arXiv:2603.20038v1 [quant-ph] 20 Mar 2026

用于乘积态量子k-SAT 的搜索驱动子句学习（PRODSAT-QSAT）

Samuel Gonz´ález-Castillo JoonHyung Lee Alfons Laarman

2026年3月23日

摘要

我们研究 PRODSAT-QSAT(k)：给定秩一k-局域投影算符，判断量子k-SAT 实例是否存在满足约束的乘积态。我们提出一种 CDCL 式反驳框架，在每个量子比特的布洛赫球上搜索有限划分，同时由可靠的理论求解器通过对每个约束的投影幅度进行几何过近似来检查区域可行性。当理论求解器证明某个区域中的任何状态都无法满足某个约束时，它会生成可靠的冲突子句以阻塞该区域；累积的阻塞子句可得出乘积态不可满足（UN-PRODSAT）的全局结论。我们对该问题进行形式化，证明子句学习规则的可靠性，并介绍一种实用算法及其实现。

目录

1 引言 1 2 几何预备知识 3 3 理论求解器 7 4 主要架构（定理 1 的证明）84.1 PRODSAT 结果的附加假设.13 4.2 关于 SAT 求解器必要性的说明.14**5 实现 15 6 结论 17**

1 引言

量子k-SAT（k-QSAT）要判断是否存在一个n量子比特状态，能被一组局域投影算符 Π_j 消去。等价地，给定一个哈密顿量$H \coloneqq \sum_j \Pi_j$，该问题要判断H 是否无挫折，即其基态能量是否为零。这是经典问题k-SAT 的量子类比，但研究得要少得多。该领域已经取得了一些重要发现。Kitaev [12] 最早研究了这个问题。已证明 2-QSAT 易于求解 [5, 7]。文献 [15], 已证明存在乘积态解的条件，以及寻找这类解的算法，但仍有许多问题有待探索。例如，文献 [15] 给出的算法具有双重

指数复杂度。它也引出了关于乘积态相之外的相的问题，即纠缠态和不可满足。在类比k-SAT 对$k \geq 3$ 为 NP完全的情况下，k-QSAT 对$k \geq 3$ 为 QMA完全，如文献 [9] 所示。

即使在规模较小时，启发式求解器也必须在非凸高维景观中进行搜索。从算法角度来看，在结构化的 ans¨atze 中寻找可满足状态是很自然的。本文聚焦于乘积态 ansatz，它既具有物理意义，又便于算法处理。具体而言，我们旨在利用冲突驱动子句学习（CDCL） [1, 16] 原理，证明有限规模k-QSAT 不存在乘积态解（UN-PRODSAT）。

乘积态定义如下:

$$|\psi\rangle = \bigotimes_{j=1}^{n} |\psi_j\rangle, \qquad |\psi_j\rangle \in \mathbb{C}^2, \ \big\||\psi_j\rangle\big\| = 1. \tag{1}$$

我们针对这一设定开发了一个子句学习框架，用以引导在状态空间中进行二分搜索。核心思路是将单量子比特流形离散化为有限个区域，并让 CDCL SAT 求解器 [1, 16, 17] 搜索这些区域，同时由连续理论求解器利用区间算术检查区域内的可行性。当可行性检查失败时，可以生成一个可靠的冲突子句 。

问题定义。固定n 个量子比特。对于每个约束$j \in [m],$，令$S_j \subseteq [n]$ 为一个子集，满足 $|S_j| = k$，并令 Π_j 为一个支撑在S_j上的秩一投影算子:

$$\Pi_j = |v_j\rangle\langle v_j|, \qquad |v_j\rangle \in (\mathbb{C}^2)^{\otimes k}, \ \big\||v_j\rangle\big\| = 1.$$

对于全局态 $|\psi\rangle \in (\mathbb{C}^2)^{\otimes n}$，令 $\left|\psi_{S_j}\right\rangle$ 表示S_j上的投影态。我们将 Π_j 解释为排除局部态 $|v_j\rangle$: 全局态 $|\psi\rangle$ 满足约束j 当且仅当

$$\Pi_j |\psi\rangle = 0 \quad \Longleftrightarrow \quad \left\langle v_j \middle| \psi_{S_j} \right\rangle = 0.$$

定义 1(PRODSAT-QSAT(k))。给定秩一 k-局域投影算符 $\Pi_1, \ldots, \Pi_m$，判断是否存在一个乘积态 $|\psi\rangle = \otimes n_{j=1} |\psi_j\rangle$ ，使得对所有 $j \in [m]$均成立。

贡献。

- 我们形式化了 *PRODSAT-QSAT(k)* ，并设计了一种专门用于证明乘积态不可满足性的算法架构。
- 我们给出了一条可靠的子句学习规则，该规则基于通过区间排除投影振幅中的零点。
- 我们提出了用于子句学习的实用规则，并提供了该算法的实现。

特别地，我们引入了一个理论求解器（算法1），能够检测一族乘积态空间区域中各个QSAT(k)约束的乘积态不可满足性。在此基础上，我们利用子句学习框架定义了一种算法（算法3），该算法由以下结果提供支持:

定理 1. 固定 k，并考虑一个 *QSAT(k)* 实例。算法 *3* 在最坏情况下的运行时间关于量子比特数呈指数级。如果它能够证明输入实例属于 *UN-PRODSAT*，则返回 *UN-PRODSAT*。否则，返回 *MAYBE(A, ρ)*，其中 A 和 ρ 分别是理论求解器构造的集合过近似的面积总和与最大模平方总和。

在上述结果中，参数 A 和 ρ 在算法无法证明实例属于 UN-PRODSAT 时，可作为启发式指标，用于判断该实例属于 PRODSAT 的可能性有多大。参数值越小，这种可能性就越高。

论文提纲。

- 在第2节中，我们介绍理论求解器使用的几何工具。
- 在第3节中，我们介绍理论求解器（算法1），并讨论和证明其可靠性保证（总结于命题1）。
- 在第4节中，我们讨论整体的基于CDCL的架构和子句学习规则，并证明定理 1。
- 在第5节中，我们讨论实现并分享实证观察。
- 最后，我们在第6节中总结研究结果。

2 几何预备知识

对于乘积态 $|\psi\rangle$，将每个单量子比特态（忽略整体相位）写成布洛赫坐标：

$$\left|\psi_j\right\rangle = \left|\Psi(\theta_j,\varphi_j)\right\rangle \;=\; \cos(\theta_j/2)\,|0\rangle + e^{i\varphi_j}\sin(\theta_j/2)\,|1\rangle\,, \qquad \theta_j\in[0,\pi],\ \varphi_j\in[0,2\pi).$$

对于支撑在 $S_i=\{i_1,\dots,i_k\}$ 上的约束 $\Pi_i = |v_i\rangle\langle v_i|$，这等价于

$$\langle v_i| \cdot \bigotimes_{j\in S_i} \left|\Psi(\theta_j,\varphi_j)\right\rangle .$$

给定一些复数 $p_1,\dots,p_\ell$（其中 $\ell\geq 2$），我们用 $\mathrm{poly}[p_1,\dots,p_\ell]$ 表示复平面中以边 $[p_1,p_2],\dots,[p_{\ell-1},p_\ell],[p_\ell,p_1]$ 构成的多边形。注意，我们将线段也视为多边形（当 $\ell=2$ 时）。此外，给定任意实数 r_0,$r1,\varphi_0,\varphi_1$，且满足 $0\leq r_0\leq r_1$ 和 $\varphi_0\leq\varphi_1$，我们将考虑环形扇区

$$\mathrm{Ann}(r_0,r_1;\varphi_0,\varphi_1) \coloneqq [r_0,r_1]e^{i[\varphi_0,\varphi_1]},$$

如图 1a 所示。

给定任意两个 集合A、$B\subseteq\mathbb{C}$，我们分别将它们的闵可夫斯基和与乘积定义为 ，

$$A\oplus B \coloneqq \{a+b\mid a\in A, b\in B\}, \qquad A\odot B \coloneqq \{a\cdot b\mid a\in A, b\in B\}.$$

此外，对于任意集合$A\subseteq\mathbb{C}$，我们用 conv$A$ 表示其凸包：包含A的最小凸子集。显然，特别地，对于任意点p_1,.....、$p_n\in\mathbb{C}$，有 [14]

$$\mathrm{conv}\{p_1,\dots,p_\ell\} \coloneqq \left\{ \sum_{r=1}^{\ell}\lambda_r p_r \;\middle|\; \lambda_r\in\mathbb{R}_{\geq 0}, \sum_{r=1}^{\ell}\lambda_r = 1 \right\},$$

并且，如果 poly$[p_1,$.....、$p_\ell]$ 是一个凸多边形，那么 poly$[p_1,$.....、$p_\ell]=$conv$\{p_1,$.....、$p_\ell\}$。

如果形如 $|\psi(\theta,\varphi)\rangle$ 的状态 $|\psi\rangle$ 满足 $\theta\in[\theta_0,\theta_1]$ 和 $\varphi\in[\varphi_0,\varphi_1]$，则 n

$$\langle 0\mid\psi\rangle\in\cos([\theta_0/2,\theta_1/2]), \qquad \langle 1\mid\psi\rangle\in\sin([\theta_0/2,\theta_1/2])e^{i[\varphi_0,\varphi_1]},$$

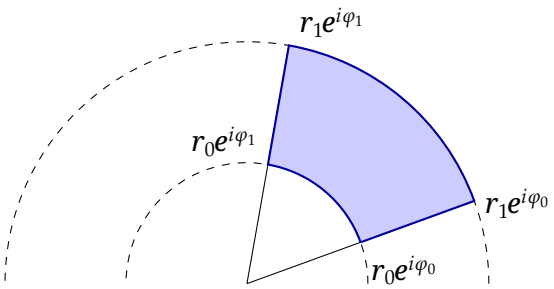
因此，振幅 $\langle 0 \mid \psi \rangle$ 位于区间 $[s_0, s_1] := \cos([\theta_0/2, \theta_1/2])$ 内，可将其识别为环形扇区 $\mathrm{Ann}(s_0, s_1; 0, 0)$；而振幅 $\langle 1 \mid \psi \rangle$ 位于环形扇区 $\mathrm{Ann}(r_0, r_1;\ \varphi_0, \varphi_1)$ 内，其中 $[r_0, r_1] := \sin([\theta_0/2, \theta_1/2])$。因此，两个振幅的取值都在环形扇区内。

计算环形扇区的闵可夫斯基积非常简单，以下结果对此作了说明。

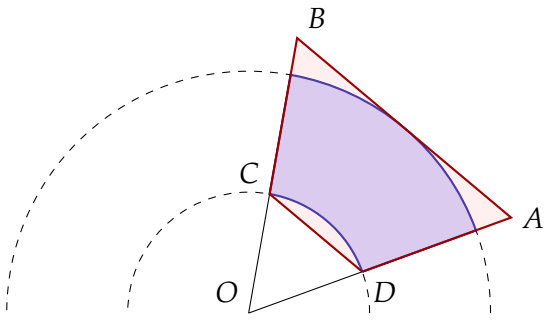
引理 1. 设$\mathrm{Ann}(r_0, r_1, \varphi_0, \varphi_1)$ 和$\mathrm{Ann}(s_0, s_1, \theta_0, \theta_1)$ 是环形扇区。它们的闵可夫斯基积可计算为

$$\mathrm{Ann}(r_0, r_1, \varphi_0, \varphi_1) \odot \mathrm{Ann}(s_0, s_1, \theta_0, \theta_1) = \mathrm{Ann}(r_0 s_0, r_1 s_1, \varphi_0 + \theta_0, \varphi_1 + \theta_1).$$

证明。 考虑到加法和乘法是连续且单调的，我们只需利用闭区间是连通且紧致的这一事实，并注意到，对于任意实数r、s，θ、φ，有$r \exp(i\varphi) \cdot s \exp(i\theta) = rs \exp(i(\varphi + \theta))$。 □



(a) 环形扇区 Ann（r_0, r_1；φ_0, φ_1）。



(b) 多边形包络的表示（红色）以及环形扇区（蓝色），遵循与引理 2 相同的记号。

图 1：环形扇区及其多边形包络。

上述引理提供了一种简单的方法来计算环形扇区的闵可夫斯基积。然而，直接计算它们的闵可夫斯基和并不容易。为了能够对这些环形扇区进行运算，我们将用凸多边形将其包络起来，依据如下结果。

引理 2。 给定一些实数 $0 \le r_0 \le r_1$ 和 $0 \le \varphi_0 \le \varphi_1 \le 2\pi$，满足 $\varphi_1 - \varphi_0 < \pi$，考虑环形扇区 $\Gamma := $ 环形扇区$(r_0, r_1;\ \varphi_0,\ \varphi_1)$，并令 $\Delta := \varphi_1 - \varphi_0$。

固定这些点

$$A := \frac{r_1}{\cos(\Delta/2)} e^{i\varphi_0}, \qquad B := \frac{r_1}{\cos(\Delta/2)} e^{i\varphi_1}, \qquad (C, D) := \left(r_0 e^{i\varphi_1}, r_0 e^{i\varphi_0} \right),$$

我们将 Γ 的软多边形包络 定义为多边形 $P :=$ 多边形$[A,\ B,\ C,\ D]$,，如图 *1b* 所示。它满足以下性质：

(i) 软多边形包络 P 是一个凸多边形。

(ii) 环形扇区 Γ 包含在其软多边形包络 P 中。

(iii) 软多边形包络 P 可以任意逼近环形扇区 Γ，且随着 $\varphi_1 \to \varphi_0$。形式化地，若 $\mu(\cdot)$ 表示复平面中某个区域的表面积，

$$\lim_{\varphi_1 \to \varphi_0} \frac{\mu(P)}{\mu(\Gamma)} = 1.$$

证明。令O 表示原点（$O := 0 \in \mathbb{C}$）。

(i) 只需验证，由于 $\varphi_1 - \varphi_0 \leq \pi$，所有顶点处的角都小于或等于 π，结论便可由 [14]中的定理 18.7得出。

(ii) 很容易证明，Γ 的内部与P 所围成的面积的交集非空，因此只需证明P 的周长位于 Γ 的外部或边界上。

定义点 $\mathrm{A}' = \mathrm{r}_1 \mathrm{e}^{i\varphi_0}$ 和 $\mathrm{B}' = \mathrm{r}_1 \mathrm{e}^{i\varphi_1}$，使得 $\mathrm{AD} = \mathrm{AA}' \cup \mathrm{A}'\mathrm{D}$，且 $\mathrm{BC} = \mathrm{BB}' \cup \mathrm{B}'\mathrm{C}$。显然，线段 $\mathrm{A}'\mathrm{D}$ 和 $\mathrm{B}'\mathrm{C}$ 位于边界上，而 AA' 和 BB' 位于外部（不包括 A' 和 B' 本身），因为它们的模大于 r_1。此外，CD（不包括 C 和 D）位于外部，因为它是连接圆周上两个半径为 r_0的点的弦。

最后，关于 AB，只需注意到它与半径为 r_1，的圆相切，因此除其中点（显然位于边界上）外，它都位于圆的外部。

(iii) 一方面，环形扇形的面积为

$$\mu(\Gamma) = \frac{\Delta}{2}(r_1^2 - r_0^2).$$

另一方面，软多边形包络 P 的面积等于等腰三角形 OAB 的面积减去等腰三角形ODC 的面积。此外，

$$\mu(OAB) = r_1^2 \tan(\Delta/2), \qquad \mu(ODC) = \frac{1}{2} r_0^2 \sin(\Delta),$$

因此

$$\begin{aligned} \mu(P) = \mu(OAB) - \mu(ODC) &= r_1^2 \tan(\Delta/2) - \frac{r_0^2}{2} \sin(\Delta) \\ &= \frac{\Delta}{2}(r_1^2 - r_0^2) + \mathcal{O}(\Delta)^3, \end{aligned}$$

由此即可得出结论。

□

前述结果表明，环形扇区的软多边形包络是一个凸集，它包含整个扇区；随着扇区变小，该包络对扇区的近似效果也会更好。此外，由于它是一个多边形，操作起来要容易得多。然而，我们只能为角度差 $\varphi_1 - \varphi_0$ 小于 π 的环形扇区计算该包络，而且角度差越小，近似效果越好。考虑到这一点，我们可以如下定义一般多边形包络。

定义2. 令 $S(r_0, r_1, \varphi_0, \varphi_1)$ 表示环形扇区 $\mathrm{Ann}(r_0, r_1, \varphi_0, \varphi_1)$的软多边形包络。

考虑一个精度因子$F \in \mathbb{N}$。给定任意一个形如 $\Gamma := \mathrm{Ann}(r_0, r_1, \varphi_0, \varphi_1)$的环形扇区（角度不受限制），令 $\Delta = (\varphi_1 - \varphi_0)$ 模 2π，并固定

$$N := \begin{cases} F & \text{if } \Delta < \pi/8, \\ 2F & \text{if } \Delta < \pi/4, \\ 4F & \text{if } \Delta < \pi/2, \\ 8F & \text{if } \Delta < \pi, \\ 16F & \text{otherwise.} \end{cases}$$

我们将 Γ 的多边形包络定义为

$$\mathrm{polyenc}\,(\Gamma) := \mathrm{conv}\left[\bigcup_{r=0}^{N-1} S\left(r_0, r_1, \varphi_0 + (\varphi_1 - \varphi_0)\frac{r}{N}, \varphi_0 + (\varphi_1 - \varphi_0)\frac{r+1}{N}\right)\right].$$

在刚才引入的定义中，无论选择何种精度因子，我们总能得到一个定义良好的多边形包络，因为软多边形包络中所用的角度远小于 π。精度因子可自行调整；因子越大，近似效果越好，但形状也会更复杂。我们取$F := 4$。

请注意，在多边形包络的定义中，我们必须取软多边形包络并集的凸包。事实上，对于具有非零测度的环形扇区，其并集永远不会是凸集，因此凸包总会产生严格的过近似。不过，就我们的用途而言，这种过近似足够准确（见备注1）。为了高效计算凸包，可以使用单调链算法 [2],：该算法将集合中的点按其实部排序（实部相同则按虚部排序），然后线性遍历这些点，计算下凸包和上凸包。我们将在实现中使用该算法。不过，值得指出的是，另一种选择是使用Kirkpatrick–Seidel算法 [11], 。该算法更为复杂，因此不太适合我们所考虑的小规模输入，但其渐近复杂度更优。具体而言，若n 表示输入点的数量，h 表示凸包中的点数，则单调链算法的渐近复杂度为 $\mathcal{O}(n\log n)$，而Kirkpatrick–Seidel算法的复杂度为 $\mathcal{O}(n \log h)$。

本节最后，我们将讨论如何判断复平面中某些凸多边形的闵可夫斯基和是否包含 0。给定若干凸多边形，始终可以高效地计算出与它们的闵可夫斯基和对应的凸多边形的顶点。如下列结果所示，判断 0 是否属于这个和是很直接的。

引理3. 考虑复平面中的一个凸多边形 $P := \mathrm{poly}[p_1, \ldots, p_\ell] = \mathrm{conv}\{p_1, \ldots, p_\ell\}$ 。当且仅当对于 $j = 1, \ldots, \ell$，叉积 $p^j \times (p^{j+1} - p^j)$的符号都相同时，点0 才位于 P 的内部或边界，其中令 $p^{\ell+1} := p_1$。

证明。 凸多边形等于由其各边所定义的半平面的交集 [19,§1–2], 因此，对于固定的 $\sigma \in \{+1, -1\}$,

$$P = \bigcap_{j=1}^{\ell} \left\{p \in \mathbb{C} \mid \mathrm{sign}\left((p_j - p) \times (p_{j+1} - p_j)\right) \in \{\sigma, 0\}\right\}.$$

由此事实可直接得出该结果。 □

3 理论求解器

不失一般性，考虑一个支撑在 $[k]$ 上的秩一投影算子 $\Pi := |v\rangle\langle v|$ 。给定一些区间

$$[\theta_{10},\theta_{11}],\ldots,[\theta_{k0},\theta_{k1}] \subseteq [0,\pi], \qquad [\varphi_{10},\varphi_{11}],\ldots,[\varphi_{k0},\varphi_{k1}] \subseteq [0,2\pi], \tag{$*$}$$

we w考虑一些单量子比特态 $|\psi_1\rangle$、.....、$|\psi_k\rangle$，使得对于每个 $j = 1$、.....、k

$$|\psi_j\rangle = |\psi(\theta_j,\varphi_j)\rangle = \cos(\theta_j/2)\,|0\rangle + e^{i\varphi_j}\sin(\theta_j/2)\,|1\rangle\,, \qquad \theta_j \in [\theta_{j0},\theta_{j1}], \qquad \varphi_j \in [\varphi_{j0},\varphi_{j1}].$$

我们的目标是确定，在这些假设下，乘积态 $|\psi\rangle := |\psi_1\rangle \otimes \cdots \otimes |\psi_k\rangle$ 是否可能与 $|v\rangle$ 正交（即属于 $|v\rangle\langle v|$ 的核）。为此，我们将引入一种算法，该算法将返回两种可能的输出：

- UN-PRO，DSAT if it can certify that $\langle v|\,(|\psi_1\rangle \otimes \cdots |\psi_k\rangle) \neq 0$ for all allowed configurations
- 否则为 MAYBE(A， ρ)，其中指定了一个面积为 A 、最大模平方为 ρ 的集合，该集合覆盖 $\langle v \mid \psi\rangle$ 的所有可能取值，其中包括 0。

为了确定对于所有允许的配置是否有 $\langle v|\,(|\psi_1\rangle \otimes \cdots \otimes |\psi_k\rangle) = 0$，我们首先考虑对于每个 $j = 1$、.....、k 的区间

$$P_j^0 := \cos([\theta_{j0}/2,\theta_{j1}/2]),$$

以及环形扇区

$$P_j^1 := \sin([\theta_{j0}/2,\theta_{j1}/2]) \cdot \exp(i[\varphi_{j0},\varphi_{j1}]),$$

不难验证，$\langle 0 \,|\, \psi_j\rangle \in P_j^0$和 $\langle 1 \,|\, \psi_j\rangle \in P_j^1$。

利用恒等分解，我们知道

$$\begin{aligned}
\langle v|\,(|\psi_1\rangle \otimes \cdots \otimes |\psi_k\rangle) &= \langle v|\, I_{2^k}(|\psi_1\rangle \otimes \cdots \otimes |\psi_k\rangle) \\
&= \langle v|\left(\sum_{t=0}^{2^k-1} |t\rangle\langle t|\right)(|\psi_1\rangle \otimes \cdots \otimes |\psi_k\rangle) \\
&= \sum_t \langle v \mid t\rangle \cdot \langle t_1 \mid \psi_1\rangle \cdots \langle t_k \mid \psi_k\rangle \\
&\in \bigoplus_t \left(\langle v \mid t\rangle \cdot P_1^{t_1} \odot \cdots \odot P_k^{t_k}\right)
\end{aligned}$$

其中，我们用$t_1,...,\ t_k$表示k位二进制位的t，而 $\oplus$ 和 $\odot$ 分别表示集合的闵可夫斯基和与乘积。因此，对于每一种可能的配置，要使 $\langle v \mid \psi\rangle \neq 0$ 成立，一个充分条件是

$$0 \notin \bigoplus_t \left(\langle v \mid t\rangle \cdot P_1^{t_1} \odot \cdots \odot P_k^{t_k}\right). \tag{S}$$

该条件可以使用我们在第2节中介绍的几何工具进行验证。可以使用引理 1计算闵可夫斯基积，随后将其与标量 $\langle v \mid t\rangle$ 相乘则是直接的。这样，表达式的右侧就变成了环形扇区的闵可夫斯基和。利用引理 2和定义2，我们可以求出这些扇区的多边形包络；正如我们在第2节末尾所讨论的，还可以高效地确定0是否属于它们的闵可夫斯基和。

备注1。 正如我们在第*2*节中提到的，在定义环形扇区的多边形包络时，我们取一个已经覆盖该扇区的集合的凸包——而凸包始终是真超集。然而，如果该凸包将参与含有大量被加集合的闵可夫斯基和，由于*Shapley–Folkman*引理 *[18]*，它仍然能够提供良好的过近似。该结果表明，在集合 $S := A_1 \oplus \cdots \oplus A_\ell \subseteq \mathbb{C}$的闵可夫斯基和中，凸包 conv$S$ 中的每个元素 s 都可以表示为一个和 $s = a_1 + \ldots + a_\ell$，其中每个元素 a^j 都属于 A^j，除了至多两个索引 j_1, j_2 所对应的 $a^{j_1} \in \mathrm{conv}(A^{j_1}) \setminus A^{j_1}$（对于 a^{j_2}也是如此）。

这定义了理论求解器的行为：如果利用我们已介绍的工具验证了 $\langle v \mid \psi \rangle \neq 0$ 的充分条件 (S)，理论求解器将返回 UN-PRODSAT。否则，它将返回 MAYBE，并同时给出 (S) 中闵可夫斯基和的面积和最大模，该和包含 0。显然，若结果为 UN-PRODSAT，我们就能完全确定，在所有允许的配置下，都不存在满足由$|v\rangle\langle v|$定义的约束的状态。另一方面，MAYBE 结果并不能保证在允许的配置中存在满足该约束的状态——不过，在这种情况下，返回的面积越小，我们对存在解的确信程度就越高。

以下结果概括并总结了本节所定义的过程。

命题1. 考虑一个支撑为 $s_1, \ldots, s_k$ 的约束 Π，该约束由一个 k 量子比特态 $|v\rangle$ 定义；再考虑一些区间 $[\theta_{j0}, \theta_{j1}] \subseteq [0, \pi]$，其中 $[\varphi_{j0}, \varphi_{j1}] \subseteq [0, 2\pi]$ 对应 $j = 1, \ldots k$，并定义

$$\Omega := \{ |\Psi(\theta_1, \varphi_1)\rangle \otimes \cdots \otimes |\Psi(\theta_n, \varphi_n)\rangle \mid \theta_{s_j} \in [\theta_{j0}, \theta_{j1}], \varphi_{s_j} \in [\varphi_{j0}, \varphi_{j1}], j = 1, \ldots, k \}$$

为所有满足以下条件的 n量子比特乘积态的集合：量子比特 s^j 的布洛赫球角度被 $[\theta_{0j}, \theta_{1j}]$ 和 $[\varphi_{j0}, \varphi_{j1}]$ 界定，其中 $j = 1$，……，k。

将 v，（s_1,……，s_k），（$[\theta_{j_0}, \theta_{j_1}]$，$[\varphi_{j_0}, \varphi_{j_1}]$）输入算法*1*：

（*i*）如果结果为 *UN-PRODSAT*，则 Ω 中没有状态满足约束 Π。

（*ii*）如果结果为 *MAYBE(A, ρ)*，则在 $|\psi\rangle \in \Omega$ 时取值 $\Pi|\psi\rangle$ 的集合可能包含零，并且被一个面积为 A、最大模平方为 ρ 的集合所覆盖。

4 主体架构（定理 1 的证明）

本节将构造算法 3，并证明定理 1。

令 $x_1, \ldots, x_\ell$为 ℓ 个布尔值，其中 $\ell \geq 1$。令

$$B(x_1, \ldots, x_\ell) := \sum_{j=1}^{\ell} x_j \cdot \frac{2\pi}{2^j},$$

我们定义区间

$$I(x_1, \ldots, x_\ell) := B(x_1, \ldots, x_\ell) + \left[0, \frac{2\pi}{2^\ell}\right] = \left[B, \frac{2\pi}{2^\ell} + B\right].$$

此区间实际上表示在 $[0, 2\pi]$中执行 ℓ 次二分搜索的结果：若在第 j次迭代中 $x_j = 0$，则选择左半部分；若$x_j = 1$，则选择右半部分。显然，所有此类区间构成的集合，除去零测集上的重叠

算法1 理论求解器

输入： 由一个k量子比特状态 $|v\rangle$ 定义的约束，其支撑为$(s_1,\ldots,s_k)$。
输入： 由k 个区间有序对$([\theta_{j0},\theta_{j1}],[\varphi_{j0},\varphi_{j1}])_{j=1}^{k}$组成的序列。

多边形加项 $\leftarrow$ []
对 $t = (\overline{t_1,\ldots,t_k})_2 \leftarrow 0,\ldots 2^k-1$ **执行** ▷ $t_{(\cdot)}$ 是t的二进制位。
    环形扇区 $\leftarrow$ []
    对于$j \leftarrow 1,\ldots,k$ **执行**
        如果 $t_j = 1$**那么**
            追加(环形扇区, $\sin([\theta_{j0}/2,\theta_{j1}/2])\cdot\exp(i[\varphi_{j0},\varphi_{j1}])$)
        else
            追加(环形扇区, $\cos([\theta_{j0}/2,\theta_{j1}/2])$)
        结束如果
    结束循环
    追加(多边形加项, polyenc（$\langle v\mid t\rangle\cdot\bigodot$环形扇区）)
结束循环
$P \leftarrow \bigoplus$多边形加项
如果 $0\in P$ **则**
    返回 MAYBE(面积(P), $\max_{p\in P}|p|^2$)
否则
    返回 UN-PRODSAT
结束条件
判断

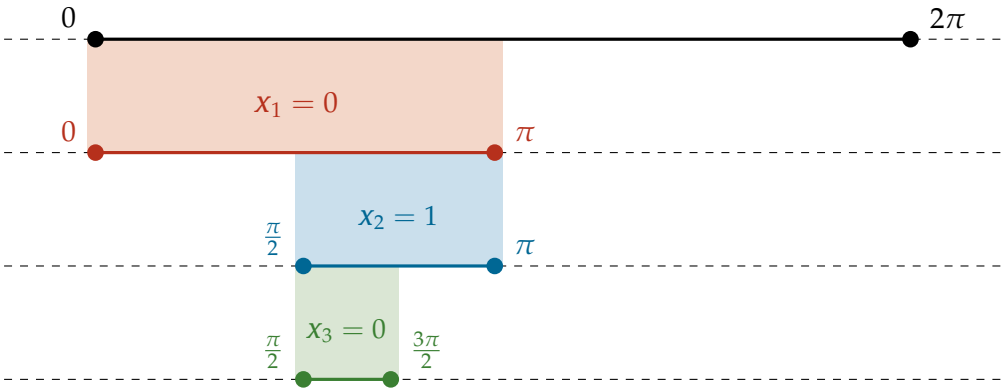


图 2：黑色、红色、蓝色和绿色线段分别表示区间 $I()$、$I(0)$、$I(0,1)$ 和 $I(0,1,0)$。传递给 I 的每个额外比特都代表在区间 $[0,2\pi]$ 中进行二分搜索的新一步。

子集，是 $[0, 2\pi]$ 的一个划分。此外，对于固定的 ℓ，所有形如 $I(0, x_2, \ldots, x_\ell)$ 的区间的集合，以类似的方式划分区间 $[0, \pi]$。在图 2 中，我们展示了这些区间构造方式的可视化表示。

考虑一个如定义 1 中所述的 QSAT(k) 实例 Q，并固定一个搜索深度 $D \in \mathbb{N}$。现在我们将形式化描述一个过程，旨在确定是否存在 $2n$ 个区间

$$\left\{\Theta_j := I(0, x_j^1 \ldots, x_j^{D-1}), \quad \Phi_j := I(y_j^1, \ldots, y_j^D)\right\}_{j=1}^{n},$$

其中 x_j^d、$y_j^d \in \mathbb{B}$，对于这些区间，问题可能存在一个解 $|\psi\rangle$，使得 $|\psi_j\rangle = |\Psi(\theta_j、\ \varphi_j)\rangle$以及 $\theta_j \in \Theta_j$、$\varphi_j \in \Phi_j$。如果算法能够证明不存在这样的区间，则返回 UN-PRODSAT；否则，作为理论求解器，它将返回 MAYBE(A，ρ)。此外，在关于问题实例的附加假设下，如果算法能够验证存在一个满足赋值，则还可能返回 SAT。

我们将问题建模如下。首先，考虑定义在由$n \times (2D - 1)$ 个变量组成的集合上的命题逻辑语言：

$$\mathcal{V} := \left\{\hat{\theta}_j^{d'}, \hat{\varphi}_j^d \;\middle|\; j \in [n],\ d \in [D],\ d' \in [D-1]\right\}.$$

为方便起见，我们记作

$$\hat{\theta}_j := (\hat{\theta}_j^1, \ldots, \hat{\theta}_j^{D-1}), \qquad \hat{\varphi}_j := (\hat{\varphi}_j^1, \ldots, \hat{\varphi}_j^D).$$

按照标准术语，我们将该语言中的*literal* 定义为v 或 $\neg v$ 的形式，其中$v \in \mathcal{V}$；将*clause* 定义为文字的析取，将*(CNF) formula*定义为子句的合取。此外，我们将该语言上的*total assignment*定义为函数 $\sigma : \mathcal{V} \longrightarrow \mathbb{B}$，将*partialassignment* 定义为函数 $\sigma' : D \subset \mathcal{V} \longrightarrow \mathbb{B}$。给定一个部分赋值或全赋值，子句和公式的真值（真、假或未定义）按显然的方式定义。

如果假设一个n量子比特状态具有如下形式，则称全赋值 σ 是(Q,D)-*invalid*：

$$|\psi\rangle = |\Psi(\theta_1, \varphi_1)\rangle \otimes \cdots \otimes |\Psi(\theta_n, \varphi_n)\rangle, \qquad \theta_j \in I(0, \sigma(\hat{\theta}_j)), \quad \varphi_j \in I(\sigma(\hat{\varphi}_j)),$$

存在一个约束，使得诱导区域中的任何乘积态都不满足该约束；在这种情况下，我们也称该约束在赋值 σ下不满足。反之，如果在这些条件下，所有约束对 $|\psi\rangle$ 均得到满足，则称 σ 是(Q,D)-*valid*。此外，如果一个子句在任意(Q,D)-valid全赋值下均得到满足，则称该子句为(Q, D)-*valid* 。

介绍了这些基本定义后，下面的结果是我们方法的关键。

引理 4. *考虑一个如上所建模的 QSAT 实例 Q。*

(i) 如果一个由(Q,D)-有效子句合取而成的（*CNF*）公式不可满足，那么 *QSAT* 实例 Q 就是 *UN-PRODSAT*。

(ii) 如果存在一个(Q,D)-有效全赋值，那么 *(i)* 中的假设就不可能成立。

证明。 对于 (i)，如果若干个(Q,D)-有效子句无法同时满足，这意味着不存在(Q,D)-有效全赋值，因此Q 是 UN-PRODSAT。事实上，如果Q 是

PRODSAT，就会存在一个满足赋值 $|\Psi(\theta_1,\varphi_1)\rangle\otimes\cdots|\Psi(\theta_n,\varphi_n)\rangle$，因此，全赋值[1]$\theta_j\in I(0,\sigma(\hat{\theta}_j))$ 和 $\varphi_j\in I(\sigma(\hat{\varphi}_j))$ 将是 (Q,D)- 有效的，因此会满足任何 (Q,D)- 有效子句。

剩余陈述显然成立。 □

此外，我们现在将讨论每个(Q,D)-无效赋值如何导出一个(Q,D)-有效子句，从而排除该赋值。

对于一个全赋值 σ 和一个变量v，令 $[v]^\sigma$ 表示文字v （当 $\sigma(v)=0$ 时），而 $\neg v$ 表示文字（当$\sigma(v)=1$时）。如果一个支撑为$s_1,\ldots,s_k$的约束在 σ下不满足，那么显然

$$\bigvee_{j=1}^{k}\left(\bigvee_{d=1}^{D}[\hat{\varphi}_{s_j}^{d}]^\sigma\vee\bigvee_{d'=1}^{D-1}[\hat{\theta}_{s_j}^{d}]^\sigma\right)$$

它是(Q,D)-有效的，因为任何在定义量子比特$s_1,\ldots,s_k$上布洛赫球角度的变量取值不同于 σ 的赋值都满足它。不过，即使该子句是(Q,D)-有效的，它也没有特别大的信息量，因为它包含$k(2D-1)$个变量。幸运的是，我们有可能删去其中若干文字。

如果某个约束在全赋值 σ下不满足，这意味着当其量子比特的布洛赫球角度落在

$$\theta_{s_j}\in I\left(0,\sigma(\hat{\theta}_{s_j}^{1}),\ldots,\sigma(\hat{\theta}_{s_j}^{D-1})\right),\qquad \varphi_{s_j}\in I\left(\sigma(\hat{\varphi}_{s_j}^{1}),\ldots,\sigma(\hat{\varphi}_{s_j}^{D})\right)$$

对于 $j\in[k]$。不过，我们仍可以反复运行算法1，以找出更小的值 D_1、.....、D_k 和D_1'、.....、D_k'，使得当角度限制为

$$\theta_{s_j}\in I\left(0,\sigma(\hat{\theta}_{s_j}^{1}),\ldots,\sigma(\hat{\theta}_{s_j}^{D_j'})\right),\qquad \varphi_{s_j}\in I\left(\sigma(\hat{\varphi}_{s_j}^{1}),\ldots,\sigma(\hat{\varphi}_{s_j}^{D_j})\right),$$

因此，(Q,D)-有效阻塞子句可简化为

$$\bigvee_{j=1}^{k}\left(\bigvee_{d=1}^{D_j}[\hat{\varphi}_{s_j}^{d}]^\sigma\vee\bigvee_{d'=1}^{D_j'}[\hat{\theta}_{s_j}^{d'}]^\sigma\right). \tag{RC}$$

具体而言，我们建议使用算法 2 来寻找这样的值。

综上所述，有一种自然的方法来解决我们的问题：

1. 考虑一个初始全赋值 σ，并初始化一个由(Q,D)-有效阻塞子句组成的数据库。

2. 使用算法1检查每个约束上的 σ。

   (a) 如果所有约束都输出 MAYBE(A_r,ρ_r)，根据引理 4，返回 MAYBE$(\sum_r A_r,\sum_r\rho_r)$。(b) 如果当前全赋值不满足某些约束，则将针对这些约束分别通过子句泛化规则（算法 2）生成的所有 (Q,D)-有效子句添加到阻塞子句数据库中。所有这些子句都将特别排除该赋值。

[1]这样的赋值必定存在，因为形如 $I(x_1,\ldots,x_D)$ 的区间的并集覆盖了 $[0,2\pi]$。

算法 2 子句泛化规则

输入： 支撑于 s_1,...、s_k 的约束 Π。**输入：** 全赋值 σ，理论求解器在 Π 上返回 UN-PRODSAT。**输入：** 搜索深度 D。

1: 我们令$C(d_1,\ldots,d_k,d'_1,\ldots,d'_k)$ 表示对 Π 调用理论求解器的结果，其中

$$\theta_{s_j} \in I\left(0,\sigma(\hat{\theta}^1_{s_j}),\ldots,\sigma(\hat{\theta}^{d'_j}_{s_j})\right), \qquad \varphi_{s_j} \in I\left(\sigma(\hat{\varphi}^1_{s_j}),\ldots,\sigma(\hat{\varphi}^{d_j}_{s_j})\right).$$

2: 初始化一些变量 $(D_1,\ldots,D_k,D'_1,\ldots,D'_k) \leftarrow (D,\ldots,D,D-1,\ldots,D-1)$. 3: 初始化一个可移动变量向量 $M \leftarrow (D_1,\ldots,D_k,D'_1,\ldots,D'_k)$
4: 对于任意变量d，令$m(d) \leftarrow d$，若$d \notin M$；否则令$m(d) \leftarrow d-1$。

5: **当**$M \neq \emptyset$**时执行**

6: 如果 *C(m(D1),...,m(Dk),m(D'1),...,m(D'k))* 为 *UN-PRODSAT*，则 *7*: *(D1,...,Dk,D'1,...,D'k) = (m(D1),...,m(Dk),m(D'1),...,m(D'k)) 8*: 否则 *9*: 对于 *M* 中的 *d* 执行▷ 我们遍历的是 *M* 中的变量，而不是它们所对应的值。 *10*: $d \leftarrow d-1$ *11*: 如果 *C(D1,...,Dk,D'1,...,D'k)* 为 *MAYBE*，则 *12*: $d \leftarrow d+1$ *13*: 从 *M* 中移除 *d* *14*: 结束 *if* *15*: 结束 *for* *16*: 结束 *if* *17*: 在 *M* 中仅保留取值严格大于零的变量 *d*。 *18*: 结束 *while* *19*: 返回 $\bigvee_{kj=1}\left(\bigvee_{D=1}^{j}\lfloor d\hat{}\varphi^d_{s_j}\rfloor^\sigma \vee \bigvee_{D'_j\,d'=1}\lfloor\theta^{d'}s_j\rfloor^\sigma\right)$

3. 运行 CDCL求解器，寻找一个新的全赋值，使其满足所有(Q,D)-有效阻塞子句。

(a) 如果 CDCL SAT 求解器返回一个（可能不完整的）赋值，则将其补全为完整赋值 σ，并返回步骤 2。(b) 如果 CDCL SAT 求解器判定所给子句集合不可满足，则返回 UN-PRODSAT。

在上述算法中，只有当我们能找到一个不可满足的(Q,D)-有效子句集合时，才返回 UN-PRODSAT，如引理 4(i) 所示。反之，只要可能已经找到一个(Q,D)-有效全赋值，我们就返回 MAYBE，因为此时无法证明 UN-PRODSAT。此外，我们返回两个值，A 和 ρ，分别对应理论求解器所生成集合的表面积之和与最大模平方。显然，这些值越小，找到 SAT 赋值的可能性就越大。

值得强调的是，我们的算法总会终止，因为对于步骤 2 中找到的每个全赋值σ，算法要么返回，要么加入一个新子句，阻止该

赋值在步骤 3 中再次被选中。

就最坏情况复杂度而言，对于固定的k，我们的算法运行时间关于量子比特的数量呈指数增长。事实上，每次调用理论求解器以及生成阻塞子句所需的时间均为K，只取决于k，而与量子比特的数量无关。此外，我们只有 $2^{n(2D-1)}$ 种可能的赋值需要测试，因此算法最多可能执行这么多次迭代。再结合 SAT 求解器的最坏情况复杂度也是 $2^{n(2D-1)}$ 这一事实，最终得到的最坏情况总复杂度为$K \cdot 4^{n(2D-1)}$。

至此，定理 1 的证明完成。算法 3 总结了我们所介绍的过程。

算法 3 PRODSAT-QSAT(k)

输入： 实例Q，其中包含一组作用于n 个量子比特的k-局域约束。

输入： 搜索深度D。

1: CDCL-SAT $\leftarrow$ 在 $\mathcal{V}$ 中的变量上初始化一个 CDCL-SAT 求解器实例。 2: $\sigma \leftarrow$ 对 $\mathcal{V}$ 中的所有二值变量初始化一个全赋值。

3: **标记**开始4: **如果**理论求解器对 A_j，ρ_j下的每个约束都输出 MAYBE(σ，则返回 MAYBE($\sum_j A_j$， $\sum_j \rho_j$)6: **否则**7: **对于**每个约束 Π，若理论求解器对其输出 UN-PRODSAT，则执行8: $C \leftarrow$ 在 Π 和 σ （搜索深度为 D）上应用子句泛化规则9: 将子句 C 传递给 CDCL-SAT10: **结束循环**11: **结束条件判断**

12: **如果** CDCL-SAT 返回不可满足，**则**
13: **返回** UN-PRODSAT
14: **否则**
15: $\sigma \leftarrow$ 满足所有阻塞子句的全赋值
16: **跳转到** 开始
17: **结束 if**

综合以上内容，我们定义了一种程序，对于任意 QSAT(k) 实例，都会返回 UN-PRODSAT 结果（此时该实例确实是 UN-PRODSAT），或无法确定的 MAYBE(A，ρ) 结果——其中，A 和 ρ 的值越小，该实例越有可能是 PRODSAT。在第二种情况下，可以使用 Buchberger算法 [15], 来获得确定结果，但其最坏情况下的计算成本为双指数级。

4.1 PRODSAT 结果的附加假设

在附加假设下，我们或许能够用自己的方法证明 PRODSAT。

命题2。 考虑一个带有若干约束 Π_j 对 $j \in [m]$, 的 *QSAT(k)* 实例，并定义哈密顿量

$$H := \sum_{j=1}^{m} \Pi_j.$$

在假设该实例为 *PRODSAT*（相应地为 *SAT*），或 H 的最小特征值大于某个 E_0, 的情况下，如果我们的过程输出 *MAYBE(*−， ρ*)* 且$\rho < E_0$，则我们可以认证其为 *PRODSAT*（相应地为 *SAT*）。

证明。 如果算法输出 MAYBE$(-,\rho)$，那么我们知道乘积态空间中存在一个区域，其中对于每个状态 $|\psi\rangle$,

$$\langle\psi| H |\psi\rangle = \sum_j \langle\psi| \Pi_j |\psi\rangle \leq \sum_j \rho_j = \rho,$$

因此，如果 $\rho < E_0$,，那么 E_0 不可能是 H 的最小特征值。 □

4.2 关于 SAT 求解器的必要性

在我们的算法中，我们使用 SAT 求解器来寻找满足一组阻塞子句的全赋值。下面的结果说明了我们选择使用 SAT 求解器是有充分理由的。

命题 3。 设 $k \leq n$ 是一对自然数。假设存在一个过程，对于量子比特数为 n 的量子 *k-SAT* 实例，给定任意一组形如 (RC) 的子句，都能返回满足这些子句的布尔赋值，或者判定这些子句不可满足。该过程可以解决含 n 个变量的布尔 *k-SAT*。

证明。 为简化记法，给定一个布尔变量 X，我们将其两个文字表示为$(X,0) := \neg X$ 和 $(X,1) := X$。

为了证明我们的结果，只需说明：在一个有 $\{x_i\}_{i\in[n]}$ 个量子比特的 Quantum k-SAT 实例中，任意由 k 元布尔子句构成的集合，都可以编码为 (RC) 形式的子句。因此，设 $\{x_i\}_{i\in[n]}$ 为布尔变量，并固定任意一个由 m k 元子句组成的集合

$$C := \left\{ (x_{\iota(t,1)}, \alpha(t,1)) \vee \cdots \vee (x_{\iota(t,k)}, \alpha(t,k)) \right\}_{t=1}^{m},$$

其中 $\iota : [m] \times [k] \longrightarrow [n]$，且 $\alpha : [m] \times [k] \longrightarrow \{0,1\}$。不失一般性，我们假设每个变量在同一个子句中至多出现一次。

现在考虑一个包含 n 个量子比特和 m 个约束 $\{\Pi_t\}_{t\in[m]}$的 Quantum k-SAT 实例，使得对每个 <style i d='t'>t $\in [m]$,， Π_t 的支撑为 $\iota(t,1)$，.....， $\iota(t,k)$。对于这样的实例，无论搜索深度 D 为何，C 中的任何子句都可以通过变量替换写成 (RC) 形式。事实上，对于任意一个作用于量子比特 i_1,.....、i_k 上的约束 Π_t，任何形如

$$\bigvee_{j=1}^{k} \left(\bigvee_{d=1}^{D_j} (\hat{\varphi}_{i_j}^d, a_{jd}) \vee \bigvee_{d=1}^{D'_j} (\hat{\theta}_{i_j}^d, b_{jd}) \right)$$

可以是任意 Di、$D'_j \in \mathbb{N}_0$ 以及任意 a^{jd}、$b_{jd} \in \{0,1\}$的阻塞子句。特别地，这对$D_j = 1$ 和$D'_j = 0$成立，因此任何形如

$$(\hat{\varphi}_{i_1}^1, a_1) \vee \cdots \vee (\hat{\varphi}_{i_k}^1, a_k)$$

可以是阻塞子句。将其具体应用于每个约束 Π_t，并适当地选取a_r 的值，可得

$$\left\{ \left(\hat{\varphi}_{\iota(t,1)}^1, \alpha(t,1) \right) \vee \cdots \vee \left(\hat{\varphi}_{\iota(t,k)}^1, \alpha(t,k) \right) \right\}_{t=1}^{m}$$

是该量子k-SAT 实例中形如 (RC) 的子句集合。将变量x_i替换为 $\hat{}\varphi_i^1$，即可得出结果。

□

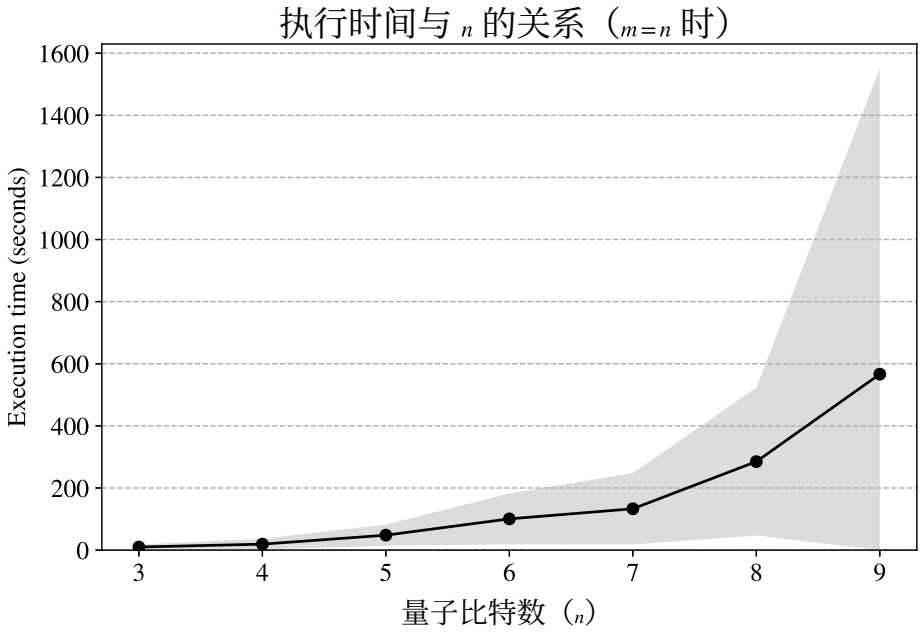


图3：我们的实现的运行时间随量子比特数变化的情况（平均值和标准差）。为公平起见，我们仅考虑约束数 m 等于量子比特数 n 的实例。

5 实现

我们已使用 Rust 实现了该算法，代码可在以下代码仓库中获取：

`https://github.com/System-Verification-Lab/cdcl-qsat`

它可以使用浮点数进行数值计算，也可以使用分圆域元素的精确表示来运行该算法。事实上，只要用户通过实现某个特征提供数表示，就可以使用任意数表示。此外，还可以使用任意 SAT 求解器运行该算法。通过 RustSAT 库 [10], ，我们提供了对 Glucose [3]（基于 Minisat [8]）、CaDiCaL [4], 和 BatSat [6]（用 Rust 重新实现的 Minisat）的内置支持。当然，用户也可以通过实现相应的特征来使用自己的 SAT 求解器。

我们在随机生成的状态上测试了这一实现，并验证它能够生成 UN-PRODSAT 结果。具体而言，在搜索深度为$D = 8$的情况下，我们对n 个量子比特和m 个约束进行了测试，其中$n = 3, \ldots, 9$，且 $m = 1, \ldots, n + 1$；对于 n 和 m 的每种配置，我们生成了 13 个随机约束集。运行时间随 n 的变化情况（针对$n = m$的情形）见图3。

表1记录了我们的算法在基准测试中成功生成 UN-PRODSAT 证书的比例。这些结果与理论预期 [15]十分吻合。表2列出了阻塞子句的数量、理论求解器的调用次数，以及执行结束时 CDCL 求解器保留的学习子句数量等指标。综合来看，这些指标表明，我们方法的平均计算成本远低于其理论最坏情况。它们还展示了我们的理论求解器以及对 SAT 求解器的使用所具有的有效性。

		约束数量 m									
		1	2	3	4	5	6	7	8	9	10
n	3	0/13	0/13	0/13	13/13						
	4	0/13	0/13	0/13	1/13	13/13					
	5	0/13	0/13	0/13	0/13	0/13	13/13				
	6	0/13	0/13	0/13	0/13	0/13	1/13	12/13			
	7	0/13	0/13	0/13	0/13	0/13	0/13	7/13	13/13		
	8	0/13	0/13	0/13	0/13	0/13	0/13	1/13	4/13	13/13	
	9	0/13	0/13	0/13	0/13	0/13	0/13	0/13	0/13	4/13	13/13

表1：我们的算法实现能够生成的 UN-PRODSAT 证书比例。这些结果与理论预期一致。

n	Avg. #子句	Max. #子句	Avg. #调用次数	Max. #调用次数	平均学习子句数（CDCL）	搜索空间
3	12 140	88 284	270 951	1 548 287	1415	2^{45}
4	20 429	199 770	466 371	3 643 611	2939	2^{60}
5	41 796	316 550	910 593	5 749 351	6152	2^{75}
6	85 866	753 176	1 824 031	13 917 850	11 078	2^{90}
7	94 706	460 644	2 152 597	17 073 270	11 812	2^{105}
8	130 422	968 536	2 761 632	23 582 225	15 211	2^{120}
9	189 815	3 016 533	4 010 449	54 923 198	20 783	2^{135}

表2：对于每个量子比特数 n，CDCL求解器在每个实例中所需的阻塞子句、理论求解器调用次数和学习到的冲突子句的平均数量（执行结束时）。我们还展示了评估一个实例所需的子句数和约束调用次数的最大值。

6 结论

本文提出了一种面向 PRODSAT-QSAT(k) 的搜索驱动子句学习框架，旨在证明具有秩一局部投影算子的量子 k-SAT 实例不存在满足约束的乘积态。其核心思想是通过二分划分将每个单量子比特布洛赫流形离散为有限个角区域，并让布尔 CDCL 核在这些区域上搜索，同时由连续理论求解器检查每个约束在相应乘积区域内是否可行。理论求解器直接在复平面中运作：通过用凸多边形包围环形扇区，并将其通过闵可夫斯基和与乘积进行组合，我们得到一个可计算的可达局部振幅集合的过近似；若该集合不包含原点，即可可靠地得出该区域为 UN-PRODSAT 的结论。

两个组件之间的交互由(Q,D)-有效阻塞子句进行协调。当发现一个全赋值是(Q,D)-无效的（也就是说，某个约束在诱导出的角区域内任何位置都无法满足）时，我们会生成一个排除该区域的子句，并将其加入子句数据库。引理 4 概括了由此得到的可靠性论证：若(Q,D)-有效子句的合取不可满足，即可证明原实例为 UN-PRODSAT。同时，如果无法在候选区域内反驳任何约束，我们的过程便返回 MAYBE(A, ρ)，其中A 和 ρ 分别表示相关集合过近似的面积和最大模平方。这些量可启发式地表明剩余不确定性的“紧密”程度：数值越小，表明实例为 PRODSAT 的证据越强；而任何报告为 UN-PRODSAT 的结果都是真正成立的结果。

我们还提供了一种实用算法（算法 3），并以 Rust 实现，同时支持浮点运算和分圆域上的精确表示。实验表明，该方法能够生成与预期乘积态可满足性范围定性一致的 UN-PRODSAT 结果。尽管如此，在深度固定时，最坏情况复杂度仍随量子比特数呈指数增长，而且我们的方法不可避免地会在某些实例上得出无法确定的结果。在这种情况下，仍可采用精确代数方法（例如布赫伯格算法 [15]）来获得确定答案，但最坏情况开销会显著更高。

理论方面的若干改进途径或可同时提升性能和结论的确定性。通过构造更紧的包络，可以减少 MAYBE 结果，尤其是利用单量子比特态两个振幅之间的相关性；目前，独立的区间界忽略了这种相关性。将这些思路从乘积态拟设推广到表达能力更强的结构化态族，也是一个值得探索的研究方向。

致谢

Joon Hyung Lee 感谢荷兰研究委员会（NWO）通过研究计划利用人工智能加速新量子算法的搜索（*BoostQA*）提供支撑，项目编号为 NGF.1623.23.033。该项目属于量子技术 *2023*研究计划。

Samuel Gonz´alez-Castillo 获得了西班牙大学部提供的 FPU资助（学术人员培训计划，FPU22/02020）。他还获得了西班牙政府经济事务与数字化转型部通过西班牙国家网络安全研究所（INCIBE）发起的西班牙网络安全战略项目资助，并获得了欧盟通过复苏、转型与韧性计划——NextGenerationEU，在数字西班牙2026议程框架下提供的资助。

Financiado por la Unión Europea NextGenerationEU
GOBIERNO DE ESPAÑA MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA
Plan de Recuperación, Transformación y Resiliencia
incibe_ INSTITUTO NACIONAL DE CIBERSEGURIDAD

参考文献

[1] Grasp：一种用于命题可满足性的搜索算法。*IEEE*计算机汇刊，48(5):506–521，2002。[2] Alex M Andrew。一个用于二维凸包的高效算法。信息处理快报，9(5):216–219，1979。[3] Gilles Audemard 和 Laurent Simon。论 Glucose SAT 求解器。国际人工智能工具期刊，27(01):1840001，2018。[4] Armin Biere、Tobias Faller、Katalin Fazekas、Mathias Fleury、Nils Froleyks 和 Florian Pollitt。CaDiCaL 2.0。载于国际计算机辅助验证会议，第133–152页。施普林格，2024。[5] Sergey Bravyi。一种用于量子2-SAT的高效算法。当代数学，536:33–48，2011。[6] Simon Cruanes 和 Masaki Hara。BatSat：一个（参数化的）Rust SAT求解器，最初基于minisat。https://github.com/c-cube/batsat。[7] Niel de Beaudrap 和 Sevag Gharibian。一种用于量子2-SAT的线性时间算法。第*31*届计算复杂性会议论文集，50:27:1–17:21，2016。[8] Niklas E´en 和 Niklas S¨orensson。一种可扩展的SAT求解器。载于可满足性测试理论与应用国际会议，第502–518页。施普林格，2003。[9] David Gosset 和 Daniel Nagaj。量子3-SAT是QMA_1-完全问题。*SIAM*计算杂志，45(3):1080–1128，2016。[10] Christoph Jabs。RustSAT：一个用于Rust中SAT求解的库。载于Jeremias Berg 和 Jakob Nordstr¨om（编），第*28*届可满足性测试理论与应用国际会议（*SAT 2025*），第341卷，莱布尼茨信息学国际会议论文集（*LIPIcs*），第15:1–15:13页，德国达格施图尔，2025年。达格施图尔城堡——莱布尼茨信息学中心。[11] David G. Kirkpatrick 和 Raimund Seidel。终极平面凸包算法？*SIAM*计算杂志，15(1):287–299，1986。[12] Alexei Yu Kitaev、Alexander Shen 和 Mikhail N. Vyalyi。经典计算与量子计算。第47号。美国数学会，2002。[13] Jean-Claude Latombe。机器人运动规划，第124卷。施普林格科学与商业媒体，2012。[14] Steven R. Lay。凸集及其应用。纯粹与应用数学。威利，第一版，1982。[15] Joon Lee、Nicolas Macris、Jean Bernoulli Ravelomanana 和 Perrine Vantalon。随机量子可满足性的Prodsat相，2024年。

[16] Joao Marques-Silva、Inˆes Lynce 和 Sharad Malik。冲突驱动子句学习 SAT 求解器。可满足性手册，第 131–153 页，2009 年。[17] Jo˜ao P. Marques-Silva、Inˆes Lynce 和 Sharad Malik。冲突驱动子句学习 SAT 求解器。载于 Armin Biere、Marijn Heule、Hans van Maaren 和 Toby Walsh（编），可满足性手册，人工智能与应用前沿第 185 卷，第 133–182 页。IOS Press，第二版，2021 年。[18] Lin Zhou。Shapley–Folkman 定理的一个简单证明。经济理论，3(2)：371–372，1993 年。[19] G ̈unter M Ziegler。多胞体讲义，第 152 卷。施普林格科学与商业媒体，2012 年。