软件学报 ISSN 1000-9825, CODEN RUXUEW
2025,36(10):4525−4541 [doi: 10.13328/j.cnki.jos.007302] [CSTR: 32375.14.jos.007302]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

正则图上对称双态自旋系统相关的细密度二分定理*

刘 莹[1,2,3]

[1](计算机科学国家重点实验室 (中国科学院 软件研究所), 北京 100190)

[2](基础软件与系统重点实验室 (中国科学院 软件研究所), 北京 100190)

[3](中国科学院大学, 北京 100049)

通信作者: 刘莹, E-mail: liuy@ios.ac.cn

摘 要: 讨论正则图上的对称双态自旋系统的配分函数计算复杂性. 利用计数指数时间假设 (#ETH) 和随机指数时间假设 (rETH), 将该问题类的经典二分定理, 细化到指数型二分定理, 又称细密度二分定理. 换而言之, 证明满足给定易解条件时, 该问题可在多项式时间内求解; 否则, #ETH 成立时, 该问题没有亚指数时间算法. 还针对平面图限制下已有插值方法在构造根号亚指数时间归约时失效的问题, 提出两种解决方案, 并利用这两种方案探讨平面限制下该问题相关的细密度复杂性和二分定理.

关键词: 计算复杂性; 细密度二分定理; 指数时间假设; 配分函数; 自旋系统

中图法分类号: TP301

中文引用格式: 刘莹. 正则图上对称双态自旋系统相关的细密度二分定理. 软件学报, 2025, 36(10): 4525–4541. http://www.jos.org.cn/1000-9825/7302.htm

英文引用格式: Liu Y. Fine-grained Dichotomies for Symmetric 2-spin System on Regular Graphs. Ruan Jian Xue Bao/Journal of Software, 2025, 36(10): 4525–4541 (in Chinese). http://www.jos.org.cn/1000-9825/7302.htm

Fine-grained Dichotomies for Symmetric 2-spin System on Regular Graphs

LIU Ying[1,2,3]

[1](Sate Key Laboratory of Computer Science (Institute of Software, Chinese Academy of Sciences), Beijing 100190, China)

[2](Key Laboratory of System Software (Institute of Software, Chinese Academy of Sciences), Beijing 100190, China)

[3](University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: This study discusses the computational complexity of the partition function of the symmetric dual-spin system on regular graphs. Based on # exponential time hypothesis (#ETH) and random exponential time hypothesis (rETH), this study develops the classical dichotomies of this problem class into the exponential dichotomies, also known as the fine-grained dichotomies. In other words, this study proves that when the given tractable conditions are satisfied, then the problem is solvable in polynomial time; otherwise, there is no sub-exponential time algorithm when #ETH holds. This study also proposes two solutions to solve the in-effectiveness of existing interpolation methods on building sqrt-sub-exponential time reductions under the restriction of planar graphs. It also utilizes these two solutions to discuss the related fine-grained complexity and dichotomy of this problem under the planar graph restriction.

Key words: computational complexity; fine-grained dichotomy; exponential time hypothesis; partition function; spin system

1 研究背景与相关工作

自旋系统是描述微观粒子相互作用的重要框架. 在统计物理中, 可用于描述伊辛模型[1]、玻茨模型[2]等重要框架, 其配分函数的值, 反映着物质的能量、磁矩等重要物理性质; 在理论计算机科学中, 自旋系统可用于描述图

* 基金项目: 科技部重点研发课题 (2023YFA1009500); 国家自然科学基金 (61932002, 62272448)
收稿时间: 2024-06-14; 修改时间: 2024-08-10; 采用时间: 2024-10-02; jos 在线出版时间: 2025-02-26
CNKI 网络首发时间: 2025-02-26

染色数目问题[3]、点独立集数目问题[4]等图论上问题, 其配分函数的值对应着问题的解; 在人工智能和概率论等领域, 自旋系统也被称为马尔可夫随机场[5], 被广泛研究. 自旋系统的配分函数计算问题, 是多领域关心的重点问题, 其求解算法与计算复杂性也得到持续的关注和研究.

自旋系统由点阵构成, 每个阵点代表一个微观粒子, 具有不同的自旋状态. 自旋系统假设只有最相近的两个粒子的自旋状态之间具有相互作用. 若粒子只有自旋向上或自旋向下两种状态, 这样的自旋系统被称为双态自旋系统, 可以抽象成一个图 $\mathcal{S}(V,E)$. V 表示粒子的集合, 每个粒子的状态用一个取值为 0 (自旋向下) 或 1 (自旋向上) 的布尔变量来表示; E 表示图中边的集合, 每对相互作用的粒子之间连一条边, 边上赋有一个二元函数 $g:\{0,1\}^2\to\mathbb{C}$ 以表示粒子间的相互作用. 这个自旋系统的配分函数定义为:

$$Z(\mathcal{S})=\sum_{\sigma:V\to\{0,1\}}\prod_{(u,v)\in E}g(\sigma(u),\sigma(v)),$$

其中, σ 表示所有状态变量的一组布尔赋值; $\sigma(u),\sigma(v)$ 表示在赋值 σ 下点 u,v 对应变量的赋值.

配分函数计算问题属于计数问题类 #P[6]. 一个问题 A 属于 #P 类, 则存在一个非确定性多项式时间图灵机对于输入 x 满足, 接受路径的数量等于 $A(x)$. 一个问题是 #P 难的, 则任意 #P 内的问题可以多项式时间内归约到该问题; 若该问题也在 #P 内, 则它是 #P 完全的. 自旋系统的配分函数计算问题, 除少数情况外, 是#P 难的. 基于计数复杂性领域重要的假设: P ≠ #P, 大多数情况下, 人们相信配分函数的精确求解不能在多项式时间内完成. 因此, 探寻其的近似算法[7−10]或量子算法[11,12]是研究的热点之一. 此外, 配分函数的计算复杂性[13], 也得到长久的关注和研究. 根据定义, 配分函数的计算复杂性与图 G 的结构和函数 g 有关. 在过去几十年, 人们从图结构和相互作用函数两个维度, 对配分函数的计算复杂性进行分析, 给出了一系列经典的二分定理[14−20]: 图结构或相互作用函数满足给定条件时, 能给出配分函数的多项式时间确定性求解算法; 否则能证明问题是 #P 难 (#P 完全) 的.

计算复杂性领域内另一被广泛相信的假设由 Impagliazzo 等人[21,22]提出, 称为指数时间假设 (exponential time hypothesis), 简写为 ETH. 该假设认为布尔可满足性问题 (satisfiability) 没有亚指数时间内的确定性算法. 利用 ETH, 可以将 NP 难的问题的计算复杂性, 进一步细化到没有亚指数时间算法[23]. 该假设被 Dell 等人[24]发展到计数版本#ETH 和随机版本 rETH, 用于证明#P 难问题没有亚指数时间算法, 如图的完美匹配数目问题[25]. 一些经典的二分定理, 也被细化到指数型二分定理, 即问题在满足易解条件时有多项式时间内的算法, 否则#ETH 成立时问题没有亚指数时间算法 (或根号亚指数时间算法), 例如计数约束求解问题的指数型二分定理[26]. 这样的指数型二分定理又称为细密度二分定理.

在自旋系统的配分函数计算问题上, 相比于经典二分定理的大量系统性结果[14−20], 细密度二分定理还处于兴起阶段, 仅有小部分结果: 关于 Tutte 多项式[24,27,28]、计数约束求解问题[26,28]和 Holant*问题[29]的细密度二分定理包含了一些受限自旋系统上配分函数计算的细密度复杂性分类; Chen 等人[30]给出了当相互作用函数为布尔值域的对称函数时, 配分函数计算的细密度二分定理.

本文考虑对称双态自旋系统的配分函数计算的细密度复杂性, 即相互作用函数为对称布尔函数. 根据已有的正则图上该问题的经典二分定理[17,19,20], 发展#ETH 和 rETH 下相关的细密度二分定理. 这是对配分函数计算复杂性的更细致探讨, 有助于更深层理解该问题的计算复杂性; 同时, 在探索过程中出现的新的归约技术和方法, 也对其他问题的细密度复杂性研究提供了工具和引导.

1.1 本文成果与结构

本文第 2 节介绍相关基础知识. 第 3 节介绍文章中使用的已有归约技术: 构件构造、全息变换和插值. 第 4 节为本文中心, 论证正则图结构上对称双态自旋系统相关的细密度复杂性.

第 4.1 节根据经典二分定理的证明, 利用已有归约手段, 证明 k 正则图上该问题的细密度二分定理.

定理 1. 给定一个底层图结构为 k 正则图的双态自旋系统, 其中 k 为大于 2 的整数. 若粒子间的相互作用函数 $g=[a,1,b]$ ($a,b\in\mathbb{C}$) 满足以下某一条件, 则该系统的配分函数可以在多项式时间内计算.

(1) $a=b=0$;

(2) $ab=1$;

(3) $ab=-1$ 且 $a^{2k}=b^{2k}$.

否则, 若#ETH 成立, 则配分函数不能在 $2^{O(N)}$ 时间内计算, 其中 N 表示系统内粒子数 (点数).

第 4.2 节进一步探讨平面 k 正则图上该问题的细密度二分定理. 当限制图结构为平面时, 已有插值技术的失效使得细密度复杂性的分析论证更为困难. 本文退一步考虑, 除了相互作用函数 g, 粒子可能会被额外的一个一元函数 $[0,1]$ 作用时, 平面图上的双态自旋系统. 一个粒子被额外的一元函数 $[0,1]$ 作用时, 例如图 1 所示 (黑点表示粒子, 边表示函数), 该粒子的状态被限制为 1 (自旋向上).

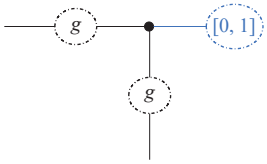


图 1 被两个相互作用函数 g 和一个一元函数 $[0,1]$ 作用的粒子

第 4.2.1 节利用比#ETH 更强的 rETH 假设, 规避了插值的使用, 从而论证了这样的系统配分函数计算问题的细密度二分定理.

定理 2. 给定一个底层图结构为 k 正则图的双态自旋系统, 其中 k 为大于 2 的整数, 该系统中粒子受到相互作用函数 $g=[a,1,b]$ ($a,b\in\mathbb{Q}$) 或独立作用函数 $[0,1]$ 的约束. 当 g 满足以下条件之一时, 该系统的配分函数可以在多项式时间内计算.

(1) $a=b=0$;

(2) $ab=1$;

(3) $ab=-1$ 且 $a^{2k}=b^{2k}$, 即 $(a,b)\in\{(1,-1),(-1,1)\}$.

否则, 若 rETH 成立, 配分函数不能在 $2^{O(\sqrt{N})}$ 时间内计算, 其中 N 表示系统内粒子数 (点数).

第 4.2.2 节给出一个利用小规模构件实现两两线性无关函数的创新性方法, 并将该方法与多项式时间插值结合, 从而论证了平面正则图上受限对称双态自旋系统的配分函数计算的细密度复杂性下界.

定理 3. 给定一个底层图结构为平面 k 正则图的双态自旋系统, 该系统中相互作用函数为 $g=[0,1,b]$, 其中 k 为大于 2 的整数且 b 为非负实数. 若#ETH 成立, 则配分函数不能在 $2^{O(\sqrt{N/(\log N)})}$ 时间内计算, 其中 N 表示系统内粒子数 (点数).

第 5 节总结本文成果并给出几个待解决问题.

2 基础知识

2.1 相关记号与定义

$\mathbb{N}$、$\mathbb{R}$ 和 $\mathbb{C}$ 分别表示自然数集、实数集和复数集. 对于某个正整数 q, $[q]$ 表示一个大小为 q 的有限集合 $\{1,2,\ldots,q\}$. 若 $q=2$, 则 $[q]$ 称为布尔域, 该域中任意元素被赋值为 0 或 1. 一个定义在 $[q]$ 上的复数值函数 F 将 $[q]^k$ 内的项映射到复数, 其中 $k\in\mathbb{N}$ 被称作函数的元. 布尔一元函数 F 通常写作 $[F(0),F(1)]$, 布尔二元函数 H 通常写作矩阵形式 $\begin{pmatrix} H(0,0) & H(0,1) \\ H(1,0) & H(1,1) \end{pmatrix}$. 一个 k 元函数 F 是对称函数, 当且仅当函数值与变量赋值的顺序无关, 即 $F(a_1,a_2,\ldots,a_k)=F(a_{\pi(1)},a_{\pi(2)},\ldots,a_{\pi(k)})$, 其中 $a_1,a_2,\ldots,a_k$ 为 F 的一组变量赋值且 $\pi:[k]\to[k]$ 为一个置换函数. 根据定义, 一个 k 元对称布尔函数 F 的值仅由变量赋值中 1 的个数 (也称汉明权重) 决定, 故 F 可写作 $[f_0,f_1,\ldots,f_k]$, 其中 f_i 表示 F 在汉明权重为 i 的变量赋值下的函数值. 例如, 二元布尔相等函数, 记作 $=_2$, 可写作 $[1,0,1]$.

张量积的一个特例就是 Kronecker 积, 用 $\otimes$ 表示. 假设 a,b,c,d 为正整数, 两个矩阵 $X=X_{\{a\times b\}}$ 和 $Y=Y_{\{c\times d\}}$ 之间的张量积 $X\otimes Y$ 是一个 $ac\times bd$ 矩阵, 且其在第 $(i,j)\in[a]\times[c]$ 行、第 $(k,l)\in[b]\times[d]$ 列的元素为 $X_{i,k}\times Y_{j,l}$. 递归定义

$X^{\otimes k}=X^{\otimes(k-1)}\otimes X$. 一个 k 元函数 F 是退化的, 当且仅当它可以写作一些一元函数的张量积, 即 $F=U_1\otimes U_2\otimes\ldots\otimes U_k$ 对某些一元函数 $U_1,\ldots,U_k$ 成立.

一个无向图 G 可以用元组 (V,E) 表示, 其中 V 表示顶点集, E 表示边集. V,E 也记作 $V(G),E(G)$. 一条边 e 的两个端点若为同一点 v, 则称 e 为点 v 的一条自环边; 若存在 l 条边有相同的两端点 v 和 u, 则将这些边视为 v 和 u 之间的一条 l 重边. 若图 G 中没有自环和重边, 则 G 为简单图, 并常用元组 (u,v) 来指代 G 中两端点为 u 和 v 的边. $N(v)$ 和 $E(v)$ 分别表示点 v 的邻点和邻边集合; $d(v)$ 表示点 v 的度数, 即点 v 的邻边条数, 其中自环计 2 条, l 重边计 l 条. 图 G 的最大度记作 $\Delta=\max\{d(v)|v\in V\}$. 若 G 是二部图, 则 V 可以分割成两个不相交的非空点集 V_L 和 V_R, 且 G 中任一边的一端点在 V_L 中且另一端点在 V_R 中. 二部图也常用元组 $(V_L\cup V_R,E)$ 表示, 其中 V_L 和 V_R 中的点常分别称为“左侧”和“右侧”的点. 若 G 是一个简单图且任意点的度数都为某个正整数 k, 则称 G 为 k 正则图. 若 G 能按照任意两条边不在非端点处相交的方式画在平面上, 则称 G 为平面图, 且符合要求的画法称为 G 的一个平面嵌入.

下面介绍两个典型的图上的#P 完全问题.

点覆盖集数目问题 (#VC): 给定一个图 $G(V,E)$, 一个点子集 $S\in V$ 被称作 G 的一个点覆盖集 (vertex cover) 当且仅当, $\forall e\in E$, e 至少有一端点在 S 内. 点覆盖集数目问题, 简记为#VC, 即为求给定图的点覆盖集数目.

布尔满足性数目问题 (#SAT): 给定一个正整数 k, 一个定义在 n 个布尔变量 $x_1,x_2,\ldots,x_n$ 上的公式 ϕ 被称为 k 合取范式, 记作 k-CNF, 当且仅当 ϕ 可以写作 m 个子句的合取, 即 $\phi=\wedge_{i\in[m]}C_i$, 其中 m 为某个正整数, C_i 表示第 i 个子句且形如 $\vee_{j\in[k]}l_{i,j}$. $l_{i,j}$ 表示第 i 个子句中的第 j 个文字, 且 $l_{i,j}$ 对应某个布尔变量 $x\in\{x_1,x_2,\ldots,x_n\}$ 或者某个变量的否 $\bar{x}=1-x$. 一组布尔赋值 $a=(a_1,\ldots,a_n)$ 满足 $\phi(a)=1$, 则称其为 ϕ 的一组满足性赋值. 给定一个 k-CNF 公式 ϕ, 判定 ϕ 是否有满足性赋值称为布尔满足性问题, 简记为 k-SAT; 求 ϕ 的满足性赋值的组数, 被称为布尔满足性赋数目问题, 简记为#k-SAT. SAT=$\{k\text{-SAT}\,|\,k\in\mathbb{N}\}$ 且#SAT=$\{\#k\text{-SAT}\,|\,k\in\mathbb{N}\}$.

2.2 Holant 问题

在第 1 节中, 本文介绍了双态自旋系统模型. 若这样的一个自旋系统 $\mathcal{S}(V,E)$ 内相互作用函数为 $g=[0,1,1]$, 即要求 $\mathcal{S}$ 内每条边至少有一端点变量赋值为 1. 给定一组点赋值, 若其满足 $\prod_{(u,v)\in E}g(\sigma(u),\sigma(v))=1$, 则此时对应赋值 1 的点构成的集合恰好是 $\mathcal{S}$ 的一个点覆盖集; 反之亦然. 故该自旋系统的配分函数的值恰好是图 $\mathcal{S}$ 的点覆盖集数目. 由此可知, #VC 问题多项式时间等价于相互作用函数为 $[0,1,1]$ 的双态自旋系统的配分函数计算问题. 与#VC 不同的是, #k-SAT 问题的实例并不总能被表达成一个自旋系统.

Holant 是更一般的计数问题模型. Holant 值概念[31]由 Valiant 提出, 并被 Cai 等人[32]发展成 Holant 问题模型. 令 q 是一个大于等于 2 的正整数, $\mathcal{F}$ 是一个定义在 $[q]$ 上的函数组成的集合. $\mathcal{F}$ 上的一个 signature grid 是一个元组 $\Omega(G,\pi)$, 其中 G 是一个点集为 V 且边集为 E 的图; π 是一个将 $\mathcal{F}$ 内某个函数映射到 G 内某个点的函数. 以 $\mathcal{F}$ 为参数函数集的 Holant 问题, 记为 $\text{Holant}_q(\mathcal{F})$, 定义如下.

$\text{Holant}_q(\mathcal{F})$ 问题: 接受 $\mathcal{F}$ 上的一个 signature grid $\Omega(G,\pi)$ 为输入, 并输出

$$\#_q(\Omega)=\sum_{\sigma:E\to[q]}\prod_{v\in V}F_v(\sigma|_{E(v)}),$$

其中, σ 表示对所有边的一组布尔赋值; $\sigma|_{E(v)}$ 表示 σ 下点 v 的邻边 $E(v)$ 的赋值; F_v 是被 π 映射到点 v 的函数, 其值由 $\sigma|_{E(v)}$ 按照一定顺序决定 (该顺序已被 π 确定).

若将 F_v 替换成 λF_v, λ 为某个非 0 常数, 根据定义, 实例的值仅变化 λ 倍; 这种简单的倍数关系是容易计算及相互归约转换的, 故在 Holant 问题中, 对任意函数 F 和非 0 常数 λ, F 和 λF 被视为等价的, 它们之间的相互转化被称为归一化 (normalization).

当 $q=2$ 时, $\mathcal{F}$ 被限定在布尔定义域 $\{0,1\}$, $\text{Holant}_2(\mathcal{F})$ 是一个布尔 Holant 问题, 简记为 $\text{Holant}(\mathcal{F})$; $\#_2(\Omega)$ 也简记为 $\#(\Omega)$. 若 $\mathcal{F}$ 能被划分成不相交的两个函数集合 $\mathcal{H}$ 和 $\mathcal{G}$, 且限制输入的 $\Omega(G,\pi)$ 中 G 为一个二部图 $(V_L\cup V_R,E)$, 其中 V_L 的每个点被 π 映射到 $\mathcal{H}$ 内的某个函数, V_R 的每个点被 π 映射到 $\mathcal{G}$ 内的某个函数. 这样的 Ω 也被称为 $\mathcal{H}|\mathcal{G}$

上的一个 signature grid. 限制下的问题是一个二部图 Holant 问题, 记作 $\text{Holant}_q(\mathcal{H}|\mathcal{G})$.

当上下文语义清晰时, 常将 $\Omega(G,\pi)$ 中 π 的信息隐藏进 G 中, 用 G 指代 Ω. 若函数集 $\mathcal{F}=\{F\}$ 只包含一个函数, 则直接用 F 表示 $\mathcal{F}$. 常用前缀 pl-表示平面限制, 如 pl-Holant($\mathcal{F}$) 表示平面 Holant($\mathcal{F}$) 问题.

根据一个相互作用函数为 g 的双态自旋系统 $\mathcal{S}$, 可以构造一个 signature grid $\Omega(G,\pi)$, 其中 G 为一个二部图 $(V_L\cup V_R,E)$, $V_L=V(\mathcal{S})$ 内每个点表示一个粒子, 被 π 映射到一个布尔相等函数; V_R 内每个点对应 $E(\mathcal{S})$ 内的一条边, 被 π 映射到 g; 若 $u\in V_R$ 对应 $E(\mathcal{S})$ 内边 (v_1,v_2), 则在 E 中加入边 (u,v_1) 和 (u,v_2). 根据定义, $\mathcal{S}$ 的配分函数的值恰好等于 $\#(\Omega)$. 反之亦然, 故该系统的配分函数计算问题, 多项式时间等价于 Holant($\mathcal{EQ}|g$), 其中 $\mathcal{EQ}$ 包含所有的布尔相等函数. 例如#VC $\equiv_{\text{poly}}$Holant($\mathcal{EQ}$|[0, 1, 1]). 若限制自旋系统的底层图结构为 k 正则图, 其中 k 为正整数, 则这样的问题多项式时间等价于 Holant($=_k|g$), 其中 $=_k$ 表示 k 元布尔相等函数.

2.3 指数时间假设

根据著名的 P ≠ NP 假设, SAT 问题被广泛相信不能在多项式时间内求解; 类似的, 根据 P ≠ #P 假设, #SAT 不能在多项式时间内求解. Impagliazzo 等人[21,22]提出了比 P ≠ NP 更强的假设: 指数时间假设 (exponential time hypothesis), 等价定义如下.

指数时间假设 (ETH): 总存在一个大于 0 的常数 ε, 使得 3-SAT 问题没有 $2^{\varepsilon N}$ 时间的确定性算法, 其中 N 表示输入 3-CNF 公式的变量个数.

Dell 等人[24]将 ETH 发展到了计数版本#ETH 和随机版本 rETH, 这两个假设定义如下.

计数指数时间假设 (#ETH): 总存在一个大于 0 的常数 ε, 使得#3-SAT 问题没有 $2^{\varepsilon N}$ 时间的确定性算法, 其中 N 表示输入 3-CNF 公式的变量个数.

随机指数时间假设 (rETH): 总存在一个大于 0 的常数 ε, 使得 3-SAT 问题没有错误率低于 1/3 的 $2^{\varepsilon N}$ 时间内的近似算法, 其中 N 表示输入 3-CNF 公式的变量个数.

利用 Impagliazzo 等人证明的稀疏化引理[22], 以上假设可以加强到没有 $2^{\varepsilon M}$ 的确定性或者近似算法, 其中 M 表示输入 3-CNF 公式的子句个数. 从定义可以看出, rETH 暗示了 ETH, 而 ETH 暗示了#ETH, 反方向未知. 这 3 个假设可按照 rETH, ETH, #ETH 顺序排列, 前一个假设不弱于后一个 (普遍认为前一个强于后一个).

根据#ETH, #3-SAT 的复杂性下界为 $2^{\Omega(N)}$ (或 $2^{\Omega(M)}$). 在以往复杂性下界证明中, 人们常常建立一个已知困难问题到目标问题的多项式时间归约, 来传递#P 难或#P 完全性质; 当传递的复杂性下界从多项式时间变为 $2^{\Omega(N)}$ 时间时, 所建立的归约类型与多项式时间归约不同. 由于本文涉及的问题都具有图形化实例, 因此本小节仅介绍图上问题之间的归约定义.

多项式时间归约 ($\leqslant_{\text{poly}}$): A 和 B 是两个图上的问题. 给定 A 的一个规模为 n 的输入图 G_A, 总存在一个算法 T 满足以下条件, 则称 A 可以多项式时间归约到 B, 记作 $A\leqslant_{\text{poly}}B$.

(1) T 构造了多个 B 的实例 $G_1,G_2,\ldots,G_p$ 去访问 B 的神谕;

(2) 根据 $B(G_1),B(G_2),\ldots,B(G_p)$, T 计算得到 $A(G)$ 且总消耗时间为 $\text{poly}(n)$.

若 $A\leqslant_{\text{poly}}B$ 且 $B\leqslant_{\text{poly}}A$, 则称 A 和 B 多项式时间归约等价, 记作 $A\equiv_{\text{poly}}B$.

亚指数时间归约 ($\leqslant_{\text{serf}}$): A 和 B 是两个图上的问题. 给定 A 的一个规模为 n 的输入图 G_A 和一个时间运行参数 $\varepsilon>0$, 总存在一个算法 T 满足以下条件, 则称 A 可以亚指数时间归约到 B, 记作 $A\leqslant_{\text{serf}}B$.

(1) T 构造了多个 B 的实例 $G_1,G_2,\ldots,G_p$ 去访问 B 的神谕, 且每个新生成实例的规模是 $O(n)$;

(2) 根据 $B(G_1),B(G_2),\ldots,B(G_p)$, T 计算得到 $A(G)$ 且总消耗时间不超过 $2^{\varepsilon n}$.

若 $A\leqslant_{\text{serf}}B$ 且 $B\leqslant_{\text{serf}}A$, 则称 A 和 B 亚指数时间归约等价, 记作 $A\equiv_{\text{serf}}B$.

与多项式时间归约相比, 亚指数时间归约放松了对归约时间的限制, 但要求新生成实例的规模与原实例的规模保持线性关系. 图的规模即为图的表达长度, 一般用图的点数或者边数指代. 在本文中, 如无特殊说明, 图的规模即为图的点数.

引理 1. A 和 B 是两个图上的问题且 $A\leqslant_{\text{serf}}B$. 若存在 $\varepsilon_A>0$ 使得, 对于 n_A 个点的图 G_A, $A(G_A)$ 不能在 $2^{\varepsilon_A n_A}$ 时

间内计算, 那么总存在 $\varepsilon_B>0$ 使得问题 B 没有 $2^{\varepsilon_B n_B}$ 时间的确定性算法, 其中 n_B 表示问题 B 输入图的点数.

证明: 假设对于任意 $\varepsilon_B>0$, 问题 B 都有 $2^{\varepsilon_B n_B}$ 时间内的算法. $A\leqslant_{\text{serf}} B$ 则存在一个带有 B 的神谕的算法 T 在 $2^{\varepsilon n_A}$ 时间内计算 $A(G_A)$, 其中 ε 为任意大于 0 的常数. 将 B 的神谕替换成假设的算法, 则结合 T, 可以得到一个求解 $A(G_A)$ 的确定性算法. 假设 T 所构造的 B 的实例的点数都不超过 cn_A, 其中 c 为某个正常数, 则确定性算法所花费的时间不超过 $2^{\varepsilon n_A}\cdot 2^{\varepsilon_B c n_A}\leqslant 2^{(\varepsilon+c\varepsilon_B)n_A}$. 选取足够小的 ε 和 ε_B, 可以使 $2^{(\varepsilon+c\varepsilon_B)n_A}\leqslant 2^{\varepsilon_A n_A}$, 与引理中已知矛盾. 故存在 $\varepsilon_B>0$ 使得问题 B 没有 $2^{\varepsilon_B n_B}$ 时间的确定性算法.

不难证明, 多项式时间归约和亚指数时间归约都具有传递性. 通过构造一系列的亚指数时间归约, 人们证明了, 当#ETH 或 rETH 成立时, 一些计数问题的亚指数时间下界.

定理 4[26]. g 是一个二元对称布尔函数且 $g\notin\{[0,1,0],[1,0,1],[a,1,b],\lambda[1,\pm i,1],\lambda[1,\pm 1,-1]\mid ab=1,i=\sqrt{-1},\lambda\in\mathbb{C}\}$. 若#ETH 成立, 则存在 $\varepsilon>0$ 使得 Holant($\{=_1,=_2,=_3\}|g$) 没有 $2^{\varepsilon n}$ 时间的确定性算法, 其中 n 表示输入图的点数.

定理 5[29]. 若#ETH 成立, 则存在 $\varepsilon>0$ 使得 Holant($=_3|$ [0, 1, 1]) 没有 $2^{\varepsilon n}$ 时间的确定性算法, 其中 n 表示输入图的点数.

定理 6[29]. 若#ETH 成立, 则存在 $\varepsilon>0$ 使得 pl-Holant($=_3|\{[0,1,1],[1,-1]\}$) 没有 $2^{\varepsilon\sqrt{n}}$ 时间的确定性算法, 其中 n 表示输入图的点数. 若 rETH 成立, 则该结论在限制输入实例的值非 0 即 1 的情况下仍成立.

3 归约方法

3.1 构件构造

一类最直接的归约方法称作构件构造, 即通过函数之间的相互连接实现新的函数. $\mathcal{F}$ 是一个定义集合 $[q]$ 上的函数集. 一个由 $\mathcal{F}$ 内函数构成的构件, 也称作 $\mathcal{F}$ 门, 是一个 $\mathcal{F}$ 上的 signature grid $G(V,E\cup X)$ (π 隐含在 G 内). E 和 X 是两个不相交的边集, E 中的边两端都连接了 V 中顶点; X 中的边一端连接 V 中顶点, 另一端悬空, 也称悬挂边. G 实现了一个 $|X|$ 元函数 Γ_G, 且变量赋值为 $(a_1,a_2,\ldots,a_{|X|})\in[q]^{\otimes|X|}$ 时, 该函数取值为:

$$\Gamma_G(a_1,a_2,\ldots,a_{|X|})=\sum_{\sigma:E\to[q]}\prod_{v\in V}F_v(\hat{\sigma}|_{E(v)}),$$

其中, σ 表示 E 内边的一组赋值; $\hat{\sigma}|_{E(v)}$ 表示 v 的邻边的 (有序) 赋值, 由 σ 和 $(a_1,a_2,\ldots,a_{|X|})$ 共同决定. 若 X 为空, Γ_G 为零元函数, 对应的常值即为 $\#_q(G)$.

令 $\mathcal{H}$ 也是定义在 $[q]$ 上的函数集. 若 G 是一个 $\mathcal{F}\mid\mathcal{H}$ 门, 即 G 是 $\mathcal{F}\mid\mathcal{H}$ 上的 signature grid, 且 X 的邻点全在 V_L 中或 V_R 中. 若 X 的邻点全在 V_L 中, 则称 G 实现了一个“左侧”的函数; 相应的, 若 X 的邻点全在 V_R 中, 则称 G 实现了一个“右侧”的函数. 图 2 展示了一个实现右侧 $=_k$ 函数的 $\{=_2\mid=_3\}$ 门, 其中正整数 $k\geqslant 5$.

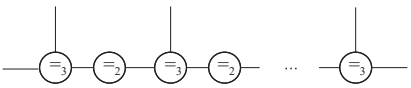


图 2 实现 $=_k$ 函数的 $\{=_2|=_3\}$ 门, 包含 $k-2$ 个 $(=_3)$ 和 $k-3$ 个 $(=_2)$

假设 $\mathcal{H}$ 内每一个函数都可以由某个常数规模的 $\mathcal{F}$ 门实现. 对于 $\text{Holant}_q(\mathcal{H})$ 的任一实例 G, 总可以在 poly($|V(G)|$) 时间内将所有点替换成对应的 $\mathcal{F}$ 门, 从而得到 $\text{Holant}_q(\mathcal{F})$ 的一个实例 G', 有 $\#_q(G)=\#_q(G')$ 且 $|V(G')|=O(|V(G)|)$.

引理 2. $\mathcal{H}$, $\mathcal{F}$ 为两个定义在 $[q]$ 上的函数集, 其中 q 为大于 1 的正整数, 且 $\mathcal{H}$ 内每一个函数都可以由某个常数规模的 $\mathcal{F}$ 门实现, 有:

$$\text{Holant}_q(\mathcal{H})\leqslant_{\text{poly}}\text{Holant}_q(\mathcal{F})\text{且}\text{Holant}_q(\mathcal{H})\leqslant_{\text{serf}}\text{Holant}_q(\mathcal{F}).$$

3.2 全息变换

另一类归约方法被称为全息变换. 有时候, 两个表面看上去不同的计数问题实际上是不同形态下的相同问题, 全息变换就是连接它们之间的等价关系的工具.

一个定义在 $[q]$ 上的 k 元函数 F 总可以写成列向量的形式, 即选取某个变量作为列标, 其余变量作为行标, 按照字母序展开, 相应位置填入对应函数值. 如 3 元布尔相等函数 (对称函数值与变量顺序无关) 可写作:

$$(=_3)=\begin{pmatrix}1&0\\0&0\\0&0\\0&1\end{pmatrix},$$

其中, 行标展开为 00, 01, 10, 11; 列标展开为 0, 1. 对任意矩阵 $T\in\mathbb{C}^{q\times q}$, $T^{\otimes k}F$ 即为 F 被 T 作用后的转换函数. 类似定义 $FT^{\otimes k}$, 其中 F 写作行向量. 对于 $[q]$ 上的函数集 $\mathcal{F}$, $T\mathcal{F}=\{T^{\otimes k}F|F\in\mathcal{F}\}$ 且 $\mathcal{F}T=\{FT^{\otimes k}|F\in\mathcal{F}\}$.

$\mathcal{H}$, $\mathcal{F}$ 为两个定义在 $[q]$ 上的函数集, 一个可逆矩阵 $T\in\mathbb{C}^{q\times q}$ 定义的全息变换为: 对于 $\text{Holant}_q(\mathcal{F}|\mathcal{H})$ 的一个二部图实例 $\Omega(G,\pi)$, 构建一个 $\text{Holant}_q(\mathcal{F}T|T^{-1}\mathcal{H})$ 的一个二部图实例 $\Omega'(G,\pi')$; 若 π 将 $v\in V(G)$ 映射到函数 $F\in\mathcal{F}$ (或 $H\in\mathcal{H}$), 则 π' 将 v 映射到函数 $FT^{\otimes d(v)}$ (或 $(T^{-1})^{\otimes d(v)}H$). Valiant[31]证明了, 对任意可逆矩阵 T, $\#_q(\Omega)=\#_q(\Omega')$, 故容易得到以下结论.

引理 3. $\mathcal{H}$, $\mathcal{F}$ 为两个定义在 $[q]$ 上的函数集, 其中 q 为大于 1 的正整数. 对于任意可逆矩阵 $T\in\mathbb{C}^{q\times q}$, 有:

$$\text{Holant}_q(\mathcal{F}|\mathcal{H})\equiv_{\text{poly}}\text{Holant}_q(\mathcal{F}T|T^{-1}\mathcal{H})\text{且}\text{Holant}_q(\mathcal{F}|\mathcal{H})\equiv_{\text{serf}}\text{Holant}_q(\mathcal{F}T|T^{-1}\mathcal{H}).$$

3.3 插　值

构件构造和全息变化这两个归约方法, 对于输入实例都一对一的构造一个新的等值实例. 插值方法与它们不同, 插值方法生成多个新的实例, 利用这多个实例的值来计算原始输入实例的值. A 和 B 是两个图上的问题, 利用插值方法构造 A 到 B 的归约一般分两步.

(1) 根据 A 的输入实例 G, 构造一个多项式 μ, 满足在某个点 a 上 $\mu(a)=A(G)$.

(2) 根据 G 构造一系列 B 的实例 $G_1,G_2,\ldots$ 满足 $B(G_1)=\mu(b_1),B(G_2)=\mu(b_2),\ldots$, 利用 B 的神谕得到 μ 在多个不同点的值, 根据拉格朗日插值恢复出 μ 内每项的系数, 从而计算得到 $\mu(a)=A(G)$.

以下例子展示了通过插值方法构建多项式时间归约[4,6]和亚指数时间归约[25].

引理 4. $\mathcal{F}$ 为一个布尔函数集, m 为正整数, $\mathcal{H}=\{[1,a_i]|a_i\in\mathbb{C},i\in[m]\}$ 为 m 个布尔一元函数构成的集合. 对于任意正整数 N, 若能构造出一系列 poly(N) 规模的 $\mathcal{F}$ 门以实现 N 个形如 $[1,x]$ 的一元函数, 其中 $x\in\mathbb{C}$, 且这 N 个函数两两线性无关, 则:

$$\text{Holant}(\mathcal{F}\cup\mathcal{H})\leqslant_{\text{poly}}\text{Holant}(\mathcal{F})\text{且}\text{Holant}(\mathcal{F}\cup\mathcal{H})\leqslant_{\text{serf}}\text{Holant}(\mathcal{F}).$$

若涉及的 $\mathcal{F}$ 门都是平面的, 则以上结论在平面限制下仍成立.

证明: (1) 多项式时间插值: 构造多项式时间归约来证明 $\text{Holant}(\mathcal{F}\cup\mathcal{H})\leqslant_{\text{poly}}\text{Holant}(\mathcal{F})$.

令 $G(V,E)$ 为 $\text{Holant}(\mathcal{F}\cup\mathcal{H})$ 的一个实例. V 中每个点都对应 $\mathcal{F}$ 内某个函数或 $\mathcal{H}$ 内某个函数 $h_i=[1,a_i]$, 记对应 h_i 的点构成的集合为 V_i', 不妨设 $|V_i'|=n_i$ 且 $\sum_{i\in[m]}n_i=n\leqslant|V|$. 为 E 的每组布尔赋值都设计一个标签 $t=(t_1,\ldots,t_m)\in\{0,1,\ldots,n_1\}\times\ldots\times\{0,1,\ldots,n_m\}$: 该组赋值下, V_i' 内有 t_i 个点的邻边被赋值为 1. 根据定义:

$$\#(G)=\sum_{\sigma:E\to[q]}\prod_{v\in V-\cup_iV_i'}F_v(\sigma|_{E(v)})\prod_{v\in V_i',i\in[m]}h_i(\sigma|_{E(v)})=\sum_t\rho_t\prod_{i\in[m]}a_i^{t_i},$$

其中, $\rho_t=\sum_{\sigma\text{with label }t}\prod_{v\in V-\cup_iV_i'}F_v(\sigma|_{E(v)})$, 即有相同标签 t 的所有可能赋值下的 $V-\cup_iV_i'$ 中点对应函数值的积. 定义 n 次 m 元多项式:

$$\mu(z_1,z_2,\ldots,z_m)=\sum_t\rho_t\prod_{i\in[m]}z_i^{t_i}.$$

有 $\mu(a_1,a_2,\ldots,a_m)=\#(G)$. 根据假设, 存在 $(n+1)$ 个大小为 poly$(n+1)$ 的 $\mathcal{F}$ 门实现了 $(n+1)$ 个两两线性无关的一元函数 $\{g_j=[1,x_j]|x_j\in\mathbb{C},j\in[n+1]\}$. 定义 $l=(l_1,l_2,\ldots,l_m)\in[n_1+1]\times[n_2+1]\times\ldots\times[n_m+1]$. 任取 l 对 G 作以下操作: $\forall i\in[m]$, 将 V_i' 内点对应函数由 h_i 替换成 g_{l_i} (本质上是替换成实现 g_{l_i} 的 $\mathcal{F}$ 门), 得到新的 $\text{Holant}(\mathcal{F})$ 的实例 G_l. G_l 的规模为 poly($|V|$). 根据定义, 可知:

$$\#(G_l)=\sum_t \rho_t \prod_{i\in[m]} x_{l_i}^{t_i}=\mu(x_{l_1},x_{l_2},\ldots,x_{l_m}).$$

由于 $\{g_j\}$ 两两线性无关, 故通过上述方式构造 $\prod_{i\in[m]}(n_i+1)\leqslant n^m$ 个实例去询问 Holant($\mathcal{F}$) 的神谕, 可以得到 μ 在 $\prod_{i\in[m]}(n_i+1)$ 个不同点上的值. 根据拉格朗日插值法, 已知 μ 在 $\prod_{i\in[m]}(n_i+1)$ 个不同点上的值, 可以在 $\mathrm{poly}(\prod_{i\in[m]}(n_i+1))$ 时间内计算出 μ 中所有未知系数 ρ_t. 从而可以计算得 $\mu(a_1,a_2,\ldots,a_m)=\#(G)$.

接下来分析以上归约所需时间. 每构造一个 G_l 需要 $\mathrm{poly}(|V|)$ 时间, 访问一次 Holant($\mathcal{F}$) 的神谕所花费的时间为 $\mathrm{O}(1)$; 计算出所有 ρ_t 需要 $\mathrm{poly}(\prod_{i\in[m]}(n_i+1))$ 时间; 计算 $\mu(a_1,a_2,\ldots,a_m)$ 需要 $\mathrm{poly}(\prod_{i\in[m]}(n_i+1))$ 时间; 故总时间为 $\mathrm{poly}(|V|)$, 即以上归约可在 $\mathrm{poly}(|V|)$ 时间内完成.

(2) 块插值: 构造亚指数时间归约来证明 Holant($\mathcal{F}\cup\mathcal{H}$)$\leqslant_{\mathrm{serf}}$Holant($\mathcal{F}$). 由于亚指数时间归约具有传递性, 因此在这里简单展示 $\mathcal{H}$ 内只有一个一元函数 $[1,a]$ 时的归约细节.

$G(V,E)$ 为 Holant($\mathcal{F}\cup\{[1,a]\}$) 的一个实例, 且其中对应函数 $[1,a]$ 的点集记作 V'. 设 $V'=n\leqslant|V|$.

任给一个正整数 d, 将 V' 分成 $r=\lceil n/d\rceil$ 个不相交集合 $B_1,B_2,\ldots,B_r$, 满足每个集合大小不超过 d. 这些集合被称为“块”. 不妨设 n 被 d 整除, 即每块大小恰好为 d. 为 E 的每组布尔赋值都新设计一个标签 $\vec{t}=(t_1,t_2,\ldots,t_r)\in\{0,1,\ldots,d\}^r$: 该组赋值下, $\forall i\in[r]$, B_i 内恰好有 t_i 个点的邻边被赋值为 1. 根据定义:

$$\#(G)=\sum_{\vec{t}\in\{0,1,\ldots,d\}^r}\rho_{\vec{t}}\prod_{i\in[r]}a^{t_i},$$

其中, $\rho_{\vec{t}}$ 仍然为标签 $\vec{t}$ 的所有可能赋值下的 $V-V'$ 中点对应函数值的积. 定义 n 次 r 元多项式:

$$\mu(z_1,z_2,\ldots,z_r)=\sum_{\vec{t}\in\{0,1,\ldots,d\}^r}\rho_{\vec{t}}\prod_{i\in[r]}z_i^{t_i}.$$

有 $\mu(a,a,\ldots,a)=\#(G)$. 根据假设, 利用 $\mathcal{F}$ 门可以实现 $(d+1)$ 个两两线性无关的一元函数, 记为 $\{g_j=[1,x_j]\mid x_j\in\mathbb{C},j\in[d+1]\}$. 由于 d 为常数, 则这些 $\mathcal{F}$ 门的规模为常数. 任取 $\vec{l}=(l_1,l_2,\ldots,l_r)\in[d+1]^r$, 对于任意 $i\in[r]$, 将 B_i 内点对应函数替换成 g_{l_i}, 得到新的 Holant($\mathcal{F}$) 的实例 $G_{\vec{l}}$. 根据定义:

$$\#(G_{\vec{l}})=\sum_{\vec{t}\in\{0,1,\ldots,d\}^r}\rho_{\vec{t}}\prod_{i\in[r]}(x_{l_i})^{t_i}=\mu(x_{l_1},x_{l_2},\ldots,x_{l_r}).$$

由于 $\{g_j\}$ 两两线性无关, 故取遍 $\vec{l}\in[d+1]^r$ 并按照上述方式构造 $(d+1)^r$ 个新的实例去询问 Holant($\mathcal{F}$) 的神谕, 能得到 μ 在 $(d+1)^r$ 个不同点上的值, 恢复出所有的系数 $\rho_{\vec{t}}$, 从而计算 $\#(G)$.

以上归约构造 $(d+1)^r$ 个新的实例, 每个实例构造需要 $\mathrm{poly}(|V|)$ 时间. 由于涉及的 $\mathcal{F}$ 门的规模为常数, 故每个新的实例规模为 $\mathrm{O}(|V|)$. 以上归约的时间为:

$$T(|V|;d)=(d+1)^r(\mathrm{poly}(|V|)+\mathrm{O}(1))+\mathrm{poly}((d+1)^r)+\mathrm{poly}((d+1)^r)\leqslant 2^{c\frac{\log(d+1)}{d}|V|},$$

其中, c 为某个常数 (与 G 相关). 任给一个时间运行参数 $\varepsilon>0$, 总可以选取足够大的常数 d, 使得 $T(|V|;d)\leqslant 2^{\varepsilon|V|}$, 故以上归约为亚指数时间归约.

由引理 4 的证明可以看出, 若原问题实例有 N 个点, 多项式时间插值归约会构造常数元 $\mathrm{O}(N)$ 次多项式, 求解其中的 $\mathrm{poly}(N)$ 个系数需要构造 $\mathrm{poly}(N)$ 个新的实例, 其中每个新的实例规模为 $\mathrm{poly}(N)$; 而块插值会构造 $\mathrm{O}\left(\frac{N}{d}\right)$ 元 $\mathrm{O}(N)$ 次多项式, 求解其中的 $\mathrm{O}(2^{\frac{\log(d+1)}{d}N})$ 个系数需要构造 $\mathrm{O}(2^{\frac{\log(d+1)}{d}N})$ 个新的实例, 其中每个新的实例规模为 $\mathrm{O}(N)$. 容易看出, 块插值牺牲归约时间, 将归约中需要构建的新实例规模限制在线性规模 $\mathrm{O}(N)$; 同时, 通过调节常数 d 的大小, 使得构造的新实例个数及整个归约的时间小于给定的 $2^{\varepsilon N}$, 从而构造了亚指数时间归约. 多项式时间插值需要 $\mathrm{O}(N)$ 个线性无关函数的构件帮助构造新的实例, 而块插值仅需要常数个 (d 个). 因此, 对于两个问题 A 和 B, 若能通过多项式时间插值方法证明 $A\leqslant_{\mathrm{poly}}B$, 则必能通过块插值方法证明 $A\leqslant_{\mathrm{serf}}B$.

4 对称双态自旋系统相关的细密度二分定理

考虑 k 正则图上的对称双态自旋系统的配分函数计算问题, 即 Holant($g|=_k$), 其中 k 为正整数, $g:\{0,1\}^2\to\mathbb{C}$ 为

对称布尔函数. 若 $g=[a,0,b]$ ($a,b\in\mathbb{C}$), 则 Holant($g|=_k$) 实例的任一连通分支内的边的赋值只有两种: 全 0 或全 1, 通过穷举即可在多项式时间内计算实例的值; 若 $k\leqslant 2$, 则 Holant($g|=_k$) 任意实例的任一连通分支为一条路径或者一个圈, 通过穷举也可在多项式时间内计算该实例的值.

故只需考虑 $k\geqslant 3$ 且 g 形如 $[a,1,b]$ 的情况.

4.1 一般图

Cai 等人[17,19,20]已经证明该问题的经典二分定理.

定理 7 [17,19,20]. 若函数 $g=[a,1,b]$ ($a,b\in\mathbb{C}$) 满足以下某一条件, 则对于任意正整数 $k\geqslant 3$, Holant($g|=_k$) 有多项式时间算法.

(1) $a=b=0$;

(2) $ab=1$;

(3) $ab=-1$ 且 $a^{2k}=b^{2k}$.

否则, 该问题是#P 难的. 该结论对 Holant($\{g,[0,1],[1,0]\}|=_k$) 仍然成立.

当 g 不满足易解条件时, Cai 等人[17,19,20]以两个已知#P 难的问题: Holant($[c,1,c]|\mathcal{EQ}$) 和 Holant($[0,1,1]|=_3$) 为起始问题, 其中 $c=\pm a\sqrt{b/a}$ 或 c 满足 $c^2=ab$ 且 $2c^3=a^3+b^3$; 并利用构件构造、全息变换和多项式时间插值, 根据 a,b 取值的不同, 分别建立了这两个问题到 Holant($g|=_k$) 的多项式时间归约链. 参考定理 7 的证明, 容易证明以下细密度二分定理.

根据定理 1, 若函数 $g=[a,1,b]$ ($a,b\in\mathbb{C}$) 满足 (1) $a=b=0$ 或 (2) $ab=1$, 或 (3) $ab=-1$ 且 $a^{2k}=b^{2k}$, 则对于任意正整数 $k\geqslant 3$, Holant($g|=_k$) 有多项式时间算法; 否则, 若#ETH 成立, 则存在 $\varepsilon>0$ 使得该问题没有 $2^{\varepsilon N}$ 时间的确定性算法, 其中 N 为输入图点数. 该结论对 Holant($\{g,[0,1],[1,0]\}|=_k$) 仍然成立.

证明: 当 g 满足易解条件时, 定理 7 论证了定理 1. 当 g 不满足易解条件时, 参考定理 7 中#P 困难性的证明[17,19,20], 总可以建立以下某一条归约链.

(1) Holant($[c,1,c]|\mathcal{EQ}$)$\equiv_{\text{serf}}$Holant($\{[c,1,c],[1,0,1]\}|=_3$) (平凡的构件构造)

$\leqslant_{\text{serf}}$Holant($\{[c,1,c]\}\cup\{[x_i,y_i,x_i]\}_{i\geqslant 1}|=_3$) (块插值)

$\leqslant_{\text{serf}}$Holant($[c,1,c]|=_3$) (构件构造, 见文献 [20] 中定理 5 的证明)

$\equiv_{\text{serf}}$Holant($[a,1,b]|=_3$) (见文献 [20] 中引理 5),

其中, $\{[x_i,y_i,x_i]\}_{i\geqslant 1}$ 表示一系列两两线性无关函数的集合; c 满足 $c^2=ab$ 且 $2c^3=a^3+b^3$.

(2) Holant($[ec,1,ec]|=_3$)$\equiv_{\text{serf}}$Holant($[c,1,c]|[1,0,0,e]$) (全息变换, 见文献 [17] 中引理 14)

$\leqslant_{\text{serf}}$Holant($\{[c,1,c],[1,1,1]\}|[1,0,\ldots,0,e]$) (平凡的构件构造)

$\leqslant_{\text{serf}}$Holant($\{[c,1,c]\}\cup\{[x_i,y_i,x_i]\}_{i\geqslant 1}|[1,0,\ldots,0,e]$) (块插值)

$\leqslant_{\text{serf}}$Holant($[c,1,c]|[1,0,\ldots,0,e]$) (构件构造, 见文献 [17] 中引理 15, 16, 17)

$\equiv_{\text{serf}}$Holant($[a,1,b]|=_k$) (全息变换, 见文献 [17] 中引理 14),

其中, $c=a\left(\frac{b}{a}\right)^{\frac{1}{2}}$, $e=\pm 1$, $\{[x_i,y_i,x_i]\}_{i\geqslant 1}$ 表示一系列两两线性无关函数的集合.

(3) Holant($[0,1,1]|=_3$)$\leqslant_{\text{serf}}$Holant($[0,1,1]|=_k$) (构件构造, 见文献 [19] 中引理 15)

$\leqslant_{\text{serf}}$Holant($[a,1,b]|\{=_k\}\cup\{[x_1,0,y_1],\ldots,[x_m,0,y_m]\}$) (构件构造, 见文献 [20] 引理 4)

$\leqslant_{\text{serf}}$Holant($[a,1,b]|\{=_k\}\cup\{[x_j,0,y_j]\}_{j\geqslant 1}$) (块插值)

$\equiv_{\text{serf}}$Holant($[a,1,b]|=_k$) (构件构造, 见文献 [19] 中第 4.2 节),

其中, m 为某个常数, $[x_1,0,y_1],\ldots,[x_m,0,y_m]$ 为 m 个二元函数, 它们仅与 a,b,k 有关; $\{[x_j,0,y_j]\}_{j\geqslant 1}$ 表示一系列两两线性无关函数的集合.

以上归约链中涉及的块插值方法皆与引理 4 中类似. 根据定理 4, 存在正数 ε_1 使得 Holant($[c,1,c]|\mathcal{EQ}$) 没有 $2^{\varepsilon_1 N}$ 时间算法, 则由归约链 (1) 得 Holant($[ec,1,ec]|=_3$) 对某个正数 ε_2 没有 $2^{\varepsilon_2 N}$ 时间算法; 又根据定理 5, 存在正数 ε_3 使得Holant($[0,1,1]|=_3$) 也没有 $2^{\varepsilon_3 N}$ 时间算法. 由此, 对任意不满足易解条件的 g 所定义的 Holant($[a,1,b]|=_k$) 问题,

总有上述某一条归约链证明了其对于某个正数 ε 没有 $2^{\varepsilon N}$ 时间的确定性算法. 故定理 1 得证.

4.2 平面图

接下来考虑 Holant($g|=_k$) 问题限制到平面图时, 该问题的细密度复杂性.

若存在 $\{g|=_k\}$ 门能实现一个四元布尔函数 $Cr(x_1,y_1,x_2,y_2)$, 该函数仅在 $x_1=x_2$ 且 $y_1=y_2$ 时取值为 1, 其余情况取值为 0, 则对于 Holant($g|=_k$) 的任意实例 G, 利用 Cr 替换其内部的交叉结构, 可以构建一个 Holant($g|=_k$) 的平面图实例 G', 有 $\#(G')=\#(G)$. 这样就构建了 Holant($g|=_k$) 到 pl-Holant($g|=_k$) 的多项式时间归约. G 为 Holant($g|=_k$) 的实例, 有 $|E(G)|=\mathrm{O}(|V(G)|)$, 故 G 中交叉数量为 $\mathrm{O}(|E(G)|^2)=\mathrm{O}(|V(G)|^2)$; 则 G' 的点数为 $\mathrm{O}(|V(G)|^2)$. 若 $\#(G)$ 不能在 $2^{\mathrm{O}(|V(G)|)}$ 时间内计算, 容易证明 $\#(G')$ 定不能在 $2^{\mathrm{O}(\sqrt{|V(G')|})}$ 时间内计算. 此情况下, 容易将一般图上问题的细密度复杂性下界发展到平面图上对应问题的细密度复杂性下界. 遗憾的是, 很多情况下 $\{g|=_k\}$ 门不能直接实现 Cr, 需要借助一些一元函数, 如 pl-Holant($\{[0,1,1],[1,-1]\}|=_3$) 的细密度复杂性下界 (定理 6) 证明就依赖于 $[1,-1]$ 帮助构造 Cr.

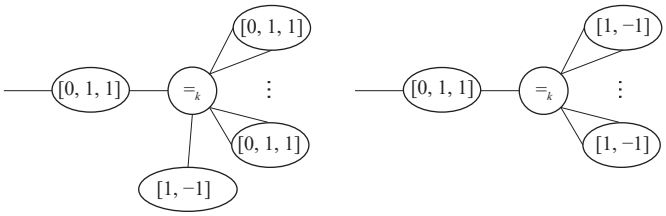
先对 pl-Holant($\{[0,1,1],[1,-1]\}|=_3$) 问题作一个简单的扩展.

引理 5. k 是不小于 3 的整数. 若#ETH 成立, 则存在 $\varepsilon>0$ 使得 pl-Holant($\{[0,1,1],[1,-1]\}|=_k$) 不能在 $2^{\varepsilon\sqrt{N}}$ 时间内计算, 其中 N 表示输入图的点数. 若 rETH 成立, 即使限制输入实例的值非 0 即 1, 该下界仍成立.

证明: 令 G 为 pl-Holant($\{[0,1,1],[1,-1]\}|=_3$) 的一个实例. 不妨设 $|V(G)|=n$.

若 k 为奇数, 利用 $(=_k)$ 与 $(k-3)$ 个 $[1,-1]$ 相连则可得到右侧的 $(=_3)$ 函数, 将 G 中的 $(=_3)$ 函数替换成这样的等价构件, 则构造了 pl-Holant($\{[0,1,1],[1,-1]\}|=_k$) 的实例 G', 有 $\#(G')=\#(G)$.

若 k 为偶数, 则利用图 3(a) 所示构件可以实现左侧 $[-1,-1]$; 再利用 $(k-3)$ 个 $[-1,-1]$ 与 $(=_k)$ 相连则可得到右侧的 $[-1,0,0,-1]$ 函数. 若 G 中有偶数个 $(=_3)$ 函数, 将每个 $(=_3)$ 替换成 $[-1,0,0,-1]$ 相应构件, 则构造了 pl-Holant($\{[0,1,1],[1,-1]\}|=_k$) 的实例 G', 且 $\#(G')=\#(G)$. 若 G 中有奇数个 $(=_3)$ 函数, 则将每个 $(=_3)$ 替换成 $[-1,0,0,-1]$ 相应构件, 得到的实例 G' 满足 $\#(G')=-\#(G)$. 在 G' 中额外加入一个连通分支构造新的实例 G'', 该连通分支是一个 $(=_k)$ 与 1 个 $[-1,0]$ 和 $(k-1)$ 个 $[1,-1]$ 相连, 其中左侧的 $[-1,0]$ 由图 3(b) 所示构件实现. 新增加的这一连通分支带来一个 -1 因子, 故 $\#(G'')=-\#(G')=\#(G)$. 重命名 $G'=G''$.



(a) 实现 $[-1,-1]$ (b) 实现 $[-1,0]$

图 3 两个 $\{[0,1,1],[1,-1]|=_k\}$ 门, 其中 k 为大于 3 的偶数

通过以上方式, 总能在 poly(n) 内构造出 G', 有 $\#(G')=\#(G)$. 假设 $|V(G')|=N$, 有 $N=\mathrm{O}(n)$.

假设引理不成立, 则对任意 $\varepsilon>0$, 使得 $\#(G')$ 能在 $2^{\varepsilon\sqrt{N}}$ 时间内计算. 那么任给一个 G, 通过上述方式构造出 G', 从而计算 $\#(G')=\#(G)$, 整个算法的总时间为: $2^{\varepsilon\sqrt{N}}+\mathrm{poly}(n)\leqslant 2^{\varepsilon c\sqrt{n}}$, 其中 c 为某个常数. 在 n 足够大时, 对于任意时间运行参数 $\varepsilon'>0$, 总可以选择足够小的 ε, 使得 $2^{\varepsilon c\sqrt{n}}\leqslant 2^{\varepsilon'\sqrt{n}}$ 时间内能计算出 $\#(G)$, 这与定理 6 矛盾, 故引理成立.

由上述证明可以看到, 在平面图问题之间传递 $2^{\Omega(\sqrt{N})}$ 时间下界, 需要构造 $2^{\mathrm{O}(\sqrt{N})}$ 时间的归约, 同时归约过程中产生的新实例的规模要求仅线性变化. 这样的归约简称为根号亚指数时间归约. 可以看到, 常数规模的构件构造和全息变换仍然适用, 但多项式时间插值和块插值均已失效; 因此, 探索平面图上 Holant($g|=_k$) 问题的细密度二分定理, 需要新的归约思路和归约方法, 以建立根号亚指数时间归约.

参考定理 1 的证明, 假设 $\mathcal{F}$ 和 $\mathcal{H}$ 是两个布尔函数集, 建立 Holant($\mathcal{F}$) 到 Holant($\mathcal{H}$) 的归约, 本质是以直接 (构件构造和全息变换) 或间接 (插值) 的方式, 利用 $\mathcal{H}$ 内的函数实现 $\mathcal{F}$ 内所有函数. 因此, $\mathcal{F}$ 内包含的函越容易被实

现 (例如个数越少、形式越简单), 构造归约的难度就越低. 如定理 1 的证明就选择了 Holant($[0,1,1]|=_3$) 问题为归约起始. 遗憾的是, 在#ETH 假设下, pl-Holant($[0,1,1]|=_3$) 问题是否有 $2^{\Omega(\sqrt{N})}$ 时间下界还未可知, 本文只能够退一步选择 pl-Holant($\{[0,1,1],[1,-1]\}|=_3$) 问题为归约起始来探讨平面 Holant($g|=_k$) 的复杂性. 这就带来了一个新的归约思路: 利用更强的假设, ETH 或 rETH, 探寻参数函数集更容易被实现的平面 Holant 问题, 以其为归约起始来降低归约构造的难度, 从而避开插值的使用. 第 4.2.1 节利用了此思路, 虽然没有直接找到参数函数集比 $\{[0,1,1],[1,-1]|=_k\}$ 更简单的平面 Holant 问题, 但根据定理 6, 在 rETH 假设下, pl-Holant($\{[0,1,1],[1,-1]\}|=_3$) 问题在限制实例的值非 0 即 1 时, 仍有 $2^{\Omega(\sqrt{N})}$ 时间下界. 这意味着, 对于任意正整数 p, 在 mod p 的环境下, 只需实现与 $\{[0,1,1],[1,-1]|=_3\}$ 在 mod p 意义下等价的函数即可, 而不需要严格实现 $\{[0,1,1],[1,-1]|=_3\}$. 这就降低了归约难度, 使得插值技术能被规避.

4.2.1 rETH 假设下规避插值

以引理 4 的证明为例, 不妨设 $\mathcal{H}=\{[1,a]\}$ ($a\in\mathbb{Z}$). 在 mod p (p 为正整数) 的环境中, 若使用多项式时间插值, 根据费马小定理, 建立的 n 次一元多项式会被降元到 $p-2$ 次多项式, 仅需构造 $p-1$ 个新的实例, 即只需构造出 $p-1$ 个形如 $[1,x]$ 的两两线性无关函数, 来进一步恢复出该多项式的系数. 在 mod p 的环境中, 形如 $[1,x]$ 的两两线性无关函数总共只有 p 个, 这就意味着在进行多项式时间插值时, $[1,a]$ 以接近 1 的概率已经被某个构件直接构造出来, 可以直接利用该构件来建立归约, 而无需进行插值. Guo 等人[33]在研究模 p 约束求解数目问题时, 提出了这一观察, 并论证了在 mod p 的环境中, 对于可表达为非退化矩阵的二元函数 A, 总能利用常数规模构件实现二元函数 A^{-1}. 令 $\mathbb{Z}_p=\{x \bmod p|x\in\mathbb{Z}\}$.

引理 6[33]. p 为正整数. 对任意非退化 2×2 矩阵 $A\in\mathbb{Z}_p$, 总存在正整数 k, 使得 $A^k \bmod p = A^{-1} \bmod p$.

任何一个有理数都可以表示为两个整数之商, 即对任意 $c\in\mathbb{Q}$, 存在 $a,b\in\mathbb{Z}$ 且 $b\neq0$, 有 $c=a/b$. 根据费马小定理, 对任意与 b 互素的质数 p, $c \bmod p = ab^{p-2} \bmod p$. 若 a,b 均与 p 互素, 则称 c 与 p 互素.

推论 1. $A\in\mathbb{Q}$ 为 2×2 矩阵, 将其每一项写成分数形式, p 是与每项分母互素的质数. 若 $A \bmod p$ 非退化, 则总存在正整数 k, 有 $A^k \bmod p = A^{-1} \bmod p$.

想规避掉插值, 则需要寻找一个合适的归约起始问题, 该问题对于某个正整数 p, 在 mod p 的环境中仍有 $2^{\Omega(\sqrt{N})}$ 时间下界. 根据定理 6, 在 rETH 假设下, 限制 pl-Holant($\{[0,1,1],[1,-1]\}|=_k$) 实例的值非 0 即 1, 问题依然有 $2^{\Omega(\sqrt{N})}$ 时间复杂性下界. 这恰好为取模运算的使用提供了便利. 换而言之, 要证明一个平面计数问题 B 在 rETH 假设下的细密度下界, 只需建立受限 pl-Holant($\{[0,1,1],[1,-1]\}|=_k$) 到 B 的归约: 对于 pl-Holant($\{[0,1,1],[1,-1]\}|=_k$) 的任一值非 0 即 1 的实例 G, 构造出 B 的一个实例 G', 选取一个合适的指数 p, 使得 $\#(G') \bmod p\neq0$ 则 $\#(G)=1$, $\#(G') \bmod p=0$ 则 $\#(G)=0$; 并保证归约时间为 $2^{O(\sqrt{|V(G)|})}$ 时间且 $|V(G')|=O(|V(G)|)$ 即可.

根据定理 2, 若函数 $g=[a,1,b]$ ($a,b\in\mathbb{Q}$) 满足 (1) $a=b=0$, 或 (2) $ab=1$, 或 (3) $ab=-1$ 且 $a^{2k}=b^{2k}$ (即 $(a,b)\in\{(1,-1),(-1,1)\}$), 则对于任意正整数 $k\geqslant3$, 平面 Holant($\{g,[0,1]\}|=_k$) 有多项式时间算法; 否则, 若 rETH 成立, 则存在 $\varepsilon>0$, 使得该问题不能在 $2^{\varepsilon\sqrt{N}}$ 时间内计算, 其中 N 表示输入图的点数.

条件被满足时, 根据定理 7, 多项式时间算法已被 Cai 等人[17,19,20]给出; 只需考虑证明定理 2 的难解性. 本节包含的证明中, $\equiv_{(\bmod p)}$ 符号简写为 $=$.

引理 7. 若 rETH 成立, 对于任意 $b\in\mathbb{Q}-\{0\}$ 和任意正整数 $k\geqslant3$, 存在 $\varepsilon>0$, 使得 Holant($[0,1,b]|=_k$) 不能在 $2^{\varepsilon\sqrt{N}}$ 时间内计算, 其中 N 表示输入图的点数.

证明: 令 p 是与 b 互素的质数, 即 $b \bmod p\neq0$. 利用图 4 所示 $\{[0,1,b]|=_k\}$ 门实现了右侧的 $[0,1,b^{k-1}]$ 函数. 又通过图 5 所示的 $\{[0,1,b]|=_k\}$ 门, 选取满足推论 1 的 t, 可实现左侧的二元函数 $\begin{pmatrix}0&1\\1&b\end{pmatrix}\left(\begin{pmatrix}0&1\\1&b^{k-1}\end{pmatrix}\begin{pmatrix}0&1\\1&b\end{pmatrix}\right)^t=\begin{pmatrix}0&1\\1&b\end{pmatrix}\left(\begin{pmatrix}0&1\\1&b^{k-1}\end{pmatrix}\begin{pmatrix}0&1\\1&b\end{pmatrix}\right)^{-1}=\begin{pmatrix}0&1\\1&b^{k-1}\end{pmatrix}^{-1}=[-b^{k-1},1,0]$. 由此也可看出, 对于一侧的非退化二元函数 A, 总可以实现另一侧二元函数 A^{-1}.

将图 4 中一个 $[0,1,b]$ 替换成 $[-b^{k-1},1,0]$, 则可得到右侧的二元不等函数 $[0,1,0]$. 对任意正整数 l, 利用形如

$[0,1,b],[0,1,0],\ldots,[0,1,b]$ 的路径构件, 可实现函数 $\begin{pmatrix}0&1\\1&b\end{pmatrix}\left(\begin{pmatrix}0&1\\1&0\end{pmatrix}\begin{pmatrix}0&1\\1&b\end{pmatrix}\right)^{l}=[0,1,lb]$. 定存在 l_1,l_2, 使得 $l_1b=-1$ 和 $l_2b=1$. 故存在 $\{[0,1,b]|=_k\}$ 门实现左侧函数 $[0,1,1]$ 和 $[0,1,-1]$.

利用图 6 结构, 可以实现右侧 $[0,1]$ 或 $[0,1]^{\otimes 2}$. 若 k 为奇数, 利用 $[0,1,1]$ 与 $[0,1]$ 相连可实现左侧函数 $[1,1]$; 利用 $[0,1,-1]$ 与 $[0,1]$ 相连可实现左侧函数 $[1,-1]$. 对于 pl-Holant($\{[0,1,1],[1,-1]\}|=_k$) 的任一实例 G, $\#(G)$ 非 0 即 1, 可以将其中的 $[0,1,1]$ 和 $[1,-1]$ 都替换成上述对应的 $\{[0,1,b]|=_k\}$ 门, 得到 pl-Holant($[0,1,b]|=_k$) 的实例 G', 有 $\#(G')=\#(G)$.

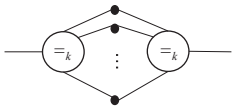


图 4 一个实现 $[0,1,b^{k-1}]$ 函数的 $\{[0,1,b]|=_k\}$ 门, 黑色实心点表示二元函数 $[0,1,b]$

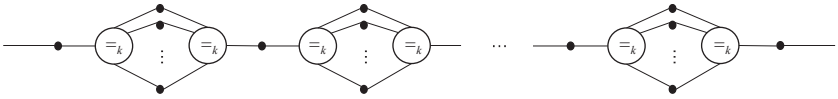
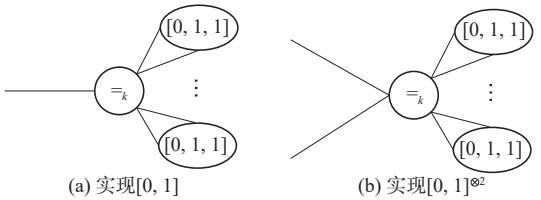


图 5 一个 $\{[0,1,b]|=_k\}$ 门



(a) 实现[0, 1] (b) 实现[0, 1]$^{\otimes 2}$

图 6 实现 $[0,1]$ (k 为奇数) 或 $[0,1]^{\otimes 2}$ (k 为偶数) 的 $\{[0,1,b]|=_k\}$ 门

若 k 为偶数, 利用两个 $[0,1,1]$ 与一个 $[0,1]^{\otimes 2}$ 相连可实现左侧函数 $[1,1]^{\otimes 2}$; 利用两个 $[0,1,-1]$ 与一个 $[0,1]^{\otimes 2}$ 相连可实现左侧函数 $[1,-1]^{\otimes 2}$. 对于 pl-Holant($\{[0,1,1],[1,-1]\}|=_k$) 的任一实例 G, $\#(G)$ 非 0 即 1, 由于 k 为偶数, 故 $[1,-1]$ 在 G 中的出现次数必为偶数次, 通过调整, 在保持平面性的同时, 可以保持每个面内 $[1,-1]$ 的个数是偶数. 利用实现 $[1,-1]^{\otimes 2}$ 的 $\{[0,1,b]|=_k\}$ 门去替换同个面内相邻的成对 $[1,-1]$; 同时将其中的 $[0,1,1]$ 也替换成对应的 $\{[0,1,b]|=_k\}$ 门, 从而得到 pl-Holant($[0,1,b]|=_k$) 的实例 G', 有 $\#(G')=\#(G)$.

上述两种情况都是利用常数规模的 $\{[0,1,b]|=_k\}$ 门实现 $[0,1,1]$ 和 $[1,-1]$, 故 $|V(G')|=O(|V(G)|)$, 且在多项式时间内能构造出 G'. 故定理得证.

在 $\bmod p$ 环境下, 仍可以使用归一化操作, 将函数 F 替换为 λF, 来进行函数 F 和 λF 之间的替换, 但需要保证 $\lambda \bmod p\neq 0$.

引理 8. a, b 为两个有理数, 满足 $(a,b)\notin\{(1,-1),(-1,1)\}$ 且 $ab\notin\{0,1\}$. 若 rETH 成立, 则存在 $\varepsilon>0$, 对任意正整数 $k\geqslant 3$ 有 pl-Holant($\{[a,1,b],[0,1]\}|=_k$) 不能在 $2^{\varepsilon\sqrt{N}}$ 时间内计算, 其中 N 表示输入图的点数.

证明: 若 $ab=-1$, 即 $[a,1,b]=\frac{1}{b}[-1,b,b^2]$, 有 $b\notin\{0,\pm1\}$. k 为偶数时, 利用一个 $(=_k)$ 与 $\frac{k-2}{2}$ 个 $[-1,b,b^2]$ 相连, 形如图 6 中构件, 可得右侧二元函数 $[\pm1,0,b^{k-2}]$; 进一步将该二元函数左右两端都与 $[-1,b,b^2]$ 相连, 则可以得到左侧二元函数 $[b^k\pm1,b^{k+1}\mp b,b^{k+2}\pm b^2]$, 归一化为 $[a',1,b']$. 容易验证, $a'b'\notin\{0,\pm1\}$ 且 $a'\neq\pm b'$. k 为奇数时, 利用一个 $(=_k)$ 与 $\frac{k-1}{2}$ 个 $[-1,b,b^2]$ 相连, 得到右侧一元函数 $[\pm1,b^{k-1}]$; 将其与 $[-1,b,b^2]$ 相连得到左侧 $[b^k\pm1,b(b^k\mp1)]$; 再将 $(=_k)$ 与 $\frac{k-3}{2}$ 个 $[-1,b,b^2]$ 和一个 $[b^k\pm1,b(b^k\mp1)]$ 相连得到 $[1\pm b^k,0,b^{k-2}(b^k\mp1)]$. 进一步将该二元函数左右两端都与 $[-1,b,b^2]$ 相连, 则可以得到左侧二元函数 $[b^k(b^k-1)+(b^k+1),b^{k+1}(b^k-1)-b(b^k+1),b^{k+2}(b^k-1)+b^2(b^k+1)]$ 或 $[b^k(b^k+1)-(b^k-1),b^{k+1}(b^k+1)+b(b^k-1),b^{k+2}(b^k+1)-b^2(b^k-1)]$, 记为 $[\alpha,\beta,\gamma]$. 由于 $b\notin\{0,\pm1\}$ 且 b 为有理数, 则

$\beta \neq 0$, 故可归一化为 $[a',1,b']$, 满足 $a'b' \notin \{0,\pm 1\}$ 且 $a' \neq b'$. 故不妨设 $ab \neq -1$.

令 p 是与 $a,b,(ab+1),(ab-1)$ 互素的质数.

(1) $k=3$. 两个 $(=_3)$ 与通过两个 $[a,1,b]$ 连接, 类似图 4 结构, 可以实现右侧 $[a^2,1,b^2]$ 函数; 由于 $ab \neq \pm 1$, 则矩阵 $\begin{pmatrix} a^2 & 1 \\ 1 & b^2 \end{pmatrix}$ 可逆, 可实现左侧 $\begin{pmatrix} a^2 & 1 \\ 1 & b^2 \end{pmatrix}^{-1}$ 即 $[b^2,-1,a^2]$ 函数. 两个 $[0,1]$ 与一个 $(=_3)$ 相连得到右侧 $[0,1]$; 将其与左侧 $[b^2,-1,a^2]$ 函数相连得左侧 $[-1,a^2]$; 再将 $[-1,a^2]$ 与 $(=_3)$ 相连得右侧的 $[-1,0,a^2]$. 通过 $\begin{pmatrix} a & 1 \\ 1 & b \end{pmatrix}\begin{pmatrix} -1 & 0 \\ 0 & a^2 \end{pmatrix}\begin{pmatrix} a & 1 \\ 1 & b \end{pmatrix} = \begin{pmatrix} 0 & a^2b-a \\ a^2b-a & a^2b^2-1 \end{pmatrix}$ 实现左侧 $[0,a,ab+1]$ 函数. 根据引理 7, 得证.

(2) $k=4$. 利用 3 个 $[0,1]$ 与 $(=_4)$ 相连可得到右侧 $[0,1]$; 进一步利用 $[0,1]$ 和左侧的 $[a,1,b]$ 连接得到左侧的 $[1,b]$; 再利用两个 $[1,b]$ 与一个 $(=_4)$ 连接, 可得右侧 $[1,0,b^2]$ 函数, 从而得到左侧逆函数 $[b^2,0,1]$. 利用两个 $[a,1,b]$、一个 $[b^2,0,1]$ 与两个 $(=_4)$ 连接, 得到右侧 $b^2[a^2,0,1]$, 进一步得到左侧逆函数 $[1,0,a^2]$. 又通过图 4 类似结构, 可实现右侧 $[a^3,1,b^3]$ 函数, 从而得到左侧对应逆函数 $[b^3,-1,a^3]$; 利用 $[0,1]$ 和左侧的 $[b^3,-1,a^3]$, 可得左侧的 $[-1,a^3]$; 再将两个 $[-1,a^3]$ 与一个 $(=_4)$ 连接, 可得右侧 $[1,0,a^6]$ 函数. 对于任意 $l \in \mathbb{N}$, 通过

$$\begin{pmatrix} 1 & 0 \\ 0 & a^2 \end{pmatrix}\left(\begin{pmatrix} 1 & 0 \\ 0 & a^6 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & a^2 \end{pmatrix}\right)^l = \begin{pmatrix} 1 & 0 \\ 0 & a^{2+8l} \end{pmatrix}$$

可得到左侧的 $[1,0,a^{2+8l}]$. 令 p 是与 $a,b,(ab+1),(ab-1)$ 互素且形如 $3+8l$ 的质数 (由于质数是无限的, 故总能取到这样的 p), 即存在 l_0 使得 $2+8l_0=p-1$, 根据费马小定理, 通过上述构造可获得左侧函数 $[1,0,a^{2+8l_0}]=[1,0,1]$ 函数. 获得了二元相等函数后无需再注意左右侧. 通过两个 $[a,1,b]$、一个 $[1,0,1]$ 与两个 $(=_4)$ 连接获得 $[a^2,1,b^2]$, 其逆矩阵为 $[b^2,-1,a^2]$; 再利用一个 $[a,1,b]$, 一个 $[b^2,-1,a^2]$ 和一个 $[1,0,1]$ 与两个 $(=_4)$ 连接获得 $[ab^2,-1,a^2b]$. $[a,1,b]$ 的逆矩阵对应 $[b,-1,a]$, 其与 $[0,1]$ 相连获得 $[-1,a]$; 利用 $(=_4)$ 与 $[-1,a]$ 和 $[1,b]$ 相连获得 $[-1,0,ab]$; 再利用两个 $[-1,0,ab]$, 一个 $[1,0,1]$ 与两个 $(=_4)$ 连接, 形如图 4 构件, 实现 $[1,0,a^2b^2]$. 可实现 $\begin{pmatrix} ab^2 & -1 \\ -1 & a^2b \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & a^2b^2 \end{pmatrix}\begin{pmatrix} a & 1 \\ 1 & b \end{pmatrix} = \begin{pmatrix} 0 & ab^2-a^2b^3 \\ a^4b^3-a & a^4b^4-1 \end{pmatrix}$ 函数; 再利用两个 $(=_4)$ 与 $[1,0,1]$, $\begin{pmatrix} 0 & ab^2-a^2b^3 \\ a^4b^3-a & a^4b^4-1 \end{pmatrix}$ 和 $\begin{pmatrix} 0 & a^4b^3-a \\ a^4b^3-a & a^4b^4-1 \end{pmatrix}$ 相连, 形如图 4 构件, 得到 $[0,a^5b^5+a^3b^3-a^6b^6-a^2b^2,a^4b^4-1]$. 根据引理 7, 得证.

(3) $k>4$. 考虑造出左侧 $(=_2)$, 从而将 $(=_k)$ 与 $\frac{k-3}{2}$ 或 $\frac{k-4}{2}$ 个 $(=_2)$ 相连获得 $(=_3)$ 或 $(=_4)$, 回到上述情况.

若 k 为奇数, 利用一个 $(=_k)$ 与 $\frac{k-1}{2}$ 个 $[a,1,b]$ 相连, 形如图 6 中构件, 实现右侧的 $[a^{\frac{k-1}{2}},b^{\frac{k-1}{2}}]$; 进一步实现左侧逆函数 $[b^{\frac{k-1}{2}},a^{\frac{k-1}{2}}]$; 再利用一个 $(=_k)$ 与 $(k-3)/2$ 个 $[a,1,b]$ 和一个 $[b^{\frac{k-1}{2}},a^{\frac{k-1}{2}}]$ 相连, 实现右端 $[b,0,a]$ 函数, 从而实现左侧逆函数 $[a,0,b]$. 通过两个 $(=_k)$ 与 $(k-2)$ 个 $[a,1,b]$ 和一个 $[a,0,b]$ 可以实现右侧的 $[a^{k-1},0,b^{k-1}]$. 对任意 $l \in \mathbb{N}$, 通过由形如 $[a,0,b],[a^{k-1},0,b^{k-1}],\ldots,[a,0,b]$ 的路径构件实现左侧的 $[a^{kl+1},0,b^{kl+1}]$. 令 p 是与 $a,b,(ab+1),(ab-1)$ 互素且形如 $kl+2$ 的质数, 即对某个 l_1 有 $p-1=kl_1+1$, 根据费马小定理, $[a^{kl_1+1},0,b^{kl_1+1}]$ 实现了左侧 $[1,0,1]$.

若 k 为偶数, 则与情况 (2) 类似, 利用 $[0,1]$ 和左侧的 $[a,1,b]$ 连接得到左侧的 $[1,b]$; 再利用 $(k-2)$ 个 $[1,b]$ 与一个 $=_k$ 连接, 可得右侧 $[1,0,b^{k-2}]$ 函数, 从而得到左侧逆函数 $[b^{k-2},0,1]$. 利用 $(k-2)$ 个 $[a,1,b]$ 和一个 $[b^{k-2},0,1]$ 与两个 $=_k$ 连接, 得到右侧 $b^{k-2}[a^{k-2},0,1]$, 进一步得到左侧逆函数 $[1,0,a^{k-2}]$. 又通过图 4 类似结构, 可实现右侧 $[a^{k-1},1,b^{k-1}]$ 函数, 从而得到左侧对应逆函数 $[b^{k-1},-1,a^{k-1}]$; 利用 $[0,1]$ 和左侧的 $[b^{k-1},-1,a^{k-1}]$, 可得左侧的 $[-1,a^{k-1}]$; 再将 $(k-2)$ 个 $[-1,a^{k-1}]$ 与一个 $=_k$ 连接, 可得右侧 $[1,0,a^{(k-1)(k-2)}]$. 对任意 $l \in \mathbb{N}$, 通过 $\begin{pmatrix} 1 & 0 \\ 0 & a^{k-2} \end{pmatrix}\left(\begin{pmatrix} 1 & 0 \\ 0 & a^{(k-1)(k-2)} \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & a^{k-2} \end{pmatrix}\right)^l = \begin{pmatrix} 1 & 0 \\ 0 & a^{(k-2)+k(k-2)l} \end{pmatrix}$ 实现左侧 $[1,,0,a^{(k-2)+k(k-2)l}]$. 令 p 是与 $a,b,(ab+1),(ab-1)$ 互素且形如 $(k-1)+k(k-2)l$ 的质数, 即存在 l_2 使得 $(k-1)+k(k-2)l_2=p-1$, 根据费马小定理, 可获得左侧函数 $[1,0,a^{2+8l_0(k-2)+k(k-2)l_2}]=[1,0,1]$ 函数.

在比#ETH 更强的 rETH 假设下, 本节借助取模运算规避插值, 从而证明了定理 2. 但该方法受到取模运算定

义的局限性, 无法推广到更一般的实数域乃至复数域.

4.2.2 插值与小规模构件结合

仍然考虑插值方法. 想构造根号亚指数归约, 多项式时间插值的局限性在于新实例规模非线性变化; 块插值的局限性在于所需构造的新实例个数过多 (归约时间过长), 即使增大 d 至非常数, 如 $\sqrt{N}$, 也无法在满足根号亚指数归约时间的同时保持新实例规模仅线性变化. 一个直观的想法是, 减少模拟线性无关函数构件的规模. 考虑一个 N 个点的平面图, 若每个点随机赋予函数 A 或 B, 则会产生 2^N 个构件 (有重复), 这暗示了 N 个点的平面构件所能实现的函数个数极多, 很多时候远超所需要的 N 个. 根据这点, 本节提出了仅利用 $\mathrm{O}(\log N)$ 规模的构件来实现 N 个两两线性无关函数的方案, 创新地将其与多项式时间插值结合, 从而在更弱的#ETH 假设下探索 pl-Holant $([a,1,b]|=_k)$ 相关的细密度复杂性.

根据定理 3, 若#ETH 成立, 对于任意正实数 b 和任意正整数 $k\geqslant 3$, 存在 $\varepsilon>0$, 使得 pl-Holant$([0,1,b]|=_k)$ 不能在 $2^{\varepsilon\sqrt{N/\log N}}$ 时间内计算, 其中 N 表示输入图的点数.

证明: 利用 $k-3$ 或 $k-2$ 个 $[1,1]$ 与 $(=_k)$ 相连实现 $(=_3)$ 或 $(=_2)$. 若 $b\neq\frac{1}{2}$, 则利用图 7 所示构件, 令 $x=-\frac{2}{b^2}$, $y=-\frac{2}{b(2b-1)}$, 实现 $[0,1,1]$; 否则, $b=\frac{1}{2}$, 令 $x=-4,y=-2$, 图 7 所示构件实现 $[0,4,0]$ 函数, 再利用路径构件: $\left[0,1,\frac{1}{2}\right],[1,0,1],[0,4,0],[1,0,1],\left[0,1,\frac{1}{2}\right]$, 实现 $[0,1,1]$. 即根据构件构造有 pl-Holant$(\{[0,1,1],[1,-1]\}|=_3)\leqslant_{\mathrm{poly}}$pl-Holant$(\{[0,1,b],[1,x],[1,y],[1,1],[1,-1]\}|=_k)$. 接下来只需要考虑在 pl-Holant$([0,1,b]|=_k)$ 内插值出这 4 个一元函数. 根据引理 3, 若对于任意正整数 n, 可以在 poly(n) 时间内构造 n 个 $\{[0,1,b]|=_k\}$ 门实现 n 个两两线性无关函数, 且每个构件的规模为 $\mathrm{O}(\log n)$, 则 pl-Holant$(\{[0,1,b],[1,x],[1,y],[1,1],[1,-1]\}|=_k)\leqslant_{\mathrm{poly}}$pl-Holant$([0,1,b]|=_k)$, 且若原实例的规模为 N, 生成的新实例的规模为 $\mathrm{O}(N\log N)$. 下面考虑构造所需的两两线性无关函数.

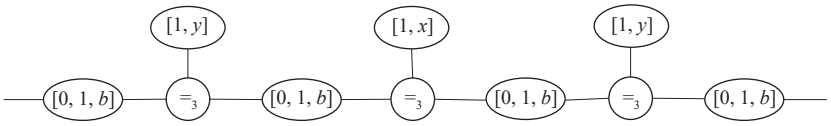


图 7 实现函数 $[0,1,1]$ 的平面构件

利用一个或两个 $[0,1,b]$ 与图 6 类似构造连接可以实现左侧一元函数 $[1,b]$ 或二元函数 $[1,b,b^2]=[1,b]\otimes[1,b]$ (归一化后). 利用图 4 类似构造可以实现函数 $[0,1,b^{k-1}]$; 从而可以构造一个二元函数 $A=\begin{pmatrix}0&1\\1&b\end{pmatrix}\begin{pmatrix}0&1\\1&b^{k-1}\end{pmatrix}=\begin{pmatrix}1&b^{k-1}\\b&b^k+1\end{pmatrix}$, 且总可以通过 A 和左侧一元函数 $[1,x]$ 连接, 实现新的左侧一元函数. 若 k 为奇数, 利用一个 $=_k$ 与 $k-3$ 个一元相等相连得到三元函数 $[1,0,0,b^{k-1}]$; 从而可以实现图 8(a) 所示构件 $B=\begin{pmatrix}0&b^k+1\\b^2&b^{k+1}+b\end{pmatrix}$, 且连接 B 和左侧一元函数 $[1,y]$ 能实现新的左侧一元函数. 记 $s=A\begin{pmatrix}1\\b\end{pmatrix}=(b^k+1)\left[1,b+\frac{b^{k-1}}{b^k+1}\right]$; 利用上述方法, 总可以递归地构造一系列 $\{[0,1,b]|=_k\}$ 门, 实现一元函数集合 $\{M_lM_{l-1}\ldots M_1s|M_i=A\text{或}B,i\in[l],l\in\mathbb{N}^+\}$.

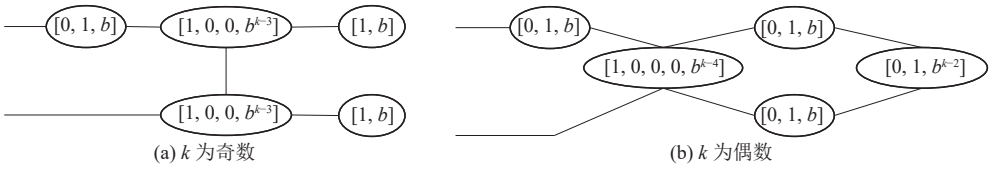


图 8 实现函数 $\begin{pmatrix}0&b^k+1\\b^2&b^{k+1}+b\end{pmatrix}$ 的构件

归纳法证明 $\{M_lM_{l-1}\ldots M_1s\}$ 集合内函数两两线性无关. 若 $l=1$, $As=[b^{2k+1}+2b^k+1,b^{2k+3}+b^{k+2}+b^{k+1}+b]$ 和 $Bs=[b^{2k+2}+b^{k+1}+b^{k+2}+b,b^{2k+3}+b^{k+3}+2b^{k+1}+2b^2]$ 线性无关. 假设从 $\{M_{l-1}\ldots M_1s\}$ 内任取两个一元函数, 归一化后记

作 $[1,x]$ 和 $[1,y]$, 它们线性无关, 即 $x\neq y$. 容易判断 $A\begin{pmatrix}1\\x\end{pmatrix}$ 和 $A\begin{pmatrix}1\\y\end{pmatrix}$, $B\begin{pmatrix}1\\x\end{pmatrix}$ 和 $B\begin{pmatrix}1\\y\end{pmatrix}$ 线性无关. $A\begin{pmatrix}1\\x\end{pmatrix}$ 归一化后为 $[1,b+(b^{k-1}+x^{-1})^{-1}]$, $B\begin{pmatrix}1\\y\end{pmatrix}$ 归一化后为 $[1,b+(b^{k-2}+b^{-2})^{-1}y^{-1}]$; 由于 $s=\left[1,b+\dfrac{b^{k-1}}{b^k+1}\right]$ 且 $b>0$, 故 $b+\dfrac{b^{k-1}}{b^k+1}>b$, 则 $x,y>b$; 又 $b+(b^{k-1}+x^{-1})^{-1}$ 单调递增, $b+(b^{k-2}+b^{-2})^{-1}y^{-1}$ 单调递减, 则 $b+(b^{k-1}+x^{-1})^{-1}<b+(b^{k-1}+b^{-1})^{-1}<b+(b^{k-2}+b^{-2})^{-1}y^{-1}$, 即 $A\begin{pmatrix}1\\x\end{pmatrix}$ 和 $B\begin{pmatrix}1\\y\end{pmatrix}$ 线性无关. 则综上可知 $\{AM_{l-1}\dots M_1 s\}\cup\{BM_{l-1}\dots M_1 s\}$ 集合内函数两两线性无关.

若 k 为偶数, 利用一个 $=_k$ 与 $k-2$ 个一元相等相连得到四元函数 $[1,0,0,b^{k-2}]$; 利用一个 $=_k$ 与 $k-2$ 个一元相等相连得到二元函数 $[1,0,b^{k-2}]$; 从而可以实现图 8(b) 所示构件 $B=\begin{pmatrix}0 & b^k+1\\ b^2 & b^{k+1}+b\end{pmatrix}$. 此时, 令 $s'=A^{\otimes 2}\begin{pmatrix}1\\b\end{pmatrix}^{\otimes 2}$, 由于 k 为偶数, 则 pl-Holant($\{[0,1,b],[1,x],[1,y],[1,1],[1,-1]\}|=_k$) 的任意实例中一元函数出现的个数为偶数, 且可以通过调整平面实例, 使其满足每一个面包含偶数个一元函数, 则 $s'=A^{\otimes 2}\begin{pmatrix}1\\b\end{pmatrix}^{\otimes 2}$ 可视为两个独立的 $s=A\begin{pmatrix}1\\b\end{pmatrix}$ 使用; 同理, 形如 $((M_lM_{l-1}\dots M_1)\otimes(M'_lM'_{l-1}\dots M'_1))s'$ 的函数可以视为两个独立的一元函数 $M_lM_{l-1}\dots M_1 s$ 和 $M'_lM'_{l-1}\dots M'_1 s$ 使用, 其中 $M_i,M'_i=A$ 或 B, $i\in[l]$. 故仍可以按照引理 3 所给方式进行插值.

上述归约方法建立了 pl-Holant($\{[0,1,1],[1,-1]\}|=_3$) 到pl-Holant($[0,1,b]|=_3$) 的多项式时间归约. 不妨设G 是 pl-Holant($\{[0,1,1],[1,-1]\}|=_3$)的一个 N 个点的实例, 则上述归约时间为 poly(N), 归约过程中产生 poly(N) 个 pl-Holant ($[0,1,b]|=_3$) 的实例, 且新实例的点数不超过 $cN\log N$, c 为某个常数. 若定理 3 不成立, 则对任意 $\varepsilon>0$, 每个新实例的值能在 $2^{\varepsilon\sqrt{cN\log N/\log(cN\log N)}}\leqslant 2^{\varepsilon\sqrt{c'N}}$ 时间内计算, c' 为某个常数. 结合上述归约算法, 对于任意 $\epsilon>0$, 在 N 足够大时, 总可以选取足够小的 ε, 使得 $\text{poly}(N)+\text{poly}(N)2^{\epsilon\sqrt{c'N}}\leqslant 2^{\epsilon\sqrt{N}}$ 时间内能计算出 $\#(G)$, 这与定理 6 矛盾.

5 总 结

本文探讨了, #ETH 假设和 rETH 假设下, Holant($[a,1,b]|=_k$) 的相关细密度复杂性. 当无额外图结构限制时, 利用已有的归约技术和证明思路, 容易发展#ETH 假设下 Holant($[a,1,b]|=_k$) 的细密度二分定理. 当要求图结构为平面时, 对归约时间和生成实例规模的进一步限制使得已有的插值方法失效. 本文提出了两种解决方案: 一种是在更强的 rETH 假设下, 只需实现出与目标函数在模 p 意义下等价的函数即可, 从而规避掉插值运算; 第 2 种方法仍在 #ETH 假设下, 利用 O($\log n$) 规模构件来实现 n 个线性无关的函数, 将其与多项式时间插值结合, 降低多项式时间插值产生的新实例的规模. 利用这两种方案, 本文论证了 pl-Holant($\{[a,1,b],[0,1]\}|=_k$) 问题 ($a,b\in\mathbb{Q}$) 在 rETH 下的细密度二分定理和 pl-Holant($[0,1,c]|=_k$) 问题 ($c\in\mathbb{R}^+$) 在#ETH 下的细密度复杂性下界. 这两个结论仍待改进. 一方面思考如何在 pl-Holant($[a,1,b]|=_k$) 内实现 $[0,1]$ 函数, 以得到 pl-Holant($[a,1,b]|=_k$) 在 rETH 下的细密度二分定理; 另一方面是找到小规模构件模拟线性无关函数的充分条件, 扩大改进后的多项式时间插值的适用范围, 以探讨 #ETH 下, 实数域乃至复数域上 pl-Holant($[a,1,b]|=_k$) 问题的细密度下界.

本文在较强的 rETH 假设下, 利用取模操作降低了函数实现的难度, 得到了平面正则图上双态自旋系统相关的 $2^{\Omega(\sqrt{N})}$ 下界结论; 在较弱的#ETH 假设下, 改进了现有插值方法, 得到了相关的 $2^{\Omega(\sqrt{N/(\log N)})}$ 下界结论. 若想在 #ETH 下得到更强的 $2^{\Omega(\sqrt{N})}$ 下界结论, 两个可能的研究思路是: (1) 寻找函数集更易于实现的归约起始问题, 从而规避插值; (2) 本文在使用插值方法时, 求解了所有的系数, 但很多时候只需求解某些特定系数即可, 从这点入手或可改进块插值方法.

References:

[1] Cipra BA. An introduction to the Ising model. The American Mathematical Monthly, 1987, 94(10): 937–959. [doi: 10.1080/00029890.1987.12000742]

[2] Martin PP. Potts Models and Related Problems in Statistical Mechanics. Singapore: World Scientific, 1991. 1–16. [doi: 10.1142/0983].

[3] Kotek T, Makowsky JA, Zilber B. On counting generalized colorings. In: Kaminski M, Martini S, eds. Computer Science Logic. Berlin: Springer, 2008. 339–353. [doi: 10.1007/978-3-540-87531-4_25]

[4] Vadhan SP. The complexity of counting in sparse, regular, and planar graphs. SIAM Journal on Computing, 2001, 31(2): 398–427. [doi: 10.1137/S0097539797321602]

[5] Goodfellow I, Bengio Y, Courville A. Deep Learning. Cambridge: MIT Press, 2016. 587–614.

[6] Valiant LG. The complexity of computing the permanent. Theoretical Computer Science, 1979, 8(2): 189–201. [doi: 10.1016/0304-3975(79)90044-6]

[7] Patel V, Regts G. Deterministic polynomial-time approximation algorithms for partition functions and graph polynomials. SIAM Journal on Computing, 2017, 46(6): 1893–1919. [doi: 10.1137/16M1101003]

[8] Kolmogorov V. A faster approximation algorithm for the Gibbs partition function. In: Proc. of the 31st Conf. on Learning Theory. Stockholm: PMLR, 2018. 228–249.

[9] Qiu GL, Zhang CH. Approximability of partition functions of ferromagnetic two-state spin systems. Computer Science, 2020, 47(5): 22–26 (in Chinese with English abstract). [doi: 10.11896/jsjkx.200200119]

[10] Bai ZL, Wang HP, Cao YZ, Wang LL. Fast sampling algorithms for spin systems on trees. Chinese Journal of Computers, 2022, 45(10): 2093–2116 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2022.02093]

[11] Laba HP, Tkachuk VM. Calculation of partition function of Ising model on quantum computer. Physics Letters A, 2023, 491: 129213. [doi: 10.1016/j.physleta.2023.129213]

[12] Cornelissen A, Hamoudi Y. A sublinear-time quantum algorithm for approximating partition functions. In: Proc. of the 2023 Annual ACM-SIAM Symp. on Discrete Algorithms (SODA). Florence: SIAM, 2023. 1245–1264. [doi: 10.1137/1.9781611977554.ch46]

[13] Barvinok A. Combinatorics and Complexity of Partition Functions. Cham: Springer, 2016. 1–7. [doi: 10.1007/978-3-319-51829-9]

[14] Dyer M, Greenhill C. The complexity of counting graph homomorphisms. Random Structures & Algorithms, 2000, 17(3–4): 260–289. [doi: 10.1002/1098-2418(200010/12)17:3/4<260::AID-RSA5>3.0.CO;2-W]

[15] Goldberg LA, Grohe M, Jerrum M, Thurley M. A complexity dichotomy for partition functions with mixed signs. SIAM Journal on Computing, 2010, 39(7): 3336–3402. [doi: 10.1137/090757496]

[16] Bulatov A, Grohe M. The complexity of partition functions. Theoretical Computer Science, 2005, 348(2-3): 148–186. [doi: 10.1016/j.tcs.2005.09.011]

[17] Cai JY, Kowalczyk M. Spin systems on *k*-regular graphs with complex edge functions. Theoretical Computer Science, 2012, 461: 2–16. [doi: 10.1016/j.tcs.2012.01.021]

[18] Cai JY, Fu ZG, Girstmair K, Kowalczyk M. A complexity trichotomy for *k*-regular asymmetric spin systems using number theory. In: Proc. of the 9th Innovations in Theoretical Computer Science Conf. (ITCS 2018). Cambridge: ITCS, 2018. 2:1–2:22. [doi: 10.4230/LIPIcs.ITCS.2018.2]

[19] Cai JY, Kowalczyk M. Partition functions on *k*-regular graphs with {0, 1}-vertex assignments and real edge functions. Theoretical Computer Science, 2013, 494: 63–74. [doi: 10.1016/j.tcs.2012.12.043]

[20] Kowalczyk M, Cai JY. Holant problems for 3-regular graphs with complex edge functions. Theory of Computing Systems, 2016, 59(1): 133–158. [doi: 10.1007/s00224-016-9671-7]

[21] Impagliazzo R, Paturi R, Zane F. Which problems have strongly exponential complexity? Journal of Computer and System Sciences, 2001, 63(4): 512–530. [doi: 10.1006/jcss.2001.1774]

[22] Impagliazzo R, Paturi R. On the complexity of *k*-SAT. Journal of Computer and System Sciences, 2001, 62(2): 367–375. [doi: 10.1006/jcss.2000.1727]

[23] Cygan M, Fomin FV, Kowalik Ł, Lokshtanov D, Marx D, Pilipczuk M, Pilipczuk M, Saurabh S. Lower bounds based on the exponential-time hypothesis. In: Cygan M, Fomin FV, Kowalik Ł, Lokshtanov D, Marx D, Pilipczuk M, Pilipczuk M, Saurabh S, eds. Parameterized Algorithms. Cham: Springer, 2015. 467–521. [doi: 10.1007/978-3-319-21275-3_14]

[24] Dell H, Husfeldt T, Marx D, Taslaman N, Wahlén M. Exponential time complexity of the permanent and the Tutte polynomial. ACM Trans. on Algorithms (TALG), 2014, 10(4): 21. [doi: 10.1145/2635812]

[25] Curticapean R. Block interpolation: A framework for tight exponential-time counting complexity. Information and Computation, 2018, 261: 265–280. [doi: 10.1016/j.ic.2018.02.008]

[26] Liu Y. Exponential time complexity of the complex weighted Boolean #CSP. In: Wu WL, Tong G, eds. Computing and Combinatorics: 29th Int'l Conf. Cham: Springer, 2024. 83–96. [doi: 10.1007/978-3-031-49190-0_6]

[27] Björklund A, Kaski P. The fine-grained complexity of computing the Tutte polynomial of a linear matroid. In: Proc. of the 32nd Annual ACM-SIAM Symp. on Discrete Algorithms. Philadelphia: SIAM, 2021. 2333–2345.

[28] Brand C, Dell H, Roth M. Fine-grained dichotomies for the Tutte plane and Boolean #CSP. Algorithmica, 2019, 81(2): 541–556. [doi: 10.1007/s00453-018-0472-z]

[29] Liu Y, Chen S. Sub-exponential time lower bounds for #VC and #matching on 3-regular graphs. In: Proc. of the 41st Int'l Symp. on Theoretical Aspects of Computer Science (STACS 2024). Clermont-Ferrand: STACS, 2024. 49: 1–49: 18. [doi: 10.4230/LIPIcs.STACS.2024.49]

[30] Chen HB, Curticapean R, Dell H. The exponential-time complexity of counting (quantum) graph homomorphisms. In: Proc. of the 45th Int'l Workshop on Graph-theoretic Concepts in Computer Science. Vall de Núria: Springer, 2019. 364–378. [doi: 10.1007/978-3-030-30786-8_28]

[31] Valiant LG. Holographic algorithms. SIAM Journal on Computing, 2008, 37(5): 1565–1594. [doi: 10.1137/070682575]

[32] Cai JY, Lu PY, Xia MJ. Holographic algorithms by Fibonacci gates and holographic reductions for hardness. In: Proc. of the 49th Annual IEEE Symp. on Foundations of Computer Science. Philadelphia: IEEE, 2008. 644–653. [doi: 10.1109/FOCS.2008.34]

[33] Guo H, Huang SX, Lu PY, Xia MJ. The complexity of weighted Boolean #CSP modulo k. In: Proc. of the 28th Int'l Symp. on Theoretical Aspects of Computer Science (STACS 2011). Dortmund: STACS, 2011. 249–260. [doi: 10.4230/LIPIcs.STACS.2011.249]

附中文参考文献:

[9] 邱国良, 张驰豪. 铁磁性双态自旋系统配分函数的可近似性. 计算机科学, 2020, 47(5): 22–26. [doi: 10.11896/jsjkx.200200119]

[10] 白宗磊, 王捍贫, 曹永知, 王璐璐. 树上自旋系统的快速采样算法. 计算机学报, 2022, 45(10): 2093–2116. [doi: 10.11897/SP.J.1016.2022.02093]

刘莹(1996—), 女, 博士生, CCF 学生会员, 主要研究领域为理论计算机科学, 计数复杂性.