arXiv:2301.10197v1 [cs.LO] 24 Jan 2023

马尔可夫决策过程模型检测算法实践者指南*

ArndHartmanns[1], Sebastian Junges[2],
Tim Quatmann[3], 和 Maximilian Weininger[4]

[1] 特文特大学，恩斯赫德，荷兰 a.hartmanns@utwente.nl[2] 拉德堡德大学，奈梅亨，荷兰 sebastian.junges@ru.nl[3] 亚琛工业大学，亚琛，德国
tim.quatmann@cs.rwth-aachen.de[4] 慕尼黑工业大学，慕尼黑，德国
maxi.weininger@tum.de

摘要。 对马尔可夫决策过程（MDP）进行无折扣可达性和期望奖励性质的模型检测，是验证在不确定性下运行的系统的关键。常用算法包括策略迭代和值迭代的各种变体；在工具竞赛中，大多数参赛者依赖后者。这些算法通常在最坏情况下需要指数时间。然而，该问题同样可以表述为线性规划，并能在多项式时间内求解。本文详细综述当前最先进的 MDP 模型检测算法，重点关注性能与正确性。我们着重介绍这些算法的根本差异，并描述各种优化和实现变体。我们使用两个概率模型检测器，在三组基准测试上对所有算法的浮点实现和精确算术实现进行了实验比较。结果表明，（乐观）值迭代是合理的默认选择，但在某些特定场景下，其他算法更为合适。因此，本文为 MDP 验证实践者——工具开发者和用户——提供了一份指南。

1 引言

马尔可夫决策过程的验证对于设计和评估带有传感器噪声的网络物理系统、生物和化学过程、网络协议以及许多其他复杂系统至关重要。马尔可夫决策过程是处理不确定性下序贯决策的标准模型，因此也是强化学习的核心。许多可信性评估和安全保障方法都在某种程度上依赖于针对时序逻辑性质对马尔可夫决策过程进行验证。概率模型检测[4,5] 为此任务提供了强大的工具。

MDP 模型检测的核心查询是发生某种坏事的最坏情况概率（可达性）和任务完成前的期望资源消耗（期望奖励）。这些是无限（无折扣）时域查询：它们询问某个随机变量在某一事件发生前（该事件构成时域）的概率或期望，但查询本身没有上界。许多更复杂的性质最终都可归结为求解可达性或期望奖励。例如，如果对某事

* 本研究由欧盟“地平线 2020”研究与创新计划资助，资助依据为玛丽·斯克沃多夫斯卡-居里资助协议第 101008233 号（MISSION）；此外还获得了 NWO VENI 资助，编号为 639.021.754。

坏事的描述采用线性时序逻辑（LTL），那么通过与合适的自动机进行乘积构造，就能将 LTL 查询归结为可达性 [6]。本文旨在确定求解无限时域可达概率和期望奖励的实践最优算法；我们采用的方法是实证评估。

马尔可夫决策过程（MDP）的分析在许多领域都得到了深入研究，并产生了三类主要算法：值迭代（VI）、策略迭代（PI）和线性规划（LP）[42]。虽然无限时域查询在验证领域中很自然，但它们不同于运筹学、规划和强化学习等领域中的标准问题。在这些领域，主要关注的是计算一种策略，使其在无限时域内（通常近似地）优化折扣后的 期望奖励；其中，未来累积的奖励会按折扣因子加权，且该因子会以指数方式优先考虑较早累积的奖励 < 1 。

验证中缺少折扣机制，影响深远。贝尔曼算子本质上描述了对期望奖励进行的一步向后更新；在有折扣的情况下，它是压缩映射，而在没有折扣时则不是。这使得基于值迭代的验证方法需要复杂得多的终止判据 [26]。事实上，对于任意固定的折扣因子[37], ，值迭代都能在多项式时间内运行；对于策略迭代以及使用单纯形算法求解线性规划，也已知有类似结果 [47]。相比之下，已知在无折扣情形下，值迭代 [9] 和策略迭代 [18] 在最坏情况下会呈指数级增长。

那么，用于马尔可夫决策过程模型检测的最佳算法是什么？可以通过线性规划公式并采用障碍法求解，得到一种多项式时间算法 [11]。基于线性规划的方法（及其向混合整数线性规划的扩展）在多目标模型检测等更高级的查询中也很突出 [19], , 用于反例生成 [20], , 以及参数化马尔可夫链的分析 [15]。不过，业界普遍认为，迭代方法，尤其是值迭代，更适合求解马尔可夫决策过程。事实上，参加 QCo mp 竞赛的所有模型检测器默认都采用值迭代的变体 [13]。这种一致性可能会造成误导。事实上，对于随机博弈和某种特定技术，使用线性规划求解底层的马尔可夫决策过程可能更为可取[3, 附录 E.4]。在运行时保障的示例中，由于数值不稳定性，策略迭代更受青睐 [32, 第 6] 节。来自 [26] 的一个玩具示例，是对基于值迭代的方法的一项著名挑战。尽管线性规划十分重要、马尔可夫决策过程易于编码，且功能强大的现成线性规划求解器随手可得，许多模型检测工具（直到最近）仍未通过线性规划求解器支持马尔可夫决策过程模型检测。

本文重新审视策略迭代和线性规划算法，以探究概率模型检测是否关注了错误的算法类别。我们报告了在模型检测器 Storm [31] 和 mcsta [28]中，采用两种独立实现开展的大规模实证研究结果。我们发现，就性能和可扩展性而言，乐观值迭代 [29] 是标准基准测试集（范围超出竞赛基准测试）上的稳健选择，但在具有挑战性的案例中，表现可能会明显逊色。我们还强调了精度和可靠性问题。数值算法，尤其是那些在极限处收敛的算法，容易得出错误结果。对于值迭代，人们认识到这一问题后，在过去

十年间 [8,26,29,33,41,43]提出了一系列改进。我们表明，策略迭代也面临类似问题。使用浮点算术时，还可能出现其他问题 [27,46]。我们使用多种线性规划求解器对各类基准测试进行求解，结果令人担忧。因此，我们还纳入了使用有理数算术进行精确计算的结果。

本研究的局限性。 对算法进行全面的实验研究，需要谨慎界定评估范围。我们研究的是能够完全装入内存的扁平马尔可夫决策过程表示（即不考虑状态空间探索过程和符号方法）。我们选择了旨在收敛到 最优值的算法。我们还排除了通过模拟或模型检测结果指导探索、逐步构建并求解（部分或抽象）马尔可夫决策过程的方法：这类方法属于正交改进，同样能从求解部分马尔可夫决策过程的更快算法中获益。此外，本研究考察的是算法，而非其实现。为降低潜在实现缺陷的影响，我们尽可能使用两种独立工具。我们的实验在单一类型的机器上运行——我们不研究不同硬件的影响。

贡献。 本文全面概述了如何对马尔可夫决策过程中的无限时域性质进行模型检测，使马尔可夫决策过程模型检测更易于理解，同时也通过澄清尚未解决的问题推动了最先进水平的发展。我们的研究建立在全面的实证评估之上，使用两个独立的代码库，基准测试取自标准基准测试套件和近期出版物，对比了 10 种线性规划求解器，并研究了多种重要预处理技术的影响。本文提供了新的见解，并重新审视了一些广为流传的观点：主要亮点包括一个简单但颇具挑战性的马尔可夫决策过程新族，它会导致所有浮点线性规划求解器得出错误结果（第 2.3节）、关于采用 ε 精确策略评估器的策略迭代可靠性的一项否定性结论（第 4节），以及对数值挑战性基准测试的评估，结果显示了值迭代在实际场景中的局限性（第 5.3节）。

2 背景

我们回顾带有可达性和奖励目标的马尔可夫决策过程，介绍求解算法及其保证，并讨论常用的优化方法。

2.1 马尔可夫决策过程

$\mathrm{D}_X := \{\mathrm{d}\colon X \to [0,1] \mid \sum_{x\in X} \mathrm{d}(x) = 1\}$ 表示 X 上的概率分布集合。

定义 1. 马尔可夫决策过程（*MDP*）*[42]*是一个元组 $\mathcal{M} = (\mathsf{S}, \mathsf{A}, \delta)$，其中状态集合 S 和动作集合 A均为有限集，并且转移函数$\delta\colon \mathsf{S} \times \mathsf{A} \to \mathsf{D}_\mathsf{S}$是部分定义的，满足 $\mathsf{A}(s) := \{ a \mid (s, a) \text{的定义域}(\delta) \} \neq \emptyset$对所有 $s \in \mathsf{S}$均成立。

$\mathsf{A}(s)$ 是状态 s 下可用动作的集合。 δ 将可用的状态-动作对映射为后继状态上的概率分布。若对所有 s，$|\mathsf{A}(s)| = 1$，则马尔可夫链（MC）是一个马尔可夫决策过程（MDP）。MDP 的语义 按通常方式定义，参见例如 [6, 第 10] 章。一个（无记忆确定性）策略——也称为策略或调度器——是函数 $\pi\colon \mathsf{S} \to \mathsf{A}$。直观而言，给定当前状态 s，它规定应执行哪个动作 $a \in \mathsf{A}(s)$。将策略 π 应用于一个MDP

会导出一个马尔可夫链 $\mathcal{M}^\pi$。该马尔可夫链中的一条路径是一个无限序列 $\rho = s_1 s_2 \ldots$，满足 $\delta(s_i, \pi(s_i))(s_{i+1}) > 0$。Paths 表示所有路径的集合，而 $\mathbb{P}_s^\pi$ 表示从状态 s 出发、定义在无限路径上的唯一概率测度 $\mathcal{M}^\pi$。

对于目标状态集合 T $\subseteq$ S 和 opt $\in\{\max, \min\}$，可达性目标 *reachability objective* $\mathrm{P}^{\mathrm{opt}}$(T) 会诱导一个随机变量 X: Paths $\to [0,1]$，其对最终到达目标状态的所有路径赋值为 1，对其他所有路径赋值为 0。$\mathrm{E}^{\mathrm{opt}}$(rew) 表示一个期望奖励目标，其中 rew: S $\to \mathbb{Q}_{\geq 0}$ 为每个状态指定一个奖励。rew(ρ) := $\sum_{i=1}^{\infty}$rew(s_i) 是路径 $\rho = s_1 s_2 \ldots$ 的累积奖励。这会得到一个随机变量 X: Paths $\to \mathbb{Q} \cup \{\infty\}$，将路径映射为其奖励。对于给定的目标及其随机变量 X，状态 $s \in$S 的值是 X 在由最优策略 π诱导的马尔可夫链的概率测度 $\mathbb{P}_s^\pi$ 下的期望，形式化地写作 V(s) := $\mathrm{opt}_{\pi \in \Pi} \mathbb{E}_s^\pi[X]$。

2.2 求解算法

值迭代（*VI*），例如 [14], 会计算一系列值向量，并在极限情况下收敛到最优值。在该算法的所有变体中，我们都从一个函数开始 x: S $\to \mathbb{Q}$，它为每个状态指定一个值的估计值。算法反复执行更新操作，以改进这些估计值。经过一些预处理后，当 x = V时，该操作具有唯一的不动点。因此，值迭代在极限情况下收敛到该值。值迭代的变体包括区间迭代[26], 可靠值迭代 [43] 和乐观值迭代 [29]。我们不详细讨论这些方法，而是请读者参阅相应的论文。

线性规划（*LP*），例如 [6, 第 10], 章将马尔可夫决策过程的转移结构和目标编码为一个线性优化问题。对于每个状态，线性规划中都有一个变量来表示其值的估计值。每个状态-动作对都被编码为关于这些变量的约束，目标集或奖励也同样如此。当且仅当每个状态对应的变量都被设为该状态的值时，线性规划才能达到唯一的最优值。我们将在第 3 节深入讨论线性规划的理论和实践方面。

策略迭代 *(PI)*，例如 [10, 第 4], 节，我们会计算一系列策略。从初始策略开始，我们评估其诱导马尔可夫链，通过切换次优选择来改进策略，并在新策略上重复这一过程。由于每个策略都比前一个策略更优，而且无记忆确定性策略的数量有限（其数量随状态数呈指数增长），最终我们会得到一个最优策略。我们将在第 4 节进一步讨论策略迭代。

2.3 保证

鉴于许多模型检测应用所涉及的重大利害关系，我们要求对算法结果 $\bar{v}$ 与真实概率或期望奖励 v之间的关系作出保证。首先，除非采用精确（有理数）运算或安全舍入 [27]，否则实现会受到浮点误差和精度不足的影响。这可能导致结果与真实值之间出现任意大的差异。其次，问题还在于算法本身的固有特性：VI 是一种近似算法，只有在极限情况下才会收敛到真实值。理论上，经过指数级迭代次数 [14]后再进行舍入，就有可能得到精确结果；但在实践中，这会导致运行时间过长。因此，多年来，各种实现

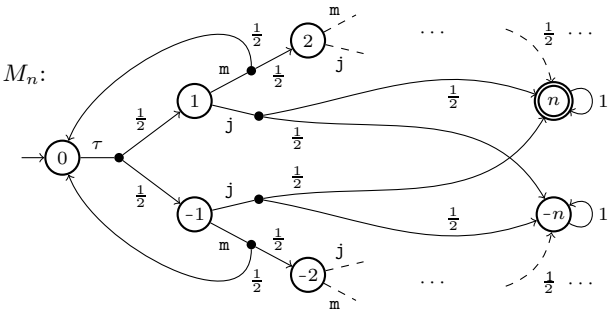


图1：适用于所有算法的困难的马尔可夫决策过程

表1：正确结果

alg.	求解器	$n \leq$
PI	–	20
LP	COPT	18
	CPLEX	18
	Glop	25
	GLPK	24
	Gurobi	18
	HiGHS	22
	lp_求解	28
	Mosek	22
	SoPlex	34

采用了简单的停止准则，可能会返回任意严重错误的结果 [25]。这一问题的发现 [25] 推动了各种可靠的 VI 变体的发展 [8,26,29,33,41,43], ，包括区间迭代、可靠值迭代和乐观值迭代。可靠的 VI 算法可保证结果的精度达到 ε，即$|v-\bar{v}| \leq \epsilon$或 $|v-\bar{v}| \leq v \cdot \epsilon$。对于线性规划和策略迭代，其保证问题尚未得到全面研究。从理论上说，两者都是精确的，但实际实现往往并非如此。我们将在第3 节和4节详细讨论这些问题。

[25, 图 2] 中的手工构造的马尔可夫链凸显了 VI 缺乏保证这一问题：标准实现会返回严重错误的结果。我们为其加入动作选择，得到图 1 中所示的马尔可夫决策过程 M_n ，用于 $n \in \mathbb{N}, n \geq 2$。它有 $2n+1$ 个状态；我们计算 $P_{min}(\{n\})$ 和 $P_{max}(\{n\})$。在所有可能的状态中均选择动作 m 的策略会导出 [25, 图 2] 中的马尔可夫链，其 $(P_{min}(\{n\}), P_{max}(\{n\}))$ 为 $=(\frac{1}{2}, \frac{1}{2})$。在每个满足 $0 < s < n$的状态 s中，我们都加入了动作 j 的选项，该动作会跳转到 n 和 $-n$。由此，所有策略下的（最优）值为 $(\frac{1}{3}, \frac{2}{3})$。对于 VI，除 n外，所有状态的初始值均设为 0；一开始，在所有状态中都选择 j，看起来是针对 P_{max}的最佳策略。随着更新后的值缓慢传播，在某个时刻，m 会逐个状态地成为最优选择。由此，我们在原马尔可夫链的缓慢收敛之上又叠加了一个“具有欺骗性”的决策问题。因此，对于 $n = 20$，Storm 和 mcsta 的 VI 得出了错误结果 (0.247, 0.500)。对于 Storm 的 PI 和各种线性规划求解器，我们在表 1 中展示了它们能够返回 ± 0.01-正确结果时最大的 n 。当 n更大时，除 Glop 和 GLPK 外，PI 和所有线性规划求解器均将 $\approx(\frac{1}{2}, \frac{1}{2})$ 作为正确解；Glop 和 GLPK 则返回 $\approx(0.\overline{3}, 0.\overline{6})$，直到分别在最小值为 $n = 29$和 52 时放弃求解。可靠的值迭代算法和 Storm 的精确算术引擎能够得出 (ϵ-)正确结果，不过对于较大的 n，前者的运行时间过长。所有工具和求解器均使用默认设置。

2.4 优化

值迭代、线性规划和策略迭代都可以从以下优化中受益。

图论算法 可用于对马尔可夫决策过程进行定性分析，即找出值为 0 或（仅针对可达性目标）为 1 的状态。这些定性方法通常比用于定量分析的数值计算快得多。因此，我们总是先应用这些方法，然后仅对剩余的非平凡值状态运行数值算法。

拓扑方法，例如 [16], 不会考虑整个马尔可夫决策过程。相反，我们首先计算强连通分量（SCCs）的拓扑排序[5]，然后分别分析每个强连通分量。这可以提高运行时间，因为我们将问题分解为更小的子问题。可以使用任意求解方法来求解这些子问题。请注意，在考虑无环马尔可夫决策过程时，拓扑方法无需调用求解方法，因为由此得到的值可以立即反向传播。

极大终端分量（最大端组件）的折叠，例如， [12,26], 可将马尔可夫决策过程转换为值等价但结构更简单的形式。折叠最大端组件后，马尔可夫决策过程具有收缩性，即我们几乎必然会到达目标状态或值为零的状态。值迭代算法依赖这一性质来保证收敛 [26,43,29]。对于策略迭代和线性规划，在应用求解方法之前简化图结构可以加快计算速度。

热启动，例如 [21,34], 可以为算法提供合适的初始条件，即我们可以向算法提供一些先验知识，使计算拥有良好的起点。我们通过先运行有限次迭代的值迭代，并利用所得估计值来猜测线性规划中变量的界，或为策略迭代选择一个良好的初始策略，从而实现热启动。更多详情请参见第 3 节和第 4 节。

3 使用线性规划实际求解马尔可夫决策过程

本节讨论基于线性规划的方法，用于求解马尔可夫决策过程中的最优策略问题。据我们所知，这是唯一一种多项式时间方法。我们将讨论各种配置。这些配置由线性规划公式、软件选择及其参数化共同构成。

3.1 如何将马尔可夫决策过程编码为线性规划?

对于目标 $\mathsf{P}_{\max}(\mathsf{T})$，我们针对变量 x_s, $s \in \mathsf{S} \setminus \mathsf{T}$ 构建如下线性规划：

$$\text{minimize} \quad \sum_{s \in \mathsf{S}} x_s \quad \text{s.t. } lb(s) \leq x_s \leq ub(s) \quad \text{and}$$
$$x_s \geq \sum_{s' \in \mathsf{S} \setminus \mathsf{T}} \delta(s,a)(s') \cdot x_{s'} + \sum_{t \in \mathsf{T}} \delta(s,a)(t) \quad \text{for all } s \in \mathsf{S} \setminus \mathsf{T}, a \in \mathsf{A}$$

我们假设 $\mathsf{S} \setminus \mathsf{T}$ 的界为 $lb(s) = 0$ 和 $ub(s) = 1$。该线性规划的唯一解η: $\{ x_s \mid s \in \mathsf{S} \setminus \mathsf{T} \} \to [0,1]$，与所需目标值 $\eta(x_s) = V(s)$ 一致。目标 $\mathsf{P}_{\min}(\mathsf{T})$ 和 $\mathsf{E}^{\mathsf{opt}}(\mathsf{rew})$ 的编码类似：

[5] 集合 $\mathsf{S}' \subseteq \mathsf{S}$ 是一个连通分量，当且仅当对于所有 $s, s' \in \mathsf{S}'$，s 均可从 s' 到达。如果 S' 是一个连通分量，且不存在包含 S' 的更大集合也是连通分量，则称 S' 为强连通分量。

最小化策略需要在线性规划中进行最大化，并反转约束关系。奖励可以作为加法因子添加到右侧。出于实际考虑，可以对线性规划公式进行调整。

界的选择。任何符合唯一解的界都不会改变答案。也就是说，任何 lb 和 ub 满足 $0 \leq lb(s) \leq V(s) \leq ub(s)$ 的取值都会产生可靠编码。虽然这些额外的界是多余的，但它们可能显著缩小搜索空间。我们研究了平凡界（例如，已知所有概率都在 $[0,1]$, 内）、由结构分析得出的界（如 [8], 所述），以及由求解器热启动产生的界。对于后者，若我们已获得值 $V' \leq V$（例如由次优策略得出），那么$V'(s)$ 是值 x_s的下界，这一点尤为重要，因为线性规划的目标是最小化。

唯一动作对应的等式。马尔可夫链，即满足 $|\mathsf{A}| = 1$ 的马尔可夫决策过程，可以使用线性方程组求解。线性规划编码利用单边不等式和目标函数来纳入非确定性选择。我们研究了为所有只有一个动作的状态添加约束。

$$x_s \leq \sum_{s' \in S \setminus T} \delta(s,a)(s') \cdot x_{s'} + \sum_{t \in T} \delta(s,a)(t) \quad \text{for all } s \in S \setminus T \text{ with } \mathsf{A}(s) = \{a\}$$

这些额外的约束可能会触发求解器中的不同优化，例如，某些求解器会使用高斯消元进行变量消元。

更简单的目标。 标准目标可确保解 η 对每个 状态都是最优的，而大多数调用只要求在某些特定状态下达到最优性——通常是初始状态 s_0 或强连通分量的入口状态。在这种情况下，可以简化目标，只优化这些状态的值。这可能会产生多个最优解：就马尔可夫决策过程而言，不再需要优化在最优策略下不会到达的状态的值。

对偶形式的编码。 将线性规划的对偶形式编码很有意义，因为它可用于线性规划的混合整数扩展，例如，用于计算部分可观测马尔可夫决策过程中的策略 [35], 或计算极小反例 [45]。对于线性规划，由于强对偶性，我们考察的求解器中的内部表示几乎等价，而且所有求解器都支持求解原始形式和对偶形式。因此，我们不再进一步考虑如何构造它们。

3.2 如何使用现有求解器求解线性规划?

我们依赖先进的线性规划求解器的性能。许多求解器已经开发出来，并且仍在不断改进，参见 [2] ，其中对通用基准测试进行了最新比较。我们在表2中列出了本研究考虑使用的线性规划求解器。各列分别概述了每种求解器所采用的许可证类型、使用精确运算还是浮点运算、是否支持多线程，以及实现了哪类算法。我们还列出了这些求解器是否可用于本研究采用的两种模型检验器[6]。

[6] Storm 已支持 Gurobi、GLPK 和 Z3。mcsta 已支持 Glop。其他所有求解器接口均已添加。

表2：可用的线性规划求解器（“intr” = 内点法）

求解器	版本	许可证	精确/浮点	并行	算法	mcsta	Storm
CPLEX[7]	22.10	学术	fp	yes	intr + 单纯形法	yes	no
COPT[8]	5.0.5	学术	fp	yes	intr + 单纯形法	yes	no
Gurobi [24]	9.5	学术	fp	yes	intr + 单纯形法	yes	yes
GLPK[9]	4.65	GPL	fp	no	intr + 单纯形法	no	yes
Glop[10]	9.4.1874	Apache	fp	no	仅单纯形法	yes	no
HiGHS[11]	1.2.2	MIT	fp	yes	intr + 单纯形法	yes	no
lp_求解[12]	5.5.2.11	LGPL	fp	no	仅单纯形法	yes	no
Mosek[13]	10.0	学术	fp	yes	intr + 单纯形法	yes	no
SoPlex [23]	6.0.1	学术	both	no	仅限单纯形法	no	yes
Z3 [40]	4.8.13	MIT	精确	no	仅使用单纯形法	no	yes

方法。我们简要介绍可用的方法，详细说明请参见 [11] 。大体而言，线性规划求解器采用两类方法之一。基于单纯形法的方法依赖高效的枢轴运算，以考察可行解单纯形中的顶点。单纯形法既可以按原始方式执行，也可以按对偶方式执行，这会改变算法的推进方向。我们的线性规划公式包含的约束多于变量，通常意味着对偶版本更为合适。内点法（通常是障碍法的一个子类）不需要沿着顶点集合进行。这些方法最坏情况下可能达到多项式时间。通常认为，单纯形法的平均情况性能更佳，但对扰动非常敏感；而内点法的性能则更为稳健。

热启动。 基于线性规划的模型检测可以通过两种热启动方式实现：一是像 [21] 中那样提供一个（可行）基点，二是提供界。不过，前一种方式有各种注意事项和限制，例如必须禁用预处理。因此，如上所述，我们仅通过提供界来进行热启动。

多线程。 我们通常会看到线性规划求解器中的两种并行化方式。一些求解器支持组合 方法，即同时运行不同的方法，并以最先得出结果的方法为准。其他求解器则对内点法和/或单纯形法本身进行并行化。

数值线性规划求解器的保证。 所有线性规划求解器都允许调整各种参数，包括 容差 ，以分别控制一个点是否被视为可行或最优。不过，表 1 中的实验已经表明

[7] https://www.ibm.com/analytics/cplex-optimizer [8] https://www.shanshu.ai/copt [9] https://www.gnu.org/software/glpk/ [10] https://developers.google.com/optimization/lp [11] https://www.maths.ed.ac.uk/hall/HiGHS/ [12] https://lpsolve.sourceforge.net/5.5/ [13] https://www.mosek.com/

这些保证并非 绝对。有限的实验表明，将这些容差逐渐减小至零确实消除了部分错误结果，但并未消除全部错误结果。

精确求解。 SoPlex 支持精确计算，并通过 Boost 库封装 GMP 有理数[14]，，同时利用浮点运算 [22]。虽然这在大多数情况下有利于提升性能，但对于数值计算具有挑战性的模型，似乎会引发错误。 Z3 仅支持精确运算（也通过其自有接口封装 GMP 数）。我们发现，转换较大的有理数可能代价高昂。此外，像 Z3 这样的 SMT 求解器采用单纯形法变体 [17]，该变体专为寻找可行点而设计，并以增量方式运行，针对具有复杂布尔结构的问题进行了优化。相比之下，我们的线性规划公式很容易满足，且是纯合取式。

4 可靠策略迭代

从初始策略出发，基于策略迭代的算法会根据诱导马尔可夫链得到的值，迭代改进策略。求解诱导马尔可夫链的算法会显著影响整体方法的性能和准确性。本节介绍 Storm 中可用的求解器、可能存在的精度问题，以及如何利用热启动；第 5 节将讨论策略迭代的性能[15]。

马尔可夫链求解器。 为了求解诱导马尔可夫链，Storm 可以使用 [31] 中列出的所有线性方程求解器，以及 Storm 实现的所有值迭代变体。在我们的实验中，我们考虑：(i) GMM++[16] 中实现的广义最小残量法（GM- RES） [44]，(ii) 采用标准（相对）终止准则的值迭代（VI） [14]，(iii) 乐观值迭代（OVI） [29]，，以及 (iv) Eigen[17] 中实现的稀疏 LU 分解，采用浮点运算或精确运算（LU^{X}）；LU 和 LU^{X} 可提供精确结果（LU 中的浮点错误除外），而 OVI 可给出 ε-精确的结果。值迭代和 GMRES 均不提供任何保证。

策略迭代的正确性。 策略迭代的准确性会受到马尔可夫链求解器的影响：首先，策略迭代的精度无法高于其底层求解器的精度；策略迭代的结果与最终马尔可夫链的求解结果具有相同精度。其次，求解器产生的不准确结果可能掩盖策略改进，从而导致策略以次优策略过早收敛。下面的示例表明，策略迭代可能返回任意错误的结果——即使中间结果达到 ε-精度。

考虑图 2 中目标为 $\text{P}_{\max}(\{G\})$ 的马尔可夫决策过程。这里只有一个非确定性选择，即状态 s_0 中的选择。最优策略是选择 b，从而得到 0.5 的值。选择 a 只能得到 0.1。然而，从初始策略 $\pi(s_0) = \text{a}$ 出发，一个 ε-精确的马尔可夫链求解器可能会对 s_0 和 s_1 都返回 $0.1 + \varepsilon$，并对 s_2 返回 $\delta/2 + (1-\delta)\cdot 0.1$。这个解

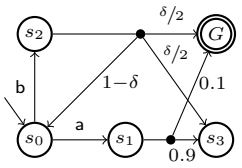


图 2：马尔可夫决策过程示例

[14] https://gmplib.org/ [15] [34] 讨论随机博弈中策略迭代的性能。 [16] https://getfem.org/gmm.html [17] https://eigen.tuxfamily.org/index.php

确实是 ε-精确的。然而，在评估 s_0 中应选择哪个动作时，我们可以选择 δ，使得 a 看起来能得到更高的值。具体而言，我们要求$\delta/2 + (1 - \delta) \cdot 0.1 < 0.1 + \varepsilon$。对于每个 $\varepsilon > 0$，都可以通过设置$\delta < 2.5 \cdot \varepsilon$来实现。在这种情况下，策略迭代会以最终策略终止，而该策略导致的值远非最优。

如果每条马尔可夫链都能精确求解，那么策略迭代就是正确的。事实上，只需确定某个动作优于所有其他动作即可。这正是 [42, 第 6.5 章和第 7.2 章6]中所述的修正策略迭代的核心。同样，[34,第4节2]建议在求解当前策略所诱导的系统时使用区间迭代，并在某个动作的下界高于所有其他动作的上界时停止。

热启动。策略迭代受益于提供一个良好的初始策略。如果初始策略已经是最优策略，策略迭代只需一次迭代便会终止。我们可以利用值迭代计算出的所有状态的估计值，来指导初始策略的选择。对于每个状态，我们选择根据估计值判定为最优的动作。这样既能充分利用值迭代快速给出良好估计值的能力 [29], ，又能同时获得策略迭代的精确性保证。

5 实验评估

为了了解不同算法的实际性能，我们进行了广泛的实验评估。我们使用了三组基准测试：定量验证基准测试集（QVBS） [30] （$qvbs$ 集）中的所有适用基准测试实例[18] 、一部分较难的 QVBS 实例（$hard$集），以及来自某项运行时监控应用的数值挑战性模型[32] （$premise$ 集，该名称源自相应的原型）。我们考察了两个概率模型检测器：Storm [31] 和 Modest 工具集的 [28] mcsta。我们使用运行 64 位 CentOS Linux 7.9 的英特尔至强白金 8160 系统；除非另有说明，每项实验分配 4 个 CPU 核心和 32GB 内存。

我们以秒为单位绘制算法运行时间，使用分位数图（如图散点图（如图3右侧）进行展示。前者用于比较多个工具或配置；对于每一种工具或配置，我们按运行时间对实例排序，并绘制相应的单调递增曲线。这里，点 $(x,\ y)$ 位于 a 线上，表示由 a 求解的第 x快的实例耗时 y 秒。后者用于比较两个工具或配置。每个点 $(x,\ y)$ 对应一个基准测试实例：横轴所示工具耗时 x，纵轴所示工具耗时 y 秒将其求解。点的形状表示模型类型；形状与类型的对应关系在所有散点图中均相同，且仅在图3的第一幅图中明确给出。本节各项论述的补充图表见附录。

图中所示的运行时间包括相应算法以及所有必要和/或注明的预处理所需的时间，但不包括构造 MDP 状态空间的时间（这与算法无关）。mcsta 报告的所有时间测量值均四舍五入到 0.1 秒的倍数。我们将超时、内存不足、错误和错误结果统一记为“n/a”。算法的超时限为 30 分钟，而包括 MDP 构造在内的总运行时间超时限为 45 分钟。

[18] 一个基准测试实例 由模型、参数赋值和目标组合而成。

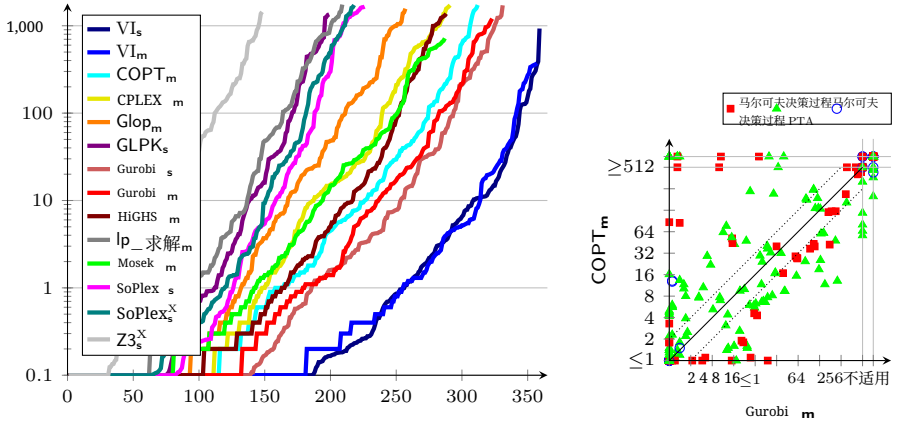


图 3：*qvbs* 集上的线性规划求解器运行时间比较

当存在参考结果 v 时，若 $|v-\bar{v}| > v \cdot 10^{-3}$（即相对误差 10^{-3}），我们就认为结果 $\bar{v}$不正确。不过，如果v 和 $\bar{v}$ 都低于 10^{-8}，我们不会将结果标记为不正确（这与前提集合有关）。尽管如此，我们仍将 VI 的（不可靠）收敛阈值设置为相对 10^{-6}；在可靠的 VI 算法中，我们纳入了 OVI，其停止准则为相对 10^{-6} 误差。因而，若只需达到我们实际测试的 10^{-3} 精度，OVI 的速度可能比图中显示的还要快。我们作此区分，是因为许多算法（包括线性规划求解器）都没有可靠的误差准则。我们用上标x标记使用有理数算术的精确算法/求解器。其他配置则使用浮点数（fp）。

5.1 QVBS基准测试

该 *qvbs* 集包含所有具有 MDP、马尔可夫自动机（MA）或概率定时自动机（PTA）模型的 QVBS 基准测试实例[19] 以及定量的可达性或期望奖励/时间目标，即不包含产生概率 0 或 1 的查询。我们仅考虑以下实例：线性规划求解器 Storm 和 mcsta 均能在 15 分钟内构建 MDP 的显式表示。由此得到 367 个实例。对于其中 344 个实例，我们从 QVBS 数据库或使用 Storm 的精确方法获得了参考结果。不同方法得到的参考结果始终一致。

对于线性规划求解器，我们有多种选择，每种求解器也有各自不同的参数，参见第 3 节。为简洁起见，我们首先比较在 *qvbs* 集合上所有可用的线性规划求解器。然后，我们评估表现最佳的求解器在采用不同配置时的效果。对于策略迭代中的马尔可夫链求解方法，我们也进行同样的比较。接下来，我们分别针对线性规划求解器和策略迭代，重点考察各自单一且合理的配置。

[19] MA 和 PTA 通过嵌入和数字时钟转换为 MDP [36]。

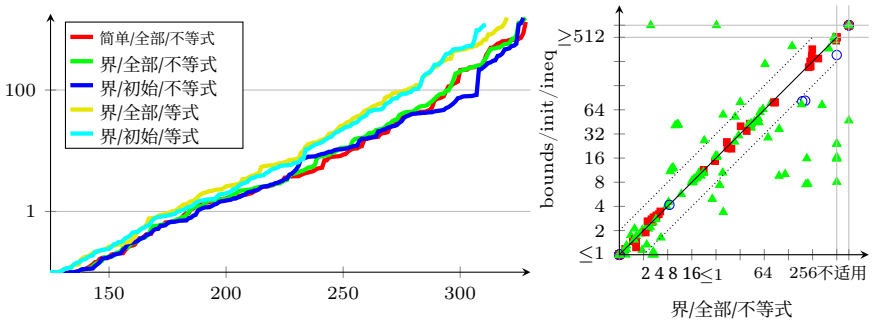


图 4：线性规划问题表述变体的性能影响（使用 Gurobi$_s$）

线性规划求解器比较。 图 3 左侧的图汇总了不同线性规划求解器的比较结果。下标 $_s$ 和 $_m$表示求解器分别嵌入在Storm 或 mcsta中。除预计算概率为0的状态（以及在 Storm 中还预计算概率为1的状态）外，我们不对马尔可夫决策过程进行任何优化或约简；所有求解器均采用默认设置，概率和期望奖励的平凡变量界分别为 $[0,1]$ 和 $[0,\infty)$。我们将值迭代作为基准。在表 3 中汇总了结果。

表3：线性规划摘要

求解器	正确	不正确	无结果
VI$_s$	359	8	0
VI$_m$	357	8	2
COPT$_m$	312	12	43
CPLEX $_m$	291	10	66
Glop$_m$	257	4	106
GLPK$_s$	199	5	163
Gurobi $_s$	331	4	32
Gurobi$_m$	323	4	40
HiGHS $_m$	288	10	69
lp_求解 $_m$	209	0	158
Mosek$_m$	287	15	65
SoPlex $_s$	226	9	132
SoPlex^X_s	218	0	149
Z3^X_s	148	0	219

在**性能** 和可扩展性方面，Gurobi 求解的基准测试数量最多
在任何给定的时间预算内，紧随其后的是 COPT。 CPLEX、HiGHS，而 Mosek则构成中游梯队。尽管精确求解器 Z3 速度非常慢，SoPlex的精确模式实际上可以与一些浮点求解器相媲美。分位数图并不能展现全部情况。在图 3的右侧，我们比较了 COPT 和 Gurobi，它们各自在大量实例上的表现都（远）胜过对方。

就结果的**可靠性** 而言，正如预期，精确求解器不会产生错误结果；速度最慢的浮点求解器 lp_solve 也是如此。COPT、CPLEX、HiGHS、Mosek 和浮点 SoPlex 在这一指标上的表现较差，产生的错误比值迭代更多。有趣的是，这些求解器大多速度更快，Gurobi 是个例外。

总体而言，Gurobi 在可靠性尚可的情况下实现了最高性能；因此，在本节余下部分中，每当我们使用非精确线性规划时，都会采用 Gurobi$_s$ 。

线性规划求解器调优。Gurobi 可以配置为使用“自动”组合方法，可能会在多个线程上并发运行多种算法，例如原始单纯形算法、对偶单纯形算法或障碍法算法。我们使用 4 个线程比较了每种选项，发现性能没有显著差异。同样，使用 1、4 和 16 个线程运行自动 方法（仅在此处，我们为每次实验分配 16 个线程），也未能带来明显的性能差异

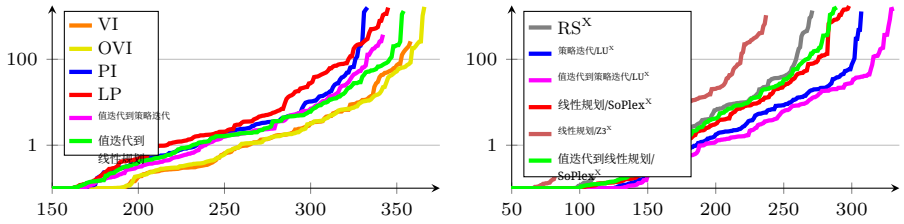
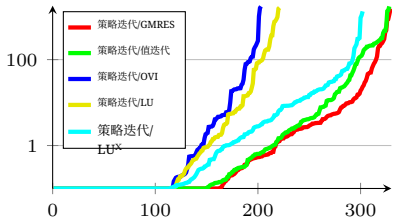


图 5：在*qvbs* 集合上对比马尔可夫决策过程模型检测算法

。不过，使用更多线程会导致更多内存不足错误。因此，我们将 Gurobi 固定为自动 模式，并使用 4 个线程。

图4 展示了使用 Gurobi 求解期望奖励目标时，为变量提供更精确的界所带来的性能影响；这些方法包括 [8,39]（使用“bounds”而非“simple”）、仅针对初始状态进行优化（“init”），而非对所有状态求和（“all”），以及对唯一动作状态使用等式（“eq”），而非大于或等于/小于或等于约束（“ineq”）。更精确的界几乎没有成本，却带来了非常小的性能提升。仅针对初始状态进行优化，总体性能会稍好一些（在分位数图中，$x = 375$ 附近的“口袋”区域尤为明显，在散点图中也清晰可见）。不过，这也使 *qvbs* 集合中错误结果增加了 2 个。对唯一动作使用等式会明显降低性能，并使错误结果增加 9 个实例。因此，在后续所有实验中，我们都使用更精确的界，但不启用另外两项优化。

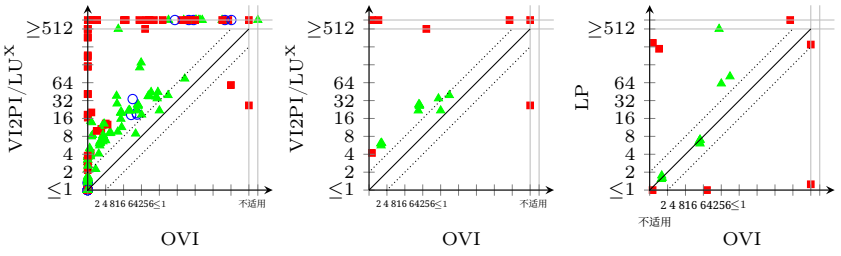
策略迭代方法比较。策略迭代中的主要选择是使用哪种算法来求解诱导出的马尔可夫链。右侧展示了Storm （参见第4节）中可用的不同算法的性能。LU^X 可实现完全精确的策略迭代。有趣的是，其性能优于浮点



版本，可能是因为浮点误差会引发虚假的策略变化。同样的影响可能也会妨碍 OVI 的使用，而 VI 则表现良好。尽管如此，GMRES 的整体表现最佳，因此我们在后续所有采用非精确策略迭代的实验中都选择了它。VI 和 GMRES 分别产生了 6 个和 4 个错误结果。对于这组基准测试，OVI 和精确方法始终给出正确结果。

QVBS 的最佳 *MDP* 算法。现在，我们比较 *QVBS* 测试集上的所有马尔可夫决策过程模型检测算法：使用浮点数，并按照上述方式配置线性规划和策略迭代；此外还包括不可靠的值迭代、可靠的 OVI，以及策略迭代和线性规划的热启动变体，分别记作“值迭代到策略迭代”和“值迭代到线性规划”。精确结果由有理数搜索（RS，本质上是 VI 的精确版本）、采用精确 LU 的 [38], PI，以及使用精确求解器（SoPlex 和 Z3）的线性规划提供。所有算法均在 Storm 中实现。

在第一项实验中，我们评估了采用拓扑方法以及折叠最大端组件（参见第 2.4节）的影响。对于实验结果，我们省略了



(a) *QVBS* OVI 与值迭代到策略迭代对比^{X} (b) 困难 OVI 与值迭代到策略迭代对比^{X} (c) 困难 OVI 与线性规划

图 6：其他直接性能比较

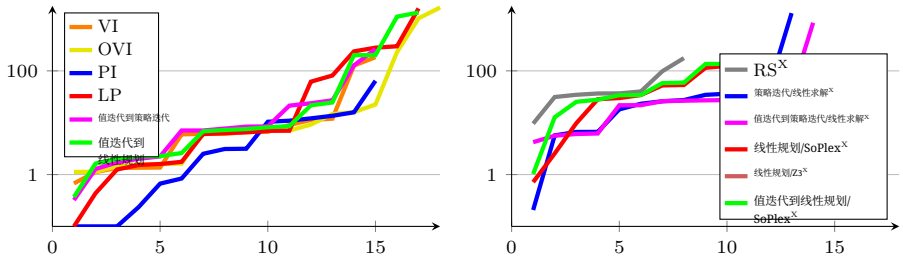


图 7：硬子集上的马尔可夫决策过程模型检测算法比较

图表。结果表明，拓扑方法显著提升了所有 算法的性能和可扩展性，因此从现在起我们始终采用这种方法。折叠最大端组件是保证 OVI 终止所必需的；对于其他算法而言，这是一种潜在的优化。不过，我们发现总体而言，它带来的性能提升微乎其微。由于 OVI 需要这一操作，且它不会降低性能，因此从现在起我们也始终采用这种操作。

图5 展示了在整个*qvbs* 数据集上所有方法的完整比较：左侧为浮点算法，右侧为精确解。在浮点算法中，OVI 显然速度最快、可扩展性最好。值迭代稍快一些，但会产生若干错误结果，因此在分位数图中的表现有所逊色。此外，OVI 在这些算法中也很特别，因为它是可靠的，即能提供有保证的 ϵ正确结果——尽管存在浮点舍入误差，但按照 [27]中的方法可以消除这些误差。在精确解方面，采用非精确值迭代热启动的策略迭代表现最佳。图6(a) 展示了计算精确解而非近似解对性能的影响。

5.2 困难的 QVBS基准测试

QVBS 包含许多为默认使用值迭代算法的工具构建的模型。其他算法实际上可能对于求解那些 VI/OVI 表现不佳的棘手实例十分重要。对于值迭代而言，这些简单实例数量众多，可能会掩盖其他算法的贡献。因此，我们将重点考察一组选出的 QVBS

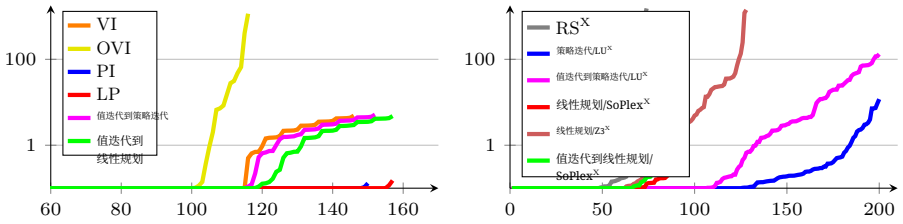


图 8：MDP 模型检测算法在前提 集上的比较

实例：这些实例对值迭代来说似乎“很难”，也就是说，在 Storm 和 mcsta 中，值迭代耗时都超过此前的马尔可夫决策过程状态空间构建阶段，且两个阶段合计耗时至少为 1 秒。这些实例占此前考察的 367 个实例中的 18 个。

在图 7中，我们展示了所有算法在该困难子集上的表现。由于值迭代返回了错误结果，OVI 再次优于值迭代。我们看到，算法之间在性能和可扩展性方面的差距已经缩小；尽管 OVI 仍然“胜出”，但线性规划尤其接近其表现，远胜于在完整的 *QVBS* 集上的情况。我们还研究了使用Gurobi以外的求解器求解线性规划的结果：Gurobi 和 COPT 仍然是速度最快、可扩展性最强的求解器。使用 mcsta时，在基本配置下，它们分别求解了 16 和 17 个实例，最慢的分别耗时 835 秒和 1334 秒；使用 topo 时，求解的实例数分别变为 17 和 15 个，最慢的分别耗时 1373 秒和 1590 秒。我们展示了 OVI 和线性规划的详细比较，并指出有少数实例中线性规划快得多（图 6(c)），并重复比较表现最佳的浮点算法和精确算法（图 6(b)）。

5.3 运行时监测基准测试

虽然 QVBS 刻意保持多样性，但我们的第三组基准测试则有意聚焦于一个方向：我们研究了运行时监测研究中的 200 个马尔可夫决策过程 [32]。原始问题是在随机噪声、不可观测且不可控的非确定性以及部分状态观测的情况下，计算继续运行受监控系统的归一化风险。这是一个条件概率查询。按照 Baier 等人的方法$n \in \{50, \ldots, 1000\}$，通过沿长度为[7]的观测历史轨迹展开马尔可夫决策过程模型，利用概率模型检测来回答这一查询。这些马尔可夫决策过程中有许多转移会返回初始状态，最终导致数值计算颇具挑战的实例（参见第 M_n 2.3节中的结构）。我们为所有实例都计算出了参考结果。

图8 比较了这组实例上的不同马尔可夫决策过程模型检测算法。与 [32], 中的观察一致，我们发现其表现与 QVBS 中的情况大不相同。在左侧的浮点解中，使用 Gurobi 的线性规划会很快终止（不到 1 秒），并且要么给出正确结果（155 个实例），要么给出完全错误的结果（大多为 0，涉及 45 个实例）。值迭代的表现类似，但速度较慢。

相比之下，OVI 不会给出错误结果，但在除 116 个以外的所有实例上都无法终止。在精确设置下，使用值迭代进行热启动会继承其相对较慢的速度，因此并无助益。精确策略迭代优于两种精确线性规划求解器。对于精确 SoPlex，在它无法求解的 112 个实例中，有 98 个是错误，可能与当前版本中已确认的错误有关。

前提集表明，最佳的马尔可夫决策过程模型检测算法取决于具体应用。在这里，对于浮点情形，线性规划似乎表现最佳，但会产生不可靠（错误）的结果；而看似差得多的 OVI 至少不会如此。鉴于这一数值挑战，应选择精确方法，而我们表明，精确方法在此确实表现良好。

6 结论

我们全面考察了马尔可夫决策过程模型检测领域的最新研究进展，表明这项任务不存在单一的最佳算法。对于数值挑战不大的基准测试，OVI 是一个合理的默认选择，其次是策略迭代和采用热启动的线性规划——不过，使用后两者意味着会牺牲可靠性，这一点已由我们实验中的若干错误结果证实。对于数值上较难的基准测试，策略迭代、线性规划以及计算精确解更具吸引力，而将它们结合起来显然更为可取。总体而言，尽管线性规划具有更优的（多项式）理论复杂度，但在我们的实际评估中，它几乎总是逊于其他（指数级）方法。即便我们使用了现代商业求解器，并对问题的线性规划编码以及求解器参数进行了调优，情况依然如此。虽然我们观察到了不同算法的行为，也对哪些因素会使前提集变得困难有了一些直觉，但识别并量化导致模型求解困难的结构性质，本身就是一个完整的研究课题。

我们的评估还提出了一个问题：在实际应用中，挑战值迭代（VI）的马尔可夫决策过程有多常见。除了 *premise* 基准测试之外，我们未能找到其他难以用值迭代求解的马尔可夫决策过程集合。值得注意的是，[34]中发现了若干难以用值迭代求解的随机博弈（SG）；作者指出，对这些随机博弈使用策略迭代（PI）优于使用值迭代。不过，在提取出诱导的马尔可夫决策过程后，我们发现它们都很容易用值迭代求解。同样，[3] 采用随机生成的方法生成了至多包含 10,000 个状态的随机博弈，其中许多对随机博弈算法而言颇具挑战性。然而，对该随机生成方法稍作修改以生成马尔可夫决策过程后，即使状态数量大幅增加，生成的马尔可夫决策过程也都能在几秒内轻松求解。相比之下，Alagöz 等人 [1] 报告称，他们的随机生成方法生成了一些模型，在这些模型上，线性规划优于策略迭代。不过，他们研究的是折扣情形，而且对随机生成方法的描述过于简略，我们无法据此复现。我们注意到，在若干散点图中，QVBS 中的 MA 实例（我们在其中检验嵌入式马尔可夫决策过程）整体上似乎比马尔可夫决策过程更具挑战性。因此，本文最后呼吁构建具有挑战性的马尔可夫决策过程基准测试——既可以建立具有独特特征的独立基准测试集，例如 *premise*，也可以将其纳入 QVBS。

参考文献

1. Alagöz, O., Ayvaci, M.U.S., Linderoth, J.T.：利用总期望折扣奖励函数最优求解马尔可夫决策过程：重新审视线性规划。计算机与工业工程 87，311–316（2015）。https://doi.org/10 031 2 1016 2015 05 . /j.cie. . . . Anand, R., Aggarwal, D., Kumar, V.：优化求解器的比较分析。统计与管理系统杂志 20(4)，623–635（2017）。https://doi.org/10 1395182 3 1080 09720510 2017 . / . . . Azeem, M., Evangelidis, A., Kretínský, J., Slivinskiy, A., Weininger, M.：用于简单随机博弈的乐观值迭代与拓扑值迭代。CoRR abs/2207.14417（2022）。https://doi.org/10.48550/arXiv.2207.14417, https://doi.org/10 14417 4 48550 2207 . /arXiv. . . Baier, C., de Alfaro, L., Forejt, V., Kwiatkowska, M.：概率系统的模型检测。载于：模型检验手册，第 963–999 页。Springer（2018） 5. Baier, C., Hermanns, H., Katoen, J.：MDP 模型检验的一万种面向。载于：计算与软件科学，计算机科学讲义，第 10000 卷，第 420–451 页。Springer（2019） 6. Baier, C., Katoen, J.：模型检验原理。MIT出版社（2008），https://mitpress.mit.edu/books/principles-model-checking 7. Baier, C., Klein, J., Klüppelholz, S., Märcker, S.：高效计算马尔可夫模型中的条件概率。载于：TACAS。计算机科学讲义，第 8413 卷，第 515–530 页。Springer（ 2014 10 1007 978 3 642 54862 8_43 ）。
8 https://doi.org/ . / - - - - . Baier, C., Klein, J., Leuschner, L., Parker, D., Wunderlich, S.：确保模型检验器的可靠性：马尔可夫决策过程的区间迭代。载于：计算机辅助验证（1）。计算机科学讲义，第 10426 卷，第 160–180 页。Springer（2017）。https://doi.org/10.1007/978-3-319-63387-9_8, https://doi.org/ 10 1007 978 3 319 63387 9_8 . / - - - - . Balaji, N., Kiefer, S.,
9 Novotný, P., Pérez, G.A., Shirmohammadi, M.：值迭代的复杂度。载于：ICALP。LIPIcs，第 132 卷，第 102:1–102:15 页。达格斯图尔城堡—莱布尼茨信息学中心（2019）。https://doi.org/10.4230/LIPIcs.ICALP.2019.102, https://doi.org/10 102 10 4230 2019 . /LIPIcs.ICALP. . . Bertsekas, D.P., Tsitsiklis, J.N.：随机最短路径问题分析。数学运筹研究 16(3)，580–595（1991）。https://doi.org/10.1287/moor.16.3.580, https://doi.org/10 580 11 1287 16 3 . /moor. . . . Boyd, S.P., Vandenberghe, L.：凸优化。剑桥大学出版社（2014） 12. Brázdil, T., Chatterjee, K., Chmelik, M., Forejt, V., Kretínský, J., Kwiatkowska, M.Z., Parker, D., Ujma, M.：使用学习算法验证马尔可夫决策过程。载于：ATVA。计算机科学讲义，第 8837 卷，第 98–114 页。Springer（2014）。https://doi.org/10.1007/978-3-319-11936-6_8, https://doi.org/ 10 1007 978 3 319 11936 6_8 . / - - - - . Budde, C.E., Hartmanns,
13 A., Klauck, M., Kretínský, J., Parker, D., Quatmann, T., Turrini, A., Zhang, Z.：定量验证中的正确性、精度与性能——QComp 2020竞赛报告。载于：ISoLA（4）。计算机科学讲义，第 12479 卷，第 216–241 页。Springer（2020） 14. Chatterjee, K., Henzinger, T.A.：值迭代。载于：模型检测 25 年。计算机科学讲义，第 5000 卷，第 107–138 页。Springer（2008）。https://doi.org/10.1007/978-3-540-69850-0_7, https://doi.org/10.1007/978-3-540-69850-0_7

15. Cubuktepe, M., Jansen, N., Junges, S., Katoen, J., Topcu, U.：马尔可夫决策过程参数综合的凸优化。IEEE Trans. Autom. Control.（2022）。https://doi.org/10 3133265 16 1109 2021 . / TAC. . . Dai, P., Mausam, Weld, D.S., Goldsmith, J.：拓扑值迭代算法。J. Artif. Intell. Res. 42, 181–209（2011），https://www.jair.org/index.php/jair/article/view/10725 17. Dutertre, B., de Moura, L.M.：一种用于 DPLL(T) 的快速线性算术求解器。载于：计算机辅助验证。计算机科学讲义，第4144卷，第81–94页。Springer（2006）18. Fearnley, J.：策略迭代的指数级下界。载于：ICALP (2)。计算机科学讲义，第6199卷，第551–562页。Springer（2010）19. Forejt, V., Kwiatkowska, M.Z., Parker, D.：概率模型检测的帕累托曲线。载于：ATVA。计算机科学讲义，第7561卷，第317–332页。Springer（2012）。https://doi.org/10.1007/978-3-642-33386-6_25, https://doi.org/ 10 1007 978 3 642 33386 6_25 . / - - - - . Funke, F., Jantsch, S., Baier, C.：概率可达性约束的 Farkas证书与最小见证。载于：TACAS (1)。计算机科学讲义，第12078卷，第324–345页。Springer（2020）21. Giro, S.：最优调度器与最优基：一种高效精确求解马尔可夫决策过程的方法。Theor. Comput. Sci. 538, 70–83（2014）。https://doi.org/10.1016/j.tcs.2013.08.020，https://doi.org/10 020 22 1016 2013 08 . /j.tcs. . . . Gleixner, A.M., Steffy, D.E., Wolter, K.：通过迭代细化提高线性规划求解器的精度。载于：符号与代数计算国际研讨会。第187–194页。ACM（2012）23. Gleixner, A.M., Steffy, D.E., Wolter, K.：线性规划的迭代细化。技术报告3，柏林扎伊布尼茨研究所，Takustr. 7, 14195 柏林（2016）。https://doi.org/10 0692 24 1287 2016 0692, 10 1287 2016 . /ijoc. . https://doi.org/ . / ijoc. . . Gurobi Optimization, LLC：Gurobi优化器参考手册（2022），https://www.gurobi.com 25. Haddad, S., Monmege, B.：马尔可夫决策过程中的可达性：改进值迭代的收敛性。载于：随机与组合优化国际会议。计算机科学讲义，第8762卷，第125–137页。Springer（2014）26. Haddad, S., Monmege, B.：适用于马尔可夫决策过程和区间马尔可夫决策过程的区间迭代算法。Theor. Comput. Sci. 735, 111–131（2018）。https://doi.org/10.1016/j.tcs.2016.12.003，https://doi.org/10 003 27 1016 2016 12 . /j.tcs. . . . Hartmanns, A.：使用浮点运算进行正确的概率模型检测。载于：TACAS (2)。计算机科学讲义，第13244卷，第41–59页。Springer（2022）。https://doi.org/10.1007/978-3-030-99527-0_3, https://doi.org/ 10 1007 978 3 030 99527 0_3 . / - - - - . Hartmanns, A., Hermanns, H.：Modest工具集：用于定量建模与验证的集成环境。载于：TACAS。计算机科学讲义，第8413卷，第593–598页。Springer（ 2014 10 1007 978 3 642 54862 8_51 ）。https://doi.org/ . / - - - - . Hartmanns, A., Kaminski, B.L.：乐观值迭代。载于：计算机辅助验证 (2)。计算机科学讲义，第12225卷，第488–511页。Springer（2020）。https://doi.org/10.1007/978-3-030-53291-8_26, https://doi.org/ 10 1007 978 3 030 53291 8_26 . / - - - - . Hartmanns, A., Klauck, M., Parker, D., Quatmann, T., Ruijters, E.：定量验证基准集。载于：TACAS。计算机科学讲义，第11427卷，第344–350页。Springer（ 2019 10 1007 978 3 030 17462 0_20 ）。https://doi.org/ . / - - - - . Hensel, C., Junges, S., Katoen, J.P., Quatmann, T., Volk, M.：概率模型检测器 Storm。Int. J. Softw. Tools Technol. Transf. 24(4), 589–610（2022）。https://doi.org/10.1007/s10009-021-00633-z

32. Junges, S., Torfah, H., Seshia, S.A.：马尔可夫决策过程的运行时监视器。载于：计算机辅助验证 (2)。计算机科学讲义，第12760卷，第553–576页。Springer（2021）33. Kelmendi, E., Krämer, J., Kretínský, J., Weininger, M.：简单随机博弈的值迭代：停止准则与学习算法。载于：计算机辅助验证 (1)。计算机科学讲义，第10981卷，第623–642页。Springer（2018）34. Kretinsky, J., Ramneantu, E., Slivinskiy, A., Weininger, M.：简单随机博弈算法的比较。信息与计算（2022），https://doi.org/10 104885 35 1016 2022 . /j.ic. . . Kumar, A., Zilberstein, S.：部分可观测马尔可夫决策过程的基于历史的控制器设计与优化。载于：人工智能规划与调度国际会议。第25卷，第156–164页（2015）36. Kwiatkowska, M.Z., Norman, G., Parker, D., Sproston, J.：使用数字时钟分析概率定时自动机的性能。形式化方法与系统设计 29 2 37 1 33 78 2006 10 1007 10703 006 0005 ( ), – ( ). https://doi.org/ . /s - - - . Littman, M.L., Dean, T.L., Kaelbling, L.P.：求解马尔可夫决策问题的复杂度。载于：不确定性人工智能。第394–402页。Morgan Kaufmann（1995）38. Mathur, U., Bauer, M.S., Chadha, R., Sistla, A.P., Viswanathan, M.：通过有理数搜索实现精确的定量概率模型检测。形式化方法与系统设计 56(1), 90–126（2020）39. McMahan, H.B., Likhachev, M., Gordon, G.J.：有界实时动态规划：具有单调上界和性能保证的实时动态规划。载于：国际机器学习会议。ACM国际会议论文集系列，第119卷，第569–576页。ACM（2005 1102423 40 10 1145 1102351 3 ）。https://doi.org/ . / . . de Moura, L.M., Bjørner, N.S.：Z：一种高效的可满足性模理论求解器。载于：TACAS。计算机科学讲义，第4963卷，第337–340页。Springer（2008）。https://doi.org/10.1007/978-3-540-78800-3_24, https://doi.org/ 10 1007 978 3 540 78800 3_24 . / - - - - . Phalakarn, K., Takisaka, T., Haas, T., Hasuo, I.：随机博弈有界值迭代中的最宽路径与全局传播。载于：计算机辅助验证 (2)。计算机科学讲义，第12225卷，第349–371页。Springer（2020）42. Puterman, M.L.：马尔可夫决策过程：离散随机动态规划。Wiley概率与统计丛书，Wiley（1994）。https://doi.org/10 9780470316887 43 1002 9780470316887, 10 1002 1 . / https://doi.org/ . / . Quatmann, T., Katoen, J.：可靠值迭代。载于：计算机辅助验证 ( )。计算机科学讲义，第10981卷，第 643 661 2018 10 1007 978 3 319 96145 3_37 –页。Springer ( )。https://doi.org/ . / - - - - . Saad, Y., Schultz, M.H.：GMRES：一种用于求解非对称线性方程组的广义最小残量算法。Siam Journal on Scientific and Statistical Computing 7, 856 0907058 45 869 1986 10 1137 – ( )，https://epubs.siam.org/doi/ . / . Wimmer, R., Jansen, N., Vorpahl, A., Ábrahám, E., Katoen, J., Becker, B.：概率自动机的高层反例。逻辑方法与计算机科学 11(1)（2015）46. Wimmer, R., Kortus, A., Herbstritt, M., Becker, B.：概率模型检测与结果可靠性。载于：设计与诊断电子电路与系统国际会议。第207–212页。IEEE计算机学会（2008）。https://doi.org/10 4538787 47 1109 2008 . /DDECS. . . Ye, Y.：对于具有固定折扣率的马尔可夫决策问题，单纯形法和策略迭代法具有强多项式复杂度。运筹学数学 36(4), 593–603（2011）

A QVBS基准测试：线性规划求解器调优

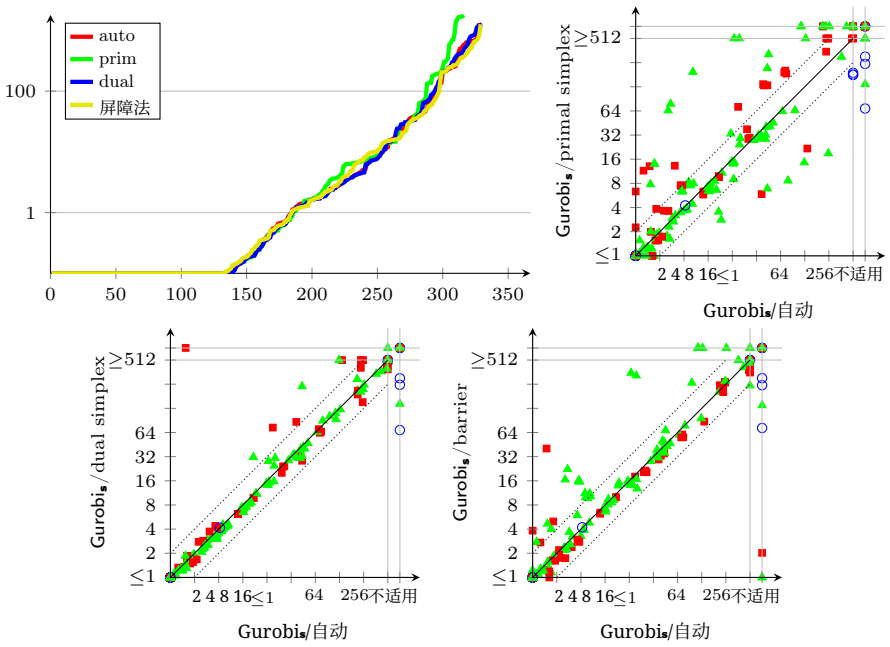


图9：比较Gurobi的配置。结果讨论请参见第线性规划求解器调优节5.1 中的相关段落。

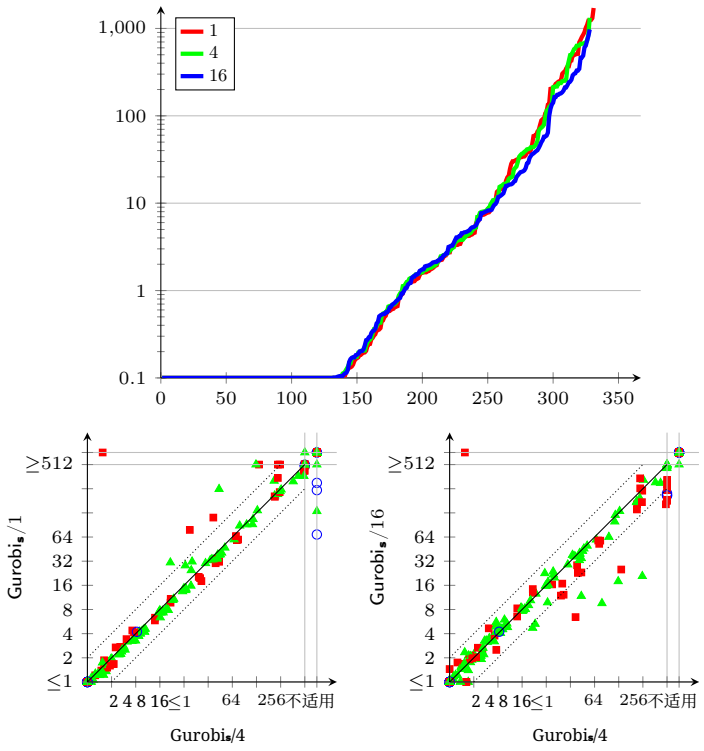


图 10：线程数对Gurobi自动方法性能的影响比较。结果讨论请参见第*5.1* 节中关于线性规划求解器调优的段落。

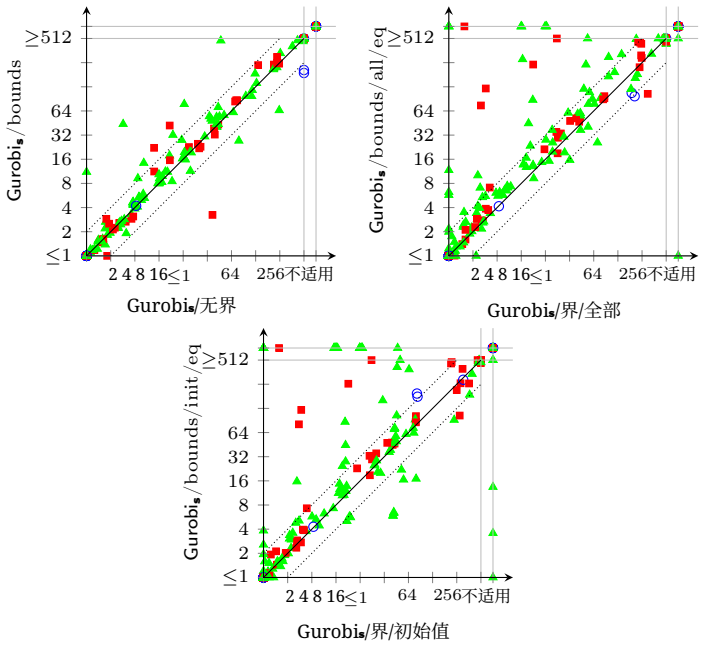


图 11：更多线性规划问题表述变体的比较。描述请参见第 4 节 5.1 中描述图的段落。

B QVBS基准测试：MEC 折叠与拓扑分解

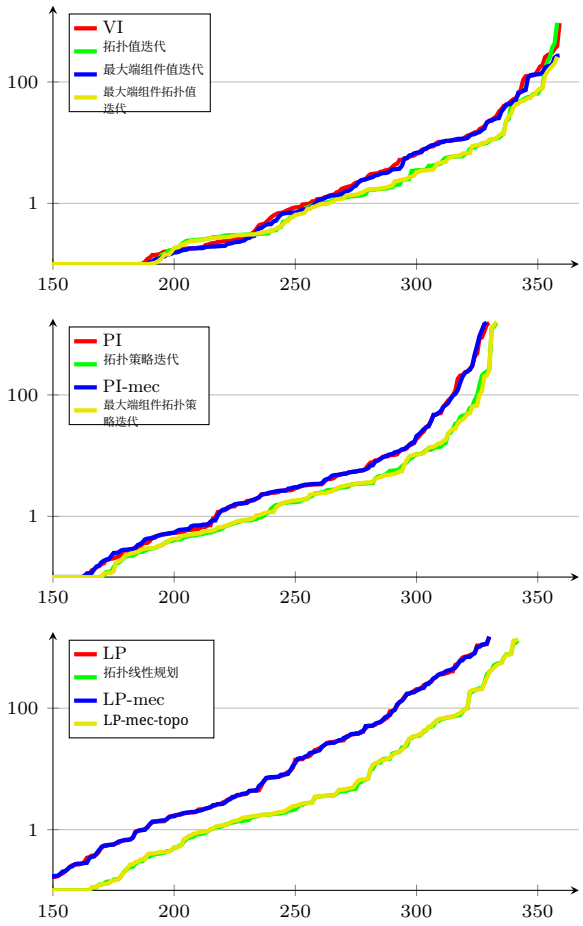


图 12：使用 MEC 折叠和拓扑分解的基础算法 VI、PI 和 LP 及其变体的比较。请参阅段落下方的讨论*QVBS* 的最佳 *MDP* 算法 第 5.1 节，以了解结果的讨论。

C Hardbenchmarks：线性规划求解器运行时间

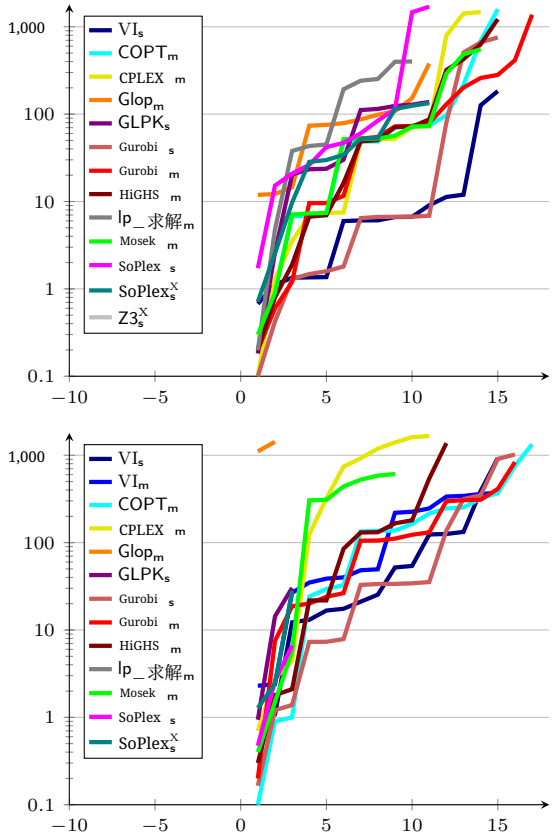


图 13：在困难子集上的线性规划求解器运行时间比较。上图使用拓扑优化，下图未使用。结果讨论请参见第5.2 节。