a)}{=} \psi_c(t_c - 1) \overset{(b)}{\le} \lambda_c, \tag{111}$$

where (a) is due to the definition of $\psi_c(t)$ in Eq. (14) and (b) is due to the combo purchase is not made before t_c.

Next, we analyze the value of n_s. Recall that s_1 is the number of all the items k such that $\lambda_{s,k} = C_s/\theta$ and $z_k \ge C_s/\theta$, and s_2 is the number of all the items k such that $\lambda_{s,k} = \theta C_s$ and $z_k \ge \theta C_s$. We only need to show that i) $n_s \le \frac{\lambda_c}{\theta C_s}$, ii) $n_s \le s_2 + \frac{\lambda_c - \theta C_s s_2}{C_s/\theta}$, and iii) $n_s \le s_1 + s_2$.

i) At the end of time-slot $t_c - 1$, for each item k such that LADTSR makes the single purchase for it, we have $\psi_k(t_c - 1) \geq \lambda_{s,k} \geq \theta C_s$. Therefore, we have $\lambda_c \geq \psi_c(t_c - 1) \geq \sum_{k=1}^{K} \min\{\psi_k(t_c - 1), \lambda_{s,k}\} \geq n_s \theta C_s$.

ii) Suppose among n_s items, there are n_1 items with a threshold equal to C_s/θ and n_2 items with a threshold equal to θC_s. Clearly, we have $n_1 \leq s_1$ and $n_2 \leq s_2$ because LADTSR will not make the single purchase for the item that has a total demand less than its single purchase threshold. Then similar to the proof for 1), we have $\lambda_c \geq n_1 C_s/\theta + n_2 \theta C_s$. As $n_s = n_1 + n_2$, we have

$$n_s = n_1 + n_2 \leq n_2 + \frac{\lambda_c - \theta C_s n_2}{C_s/\theta} = (1-\theta^2) n_2 + \frac{\theta \lambda_c}{C_s}. \tag{112}$$

As $1 - \theta^2 \geq 0$ and $n_2 \leq s_2$, we have

$$n_2 \leq (1-\theta^2) s_2 + \frac{\theta \lambda_c}{C_s} = s_2 + \frac{\lambda_c - \theta C_s s_2}{C_s/\theta}. \tag{113}$$

iii) Clearly, we have $n_s = n_1 + n_2 \leq s_1 + s_2$ because $n_1 \leq s_1$ and $n_2 \leq s_2$.

□

H Proof of Robustness

Proof. In this section, we prove the other upper bound $1 + \theta^{-1} + \theta^{-3}$. Our goal is still to derive an upper bound of CR over all the demand sequences under any prediction quality.

The analysis follows a similar pattern to that of Theorem 3.1. Specifically, we proceed with the proof in two steps:

1a) We establish the upper bound of CR for all demand sequences with the same total demand.

1b) We define a standard total demand (see Definition B.1) to simplify the upper bound we obtained in Step 1a) (see Lemma H.1).

1c) We determine the upper bound of CR over all the standard total demand, regardless of the specific demand sequence, which is $1 + \theta^{-1} + \theta^{-3}$.

Step 1a): We can directly apply the result of Lemma F.1 in the proof of consistency.

Step 1b): Given the complexity of $U_{\text{ALG}}(z)$, we establish the standard total demand for LADTSR. However, the difficulty here is that the single purchase threshold can be two values: θC_s or C_s/θ. Therefore, we need to extend our definition of standard total demand for RDTSR. First, we provide the extended definition as follows.

Definition H.1. *Given any prediction y and its associated $\lambda_{s,k}$ (for item $k = 1, \ldots, K$), a total demand z is said to be the standard total demand $z^y_{\text{std}}(m_1, m_2, n_2, x)$ if it satisfies:*

i) for items with a threshold equal to C_s/θ:
- *m_1 items have a total demand of at least $\lceil C_s/\theta \rceil$;*

ii) for items with a threshold equal to θC_s:
- *m_2 items have a total demand of at least C_s,*
- *n_2 items have a total demand equal to $\lceil \theta C_s \rceil$;*

iii) among all the remaining items, each item k has a total demand within the interval $[0, \min\{C_s, \lambda_{s,k}\})$. Let x represent the sum of their total demands;

here m_1, m_2, n_2, x can be any non-negative integers.

Algorithm 4: Construct $z^y_{\text{std}}(m_1, m_2, n_2, x)$ from z and y

```
Input  : $z$, $y$
Output: $z'$
$z' \leftarrow z$;
Calculate $\lambda_{s,k}$ for each item $k$ according to $y$;
while there exists item $k$ s.t. $\lambda_{s,k} = C_s/\theta$ and $C_s \leq z'_k < \lceil C_s/\theta \rceil$ do
    $z'_k \leftarrow \lceil C_s/\theta \rceil$;
end
while there exists item $k$ s.t. $\lambda_{s,k} = \theta C_s$ and $\lceil \theta C_s \rceil \leq z'_k < C_s$ do
    $z'_k \leftarrow \lceil \theta C_s \rceil$;
end
```

Next, we present an approach to construct a standard total demand given any total demand z, which is presented in Algorithm 4. Specifically, we still start with a total demand $z' = \{z'_1, \ldots, z'_K\}$ equal to z. Then we modify z' as follows.

i) If there exists an item k such that $\lambda_{s,k} = C_s/\theta$ and $C_s \leq z'_k < \lceil C_s/\theta \rceil$, set z'_k as $\lceil C_s/\theta \rceil$.

ii) If there exists an item k such that $\lambda_{s,k} = \theta C_s$ and $\lceil \theta C_s \rceil \leq z'_k < C_s$, set z'_k as $\lceil \theta C_s \rceil$.

iii) All the remaining items remain the same.

We will show that the constructed total demand z' follows Definition H.1 and can bound $U_{\text{CR}}(z)$ in Lemma H.1.

Lemma H.1. *Given any total demand z, the total demand z' constructed by Algorithm 4 is a standard total demand. Furthermore, we have*

$$U_{\text{CR}}(z) \leq U_{\text{CR}}(z'). \tag{114}$$

We prove Lemma H.1 in Appendix I. Hence, with any prediction y and any total demand z, Algorithm 4 allows us to find a standard total demand to bound $U_{\text{CR}}(z)$.

Step 1c): Given any total demand z, we use $z^y_{\text{std}}(m_1, m_2, n_2, x)$ to denote the constructed standard total demand. To obtain $U_{\text{CR}}(z^y_{\text{std}}(m_1, m_2, n_2, x))$, we consider two cases: A) LADTSR makes the combo purchase on $z^y_{\text{std}}(m_1, m_2, n_2, x)$; B) LADTSR does not make the combo purchase on $z^y_{\text{std}}(m_1, m_2, n_2, x)$.

Case A): We first analyze $OPT(z^y_{\text{std}}(m_1, m_2, n_2, x))$. According to Definition H.1, there are $m_1 + m_2$ items that have a total demand of at least C_s; for the remaining items, their total demand is less than C_s and the sum of their total demand is $n_2 \lceil \theta C_s \rceil + x$. Same as z_k, we use

$z^y_{\mathrm{std}}(m_1, m_2, n_2, x)_k$ to represent the total demand of item k in $z^y_{\mathrm{std}}(m_1, m_2, n_2, x)$. Then, we can rewrite Eq. (18) as

$$\begin{aligned} &OPT(z^y_{\mathrm{std}}(m_1, m_2, n_2, x)) \\ &\quad = \min\{C_c, \sum_{k=1}^{K} \min\{C_s, z^y_{\mathrm{std}}(m_1, m_2, n_2, x)_k\}\} \\ &\quad = \min\{C_c, (m_1 + m_2)C_s + n_2 \lceil \theta C_s \rceil + x\}. \end{aligned} \tag{115}$$

Next, we analyze $U_{\mathrm{ALG}}(z^y_{\mathrm{std}}(m_1, m_2, n_2, x))$. For items with a threshold C_s/θ, there are m_1 items having a total demand of at least $\lceil C_s/\theta \rceil \geq C_s/\theta$ (i.e., $s_1 = m_1$). Similarly, for items with a threshold θC_s, there are m_2 items having a total demand of at least $C_s \geq \theta C_s$ and n_2 items having a total demand of at least $\lceil \theta C_s \rceil \geq \theta C_s$ (i.e., $s_2 = m_2 + n_2$). According to Lemma F.1, we can rewrite $U_{\mathrm{ALG}}(z^y_{\mathrm{std}}(m_1, m_2, n_2, x))$ as

$$U_{\mathrm{ALG}}(z^y_{\mathrm{std}}(m_1, m_2, n_2, x)) = \lambda_c + C_c + f(m_1, m_2 + n_2)C_s. \tag{116}$$

Therefore, we can express $U_{\mathrm{CR}}(z^y_{\mathrm{std}}(m_1, m_2, n_2, x))$ as

$$\begin{aligned} &U_{\mathrm{CR}}(z^y_{\mathrm{std}}(m_1, m_2, n_2, x)) \\ &\quad = \frac{\lambda_c + C_c + f(m_1, m_2 + n_2)C_s}{\min\{C_c, (m_1 + m_2)C_s + n_2 \lceil \theta C_s \rceil + x\}}. \end{aligned} \tag{117}$$

As LADTSR makes the combo purchase in this case, according to Lemma A.2, we have

$$\sum_{k=1}^{K} \min\{z^y_{\mathrm{std}}(m_1, m_2, n_2, x)_k, \lambda_{s,k}\} \geq \lambda_c. \tag{118}$$

According to Definition H.1, there are m_1 items with a total demand exceeding its threshold C_s/θ; there are $m_2 + n_2$ items with a total demand exceeding its threshold θC_s; all the remaining items having a total demand less than its threshold and the sum of their total demands is x. Therefore, Eq. (118) becomes

$$m_1 \frac{C_s}{\theta} + (m_2 + n_2)\theta C_s + x \geq \lambda_c. \tag{119}$$

Next, we claim that we only need to consider the case when $m_2 = 0$. To see this, consider another standard total demand $z'^y_{\mathrm{std}}(m'_1, m'_2, n'_2, x')$ such that $m'_1 = m_1$, $m'_2 = 0$, $n'_2 = m_2 + n_2$, and $x' = x$. Note that the prediction for $z'^y_{\mathrm{std}}(m'_1, m'_2, n'_2, x')$ remains the same. First, we have

$$\begin{aligned} &\sum_{k=1}^{K} \min\{z'^y_{\mathrm{std}}(m'_1, m'_2, n'_2, x')_k, \lambda_{s,k}\} \\ &\quad \overset{(a)}{=} m'_1 \frac{C_s}{\theta} + (m'_2 + n'_2)\theta C_s + x' \\ &\quad = m_1 \frac{C_s}{\theta} + (m_2 + n_2)\theta C_s + x \\ &\quad \geq \lambda_c, \end{aligned} \tag{120}$$

where (a) is due to the same reason for Eq. (119) (replace m_1, m_2, n_2, x by m'_1, m'_2, n'_2, x').

Eq. (120) implies that LADTSR will make the combo purchase on $z'^y_{\mathrm{std}}(m'_1, m'_2, n'_2, x')$ according to Lemma A.2. Therefore, according to Lemma F.1, we have

$$\begin{aligned} &U_{\mathrm{CR}}(z'^y_{\mathrm{std}}(m'_1, m'_2, n'_2, x')) \\ &\quad = \frac{\lambda_c + C_c + f(m'_1, m'_2 + n'_2)C_s}{\min\{C_c, (m'_1 + m'_2)C_s + n'_2 \lceil \theta C_s \rceil + x'\}}, \end{aligned} \tag{121}$$

Then, we can show that

$$U_{\mathrm{CR}}(z'^y_{\mathrm{std}}(m'_1, m'_2, n'_2, x')) \geq U_{\mathrm{CR}}(z^y_{\mathrm{std}}(m_1, m_2, n_2, x)). \tag{122}$$

To see this, by comparing the denominator, we have

$$\begin{aligned} &\min\{C_c, (m'_1 + m'_2)C_s + n'_2 \lceil \theta C_s \rceil + x'\} \\ &\quad = \min\{C_c, (m_1 + 0)C_s + (m_2 + n_2) \lceil \theta C_s \rceil + x\} \\ &\quad \leq \min\{C_c, (m_1 + m_2)C_s + n_2 \lceil \theta C_s \rceil + x\}. \end{aligned} \tag{123}$$

For the numerator, as $m'_1 = m_1$ and $m'_2 + n'_2 = m_2 + n_2$, we have $f(m'_1, m'_2 + n'_2) = f(m_1, m_2 + n_2)$. Therefore, the numerator of $U_{\mathrm{CR}}(z'^y_{\mathrm{std}}(m'_1, m'_2, n'_2, x'))$ is identical to that of $U_{\mathrm{CR}}(z^y_{\mathrm{std}}(m_1, m_2, n_2, x))$, while $U_{\mathrm{CR}}(z'^y_{\mathrm{std}}(m'_1, m'_2, n'_2, x'))$ has a smaller denominator. So we conclude that Eq. (122) holds.

Therefore, given any $z^y_{\mathrm{std}}(m_1, m_2, n_2, x)$, there exists another standard total demand $z'^y_{\mathrm{std}}(m'_1, 0, n'_2, x')$ and we have

$$\begin{aligned} &U_{\mathrm{CR}}(z^y_{\mathrm{std}}(m_1, m_2, n_2, x)) \\ &\quad \leq U_{\mathrm{CR}}(z'^y_{\mathrm{std}}(m'_1, 0, n'_2, x')) \\ &\quad = \frac{\lambda_c + C_c + f(m'_1, n'_2)C_s}{\min\{C_c, m'_1 C_s + n'_2 \lceil \theta C_s \rceil + x'\}}, \end{aligned} \tag{124}$$

and Eq. (119) becomes

$$m'_1 \frac{C_s}{\theta} + n'_2 \theta C_s + x' \geq \lambda_c. \tag{125}$$

Next, relax m'_1, n'_2, x' to be non-negative real numbers and let $g(m'_1, n'_2, x')$ denote the RHS of Eq. (124), i.e.,

$$g(m'_1, n'_2, x') = \frac{\lambda_c + C_c + f(s'_1, s'_2)C_s}{\min\{C_c, m'_1 C_s + n'_2 \lceil \theta C_s \rceil + x'\}}. \tag{126}$$

If we can find an upper bound holds for $g(m'_1, n'_2, x')$ with the constraint Eq. (125), it is also an upper bound for $U_{\mathrm{CR}}(z'^y_{\mathrm{std}}(m'_1, 0, n'_2, x'))$. That is,

$$\begin{aligned} &\forall\, m'_1, n'_2, x' \in \mathbb{N}, U_{\mathrm{CR}}(z'^y_{\mathrm{std}}(m'_1, 0, n'_2, x')) \\ &\quad \leq \max_{m'_1, n'_2, x' \in \mathbb{N}} U_{\mathrm{CR}}(z'^y_{\mathrm{std}}(m'_1, 0, n'_2, x')) \\ &\quad \leq \max_{m'_1, n'_2, x' \in \mathbb{R}^+ \cup \{0\}} g(m'_1, n'_2, x'). \end{aligned} \tag{127}$$

Then, we claim that to find the upper bound for the function $g(m'_1, n'_2, x')$, we only need to focus on the case when $m'_1 C_s/\theta + n'_2 \theta C_s + x' = \lambda_c$. To see this, for all the possible m'_1, n'_2, and x' such that $m'_1 C_s/\theta + n'_2 \theta C_s + x' > \lambda_c$, we show that there exists another pair of $\tilde{m}_1$, $\tilde{n}_2$, and $\tilde{x}$ such that $\tilde{m}_1 C_s/\theta + \tilde{n}_2 \theta C_s + \tilde{x} = \lambda_c$ and $g(\tilde{m}_1, \tilde{n}_2, \tilde{x}) \geq g(m'_1, n'_2, x')$. Specifically, we consider two cases: if $x' = 0$ and if $x' > 0$.

i) If $x' = 0$, we consider two subcases: if $m_1' = 0$ and if $m_1' > 0$.

- If $m_1' = 0$, then Eq. (126) becomes
$$g(m_1', n_2', x') = \frac{\lambda_c + C_c + f(0, n_2')C_s}{\min\{C_c, n_2' \lceil \theta C_s \rceil\}}, \tag{128}$$
where
$$f(0, n_2') = \min\{\frac{\lambda_c}{\theta C_s}, n_2' + \frac{\lambda_c - \theta C_s n_2'}{C_s/\theta}, n_2'\}. \tag{129}$$
Also, we have $n_2'\theta C_s > \lambda_c$. Therefore, for $f(0, n_2')$ we have
$$\begin{aligned} n_2' + \frac{\lambda_c - \theta C_s n_2'}{C_s/\theta} &= (1-\theta^2)n_2' + \frac{\theta\lambda_c}{C_s} \\ &> (1-\theta^2)\frac{\lambda_c}{\theta C_s} + \frac{\theta\lambda_c}{C_s} \\ &= \frac{\lambda_c}{\theta C_s}. \end{aligned} \tag{130}$$
As $n_2' > \frac{\lambda_c}{\theta C_s}$, then we have $f(0, n_2') = \frac{\lambda_c}{\theta C_s}$. Therefore, Eq. (128) becomes
$$g(m_1', n_2', x') = \frac{\lambda_c + C_c + \frac{\lambda_c}{\theta C_s}C_s}{\min\{C_c, n_2' \lceil \theta C_s \rceil\}}. \tag{131}$$
Let $\tilde{n}_2 = \frac{\lambda_c}{\theta C_s} < n_2'$ and $\tilde{m}_1 = \tilde{x} = 0$, then we have
$$\begin{aligned} g(\tilde{m}_1, \tilde{n}_2, \tilde{x}) &= \frac{\lambda_c + C_c + \frac{\lambda_c}{\theta C_s}C_s}{\min\{C_c, \tilde{n}_2 \lceil \theta C_s \rceil\}} \\ &\geq \frac{\lambda_c + C_c + \frac{\lambda_c}{\theta C_s}C_s}{\min\{C_c, n_2' \lceil \theta C_s \rceil\}} \\ &= g(m_1', n_2', x'). \end{aligned} \tag{132}$$
- If $m_1' > 0$, because $m_1'C_s/\theta + n_2'\theta C_s > \lambda_c$, then for $f(m_1', n_2')$ we have
$$n_2' + \frac{\lambda_c - \theta C_s n_2'}{C_s/\theta} < n_2' + \frac{m_1'C_s/\theta}{C_s/\theta} = n_2' + m_1'. \tag{133}$$
Therefore, Eq. (126) becomes
$$g(m_1', n_2', x') = \frac{\lambda_c + C_c + f(m_1', n_2')C_s}{\min\{C_c, m_1'C_s + n_2' \lceil \theta C_s \rceil\}}, \tag{134}$$
where
$$f(m_1', n_2') = \min\{\frac{\lambda_c}{\theta C_s}, n_2' + \frac{\lambda_c - \theta C_s n_2'}{C_s/\theta}\}. \tag{135}$$
If $n_2'\theta C_s \leq \lambda_c$, let $\tilde{m}_1 C_s/\theta + n_2'\theta C_s = \lambda_c$ ($\tilde{m}_1 \leq m_1'$), $\tilde{n}_2 = n_2'$, and $\tilde{x} = 0$, we have
$$\begin{aligned} g(\tilde{m}_1, \tilde{n}_2, \tilde{x}) &= \frac{\lambda_c + C_c + f(\tilde{m}_1, \tilde{n}_2)C_s}{\min\{C_c, \tilde{m}_1 C_s + \tilde{n}_2 \lceil \theta C_s \rceil\}} \\ &= \frac{\lambda_c + C_c + f(m_1', n_2')C_s}{\min\{C_c, \tilde{m}_1 C_s + n_2' \lceil \theta C_s \rceil\}} \\ &\geq \frac{\lambda_c + C_c + f(m_1', n_2')C_s}{\min\{C_c, m_1' C_s + n_2' \lceil \theta C_s \rceil\}} \\ &= g(m_1', n_2', x'). \end{aligned} \tag{136}$$
If $n_2'\theta C_s > \lambda_c$, then set $\tilde{m}_1 = 0$, $\tilde{n}_2 = \frac{\lambda_c}{\theta C_s}$ and $\tilde{x} = 0$, we have
$$\begin{aligned} g(\tilde{m}_1, \tilde{n}_2, \tilde{x}) &= g(0, \tilde{n}_2, 0) \\ &\overset{(a)}{\geq} g(0, n_2', 0) \\ &\overset{(b)}{\geq} g(m_1', n_2', 0) \\ &= g(0, \tilde{n}_2, x'), \end{aligned} \tag{137}$$
where (a) is shown in the case when $m_1' = x' = 0$ and (b) is shown in Eq. (136).

ii) When $x' > 0$, if $m_1'C_s/\theta + n_2'\theta C_s \leq \lambda_c$, let $m_1'C_s/\theta + n_2'\theta C_s + \tilde{x} = \lambda_c$ ($\tilde{x} < x'$), $\tilde{m}_1 = m_1'$, and $\tilde{n}_2 = n_2'$. Then, from Eq. (126) we have
$$\begin{aligned} g(\tilde{m}_1, \tilde{n}_2, \tilde{x}) &= \frac{\lambda_c + C_c + f(\tilde{m}_1, \tilde{n}_2)C_s}{\min\{C_c, \tilde{m}_1 C_s + \tilde{n}_2 \lceil \theta C_s \rceil + \tilde{x}\}} \\ &= \frac{\lambda_c + C_c + f(m_1', n_2')C_s}{\min\{C_c, m_1' C_s + n_2' \lceil \theta C_s \rceil + \tilde{x}\}} \\ &\geq \frac{\lambda_c + C_c + f(m_1', n_2')C_s}{\min\{C_c, m_1' C_s + n_2' \lceil \theta C_s \rceil + x'\}} \\ &= g(m_1', n_2', x'). \end{aligned} \tag{138}$$
If $m_1'C_s/\theta + n_2'\theta C_s > \lambda_c$, let $\tilde{x} = 0$, similarly we have
$$g(m_1', n_2', \tilde{x}) \leq g(m_1', n_2', x'). \tag{139}$$
And we already show how to bound $g(m_1', n_2', 0)$ in the case when $x' = 0$.

Therefore, we can conclude that if $m_1'C_s/\theta + n_2'\theta C_s + x' > \lambda_c$, there exists non-negative real variables $\tilde{m}_1, \tilde{n}_2, \tilde{x}$ such that $\tilde{m}_1 C_s/\theta + \tilde{n}_2\theta C_s + \tilde{x} = \lambda_c$ and $g(\tilde{m}_1, \tilde{n}_2, \tilde{x}) \geq g(m_1', n_2', x')$.

Next, we focus on $g(\tilde{m}_1, \tilde{n}_2, \tilde{x})$ with $\tilde{m}_1 C_s/\theta + \tilde{n}_2\theta C_s + \tilde{x} = \lambda_c$. Substitute $\tilde{n}_2\theta C_s + \tilde{x} = \lambda_c - \tilde{m}_1 C_s/\theta$ into $g(\tilde{m}_1, \tilde{n}_2, \tilde{x})$, we have
$$\begin{aligned} g(\tilde{m}_1, \tilde{n}_2, \tilde{x}) &\leq \frac{\lambda_c + C_c + f(\tilde{m}_1, \tilde{n}_2)C_s}{\min\{C_c, \tilde{m}_1 C_s + \tilde{n}_2 \lceil \theta C_s \rceil + \tilde{x}\}} \\ &\leq \frac{\lambda_c + C_c + f(\tilde{m}_1, \tilde{n}_2)C_s}{\min\{C_c, \tilde{m}_1 C_s + \tilde{n}_2 \theta C_s + \tilde{x}\}} \\ &= \frac{\lambda_c + C_c + f(\tilde{m}_1, \tilde{n}_2)C_s}{\min\{C_c, \lambda_c - \tilde{m}_1(\frac{C_s}{\theta} - C_s)\}}, \end{aligned} \tag{140}$$
where
$$f(\tilde{m}_1, \tilde{n}_2) = \min\{\frac{\lambda_c}{\theta C_s}, \tilde{n}_2 + \frac{\lambda_c - \theta C_s \tilde{n}_2}{C_s/\theta}, \tilde{m}_1 + \tilde{n}_2\}. \tag{141}$$
As $f(\tilde{m}_1, \tilde{n}_2) \leq \tilde{m}_1 + \tilde{n}_2$, we have
$$g(\tilde{m}_1, \tilde{n}_2, \tilde{x}) \leq \frac{\lambda_c + C_c + (\tilde{m}_1 + \tilde{n}_2)C_s}{\min\{C_c, \lambda_c - \tilde{m}_1(\frac{C_s}{\theta} - C_s)\}}. \tag{142}$$
Also, because $\tilde{m}_1\frac{C_s}{\theta} + \tilde{n}_2\theta C_s + \tilde{x} = \lambda_c$ and $\tilde{x} \geq 0$, we have
$$\tilde{n}_2 = \frac{\lambda_c - \tilde{x} - \tilde{m}_1 C_s/\theta}{\theta C_s} \leq \frac{\lambda_c - \tilde{m}_1 C_s/\theta}{\theta C_s}. \tag{143}$$

Therefore, Eq. (142) becomes

$$g(\tilde{m}_1, \tilde{n}_2, \tilde{x}) \leq \frac{\lambda_c + C_c + (\tilde{m}_1 + \frac{\lambda_c - \tilde{m}_1 C_s/\theta}{\theta C_s})C_s}{\min\{C_c, \lambda_c - \tilde{m}_1(\frac{C_s}{\theta} - C_s)\}}. \quad (144)$$

Next, to analyze the upper bound of CR holds for any $\tilde{m}_1$, we consider two cases: if $\lambda_c = \theta^2 C_c$ and if $\lambda_c = C_c/\theta$.

If $\lambda_c = \theta^2 C_c$, then Eq. (144) becomes

$$\begin{aligned} & g(\tilde{m}_1, \tilde{n}_2, \tilde{x}) \\ & \quad \leq \frac{\theta^2 C_c + C_c + (\tilde{m}_1 + \frac{\theta^2 C_c - \tilde{m}_1 C_s/\theta}{\theta C_s})C_s}{\min\{C_c, \theta^2 C_c - \tilde{m}_1(\frac{C_s}{\theta} - C_s)\}} \\ & \quad = \frac{\theta^2 C_c + C_c + (\tilde{m}_1 + \frac{\theta^2 C_c - \tilde{m}_1 C_s/\theta}{\theta C_s})C_s}{\min\{C_c, \theta^2 C_c - \tilde{m}_1 C_s(\frac{1}{\theta} - 1)\}}. \end{aligned} \quad (145)$$

As $1/\theta - 1 \geq 0$, for the denominator we have

$$\theta^2 C_c - \tilde{m}_1 C_s(\frac{1}{\theta} - 1) \leq \theta^2 C_c \leq C_c. \quad (146)$$

Therefore, Eq. (145) becomes

$$\begin{aligned} & g(\tilde{m}_1, \tilde{n}_2, \tilde{x}) \\ & \quad \leq \frac{\theta^2 C_c + C_c + (\tilde{m}_1 + \frac{\theta^2 C_c - \tilde{m}_1 C_s/\theta}{\theta C_s})C_s}{\theta^2 C_c - \tilde{m}_1(\frac{C_s}{\theta} - C_s)} \\ & \quad = \frac{(1 + \theta + \theta^2)C_c + \tilde{m}_1 C_s(1 - \frac{1}{\theta^2})}{\theta^2 C_c + \tilde{m}_1 C_s(1 - \frac{1}{\theta})}. \end{aligned} \quad (147)$$

As $\tilde{m}_1$, $\tilde{n}_2$,and $\tilde{x}$ are non-negative and $\tilde{m}_1 C_s/\theta + \tilde{n}_2 \theta C_s + \tilde{x} = \theta^2 C_c$, we have

$$0 \leq \tilde{m}_1 = \frac{\theta^2 C_c - \tilde{x} - \theta C_s \tilde{n}_2}{C_s/\theta} \leq \frac{\theta^3 C_c}{C_s}. \quad (148)$$

Substitute $0 \leq \tilde{m}_1 \leq \frac{\theta^3 C_c}{C_s}$ into the denominator of Eq. (147) we have

$$0 < \theta^3 C_c \leq \theta^2 C_c + \tilde{m}_1 C_s(1 - \frac{1}{\theta}) \leq \theta^2 C_c. \quad (149)$$

Therefore, The RHS function in Eq. (147) is defined over $\tilde{m}_1 \in [0, \frac{\theta^3 C_c}{C_s}]$. On the other hand, by taking derivative of the RHS function to $\tilde{m}_1$, we have

$$\begin{aligned} & \frac{d}{d\tilde{m}_1}\left(\frac{(1 + \theta + \theta^2)C_c + \tilde{m}_1 C_s(1 - \frac{1}{\theta^2})}{\theta^2 C_c + \tilde{m}_1 C_s(1 - \frac{1}{\theta})}\right) \\ & \quad = \frac{C_s(1 - \frac{1}{\theta^2})(\theta^2 C_c + \tilde{m}_1 C_s(1 - \frac{1}{\theta}))}{(\theta^2 C_c + \tilde{m}_1 C_s(1 - \frac{1}{\theta}))^2} \\ & \qquad - \frac{C_s(1 - \frac{1}{\theta})((1 + \theta + \theta^2)C_c + \tilde{m}_1 C_s(1 - \frac{1}{\theta^2}))}{(\theta^2 C_c + \tilde{m}_1 C_s(1 - \frac{1}{\theta}))^2} \\ & \quad = \frac{C_s(1 - \frac{1}{\theta^2})\theta^2 C_c + C_s(1 - \frac{1}{\theta})(1 + \theta + \theta^2)C_c}{(\theta^2 C_c + \tilde{m}_1 C_s(1 - \frac{1}{\theta}))^2} \end{aligned} \quad (150)$$

As the denominator is non-negative and the numerator is independent of $\tilde{m}_1$, the RHS function in Eq. (147) is monotone to $\tilde{m}_1$. Therefore, it takes the maximum only when $\tilde{m}_1 = 0$ or $\tilde{m}_1 = \frac{\theta^3 C_c}{C_s}$. When $\tilde{m}_1 = 0$, we have

$$\begin{aligned} \frac{(1 + \theta + \theta^2)C_c + \tilde{m}_1 C_s(1 - \frac{1}{\theta^2})}{\theta^2 C_c + \tilde{m}_1 C_s(1 - \frac{1}{\theta})} &= \frac{(1 + \theta + \theta^2)C_c}{\theta^2 C_c} \\ &= 1 + \frac{1}{\theta} + \frac{1}{\theta^2}. \end{aligned} \quad (151)$$

When $\tilde{m}_1 = \frac{\theta^3 C_c}{C_s}$, we have

$$\begin{aligned} & \frac{(1 + \theta + \theta^2)C_c + \tilde{m}_1 C_s(1 - \frac{1}{\theta^2})}{\theta^2 C_c + \tilde{m}_1 C_s(1 - \frac{1}{\theta})} \\ & \quad = \frac{(1 + \theta + \theta^2)C_c + \frac{\theta^3 C_c}{C_s} C_s(1 - \frac{1}{\theta^2})}{\theta^2 C_c + \frac{\theta^3 C_c}{C_s} C_s(1 - \frac{1}{\theta})} \\ & \quad = \frac{(1 + \theta + \theta^2 + \theta^3 - \theta)C_c}{(\theta^2 + \theta^3 - \theta^2)C_c} \\ & \quad = \frac{1 + \theta + \theta^3}{\theta^3} \\ & \quad = 1 + \frac{1}{\theta} + \frac{1}{\theta^3} \geq 1 + \frac{1}{\theta} + \frac{1}{\theta^2}. \end{aligned} \quad (152)$$

Therefore, combining Eqs. (124) and (127) we have

$$U_{\mathrm{CR}}(z^y_{\mathrm{std}}(m_1, n_1, n_2, x)) \leq 1 + \frac{1}{\theta} + \frac{1}{\theta^3}. \quad (153)$$

Next, we turn to $\lambda_c = C_c/\theta$, then Eq. (144) becomes

$$\begin{aligned} & g(\tilde{m}_1, \tilde{n}_2, \tilde{x}) \\ & \quad \leq \frac{C_c/\theta + C_c + (\tilde{m}_1 + \frac{C_c/\theta - \tilde{m}_1 C_s/\theta}{\theta C_s})C_s}{\min\{C_c, C_c/\theta - \tilde{m}_1 C_s(\frac{1}{\theta} - 1)\}}. \end{aligned} \quad (154)$$

Because $\tilde{m}_1$, $\tilde{n}_2$, and $\tilde{x}$ are non-negative and $\tilde{m}_1 C_s/\theta + \tilde{n}_2 \theta C_s + \tilde{x} = C_c/\theta$, we have

$$0 \leq \tilde{m}_1 = \frac{C_c/\theta - \tilde{n}_2 \theta C_s - \tilde{x}}{C_s/\theta} \leq \frac{C_c}{C_s}. \quad (155)$$

Therefore, for the denominator in Eq. (154) we have

$$\frac{C_c}{\theta} - \tilde{m}_1 C_s(\frac{1}{\theta} - 1) \geq \frac{C_c}{\theta} - \frac{C_c}{C_s} C_s(\frac{1}{\theta} - 1) \geq C_c. \quad (156)$$

Therefore, Eq. (154) becomes

$$\begin{aligned} & g(\tilde{m}_1, \tilde{n}_2, \tilde{x}) \\ & \quad \leq \frac{C_c/\theta + C_c + (\tilde{m}_1 + \frac{C_c/\theta - \tilde{m}_1 C_s/\theta}{\theta C_s})C_s}{C_c} \\ & \quad = (1 + \frac{1}{\theta} + \frac{1}{\theta^2}) + \frac{C_s(1 - \frac{1}{\theta^2})}{C_c}\tilde{m}_1 \\ & \quad \overset{(a)}{\leq} 1 + \frac{1}{\theta} + \frac{1}{\theta^2} \leq 1 + \frac{1}{\theta} + \frac{1}{\theta^3}, \end{aligned} \quad (157)$$

where (a) is due to the term $\frac{C_s(1-\theta^{-2})}{C_c}\tilde{m}_1 \leq 0$. Therefore, combining Eqs. (124) and (127) we have

$$U_{\mathrm{CR}}(z^y_{\mathrm{std}}(m_1, n_1, n_2, x)) \leq 1 + \frac{1}{\theta} + \frac{1}{\theta^3}. \quad (158)$$

Now, we conclude that the CR of LADTSR is upper bounded by $1+\theta^{-1}+\theta^{-3}$ in Case A.

Case B): In this case, LADTSR does not make the combo purchase. We consider two subcases: I) If the optimal offline algorithm does not make the combo purchase. II) If it makes the combo purchase.

Case B-I): We show that the CR upper bound $1+\theta^{-1}+\theta^{-3}$ holds for all the total demand directly without relying on the standard total demand. In this case, the problem degenerates into the ski-rental problem (i.e., z_k is the ski day for item k and C_s is the purchase cost). Therefore, we can directly apply Theorem 2.2 in their work, i.e., for each item k, the cost of LADTSR for item k (denoted by $ALG_k(z)$) is at most $(1+\theta^{-1})OPT_k(z)$, where $OPT_k(z)$ is the cost of the optimal offline algorithm for item k. As the total cost of both algorithms in Case B-I is the sum of their cost for each item, we have

$$\begin{aligned} CR(z) &= \frac{\sum_{k=1}^{K} ALG_k(z)}{\sum_{k=1}^{K} OPT_k(z)} \\ &\le (1+\theta^{-1})\frac{\sum_{k=1}^{K} OPT_k(z)}{\sum_{k=1}^{K} OPT_k(z)} \\ &= 1+\frac{1}{\theta} \le 1+\frac{1}{\theta}+\frac{1}{\theta^3}. \end{aligned} \tag{159}$$

Case B-II): We can show that this case is dominated by Case A. First, we have $OPT(z^{y}_{\text{std}}(m_1,m_2,n_2,x)) = C_c$. For the cost of LADTSR, rewrite Eq. (69) we have

$$\begin{aligned} &U_{\text{ALG}}(z^{y}_{\text{std}}(m_1,m_2,n_2,x)) \\ &= m_1(\frac{C_s}{\theta}+C_s)+(m_2+n_2)(\theta C_s+C_s)+x. \end{aligned} \tag{160}$$

Therefore, for $U_{CR}(z^{y}_{\text{std}}(m_1,m_2,n_2,x))$ we have

$$\begin{aligned} &U_{CR}(z^{y}_{\text{std}}(m_1,m_2,n_2,x)) \\ &= \frac{m_1(\frac{C_s}{\theta}+C_s)+(m_2+n_2)(\theta C_s+C_s)+x}{C_c}. \end{aligned} \tag{161}$$

As LADTSR does not make the combo purchase, according to Lemma A.2, we have

$$\begin{aligned} &\sum_{k=1}^{K}\min\{z^{y}_{\text{std}}(m_1,m_2,n_2,x)_k,\lambda_{s,k}\} \\ &= m_1\frac{C_s}{\theta}+(m_2+n_2)\theta C_s+x \\ &< \lambda_c. \end{aligned} \tag{162}$$

Then Eq. (161) becomes

$$U_{CR}(z^{y}_{\text{std}}(m_1,m_2,n_2,x)) < \frac{\lambda_c+(m_1+m_2+n_2)C_s}{C_c}. \tag{163}$$

Then we can find another standard total demand $z'^{y'}_{\text{std}}(m'_1,m'_2,n'_2,x')$ such that

$$\begin{aligned} &m'_1 \ge m_1, m'_2 = m_2, n'_2 = n_2, x' = x, \\ &m'_1\frac{C_s}{\theta}+(m'_2+n'_2)\theta C_s + x' \ge \lambda_c. \end{aligned} \tag{164}$$

Note that $z'^{y'}_{\text{std}}(m'_1,m'_2,n'_2,x')$ has a different prediction y' as it has more items compared to $z^{y}_{\text{std}}(m_1,m_2,n_2,x)$. For these extra items, we can assume their predicted total demand is 0 because the single purchase threshold for these items is C_s/θ, which implies their predicted total demand is less than C_s. For any other item, we assume the predicted total demand is the same as y. Let K' be the number of items in $z'^{y'}_{\text{std}}(m'_1,m'_2,n'_2,x')$. Then, we have

$$\sum_{k=1}^{K'}\min\{y'_k,C_s\} = \sum_{k=1}^{K}\min\{y_k,C_s\}. \tag{165}$$

According to Lines 8-12 of LADTSR, it means LADTSR has the same combo purchase threshold on both y and y'. According to the construction (i.e., Eq. (164)) and Lemma A.2, LADTSR will make the combo purchase on $z'^{y'}_{\text{std}}(m'_1,m'_2,n'_2,x')$.

Therefore, we have

$$\begin{aligned} &U_{CR}(z'^{y'}_{\text{std}}(m'_1,m'_2,n'_2,x')) \\ &= \frac{\lambda_c+C_c+f(m'_1,m'_2+n'_2)C_s}{C_c}. \end{aligned} \tag{166}$$

Next we show that $m_1+m_2+n_2 \le f(m'_1,m'_2+n'_2)$.

i) First, from Eq. (162) we have

$$\begin{aligned} (m_1+m_2+n_2)\theta C_s &\le m_1\frac{C_s}{\theta}+(m_2+n_2)\theta C_s+x \\ &< \lambda_c. \end{aligned} \tag{167}$$

Therefore, we have $m_1+n_1+m_2 < \frac{\lambda_c}{\theta C_s}$.

ii) Second, as $m'_1 \ge m_1$, $m'_2 = m_2$, and $n'_2 = n_2$, we have

$$m_1+m_2+n_2 \le m'_1+m'_2+n'_2. \tag{168}$$

iii) Finally, from Eq. (162) we have

$$m_1 < \frac{\lambda_c-\theta C_s(m_2+n_2)}{C_s/\theta}. \tag{169}$$

Therefore, we have

$$\begin{aligned} m_1+m_2+n_2 &\le m_2+n_2+\frac{\lambda_c-\theta C_s(m_2+n_2)}{C_s/\theta} \\ &= m'_2+n'_2+\frac{\lambda_c-\theta C_s(m'_2+n'_2)}{C_s/\theta}. \end{aligned} \tag{170}$$

Therefore, we conclude that $m_1+m_2+n_2 \le f(m'_1,m'_2+n'_2)$. Furthermore, we have $U_{CR}(z^{y}_{\text{std}}(m_1,m_2,n_2,x)) \le U_{CR}(z'^{y'}_{\text{std}}(m'_1,m'_2,n'_2,x'))$. On the other hand, we have $OPT(z'^{y'}_{\text{std}}(m'_1,m'_2,n'_2,x')) = C_c$ with increased m'_1. Therefore, we have

$$U_{CR}(z'^{y'}_{\text{std}}(m'_1,m'_2,n'_2,x')) \ge U_{CR}(z^{y}_{\text{std}}(m_1,m_2,n_2,x)).$$

As we have shown $U_{CR}(z'^{y'}_{\text{std}}(m'_1,m'_2,n'_2,x')) \le 1+\theta^{-1}+\theta^{-3}$ in Case A, we conclude the proof for Case B-II.

Finally, by combining Case A and B, we conclude that the CR of LADTSR is upper bounded by $1+\theta^{-1}+\theta^{-3}$. □

I Proof of Lemma H.1

Proof. First, we show that the total demand z' constructed by Algorithm 4 is a standard total demand. First, recall that there are two loops in Algorithm 4, i.e., Lines 3-4, and Lines 6-7. For each item, we consider two cases: $\lambda_{s,k} = C_s/\theta$ and $\lambda_{s,k} = \theta C_s$.

- If $\lambda_{s,k} = C_s/\theta$, after the first loop, the total demand z'_k can be either $z'_k \geq \lceil C_s/\theta \rceil$ or $[0, C_s)$. That is, except $z'_k \geq \lceil C_s/\theta \rceil$, the total demand can only be within the interval $[0, \min\{C_s, \lambda_{s,k}\})$.
- If $\lambda_{s,k} = \theta C_s$, after the second loop, there are only three cases for the total demand z'_k: $z'_k \geq C_s$, $z'_k = \lceil \theta C_s \rceil$, and $0 \leq z'_k < \theta C_s$. That is, except $z'_k \geq C_s$ and $z_k = \lceil \theta C_s \rceil$, the total demand can only be within the interval $[0, \min\{C_s, \lambda_{s,k}\})$.

Combining two cases, we conclude that z' satisfies (a), (b), (c) at the same time.

Second, we show that $U_{\text{CR}}(z') \geq U_{\text{CR}}(z)$. Initially, we have $z' = z$. Then, For each iteration in the first loop, we will show that $U_{\text{CR}}(z')$ does not decrease in Lemma I.1. For each iteration in the second loop, we will show that $U_{\text{CR}}(z')$ does not decrease in Lemma I.2. Then we can conclude that $U_{\text{CR}}(z') \geq U_{\text{CR}}(z)$. □

Lemma I.1. *Given any total demand z, if there exists item j such that, $\lambda_{s,j} = C_s/\theta$ and $C_s \leq z_j < \lceil \lambda_{s,j} \rceil$, then there exists another total demand $z' = \{z'_1, \ldots, z'_K\}$ such that*

$$z'_k = \begin{cases} \lceil \lambda_{s,j} \rceil, & k = j, \\ z_k, & k \neq j, \end{cases} \tag{171}$$

and we have $U_{\text{CR}}(z) \leq U_{\text{CR}}(z')$.

Proof. The proof has two steps: 1) We show that $OPT(z) = OPT(z')$. 2) We show that $U_{\text{ALG}}(z) \leq U_{\text{ALG}}(z')$.

Step 1): For $OPT(z)$ we have

$$\begin{aligned} OPT(z) &\overset{(a)}{=} \min\{C_c, \sum_{k=1}^{K} \min\{C_s, z_k\}\} \\ &\overset{(b)}{=} \min\{C_c, C_s + \sum_{k=1, k\neq j}^{K} \min\{C_s, z_k\}\} \\ &\overset{(c)}{=} \min\{C_c, C_s + \sum_{k=1, k\neq j}^{K} \min\{C_s, z'_k\}\} \\ &\overset{(d)}{=} \min\{C_c, \sum_{k=1}^{K} \min\{C_s, z'_k\}\} \overset{(e)}{=} OPT(z'), \end{aligned} \tag{172}$$

where (a) is due to the definition of $OPT(z)$ (i.e., Eq. (18)), (b) is due to $z_j \geq C_s$, (c) is due to for all the items k such that $k \neq j$ we have $z_k = z'_k$ (i.e., $\min\{C_s, z_k\} = \min\{C_s, z'_k\}$) according to Eq. (171), (d) is due to for item j, we have $z'_j = \lceil C_s/\theta \rceil \geq C_s$ according to Eq. (171), and (e) is due to the definition of $OPT(z)$.

Step 2): We consider two cases: A) If LADTSR makes the combo purchase on z. B) If LADTSR does not make the combo purchase on z.

Case A): In this case, according to Lemma A.2, we have $\sum_{k=1}^{K} \min\{z_k, \lambda_{s,k}\} \geq \lambda_c$. On the other hand, we have $\sum_{k=1}^{K} \min\{z'_k, \lambda_{s,k}\} \geq \sum_{k=1}^{K} \min\{z_k, \lambda_{s,k}\}$ since $z'_k \geq z_k$ for item $k = 1, \ldots, K$ according to Eq. (171). Therefore, LADTSR will make the combo purchase on z' as well.

Then, according to Lemma F.1, we have

$$U_{\text{ALG}}(z) = \lambda_c + C_c + f(s_1, s_2) C_s, \tag{173}$$

where

$$f(s_1, s_2) = \min\{\frac{\lambda_c}{\theta C_s}, s_1 + \frac{\lambda_c - \theta C_s s_1}{C_s/\theta}, s_1 + s_2\}. \tag{174}$$

Let s'_1 (resp. s'_2) denote the number of items in z' such that its threshold is equal to C_s/θ (resp. θC_s) and its total demand exceeds its threshold. Because $s'_1 \geq s_1$ and $s'_2 \geq s_2$ since $z'_k \geq z_k$ for item $k = 1, \ldots, K$, we have $U_{\text{ALG}}(z) \leq U_{\text{ALG}}(z')$.

Case B): In this case, we consider two subcases: I) If LADTSR makes the combo purchase on z'. II) If LADTSR does not make the combo purchase on z'.

Case B-I): Let s_0 denote the number of items that have a total demand less than its single purchase threshold. Clearly, we have $K = s_0 + s_1 + s_2$. Suppose for item $k = 1, \ldots, s_0$, we have $z_k < \lambda_{s,k}$; for item $k = s_0+1, \ldots, s_0+s_1$, we have $\lambda_{s,k} = C_s/\theta$ and $z_k \geq C_s/\theta$; and for item $k = s_0 + s_1 + 1, \ldots, s_0 + s_1 + s_2$, we have $\lambda_{s,k} = \theta/C_s$ and $z_k \geq \theta C_s$. According to lemma F.1 we have

$$U_{\text{ALG}}(z) = \sum_{k=1}^{s_0} z_k + s_1(C_s/\theta + C_s) + s_2(\theta C_s + C_s). \tag{175}$$

Because LADTSR does not make the combo purchase on z, according to Lemma A.2, we have

$$\sum_{k=1}^{K} \min\{z_k, \lambda_{s,k}\} = \sum_{k=1}^{s_0} z_k + s_1 C_s/\theta + s_2 \theta C_s < \lambda_c. \tag{176}$$

Combining Eqs. (175) and (176) we have

$$U_{\text{ALG}}(z) < \lambda_c + (s_1 + s_2) C_s. \tag{177}$$

On the other hand, for z', because LADTSR makes the combo purchase on it, we have

$$U_{\text{ALG}}(z) = \lambda_c + C_c + f(s'_1, s'_2) C_s, \tag{178}$$

where

$$f(s'_1, s'_2) = \min\left\{\frac{\lambda_c}{\theta C_s}, s'_1 + s'_2, s'_2 + \frac{\lambda_c - \theta C_s s'_2}{C_s/\theta}\right\}. \tag{179}$$

Therefore, we only need to show that $s_1 + s_2 \leq f(s'_1, s'_2)$.

- First, from Eq. (176) we have

  $$(s_1 + s_2)\theta C_s \leq s_1 \frac{C_s}{\theta} + s_2 \theta C_s + x < \lambda_c. \tag{180}$$

  Therefore, we have $s_1 + s_2 < \frac{\lambda_c}{\theta C_s}$.
- Second, we have $s_1 \leq s'_1$ and $s_2 \leq s'_2$ because $z'_k \geq z_k$ for item $k = 1, \ldots, K$. Therefore, we have

  $$s_1 + s_2 \leq s'_1 + s'_2. \tag{181}$$

- Finally, from Eq. (176) we have

$$s_1 < \frac{\lambda_c - \theta C_c s_2 - \sum_{k=1}^{s_0} z_k}{C_s/\theta} \le \frac{\lambda_c - \theta C_s s_2}{C_s/\theta}. \tag{182}$$

Therefore, we have

$$s_1+s_2 \le s_2+\frac{\lambda_c - \theta C_s s_2}{C_s/\theta} = (1-\theta^2)s_2+\frac{\lambda_c}{C_s/\theta}. \tag{183}$$

As $s_2 \le s_2'$ and $1-\theta^2 \ge 0$, we have

$$s_1+s_2 \le (1-\theta^2)s_2'+\frac{\lambda_c}{C_s/\theta} = s_2'+\frac{\lambda_c - \theta C_s s_2'}{C_s/\theta}. \tag{184}$$

Therefore, we conclude that $s_1 + s_2 \le f(s_1', s_2')$ and $U_{\text{ALG}}(z) \le U_{\text{ALG}}(z')$.

Case B-II): For item j, we have $U_{\text{ALG}}^j(z') = C_s/\theta + C_s \ge C_s/\theta + 1 \ge \lceil C_s/\theta \rceil$ because $z_j' = \lceil C_s/\theta \rceil \ge C_s/\theta$ and $U_{\text{ALG}}^j(z) = z_j < \lceil C_s/\theta \rceil$ because $z_j < C_s/\theta$. Therefore, we have $U_{\text{ALG}}^j(z) \le U_{\text{ALG}}^j(z')$.

For item k such that $k \neq j$, we have $U_{\text{ALG}}^k(z') = U_{\text{ALG}}^k(z)$ since $z_k' = z_k$ according to Eq. (171).

Therefore, we have $U_{\text{ALG}}^j(z) \le U_{\text{ALG}}^j(z')$.

We conclude the proof by combining Cases A and B. □

Lemma I.2. *Given any total demand z, if there exists item j such that* $\lambda_{s,j} = \theta C_s$ *and* $\lceil \lambda_{s,j} \rceil \le z_j < C_s$*, then there exists another total demand* $z' = \{z_1', \dots, z_K'\}$ *such that*

$$z_k' = \begin{cases} \lceil \lambda_{s,j} \rceil, & k = j, \\ z_k, & k \neq j, \end{cases} \tag{185}$$

and we have $U_{\text{CR}}(z) \le U_{\text{CR}}(z')$.

Proof. Similar to Lemma D.1, we only need to show that 1) $OPT(z) \ge OPT(z')$ and 2) $U_{\text{ALG}}(z) = U_{\text{ALG}}(z')$.

Step 1): For $OPT(z)$, we have

$$\begin{aligned} OPT(z) &\overset{(a)}{=} \min\{C_c, \sum_{k=1}^{K} \min\{C_s, z_k\}\} \\ &\overset{(b)}{=} \min\{C_c, z_j + \sum_{k=1,k\neq j}^{K} \min\{C_s, z_k\}\} \\ &\overset{(c)}{\ge} \min\{C_c, z_j' + \sum_{k=1,k\neq j}^{K} \min\{C_s, z_k'\}\} \\ &\overset{(d)}{=} \min\{C_c, \sum_{k=1}^{K} \min\{C_s, z_k'\}\} \overset{(e)}{=} OPT(z'), \end{aligned} \tag{186}$$

where (a) is due to the definition of $OPT(z)$ (i.e., Eq. (18)), (b) is due to $\min\{C_s, z_j\} = z_j$ because $z_j < C_s$, (c) is due to $z_k' \le z_k$ for item $k = 1, \dots, K$ according to Eq. (185), (d) is due to $\min\{C_s, z_j'\} = z_j'$ because $z_j' = \lceil \theta C_s \rceil \le C_s$, and (e) is due to the definition of $OPT(z)$.

Step 2): For LADTSR, we first show that the combo purchase decision of LADTSR is the same on z and z'. According to Lemma A.2, we only need to show that $\sum_{k=1}^{K} \min\{z_k, \lambda_{s,k}\} = \sum_{k=1}^{K} \min\{z_k', \lambda_{s,k}\}$. This is due to two facts:

- For item j, we have $\min\{z_j, \lambda_{s,j}\} = \min\{z_j', \lambda_{s,j}\} = \lambda_{s,j}$ since $z_j, z_j' \ge \lceil \lambda_{s,j} \rceil \ge \lambda_{s,j}$ according to Eq. (185).
- For item k such that $k \neq j$, we have $z_k = z_k'$ according to Eq. (185).

Thus, we get $\sum_{k=1}^{K} \min\{z_k, \lambda_{s,k}\} = \sum_{k=1}^{K} \min\{z_k', \lambda_{s,k}\}$.

Next, we consider two cases: whether LADTSR makes the combo purchase on z and z' or not.

If LADTSR makes the combo purchase, according to Lemma F.1 we have

$$U_{\text{ALG}}(z) = \lambda_c + C_c + f(s_1, s_2) C_s, \tag{187}$$

where

$$f(s_1, s_2) = \min\{\frac{\lambda_c}{\theta C_s}, s_2 + \frac{\lambda_c - \theta C_s s_2}{C_s/\theta}, s_1 + s_2\}. \tag{188}$$

Let s_1' (resp. s_2') denote the number of items in z' with a threshold equal to C_s/θ (resp. θC_s) and a total demand exceeds its threshold. Then we only need to show that $f(s_1', s_2') = f(s_1, s_2)$.

For item j, we have $z_j \ge \lambda_{s,j}$ and $z_j' = \lceil \lambda_{s,j} \rceil \ge \lambda_{s,j}$ and for any other item k such that $k \neq j$, we have $z_k = z_k'$ according Eq. (185). Therefore, we have $s_1' = s_1$ and $s_2' = s_2$. Furthermore, we have $U_{\text{ALG}}(z) = U_{\text{ALG}}(z')$.

If LADTSR does not make the combo purchase, in Eq. (69), we have

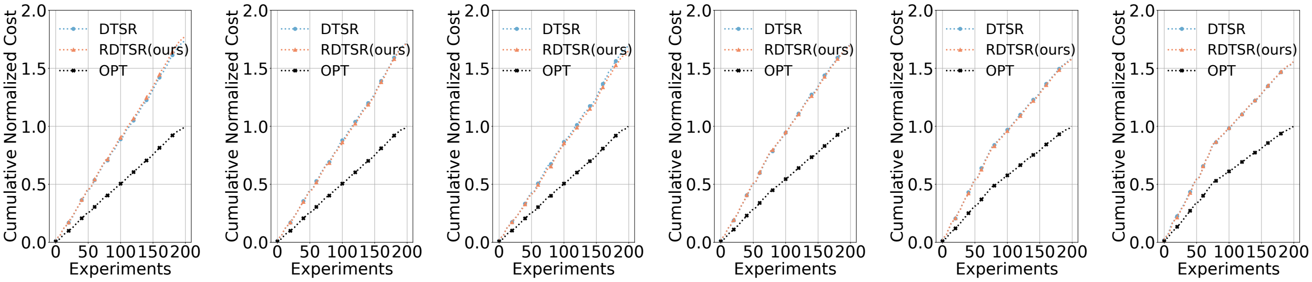
$$U_{\text{ALG}}^k(z) = \begin{cases} z_k, & z_k < \lambda_{s,k}, \\ \lambda_{s,k} + C_s, & \text{otherwise.} \end{cases} \tag{189}$$

For item j, we have $z_j, z_j' \ge \lambda_{s,j}$ and $U_{\text{ALG}}^j(z) = U_{\text{ALG}}^j(z') = \lambda_{s,j} + C_s$; for item k such that $k \neq j$, we have $z_k = z_k'$ and hence $U_{\text{ALG}}^k(z) = U_{\text{ALG}}^k(z')$. Therefore, we have $U_{\text{ALG}}(z) = \sum_{k=1}^{K} U_{\text{ALG}}^k(z) = U_{\text{ALG}}^k(z')$.

Combining two cases, we have $U_{\text{ALG}}(z) = U_{\text{ALG}}(z')$. □

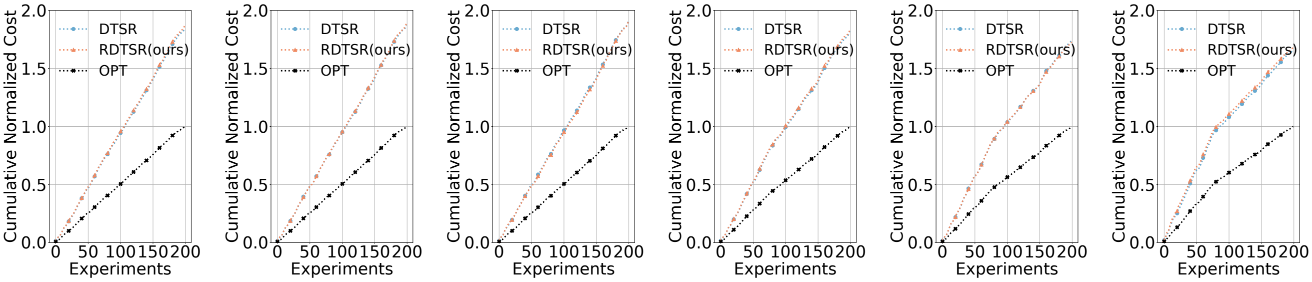
J Average Performance Evaluation

In this section, we compare the average performance of RDTSR with DTSR. We adopt the metric cumulative normalized cost used in (Wu, Bao, and Yuan 2021), which is defined as the sum of the algorithm's cost divided by the sum of the offline optimal cost over all the sequences. We vary the value of the combo purchase price C_c according to (Wu, Bao, and Yuan 2021) and for each value of C_c, we run our experiments for 200 independent demand sequences. In Figs. 3, 4, and 5, we show the average performance comparison under all the three datasets, i.e., Synthetic (unit demands and multi-unit demands), Azure, and AppUsage datasets. For the synthetic dataset, the uniform setting represents all 200 sequences are uniform sequences; the mixed setting represents 40% of sequences are uniform and 60% of sequences are long-tailed. From Fig. 3, we can observe that our algorithm RDTSR has a similar performance compared to DTSR. This indicates that over the long term (in this experiment, 200 rounds), two algorithms have the closing total costs. Similar results can also be obtained from Figs. 4 and 5.



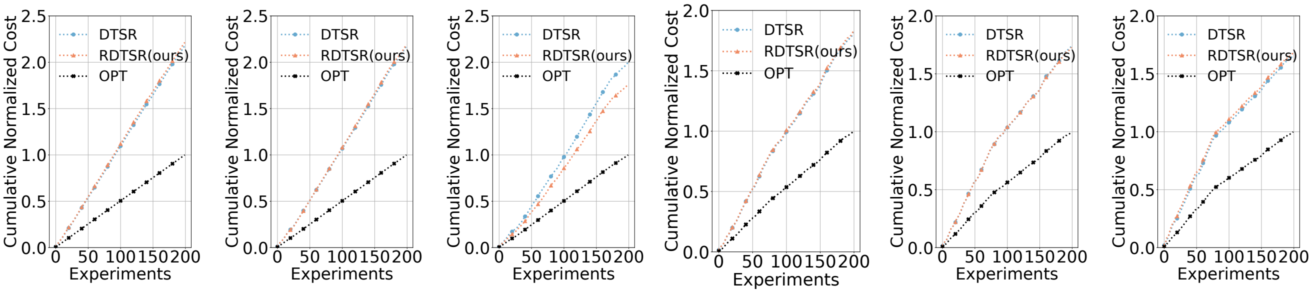
(a) $C_c = 20$ (b) $C_c = 25$ (c) $C_c = 31$ (d) $C_c = 20$ (e) $C_c = 25$ (f) $C_c = 33$

Figure 3: Average performance on the synthetic dataset (unit demands). Figs. 3a, 3b, and 3c are under the uniform sequence setting; Figs. 3d, 3e, and 3f are under the mixed sequence setting.



(a) $C_c = 20$ (b) $C_c = 25$ (c) $C_c = 31$ (d) $C_c = 20$ (e) $C_c = 25$ (f) $C_c = 33$

Figure 4: Average performance on the synthetic dataset (multi-unit demands). Figs. 4a, 4b, and 4c are under the uniform sequence setting; Figs. 4d, 4e, and 4f are under the mixed sequence setting.



(a) $C_c = 20$ (b) $C_c = 25$ (c) $C_c = 30$ (d) $C_c = 20$ (e) $C_c = 25$ (f) $C_c = 30$

Figure 5: Average performance on the real-world datasets. Figs. 5a, 5b, and 5c are under the Azure dataset; Figs. 5d, 5e, and 5f are under the AppUsage dataset.