arXiv:2302.02188v1 [math.OC] 4 Feb 2023

有限阿贝尔群上函数的下界

Jiangting Yang 数学机械化重点实验室，AMSS 中国科学院大学 北京，100190，中国
yangjianting@amss.ac.cn

Ke Ye 数学机械化重点实验室，AMSS 中国科学院大学 北京，100190，中国
keyk@amss.ac.cn

Lihong Zhi 数学机械化重点实验室，AMSS 中国科学院大学 北京，100190，中国
lzhi@mmrc.iss.ac.cn

摘要

在有限集上计算函数最优解的问题，是数学和计算机科学中的一个重要问题。许多组合问题，例如最大可满足性问题和最大割问题，都可以视为超立方体 $C_2^n = \{-1, 1\}^n$ 上的优化问题，该超立方体由 2^n 个点组成。人们已经注意到，如果有限集具有阿贝尔群结构，那么可以通过傅里叶平方和（FSOS）高效地认证其上的非负函数。受这些工作的启发，本文致力于建立一个框架，以高效地求出有限阿贝尔群上函数的下界。我们使用半正定规划求解器 SDPNAL+ 实现了该算法，用于计算 f 在 G 上的经验证下界，并在 2009年和 2016年MAX-SAT竞赛中的最大 2-SAT 和 MAX-3SAT 基准问题上进行测试。此外，我们还在 C_3^n 上的随机函数上测试了该算法。这些实验表明，我们的算法优于以往已知的方法。

关键词

阿贝尔群，傅里叶平方和表示，半定规划，稀疏格拉姆矩阵，SAT，最大可满足性问题

ACM 引用格式：

Jiangting Yang、Ke Ye 和 Lihong Zhi。2023年。有限阿贝尔群上函数的下界。载于 *ACM*会议论文集（*Conference'17*）。ACM，纽约，纽约州，美国，10 页。https://doi.org/10.1145/nnnnn.nnnn

1 引言

计算有限集上函数最优解的问题，是数学和计算机科学中的一个重要而又具有挑战性的问题。此类问题包括背包问题[30, 31],集合覆盖问题[11, 23],旅行商问题[19,39], 车辆路径问题[9, 45], 最大 k-SAT 问题 [6, 21] 及其众多变体。尽管这些问题都可以表述为整数规划，但除非**P== NP**[21, 23, 24, 34, 35, 45], 否则大多数问题都不存在多项式时间算法。因此，人们采用各种近似算法来解决这一问题 [5, 15, 28, 42, 46, 47]。半定规划（SDP）是设计和分析近似算法的最有力且研究最广泛的技术之一[3, 4, 13, 20, 22, 25, 27, 37, 43, 46, 54]。在 SDP 技术的成功应用中，最著名的有最大 2-SAT[16], MAX-3SAT[20] 和最大割[22]。另一方面，文献 [12, 41, 52, 53] 指出，如果一个有限集具有阿贝尔群结构，那么就可以通过傅里叶平方和（FSOS）高效地认证其上的非负函数。受这些工作的启发，本文旨在建立一个框架，利用 FSOS 和 SDP 来解决以下问题。

问题1.1（由 FSOS 给出的下界）。给定有限阿贝尔群上的函数 f，高效地求出 f 的下界。

设 $S \subseteq \mathbb{C}^n$ 为一个代数簇

从代数角度看，将 $\mathbb{C}[S]$ 视为 $\mathbb{C}[z_1, \ldots, z_n]/I(S)$ 是一种有利的观点，因为后一个环具有丰富的几何和代数结构。然而，出于计算目的，将函数视为一个等价类并不方便，因为同一个等价类可以由两个不同的多项式表示。如果 $S = G$ 是一个有限群，那么 $\mathbb{C}[G]$ 上还有另一种极其有助于计算的代数结构[12, 41, 52, 53]。具体而言，可以通过傅里叶变换 [14]，将 $\mathbb{C}[G]$ 与 G 的群环对应起来。这种观点的优点在于，G 上的函数 f 可以展开为 $f = \sum_{\chi \in \widehat{G}} f(\chi)\chi$，

其中 $\widehat{G}$ 是 G 的对偶群，且 $f(\chi)$ 是 f 在 $\chi \in \widehat{G}$ 处的傅里叶系数。这一广为人知的观点使我们能够引入解析工具来求解问题 1.1。

相关工作与我们的贡献

近年来，文献 [49] 提出了一种名为 TSSOS 层级的通用多项式优化方法。该新方法遵循文献 [26], 中确立的经典方法论，但利用多项式的稀疏性来缩减半定规划的规模。结合 TSSOS 层级与文献 [48] 中针对相关稀疏性的方法，[50] 提出了适用于大规模稀疏多项式优化的 CS-TSSOS 层级。特别地，TSSOS 和 CS-TSSOS 层级均适用于有限阿贝尔群上的优化问题。

许多组合问题，例如最大可满足性问题和 MAX-CUT，都可以看作超立方体 $C_2^n = \{-1,1\}^n$ 上的优化问题。由于其在计算机科学中的重要性，人们为最大可满足性问题开发了各种求解器。例如，在 [46], 中，研究了 $\mathbb{C}[C_2^n]$ 的商结构以制定支撑集选择策略，由此可以通过半定规划求解最大可满足性问题；通过结合

特此免费授予许可，允许为个人或课堂使用而制作本作品全部或部分内容的数字副本或纸质副本，前提是不得为营利或商业利益而复制或分发，且副本须附有本声明及首页上的完整引文。对于本作品中归他人而非 ACM 所有的组件，必须尊重其版权。注明出处后允许摘录。若要以其他方式复制、重新发表、发布到服务器或分发至列表，须事先获得特别许可和/或支付费用。请向 permissions@acm.org 申请许可。*Conference'17*，*2017*年7月，美国华盛顿特区©2023年 计算机协会。ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnn.nnnn

专为最大可满足性问题设计的几种优化技术，[51] 提出了一种高效的基于半定规划的 MIXSAT 算法；基于归结反驳，[36]中提出了一种名为 MS-builder 的求解器。

一方面，TSSOS 和 CS-TSSOS 层级可以处理一般的多项式优化问题，而 MIXSAT 和 MS-builder 等专门设计的求解器只能处理最大可满足性问题。另一方面，然而，我们自然会认为，在最大可满足性问题上，这些专门设计的求解器会优于通用方法。

我们的 FSOS 优化框架兼顾了通用性与效率。事实上，我们的方法适用于有限阿贝尔群上的任意优化问题，包括超立方体 C_2^n、循环群 $\mathbb{Z}_N$ 及其乘积。

本文的主要贡献是提出了一种近似算法，用于求阿贝尔群 f 上函数 G的最小值。具体而言，对于每个正整数 k，我们提出一个与某些 $S_f \subseteq \widehat{G}$ 相关的半定规划（引理 3.2）（算法 1），其中$|S_f| = k$。该半定规划的解为 f提供了一个良好的下界（定理 3.5）。由于该半定规划的规模为 $k \times k$，我们的近似算法适用于定义在大型阿贝尔群上的函数。我们方法的另一特点是用于该半定规划的支撑集选择策略。不同于 [46], 中提出的策略（该策略对于超立方体上的所有函数几乎相同），我们的策略利用傅里叶系数的大小，为不同函数提供不同的基。此外，我们还分别基于格拉姆矩阵的零空间和矩矩阵的低秩近似，提出了两种舍入方法。我们在附录2中的数值示例上测试了算法（算法 A）和舍入方法，从中可以清楚地看到，与上述方法相比，我们的算法具有明显优势。

应用

下面我们介绍该算法的一些潜在应用。尽管我们可以在所有这些有趣的问题上测试该算法，并在附录A中展示数值实验，但受篇幅限制，我们只关注最大 2-SAT 和 MAX-3SAT。

可满足性问题。合取范式公式的特征函数（参见第2.3小节）可以看作群 k 上的非负函数，其中k-合取范式[1]公式包含 n个变量C_2^n。因此，我们的框架为最大kSAT[2] 及其变体（如不可满足性问题和最小可满足性问题）提供了一种近似算法。

最大割问题。设 $G = (V, E)$ 是一个无向图，其中 $|V| = n$ 的边权重为 $\{w_{ij} : \{i, j\} \in E\}$。对于 G，最大割问题是要找到一个划分 $V = S \sqcup T$，使得连接 S 和 T 的边的权重之和尽可能大。因此，它可以表述为 $\mathbb{Z}_2^n$ 上的一个优化问题。

$$\max \sum_{\{i,j\}\in E} w_{ij} x_i (1 - x_j)$$
$$s.t.\ x_i \in \mathbb{Z}_2 = \{0, 1\}, 1 \le i \le n$$

格点模型的基态。格点模型是在 $\mathbb{R}^3$ 中固定的一组位置，粒子分布在这些位置上并相互作用。格点模型的基态是使哈密顿量最小化的粒子构型。从数学上说，求格点模型的基态可以改写为关于 $\mathbb{Z}_{d_1} \times \cdots \times \mathbb{Z}_{d_n}$ 的优化问题，其中 n 是粒子数，而 d_j 由第 j 个粒子的类型决定。在 [18], 中，考虑 $d_1 = \cdots = d_n = 2$ 的情况。

2 预备知识

本节首先回顾阿贝尔群上的傅里叶分析，随后简要介绍阿贝尔群上的傅里叶平方和（FSOS）。最后，我们定义（加权） k-合取范式公式的特征函数，它们是阿贝尔群上整数值函数的重要例子。

2.1 群上的傅里叶分析

本小节简要概述群论和表示论的基础知识。有关更多细节，请读者参阅 [14, 33, 40]。

设 G是一个有限阿贝尔群。 G的一个特征标 是一个群同态 $\chi\colon G \to \mathbb{C}^\times$。这里， $\mathbb{C}^\times$ 是赋予复数乘法这一群运算的 $\mathbb{C}\setminus\{0\}$。 G的所有特征标构成的集合记为 $\widehat{G}$，称为 G的对偶群[3]。

根据 [14, 第 1], 章， G 上的任意函数 f都可以表示为傅里叶展开：

$$f = \sum_{\chi\in\widehat{G}} \widehat{f}(\chi)\chi,$$

其中， $\widehat{f}(\chi) := \frac{1}{|G|}\sum_{g\in G} f(g)\overline{\chi(g)}$称为 f在 $\chi \in \widehat{G}$处的傅里叶系数。 f的支撑集 为

$$\operatorname{supp}(f) := \left\{\chi \in \widehat{G} : \widehat{f}(\chi) \neq 0\right\}.$$

例如，超立方体 $C_2^n = \{-1, 1\}^n$的对偶群是

$$\widehat{C_2^n} = \{z^\alpha : \alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{Z}_2^n\} \simeq \mathbb{Z}_2^n.$$

这里， $\mathbb{Z}_2 = \mathbb{Z}/2\mathbb{Z} = \{0, 1\}$是加法群，且对每个 $\beta \in \mathbb{N}^n$都有 $z^\beta := z_1^{\beta_1} \ldots z_n^{\beta_n}$。因此，函数 $f\colon C_2^n \to \mathbb{C}$ 可以表示为多线性单项式的线性组合：

$$f = \sum_{\alpha\in\mathbb{Z}_2^n} f_\alpha z^\alpha.$$

2.2 有限阿贝尔群上的傅里叶平方和

本小节讨论 [12, 41, 52] 中发展的 FSOS 理论。设 f 是有限阿贝尔群 G 上的一个非负函数。 f 的 *FSOS* 证书 是复值函数 $_{i\in I}$ 的一个有限族 $\{h_i\}$，这些函数定义在 G 上，并满足 $f = \sum_{i\in I} |h_i|^2$。其 稀疏度 为

[1] A k-合取范式公式是指每个子句至多包含 k 个文字的合取范式公式。[2] 这里我们采用最常见的 [24] 最大-kSAT 定义，即要求计算 k-合取范式公式中可同时满足的子句的最大数量。不过，最大-kSAT 还有其他定义。例如， [2] 将其定义为一个**NP**-困难问题，即寻找一个极大可满足赋值的问题。在 [35], 中，MAX-kSAT 指的是**NP**-完全问题，其目标是判定是否存在一个赋值，能够满足至少给定数量的子句。

[3] 由于 G 是一个阿贝尔群， $\widehat{G}$ 确实是一个群，并且 $\widehat{G} \simeq G$。

$\{h_i\}_{i\in I}$ 定义为 $|\bigcup_{i\in I}\mathrm{supp}(h_i)|$。如果 f 存在一个稀疏度较低的 FSOS 证书，我们就称 f 是稀疏的。下面的命题确保了 FSOS 证书的存在性。

命题 2.1。 *[12,命题 2]* 有限阿贝尔群上的非负函数 f 存在 *FSOS* 证书。

函数 f 在 G 上非负，当且仅当存在厄米正半定矩阵 $Q=(Q_{\chi,\chi'})_{\chi,\chi'\in\widehat{G}}\in\mathbb{C}^{|G|\times|G|}$ 使得：

$$\sum_{\chi'\in\widehat{G}} Q_{(\chi')^{-1}\chi,\chi}=f_\chi,\quad \forall\chi\in\widehat{G}. \tag{1}$$

任意 $|G|\times|G|$ 复矩阵 $Q\succeq 0$ 若满足（1），则称为 格拉姆矩阵，对应于 f。格拉姆矩阵在 FSOS 的理论研究和计算研究中都非常重要。事实上，如果 Q 是 f 的格拉姆矩阵，且 $Q=M^*M$ 对于某个矩阵

$$M=(M_{j,\chi})_{1\le j\le r,\chi\in\widehat{G}}\in\mathbb{C}^{r\times|G|},$$

那么我们有 $f=\sum_{j=1}^r\left|\sum_{\chi\in\widehat{G}}M_{j,\chi}\chi\right|^2$。显然，这种构造在（稀疏）格拉姆矩阵与 f 的（稀疏）FSOS 证书之间建立了一一对应关系。因此，计算 f 的稀疏 FSOS 证书的问题，等价于计算 f 的稀疏格拉姆矩阵的问题。

定理 2.2。 *[52, 引理 3.6]* 设 f 是有限阿贝尔群 G 上的非负非零函数。假设

$$\sqrt{f}=\sum_{\chi\in\widehat{G}}a_\chi\chi$$

是 $\sqrt{f}$ 的傅里叶展开。那么，

$$\min_{Q:\text{gram matrix of }f}\|Q\|_0$$

的凸松弛的最优解是秩一半正定矩阵 $H\in\mathbb{C}^{|G|\times|G|}$，其中

$$H_{\chi,\chi'}=\overline{a_\chi}a_{\chi'}.$$

这里，我们再次分别以 $\widehat{G}$ 中的元素为 H 的行和列编索引。

通过快速傅里叶变换（FFT）和逆快速傅里叶变换（iFFT），$\sqrt{f}$ 可以在准线性时间内计算，关于 $|G|$。然而，如果群 $G=C_2^n$，那么计算 $\sqrt{f}$ 的复杂性为 $O(n\cdot 2^n)$，这关于 n 是指数级的。在第 3 节中，我们提出一种高效估计 $\sqrt{f}$ 的方法。

2.3 整数值函数与加权 k-合取范式公式的特征函数

设 G 为一个有限阿贝尔群。本文中，定义在 G 上的整数值函数[4] 是指满足以下条件的函数 $f\colon G\to\mathbb{Z}$：对于所有 $g\in G$，

$|f\ g\ |=O$ $|G|$ () (polylog( ))。

[4]对 f 取值范围上界的假设，是用来控制算法 1 的复杂性。除此之外，本文中的所有结论实际上对任意 $f\colon G\to\mathbb{Z}$ 都成立。

整数值函数的典型例子是加权 k-合取范式公式的特征函数。一个加权 k-合取范式公式 ϕ 由逻辑子句组成

$$\left\{\left(\vee_{i\in S_j^+}x_i\right)\vee\left(\vee_{i\in S_j^-}\neg x_i\right)\right\}_{j\in J}$$

以及整数权重 $\{w_j\}_{j\in J}$。这里，x_i 是取值为 {True，False} 的逻辑变量，而 $\vee$，$\neg$ 分别是逻辑运算“或”和“非”。我们指出，若对所有 $j\in J$ 都有 $w_j=1$，则 ϕ 就是一个 k-合取范式 [10]：

$$\sum_{j\in J}\left(\vee_{i\in S_j^+}x_i\right)\vee\left(\vee_{i\in S_j^-}\neg x_i\right).$$

我们定义 ϕ 的特征函数如下：

$$f_\phi(z_1,\ldots,z_n)=\sum_{j\in J}\frac{w_j}{2^l}\left(\prod_{i\in S_j^+}(1+z_i)\cdot\prod_{i\in S_j^-}(1-z_i)\right),$$

其中 $(z_1,\ldots\ldots,z_n)\in C_2^n$。显然，对于 ϕ，加权 MAX-SAT 问题等价于求 f_ϕ 的最小值。

若定义

$$\tau:\{\text{False},\text{True}\}\to C_2,\quad \tau(\text{False})=1,\tau(\text{True})=-1,$$

那么，k-合取范式公式的特征函数具有以下性质。

命题 2.3。 *[52]* 设 ϕ 是一个含有 n 个变量和 m 个子句的 k-合取范式公式，且 f_ϕ 是 ϕ 的特征函数。那么

(i) 对于 $x_1,x_2,\cdots x_n\in\{False, True\}$，$f_\phi(\tau(x_1)\ \tau(x_2),\cdots\ \tau(x_n))$ 是 ϕ 中被 $x_1,x_2,\cdots x_n$ 判为假的子句数。*(ii)* f_ϕ 的次数至多为 k。*(iii)* $supp(f_\phi)$ 的基数至多为 $min\left\{2^km,\sum_{i=0}^k\binom{n}{i}\right\}$。*(iv)* f_ϕ 可通过 $O(2^kkm)$ 次运算计算得到。特别地，若固定 k，则 f_ϕ 可通过 $O(m)$ 次运算计算得到。*(v)* 对于每个 $x\in\{False, True\}^n$，$f_\phi(\tau(x))$ 是 ϕ 中被 x 判为假的子句数。

(vi) f_ϕ 的像集包含于$\{0,\ldots,m\}$。

此外，以下陈述等价：

(a) ϕ 的可满足子句数至多为$(m-L)$。*(b)* ϕ 的被赋值为假的子句数至少为 L。*(c)* 对某个函数 $\delta\colon C_2^n\to[0,1)$，$f_\phi-L+\delta\ge 0$。*(d)* 对任意函数 $\delta\colon C_2^n\to[0,1)$，$f_\phi-L+\delta\ge 0$。

根据命题2.3，求解 CNF 公式 ϕ 的最大可满足性问题等价于求解问题1.1，其中 f_ϕ。

3 FSOS 下界

本节中，我们给出整数值函数 f 在 G 上的问题1.1 的解法。在继续之前，值得说明的是，若已知 L 的一个下界 f，则可以通过计算 $f-L$ 的一个正下界，得到 f 的一个更好的下界 $f-L$，除非 L 已经是 f 的最小值。

我们对问题1.1 的解法基于下面的引理。

引理 3.1。 设 $f\colon G\mapsto\mathbb{R}$ 是有限阿贝尔群 G 上的函数，α_0 是一个实数。那么 α_0 是 f 的下界，当且仅当存在一个由复值函数构成的有限族$\{h_i\}_{i\in I}$

在 G上，使得 $f-\alpha_0=\sum_{i\in I}|h_i|^2$。因此， f的下界集合与以下可行集之间存在一一对应关系：

$$\max_{h_i:G\mapsto\mathbb{C},i\in I}\alpha,\quad s.t.\ f-\alpha=\sum_{i\in I}|h_i|^2. \tag{2}$$

3.1 支撑集选择

我们注意到， G子集的包含关系在（2）的可行集上诱导出一个滤过。也就是说，对于每个 $S\subseteq\widehat{G}$，我们定义

$$F_S:=\left\{\{h_i\}_{i\in I}:f-\alpha=\sum_{i\in I}|h_i|^2 \text{ for some } \alpha,\bigcup_{i\in I}\mathrm{supp}(h_i)\subseteq S\right\}.$$

于是我们有

$$\text{feasible set of (2)}=F_{\widehat{G}}:=\bigcup_{S\subseteq\widehat{G}}F_S \tag{3}$$

且 $F_{S_1}\subseteq F_{S_2}$当 $S_1\subseteq S_2\subseteq\widehat{G}$时。因此，对于给定的 $S\subseteq\widehat{G}$，我们可以考虑以下问题：

$$\max_{\{h_i\}_{i\in I}\in F_S}\alpha. \tag{4}$$

根据下面的引理，(4) 可以改写为一个半定规划。

引理 3.2。 对于每个 $S\subseteq G$， (4) 等价于

$$\max_{Q\in\mathbb{C}^{S\times S}}\ \widehat{f}(\chi_0)-\mathrm{trace}(Q), \tag{5}$$

$$s.t.\quad \sum_{\chi'\in\widehat{G}}Q_{(\chi')^{-1}\chi,\chi}=\widehat{f}(\chi),\ \chi\neq\chi_0\in\widehat{G} \tag{6}$$

$$Q\succeq 0. \tag{7}$$

这里， $\chi_0\in\widehat{G}$表示 $\widehat{G}$中的单位元， $Q\in\mathbb{C}^{S\times S}$表示$Q$是一个矩阵，其大小为 $|S|$，行和列均由 S中的元素索引。

证明。条件 (6) 和 (7) 确保 Q 是 $f-\widehat{f}$的格拉姆矩阵（χ_0）， $+$迹为Q。设 Q是 (5) 的一个可行解，并令$Q=H^*H$，其中 $H\in\mathbb{C}^{r\times S}$。我们设 $h_i=\sum_{\chi\in S}H(i,\chi)\chi$， $i=1,\ldots\ldots,r$。那么，$\{h_i\}\in F_S$是 (4) 的一个可行解，且 $\alpha=f(\chi_0)-$迹为Q。反过来，若$\{h_i\}_{i\in I}$是 (4) 的一个可行解，则可以定义 $H\in\mathbb{C}^{I\times S}$，其中 $H(i,\chi)=\widehat{h_i}$（χ）。因此， $Q=H^*H$是 (4) 的一个可行解，且 $\widehat{f}$（χ_0）$-$迹为Q $=\alpha$。□

根据引理 3.1 和 (3)，(5) 的每个最优解都给出了 f 的一个下界。这一下界的质量取决于 S 的选择。例如，若对于 (2) 的某个最优解 $\{h_i\}_{i\in I}$，有 $S\supseteq\bigcup_{i\in I}\mathrm{supp}(h_i)$，则显然，(5) 的一个最优解给出的下界是 f 的最小值。此外，(5) 的规模完全由 $|S|$ 决定。因此，为了高效地计算 f 的一个较好下界，我们需要选择较小的 S，使其对于 (2) 的某个最优解 $\{h_i\}_{i\in I}$，尽可能多地包含在 $\bigcup_{i\in I}\mathrm{supp}(h_i)$ 中的元素。本节剩余部分将在这一原则的指导下，讨论如何选择 $S\subseteq\widehat{G}$。

设 f是阿贝尔群 G 上的一个非负函数。我们可以将 $\sqrt{f}$ 展开为：

$$\sqrt{f}=\sum_{i=1}^{|G|}a_i\chi_i,$$

其中 $|a_1|\geq|a_2|\geq\cdots\geq|a_{|G|}|$且 $\widehat{G}=\{\chi_i\}_{i=1}^{|G|}$。根据定理2.2，$S=\{\chi_i\}_{i=1}^{k}$已经是 S（5）的一个合适选择。这里 $k\leq n$是某个给定的正整数。然而，正如我们在小节 2.2末尾所指出的，计算 $\sqrt{f}$的傅里叶展开的复杂性关于 $|G|$是准线性的；若 $G=C_2^n$，这会导致关于 n的指数级复杂性。因此，为了降低复杂性，我们可以近似计算 $\sqrt{f}$的傅里叶展开，而不是精确计算它。为此，需要估计近似误差，这正是下一个命题的内容。

命题 3.3。 *[53, 命题 4.12]* 设 f 是 G 上的一个非负函数，并设 $\varepsilon>0$ 是一个固定数。若 p 是一个单变量多项式，满足最大值$_{g\in G}\left|p(f(g))-\sqrt{f(g)}\right|\leq\varepsilon$，则有

$$\max_{\chi\in\widehat{G}}\left|\widehat{p\circ f}(\chi)-\widehat{f}(\chi)\right|\leq\varepsilon.$$

根据上述讨论，我们可以通过算法 S构造一个基数为 $|S|=k$的1。如果 f是整数值的，且 $l\leq f\leq m$

算法1 支撑集的选择

输入： 非负函数 f，定义在 G上；参数 l、 m、 d、 $k\in\mathbb{N}$。**输出：** $S\subseteq\widehat{G}$，满足（5），其中 $|S|=k$。

1：用次数至多为 d的多项式 $p(t)$在 $[l,m]$中的整数点处近似 $\sqrt{t}$。2：计算复合函数 $p\circ f=\sum_{i=1}^{|G|}a_i\chi_i$，其中 $|a_1|\geq\cdots\geq|a_{|G|}|$且 $\chi_i\geq\chi_{i+1}$ 若 $|a_i|=|a_{i+1}|$，$1\leq i\leq|G|-1$，则 $\deg(\ )\geq\deg(\ )$。**返回** $S=\{\chi_1,\ldots\ldots,\chi_k\}$。

对于某个 $0<l\leq m$，那么通过求解一个线性规划 p，便可轻松计算出算法1 的第1 步 [53]中的多项式。提醒读者注意， $f:G\to\mathbb{Z}$满足 f（g）$=O$（polylog（$|G|$））（参见第2.3小节）。因此，算法1 的第1步可在 O（polylog（$|G|$））时间内完成。

根据命题 3.3， $p\circ f$是 $\sqrt{f}$的一个估计，其系数误差上界为 $\max_{l\leq i\leq m}\left|p(i)-\sqrt{i}\right|$。这保证了算法1得到的集合 S确实由 $\sqrt{f}$的前 k项组成，前提是 $p(t)$在 l和 m之间的整数点处对 $\sqrt{t}$的近似足够好。此外，实践中只需令 d足够小。事实上，如果 f是某个合取范式公式的特征函数，那么 $d=1$ 或 2 已经能为我们提供良好的下界，正如我们将在附录 A 中看到的那样。

令 $f_{\text{最小值}}$为 f在 G上的最小值。我们注意到，一方面，对于每个 $a\leq f_{\text{最小值}}$，我们都可以应用算法 1 对 $f-a$ 进行处理，从而得到一个 $S_a\subseteq\widehat{G}$。另一方面，根据（2）、（4）和（5），当且仅当S 包含 $S_{f_{\min}}$时，我们才能从（5）的最优解中恢复$f_{\text{最小值}}$。下一个命题的推论是，对于每个$a\leq f_{\text{最小值}}$， S_a 都包含 $S_{f_{\min}}$ 中的元素，且这些元素在f中的系数足够大。特别地，如果 f非负，那么我们甚至可以取 S_0 作为 $S_{f_{\min}}$的估计。

命题 3.4. 设 f 是定义在 G 上的函数，且 $a>\widetilde{a}\geq 0$ 是 f 的下界。假设 c_χ 是 $\sqrt{f-a}$ 中 χ 的系数。若 $|c_\chi|>\sqrt{a-\widetilde{a}}$，则 $\chi\in$支撑$(\sqrt{f-\widetilde{a}})$。此外，若 $G=C_2^n$ 且 $f\leq m$ 对某个 $m\in\mathbb{R}$ 成立，则 $|c_\chi|>2^{\frac{a-\widetilde{a}}{2}}\left(\frac{1}{\sqrt{a-\widetilde{a}}}-\frac{1}{\sqrt{m-a+\sqrt{m-\widetilde{a}}}}\right)$蕴含 $\chi\in$支撑$(\sqrt{f-\widetilde{a}})$。

证明。回顾一下，$c_\chi := \frac{1}{|G|}\sum_{g\in G}\chi(g)\sqrt{f(g)-a}$。类似地，我们可以写成 $\sqrt{f-\widetilde{a}} = \sum_{\chi\in\widehat{G}}\widetilde{c_\chi}\chi$。对于每个 $\chi\in\widehat{G}$，注意到

$$\begin{aligned}|c_\chi - \widetilde{c_\chi}| &= \frac{1}{|G|}\Big|\sum_{g\in G}\overline{\chi(g)}\left(\sqrt{f(g)-a}-\sqrt{f(g)-\widetilde{a}}\right)\Big| \\ &= \frac{a-\widetilde{a}}{|G|}\Big|\sum_{g\in G}\frac{\overline{\chi(g)}}{\sqrt{f(g)-a}+\sqrt{f(g)-\widetilde{a}}}\Big| \\ &\le \sqrt{a-\widetilde{a}}.\end{aligned}$$

因此，如果 $|c_\chi| > \sqrt{a-\widetilde{a}}$那么 $\widetilde{c_\chi}\neq 0$。

如果 $G = C_2^n$，那么 $\chi\in\widehat{G}\simeq\mathbb{Z}_2^n$是一个多线性单项式。因此，对于任意 $g\in C_2^n = \{-1,\ 1\}^n$，$\chi(g) = \pm 1$。此外，直接验证可知，我们可以构造一个双射 $\psi\colon\ C_2^n\to C_2^n$，使得 $\chi(g) = -\chi(\psi(g))$ 且 $\psi^2 = \mathrm{Id}$。因此我们得到

$$\begin{aligned}|c_\chi - \widetilde{c_\chi}| &= \frac{a-\widetilde{a}}{|G|}\Big|\sum_{g\in G}\frac{\overline{\chi(g)}}{\sqrt{f(g)-a}+\sqrt{f(g)-\widetilde{a}}}\Big| \\ &\le \frac{a-\widetilde{a}}{|G|}\sum_{g\in G,\chi(g)=1}\left(\frac{1}{\sqrt{a-\widetilde{a}}}-\frac{1}{\sqrt{m-a}+\sqrt{m-\widetilde{a}}}\right) \\ &= \frac{a-\widetilde{a}}{2}\left(\frac{1}{\sqrt{a-\widetilde{a}}}-\frac{1}{\sqrt{m-a}+\sqrt{m-\widetilde{a}}}\right).\end{aligned}$$

□

3.2 含错误的 FSOS

本小节将讨论我们对问题 1.1 的解（参见第 3 节）的一个显著特征。也就是说，一个违反条件（6）和（7）的半正定规划问题（5）的解，仍然可以为我们提供 f 的紧下界。这就是下一个定理的内容。

定理 3.5（带误差的FSOS）。设 G 为有限阿贝尔群，且 S 是 $\widehat{G}$ 的一个子集。给定函数 $f\colon\ G\mapsto\mathbb{R}$ 和厄米矩阵 $Q\in\mathbb{C}^{S\times S}$，我们有

$$\min_{g\in G} f(g) \ge -\|\widehat{e}\|_1 + \lambda|S|,$$

其中，λ 是 Q的最小特征值，$e = f - v_S^* Q v_S$，而 $v_S = (\chi)_{\chi\in S}$ 是由 S中的所有特征标组成的向量。

证明。对于任意 $g\in G$，我们有

$$f(g) - e(g) = v_S(g)^* Q v(g) \ge \lambda v_S(g)^* v(g).$$

我们注意到，对于每个 $g\in G$和 $\chi\in\widehat{G}$，都有 $|\chi(g)| = 1$。因此，$v_S(g)^* v_S(g) \le |S|$，且对于任意 $g\in G$，都有 $|e(g)|\le\|\widehat{e}\|_1$。这意味着

$$f(g) \ge -\|\widehat{e}\|_1 + \lambda|S|,\quad g\in G.$$

□

根据定理3.5，$-\|\widehat{e}\|_1 + \lambda|S|$是 f的下界，即使 $Q\not\succeq 0$或 $e\neq 0$也是如此。下面的例子表明，即使 f不具有支撑在 S上的FSOS，我们的方法也能给出其紧下界。

例3.6。令 $f\colon\ C_2^3\to\mathbb{R}$ 为如下定义的函数

$$f(z_1,z_2,z_3) = 4 + z_1 + z_2 + z_3 + z_1z_2z_3.$$

我们可以验证，半定规划 (5) 对于 $S = \{1、\ z_1、\ z_2、\ z_3\}$没有可行解，即 f没有支撑集为 S的 *FSOS*。然而，若令$e(z_1,\ z_2,\ z_3) := z_1z_2z_3$，则

$$f - e = 1 + \frac{1}{2}\sum_{i=1}^{3}(1+z_i)^2,$$

由此我们得到 $f\ge 1 - \|\widehat{e}\|_1 = 0$。

下面我们记录定理 3.5 的一个代数版本，这正是第 4 节所需的形式。

推论 3.7。设 f、S、Q、e、λ如上定义。则 $Q-\lambda\mathrm{Id}\succeq 0$是函数 $f - e - \lambda|S|$ 的一个格拉姆矩阵。

我们注意到，f可以表示为多项式函数 $F_{Q-\lambda\mathrm{Id}} + F_{\lambda\mathrm{Id}}$，其中 $F_M(z) = z^{\mathsf{T}}Mz$是由对称矩阵 M确定的非负多项式。由于 $Q-\lambda\mathrm{Id}\succeq 0$，我们得到 $F_{\lambda\mathrm{Id}}$的最小值为 f提供了一个下界。不幸的是，当 $z\in\mathbb{R}^n$时，如果 $\lambda<0$，$F_{\lambda\mathrm{Id}}$的最小值为 $-\infty$。这一现象使我们的方法有别于传统的多项式优化方法。

例如，我们考虑 $f = 2z_1z_2$ 在 C_2^2上，使得 Q是

$$\begin{array}{c} \\ z_1 \\ z_2 \end{array}\begin{array}{c} \begin{array}{cc} z_1 & z_2 \end{array} \\ \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \end{array}$$

以及 Q的最小特征值为 -1 根据推论3.7，我们得出 $Q+\mathrm{Id}\succeq 0$是 $f+2$的格拉姆矩阵，并且 $f\ge -2$ 在 C_2^2上。作为比较，f可表示为多项式

$$F_{Q+\mathrm{Id}} + F_{-\mathrm{Id}} = (z_1+z_2)^2 - (z_1^2+z_2^2)$$

且 $F_{-\mathrm{Id}} = -(z_1^2+z_2^2)$ 的最小值为 $-\infty$，无法为 f 提供非平凡的下界。

本节最后简要总结带误差的FSOS的主要优点。感兴趣的读者可参阅附录A中的数值示例。

(1) 提前终止：现有方法 [46, 51] 需要等待算法收敛到满足条件（6）和（7）的解。根据定理3.5，即使不满足这些条件，我们的方法也能找到一个下界。这一特性使我们能够为求解半正定规划问题设置不同的时间限制。显然，时间限制设置得越长，得到的下界就越好。

(2) 适用于更多半定规划求解器：由于条件（7）不必精确满足，我们可以使用效率更高的半定规划求解器 SDPNALplus [44]来求解引理 3.2中的问题。因此，可以成功求解基准测试集[5] [6] 中规模大得多的 MAX-kSAT 问题，而本文提出的基于和平方的算法 [46] 无法处理这些问题。

(3) 规模缩减：给定一个单项式基 S，[46]中的算法无法得到下界，若支撑集$(f)\not\subseteq\{\chi\chi'\colon\ \chi,\chi'\in S\}$，因为 f在 S上没有 FSOS 支撑。然而，根据定理 3.5，即使 (5) 的可行解不满足 (6，仍然可以为我们提供下界。这使得我们能够缩减半正定规划问题的规模。

[5] http://www.maxsat.udl.cat/09/index.php [6] http://www.maxsat.udl.cat/16/index.html

4 下界的计算

本节将第 3 节中的讨论转化为一种算法，用于计算函数 f 在有限阿贝尔群 G 上的下界。首先，我们回顾一下，每个定义在 h 上 G 的函数都具有如下傅里叶展开：

$$h = \sum_{\chi \in \widehat{G}} \widehat{h}(\chi)\chi.$$

$\widehat{\ }$为简单起见，我们记作 $h_0 := h(1)$，其中 1 是 $\widehat{G}$中的单位元。

4.1 我们的算法

设 S 是 $\widehat{G}$的一个子集，并设 $Q \in \mathbb{C}^{S\times S}$ 是一个厄米矩阵。我们定义 $e := f - f_0 - v_S^* Q v_S$其中 v_S是由 $|S|$中各特征标构成的 S。将定理3.5 应用于 $f - f_0$，得到

$$f(g) \geq f_0 + e_0 - \|\widehat{e - e_0}\|_1 + \lambda_{\min}(Q)|S|, \quad g \in G$$

其中， $\lambda_{\text{最小值}}(Q)$ 是 Q的最小特征值。我们还注意到

$$e_0 = -\sum_{\chi=\chi'} Q(\chi,\chi') = -\operatorname{trace}(Q).$$

因此，我们可以通过求解以下无约束凸优化问题，得到 f的一个下界：

$$\min_{Q\in\mathbb{C}^{S\times S},\, Q=Q^*} F(Q), \tag{8}$$

其中

$$F(Q) = \operatorname{trace}(Q) + \|E(Q)\|_1 - \lambda_{\min}(Q)|S|$$

并且 $E\colon \mathbb{C}^{S\times S} \mapsto \mathbb{C}^{|G|}$是一个仿射映射，它将 Q映射为由 $e - e_0$的傅里叶系数组成的（稀疏）向量。我们注意到， F的次梯度为

$$\partial F = \mathrm{Id} + \operatorname{sign}(E(Q))\partial E - |S|uu^*,$$

其中 ∂E 是 E的梯度，符号函数(x) 是符号函数，而 u 是 Q 对应于 $\lambda_{\text{最小值}}(Q)$的单位特征向量。因此，我们得到用于求下界的算法2 f。

算法2有限阿贝尔群上函数的下界。

输入 函数 f，定义在 G 上；正整数 l、 m、 d 和 k。
输出 f 的一个下界。

1: 选择满足 l、 m、 d 和 k 的 S。 ▷ 通过算法 1
2: 求解 5）以求 Q_0。 ▷ 通过 SDPNAL+
3: 用8求解 Q。 ▷ 使用梯度下降，初始点为 Q_0
4: **返回** $f_0 - F(Q)$。

4.2 舍入

舍入是组合优化问题中基于半正定规划的算法的重要步骤，尤其是在同时关注最优值和最优点时。文献中已有多种舍入技术。例如，随机超平面舍入法 [15]及其改进版本 [13],偏斜舍入过程 [29] 和随机化舍入技术 [46]。在所有这些舍入策略中， [46] 中提出的方法可以很容易地适用于我们的情形。

我们提出了两种舍入方法：一种基于格拉姆矩阵的零空间，另一种基于矩矩阵的低秩近似。

(1) 设 $Q \in \mathbb{R}^{S\times S}$ 是（5）的一个解。给定 $f\colon\ G \to \mathbb{R}\widehat{\ }$和 $S \subseteq G$ 包含所有次数至多为一的特征标，我们假设

$$f - \alpha = v_S^* Q v_S,$$

其中 α是 f的最小值， v_S是由 S和 $Q \geq 0$中的特征标组成的 $|S|$ 维列向量。
如果 Q 的零空间是一维的，那么将 Q 的任意零向量的第一个元素归一化为 1，即可得到目标解。得到目标零向量 $v \in \mathbb{R}^{|S|}$后，我们可以通过对 v 的元素进行舍入来恢复 $\tilde{g} \in G$。

如果 Q 的零空间维数为 $d > 1$，那么仅靠归一化就不够了，因为此时 Q 的零向量的元素可能代数不一致。我们可以根据 Stickelberger 定理提取解，该定理在文献中常用于求解多项式系统 [7, 8, 17, 32, 38]。

(2) The moment matrix $H \geq 0$ is the solution of the dual problem of (5). Let $H = \sum_{i=1}^{|S|} \mu_i u_i u_i^*$ be the eigenvalue decomposition of H, with $\mu_1 \geq \mu_2 \geq \mu_3, \ldots \geq \mu_{|S|}$. Then $\widetilde{H} = \mu_1 u_1 u_1^* \in \mathbb{R}^{S\times S}$ is the rank-one approximation of H. We can recover the solution $\widetilde{x}$ from $\widetilde{H}$ by setting $\widetilde{x}_i = \widetilde{H}_{1,x_i}$.

实际上，通过算法 2 得到的矩阵 Q 可能不是半正定的。我们需要将其更新为 $Q + \lambda_{\text{最小值}}(Q)\mathrm{Id}$。我们稍感意外地发现，$Q + \lambda_{\text{最小值}}(Q)$ Id 的数值余秩常常为 1，这使得我们能够通过其零向量高效地恢复最优解。

最后，我们通过几个例子来说明上述舍入过程。

例4.1 （一维零空间）。 在例 3.6 中，我们看到

$$f(z_1,z_2,z_3) = 4 + z_1 + z_2 + z_3 + z_1z_2z_3$$

是定义在 C_2^3 上的非负函数。对于

$$S = \{1, z_1, z_2, z_3, z_1z_2z_3\},$$

我们可以通过*SDPNAL+*得到一个厄米矩阵（但不是半正定矩阵）：

$$Q = \begin{array}{c} \\ 1 \\ z_1 \\ z_2 \\ z_3 \\ z_1z_2z_3 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & z_1 & z_2 & z_3 & z_1z_2z_3 \end{array} \\ \begin{bmatrix} 1.9546 & 0.5000 & 0.5000 & 0.5000 & 0.5000 \\ 0.5000 & 0.4536 & 0.0000 & 0.0000 & 0.0000 \\ 0.5000 & 0.0000 & 0.4536 & 0.0000 & 0.0000 \\ 0.5000 & 0.0000 & 0.0000 & 0.4536 & 0.0000 \\ 0.5000 & 0.0000 & 0.0000 & 0.0000 & 0.4536 \end{bmatrix} \end{array},$$

其特征值为

$$-0.0462, 0.4536, 0.4536, 0.4536, 2.4544.$$

对应于 -0.0462 的归一化特征向量为

$$v = \begin{array}{c} \begin{array}{ccccc} 1 & z_1 & z_2 & z_3 & z_1z_2z_3 \end{array} \\ \begin{bmatrix} 1 & -1.000447 & -1.000447 & -1.000447 & -1.000447 \end{bmatrix} \end{array}.$$

我们通过对 v 中的元素进行舍入来恢复 $z_1 = z_2 = z_3 = -1$； v 是 $f = 0$ 的最优解。

例 4.2 (更高维的零空间)。 我们考虑函数 $f(z_1, z_2) = (1+z_1+z_2+z_1z_2)^2$，定义在 C_2^2 上，取值于 $S = \{1, z_1, z_2, z_1z_2\}$。我们注意到 f 有一个格拉姆矩阵

$$Q = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix},$$

Q 的零空间由以下向量张成

$$\begin{bmatrix} 1 & 0 & 0 & -1 \end{bmatrix}^T, \quad \begin{bmatrix} 1 & 0 & -1 & 0 \end{bmatrix}^T, \quad \begin{bmatrix} 1 & -1 & 0 & 0 \end{bmatrix}^T.$$

利用这些零向量，我们构造乘法矩阵[7] 关于 $z_1 - z_2/2$:

$$\begin{bmatrix} 0 & 1 & -1/2 \\ 3/2 & 1/2 & 1/2 \\ -3/2 & -1 & -1 \end{bmatrix}$$

其特征向量为

$$\begin{bmatrix} -1 & -1 & 1 \end{bmatrix}^T, \quad \begin{bmatrix} -1 & 1 & 1 \end{bmatrix}^T, \quad \begin{bmatrix} 1 & -1 & 1 \end{bmatrix}^T.$$

将这些特征向量的第一个元素归一化为 1 后，我们得到 $f(z_1, z_2) = 0$的三个解：$(1, -1)$、$(-1, -1)$ 和 $(-1, 1)$。

A 数值实验

本节进行三次数值实验，以检验算法 2 以及第 4.2 节所讨论的舍入技术。我们使用 Matlab (2016b) 和 SDPNAL+ [44]实现该算法。代码可在 github.com/jty-AMSS/Fast-Lower-Bound- On-FAG 获取。所有实验均在一台配备英特尔酷睿 i9-10900X@3.70GHz 中央处理器和 128GB 随机存取存储器的台式电脑上进行。

A.1 随机函数的下界

第一个实验旨在展示算法 2 的正确性和效率。我们在两个有限阿贝尔群上随机生成非负整数值函数，并分别使用算法 2、TSSOS[49] 和 CS-TSSOS [50] 计算它们的下界。不失一般性，我们只考虑最小值为零的函数。

为便于比较，我们在表 1 和 2 中记录了每种算法的运行时间和结果。该实验表明，对于随机实例，算法2比通用方法 TSSOS 和 CS-TSSOS 更高效。

A.1.1 实验 C_2^{25}。 我们生成十个次数为三且稀疏度至多为 500 的多项式 C_2^{25}，方法是从集合 $\{m \in \mathbb{Z} : -5 \le m \le 5\}$ 中随机选取其非常数系数。这些多项式的常数项经过选取，使其最小值为零。

对于这十个多项式中的每一个 f，我们使用以下参数执行算法 2:

$$d = 2, \quad k = 3\,|\mathrm{supp}(f)|, \quad l = 0,$$
$$m = \text{sum of absolute value of coefficients of } f.$$

我们使用 TSSOS 和 CS-TSSOS，并结合 Mosek [1] 在松弛阶数 2 下计算这些函数的下界。结果见表 1，其中"sp"表示稀疏度，"bound"表示相应算法得到的下界。

[7]任何随机线性组合 z_1 和 z_2 也同样适用。

No	sp	算法 2		TSSOS		CS-TSSOS	
		界	time	界	time	界	time
1	451	-6.6e-01	903	-2.75	**407**	9.0e-09	821
2	440	-1.1e-01	**587**	-3.95	1208	-3.44	1302
3	453	-4.9e-02	**519**	6.3e-08	1104	4.5e-09	1634
4	453	-1.6e-06	1464	-2.5	**1091**	-1.92	1510
5	451	-4.4e-01	804	-4.06	**636**	2.2e-06	1535
6	457	-6.7e-01	**581**	9.0e-11	1081	2.3e-10	2031
7	452	-2.2e-02	**636**	4.0e-10	847	7.0e-09	1895
8	455	-6.5e-03	**775**	4.8e-09	797	7.0e-11	1126
9	443	-1.3e-01	**554**	4.9e-10	716	2.4e-11	1284
10	454	-1.9e-02	**561**	2.0e-08	807	3.0e-09	1352

表 1： C_2^{25}**上的随机示例**

我们注意到，在表 1中，算法2 成功地恢复了这十个函数的最小值零，因为我们的函数取整数值。相比之下，TSSOS 和 CS-TSSOS 分别在四个和两个实例中未能恢复最小值。此外，算法 2 在十个函数中的七个上比 TSSOS 更快，在九个上比 CS-TSSOS 更快。

A.1.2 实验： C_3^{15}。我们通过以下步骤在 C_3^{15}上生成十个函数：

(1) 设置 $f = 0$；(2) 随机生成一个整数值函数 h，定义在 C_3^2 上，使得 $0 \le h \le 10$；(3) 随机选取一个投影映射 τ：$C_3^{15} \to C_3^2$；(4) 更新 $f \leftarrow f + h \circ \tau$；(5) 重复步骤（2）-（4），直到 f的稀疏度大于190。（6）更新 $f \leftarrow f -$ 最小值$_{g \in C_3^{15}} f(g)$。

通过上述过程得到的函数均为整数值，且其最小值为零。尽管算法 2 能够处理定义在 C_3^{15}上、稀疏度最高达 350的函数，但我们将稀疏度上限设为 190，因为 CS-TSSOS 只能处理稀疏度约为 200的函数。

我们使用带有2 的算法计算它们的下界。

$$d = 2, \quad k = 3\,|\mathrm{supp}(f)|, \quad l = 0$$
$$m = \text{sum of maximum value of } h \text{ in step (2)}.$$

我们还使用 Mosek 和松弛阶数[8] 4 对这些函数应用 CS-TSSOS。

本实验中我们未测试 TSSOS，因为它仅适用于具有实变量和实系数的函数，而 $C_3 = \{1, \omega, \omega^2\}$ 其中 $\omega = \exp(2\pi i/3) \in \mathbb{C}$。结果列于表2中，其中"sp"表示每个函数的稀疏度，"bound"表示相应算法计算出的下界。

如表 2所示，算法 2和 CS-TSSOS 均成功恢复了所有实例的最小值，但算法 2 比 CS-TSSOS 快得多。

我们注意到，表 2 中的十个函数具有相同的稀疏度 191。稍加思考便会发现，这是我们生成这些函数的过程所导致的。实际上，过程中的函数 $h \circ \tau$包含 8 个单项式 1，z_i，z_j，z_i^2，z_j^2，z_iz_j，$z_i^2z_j$，$z_iz_j^2$

[8]如果使用较小的松弛阶数，CS-TSSOS 将无法完成计算。

No	sp	算法 2		CS-TSSOS	
		界	time	界	time
1	191	-1.20e-01	**237.0**	-1.98e-05	3148.1
2	191	-3.42e-12	**224.8**	2.38e-08	1923.3
3	191	-3.53e-03	**223.6**	2.76e-09	444.5
4	191	-1.74e-01	**887.1**	-8.55e-06	6623.8
5	191	-3.31e-12	**876.6**	1.17e-07	1745.5
6	191	-2.40e-12	**221.7**	5.34e-07	752.4
7	191	-3.82e-12	**225.3**	3.92e-08	362.9
8	191	-6.92e-02	**402.5**	8.33e-09	2144.8
9	191	-1.53e-01	**227.0**	4.21e-07	1695.6
10	191	-5.71e-12	**404.3**	4.61e-10	1105.3

表 2：C_3^{15}上的随机示例

对于某个 $1 \le i < j \le 15$。一旦 f在某次迭代中包含全部 15 个变量，它就必须包含全部 31 个单项式 z_j^k，$1 \le j \le 15$，$0 \le k \le 2$。因此，每次更新 $f + h \circ \tau$只会使稀疏度增加 0 或 4，所得函数的稀疏度形式为 $31 + 4k$，$k \in \mathbb{N}$。由于稀疏度超过 190 时我们便会终止该过程，因此预期稀疏度为 $191 = 31 + 4 \times 40$。

A.2 最大可满足性问题的上界

作为一项应用，我们在最大可满足性问题上测试算法 2 。首先回顾命题 2.3-(v)：对于给定的合取范式公式 ϕ，其含有 n 个变量和 m 个子句，则 ϕ中可同时满足的子句的最大数量等于 $m -$ 最小值$_{g \in C_2^n} f_\phi(g)$。因此，求解 ϕ的最大可满足性问题，等价于计算其特征函数 f_ϕ在 C_2^n上的最小值。

我们测试的最大可满足性问题取自 2016 年和 2009 年 MAX-SAT 竞赛的基准问题集[9] [10]。在适用情况下，我们还使用 TSSOS 和 CS-TSSOS 求解这些问题。结果汇总于表 3 和 4 中。从前者可以得出，算法 2 在最大可满足性问题上比 TSSOS 和 CS-TSSOS 更有效；而从后者可以看出算法 2的灵活性。

A.2.1 无权*MAX-2SAT*问题实验。 在本次实验中，我们考虑了 2016 年 MAX-SAT 竞赛中随机生成的无权 MAX-2SAT 问题。这类问题有 120 个变量，子句数范围为 1200 至 2600。我们应用算法2 计算相应特征函数在 C_2^{120}上的下界，参数如下：

$$d = 1, \quad k = |\mathrm{supp}(f)|, \quad l = 0,$$
$$m = \text{number of clauses}.$$

我们使用 Mosek 对这些函数应用松弛阶数为 1 的 TSSOS 和 CS-TSSOS。值得指出的是，任何更高阶的松弛都会导致内存泄漏。这是因为更高阶松弛中的格拉姆矩阵大小呈指数级增长。[11] 例如，第一次松弛中的格拉姆矩阵大小为 121×121 而第二次松弛中的格拉姆矩阵大小为 7261×7261 。

[9] http://www.maxsat.udl.cat/09/index.php [10] http://www.maxsat.udl.cat/16/index.html [11] 共有 $\sum_{j=0}^{k} \binom{120}{k}$ 个单项式参与了第 k阶松弛。

数值结果见表 3，其中"子句"表示每个合取范式公式中的子句数，"最小值"表示特征函数的最小值，"下界"表示各方法得到的特征函数下界。

从表3中可以看出，算法 2得到的下界非常接近特征函数的最小值，且优于 TSSOS 和 CS-TSSOS 得到的结果。从理论上讲，提高 TSSOS 和 CS-TSSOS 中松弛的阶数有望改善下界的质量。遗憾的是，正如我们所指出的，由于所考虑的 MAX-2SAT 问题规模巨大，导致内存泄漏，因此无法采用更高阶的松弛。

A.2.2 加权 *MAX-3SAT* 问题的实验。我们在 2009 年 MAX-SAT 竞赛中的加权 MAX-3SAT 基准问题上测试算法2 。这些 3-CNF公式包含70 个变量和400 个子句。值得注意的是，TSSOS 和 CS-TSSOS 均无法处理这些问题，因为它们的特征函数次数为3，因此松弛阶数至少需要为2。这又会导致内存泄漏。

我们将算法2 应用于这些 MAX-3SAT 问题的特征函数，并使用以下两组参数：

$$d = 1, \quad k = |\mathrm{supp}(f)|, \quad l = 0, \quad m = \text{sum of weights}.$$
$$d = 2, \quad k = 1.5\,|\mathrm{supp}(f)|, \quad l = 0, \quad m = \text{sum of weights}.$$

本实验旨在展示算法2的灵活性。也就是说，可以通过参数 d和 k自由控制下界的质量与时间成本之间的权衡。作为对比，在 TSSOS 和 CS-TSSOS 等其他基于 SOS 的算法中，这种权衡由松弛阶数控制，而这很容易导致内存泄漏（参见A.2.1）。实验结果见表4，其中各标签的含义与表3相同。

A.3 最大 2-SAT 问题的舍入

令 S_k 为算法 1 在输入 $l = 0$、$d = 2$、 m 时选出的支撑集，其中 $l = 0$、$d = 2$、 m 分别为子句数，并选择 k 使得 $S = S_k \cup M_1$ 的基数为 $|M_p|$。其中， M_1 是次数至多为 1 的单项式集合，M_p 是包含 M_1 以及单项式$z_i z_j$的单项式基；当逻辑变量 x_i和 x_j出现在同一子句中时，单项式$z_i z_j$也包含在其中，参见 [46, 定义 1]。我们在随机生成的最大 2-SAT 问题上，将第 4.2 节介绍的舍入技术与 [46] 中采用缩放因子 ρ_i^N和 $2^{-(i-1)}$ 的舍入技术进行比较。对于基于格拉姆矩阵的舍入方法，SDPNAL+的最大迭代次数设为 $|S|$。对于基于矩矩阵的舍入方法，SDPNAL+ 的最大迭代次数设为 $\max(300, |S|)$。

A.3.1 随机问题实验。 在本实验中，我们在 [46, 表 7]中进行实验。对于 $n = 25$、30、35、40 和$m = 3n$， $5n$， $7n$，我们随机生成 100 个最大 2-SAT 实例，其中包含 n 个变量和 m 个子句，将我们的舍入技术与 [46], 中给出的方法进行比较，并记录每种方法找到最优解的频率。结果见表 5，其中

No	子句	min	算法 2		TSSOS		CS-TSSOS	
			时间界		时间界		时间界	
1	1200	161	**159.5**	370	146.7	45	146.7	52
2	1200	159	**156.7**	327	143.1	49	143.1	55
3	1200	160	**159.0**	362	146.8	46	146.8	64
4	1300	180	**177.5**	450	162.4	52	162.4	73
5	1300	172	**170.6**	417	156.2	47	156.2	65
6	1300	173	**171.6**	432	158.8	44	158.8	58
7	1400	197	**194.8**	506	179.8	46	179.8	75
8	1400	191	**189.3**	499	174.3	51	174.3	87
9	1400	189	**187.2**	504	172.1	58	172.1	78
10	1500	211	**209.9**	589	194.5	53	194.5	93
11	1500	213	**210.1**	573	194.4	50	194.4	88
12	1500	207	**205.7**	531	191.3	50	191.3	87
13	1600	233	**231.2**	668	215.6	50	215.6	90
14	1600	239	**235.0**	668	218.7	48	218.7	85
15	1600	233	**230.5**	617	215.2	52	215.2	68
16	1700	257	**255.1**	745	238.3	48	238.3	91
17	1700	248	**245.7**	749	229.2	53	229.2	105
18	1700	239	**238.9**	738	225.3	50	225.3	86
19	1800	291	**285.8**	788	268.4	56	268.4	87
20	1800	262	**261.3**	896	244.2	53	244.2	99
21	1800	279	**277.5**	836	259.8	50	259.8	99
22	1900	293	**292.1**	1002	275.3	67	275.3	119
23	1900	296	**294.3**	987	275.8	53	275.8	132
24	1900	294	**291.8**	994	273.0	53	273.0	113
25	2000	307	**306.4**	1103	288.8	57	288.8	149
26	2000	321	**318.1**	1131	299.8	55	299.8	140
27	2000	307	**306.0**	1240	288.3	58	288.3	139
28	2100	336	**335.5**	1202	317.7	68	317.7	141
29	2100	336	**332.2**	1215	313.4	57	313.4	141
30	2100	332	**330.3**	1224	311.3	57	311.3	104
31	2200	358	**355.6**	1467	334.6	54	334.6	163
32	2200	371	**365.8**	1351	345.2	52	345.2	114
33	2200	359	**358.1**	1381	338.3	59	338.3	130
34	2300	380	**377.6**	1412	356.8	58	356.8	127
35	2300	383	**381.5**	1535	359.2	61	359.2	136
36	2300	365	**364.7**	1510	345.4	53	345.4	146
37	2400	389	**387.9**	1686	368.7	56	368.7	151
38	2400	402	**400.0**	1757	378.1	56	378.1	125
39	2400	380	**380.0**	1951	363.0	69	363.0	153
40	2500	418	**416.4**	1963	395.5	60	395.5	133
41	2500	435	**432.4**	1876	411.5	55	411.5	143
42	2500	425	**424.2**	1934	402.4	68	402.4	186
43	2600	439	**436.7**	2004	414.0	57	414.0	148
44	2600	458	**455.8**	2010	434.2	57	434.2	221
45	2600	440	**439.1**	1983	418.7	59	418.7	154

表3：无权MAX-2SAT问题

No	min	$d=1, k=\lvert$支持$(f)\rvert$		$d=2, k=3\lvert$支持$(f)\rvert/2$	
		时间界		时间界	
1	12	6.2853	615	8.8433	2265
2	19	10.7006	582	13.6655	2126
3	5	4.2047	601	4.8462	2176
4	20	12.1123	584	15.1769	2098
5	14	7.4614	605	9.7210	2183
6	12	8.0066	595	10.0030	2131
7	17	10.1713	605	12.7487	2157
8	7	5.4025	635	6.4714	2325
9	17	10.6177	597	13.6209	2121
10	12	8.2999	573	10.6198	2108

表4：加权 MAX-3SAT 问题

“Vars”表示变量的数量，“clause”表示子句的数量，“Gram”和“Moment”分别表示基于格拉姆矩阵和矩矩阵的方法找到最优解的频率，ρ_i^N以及 2^{-}（$i-1$）表示 [46]中所介绍的推荐方法的频率。表 5 显示，基于低秩矩近似的舍入方法找到最优解的频率最高。

Vars	子句	Gram	矩	ρ_i^N	$2^{-}(i-1)$
25	75	63	**96**	70	76
25	125	75	**93**	67	81
25	175	80	**91**	64	87
30	90	71	**95**	67	68
30	150	70	**91**	58	70
30	210	68	**87**	63	73
35	105	74	**93**	56	65
35	175	77	**91**	55	78
35	245	68	**87**	52	72
40	120	79	**92**	51	56
40	200	72	**87**	44	59
40	280	67	**83**	54	76

表5：随机 MAX-2SAT 问题的舍入

A.3.2 基准问题实验。我们分别测试了第4.2节和 [46] 中介绍的舍入技术，测试对象为2016年 MAX-SAT 竞赛中的随机生成的无权MAX-2SAT问题。表6 表明，我们的两种舍入方法均优于 [46]中介绍的方法。

此外，我们指出，由于 [46] 使用内点法求解半正定规划问题，其舍入技术通常至少需要6000秒。相比之下，我们的方法耗时不到1000秒，因为我们的方法允许使用 SDPNAL+ 求解半正定规划问题。

参考文献

[1] MOSEK ApS。2019年。*MOSEK MATLAB*优化工具箱手册。版本 *9.3.11*。https://docs.mosek.com/9.3/toolbox/index.html[2] Sanjeev Arora 和 Boaz Barak。2009年。计算复杂性：一种现代方法。剑桥大学出版社。

No	子句	min	Gram	矩	ρ_i^N	$2^-(i-1)$
1	1200	161	**162**	**162**	225	227
2	1200	159	**159**	164	215	194
3	1200	160	**160**	**160**	162	**160**
4	1300	180	**180**	185	226	243
5	1300	172	**173**	178	225	230
6	1300	173	**173**	174	245	253
7	1400	197	**198**	202	234	270
8	1400	191	**192**	199	255	246
9	1400	189	**189**	**189**	227	231

表6：最大 2-SAT 基准上的舍入

[3] Per Austrin。2007年。平衡最大 2-SAT 未必是最难的问题。载于第三十九届 ACM 计算理论年度研讨会论文集。189–197。 [4] Per Austrin。2010年。迈向任意 2-CSP 的精确不可近似性结果。《SIAM计算杂志》39, 6 (2010年), 2430–2463。 [5] Nicos Christofides。1976 年。旅行商问题新启发式算法的最坏情况分析。技术报告。卡内基梅隆大学匹兹堡分校管理科学研究组。 [6] Stephen A Cook。1971年。定理证明过程的复杂性。载于第三届 ACM 计算理论年度研讨会论文集。151–158。[7] R. Corless、P. Gianni 和 B. Trager。一种用于具有重根的零维多项式系统的重排舒尔分解方法。载于 1997 年 ISSAC'97 论文集。133–140。[8] R. Corless、P. Gianni、B. Trager 和 S. Watt。多项式系统的奇异值分解。载于 1995 年 ISSAC'95 论文集。96–103。 [9] George B Dantzig 和 John H Ramser。1959年。卡车调度问题。《管理科学》6, 1 (1959年), 80–91。 [10] Martin Davis、Ron Sigal 和 Elaine J Weyuker。1994年。可计算性、复杂性与语言：理论计算机科学基础。Elsevier。 [11] Irit Dinur 和 David Steurer。2014年。并行重复的分析方法。载于 STOC'14——2014年 ACM 计算理论研讨会论文集。ACM，纽约，624–633。 [12] Hamza Fawzi、James Saunderson 和 Pablo A Parrilo。2016年。有限阿贝尔群上的稀疏平方和与改进的半正定提升。《数学规划》160, 1-2 (2016年), 149–191。 [13] U. Feige 和 M. Goemans。1995年。近似二次幂证明系统的值及其在最大 2-SAT 和最大有向割中的应用。载于第三届以色列计算与系统理论研讨会论文集。182–189。 [14] William Fulton 和 Joe Harris。2013年。表示论：入门教程。第 129 卷。Springer Science & Business Media。 [15] Michel X Goemans 和 David P Williamson。1995年。使用半定规划改进最大割和可满足性问题的近似算法。《ACM杂志》(JACM) 42, 6 (1995年), 1115–1145。 [16] Michel X. Goemans 和 David P. Williamson。1995年。使用半定规划改进最大割和可满足性问题的近似算法。《计算机协会杂志》42, 6 (1995年), 1115–1145。 [17] Didier Henrion 和 Jean-Bernard Lasserre。2005年。在 GloptiPoly 中检测全局最优性并提取解。Springer Berlin Heidelberg，柏林，海德堡，293–310。[18] Wenxuan Huang、Daniil A. Kitchaev、Stephen T. Dacek、Ziqin Rong、Alexander Urban、Shan Cao、Chuan Luo 和 Gerbrand Ceder。2016年。通过凸优化和最大可满足性问题求得并证明广义伊辛模型的精确基态。《物理评论B》94（2016年10月），134424。第 13 期。 [19] Anna R. Karlin 、 Nathan Klein 和 Shayan Oveis Gharan。 [2021] ©2021。一种（略有改进的）度量旅行商问题近似算法。载于 STOC'21——第 53 届 ACM SIGACT 计算理论研讨会论文集。ACM，纽约，32–45。 [20] H. Karloff 和 U. Zwick。1997年。一种用于最大 3-SAT 的 7/8 近似算法？载于第 38 届计算机科学基础年度研讨会论文集。406–415。 [21] Richard M Karp。1972年。组合问题之间的可约性。载于《计算机计算复杂性》。Springer，85–103。 [22] Subhash Khot、Guy Kindler、Elchanan Mossel 和 Ryan O'Donnell。2007年。最大割及其他二变量约束满足问题的最优不可近似性结果？《SIAM计算杂志》37, 1 (2007年), 319–357。 [23] Bernhard H Korte、Jens Vygen、B Korte 和 J Vygen。2011年。组合优化。第 1 卷。Springer。 [24] Mark W Krentel。1988年。优化问题的复杂性。《计算机与系统科学杂志》36, 3 (1988年), 490–509。

[25] Adam Kurpisz、Samuli Leppänen 和 Monaldo Mastrolilli。2016年。二元多项式优化问题的紧平方和下界。载于第43届自动机、语言与编程国际研讨会（ICALP 2016）（莱布尼茨国际信息学会议论文集（LIPIcs）），Ioannis Chatzigiannakis、Michael Mitzenmacher、Yuval Rabani 和 Davide Sangiorgi（编），第55卷。Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik，德国 Dagstuhl，78:1–78:14。[26] Jean B. Lasserre。2001年。多项式全局优化与矩问题。SIAM 优化期刊 11，3（2001），796–817。[27] Jean B Lasserre。2016年。0/1规划的最大割公式。运筹学快报 44，2（2016），158–164。[28] Monique Laurent。2003年。0–1规划中 Sherali-Adams、Lovász-Schrijver 和 Lasserre 松弛的比较。《数学规划》28，3（2003），470–496。[29] Michael Lewin、Dror Livnat 和 Uri Zwick。2002年。最大2-SAT和最大有向割问题的改进舍入技术。载于《整数规划与组合优化》，William J. Cook 和 Andreas S. Schulz （编）。Springer Berlin Heidelberg，柏林，海德堡，67–82。[30] Silvano Martello 和 Paolo Toth。1990年。背包问题：算法与计算机实现。John Wiley & Sons, Inc.。[31] George B Mathews。1896年。论整数分拆。伦敦数学会会刊 1，1（1896），486–490。[32] H. Möller 和 H. Stetter。1995年。通过矩阵特征值问题求解具有重根的多元多项式方程。数值数学 70（1995），311–329。[33] Ryan O'Donnell。2014年。布尔函数分析。剑桥大学出版社，纽约。xx+423 页。[34] Christos H. Papadimitriou。1977年。欧几里得旅行商问题是*NP*完全的。理论计算机科学 4，3（1977），237–244。[35] Christos H. Papadimitriou。1994年。计算复杂性。Addison-Wesley Publishing Company，马萨诸塞州雷丁。xvi+523 页。[36] Matthieu Py、Mohamed Sami Cherif 和 Djamal Habet。2021年。最大可满足性问题的证明构造器。载于可满足性测试理论与应用国际会议。Springer，488–498。[37] Prasad Raghavendra。2008年。适用于每个约束满足问题的最优算法和不可近似性结果？ [扩展摘要]。载于 STOC'08。ACM，纽约，245–254。[38] Greg Reid 和 Lihong Zhi。2009年。通过符号-数值归约至几何对合形式求解多项式系统。符号计算杂志 44，3（2009），280–291。纪念 Daniel Lazard 的多项式系统求解专辑。[39] Julia Robinson。1949年。论哈密顿游戏（一个旅行商问题）。技术报告。兰德空军项目，弗吉尼亚州阿灵顿。[40] Walter Rudin。1962年。群上的傅里叶分析。第121967卷。Wiley在线图书馆。[41] Shinsaku Sakaue、Akiko Takeda、Sunyoung Kim 和 Naoki Ito。2017年。针对二元多项式优化问题、采用截断矩矩阵的精确半定规划松弛。SIAM 优化期刊 27，1（2017），565–582。[42] Petr Slavík。1996年。集合覆盖贪心算法的紧确分析。载于第二十八届 ACM计算理论研讨会论文集（宾夕法尼亚州费城，1996年）。ACM，纽约，435–441。[43] Lucas Slot 和 Monique Laurent。2022年。二元多项式优化的平方和层级。《数学规划》（2022），1–40。[44] Defeng Sun、Kim-Chuan Toh、Yancheng Yuan 和 Xin-Yuan Zhao。2020年。SDP-NAL+：带边界约束的半定规划 Matlab 软件（1.0版）。优化方法与软件 35，1（2020），87–115。[45] Paolo Toth 和 Daniele Vigo。2002年。车辆路径问题。SIAM。[46] H. van Maaren、L. van Norden 和 M.J.H. Heule。2008年。基于平方和的最大可满足性问题近似算法。离散应用数学 156，10（2008），1754–1779。[47] Vijay V Vazirani。2001年。近似算法。第1卷。Springer。[48] Hayato Waki、Sunyoung Kim、Masakazu Kojima 和 Masakazu Muramatsu。2006年。具有结构化稀疏性的多项式优化问题的平方和与半定规划松弛。SIAM 优化期刊 17，1（2006），218–242。[49] Jie Wang、Victor Magron 和 Jean-Bernard Lasserre。2021年。TSSOS：利用项稀疏性的矩-平方和层级。SIAM 优化期刊 31，1（2021），30–58。[50] Jie Wang、Victor Magron、J. B. Lasserre 和 Ngoc Hoang Anh Mai。2022年。CS-TSSOS：大规模多项式优化中的相关与项稀疏性。ACM数学软件汇刊 48，4（2022），1–26。[51] Po-Wei Wang 和 J Zico Kolter。2019年。最大2-SAT问题的低秩半定规划。载于 AAAI人工智能会议论文集，第33卷。1641–1649。[52] Jianting Yang、Ke Ye 和 Lihong Zhi。2022年。在有限阿贝尔群上以准线性时间计算稀疏傅里叶平方和。arXiv 预印本 arXiv:2201.03912（2022）。[53] Jianting Yang、Ke Ye 和 Lihong Zhi。2022年。通过傅里叶平方和为最大可满足性问题提供短证书。arXiv 预印本 arXiv:2207.08076（2022）。[54] Richard Y Zhang 和 Javad Lavaei。2021年。通过对偶化团树转换实现具有近线性时间复杂度保证的稀疏半定规划。《数学规划》188，1（2021），351–393。