arXiv:2301.01631v1 [cs.DS] 4 Jan 2023

R´enyi-Ulam 猜测游戏与不完美建议下的在线计算

Spyros Angelopoulos[1] 和 Shahin Kamali[2]

[1]法国国家科学研究中心和 LIP6-索邦大学，法国巴黎。spyros.angelopoulos@lip6.fr[2]加拿大约克大学。kamalis@yorku.ca

摘要

我们研究了不完美建议下在线计算这一新兴问题设定：在线算法借助某种预测得到增强，该预测以可能存在错误的二进制字符串形式编码。算法并不知道建议错误的情况，但会设定一个期望的容错度，即它能够容忍的建议比特错误数量的上界。这一模型推广了不可信建议模型 [Angelopoulos 等人，ITCS 2020], ；在该模型中，算法的性能仅在错误程度的极端情形下进行评估（即建议完全无误，或由对手生成）。

在这项工作中，我们建立了说谎回答者博弈（也称为 R´enyi-Ulam 博弈）与具有不完美建议的在线算法的设计和分析之间的联系。具体而言，我们展示了如何为时间序列搜索、在线竞价和分数背包等已得到深入研究的在线问题，推导其竞争比的上界和下界。我们的方法首次为这一模型下的在线问题给出了下界。我们还指出并利用了具有不完美建议的竞争分析与多处理器系统容错之间的联系。最后，我们展示了如何通过资源增强和鲁棒化，消除对容差参数的依赖。

关键词：在线计算、含噪查询、R´enyi-Ulam 博弈、超越最坏情况分析、容错算法。

1 引言

在线计算，尤其是竞争分析，已成为在不确定条件下对算法进行理论分析的经典框架。虽然早期的标准在线计算定义 [37]假设算法对请求序列一无所知，但在实际情形中，算法确实可能掌握某些有限但可能不准确的信息（例如，某种前瞻信息，或有关典型序列的历史信息）。因此，我们需要更细致的模型，以刻画借助外部信息增强的在线算法的能力与局限。

理论计算机科学中的一种方法是建议复杂度框架；参见 [19, 11, 21], 这篇综述 [12] 以及这本书 [25]。在建议复杂度模型（尤其是磁带模型 [10]）中，在线算法会收到一个字符串，其中编码了与请求序列相关的信息，并可借此提升其性能。其目标是量化建议大小（以比特数计）与算法竞争比之间的权衡。该模型的要求十分严格：假定建议没有错误，且可以由全知全能的预言机提供。因此，正如 [34], 所指出的，该模型主要具有理论意义。

另一种更实用的方法，研究预测对提升竞争比的影响。在这种模型中，在线算法会获得一些有关请求序列的不完美信息，但对其大小没有限制。人们关注的是这样一类算法：其性能会随预测误差增加而平缓下降；具体而言，当预测没有错误时，算法表现良好（这称为算法的一致性），同时在面对任何可能的错误时仍保持鲁棒（这称为算法的鲁棒性）。这一研究方向由 [31] 和 [35], 的工作开创，此后已有大量在线问题在该模型下得到研究（例如，参见 [34]这篇综述以及 [1]在线文集）。

建议复杂度与预测模型的一种结合是不可信建议模型，该模型在 [7] 中被提出。在这一模型中，部分建议比特可能有误，算法的性能则根据建议错误情况的两种极端情形进行评估。在一种极端情形下，建议无错误；而在另一种极端情形下，建议由一名旨在最大限度降低算法性能的（恶意）对手生成。借用带预测的算法的术语，这两种竞争比分别称为一致性和鲁棒性。目标是找出在这两项极端指标上帕累托有效，理想情况下达到帕累托最优的算法。近期已有多种在线问题在帕累托最优性框架下得到研究（涵盖不可信建议模型和预测模型）；例如，参见 [38, 28, 26, 5, 8]。

1.1 不完美建议下的在线计算

本文聚焦于一种新兴模型，在该模型中，建议可能并不完美。出发点是，不可信建议的帕累托框架只关注极端竞争比，即一致性和鲁棒性。一个更一般的问题是，根据建议错误评估在线算法的性能。给定一个大小为 k 的建议字符串，我们用 $\eta \leq k$ 表示错误比特的数量。因此，我们旨在研究这一设定下在线算法的能力与局限性，也就是从竞争比上界和下界两个方面进行研究。

自然地，算法事先并不知道确切的建议错误。相反，算法会定义一个特定于应用的参数 $H \leq k$，用来确定所需的错误容忍度，或者等价地，设定对建议错误的预期上界。这个参数经常出现在说谎回答者博弈的分析中，例如与我们研究相关的 R´ényi-Ulam 博弈 [36], ，我们很快就会对此进行讨论。近期关于弱预测的学习增强型在线算法的研究也进一步说明了这一参数的必要性：在这类研究中，预测是输入某个相关参数的上界（例如，采用频率预测的在线背包 [23], ，其中预测是在线出现的物品大小的上界）。我们的目标是从上界和下界两个角度，量化建议大小、错误容忍度与竞争比之间的权衡关系。

对不完美建议模型的一种不同解读是，将每个建议位视为针对输入提出的一个二元查询的回答。因此，可以将 k 位建议字符串看作通过 k 个二元查询获得的预测，而这些查询并非都能得到正确回答。众所周知，查询有助于提升机器学习应用中近似算法的性能；例如，带噪查询聚类 [33], 中的查询会询问两个点是否应属于同一个簇，并且每个查询都能以 p 的概率得到正确回答，这一概率为算法所知。在本文中，我们研究带噪查询在线算法的能力及其局限性。不过，与 [33], 不同，我们不对查询回答作任何概率假设。据我们所知，不完美建议模型（尤其是基于二元查询的解读）仅应用于合同调度 [8], 和时间序列搜索 [9], 问题，且研究仅从上界的角度展开。

1.2 贡献

我们建立了说谎回答者博弈与具有不完美建议的在线算法设计和分析之间的联系。具体而言，我们展示了如何利用 R´ényi-Ulam 博弈分析中的结果，并在竞争分析方面得出正面和负面的结果。我们将这些工具应用于时间序列搜索、在线竞价和在线分数背包等研究充分的问题。我们的结果改进了这些问题已知的上界（在已有相关结果的情况下），同时也首次给出了此类设定下在线问题竞争比的下界。

更具体地说，我们首先在第 3 节中以时间序列搜索问题作为热身，展示这些技术如何帮助我们改进 [9] 的结果；我们还展示如何使用基于二元熵函数的近似来评估竞争比。在第 4 节中，我们研究一个更复杂的应用，即在线竞价问题；该问题最初由 [7] 在不可信建议的背景下研究。这里的关键在于建立近乎最优的下界。为此，我们在 $l \leq 2^k$ 个处理器上构建在线竞价的多处理器版本，其中一定数量的处理器可能发生故障；随后，我们将该问题的竞争比与不完美建议设定联系起来，具体做法是将处理器层面的容错与 R´enyi-Ulam 猜谜游戏中的固有误差联系起来。在第 5 节中，我们研究在线分数背包问题。在此，我们提出一种算法，只要 $H < k/2$，其竞争比就会以关于 k 的指数速度收敛到 1。我们还给出了一个几乎匹配的下界，表明该算法接近最优。对于上界，关键在于分配查询，以近似实例中适当定义的两个参数。对于下界，我们使用信息论论证。具体而言，我们证明了一个来自 R´enyi-Ulam 猜谜游戏的归约：如果存在竞争比优于某个特定值的算法，就可以将游戏进行到超出理论性能界的程度，从而导致矛盾。

如上所述，参数 H 表示算法期望达到的错误容忍度，因此取决于具体应用。在第 6 节中，我们论证了即使事先不知道确切的容差，这些结果在相应场景中仍然有用。我们通过两种不同的方式实现这一点：第一，利用资源增广论证，即比较使用大小为 k 的完美（无错误）建议的算法，与使用 $l > k$ 个建议比特、但建议错误可能非常高的算法的性能。第二，通过使算法鲁棒化，也就是说，要求即使错误超过容差参数，算法仍能表现良好。

我们开发的技术也可应用于其他在线问题。具体而言，我们针对在线竞价问题提出的方法构成了以下通用框架：对于上界，目标是定义一组“候选”算法，使它们的最坏情况性能排名彼此接近。随后，可以利用建议从中选出一个合适的候选算法，使其表现接近最优。对于下界，目标是证明，在任何一组候选算法中，错误查询都可能不得不总是返回一个解，而该解与最佳解之间的“排名”差距足够大；然后，还需要从下界的角度将“排名”这一概念与性能联系起来。最后这部分突显了带有不完美建议的在线问题与其在并行系统中（不带建议）的容错版本之间的联系。另一方面，我们针对时间序列和分数背包问题的方法展示了另一种通用技术：对于上界，应找出问题的一些重要参数，然后适当分配查询，以便在存在回答错误的情况下对这些参数进行近似。对于下界，则应通过信息论论证，建立从 R´enyi-Ulam 博弈到该在线问题的归约。

2 有说谎应答者的游戏

我们回顾一些关于说谎回答者博弈的核心结果；这些结果将是分析带有不完美建议的在线问题的关键。我们尤其关注 [36], ，它研究提问者与回答者之间的博弈，涉及一个从域 $\mathcal{D}$ 中选出的未知价值 x。提问者可以提出一般形式的查询：“ x 是否属于 S”，其中 S 是 $\mathcal{D}$ 的某个子集，这类查询称为子集查询。即使提问者提出简单得多的查询，上界仍然成立；这类查询是形如“ x 是否至多为 a”的比较查询，其中给定了某个 a。 [36] 中的上界和下界都用二项式系数的部分和表示。形式化地，我们定义：

$$\left(\!\!\binom{N}{m}\!\!\right) := \sum_{j=0}^{m} \binom{N}{j}, \text{ for } m \le N.$$

我们尤其关注以下在连续空间上进行的游戏 :

CONTINIOUSSEARCH(k, H) 游戏：在这个游戏中， x 是一个实数，且满足 $x \in \mathcal{D} = (0, 1]$, ，提问者提出 k 个查询，其中至多 H 个可能收到错误响应。提问者的目标是找到一个区间 I_x ，使得 $x \in I_x$ ，并将 $|I_x|$ 最小化。

引理 1. *[36]* 对于 CONTINIOUSSEARCH(k, H)，任何满足 $H \le k/2$ 的提问者策略都满足 $|I_x| \ge \left(\!\!\binom{k}{H}\!\!\right)/2^k$。此外，对于 $H \le k/2$，存在一种名为 C-WEIGHTING 的策略，它使用比较查询并输出一个区间 $I_{W,x}$，满足 $|I_{W,x}| \le \left(\!\!\binom{k-H}{H}\!\!\right)/2^{k-H}$。

以下 所述游戏将有助于我们分析在线时间序列和分数背包问题 k.

FIND(k, H) 游戏：在此游戏中，给定 k、 $H \le k/2$ 和 $\mathcal{D} = \{1, \ldots, m\}$，目标是使用 k 次查询找出未知的 $x \in \mathcal{D}$，其中最多有 H 次查询可能得到错误回答。

以下定理可直接由引理 1 证明：

定理 2. 若提问者能够在 FIND(k, H) 游戏中识别任意数字 $x \in \{1, 2, \ldots, \mu(k, H)\}$，则满足条件的最大正整数为 $\mu(k, H)$，且 $2^{k-H} / \left(\!\!\binom{k-H}{H}\!\!\right) \le \mu(k, H) \le 2^k / \left(\!\!\binom{k}{H}\!\!\right)$。

我们定义另外两个与分析相关的游戏。第一个游戏与在循环排列中搜索有关，将用于在线竞价的上界分析。

MINCYCLIC(n, k, H) 游戏：给定一个数组 $A[0 \ldots n-1]$，其元素是 $\{0, \ldots, n-1\}$ 的一个未知循环置换，目标是使用 k 次查询（其中最多 $H \le k/2$ 次可能出错），输出数组中一个元素尽可能小的索引。

定理 3（附录）。存在一种针对 MINCYCLIC(n, k, H) 的提问者策略，该策略基于 k 次比较查询，并输出一个索引 j，使得对所有 $H \le k/2$， $A[j] \le \lceil n \left(\!\!\binom{k-H}{H}\!\!\right)/2^{k-H} \rceil$。

最后，我们定义一种与一般排列搜索相关的游戏，它将有助于建立在线竞价竞争比的下界。

SEARCH(n, k, H) 游戏：给定一个数组 $A[0, \ldots, n-1]$，其元素是 $\{0, \ldots, n-1\}$ 的一个未知排列，目标是使用 k 次查询（其中最多 H 次可能出错），从而输出数组中一个元素尽可能小的索引。

定理 4（附录）。对于 SEARCH(n, k, H) 游戏中的任意提问者策略，都存在一种应答者策略，使得如果 e 是返回的 A 中的元素，那么 $A[e] \ge \lfloor n \left(\!\!\binom{k}{H}\!\!\right)/2^k \rfloor$。

3 预热：在线时间序列搜索

在线（时间序列）搜索问题描述了不确定性下决策中的一种简单却基础的情境。在这个问题中，玩家必须在某个时间范围内（例如， d天内）卖出一项不可分割资产，而玩家并不知道这个时间范围。每天 i都会揭示一个价格 p_i，玩家有两个选择：要么接受该价格并获得利润 p_i（此时游戏结束），要么拒绝该价格（此时游戏继续到第 $i+1$天）。如果玩家到第 d天仍未接受任何价格，那么默认接受最后一个价格 p_d。玩家算法的竞争比是对所有价格序列取最坏情况时，序列中的最高价格与玩家接受的价格之比。

该问题最早由 [20] 提出并研究，他们给出了一种简单的确定性算法，其竞争比为 $\sqrt{M/m}$，其中 M和 m 分别是序列中最高价和最低价的上界与下界，并假设算法已知这些值。对于确定性算法而言，该界是最优的。时间序列搜索是在线金融优化中的一种基本范式，其多种变体和推广已得到研究 [18, 30, 39, 17]；另见综述 [27]。该问题还被用作案例研究，用于评估在线算法的多种性能度量，包括竞争分析之外的度量 [13, 2]。

近期， [9], 在不完美建议框架下研究了时间序列搜索，并在假设 $H \le k/4$的情况下，证明使用 k位建议且容错度为 H时，竞争比的上界为 $(M/m)^{2^{2H-k/2}}$。请注意，目前尚无针对 $H \in (k/4, k/2]$的上界。如果建议无错误[1]，即在建议复杂度模型中，那么竞争比的紧确界 $(M/m)^{\frac{1}{2^k+1}}$由 [17] 给出。

作为第2节中讨论的 FIND(k, H) 游戏的一项应用，我们证明以下结果。

定理 5. 考虑在线时间序列搜索问题，其中不完美建议的大小为 k，容差为$H \le k/2$。存在一种算法，使用 k 次比较查询，且其竞争比至多为 $(M/m)^{\frac{1}{U+1}}$，其中 $U = \lfloor 2^{k-H} / \left(\!\binom{k-H}{H}\!\right) \rfloor$，对任意 $H \le k/2$ 均成立。相比之下，任何基于 k 次子集查询的（确定性）算法，其竞争比都小于 $(M/m)^{\frac{1}{L+1}}$，其中 $L = \lceil 2^k / \left(\!\binom{k}{H}\!\right) \rceil$。

证明。我们首先证明上界。令 a_1、 $\dots a_U$、 r 满足 $r = \frac{a_1}{m} = \frac{a_2}{a_1} = \dots = \frac{a_U}{a_{U-1}} = \frac{M}{a_U}$，因此 $r = (M/m)^{1/(U+1)}$。该算法使用 k 次比较查询，在集合 $\{a_i\}_{i=1}^{U}$ 中找出最佳保留价格，即阈值 p；高于该阈值时，算法将始终接受序列中的价格。根据定理 2，这一点成立，因为 $U \le 2^{k-H} / \left(\!\binom{k-H}{H}\!\right)$。随后，算法将 p 作为其保留价格，也就是说，它接受请求序列中第一个大于或等于 p 的价格。根据集合 $\{a_i\}_{i=1}^{U}$ 的定义，很容易得出该算法的竞争比至多为 r，由此完成上界的证明。

现在证明下界。反设存在一个用于时间序列搜索的算法 A，该算法使用 k 位不完美建议，且竞争比小于 $C = (M/m)^{\frac{1}{L+1}}$。我们将证明，届时可以在 FIND$(k, H)$ 游戏中使用 A，通过 k 次查询识别出 $\{1, \dots, L+1\}$ 中的一个未知价值，这与定理 2 的上界相矛盾。

为得出矛盾，定义 $a_1, \dots, a_L$使得 $r' = \frac{a_1}{m} = \frac{a_2}{a_1} = \dots = \frac{a_L}{a_{L-1}} = \frac{M}{a_L}$，因此$r' = (M/m)^{\frac{1}{L+1}} = C$。考虑在线算法 A 与对手之间的一场博弈，其中请求序列由 $\{m, a_1, \dots, a_L, M\}$中的价格组成。更确切地说，考虑所有形如 $\sigma_i = m, a_1, \dots, a_i, m$的请求序列集合，其中 $i \in [1, L+1]$，且 a_{L+1}定义为等于

M。在 σ_i中，若要达到严格小于 C竞争比， A 必须接受价格 a_i （序列中的最后一个请求）。等价地， A 使用 k 次查询，最多有 H 次错误，并在集合 $\{a_j\}_{j=1}^{L+1}$中找到 a_i，这与定理 2矛盾。 □

3.1 界的比较

为了比较定理 5 的上界和下界，我们需要能够计算二项式系数部分和。由于这个部分和没有闭式，我们将采用 [32] 给出的以下有用近似。令 $\mathcal{H}$ 表示二元熵函数。那么

$$\frac{2^{N\mathcal{H}(\frac{m}{N})}}{\sqrt{8m(1-\frac{m}{N})}} \leq \left(\!\!\binom{N}{m}\!\!\right) \leq 2^{N\mathcal{H}(\frac{m}{N})}, \quad \text{for } 0 < m < N/2. \tag{1}$$

我们还将使用二元熵函数的以下性质

$$4p(1-p) \leq \mathcal{H}(p) \leq (4p(1-p))^{1/\ln 4}, \text{ for all } p \in (0,1). \tag{2}$$

我们首先证明，定理 5 中的算法优于 [9] 中的算法。首先注意， [9] 假设 $H \leq k/4$，而定理 5 适用于所有 $H \leq k/2$。此外，对于 H和 k的所有取值，我们都改进了竞争比。为此，只需证明 $\left(\!\!\binom{k-H}{H}\!\!\right)/2^{k-H} < 2^{2H-k/2}$；由 (1) 可知，当 $2^{(k-H)(\mathcal{H}(\frac{H}{k-H})-1)} < 2^{2H-k/2}$ 时成立，或者等价地，当 $(k-H)(\mathcal{H}(\frac{H}{k-H}) - 1) < 2H - k/2$ 时成立。令 τ满足 $\tau = H/k$ （因此 $\tau \leq 1/2$），那么后者等价于证明 $\mathcal{H}(\frac{\tau}{1-\tau}) < \frac{1+2\tau}{2-2\tau}$。根据 (2)，只需证明

$$(\frac{4\tau(1-2\tau)}{(1-\tau)^2})^{1/\ln 4} < \frac{1+2\tau}{2-2\tau},$$

这对所有 $\tau \leq 1/2$ 都成立。

接下来，我们考察定理 5 的上界和下界彼此有多接近。回想一下，这些界的形式分别为 $(M/m)^{1/(U+1)}$ 和 $(M/m)^{1/(L+1)}$。利用 (1)，并为简便起见忽略向下取整和向上取整，我们得到

$$U \geq 2^{k(1-\tau)(1-\mathcal{H}(\frac{\tau}{1-\tau}))} \text{ and } L \leq \sqrt{8k\tau(1-\tau)}2^{k(1-\mathcal{H}(\tau))}.$$

上述不等式结合 (2) 表明，上界和下界彼此非常接近，因为对于任意固定的 τ，都有 $U \geq 2^{\Theta(k)}$ 和 $L \leq 2^{\Theta(k)}$。

4 在线竞价

在线竞价由 [16] 引入，作为一种典型问题，用于形式化在线和离线优化问题中的基于加倍的策略，例如在线上搜索目标、最小延迟和层次聚类。在这个问题中，玩家希望猜出一个隐藏的未知实数值 $u \geq 1$。为此，玩家定义一个由正数递增出价组成的（无限）序列 $X = (x_i)$，这就是它的策略。使用策略 X 发现隐藏值 u 的成本，记作 $c(X,u)$，定义为 $\sum_{i=1}^{j_u} x_i$，其中 j_u 满足 $x_{j_u-1} < u \leq x_{j_u}$。因此，自然可以将竞价者策略 X 的竞争比定义为 $\mathrm{Cr}(X) = \sup_u \frac{c(X,u)}{u}$。

在问题的标准版本中，也就是说，不使用建议时，倍增策略 $x_i = 2^i$ 可实现最优竞争比 4。在线竞价也在不可信建议模型下得到研究， [7], 并给出了一致性与鲁棒性权衡的界。该问题还与合同调度有关，后者在 [8], 中进行了研究；另请参见第 4.1.3 节中的讨论。

4.1 不完美建议下的在线竞价

4.1.1 上界

上界背后的思路如下。我们将考虑一个由 2^k几何递增序列构成的空间（见定义6）。在完美建议的理想情况下，可以利用 k 建议比特来确定该空间中的最佳策略。存在建议错误时，我们将展示如何利用该空间的循环结构，并结合 MINCYCLIC 博弈的上界（定理 3），找到一个与最优策略相差不大的策略。

我们首先定义几何递增出价序列的空间。

定义6. 对于给定的 $b > 1$和 $l \in \mathbb{N}^+$，定义 $\mathcal{X}_{b,\ l}$ 为出价序列 $\{X_0, \dots X_{l-1}\}$ 的集合，其中 $X_i = (b^{i+jl})_{j=0}^{\infty}$，对所有 $i \in [0, l-1]$ 均成立。

根据 $\mathcal{X}_{b,\ l}$ 的定义，很容易看出，对于任何潜在目标 u，都存在一个 $\{0, \dots l-1\}$ 的循环置换 π，它根据策略 $\mathcal{X}_{b,\ l}$ 的表现确定其排序。更确切地说，假设 $X_{\pi(0)}$ 是以至少 C 的成本发现 u 的最佳序列。那么， $X_{\pi(i)}$ 能以至多 $b^i C$ 的成本发现 u。这一性质有助于我们证明以下上界：

定理 7（附录）。存在一种基于 k 个比较查询的出价策略，其竞争比至多为 $\frac{1+U}{2^k}\left(1+\frac{2^k}{1+U}\right)^{1+\frac{1+U}{2^k}}$，其中 $U = \lceil 2^H \left(\!\!\binom{k-H}{H}\!\!\right) \rceil$.

4.1.2 下界

下界背后的思路如下。拥有 k 个建议比特时，能做到的最好情况是从一个至多包含 2^k 个策略的集合 $\mathcal{X}$ 中选出最佳策略。请注意，如果建议无错误， $|\mathcal{X}|$ 可以大到 2^k；然而，在存在错误的情况下，算法可能会选择缩小 $|\mathcal{X}|$。

我们的方法结合了两个思路。第一个思路利用 SEARCH(n, k, H) 游戏的抽象形式以及定理 4 的下界。这个结果使我们能够为所选策略的排名设定下界，其中最佳策略的排名为 0。第二个思路是定义一个度量，用来表示在 $\mathcal{X}$ 中，排名为 j 的策略相较于 $\mathcal{X}$ 中的最佳策略会差多少。我们将借助并行性和容错性的概念来实现这一点。

更准确地说，给定整数 p 和 ϕ，且满足 $\phi < p$，我们将容错并行竞价问题记作 FPB(p, ϕ)，定义如下。玩家可以并行运行 p 个竞价策略；然而，这些策略中有 ϕ 个可能发生故障，即永远无法发现目标；例如，我们可以把故障策略理解为玩家在某个时刻突然停止出价，这类似于“拜占庭式”故障。此时，发现目标 u 的成本定义为 $p - \phi$ 个无故障策略中有一个发现该目标所需的最低成本。需要注意的是，故障由旨在使该成本最大化的对手决定。竞争比也据此定义。

下一个定理是关于 FPB(p, ϕ) 的主要技术结果，它给出了该问题中任意策略的竞争比下界，该下界是参数 p、 ϕ 和 $\alpha_{\bar{X}}$ 的函数。这里， X 定义为 p-并行策略 X 中所有出价按非递减顺序排列后得到的序列。此外，给定一个正实数序列 X，我们将 α_X 定义为 $\limsup_{i\to\infty} x_i^{1/i}$。

定理 8（附录）。每个用于 FPB(p, ϕ) 的 p-并行策略 X 的竞争比 Cr(X) $\geq \frac{\alpha_{\bar{X}}^{p+1+\phi}}{\alpha_{\bar{X}}^{p}-1}$.

证明概要。我们利用 p-并行策略的性质，证明任何此类策略都满足 $\mathrm{Cr}(X) \geq \sup_q \frac{\sum_{i=0}^{q+\phi+1} \bar{x}_i}{\sum_{i=q}^{q-(p-1)} \bar{x}_i}$。然后，我们利用 Gal 函数定理 [22] 得出结果。许多技术细节请参见附录。

□

现在，我们结合上述思路，给出该问题下界的推导方法。需要强调一个微妙之处：与大小为 k 的无误差建议不同，在无误差建议下，应始终从恰好包含 2^k 个策略的集合中选择最佳策略；而在存在错误时，这个集合的大小完全可能是 $l < 2^k$。这是因为，随着 l 减小，错误对竞争比的影响也随之减小。换言之，我们需要对所有取值 $l \leq 2^k$ 证明该结果，而不只是对 $l = 2^k$。

定理 9. 对于不完美建议模型中的任意出价序列 X 和 k 次子集查询，有 $\mathrm{Cr}(X) \geq \frac{1}{L}(1+L)^{1+1/L}$，其中 $L = 2^k / \left(\!\binom{k}{H}\!\right)$。

证明. 每种出价策略都会利用查询的响应，从一个包含 $\mathcal{X} = \{X_0, \ldots, X_{l-1}\}$ 个候选序列的集合中选出一个策略，其中 $l \leq 2^k$。对于给定的目标值 u，可以对 $\mathcal{X}$ 中的 l 个序列排序，使得 $X_{\pi(i)}$ 的竞争比不劣于 $X_{\pi(i+1)}$；具体而言，按序列的表现从高到低排列。由定理 4 可知，该策略将选择一个序列 X_j，使得 $\pi(j) \geq \lfloor l \left(\!\binom{k}{H}\!\right) / 2^k \rfloor$。所选序列的竞争比至少与由 $\mathcal{X}$ 定义的 l 并行策略的竞争比相同；在该策略中，最多可能有 $\phi_l = \lfloor l \left(\!\binom{k}{H}\!\right) / 2^k \rfloor$ 个序列发生故障。由定理 8，

$$\mathrm{Cr}(X) \geq \frac{\alpha_{\bar{X}}^{l+1+\phi_l}}{\alpha_{\bar{X}}^{l} - 1}, \quad \text{with } \phi_l = \lfloor l \left(\!\binom{k}{H}\!\right) / 2^k \rfloor. \tag{3}$$

现在我们考虑两种情况。首先假设 $l < L$。在这种情况下，情况 $\phi_l = 0$，因此由 (3) 可得 $\mathrm{Cr}(X) \geq \alpha_{\bar{X}}^{l+1}/(\alpha_{\bar{X}}^{l} - 1)$，其在 $\alpha_{\bar{X}} = (l+1)^{1/l} > 1$ 时取最小值，因此 $\mathrm{Cr}(X) \geq \frac{1}{l}(l+1)^{1+1/l}$。该函数关于 l 单调递减，且由于 $l < L$，我们有 $\mathrm{Cr}(X) \geq \frac{1}{L}(1+L)^{1+1/L}$. 接下来，假设 $l \in [L, 2^k]$。在这种情况下，由 (3) 可得 $\mathrm{Cr}(X) \geq \frac{\alpha_{\bar{X}}^{l(1+1/L)}}{\alpha_{\bar{X}}^{l} - 1}$. 上述表达式在 $\alpha_{\bar{X}} = (1+L)^{1/l}$ 时取最小值，代入后再次得到 $\mathrm{Cr}(X) \geq \frac{1}{L}(1+L)^{1+1/L}$.

□

4.1.3 界的比较

我们在附录中证明，这两个界的比值约为

$$\log \frac{\mathrm{UB}}{\mathrm{LB}} \leq \frac{\sqrt{8k\tau(1-\tau)}k(1-\tau)(1-\mathcal{H}(\frac{\tau}{1-\tau}))}{2^{k(1-\tau)(1-\mathcal{H}(\frac{\tau}{1-\tau}))}} - \frac{k(1-\mathcal{H}(\tau))}{2^{k(1-\mathcal{H}(\tau))}},$$

其中 $\tau = H/k$。我们推断，随着 k 增大，对于任意固定的 τ 值，上界和下界会变得非常接近。

5 在线分数背包

在线分数背包问题中，请求序列由物品组成，其中物品 i 的价值为$v_i \in \mathbb{R}^+$，大小为 $s_i \in (0, 1]$。在线算法拥有一个容量为单位容量的背包，在考虑物品 i时，可以不可撤销地接受该物品的分数 $f_i \in (0, 1]$，并受容量约束限制。更准确地说，算法旨在最大化 $\sum(f_i \cdot v_i)$，满足约束 $\sum(f_i \cdot s_i) \leq 1$。$i$ i

令 $d_i = v_i/s_i$ 表示物品 i的密度。尽管该问题的离线版本可以通过贪心算法（按密度的非递减顺序对所有物品排序，并按此顺序接受物品，直至背包装满）得到一个简单的最优解，但在线版本更具挑战性。假设 $d_i \in [L, U]$，对于 L, U 算法已知。[14, 15] 分别给出了该问题竞争比的匹配上界 $O(\log(U/L))$ 和下界 $\Omega(\log(U/L))$，而 [40] 证明了确定性算法的最优界为 $\ln(U/L) + 1$。在线分数背包应用于赞助搜索拍卖和在线广告分配，也已在其他若干设定中得到研究，例如 [3, 24]。本节将研究不完美建议设定下的这一问题。

5.1 上界

与以往所有研究一样，我们假设所有物品的密度都属于 $[L, U]$，其中 L 和 U 为已知值。令 d^* 表示最优解中以正分数纳入的物品的最小密度。也就是说，最优算法 OPT 接受密度大于 d^* 的物品的全部（比例为 1），并用密度为 d^* 的物品填满剩余空间。不幸的是，仅知道 d^*（即使知道其精确值），也不足以让在线算法达到 OPT 的效率。例如，对于只由密度为 d^* 的物品组成的序列，接受密度大于 d^* 的物品的全部（比例为 1）的算法，其竞争比无界。同样，对于序列中较早出现密度为 d^* 的物品、较晚出现密度更高物品的情况，接受密度至少为 d^* 的物品的全部（比例为 1）的算法，其竞争比也无界。然而，如果用 $c^* \in$ 表示最优解中背包的比例（0, 1），其中这部分要么为空，要么由密度为 d^* 的物品占据，那么同时知道 d^* 和 c^* 的精确值就足以达到最优。于是，我们的方法将尝试使用 k 个比较查询来近似 c^* 和 d^* 的值，再利用这些近似值来选择分数物品。

5.1.1 算法与分析

我们介绍该在线算法。首先定义两种划分，分别与参数 d^* 和 c^* 有关。对于 d^*，将区间 $[L, U]$ 划分为 s 个子区间 $I_1, \ldots, I_s$，使得 $I_i = [d_{i-1}, d_i)$，其中 s 将在稍后指定。我们还设定 $L = d_0$、$U = d_s$。数值 d_i 的定义满足：$\beta = \frac{d_1}{d_0} = \frac{d_2}{d_1} = \ldots = \frac{d_s}{d_{s-1}}$。因此，我们有 $\beta = (U/L)^{1/s}$ 和 $d_i = L \cdot \beta^i$，并注意到，对于某个 $x \in [1, s]$，有 $d^* \in I_x$。

对于参数 c^*，我们将区间 $[0, 1]$ 划分为 m 个子区间 $I'_1, \ldots, I'_m$，使得 $I'_i = [c_{i-1}, c_i)$；我们有 $c_0 = 0$ 和 $c_m = 1$。m 的值稍后确定；数值 c_i 的定义满足 $c_1 = c_2 - \frac{c_1}{\beta} = c_3 - \frac{c_2}{\beta} = \ldots = c_m - \frac{c_{m-1}}{\beta}$.

由此可直接得出，对于 $i \geq 1$，有 $c_i = \frac{\beta^{m+i-1} - \beta^{m+i-2}}{\beta^m - 1}$。特别地，$c_1 = \frac{\beta^m - \beta^{m-1}}{\beta^m - 1}$，且 $\frac{1}{1-c_1} = \frac{\beta^m - 1}{\beta^{m-1} - 1}$。还需注意，对于某个 $y \in [1, m]$，$c^* \in I'_y$。

若 $s \cdot m \leq \lfloor 2^{k-H} / \binom{k-H}{H} \rfloor$，定理 2 表明，算法可以使用 k 次比较查询，以确定 x 和 y。已知这些值后，算法会在背包中预留容量 $c = c_{y-1}$，用于存放密度范围为 $I_x = [d_{x-1}, d_x)$ 内的物品，我们称之为关键物品。算法将剩余容量 $1 - c$ 用于存放密度大于 d_x 的物品，我们称之为重物品。只要为关键物品预留的容量 c 足够，算法就会接纳所有关键物品的分数 1。同样，算法会接纳分数 1 的重物品，并将其放入背包中为它们专门预留的空间。由于 $c^* \in I_y$，我们有 $1 - c > 1 - c^*$；也就是说，为重物品预留的容量至少等于这些物品的总大小。换言之，算法能够接纳所有重物品。算法拒绝所有密度小于 d_{x-1} 的物品。

定理 10（附录）。对于任意 $H \le k/2$，上述算法的竞争比至多为

$$\min_{s,m\in\mathbb{N}} \quad f_m(\beta) \qquad \text{where} \quad \beta = (U/L)^{1/s}, \text{ and } f_m(\beta) = \frac{\beta^m - 1}{\beta^{m-1} - 1}$$
$$\text{subject to} \quad s \cdot m \le \lfloor 2^{k-H} / \left(\!\!\binom{k-H}{H}\!\!\right) \rfloor.$$

5.2 下界

我们将证明，任何采用不完美建议的算法，其竞争比都有一个下界 $C(k, H)$。为导出矛盾，假设存在一个竞争比优于 $C(k, H)$ 的算法 A。我们的证明基于从 FIND(k, H) 游戏进行的归约。具体而言，我们证明，基于 A，可以得到一个用于 FIND(k, H) 的提问者策略，该策略能够找到一个满足 $p = \lceil 2^k / \left(\!\!\binom{k}{H}\!\!\right) \rceil + 1$ 的价值 $z \in \{1, \ldots, p\}$，这与定理 2 矛盾。

我们先介绍证明背后的直觉。设 s 和 m 为任意两个正整数，使得 $s \cdot m \le p$ 和 $s \cdot (m+1) > p$。定义 $\beta = (U/L)^{1/s}$，并令 $d_i = U \cdot \beta^i$，其中 $i \in [1, s]$。给定一对整数 (x, y)，其中 $x \in \{1, \ldots, s\}$ 且 $y \in \{1, \ldots m+1\}$，定义序列

$$\sigma_{x,y} = ((d_1, 1), (d_2, 1), \ldots, (d_{x-1}, 1), (d_x, c_y),$$

其中，(d_i, j) 表示由 j/ϵ 个物品组成的一个子序列，每个物品的大小为 ϵ，密度为 d_i，且 ϵ 是无穷小量。$c_y \in [0, 1]$ 在证明中有适当定义。对于此序列，OPT$(\sigma_{x,\ y}) = (1 - c_y) d_{x-1} + c_y d_x$。这样的序列共有 $s \cdot (m+1) > p$ 个，且 $\sigma_{x,\ y}$ 是 $\sigma_{x,\ y+1}$ 的一个前缀序列，并且 $\sigma_{x,\ m}$ 是 1 的 $\sigma_{x+1,\ 1}$ 的一个前缀序列。在证明中，我们考虑这种形式的请求序列，并证明如果 A 是 $C(k, H)$-竞争的，那么它的决策可以帮助找出任意给定的 $z \in \{1, \ldots, p\}$，这与定理 2 矛盾。技术细节见附录。

定理 11（附录）。对于分数背包问题，若物品的密度属于 $[L, U]$，则不存在使用 k 次子集查询（其中 $H \le k/2$ 次可能得到错误响应）的确定性算法能够实现优于以下值的竞争比

$$C(k, H) = \min_{s,m\in\mathbb{N}} \quad g_m(\beta) \quad \text{where} \quad \beta = (U/L)^{1/s}, \ g_m(\beta) = \left(\frac{\beta^2 - \beta + 1}{2\beta + 1}\right)^{1/(m+1)}$$
$$\text{subject to} \quad s \cdot m \le \lceil 2^k / \left(\!\!\binom{k}{H}\!\!\right) \rceil + 1.$$

界的比较 令 $\tau = H/k$。由于 $\frac{\beta^m - 1}{\beta^{m-1} - 1} \le \beta$，利用 (1)，定理 10 的上界至多为 $(U/L)^q$，其中 $q \le 1/2^{k(1-\tau)(1-\mathcal{H}(\frac{\tau}{1-\tau}))}$。此外，由于 $\frac{\beta^2 - \beta + 1}{2\beta + 1} \ge \frac{\beta}{3}$（对所有 $\beta \ge 3$），定理 11 的下界至少为 $(U/L)^{q'}(1/3)^{q'}$，其中 $q' \ge 1/(2\sqrt{8k\tau(1-\tau)2^{k(1-\mathcal{H}(\tau))}} + 1)$，对所有 $U/L \ge 3$ 均成立。为简便起见，我们省略了向下取整和向上取整。

6 免除对容差参数的假设

在我们迄今研究的不完美建议设定中，算法会定义一个针对具体应用的容差参数，用于衡量其期望的错误容忍度（或者等价地说，预期的上界

）。从某种意义上说，这个参数是必需的，因为 [36] 中 R´enyi-Ulam 博弈的分析涉及误差极值（即 H），而不是实例特定误差值（即 η）。不过，在本节中，我们将讨论如何减少预先确定容差参数的必要性。我们提出两种不同的方法，分别基于资源增强和鲁棒化，并将在下文中讨论。我们以时间序列搜索和在线竞价问题为例进行说明，尽管我们的方法也可能适用于其他在线问题，只是计算会更加复杂。

6.1 资源增强

在这种设定下，我们比较一个拥有 l 个建议比特、但不知道误差界的无视型在线算法 A，与一个拥有 k 个理想（即无误差）建议比特的在线算法 B。具体而言，我们希望找到最小的 $l \geq k$（作为 k 的函数），使得无论算法 B 的建议中有多少错误，算法 A 的表现都至少与其一样好。

以下定理表明，对于任何常数 c，即使建议比特中有比例为 $1/3 - c$ 的比特可能出错，采用 $O(1)$ 倍资源增强也足以获得一种无视型算法，其效率至少与任何在无误差建议这一理想设定下运行的算法一样高。

定理 12（附录）。考虑时间序列问题和在线竞价问题。对于所有充分大的 k，以及任意 $c \in (0, 1/3)$，存在一个建议大小为 l 的无视型在线算法 A，其竞争比至少与任何拥有 k 个完美（即无误差）建议比特的在线算法 B 一样好；其中 $l = {}^{1}(\frac{2}{3}+c)(1-\mathcal{H}(\frac{\frac{1}{3}-c}{\frac{2}{3}+c}))k + 1$，且算法 A 的建议中可有任意误差 $\eta \leq (1/3 - c)l$。

6.2 鲁棒化

在这种设定下，我们扩展了不完美建议框架：不仅要求算法在建议错误不超过容忍度 H 时将竞争比降至最低，还要求当错误超过 H 时（特别是建议由对抗性方式生成时），其竞争比不超过某个指定的鲁棒性要求 r，其中 r。我们称这类在线算法为 r-鲁棒算法。因此，该模型可以视为对 [7] 的不完美建议模型和不可信建议模型的扩展。

对于时间序列问题，我们得到了以下结果，它推广了定理 5。特别地，请注意，对于 $\rho = 1$，定理 5 是定理 13 的特例。

定理 13（附录）。考虑在线时间序列搜索问题，其中不完美建议的大小为 k、容忍度为 $H \leq k/2$，鲁棒性为 $r = (M/m)^{\rho}$，其中 $\rho \in (1/2, 1]$。存在一种 r-鲁棒算法，使用 k 次比较查询，且竞争比至多为 $(M/m)^{\frac{2\rho-1}{U+1}}$，其中 $U = \lfloor 2^{k-H} / \left(\!\binom{k-H}{H}\!\right) \rfloor$，对任意 $H \leq k/2$ 均成立。此外，任何基于 k 次子集查询的（确定性）算法，其竞争比都优于 $(M/m)^{\frac{2\rho-1}{L+1}}$，其中 $L = \lceil 2^{k} / \left(\!\binom{k-H}{H}\!\right) \rceil$。

在线竞价中对 r-鲁棒算法的分析更具挑战性，尤其是在不可能性结果方面。下面概述这一方法。对于上界，我们可以参照定理 7 的思路进行分析；但是，集合 $\mathcal{X}_{b,\ 2^k}$ 中的每个出价序列都必须单独满足 r-鲁棒性。这一点很容易实现，并且要求 b much 满足 $b^2/(b-1) \leq r$。下界则更微妙：证明沿用定理 9 的思路，但利用了这样一个事实：如果 X_0，...，X_{l-1} 中的所有 l序列都必须满足 r- 鲁棒性，那么 $\alpha_{\bar{X}}^2/(\alpha_{\bar{X}} - 1) \leq r$。由此得到以下结果：

定理 14（附录）。对于每个 $r \geq 4$，都存在一种 r-鲁棒出价策略，使用 k 位不完美建议，且其竞争比至多为

$$\min_{b>1} \frac{b^{2^k+U+1}}{b^{2^k}-1}, \quad \textit{subject to } b^{2^{k+1}}/(b^{2^k}-1) \leq r, \quad \textit{and where } U = \lceil 2^H \left(\!\!\binom{k-H}{H}\!\!\right) \rceil.$$

此外，每个使用 k 位不完美建议的 r-鲁棒出价策略，其竞争比至少为

$$\min_{\alpha>1} \frac{\alpha^{2^k+L+1}}{\alpha^{2^k}-1} \quad \textit{subject to } \alpha^{2k}/(\alpha^k-1) \leq r, \quad \textit{and where } L = \lfloor \left(\!\!\binom{k}{H}\!\!\right) \rfloor.$$

参考文献

[1] 带预测的算法研究成果库。https://algorithms-with-prediction s.github.io/al 2022年。访问日期：2022-09-01。[2] Iftikhar Ahmad、Marcus Pirron 和 G¨unter Schmidt。使用不同性能度量分析基于威胁的算法。RAIRO：Recherche Op´erationnelle，55：2393，2021年。[3] Susanne Albers、Arindam Khan 和 Leon Ladewig。随机顺序模型下背包与间隙问题的改进在线算法。Algorithmica，83(6)：1750–1785，2021年。[4] Steve Alpern 和 Shmuel Gal。搜索博弈与会合问题理论。Kluwer Academic Publishers，2003年。[5] Spyros Angelopoulos。带提示的在线搜索。载于第12届理论计算机科学创新会议（ITCS）论文集，页码 51:1–51:16，2021 年。[6] Spyros Angelopoulos 、 Diogo Ars´enio 和 Shahin Kamali。含噪建议下的竞争性排序。CoRR，abs/2111.05281，2021年。[7] Spyros Angelopoulos、Christoph D¨urr、Shendan Jin、Shahin Kamali 和 Marc P. Renault。带不可信建议的在线计算。载于第11届理论计算机科学创新国际会议（ITCS）论文集，页码 52:1–52:15，2020年。[8] Spyros Angelopoulos 和 Shahin Kamali。基于预测的合同调度。载于第35届AAAI人工智能会议论文集。[9] Spyros Angelopoulos、Shahin Kamali 和 Dehou Zhang。基于最佳价格和查询的预测进行在线搜索。载于第36届AAAI人工智能会议论文集，页码 9652–9660，2022年。[10] Hans-Joachim B¨ockenhauer、Dennis Komm、Rastislav Kr´alovic 和 Richard Kr´alovic。关于 k 服务器问题的建议复杂度。J. Comput. Syst. Sci.，86：159–170，2017年。[11] Hans-Joachim B¨ockenhauer、Dennis Komm、Rastislav Kr´aloviˇc、Richard Kr´aloviˇc 和 Tobias M¨omke。关于在线问题的建议复杂度。载于第 20 届算法与计算国际研讨会（ ISAAC ）论文集，页码 331–340，2009年。

[12] Joan Boyar、Lene M. Favrholdt、Christian Kudahl、Kim S. Larsen 和 Jesper W. Mikkelsen。带建议的在线算法：综述。SIGACT News，47(3)：93–129，2016年。[13] Joan Boyar、Kim S. Larsen 和 Abyayananda Maiti。通过在线搜索比较性能度量。理论计算机科学，532：2–13，2014年。[14] Niv Buchbinder 和 Joseph Naor。用于覆盖与装箱问题的在线原始-对偶算法。载于第13届欧洲算法研讨会（ESA）论文集，页码 689–701。施普林格，2005年。[15] Niv Buchbinder 和 Joseph Naor。通过原始-对偶方法改进在线路由与装箱的界限。载于第47届年度IEEE计算机科学基础研讨会（FOCS）论文集，页码 293–304。IEEE，2006年。[16] Marek Chrobak 和 Claire Kenyon-Mathieu。SIGACT News 在线算法专栏第10篇：通过加倍实现竞争性。SIGACT News，37(4)：115–126，2006年。[17] Jhoirene Clemente、Juraj Hromkovi˘c、Dennis Komm 和 Christian Kudahl。在线搜索问题的建议复杂度。载于第27届国际组合算法研讨会（IWOCA）论文集，页码 203–212，2016年。[18] Peter Damaschke、Phuong Hoai Ha 和 Philippas Tsigas。具有时变价格界限的在线搜索。Algorithmica，55(4)：619–642，2009年。[19] Stefan Dobrev、Rastislav Kr´alovi˘c 和 Dana Pardubsk´a。衡量输入中的问题相关信息。RAIRO Theor. Informatics Appl.，43(3)：585–613，2009年。[20] Ran El-Yaniv、Amos Fiat、Richard M Karp 和 Gordon Turpin。最优搜索与单向交易在线算法。Algorithmica，30(1)：101–139，2001年。[21] Yuval Emek、Pierre Fraigniaud、Amos Korman 和 Adi Ros´en。带建议的在线计算。理论计算机科学，412(24)：2642–2656，2011年。[22] Shmuel Gal。一般搜索博弈。《以色列数学杂志》，12：32–45，1972年。[23] Sungjin Im、Ravi Kumar、Mahshid Montazer Qaem 和 Manish Purohit。具有频率预测的在线背包。载于第34届神经信息处理系统年度会议（NeurIPS）论文集，页码 2733–2743，2021年。[24] Thomas Kesselheim、Andreas T ¨onnis、Klaus Radke 和 Berthold V¨ocking。随机顺序模型下，原始方法优于在线装箱线性规划中的对偶方法。载于第46届ACM计算理论年度研讨会（STOC）论文集，页码 303–312，2014年。[25] Dennis Komm。《在线计算导论》。施普林格，2016年。[26] Russell Lee、Jessica Maghakian、Mohammad H. Hajiesmaili、Jian Li、Ramesh K. Sitaraman 和 Zhenhua Liu。带不可信建议的在线峰值感知能源调度。载于第12届ACM未来能源系统国际会议（eEnergy）论文集，页码 107–123。ACM，2021年。[27] Bin Li 和 Steven CH Hoi。在线投资组合选择：综述。《 ACM 计算调查》（ CSUR ），46(3)：1–36，2014年。

[28] Tongxin Li、Ruixiao Yang、Guannan Qu、Guanya Shi、Chenkai Yu、Adam Wierman 和 Steven H. Low。针对不可信预测的线性二次控制中的鲁棒性与一致性。《ACM Meas. Anal. Comput. Syst.》，6(1):18:1–18:35，2022年。[29] Alejandro L´opez-Ortiz、Spyros Angelopoulos 和 Angele Hamel。随时问题求解中契约算法的最优调度。《J. Artif. Intell. Res.》，(51):533–554，2014年。[30] Julian Lorenz、Konstantinos Panagiotou 和 Angelika Steger。用于期权定价的最优 k 搜索算法。《Algorithmica》，55(2):311–328，2009年。[31] Thodoris Lykouris 和 Sergei Vassilvitskii。基于机器学习建议的竞争性缓存。《J. ACM》，68(4):24:1–24:25，2021年。[32] Florence Jessie MacWilliams 和 Neil James Alexander Sloane。《纠错码理论》，第16卷。爱思唯尔，1977年。[33] Arya Mazumdar 和 Barna Saha。带噪查询聚类。载于《第31届神经信息处理系统会议（NeurIPS）论文集》，第5788–5799页。2017年。[34] Michael Mitzenmacher 和 Sergei Vassilvitskii。带预测的算法。载于《超越算法最坏情况分析》，第646–662页。剑桥大学出版社，2020年。[35] Manish Purohit、Zoya Svitkina 和 Ravi Kumar。通过机器学习预测改进在线算法。载于《第32届神经信息处理系统会议（NeurIPS）论文集》，第9661–9670页，2018年。[36] Ronald L. Rivest、Albert R. Meyer、Daniel J. Kleitman、Karl Winklmann 和 Joel Spencer。应对二分搜索过程中的错误。《J. Comput. Syst. Sci.》，20(3):396–404，1980年。[37] Daniel Sleator 和 Robert E. Tarjan。列表更新与分页规则的摊销效率。《Communications of the ACM》，28:202–208，1985年。[38] Alexander Wei 和 Fred Zhang。学习增强型在线算法的最优鲁棒性-一致性权衡。载于《第33届神经信息处理系统会议（NeurIPS）论文集》，2020年。[39] Yinfeng Xu、Wenming Zhang 和 Feifeng Zheng。在线时间序列搜索问题的最优算法。《理论计算机科学》，412(3):192–197，2011年。[40] Yunhong Zhou、Deeparnab Chakrabarty 和 Rajan Lukose。关键词拍卖和在线背包问题中的预算约束竞价。载于《互联网与网络经济学国际研讨会（WINE）论文集》，第566–576页。施普林格，2008年。

A 附录

A.1 第2节的省略内容

定理 3 的证明。我们将把 MINCYCLIC(n, k, H) 归约为以下在 [36] 中研究过的博弈：

IDENTIFY(m, H) 博弈：在该博弈中，x 是 $\{0, 1, \ldots, m-1\}$ 中的一个整数，其中某个已知的 m，目标是在最多 H 次查询可能得到错误回答的情况下，用尽可能少的查询识别出 x。

我们将在分析中使用以下结果：

引理 15. *[36]* 识别 x 所需的查询数量 $Q(m, H)$，在 IDENTIFY(m, H) 的一个实例中满足

$$\min\{k' | 2^{k'} \geq m \cdot \left(\!\!\binom{k'}{H}\!\!\right)\} \leq Q(m, H) \leq \min\{k | 2^{k-H} \geq m \left(\!\!\binom{k-H}{H}\!\!\right)\}.$$

给定 MINCYCLIC(n, k, H) 的一个实例，我们构造一个 IDENTIFY(m, H) 的实例，其中 $m = 2^{k-H} / \left(\!\!\binom{k-H}{H}\!\!\right)$（注意，两个实例的 H 相同）。将区间 $[0, \ldots, n-1]$ 划分为 m个互不相交的子区间，每个子区间的长度至多为 $\lceil n/m \rceil$。令 x 为 $[0, \ldots, n-1]$ 中满足$A[x] = 0$的索引，并令 I_x 表示包含 x的区间。WEIGHTING 策略针对 [36] 提出查询 q，询问“对于某个 $b \in \{0, m-1\}$，$I_x \leq b$ 是否成立？”，这对应于 MINCYCLIC(n, k, H) 实例中的查询 $\mu(x)$，询问“$x \leq f(b)$ 是否成立？”，其中 $f(b)$ 是区间 I_b中的最大价值。注意，只有当 $\mu(q)$ 的答案为“是”时，q 的答案才是“是”。随后将对 $\mu(q)$ 的回答提供给 WEIGHTING，由其更新状态并进行下一次查询。对于如上定义的 m，我们有 $2^{k-H} \geq m \cdot \left(\!\!\binom{k-H}{H}\!\!\right)$；使用 WEIGHTING，我们可以通过 k 次查询找到 I_x。接下来，我们返回 I_x中的最大整数 $f(x)$。鉴于 x 位于 I_x 中，且区间长度至少为 $\lceil n/m \rceil$，我们得出返回的索引 j 满足 $A[j] =\leq \lceil n/m \rceil = \lceil n \left(\!\!\binom{k-H}{H}\!\!\right) / 2^{k-H} \rceil$。

□

定理 4 的证明。考虑如下游戏，它被定义为 CONTINIOUSSEARCH(k, H) 游戏，唯一的区别在于，目标是猜出一个尽可能接近某个 $r \in [0, n)$ 的值，其中 n 固定（为了证明的目的，我们可以认为 n 足够大）。收到对 k 个查询的响应后，提问者返回一个数 $r' \in [0, n)$，目标是最小化 $|r' - r|$。我们将证明从这个游戏出发的一个归约，以帮助我们建立下界。反设存在一种策略，记为 ALG，用于 SEARCH(n, k, H)，它返回一个满足 $\pi(e) < \lfloor n \left(\!\!\binom{k}{H}\!\!\right) / 2^k \rfloor - 1$ 的元素 e。我们基于 ALG 为连续游戏中的提问者设计一种策略。给定定义连续游戏实例 G 的值 r、k 和 H，该实例所处的连续区间为 $I = [0, n)$，我们构造一个在包含 n 个元素的空间 A 上的 SEARCH(n, k, H) 实例 S，其中 k 和 H 的值相同。考虑一个双射映射 β，它将 A ($i \in \{0, \ldots, n-1\}$) 中排名为 i 的元素映射到 I 中的一个区间 $\beta(i) = [i, i+1)$。类似地，定义一个双射映射 μ，对应于为 G 提出的查询和为 S 提出的查询。任何属于为 S 提出的子集查询 q 的索引范围 $[i, j]$，都会映射到为 G 提出的查询 $\mu(q)$ 中的一个区间 $[i, j+1)$。令 r 表示 G 中被搜索的值，并令 x 表示 A 中满足 r 属于 $\beta(x)$ 的一个索引。为了搜索 r，我们考虑 ALG 为 S 提出的查询；对于任意此类查询 q，我们向 G 提出 $\mu(q)$。随后将 $\mu(q)$ 的响应提供给 ALG，使其更新状态并提出下一个查询。

回想一下，我们假设 ALG 输出一个元素 e，使得 $\pi(e) < \lfloor n \left(\!\!\binom{k}{H}\!\!\right) / 2^k \rfloor - 1$，并且 $\beta(\pi(e)) = [\pi(e)\ \pi(e) + 1)$。作为 G的输出，我们返回 $r' = \pi(e) + 1$ 作为 G的答案。注意，在 I中，β 的值域中恰好有 $e+1$ 个区间位于 r 和 r' 之间。也就是说$|r - r'| < \lfloor n \left(\!\!\binom{k}{H}\!\!\right) / 2^k - 1 \rfloor + 1 \leq n(\left(\!\!\binom{k}{H}\!\!\right) / 2^k$。然而，这与引理 1 矛盾。 □

A.2 第 4 节中省略的内容

定理 7 的证明。我们将定理 3 的算法应用于 $\mathcal{X}_{b,2^k}$中所有序列的索引集合，其中 $n = 2^k$，$b > 1$ 将在之后选定。输出是 $\mathcal{X}_{b,2^k}$ 中一个策略的索引，该策略在这些序列中的排名不超过

U。根据 $\mathcal{X}_{b,2^k}$ 的定义，特别是其循环性质，这意味着选定的策略能够以至多比 $\mathcal{X}_{b,2^k}$中的最佳策略大 b^U 倍的成本发现目标。由此可知，所选策略的竞争比至多为 $\frac{b^{2^k+1+U}}{b^{2^k}-1}$。当 $b=(\frac{2^k+U+1}{U+1})^{1/2^k}$ 时，该表达式取得最小值，由此得到我们的策略的竞争比至多为

$$\frac{1+U}{2^k}\left(1+\frac{2^k}{1+U}\right)^{1+\frac{1+U}{2^k}}.$$

□

定理 8 的证明。考虑一个 p-并行策略 X，它由 p 个竞价策略 $X_0,\ldots,X_{p-1}$ 定义，每个策略都在一个专用处理器上运行。令 $x_{j,i}$ 表示 X_j 中的出价 i；如果 $x_{j,i}$ 先于 j 中的出价 $x_{j,i'}$，则称 $i<i'$。我们将 X_j 中某个出价的前缀成本定义为在 X_j 中先于该出价的所有出价的价值之和。对于 $j\in[0,\ldots,p-1]$，我们用 $u_X(c,j)$ 表示 X_j 中满足以下条件的最大出价的价值：该出价的前缀成本与其价值之和不超过 c。我们还将 $u_{X,\phi}(c)$ 定义为集合 $\{u_X(c,j)\}_{j=0}^{p-1}$ 中第 $(\phi+1)$ 大的量。

考虑对 X 中所有出价进行任意索引，即第 i 个出价是 X_j 中的第 m 个出价，其中某个 m,j。我们将该出价表示为一对数 (C_i,D_i)，其中 C_i 是所属序列中先于出价 i 的所有出价的成本，而 D_i 是该出价本身（即其价值）。请注意，这种表示忽略了该出价所属的具体序列，因为正如我们将看到的，这对证明而言并不重要。给定一个表示为 (C_i,D_i) 的出价，我们定义 d_i 等于 $u_{X,\phi}(C_i+D_i)$：我们称这个值为相对于 D_i、在 X 中第 $(\phi+1)$ 大的出价。

回顾 X 表示 $\mathbb{N}$ 中所有出价值按非递减顺序排列后构成的序列。因此，X 中的每个出价都根据其长度映射到该序列中的一个元素（平局时任意打破）。

固定一个形式为 $b_{i_0}=(C_{i_0},D_{i_0})$的出价 i_0，不失一般性地假设 b_{i_0}属于序列 X_0。令 $c=C_{i_0}+D_{i_0}$。对于所有 $m\in[1,p-1]$，令$b_{i_m}=(C_{i_m},D_{i_m})$表示 X_m中满足其前缀成本与出价价值之和不超过 c的最大出价。对于每个 $m\in[0,p-1]$，定义I_m 为 $\mathbb{N}$中满足以下条件的索引集合：$i\in I_m$ 当且仅当存在一个价值为 x_i 的出价，使得其前缀成本加上x_i 不超过 c。根据竞争比的定义，我们有

$$\mathrm{Cr}(X)\geq\frac{\sum_{i\in I_m}x_i}{d_{i_m}},\qquad\text{for all } m\in[0,p-1].$$

因此，

$$\mathrm{Cr}(X)\geq\max_{0\leq m\leq p-1}\frac{\sum_{i\in I_m}x_i}{d_{i_m}},$$

并利用对所有 a、b、c、$d>0$均成立的性质$\max\{a/b,c/d\}\geq\frac{a+b}{c+d}$，我们得到

$$\mathrm{Cr}(X)\geq\frac{\sum_{m=0}^{p-1}\sum_{i\in I_m}x_i}{\sum_{m=0}^{p-1}d_{i_m}}.\tag{4}$$

接下来，我们将从下方界定式 (4) 中分数的分子，并从上方界定其分母。我们先给出一个有用的观察：对于成本 c（前文已定义），不失一般性地，我们可以假设，任何价值为 d_{i_0} 或更低的出价，其前缀成本都不超过 c 减去该出价的价值。这由 d_{i_0} 的定义可知：如果存在这样的出价，那么只需从 X 中“移除”该出价，

便可得到一个竞争比不更差的 p 并行序列（换言之，这样的出价毫无用处，而且可以构造出一个不包含该出价、竞争比不高于 X 的序列）。

利用上述观察可知，式 (4) 的分子包含所有价值不超过 d_{i_0} 的出价作为加数，此外还至少包含 $\phi+1$ 个不小于 d_{i_0} 的出价（其中 ϕ 个出价来自 d_{i_0} 的定义，另一个是出价 b_{i_0}）。令 q 表示一个满足 $d_{i_0}=\bar{x}_q$ 的指标，则有

$$\sum_{m=0}^{p-1}\sum_{i\in I_m} x_i \geq \sum_{i=0}^{q+\phi+1}\bar{x}_i.$$

现在，我们利用定义中隐含的单调性，对分母给出上界。这里涉及相对于给定出价价值的（$\phi+1$）最大价值，以及出价 b_{i_0}、……、$b_{i_{p-1}}$ 的定义。具体而言，对于每个满足 $m\in[1,p-1]$ 的出价 b_{i_m}，都有 $d_{i_m}\leq d_{i_0}$。因此可得

$$\sum_{m=0}^{p-1} d_{i_m} \leq \sum_{i=q}^{q-(p-1)}\bar{x}_i.$$

结合这两个界，可得

$$\mathrm{Cr}(X)\geq \sup_{0\leq q<\infty}\frac{\sum_{i=0}^{q+\phi+1}\bar{x}_i}{\sum_{i=q}^{q-(p-1)}\bar{x}_i}.$$

在证明的最后一步，我们将用到搜索理论中的一个结果，即 Gal 函数定理，陈述如下：

定理 16（Gal [22]）。设 q 为正整数，且 $X=(x_i)_{i=0}^{\infty}$ 为正数序列，满足 $\sup_{n\geq 0} x_{n+1}/x_n<\infty$ 和 $\alpha_X>0$。假设 F_i 是满足以下性质的泛函序列：

(1) $F_i(X)$ 仅取决于 x_0、x_1、$\cdots x_{i+q}$,

(2) 对所有正数序列 X, $F_i(X)$ 关于每个变量都是连续的,

(3) 对所有 $a>0$, $F_i(aX)=F_i(X)$,

(4) 对所有正数序列 X、Y, $F_i(X+Y)\leq \max(F_i(X)\ F_i(Y))$, 并且

(5) $F_{i+j}(X)\geq F_i(X^{+j})$, *for all* $j\geq 1$, *where* $X^{+j}=(x_j,x_{j+1},\ldots)$.

Then

$$\sup_{0\leq k<\infty} F_k(X)\geq \sup_{0\leq k<\infty} F_k(G_{\alpha_X}),$$

其中 G_a 定义为几何序列 $(a^i)_{i=0}^{\infty}$。

现在定义泛函 $F_q(\bar{X})=\frac{\sum_{i=0}^{q+\phi+1}\bar{x}_i}{\sum_{i=q}^{q-(p-1)}\bar{x}_i}$，对每个 q 均成立。该泛函满足定理 16 的条件 (1)-(5)（参见 [4] 中的例 7.3）。根据 Gal 定理，可得

$$\mathrm{Cr}(X)\geq \sup_{0\leq q<\infty}\frac{\sum_{i=0}^{q+\phi+1}\alpha_{\bar{X}}^{i}}{\sum_{i=q}^{q-(p-1)}\alpha_{\bar{X}}^{i}}.$$

如果 $\alpha_{\bar{X}}\leq 1$，那么很容易证明，上述表达式表明 $\mathrm{Cr}(X)=\infty$；例如参见 [29]。否则，也就是说，如果 $\alpha_{\bar{X}}>1$，经过一些简单计算即可得到所需结果。 □

上界与下界的比较。我们比较第 4.1.1 节和第 4.1.2 节中的上界与下界。定义 f为函数 $f(x) = \frac{1}{x}(1+x)^{1+1/x}$，并注意到 f关于 x单调递减，且 $f(1) = 4$，以及 $\lim_{x\to\infty} f(x) = 1$。于是，定理 7 的上界等于 $f(2^k/(U+1))$，而定理 9 的下界等于 $f(L)$，其中 U、L 的定义见相应定理的陈述。

对于每个 $y > x$，我们都有

$$\frac{f(y)}{f(x)} = \frac{\frac{1}{y}(1+y)^{1+1/y}}{\frac{1}{x}(1+x)^{1+1/x}} \le \frac{(1+y)^{1/y}}{(1+x)^{1/x}}.$$

此外，运用一些更基础的微积分知识，

$$\log\frac{f(y)}{f(x)} \le \log\frac{(1+y)^{1/y}}{(1+x)^{1/x}} \le \log\frac{y^{1/y}}{x^{1/x}} = \frac{1}{y}\log y - \frac{1}{x}\log x.$$

我们将使用上述不等式来比较 $f(2^k/(U+1))$ 和 $f(L)$。为简化计算，我们假设上界为 $f(2^k/U)$，因为当 k 变大时，分子中加上的“1”几乎不会影响竞争比。出于同样的原因，我们忽略 U 表达式中的上取整。利用式 (1) 的二项式系数部分和近似值，并定义 $\tau = k/H$，可得上界与下界之比 UB/LB 满足

$$\log\frac{\mathrm{UB}}{\mathrm{LB}} \le \frac{\sqrt{8k\tau(1-\tau)}k(1-\tau)(1-\mathcal{H}(\frac{\tau}{1-\tau}))}{2^{k(1-\tau)(1-\mathcal{H}(\frac{\tau}{1-\tau}))}} - \frac{k(1-\mathcal{H}(\tau))}{2^{k(1-\mathcal{H}(\tau))}},$$

□

A.3 第 5 节中省略的材料

定理 10 的证明。注意，在线算法和 OPT 都接受了比例为 1 的重物品。因此，重物品对算法和 OPT 收益的贡献相同，记为 Δ；我们有 $\delta \ge (1-c^*)d_x \ge (1-c_y)d_x$；这是因为 $c^* \in [c_{y-1}, c_y)$，且所有重物品的密度都大于 d_x。算法用关键物品填满大小为 c 的预留空间，这些关键物品的密度至少为 d_{x-1}；而 OPT 用关键物品填充大小为 $c^* < c_y$ 的空间，这些关键物品的密度至多为 d_x。因此，我们有

$$\begin{aligned}\mathrm{Cr}(A) \le \frac{\Delta + c_y\cdot d_x}{\Delta + c_{y-1}\cdot d_{x-1}} &\le \frac{(1-c_y)d_x + c_y\cdot d_x}{(1-c_y)d_x + c_{y-1}\cdot d_{x-1}} \\ &= \frac{1}{1-(c_y - c_{y-1}/\beta)} = \frac{1}{1-c_1} = \frac{\beta^m - 1}{\beta^{m-1}-1}.\end{aligned}$$

□

定理 11 的证明。反设存在一个算法 A，其竞争比优于 $C(k,H)$。我们将证明，随后可以在 FIND(k,H) 游戏中使用 A，通过 k 次查询识别出 $\{1,\dots,p\}$ 中的一个未知值，这与定理 2 的上界矛盾。

固定使 $g_m(\beta)$ 最小的 (s,m') 的值，并满足 $s.m' \le p$，令 $m = m'+1$。定义 $p = s\cdot m$；我们有 $p > \lceil 2^k / \left(\!\binom{k}{H}\!\right) \rceil$，否则，数对 (s,m) 会使 $g_m(\beta)$ 的值更小（注意，$g_m(\beta)$

是 m 的递减函数）。假设我们要识别一个未知值 $z \in \{1, \ldots, s \cdot m\}$；这等价于寻找一个满足 $x \in \{1, \ldots s\}$ 和 $y \in \{1, \ldots, m\}$ 的数对 (x, y)。对于 $i \in \{1, \ldots, s\}$，令 $d_i = L \cdot \beta^i$。此外，对于 $j \in \{0, \ldots, m\}$，定义 c_j，使得以下条件成立：

$$c_0 = \min\{1/\beta, 1 - 1/\beta\}, \quad c_m = \max\{1/\beta, 1 - 1/\beta\},$$
$$r = \frac{c_1 + \beta(1 - c_1)}{c_0 + \beta(1 - c_0)} = \frac{c_2 + \beta(1 - c_2)}{c_1 + \beta(1 - c_1)} = \ldots = \frac{c_{i+1} + \beta(1 - c_{i+1})}{c_i + \beta(1 - c_i)} = \ldots = \frac{c_{m-1} + \beta(1 - c_{m-1})}{c_m + \beta(1 - c_m)}.$$

可以验证，$r = (\frac{\beta^2 - \beta + 1}{2\beta + 1})^{1/m}$。对于任意 $x \in \{1, \ldots, s\}$ 和 $y \in \{1, \ldots, m\}$，构造如下输入序列：

$$\sigma_{x,y} = (1, d_0), (1, d_1), (1, d_2), \ldots, (1, d_{x-1}), (c_y, d_x).$$

这里，(a, b)表示一个物品序列，其中所有物品的大小均为无穷小尺寸 ϵ，密度为 b，总大小为 a。最优解用密度为 d_x的物品填满容量 c_y，并用密度为 d_{x+1}的物品填满剩余的 $1 - c_y$ 容量。我们有

$$\text{OPT}(\sigma_{x,y}) = (1 - c_y) \cdot d_{x-1} + c_y \cdot d_x.$$

接下来，我们将说明如何利用竞争比优于 $C(k, H)$的算法 A，正确求出未知值 $x \in \{1, \ldots s\}$ 和 $y \in \{1, \ldots, m\}$。假设下一个物品的密度为 d_α。令 w_α 表示密度为 $\leq d_{\alpha-1}$的物品的总大小，并假设对于某个 $q \in \{[1, \ldots, m\},$，有 $w_\alpha \in [1 - c_q, 1 - c_{q-1})$，也就是说，$1 - w_\alpha \in (c_{q-1}, c_q]$。定义 $\Delta_\alpha = c_{q+1} - c_q$。当算法 A 的背包空余空间小于 Δ_α时，算法便会“猜测” $x = \alpha$和 $y = q$。接下来，我们将证明 A 会在某个时刻作出这些猜测，且所作猜测正确。为导出矛盾，假设 A 没有作出猜测，或者至少有一个猜测不正确。我们将证明 A 的竞争比会大于 $C(k, H)$。假设对于某个 $q \in \{1, \ldots, m\}$，有 $w_x \in [1 - c_q, 1 - c_{q-1})$，也就是说，$1 - w \in (c_{q-1}, c_q]$。需要考虑四种可能性：

A 不进行猜测：由于 A 不进行猜测，背包中的空余空间至少为$\Delta_x = c_{q+1} - c_q$。此外，密度为 $\leq d_{x-1}$ 的物品的总大小至少为 $1 - c_q$。因此，密度为 d_x 的物品对背包价值的贡献至多为 $c_q \cdot d_x$，而其他物品的贡献至多为 $(1 - c_q - \Delta_x)d_{x-1} = (1 - c_{q+1})d_{x-1}$。因此，A 的背包总价值至多为 $(1 - c_{q+1})d_{x-1} + c_q d_x \leq (1 - c_{y+1})d_{x-1} + c_y d_x$。我们可以写成：

$$\text{Cr}(A) \geq \frac{(1 - c_y)d_{x-1} + c_y d_x}{(1 - c_{y+1})d_{x-1} + c_y d_x} > r.$$

对 y 的错误猜测：假设算法停止，但对 y 作出了错误猜测，即 $q \neq y$。首先，假设 $q \leq y - 1$，也就是说，A 为密度为 d_x 的物品预留的空间过少。密度为 $\leq d_{x-1}$ 的物品的总大小至少为 $1 - c_q$。因此，密度为 d_x 的物品的总大小至多为c_q。所以，背包的最终价值至多为 $(1 - c_q)d_{x-1} + c_q d_x \leq (1 - c_{y-1})d_{x-1} + c_{y-1} d_x$。我们可以写成：

$$\text{Cr}(A) \geq \frac{(1 - c_y)d_{x-1} + c_y d_x}{(1 - c_{y-1})d_{x-1} + c_{y-1} d_x} = r.$$

接下来，假设 $p \geq y + 1$，也就是说，为密度为 d_x 的物品预留了过多空间，其中一部分空间处于空闲状态。背包中密度至多为 d_{x-1} 的物品的总大小将为

至多为 $1-c_{q+1}$，而密度为 d_x 的物品的总大小至多为 $1-c_q$。因此，我们有$A(\sigma_x,\ y) \le (1-c_{q+1})d_{x-1}+c_q d_x \le (1-c_{y+1})d_{x-1}+c_y d_x$。我们可以写成

$$\mathrm{Cr}(A) \ge \frac{(1-c_y)d_{x-1}+c_y d_x}{(1-c_{y+1})d_{x-1}+c_y d_x} > r.$$

对 x 的错误猜测：假设 $p=y$ 但 $\alpha < x$（注意，α 不可能大于 x）。当算法将容量 c_y 装满密度为 α 的物品，剩余部分装入密度为 $d_{\alpha-1}$ 的物品时，背包的价值最大，即 $A(\sigma_x,\ y) \le (1-c_y)d_{\alpha-1}+\beta c_y d_\alpha \le (1-c_y)d_{x-2}+\beta c_y d_{x-1}$。我们可以写成

$$\mathrm{Cr}(A) \ge \frac{(1-c_y)d_{x-1}+c_y d_x}{(1-c_y)d_{x-2}+\beta c_y d_{x-1}} = \beta > r.$$

总之，只要 $\mathrm{Cr}(A) < g_m(\beta)$，就可以使用 A 正确猜出 x 和 y 的值，也就是说，它可以通过 k 次查询识别 $z \in \{1,\dots,p\}$。然而，这与定理 2 的下界矛盾。因此，关于 A 的竞争比的初始假设不成立，我们得出 $\mathrm{Cr}(A) \ge g_m(\beta)$。

□

A.4 第 6 节的省略材料

定理 12 的证明。考虑定理 5 和定理 7 的算法（上界），其包含 l 不完美建议位。根据假设，建议错误 η 至多为 $(1/3-c)l$，因此建议比特中至多有比例为 $1/3-c$的比特出错。由第 3.1 节和第 4.1.3 节中的讨论可知，只要

$$l(1-(1/3-c))(1-\mathcal{H}(\frac{1/3-c}{1-(1/3-c)}) > k,$$

对于所有足够大的 k，结论便成立。 □

定理 13 的证明。定义 p_1 和 p_2，使得它们分别满足 $M/p_1 = (M/m)^\rho$和 $p_2/m = (M/m)^\rho$，并注意到 $m \le p_1 \le p_2 \le M$。那么，当且仅当时间序列搜索算法将其保留价格设为某个 $p \in [p_1,p_2]$时，该算法是 r-鲁棒的。证明思路与定理 5 的证明类似；我们不再利用查询在 $[m,M]$ 中寻找合适的保留价格，而是转而在 $[p_1,p_2]$中寻找保留价格。特别地，根据 p_1的定义，p_2，我们有 $p_2/p_1 = (M/m)^{2\rho-1}$。 □

定理 14 的证明。对于上界，证明与定理 7 的证明类似。唯一的区别在于，必须在以下条件下优化 b：$\mathcal{X}_{b,\ 2^k}$中的每个策略都必须单独具备 r-鲁棒性。我们知道，基数为 b 的在线竞价几何策略的竞争比至多为 $b^2/(b-1)$。由于每个策略都是基数为 b^{2^k} 的几何策略，因此，只要 $b^{2^{k+1}}/(b^{2^k}-1) \le r$，$\mathcal{X}_{b,\ 2^k}$ 中的每个策略都是 r-鲁棒的，因而不完美建议所选出的策略也具有这一性质。

对于下界，我们借助 [6] 中证明的以下性质：如果集合 X_0、.....、X_{l-1} 中的所有 l 出价序列都具有 r-鲁棒性，那么必有 $\alpha_{\bar{X}}^2/(\alpha_{\bar{X}}-1) \le r$。回顾一下，$\bar{X}$ 是 l 个策略中所有出价的并集所构成的序列，并按非递减顺序排列。接着，证明沿用定理 9 的证明思路，并注意到当$l=2^k$时，竞争比达到最小。 □