arXiv:2601.06723v1 [cs.DS] 10 Jan 2026

使用期望预算进行分划拟阵的拟阵基测试近似

Lisa Hellerstein * Benedikt M. Plank † Kevin Schewior ‡

2026年1月13日

摘要

我们考虑以下随机布尔函数求值问题，它与文献中的若干问题密切相关。给定底集 E 上的一个拟阵 $\mathcal{M}$（采用紧凑表示），且每个元素 $i \in E$ 都以已知概率$p_i \in (0, 1)$独立激活。可以查询这些元素，查询后会揭示相应元素是否激活。目标是找到一种自适应查询策略，以确定是否存在一个基，其中所有元素都处于激活状态 $\mathcal{M}$ ，同时使查询次数的期望值最小。

当 $\mathcal{M}$ 是均匀拟阵时，这就是评估一个 k-取-n 函数的问题，该问题最早于20世纪70年代得到研究。这个问题已得到充分理解，并且存在一种最优的自适应策略，可以在多项式时间内计算出来。

将 $\mathcal{M}$ 改为分划拟阵后，我们证明了以往的方法无法给出常数因子近似。我们的主要成果是一种多项式时间常数因子近似算法，该算法为这一分划拟阵问题生成随机化策略。我们通过将一种新技术与若干成熟技术相结合，得到了这一结果。我们的算法在满足 期望 成本约束的情况下，自适应地交错处理若干个新型随机查询问题的实例。我们相信，这类问题本身就具有研究价值，将会激发后续工作，并有望获得更多应用。

*纽约大学计算机科学与工程系，纽约，美国（lisa.hellerstein@nyu.edu）。

†柏林，德国（b.plank@mail.de）。

‡科隆大学数学与计算机科学系，K¨oln，德国；南丹麦大学数学与计算机科学系，丹麦欧登塞 (k.schewior@uni-koeln.de)。

目录

1 引言 3

2 预备知识 12

3 $O(1)$-近似算法：分划拟阵上的 MBT 14

4 在一般拟阵上最优地寻找一个 1-证书 18

5 在分划拟阵上近似最优地寻找一个 0-证书 21

6 $O(1)$-近似算法：均匀拟阵上的 MP 23

7 结论 33

A 第 1 节缺失的材料 37

B 第 2.1 节缺失的材料 39

C 第 6 节缺失的材料 41

1 引言

随机布尔函数求值（SBFE）问题是随机环境中的一类信息获取问题。本文聚焦于其中基础的单位成本版本。给定一个（紧凑表示的）函数 $f\colon \{0,1\}^n \to \{0,1\}$ 以及布尔随机变量 $x_1, \ldots, x_n$，其中对所有 $i \in [n]$，变量 x_i 以已知概率 $p_i \in (0, 1)$ 独立地取值为 1（我们也称元素 i 为激活）。必须通过依次测试变量的取值来求值函数 $f(x_1, \ldots, x_n)$。换言之，必须继续测试，直到找到函数取值的证书。

问题在于（自适应地）确定测试顺序，从而使执行测试次数的期望值最小化。此类问题的算法无需输出显式的测试策略（决策树）；只需在任何情况下输出下一项要执行的测试即可。有关该领域近期成果的综述，参见 [43]。

近期关于 SBFE 问题的论文关注的是（通常为多项式时间）近似算法 [12, 19, 17, 37, 28, 35]：如果某算法执行的测试次数的期望不超过最优策略执行的测试次数期望的 c 倍，则称其为 c-近似算法。值得注意的是，对于其中若干问题，目前尚不清楚该问题是否为 NP难，以及是否存在常数因子近似算法。设计 SBFE 问题近似算法的一般难点在于，缺乏能够自然地为最优测试策略成本提供有力下界的方法。

SBFE 文献中的一个主要方向聚焦于“计数函数”：活跃变量会使计数器递增，而函数值仅取决于计数器的值：k-of-n 函数，当且仅当至少有 k 个活跃元素时输出 1 [26, 39, 38, 5, 35]；线性阈值函数，即 k-of-n 函数的加权对应形式 [10, 12, 30]；一般的对称函数 [20, 22, 17, 37, 32]；以及超越布尔情形和单个计数器的投票函数 [28]。Ghuge等人研究了一个名为 Explainable Stochastic d-Halfspace Evaluation 的问题，该问题考虑了 SBFE 问题的一种变体：对于形如 $h(g_1, \ldots\ldots, g_d)$ 的函数，必须找到一个解释（一种特定类型的证书）；其中每个 g_i 都是线性阈值函数，而 h 是任意布尔函数 [17]。针对上述所有问题，都已开发出精确算法或近似算法。

另一个方向是研究与图相关的函数。在这里，变量对应于给定图 E 的边 $G = (V, E)$，而 f 的值取决于 G' 的子图 G，该子图由所有对应变量处于激活状态的边组成。这类问题中一个有趣的问题涉及 s-t连通性：当且仅当 G' 中存在一条 s-t 路径时，$f(x) = 1$（对于某个给定的 $s, t \in V$）。该问题是 NP难的 [16, 24]。当 G 是串并联图时，其特殊情形等价于布尔读一次公式的 SBFE 问题（参见 [21]）；该问题自20世纪70年代以来一直受到研究，但其 NP-困难性至今仍未解决 [42, 23, 12]。即使对于这一特殊情形，是否存在多项式时间的非平凡近似算法仍是一个开放问题。

在本文中，我们引入以下框架，该框架与上述两个方向都密切相关。设 $\mathcal{M} = (E, \mathcal{I})$ 为一个拟阵，其中 $E = \{1, \ldots, n\}$ 是底集，$\mathcal{I}$是独立系统。我们定义 $f^{\mathcal{M}}$ 为一个函数：当且仅当存在一个 $\mathcal{M}$ 的基，且该基仅由活跃元素（变量）组成时，其值为 1。[1] 对于这个函数对应的 SBFE 问题，我们的目标是最小化期望 查询次数（等价地，假设测试成本为单位成本），我们将其称为拟阵基测试问题（MBT）。

当 $\mathcal{M}$ 是一个均匀拟阵（即，对于某个给定的 k，当且仅当 $|I| \le k$ 时，$E' \in \mathcal{I}$）与 o

[1]等价地，用拟阵术语来说，$f^{\mathcal{M}}(x) = 0$ 当且仅当存在一个 $\mathcal{M}$ 余回路，且它仅由未激活元素组成。关于余回路及其他拟阵概念的定义，请参阅例如 [36]。本文仅假设读者具备非常基础的拟阵知识。

适用于 k-of-n 函数的 SBFE 问题。当 $\mathcal{M}$ 是一个图拟阵时（即，当且仅当 $(V,\ E')$ 是给定图 $G = (V,\ E)$ 的一个无环子图时，$E' \in \mathcal{I}$），这就是上述 s-t连通性函数的全局连通性版本。

尽管我们的一些结果涉及更一般的问题，但本文主要关注 $\mathcal{M}$ 为划分拟阵时的 MBT。在这种情况下，拟阵 $\mathcal{M}$ 由$\mathcal{P} = \{P_1, \ldots, P_d\}$ 的一个划分以及上界 k_1、.....、k_d表示，使得 $E' \in \mathcal{I}$当且仅当 $|E' \cap P_i| \leq k_i$ 对所有$i \in \{1, \ldots, d\}$均成立。**我们的主要结果是一个 $O(1)$-近似算法，用于求解划分拟阵上的 MBT**。

划分拟阵上的 MBT 问题等价于布尔函数$f = g_1 \wedge g_2 \wedge \ldots \wedge g_d$的 SBFE 问题，其中 g_i 定义在互不相交的变量集合上，并且每个 g_i 都是某个 $1 \leq k_i \leq n_i$的 k_i-of-n_i函数。例如，$f(x_1, x_2, x_3, x_4, x_5) = g_1(x_1, x_2) \wedge g_2(x_3, x_4, x_5)$，其中 $g_1(x_1, x_2)$ 是 1-of-2 函数（布尔或），而 $g_2(x_3, x_4, x_5)$ 是 2-of-3 函数（多数函数）。由于 k_i-of-n_i 函数是线性阈值函数，且 g_i 定义在互不相交的变量集合上，因此可以证明，划分拟阵上的 MBT 问题是可解释随机 d-Halfspace Evaluation 的特例。Ghuge等人针对这一更一般的问题提出了一种近似算法，其近似因子为 $O(d^2 \log d)$ [17]。

MBT 在分划拟阵上的一个特殊情形是每个 $k_i = 1$的情形；等价地，f 可由只读一次 *CNF* 公式表示。与 k-of-n函数一起，由只读一次 CNF（对偶地，DNF）公式表示的函数，是少数已知可通过多项式时间算法精确求解 SBFE 问题的布尔函数类别之一 [7].[2]

虽然分划拟阵可以说是从均匀拟阵再进一步的一类结构，而且看起来很简单，但分划拟阵上的 MBT 问题带来了全新的重大挑战。对于上述 SBFE 表述，我们知道如何分别最优地求值每个 g_i，因为每个都是一个 k_i-of-n_i 函数。但由于 f 是 g_i的合取，我们需要决定何时以及如何在测试各个 g_i 中的变量之间切换。一种自然的方法是按深度优先顺序执行测试：一旦开始测试某个 g_i中的变量，就先完成对该 g_i 的求值，再测试另一个 g_i中的变量。事实上，对于只读一次 CNF 函数这一特殊情形，深度优先方法是最优的。然而，当 g_i 是任意的 k_i-of-n_i函数时，深度优先方法的近似比无法由 d（ g_i的数量）的任何函数所界定。另一方面，Ghuge 等人采用轮询方法，为可解释随机 d-半空间求值得到$O(d^2 \log d)$近似，但不可避免地损失了 d这一因子。**那么，应该如何决定何时从测试一个 g_i 中的变量切换到测试另一个 g_i呢？** 我们通过将这一问题归约为一种新型最大化问题来回答，且该问题带有对期望成本的约束。我们将这类问题称为“期望预算问题”。

我们所研究的特定期望预算问题称为 MP0，具体如下：给定一个单一的g_i、概率 p_i 和预算 B，找出一种策略，使获得满足 $f(x) = 0$ 的证书的概率最大，同时约束策略的期望成本至多为B。**这里，“期望”一词至关重要，它会导致可达概率相对于成本的硬性逐次实现约束出现任意大的差距**。对于 $B < 1$，这一点显而易见；但更重要的是，即便预算任意大，这一点也依然成立。从算法角度看，在期望成本约束下，难点在于判断测试结果何时足够乐观，值得继续进行测试，以及何时应当直接中止。有趣的是，对于 MP0，按照 k-of-n 函数的 SBFE 问题的最优策略所采用的相同（自适应）顺序进行测试，并不能得到常数因子近似，即使中止该策略

[2]根据布尔对偶性，分划拟阵上的 MBT 问题实际上等价于上述针对 g_i 析取（而非合取）的 SBFE 问题。同样，读一次 DNF 公式的 SBFE 问题实际上等价于 CNF 公式的 SBFE 问题。

以最优方 p式 y. 本文展示了如何为 MP 实现常数因子近似 0.

在文献中，还有若干问题需要自适应地在 d不同选项之间进行选择，例如马尔可夫链。一种常常能够导向最优性的深刻解法概念，基于著名的吉廷斯指数及相关指数 [18, 44, 13, 41, 25]。我们采用的方法是化归到上述期望预算问题，这与众不同，并有望应用于其他 SBFE 问题以及其他随机组合优化问题。虽然我们只能得到一个 O（1）近似解，但我们所解决的问题面临着一项重大挑战，而这在吉廷斯指数通常处理的问题中并不存在：我们不仅要在 d 不同选项 g_i之间进行选择，还要选择要执行哪项测试，以便进一步评估该选项 g_i。换句话说，在每一步中，我们选择的不只是要推进哪条马尔可夫链，还要选择一个马尔可夫决策过程及该过程中的一个动作；这项任务通常要困难得多（例如，参见 [15, 40]）。

我们总结一下我们的贡献。我们的主要成果包括：针对划分拟阵上的 MBT，给出一个 O(1)-近似算法；以及针对均匀拟阵上的期望预算问题，给出一个 O(1)-近似算法（事实上，该算法也适用于划分拟阵）。从更具概念性的层面来看，我们定义了 MBT——一类新的 SBFE 问题——以及期望预算问题。我们相信，后者的贡献尤其为未来研究开辟了引人入胜的新方向。

1.1 技术概述

在进一步讨论相关工作之前，我们先非正式地概述一下如何获得 O针对划分拟阵上的 MBT 问题的 (1)-近似算法。在图 1中，我们给出了一幅示意图，展示本文主要成果之间的关系。希望跳过这一概述、直接阅读详细结果的读者可以放心这样做，不会影响行文的连贯性。

关于均匀拟阵的已知结果。我们首先回顾一下均匀拟阵上 MBT 的精妙最优策略 [26] （另请参见例如 [6, 5]）。直观地说，按照概率递减的顺序进行测试，可以最大化在任意步数（测试）之后认证 $f^{\mathcal{M}}(x) = 1$ （下文中的 1-证书），即找到 k 个活跃元素的概率。同样地，按照相反的顺序进行测试，可以最大化在任意步数之后认证 $f^{\mathcal{M}}(x) = 0$（0-证书）的概率，即找到 $n-k+1$ 个非活跃元素。关键观察是，第一个策略至少必须执行其测试中的前 k 个测试后才能终止，而第二个策略至少必须执行其测试中的前 $n-k+1$ 个测试后才能终止。因此，在终止之前，两者都需要执行概率排名第 k大的测试。于是，存在一种策略，能够在任意步数之后最大化找到证书（即 0-证书或 1-证书）的概率，并且首先执行概率排名第 k大的测试。通过执行该测试，然后递归处理，即可计算出这一策略。该策略还能够最小化找到证书之前的期望查询次数。

事实上，由于本文关注近似算法，我们也可以考虑以下针对均匀拟阵 MBT 的简单近似算法：以轮转方式交替执行按概率递增顺序进行的测试和按概率递减顺序进行的测试。对于任何 $\ell > 0$，经过 2ℓ 步后，概率最高的 ℓ 个元素和概率最低的 ℓ 个元素都已完成测试。因此，该策略在 ℓ 步后找到证书的概率不低于最优策略，并且该策略是一个 2-近似。

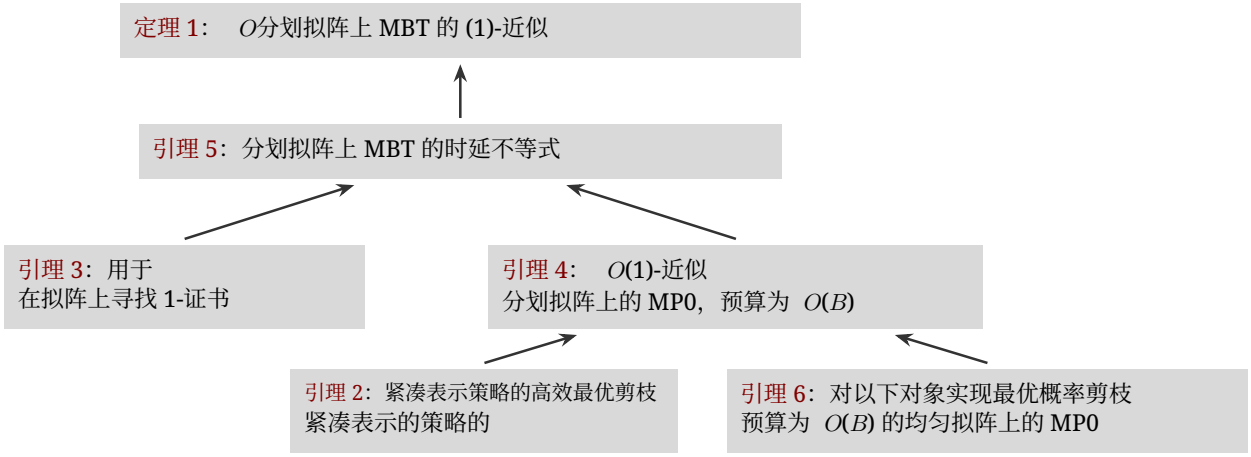

图 1：结果概览及其相互关系。

单独讨论值为1 的情形。 上述部分结果可以推广到任意拟阵 $\mathcal{M}$：考虑按概率递减顺序进行测试的策略，但作出如下修改：跳过任何不会对构成一个基有所贡献的元素，即跳过任何会与已发现为激活的元素构成相关集的元素。我们将此策略称为 OPT_1。通过对均匀拟阵证明的直接推广，我们在引理 3(第 4 节) 中证明，OPT_1 使经过任意步数后找到 1-证书的概率最大。事实上，接下来我们将着重于寻找 0-证书，并最终将处理这一任务的策略与 OPT_1, 结合起来，仍采用一种轮转形式。

关于值为 0情形的一些直觉。显然，在划分拟阵上，寻找 0-证书与寻找 1-证书截然不同——只需证明某个划分类 i 中的活跃元素少于 k_i 个。因此，人们可能会想采用一种“深度优先”策略：根据某种标准选择一个划分类 P_i，并使用均匀拟阵的最优评估策略来确定其中是否有 k_i 个活跃元素 P_i。如果不是这种情况（即已找到一个 0-证书），就停止测试；否则继续处理另一个 P_i。

我们给出一个运行示例，其中 $f(x) = 0$ 以极高的概率发生（因此可以忽略 $f(x) = 1$ 的情况），在该示例中，没有任何深度优先策略是常数因子近似。在我们的实例族中[3]，$d = 1/\varepsilon^2$，并且对于每个 $i \in [d]$ 划分类 i，都包含以下 n_i 个变量：

- $1/\varepsilon$ 个变量 x_i^1，……，$x_i^{1/\varepsilon}$，其取值为 1 的概率为 0，
- 个变量 y_i，其取值为 1 的概率为 $1-\varepsilon$，以及
- $1/\varepsilon^3$ 个变量 z_i^1，……，z_i^{1/ε^3}，其取值为 1 的概率为 $1-\varepsilon^2$，

对于某个较小的 $\varepsilon > 0$（见 图 2），并且我们还令 $k_i = 1/\varepsilon^3 + 1 = n_i - 1/\varepsilon$。注意，任意划分类都以趋近于 1 的概率拥有一个 0-证书（当 $\varepsilon \to 0$ 时）。直观来说，难点在于找出能够以较低成本获得该证书的划分类。我们考察最优策略和一种深度优先策略：

- 最优策略会在不同的划分类中测试 y_i，直到找到一个划分类 i，其中 y_i 的值为 0（这种情况以概率 $1-(1-\varepsilon)^{1/\varepsilon^2}$ 发生，该概率足够大，因此可以忽略另一种情况）；期望成本低于 $1/\varepsilon$，然后测试该划分类中的所有 x_i^j

[3]为便于展示，我们在一些反例中允许概率取 0 和 1。即使变量的取值已知，若要将其纳入证书，仍必须对其进行测试。不难看出，这类概率可以分别用非常接近 0 和 1 的概率来模拟。此外，为便于展示，我们对变量采用了不同于 $x_1, \ldots, x_n$ 的命名方式。

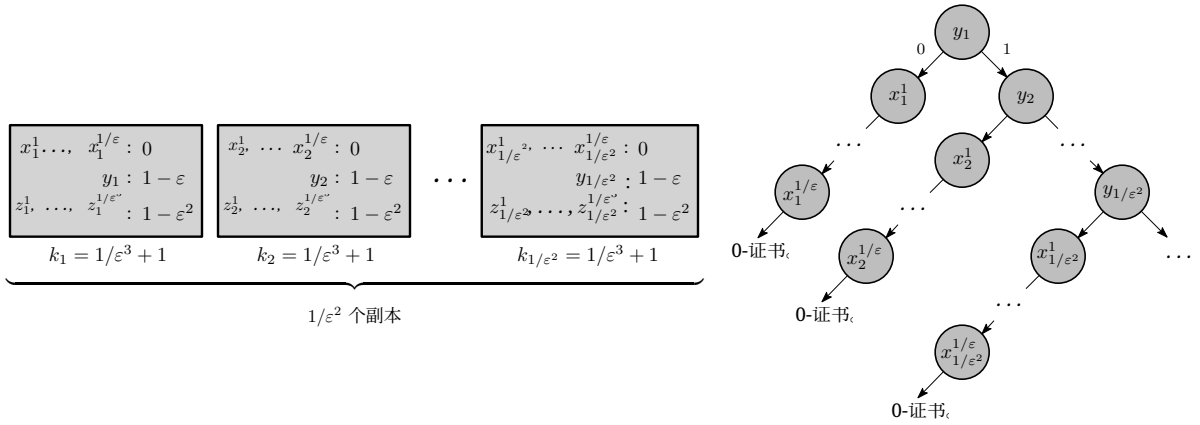

图 2：左：深度优先策略和非自适应策略表现不佳的实例（记法：“变量名：概率”）。右：该实例的最优策略。

类。期望总成本为 $O(1/\varepsilon)$（因为 $\varepsilon \to 0$）。

- 相比之下，考虑一种深度优先策略。在它考察的第一个划分类 i 中，y_i 取值为 1 的概率为 $1-\varepsilon$；此时，计入 x_i^j 中的 0 后，还缺少一个 0 才能构成 0-证书。深度优先策略很可能（概率为 $1-(1-\varepsilon^2)^{1/\varepsilon^3}$）在 z_i^j 中找到另一个 0，但期望而言，它必须测试其中大约 $1/\varepsilon^2$ 个元素才能找到这样的 0。因此，深度优先策略的期望总成本为 $\Omega(1/\varepsilon^2)$。

注意，同一个实例表明，非自适应策略（以固定顺序进行测试、不受结果影响的策略）并非常数因子近似。事实上，非自适应策略无法将对某个划分类 i 中 x_i^j 的投入与 y_i取值为 0 联系起来。因此，在任何划分类中，它找到 0-证书的速率都只能达到 $O(\varepsilon^2)$（每找到一个证书所需的成本），从而总期望成本仍为 $\Omega(1/\varepsilon^2)$。

在第 1.2 节中，我们将讨论随机布尔函数评估文献中的其他相关方法，并说明这些方法无法用于实现分划拟阵上 MBT 的常数因子近似。

我们方法的概述。 如前所述，我们的新方法的核心是近似求解以下新问题，而该方法确实能够实现常数因子近似。给定预算B、布尔函数 f和概率 p_i，求一个测试策略，使其使用的预算（成本）不超过预算B （期望意义下），并最大化找到 f 的 0-证书的概率。

关键在于，该策略可以在尚未找到证书时停止测试，并且可以使用内部随机化。我们将此问题称为最大概率 0-证书问题（MP0）。具体而言，我们将展示如何近似求解函数 $f^{\mathcal{M}}$，其中 $\mathcal{M}$ 是均匀拟阵，这对应于其中一个划分类。与此同时，我们还将近似求解 $f^{\mathcal{M}}$，其中 $\mathcal{M}$ 是分划拟阵。事实上，我们认为，必须满足期望意义下的预算约束本身就是自然且有趣的问题。令人惊讶的是，据我们所知，随机组合优化领域的其他工作都没有研究具有这一特点的问题。

为了直观地理解，为什么近似求解均匀拟阵上的 MP0 有助于近似求解分划拟阵上的 MBT，不妨考虑反例中某个划分类 i 的 MP0；该反例展示了深度优先策略，见 图 2。给定预算 $B=2$，这里 MP0 的最优解如下：测试 y_i；如果其值为 0，则继续测试所有 x_i^j 变量，否则停止。所得策略的期望成本至多为 $B=2$，且找到 0-

证书的概率为 ε。请注意，这一策略恰好就是我们对整个反例所描述的最优策略中，用于在划分类 i 内执行测试的子策略，并在该子策略于划分类 i 中停止测试时将其剪枝后所得的策略。因此，我们可以期望，通过为不同划分类的 MP0 组合（近似）最优解，并分配适当的预算，能够得到分划拟阵上 MBT 的（近似）最优解。

我们针对分划拟阵上的 MBT 寻找 0-证书的策略分阶段进行，从阶段 0 开始，各阶段的长度按指数递增。证明近似保证的关键在于采用基于延迟的论证。更确切地说，我们证明一个递推关系，将以下三个量联系起来：

(i) 当前阶段结束后尚未找到证书的概率，(ii) 前一阶段结束后尚未找到证书的概率，以及 (iii) 最优策略在一个可比（略短）的时间段内尚未找到证书的概率。

此类递推关系此前已有应用 [29, 14, 17], 但与以往工作的关键区别在于，我们各 个 阶段的成本可能因变量的取值（以及算法的内部随机化）而有很大差异；其成本只有一个上界（即所述指数值），且该上界是按期望计算的。（请注意，在我们对最优策略进行分析时所考察的时间段确实具有确定的长度。）

为了得到这一递推关系，事实证明，只需近似求解分划拟阵上的 MP0（其中预算是当前阶段期望成本的上界）即可。我们在主要结果 定理 1 (第3节) 的证明中表明，这已足够。分划拟阵上的 MP0 的最优策略可能相当复杂；我们不排除它会根据观察结果自适应地在各划分类之间任意频繁地来回切换。然而，正如我们在 引理 4 (第5节) 中所证明的，对于近似求解，只需将该阶段的预算分配给不同的划分类（以近似最优的方式），然后以任意顺序运行相应 MP0 实例（在均匀拟阵上）的（近似最优）解，而无需来回切换。在分配预算时，我们力求最大化分划拟阵所有划分类各自找到 0-证书的概率之和。可以很容易地证明，该和与为分划拟阵本身找到 0-证书的概率相比，至少达到 $1 - 1/e$ 倍。

为了最大化找到 0-证书的概率之和，我们考虑由此前所有阶段诱导的子实例。我们对均匀拟阵上的 MP0 进行近似求解，并同时处理所有预算。我们在第6节 (结合第 2.1 节)中证明，可以高效地计算（一个紧凑表示），得到一个分段线性、单调且凹的函数 $h : \mathbb{R}_+ \to [0, 1]$使得，对于任意预算 B， $h(B)$ 至少等于使用预算 $c \cdot B$ 时，在某个常数 c下能够达到的概率。 图 3 展示了在前述示例（见图 2）中该函数的形状，其中 $c = 1$。我们还展示了如何针对所有预算 B，计算出一种策略，使其达到 $h(B)$的概率，且期望成本为 B。由此可见，近似最大化概率之和的问题可以视为一个连续背包问题，而按密度排序的贪心算法是该问题的最优算法。

求解均匀拟阵上的 MP0。剩下需要近似求解的问题是均匀拟阵上的 MP0。[4]

请注意，当预算等于同一均匀拟阵上对应 MBT 实例的最小期望求值成本时，本小节开头讨论的均匀拟阵上 MBT 的评估策略是最优的。它找到 0-证书的概率为

[4]对于均匀拟阵，寻找 1-证书对应的问题实际上是等价的问题，因为 k-of-n 函数的补函数是一个 $(n - k + 1)$-of-n 函数。

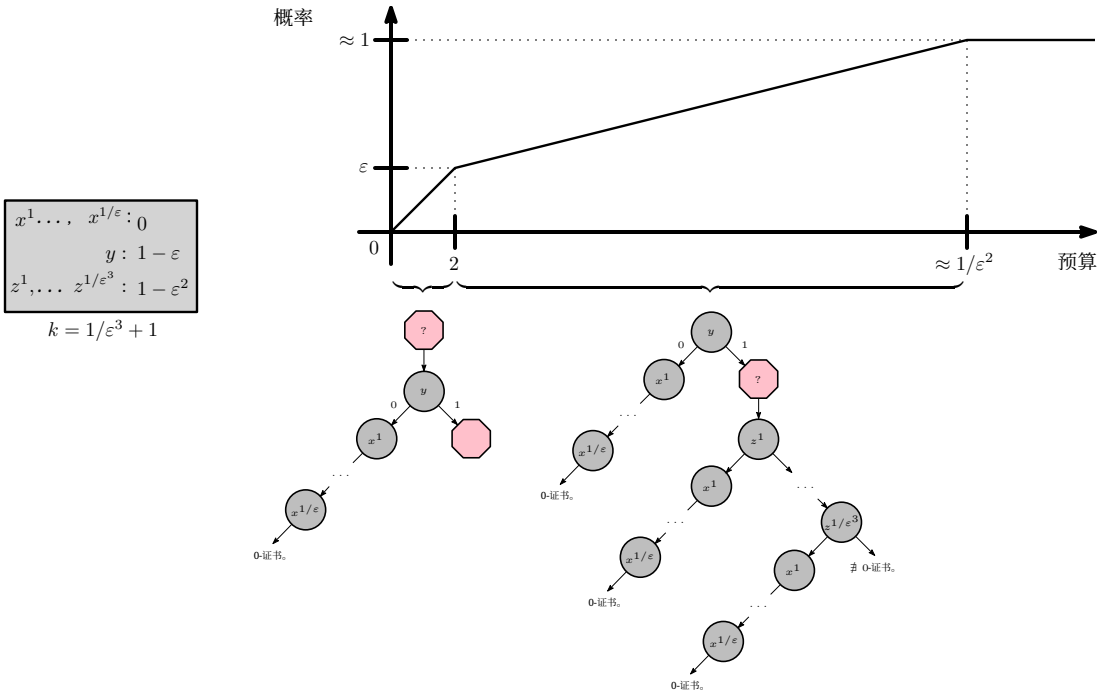

图 3：左：一个均匀拟阵（来自深度优先策略和非自适应策略反例的一个划分类）。右：一个（未按比例绘制的）图，展示了该均匀拟阵中预算与找到 0-证书的可达概率之间的函数关系，以及相应的策略。红色八边形表示停止；当图中出现“? ”时，仅表示以某个概率停止。左侧的（随机化）策略适用于预算介于 0 和 2 之间的情况，右侧的策略适用于预算介于 2 和约 $\frac{1}{\varepsilon^2}$ 之间的情况。

—

等于 $f^{\mathcal{M}}(x) = 0$。请注意，在图 3所示的例子中，对同一策略进行剪枝（即提前停止，也可以采用随机化方式）同样能得到最优策略。因此，人们或许可以大胆猜想，这一结论普遍成立；或者，至少总能通过对最优评估策略进行剪枝，获得常数因子近似。为说明这一猜想不成立，考虑均匀拟阵上的一个 MP0 实例，其

- 一个取值为 1 的概率为 $1-$ 的单个变量 ε and
- $1/\varepsilon$ 取值为 1 的概率为 $1-\varepsilon^2$ 的变量，

其中 $n = 1 + 1/\varepsilon$，且 $k = n - 1$。对最优评估策略进行任何剪枝，都必须先测试概率为 $(1-\varepsilon^2)$ 的变量；只有在找到一个 0（或测试完所有概率为 $(1-\varepsilon)$ 的变量）后，才能转而测试概率为 $(1-\varepsilon^2)$ 的变量。预算为 2 时，找到 0- 证书的概率至多为 $O(\varepsilon^3)$。相比之下，预算为 2 时，MP0 的最优策略会先测试概率为 $(1-\varepsilon)$ 的变量，查看其结果是否为 0；只有在该事件发生时，才会投入资源测试概率为 $(1-\varepsilon^2)$ 的变量，从而以 $\Omega(\varepsilon^2)$ 的概率找到 0-证书。我们在 附录 A 中给出严格论证。这个例子表明，MP0 与评估问题有很大不同。

我们仍可期待存在另一种这样的“普适”策略——一种经过剪枝后，针对任意预算都能达到常数因子近似的策略。事实上，只要证明某种策略具有这样的结构性结果（第 2.1 节），我们就能直接得到一个算法：正如我们在引理 2 （第 2.1 节）中所证明的，将树剪枝至某个预算的最优剪枝 B 可以通过反复剪去密度最低的子树来计算。密度是两个量的比值，二者均以到达该子树根节点为条件：（1）在 $\mathcal{M}$中找到 0-证书的概率；（2）

该子树中产生的期望成本。这种贪心算法以连续的方式进行剪枝（每次选中的子树被剪枝的概率从 0 连续增加至 1），并在预算约束恰好满足时停止。方便的是，只需对该算法稍作修改，就能计算出一个分段线性函数，将任意预算映射到剪枝至该预算时所达到的目标值，其断点对应于贪心算法切换子树的位置。当且仅当原策略具有上述普适性时，该函数才满足前述近似保证。

一个小小的复杂之处在于，直接实现贪心算法需要完整表示策略。幸运的是，尽管将策略表示为一棵树需要指数级空间，但可以合并已发现相同数量的 0 和相同数量的 1 的节点和，从而得到一个规模为多项式的有向无环图（DAG）。这个 DAG 可以直接计算出来（无需显式合并），而且可以轻松地调整贪心算法，使其在 DAG 上运行，从而得到一个多项式时间算法。

但这样的通用策略可能是什么呢？另一个自然的候选方案是按概率从低到高进行测试——毕竟，我们要找的是一个 0-证书。然而，考虑这样一个实例：有 $k = 1$ 个变量，其取值为 1 的概率为 0，另有一个变量，其取值为 1 的概率为 $1 - \varepsilon$。给定预算 2，首先测试取值为 1 的概率为 $1-\varepsilon$ 的变量（只有当测试结果为 0 时才继续），就能以概率 ε 找到一个 0-证书。另一方面，如果先测试取值为 1 的概率为 0 的变量，那么要测试取值为 1 的概率为 $1 - \varepsilon$ 的变量并找到 0-证书，就必须先完成所有其他测试。完成所有测试的成本为 $1/\varepsilon + 1$。因此，在预算为 2 的情况下，最好的做法是以概率 $2/(1/\varepsilon+1)$ 执行所有测试，否则完全不进行测试。最终找到 0-证书的概率为 $O(\varepsilon^2)$。

按概率递减顺序测试变量，显然也不符合上述意义上的普适性；例如，任取一个实例，再添加大量概率非常接近 1 的变量——测试这些变量并不能充分推进寻找 0-证书的进程。（注意，我们为最优评估策略给出的例子也是一个反例，只是更加复杂。）

鉴于之前的反例，我们希望设计一种策略：(i)“了解”找到 0-证书是否会代价高昂，以便在这种情况下可以对其进行剪枝；同时 (ii) 能够足够快速地推进寻找 0-证书的进程。我们称这种策略为 InsideOut。它是以下两个序列的轮转：

- 按递减顺序排列的、满足 $p_i \le 1/2$ 的变量 i 序列，以及
- 按递增顺序排列的、满足 $p_i > 1/2$ 的变量 i 序列。

直观而言，这兼顾了两个目标：它满足目标 (i)，因为高熵测试相对较早地出现在序列中。它也满足目标 (ii)，因为如果存在以较高概率处于激活状态的测试，我们就会在序列中较早地纳入一些（可能不同的）以相近概率处于激活状态的测试（即，其概率至少为 $1/2$）。

在第6节中，我们证明了 InsideOut 是上述意义下的通用策略。为此，我们修改 MP0 的任意一个解，得到 InsideOut 的一种剪枝方式，使期望成本至多增加一个常数因子，同时不降低找到 0-证书的概率。我们的证明对序列长度使用归纳法，并运用了细致的情形分析和收费论证。这是本文技术性最强的部分。我们在第6节开头对证明作了更详细的概述。

1.2 相关工作的进一步讨论

替代方法。 对于分划拟阵上的 MBT 问题，可以采用一些已有方法，但这些方法无法得到常数因子近似界。若干研究

针对 SBFE 问题，将分布划分为若干个 ℓ 部分，并针对每个部分设计算法，使其在给定相应部分的条件下，成本近似最小。通过在这些算法之间轮转，可以得到一个适用于整个问题的算法，其近似保证可由 ℓ和最大的单个近似保证 [3, 22, 17, 37]界定。对于我们的问题，一种直接的方法是在 OPT_1 与 d 分别为每个划分类寻找 0-证书的策略之间轮转，但这种方法不可避免地会使近似保证损失一个与 d 呈线性关系的因子。

如上所述，划分拟阵上的 MBT 问题是 Ghuge 等人研究的可解释随机 d 半空间评估问题的特例 [17], , 该问题涉及对函数 $h(g_1, \ldots\ldots, g_d)$ 求值，其中 g_i 是线性阈值函数。值得注意的是，他们的 g_i 可以定义在相互重叠的变量集合上。他们的算法与我们的算法一样，分阶段执行，并采用指数递增预算。然而，在每个阶段 j，他们为每个 g_i 分配相同的测试预算 B_j（其中预算 B_j 是对最大成本的硬约束）。在该阶段内，他们依次遍历 g_i，先对一个 g_i 执行测试，然后再处理下一个。这种迭代方式导致其近似保证依赖于 d 。

Deshpande 等人提出了一种截然不同的方法来构造 SBFE 问题的近似算法。该方法是将函数 f 的 SBFE 问题归约为一个针对子模效用函数 g的随机子模覆盖问题，该子模效用函数由 f构造而成。随后，使用贪心算法求解针对 g的随机子模覆盖问题，从而得到原始 SBFE 问题的一个近似因子；该因子取决于所构造的 g的某些性质。对于某些类型的函数 f，如果仔细构造效用函数 g ，则可以得到较小的近似因子。然而，我们在 A.2节 中证明，当 f 是定义在互不相交变量集上的 k_i-n_i 个函数的合取（或析取）时，不可能构造 g ，使得所得近似因子为 $o(n^\beta)$，其中任意常数 $0 < \beta < 1$。

任意成本测试。上述若干 SBFE 问题也在更一般的设定下得到了研究：每个变量 x_i 都具有一个非负测试成本 c_i，目标是最小化总期望测试成本，例如 [20, 17, 28]。任意成本情形下，均匀拟阵上的 MBT 最优策略（即 k-of-n 函数）与单位成本情形下的最优策略 [38, 5]密切相关。对于分划拟阵上的 MBT 问题，我们将一个开放问题留待研究：我们的主要结果，即存在多项式时间常数因子近似算法，是否也适用于任意成本。值得注意的是，将分划拟阵上的 MBT 问题归约为均匀拟阵上的 MP0 的过程在任意成本下仍然成立，只需对 OPT_1 进行简单修改，使用概率—成本比即可。然而，我们算法中最困难的部分是求解 MP0，而这个问题在任意成本下似乎要困难得多。

其他基准。 我们指出，我们的基准并非（期望）证书大小（例如，见 [8, 34, 31, 3]），也不是等价的全知“先知”的表现 [33, 9]。事实上，对于 MBT 问题，很容易看出，即便是均匀拟阵，也不可能在这一基准下实现常数因子近似（竞争比）。我们不采用这种“在线算法视角”，而是采用“近似算法视角”，并以最优策略作为基准。大多数关于 SBFE 问题的研究都采用这一基准；在此基准下，我们实际上可以实现常数因子近似比。

2 预备知识

本节定义与布尔函数、策略和剪枝相关的术语。我们不重复第1节中已有的正式定义。我们在全文中用 $[n]$ 表示 $\{1,\ldots,n\}$。

设 $f\colon\ \{0,1\}^n \to \{0,1\}$ 是一个布尔函数。我们将 f 视为关于 n 个输入变量x_1、.....、 x_n的函数，即 $f(x_1, ..., x_n)$。赋值 π（针对这些变量）是 $\{0,1\}^n$的一个元素。于是，π_i是对输入变量 x_i 的赋值。部分赋值 σ是 $\{0,1,*\}^n$的一个元素，其中$\sigma_i = *$ 表示 σ 没有为 x_i赋值。由于我们将 x_i 视为随机变量，因此常将对 x_i 的赋值称为（ x_i的）实现。输入变量上的部分赋值 σ 的扩展是这些变量上的一个部分赋值 σ'，满足对所有 $\sigma_i \neq *$都有 $\sigma'_i = \sigma_i$。部分赋值 σ 会诱导出一个从 f到 f 的函数 f_σ：取所有满足 $\sigma_i \neq *$的输入x_i，并将每个此类 x_i 的值固定为 σ_i，于是函数值只取决于那些未被 σ赋值的 x_i。例如，如果 $f(x_1,x_2,x_3,x_4) = (x_1 \vee x_2) \wedge (x_3 \vee x_4)$，且$\sigma = (1,*,0,*)$，那么 $f_\sigma(x_2,x_4) = f(1,x_2,0,x_4) = x_4$。对于 $\ell \in \{0,1\}$，f的一个 ℓ-证书是一个部分赋值 $\sigma \in \{0,1,*\}^n$，使得对 σ的所有扩展 $\sigma' \in \{0,1\}^n$，$f(\sigma') = \ell$成立。f 的证书是 f的 1-证书或 0-证书。

我们通过二叉决策树定义一种（确定性）测试策略 S，用于 f，二叉决策树是该策略的一种可能表示形式。决策树的每个内部节点都标记有一个变量 x_i，对应于执行一次“测试”或“查询”以确定 x_i的值。我们将执行此测试称为测试 x_i。叶节点对应于“停止”或“终止”。每个内部节点的一个子节点对应于 $x_i = 1$，另一个对应于 $x_i = 0$。在每条根叶路径上，变量 x_i 最多只能出现一次。给定一个赋值 $\pi \in \{0,1\}^n$，该策略会针对 π诱导出一系列要执行的测试，即根叶路径上各节点中的 x_i；这一路径是通过对每次测试获得结果 π_i 来确定的x_i。该根叶路径对应于一个部分赋值，其中对于路径上测试的所有 x_i，$x_i = \pi_i$，而对于其他变量则为 $x_i = *$。如果对于所有 $\pi \in \{0,1\}^n$，该部分赋值都是 f 的证书，并且路径末端的叶节点标记有 f（π）的值，那么我们称该策略计算或评估 f。

一个针对 f 的随机化测试策略 S 可以用一棵二叉决策树表示，该树包含两种类型的内部节点：标记为 x_i 的节点（其中 $i \in [n]$），对应于对 x_i 执行测试；以及标记有某个概率 α_v 的节点 v（其中 $0 < \alpha_v < 1$）。后一类节点对应于以概率 α 正面朝上掷一枚硬币，其中该节点的一个子节点对应正面朝上，另一个对应反面朝上。不过，从根节点到叶节点的每条路径上，每个变量最多只能出现一次。等价地，我们可以将随机化测试策略定义为确定性测试策略上的一个分布。除非另有说明，我们使用测试策略（或简称策略）一词来指确定性测试策略。

因此，我们所考虑的随机变量有两个随机性来源：赋值 $\pi \in \{0,1\}^n$ 上的概率分布，以及策略的内部随机化。在计算概率和求期望时，我们会同时考虑这两个随机性来源。对于一个（可能是随机化的）策略 S，令 cost(S) 为如下随机变量：在随机赋值 π 上运行 S 所产生的总成本。

如果布尔函数 $f = f(x_1,\ .....,\ x_n)$ 的函数值只取决于$\sum_{i=1}^n x_i$，则称其为对称函数。考虑针对这类函数的一种可能带有随机化的测试策略 S。如果其决策只取决于测试结果中目前观察到的 0 和 1 的数量（而不取决于哪些测试产生了这些 0 和 1），以及可能的内部随机化，则称 S 为初等策略。策略 S 可以等价地表示为一个有向无环图（DAG）：将观察到相同数量的 0 和相同数量的 1 的节点合并即可。与二叉决策树表示不同，二叉决策树的大小通常无法用 n的多项式来界定，而该 DAG 的大小是 n的多项式。我们称之为决策DAG

（表示），用于表示初等策略 S。对于具有固定表示形式（即二叉决策树，或初等策略情况下的DAG）的策略 S，我们用 $V(S)$ 表示该表示中的节点集合。

对（树或一般有向无环图）表示的可能随机化测试策略的节点进行剪枝，意味着在该节点处终止。也就是说，在树表示的情况下，整个子树都会被移除，并替换为一个叶节点。对节点进行随机剪枝（概率为α）意味着在该节点引入一次随机选择：以概率 $1-\alpha$继续（按照原方式），以概率 α立即终止。（随机化）测试策略 S' 是一个（随机化）剪枝后的（可能随机化）测试策略 S，如果存在一个序列 $S = S_1, S_2, \ldots\ldots, S_k = S'$，其中某个 $k \geq 1$ 满足：对于所有 $i \in [k-1]$，S_{i+1} 都是通过对 S_i的表示中的某个节点进行（可能随机化）剪枝，由 S_i 产生的。与随机化测试策略的定义一样，策略 S 的随机剪枝也可以定义为（确定性）剪枝 S上的一个分布。

2.1 计算最优剪枝

以下定义和陈述针对对称函数 $f\colon \{0,1\}^n \to \{0,1\}$（且仅在 f 是 k-of-n 函数时使用）。给定某个 $L \subseteq \{0,1\}$，令 $U\colon \{0,1,*\}^n \to \{0,1\}$ 为一个效用函数，它为每个属于某个 $\ell \in L$ 的 ℓ-证书的部分赋值赋予效用值 1，否则赋予效用值 0。对于 f 的一个（可能是随机化的）测试策略 S，我们将 $U(S)$ 定义为 $U(\sigma_S)$，其中 σ_S 是应用策略 S 后产生的随机部分赋值。因此，$U(S)$ 是一个随机变量。我们称 U 为（针对 f 的）证书效用函数。

对于一个（可能是随机化的）策略 S，令 randPrunings(S) 表示 S 的所有随机剪枝组成的集合。令 $q_S\colon \mathbb{R}_{\geq 0} \to \mathbb{R}$，其中

$$q_S(B) := \max\{\mathbb{E}\big[U(S')\big] \mid S' \in \text{randPrunings}(S) \wedge \mathbb{E}[\text{cost}(S')] \leq B\}.$$

也就是说，q_S 是对 S 进行任意随机剪枝且期望成本至多为 B时所能达到的最大期望效用。

为了计算这个函数，给定一个（可能是初等的）策略 S 以及相应决策树或有向无环图中的一个节点 v，定义随机变量 $U(S, v)$如下：如果 S 访问了 v，且在对 v进行测试之前尚未找到 ℓ-证书，则其值为 $U(S)$；否则，其值为 0。类似地，定义随机变量 cost(S, v)：如果 S 访问了 v，则其值为执行 v 处的测试以及 S 之后执行的所有测试所需的成本；如果 S 未访问 v，则其值为 0。显然，对于根节点 r，有 $U(S, r) = U(S)$，且 cost(S, r) = cost(S)。

我们首先给出一个引理，说明对于初等策略，剪枝相应的决策树还是相应的决策DAG并无区别。这是一个非常直观的结论，因为从树得到DAG只需合并等价节点；不过，为完整起见，我们在B 节给出证明。该引理是获得运行时间为输入规模的多项式的算法所必需的。

引理 1. </style}令 为对称函数的初等测试策略。令 为 的（随机化）剪枝。那么存在 的一个（随机化）剪枝，它是初等的，并且满足 成本(R)] = $\mathbb{E}$[成本(R')]，且 $\mathbb{E}[U(R)] = \mathbb{E}[U(R')]$。

给定策略 S（以树或有向无环图表示），过程 ComputePruningFunction(S)计算函数 q_S的表示。正如我们将在下文证明的，q_S 是一个分段线性函数，因此可由其断点表示。该过程通过维护一个策略 S'（初始为 S），并反复选择使比值 $\mathbb{E}[U(S', v) \mid S'$ 达到 $v]$ 与 $\mathbb{E}[$成本(S', v) $\mid S'$ 达到 $v]$之比最小的节点 v，来计算这些断点。对于每个这样的 v，它通过使其停止来修改 S'

算法 1： ComputePruningFunction(S)

$S' \leftarrow S$;

BP $\leftarrow \{(\infty,\ \mathbb{E}[U(S)]),\ (\mathbb{E}[\text{成本}(S)],\ \mathbb{E}[U(S)])\}$ (断点集合);

当 $\mathbb{E}[\text{成本}(S')] > 0$ **时**

  选择 $v^* \in \arg\min_{v \in V(S')} \frac{\mathbb{E}[U\ S'\ v\ |S'\quad\quad v]}{\mathbb{E}[\quad S'\ v\ |S'\quad\quad v](,\ )\text{达到成本}(,\ )\text{达到}\ v]}$;

  通过在 v^* 处剪枝来修改 S'；

  BP $\leftarrow$ BP $\cup \{(\mathbb{E}[\text{成本}(S')],\ \mathbb{E}[U(S')])\}$;

返回 BP

算法 2： ComputePrunedStrategy(S, B)

$S' \leftarrow S$;

当 $\mathbb{E}[\text{成本}(S')] > B$ **时**

  选择 $v^* \in$ 使.....最小的参数$_{v \in V(S')} \frac{\mathbb{E}[U\ S'\ v\ |S'\quad\quad v]}{\mathbb{E}[\quad S'\ v\ |S'\quad\quad v](,)\text{达到成本}(,)\text{达到}\ v]}$;

  $p \leftarrow$ 最小值$_{\text{本}(,)]}\left\{1, \frac{\mathbb{E}[\text{成本}(\quad S'\ ]-B\ )\text{成}}{\mathbb{E}[\quad S'\ v^*}\right\}$;

  在 v^*处对 S' 进行随机剪枝（概率为 p）；

返回 S'

于 v，也就是（确定性地）对节点 v进行剪枝。以这种方式获得的策略（直到达到空策略为止）对应于 q_S的断点。更一般地， B 定义域中的每个值都 q_S 对应于这样一种策略：允许对过程中随机化选择的每个节点 v 进行剪枝即可得到该策略。过程 ComputePrunedStrategy(S, B) 返回对应于 B的策略。相关过程的伪代码分别见算法 1 和 算法 2。

我们在下述引理中总结这些过程的主要性质，证明见 B 节。我们假设输入策略 S 的输入形式为二叉决策树；或者，当 S 是初等策略时，输入形式为有向无环图。输出表示形式与输入类型相同。

引理 *2. </style}*设 S 是对称函数 f的一个测试策略，且 U 是 f的证书效用函数。那么：

(i) 函数 q_S 是分段线性的、单调且凹的。*(ii)* q_S 的断点通过 *ComputePruningFunction(S)* 计算。此外，对每个$B \in \mathbb{R}_{\geq 0}$，*ComputePrunedStrategy(S,B)* 会计算出一个策略 $S' \in$ 随机剪枝集合(S) ，其 $\mathbb{E}[U(S')] = q_S(B)$ 且 $\mathbb{E}[\text{成本}(S')] \leq B$ 。*(iii) ComputePruningFunction(S)* 和 *ComputePrunedStrategy(S,B)* 可实现为在关于 S输入表示大小的多项式时间内运行。

3 分划拟阵上 MBT 的一个 $O(1)$-近似算法

本节将描述并分析我们针对分划拟阵上的 MBT 的 O(1)-近似算法。我们自上而下地介绍该算法，并将子程序的描述和分析留待后续章节。我们将该算法称为 ALG，并证明以下定理。

算法 3： ALG

$\sigma \leftarrow (*, \ldots\ldots, *)$;
对于 $\ell = 0, 1, \ldots\ldots$ **执行**
  执行 $\text{OPT}_1(\sigma,\ 2^\ell)$;
  执行 $\text{SteepestAscent}(\sigma,\ 6 \cdot 2^\ell)$;
  更新 σ，纳入刚刚执行的测试结果;
  如果 σ 是 $v \in \{0,1\}$ 的 v-证书，**则**
    返回“$f^{\mathcal{M}}(x) = v$”

定理 1. *ALG* 是针对分划拟阵上 *MBT* 的多项式时间 $O(1)$-近似算法，也就是说，这是针对函数 $f = f^{\mathcal{M}}$ *where* $\mathcal{M}$ 为分划拟阵的 *SBFE* 问题。

ALG通过结合两种策略的元素来实现这一点：确定性策略OPT_1 和随机化策略Steepest Ascent；两者都是针对 $f^{\mathcal{M}}$ 的评估策略（在预算充足时），但目标不同：前者寻找1-证书，后者寻找0-证书。我们将在第 4 节和第5节分别描述并分析这两种策略。除底层实例外，这两种算法都接收两个输入参数：一个部分实现（已观察到的取值）以及算法预期使用的预算。在当前对ALG的分析中，我们在此陈述关于这两种策略的两个关键引理。

下面给出关于OPT_1的引理。

引理 3。 设 $\mathcal{M}$ 是一个拟阵，且 $f := f^{\mathcal{M}}$。设 σ 是 f 和 $B \in \mathbb{N}$的输入变量的一个部分赋值。*$\text{OPT}_1(\sigma,B)$*的每次实现成本至多为 B。对于任意（可能经过随机化的）测试策略 Z，若其针对函数 f_σ 的每次实现成本至多为 B，则*$\text{OPT}_1(\sigma,B)$*已找到 f_σ 的*1*-证书的概率，至少与 Z 已找到 f_σ的*1*-证书的概率相同。最后，*OPT_1* 在多项式时间内运行。

相比之下，（更复杂的）SteepestAscent 算法专为分划拟阵的情形设计。我们有如下引理。

引理 4. 设 $\mathcal{M}$ 是一个分划拟阵，且 $f := f^{\mathcal{M}}$。设 σ 是 f 和 $B \in \mathbb{R}_{\geq 0}$ 的输入变量的一个部分赋值。*SteepestAscent*(σ, B) 的期望成本至多为 B。*SteepestAscent*(σ, B) 找到 f_σ 的 *0*-证书的概率，至少是任何（可能随机化的）测试策略在期望成本为 $B/6$ 的条件下找到 f_σ 的 *0*-证书的概率的 $1 - 1/e$ 倍。最后，*SteepestAscent*(σ, B) 可在多项式时间内运行。

可以证明，由于 q_S（引理 2 的凹性，该结果意味着在分划拟阵上，MP0 的近似比为 $O(1)$。出于技术原因，我们需要这一更强的结论来证明主定理。

ALG 分阶段运行 $\ell \in \{0, 1, \ldots\}$。在阶段 ℓ中，该阶段以部分实现 σ开始，ALG 执行 $\text{OPT}_1(\sigma, 2^\ell)$，随后执行 $\text{SteepestAscent}(\sigma, 6 \cdot 2^\ell)$。然而，在执行 SteepestAscent 期间，每当下一个测试已由 OPT 执行过$_1$，，ALG 就不会再次实际执行该测试。（事实上，我们的模型不允许重复执行测试。）相反，它会使用 OPT 先前执行该测试时得到的结果$_1$。伪代码见算法 3。

我们分析 ALG。为此，令 OPT 为针对 f 的最优评估策略，也就是说，在所有针对 f 的评估策略中，它的期望成本最低。确定性策略可以达到最低期望成本，因此我们假设 OPT 是确定性的。我们使用以下定义：

- 令 μ_ℓ 表示 ALG 到第 ℓ 阶段结束时尚未找到 f 的证书的概率。
- 令 μ_ℓ^* 表示 OPT 产生的成本至少为 2^ℓ 的概率，也就是说，OPT 在产生成本 2^ℓ 之前没有找到 f 的证书。

在相关工作中，[29, 14, 17], 已有研究证明了与以下结论类似的命题。虽然我们的设定不同，但证明采用了类似的思路。

引理 5. 对所有 $\ell \geq 1$，以下结论均成立：

$$\mu_\ell \leq \frac{1}{e} \cdot \mu_{\ell-1} + \mu_\ell^* .$$

证明。令 σ 为一个随机变量，其取值是部分实现，表示 ALG 在阶段 0 至 $\ell - 1$ 执行的测试所产生的结果。令 W 为所有此类非 f 证书的部分实现构成的集合。因此，如果 $\sigma \in W$，则 ALG 到阶段 $\ell - 1$ 结束时尚未完成。注意，由于 ALG 是随机化的，σ 由 x_i 的实现以及 ALG 所做的随机选择共同决定。接下来，我们将比较在 σ 的条件下，ALG 在阶段 ℓ 中的情况与 OPT 在成本 2^ℓ 内的情况。

我们假设 $\sigma \in W$。因此，ALG 接下来的任务是为诱导函数 f_σ寻找一个证书 $f = f^{\mathcal{M}}$对于分划拟阵 $\mathcal{M}$，显然 $f_\sigma = f^{\mathcal{M}'}$对于分划拟阵 $\mathcal{M}'$。因此，引理 3 和引理 4 适用于 f_σ。令 OPT$_\sigma$ 表示由 OPT 获得的策略，具体做法是：每当 OPT 对一个 x_i 满足 $\sigma_i \neq *$的对象执行测试时，跳过该测试，并继续执行，就好像测试已经完成且结果为 σ_i。显然，OPT$_\sigma$ 是 f_σ的评估策略。

现在考虑任意固定的 $\pi \in \{0,1\}^n$，其为 σ 的扩展。如果在 x 上执行的 OPT 在成本 2^ℓ 内找到 f 的 0-证书（1-证书），那么在 x 上执行的 OPT$_\sigma$ 显然会在成本 2^ℓ 内找到 f_σ 的 0-证书（1-证书）。因此，在给定 σ 的条件下，OPT$_\sigma$ 在成本 2^ℓ 内找到 f_σ 的 0-证书（1-证书）的概率，大于或等于 OPT 在成本 2^ℓ内找到 f 的 0-证书（1-证书）的概率。

在 σ 的条件下，ALG 在第 ℓ 阶段执行策略 SteepestAscent$(\sigma, 6 \cdot 2^\ell)$。因此，根据引理 4 可知

$$\begin{aligned}
&\Pr[\text{ALG finds a 0-certificate of } f \text{ in phase } \ell \mid \sigma] \\
=&\Pr[\text{ALG finds a 0-certificate of } f_\sigma \text{ in phase } \ell \mid \sigma] \\
\geq&\left(1 - \frac{1}{e}\right) \cdot \Pr[\text{OPT}_\sigma \text{ finds a 0-certificate of } f_\sigma \text{ within cost } 2^\ell \mid \sigma] \\
\geq&\left(1 - \frac{1}{e}\right) \cdot \Pr[\text{OPT finds a 0-certificate of } f \text{ within cost } 2^\ell \mid \sigma] .
\end{aligned} \tag{1}$$

利用引理 3 进行类似推理，可得寻找 1-证书的界。在 σ 的条件下，ℓ阶段中，ALG 执行 OPT 在 f_σ 上会执行的前 2^ℓ 次测试 $_1$。一个 f_σ 的 1- 证书可导出 f 的 1- 证书。因此，根据 引理 3 可得

$$\begin{aligned}
&\Pr[\text{ALG finds a 1-certificate of } f \text{ in phase } \ell \mid \sigma] \\
\geq&\Pr[\text{OPT finds a 1-certificate of } f \text{ within cost } 2^\ell \mid \sigma] .
\end{aligned} \tag{2}$$

由方程 (1)和方程 (2)，并利用以下事实：对于任意策略，寻找 1-证书和寻找

0-证书是互不相交的事件，我们有

$$\begin{aligned}
&\Pr[\text{ALG finds a certificate of } f \text{ in phase } \ell \mid \sigma] \\
=\ &\Pr[\text{ALG finds a 0-certificate of } f \text{ in phase } \ell \mid \sigma] \\
&+ \Pr[\text{ALG finds a 1-certificate of } f \text{ in phase } \ell \mid \sigma] \\
\geq\ &\left(1 - \frac{1}{e}\right) \cdot \Pr[\text{OPT finds a certificate of } f \text{ within cost } 2^\ell \mid \sigma]\,.
\end{aligned}$$

由此可得

$$\begin{aligned}
&\Pr[\text{ALG finds a certificate of } f \text{ in phase } \ell] \\
=\ &\sum_{\sigma \in W} \Pr[\sigma] \cdot \Pr[\text{ALG finds a certificate of } f \text{ in phase } \ell \mid \sigma] \\
\geq\ &\left(1 - \frac{1}{e}\right) \cdot \sum_{\sigma \in W} \Pr[\sigma] \cdot \Pr[\text{OPT finds a certificate of } f \text{ within cost } 2^\ell \mid \sigma] \\
=\ &\left(1 - \frac{1}{e}\right) \cdot \Pr[A \text{ and } B]\,,
\end{aligned} \tag{3}$$

其中， A 表示截至阶段 $\ell - 1$结束时 ALG 未找到 f的证书这一事件，而 B 表示 OPT 在成本 2^ℓ内找到 f 的证书这一事件。

现在考虑公式（3）的首行和末行。首行中的量等于 $\mu_{\ell-1} - \mu_\ell$。要为末行给出下界，注意到并集界意味着 $\Pr[A \text{ 且 } B] \geq \Pr[A] - \Pr[\neg B]$。我们还有 $\Pr[A] = \mu_{\ell-1}$。根据 B的定义，$\Pr[\neg B]$ 表示 OPT 产生至少 $2^\ell + 1$ 成本的概率，因此 $\Pr[\neg B] \leq \mu^*_\ell$。所以，末行中的量大于或等于 $(1 - 1/e) \cdot (\mu_{\ell-1} - \mu^*_\ell)$：

$$\mu_{\ell-1} - \mu_\ell \geq \left(1 - \frac{1}{e}\right) \cdot (\mu_{\ell-1} - \mu^*_\ell).$$

重新排列后可得

$$\mu_\ell \leq \frac{1}{e} \cdot \mu_{\ell-1} + \left(1 - \frac{1}{e}\right) \cdot \mu^*_\ell \leq \frac{1}{e} \cdot \mu_{\ell-1} + \mu^*_\ell\,,$$

如所述。 □

现在我们可以证明定理 1了。证明由引理 5（以及 引理 3 和 4的成本上界）可得，且相当标准（参见例如 [14, 17]）。为完整起见，我们在此重述证明。

证明*定理 1*：根据引理 3 和引理 4，阶段 0 到 ℓ 的 ALG 的期望成本至多为 $7 \cdot \sum_{j=0}^{\ell} 2^j \leq 14 \cdot 2^\ell$。ALG 以 $1 - \mu_0$ 的概率在阶段 0 结束，并以 $\ell \geq 1$ 的概率在阶段 $\mu_{\ell-1} - \mu_\ell$结束。因此，

$$\mathbb{E}[\text{cost}(\text{ALG})] \leq 14 \cdot (1 - \mu_0) + \sum_{\ell=1}^{\infty} 14 \cdot 2^\ell (\mu_{\ell-1} - \mu_\ell) = 14 + 14 \cdot \sum_{\ell=0}^{\infty} 2^\ell \mu_\ell. \tag{4}$$

类似地，根据 μ^*_ℓ 的定义，我们可以将 OPT 的成本界定为

$$\mathbb{E}[\text{cost}(\text{OPT})] \geq \sum_{\ell=0}^{\infty} 2^\ell \cdot (\mu^*_\ell - \mu^*_{\ell+1}) \geq \mu^*_0 + \frac{1}{2} \cdot \sum_{\ell=1}^{\infty} 2^\ell \mu^*_\ell = 1 + \frac{1}{2} \cdot \sum_{\ell=1}^{\infty} 2^\ell \mu^*_\ell, \tag{5}$$

其中，在最后一个等式中，我们使用了 $\mu_0^* = 1$ 这一事实。

我们定义 $\Gamma := \sum_{\ell=0}^{\infty} 2^\ell \mu_\ell$。于是有

$$\begin{aligned}\Gamma = \sum_{\ell=0}^{\infty} 2^\ell \mu_\ell &\le \mu_0 + \frac{1}{e} \cdot \sum_{\ell=1}^{\infty} 2^\ell \mu_{\ell-1} + \sum_{\ell=1}^{\infty} 2^\ell \mu_\ell^* \\ &\le \mu_0 + \frac{1}{e} \cdot \sum_{\ell=1}^{\infty} 2^\ell \mu_{\ell-1} + 2 \cdot (\mathbb{E}[\mathrm{cost}(\mathrm{OPT})] - 1) \\ &= \mu_0 + 2 \cdot \frac{1}{e} \cdot \left(\sum_{\ell=0}^{\infty} 2^\ell \mu_\ell \right) + 2 \cdot (\mathbb{E}[\mathrm{cost}(\mathrm{OPT})] - 1) \\ &\le \frac{2}{e} \cdot \Gamma + 2 \cdot \mathbb{E}[\mathrm{cost}(\mathrm{OPT})] - 1,\end{aligned}$$

其中，第一个不等式由引理 5得出，第二个不等式由公式（5）得出，最后一个不等式则由以下事实得出：$\mu_0 \le 1$。因此 $\Gamma \le 2 \cdot (1 - \frac{2}{e})^{-1} \cdot \mathbb{E}[\text{成本(OPT)}]$。由公式（4）可得 $\mathbb{E}[\text{成本(ALG)}] \le 14 + 14 \cdot (1 - \frac{2}{e})^{-1} \cdot \mathbb{E}[\text{成本(OPT)}] = O(1) \cdot \mathbb{E}[\text{成本(OPT)}]$。

最后，运行时间的界由引理 3 和引理 4 中的界以及算法3的定义得出。

□

4 在一般拟阵上最优地寻找1-证书

本节的目标是定义 OPT_1 并证明引理 3，为方便起见，我们在此重述如下。

引理 3. 设 $\mathcal{M}$ 是一个拟阵，且 $f := f^{\mathcal{M}}$。设 σ 是 f 和 $B \in \mathbb{N}$的输入变量的一个部分赋值。*$\mathrm{OPT}_1(\sigma, B)$* 的每次实现成本至多为 B。对于函数 f_σ 的任意测试策略 Z（可能是随机化的），若其每次实现成本至多为 B，则 *$\mathrm{OPT}_1(\sigma, B)$* 找到 f_σ 的一个*1*-证书的概率，至少与 Z 找到 f_σ的一个*1*-证书的概率相同。最后，*OPT_1* 可在多项式时间内运行。

虽然我们将在评估分划拟阵 $\mathcal{M} = (E, \mathcal{I})$ 上的函数 $f = f^{\mathcal{M}}$ 时使用本节结果，但这些结果对任意拟阵 $\mathcal{M}$ 同样成立。

算法 OPT_1，在给定部分实现 σ后，会贪心地尝试构造 $\mathcal{M}_\sigma$的一个基 S，其中 $\mathcal{M}_\sigma = (E, \mathcal{I}_\sigma)$ 是由 σ诱导的拟阵。算法首先将 S 初始化为空集。然后按 p_i 递减的顺序考察 x_i 上的测试；若测试 x_i 与 S中已有的元素独立，则执行该测试；若对 x_i 的测试结果为1，则将 i 加入 S。一旦 S 成为一个基，算法便停止测试；这意味着 $f^{\mathcal{M}_\sigma}(x) = 1$，因此 $f^{\mathcal{M}}(x) = 1$。如果处理完所有测试后，S仍不是一个基，那么 $f^{\mathcal{M}}(x) = 0$。（注意，可能更早就已找到一个0-证书，但为简便起见，我们忽略这一点。）伪代码见算法4。

引理 3 的证明。我们通过对 B 进行归纳法来证明该引理。首先注意到，f_σ 与 $f^{\mathcal{M}_\sigma}$ 相同（其中 $\mathcal{M}_\sigma$ 也是一个拟阵）。因此，不失一般性，我们可以假设 $\sigma = (*, \ldots, *)$。接下来，为简单起见，我们将在记号 $\mathrm{OPT}_1(\sigma, B)$ 中省略 σ，并且在 $\mathcal{M}_\sigma$ 和 f_σ 中也省略它。现在固定一个 Z。同样，不失一般性，可以假设 Z 是一种使找到 1-证书的概率最大的确定性策略。最后，再次不失一般性，假设 f 的输入变量，即 $x_1, \ldots, x_n$，编号满足 $p_1 \ge p_2 \ge \ldots \ge p_n$。

注意，$\mathrm{rank}(\mathcal{M}) = 0$ 的情形是平凡的。因此，假设 $\mathrm{rank}(\mathcal{M}) \ge 1$。根据这一假设，存在某个 x_ℓ，它是 $\mathrm{OPT}_1(B)$ 执行的第一个测试（注意，$\ell \ne 1$ 是可能的，因为

算法4： $\mathrm{OPT}_1(\sigma,\ B)$

$S \leftarrow \emptyset$;
$c \leftarrow 0$;
$L \leftarrow$ 测试列表 x_i，按 p_i降序排列的 $\sigma_i = *$；
for x_i 在 L **中执行**
  如果 $S \cup \{x_i\} \in \mathcal{I}_\sigma$ **则**
    对 x_i 执行测试;
    $c \leftarrow c + 1$;
    如果 $x_i = 1$，**则**
      $S \leftarrow S \cup \{x_i\}$;
  如果 S 是 $\mathcal{M}$ 的基$_\sigma$ **then**
    返回 "$f_\sigma(x) = 1$"
  若 $c = B$ **则**
    中断
返回 "在成本 B以内未找到 f_σ 的 1-证书"

$\mathcal{M}$ 可能包含环，即满足 $\{x_i\}$ 不独立的元素 x_i ）。令 x_j 为 Z 执行的第一个测试。不失一般性，假设 x_j 也不是环。

对于基本情况，假设 $B = 1$。那么 $\mathrm{OPT}_1(B)$ 对 x_ℓ 进行单次测试，而 Z对 x_j进行单次测试。若 $j = \ell$，则 $\mathrm{OPT}_1(B)$ 和 Z 相同，结论显然成立。假设 $j \neq \ell$。集合 $\{x_j\}$ 和 $\{x_\ell\}$ 都是独立的。由于 $\mathcal{M}$ 的所有基大小相同，因此 $\{x_j\}$ 和 $\{x_\ell\}$ 要么都是 $\mathcal{M}$的基，要么都不是。若它们都是基，则 $\mathrm{OPT}_1(B)$ 以概率 p_ℓ 找到一个 1-证书，而 Z 以概率p_j找到一个 1-证书。根据 OPT_1 的定义以及 x_j 不是环这一事实，$p_\ell \geq p_j$。若 $\{x_j\}$ 和 $\{x_\ell\}$ 都不是基，则 $\mathrm{OPT}_1(B)$ 和 Z 找到 1-证书的概率均为零。基本情况的证明至此完成。

对于归纳步骤，假设 $B > 1$。令 σ_0 为将 $x_j = 0$ 设定为值的部分实现，并令σ_1 为将 $x_j = 1$设定为值的部分实现。如果对 f而言，在x_j上进行 Z的首次测试未能得到 1-证书，Z 就会继续测试，以便找到 f_{σ_0} 的 1-证书（如果 $x_j = 0$）或 f_{σ_1} 的 1-证书（如果 $x_j = 1$）。它会以一种能最大化在剩余预算 $B - 1$内获得 1-证书概率的方式进行测试。根据归纳假设，通过在f_{σ_0} 或f_{σ_1}上执行 $\mathrm{OPT}_1(B - 1)$（视情况而定），也能最大化这一概率。因此，不失一般性，我们可以假设 Z 在完成首次测试后执行 $\mathrm{OPT}_1(B - 1)$。

如果 $j = \ell$，那么 Z 与 $\mathrm{OPT}_1(B)$ 完全相同，而 $\mathrm{OPT}_1(B)$ 也能最大化在预算 B内获得 1-证书的概率。因此，假设 $j \neq \ell$。在这种情况下，测试完x_j后，如果Z 还没有 1-证书，那么它接下来考虑测试的变量就是 x_ℓ （无论$x_j = 1$ 还是 $x_j = 0$）。我们考虑对 Z 进行修改，交换x_j 和 x_ℓ的作用。~我们将此称为 Z。现在我们证明，Z 找到 f 的 1-证书的概率，至少与 Z 找到 f的 1-证书的概率一样大。这就完成了引理的证明：由于 Z 的首次测试~是 x_ℓ，再次根据归纳法可知，OPT_1 能最大化在预算 B内找到 1-证书的概率。

我们区分两种情况：

情况 1：变量 x_j 和 x_ℓ 在 $\mathcal{M}$ 中独立（即，$\{x_j, x_\ell\}$ 是一个独立集）。

~在这种情况下，$\{x_j\}$ 和 $\{x_\ell\}$ 都不是 $\mathcal{M}$的基。由于 $B > 1$、Z和 Z 总是执行前两个测试，即对x_j 和 x_ℓ 进行测试（顺序相反）。对于这两个测试的每一对可能结果

~ ~， Z 和 Z 随后采用相同的测试策略。因此， Z 和 Z 找到 1-证书的概率相同。

情形 2：x_j 和 x_ℓ 存在依赖关系。

如上所述， $\{x_j\}$ 和 $\{x_\ell\}$ 要么都是 $\mathcal{M}$ 的基，要么都不是。假设二者都是。那么由于 $B > 1$， Z 和 Z 都会在前两个测试中获得一个 1-证书，除非 $x_\ell = x_j = 0$。~在这种情况下，在分别对 x_j 和 x_ℓ进行测试后， Z 和 Z 随后会采取相同的~策略。因此， Z 和 Z 找到 1-证书的概率相同。

~因此，假设 $\{x_j\}$ 和 $\{x_\ell\}$ 都不是 $\mathcal{M}$的基。那么 Z 和 Z 都会先以相反的顺序考虑是否测试x_j 和 x_ℓ。具体来说， Z 先测试x_j。如果 $x_j = 1$，由于依赖关系，它~会跳过接下来对 x_ℓ 的测试。如果 $x_j = 0$，它接下来会测试 x_ℓ 。策略 Z 先测试 x_ℓ。如果 $x_\ell = 1$，由于依赖关系，它会跳过接下来对x_j 的测试。如果 $x_\ell = 0$，它接下来会测试x_j 。

对于 s， $t \in \{0,1\}$，令 P_{st}表示 $x_j = s$和 $x_\ell = t$的概率。此外，对于 u， $v \in \{0,1,*\}$，令$f_{uv} = f_{\sigma'}$，其中 σ' 是这样一个部分实现：若 $u \neq *$，则将 u 赋给 x_j ；若 $v \neq *$，则将 v 赋给 x_ℓ 。

Z 找到 f 的 1-证书的概率为

$$\begin{aligned}&(P_{11}+P_{10})\cdot \Pr[\mathrm{OPT}_1(B-1) \text{ finds 1-certificate of } f_{1*}]\\ +&P_{01}\cdot \Pr[\mathrm{OPT}_1(B-2) \text{ finds 1-certificate of } f_{01}]\\ +&P_{00}\cdot \Pr[\mathrm{OPT}_1(B-2) \text{ finds 1-certificate of } f_{00}].\end{aligned}$$

~ Z 找到 f 的 1-证书的概率为

$$\begin{aligned}&(P_{11}+P_{01})\cdot \Pr[\mathrm{OPT}_1(B-1) \text{ finds 1-certificate of } f_{*1}]\\ +&P_{10}\cdot \Pr[\mathrm{OPT}_1(B-2) \text{ finds 1-certificate of } f_{10}]\\ +&P_{00}\cdot \Pr[\mathrm{OPT}_1(B-2) \text{ finds 1-certificate of } f_{00}].\end{aligned}$$

~现在我们证明，对于 Z，第一个量不大于对于 Z的第二个量。在拟阵理论中，若拟阵的两个元素 e、e' 满足 e ，且 e' 不是环，并且$\{e, e'\}$ 是一个相关集，则称 e 和 e' 是该拟阵的平行元素。若拟阵中的 e 和 e'是平行元素，则对于该拟阵中每个包含 e的基 I ，集合 $(I\backslash\{e\}) \cup \{e'\}$ 也是该拟阵的一个基（参见 [36], 命题 1.7.2）。因此，对于该拟阵中每个包含 e 的独立集 I ，集合 $(I \setminus \{e\})\cup\{e'\}$ 也是独立集；对于包含 e'的独立集 I ，对称的结论同样成立。

因此，函数 f_{1*} 和 f_{*1} 分别对应于 $\mathcal{M}$ 的子式 $\mathcal{M}_{1*}$ 和 $\mathcal{M}_{*1}$ ，二者具有相同的独立集族，其中没有任何一个包含 x_j 或 x_ℓ。由此可知，对于任意预算 B'， $\mathrm{OPT}_1(B')$ 找到 1-证书的概率对于 f_{*1}、 f_{1*}、 f_{01} 和 f_{10} 都相同。此外，由于 $\mathrm{OPT}_1(B')$ 找到函数的 1-证书的概率随 B' 单调不减，且因为 $P_{10} \leq P_{01}$，所以对于 Z，上面的第一个量不超过 ~ ~下面针对 Z 的第二个量。因此，如所声称的那样， Z 在预算 B 内找到 1- 证书的概率至少与 Z 找到 1- 证书的概率相同。

最后，运行时间的上界显而易见。 □

5 在划分拟阵上近似最优地寻找0-证书

本节介绍 SteepestAscent(σ, B)，这是我们用于在期望预算约束下最大化找到分划拟阵 0-证书的概率的近似算法。我们证明引理 4，为方便起见，我们在此重述该引理。

引理 4。 设 $\mathcal{M}$ 是一个划分拟阵，且 $f := f^{\mathcal{M}}$。设 σ 是 f 的输入变量的部分赋值，$B \in \mathbb{R}_{\geq 0}$。*SteepestAscent*$(\sigma, B)$ 的期望成本至多为 B。*SteepestAscent*(σ, B) 找到 f_σ 的 *0*-证书的概率，至少是任何（可能是随机化的）测试策略在期望成本 $B/6$ 内找到 f_σ 的 *0*-证书的概率的 $1 - 1/e$ 倍。最后，*SteepestAscent*(σ, B) 在多项式时间内运行。

SteepestAscent 的伪代码见算法 5。SteepestAscent 通过计算我们称为 IO（或 InsideOut）的策略的适当剪枝来工作，并将其应用于 $g_{1,\sigma}$、……、$g_{d,\sigma}$ 中的每一个。（回顾一下，g_j 是与划分拟阵的第 j 个划分类对应的 k_j-of-n_j 函数，而 $g_{j,\sigma}$ 是 σ 在其上诱导的函数。）我们将这些策略记为 IO$_{1,\sigma}$、……、IO$_{d,\sigma}$。IO 是一种初等策略，在我们的算法中，它由一个有向无环图表示。

现在我们给出一个关于 IO 剪枝质量的引理。关于 IO 的进一步讨论以及该引理的证明，我们留待第 6节。

引理 6. 设 $\mathcal{M}$ 是一个均匀拟阵，且 $f := f^{\mathcal{M}}$。等价地，设 f 是一个 k-*of*-n 函数。设 S 是针对 f 的一个（可能是随机化的）测试策略，其期望成本为 B，并且以概率 P 找到 f 的一个 *0*-证书。则存在一个对初等策略 *IO* 的随机剪枝 S'，其期望成本至多为 $6 \cdot B$，并且以至少 P 的概率找到 f 的一个 *0*-证书。

回顾一下，给定预算 B 和针对 f 的测试策略 S，函数 ComputePrunedStrategy（定义于 第 2.1 节）可用于生成 S 的随机剪枝，使其在所有此类剪枝中，能以最大的概率在期望成本不超过 B 的情况下找到 f 的 0-证书。利用 引理 2，由此我们得到一个针对均匀拟阵上 MP0 的多项式时间近似算法，该算法通过计算初等策略 IO 的随机剪枝来实现。

为了确定用于剪枝策略 IO 的预算 B_1、……、B_d，算法 SteepestAscent 求解一个连续背包问题。为此，SteepestAscent 首先对 IO$_{1,\sigma}$、……、IO$_{d,\sigma}$调用过程 ComputePruningFunctions（定义于第 2.1 节），分别获得函数$_{1,\sigma}$、……、IO$_{d,\sigma}$。从q_1、……、q_d开始，在全部预算投入完毕时停止，它采用连续贪心方法在各 $B_1 = \cdots = B_d = 0$ 划分类之间持续分配预算，以最大化效用之和 $d(\sum_{j\in[d]} q_j)$。我们在算法 5中给出伪代码，使用

$$\partial_+ q(a) := \lim_{x \to a, x > a} \frac{q(x) - q(a)}{x - a}$$

表示函数 q 在 $\mathbb{R}$ 处于位置 a的右导数。注意，SteepestAscent(σ,B) 在多项式时间内运行，因为每个q_j 的断点数都是多项式级的（参见引理 2）。

现在证明引理 4。

引理 *4*的证明。 成本上界直接由 SteepestAscent(σ, B) 的定义得出。因此，我们将重点放在概率上界上。令 OPT 为一种期望成本至多为 $B/6$的（可能是随机化的）策略。

考虑与 SteepestAscent(σ, B) 对应的主循环迭代结束时 B_1, ……, B_d 的取值。对于任意 $j \in [d]$, 记 S_j 为 ComputePrunedStrategy(IO$_{j,\sigma}$, B_j) 这一事件

算法 5： SteepestAscent(σ， B)

q_1，……， $q_d \leftarrow$ ComputePruningFunction(IO$_1$，σ) ……，ComputePruningFunction(IO$_d$，σ);B_1，……， $B_d \leftarrow 0$;**当** $\sum_{j\in[d]} B_j < B$ **时**，选择 $j^* \in \arg\max_{j\in[d]} \partial_+ q_j(B_j)$;$\Delta \leftarrow \Delta > 0$ 满足以下条件的最小值：q_{j^*} 在 $B_{j^\star} + \Delta$ 或 $\sum_{j\in[d]} B_j + \Delta = B$处有断点；$B_{j^\star} \leftarrow B_{j^\star} + \Delta$;**对于** $j \in [d]$ ，执行 ComputePrunedStrategy(IO$_j$，σ， B_j);

为 $g_{\sigma,\ j}$找到一个 0-证书。同样，记 B'_j 为 OPT 在$g_{\sigma,\ j}$上的期望成本，并记 T_j 为 OPT 为$g_{\sigma,\ j}$找到 0-证书的事件。我们首先注意到

$$\sum_{j\in[d]} \Pr[S_j \mid \sigma] = \sum_{j\in[d]} q_j(B_j) \geq \sum_{j\in[d]} q_j(6 \cdot B'_j) \geq \sum_{j\in[d]} \Pr[T_j \mid \sigma]\,. \tag{6}$$

这个等式直接由 引理 2 (ii) 得出。对于第一个不等式，首先注意到，随机化策略 SteepestAscent(σ, B) 在约束 $\sum_{j\in[d]} q_j(B_j)$ 下使 $\sum_{j\in[d]} B_j = B$ 最大化（利用 q_j的凹性；引理 2 (i)）。于是，由于 $B \geq 6 \cdot \sum_{j\in[d]} \cdot B'_j$ 和 q_j 是单调的（同样由引理 2 (i)），该不等式便得证。第二个不等式直接由 引理 6 得出。

下面，我们将事件视为取值为 0 或 1 的随机变量。设 $S'_1, \ldots, S'_d$ 为事件，其边缘概率分别与 $S_1, \ldots, S_d$ 相同，但彼此相关的方式使 $\mathbb{E}[\max(S'_1, \ldots, S'_d)]$ 最大。注意，$\max(S'_1, \ldots, S'_d)$ 表示至少有一个 $S'_1, \ldots, S'_d$ 发生这一事件；并且，当 $S'_1, \ldots, S'_d$ 不同时发生时， $\mathbb{E}[\max(S'_1, \ldots, S'_d)]$ 取得最大值，除非它们的概率之和为 1（此时 $\max(S'_1, \ldots, S'_d) = 1$ 以概率 1 发生）。于是我们得到

$$\begin{aligned}
&\Pr[\text{SteepestAscent}(\sigma, B) \text{ finds 0-certificate of } f_\sigma \mid \sigma] \\
&= \mathbb{E}[\max(S_1, \ldots, S_d) \mid \sigma] && \text{(Def. of } S_1, \ldots, S_d) \\
&\geq \left(1 - \frac{1}{e}\right) \cdot \mathbb{E}[\max(S'_1, \ldots, S'_d) \mid \sigma] && (*) \\
&= \left(1 - \frac{1}{e}\right) \cdot \min\left(1, \sum_{j\in[d]} \Pr[S_j \mid \sigma]\right) && \text{(Def. of } S'_1, \ldots, S'_d) \\
&\geq \left(1 - \frac{1}{e}\right) \cdot \min\left(1, \sum_{j\in[d]} \Pr[T_j \mid \sigma]\right) && \text{(Equation (6))} \\
&\geq \left(1 - \frac{1}{e}\right) \cdot \Pr[\text{OPT finds 0-cert. within exp. cost } B/6 \mid \sigma] && \text{(Def. of } T_1, \ldots, T_d, \text{ union bound)},
\end{aligned}$$

其中，标有（*）的步骤是一个标准论证，不过也可以通过调用相关性差距 [1]。

最后，运行时间的多项式界可由以下事实结合得出：IO 是初等的，引理 2，以及算法 5的定义。□

算法 6： 轮转($\mathcal{A}$，$\mathcal{B}$，f)

$\sigma \leftarrow (*, \ldots\ldots, *)$;$i \leftarrow 0$;**当** σ 不是 f 的证书**时**执行 i中的第 -th 次测试（如果存在），以及 $\mathcal{A}$ 中的第 -th 次测试（如果存在）；更新 σ，使其包含刚刚执行的测试结果；$i \leftarrow i+1$;

算法 7： InsideOut(f)

$\mathcal{L}_0 \leftarrow \{x_1, \ldots, x_n\}$ 中所有元素按 p_i 降序排列的序列，其中 $p_i \le \frac{1}{2}$；
$\mathcal{L}_1 \leftarrow \{x_1, \ldots, x_n\}$ 中所有元素按 p_i 升序排列的序列，其中 $p_i > \frac{1}{2}$；
执行轮转($\mathcal{L}_0$,$\mathcal{L}_1$,f);

6 一种用于均匀拟阵上 MP0 的 $O(1)$-近似算法

本节正式定义 IO（InsideOut），并证明引理 6成立。为方便起见，我们在此重述该引理。

引理 6。 设 $\mathcal{M}$ 是一个均匀拟阵，且 $f := f^{\mathcal{M}}$。等价地，设 f 是一个 k-*of*-n 函数。设 S 是针对 f 的（可能经过随机化的）测试策略，其期望成本为 B，并且以概率 P找到 f 的一个 *0*-证书。则存在一个对初等策略 *IO* 进行随机剪枝的方案 S'，其期望成本至多为 $6 \cdot B$，并且以至少 P的概率找到 f 的一个 *0*-证书。

设 f 是一个 k-of-n 函数。设 $\mathcal{A}$ 和 $\mathcal{B}$ 是来自集合 $\{x_1, \ldots, x_n\}$的互不相交的有序随机变量列表，且每个变量都出现在这两个列表之一中。我们首先介绍轮转测试策略 RoundRobin($\mathcal{A}, \mathcal{B}, f$)，其伪代码见算法 6。

我们将伪代码中 while循环的第 i 次迭代称为轮转的第 i轮次。在某次迭代 i 中，如果两个列表中只有一个包含第 i 个元素（当一个列表比另一个长时可能出现这种情况），那么该轮次就只包含这一个元素。此外，RoundRobin($\mathcal{A}$，$\mathcal{B}$，f) 的写法满足我们称为轮次完成（RC）的性质，即策略不能在一轮次中途终止。因此，即使在某一轮次中测试第一个随机变量后就能确定 f 的值，如果该轮次中还存在另一个随机变量，也必须先对其进行测试，策略才能终止。从算法角度来看，轮次完成性质并非必需，但它简化了我们的表述和分析。RoundRobin($\mathcal{A}$，$\mathcal{B}$，f) 会对 f 进行求值，因为它会在必要时测试所有随机变量，并且不会在没有证书的情况下终止。

我们的基础策略 InsideOut(f)（参见 算法 7）执行 RoundRobin($\mathcal{L}_0$，$\mathcal{L}_1$，f)，其中 $\mathcal{L}_0$ 是由 $p_i \le \frac{1}{2}$按 p_i降序排列的元素序列，而 $\mathcal{L}_1$ 是 $\{x_1, \ldots, x_n\}$ 中所有 $p_i > \frac{1}{2}$按 p_i升序排列后组成的序列。对于分布相同的随机变量，我们假定其顺序任意但固定。下文中，为简洁起见，我们通常省略参数 f，并将上述过程简称为 RoundRobin($\mathcal{A}$，$\mathcal{B}$) 和 InsideOut。

如果一个策略 j-伪 *IO* 与 InsideOut 相同，但x_j 在 $\mathcal{L}_0$ 或 $\mathcal{L}_1$ 中的第一个元素可能未按正确的排序顺序排列，我们就称该策略为 j-伪 *IO* 。也就是说，$\mathcal{L}_0$ 仍包含具有 $p_i \le \frac{1}{2}$ 的随机变量，而 $\mathcal{L}_1$ 仍包含具有 $p_i > \frac{1}{2}$ 的随机变量，但这两个序列中的一个以 x_j 开头，且排序标准（对于 $\mathcal{L}_0$，p_i 按降序排列；对于 $\mathcal{L}_1$，p_i 按升序排列）只需从该序列的第二个元素开始满足。

利用 f 是一个 k-n 函数这一事实，很容易证明 IO 是一种初等策略（定义见 第 2 节）。IO 按固定顺序测试其元素，直到在一轮结束时可以确定 f 的值。在第 ℓ 轮结束时，已有固定数量的 n_ℓ 个元素经过测试。仅当这些 n_ℓ 个实现中恰好有 k 个或 $k+1$ 1 个时，IO 才会在这一轮结束时终止，其余的均为 0；或者恰好有 $n-k+1$ 个或 $n-k+2$ 0 个时，其余的均为 1。因此，是否终止仅取决于观测到的 1 的数量和 0 的数量。

6.1 IO的分析

我们先给出一些定义。设 S 是针对 k-of-n 函数 f 的（可能是随机化的）测试策略。为使记号简洁，本节还允许 $k \le 0$ 和 $k > n$。第一种情况下，f 在所有实现中的值均为 1；第二种情况下，f 在所有实现中的值均为 0。第一种情况下，$(*, *, \ldots, *)$ 是 f 的 1- 证书；第二种情况下，它是 f 的 0-证书。这两种情况下，无需执行任何测试即可找到该证书。这里，S 的效用记作 $U(S)$，它是一个随机变量：若 S 找到 f 的 0-证书，其值为 1；否则为 0。对于整数 m，我们将 $U_m(S)$ 定义为一个指示随机变量，表示 S 在终止[5]之前，观察到的（在已测试变量的实现中）至少有 m 0 个。我们采用简写 $\bar{k} := n - k + 1$.¯。找到一个 k-of-n 函数的 0-证书，意味着观察到至少 k 个测试结果为 0，因此 $U(S) \equiv U_{\bar{k}}(S)$。我们用 $\bar{p}_i := 1 - p_i$ 表示对 x_i 的测试结果为 0 的概率。最后，我们所说的（随机化）IO的RC-剪枝，是指满足轮次完成性质的（随机化）IO的剪枝。

在介绍引理 6的证明之前，我们先给出以下证明概要。概括而言，我们通过对 n进行归纳来证明该结果，并通过要求 S' 是初等策略 IO 的随机化 RC-剪枝（而不只是随机剪枝）来加强引理陈述。随机化测试策略 S 对应于一个确定性策略上的分布。固定一个来自该分布 S 支撑集的策略。在归纳论证中，我们指定一系列针对 S的变换，这些变换在期望意义下保持其效用，同时仅使其成本增加一个有界量。首先，我们保留 x_i 中 S 的第一个元素，并利用归纳假设，将S 根节点的左子树替换为 IO 在该诱导子实例上的随机化 RC-剪枝；右子树也同样处理。因此，两个子树都会执行在相同的两个列表 $\mathcal{L}_0$ 和 $\mathcal{L}_1$之间进行轮转的（随机化、经剪枝）版本。所得的树尚未成为 IO 在完整实例上的随机剪枝，主要是因为

(i) 对 x_i 的测试在开始轮转之前进行，并且

(ii) 首先测试 x_i 可能违反 IO 所采用的测试顺序。

我们通过将 x_i 移至后续轮转的第一轮来处理 (i)。更具体地说，如果$p_i \le \frac{1}{2}$，我们就把它前置到 $\mathcal{L}_0$；否则,¯ 我们就把它前置到 $\mathcal{L}_1$。这样会使受影响列表中的元素移至相邻轮次，且不会增加额外成本。由此得到的策略是 i-伪 IO 的随机化 RC-剪枝。要处理 (ii)，我们需要将这一基于 i-伪 IO 的策略转换为标准 IO 的随机化 RC-剪枝，这要求将 x_i 移至其列表中的正确位置。证明接下来最复杂的部分将表明，可以通过一种方式完成这一操作，使期望效用保持不变，且期望成本不会增加太多（参见 引理 7）。与引理 3 （见第 4 节）的证明一样，该引理证明了贪心算法在任意拟阵上寻找 1-证书的最优性；这里的证明也采用归纳法，并包含情形分析。不过，此处的情形分析复杂得多。引理 3 的证明依赖于一种标准的

[5]为方便起见，这包括对 $m \le 0$ 使用 $U_m(S) = 1$。

交换论证：在策略中仅交换两个测试的作用；而这里则需要对策略进行更复杂的重构。这种复杂性源于 IO 的性质（包括其轮次结构）、0-证书的性质，以及此处的论证以期望成本为依据这一事实。

为了分析移动 x_i 所带来的期望成本增加，我们引入如下定义的 摊销成本 。对于使用 $p_i \le \frac{1}{2}$ 测试 x_i，当 $x_i = 0$ 时，摊销成本 acost(i) 为 1，否则为 0。对于 $p_i > \frac{1}{2}$，当 $x_i = 1$ 时，摊销成本为 1，否则为 0。因此，acost(i) 成为了一个取决于 x_i 的随机变量。我们使用随机变量 acost(S) 表示策略 S 的摊销成本。摊销成本与（实际）成本的期望之间可相差一个常数因子：对于 $p_i \le \frac{1}{2}$ 和 $p_i > \frac{1}{2}$，在 x_i 上进行测试的期望摊销成本至少为 $1\ 2$ （至多为 1）。因此，根据期望的线性性，

$$\frac{1}{2} \cdot \mathbb{E}[\mathrm{cost}(S)] \le \mathbb{E}[\mathrm{acost}(S)] \le \mathbb{E}[\mathrm{cost}(S)] \tag{7}$$

对于任意策略 S,

现在我们给出用于处理 (ii) 的技术引理：

引理 7。 令 $i \in [n]$，并令 $\mathcal{I}_i$ 表示 i-伪 *IO* 策略，用于 k-n-函数 $f(x_1, \ldots\ldots, x_n)$[6]。令 S 为 $\mathcal{I}_i$的随机化 *RC*-剪枝，其期望摊销成本为 $\mathbb{E}[\mathrm{acost}(S)] = B$。则存在一个 S'' 的随机化 *RC*-剪枝，该剪枝针对 f的 *IO* 策略，且 S'' 具有以下性质：

- $\mathbb{E}[\mathrm{acost}(S'')] \le B + 1$
- $\mathbb{E}[U_{\bar{k}}(S'')] \ge \mathbb{E}[U_{\bar{k}}(S)]$

我们定义出以下证明 引理 7 直到 第 6.1.1 节 并继续证明 引理 6 .

引理 *6* 的证明。 对 n（测试次数）使用归纳假设。

归纳假设：设 S 是针对 $f(x_1, \ldots\ldots, x_n)$ 的一种（可能是随机化的）测试策略，其期望成本为 B，期望效用为 P。对于 $C_1 \ge 6$，存在一种 IO 的随机化 RC 剪枝 S'，其 $\mathbb{E}[\mathrm{acost}(S')] \le \frac{C_1}{2} \cdot B$，期望效用为 $\mathbb{E}[U(S')] \ge P$。

请注意，归纳假设与引理 6 的陈述不同之处在于，它对剪枝的期望摊销成本进行了界定。它还要求剪枝满足轮次完成性质。根据公式（7），归纳假设至少与引理陈述一样强。

对于 $n = 1$，任意随机化策略都会以某种概率对唯一的元素执行测试，否则便立即停止。这是一种对 IO 的随机化 RC 剪枝（无需进行任何变换）。根据 公式（7），期望摊销成本至多为 S 的期望成本，并且当 $C_1 \ge 2$ 时，该陈述成立。

假设归纳假设对 $n - 1 \ge 1$ 成立。我们将证明它对 n 也成立。

为此，考虑针对 f 的随机化策略 S，它对应于确定性策略上的一个分布。令 J 为索引集，使得其中满足 $j \in J$ 的 S_j 是该分布支撑集中的确定性策略，其中 S_j 以概率 P_j 被选中。对于每个 $j \in J$，我们将构造一个与之关联的 IO R''_j 的随机化 RC-剪枝，使得

$$\mathbb{E}\big[\mathrm{acost}\big(R''_j\big)\big] \le \frac{C_1}{2} \cdot \mathbb{E}[\mathrm{cost}(S_j)] \qquad \text{and} \tag{8}$$

[6]这使得 x_i 成为 IO 中可能不在正确位置的元素。

$$\mathbb{E}\left[U_{\bar{k}}\left(R''_j\right)\right] \geq \mathbb{E}[U_{\bar{k}}(S_j)]\,. \tag{9}$$

固定 $j \in J$。如果 S_j 不测试任何随机变量，那么它显然是 IO 的一个 RC-剪枝。令 $R''_j = S_j$。在这种情况下，如果 $\bar{k} \leq 0$，则 $U_{\bar{k}}(S_j) = 1$；否则 $U_{\bar{k}}(S_j) = 0$。于是，对于 $C_1 \geq 0$，有 $0 = \mathbb{E}\left[\mathsf{acost}\left(R''_j\right)\right] \leq \frac{C_1}{2} \cdot \mathbb{E}[\mathsf{cost}(S_j)] = 0$，且 $\mathbb{E}\left[U_{\bar{k}}\left(R''_j\right)\right] \geq \mathbb{E}[U_{\bar{k}}(S_j)]$。

对于 S_j 至少测试一个元素的情况，设 x_i 为第一个被测试的元素。如果 $k \leq 0$，那么我们可以将 R''_j 设为不测试任何随机变量的策略，从而得到 $U_{\bar{k}}\left(R''_j\right) = 1$ 和 $\mathbb{E}\left[\mathsf{acost}\left(R''_j\right)\right] = 0$。下面给出 $\bar{k} \geq 1$ 的情况。令 S_j^0 和 S_j^1 分别表示 S_j 根据 x_i 的值为 0 或 1 而采用的（可能为空的）子策略。因此我们有

$$\mathbb{E}[\mathrm{cost}(S_j)] = 1 + \bar{p}_i \cdot \mathbb{E}\left[\mathrm{cost}\left(S_j^0\right)\right] + (1 - \bar{p}_i) \cdot \mathbb{E}\left[\mathrm{cost}\left(S_j^1\right)\right] \tag{10}$$

and

$$\mathbb{E}[U_{\bar{k}}(S_j)] = \bar{p}_i \cdot \mathbb{E}\left[U_{\bar{k}-1}\left(S_j^0\right)\right] + (1 - \bar{p}_i) \cdot \mathbb{E}\left[U_{\bar{k}}\left(S_j^1\right)\right].$$

对于 x_i 的任一结果，我们都可以得到一个关于 $n-1$ 个变量的期望预算问题诱导实例：若 $x_i = 0$，则该诱导实例对应一个 k-of-$(n-1)$ 函数；若 $x_i = 1$，则该诱导实例对应一个 $(k-1)$-of-$(n-1)$ 函数。因此，根据归纳假设，我们可以将 S_j^0 替换为 IO 的一个随机化 RC-剪枝 R_j^0（针对该诱导子实例），其期望效用至少为 $\mathbb{E}\left[U_{\bar{k}-1}\left(S_j^0\right)\right]$，且 $\mathbb{E}\left[\mathsf{acost}\left(R_j^0\right)\right] \leq \frac{C_1}{2} \cdot \mathbb{E}\left[\mathsf{cost}\left(S_j^0\right)\right]$；并将 S_j^1 替换为 IO 的一个随机化 RC-剪枝 R_j^1（同样针对该诱导子实例），其期望效用至少为 $\mathbb{E}\left[U_{\bar{k}}\left(S_j^1\right)\right]$，且$\mathbb{E}\left[\mathsf{acost}\left(R_j^1\right)\right] \leq \frac{C_1}{2} \cdot \mathbb{E}\left[\mathsf{cost}\left(S_j^1\right)\right]$。我们可以假设两个分支中剪枝的分布相互独立。令 R_j 表示由此得到的随机化策略，现在考虑用 R_j 替换 S_j。

由于测试 x_i 的摊销成本至多为 1，因此我们有

$$\mathbb{E}[\mathrm{acost}(R_j)] \leq 1 + \bar{p}_i \cdot \mathbb{E}\left[\mathrm{acost}\left(R_j^0\right)\right] + (1 - \bar{p}_i) \cdot \mathbb{E}\left[\mathrm{acost}\left(R_j^1\right)\right] \tag{11}$$

$$\leq 1 + \frac{C_1}{2}\left[\bar{p}_i \cdot \mathbb{E}\left[\mathrm{cost}\left(S_j^0\right)\right] + (1 - \bar{p}_i) \cdot \mathbb{E}\left[\mathrm{cost}\left(S_j^1\right)\right]\right]. \tag{12}$$

只要 $C_1 \geq 2$，这就意味着 $\mathbb{E}[\mathsf{acost}(R_j)] \leq \frac{C_1}{2} \cdot \mathbb{E}[\mathsf{cost}(S_j)]$；然而，正如证明阐述中所提到的，$R_j$ 通常还不是 IO 的 RC 剪枝上的一个分布。

我们首先恢复轮次完成性质。注意，尽管 R_j^0 和 R_j^1 的所有分量采用了不同的停止准则，但它们都基于相同的（可能为空的）列表 $\mathcal{A}$ 和 $\mathcal{B}$；这些列表由元素的概率得出，既不取决于停止准则，也不取决于所需的 0 的数量。这两个列表中，一个包含概率不超过 $1/2$ 的随机变量（x_i 除外），另一个包含概率大于 $1/2$ 的随机变量（x_i 除外）。如果 p_i 不超过 $1/2$，则令 $\mathcal{A}$ 为前一个列表；否则令 $\mathcal{A}$ 为后一个列表——也就是说，无论哪种情况，$\mathcal{A}$ 都是包含概率与 x_i 的概率处于同一范围内的元素的列表。相应地，$\mathcal{B}$ 表示另一个列表。令 $\mathcal{A}'$ 表示将 x_i 前置于 $\mathcal{A}$ 所得到的变量序列。如果 $\mathcal{B}$ 为空，那么 R_j 的任一确定性分量已经基于“轮次结构”$\{x_i\}\{x_{a_1}\}\{x_{a_2}\}\ldots$，其中花括号表示测试轮次以及之后可能发生的停止，因此它已经是一个符合轮次完成性质的 i-伪 IO 策略。如果 $\mathcal{B}$ 非空，那么 R_j 的任一确定性分量具有形式为 $\{x_i\}\{x_{b_1}\}\{x_{b_2}\}\ldots$ 或 $\{x_i\}\{x_{b_1}, x_{a_1}\}\{x_{b_2}, x_{a_2}\}\ldots$ 的轮次结构（取决于 $\mathcal{A}$ 是否为空），其中轮次以 ℓ 编号；在包含 $\ell \geq 2$ 的第 ℓ 轮中，如果变量 $x_{b_{\ell-1}}$ 和 $x_{a_{\ell-1}}$ 存在，则对其进行评估；而 R_j 可能会根据结果和内部随机化停止。

另一方面，符合轮次完成性质的伪 IO 策略所需的轮次结构为 $\{x_i, x_{b_1}\}\{x_{b_2}\}\ldots$ 或 $\{x_i, x_{b_1}\}\{x_{a_1}, x_{b_2}\}\ldots$。固定 R_j 的一个确定性分量以及一个实现 σ。考虑通过修改 R_j 的分量，使其遵循伪 IO 所需的轮次结构而得到的确定性策略；该策略按原先相同的顺序测试相同的元素，并测试满足 RC 性质所需的任何额外元素。对于任意实现 σ，修改轮次结构后，以下性质成立：

- 原策略评估的所有随机变量也都会被评估。
- 至多会额外评估一个随机变量。
- 对于序列 $\mathcal{A}'$ 和 $\mathcal{B}$，轮次完成性质均得到满足。

最后，移除找到证书后进行的任何测试轮次。因此，修改后的策略是 i-伪 IO $\mathcal{I}_i$ 的一种 RC-剪枝，其额外期望摊销成本至多为 1（来自至多一次额外测试），且期望效用至少与原策略相同（修改后仍会找到任何已找到的证书）。对 R_j 的每个确定性分量应用此变换，并将所得策略记为 R'_j；该策略现在是 i-伪 IO $\mathcal{I}_i$ 的 RC-剪枝的一种分布，因此是 $\mathcal{I}_i$ 的一种随机化 RC-剪枝，且

$$\mathbb{E}\big[\mathrm{acost}\big(R'_j\big)\big] \le 1 + \mathbb{E}[\mathrm{acost}(R_j)] \tag{13}$$
$$\mathbb{E}\big[U_{\bar{k}}\big(R'_j\big)\big] \ge \mathbb{E}[U_{\bar{k}}(R_j)]. \tag{14}$$

根据引理 7， 存在 IO 的一种随机化 RC-剪枝 R''_j，具有以下性质 ：

$$\mathbb{E}\big[\mathrm{acost}\big(R''_j\big)\big] \le 1 + \mathbb{E}\big[\mathrm{acost}\big(R'_j\big)\big] \tag{15}$$
$$\mathbb{E}\big[U_{\bar{k}}\big(R''_j\big)\big] \ge \mathbb{E}\big[U_{\bar{k}}\big(R'_j\big)\big]. \tag{16}$$

结合方程(15)、(13)、(12) 和(10)，我们得到

$$\mathbb{E}\big[\mathrm{acost}\big(R''_j\big)\big] \le 2 + 1 + \frac{C_1}{2}\big[\bar{p}_i \cdot \mathbb{E}\big[\mathrm{cost}\big(S_j^0\big)\big] + (1-\bar{p}_i)\cdot \mathbb{E}\big[\mathrm{cost}\big(S_j^1\big)\big]\big] \tag{17}$$
$$\le \frac{C_1}{2}\cdot\big[1 + \bar{p}_i \cdot \mathbb{E}\big[\mathrm{cost}\big(S_j^0\big)\big] + (1-\bar{p}_i)\cdot \mathbb{E}\big[\mathrm{cost}\big(S_j^1\big)\big]\big] \tag{18}$$
$$\le \frac{C_1}{2}\cdot \mathbb{E}[\mathrm{cost}(S_j)] \tag{19}$$

其中，只要满足 $\frac{C_1}{2} \ge 3$，第二个不等式就成立。

为了完成证明，回顾初始的随机化策略 S，该策略以 P_j的概率选择确定性策略S_j S''。考虑策略 R''_j，其以 P_j的概率选择策略S''。这是 IO 的随机化 RC 剪枝上的一个分布，因此其本身也是 IO 的随机化 RC 剪枝。我们已经证明，对于每个 $j \in J$，方程（8）和（9）均成立。对 J应用期望的线性性可知，这意味着归纳假设成立；结合公式（7），引理 6对任意 $C_1 \ge 6$均成立。

□

6.1.1 引理 7 的证明

接下来给出引理 7的证明。这里展示两个示例，其余情况见附录。

引理 7的证明。设 $\mathcal{A}=(x_i,\ x_{a_1},\ \ldots\ldots,\ x_{a_m})$ 和 $\mathcal{B}=(x_{b_1},\ \ldots\ldots,\ x_{b_\ell}$是 $\mathcal{I}_i$进行轮转时所用的两个列表，因此，根据 x_i 的定义，$\mathcal{A}$是列表 $\mathcal{I}_i$中可能不在正确位置的元素。注意，对于 $\mathcal{I}_i$ 的任意 RC-剪枝，该元素都相同，而且 m 和 ℓ 也可能为 0。证明通过对 m进行归纳法得到。

如果 $k\le 0$，我们可以将 S'' 设为不测试任何随机变量的策略。这是 IO 的一种 RC-剪枝，因此引理陈述可直接由 $U_{\bar{k}}(S'')=1$和 acost$(S'')=0$得出。如果 $m=0$，列表 $\mathcal{A}$ 已经按正确顺序排列，显然引理成立。如果 $m>0$，并且根据 InsideOut 的定义，x_i已经处于正确位置，那么引理也成立。

对于其余情况，x_{a_1}在 $\mathcal{A}$中，且$|\frac{1}{2}-p_i|\ge|\frac{1}{2}-p_{a_1}|$。对于归纳步骤，令 $M\ge 1$，并假设引理陈述对所有满足 $m<M$的实例都成立。我们将证明引理对$m=M$成立。考虑一个满足 $m=M$的实例。列表 $\mathcal{A}$同时包含 x_i和 x_{a_1}，总共由 $m+1$个元素组成。下文中，对于 $k\in\{1,2\}$，如果变量 x_{b_k} 满足存活 条件，即 $k\le\ell$（$\mathcal{B}$的长度），那么我们称其为存活变量；也就是说，如果 x_{b_k} 作为一个元素存在于 $\mathcal{B}$中。我们也称 x_i 和 x_{a_1}为存活变量，它们都属于 $\mathcal{A}$。

令 J 为索引集，使得在与随机化策略 S 对应的、 $\mathcal{I}_i$ 的确定性 RC-剪枝分布的支撑集中，带有 $j\in J$ 的 S_j 是 $\mathcal{I}_i$ 的确定性 RC-剪枝。根据期望的线性性，只需证明对于每个 S_j，都存在一个 IO 的随机化 RC 剪枝 R_j，使得

$$\mathbb{E}[\text{acost}(R_j)]\le\mathbb{E}[\text{acost}(S_j)]+1 \tag{20}$$

$$\mathbb{E}[U_{\bar{k}}(R_j)]\ge\mathbb{E}[U_{\bar{k}}(S_j)]. \tag{21}$$

于是，只要将 S 支撑集中的每个 S_j 替换为 R_j（不改变与 S_j 相关的概率），便可得到所需结果；所得策略是 IO 的随机化 RC 剪枝上的一个分布，因此其本身也是 IO 的随机化 RC 剪枝。

现在固定 j。如果 S_j 不测试任何元素，那么它已经是一个（平凡的）IO 的 RC 剪枝。因此，假设 S_j 至少进行一轮测试。

由于 S_j 是 $\mathcal{I}_i$的 RC 剪枝，其第一轮元素是 $\{x_i,x_{b_1}\}$ 的存活元素，而（如果执行）其第二轮元素则是 $\{x_{a_1},x_{b_2}\}$的存活元素。

我们不会直接从 S_j 构造 R_j。相反，我们先在对变量 x_{b_1} 和 x_{b_2}（如果它们存活）取固定但任意的值进行条件化后，为 S_j 构造替换，然后将 S_j 的替换组合起来形成 R_j。直观地说，条件化确保对 x_{b_1} 或 x_{b_2} 的测试只有一种可能的结果。

令 σ 表示由这两个元素所选实现构成的部分实现。在给定 σ的条件下，在前 2 轮中，S_j 只能依赖于 x_i 的取值和 x_{a_1}的取值（例如，图 4左侧树的前两层对此有所展示）。我们构造一个替换，其形式为对 IO 进行随机化 RC 剪枝，并且同样以 σ为条件。根据 IO 策略的定义，这种随机化 RC 剪枝会在第一轮测试 $\{x_{a_1},x_{b_1}\}$ 中的存活元素，并在第二轮测试 $\{x_{a_2},x_{b_2}\}$ 中的存活元素（假设该轮次确实进行）。

下面我们证明，对于这种随机化 RC- 剪枝，在给定 方程 (20) 和 (21) 的条件下，上述方程成立。我们通过情形分析来完成证明，其中每一种（我们稍后会看到共有七种）情形，都对应于 σ在第一轮测试 S_j (给定 σ 的条件下）完成后的停止行为——有证书或没有证书[7]。在情形分析中，我们将 x_{b_1} 和 x_{b_2} 都视为存活。如果情况并非如此，论证也很容易推广，因为情形分析是以这些变量的固定取值为条件的。

根据对 x_{b_1} 和 x_{b_2} 的四种取值组合应用全期望定律，只要能以

[7]具体而言，情形之间的区别以及相关的替换，并不取决于第二轮中被测试元素的实现。

良好定义的方式将不同的替换合并为策略 R_j，这些不等式就总体成立。正如我们稍后将讨论的，这实际上是可行的，因为在我们的替换中，在测试 x_{b_1} 或 x_{b_2} 之前（也只有在测试过、因而知道它们的取值之后），关于下一项测试以及是否停止的决策都不会依赖于它们的取值。

因此，我们固定 x_{b_1} 和 x_{b_2} 的取值，并将分为两个层级的分析细分为若干情况——在较低层，我们根据 S_j 在前两个测试轮次中的性质进行分析，如上所述；在顶层，则根据 $\bar{p}_i \geq \frac{1}{2}$ 是否成立进行分析。我们通过定义 $\bar{k}_1 := k - (1 - x_{b_1})$ 和 $\bar{k}_2 := k - (2 - x_{b_1} - x_{b_2})$，将 x_{b_1} 和 x_{b_2} 的取值纳入考虑。

对于每种情况，我们对策略 S_j 进行变换：该变换要么直接得到 IO 的随机化 RC-剪枝，要么交换元素 x_i 和 x_{a_1}，使处于错误位置的元素现在位于较低层，从而可以对该子实例应用归纳假设。我们将证明，归纳假设中的不等式在变换后仍然成立。

考虑 S_j。我们将明确分析其前 2 个轮次中的成本和效用变化。令 T^0 表示当 $x_i = 0$ 时 S_j 的子策略，以及当 $x_i = 1$ 时 T^1 的子策略；令 T^{uv} 表示对于 $u,\ v \in \{0, 1\}$，在针对 x_i 和 x_{a_1} 的相应结果组合执行前两个测试轮次后，在诱导子实例上的策略（其中包括不进行任何测试、直接停止并带有或不带有证书的策略）；对于表示变换中间步骤的策略 S'_j，在第一轮中 x_{a_1} 的实现为 0 和 1 时，我们还分别使用 T'^0 和 T'^1 表示其子策略。

情况 1： $\bar{p}_i \geq \frac{1}{2}$

下文将其概括为情况 1。我们假设 $\bar{p}_i \geq \frac{1}{2}$，因此 $\bar{p}_{a_1} \leq \bar{p}_i$。注意，根据 acost 的定义，这意味着 $\mathbb{E}[\mathsf{acost}(a_1)] = \bar{p}_{a_1}$ 且 $\mathbb{E}[\mathsf{acost}(i)] = \bar{p}_i$。

回想一下，不测试任何随机变量的策略已在前文处理。因此，我们可以根据第一轮测试后子策略 T^0 和 T^1 的性质，区分若干子情况。具体而言，对于每个子策略，我们根据其是否属于以下情况进行分类：(a) 由于第一轮 S_j 的结果找到了一个 0-证书，因而立即终止且不再进行任何测试；(b) 未找到 0-证书便终止，且不再进行任何测试；或 (c) 对 x_{a_1} 和 x_{b_2} 进行测试，并可能在此之后继续进行更多测试。我们将这三种情况分别记为 证书、中止 和 继续，并将它们的组合用作相应情况的标识符——例如，证书-继续 表示 T^0 不测试任何元素，因为已经找到了一个 0-证书，而 T^1 则至少继续进行一轮测试。

因此，T_0 和 T_1各有 3 种可能的性质，T_0和 T_1 的性质组合共有 9 种。然而，对于伪 IO 的 RC-剪枝 S_j，如果 T_0 不具有证书 性质，那么 T_1 就不可能具有该性质（因为我们正在寻找 0-证书），因此子情况的数量减少到 7 种。

这里给出情况 1 的两个子情况——情况 1.1（继续-继续），它展示了通用方法；以及情况 1.2（继续-中止），它要求以介于 0 和 1 之间的概率作出停止决策。情况 1 的其他复杂程度各异的子情况见附录。其中有一种情况（中止-继续）比较特殊：若 $x_i = 0$，在没有证书的情况下停止，但若 $x_i = 1$则继续，这种做法并非最优，因此可将其归约为另外两种情况。情况 2（$\bar{p}_i < \frac{1}{2}$）的 7 个子情况也在附录中讨论。遗憾的是，这些子情况不能直接通过对称性从情况 1 推得，因为只有找到 k 0 才能获得效用。

情况 1.1 (继续-继续)：无论 x_i 的值如何，都会执行两轮测试。这种情况下决策树顶部的示意图见图 4，其中展示了前两轮测试的变量。仅取决于对 x_i 和 x_{a_1} 的测试结果（因为我们已基于对 x_{b_1} 和 x_{b_2} 的测试结果进行条件化），策略随后继续执行子树（子策略）T^{11}、T^{10}、T^{01} 或 T^{00}。考虑 S'_j，它由以下方式得到

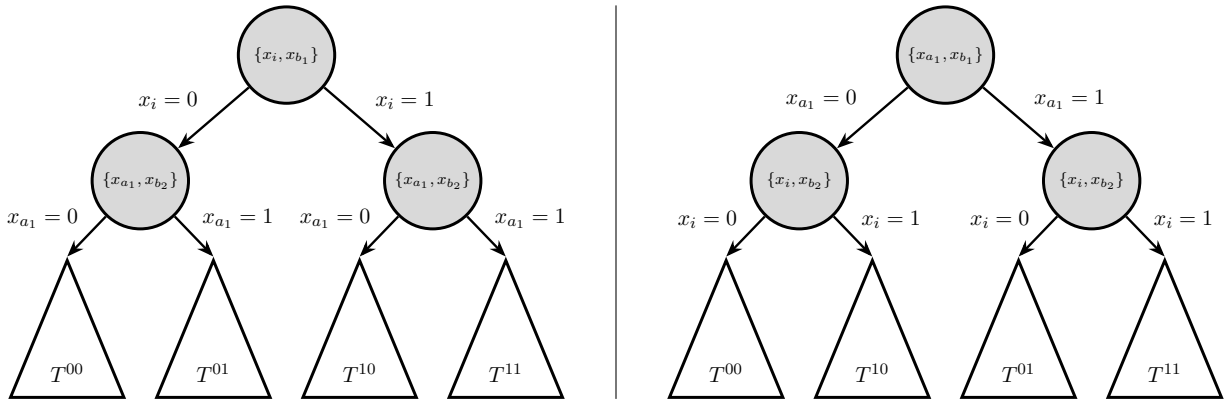

图 4：情况 1.1（继续-继续）。 S_j 在左侧，S'_j 在右侧（注意， T^{01}和 T^{10} 的角色互换了，因此到达这些子树的概率与之前相同）。

通过交换元素 x_i 和 x_{a_1}的位置，以及子树 T^{01} 和 T^{10}的位置，从 S_j 得到。在 S'_j中，第二轮之后的 4 棵子树各自被到达的概率与 S_j 中相同。此外，到达每棵子树时，从第 1 轮和第 2 轮的观测中得到的结果 0 的数量也相同，因此它们都在原问题的同一个子实例上执行。由此可知，期望摊销成本和期望效用保持不变。注意，对于 S'_j，第一轮中的元素现在处于应有的位置，从而满足 IO 的剪枝性质。

在 S'_j中第一轮测试之后，子实例上的策略 T'^1和 T'^0现在是针对这些子实例的伪 IO 策略的剪枝；该伪 IO 策略在一个包含 $M-1$ 个其他元素的列表开头，将 x_i 作为次序外元素。根据归纳假设，存在一个定义在该子实例上的 IO 随机剪枝 R^0 ，其期望值不低于 T'^0的$U_{\bar{k}_1-1}$，且额外的期望摊销成本至多为 1；还存在一个 IO 的随机剪枝 R^1，其期望值不低于 T'^1的$U_{\bar{k}_1}$，并且额外的期望摊销成本同样至多为 1。

令 R_j为通过将 S'_j中的 T'^0替换为 R^0，并将 T'^1替换为 R^1而得到的策略（使用 R^0 和 R^1中的分布的笛卡尔积）。因此， R_j 是 IO 剪枝上的分布，因为所有元素现在都已按正确顺序排列。

对于效用，我们有：

$$\begin{aligned}
\mathbb{E}[U_{\bar{k}}(R_j)] &= \bar{p}_{a_1} \cdot \mathbb{E}\big[U_{\bar{k}_1-1}\big(R^0\big)\big] + (1-\bar{p}_{a_1}) \cdot \mathbb{E}\big[U_{\bar{k}_1}\big(R^1\big)\big] \\
&\geq \bar{p}_{a_1} \cdot \mathbb{E}\big[U_{\bar{k}_1-1}\big(T'^0\big)\big] + (1-\bar{p}_{a_1}) \cdot \mathbb{E}\big[U_{\bar{k}_1}\big(T'^1\big)\big] \\
&= \bar{p}_{a_1} \cdot \big[\bar{p}_i \cdot \mathbb{E}\big[U_{\bar{k}_2-2}\big(T^{00}\big)\big] + (1-\bar{p}_i) \cdot \mathbb{E}\big[U_{\bar{k}_2-1}\big(T^{10}\big)\big]\big] \\
&\quad + (1-\bar{p}_{a_1}) \cdot \big[\bar{p}_i \cdot \mathbb{E}\big[U_{\bar{k}_2-1}\big(T^{01}\big)\big] + (1-\bar{p}_i) \cdot \mathbb{E}\big[U_{\bar{k}_2}\big(T^{11}\big)\big]\big] \\
&= \bar{p}_i \cdot \big[\bar{p}_{a_1} \cdot \mathbb{E}\big[U_{\bar{k}_2-2}\big(T^{00}\big)\big] + (1-\bar{p}_{a_1}) \cdot \mathbb{E}\big[U_{\bar{k}_2-1}\big(T^{01}\big)\big]\big] \\
&\quad + (1-\bar{p}_i) \cdot \big[\bar{p}_{a_1} \cdot \mathbb{E}\big[U_{\bar{k}_2-1}\big(T^{10}\big)\big] + (1-\bar{p}_{a_1}) \cdot \mathbb{E}\big[U_{\bar{k}_2}\big(T^{11}\big)\big]\big] \\
&= \mathbb{E}[U_{\bar{k}}(S_j)]
\end{aligned}$$

其中不等式中用到了归纳假设。等式由这些表达式的定义以及比较系数得出。对于期望摊销成本：

$$\begin{aligned}
\mathbb{E}[\mathrm{acost}(R_j)] &= \mathrm{acost}(b_1) + \mathbb{E}[\mathrm{acost}(a_1)] + \bar{p}_{a_1} \cdot \mathbb{E}\big[\mathrm{acost}\big(R^0\big)\big] + (1-\bar{p}_{a_1}) \cdot \mathbb{E}\big[\mathrm{acost}\big(R^1\big)\big] \\
&\leq \mathrm{acost}(b_1) + \mathbb{E}[\mathrm{acost}(a_1)] + \bar{p}_{a_1} \cdot \big[\mathbb{E}\big[\mathrm{acost}\big(T'^0\big)\big] + 1\big] + (1-\bar{p}_{a_1}) \cdot \big[\mathbb{E}\big[\mathrm{acost}\big(T'^1\big)\big] + 1\big] \\
&= \mathrm{acost}(b_1) + \mathrm{acost}(b_2) + \mathbb{E}[\mathrm{acost}(a_1) + \mathrm{acost}(i)] + 1
\end{aligned}$$

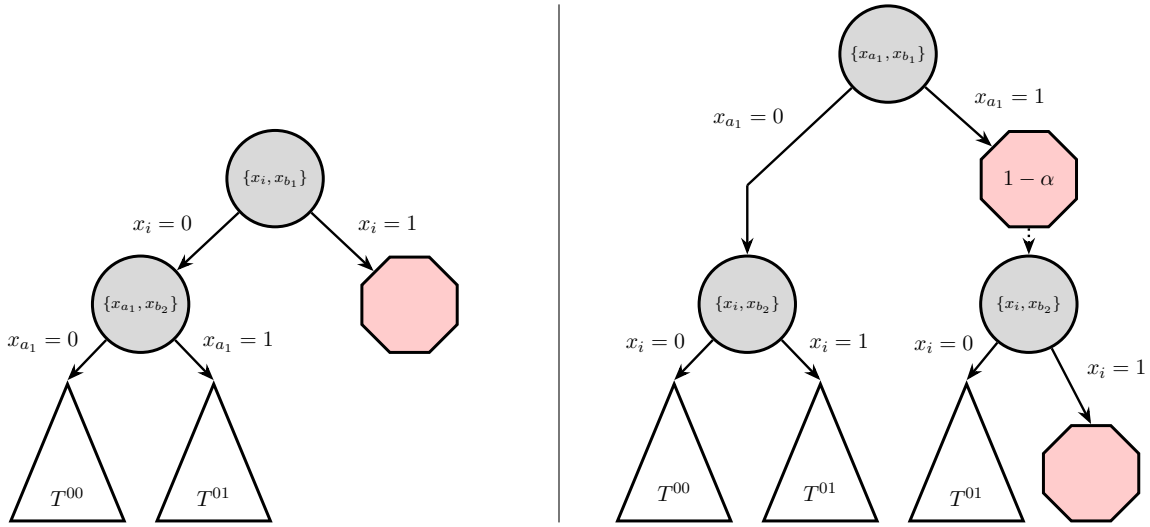

图5：情况 1.2（继续-中止）。左侧为 S_j；右侧为 S'_j，正文将其定义为在两种确定性策略（S''_j 和 S''_j）之间进行随机˜选择，并可用这种方式表示。八边形节点对应于在没有 0-证书的情况下停止（确定性地停止或以$1-\alpha$ 的概率停止，否则继续）。在这种情况下，$\alpha := \frac{\bar{p}_i - \bar{p}_{a_1}}{\bar{p}_i \cdot\ 1-\bar{p}_{a_1}}$（）。

$$
\begin{aligned}
&\quad + \bar{p}_{a_1} \cdot \big[\bar{p}_i \cdot \mathbb{E}\big[\mathrm{acost}\big(T^{00}\big)\big] + (1-\bar{p}_i) \cdot \mathbb{E}\big[\mathrm{acost}\big(T^{10}\big)\big]\big] \\
&\quad + (1-\bar{p}_{a_1}) \cdot \big[\bar{p}_i \cdot \mathbb{E}\big[\mathrm{acost}\big(T^{01}\big)\big] + (1-\bar{p}_i) \cdot \mathbb{E}\big[\mathrm{acost}\big(T^{11}\big)\big]\big] \\
&= \mathrm{acost}(b_1) + \mathrm{acost}(b_2) + \mathbb{E}[\mathrm{acost}(i) + \mathrm{acost}(a_1)] + 1 \\
&\quad + \bar{p}_i \cdot \big[\bar{p}_{a_1} \cdot \mathbb{E}\big[\mathrm{acost}\big(T^{00}\big)\big] + (1-\bar{p}_{a_1}) \cdot \mathbb{E}\big[\mathrm{acost}\big(T^{01}\big)\big]\big] \\
&\quad + (1-\bar{p}_i) \cdot \big[\bar{p}_{a_1} \cdot \mathbb{E}\big[\mathrm{acost}\big(T^{10}\big)\big] + (1-\bar{p}_{a_1}) \cdot \mathbb{E}\big[\mathrm{acost}\big(T^{11}\big)\big]\big] \\
&= \mathbb{E}[\mathrm{acost}(S_j)] + 1
\end{aligned}
$$

其中，不等式由归纳假设得出，第一、第二和最后一个等式由这些表达式的定义得出，第三个等式则由比较系数得出。对于 $\ell \in \{0,1\}$，论证和估算同样适用。

情况 1.2（继续-中止）：这是指在 $x_i = 0$ 时继续求值（包括 T^{00}和 T^{01}可能不测试任何元素），并在 $x_i =$¯1 时停止且不产生 0-证书；隐含地，这意味着 $k \geq 2$。第一个变换步骤是交换元素 x_i 和 x_{a_1}。然而，这会降低到达子树 T^{01} 的概率，从而降低期望效用，因为$\bar{p}_i \cdot (1-\bar{p}_{a_1}) \geq \bar{p}_{a_1} \cdot (1-\bar{p}_i)$，这是由于 $\bar{p}_i \geq \bar{p}_{a_1}$。为此进行补偿，我们定义两种确定性策略，二者都将 x_{a_i} 交换到第一轮，并在两者之间随机选择。第一种策略记为˜ S''_j，除交换元素 x_i和 x_{a_i}外，与 S_j完全相同。以 $1-\alpha$的概率选择该策略（其中 α 如下所示）。第二种策略记为 S''_j：在发现 $x_{a_1} = 1$后不停止，而是继续进行第二轮。在第二轮之后，如果 $x_i = 0$，则执行T^{01}（在相同的子实例上，且迄今为止发现的 0 的数量相同）；否则，如果 $x_i = 1$，则停止且不产生证书。以 α的概率选择策略 S''_j。所得的随机化策略˜S'_j由 S''_j和 S''_j组成，如图5右侧所示的单棵树所描绘。再次，x_{a_1} 现在处于其在 IO 中的位置。我们令 α 为 $\frac{\bar{p}_i - \bar{p}_{a_1}}{\bar{p}_i \cdot\ 1-\bar{p}_{a_1}}$（）$\in$（0, 1）。如下所示，这确保整体期望效用保持不变。

我们将 s　　第一轮之后的子策略，对于 $x_{a_1} = 0$在 S'_j中具有 T'^0（S''_j也相同　　d

$\tilde{S}''_j$），以及对于 $x_{a_1} = 1$，通过 T'^1 为 S''_j，并通过 $\tilde{T}'^1$ 为 $\tilde{S}''_j$。注意，T'^0、T'^1 和 $\tilde{T}'^1$ 分别是对应子实例上伪 IO 的剪枝；$\tilde{T}'^1$（不测试任何元素）显然已经是 IO 的剪枝。它们也是 RC-剪枝，因为 S_j 遵循轮次完成性质。根据归纳假设，我们可以将 T'^0 和 T'^1 分别替换为分布 R^0 和 R^1，它们是对应子实例上 IO 剪枝的分布，同时保持期望效用，并对任何额外的期望摊销成本加以限制。由此得到的策略 R_j 现在遵循 IO 所要求的顺序，并且是剪枝上的分布。

我们计算期望效用和期望摊销成本：

$$\begin{aligned}
\mathbb{E}[U_{\bar{k}}(R_j)] &= \bar{p}_{a_1} \cdot \mathbb{E}\big[U_{\bar{k}_1-1}\big(R^0\big)\big] + (1-\bar{p}_{a_1}) \cdot \big[\alpha \cdot \mathbb{E}\big[U_{\bar{k}_1}\big(R^1\big)\big] + (1-\alpha)\cdot 0\big] \\
&\geq \bar{p}_{a_1} \cdot \mathbb{E}\big[U_{\bar{k}_1-1}\big(T'^0\big)\big] + \frac{\bar{p}_i - \bar{p}_{a_1}}{\bar{p}_i} \cdot \mathbb{E}\big[U_{\bar{k}_1}\big(T'^1\big)\big] \\
&= \bar{p}_{a_1} \cdot \big[\bar{p}_i \cdot \mathbb{E}\big[U_{\bar{k}_2-2}\big(T^{00}\big)\big] + (1-\bar{p}_i)\cdot \mathbb{E}\big[U_{\bar{k}_2-1}\big(T^{01}\big)\big]\big] + \frac{\bar{p}_i - \bar{p}_{a_1}}{\bar{p}_i} \cdot \bar{p}_i \cdot \mathbb{E}\big[U_{\bar{k}_2-1}\big(T^{01}\big)\big] \\
&= \bar{p}_i \cdot \bar{p}_{a_1} \cdot \mathbb{E}\big[U_{\bar{k}_2-2}\big(T^{00}\big)\big] + \bar{p}_i \cdot (1-\bar{p}_{a_1}) \cdot \mathbb{E}\big[U_{\bar{k}_2-1}\big(T^{01}\big)\big] \\
&= \mathbb{E}[U_{\bar{k}}(S_j)]\,.
\end{aligned}$$

在该不等式中，我们利用了 $\tilde{T}'^1$ 不会对效用有所贡献这一点、归纳假设对其他情况给出的期望效用保证，并代入 α 的定义。第二个等式由子策略的定义得出。对于第三个等式，我们通过代数变形重新排列各项，使其符合 S_j 的结构。

摊销成本的估算更为复杂；根据 x_{b_1} 和 x_{b_2} 的实现，交换 x_i 和 x_{a_1} 可能会产生额外的摊销成本：

$$\begin{aligned}
\mathbb{E}[\mathrm{acost}(R_j)] &= \mathbb{E}[\mathrm{acost}(a_1)] + \mathrm{acost}(b_1) \\
&\quad + \bar{p}_{a_1} \cdot \mathbb{E}\big[\mathrm{acost}\big(R^0\big)\big] + (1-\bar{p}_{a_1}) \cdot \big[\alpha \cdot \mathbb{E}\big[\mathrm{acost}\big(R^1\big)\big] + (1-\alpha)\cdot 0\big] \\
&\leq \mathbb{E}[\mathrm{acost}(a_1)] + \mathrm{acost}(b_1) + \bar{p}_{a_1} \cdot \big(1 + \mathbb{E}\big[\mathrm{acost}\big(T'^0\big)\big]\big) + (1-\bar{p}_{a_1}) \cdot \alpha \cdot \big(1 + \mathbb{E}\big[\mathrm{acost}\big(T'^1\big)\big]\big) \\
&= \mathbb{E}[\mathrm{acost}(a_1)] + \mathrm{acost}(b_1) \\
&\quad + \bar{p}_{a_1} \cdot \big[\mathbb{E}[\mathrm{acost}(i)] + \mathrm{acost}(b_2) + \bar{p}_i \cdot \mathbb{E}\big[\mathrm{acost}\big(T^{00}\big)\big] + (1-\bar{p}_i)\cdot \mathbb{E}\big[\mathrm{acost}\big(T^{01}\big)\big]\big] \\
&\quad + (1-\bar{p}_{a_1}) \cdot \alpha \cdot \big[\mathbb{E}[\mathrm{acost}(i)] + \mathrm{acost}(b_2) + \bar{p}_i \cdot \mathbb{E}\big[\mathrm{acost}\big(T^{01}\big)\big]\big] \\
&\quad + 1 \cdot [\bar{p}_{a_1} + \alpha \cdot (1-\bar{p}_{a_1})] \\
&= \mathbb{E}[\mathrm{acost}(a_1)] + \mathrm{acost}(b_1) \\
&\quad + \bar{p}_i \cdot \bar{p}_{a_1} \cdot \mathbb{E}\big[\mathrm{acost}\big(T^{00}\big)\big] + [\bar{p}_{a_1} \cdot (1-\bar{p}_i) + (\bar{p}_i - \bar{p}_{a_1})] \cdot \mathbb{E}\big[\mathrm{acost}\big(T^{01}\big)\big] \\
&\quad + [\bar{p}_{a_1} + \alpha \cdot (1-\bar{p}_{a_1})] \cdot [\mathbb{E}[\mathrm{acost}(i)] + \mathrm{acost}(b_2)] + [\bar{p}_{a_1} + \alpha \cdot (1-\bar{p}_{a_1})] \\
&= \mathbb{E}[\mathrm{acost}(i)] + \mathrm{acost}(b_1) \\
&\quad + \bar{p}_i \cdot \big[\mathbb{E}[\mathrm{acost}(a_1)] + \mathrm{acost}(b_2) + \bar{p}_{a_1} \cdot \mathbb{E}\big[\mathrm{acost}\big(T^{00}\big)\big] + (1-\bar{p}_{a_1}) \cdot \mathbb{E}\big[\mathrm{acost}\big(T^{01}\big)\big]\big] \\
&\quad + (1-\bar{p}_i) \cdot \mathbb{E}[\mathrm{acost}(a_1)] - [1 - \bar{p}_{a_1} - \alpha \cdot (1-\bar{p}_{a_1})] \cdot \mathbb{E}[\mathrm{acost}(i)] \\
&\quad + [\bar{p}_{a_1} + \alpha \cdot (1-\bar{p}_{a_1}) - \bar{p}_i] \cdot \mathrm{acost}(b_2) + [\bar{p}_{a_1} + \alpha \cdot (1-\bar{p}_{a_1})] \\
&\leq \mathbb{E}[\mathrm{acost}(S_j)] + 1\,.
\end{aligned}$$

在第一个等式中，我们使用了策略定义，并注意到 $\tilde{T}'^1$ 不会产生任何摊销成本。第一个不等式使用归纳假设给出的成本上界，来替换伪 IO 剪枝。第二个等式代入 T'^0 和 T'^1 的定义并重新排列。第三个和第四个不等式进一步重新排列，并使用 α 的定义。在最后一步，我们取回 S_j 的期望摊销成本，并将剩余两项分别以上界 0 和 1 进行估计。

表达式中各项的取值分别为 0 和 1；相关计算推迟到 C.2节。与之前一样，结果也适用于 $\ell \in \{0,1\}$ 的情况。

至此，我们完成了对情况 1 的两个示例子情况的介绍与分析。情况 1 的其余子情况以及情况 2（$\bar{p}_i < \frac{1}{2}$）的分析留待C.1节。

在情形分析中对 σ 进行条件化后，我们现在说明，这些替换实际上可以以良好定义的方式合并为一个策略 R_j。由于所有提出的替换都是对 IO 的（随机化）RC 剪枝，因此要测试的元素选择相同。因此，良好定义的 R_j 可能出现的问题，只会在某个替换以非零概率在第二轮完成之前停止时发生——此后，x_{b_1} 和 x_{b_2} 的结果已知。所有情况下提出的替换都不会在第一轮之前停止。有些替换在第一轮之后会以非零概率停止，且停止行为可能取决于 x_{a_1} 的取值；然而，我们只会在原策略针对 x_i 的两个可能取值中至少一个也会在第一轮之后停止的情况下使用此类替换。回想一下，这些情况是根据前两轮中的停止行为区分的：对于 x_{b_1} 的固定取值，如果原策略在 $x_{b_2} = 0$ 时于第一轮之后停止，那么当 $x_{b_2} = 1$ 时，它也会在第一轮之后以相同方式停止。因此，对于 x_{b_1} 的固定取值，我们情形分析中的相关情况不受 x_{b_2} 取值的影响。进而，这意味着原策略将被替换为（随机化）策略，而这些策略对于 $x_{b_2} = 1$ 和 $x_{b_2} = 0$ 在第一轮之后具有相同的停止行为。因此，在这些元素尚未被查询之前，所提出的替换策略不会依赖于测试 x_{b_1} 和 x_{b_2} 的结果，从而我们得到一个良好定义的策略 R_j。

总而言之，这意味着归纳假设对于 $m = M$同样成立，从而完成归纳步骤，并因此证明引理 7。

□

7 结论

我们的工作提出了一个自然问题：对于划分拟阵上的 MBT，是否也能在测试 x_i 会产生相关成本 c_i 的情形下实现常数因子近似算法$_1$。事实上，对于为一般拟阵寻找 1-证书的问题，通过按照比率 p_i/c_i对测试进行排序，可以直接将 OPT 推广到这一情形。然而，如何以一种能够为 MP0 提供常数因子近似的方式推广 InsideOut策略，目前并不明确。此外，第6节 中的当前分析已经高度技术化，并且严重依赖单位成本假设。因此，要处理任意成本，算法设计及其分析似乎都需要新的思路。

在概念层面上，本文将 MBT 引入为一类新的 SBFE 问题，同时也引入了期望预算问题。我们希望，我们的结果和技术——尤其是对期望预算的考虑——能够为（近似地）解决一般拟阵上的 MBT，以及为其他 SBFE 问题获得算法铺平道路。

致谢。

我们感谢 Stefan Walzer 就相关 SBFE 问题中的困难实例进行的早期讨论，也感谢匿名审稿人提供的宝贵建议和反馈。Lisa Hellerstein 部分获得了 NSF奖项1909335 的资助。Kevin Schewior 部分获得了丹麦独立研究基金（自然科学领域）项目的资助，项目编号为 DFF-0135-00018B 和 DFF-4283-00079B。

参考文献

[1] Agrawal, S., Ding, Y., Saberi, A., and Ye, Y. 相关性鲁棒随机优化。发表于离散算法研讨会（SODA）（2010），第 1087–1096 页。.[2] Allen, S. R., Hellerstein, L., Kletenik, D., and Unlüyurt, T. DNF 公式的求值。CoRR abs/1310.3673（2013）。.[3] Allen, S. R., Hellerstein, L., Kletenik, D., and Unlüyurt, T. 单调DNF公式的求值。Algorithmica 77, 3（2017），661–685。[4] Bach, E., Dusart, J., Hellerstein, L., and Kletenik, D. 布尔函数的子模目标值。离散应用数学 238（2018），1–13。[5] Ben-Dov, Y. 相干系统特殊结构的最优测试程序。管理科学 27，12（1981），1410–1420。.[6] Boros, E., and Unlüyurt, T. 双正则系统诊断。数学与人工智能年刊 26，1-4（1999年9月），171–191。.[7] Boros, E., and Unlüyurt, T. 小深度串并联系统的序贯测试。载于《建模、优化与仿真计算工具：计算机科学与运筹学交叉领域》，M. Laguna 和 J. L. G. Velarde 编。Springer，2000，第 39–73 页。[8] Charikar, M., Fagin, R., Guruswami, V., Kleinberg, J., Raghavan, P., and Sahai, A.定价信息的查询策略。发表于 ACM计算理论研讨会（STOC）（2000），第 582–591 页。[9] Correa, J. R., Foncea, P., Hoeksma, R., Oosterwijk, T., and Vredeveld, T. 先知不等式的最新进展。SIGecom Exchanges 17，1（2018），61–70。[10] Cox, L., Qiu, Y., and Kuehner, W. 不确定自变量值下离散分类函数的启发式最低成本计算。运筹学年刊 21（1989），1–29。[11] Cui, Y., and Nagarajan, V. 最小成本自适应子模覆盖。发表于算法简洁性研讨会（SOSA）（2023），第 12–27 页。[12] Deshpande, A., Hellerstein, L., and Kletenik, D. 随机次模集合覆盖的近似算法及其在布尔函数求值和最小背包问题中的应用。ACM算法汇刊 12，3（2016），1–28。[13] Dumitriu, I., Tetali, P., and Winkler, P. 关于用两个球打高尔夫。SIAM离散数学杂志 16，4（2003），604–615。[14] Ene, A., Nagarajan, V., and Saket, R. 随机 k-TSP 的近似算法。发表于软件技术基础与理论计算机科学会议（FSTTCS）（2017），第 27:27–27:14 页。[15] Fu, H., Li, J., and Liu, D. 具有非强制检查的潘多拉盒子问题：困难性与近似方案。发表于 ACM计算理论研讨会（STOC）（2023），第 789–802 页。[16] Fu, L., Fu, X., Xu, Z., Peng, Q., Wang, X., and Lu, S. 不确定网络中的源点—目的地点连通性判定：建模与求解。IEEE/ACM Transactions on Networking 25，6（2017），3237–3252。

[17] Ghuge, R., Gupta, A., and Nagarajan, V. 非自适应随机评分分类与可解释半空间求值。载于国际整数规划与组合优化会议（IPCO）（2022），第277–290页。[18] Gittins, J. 多臂老虎机分配指数。约翰·威立，1989年。[19] Gkenosis, D., Grammel, N., Hellerstein, L., and Kletenik, D. 随机评分分类问题。载于欧洲算法研讨会（ESA）（2018），第36:1–36:14页。[20] Gkenosis, D., Grammel, N., Hellerstein, L., and Kletenik, D. 对称布尔函数的随机布尔函数求值问题。《离散应用数学》309（2022），269–277。[21] Golumbic, M., and Gurvich, V. 读一次函数。载于《布尔函数理论、算法与应用》，Y. Crama 和 P. Hammer 编。剑桥大学出版社，2011年。[22] Grammel, N., Hellerstein, L., Kletenik, D., and Liu, N. 单次成本随机评分分类问题的算法。Algorithmica 84，10（2022），3054–3074。[23] Greiner, R., Hayward, R., Jankowska, M., and Molloy, M. 寻找与或树的最优满意化策略。《人工智能》170，1（2006），19–58。[24] Guo, M., Li, J., Neumann, A., Neumann, F., and Nguyen, H. X. 有限查询图连通性测试。载于AAAI人工智能会议（AAAI）（2024），第20718–20725页。[25] Gupta, A., Jiang, H., Scully, Z., and Singla, S. 马尔可夫信息价格。载于国际整数规划与组合优化会议（IPCO）（2019），第233–246页。[26] Halpern, J. 技术说明：k取n系统的故障测试。《运筹学》22，6（1974），1267–1271。[27] Hellerstein, L., Kletenik, D., and Parthasarathy, S. 随机次模覆盖的紧确界。《人工智能研究杂志》71（2021），347–370。[28] Hellerstein, L., Liu, N., and Schewior, K. 快速判定选举获胜者。载于第15届理论计算机科学创新会议论文集（ITCS）（2024），第61:1–61:14页。[29] Im, S., Nagarajan, V., and van der Zwaan, R. 最小延迟次模覆盖。《ACM算法汇刊》13，1（2016），13:1–13:28。[30] Jiang, H., Li, J., Liu, D., and Singla, S. 随机k-TSP的算法与适应性差距。载于理论计算机科学创新会议（ITCS）（2020），第45:1–45:25页。[31] Kaplan, H., Kushilevitz, E., and Mansour, Y. 带属性成本的学习。载于ACM计算理论研讨会（STOC）（2005），第356–365页。[32] Liu, N. 随机评分分类问题的两种6-近似算法。CoRR abs/2212.02370（2022）。[33] Lucier, B. 先知不等式的经济学视角。SIGecom Exchanges 16，1（2017），24–47。

[34] Megow, N., and Schlöter, J. 通过覆盖技术，在可探索的随机不确定性下进行集合选择。载于国际整数规划与组合优化会议（IPCO）（2023），第319–333页。[35] Nielsen, M. A., Rohwedder, L., and Schewior, K. 对 k-取n函数的非自适应求值：紧确界差距与单位成本多项式时间近似方案。载于组合优化问题近似算法国际会议（APPROX）（2025），第26:1–26:18页。[36] Oxley, J. 拟阵理论。牛津大学出版社，2011年。[37] Plank, B. M., and Schewior, K. 具有较小近似比的随机评分分类简易算法。SIAM离散数学杂志38，3（2024），2069–2088。[38] Salloum, S., and Breuer, M. 若干对称相干系统的最优测试算法。《数学分析与应用杂志》101，1（1984），170–194。[39] Salloum, S. N., and Breuer, M. A. k取n:g系统的快速最优诊断程序。IEEE可靠性汇刊46（1997），283–290。[40] Scully, Z., and Terenin, A. 吉廷斯指数：不确定性下决策的设计原则。《运筹学教程》（2025），28–70。[41] Singla, S. 组合优化中的信息价格。载于ACM-SIAM离散算法研讨会（SODA）（2018），第2523–2532页。..[42] Unlüyurt, T. 复杂系统的序贯测试：综述。《离散应用数学》142，1-3（2004），189–205。.[43] Unlüyurt, T. 序贯测试问题：后续综述。《离散应用数学》377（2025），356–369。[44] Weitzman, M. L. 最优搜索最佳替代方案。《计量经济学》47，3（1979），641–654。

来自缺失材料第1节

A.1 均匀拟阵上 MP0 最优评估策略剪枝的反例

引理8. 令 $U(A)$ 表示一个随机变量，用于指示策略 A 是否找到了一个0-证书。对于任意正整数 M，存在一个 k-*of*-n 函数的实例以及一个策略R，具有以下性质：

- $\mathbb{E}[U(S')] \le \frac{1}{M}\cdot\mathbb{E}[U(R)]$ 对于任意对 S 进行的随机剪枝策略 S'，若其 $\mathbb{E}[\text{成本}(S')] \le \mathbb{E}[\text{小于成本}(R)]$
- $\mathbb{E}[\text{成本}(S'')] \ge M\cdot\mathbb{E}[\text{成本}(R)]$ 对于任意对 S 进行的随机剪枝策略 S''，若其 $\mathbb{E}[U(S'')] \ge \mathbb{E}[U\text{小于}(R)]$

其中 S 是最优评估策略。

证明。考虑以下实例，其中有 $n = 1+\frac{1}{\varepsilon}$ 和 $k = n-1$；元素 1 的概率为 $p_1 = 1-\varepsilon$，其余 $n-1$ 个元素激活的概率为 $p_2 = \ldots = p_n = 1-\varepsilon^2$。稍后我们将适当地设定 ε，使其成为 M 的函数。

令 R 为以下（剪枝后的）策略：首先测试 x_1。如果 $x_1 = 0$，则继续进行完整评估（即，直到找到第二个未激活变量，或直到查询完所有剩余元素为止）。如果 $x_1 = 1$，则停止。执行 R 的期望成本由 x_1 带来的 1，以及其余每个元素至多带来的 ε 组成，因为检查这些元素的概率至多为 ε。因此，$\mathbb{E}[\text{cost}(R)] \le 1+(n-1)\cdot\varepsilon = 2$。$R$ 的策略树有 $n-1=\frac{1}{\varepsilon}$ 个以 0-证书结束的叶节点。到达这些叶节点中每个节点的概率至少为 $\varepsilon\cdot\left(1-\varepsilon^2\right)^{n-2}\cdot\varepsilon^2 \ge \varepsilon\cdot\left(1-(n-1)\varepsilon^2\right)\cdot\varepsilon^2 = \varepsilon^3\cdot(1-\varepsilon)$，因此 $\mathbb{E}[U(R)] \ge \varepsilon^2\cdot(1-\varepsilon)$。

S 最优评估策略执行如下操作：首先测试 x_2。如果 $x_2 = 1$，则继续测试 x_3，依此类推，直到遇到一个未激活变量，或直到元素 x_2 至 x_n 已被查询（此时 S 在没有找到 0-证书的情况下停止）；遇到第一个未激活元素后，继续测试 x_1，若 $x_1 = 1$，则查询所有剩余元素，直到找到第二个未激活变量，或所有元素都已测试。

设 $\tilde{S}$ 是 S 的一个剪枝。设 $m \in \{1,\ldots,n-1\}$ 是这样一个活跃元素的数量：如果 S 在停止且未找到证书之前还没有发现任何未激活变量，它会查询这些元素。我们可以宽松地忽略找到第一个未激活元素之后产生的任何成本，从而得到 $\mathbb{E}\left[\text{成本}\left(\tilde{S}\right)\right]$ 的下界 $m\cdot\left(1-\varepsilon^2\right)^{m-1} \ge m\cdot\left(1-m\varepsilon^2\right)$，因为树中只要到达这些元素，每个元素都会因测试而产生 1 的成本。以选中的第一个元素（x_2）为未激活为条件，有一个以 0-证书结尾的叶节点，其到达概率为 ε；另有 $n-2$ 个以 0-证书结尾的叶节点，每个叶节点的到达概率至多为 ε^2。根据并集界，我们所能期望的找到 0-证书的最高概率至多为 $\varepsilon+(n-2)\cdot\varepsilon^2 \le 2\varepsilon$；若以第 i 个被选中的元素是第一个未激活元素为条件，该值只会更低。根据全期望定律，并将第 i 个元素是第一个未激活元素的概率上界设为 $1\cdot\varepsilon^2$，可得 $\mathbb{E}\left[U\left(\tilde{S}\right)\right] \le m\cdot\varepsilon^2\cdot 2\varepsilon = 2\cdot m\cdot\varepsilon^3$。总体而言，我们有：

$$\frac{\mathbb{E}\left[\text{cost}\left(\tilde{S}\right)\right]}{\mathbb{E}\left[U\left(\tilde{S}\right)\right]} \ge \frac{m\cdot\left(1-m\varepsilon^2\right)}{2\cdot m\cdot\varepsilon^3} = \frac{1}{\varepsilon^3}\cdot\frac{1-m\varepsilon^2}{2} \ge \frac{1}{\varepsilon^3}\cdot\frac{1-\varepsilon}{2} \tag{22}$$

根据期望的线性性，该比率的这一界也适用于 S 的任意剪枝上的分布。

由 公式（22），随机剪枝 S' 的 S 与 $\mathbb{E}[\text{成本}(S')] \le \mathbb{E}[\text{成本}(R)] \le 2$ 找到 0-证书的概率至多为 $\mathbb{E}[U(S')] \le \varepsilon^3\frac{4}{1-\varepsilon}$；同时，$\mathbb{E}[U(R)] \ge \varepsilon^2\cdot(1-\varepsilon)$，

因此 $\frac{\mathbb{E}[U(R)]}{\mathbb{E}[U(S')]} \geq \frac{1}{\varepsilon} \cdot (1-\varepsilon)^2$。取 $\varepsilon \leq \frac{1}{8M}$，可得 $\mathbb{E}[U(S')] \leq \frac{1}{M} \cdot \mathbb{E}[U(R)]$, 从而证明了第一个引理陈述。公式（22）还意味着，任何 S'' 在 S 的剪枝上的$\mathbb{E}[U(S'')] \geq \mathbb{E}[U(R)] \geq \varepsilon^2 \cdot (1-\varepsilon)$分布都会产生成本 $\mathbb{E}[\text{成本}(S'')] \geq \frac{1}{\varepsilon} \cdot (1-\varepsilon)^2 \geq 2M \geq M \cdot \mathbb{E}[\text{成本}(R)]$对于 $\varepsilon \leq \frac{1}{8M}$，从而得到第二个引理陈述。 □

A.2 通过 Deshpande 等人的子模目标值方法可实现的近似因子

考虑某类布尔函数 f对应的 SBFE 问题。Desh- pande 等人用于为该问题构造优良近似算法的方法, 基于构造一个目标函数 g，该函数与正在评估的函数 f 相关联 [12]。目标函数 g 是（依赖于状态的）单调次模函数。它为通过测试获得的信息赋予整数效用值，使得当所获信息足以确定 Q 的值时，恰好达到一个固定效用值 g（称为 f的目标值）。这样，关于 f 的 SBFE 问题就可以归约为关于效用函数 g的随机子模覆盖问题。可以为 g 构造许多不同的目标函数 f，它们具有不同的性质, 包括不同的 Q取值。

Deshpande 等人使用两种贪心算法之一来求解由此得到的随机子模覆盖问题 g 。第一种是自适应贪心，它是一种针对随机子模覆盖问题的 $O(\log Q)$-近似算法 [12, 27, 11], ，并可为相关的 SBFE 问题提供 $O(\log Q)$-近似。另一种方法是使用双重自适应贪心；Deshpande 等人证明，该算法的近似界取决于 g的其他性质。这个界的定义有些技术性，因此我们仅将其称为 α。

因此，Deshpande 等人针对 SBFE 问题获得良好近似界的方法如下：尝试设计 g，使得 $O(\log Q)$ 或 α 较小，然后以 g为依据，使用自适应贪心或双重自适应贪心作为评估f的近似算法。通过这种方法，当 f是由大小关于 n为多项式的决策树表示的布尔函数时（通过构造目标值为 Q 且关于 n为多项式的 g），他们获得了一个 $O(\log n)$ 近似算法；当 f 是布尔线性阈值函数时（通过构造具有常数大小 α的 g），他们获得了一个常数因子近似算法。

如果 f 是一个取值依赖于其所有 n 个输入变量的布尔函数，那么针对 f 的任意目标函数 g 的目标值 Q至少为 n [4]。由此可知，上述使用自适应贪心及其 $O(\log Q)$ 界的方法，无法帮助 SBFE 问题获得常数因子近似。事实上，Deshpande 等人针对线性函数 f以及函数 f 都证明了以下结论：后者是若干互不相交的 k_i-of-n_i 函数的析取，其中 $k_i = n_i$ （即读一次析取范式公式）。对于这两类函数，针对 f 的任意目标函数 g 的目标值 Q 都必须关于 n呈指数增长，因此 $\log Q$ 关于 n呈线性增长。由于后一类函数的 SBFE 问题实际上等价于分划拟阵上的 MBT，因此无法基于 $O(\log Q)$ 界，为这些函数 f实现任何次线性近似。

这仍然留下了一种可能性：对于 f 一个 k_i元n_i 函数的析取（或合取），g 或许可以构造得使 α 为常数，就像 Deshpande 等人为线性阈值函数所做的那样。不过，直接根据 Deshpande 等人的结果也可得出，对于任意布尔函数f，当与测试相关的概率 p_j 都等于 $1/2$时，α 给出了 DGREEDY/(2·CERT) 的上界。这里，DGREEDY 是采用双重自适应贪心（相对于 f的任意目标函数 g）作为评估 f的策略时的期望成本；CERT 则是认证 $f(x)$的最小期望成本（即，在随机 x上评估 f 的最小离线成本的期望），其中所有p_j 都等于 $1/2$。[8] 在 [2], 中已证明（定理 6），对于任意

[8]同样，$\log Q$ 以上 当 p_j 全部等于 $_1{}_2$ 时，AGREEDY/(2 · CERT) 的界，其中 AGREEDY 是 th e

常数 β，其中 $0 < \beta < 1$，存在一个函数 f，它是 k_i-of-n_i 个函数的析取，且满足$k_i = n_i$，使得 OPT/CERT 的比值为 $\Omega(n^\beta/\log n)$。这里，OPT 是评估 f 的任意自适应策略的最小期望成本。由于双重自适应贪心就是这样一种策略，因此对于每个 $0 < \beta < 1$，都有 $\alpha = \Omega(n^\beta)$。这排除了通过双重自适应贪心及其 α 界，为分划拟阵上的 MBT 问题实现任何 $o(n^\beta)$ 近似界的可能性。

B 第 2.1 节缺失的材料

引理 1 的证明。考虑初等测试策略 S，以及二叉决策树中对应于 S 的两个节点 u 和 v；到达这两个节点时，活跃元素和非活跃元素的数量相同。尽管到达 u 和 v 的元素选择和/或顺序可能不同，且在 u 和 v 处产生的子实例也可能不同，但分别以 u 和 v 为根的子策略 $S(u)$ 和 $S(v)$ 是等价的，因为 S 是初等的。特别是，在 u 和 v 处，子树中的测试集合完全相同。$S(v)$ 是 u 处子实例的一个定义良好的子策略，而 $S(u)$ 则是 v 处子实例的一个定义良好的子策略；由于 f 是对称函数，如果在 v 处运行任一子策略会返回一个证书，当且仅当在 u 处运行它会返回一个证书。因此，$\mathbb{E}[U(S, u) \mid S \text{ 到达 } u] = \mathbb{E}[U(S, v) \mid S \text{ 到达 } v]$。此外，对于任何能够同时到达 u 和 v 的 S 的剪枝 S，以 u 和 v 为根的子策略 $S(u)$ 和 $S(v)$ 都可以视为对 S 的同一子策略进行剪枝所得。它们的区别仅在于停止方式，而不在于元素的选择或顺序，因为它们是通过对初等策略进行剪枝得到的。

现在，考虑与随机化策略 R 对应的分布的支撑集中 R_j 从 S 得到的（有限集合）剪枝，其中 $j \in J$ 适用于相应的集合 J，并考虑以下过程：固定 $j \in J$，并固定元组 $\tau = (s, t)$，其中 $s, t \in \{0, \dots, n\}$，$s + t \le n$。如果在恰有 s 个活跃元素和 t 个非活跃元素的任一节点处，R_j 都继续采用相同的测试策略，我们就称它对于元组 τ 是初等的。

如果 R_j 对于 τ 尚非初等，则执行以下操作。令 $V = V(R_j, \tau)$ 表示与 R_j 对应的二叉决策树中（可能为空的）节点集合，其中恰好观察到 s 个活跃元素和 t 个非活跃元素。对于 $v \in V$，令 $R_j(v)$ 表示 R_j 在测试了 v 处的变量后所采用的（确定性）测试策略。令 $\Pr[R_j, v] > 0$ 表示到达 v 的概率，并设 $\Pr[R_j, V] := \sum_{v \in V} \Pr[R_j, v]$。对于每个 v，令 R'^v_j 为从 R_j 得到的测试策略，即在 V 中的每个节点处继续采用 $R_j(v)$。注意，这一定义良好，只会改变以 V 中节点为根的子树中的决策树；特别地，对于任意 $u \in V$，$\Pr[R_j, u] = \Pr\left[R'^v_j, u\right]$。$R'^v_j$ 是 S 的一种剪枝，并且根据构造，它对于 τ 是初等的。

令 R'_j 为剪枝上的分布 $\{R'^v_j \mid v \in V\}$，它以概率 $\frac{\Pr[R_j, v]}{\Pr[R_j, V]}$ 选择 R'^v_j。此变换保持期望效用和成本不变：

$$\begin{aligned}\mathbb{E}\left[U\left(R'_j\right)\right] &= \sum_{v \in V} \frac{\Pr[R_j, v]}{\Pr[R_j, V]} \cdot \left[\tilde{U} + \Pr[R_j, V] \cdot \mathbb{E}[U(R_j, v) \mid R_j \text{ reaches } v]\right] \\ &= \tilde{U} + \sum_{v \in V} \Pr[R_j, v] \cdot \mathbb{E}[U(R_j, v) \mid R_j \text{ reaches } v] = \mathbb{E}[U(R_j)] \ ,\end{aligned}$$

对于适当的值 $\tilde{U}$。类似地，我们得到 $\mathbb{E}\left[\text{成本}\left(R'_j\right)\right] = \mathbb{E}[\text{成本}(R_j)]$。

R'_j 是剪枝上的分布，其中每个剪枝对于 τ 都是初等的。在 R 中，对每个 $j \in J$，将 R_j 替换为 R'_j，我们得到 R'，它是一个以剪枝上的分布为元素的分布，因此其本身也是一个对于 τ 初等的剪枝上的分布。

此操作（对于元组 τ）不会破坏其他元组保持初等性：令 $\preceq$ 表示由元组上的整数比较诱导的偏序。令 R_j 表示剪枝上的分布

自适应贪心的期望成本。

对于某个其他元组 τ' 而言是初等的。如果 τ 和 τ' 关于 $\preceq$ 不可比较，则 $V(R_j, \tau')$ 处的任何子树都不会发生变化，因为它们与 $V(R_j, \tau)$ 处的子树不相交。

如果 $\tau' \prec \tau$，且 R_j 中针对 τ' 的剪枝是初等的，那么 $V(R_j, \tau')$ 处的子树完全相同；$R_j'^{v}$ 的构造会在这些子树中执行相同的替换。因此，$V(R_j'^{v}, \tau')$ 处的子树仍然完全相同，所以 $R_j'^{v}$ 对于 τ' 仍然是初等的（因为 $\tau' \prec \tau$，在 R_j 中被剪枝的、与 τ' 对应的节点，在 $R_j'^{v}$ 中仍被剪枝）。

否则，如果 $\tau \prec \tau'$，那么 $V(R_j, \tau')$ 中的每个节点都属于一棵以 $V(R_j, \tau)$ 为根节点的子树。在 $R_j'^{v}$ 中，$V(R_j, \tau)$ 中的每棵此类子树都被替换为 $R_j(v)$。由于 R_j 对 τ' 是初等的，$R_j(v)$ 中与 τ' 对应的节点处的所有子策略都完全相同；因此，替换后，$V(R_j'^{v}, \tau')$ 处的任何子策略仍与该子策略完全相同，包括位于 $V(R_j'^{v}, \tau')$ 中但不在 $V(R_j, \tau')$ 中的某个节点处的任何子策略；该子策略此前曾在路径中的某处被剪枝（但现在以非零概率到达）。

因此，对每个可能的元组迭代应用上述操作，在保持期望成本和效用不变的情况下，得到的随机化策略 R' 是 S 的剪枝上的一个分布，其中每个剪枝对任意元组都是初等的；因此，该策略本身就是 S 的一个初等随机剪枝。

□

在引理 2的证明中，我们将使用以下显然事实。

事实 1。对于所有策略 S 和预算 B，*ComputePruning-Function(S)* 输出的函数在 B 处的取值恰好为 $\mathbb{E}[U(\textit{ComputePrunedStrategy}(S, B))]$。

现在我们给出该引理的证明。

引理 *2*的证明。关于 (i)，根据定义，q_S 显然是单调的。凹性和分段˜线性可由以下观察得出：每个随机剪枝 S 的 S 都可以视为一个分布，该分布定义在 S的有限个确定性剪枝所构成的策略集合上。根据˜˜成本和效用的期望线性性，$\mathbb{E}[$成本$(S)]$ 和 $\mathbb{E}[U(S)]$ 分别是 S的确定性剪枝对应取值的凸组合。

关于凹性，更正式地说，反设 q_S 不是凹函数，即存在B_ℓ、$B_r \in \mathbb{R}_{\geq 0}$和 $\alpha \in (0, 1)$，使得 $q_S((1-\alpha)B_\ell + \alpha B_r) < (1-\alpha)\cdot q_S(B_\ell) + \alpha \cdot q_S(B_r)$。令 $S_\ell, S_r \in$随机剪枝集合(S)，且 $\mathbb{E}[$成本$(S_\ell)] \leq B_\ell$, $\mathbb{E}[$成本$(S_r)] \leq B_r$, $\mathbb{E}[U(S_\ell)] = q_S(B_\ell)$，并且 $\mathbb{E}[U(S_r)] = q_S(B_r)$；再令 $S^* \in$ 随机剪枝集合(S)为如下策略：以概率$1-\alpha$选择 S_ℓ，以概率 α选择 S_r。根据期望的线性性，$\mathbb{E}[U(S^*)] = (1-\alpha)\cdot q_S(B_\ell) + \alpha \cdot q_S(B_r)$，且 $\mathbb{E}[$成本$(S^*)] \leq (1-\alpha)B_\ell + \alpha B_r$，因此 $q_S((1-\alpha)B_\ell + \alpha B_r) \geq (1-\alpha)\cdot q_S(B_\ell) + \alpha \cdot q_S(B_r)$，这与 q_S 不是凹函数的假设矛盾。

我们针对 S的二叉决策树表示证明 (ii)；结合引理 1，将有向无环图的剪枝视为相应策略树的剪枝，即可将证明推广到有向无环图表示。

通过对 S的高度进行归纳法来证明。若 S 为空，则基本情况显然成立。对于归纳步骤，令 k 为 S 的高度，并固定一个预算 B。根据事实 1，只需证明 ComputePrunedStrategy(S,B) 能计算出一个策略 $S' \in \text{randPrunings}(S)$，使 $\mathbb{E}[U(S')]$最大化，同时满足 $\mathbb{E}[\text{cost}(S')] \leq B$。

为证明后一项结论成立，考虑一个策略 $S^* \in$ randPrunings(S)，它在约束 $\mathbb{E}[$cost$(S^*)] \leq B$ 下确实能最大化$\mathbb{E}[U(S^*)]$。我们将把 S^* 转化为由 ComputePrunedStrategy(S, B) 计算出的策略，同时不降低效用，也不增加成本。为此，设 p 为 S^*wo du 立即停止的概率，x_i 为 S^*测试的第一个变量，S_0^* 和 S_1^* 分别为 S^*的左子树和右子树。类似地，设 S_0和 S_1 分别为 S的左子树和右子树。

首先，在 S^* 中将 S_j^* 替换为 ComputePrunedStrategy(S_j, $\mathbb{E}$[cost(S_j^*)])，其中 $j \in \{0,1\}$。根据归纳法，这会使 S^* 的期望成本和效用都保持不变。再根据归纳法以及S^*的最优性，选择量 p, $y_0 := \mathbb{E}[\text{cost}(S_0^*)]$, 和 $y_1 := \mathbb{E}[\text{cost}(S_1^*)]$，使其效用

$$(1-p)\cdot((1-p_i)\cdot q_{S_0}(y_0)+p_i\cdot q_{S_1}(y_1))$$

在期望成本满足约束的条件下最大化

$$(1-p)\cdot(1+(1-p_i)\cdot y_0+p_i\cdot y_1)=B.$$

由于 q_{S_0} 和 q_{S_1} 是单调凹函数，因此不难看出，可以通过以下过程计算最优取值。从 a 开始: $:= \mathbb{E}[\text{cost}(S_0)]$、$b := \mathbb{E}[\text{cost}(S_1)]$, 和$p$: $= 1$，分两个阶段持续减小变量，直至满足预算约束。具体而言，在阶段 1 中，减小一个满足 y_z 的变量，使右导数$\dot{q}^-_{S_z}(y_z)$最小；若出现并列，则任意选择，但只在断点处切换。持续执行此操作，直至上述量达到目标值与预算约束左侧之比，即，

$$\frac{(1-p_i)\cdot q_{S_0}(y_0)+p_i\cdot q_{S_1}(y_1)}{1+(1-p_i)\cdot y_0+p_i\cdot y_1}\le \min\{\dot{q}^-_{S_0}(y_0),\dot{q}^-_{S_1}(y_1)\}. \tag{23}$$

（注意，该不等式在开始时也可能已经满足。）只要该不等式仍以等式形式成立，就可以继续或结束阶段 1，选择方式仍然任意（但只在断点处改变选择）。如果继续阶段 1 会使该不等式不再以等式形式成立，就必须结束阶段 1。在阶段 2 中，只需减小 p，直至满足预算约束。不难看出，此过程与过程 ComputePrunedStrategy(S, B) 之间存在一一对应关系：第一阶段对应于在根节点以下进行剪枝，第二阶段对应于在根节点处进行剪枝。因此，ComputePrunedStrategy(S,B) 也是上述优化问题的一个最优解。

最后，注意到这两个算法显然都能在多项式时间内运行，因为它们主循环的迭代次数至多为 $|V(S)|$，这意味着 (iii) 成立。 □

C 缺失材料自 第 6 节

C.1 引理 7 的证明续篇 引理 7

情况 1 的续篇： $\bar{p}_i \ge \frac{1}{2}$ –

回顾一下，在情况 1 中，$\bar{p}_i \ge \frac{1}{2}$ 因此 $\bar{p}_{a_1} \le \bar{p}_i$，$\mathbb{E}[\text{acost}(a_1)] = \bar{p}_{a_1}$，且 $\mathbb{E}[\text{acost}(i)] = \bar{p}_i$。

情况 1.3（证书-继续）：这是指在考虑了第 1 轮测试的 x_{b_1} 的结果后，S_j 还需要再得到一个 0 才能获得 0-证书的情况。令 S'_j 为交换元素 x_i 和 x_{a_1} 位置后得到的策略，如图 6 所示。由于 x_i 或 x_{a_1} 中任一者为 0 就足以找到证书，因此除两者结果均为 1 的情况外，我们都会成功终止；在这种情况下，则执行任意一个（可能为空的）子策略 T^{11}。这些事件在 S_j 和 S'_j 中发生的概率相同，但由于对 x_{b_2} 的测试现在比以前更可能执行，S'_j 会产生一些额外的期望摊销成本。令 T'^1 为 S'_j 在情况 $x_{a_1} = 1$ 下的子策略。T'^1 是一个列表更短的伪 IO 剪枝，因此根据归纳假设，我们可以将其替换为子实例上 IO 剪枝的分布 R^1，且在期望意义下效用不会降低，额外（摊销）成本也有界。将 S'_j 中的 T'^1 替换为 R^1 后得到的策略 R_j，是 IO 剪枝上的一个分布

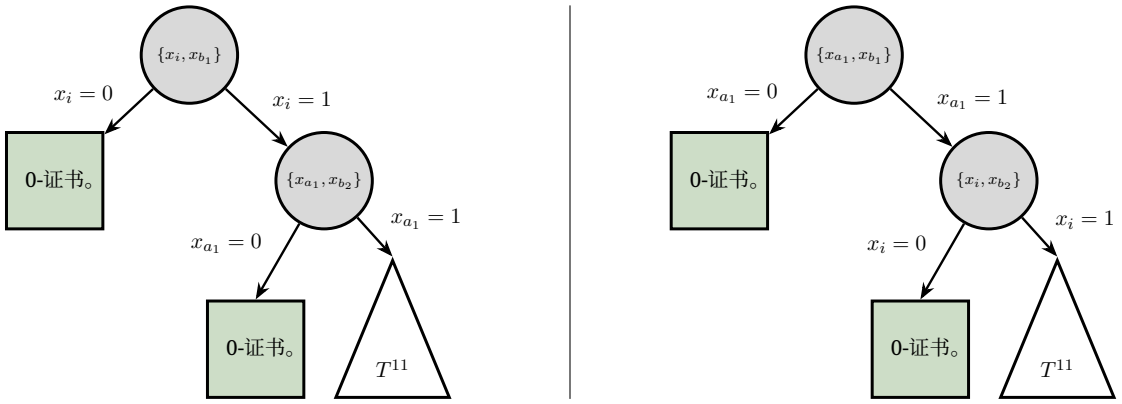

图 6：情况 1.3（证书-继续）。 S_j 在左侧， S'_j 在右侧。

因为现在所有元素的顺序都恰当。根据归纳假设，很容易推出 $\mathbb{E}[U_{\bar{k}}(R_j)] \geq \mathbb{E}[U_{\bar{k}}(S_j)]$。期望摊销成本的变化如下：

$$
\begin{aligned}
\mathbb{E}[\mathrm{acost}(R_j)] &= \mathbb{E}[\mathrm{acost}(a_1)] + \mathrm{acost}(b_1) + (1-\bar{p}_{a_1}) \cdot \mathbb{E}\big[\mathrm{acost}\big(R^1\big)\big] \\
&\leq \mathbb{E}[\mathrm{acost}(a_1)] + \mathrm{acost}(b_1) + (1-\bar{p}_{a_1}) \cdot \big[1 + \mathbb{E}\big[\mathrm{acost}\big(T'^1\big)\big]\big] \\
&= \mathbb{E}[\mathrm{acost}(a_1)] + \mathrm{acost}(b_1) + (1-\bar{p}_{a_1}) \cdot \big[\mathbb{E}[\mathrm{acost}(i)] + \mathrm{acost}(b_2) + (1-\bar{p}_i) \cdot \mathbb{E}\big[\mathrm{acost}\big(T^{11}\big)\big]\big] \\
&\quad + (1-\bar{p}_{a_1}) \cdot 1 \\
&= \mathbb{E}[\mathrm{acost}(i)] + \mathrm{acost}(b_1) + (1-\bar{p}_i) \cdot \big[\mathbb{E}[\mathrm{acost}(a_1)] + \mathrm{acost}(b_2) + (1-\bar{p}_{a_1}) \cdot \mathbb{E}\big[\mathrm{acost}\big(T^{11}\big)\big]\big] \\
&\quad - \bar{p}_{a_1} \cdot \mathbb{E}[\mathrm{acost}(i)] + \bar{p}_i \cdot \mathbb{E}[\mathrm{acost}(a_1)] \\
&\quad + \mathrm{acost}(b_2) \cdot [(1-\bar{p}_{a_1}) - (1-\bar{p}_i)] + (1-\bar{p}_{a_1}) \cdot 1 \\
&\leq \mathbb{E}[\mathrm{acost}(S_j)] + [1 + \bar{p}_i - 2\bar{p}_{a_1}] \\
&\leq \mathbb{E}[\mathrm{acost}(S_j)] + 1
\end{aligned}
$$

在第一个不等式中，我们应用归纳假设给出的期望摊销成本上界。在第三个等式中，我们重新整理，以得到 S_j 的期望摊销成本；在第二个不等式中，我们利用 $-\bar{p}_{a_1} \cdot \mathbb{E}[\mathrm{acost}(i)] + \bar{p}_i \cdot \mathbb{E}[\mathrm{acost}(a_1)] = 0$，以及 $\mathrm{acost}(b_2) \leq 1$。对于最后一个不等式，我们使用 $\bar{p}_i \leq 1$ 和 $2\bar{p}_{a_1} \geq 1$。对于 $\ell \in \{0,1\}$，论证和估计同样成立。至此，情况 1.3 得证。

情形1.4（中止-继续）：在此情形下， S_j 在遇到第一个 0 后停止且不提供证书，否则继续（如图7所示）。直观而言，在观察到第一个元素为 0 后停止且不提供证书，并非我们能采取的最佳做法。事实上， S_j 被以下策略（弱）支配，其中 T^1 是 S_j 在 $x_i = 1$下的子策略：

以概率 $\alpha = \frac{1-\bar{p}_i}{\bar{p}_i} \leq 1$，执行以下策略，称为 S''_j：进行第一轮测试。如果 $x_i = 1$，则停止（不生成证书）。如果 $x_i = 0$，则继续进行第2轮（对 x_{a_1} 和 x_{b_2}进行测试）。根据 x_{a_1}的结果，如果 $x_{a_1} = 0$，则继续执行 T'^{10}；否则执行 T'^{11}。子策略T'^{11}和 T'^{10}由 T^{11}和 T^{01}推导而来，并经过修改，使其在找到证书的那一轮结束后停止。之所以需要进行此修改，是因为根据我们的定义，轮次完成性质要求策略在找到证书的那一轮结束后停止，而且与 T^{11}和 T^{10}相比， T'^{11}和 T'^{10}在测试 x_i时会额外从 0 开始。该变换会使效用弱增，并使期望摊销成本弱减，因为可能以更低的摊销成本获得更多证书。 $x_i = 0$对应的子策略记为 T'^0，它是该子实例上 IO 的一种剪枝。因此， S''_j是 i-伪 IO 的一种剪枝。

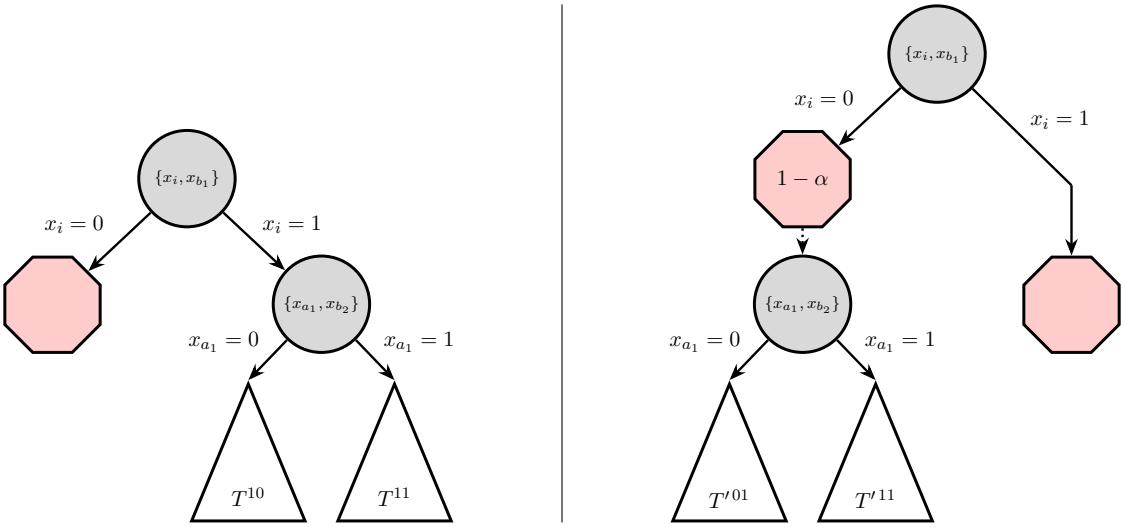

图7：情况 1.4（中止-继续）。左侧为 S_j；右侧为 S'_j，正文将其定义为分布$\tilde{}$，即在两个确定性策略（S''_j和 S''_j）上的分布，并可表示如上：如果第一轮中 $x_i = 0$，则 S'_j 以概率 $1 - \alpha$ 为 α 而停止：$= \frac{1-\bar{p}_i}{\bar{p}_i}$；或者以剩余概率，按照 S_j 在 $x_i = 1$ 之后继续的方式继续（可能会更早找到证书）。

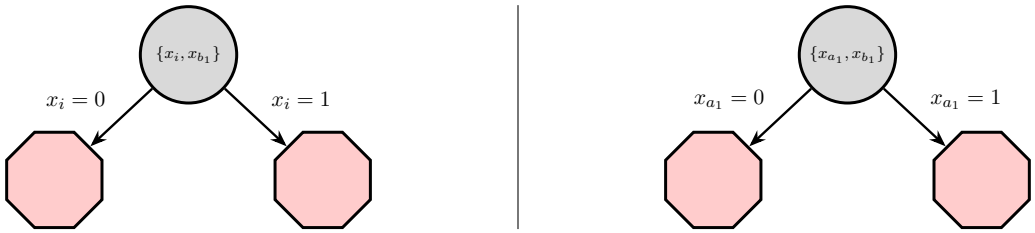

图8：情况 1.5（中止-中止）。左侧为 S_j，右侧为 S'_j。

以剩余概率 $1-\alpha$，我们对 x_i 和 x_{b_1} 进行测试，并在不生成证书的情况下停止。$\tilde{}$我们将此策略记为 S''_j。它也是 i-伪 IO 的一种剪枝。

最终得到的整体随机化策略记为 S'_j，其本身是对 $\mathcal{I}_i$进行随机剪枝后得到的策略。注意，在 S'_j 中，T'^0 以概率 $\alpha \cdot \bar{p}_i = 1 - \bar{p}_i$ 执行，这与 S_j中 T^1 的执行概率相同。对于 x_i 和 x_{b_1}，其期望摊销成本在 S'_j 中与 S_j相比没有变化，因此得到 $\mathbb{E}\left[\mathsf{acost}\left(S'_j\right)\right] \leq \mathbb{E}[\mathsf{acost}(S_j)]$。

对S''_j进行了剪枝（情况 1.2），并进行了分析）。因此，使用其中提出的替换方案，这种情况可以归约为上述情况。对于$\ell \in \{0, 1\}$，论证和估计也以相同方式成立。

请注意，S''_j的期望摊销成本在相同条件下不会更低。然而，使用 S''_j可以避免在尚未选择测试 x_{b_1} 的情况下，停止决策可能依赖于其结果。情况 1.2 和情况 1.5 中的替换方案是对 IO 进行的随机化 RC 剪枝，在第一轮结束之前不会终止，因此可以将其用于定义明确的总体策略（参见论文主体部分在情形分析之后关于如何针对 x_{b_1} 和 x_{b_2} 的所有可能结果获得总体策略的论证）。

情况 1.5（中止-中止）：这是指 S_j <style id='8}>在 x_i<style id='11}>的两种实现下均未能获得 0-证书便停止的情况，如 图 8<style id='15}>所示。我们通过将对 x_i <style id='21}>的测试替换为对 x_{a_1}<style id='24}>的测试，得到 S'_j。

由此得到的策略是对 IO 的剪枝。在两种情况下，效用均为 0。期望摊销成本只会降低： $\mathbb{E}\left[\text{acost}\left(S'_j\right)\right] = \mathbb{E}[$ acost(a_1)] +acost(b_1) $= \bar{p}_{a_1}$ +acost(b_1) $\leq \bar{p}_i$ +acost(b_1) $=\mathbb{E}[$acost(i)] + acost(b_1) $= \mathbb{E}[$acost(S_j)]。对于 $\ell \in \{0, 1\}$，论证和估算方法相同。因此，选择 $R_j = S'_j$ 后，归纳假设在这种情况下也成立。

情况 1.6（证书-中止）：这是指这种情况：在考虑了 S_j<style id='8}>的结果后，x_{b_1}还需要一个 0 才能得到 0-证书；如果 $x_i = 1$，则会在没有证书的情况下停止（如图 9所示）。

第一步变换仍然是将第一轮中对 x_i 的测试替换为 x_{a_1}；将由此得到的策略记为$\tilde{S}''_j$。然而，与情况 1.2 一样，这会降低第一轮找到 0-证书的概率。为弥补这一点，我们仅以概率 $1-\alpha$ 运行 S''_j，用于 $\alpha = \frac{\bar{p}_i - \bar{p}_{a_1}}{\bar{p}_i \cdot 1-\bar{p}_{a_1}} \leq 1$ ：( )。以剩

余概率，我们运行以下策略：同样从x_{a_1} 开始，并在第一轮进行 x_{b_1}。然而，与S''_j不同，在 $x_{a_1} = 1$的情况下，不是停止，而是继续进行第二轮测试（x_i 和 x_{b_2}）。由于只缺少一个 0，如果 $x_i = 0$， S''_j就已找到证书。如果 $x_i = 1$，则停止（结果取决于对 x_{b_2}的测试：如果 $x_{b_2} = 0$，则停止并获得证书；否则，停止但没有证书）。

令S''_j得到的策略为S'_j。注意， S'_j 已经是对 IO 的随机剪枝：由于最多只进行两轮测试，因此元素的正确顺序已经得到保证，无需通过调用归纳假设来进行进一步替换。

还需要界定期望效用和期望摊销成本。对于情况 $x_{b_2} = 1$，我们给出了相应的界；由于 x_{b_2} 的实现不会影响 S_j，这些结果也意味着 $x_{b_2} = 0$ 具有相同的界（效用更高，成本的界不变）。

$$\begin{aligned}\mathbb{E}\left[U_{\bar{k}}\left(S'_j\right)\right] &\geq \bar{p}_{a_1} + \alpha \cdot (1 - \bar{p}_{a_1}) \cdot \bar{p}_i \\ &= \bar{p}_{a_1} + \bar{p}_i - \bar{p}_{a_1} \\ &= \mathbb{E}[U_{\bar{k}}(S_j)]\end{aligned}$$

第一个不等式利用了 S'_j 的构造，并假设 $x_{b_2} = 1$ （效用可能最低）。

$$\begin{aligned}\mathbb{E}\left[\text{acost}\left(S'_j\right)\right] &= \mathbb{E}[\text{acost}(a_1)] + \text{acost}(b_1) + \alpha \cdot (1 - \bar{p}_{a_1}) \cdot (\mathbb{E}[\text{acost}(i)] + \text{acost}(b_2)) \\ &= \mathbb{E}[\text{acost}(i)] + \text{acost}(b_1) + \mathbb{E}[\text{acost}(a_1)] - (1 - \alpha \cdot (1 - \bar{p}_{a_1})) \cdot \mathbb{E}[\text{acost}(i)] \\ &\quad + \alpha \cdot (1 - \bar{p}_{a_1}) \cdot \text{acost}(b_2) \\ &\leq \mathbb{E}[\text{acost}(S_j)] + [\bar{p}_{a_1} - \bar{p}_i + \alpha \cdot (1 - \bar{p}_{a_1}) \cdot \bar{p}_i] + [\alpha \cdot (1 - \bar{p}_{a_1})] \\ &\leq \mathbb{E}[\text{acost}(S_j)] + 0 + \left[1 - \frac{\bar{p}_{a_1}}{\bar{p}_i}\right] \\ &\leq \mathbb{E}[\text{acost}(S_j)] + 1\end{aligned}$$

在第一个不等式中，我们利用了 $\mathbb{E}[$acost(i)] $= \bar{p}_i$、 $\mathbb{E}[$acost(a_1)] $= \bar{p}_{a_1}$ 以及 acost(b_2) ≤ 1。在第二个不等式中，我们应用了 α的定义。正如前文所述，这对 x_{b_1} 和x_{b_2}的任意实现都成立；即使这些元素不存在（也就是说，如果 $\ell \in \{0, 1\}$），结论仍然成立。因此，令 $R_j = S'_j$，我们便可得出归纳假设对情况 1.6 也成立。

情况 1.7（证书-证书）：与情形分析中一样，我们只考虑满足以下条件的实例：

$k \geq 1$，因此只有在以下条件成立时才会出现这种情况： $x_{b_1} = 0$ （因此 $\ell \geq 1$），并且 $k = 1$。 S_j 不是 IO 的剪枝，因此我们将其替换为策略 S'_j，该策略在第一轮先测试 x_{a_1} 和 x_{b_1}，并且还能为 x_{a_1} 的两种结果都找到证书（另见图 10）。请注意， S'_j 是 IO 的剪枝。两种情况下的效用均为 1。与情况 1.5 类似，很容易看出 $\mathbb{E}\left[\text{acost}\left(S'_j\right)\right] \leq \mathbb{E}[\text{acost}(S_j)]$。

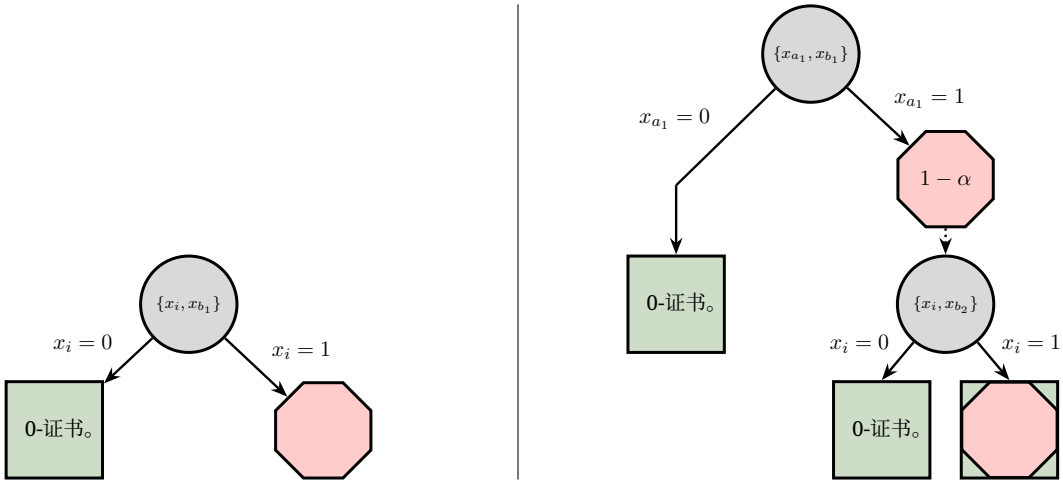

图 9：情况 1.6（证书-中止）。左侧为 S_j；右侧为 S'_j，正文中将其定义为两个确定性策略上的一个分布。它可以用如下图示表示：如果第一轮中 $x_{a_1} = 1$，则以概率 $1 - \alpha$ 执行 $\alpha := \frac{\bar{p}_i - \bar{p}_{a_1}}{\bar{p}_i \cdot (1 - \bar{p}_{a_1})}$，执行会在没有证书的情况下停止；以剩余概率进行第二个测试轮次（x_i 和 x_{b_2}）。右下角的节点表示结果取决于（固定的）值 x_{b_2}：如果 $x_{b_2} = 0$，则有证书（正方形）；否则没有（八边形）。

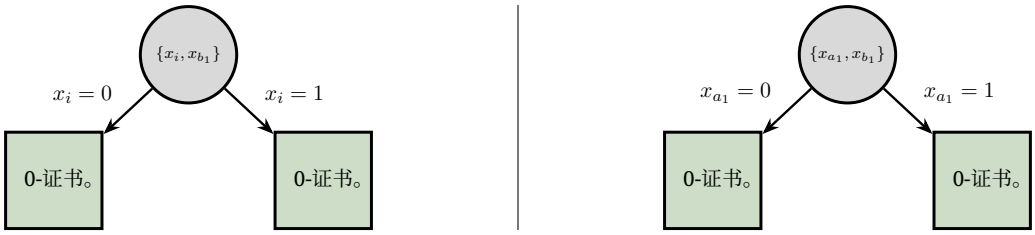

图 10：情况 1.7（证书-证书）。S_j 在左侧，S'_j 在右侧。

设定 $R_j = S'_j$ 时，归纳假设成立。同样，这些界对 $\ell = 1$ 也成立（如上所述，如果 $\ell = 0$，则情况 1.7 不会出现）。

情况 1 的分析至此完成。

情形 2： $\bar{p}_i < \frac{1}{2}$

在这种情况下，我们假设 $\bar{p}_i < \frac{1}{2}$。这意味着 $\bar{p}_{a_1} \geq \bar{p}_i$，进一步根据 acost 的定义，有 $\mathbb{E}[\text{acost}(a_1)] = (1 - \bar{p}_{a_1})$ 以及 $\mathbb{E}[\text{acost}(i)] = (1 - \bar{p}_i)$。参照情况 1 进行论证，总体上需要考虑 7 个子情况。下面对这些子情况进行分析。

情况 2.1（继续-继续）：这里同样是前两轮总能完成的情况。我们采用与情况 1.1 相同的变换和界，这些变换和界并不依赖于概率取值范围的性质。因此，计算结果可以直接沿用。

情况 2.2（继续-中止）：这是这样一种情况：S_j 在第一轮后停止且不提供证书，前提是 $x_i = 1$，否则继续，如图 11所示。我们将 S_j 转换为 S'_j，它~将是两种策略的随机化组合：S''_j 和 S''_j；二者都从 S_j 开始，交换 x_i 和 x_{a_1}；如果 $x_{a_1} = 1$，则如原策略一样，在第一轮后直接停止且不提供证书。

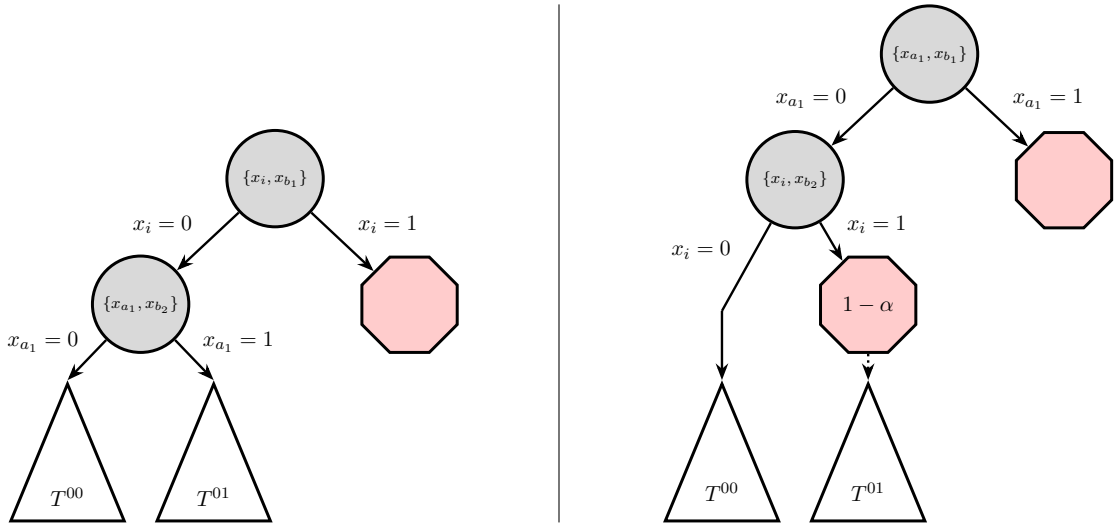

图 11：情形 2.2（继续-中止）。 S_j 在左侧； S'_j 在右侧，可定义为一种随机化策略：对于 α，以概率 $1-\alpha$ 提前停止：$= \frac{\bar{p}_i \cdot}{(1-\bar{p}_{a_1})\ \bar{p}_{a_1} \cdot (1-\bar{p}_i)}$，若 $x_{a_1} = 0$ 且 $x_i = 1$。

如果 $x_{a_1} = 0$，则 S''_j 以概率 α 被选取：$= \frac{\bar{p}_i \cdot}{(1-\bar{p}_{a_1})\ \bar{p}_{a_1} \cdot\ (1-\bar{p}_i)} \leq 1$，并继续执行以下子策略，记为 T'^0：继续对 x_i 和 x_{b_2} 进行测试；如果 $x_i = 0$，则继续执行 T^{00}，否则执行 T^{01}。以剩余概率 $1-\alpha$，策略 $\tilde{S}''_j$ 继续执行 $\tilde{T}'^0$，定义如下：同样测试元素 x_i 和 x_{b_2}；但是，此后仅当 $x_i = 0$（使用 T^{00}）时才继续，否则停止求值且不提供证书。

令 S'_j 为基于 S''_j 和 S''_j 得到的随机化策略。注意，这确保了 T^{00} 和 T^{01} 分别在 S_j 和 S'_j 中相同的子实例上执行；根据 α 的选择，变换前后到达 T^{00} 和 T^{01} 的总体概率相同，从而保持效用。

根据 S''_j 和 $\tilde{S}''_j$ 的构造， T'^0 和 $\tilde{T}'^0$ 是针对 $x_{a_1} = 0$ 在第一轮后得到的子实例上的 i-伪 IO 的随机化 RC 剪枝。应用归纳假设后，我们可以分别用 IO 的随机剪枝 R^0 和 $\tilde{R}^0$（在该子实例上）替换二者，从而得到最终的随机化总体策略 R_j。在 R_j 中，所有元素都按期望顺序排列，且它是 IO 的随机剪枝。下文将界定效用和摊销成本。

$$
\begin{aligned}
\mathbb{E}[U_{\bar{k}}(R_j)] &= \alpha \cdot \bar{p}_{a_1} \cdot \mathbb{E}\big[U_{\bar{k}_1-1}\big(R^0\big)\big] + (1-\alpha) \cdot \bar{p}_{a_1} \cdot \mathbb{E}\Big[U_{\bar{k}_1-1}\Big(\tilde{R}^0\Big)\Big] \\
&\geq \bar{p}_{a_1} \cdot \Big[\alpha \cdot \mathbb{E}\big[U_{\bar{k}_1-1}\big(T'^0\big)\big] + (1-\alpha) \cdot \mathbb{E}\Big[U_{\bar{k}_1-1}\Big(\tilde{T}'^0\Big)\Big]\Big] \\
&= \bar{p}_{a_1} \cdot \big[\bar{p}_i \cdot \mathbb{E}\big[U_{\bar{k}_2-2}\big(T^{00}\big)\big] + \alpha \cdot (1-\bar{p}_i) \cdot \mathbb{E}\big[U_{\bar{k}_2-1}\big(T^{01}\big)\big]\big] \\
&= \bar{p}_i \cdot \bar{p}_{a_1} \cdot \mathbb{E}\big[U_{\bar{k}_2-2}\big(T^{00}\big)\big] + \bar{p}_i \cdot (1-\bar{p}_{a_1}) \cdot \mathbb{E}\big[U_{\bar{k}_2-1}\big(T^{01}\big)\big] \\
&= \mathbb{E}[U_{\bar{k}}(S_j)]\,.
\end{aligned}
$$

第一个等式使用了 R_j、 R^0 和 $\tilde{R}^0$ 的定义；在第一个不等式中，我们重新排列各项并应用归纳假设（以及相应的替换）。第二个等式使用了 T'^0 和 $\tilde{T}'^0$ 的构造（在给定 $x_i = 0$ 的条件下，T^{00} 的执行与内部随机化相互独立）。第三个等式应用了 α 的定义。

$$
\begin{aligned}
\mathbb{E}[\mathrm{acost}(R_j)] &= \mathbb{E}[\mathrm{acost}(a_1)] + \mathrm{acost}(b_1) + \alpha \cdot \bar{p}_{a_1} \cdot \mathbb{E}\big[\mathrm{acost}\big(R^0\big)\big] + (1-\alpha) \cdot \bar{p}_{a_1} \cdot \mathbb{E}\Big[\mathrm{acost}\Big(\tilde{R}^0\Big)\Big] \\
&\le \mathbb{E}[\mathrm{acost}(a_1)] + \mathrm{acost}(b_1) + \alpha \cdot \bar{p}_{a_1} \cdot \big[\mathbb{E}\big[\mathrm{acost}\big(T'^0\big)\big] + 1\big] + (1-\alpha) \cdot \bar{p}_{a_1} \cdot \Big[\mathbb{E}\Big[\mathrm{acost}\Big(\tilde{T}'^0\Big)\Big] + 1\Big] \\
&= \mathbb{E}[\mathrm{acost}(a_1)] + \mathrm{acost}(b_1) + \bar{p}_{a_1} \cdot 1 \\
&\quad + \bar{p}_{a_1} \cdot \big[\mathbb{E}[\mathrm{acost}(i)] + \mathrm{acost}(b_2) + \bar{p}_i \cdot \mathbb{E}\big[\mathrm{acost}\big(T^{00}\big)\big] + \alpha \cdot (1-\bar{p}_i) \cdot \mathbb{E}\big[\mathrm{acost}\big(T^{01}\big)\big]\big] \\
&= \mathbb{E}[\mathrm{acost}(i)] + \mathrm{acost}(b_1) \\
&\quad + \bar{p}_i \cdot \big[\mathbb{E}[\mathrm{acost}(a_1)] + \mathrm{acost}(b_2) + \bar{p}_{a_1} \cdot \mathbb{E}\big[\mathrm{acost}\big(T^{00}\big)\big]\big] + \bar{p}_{a_1} \cdot (1-\bar{p}_i) \cdot \alpha \cdot \mathbb{E}\big[\mathrm{acost}\big(T^{01}\big)\big] \\
&\quad + [(1-\bar{p}_i) \cdot \mathbb{E}[\mathrm{acost}(a_1)] - (1-\bar{p}_{a_1}) \cdot \mathbb{E}[\mathrm{acost}(i)]] + (\bar{p}_{a_1} - \bar{p}_i) \cdot \mathrm{acost}(b_2) + \bar{p}_{a_1} \\
&\le \mathbb{E}[\mathrm{acost}(i)] + \mathrm{acost}(b_1) \\
&\quad + \bar{p}_i \cdot \big[\mathbb{E}[\mathrm{acost}(a_1)] + \mathrm{acost}(b_2) + \bar{p}_{a_1} \cdot \mathbb{E}\big[\mathrm{acost}\big(T^{00}\big)\big] + (1-\bar{p}_{a_1}) \cdot \mathbb{E}\big[\mathrm{acost}\big(T^{01}\big)\big]\big] \\
&\quad + 0 + 2\bar{p}_{a_1} - \bar{p}_i \\
&\le \mathbb{E}[\mathrm{acost}(S_j)] + 1\,.
\end{aligned}
$$

第一个等式应用了 R_j 的构造；请注意，对于 $x_{a_1} = 1$，第一轮之后不会产生额外成本。第一个不等式使用了归纳假设。在第二个等式中，我们使用了 T'^0 和 $\tilde{T}'^0$ 的定义（两者都由 T^{00} 执行），并在第三个等式中重新排列各项。第二个不等式使用了 α 的定义、在情形 2 的范围内 {7}acost(a_1)] = (1 − $\bar{p}_{a_1}$) 和 $\mathbb{E}$[acost(i)] = (1 − $\bar{p}_i$) 这一事实，以及 acost(b_2) ≤ 1 这一界。在最后一步，我们取回了 S_j 的期望摊销成本，并由此得到归纳假设。和之前一样，结论也适用于情形 $\ell \in \{0, 1\}$。

情况 2.3（证书-继续）：这里同样属于只需一个 0 的情况（计入第 1 轮测试 x_{b_1} 的结果后）——可与情况 1.3 对比。令 S'_j 为通过交换元素 x_i 和 x_{a_1} 的位置得到的策略，见图 6。由于 x_i 或 x_{a_1} 中任意一个出现 0 就足以找到证书，因此除两者结果均为 1 的情况外，我们都会成功终止；若两者结果均为 1，则执行某个子策略 T^{11}。这些事件在 S_j 和 S'_j 中发生的概率相同；与情况 1.3 不同的是，对 b_2 的测试的期望摊销成本降低了。令 T'^1 为 S'_j 在 $a_1 = 1$ 情况下的子策略。 T'^1 是 $\mathcal{I}_i$ 的 RC-剪枝，因此我们可以将其替换为该子实例上 IO 的随机剪枝 R^1，而不会降低期望效用，且额外的期望（摊销）成本有界。将 T'^1 替换为 R^1 后得到的策略 R_j 是 IO 的随机剪枝，因为现在所有元素的顺序都恰当。总之，按照与情况 1.3 相同的论证（不使用概率和摊销成本的取值范围），我们有 $\mathbb{E}[U_{\bar{k}}(R_j)] \ge \mathbb{E}[U_{\bar{k}}(S_j)]$。关于期望摊销成本，我们观察到：

$$
\begin{aligned}
&\mathbb{E}[\mathrm{acost}(R_j)] = \mathbb{E}[\mathrm{acost}(a_1)] + \mathrm{acost}(b_1) + (1-\bar{p}_{a_1}) \cdot \mathbb{E}\big[\mathrm{acost}\big(R^1\big)\big] \le \mathbb{E}[\mathrm{acost}(a_1)] + \mathrm{acost}(b_1) \\
&+ (1-\bar{p}_{a_1}) \cdot \big[\mathbb{E}\big[\mathrm{acost}\big(T'^1\big)\big] + 1\big] = \mathbb{E}[\mathrm{acost}(a_1)] + \mathrm{acost}(b_1) + (1-\bar{p}_{a_1}) \cdot \big[\mathbb{E}[\mathrm{acost}(i)] + \\
&\mathrm{acost}(b_2) + (1-\bar{p}_i) \cdot \mathbb{E}\big[\mathrm{acost}\big(T^{11}\big)\big]\big] + (1-\bar{p}_{a_1}) \cdot 1 = \mathbb{E}[\mathrm{acost}(i)] + \mathrm{acost}(b_1) + (1-\bar{p}_i) \cdot \big[\mathbb{E}[\\
&\mathrm{acost}(a_1)] + \mathrm{acost}(b_2) + (1-\bar{p}_{a_1}) \cdot \mathbb{E}\big[\mathrm{acost}\big(T^{11}\big)\big]\big] - \bar{p}_{a_1} \cdot \mathbb{E}[\mathrm{acost}(i)] + \bar{p}_i \cdot \mathbb{E}[\mathrm{acost}(a_1)] + \\
&\mathrm{acost}(b_2) \cdot [(1-\bar{p}_{a_1}) - (1-\bar{p}_i)] + (1-\bar{p}_{a_1}) \cdot 1 \le \mathbb{E}[\mathrm{acost}(S_j)] + (1 + \mathrm{acost}(b_2)) \cdot (\bar{p}_i - \bar{p}_{a_1}) + (\\
&1 - \bar{p}_{a_1}) \cdot 1 \le \mathbb{E}[\mathrm{acost}(S_j)] + 1\,.
\end{aligned}
$$

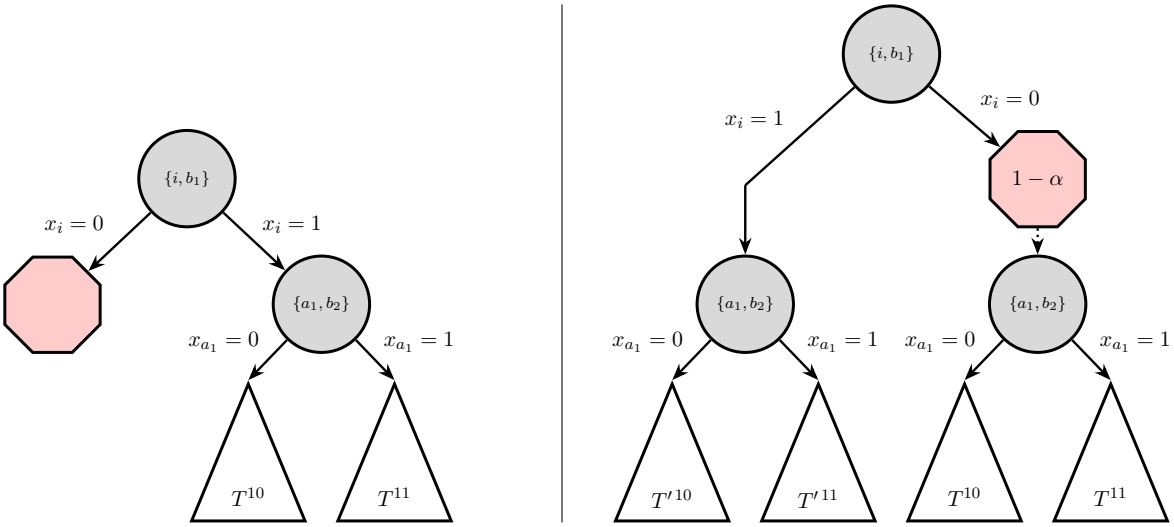

图12：情形 2.4（中止-继续）。左侧为 S_j；右侧为 S'_j，正文将其定义为在两种确定性策略（S''_j 和 S''_j）之间进行随机选择，并且可以表示为如上所示：如果$x_i = 0$，S'_j 就按照 S_j 在 $x_i = 1$ 之后继续的方式继续（可能会更早找到证书）。如果$x_i = 1$，它就以概率 α 按照与 S_j 相同的方式继续：$= \frac{1-2\bar{p}_i}{1-\bar{p}_i}$或者在未找到证书的情况下停止。

在第一个不等式中，我们应用归纳假设中关于期望摊销成本的界。在第三个等式中，我们重新整理以分离出 S_j 的期望摊销成本；在第二个不等式中，我们利用 $\mathbb{E}[\mathsf{acost}(i)] = 1 - \bar{p}_i$ 且 $\mathbb{E}[\mathsf{acost}(a_1)] = 1 - \bar{p}_{a_1}$。最后一个不等式成立是因为 $\bar{p}_i - \bar{p}_{a_1} \leq 0$。对于 $\ell \in \{0, 1\}$，论证和估计方式相同。

情形 2.4（中止-继续）：与情形1.4一样，我们论证 S_j 可以替换为 $\mathcal{I}_i$ 的剪枝上的分布，而这些剪枝此前已经考虑过，因此问题可归结为那些情形。同样，整体思路是在 $x_i = 0$ 之后而不是在 $x_i = 1$ 处继续评估，从而在期望意义下获得更低的摊销成本和更高的效用。然而，由于（$1 - \bar{p}_i$）$\geq \bar{p}_i$，我们需要以一定概率在 $x_i = 1$ 之后继续评估，以维持效用。令 T^1 为 S_j 中针对 $x_i = 1$ 的子策略，其子策略为 T^{11} 和 T^{10}。

S'_j（另见图12）构造如下：以概率 $\alpha := \frac{1-2\bar{p}_i}{1-\bar{p}_i} \leq 1$ 执行以下操作，记为 S''_j：按照之前的方式，在 x_i 和 x_{b_1}上进行测试，完成第 1 轮。如果 $x_i = 0$，则继续执行 T'^0；T'^0通过以下方式修改 T^1 得到：在某一轮结束时，一旦找到 $k_1 - 1$ 个值为 0 的变量，就立即终止，而不是等到找到 k_1 个（回想一下，现在我们可以使用来自 x_i 的额外一个 0，因此证书所需的 0 少一个）。令 T'^{10}和 T'^{11}分别为 T'^0在第二轮后针对 $x_{a_1} = 0$和 $x_{a_1} = 1$所采用的子策略。这样，T'^0满足轮次完成性质。此外，对于所有 T^1 以证书终止的实现，都会找到证书，从而得到更高或相等的 $U_{\bar{k}_1 - 1}$。与此同时，期望摊销成本不会增加。如果 $x_i = 1$，则像之前一样继续执行 T^1。

以剩余概率 $1 - \alpha$，若 $x_i = 0$，也继续执行 T'^0；否则停止且不提供证书。将此策略记为 S''_j。

选择概率 α 的方式是，使得在 S_j 和 S'_j 中，对 x_{a_1} 和 x_{b_2} 进行测试的概率相同，均为 $1 - \bar{p}_i = \bar{p}_i + \alpha \cdot (1 - \bar{p}_i)$，因此对期望摊销成本的贡献相同。此外，在 S'_j 中，T^{11} 或 T'^{11}，以及 T^{10} 或 T'^{10} 的出现概率，分别与 S_j 中的 T^{11} 和 T^{10} 相同，且期望摊销成本不高于后者、期望效用不低于后者。总体而言，$\mathbb{E}[\mathsf{acost}(S_j)] \geq \mathbb{E}\left[\mathsf{acost}\left(S'_j\right)\right]$，且 $U_{\bar{k}}(S_j) \leq U_{\bar{k}}\left(S'_j\right)$。进一步地，$S'_j$ 是

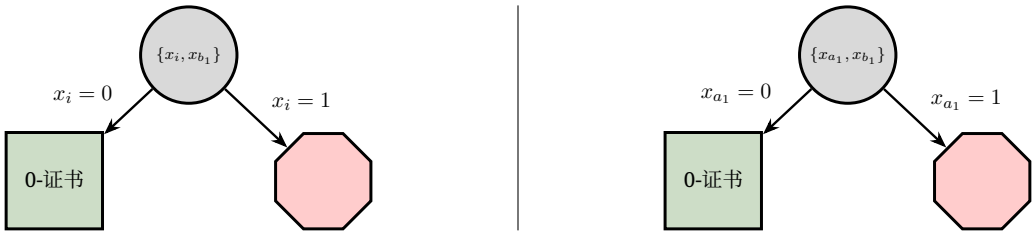

图 13：情形 2.6（证书-中止）。 S_j 在左侧，S'_j 在右侧。

˜是对 S''_j 和 S''_j 的随机化，两者都是 $\mathcal{I}_i$ 的剪枝，分别对应案例 2.1 和 2.2。因此，这种情况可以归结为这两个案例。

与情形 1.4 类似，由情形 2.1 和 2.2 中所用替换得到的替换策略，是不会在第一轮结束前终止的 IO 随机化 RC 剪枝，因此可用于构造定义良好的总体策略（参见论文主体部分在情形分析之后，针对 x_{b_1} 和 x_{b_2} 的所有可能结果构造总体策略的论证）。

情况 2.5（中止-中止）：这是指对于 x_i 测试的两种结果，S_j 均在未获得 0-证书的情况下停止的情形（参见图 8）。我们通过将元素 x_i 替换为 x_{a_1}来得到 S'_j 。所得策略是 IO 的一种剪枝。两种情况下的效用始终为 0。期望摊销成本不会增加： $\mathbb{E}\left[\text{acost}\left(S'_j\right)\right] = \mathbb{E}[\text{acost}(a_1)] + \text{acost}(b_1)$ $= 1 - \bar{p}_{a_1} + \text{acost}(b_1) \leq 1 - \bar{p}_i + \text{acost}(b_1) = \mathbb{E}[\text{acost}(i)] + \text{acost}(b_1) = \mathbb{E}[\text{acost}(S_j)]$。对于 $\ell \in \{0, 1\}$，论证和估计同样成立。因此，在此情形下，归纳假设对 $R_j = S'_j$也成立。

情形 2.6（证书-中止）：结构（如图 13所示）表明，在考虑对 x_{b_1}的测试结果后，距离得到证书恰好还缺少一个 0。通过将元素 x_i 替换为x_{a_1}，我们得到 S'_j（S_j）。摊销成本不增加这一点 $\mathbb{E}\left[\text{acost}\left(S'_j\right)\right] \leq \mathbb{E}[\text{acost}(S_j)]$ 很容易类比情况 2.5 得出。由于 S'_j 是 IO 的剪枝，因此令 $R_j = S'_j$，即可完成情况 2.6。

情况 2.7（证书-证书）：在这种情况下，对于测试 x_i 的¯两种结果，我们都在第一轮后以证书结束（参见图 10）。与情况 1.7 一样，我们受到 $k \geq 1$的限制，因此仅当 $x_{b_1} = 0$ （从而 $\ell \geq 1$）且 $k = 1$时，这种¯情况才会出现。 S_j 不是 IO 的剪枝，因此我们将其替换为 S'_j，后者在第一轮开始时测试 x_{a_1} 和 x_{b_1}，并且对于 x_{a_1}的两种实现也都会找到证书。注意， S'_j 是 IO 的剪枝。在这两种情况下，效用始终为 1。此外，按照情况 2.5 中相同的计算，我们得到 $\mathbb{E}\left[\text{acost}\left(S'_j\right)\right] \leq \mathbb{E}[\text{acost}(S_j)]$。同样，令 $R_j = S'_j$即可满足归纳假设。对于 $\ell = 1$ ，这些界也成立（与情况 1.7 类似，若 $\ell = 0$，则情况 2.7 不会出现）。

至此，情况 2 的所有可能性均已涵盖，情形分析到此结束。

C.2 引理7 的证明计算

情况 1.2 的辅助计算：回想一下 $\alpha = \frac{\bar{p}_i - \bar{p}_{a_1}}{\bar{p}_i \cdot \ 1 - \bar{p}_{a_1}}$ ( )。 ————

本节将推迟计算的两个界中的第一个等号成立：

$$(1 - \bar{p}_i) \cdot \mathbb{E}[\text{acost}(a_1)] - [1 - \bar{p}_{a_1} - \alpha \cdot (1 - \bar{p}_{a_1})] \cdot \mathbb{E}[\text{acost}(i)]$$
$$= \bar{p}_{a_1} - \bar{p}_i \cdot \bar{p}_{a_1} - \bar{p}_i + \bar{p}_{a_1} \cdot \bar{p}_i + \bar{p}_i \cdot (1 - \bar{p}_{a_1}) \cdot \alpha$$

$$= \bar{p}_{a_1} - \bar{p}_i + (\bar{p}_i - \bar{p}_{a_1}) = 0\,.$$

在第一步中，我们利用 $\mathbb{E}[\mathsf{acost}(a_1)] = \bar{p}_{a_1}$ 和 $\mathbb{E}[\mathsf{acost}(i)] = \bar{p}_i$，并在第二步中使用 α 的定义。

剩余表达式界定了递归带来的额外成本，并通过以高于 S_j 中的概率测试 x_{b_2}：

$$\begin{aligned}
&[\bar{p}_{a_1} + \alpha \cdot (1 - \bar{p}_{a_1}) - \bar{p}_i] \cdot \mathrm{acost}(b_2) + [\bar{p}_{a_1} + \alpha \cdot (1 - \bar{p}_{a_1})] \\
&\quad \leq [\alpha \cdot (1 - \bar{p}_{a_1}) - \alpha \cdot \bar{p}_i \cdot (1 - \bar{p}_{a_1})] + [1 - (1 - \bar{p}_{a_1}) + \alpha \cdot (1 - \bar{p}_{a_1})] \\
&\quad = 1 + \alpha \cdot (1 - \bar{p}_{a_1}) \cdot [1 - \bar{p}_i] - [(1 - \bar{p}_{a_1}) \cdot (1 - \alpha)] \\
&\quad = 1 + (1 - \bar{p}_{a_1}) \cdot \left[(1 - \bar{p}_i) \cdot \frac{\bar{p}_i - \bar{p}_{a_1}}{\bar{p}_i \cdot (1 - \bar{p}_{a_1})} - \frac{\bar{p}_i \cdot (1 - \bar{p}_{a_1}) - (\bar{p}_i - \bar{p}_{a_1})}{\bar{p}_i \cdot (1 - \bar{p}_{a_1})} \right] \\
&\quad = 1 + \frac{1}{\bar{p}_i} \cdot [(1 - \bar{p}_i) \cdot (\bar{p}_i - \bar{p}_{a_1}) - (\bar{p}_{a_1} - \bar{p}_{a_1} \cdot \bar{p}_i)] \\
&\quad = 1 + \frac{1 - \bar{p}_i}{\bar{p}_i} \cdot (\bar{p}_i - 2\bar{p}_{a_1}) \\
&\quad \leq 1\,.
\end{aligned}$$

在第一步中，我们使用 $\mathsf{acost}(b_2) \leq 1$ 和 $\bar{p}_i - \bar{p}_{a_1} = \alpha \cdot \bar{p}_i \cdot (1 - \bar{p}_{a_1})$。在最后一步中，我们使用 $(1 - \bar{p}_i) \geq 0$，并且在此处所考虑的情况下使用 $2\bar{p}_{a_1} \geq 1 \geq \bar{p}_i$。