通过最佳可能竞争分析改进的多选项滑雪租赁问题学习增强算法

Yongho Shin、Changyeol Lee、Gukryeol Lee 和 Hyung-Chan An*

计算机科学系，延世大学，首尔，韩国

arXiv:2302.06832v1 [cs.DS] 14 Feb 2023

摘要

本文提出了多选项滑雪租赁问题的改进学习增强算法。学习增强算法将机器学习预测作为输入的附加部分，并将这些预测纳入给定问题的求解过程。由于这类算法具有独特优势，能够将机器学习预测的能力与严格的性能保证相结合，因此近年来在在线优化问题的背景下得到了广泛研究。尽管滑雪租赁问题是在线优化领域的典型问题之一，但此前对于多选项滑雪租赁问题，无论是否采用学习增强，都只有确定性算法已知。我们提出了该问题的首个随机化学习增强算法，其性能超越了以往确定性算法所给出的性能保证。我们的学习增强算法建立在一种新的、经证明为该问题最佳可能的随机化竞争算法之上。此外，我们还给出了确定性算法和随机化算法的下界，并通过计算实验评估了我们算法在性能方面的改进。

1 引言

学习增强算法将机器学习预测作为输入的附加部分，并利用该预测来求解给定问题。这类算法的一大优势是，既能借助强大的机器学习预测，又能同时提供可证明的性能保证。这些保证往往优于不使用预测的经典算法所能提供的保证，而且无需对机器学习预测的生成方式作出任何假设，从而保留了其黑盒特性。近年来，学习增强算法在在线优化领域取得的成功尤为显著。在这一领域中，输入会在多个时间步中以在线方式逐步提供，我们必须在每个时间步作出不可撤销的决策。学习增强算法在这种场景下大有用武之地，这并不令人意外，因为设计经典竞争算法的挑战通常在于：在对未来输入一无所知的情况下避免最坏情形，而机器学习预测可以替代这部分缺失的信息。Lykouris 和 Vassilvitskii（2021）的开创性工作，以及后续研究（例如 Kumar 等人，2018年；Bamas 等人，2020年；Lattanzi 等人，2020年），充分展现了学习增强算法在在线优化领域的成功。

滑雪租赁问题 是在线优化领域中经典的问题之一，无论是否采用学习增强，都已得到广泛研究。它们简洁地体现了在线优化的核心特性，许多在线优化算法技术也由此发展而来。对于学习增强算法而言，滑雪租赁问题也自然而然地成为了重要的试验平台（Kumar 等人，2018年；Gollapudi 和 Panigrahi，2019年；Bamas 等人，2020年；Wei 和 Zhang，2020年）。

尽管滑雪租赁问题出现在各种应用中（Fleischer，2001年；Karlin 等人，2001年；Meyerson，2005年），但或许最容易将其表述为一个简单的滑雪板租赁问题（因此得名），如下所示。在多选项滑雪租赁问题中，我们给定一组 n 租赁选项：对于$i = 1, \cdots, n$，我们可以选择以成本 $c_i \in \mathbb{Q}_+$租赁滑雪板 $d_i \in \mathbb{Z}_+ \cup \{\infty\}$ 天。设 T为我们将要滑雪的天数，但这取决于天气等因素，因此我们无法提前得知 T；只有在某一天结束时，我们才会知道那天确实是滑雪的最后一天。该问题的目标是在确保我们租到滑雪板，满足整个 T 天的需求的同时，最小化总租赁成本。传统上，租赁或购买情形仅有两个选项（d_1, c_1） =(1,1)，并且

*通讯作者： hyung-chan.an@yonsei.ac.kr

$(d_2, c_2) = (\infty, B)$，已得到广泛研究，既包括不带学习增强的经典竞争分析框架（Karlin 等人，1988年、1994年；Buchbinder 等人，2007年），也包括学习增强框架（Kumar 等人，2018年；Gollapudi 和 Panigrahi，2019年；Angelopoulos 等人，2020年；Bamas 等人，2020年；Banerjee，2020年；Wei 和 Zhang，2020年）。一般的多选项问题也已在这两种框架下得到研究（Zhang 等人，2011年；Ai 等人，2014年；Wang 等人，2020年；Anand 等人，2021年）。

本文提出了首个适用于多选项滑雪租赁问题的随机化学习增强算法。此前，Anand 等人（2021年）为同一问题提出了一种确定性学习增强算法。他们的算法性能保证采用标准的一致性-鲁棒性方案来表述：该算法具有 $(1+\lambda)$-一致性和 $(5+5/\lambda)$-鲁棒性，其中 λ 是权衡参数（或“置信度”），它决定算法对预测的“信任”或“忽略”程度，并决定性能保证。[1] 他们的学习增强算法通过修改经典竞争算法，使其使用给定的机器学习预测而得（Anand 等人，2021年）；因此，为了获得改进的学习增强算法，自然需要深入研究如何为经典问题设计更具竞争力的算法。

第 3 节是预备部分，我们首先考察确定性算法。我们在第 3.1 节提出一种不含学习增强的竞争算法，它与 Anand 等人的算法非常相似，竞争比也相同。尽管如此，我们仍然介绍该算法，因为其中细微的差别最终证明有助于获得第 3.2 节提出的改进学习增强算法。为实现这一改进，我们允许学习增强算法“忽略”预测。此前的算法（Anand 等人，2021年）会在内部假设（缩放后的）预测准确，从而计算最优解，并坚持将该解纳入算法的输出，以此保证一致性。相比之下，当权衡参数表明机器学习预测不太可靠时，我们的算法能够选择不这样做。尽管这听起来合乎情理，但这是我们算法的一项新特性，并由此改善了性能权衡。

在第 4 节中，我们展示随机化如何改进我们的算法。第 4.1 节介绍了一种用于多选项滑雪租赁问题的 e-竞争算法。在 Anand 等人（2021）采用的倍增方案中，可以通过计算算法将使用的倍增预算，构造一个具有恶意的实例。我们可以通过预算随机化来防止对手利用这一点；我们的算法表明，只需少量随机性，就足以得到一个 e-竞争算法，而且可以证明这是可能达到的最佳结果（参见第 5.2 节）。在第 4.2 节中，我们基于这一 e-竞争算法改进学习增强算法的性能权衡。我们的学习增强算法具有 $\chi(\lambda)$-一致性和 (e^λ/λ)-鲁棒性，其中

$$\chi(\lambda) := \begin{cases} 1+\lambda, & \text{if } \lambda < \frac{1}{e}, \\ (e+1)\lambda - \ln\lambda - 1, & \text{if } \lambda \geq \frac{1}{e}, \end{cases}$$

优于此前的算法。

在第 5 节中，我们给出下界。我们首先提出一个称为按钮问题 的辅助问题，它更便于进行下界论证；随后，我们将其归约到多选项滑雪租赁问题。在第 5.2 节中，我们证明第 4.1 节中的算法是可能达到的最佳算法：也就是说，对于所有常数 $\rho < e$，不存在适用于多选项滑雪租赁问题的随机化 ρ-竞争算法。为证明这一点，我们仔细构造了一个线性规划（LP），用于界定给定实例下任何随机化算法的竞争比。然后，我们通过解析地构造该线性规划对偶问题的可行解，得到所需的下界。第 5.3 节首先证明确定性算法的竞争比下界为 4。尽管 Zhang 等人（2011）已经给出了相同的下界 4，我们仍在此给出这一证明，以便将其扩展为第 5.3 节中确定性学习增强算法的下界。

最后，在第 6 节中，我们通过实验评估学习增强算法的性能。我们在与 Kumar 等人（2018年）类似的设置下开展计算实验，以衡量算法的性能，从而展示我们的算法所带来的性能提升。

相关工作 近年来，学习增强算法已被应用于广泛的传统操作优化问题。许多研究将机器学习预测引入在线优化问题 a-s

[1]直观地说，一致性是一种仅在机器学习预测准确时才有效的性能保证，而鲁棒性则始终有效。二者都以算法输出与事后得知真实最优值之比的最坏情况作为衡量标准；形式化定义见第 2 节。

例如，具体包括缓存（Rohatgi，2020年；Lykouris 和 Vassilvitskii，2021年；Im 等，2022年）、匹配（Antoniadis 等，2020年b；Lavastida 等，2021年）以及图/度量问题（Antoniadis 等，2020年a；Azar 等，2022年；Jiang 等，2022年）。感兴趣的读者可参阅 Mitzenmacher 和 Vassilvitskii（2022年）的综述，以获得更全面的回顾。

滑雪租赁问题本身就是一种传统的在线优化问题，也不例外地受到广泛研究。Karlin 等人（1994年）提出了一种用于租或买问题的随机化（$e/(e-1)$）竞争算法，这是可能达到的最佳算法。Zhang 等人（2011年）针对边际成本递减假设下的多选项滑雪租赁问题，提出了一种确定性的4-竞争算法。该问题也在多种情境下得到研究，例如多商店滑雪租赁问题（Ai 等人，2014年）、Banchard问题（Fleischer，2001年）、动态TCP确认问题（Karlin 等人，2001年）以及停车许可证问题（Meyerson，2005年）。

2 预备知识

在未使用学习增强的多选项滑雪租赁问题中，给定一组 n 租赁选项：每个选项 i 覆盖 $d_i \in \mathbb{Z}_+ \cup \{\infty\}$ 天，成本为 $c_i \in \mathbb{Q}_+$。该问题的一个解是一个选项序列。在此问题中，每个滑雪日开始时，如果该日尚未被当前解覆盖，我们就必须选择一个选项，并将其加入“当前”输出解。设 T 为滑雪天数，该数值仅在第 T天结束时才会揭示。问题的目标是以最小成本的解覆盖这 T 天。若某算法输出的期望成本不超过最小成本的 γ 倍，我们就称该算法为 γ-竞争 。

在学习增强的多选项滑雪租赁问题中，我们还会得到一个关于滑雪天数的预测 $\hat{T}$。学习增强算法的性能通过标准的一致性-鲁棒性分析来衡量。也就是说，如果预测准确（即 $T = \hat{T}$），且算法满足 $\mathbb{E}[\text{SOL}] \le \chi \cdot \text{OPT}(T)$，我们就称该算法为 χ-一致；如果对于所有 T，无论预测 $\hat{T}$ 有多准确，算法都满足 $\mathbb{E}[\text{SOL}] \le \rho \cdot \text{OPT}(T)$，则称该算法为 ρ-鲁棒。

给定两个解 SOL_1 和 SOL_2，，如果将 SOL_2 中的选项添加到 SOL_1的末尾，我们就称将SOL_2 追加到 SOL_1 。例如，如果 SOL_1 选择选项 1，而 SOL_2 选择选项 2 、选项 3 和选项 3，那么将 SOL_2 追加到 SOL_1 后，得到的解将选择选项 1、选项 2、选项 3 和选项 3。

对于每个 $t \in \mathbb{Z}_+$，令 OPT(*t*) 表示一个覆盖至少 t 天的最低成本解。若有多个解，则任意选择。为简化记号，在上下文明确时，我们可能会稍微滥用记号，用 OPT(*t*) 表示其成本，而非解本身。不失一般性，假设 $\text{OPT}(1) \ge 1$；否则，我们可以将每个选项的成本除以 OPT(1)。在本文接下来的部分中，算法会将某个 t 对应的最优解 OPT(*t*) 添加到“当前”解中。注意，对于任何 t ，无需知道实际的滑雪天数 T，也可以计算出这个 OPT(*t*)：可以使用标准的动态规划技术求得 OPT(*t*)。

3 确定性算法

本节介绍我们针对多选项滑雪租赁问题提出的确定性算法。我们先在第 3.1 节介绍一种 4-竞争算法。尽管该算法与 Anand 等人于 2021年提出的算法非常相似，我们仍在此介绍自己的算法，因为它能导出一种改进的学习增强算法，详见第 3.2 节。

为便于说明，我们将描述算法的执行过程，仿佛它永不终止；换言之，总会有新的一天到来进行滑雪。不过，实际算法一旦得知已经到达最后一天，就会立即终止。对于任意 $j \ge 1$，令 $b(j)$ 表示在成本不超过 j 的解中，覆盖天数最多的解（或其成本），即 $b(j) :=$最优值($t^\star$)，其中$t^\star := \max \{t \in \mathbb{Z}_+ \cup \{\infty\} \mid \text{最优值}(t) \le j\}$。也就是说， $b(j)$ 是预算为 j 时能采取的“最佳”方案。

3.1 竞争算法

该算法分为若干次迭代。对于每次迭代 i（对于 $i = 1, 2, \cdots$），令 SOL_i 表示我们的解在第 i 次迭代结束时的总成本（从第一次迭代开始计算）。在第一次迭代中，我们将 OPT(1) 添加到解中。因此，$\text{SOL}_1 :=$OPT(1)。令 $\tau_1 := 1$。在之后的每次迭代 $i \ge 2$中，我们将 $b(\text{SOL}_{i-1})$ 添加到解中。令 τ_i 表示在第 次迭代中新近覆盖的天数

，或者换句话说，即由 $b(\mathrm{SOL}_{i-1})$ 覆盖的天数。注意，如果 $b(\mathrm{SOL}_{i-1})$ 包含一个购买选项（即，一个满足 $d_j = \infty$ 的选项 j），那么 τ_i 变为 ∞，并且不再有后续迭代。

我们注意到，与 Anand 等人（2021年）的算法不同之处在于，受 Zhang 等人（2011年）的启发，我们在每次迭代 i 中附加 $b(\mathrm{SOL}_{i-1})$，而不是 $b(2$ 最优值$(\tau_{i-1}))$。

定理 1. 该算法是 *4*-竞争的。

该定理的证明见附录 A。

3.2 学习增强算法

在本小节中，我们描述确定性学习增强算法。给定预测 $\widehat{}$diction T on T 和置信度 $\lambda \in [0,1]$。该算法最多由以下三个阶段组成。

$\widehat{}$**第一忽略阶段** 如果 $\mathrm{OPT}(1) \le \lambda\,\mathrm{OPT}(T)$，我们就在一开始进入该阶段。否则，我们直接进入尊重阶段。在此阶段，我们只需运行前述 4-竞争算法。令 $\rightsquigarrow i^\star$ 为竞争算法产生的总成本首次超过 $\lambda\,\mathrm{OPT}(T)$ 的迭代，即

$$\mathrm{SOL}_{i^\star-1} \le \lambda\,\mathrm{OPT}(\widehat{T}) < \mathrm{SOL}_{i^\star}\,. \tag{1}$$

$\widehat{}$注意，始终存在这样的 $i^\star \ge 2$，因为只有在 $\mathrm{OPT}(1) \le \lambda\,\mathrm{OPT}(T)$ 时，我们才会进入这一阶段。处理完迭代$i^\star$后，我们会根据$\widehat{}\mathrm{SOL}_{i^\star}$转入尊重阶段或第二忽略阶段。如果 $\mathrm{SOL}_{i^\star} \le \mathrm{OPT}(T)$，我们就进入尊重阶段；否则，就进入第二忽略阶段。

尊重阶段 直观而言，在这一阶段，算法会尊重预测。在这一阶段，我们会附加 $\mathrm{OPT}(T)$，然后转入第二忽略阶段。

第二忽略阶段 在这一阶段，我们会稍作修改后运行 4-竞争算法，具体如下。

首先，令 SOL'_0 表示迄今产生的总价格。因此，如果前一阶段是尊重阶段，则有 $\mathrm{SOL}'_0 := \mathrm{SOL}_{i^\star} + \mathrm{OPT}(\widehat{T})$（如果跳过了第一忽略阶段，则有 $\mathrm{SOL}'_0 := \mathrm{OPT}(\widehat{T})$）。另一方面，如果前一阶段是第一忽略阶段，则有 $\mathrm{SOL}'_0 := \mathrm{SOL}_{i^\star}$。

然后，我们选择 τ'_0，使其成为迄今覆盖天数的下界。如果前一阶段$\widehat{}$是尊重阶段，我们选择 $\tau'_0 := T$。如果紧接在前的阶段是第一忽略阶段，我们选择 $\tau'_0 := \tau_{i^\star}$。

现在，对于每次迭代 $i \ge 1$，我们将 $b(\mathrm{SOL}'_{i-1})$ 添加到解中，并令 τ'_i 表示 $b(\mathrm{SOL}'_{i-1})$所覆盖的天数。算法描述到此结束。

下面给出该学习增强算法的主要定理。证明见附录 A。

定理 2. 该算法是一个确定性的 $\max\{1+2\lambda, 4\lambda\}$-一致、 $\left(2+\frac{2}{\lambda}\right)$-鲁棒算法。 –

4 随机化算法

本节中，我们给出多选项滑雪租赁问题的随机化算法。我们先介绍我们的 e-竞争算法，然后介绍我们的学习增强算法。对于任意给定的 $\lambda \in [0,1]$，该算法均为 $\chi(\lambda)$-一致且 (e^λ/λ)-鲁棒，其中

$$\chi(\lambda) := \begin{cases} 1+\lambda, & \text{if } \lambda < \frac{1}{e}, \\ (e+1)\lambda - \ln\lambda - 1, & \text{if } \lambda \ge \frac{1}{e}. \end{cases}$$

与上一节一样，我们将描述算法的执行过程，仿佛它永不终止。还要回顾一下，对于任意 $j \ge 1$, $b(j)$ 表示在成本不超过 j 的解（或其成本）中覆盖天数最多的解。如果 $j < \mathrm{OPT}(1)$，令 $b(j) := \emptyset$ 为空解。

4.1 竞争算法

令 SOL 为我们维护的解，初始时 SOL $:= \emptyset$。一开始，我们从概率密度函数为 $f(\alpha) := 1/\alpha$ 的分布中采样 $\alpha \in [1, e)$。随后，算法分为若干阶段执行。在第 i 阶段（对于 $i = 0,\ 1,\ \cdots$），我们将 $b(\alpha e^i)$ 追加到 SOL 中。

定理 3。 *给定算法是一个随机化 e-竞争算法。*

该定理的证明推迟至附录 B.1。

4.2 学习增强算法

现在我们介绍我们的随机化学习增强算法。回顾一下，我们已知一个预测 T以及置信度 $\lambda \in [0, 1]$。

假设 我们需要若干假设来描述该算法。首先，假设 OPT($\widehat{T}$) $= e^k$ 对于某个整数 k成立。该假设不失一般性，因为如果我们有$e^{k-1} <$ OPT($\widehat{T}$) $< e^k$，则可以将每个选项的成本乘以 $\frac{e^k}{\mathrm{OPT}(\widehat{T})}$。我们还假设 $\lambda \in (0, 1)$；我们将在本节末尾讨论 $\lambda \in \{0, 1\}$ 的情况。最后，假设 λOPT(T) $\geq e$。注意，只需将每个选项的成本乘以适当的 e 次幂，就能轻松满足这一假设。

下文中，对于某个 $q \in \{0, 1, 2, \cdots, k-1\}$ 和 $r \in (0, 1]$，我们可能写作 $\lambda := e^{-q-r}$。这里需要说明 r的取值范围：它严格为正。例如，如果 $\lambda = e^{-i}$ 对某个整数 i成立，我们就将这个 λ视为具有 $q = i - 1$ 和 $r = 1$。

算法描述 令 $\mathcal{A}$ 为第 4.1 节中给出的算法。我们运行 $\mathcal{A}$，并执行与 $\mathcal{A}$相同的阶段。在每个阶段 i，我们附加与 $\mathcal{A}$相同的解，但有以下例外：如果 $\alpha e^i \in [\lambda$OPT($\widehat{T}$), OPT($\widehat{T}$)) $= [e^{k-q-r}, e^k)$，则在该阶段附加 OPT($\widehat{T}$)，而不是 $b(\alpha e^i)$。如果在先前的某个阶段已经附加过 OPT(T)，我们就不再重复附加，而是直接跳过。

定理 4。 *对于 $\lambda \in (0, 1)$，该算法是 $\chi(\lambda)$-一致且 (e^λ/λ)-鲁棒的，其中 χ 定义如下：*

$$\chi(\lambda) := \begin{cases} 1 + \lambda, & \text{if } \lambda < \frac{1}{e}, \\ (e+1)\lambda - \ln \lambda - 1, & \text{if } \lambda \geq \frac{1}{e}. \end{cases}$$

一致性分析 首先证明我们的算法是 $\chi(\lambda)$-一致的。令 SOL 表示我们的算法截至结束时产生的总成本。对于每个阶段 $i = 0, 1, \cdots, k-q-2$，算法会追加与 $\mathcal{A}$ 相同的 $b(\alpha e^i)$。因此，直到阶段 $(k-q-2)$，我们的算法产生的期望成本也至多为

$$\int_1^e \left(\sum_{i=0}^{k-q-2} \alpha e^i \right) f(\alpha) d\alpha \leq e^{k-q-1}. \tag{2}$$

暂且假设 $\lambda < e^{-1}$，即 $q \geq 1$。假设算法进入阶段 $(k-q-1)$。如果$\alpha e^{k-q-1} \geq e^{k-q-r}$（或简写为 $\alpha \geq e^{1-r}$），算法会追加 OPT($\widehat{T}$) 并随即终止。否则，如果 $\alpha < e^{1-r}$，算法会追加 $b(\alpha e^{k-q-1})$，并可能继续进入下一阶段。在阶段 $(k-q)$，算法会追加 OPT($\widehat{T}$) 并终止，因为 $\alpha e^{k-q} \in [e^{k-q-r}, e^k)$。结合方程 (2)，我们的算法的总期望成本可界定为

$$\begin{aligned} \mathbb{E}[\mathrm{SOL}] &\leq e^{k-q-1} + \int_1^{e^{1-r}} \alpha e^{k-q-1} f(\alpha) d\alpha + e^k \\ &\leq e^{k-q-r} + e^k \\ &= (1+\lambda) \cdot \mathrm{OPT}(\widehat{T}). \end{aligned} \tag{3}$$

现在我们转向 $\lambda \leq e^{-1}$ 的情形，也就是 $q = 0$。这里与前一种情形有一处不同：当算法进入阶段 $(k-1)$ 时，如果 $\alpha < e^{1-r}$，算法不仅会追加 $b(\alpha e^{k-1})$，还会

也可能 p 继续进入阶段 k，并追加 $b(\alpha e^k)$，因为 $\alpha e^k \geq \mathrm{OPT}(\widehat{T})$。因此，再次根据方程 (2),

$$\begin{aligned}\mathbb{E}[\mathrm{SOL}] &\leq e^{k-1} + \int_1^{e^{1-r}} (\alpha e^{k-1} + \alpha e^k) f(\alpha) d\alpha \\ &\quad + \int_{e^{1-r}}^{e} e^k f(\alpha) d\alpha \\ &\leq e^{k+1-r} + e^{k-r} + e^k(r-1) \\ &= ((e+1)\lambda - \ln\lambda - 1)\,\mathrm{OPT}(\widehat{T})\end{aligned} \tag{4}$$

其中最后一个等号成立，是因为 $\lambda = e^{-r}$。

鲁棒性分析 现在我们证明，该算法是 (e^λ/λ)-鲁棒的。根据 OPT(T) 的不同，下面分几种情形讨论。

情形 1. $\mathrm{OPT}(T) < e^{k-q-2}$. 注意，在这种情形下，该算法的执行与 $\mathcal{A}$ 相同，根据定理 3，可得$\mathbb{E}$[SOL] $\leq e\,\mathrm{OPT}(T)$。

情形 2. $e^{k-q-2} \leq \mathrm{OPT}(T) < e^{k-q-1}$. 设 $\mathrm{OPT}(T) = \beta e^{k-q-2}$，其中 $\beta \in [1, e)$。注意，在阶段 $(k-q-2)$ 之前，该算法追加的解与 $\mathcal{A}$ 相同。如果 $\alpha \geq \beta$ 属于 $\mathcal{A}$，该算法会在阶段 $(k-q-2)$ 终止。否则，该算法可能继续进入阶段 ($k-q-1$)。

假设 $\beta < e^{1-r}$。注意，如果该算法进入阶段 $(k-q-1)$，则有 $\alpha e^{k-q-1} < \lambda\,\mathrm{OPT}(T)$，这意味着该算法的执行与 $\mathcal{A}$ 相同。因此，根据定理 3，我们再次得到 $\mathbb{E}[\mathrm{SOL}] \leq e\,\mathrm{OPT}(T)$。

现在假设 $\beta \geq e^{1-r}$。注意，无论是 $q \geq 1$ 还是 $q = 0$，如果 $\alpha < e^{1-r}$，算法都会按照 $\mathcal{A}$ 追加 b (αe^{k-q-1})，然后终止；否则，如果 $e^{1-r} \leq \alpha < \beta$，算法$\widehat{\ }$追加 OPT($T$)，然后终止。因此我们有

$$\begin{aligned}\mathbb{E}[\mathrm{SOL}] &\leq \int_1^e \left(\sum_{i=0}^{k-q-2} \alpha e^i\right) f(\alpha) d\alpha \\ &\quad + \int_1^{e^{1-r}} \alpha e^{k-q-1} f(\alpha) d\alpha \\ &\quad + \int_{e^{1-r}}^{\beta} e^k f(\alpha) d\alpha \\ &= \left(e^{k-q-1} - 1\right) + (e^{1-r} - 1) e^{k-q-1} \\ &\quad + (\ln\beta + r - 1) e^k \\ &\leq e^{k-q-r} + (\ln\beta + r - 1) e^k.\end{aligned}$$

现在，我们可以将这种情况下的鲁棒性比率上界估计如下:

$$\begin{aligned}\frac{\mathbb{E}[\mathrm{SOL}]}{\mathrm{OPT}(T)} &\leq \frac{e^{k-q-r} + (\ln\beta + r - 1) e^k}{\beta e^{k-q-2}} \\ &= \frac{e^{2-r}}{\beta} + \frac{(\ln\beta + r - 1) e^{q+2}}{\beta} \\ &= \frac{e^{2-r}}{\beta} + \frac{(\ln\beta + r - 1) e^{2-r}}{\lambda\beta}\end{aligned}$$

其中，最后一个等式来自 $\lambda = e^{-q-r}$ 的定义。以下断言完成了这种情况下的证明。

断言 1. 对于每个 $\beta > 0$，我们有 $h(\beta) := \frac{e^{2-r}}{\beta} + \frac{(\ln\beta + r - 1) e^{2-r}}{\lambda\beta} \leq \frac{e^\lambda}{\lambda}$。

这一结论可通过简单的微积分推导得出，具体推导见附录 B.2。

情形 3. $e^{k-q-1} \le$最优值$(T) < \lambda$最优值$(\widehat{T}) = e^{k-q-r}$. 令最优值$(T) = \beta e^{k-q-1}$ 对于某个 $\beta \in [1, e^{1-r})$。暂且假设 $q \ge 1$。请注意，我们的算法执行过程可描述为以下步骤。

(a) 对于每个阶段 $i = 0, 1, \cdots, k-q-2$, 算法追加与 $\mathcal{A}$ 相同的解。(b) 在阶段 $(k-q-1)$ 中，如果 $\alpha < e^{1-r}$，算法按照 $\mathcal{A}$ 追加 $b(\alpha e^{k-q-1})$。否则，算法追加最优值(T)。如果 $\alpha \ge \beta$，算法随即终止。(c) 如果 $\alpha < \beta < e^{1-r}$，算法进入阶段 $(k-q)$，并追加最优值$(\widehat{T})$，因为对于 $q \ge 1$, $\alpha e^{k-q} \in [e^{k-q-r}, e^k)$。随后算法终止。

Fro根据这一描述，我们可以将算法产生的总期望成本上界写为 s:

$$\begin{aligned}\mathbb{E}[\mathrm{SOL}] &\le \int_1^e \sum_{i=0}^{k-q-2} \alpha e^i f(\alpha) d\alpha + \int_1^{e^{1-r}} \alpha e^{k-q-1} f(\alpha) d\alpha \\ &\quad + \int_{e^{1-r}}^{e} e^k f(\alpha) d\alpha + \int_1^{\beta} e^k f(\alpha) d\alpha \\ &\le e^{k-q-r} + e^k \cdot (\ln \beta + r),\end{aligned}$$

由此可得，此情形下的鲁棒性比率上界为

$$\frac{e^{k-q-r} + e^k \cdot (\ln \beta + r)}{\beta \cdot e^{k-q-1}} = \frac{e^{1-r}}{\beta} + \frac{e^{1-r}(\ln \beta + r)}{\lambda \beta}$$

其中等号成立是因为根据定义，$\lambda = e^{-q-r}$。我们声称，右侧还可以进一步界定为 $\frac{e^\lambda}{\lambda}$，从而完成此情形的证明。该论断的证明见附录 B.2。

命题 2. 对每个 $\beta > 0$，我们有 $h(\beta) := \frac{e^{1-r}}{\beta} + \frac{e^{1-r}(\ln \beta + r)}{\lambda \beta} \le \frac{e^\lambda}{\lambda}$。

现在我们转向 $q = 0$ 的情况，即 OPT$(T) = \beta \cdot e^{k-1}$ 且 $\lambda = e^{-r}$。注意，与 $q \ge 1$ 的情况相比，执行过程存在一处差异；在步骤 (c) 中，算法追加 $b(\alpha \cdot e^k)$，而非 OPT$(\widehat{T})$，因为算法现在进入了阶段 k，因此 $\alpha \cdot e^k \notin [e^{k-q-r}, e^k)$。注意，此情况下的总期望价格可由下式界定

$$\begin{aligned}\mathbb{E}[\mathrm{SOL}] &\le \int_1^e \sum_{i=0}^{k-2} \alpha e^i f(\alpha) d\alpha + \int_1^{e^{1-r}} \alpha e^{k-1} f(\alpha) d\alpha \\ &\quad + \int_{e^{1-r}}^{e} e^k f(\alpha) d\alpha + \int_1^{\beta} \alpha \cdot e^k f(\alpha) d\alpha \\ &\le e^{k-r} + e^k \cdot (\beta + r - 1),\end{aligned}$$

由此得到此情况下鲁棒性比率的如下上界：

$$\frac{e^{k-r} + e^k \cdot (\beta + r - 1)}{\beta \cdot e^{k-1}} = e \cdot \left(1 + \frac{\lambda + r - 1}{\beta}\right) \le e \cdot (\lambda + r),$$

其中，不等式可由 $\beta \ge 1$ 这一事实推出。以下命题完成了此情况的证明。回顾 $\lambda = e^{-r}$，因此对于此情况，$r = -\ln \lambda$。

命题 3. 对于每个 $x > 0$，都有 $e(x - \ln x) \le \frac{e^x}{x}$。

该命题的证明见附录 B.2。

剩下需要证明的是 OPT$(T) \ge \lambda$OPT(T) 的情形。这些剩余情形的证明同样见附录 B.2。

最后的说明 在分析中，我们假设 $\lambda \in (0, 1)$。不过，我们可以看到，当 $\lambda = 0$ 或 $\lambda = 1$ 时，定理 4 仍然成立。事实上，如果 $\lambda = 0$，该算法可能相当于一开始就附加 OPT(T)。请注意，这是一致的 1-一致算法，但 ∞-鲁棒。另一方面，如果 $\lambda = 1$，该算法与 $\mathcal{A}$ 本身完全相同，这意味着该算法是 e-一致且 e-鲁棒的。

5 下界

本节中，我们给出多选项滑雪租赁问题的下界。

5.1 辅助问题

我们定义一个称为按钮问题的辅助问题。在该问题中，给定一个有序的 m 按钮列表，其中一些按钮被指定为目标。我们知道这些按钮的“目标属性”是单调的，即存在某个 $J \le m$，使得按钮 1 至 $(J-1)$ 不是目标，而按钮 J 至 m 全部都是目标。然而，我们事先不知道第一个目标按钮（即按钮 J）的位置。要感知按钮 j 是否为目标，我们必须点击它并支付 $b_j \in \mathbb{Q}_+$ 作为价格；随后我们便会知道该按钮是否为目标。按钮的价格在开始时已全部给定。我们还假设价格非递减，即 $b_1 \le b_2 \le \cdots \le b_m$。算法会不断点击按钮，直到点击到一个目标按钮，而自然目标是最小化总价格。

如果一个按钮问题算法满足 $\mathbb{E}[\mathrm{SOL}] \le \gamma \cdot b_J$，我们就称其具有 γ-竞争比，其中 SOL 是该算法点击目标按钮之前所 incurred 的总价格。在该问题的学习增强版本 $\widehat{\ }$ 中，算法会获得关于第一个目标按钮 J 的预测 $\widehat{J}$。如果预测准确（即，$\widehat{J} = J$），且算法满足 $\mathbb{E}[\mathrm{SOL}] \le \chi \cdot b_{\widehat{J}}$，我们就称该按钮问题算法具有 χ-一致性；如果无论预测的准确程度如何 $\widehat{J}$，对于所有 J，算法都满足 $\mathbb{E}[\mathrm{SOL}] \le \rho \cdot b_J$，则称其具有 ρ-鲁棒性。

引理 1 指出，按钮问题的下界可直接推广，从而得到多选项滑雪租赁问题几乎相同的下界。

引理 1. 假设存在一个针对多选项滑雪租赁问题的随机化 χ-一致且 ρ-鲁棒的算法，满足 $1 \le \chi \le \rho$。那么，对于所有常数 $\varepsilon \in (0,1)$，都存在一个针对按钮问题的随机化 $(\chi+\varepsilon)$-一致且 $(\rho+\varepsilon)$-鲁棒算法。

下面介绍从多选项滑雪租赁问题到按钮问题的归约算法。设 $\mathcal{A}$ 是用于多选项滑雪租赁问题的 χ-一致、 ρ-鲁棒算法。不失一般性，我们可以假设价格 $\{b_j\}_{j=1,\ \cdots,\ m}$ 均为整数；否则，只需将这些价格乘以一个公分母即可满足这一假设。因此，我们有 $b_1 \ge 1$。

考虑以下用于按钮问题的算法。令 $C := \lceil \rho/\varepsilon \rceil \cdot b_m$。注意到 $C \ge 2$。该算法按如下方式为滑雪租赁问题构造一个实例。令租赁选项的数量 n 为 b_m；对于 $i = 1,\ \cdots,\ n$，设 $(d_i,\ c_i) := (C^i,\ i)$，并令 $\widehat{T} := C^{b_{\widehat{J}}}$。该算法在此构造的实例上内部运行 $\mathcal{A}$。每当 $\mathcal{A}$ 选择一个选项，例如选项 i，我们就点击成本不超过 $c_i = i$ 的 最后一个按钮。（如果不存在这样的按钮，我们就不做任何操作。）如果点击的按钮是目标按钮，我们就向 $\mathcal{A}$ 报告已到达最后一个滑雪日，并终止整个算法。

然而，仅采用这一流程，我们可能永远无法点击目标按钮，因为 $\mathcal{A}$ 可能只选择“便宜”的选项。为了确保算法始终会终止，我们添加如下条件：当 $\mathcal{A}$ 迄今产生的总成本达到或超过 C 时，我们点击最后一个按钮 m 并终止。在这种情况下，我们称算法被 迫使 终止。设 SOL 为所构造算法产生的成本。注意，除非所构造的算法被迫终止，否则 SOL 不会超过 $\mathcal{A}$ 产生的成本。至此，从多选项滑雪租赁问题到按钮问题的归约结束。

引理 1 的证明由下面的引理推出。其证明将在附录 C 中给出 .1

引理 2. 该归约算法是按钮问题的一个随机化 $(\chi+\varepsilon)$-一致 $(\rho+\varepsilon)$-鲁棒算法。

5.2 随机化算法的竞争比

对于随机化算法，我们得到了竞争比的以下下界。结合定理 3，这表明第 4.1 节中给出的算法确实是最优的。

定理 5。 对于任意常数 $\varepsilon > 0$，不存在竞争比为 $e - \varepsilon$ 的随机化算法。

下面我们简要证明该定理。对于任意实例 $\{b_j\}_{j=1,\ \cdots,\ m}$，我们可以构造以下线性规划，其最优值构成任何随机化算法竞争比的下界

适用于按钮问题。

$$
\begin{aligned}
&\min \gamma \\
&\text{s.t. } \textstyle\sum_{j=1}^{m} x_j = 1, \\
&\quad \textstyle\sum_{j=t+1}^{m} y_{t,j} = x_t + \sum_{j=1}^{t-1} y_{j,t}, \\
&\qquad \forall t = 1, \cdots, m-1 \\
&\quad \textstyle\sum_{j=1}^{m} b_j \cdot \left(x_j + \sum_{t=1}^{\min(J,j)-1} y_{t,j} \right) \le \gamma \cdot b_J, \\
&\qquad \forall J = 1, \cdots, m, \\
&\quad x_j \ge 0, \quad \forall j = 1, \cdots, m, \\
&\quad y_{t,j} \ge 0, \quad \forall t = 1, \cdots, m-1, \forall j = t+1, \cdots, m.
\end{aligned}
$$

接下来，我们精心构造一个实例族，并据此求出该线性规划的对偶问题。随后，我们通过解析确定一个对偶可行解，其值收敛于 $(e-\varepsilon)$。由弱对偶性可知，我们可以完成定理 5 的证明。完整证明见附录 C.2。

5.3 确定性算法的一致性与鲁棒性之间的权衡

对于确定性算法，我们考虑常规按钮问题的一个变体，称为 (K, β)-连续按钮问题，其中按钮以连续形式给出，定义在 $[1, m]$ 上，并具有价格函数$b: [1, m] \to \mathbb{R}_+$，该函数满足 b 是一个非递减的 K-Lipschitz 连续函数，且 $b(1) = \beta > 0$。下面的引理说明，我们可以转而考虑这一变体，以获得一致性与鲁棒性之间的权衡。

引理 3. 设 $f:(0,1) \to \mathbb{R}_+$ 是一个连续函数。假设对于所有 $\varepsilon > 0$ 和 $\lambda \in (0, 1)$，任何用于(K, β)-连续按钮问题的确定性$(1+\lambda)$-一致算法，其鲁棒性都必须严格大于 $f(\lambda) - \varepsilon$。那么，对于所有 $\varepsilon > 0$ 和 $\lambda \in (0, 1)$，任何用于常规按钮问题的确定性$(1+\lambda)$-一致算法，其鲁棒性也必须严格大于 $f(\lambda) - \varepsilon$。

我们首先证明，对于连续按钮问题，任何确定性算法都无法达到优于 4 的竞争比；该结论的证明见附录 C.3.1。事实上，Zhang 等人（2011）给出了确定性算法竞争比的相同下界，但我们仍然给出自己的证明，因为它有助于推导出确定性学习增强算法的一致性与鲁棒性之间的以下权衡关系。

定理6。 对于所有常数 $\lambda \in (0、1)$ 和 $\varepsilon > 0$，任何确定性$(1+\lambda)$-一致算法的鲁棒性都必须大于 $2 + \lambda + 1/\lambda - \varepsilon$。

该定理的完整证明放在附录 C.3.2 中。

6 计算评估

实验设置 本节通过计算评估三种学习增强算法的平均竞争比如何随预测 T 偏离 T 而变化：第 3.2 节中的确定性算法（Our-Det）、第 4.2 节中的随机化算法（Our-Rand），以及 Anand 等人的算法（Anand 等人）。对于租赁或购买（即双选项）滑雪租赁问题，Kumar 等人（2018）通过向 T 添加高斯误差来获得预测 T，并测量其算法的竞争比。他们调整高斯分布的标准差，使 T 偏离 T，并针对不同的权衡参数 λ 进行实验。我们的实验采用类似的安排：针对三种算法，测量 $\lambda \in \{0.1, 0.3, 0.5, 0.7\}$ 和 $\sigma \in \{0, 1, \cdots, 50\}$ 的每种组合；每种组合对应的平均竞争比绘于图 1 中。

对于固定的 (λ, σ)，我们生成 10，000 个独立随机实例。与 Kumar 等人（2018年）不同，我们的实验需要多个租赁选项。每个选项的租赁期限和成本均随机生成；滑雪天数 T 是随机选取的，同时考虑租赁期限的最大范围 d^{max}；预测误差 η从 $N(0, \sigma)$ 中采样，并进行适当的截断和舍入操作[2]。

[2]该数　租赁选项的数量设为 $n := 15$ 我们从 $\{1, 2, \cdots, d^{max} := 50\}$ 中均匀采样 n 个数字　at

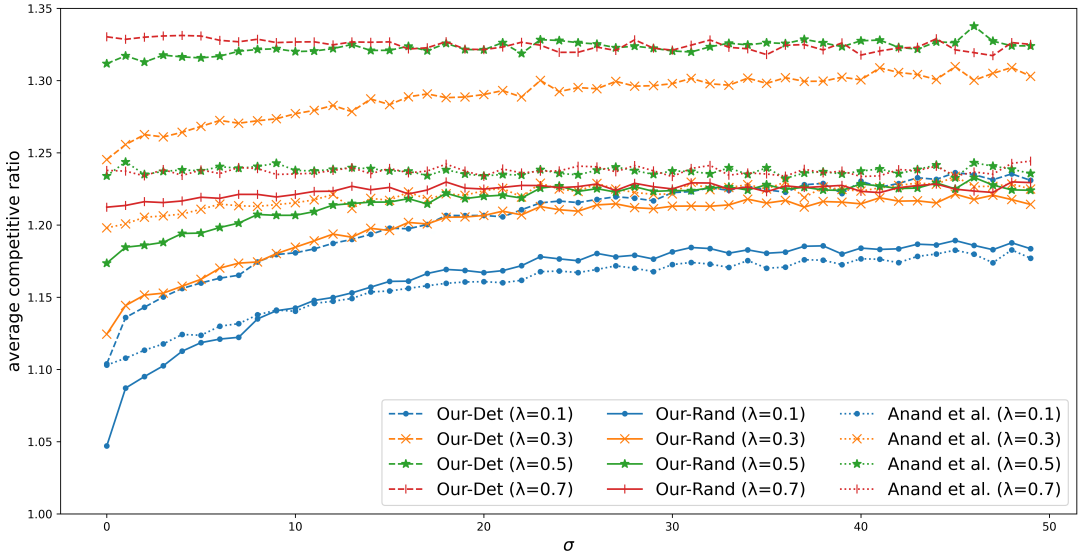

图 1：图中展示了 Our-Det（虚线）、Our-Rand（实线）和 Anand 等人（点线）所达到的平均竞争比随 σ 变化的情况。不同颜色/标记对应不同的权衡参数 λ 值。

讨论 从图 1 可以看出，Our-Rand 通常优于另外两种确定性算法。特别是，当 σ 较小时，Our-Rand 的表现远胜另外两种算法。当$\lambda = 0.7$时，图中的曲线开始显得更加“平坦”，表明这三种算法的行为受预测 T的影响都较小。另一方面，当 $\lambda = 0.1$时，准确预测带来的影响要显著得多。一个有趣的观察是，Our-Det 的表现略逊于 Anand 等人。我们认为，这是因为与 Our-Det 相比，Anand 等人的算法往往更积极地使用较长的租赁选项。

参考文献

L. Ai、X. Wu、L. Huang、L. Huang、P. Tang 和 J. Li。多商店滑雪租赁问题。载于*2014*年 *ACM* 计算机系统测量与建模国际会议，第 463–475 页，2014年。

K. Anand、R. Ge、A. Kumar 和 D. Panigrahi。学习增强型在线算法的回归方法。神经信息处理系统进展（*NeurIPS*），34：30504–30517，2021年。

S. Angelopoulos、C. D¨urr、S. Jin、S. Kamali 和 M. P. Renault。基于不可信建议的在线计算。载于第 *11* 届理论计算机科学创新会议（*ITCS 2020*），第 151 卷，第 52:1–52:15 页。Schloss Dagstuhl - Leibniz-Zentrum f¨ur Informatik，2020年。

A. Antoniadis、C. Coester、M. Elias、A. Polak 和 B. Simon。带有不可信预测的在线度量算法。载于国际机器学习会议 *(ICML)*，第 345–355 页。PMLR，2020年。A. Antoniadis、T. Gouleakis、P. Kleer 和 P. Kolev。秘书问题与在线匹配问题，

无放回随机抽样；令 d_1、$\cdots$、d_n 为抽样得到的数字，并按 $d_1 < \cdots < d_n$ 的顺序排列。接下来，我们从 (0, 1) 中独立地均匀随机抽取 n 个值；令 r_1、$\cdots$、r_n 为满足 $r_1 \leq \cdots \leq r_n$ 的抽样值。我们本可以直接定义 c_i：对所有 i 均为 $= r_i \cdot d_i$，但这可能导致某些 j 出现 $c_j > c_{j-1}$，从而使选项 1 至选项 $j-1$ 变得无用。为防止这种情况，我们按如下方式归纳构造 c_1、$\cdots$、c_n。令 c_1：$= r_1 \cdot d_1$。对于 $i = 2$、$\cdots$、n，若 $r_i \cdot d_i < c_{i-1}$，则将 r_i、$\cdots$、r_n 乘以 $\frac{c_{i-1}}{r_i \cdot d_i}$ 并加上一个小常数 10^{-4}；否则不做任何处理。然后令 c_i：$= r_i \cdot d_i$。这确保了 $\{d_i\}_{i=1}$、$\cdots$、n 以及 $\{c_i\}_{i=1}$、$\cdots$、n 均为非递减的。令 $\{(d_i, c_i)\}_{i=1}$、$\cdots$、n 为租赁选项集合。滑雪天数 T 从 $\{1, \cdots, 10 \cdot d^{max}\}$ 中均匀随机抽取。最后，我们抽取误差 $\eta \sim N(0, \sigma)$，并令 $\widehat{T}$：$=\max\{\lfloor T + \eta \rceil, 1\}$ 为关于 T 的预测。

机器学习建议。神经信息处理系统进展（*NeurIPS*），33:7933–7944，2020年。Y. Azar、D. Panigrahi 和 N. Touitou。带预测的在线图算法。载于*2022*年*ACM-SIAM*离散算法年度研讨会（*SODA*）论文集，第 35–66 页。SIAM，2022年。E. Bamas、A. Maggiori 和 O. Svensson。用于学习增强算法的原始-对偶方法。神经信息处理系统进展（*NeurIPS*），33:20083–20094，2020年。S. Banerjee。通过序贯决策和机器学习预测改进在线租赁或购买算法。神经信息处理系统进展（*NeurIPS*），33:21072–21080，2020年。N. Buchbinder、K. Jain 和 J. Naor。用于最大化广告拍卖收入的在线原始-对偶算法。载于 第*15*届欧洲算法年度研讨会（*ESA 2007*），第 253–264 页。施普林格，2007年。R. Fleischer。论 Bahncard问题。理论计算机科学，268(1):161–174，2001年。S. Gollapudi 和 D. Panigrahi。带专家建议的在线租赁或购买算法。载于国际机器学习会议（*ICML*），第 2319–2327 页。PMLR，2019年。S. Im、R. Kumar、A. Petety 和 M. Purohit。简约学习增强缓存。载于国际机器学习会议（*ICML*），第 9588–9601 页。PMLR，2022年。S. H. Jiang、E. Liu、Y. Lyu、Z. G. Tang 和 Y. Zhang。带预测的在线设施选址。载于第十届国际学习表征会议（*ICLR 2022*）。OpenReview.net，2022年。A. R. Karlin、M. S. Manasse、L. Rudolph 和 D. D. Sleator。竞争性侦听缓存。*Algorithmica*，3:79–119，1988年。A. R. Karlin、M. S. Manasse、L. A. McGeoch 和 S. Owicki。非一致问题的竞争性随机算法。*Algorithmica*，11(6):542–571，1994年。A. R. Karlin、C. Kenyon 和 D. Randall。动态TCP确认及其他关于$e/(e-1)$的故事。载于第三十三届*ACM*计算理论研讨会（*STOC*）论文集，第 502–509 页，2001年。R. Kumar、M. Purohit 和 Z. Svitkina。通过机器学习预测改进在线算法。神经信息处理系统进展（*NeurIPS*），31，2018年。S. Lattanzi、T. Lavastida、B. Moseley 和 S. Vassilvitskii。通过学习得到的权重进行在线调度。载于第十四届*ACM-SIAM*离散算法年度研讨会（*SODA*）论文集，第 1859–1877 页。SIAM，2020年。T. Lavastida、B. Moseley、R. Ravi 和 C. Xu。用于在线匹配、流和负载均衡的可学习且实例鲁棒的预测。载于 第*29*届欧洲算法年度研讨会（*ESA 2021*），第 204 卷，第 59:1–59:17 页。Schloss Dagstuhl - Leibniz-Zentrum f¨ur Informatik，2021年。T. Lykouris 和 S. Vassilvitskii。带机器学习建议的竞争性缓存。*ACM*期刊（*JACM*），68(4):1–25，2021年。A. Meyerson。停车许可证问题。载于第*46*届*IEEE*计算机科学基础研讨会（*FOCS'05*），第 274–282 页。IEEE，2005年。M. Mitzenmacher 和 S. Vassilvitskii。带预测的算法。*ACM*通讯，65 (7):33–35，2022年。D. Rohatgi。带机器学习建议的在线缓存的近最优界。载于第十四届*ACM-SIAM*离散算法年度研讨会（*SODA*）论文集，第 1834–1845 页。SIAM，2020年。S. Wang、J. Li 和 S. Wang。带机器学习建议的多商店滑雪租赁在线算法。神经信息处理系统进展（*NeurIPS*），33:8150–8160，2020年。A. Wei 和 F. Zhang。学习增强在线算法的最优鲁棒性-一致性权衡。神经信息处理系统进展（*NeurIPS*），33:8042–8053，2020年。G. Zhang、C. K. Poon 和 Y. Xu。具有多种折扣选项的滑雪租赁问题。信息处理快报，111(18):903–906，2011年。

第 3 节中的延后证明

为了证明定理 1 和定理 2，我们需要对第 3.1 节中提出的确定性竞争算法作出如下观察。

观察 1. 对于每次迭代 $i \geq 2$，我们有 最优值(τ_i) $\leq \mathrm{SOL}_{i-1} \leq$ 最优值($\tau_i + 1$) 且 $\mathrm{SOL}_i \leq 2\,\mathrm{SOL}_{i-1}$。

证明。 第一组不等式由构造得出。（注意，第二个不等式不严格的唯一原因是存在 $\tau_i = \infty$ 的可能性。）第二个不等式由 b 的定义以及 $\mathrm{SOL}_i = \mathrm{SOL}_{i-1} + b(\mathrm{SOL}_{i-1})$ 这一事实得出。 □

现在我们可以证明这两个主要定理了。

定理7 (定理 1 重述)。 该算法是 *4*-竞争的。

证明。 如果 $T = \tau_1 = 1$，算法将达到最优。如果 $\tau_1 + 1 \leq T \leq \tau_2$，注意 $\mathrm{SOL}_2 \leq 2\mathrm{SOL}_1 = 2\,\mathrm{OPT}(1) \leq 2\,\mathrm{OPT}(T)$。对于 $i \geq 3$，如果 $\tau_{i-1} + 1 \leq T \leq \tau_i$，算法将在第 i 次迭代或更早终止。我们有

$$\mathrm{SOL}_i \leq 2\,\mathrm{SOL}_{i-1} \leq 4\,\mathrm{SOL}_{i-2} \leq 4\,\mathrm{OPT}(\tau_{i-1} + 1) \leq 4\,\mathrm{OPT}(T)$$

根据观察 1。 □

定理 8 (定理 2 重述)。 该算法是一种确定性的最大$\{1+2\lambda, 4\lambda\}$-一致、$\left(2 + \frac{2}{\lambda}\right)$-鲁棒算法。

证明。 首先证明该算法的一致性。如果算法从尊重阶段开始，很容易看出该算法是最优的。因此，从现在开始，我们假设算法从第一忽略阶段开始。如果算法转入尊重阶段，它将在尊重$\widehat{\ }$阶段终止。总成本恰为 $\mathrm{SOL}_{i^\star} + \mathrm{OPT}(T)$。根据观察 1 和公式（1），该成本至多为 $(1 + 2\lambda)\,\mathrm{OPT}(T)$。如果算法在第一忽略阶段期间终止，可以设想我们继续执行算法，直到至少完成第一阶段。在这种情况下，总成本至多为 $\widehat{\mathrm{SOL}}_{i^\star}$，一致性由 $\mathrm{SOL}_{i^\star} \leq 2\lambda\,\mathrm{OPT}(T)$ 得出。

现在考虑算法在第一忽略阶段之后直接进入第二忽略阶段的情况。当 $\mathrm{SOL}_{i^\star} > \mathrm{OPT}(\widehat{T})$ 时，就会发生这种情况。注意，$b(\mathrm{SOL}'_0) = b(\mathrm{SOL}_{i^\star})$ 至少覆盖 $\succ T$ 天，这意味着算法必定在第一次迭代后终止；在这次迭代中，我们追加了 $b(\mathrm{SOL}'_0)$。因此，总成本可以界定为

$$\mathrm{SOL}'_1 = 2\,\mathrm{SOL}_{i^\star} \leq 4\,\mathrm{SOL}_{i^\star - 1} \leq 4\lambda\,\mathrm{OPT}(\widehat{T})$$

根据观察 1 和公式（1）。

现在我们来证明鲁棒性。注意，如果算法没有进入尊重阶段，那么该算法是 4-鲁棒的，因为该算法的执行与我们在第 3.1 节中的 4-竞争算法完全相同。因此，只需考虑算法确实进入尊重阶段的情况。

我们先考虑算法从第一忽略阶段开始，并在尊重阶段终止的情况。根据算法进入尊重阶段这一假设，我们有 $T > \tau_{i^\star}$。注意，

$$\lambda\,\mathrm{OPT}(\widehat{T}) < \mathrm{SOL}_{i^\star} \leq 2\,\mathrm{SOL}_{i^\star - 1} \leq 2\,\mathrm{OPT}(\tau_{i^\star} + 1) \leq 2\,\mathrm{OPT}(T),$$

其中，第一个不等式由公式（1）得出，其余不等式则由观察 1 和 $T \geq \tau_{i^\star} + 1$ 得出。这意味着 $\mathrm{OPT}(\widehat{T}) \leq \frac{2}{\lambda}\mathrm{OPT}(T)$。因此，总成本至多为 $\mathrm{SOL}_{i^\star} + \mathrm{OPT}(\widehat{T}) \leq \left(2 + \frac{2}{\lambda}\right)\mathrm{OPT}(T)$。

现在考虑这样一种情况：算法从第一阶段开始，进入尊重阶段，并在第二忽略阶段终止。假设对于某个 $i \geq 1$，$\tau'_{i-1} + 1 \leq T \leq \tau'_i$。注意，此时算法在第 i 次迭代或更早时终止。由于算法没有在尊重阶段终止，因此有 $T > T = \tau'_0$。如果 $i = 1$，则注意到

$$\mathrm{SOL}'_1 \leq 2\,\mathrm{SOL}'_0 \leq 4\,\mathrm{OPT}(\widehat{T}) \leq 4\,\mathrm{OPT}(\tau'_0 + 1) \leq 4\,\mathrm{OPT}(T)$$

其中，第二个不等式源于这样一个事实：只有当 $\widehat{\mathrm{SOL}}_{i^\star} \leq \mathrm{OPT}(T)$ 时，算法才会进入尊重阶段。如果 $i \geq 2$，请注意

$$\mathrm{SOL}'_i \leq 4\,\mathrm{SOL}'_{i-2} \leq 4\,\mathrm{OPT}(\tau'_{i-1} + 1) \leq 4\,\mathrm{OPT}(T) \tag{5}$$

其中 前两个不等式由观察 阶段 1。

最后，考虑算法从尊重阶段开始的情形。仅当 $\widehat{\text{OPT}}(1) > \lambda \text{OPT}(T)$ 时才会发生这种情况。若 $T \le T$，注意到算法会在尊重阶段终止，且算法产生的总成本至多为

$$\text{OPT}(\widehat{T}) < \frac{1}{\lambda}\text{OPT}(1) \le \frac{1}{\lambda}\text{OPT}(T).$$

若 $T > T$，算法会在第二忽略阶段终止。假设对于某个 i，有 $\tau'_{i-1} + 1 \le T \le \tau'_i$。于是，算法会在第 i 次迭代或更早终止。若 $i = 1$，则有

$$\text{SOL}'_1 \le 2\,\text{SOL}'_0 = 2\,\text{OPT}(\widehat{T}) \le 2\,\text{OPT}(\tau'_0 + 1) \le 2\,\text{OPT}(T).$$

若 $i \ge 2$，公式 (5) 再次成立，证明完成。 □

B 第 4 节中的延后证明

B.1 第 4.1 节中的延迟证明

定理 9 (重述定理 3)。 该算法是一个随机化的 e-竞争算法。

证明。 令 $\text{OPT}(T) = \beta e^{i^\star}$，其中 $i^\star \in \mathbb{Z}_{\ge 0}$ 且 $\beta \in [1, e)$。假设算法进入阶段 $i^\star$；否则，算法的开销只会更低。若 $\alpha \ge \beta$，注意到算法会在此阶段终止，因为根据 b 的定义， $b(\alpha e^{i^\star})$ 覆盖了 T 天数，并且 $\alpha e^{i^\star} \ge \text{OPT}(T)$。另一方面，若我们采样 α 使得 $\alpha < \beta$，算法可能进入下一阶段（$i^\star + 1$）。注意，算法随后会终止，因为 $\alpha e^{i^\star+1} \ge \text{OPT}(T)$。回想一下，在每个阶段 i 中，算法的开销至多为 αe^i。因此我们有

$$\begin{aligned}
\mathbb{E}[\text{SOL}] &\le \int_1^\beta \left(\sum_{i=0}^{i^\star+1} \alpha e^i\right) f(\alpha)d\alpha + \int_\beta^e \left(\sum_{i=0}^{i^\star} \alpha e^i\right) f(\alpha)d\alpha \\
&= \sum_{i=0}^{i^\star} e^i \cdot \int_1^e \alpha f(\alpha)d\alpha + e^{i^\star+1} \cdot \int_1^\beta \alpha f(\alpha)d\alpha \\
&= \frac{e^{i^\star+1} - 1}{e - 1} \cdot (e - 1) + e^{i^\star+1} \cdot (\beta - 1) \\
&= (e^{i^\star+1} - 1) + e^{i^\star+1} \cdot (\beta - 1) \\
&= \beta \cdot e^{i^\star+1} - 1 \le e \cdot \text{OPT}(T).
\end{aligned}$$

□

B.2 第 4.2 节中的延迟证明

定理 10 （重述定理 4）。 对于 $\lambda \in$ （0，1），该算法是 χ（λ）-一致且 （e^λ/λ）-鲁棒的，其中 χ 定义如下：

$$\chi(\lambda) := \begin{cases} 1 + \lambda, & \text{if } \lambda < \frac{1}{e}, \\ (e+1)\lambda - \ln\lambda - 1, & \text{if } \lambda \ge \frac{1}{e}. \end{cases}$$

鲁棒性分析 我们继续证明随机化学习增强算法的鲁棒性。首先证明断言 1、2 和 3。

断言 4 （重述断言 1）。 我们有

$$h(\beta) := \frac{e^{2-r}}{\beta} + \frac{(\ln\beta + r - 1)e^{2-r}}{\lambda\beta} \le \frac{e^\lambda}{\lambda}$$

对于每个 $\beta > 0$。

证明。 我们对 $h(\beta)$ 关于 β 的偏导数求导如下:

$$\frac{\partial h(\beta)}{\partial \beta}=\frac{e^{2-r}}{\lambda}\left(\frac{1-\ln \beta}{\beta^2}-\frac{\lambda+r-1}{\beta^2}\right)=\frac{e^{2-r}}{\lambda \beta^2} \cdot(2-\lambda-r-\ln \beta),$$

这意味着 $h(\beta)$ 在 $\beta=e^{2-\lambda-r}$ 处取得 $\frac{e^{\lambda}}{\lambda}$ 的最大值。 □

断言 5 (重述命题 2)。 我们有

$$h(\beta):=\frac{e^{1-r}}{\beta}+\frac{e^{1-r}(\ln \beta+r)}{\lambda \beta} \leq \frac{e^{\lambda}}{\lambda}$$

对每个 $\beta>0$ 都成立。

证明。 计算 $h(\beta)$ 关于 β 的偏导数，可得

$$\frac{\partial h(\beta)}{\partial \beta}=\frac{e^{1-r}}{\lambda}\left(\frac{1-\ln \beta}{\beta^2}-\frac{\lambda+r}{\beta^2}\right)=\frac{e^{1-r}}{\lambda \beta^2} \cdot(1-\lambda-r-\ln \beta),$$

因此， $h(\beta)$ 在 $\beta=e^{1-\lambda-r}$ 处取得 $\frac{e^{\lambda}}{\lambda}$ 的最大值 □

断言 6 (重述命题 3)。 我们有

$$e(x-\ln x) \leq \frac{e^x}{x}$$

对每个 $x>0$ 均成立。

证明。 令 $h(x):=\frac{e^x}{x}-ex+e\ln x$。只需证明，对每个 $x>0$， $h(x)\geq 0$。注意到

$$h'(x)=\frac{e^x(x-1)}{x^2}-e+\frac{e}{x}=\frac{(x-1)(e^x-ex)}{x^2},$$

这意味着 $h(x)$ 在 $x=1$ 处取得最小值 0。 □

接下来分析 $\mathrm{OPT}(T)\geq e^{k-q-1}$ 的情形。

情形 4. $\lambda \mathrm{OPT}(\widehat{T})=e^{k-q-r} \leq \mathrm{OPT}(T)<\mathrm{OPT}(\widehat{T})=e^k$. 值得注意的是，在这种情况下，该算法 $\widehat{}$ 的期望成本至多为预测准确时的成本，即 $T=T$。因此，一致性分析中得到的界也可用于这种情况。对于 $q\geq 1$，根据公式（3），我们可以得到这种情况下鲁棒性比率的上界如下:

$$\frac{\mathbb{E}[\mathrm{SOL}]}{\mathrm{OPT}(T)} \leq \frac{e^k+e^{k-q-r}}{e^{k-q-r}}=1+\frac{1}{\lambda} \leq \frac{e^{\lambda}}{\lambda}$$

其中，等式由 $\lambda=e^{-q-r}$ 的定义得出，最后一个不等式成立是因为对于每个 x，都有 $1+x\leq e^x$。

若 $q=0$，根据方程 (4)，鲁棒性比率可被界定为

$$\frac{\mathbb{E}[\mathrm{SOL}]}{\mathrm{OPT}(T)} \leq \frac{e^{k-r}+e^{k+1-r}+e^k(r-1)}{e^{k-r}}=1+e+\frac{r-1}{\lambda}$$

其中，等式由 $\lambda=e^{-r}$的定义得出。在此我们声称，对于这种情况下的 λ，右侧仍可被界定为 $\frac{e^{\lambda}}{\lambda}$，正如所需。回顾一下，$\lambda \in \left[\frac{1}{e}, 1\right)$ 且 $r=-\ln \lambda$。

命题 7. 对于 $x \in \left[\frac{1}{e}, 1\right]$，我们有

$$1+e-\frac{\ln x+1}{x} \leq \frac{e^x}{x}.$$

证明。 令 $h(x):=e^x-(1+e)x+\ln x+1$。只需证明 $h(x)\geq 0$ 对 $x \in \left[\frac{1}{e}, 1\right]$成立。对其求导，可得

$$h'(x)=e^x-(1+e)+\frac{1}{x}=\left(e^x+\frac{1}{x}\right)-(e+1).$$

注意， $h'(x)$ 在 $x > 0$ 上也是凸函数。此外，我们有

$$h'\left(\frac{1}{e}\right) = e^{e^{-1}} - 1 > 0,$$

连同 $h'(1) = 0$。因此我们可以得出结论： $h(x)$ 在 $\left[\frac{1}{e}, 1\right]$ 上的最小值必定是 $h\left(\frac{1}{e}\right)$ 或 $h(1)$。注意到 $h(1) = 0$，并且

$$h\left(\frac{1}{e}\right) = e^{e^{-1}} - \frac{1+e}{e} + \ln\left(\frac{1}{e}\right) + 1 = e^{e^{-1}} - \left(1 + \frac{1}{e}\right) \geq 0$$

其中，不等式是由对每个 x 都有 $e^x \geq 1 + x$ 得出的。 □

情况5. 最优值$(T) >$ 最优值$(\widehat{T}) = e^k$. 设最优值$(T) := \beta \cdot e^{k+p}$，其中 $p \in \mathbb{Z}_{\geq 0}$ 且 $\beta \in [1, e)$。我们首先考虑 $q \geq 1$ 的情况。注意，在算法到达阶段 k 之前，根据公式（3），算法的期望开销至多为 $e^k + e^{k-q-r}$。另请注意，从阶段 k 开始，我们再次遵循 $\mathcal{A}$。总计，我们有

$$\begin{aligned}\mathbb{E}[\mathrm{SOL}] &\leq e^k + e^{k-q-r} + \int_1^e \left(\sum_{i=k}^{k+p} \alpha \cdot e^i\right) f(\alpha) d\alpha + \int_1^\beta \alpha \cdot e^{k+p+1} f(\alpha) d\alpha \\ &= \beta \cdot e^{k+p+1} + e^{k-q-r},\end{aligned}$$

这意味着该情况的鲁棒性比率可以界定为

$$\frac{\beta \cdot e^{k+p+1} + e^{k-q-r}}{\beta \cdot e^{k+p}} = e + \frac{\lambda}{\beta e^p} \leq e + \lambda$$

其中，等号由 $\lambda = e^{-q-r}$ 的定义得出，而不等式可由 $\beta \geq 1$ 和 $p \geq 0$ 得出。现在，通过以下断言即可完成证明。回想一下，在这种情况下 $\lambda < \frac{1}{e}$。

断言8. 对于 $x \in \left(0, \frac{1}{e}\right]$，我们有

$$x + e \leq \frac{e^x}{x}.$$

证明. 注意， $\frac{e^x}{x}$ 在 $x \in$ 上递减 $(0, 1]$，而 $x + e$ 在 $x \in \mathbb{R}$ 上递增。直接计算可得

$$\frac{e^{e^{-1}}}{e^{-1}} > e + \frac{1}{e}$$

如所述。 □

最后，假设 $q = 0$，即 $\lambda = e^{-r}$。注意，在这种情况下，算法执行 $\widehat{\ }$，这与 $\mathcal{A}$ 几乎完全相同，唯一的例外是：在阶段 $(k-1)$，如果 $\alpha \geq e^{1-r}$，算法会附加 OPT(T)，而不是 b $(\alpha \cdot e^{k-1})$。因此，相应地调整定理 3 的证明后，我们可以推出

$$\begin{aligned}\mathbb{E}[\mathrm{SOL}] &\leq \int_1^e \left(\sum_{i=0}^{k-2} \alpha \cdot e^i\right) f(\alpha) d\alpha + \int_1^{e^{1-r}} \alpha \cdot e^{k-1} f(\alpha) d\alpha + \int_{e^{1-r}}^e e^k f(\alpha) d\alpha \\ &\quad + \int_1^e \left(\sum_{i=k}^{k+p} \alpha \cdot e^i\right) f(\alpha) d\alpha + \int_1^\beta \alpha \cdot e^{k+p+1} f(\alpha) d\alpha \\ &\leq e^{k-r} + r \cdot e^k + \beta \cdot e^{k+p+1} - e^k \\ &= \beta \cdot e^{k+p+1} + (\lambda + r - 1) \cdot e^k.\end{aligned}$$

因此，我们可以得到此情形下鲁棒性比率的上界，如下所示：

$$\frac{\mathbb{E}[\mathrm{SOL}]}{\mathrm{OPT}(T)} \leq e + \frac{\lambda + r - 1}{\beta e^p} \leq e + \lambda + r - 1$$

其中最后一个不等式是因为 $\beta \geq 1$ 和 $p \geq 0$。注意，下一个命题完成了这一情形的证明。回顾一下， $r = -\ln \lambda$。

命题 9。我们有

$$x - \ln x + e - 1 \leq \frac{e^x}{x}$$

对于任意 $x > 0$。

证明。令 $h(x) := \frac{e^x}{x} - x + \ln x$。只需证明对于所有 $x > 0$, $h(x) \geq e - 1$。注意,

$$h'(x) = \frac{e^x(x-1)}{x^2} - 1 + \frac{1}{x} = \frac{(e^x - x)(x-1)}{x^2},$$

这意味着 $x > 0$ 的 $h(x)$ 的最小值为 $h(1) = e - 1$。 □

C 第 5 节中的延后证明

C.1 第 5.1 节中的延后证明

引理 4（引理 2 重述）。该归约算法是针对按钮问题的随机化（$\chi+\varepsilon$）-一致（$\rho+\varepsilon$）-鲁棒算法。

证明。我们先证明该算法是 $(\rho+\varepsilon)$-鲁棒的。令 k为第一个目标按钮的价格，即 $k = b_J$。现在考虑在 $T = C^k$下执行 $\mathcal{A}$的情形。容易看出，$\mathrm{OPT}(T) = k = b_J$，其中 $\mathrm{OPT}(T)$ 是覆盖 T 天数的最优解的成本。注意，如果 $k = 1$，那么归约算法会在 $\mathcal{A}$ 选择任意选项后立即终止。另一方面， $\mathcal{A}$ 选择的第一个选项的期望成本至多为 ρ ，因为即使在 $T = 1$ （且 $\mathrm{OPT}(T) = 1$）时， $\mathcal{A}$ 也必须是 ρ-竞争的。这意味着所构造的按钮问题算法具有 ρ-鲁棒性。因此，从现在起我们假设 $k \geq 2$。这意味着 $n = b_m \geq b_J \geq 2$。

我们声称，如果 $\mathcal{A}$ 仅使用选项 1 到 $(k-1)$ 覆盖 T天数，那么 $\mathcal{A}$ 产生的总成本 $c(\mathcal{A})$ 至少为 C。为说明这一点，令 a_i 为 $\mathcal{A}$ 选择选项 i的次数，其中 $i = 1, \cdots, k-1$。由于 $\mathcal{A}$ 的输出覆盖了 T 天数，因此我们有

$$\sum_{i=1}^{k-1} C^i \cdot a_i \geq T = C^k,$$

从而

$$\sum_{i=1}^{k-1} \frac{k-1}{C^{k-1-i}} \cdot a_i \geq C \cdot (k-1).$$

现在可以如下证明我们的论断:

$$c(\mathcal{A}) = \sum_{i=1}^{k-1} i \cdot a_i \geq \sum_{i=1}^{k-1} \frac{k-1}{C^{k-1-i}} \cdot a_i \geq C \cdot (k-1) \geq C$$

其中，第一个不等式成立，是因为 $h(x) := (k-1)/C^{k-1-x}$ 是一个凸函数，满足 $h(1) = (k-1)/C^{k-2} \leq 1$ 且 $h(k-1) = k-1$。回忆一下， $k \geq 2$ 且 $C \geq 2$。

上述论断意味着，如果 $c(\mathcal{A}) < C$， $\mathcal{A}$ 选择了一个成本至少为 $k = b_J$ 的选项。因此，该构造构造的算法在这种情况下不会被迫终止，因此我们有

$$\mathrm{SOL} \leq c(\mathcal{A}). \tag{6}$$

另一方面，如果 $c(\mathcal{A}) \geq C$，构造的算法就会被迫以额外的价格终止，该价格为

$$b_m = n \leq \frac{\varepsilon}{\rho} \cdot C \leq \frac{\varepsilon}{\rho} \cdot c(\mathcal{A}), \tag{7}$$

从而得到

$$\mathrm{SOL} \leq \left(1 + \frac{\varepsilon}{\rho}\right) c(\mathcal{A}). \tag{8}$$

方程（6）和（8）共同表明，无论 $c(\mathcal{A})$ 是否 $< C$， $\mathrm{SOL} \leq \left(1 + \frac{\varepsilon}{\rho}\right) \cdot c(\mathcal{A})$ 都成立。因此，我们有

$$\mathbb{E}[\mathrm{SOL}] \leq \left(1 + \frac{\varepsilon}{\rho}\right) \cdot \mathbb{E}[c(\mathcal{A})] \leq (\rho + \varepsilon) \cdot b_J, \tag{9}$$

其中 s　　第二个不等式由以下事实推导得出：$\mathcal{A}$ 是 ρ-ro　　棒。

为说明所构造的算法是（$\chi+\varepsilon$）一致的，注意到，由于 $\chi \le \rho$，根据方程（7）有 $b_m \le (\varepsilon/\chi)\cdot c(\mathcal{A})$。利用 $\mathcal{A}$ 对于 $\widehat{T}=C^{b_{\widehat{J}}}$ 是 χ-一致这一事实调整公式（9），即可得到所需的 $\mathbb{E}[\mathrm{SOL}] \le (\chi+\varepsilon)\cdot b_{\widehat{J}}$。

最后，需要指出的是，上述论证仅在 ρ 有限时成立。假设 $\mathcal{A}$ 不鲁棒（即 $\rho=\infty$）。在这种情况下，我们也无需考虑所构造算法的鲁棒性。令 $C := \lceil \chi/\varepsilon \rceil \cdot b_m$，可以看出，上述关于一致性的论证仍然成立。 □

C.2 第 5.2 节中的延后证明

定理 11 (定理 5 重述)。 对于任意常数 $\varepsilon > 0$，不存在随机化算法能够达到 $e-\varepsilon$ 的竞争比。

证明。根据 引理 1，只需给出一个按钮问题的实例族，使得
任何算法的
竞争比至少为 $(e-\varepsilon)$，其中给定常数为 $\varepsilon>0$。
现在我们引入一组用于定义实例的参数。选择这些参数的顺序很重要，我们将在证明末尾讨论这一点。现在，令 δ 为一个足够大的数；直观地说， δ 对应于按钮价格的粒度。参数 c 满足$c/\delta \ge -\ln($ $e-\varepsilon)$ $-\ln\ln(e/(e-\varepsilon))$。选择整数 m，使得 $m>c$且 $m/\delta \ge \ln(e/\varepsilon)$。考虑由这些参数定义的以下按钮问题实例：按钮数量为 m，其价格为 $b_j := e^{j/\delta}$，其中 $j=1,\ \cdots,\ m$。

固定任意一个按钮问题算法。不失一般性，我们可以假设算法点击的按钮索引严格递增。也就是说，如果算法在某个时刻点击按钮 j，那么它会在接下来的轮次点击按钮 $j'>j$。暂且假设只有最后一个按钮 m是目标：即 $J=m$。对于 $j=1,\ \cdots,\ m$，令 x_j为算法在第一轮点击按钮 j的边缘概率。对于 $t=1,\ \cdots,\ m-1$以及 $j=t+1,\ \cdots,\ m$，令$y_{t,\ j}$为算法在某一轮点击按钮 t，并在紧接着的下一轮点击按钮 j 的边缘概率。

注意， x_j不依赖于 J：算法在不了解 J 的情况下就会选择第一个按钮。同样，对于所有 $t<J$，$y_{t,\ j}$ 也不依赖于 J。直观地说，对于任意$J_1<J_2$，算法针对 $J=J_2$ 的执行“前缀”与针对 $J=J_1$ 的执行相同，因为算法不了解 J。

现在，我们将这些 x和 y写成线性规划中的变量，并用约束来描述算法必须满足的性质。我们可以在假设 $J=m$的情况下写出这些约束，因为只需取一个“前缀”，就能得到 $J\neq m$ 情况下的边际概率。以下线性规划的值是该算法竞争比的下界。

$$
\begin{aligned}
&\text{minimize } \gamma \\
&\text{subject to } \sum_{j=1}^{m} x_j = 1, \\
&\qquad \sum_{j=t+1}^{m} y_{t,j} = x_t + \sum_{j=1}^{t-1} y_{j,t}, && \forall t = 1,\cdots,m-1 \\
&\qquad \sum_{j=1}^{m} b_j \cdot \left(x_j + \sum_{t=1}^{\min(J,j)-1} y_{t,j} \right) \le \gamma \cdot b_J, && \forall J = 1,\cdots,m, \\
&\qquad x_j \ge 0, && \forall j = 1,\cdots,m, \\
&\qquad y_{t,j} \ge 0, && \forall t = 1,\cdots,m-1, \forall j = t+1,\cdots,m.
\end{aligned}
$$

注意，第一项约束必须得到满足，因为 x 必须构成一个概率分布。注意，第二项约束的两边都是表示算法点击按钮 t 的边缘概率的一种方式（回顾我们假设 $J=m$）。因此，这些等式必须得到满足。注意， $x_j+\sum^{\min(J',j)}{}_{t=1}$　　$^{-1}\, y_{t,\ j}$ 是在 $J=J'$ 时按钮 j 被点击的边缘概率：我们使用 $\min(J',$ $j)-1$ 来确保取适当的前缀。这表明，只要 γ 不小于真实的竞争比，第三项约束就必须得到满足。

为了计算该线性规划的最坏情况值，我们考虑其对偶问题，如下所示：

$$
\begin{aligned}
&\text{最大化 } w \\
&\text{subject to } \sum_{j=1}^{m} b_j v_j = 1, \\
&\qquad w \le u_t + b_t \sum_{j=1}^{m} v_j, && \forall t = 1, \cdots, m-1 \\
&\qquad w \le b_m \sum_{j=1}^{m} v_j, && \text{(D1)} \\
&\qquad u_s - u_t \le b_t \sum_{j=s+1}^{m} v_j, && \forall s = 1, \cdots, m-2, \forall t = s+1, \cdots, m-1, \\
&\qquad u_s \le b_m \sum_{j=s+1}^{m} v_j, && \forall s = 1, \cdots, m-1, \\
&\qquad w \in \mathbb{R}, \\
&\qquad u_t \in \mathbb{R}, && \forall t = 1, \cdots, m-1, \\
&\qquad v_j \ge 0, && \forall j = 1, \cdots, m.
\end{aligned}
$$

我们希望找到一个可行解，使其满足 (D1) 且目标值接近 $e - \varepsilon$。为此，我们考虑以下辅助线性规划。

$$
\begin{aligned}
&\text{maximize } w \\
&\text{subject to } w \le u_t + b_t \sum_{j=1}^{m} v_j, && \forall t = 1, \cdots, m, \\
&\qquad u_s - u_t \le b_t \sum_{j=s+1}^{m} v_j, && \forall s = 1, \cdots, m-1, \forall t = s+1, \cdots, m, && \text{(D2)} \\
&\qquad u_m = 0, \\
&\qquad w \in \mathbb{R}, \\
&\qquad u_t \in \mathbb{R}, && \forall t = 1, \cdots, m, \\
&\qquad v_j \ge 0, && \forall j = 1, \cdots, m.
\end{aligned}
$$

注意，通过将 (D1) 中的每个变量除以 $\sum_{j=1}^{m} b_j v_j$，一旦我们获得 (D2) 的一个可行解，就可以轻松构造出一个可行解，o

我们按如下方式构造 (D2) 的一个可行解：对于所有 $j = 1, \cdots, m$，

$$
\begin{aligned}
v_j &:= \int_{(j-1)/\delta}^{j/\delta} e^{-z} dz, \\
u_j &:= \frac{e - \varepsilon}{\delta} \left(m - c - j\right)_+, \text{ and} \\
w &:= \min_{t=1,\cdots,m} \left\{ u_t + e^{t/\delta} \sum_{j=1}^{m} v_j \right\},
\end{aligned}
$$

其中 $(\cdot)_+ := \max\{\cdot, 0\}$。注意，选择 w 可满足 (D2) 的第一组约束。显然，第三组和第四组约束也都得到满足。下一个引理将证明该解满足第二组约束。

引理 5. 如果 $c/\delta \ge -\ln(e - \varepsilon) - \ln\ln(e/(e - \varepsilon))$，则有

$$
u_s - u_t \le b_t \sum_{j=s+1}^{m} v_j
$$

对于任意 $s < t$。

证明。 我们先考虑 $s < t \le m - c$ 的情形。注意到

$$u_s - u_t = (e - \varepsilon) \cdot \frac{t - s}{\delta} \quad \text{and}$$
$$b_t \sum_{j=s+1}^{m} v_j = e^{t/\delta} \int_{s/\delta}^{m/\delta} e^{-z} dz = e^{(t-s)/\delta} - e^{(t-m)/\delta}.$$

我们将证明

$$(e - \varepsilon) \cdot \frac{t - s}{\delta} \le e^{(t-s)/\delta} - e^{-c/\delta} \le e^{(t-s)/\delta} - e^{(t-m)/\delta},$$

其中，最后一个不等式由 $t \le m - c$ 得出，从而证明了此情形下的引理。代入$z := (t - s)/\delta$ 并重新整理各项后，只需证明对于任意 $z > 0$,

$$e^z - (e - \varepsilon) \cdot z \ge e^{-c/\delta}.$$

对左侧关于 z 求导，可以很容易地看出，当 $z = \ln(e - \varepsilon)$ 时，左侧取得最小值。注意到

$$e^{-c/\delta} \le (e - \varepsilon) - (e - \varepsilon) \ln(e - \varepsilon)$$

根据引理中的条件，即可证明该结论。

对于 $s < m - c \le t$，我们有

$$u_s - u_t = (e - \varepsilon) \cdot \frac{m - c - s}{\delta} \quad \text{and}$$
$$b_t \sum_{j=s+1}^{m} v_j = e^{t/\delta} \left(e^{-s/\delta} - e^{-m/\delta} \right).$$

注意，对于固定的 s，当 $t = m - c$ 时， $b_t \sum_{j=s+1}^{m} v_j$ 取最小值，而 $u_s - u_t$ 不受 t 影响。因此，只需证明当 $t = m - c$ 时引理成立。这已在前一种情形中证明。

最后，对于 $m - c \le s < t$，我们有 $u_s - u_t = 0$。 □

下面的引理给出了 w 的下界。

引理 6. 如果 $m/\delta \ge \ln(e/\varepsilon)$，则有

$$h(t) := u_t + e^{t/\delta} \sum_{j=1}^{m} v_j \ge \frac{e - \varepsilon}{\delta} (m - c)$$

对于任意 t。

证明。 回顾一下，对于 $t \ge m - c$， $u_t = 0$。因此，只需考虑 $t \le m - c$。我们有 e

$$h(t) = \frac{e - \varepsilon}{\delta} (m - c - t) + e^{t/\delta} \int_{0}^{m/\delta} e^{-z} dz$$
$$= \frac{e - \varepsilon}{\delta} (m - c - t) + e^{t/\delta} \left(1 - e^{-m/\delta} \right).$$

对 $h(t)$ 关于 t 求偏导数，可得

$$\frac{\partial h(t)}{\partial t} = -\frac{e - \varepsilon}{\delta} + \frac{e^{t/\delta}}{\delta} \cdot \left(1 - e^{-m/\delta} \right).$$

这意味着，当 $t = t^\star$ 且 $t^\star$ 满足以下条件时， $h(t)$ 取得最小值

$$e^{t^\star/\delta} = \frac{e - \varepsilon}{1 - e^{-m/\delta}} \le e,$$

其中，不等式由 $m/\delta \ge \ln(e/\varepsilon)$ 得出。这里我们还可以观察到 $t^\star \le \delta$。现在我们有

$$h(t^\star) = \frac{e - \varepsilon}{\delta} (m - c - t^\star) + (e - \varepsilon) \ge \frac{e - \varepsilon}{\delta} (m - c),$$

其中，不等式可由 $t^\star \le \delta$ 推导得出。 □

根据这些引理，我们可以得出结论：$(w,\ u,\ v)$ 是 (D2) 的一个可行解，其目标值至少为 $\frac{e-\varepsilon}{\delta}(m-c)$。我们已经论证，通过将每个变量都除以 $\sum_{j=1}^{m} b_j v_j$ 进行归一化，可以推导出 (D1) 的一个可行解。下一个引理表明，$\sum_{j=1}^{m} b_j v_j$ 足够小。

引理 7。有 $\sum_{j=1}^{m} b_j v_j \le (m/\delta)\cdot e^{1/\delta}$。

证明。注意到

$$\begin{aligned}\sum_{j=1}^{m} b_j v_j &= \sum_{j=1}^{m} e^{j/\delta}\int_{(j-1)/\delta}^{j/\delta} e^{-z}dz \\ &= e^{1/\delta}\sum_{j=1}^{m} e^{(j-1)/\delta}\int_{(j-1)/\delta}^{j/\delta} e^{-z}dz \\ &\le e^{1/\delta}\sum_{j=1}^{m}\int_{(j-1)/\delta}^{j/\delta} e^{z}\cdot e^{-z}dz \\ &= \frac{m}{\delta}\cdot e^{1/\delta}.\end{aligned}$$

□

因此，我们可以得到一个满足 (D1) 的解，其目标值至少为

$$\frac{(e-\varepsilon)\cdot(m/\delta - c/\delta)}{(m/\delta)\cdot e^{1/\delta}}. \tag{10}$$

给定 ε，我们先将 c/δ 固定为满足 $c/\delta \ge -\ln(e-\varepsilon) - \ln\ln(e/(e-\varepsilon))$ 的某个常数。然后，我们选择足够大的 δ，由此确定 c。之后，我们选择足够大的 m。这表明，方程（10）可以任意接近 $(e-\varepsilon)$。现在，由线性规划的弱对偶性可得所需结论，定理 5 的证明至此完成。

□

C.3 第 5.3 节中的延后证明

引理 8。设 $f:(0,1)\to\mathbb{R}_+$ 是一个连续函数。假设对于所有 $\varepsilon>0$ 和 $\lambda\in(0,1)$，任何针对$(1+\lambda)$-一致性算法的确定性算法，在(K,β)-连续按钮问题上的鲁棒性都必须严格大于 $f(\lambda)-\varepsilon$。那么，对于所有 $\varepsilon>0$ 和 $\lambda\in(0,1)$，任何$(1+\lambda)$-一致性算法的确定性算法，在常规按钮问题上的鲁棒性也必须严格大于 $f(\lambda)-\varepsilon$。

证明。反设，对于某个 $\lambda\in(0,1)$ 和 $\varepsilon>0$，存在一个用于常规按钮问题的确定性 $(1+\lambda)$-一致且 $(f(\lambda)-\varepsilon)$-鲁棒算法 $\mathcal{A}$。不失一般性，假设 $\varepsilon<f(\lambda)$。通过选择足够小的 $\varepsilon'>0$，我们可以满足 $\varepsilon'<\varepsilon/3$，$\lambda+\varepsilon'\in(0,1)$，并且 $f(\lambda+\varepsilon')>f(\lambda)-\varepsilon/3$。令 $F:=\max\{2,\sup_{\lambda\in(0,1)} f(\lambda)\}$。我们选择 $\delta\in(0,(\varepsilon'\beta)/(FK)]$，使得 $(m-1)/\delta$成为一个整数。

现在考虑用于 (K,β)-连续按钮问题的以下算法。令 b 为给定的价格函数，$J\in[1,m]$ 为预测。该算法通过创建 $((m-1)/\delta+1)$ 个按钮来构造一个常规按钮问题实例，这些按钮的值为 $b((i-1)\delta+1)$，其中 $i=1,\cdots,(m-1)/\delta+1$。然后，算法在此构造的实例上内部执行 $\mathcal{A}$，并为其提供预测$\tilde{J}:=\left\lceil(\widehat{J}-1)/\delta\right\rceil+1$。每当 $\mathcal{A}$ 点击一个按钮，例如 b_k，该算法便会点击连续问题中相应的按钮 $(k-1)\delta+1$。当算法点击到一个目标时，它会向 $\mathcal{A}$ 报告最后一个按钮是目标，然后终止。请注意，该算法产生的总成本恰好等于 $\mathcal{A}$产生的总成本。

我们声称，对于$(K,\ \beta)$-连续按钮问题，该算法是$(1+\lambda+\varepsilon')$-一致的，并且是$(f(\lambda)-(2\varepsilon)/3)$-鲁棒的。由于 $f(\lambda)-(2\varepsilon)/3<f(\lambda+\varepsilon')-\varepsilon/3$，这会导致矛盾。剩下需要验证一致性和鲁棒性。

假设当连续按钮问题的第一个目标 J 等于 $\widehat{J}$。请注意，当 $\mathcal{A}$ 点击索引为 J 或更高的按钮时，算法将终止。这意味着，提供给 $\mathcal{A}$ 的预测 $\tilde{J}$ 也是准确的（对于常规按钮问题）。从（$1+\lambda$）一致性

关于 $\mathcal{A}$, 算法产生的总成本为 most

$$\begin{aligned}(1+\lambda)b_{\tilde{J}} &= (1+\lambda)b\left(\left\lceil (\widehat{J}-1)/\delta \right\rceil \cdot \delta + 1\right)\\ &\le (1+\lambda)(b(\widehat{J}) + K\delta)\\ &\le (1+\lambda+\varepsilon')b(\widehat{J}),\end{aligned}$$

其中，第一个不等式由 b 的 K-Lipschitz 连续性推出，第二个不等式则由 b 的单调性以及 ε' 的选取推出。这表明该算法具有 $(1+\lambda+\varepsilon')$-一致性。

现在我们验证其鲁棒性。令 J 为连续按钮问题中的第一个目标按钮。当 $\mathcal{A}$ 点击索引为 $\lceil (J-1)/\delta \rceil + 1$ 或更高的按钮时，算法终止。因此，根据 $\mathcal{A}$ 的 $(f(\lambda)-\varepsilon)$-鲁棒性，该算法产生的总成本至多为

$$\begin{aligned}(f(\lambda)-\varepsilon)b\left(\lceil (J-1)/\delta \rceil \cdot \delta + 1\right) &\le (f(\lambda)-\varepsilon)(b(J)+K\delta)\\ &\le (f(\lambda)-\varepsilon+\varepsilon')b(J),\end{aligned}$$

其中，第一个不等式同样由 b 的 K-Lipschitz 连续性推出，第二个不等式则由 b 的单调性以及 ε' 的选取推出。注意， $f(\lambda)-\varepsilon+\varepsilon' < f(\lambda)-(2\varepsilon)/3$，这表明该算法具有所需的鲁棒性。

□

C.3.1 竞争比下界

在介绍确定性学习增强算法的一致性与鲁棒性之间的权衡之前，我们先证明，对于 $(K,\ \beta)$-连续按钮问题，任何确定性算法的竞争比都不可能严格优于 4。令 m 为一个足够大的数，并令 $b(j) = j$for $j \in [1, m]$。注意， b 是 1-Lipschitz 连续的，且 $b(1) = 1 > 0$。

固定一个任意的连续按钮问题确定性算法。令 n 为该算法点击的按钮数；对于任意 $i = 1,\ \cdots,\ n$，令 x_i 为算法在第 i 轮点击的按钮。注意，根据 b的选取，该按钮的价格也为 x_i。此外，不失一般性，我们可以假设 $x_1 < x_2 < \cdots < x_n$。

为了使算法具有竞争力，它必须对任意 $J \in \{1, x_1+\varepsilon, x_2+\varepsilon,\ \cdots,\ x_{n-1}+\varepsilon\}$ 都具有竞争力，其中 ε 是一个无穷小量。因此，我们得到如下无限约束序列：对于任意整数 $i = 0,\ 1,\ 2,\ \cdots,\ n-1$,

$$\sum_{j=1}^{i+1} x_j \le \gamma x_i, \tag{11}$$

其中， γ 表示该算法的竞争比，且 $x_0 := 1$。

令 $\{a_i\}$ 为按以下方式递归定义的无限序列：

$$a_1 = \gamma - 1 \text{ and } a_i = \gamma - \frac{\gamma}{a_{i-1}} \text{ for } i \ge 2.$$

注意，该无限序列仅取决于 γ。

我们用归纳法证明，对于任意 $i = 1,\ \cdots,\ n-1$，都有

$$x_{i+1} \le a_i x_i \text{ and } \sum_{j=1}^{i+1} x_j \ge \frac{\gamma}{a_i} \cdot x_{i+1}. \tag{12}$$

对于 $i = 1$，容易看出，由方程（11）可得 $x_2 \le (\gamma-1)x_1$，其中 $i = 1$。此外 ,

$$x_1 + x_2 \ge \frac{1}{\gamma-1} \cdot x_2 + x_2 = \frac{\gamma}{a_1} \cdot x_2,$$

其中，第一个不等式来自 $x_2 \le a_1 x_1$，这证明了该命题的后半部分。对于 $i \ge 2$，我们有

$$x_{i+1} \le \gamma x_i - \sum_{j=1}^{i} x_j \le \left(\gamma - \frac{\gamma}{a_{i-1}}\right) x_i = a_i x_i$$

其中，第一个不等式来自方程（11），第二个则由归纳假设得出。我们还有

$$\begin{aligned}\sum_{j=1}^{i+1} x_j &\geq \frac{\gamma}{a_{i-1}} \cdot x_i + x_{i+1} \\ &\geq \frac{\gamma}{a_{i-1}} \cdot \frac{1}{a_i} \cdot x_{i+1} + x_{i+1} \\ &= \left(\frac{\gamma}{a_{i-1}} \cdot \frac{1}{\gamma - \frac{\gamma}{a_{i-1}}} + 1 \right) x_{i+1} \\ &= \frac{\gamma}{a_i} \cdot x_{i+1},\end{aligned}$$

根据 $\{a_i\}_i$ 的递推关系。至此，命题得证。

接下来的两个引理展示了 $\{a_i\}_i$ 的一些有用性质。这些性质对于证明确定性算法的竞争比下界至关重要。

引理9。 如果序列 $\{a_i\}_i$ 收敛，则有 $\gamma \geq 4$。

证明。 令 $\alpha := \lim_{i\to\infty} a_i$。根据递推关系，我们可以得到

$$\alpha = \gamma - \frac{\gamma}{\alpha} \iff \frac{\alpha^2 - \gamma\alpha + \gamma}{\alpha} = 0.$$

为了得到一个实数解，我们应有 $\gamma^2 - 4\gamma = \gamma(\gamma - 4) \geq 0$。 □

引理10。 如果 $1 \leq \gamma < 4$，则序列 $\{a_i\}_i$ 中存在一个非正元素。

证明。 反设 $\{a_i\}_i$ 全为正数。注意，对于任意 $i \geq 1$，

$$a_i - a_{i+1} = a_i - \left(\gamma - \frac{\gamma}{a_i}\right) = \frac{a_i^2 - \gamma a_i + \gamma}{a_i} > 0,$$

由于 $1 \leq \gamma < 4$，不等式成立，这意味着该序列严格递减。因此 $\{a_i\}_i$，这与引理9矛盾。

□

通过选取足够大的 m，我们可以使 n 任意大。根据引理10，若 $\gamma < 4$，则 $\{a_i\}$中存在一个非正元素。然而，在这种情况下，该算法无法满足公式（12）。因此，必须有 $\gamma \geq 4$，证明完毕。

C.3.2 一致性与鲁棒性之间的权衡

现在我们可以证明定理6。给定一个常数 $\lambda \in (0,1)$。考虑同一个$\widehat{\ }$实例，其中 m 是一个足够大的数，且对于 $j \in [1, m]$，$b(j) = j$。令预测 $J \leq m\widehat{\ }$也足够大。任取一个利用预测 J的确定性算法。由于我们希望保证该算法具有 $(1+\lambda)$-一致性，该算法在点击索引为 J 或更大的按钮之前，至多已产生 $\lambda \cdot J$。令 n 为该算法在点击索引为 J 或更大的按钮之前已点击的按钮数。因此，我们得到关于该算法的以下约束：

$$\sum_{j=1}^{n} x_j \leq \lambda \cdot \widehat{J}, \tag{13}$$

其中，x_j 再次表示上一节中第 j 轮点击的按钮的价格。

与此同时，我们还希望算法具有鲁棒性。因此，对于 $i = 1, 2, \cdots, n-1$，算法应满足方程(11)，其中 $\gamma \geq 4$现在表示鲁棒性比率；此外，还需满足一个约束$\widehat{\ }$，用于刻画以下情况：当 $J = x_n + \varepsilon$ 时，算法在点击按钮 x_n 后至少产生 J。换言之，我们有

$$\sum_{i=1}^{n} x_i + \widehat{J} \leq \gamma x_n. \tag{14}$$

注意，由方程 (13) 和 (14)，我们有

$$\left(1+\frac{1}{\lambda}\right)\cdot\sum_{j=1}^{n}x_j\leq\gamma x_n.$$

将此不等式与公式（12）结合，可得

$$\left(1+\frac{1}{\lambda}\right)\cdot\frac{\gamma}{a_{n-1}}x_n\leq\gamma x_n,$$

从而得到

$$1+\frac{1}{\lambda}\leq a_{n-1}.$$

当 m 且 J 足够大时，我们也有一个足够大的 n。因此，我们可以在上述不等式中用$\alpha:=\lim_{i\to\infty}a_i$替换 a_{n-1}。根据引理9的证明，很容易看出 $\alpha=\frac{\gamma+\sqrt{\gamma^2-4\gamma}}{2}$。因此，对于 $\lambda\in(0,\ 1)$，我们可以得到

$$\begin{aligned}1+\frac{1}{\lambda}\leq\frac{\gamma+\sqrt{\gamma^2-4\gamma}}{2}&\implies 2+\frac{2}{\lambda}-\gamma\leq\sqrt{\gamma^2-4\gamma}\\&\implies\left(2+\frac{2}{\lambda}-\gamma\right)^2\leq\gamma^2-4\gamma\\&\implies\frac{4}{\lambda^2}+\frac{4}{\lambda}(2-\gamma)+4\leq 0\\&\implies\gamma\geq 2+\lambda+\frac{1}{\lambda}.\end{aligned}$$

结合引理 3，定理6的证明至此完成。