arXiv:2602.10296v1 [cs.CC] 10 Feb 2026

计算压缩不动点的二次加速

Xi Chen* Yuhao Li† MihalisYannakakis‡哥伦比亚大学 哥伦比亚大学 哥伦比亚大学
xichen@cs.columbia.edu yuhaoli@cs.columbia.edu mihalis@cs.columbia.edu

摘要

我们研究求取压缩映射 $f\colon [0,1]^k \mapsto [0,1]^k$ 在 ℓ_∞-范数和 ℓ_1-范数下的 ε-不动点问题。对于这两种范数，我们都给出一种运行时间为 $O(\log^{\lceil k/2 \rceil}(1/\varepsilon))$ 的算法，其中 k 为任意常数。这改进了 Shellman 和 Sikorski [SS03]，针对 ℓ_∞-范数提出的此前最佳 $O(\log^k(1/\varepsilon))$ 时间算法，以及 Fearnley、Gordon、Mehta 和 Savani [FGMS20] 针对 ℓ_1-范数提出的此前最佳 $O(\log^k(1/\varepsilon))$ 时间算法。

*由美国国家科学基金会拨款 IIS-1838154、CCF-2106429 和 CCF-2107187 资助。
†由美国国家科学基金会拨款 IIS-1838154、CCF-2106429 和 CCF-2107187 资助。
‡由美国国家科学基金会拨款 CCF-2107187、CCF-2212233 和 CCF-2332922 资助。

1 引言

定义在度量空间 $(\mathcal{M}, d)$ 上的映射 $f\colon \mathcal{M} \mapsto \mathcal{M}$，若对某个$\gamma \in (0,1]$，满足 $d(f(x)\ f(y)) \leq (1-\gamma) \cdot d(x,y)$，且对所有 $x, y \in \mathcal{M}$ 均成立，则称其为 $(1-\gamma)$- 压缩映射。Banach [Ban22] 于 1922 年证明，每个压缩映射都有唯一的不动点。值得注意的是，他的证明具有构造性：从任意一点 $x \in \mathcal{M}$ 出发，反复应用压缩映射 f，总会收敛到其唯一不动点。Banach 的奠基性成果如今称为 Banach不动点定理，在过去一个世纪中深刻影响了众多领域，包括数学 [CL55, Nas56, G¨ un89],经济学和博弈论 [Sha53, SLJ89], 以及计算机科学 [Bel57, How60, Den67]。从算法角度来看，许多基础计算问题都可以归约为求唯一不动点或高精度近似不动点，这推动了对压缩映射不动点计算的研究 [STW93, HKS99, SS02, SS03, DP11, DTZ18, FGMS20, CLY25, BFG+25, HLSW25]。

研究压缩不动点计算时，最自然且重要的度量空间之一是立方体 $[0,1]^k$，其范数为 ℓ_∞-范数。特别是，形式化验证中的一些经典博弈，例如奇偶博弈 [EJ91, CJK+22]、平均收益博弈 [EM79, ZP96] 以及简单随机博弈 [Con92], ，都可以归约为在 ℓ_∞-范数下求一个 $(1-\gamma)$-压缩映射 $f\colon [0,1]^k \mapsto [0,1]^k$ 的 ε-不动点[1]；其中维度 k 是多项式级的，而 ε 和 γ 都关于博弈的输入规模呈指数级倒数。这些博弈不仅在验证中十分重要，其复杂性地位也引人关注：它们的判定问题属于 UP∩coUP，但是否存在多项式时间算法仍是长期悬而未决的问题。事实上，证明它们属于 UP ∩ coUP 的最简单方法基于压缩不动点，并且关键地利用了 Banach 定理 [Jur98, EY10] 所保证的唯一性。

近来，一个名为 ARRIVAL 的计算问题引起了广泛关注[DGK+17, Kar17, GHH+18, GHH21, Has25]。ARRIVAL 可以归约为求一个关于 ℓ_1-范数 [GHH+18, Has25], 的 $(1-\gamma)$-压缩映射 $f\colon [0,1]^k \mapsto [0,1]^k$ 的 ε-不动点，且 k （多项式）和 γ （指数的倒数）对输入规模的依赖也类似。它的复杂性状况也类似：已知 ARRIVAL 的判定版本属于 UP ∩coUP（再次说明，证明这一点的一种方法是利用压缩映射不动点），但目前尚无已知的多项式时间算法。

黑箱模型。 本文旨在拓展关于 $[0,1]^k$ 上压缩映射不动点的算法边界，同时涵盖 ℓ_∞-范数和 ℓ_1-范数。我们将在黑箱 模型下开展研究，在该模型中，算法可以通过查询预言机访问未知的压缩映射 $f\colon [0,1]^k \mapsto [0,1]^k$：给定输入 k， ε， γ 以及范数的选择（ ℓ_∞ 或 ℓ_1），算法在每一轮都可以向预言机发送一个点 $x \in [0,1]^k$，并接收作为响应的 $f(x)$。

我们用 $\textsc{Contraction}_\infty(\varepsilon,\ \gamma,\ k)$ 和 $\textsc{Contraction}_1(\varepsilon,\ \gamma,\ k)$ 表示这两个问题，它们分别对应 ℓ_∞-范数和 ℓ_1-范数。算法的查询复杂度 指的是其在最坏情况下使用的预言机查询次数，而本文关注的是时间复杂度，即假设预言机每次查询均在一个时间步内作答，找到一个 ε-不动点所需的总计算时间。显然，时间复杂度总是至少与查询复杂度一样大，但反过来则不一定成立。

关于压缩不动点计算的已有工作。 虽然巴拿赫值迭代法适用于任意度量空间（$\mathcal{M},\ d$），但要收敛到一个 ε-不动点需要 $O((1/\gamma)\log(1/\varepsilon))$ 时间，因此，上文讨论的所有应用都需要指数时间算法。

[1]若点 $x \in \mathcal{M}$ 是 f的 ε-近似不动点，或简称 ε-不动点，则有 $d(x,\ f(x)) \leq \varepsilon$。对于关于 ℓ_p-范数的 $[0,1]^k$，这意味着一个满足 $\|f(x) - x\|_p \leq \varepsilon$ 的 $x \in [0,1]^k$。

最早专门针对 $\textsc{Contraction}_\infty(\varepsilon,\gamma,k)$ 在 $[0,1]^k$ 上的 ℓ_∞-范数设计的算法由 Shellman 和 Sikorski 提出。当 $k=2$ [SS02] 时，他们给出了一个运行时间为 $O(\log(1/\varepsilon))$ 的算法（注意，简单的二分搜索就能在 $O(\log(1/\varepsilon))$ 时间内求解 $\textsc{Contraction}_\infty(\varepsilon,\gamma,1)$），后来又针对任意常数 k [SS03] 提出了一个运行时间为 $O(\log^k(1/\varepsilon))$ 的算法。对于 ℓ_1-范数，Fearnley、Gordon、Mehta 和 Savani [FGMS20] 给出了求解 $\textsc{Contraction}_1(\varepsilon,\gamma,k)$ 的算法，对于任意常数 k，其运行时间同样为 $O(\log^k(1/\varepsilon))$。

值得注意的是，当 k 为常数且另外两个主要参数 γ 和 ε 呈多项式关系（甚至它们的对数呈多项式关系）时，这些时间复杂度上界远优于巴拿赫值迭代法；在我们的工作之前，对于这种情形，它们仍是最好的上界。我们还注意到，这些上界不依赖于 γ；事实上，[SS02、SS03、FGMS20] 的算法能够解决更一般的问题：$\textsc{NonExp}_\infty(\varepsilon,k)$，即在 $[0,1]^k$ 上、采用 ℓ_∞-范数的非扩张[2] 映射中寻找一个 ε-不动点的问题；以及$\textsc{NonExp}_1(\varepsilon,k)$，即采用 ℓ_1-范数时的同一问题（见定义 3）。

在介绍本文关于时间复杂度的主要贡献之前，我们先指出，$\textsc{NonExp}$ 和 $\textsc{Contraction}$ 的查询复杂度方面已经取得了显著进展。在 [CLY25] 中，Chen、Li 和 Yannakakis 给出了一个求解 $\textsc{NonExp}_\infty(k,\varepsilon)$ 的算法，其查询复杂度为$O(k\log(1/\varepsilon))$。随后，Haslebacher、Lill、Schnider 和 Weber [HLSW25] 针对 $p\geq 1$ 的任意 ℓ_p-范数，得到了一个求解$\textsc{Contraction}_p(\varepsilon,\gamma,k)$ 的算法，使用 $O(k^2\log(1/(\varepsilon\gamma)))$ 次查询。然而，这些算法在每一轮中都使用蛮力搜索，以 $\Omega((1/\varepsilon)^k)$ 的时间复杂度寻找下一个查询点。

我们的贡献。我们的主要结果是，相较于 [SS03、FGMS20]针对 $\textsc{NonExp}_\infty(\varepsilon,k)$ 和 $\textsc{NonExp}_1(\varepsilon,k)$ 的算法，实现了二次加速，具体如下：

定理 1. 对于任意常数 k，存在一个 $O(\log^{\lceil k/2\rceil}(1/\varepsilon))$ 时间算法，用于求解 $\textsc{NonExp}_\infty(\varepsilon,k)$。

定理2. 对于任意常数 k，存在一个 $O(\log^{\lceil k/2\rceil}(1/\varepsilon))$ 时间算法，用于求解 $\textsc{NonExp}_1(\varepsilon,k)$。

两个定理中隐含的常数分别为 c^k，其中常数 c和 k^k。下面我们从高层次概述证明 定理 1和 定理2的算法。

1.1 技术概述

定理 1 的证明首先引入问题 $\textsc{NonExp}_\infty(\varepsilon,k)$ 的一种推广，称为$\textsc{NonExp}^\dagger_\infty(\varepsilon,k)$，其中未知的非扩张映射 f从 $[0,1]^k$ 映射到 $[-\varepsilon,1+\varepsilon]^k$（而非$[0,1]^k$）。仍然可以证明，$f$必定存在一个 ε-不动点 $x\in[0,1]^k$，满足 $\|f(x)-x\|_\infty\leq\varepsilon$；该问题的目标是求出 f的这样一个 ε-不动点。（$\textsc{NonExp}^\dagger_\infty(\varepsilon,k)$ 的形式化定义见定义 4。）鉴于 $\textsc{NonExp}_\infty(\varepsilon,k)$ 是 $\textsc{NonExp}^\dagger_\infty(\varepsilon,k)$ 的特例，只需给出一个求解 $\textsc{NonExp}^\dagger_\infty(\varepsilon,k)$ 的算法，其运行时间为 $O(\log^{\lceil k/2\rceil}(1/\varepsilon))$。

证明定理 1 分两步进行。首先，我们给出一个 $O(\log(1/\varepsilon))$-时间算法，用于$\textsc{NonExp}^\dagger_\infty(\varepsilon,k)$，其中 $k=1$ 或为 2。当 $k=1$ 时，可以通过简单的二分搜索解决；而当 $k=2$时，情况则更复杂，需要将原问题直接归约到 $\textsc{NonExp}_\infty(\varepsilon,2)$，已知该问题存在一个 $O(\log(1/\varepsilon))$-时间算法 [SS02]。

接下来，我们证明一个关于 $\textsc{NonExp}^\dagger_\infty(\varepsilon,k)$ 的新分解定理（定理 3）：如果 $\textsc{NonExp}^\dagger_\infty(\varepsilon,a)$ 和 $\textsc{NonExp}^\dagger_\infty(\varepsilon,b)$ 分别可以在 $t(\varepsilon,a)$ 和 $t(\varepsilon,b)$ 时间内求解，那么 $\textsc{NonExp}^\dagger_\infty(\varepsilon,a+b)$ 可以在 $O(t(\varepsilon,a)\cdot t(\varepsilon,b))$ 时间内求解。结合这两步即可得出定理 1。

据我们所知，这是关于压缩不动点的第一个分解定理；此前，此类分解定理仅针对单调映射中的塔斯基不动点

[2]若对所有 x，$y\in\mathcal{M}$，均有 $d(f(x)\ f(y))\leq d(x,\ y)$，则称映射 $f\colon\mathcal{M}\to\mathcal{M}$ 是非扩张的。

才为人所知 [FPS22, CL22]。从较高层面来看，给定对非扩张 $f: [0,1]^{a+b} \to [-\varepsilon, 1+\varepsilon]^{a+b}$ 的预言机访问，算法（算法 1）通过在一个自适应构建的非扩张映射 $g: [0,1]^b \to [-\varepsilon, 1+\varepsilon]^b$ 上运行一个 $t(\varepsilon, b)$ 时间算法 $\mathcal{A}_2$ 来工作，该映射是通过逐个查询构建的：对于 $\mathcal{A}_2$ 希望在第 t 轮查询的每个点 $q^t \in [0,1]^b$，算法 1 都需要处理 f，并返回答案 $v^t \in [-\varepsilon, 1+\varepsilon]^b$，作为 $g(q^t)$ 的结果。我们希望满足以下三个条件：

1. 算法 1 返回给 $\mathcal{A}_2$ 的所有关于 g 的结果 $(q^1, v^1) \ldots, (q^t, v^t)$，都与 g 是从 $[0,1]^b$ 到 $[-\varepsilon, 1+\varepsilon]^b$ 的非扩张映射这一假设相符。（这保证了 $\mathcal{A}_2$ 会在 $t(\varepsilon, b)$ 时间内终止，特别是会在 $t(\varepsilon, b)$ 轮次内终止，并得到一个 g 的 ε-不动点。）2. 当 $\mathcal{A}_2$ 在第 t 轮终止，且 q^t 是 g 的一个 ε-不动点，即 $\|v^t - q^t\|_\infty \le \varepsilon$ 时，我们可以高效地利用它，在一定时间内求得原映射 f 的一个 ε-不动点。3. 对于每一轮，给定来自 $\mathcal{A}_2$ 的查询 q^t，算法 1 计算其对 $\mathcal{A}_2$ 的响应 v^t 所花费的时间不应超过 $t(\varepsilon, a)$ 时间（即运行最佳算法以求解一个 $\textsc{NonExp}^\dagger_\infty(\varepsilon, a)$ 实例所需的时间）。

设计 算法 1 背后的挑战在于同时满足全部三个条件。特别是，为了返回一个满足第一个条件的 v^t，算法 1 不能只是运行一个 $t(\varepsilon, a)$- 时间算法 $\mathcal{A}_1$ 来寻找一个 ε- 不动点，其中 f 通过将最后的 $[0,1]^a$ 坐标设置为 b，将其限制在该 q^t立方体中。相反， 算法 1的做法是聚焦于一个经过精心构造的 a维立方体，该立方体由 $p^t_{\min}$ 和 $p^t_{\max}$ 定义（详见 算法 1 ）。采用稍微更困难的问题 $\textsc{NonExp}^\dagger_\infty$ （而不是 $\textsc{NonExp}_\infty$）对于分解定理的成功也至关重要；从本质上说，这类似于某些归纳只有在以稍微更强的基础和归纳假设开始时才能奏效的情形。

定理2的证明采用了类似的策略，即直接解决低维度情形，然后对更高维度使用递归。然而，由于 ℓ_∞ 范数和 ℓ_1 范数之间存在根本差异，这两个步骤都存在关键区别。特别是，我们的目标是给出一个用于 $\textsc{Contraction}_1(\varepsilon, \gamma, k)$ 的算法（算法 2），其运行时间为 $O(k \log^{\lceil k/2 \rceil}(1/\varepsilon\gamma))$；定理2则由从 $\textsc{NonExp}_1(\varepsilon, k)$ 到 $\textsc{Contraction}_1(\varepsilon/2, \varepsilon/(2k)\ k)$ 的归约得到（见观察 1）。

当 $k = 1$ 时，同样简单的二分搜索仍然适用（事实上，对于所有 $p \in [1, \infty,\ \ell_p$ 的行为完全相同）$\cup \{\infty\}$，当 $k = 1$ 时）。对于 $k = 2$ 的情况，我们的高层次 $O(\log(1/\varepsilon\gamma))$ 时间算法如下：假设我们查询中点 $x = (1/2, 1/2)$，并得到 $f(x)$，使得 $\|f(x) - x\|_1 > \varepsilon$。为简单起见，假设 $f(x)_1 \ge x_1$且 $f(x)_2 \ge x_2$。那么我们可以推断，对于所有满足 $\|x - y\|_1 \le \|f(x) - y\|_1$的 $y \in [0,1]^2$，都有$f(y) \ne y$；这是因为 $\|f(x) - f(y)\|_1 \le (1-\gamma)\|x - y\|_1 < \|f(x) - y\|_1$。接下来，关键思路是通过考察这些 y 的集合来分析它们，具体做法是寻找一个“主导坐标”，即一个满足 $|f(x)_i - x_i| \ge |f(x)_{3-i} - x_{3-i}|$的坐标 $i \in \{1, 2\}$。假设 $f(x)_1 - x_1 \ge f(x)_2 - x_2$，那么我们声称，对于所有 $y \in [0, 1/2] \times [0, 1]$，都有 $f(y) \ne y$。这是因为 $\|f(x) - y\|_1 - \|x - y\|_1 \ge (f(x)_1 - x_1) - |f(x)_2 - x_2| \ge 0$。直观来说，这意味着我们可以通过一次查询将搜索空间缩减一半，而剩余空间仍然是一个形状规整的矩形，因此我们知道下一次查询点应当是该矩形的中心。补充更多细节（例如如何利用条件 $\|f(x) - x\|_1 > \varepsilon$）后，我们得到了一个用于 $\textsc{Contraction}_1(\varepsilon, \gamma, 2)$ 的$O(\log(1/\varepsilon\gamma))$ 时间算法。

对于更高维度 $k > 2$，由于一些技术难题，目前尚不清楚能否应用分解方法。作为替代，我们直接将上述方法推广，通过识别一个“主导坐标”来分析剩余的搜索空间。这里，主导坐标的初步定义可以是满足 $|f(x)_i - x_i| \ge \sum_{j \ne i} |f(x)_j - x_j|$的 i。然而，很容易看出，在更高维度下，这样的 i 可能并不存在。事实证明，对于

要使该方法奏效，只需找到一个弱支配坐标，其定义为满足 i 的$|f(x)_i - x_i| \geq \sum_{j\neq i} |f(x)_j - x_j| - \varepsilon\gamma/4$。递归正是在这里发挥作用。我们固定前两个坐标，并在一个 $k-2$ 维矩形中递归地寻找一个关于剩余 $\varepsilon\gamma/4$的不动点 $k-2$。这样， $i=1$ 或 $i=2$ 将成为弱支配坐标。请注意，递归中近似精度的提升幅度 $\varepsilon\gamma/4$而非 ε 非常重要，这也是为什么定理2 中隐藏的常数为 k^k。

1.2 其他相关工作

长期以来，人们已知在 ℓ_2-范数下，求解压缩映射的近似不动点问题存在多项式时间算法 [STW93、HKS99]。然而，对于其他ℓ_p-范数（例如 ℓ_∞ 和 ℓ_1 本文研究的范数），已知的最佳时间界仍然关于 k呈指数级。

作为缺乏多项式时间算法的自然完全搜索问题，计算压缩不动点这一计算问题自然属于 TFNP [MP91] 的范畴。事实上，它是促使 CLS [DP11] 类被引入的动机问题之一。最近，为了更好地刻画具有内在唯一解的问题的复杂性， [FGMS20] 定义了子类 UEOPL，并证明（除其他问题外）计算分段线性压缩映射的精确不动点属于 UEOPL；这是我们在此考察的黑盒问题的一个较简单的白盒变体。就在最近，Batziou等人 [BFG+25]证明，在黑盒模型中，求解同时满足单调性和在 ℓ_∞-范数下为压缩映射的映射的近似不动点属于 UEOPL。他们还给出了一种时间复杂度为 $O(\log^{\lceil k/3\rceil}(1/\varepsilon))$ 的算法，其速度快于目前已知的针对仅单调映射的最佳算法 [CL22]，也快于针对仅满足压缩性的映射的最佳算法（[SS03] 以及本文的算法）。

尽管人们已投入大量精力研究TFNP 中关于 ℓ_p-范数下压缩不动点的刻画（例如，参见 [EY10、DP11、DTZ18、FGMS20、Hol21、FGHS23]），但目前仍不清楚它们是否对文献中研究的任何TFNP类都是困难问题。如前所述，压缩不动点的查询复杂度研究已取得重大进展 [CLY25、HLSW25]，其中尤其排除了从已知TFNP 类到ℓ_p-范数下压缩不动点的黑箱归约；有兴趣的读者可参阅 [CLY25]中的讨论，了解更多详情。

2 预备知识

我们首先给出压缩映射和非扩张映射的定义。

定义 *1*（压缩）。设 $0<\gamma<1$ 和 $(\mathcal{M}, d)$ 是一个度量空间。若对于所有 $x, y \in \mathcal{M}$，映射 $f: \mathcal{M} \mapsto \mathcal{M}$ 满足 $d(f(x), f(y)) \leq (1-\gamma)\cdot d(x, y)$，则称其为关于 $(\mathcal{M}, d)$ 的 $(1-\gamma)$- 压缩映射。若对于所有 $x, y \in \mathcal{M}$，映射 $f: \mathcal{M} \mapsto \mathcal{M}$ 满足 $d(f(x), f(y)) \leq d(x, y)$，则称其为非扩张映射。

根据巴拿赫不动点定理，每个压缩映射都有唯一的不动点，即 x^* with $f(x^*) = x^*$。根据布劳威尔不动点定理，关于 ℓ_p-范数的 $[0,1]^k$ 上的每个非扩张映射 $p \in [1,\infty)\cup\{\infty\}$ 都有一个不动点（不一定唯一）。本文研究在 k维立方体 $[0,1]^k$ 上，关于 ℓ_p-范数的 $(1-\gamma)$-压缩映射 f 的 ε-不动点的求解时间复杂度，其中 $p \in \{1,\infty\}$:

定义 2 ($\textsc{Contraction}_p(\varepsilon, \gamma, k)$)。在问题 $\textsc{Contraction}_p(\varepsilon, \gamma, k)$中，我们可以预言机访问一个关于 $[0,1]^k$、相对于 ℓ_p-范数的$(1-\gamma)$-压缩映射 f，即 f满足

$$\|f(x) - f(y)\|_p \leq (1-\gamma)\cdot\|x-y\|_p, \quad \textit{for all } x, y \in [0,1]^k.$$

目标是求出 f的一个 ε-不动点，即一个点 $x \in [0,1]^k$，使得 $\|f(x) - x\|_p \le \varepsilon$。

我们针对关于 ℓ_p-范数的 $[0,1]^k$ 上的非扩张映射，定义类似的问题：

定义 3 ($\textsc{NonExp}_p(\varepsilon, k)$)。在问题 $\textsc{NonExp}_p(\varepsilon, k)$中，我们获得关于 ℓ_p-范数的非扩张映射 $f: [0,1]^k \to [0,1]^k$ 的预言机访问，即 f 满足

$$\|f(x) - f(y)\|_p \le \|x - y\|_p, \quad \textit{for all } x, y \in [0,1]^k.$$

目标是求出 f 的一个 ε-不动点，即一个点 $x \in [0,1]^k$，使得 $\|f(x) - x\|_p \le \varepsilon$。

我们将使用以下记号。给定一个正整数 m，我们用 $[m]$ 表示集合$\{1, \dots, m\}$。给定两个向量 $\alpha, \beta \in \mathbb{R}^m$，若对于所有 $i \in [m]$，都有 $\alpha_i \le \beta_i$，则称 $\alpha \preceq \beta$。给定 $\alpha, \beta \in \mathbb{R}^m$，使得 $\alpha \preceq \beta$，我们用 $\mathcal{B}(\alpha, \beta) := \{x : \alpha \preceq x \preceq \beta\}$ 表示由 α 和 β 定义的盒。给定$\alpha, \beta \in \mathbb{R}^m$，使得 $\alpha \preceq \beta$ 且 $\varepsilon > 0$，我们用 $\mathcal{B}(\alpha, \beta, \pm\varepsilon)$ 表示

$$\mathcal{B}(\alpha, \beta, \pm\varepsilon) := \big\{x \in \mathbb{R}^m : (\alpha_1 - \varepsilon, \cdots, \alpha_m - \varepsilon) \preceq x \preceq (\beta_1 + \varepsilon, \cdots, \beta_m + \varepsilon)\big\}.$$

我们的主要算法将在第 3 节中针对 $\textsc{NonExp}_\infty(\varepsilon,\ k)$ 处理以下推广形式：

定义 4 ($\textsc{NonExp}^\dagger_\infty(\varepsilon,\ k)$)。在 $\textsc{NonExp}^\dagger_\infty(\varepsilon,\ k)$ 中，给定 $\alpha_{\min}$、$\alpha_{\max} \in [0,1]^k$，满足 $\alpha_{\min} \preceq \alpha_{\max}$，并且可通过预言机访问关于 ℓ_∞-范数的非扩张映射 $f: \mathcal{B}(\alpha_{\min},\ \alpha_{\max}) \to \mathcal{B}(\alpha_{\min},\ \alpha_{\max},\ \pm\varepsilon)$。目标是找到 f 的一个 ε-不动点，即一个点 $x \in \mathcal{B}(\alpha_{\min},\ \alpha_{\max})$，使得$\|f(x) - x\|_\infty \le \varepsilon$。注意，这样的点总是存在。为说明这一点，我们记 $g: \mathcal{B}(\alpha_{\min},\ \alpha_{\max}) \to \mathcal{B}(\alpha_{\min},\ \alpha_{\max})$ 表示通过如下方式截断 f 后得到的映射：

$$g(x)_i := \max\big(\min(f(x)_i, \alpha_{\max,i}), \alpha_{\min,i}\big), \quad \textit{for every } i \in [k].$$

容易验证，g 仍然是非扩张的，因此根据布劳威尔不动点定理，存在一个不动点 x。g 的任意不动点都必然是 f 的一个 ε-不动点。

3 $\textsc{NonExp}_\infty(\varepsilon,\ k)$ 的算法

为了证明 定理 1，只需给出一种算法，能够在 $O(\log^{\lceil k/2 \rceil}(1/\varepsilon))$ 时间内求解 $\textsc{NonExp}^\dagger_\infty(\varepsilon,\ k)$（这是 $\textsc{NonExp}_\infty(\varepsilon,\ k)$ 的推广）。为此，我们首先指出，当 $k \in \{1, 2\}$ 时，根据 Shellman 和 Sikorski [SS02]，$\textsc{NonExp}^\dagger_\infty(\varepsilon,\ k)$ 可在 $O(\log(1/\varepsilon))$ 时间内求解：

引理 1 。当 $k \in \{1, 2\}$ 时，存在一种算法，可在 $O(\log(1/\varepsilon))$ 时间内求解 $\textsc{NonExp}^\dagger_\infty(\varepsilon, k)$。

证明。首先，问题 $\textsc{NonExp}^\dagger_\infty(\varepsilon,\ 1)$ 可以通过二分搜索轻松求解，所需时间为 $O(\log(1/\varepsilon))$，因此下文将重点讨论 $\textsc{NonExp}^\dagger_\infty(\varepsilon,\ 2)$。

假设给定 $\alpha_{\min},\ \alpha_{\max} \in [0,1]^2$ 以及 $\alpha_{\min} \preceq \alpha_{\max}$，并可通过预言机访问一个非扩张映射 $f: \mathcal{B}(\alpha_{\min},\ \alpha_{\max}) \to \mathcal{B}(\alpha_{\min},\ \alpha_{\max},\ \pm\varepsilon)$，它采用 ℓ_∞-范数。我们将 $g: \mathcal{B}(\alpha_{\min},\ \alpha_{\max}) \to \mathcal{B}(\alpha_{\min},\ \alpha_{\max})$ 定义为 f 的截断，与上文相同：

$$g(x)_i := \max\big(\min(f(x)_i, \alpha_{\max,i}), \alpha_{\min,i}\big), \quad \text{for both } i \in \{1, 2\}.$$

鉴于 g 也是非扩张的，我们可以使用 Shellman 和 Sikorski 的算法 [SS02]，在时间 ε内求出 g 的一个 $O(\log(1/\varepsilon))$ 不动点。将这样的点记为 $x^* \in \mathcal{B}(\alpha_{\min},\ \alpha_{\max})$。

如果 $\|f(x^*) - x^*\|_\infty \le \varepsilon$，那么我们就完成了。否则，必定至少有一个 $f(x^*)$ 的坐标 $i \in \{1, 2\}$ 被截断，并且 $|f(x^*)_i - x^*_i| > \varepsilon$。不失一般性，假设对于两者 $f(x^*)_i \ge x^*_i$，我们分两种情况讨论：(1) 对两者 $f(x^*)_i \notin [\alpha_{\min,i}, \alpha_{\max,i}]$，以及 (2) 恰好存在一个坐标 $i \in \{1, 2\}$，使得 $f(x^*)_i \notin [\alpha_{\min,i}, \alpha_{\max,i}]$。

1. 对于第一种情况，我们知道对于两者 $f(x^*) > \alpha_{\max,\ i}$。我们证明 $\alpha_{\max}$ 必定是 f 的一个所需的 ε- 不动点。一方面，由于 $f(\alpha_{\max}) \in \mathcal{B}(\alpha_{\min},\ \alpha_{\max},\ \pm\varepsilon)$，我们知道对于两者 $f(\alpha_{\max})_i - \alpha_{\max,\ i} \le \varepsilon$。另一方面，鉴于$\|x^* - \alpha_{\max}\|_\infty \le \varepsilon$，我们有 $\|f(\alpha_{\max}) - f(x^*)\|_\infty \le \varepsilon$，因此，

$$f(\alpha_{\max})_i \ge f(x^*)_i - \varepsilon > \alpha_{\max,i} - \varepsilon$$

对两者 $i \in \{1, 2\}$ 都成立。这表明 $\|f(\alpha_{\max}) - \alpha_{\max}\|_\infty \le \varepsilon$。

2. 第二种情况更为复杂。不失一般性，假设 $f(x^*)_1 \le \alpha_{\max,1}$ 且 $f(x^*)_2 > \alpha_{\max,2}$。我们有 $f(x^*)_2 > x_2^* + \varepsilon > \alpha_{\max,2}$。考虑点 y，其中$y_1 = x_1^*$且 $y_2 = \alpha_{\max,2}$。因此$\|y - x^*\|_\infty = \alpha_{\max,2} - x_2^* < \varepsilon$，从而 $\|f(y) - f(x^*)\|_\infty < \varepsilon$。由此我们得到 $|f(y)_2 - y_2| \le \varepsilon$： $f(y)_2 - y_2 \le \varepsilon$ 由 $f(y)_2 \le \alpha_{\max,2} + \varepsilon = y_2 + \varepsilon$ 得出；$f(y)_2 \ge y_2 - \varepsilon$，因为否则 $\|f(y) - f(x^*)\|_\infty \ge |f(y)_2 - f(x^*)_2| > \varepsilon$，矛盾。

此外，我们有 $f(y)_1 > f(x^*)_1 - \varepsilon \ge x_1^* - \varepsilon = y_1 - \varepsilon$。若 $f(y)_1 \le y_1 + \varepsilon$，则 y 是 f 的所需 ε-不动点，我们便完成了。否则，我们有 $f(y)_1 > y_1 + \varepsilon$，并且我们将在下文证明，定义为 $y_2' := y_2 = \alpha_{\max,\ 2}$ 的 y'，以及

$$y_1' := \min(\alpha_{\max,1}, y_1 + \varepsilon) = \min(\alpha_{\max,1}, x_1^* + \varepsilon),$$

必须是 f 的所需 ε-不动点。

首先注意到 $\|y' - x^*\|_\infty,\ \|y' - y\|_\infty \le \varepsilon$，因此 $\|f(y') - f(x^*)\|_\infty,\ \|f(y') - f(y)\|_\infty \le \varepsilon$。类似的论证可得 $|f(y')_2 - y_2'| \le \varepsilon$。此外，我们还有

$$f(y')_1 \ge f(y)_1 - \varepsilon > y_1 \ge y_1' - \varepsilon.$$

最后，我们有 $f(y')_1 \le y_1' + \varepsilon$，因为：若 $y_1' = \alpha_{\max,\ 1}$，则 $f(y')_1 \le \alpha_{\max,\ 1} + \varepsilon = y_1' + \varepsilon$；若$y_1' < \alpha_{\max,\ 1}$，则 $f(y')_1 - f(x^*)_1 \le \varepsilon$，这意味着 $f(y')_1 \le f(x^*)_1 + \varepsilon \le y_1' + \varepsilon$。

至此，引理的证明完成。 □

定理 1 可由结合 引理 1 与以下分解定理得出：

定理 3. 假设 $\textsc{NonExp}_\infty^\dagger(\varepsilon, a)$ 可在 $q(\varepsilon, a)$ 次查询和 $t(\varepsilon, a)$ 时间内求解，且 $\textsc{NonExp}_\infty^\dagger(\varepsilon, b)$ 可在 $q(\varepsilon, b)$ 次查询和 $t(\varepsilon, b)$ 时间内求解。那么， $\textsc{NonExp}_\infty^\dagger(\varepsilon, a + b)$ 可在 $q(\varepsilon, a) \cdot q(\varepsilon, b)$ 次查询和 $O((q(\varepsilon, b) + t(\varepsilon, a)) \cdot t(\varepsilon, b))$ 时间内求解。

请注意，时间复杂性中存在一个额外项 $q(\varepsilon, b)$，这是由 算法 1 的第 5 行导致的。这并不成问题，因为在应用分解定理得到 定理 1时，我们可以选择 $b \le a$，使得 $q(\varepsilon, b)$ 项可以被吸收到 $t(\varepsilon, a)$ 项中。

定理 3 的证明。假设 $\mathcal{A}_1$ 是一个求解 $\textsc{NonExp}_\infty^\dagger(\varepsilon, a)$ 的算法，使用 $q(\varepsilon, a)$ 次查询和 $t(\varepsilon, a)$ 时间，并且 $\mathcal{A}_2$ 是一个求解 $\textsc{NonExp}_\infty^\dagger(\varepsilon, b)$ 的算法，使用 $q(\varepsilon, b)$ 次查询和 $t(\varepsilon, b)$ 时间。我们将证明算法 1 能够求解 $\textsc{NonExp}_\infty^\dagger(\varepsilon, a + b)$，使用 $q(\varepsilon, a) \cdot q(\varepsilon, b)$ 次查询和$O((q(\varepsilon, b) + t(\varepsilon, a)) \cdot t(\varepsilon, b))$ 时间。

概述。 总体而言，算法 1 在函数 $\mathcal{A}_2$ 上 $\textsc{NonExp}_\infty^\dagger(\varepsilon,\ b)$ 运行，该函数为 $h:\ \mathcal{B}(\beta_{\min},\ \beta_{\max}) \to \mathcal{B}(\beta_{\min},\ \beta_{\max},\ \pm\varepsilon)$，它利用非扩张映射的输入即时构造

$$f : \mathcal{B}\big((\alpha_{\min}, \beta_{\min}), (\alpha_{\max}, \beta_{\max})\big) \to \mathcal{B}\big((\alpha_{\min}, \beta_{\min}), (\alpha_{\max}, \beta_{\max}), \pm\varepsilon\big).$$

算法 1： 通过 $\textsc{NonExp}^{\dagger}_{\infty}(\varepsilon,\ a)$ 和 $\textsc{NonExp}^{\dagger}_{\infty}(\varepsilon,\ b)$ 求解 $\textsc{NonExp}^{\dagger}_{\infty}(\varepsilon,\ a+b)$ 的算法

输入： 一个 $\varepsilon \in (0,\ 1]$, $\alpha_{\min},\ \alpha_{\max} \in [0,1]^a$, $\beta_{\min},\ \beta_{\max} \in [0,1]^b$，满足 $\alpha_{\min} \preceq \alpha_{\max}$ 且 $\beta_{\min} \preceq \beta_{\max}$，以及对一个非扩张映射的预言机访问 $f\colon \mathcal{B}((\alpha_{\min},\ \beta_{\min}),\ (\alpha_{\max},\ \beta_{\max})) \to \mathcal{B}((\alpha_{\min},\ \beta_{\min}),\ (\alpha_{\max},\ \beta_{\max}) \pm\varepsilon)$。

输出： 一个点 $x^* \in \mathcal{B}((\alpha_{\min},\ \beta_{\min}),\ (\alpha_{\max},\ \beta_{\max}))$，满足 $\|f(x^*) - x^*\|_\infty \le \varepsilon$。

设 $\mathcal{A}_1$ 是一个用于 $\textsc{NonExp}^{\dagger}_{\infty}(\varepsilon,\ a)$ 的算法，且 $\mathcal{A}_2$ 是一个用于 $\textsc{NonExp}^{\dagger}_{\infty}(\varepsilon,\ b)$ 的算法。

对于 $t = 1, 2, \cdots$ **执行**

对于之前的每一轮次 $r \in [t-1]$，令 $q^r \in \mathcal{B}(\beta_{\min}, \beta_{\max})$ 为 $\mathcal{A}_2$ 所查询的点且 $v^r \in \mathcal{B}(\beta_{\min}, \beta_{\max}, \pm\varepsilon)$ 为答案。

给定序列 $((q^1, v^1)\ \cdots, (q^{t-1}, v^{t-1}))$，令 $q^t \in \mathcal{B}(\beta_{\min}, \beta_{\max})$ 为第 t 次查询 $\mathcal{A}_2$ 的。

令（当 $t = 1$ 时，设 $p^1_{\min} \leftarrow \alpha_{\min}$ 且 $p^1_{\max} \leftarrow \alpha_{\max}$）

$$p^t_{\min,\ i} \leftarrow \max\left(\alpha_{\min,\ i},\ \max_{r\in[t-1]}(y_i^{(r,*)} - \|q^t - q^r\|_\infty)\right) \text{ 对所有 } i \in [a]$$

and

$$p^t_{\max\, i} \leftarrow \min\left(\alpha_{\max,i},\ \min_{r\in[t-1]}(y_i^{(r,*)} + \|q^t - q^r\|_\infty)\right) \text{ 对所有 } i \in [a].$$

定义一个新函数 $g : \mathcal{B}(p^t_{\min}\, p^t_{\max}) \to \mathcal{B}(p^t_{\min}\, p^t_{\max}, \pm\varepsilon)$ 如下：

$$g(x) = \Big(f(x, q^t)_1, \cdots, f(x, q^t)_a\Big), \quad \text{for every } x \in \mathcal{B}(p^t_{\min}, p^t_{\max}).$$

运行算法 $\mathcal{A}_1$ 以找到一个解 $y^{(t,*)} \in \mathcal{B}(p^t_{\min}\, p^t_{\max})$ 对 $\textsc{NonExp}^{\dagger}_{\infty}(\varepsilon, a)$ 在 g 上。

将 $v^t \in [0,1]^b$ 构造为对 q^t 的查询结果：

$$v^t_{j-a} = f(y^{(t,*)}, q^t)_j, \quad \text{for each } j \in [a+b] \setminus [a].$$

如果 $\|v^t - q^t\|_\infty \le \varepsilon$，**返回** $(y^{(t,*)},\ q^t)$。

设 $q^1, \ldots, q^{t-1} \in \mathcal{B}(\beta_{\min}, \beta_{\max})$ 是 $t-1$ 目前为止已进行的查询，其中 $\mathcal{A}_2$ 已进行查询，且某个 $t \ge 1$；设 $v^1, \ldots, v^{t-1} \in \mathcal{B}(\beta_{\min}, \beta_{\max}, \pm\varepsilon)$ 是在 h 上得到的查询结果。令 $q^t \in \mathcal{B}(\beta_{\min}, \beta_{\max})$ 为 $\mathcal{A}_2$ 在第 t 轮进行的新查询。我们将利用 f 以及先前的查询和相应结果 $((q^1, v^1)\ \cdots, (q^{t-1}, v^{t-1}))$，给出一个 $v^t \in \mathcal{B}(\beta_{\min}, \beta_{\max}, \pm\varepsilon)$，作为对查询 $h(q^t)$ 的回答，使得

1. 引理 3：所有结果（q^1，v^1）.....，（q^t，v^t）均符合非扩张性质，即$\|v^r - v^{r'}\|_\infty \le \|q^r - q^{r'}\|_\infty$ 对所有 r，$r' \in [t]$；并且

2. 引理 4：当 q^t 是 $\textsc{NonExp}^{\dagger}_{\infty}$（$\varepsilon$，$b$）在 h上的解，即 $\|v^t - q^t\|_\infty \le \varepsilon$时，我们可在原始输入函数 $\textsc{NonExp}^{\dagger}_{\infty}$（$\varepsilon$，$a+b$上得到一个解 f。

注意，答案的成对一致性（引理 4）意味着存在一个非扩张函数 h使得 $h(q^r) = v^r$ 对所有 $r \in [t]$都成立，这由 [CLY25]的引理 11 得出。为了获得答案 v^t 以回答查询 q^t，直观来说，我们固定最后的 b 个坐标为 q^t，并且只针对前 ε个坐标寻找一个 f -不动点 a。我们通过在一个函数上运行 $\mathcal{A}_1$ 来实现这一点

g： $\mathcal{B}(p^t_{\min},\ p^t_{\max}) \to \mathcal{B}(p^t_{\min},\ p^t_{\max},\ \pm\varepsilon)$，其中 $\alpha_{\min} \preceq p^t_{\min} \preceq p^t_{\max} \preceq \alpha_{\max}$。函数 g 定义为将 f 投影到最后 b 个坐标固定为 q^t 的切片上。此过程最关键的部分是使用前几轮中的点对（$q^r,\ v^r$），$r \in [t-1]$，来初始化搜索空间 $\mathcal{B}(p^t_{\min},\ p^t_{\max})$，而不是使用完整域 $\mathcal{B}(\alpha_{\min},\ \alpha_{\max})$。

正确性。我们证明关于 算法 1 的一系列引理：

引理 2。在每一轮 t 中，我们有

1. $\alpha_{\min} \preceq p^t_{\min} \preceq p^t_{\max} \preceq \alpha_{\max}$；
2. g 关于 ℓ_∞-范数是一个非扩张映射；并且
3. $g(x) \in \mathcal{B}(p^t_{\min}, p^t_{\max}, \pm\varepsilon)$ 对所有 $x \in \mathcal{B}(p^t_{\min}, p^t_{\max})$ 均成立；4. $\|f(y^{(t,*)}, q^t)_{[a]} - y^{(t,*)}\|_\infty \le \varepsilon$。

证明。我们对 t 进行归纳来证明该引理。当 $t=1$ 时，基例显然成立，因为 $p^1_{\min} = \alpha_{\min}$ 且 $p^1_{\max} = \alpha_{\max}$。在归纳步骤中，我们证明上述四项对于 $t>1$ 均成立。根据归纳假设，我们假设对于任意 $r \in [t-1]$，都有 (1) $\alpha_{\min} \preceq p^r_{\min} \preceq p^r_{\max} \preceq \alpha_{\max}$，以及 (4) $\|f(y^{(r,*)},\ q^r)_{[a]} - y^{(r,*)}\|_\infty \le \varepsilon$。我们依次证明每一项。

第一项。$\alpha_{\min} \preceq p^t_{\min}$ 和 $p^t_{\max} \preceq \alpha_{\max}$ 可直接由定义得出。要证明 $p^t_{\min} \preceq p^t_{\max}$，只需证明

$$y^{(r,*)}_i - \|q^t - q^r\|_\infty \le y^{(r',*)}_i + \|q^t - q^{r'}\|_\infty$$

对所有 $i \in [a]$ 和 $r,\ r' \in [t-1]$均成立。等价地，我们要证明

$$\|y^{(r,*)} - y^{(r',*)}\|_\infty \le \|q^t - q^{r'}\|_\infty + \|q^t - q^r\|_\infty$$

对所有 $r < r' \in [t-1]$均成立。任取 $r < r' \in [t-1]$。根据归纳假设，我们有$p^r_{\min} \preceq p^r_{\max}$。于是我们知道 $y^{(r',*)} \in \mathcal{B}(p^r_{\min},\ p^r_{\max})$。这意味着

$$y^{(r,*)}_i - \|q^{r'} - q^r\|_\infty \le y^{(r',*)}_i \le y^{(r,*)}_i + \|q^{r'} - q^r\|_\infty$$

对所有 $i \in [a]$均成立。因此我们知道 $\|y^{(r,*)} - y^{(r',*)}\|_\infty \le \|q^{r'} - q^r\|_\infty \le \|q^t - q^r\|_\infty + \|q^t - q^{r'}\|_\infty$.

第二项。对于第二项，请注意

$$\|g(x) - g(z)\|_\infty = \|f(x, q^t) - f(z, q^t)\|_\infty \le \|x - z\|_\infty.$$

其中最后一个不等式是因为 f 是一个非扩张函数。

第三项。对于第三项，我们首先注意到 $\mathcal{B}(p^t_{\min},\ p^t_{\max})$ 的一个等价定义，即：

$$\mathcal{B}(p^t_{\min}, p^t_{\max}) = \left(\bigcap_{r \in [t-1]} \mathcal{B}(y^{(r,*)}, y^{(r,*)}, \pm\|q^t - q^r\|_\infty) \right) \cap \mathcal{B}(\alpha_{\min}, \alpha_{\max}). \tag{1}$$

类似地，$\mathcal{B}(p^t_{\min}, p^t_{\max}, \pm\varepsilon)$ 的一个等价定义如下：

$$\mathcal{B}(p^t_{\min}, p^t_{\max}, \pm\varepsilon) = \left(\bigcap_{r \in [t-1]} \mathcal{B}(y^{(r,*)}, y^{(r,*)}, \pm(\|q^t - q^r\|_\infty + \varepsilon)) \right) \cap \mathcal{B}(\alpha_{\min}, \alpha_{\max}, \pm\varepsilon). \tag{2}$$

回顾一下，f: $\mathcal{B}((\alpha_{\min},\beta_{\min}),(\alpha_{\max},\beta_{\max})) \to \mathcal{B}((\alpha_{\min},\beta_{\min}),(\alpha_{\max},\beta_{\max}) \pm\varepsilon)$，因此有 $g(x) \in \mathcal{B}(\alpha_{\min},\alpha_{\max},\pm\varepsilon)$。

现在固定任意 $r \in [t-1]$ 和任意 $x \in \mathcal{B}(y^{(r,*)}, y^{(r,*)}, \pm\|q^t - q^r\|_\infty)$。我们将证明

$$g(x) \in \mathcal{B}(y^{(r,*)}, y^{(r,*)}, \pm(\|q^t - q^r\|_\infty + \varepsilon)).$$

首先，注意到

$$\|g(x) - y^{(r,*)}\|_\infty = \|f(x,q^t)_{[a]} - y^{(r,*)}\|_\infty \le \|f(x,q^t)_{[a]} - f(y^{(r,*)},q^r)_{[a]}\|_\infty + \|f(y^{(r,*)},q^r)_{[a]} - y^{(r,*)}\|_\infty.$$

由于 f 是非扩张的，我们知道 $\|f(x,q^t)_{[a]} - f(y^{(r,*)},q^r)_{[a]}\|_\infty \le \|(x,q^t) - (y^{(r,*)},q^r)\|_\infty$。由于 $x \in \mathcal{B}(y^{(r,*)}, y^{(r,*)}, \pm\|q^t - q^r\|_\infty)$，我们知道 $\|(x,q^t) - (y^{(r,*)},q^r)\|_\infty \le \|x - y^{(r,*)}\|_\infty$。此外，由于 $r < t$，根据归纳假设，我们有 $\|f(y^{(r,*)},q^r)_{[a]} - y^{(r,*)}\|_\infty \le \varepsilon$，因此，

$$\|g(x) - y^{(r,*)}\|_\infty \le \|x - y^{(r,*)}\|_\infty + \varepsilon, \quad \text{which implies} \quad g(x) \in \mathcal{B}(y^{(r,*)}, y^{(r,*)}, \pm(\|q^t - q^r\|_\infty + \varepsilon)).$$

这部 分完成了第三项的证明，根据方程（1）和方程（2）。

第四项。鉴于 $g(x)$: $\mathcal{B}(p^t_{\min}, p^t_{\max}) \to \mathcal{B}(p^t_{\min}, p^t_{\max}, \pm\varepsilon)$，我们知道，算法 $\mathcal{A}_1$ 输出的点 $y^{(t,*)}$ 是一个 ε-不动点。因此 $\|g(y^{(t,*)}) - y^{(t,*)}\|_\infty \le \varepsilon$，这意味着$\|f(y^{(t,*)}, q^t)_{[a]} - y^{(t,*)}\|_\infty \le \varepsilon$。 □

现在我们可以证明正确性所需的两个引理了算法 1：

引理 3。对于每一轮次 $t \ge 1$ 以及每个 $r < t$，我们都有 $\|v^r - v^t\|_\infty \le \|q^r - q^t\|_\infty$。

证明。由于 v^t 由 f 定义，因此显然有 $v^t \in \mathcal{B}(\beta_{\min}, \beta_{\max}, \pm\varepsilon)$。注意

$$\|v^t - v^r\|_\infty = \max_{j \in [a+b]\setminus[a]} |f(y^{(t,*)},q^t)_j - f(y^{(r,*)},q^r)_j| \le \|(y^{(t,*)},q^t) - (y^{(r,*)},q^r)\|_\infty \le \|q^t - q^r\|_\infty,$$

其中，最后一个不等式由 $y^{(t,*)} \in \mathcal{B}(y^{(r,*)}, y^{(r,*)}, \pm\|q^t - q^r\|_\infty)$ 得出。 □

引理 4. 对于每一轮 t，若 $\|v^t - q^t\|_\infty \le \varepsilon$，则 $\|f(y^{(t,*)},q^t) - (y^{(t,*)},q^t)\|_\infty \le \varepsilon$。

证明。 根据算法 $\mathcal{A}_1$ 的正确性，我们知道

$$\max_{j \in [a]} |f(y^{(t,*)},q^t)_j - (y^{(t,*)},q^t)_j| \le \varepsilon.$$

若 $\|v^t - q^t\|_\infty \le \varepsilon$，则我们知道

$$\max_{j \in [a+b]\setminus[a]} |f(y^{(t,*)},q^t)_j - (y^{(t,*)},q^t)_j| \le \varepsilon.$$

于是引理得证。 □

查询复杂度与时间复杂度。对于算法 1 的每一轮次，算法 $\mathcal{A}_1$ 被调用一次，其使用 $q(\varepsilon,a)$ 次查询以及 $t(\varepsilon,a)$ 时间。外层算法 $\mathcal{A}_2$ 的轮次不超过$q(\varepsilon,b)$，这意味着算法 1 的查询复杂度至多为 $q(\varepsilon,a) \cdot q(\varepsilon,b)$。在每一轮次中，第 5 行耗时 $O(q(\varepsilon,b))$，因此算法 1 的每一轮次需要 $O(q(\varepsilon,b) + t(\varepsilon,a))$ 时间。总体时间复杂度为 $O((q(\varepsilon,b) + t(\varepsilon,a)) \cdot t(\varepsilon,b))$。 □

算法 2： 用于 $\textsc{Contraction}_1(\varepsilon,\ \gamma,\ k)$ 的递归算法

输入： 一个偶数 k，以及对一个 $(1-\gamma)$-压缩 f 的预言机访问：$[0,1]^k \mapsto [0,1]^k$ 关于 ℓ_1-范数。

输出： 一个点 $x \in [0,1]^k$，使得 $\|f(x) - x\|_1 \le \varepsilon$。

令 $p_{\min} = (0,0)$ 与 $p_{\max} = (1,1)$。

对于 $t = 1, 2, \cdots$ **执行**

  令 $p_{\mathrm{mid}} = (p_{\min} + p_{\max})/2$。

  如果 $k > 2$，**则**

    按如下方式定义一个新函数 ϕ：$[0,1]^{k-2} \mapsto [0,1]^{k-2}$

$$\phi(y) := \big(f(p_{\mathrm{mid}}, y)_3, \ldots, f(p_{\mathrm{mid}}, y)_k\big).$$

    递归地找到一个点 $y^* \in [0,1]^{k-2}$，使得 $\|\phi(y^*) - y^*\|_1 \le \varepsilon\gamma/4$。

  令 $x = (p_{\mathrm{mid}}, y^*)$ 并**查询** $f(x)$。// 当 $k = 2$ 时，y^* 定义为空。

  如果 $\|f(x) - x\|_1 \le \varepsilon$，**那么**

    返回 x，作为 f 的 ε-不动点。

  令 $i \in \{1,2\}$ 满足 $|f(x)_i - x_i| \ge |f(x)_{3-i} - x_{3-i}|$（任意打破平局）。

  // 找到一个（弱）主导坐标。

  如果 $f(x)_i > x_i$，**那么**

    令 $p_{\min,\ i} = x_i + \varepsilon\gamma/8$。

  else

    令 $p_{\max,\ i} = x_i - \varepsilon\gamma/8$。

4 $\textsc{NonExp}_1(\varepsilon,\ k)$ 的算法

本节给出一个用于 算法 2，其针对 $\textsc{Contraction}_1(\varepsilon,\ \gamma,\ k)$ 的时间复杂度为$O(k\log^{\lceil k/2\rceil}(1/\varepsilon\gamma))$。 定理2 由下文的观察 1 即可推出：

观察 1。 设 $f\colon [0,1]^k \mapsto [0,1]^k$ 是关于 ℓ_1-范数的非扩张映射。考虑映射 $g\colon [0,1]^k \mapsto [0,1]^k$，定义为 $g(x) := (1-\varepsilon/(2k))f(x)$。显然，$g$ 是关于 ℓ_1-范数的$(1-\varepsilon/(2k))$压缩。令 $x \in [0,1]^k$ 为任意满足 $\|g(x) - x\|_1 \le \varepsilon/2$的点。我们有

$$\varepsilon/2 \ge \|g(x) - x\|_1 = \|(1-\varepsilon/(2k))f(x) - x\|_1 \ge \|f(x) - x\|_1 - (\varepsilon/2k)\cdot\|f(x)\|_1 \ge \|f(x) - x\|_1 - \varepsilon/2.$$

这给出了从 $\textsc{NonExp}_1(\varepsilon, k)$ 到 $\textsc{Contraction}_1(\varepsilon/2, \varepsilon/(2k)\ k)$的黑箱归约，同时具备查询高效和时间高效的特点。

不失一般性，我们假设 k 为偶数；否则，可以轻松地补充一个维度，并令 $f(x)_{k+1} = 0$ 对所有 $x \in [0,1]^{k+1}$成立。我们先概述一下 算法 2。

算法 2 的概述。 令 $x^* = \mathrm{Fix}(f)$ 为 f 的未知不动点。该算法维护两个变量 $p_{\min},\ p_{\max} \in [0,1]^2$，分别对应前两个坐标，并确保在执行过程中始终满足 $p_{\min,\ i} \le x^*_i \le p_{\max,\ i}$，对两个 $i \in \{1,2\}$ 均成立。换言之，x^* 在前两个坐标上的投影位于由 $p_{\min}$ 和 $p_{\max}$ 定义的矩形内。初始时，$p_{\min}$ 设为 (0，0)，$p_{\max}$ 设为 (1，1)，因此显然满足这一条件。

每一轮中，算法将前两个坐标固定为 $p_{\text{mid}} = (p_{\min} + p_{\max})/2$，并递归地为其余 $k-2$ 个坐标寻找一个近似不动点 y^*，其近似保证提高为 $\varepsilon\gamma/4$，而非 ε。随后令 $x = (p_{\text{mid}},\ y^*)$。

算法接着查询 $f(x)$。如果 x 结果是一个 ε-不动点，那么我们就完成了。否则，我们知道

$$\|f(x) - x\|_1 > \varepsilon \quad \text{and} \quad \sum_{i=3}^{k} |f(x)_i - x_i| \le \varepsilon\gamma/4.$$

因此，直观而言，$f(x)$ 与 x 之间的大部分偏差都来自前两个坐标。接着，我们选择 $i \in \{1,2\}$，使得 $|f(x)_i - x_i| \ge |f(x)_{3-i} - x_{3-i}|$。于是我们知道

$$|f(x)_i - x_i| \ge \sum_{j \ne i} |f(x)_j - x_j| - \varepsilon\gamma/4,$$

这意味着，i 上的偏差“几乎”支配了所有其他坐标上的偏差。这将使我们能够将 $p_{\max,i} - p_{\min,i}$ 缩小一半（事实上，略多于一半）。

现在，我们用下面的技术引理将其形式化。

引理 5. 设 $k \ge 2$。设 $f\colon [0,1]^k \mapsto [0,1]^k$ 是一个 $(1-\gamma)$-压缩映射，并设 $x^* = \texttt{Fix}(f)$。若点 $x \in [0,1]^k$ 满足 $\|f(x) - x\|_1 > \varepsilon$，但 $\sum_{j=3}^{k} |f(x)_j - x_j| \le \varepsilon\gamma/4$，则有

- $x_i^* > x_i + \varepsilon\gamma/8$ 如果 $f(x)_i > x_i$；
- $x_i^* < x_i - \varepsilon\gamma/8$ 如果 $f(x)_i < x_i$，

其中 $i \in \{1,2\}$ 满足 $|f(x)_i - x_i| \ge |f(x)_{3-i} - x_{3-i}|$。

证明。不失一般性，假设 $i = 1$，即 $|f(x)_1 - x_1| \ge |f(x)_2 - x_2|$。为进一步简化记号，我们还假设 $f(x)_1 > x_1$，以及 $f(x)_2 \ge x_2$。于是我们的目标是证明$x_1^* > x_1 + \varepsilon\gamma/8$，并由对称性可得该引理。

鉴于 f 是一个 $(1-\gamma)$-压缩，且 x^* 是 f 的不动点，我们有

$$\|f(x) - x^*\|_1 = \|f(x) - f(x^*)\|_1 \le (1-\gamma) \cdot \|x - x^*\|_1. \tag{3}$$

此外，我们注意到 $\|x - x^*\|_1 > \varepsilon/2$，否则我们有

$$\|f(x) - x\|_1 \le \|f(x) - x^*\|_1 + \|x^* - x\|_1 = \|f(x) - f(x^*)\|_1 + \|x^* - x\|_1 \le 2\|x^* - x\|_1 \le \varepsilon.$$

反证法。为了导出矛盾，我们假设 $x_1^* \le x_1 + \varepsilon\gamma/8$。由此，我们将给出 $\|x - x^*\|_1 - \|f(x) - x^*\|_1$ 的一个非平凡上界，如下所示：

$$\|x - x^*\|_1 - \|f(x) - x^*\|_1 \le \varepsilon\gamma/2. \tag{4}$$

然而，由（3），我们知道

$$\gamma\|x - x^*\|_1 \le \|x - x^*\|_1 - \|f(x) - x^*\|_1 \le \varepsilon\gamma/2,$$

这意味着 $\|x - x^*\|_1 \le \varepsilon/2$，从而导致矛盾。

为证明 方程（4），我们将 $\|x - x^*\|_1 - \|f(x) - x^*\|_1$ 划分为两部分：

$$A = \sum_{j=1,2} \left(|x_j - x_j^*| - |f(x)_j - x_j^*|\right) \quad \text{and} \quad B = \sum_{j=3}^{k} \left(|x_j - x_j^*| - |f(x)_j - x_j^*|\right).$$

首先，显然由三角不等式和引理的假设可得，

$$B \le \sum_{j=3}^{k} |x_j - f(x)_j| \le \varepsilon\gamma/4.$$

为了界定 A，注意到根据假设，$\|f(x) - x\|_1 > \varepsilon$ 但 $\sum_{j=3}^{k} |f(x)_j - x_j| \le \varepsilon\gamma/4$，我们有 $f(x)_1 - x_1 > (\varepsilon - \varepsilon\gamma/4)/2$。这意味着

$$x_1^* \le x_1 + \varepsilon\gamma/8 < x_1 + (\varepsilon - \varepsilon\gamma/4)/2 < f(x)_1.$$

No还注意到，由于 $x_1^* \le x_1 + \varepsilon\gamma/8$，我们有 $|x_1 - x_1^*| \le (x_1 - x_1^*) + \varepsilon\gamma/4$。因此

$$\begin{aligned}
A &= |x_1 - x_1^*| - |f(x)_1 - x_1^*| + |x_2 - x_2^*| - |f(x)_2 - x_2^*| \\
&= |x_1 - x_1^*| - (f(x)_1 - x_1^*) + |x_2 - x_2^*| - |f(x)_2 - x_2^*| \\
&\le \varepsilon\gamma/4 + (x_1 - x_1^*) - (f(x)_1 - x_1^*) + |x_2 - x_2^*| - |f(x)_2 - x_2^*| \\
&= \varepsilon\gamma/4 + (x_1 - f(x)_1) + |x_2 - x_2^*| - |f(x)_2 - x_2^*| \\
&\le \varepsilon\gamma/4 + x_1 - f(x)_1 + f(x)_2 - x_2 \\
&\le \varepsilon\gamma/4,
\end{aligned}$$

其中，最后一个不等式由 $|f(x)_1 - x_1| \ge |f(x)_2 - x_2|$ 得出。至此，我们证明了 $\|x - x^*\|_1 - \|f(x) - x^*\|_1 \le \varepsilon\gamma/2$，也完成了引理的证明。 □

根据引理 5，我们知道在每一轮结束时，对于两个$i \in \{1, 2\}$，都有 $p_{\min,\,i} \le x_i^* \le p_{\max,\,i}$。此外，如果一轮开始时 $\|p_{\max} - p_{\min}\|_\infty < \varepsilon\gamma/8$，那么该轮中第 9 行的 x 必须是 f 的一个 ε-不动点。由于初始时 $p_{\max,\,i} - p_{\min,\,i} = 1$，我们知道该算法必须在 $O(\log(1/\varepsilon\gamma))$ 轮内终止，以找到一个 ε-不动点。

该算法时间复杂度的递归表达式为

$$T(\varepsilon, \gamma, k) = O(\log(1/\varepsilon\gamma)) \cdot \big(O(1) + T(\varepsilon\gamma, \gamma, k-2)\big).$$

求解该递推关系可得 $T(\varepsilon, \gamma, k) = O(k \log^{\lceil k/2 \rceil}(1/\varepsilon\gamma))$.

5 讨论与未来方向

在本文中，我们给出了针对 $\textsc{NonExp}_\infty(\varepsilon, k)$ 和 $\textsc{NonExp}_1(\varepsilon, k)$ 的 $O(\log^{\lceil k/2 \rceil}(1/\varepsilon))$ 时间算法。一个自然的下一步是给出针对 $\textsc{NonExp}_\infty(\varepsilon, 3)$ 或 $\textsc{NonExp}_1(\varepsilon, 3)$ 的 $O(\log(1/\varepsilon))$ 时间算法。特别地，针对 $\textsc{NonExp}_\infty(\varepsilon, 3)$ 的此类算法将根据定理 3 蕴含针对一般 k 的 $O(\log^{\lceil k/3 \rceil}(1/\varepsilon))$ 时间算法。然而，仍存在一些挑战。从较高层面来看，针对 $\textsc{NonExp}_\infty(\varepsilon, 2)$ 和 $\textsc{NonExp}_1(\varepsilon, 2)$ 的 $O(\log(1/\varepsilon))$ 时间算法都源于一个关键观察：搜索空间始终可以保持为凸。当 $k = 3$ 时，这两个问题都不再如此，即使只进行一次查询之后也是如此（参见 [CLY25] 中关于 ℓ_∞-范数情形的图）。

另一个方向是理解时间复杂度与查询复杂度之间的权衡。请注意，存在高度查询高效的算法 [CLY25, HLSW25] 用于压缩不动点；然而，这些查询高效算法的已知实现都具有时间复杂度 $\Omega((1/\varepsilon)^k)$，这是由于其暴力性质。另一方面，当关注诸如 [SS02, SS03, FGMS20] 以及本文等工作中的时间高效算法时，它们的查询复杂度与时间复杂度基本相同。我们能否取得一些中间结果？例如，具体而言，能否给出一个针对 $\textsc{NonExp}_\infty(\varepsilon,3)$ 或 $\textsc{NonExp}_1(\varepsilon,3)$、时间复杂度为 $O(\log^2(1/\varepsilon))$、但仅使用 $O(\log(1/\varepsilon))$ 次查询的算法？

参考文献

[Ban22] 斯特凡·巴拿赫。关于抽象集合上的运算及其在积分方程中的应用。《数学基础》，3(1):133–181，1922。1[Bel57] 理查德·贝尔曼。一个马尔可夫决策过程。《数学与力学杂志》，第679–684页，1957。1[BFG+25] Eleni Batziou、John Fearnley、Spencer Gordon、Ruta Mehta 和 Rahul Savani。单调压缩映射。载于第57届 ACM 理论计算会议论文集，第507–517页，2025。1, 1.2[CJK+22] Cristian S. Calude、Sanjay Jain、Bakhadyr Khoussainov、李伟和 Frank Stephan。准多项式时间内判定奇偶博弈。《 SIAM 计算杂志》，51(2):17–152，2022。1[CL55] E. A. Coddington 和 N. Levinson。《常微分方程理论》。McGraw Hill，1955。1[CL22] Xi Chen 和 Yuhao Li。求解塔斯基不动点的改进上界。载于 EC，第1108–1118页。ACM，2022。1.1, 1.2[CLY25] Xi Chen、Yuhao Li 和 Mihalis Yannakakis。通过多项式数量的查询计算压缩映射的不动点。《J. ACM》，72(4)，2025年7月。1, 1.2, 9, 5[Con92] Anne Condon。随机博弈的复杂性。《信息与计算》，96(2):203–224，1992。1[Den67] Eic V. Denardo。动态规划背后的压缩映射。《SIAM 评论》，9(2):165–177，1967。1[DGK+17] J´érôme Dohrau、Bernd G¨artner、Manuel Kohler、Ji´rí Matoušek 和 Emo Welzl。ARRIVAL：一个属于 NP ∩ coNP 的零玩家图博弈。载于《离散数学之旅：献给 Ji´rí Matoušek》，第367–374页。Springer，2017。1[DP11] Constantinos Daskalakis 和 Christos H. Papadimitriou。连续局部搜索。载于第二十二届 ACM-SIAM 离散算法年度研讨会论文集（SODA 2011），第790–804页。SIAM，2011。1, 1.2

[DTZ18] Constantinos Daskalakis、Christos Tzamos 和 Manolis Zampetakis。巴拿赫不动点定理的逆命题及其 CLS 完全性。载于 Ilias Diakonikolas、David Kempe 和 Monika Henzinger（编），第50届 ACM SIGACT 计算理论研讨会论文集（STOC 2018），美国加利福尼亚州洛杉矶，2018年6月25日至29日，第44–50页。ACM，2018。1, 1.2[EJ91] E. A. Emerson 和 C. Jutla。树自动机、 μ-演算与确定性。载于 IEEE 计算机科学基础研讨会论文集，第368–377页，1991。1[EM79] Andrzej Ehrenfeucht 和 Jan Mycielski。平均收益博弈的位置策略。《国际博弈论杂志》，8(2):109–113，1979。1[EY10] Kousha Etessami 和 Mihalis Yannakakis。纳什均衡及其他不动点的复杂性。《SIAM 计算杂志》，39(6):2531–2597，2010。1, 1.2[FGHS23] John Fearnley、Paul Goldberg、Alexandros Hollender 和 Rahul Savani。梯度下降的复杂性：CLS = PPAD ∩ PLS。《J. ACM》，70(1):7:1–7:74，2023。1.2[FGMS20] John Fearnley、Spencer Gordon、Ruta Mehta 和 Rahul Savani。唯一势线终点。《计算机与系统科学杂志》，114:1–35，2020。（文档），1, 1.2, 5[FPS22] John Fearnley、D¨om¨ot¨or P´alv¨olgyi 和 Rahul Savani。求解塔斯基不动点的更快算法。《ACM 算法汇刊》（TALG），18(3):1–23，2022。1.1[GHH+18] Bernd G¨artner、Thomas Dueholm Hansen、Pavel Hub´acek、Karel Kr´al、Hagar Mosaad 和 Veronika Sl´ıvov´a。ARRIVAL：CLS 中的下一站。载于第45届国际自动机、语言与编程研讨会（ICALP 2018），第60–1页。Schloss Dagstuhl–Leibniz 信息学中心，2018。1[GHH21] Bernd G¨artner、Sebastian Haslebacher 和 Hung P Hoang。ARRIVAL 的亚指数时间算法。载于第48届国际自动机、语言与编程研讨会（ICALP 2021），第69–1页。Schloss Dagstuhl–Leibniz 信息学中心，2021。1[G¨un89] Matthias G¨unther。论 J. Nash 的嵌入定理。《数学新闻》，144(1):165–187，1989。1[Has25] Sebastian Haslebacher。ARRIVAL：递归框架-压缩。载于第52届国际自动机、语言与编程研讨会（ICALP 2025），第95–1页。Schloss Dagstuhl–Leibniz 信息学中心，2025。1[HKS99] Z. Huang、Leonid G. Khachiyan 和 Christopher (Krzysztof) Sikorski。近似求解弱收缩映射的不动点。《复杂性杂志》，15(2):200–213，1999。1, 1.2[HLSW25] Sebastian Haslebacher、Jonas Lill、Patrick Schnider 和 Simon Weber。 ℓ_p-压缩映射的不动点查询高效算法。载于 FOCS，2025。即将出版。1, 1.2, 5[Hol21] Alexandros Hollender。全搜索复杂性类的结构性结果及其在博弈论和优化中的应用。博士论文，英国牛津大学，2021。1.2

[How60] Ronald A Howard. 动态规划与马尔可夫过程。John Wiley，1960。1[Jur98] Marcin Jurdzi ´nski. 判定奇偶博弈的获胜者属于 UP ∩co-UP。Information Processing Letters，68(3):119–124，1998。1[Kar17] CS Karthik. 火车是否到达目的地：寻找见证的复杂性。Information Processing Letters，121:17–21，2017。1[MP91] Nimrod Megiddo and Christos H. Papadimitriou. 关于全函数、存在性定理与计算复杂性。Theor. Comput. Sci.，81(2):317–324，1991。1.2[Nas56] 约翰·纳什。黎曼流形的嵌入问题。Annals of mathematics，63(1):20–63，1956。1[Sha53] L. Shapley. 随机博弈。Proc. Nat. Acad. Sci.，39(10):1095–1100，1953。1[SLJ89] Nancy L Stokey and Robert E Lucas Jr. 经济动力学中的递归方法。哈佛大学出版社，1989。1[SS02] Spencer D. Shellman and Kris Sikorski. 用于不动点的二维二分包络算法。J. Complex.，18(2):641–659，2002。1，1.1，3，3，5[SS03] Spencer D. Shellman and Christopher (Krzysztof) Sikorski. 无穷范数不动点问题的递归算法。J. Complex.，19(6):799–834，2003。(document)，1，1.2，5[STW93] Christopher (Krzysztof) Sikorski, Chey-Woei Tsay, and Henryk Wozniakowski. 一种用于不动点计算的椭球算法。J. Complex.，9(1):181–200，1993。1，1.2[ZP96] Uri Zwick and Mike Paterson. 图上平均收益博弈的复杂性。Theor. Comput. Sci.，158(1&2):343–359，1996。1