中国科学:信息科学 2026 年 第 56 卷 第 5 期: 1154–1194
SCIENTIA SINICA Informationis
论文

《中国科学》杂志社
SCIENCE CHINA PRESS

一般约束满足问题解的采样洛瓦兹局部引理

何昆[1], 王淳扬[2*], 尹一通[2]

1. 中国人民大学数据工程与知识工程教育部重点实验室, 北京 100872
2. 南京大学计算机软件新技术国家重点实验室, 新基石科学实验室, 南京 210023
* 通信作者. E-mail: wcysai@smail.nju.edu.cn

收稿日期: 2025–04–18; 修回日期: 2025–07–08; 接受日期: 2025–11–28; 网络出版日期: 2026–04–29

国家自然科学基金 (批准号: 62472430) 资助项目

摘要 本文针对一般约束满足问题 (constraint satisfaction problems, CSP) 在类似洛瓦兹局部引理 (Lovász local lemma) 的条件下提供了一种快速的解均匀采样算法. 在给定的类局部引理条件下, 该采样算法即可在期望近线性的时间复杂度内输出一个几乎均匀分布的可满足赋值. 该结论是迄今首个针对度数不受限 (即最大度数不可被视为常数) 的一般约束满足问题的高效采样洛瓦兹局部引理. 在此之前, 类似的结论仅对于常数度数或"原子"约束 (即每个约束仅有少量局部违背配置) 的特殊约束满足问题成立. 此外, 本文建立的局部引理条件中的关键项同时改进了此前针对常数度数 CSP 的最好条件, 以及针对原子 CSP (包括 k- 合取范式 (k-conjunctive normal form satisfiability, k-SAT) 这一重要特殊情形) 的最好条件. 我们的采样方法不同于以往基于马尔可夫链 (Markov chains) 的在局部引理条件下的快速采样算法. 算法的关键步骤是一种递归的边缘分布采样算法, 这本身是一个具有独立研究价值的工具. 在局部引理条件下, 该边缘采样算法可以根据变量的边缘分布为其采样一个随机值, 且其代价与 CSP 的规模无关.

关键词 采样, 约束满足问题, 洛瓦兹局部引理

1 引言

约束满足问题 (constraint satisfaction problems, CSP) 是计算机科学中最基本的问题之一. 一个 CSP 是由一组定义在若干变量上的约束所描述的. 在形式上, 一个 CSP 实例被称为一个 CSP 公式, 记作 $\Phi = (V, \mathcal{Q}, \mathcal{C})$, 其中, V 是一个包含 $n = |V|$ 个变量的集合; $\mathcal{Q} \triangleq \bigotimes_{v\in V} Q_v$ 是所有变量赋值的笛卡儿积 (Cartesian product) 空间, 这里每个 Q_v 是一个有限的取值域, 大小为 $q_v \triangleq |Q_v| \geqslant 2$, 表示变量 v 的可能取值; $\mathcal{C}$ 是一组局部约束的集合, 每个约束 $c \in \mathcal{C}$ 是定义在变量子集 $\mathsf{vbl}(c) \subseteq V$ 上的函数, 形式为 $c : \bigotimes_{v\in\mathsf{vbl}(c)} Q_v \to \{\mathtt{True}, \mathtt{False}\}$.

若某个赋值 $\boldsymbol{x} \in \mathcal{Q}$ 满足所有约束, 即

$$\Phi(\boldsymbol{x}) \triangleq \bigwedge_{c\in\mathcal{C}} c\left(\boldsymbol{x}_{\mathsf{vbl}(c)}\right) = \mathtt{True},$$

引用格式: 何昆, 王淳扬, 尹一通. 一般约束满足问题解的采样洛瓦兹局部引理. 中国科学: 信息科学, 2026, 56: 1154–1194, doi: 10.1360/SSI-2025-0154
He K, Wang C Y, Yin Y T. Sampling Lovász local lemma for solutions to general constraint satisfaction problems. Sci Sin Inform, 2026, 56: 1154–1194, doi: 10.1360/SSI-2025-0154

则称该赋值 $\boldsymbol{x}$ 是 Φ 的一个解 (solution).

以下是 CSP 公式 $\Phi=(V,\mathcal{Q},\mathcal{C})$ 的几个关键参数:

- 值域大小 (domain size) $q=q_\Phi\triangleq\max_{v\in V}|Q_v|$ 以及宽度 (width) $k=k_\Phi\triangleq\max_{c\in\mathcal{C}}|\mathsf{vbl}(c)|$;
- 约束度数 (constraint degree) $\Delta=\Delta_\Phi\triangleq\max_{c\in\mathcal{C}}|\{c'\in\mathcal{C}\mid\mathsf{vbl}(c)\cap\mathsf{vbl}(c')\neq\emptyset\}|$[1);
- 违约概率 (violation probability) $p=p_\Phi\triangleq\max_{c\in\mathcal{C}}\mathcal{P}[\neg c]$, 其中 $\mathcal{P}$ 是对每个 $v\in V$ 从 Q_v 中独立均匀采样所得的概率分布.

著名的洛瓦兹局部引理 (Lovász local lemma, LLL) [1] 提供了 Φ 可满足性的一个充分条件. 具体而言, 如果 CSP 公式 Φ 满足以下条件, 则一定存在解

$$\mathrm{e}p\Delta\leqslant 1. \tag{1}$$

根据 Shearer [2] 给出的下界, 如果只知道 p 与 Δ, 则这个 "LLL 条件" 在可满足性判定方面是本质上紧的. 另外, 算法化 LLL (或称为构造性 LLL) 致力于高效地找出一个 CSP 的解. Moser 和 Tardos [3] 的算法是该方向的重大突破, 它在满足上述 LLL 条件的前提下, 能够高效地构造一个解.

采样洛瓦兹局部引理 (sampling LLL). 我们关注的问题是洛瓦兹局部引理的采样版本, 这是近年来受到广泛关注的研究方向 [4~15]. 在 CSP 的背景下, 该问题旨在设计一种高效的采样算法, 用于在类似 LLL 的参数范围内近似均匀地产生解. 这一问题与估计解空间体积或统计物理系统的配分函数密切相关, 并受到诸如图模型中的概率推理 [5] 和网络可靠性分析 [4,16,17] 等基本任务的驱动.

相比之下, 采样版的 LLL 问题比传统的算法化 LLL 更具计算挑战性, 因为后者只需要找到任意一个解, 而无需考虑所找的解是否服从均匀分布. 例如, 当用于采样时, Moser-Tardos 算法仅能在一些限制性的 CSP 类别中保证正确性 [4]. 根据文献 [15,18] 中的计算复杂性下界分析, 即使对于诸如 k-SAT 或超图染色等典型 CSP 子类, 也需要更强的 (类) LLL 条件 ($c\geqslant 2$) 才能实现高效采样:

$$p\Delta^c\lesssim 1, \tag{2}$$

其中符号 $\lesssim$ 忽略低阶项和常数因子.

在 Moitra [5] 的开创性工作中, 他提出了一种非常具有创新性的算法, 能够在 $p\Delta^{60}\lesssim 1$ 的条件下, 近似均匀地采样 k- 合取范式 (k-conjunctive normal form satisfiability, k-SAT) 公式的解. 该算法是基于确定性的近似计数方法, 通过在经过适当分解的公式上求解线性规划来实现, 其运行时间为 $n^{\mathrm{poly}(k,\Delta)}$. 此基于线性规划的方法后来被推广到超图染色问题 [6] 和随机 k-SAT 公式 [14], 最终 Jain 等 [11] 将其扩展到所有满足改进 LLL 条件 $p\Delta^7\lesssim 1$ 的 CSP. 不过, 这些基于确定性近似计数的算法普遍存在 $n^{\mathrm{poly}(k,\Delta)}$ 的时间复杂度, 因此只有在假设 k 和 Δ 都是常数, 即给定 CSP 实例是稀疏时才是高效的.

历史上, 快速混合 (rapid mixing) 的马尔可夫链 (Markov chains) 一直是经典的采样算法, 并且通常具有接近线性的时间效率. 然而, 在类洛瓦兹局部引理条件下进行采样时, 该方法曾经存在一个根本性的障碍: 尽管 CSP 的解在该条件下普遍存在, 但问题的解空间可能高度不连通, 从而使得使用局部转移 (local transition) 的马尔可夫链无法快速混合.

2020 年, Feng 等 [8] 在一项突破性工作中通过高效模拟的快速混合投影 (projected) 随机游走, 成功绕过了连通性障碍. 他们利用文献 [5] 中发明的标记/取消标记 (mark/unmark) 策略, 在部分适当选取变量的子集上运行马尔可夫链. 假设 (类) LLL 条件为 $p\Delta^{20}\lesssim 1$, 该新算法能够在时间复杂度为 $\mathrm{poly}(k,\Delta)\cdot n^{1.0001}$ 的范围内生成几乎均匀的 k-SAT 解, 实现了与变量数量 n 近乎线性的运行时间.

随后的研究观察到变量的标记/取消标记实际上是压缩变量状态的布尔特例. 基于这一观察, Feng 等 [9] 将马尔可夫链的快速采样方法推广到了满足 LLL 条件 $p\Delta^{350}\lesssim 1$ 的所有原子型 CSP 问题中. 之后 Jain 等 [10] 的一项工作中引入了一种名为 2- 树 (2-tree) 的组合结构, 并借此进行信息逾渗 (information percolation) 分析, 从

1) 需要注意的是, 约束度数 Δ 应当与依赖度数 (dependency degree) D 区分开, 后者指的是依赖图 (dependency graph) 的最大度数: $D\triangleq\max_{c\in\mathcal{C}}|\{c'\in\mathcal{C}\setminus\{c\}\mid\mathsf{vbl}(c)\cap\mathsf{vbl}(c')\neq\emptyset\}|$. 显然有 $\Delta=D+1$.

而显著改进了所需的 LLL 条件至 $p\Delta^{7.04} \lesssim 1$. 这一分析方法随后也被文献 [12] 采用于支持向过去耦合 (coupling from the past, CFTP) 技术, 实现了完美采样, 并进一步将条件优化为 $p\Delta^{5.71} \lesssim 1$. 然而, 所有这些已有的快速 LLL 采样算法仍然仅适用于原子型约束满足问题 (atomic CSP, 即单个约束仅会被常数数量的赋值所违反的 CSP, 例如, k-SAT 和 q 为常数时的超图 q 染色问题均属于原子型 CSP).

对于一般 CSP 的挑战. 对于既非原子型度数又不受限的一般 CSP, 实现高效采样仍然需要新的技术手段. 现有所有对于度数不受限 CSP 在 LLL 条件下的高效采样算法均依赖于将解空间投影至一个更小的空间, 以规避解空间连通性所带来的障碍. 为了高效地在投影空间中模拟随机游走, 并从随机图像中还原出一个随机解, 人们通常希望 CSP 公式在大多数情况下能够良好地分解 (factorize) 为若干小簇 —— 这是因为在当前图像下, 许多约束已被满足, 从而使得未被满足的部分呈现出局部化的结构. 这一性质在原子 CSP 中确实成立. 然而, 对于一般的非原子 CSP, 情况则可能不再如此. 由于一个违约事件 (即约束被违反) 在此情形下可能是非初等的, 其结构更为复杂, 因此在投影后难以被干净地消除, 从而破坏了分解结构的形成. 这一挑战显著限制了现有方法在非原子 CSP 上的适用性.

一般 CSP 的非原子性可能在采样 LLL 中相比在构造性 LLL 中更具有挑战性. 为了理解这一点, 注意到一般 CSP 可以通过原子型 CSP 来进行模拟: 将每个具有 N 个禁止赋值的约束 c, 替换为在相同变量集上每个禁止一个赋值的 N 个原子型约束. 这样的模拟会将约束度 Δ 最多增加 N 倍, 同时将违反概率 p 减少至最低 N 分之一. 对于经典的 LLL 条件 (1), p 和 Δ 是齐次的 (homogeneous), 这不会改变 LLL 条件; 然而, 式 (2) 中的采样 LLL 的条件会显著减小, 因为在该情况下, p 和 Δ 由于文献 [15,18] 中下界的存在而必然非齐次. 这种情况似乎表明, 一般 CSP 的非原子性可能会给 LLL 采样带来比存在性/构造性 LLL 更大的挑战. 事实上, 在本文之前, 对于一般 CSP, 尚不清楚在类似于上述 LLL 条件下, 采样问题是否是多项式时间可解的. 因此, 本文所考虑的开放问题如下:

是否存在对于一般 CSP 在 LLL 条件下的高效采样算法?

1.1 主要结论

本文对上述开放问题给出了确定的答案. 我们提出了一种与先前所有的基于马尔可夫链的快速采样方法不同的全新的算法, 首次实现了在 LLL 条件下对一般 CSP 解的高效采样, 并且我们所得出的 LLL 条件相较之前的工作也有所改进.

因为定义在超常数个变量上的任意约束函数可能非常复杂, 既难以表达又难以计算, 所以引入一个约束值计算的抽象化 (abstraction). 这种抽象化在之前算法化局部引理 [3,19] 的工作中也曾被引入. 具体而言, 假设存在以下满足性判定器 (evaluation oracle), 用于检查部分指定的赋值是否已经满足约束条件.

假设1 (满足性判定器) 对于 $\Phi = (V, \mathcal{Q}, \mathcal{C})$, 存在一个满足性判定器, 使得对于任何约束 $c \in \mathcal{C}$ 和任何在变量子集 $\Lambda \subseteq \mathsf{vbl}(c)$ 上指定的赋值 $\sigma \in \mathcal{Q}_\Lambda \triangleq \bigotimes_{v\in\Lambda} Q_v$, 该判定器会判定约束 c 是否已经被 σ 满足, 即对于所有满足 $\tau_\Lambda = \sigma_\Lambda$ 的 $\tau \in \mathcal{Q}_{\mathsf{vbl}(c)}$, 都有 $c(\tau) = \texttt{True}$.

对于特定类别的 CSP, 如 k-SAT 或超图染色问题, 这样的判定器很容易实现. 回顾之前定义的有关 CSP 公式 Φ 的参数 q, k, p 和 Δ. 假设存在这样的满足性判定器, 我们给出以下对于一般 CSP 公式在 LLL 条件内几乎均匀的高效采样算法.

定理1 (非正式版本) 存在一个算法, 输入任意 $\varepsilon \in (0, 1)$ 和任意符合以下条件的 n 个变量的 CSP 公式 $\Phi = (V, \mathcal{Q}, \mathcal{C})$:

$$k \cdot p \cdot q^2 \cdot \Delta^5 \leqslant \frac{1}{150\mathrm{e}^3}, \tag{3}$$

该算法在期望时间 $\mathrm{poly}(q, k, \Delta) \cdot n \log\left(\frac{n}{\varepsilon}\right)$ 内终止, 并且输出一个 Φ 的赋值, 其与真实的解均匀分布 μ_Φ 之间的全变差距离不超过 ε.

该定理的正式表述见定理 8 (关于终止性和采样的正确性) 和定理 11 (关于采样的效率). 当 $p \leqslant (qk)^{-\omega(1)}$ 时, 条件 (3) 变为 $p\Delta^{5+o(1)} \lesssim 1$. 之前针对一般 CSP 解的最佳采样条件是 $q^3kp\Delta^7 \lesssim 1$, 其由文献 [11] 中基于确

定性近似计数的算法获得, 且运行时间为 $(n/\varepsilon)^{\mathrm{poly}(k,\Delta,\log q)}$. 此处的条件也改善了本文之前的用于采样原子型的 CSP 最佳界限 $p\Delta^{5.714} \lesssim 1$, 其中包括著名的 k-SAT 问题 [10,12].

注释1　在本文的会议版本 [20,21] 发表之后, 2024 年 11 月, Wang 和 Yin [22] 将对于 k-SAT 的近似采样界限从本文的 $p\Delta^5 \lesssim 1$ 提升到了 $p\Delta^{4.82} \lesssim 1$. 然而, 文献 [22] 中的技术只能应用到原子型的 CSP 中, 而本文的侧重点则在于对于一般 CSP 的高效采样. 时至今日, 本文给出的结果仍然是唯一的对于一般 CSP 的高效采样算法.

令 Z 为 Φ 的解个数. 如果一个 $\hat{Z}$ 满足 $(1-\varepsilon)Z \leqslant \hat{Z} \leqslant (1+\varepsilon)Z$, 则称 $\hat{Z}$ 为 Z 的 ε- 近似. 通过常规地执行文献 [8] 中的非自适应退火 (nonadaptive annealing) 过程, 定理 1 中的近似采样算法可以作为黑盒 (black-box) 使用, 从而对于任意 $\varepsilon \in (0,1)$, 以高概率在时间 $\mathrm{poly}\,(q,k,\Delta) \cdot \tilde{O}\left(n^2\varepsilon^{-2}\right)$ 内给出 Z 的 ε- 近似.

1.1.1　完美采样 (perfect sampling) 算法

假设 1 中的满足性判定器实际上检查的是 $\mathcal{P}\left[\neg c \mid \sigma\right]$ 的符号, 其中 $\mathcal{P}\left[\neg c \mid \sigma\right]$ 指给定部分指定赋值 σ 的情况下, 约束 c 被违反的概率. 如果进一步可以高效地估计这个概率, 则定理 1 中的采样可以变得完美 (perfect), 即输出的解完全符合所有解的均匀分布.

定理2 (非正式版本)　对于一类输入的 CSP, 若存在这样的 FPTAS (fully polynomial-time approximation scheme) 来估计条件违反概率 (conditional violation probability):

- 对于任意约束 $c \in \mathcal{C}$, 任意在子集 $\Lambda \subseteq \mathsf{vbl}(c)$ 上指定的赋值 $\sigma \in \mathcal{Q}_\Lambda$, 以及任意 $0 < \varepsilon < 1$, 能够在时间 $\mathrm{poly}(q,k,1/\varepsilon)$ 内确定性地返回 $\mathcal{P}\left[\neg c \mid \sigma\right]$ 的 ε- 近似, 则定理 1 中的采样算法在式 (3) 的条件下, 于期望时间 $\mathrm{poly}(q,k,\Delta) \cdot n$ 内返回一个输入 CSP 的完美的均匀解.

定理的正式陈述见于定理 8 (采样算法的终止性和正确性) 和定理 10 (采样算法的计算效率). 首先构造一个完美采样算法. 随后, 通过蒙特卡罗 (Monte Carlo) 法实现了定理 2 中假设的 FPTAS, 从而引入了有限的误差, 并最终得到了定理 1 所描述的近似采样算法. 对于那些由简单局部约束定义的特定 CSP, 通常能够得到条件违反概率 $\mathcal{P}\left[\neg c \mid \sigma\right]$ 的闭合表达式, 从而使得计算变得十分简便. 在这种情况下, 可以在无需假设 1 和定理 2 中假设的条件违反概率的 FPTAS 的条件下, 直接构造出完美采样算法. 以下是两个允许线性时间的完美采样算法的一般 CSP 示例.

例1 (δ- 稳健 k-SAT)　n 个变量是布尔型的, 每个子句包含恰好 k 个文字, 且只有当至少 δk 个文字的值为 `True` 时, 子句才被满足.

- 对于 δ- 稳健 k-SAT, 其中变量度数为 d (每个变量至多出现在 d 个子句中), 并且满足

$$0 < \delta < \frac{1}{2}, \quad k \geqslant \frac{24\ln k + 20\ln d + 40}{(1-2\delta)^2},$$

我们的算法在期望时间 $\mathrm{poly}(k,d) \cdot n$ 内返回一个完美的均匀解.

例2 (δ- 稳健超图 q- 染色)　每个顶点都被染色为 q 种颜色之一, 每个超边都是 k- 均匀的, 当且仅当没有 $(1-\delta)k$ 个顶点有相同的颜色时, 超边被满足.

- 对于具有最大顶点度数 d 的 n 个顶点的 k- 均匀超图, 并且满足

$$(1-\delta)k \geqslant 15, \quad q \geqslant \frac{7d^{\frac{5}{(1-\delta)k-3}} \cdot 4^{\frac{1}{(1-\delta)}}}{(1-\delta)^{1.25}},$$

我们的算法在期望时间 $\mathrm{poly}(q,k,d) \cdot n$ 内返回一个完美的均匀合法染色.

1.1.2　边缘采样算法

我们的采样算法的核心组件是一个边缘采样算法 (marginal sampler). 令 $\mu = \mu_\Phi$ 表示 CSP 公式 Φ 上所有解的均匀分布, 对于每个 $v \in V$, 令 μ_v 表示由 μ 在变量 v 导出的边缘分布.

定理3 (非正式表述)　存在一种算法, 该算法以任意 $\varepsilon \in (0,1)$、满足条件 (3) 的 CSP 公式 $\Phi = (V, \mathcal{Q}, \mathcal{C})$ 以及任意 $v \in V$ 作为输入, 能够返回一个随机值 $x \in Q_v$, 其分布与 μ_v 的全变差距离不超过 ε. 该算法的期望运行时间为 $\mathrm{poly}(q, k, \Delta, \log(1/\varepsilon))$.

在与定理 2 相同的假设下, 该边缘采样算法也是完美的. 此外, 该边缘采样算法的另一个重要应用是概率推断算法, 如下所述.

定理4 (非正式表述)　存在一种算法, 该算法以任意 $\varepsilon, \delta \in (0,1)$、满足条件 (3) 的 CSP 公式 $\Phi = (V, \mathcal{Q}, \mathcal{C})$ 以及任意 $v \in V$ 作为输入, 能够在时间 $\mathrm{poly}(q, k, \Delta, 1/\varepsilon, \log(1/\delta))$ 内, 以至少 $1-\delta$ 的概率, 对每个 $x \in Q_v$ 计算 $\mu_v(x)$ 的 ε- 近似值.

上述两个定理将在定理 14 中被正式陈述和证明. 通过自归约 (self-reduction) 技术, 当对 Φ 施加一个可行的部分指定赋值 σ 时, 只要新的实例 Φ^σ 仍然满足 LLL 条件 (3), 那么定理 3 和 4 中的采样与推理算法仍然适用于条件边缘分布 μ_v^σ. 值得强调的是, 上述边缘采样与概率推断算法均为计算复杂度独立于总变量数 n 的局部算法. 在以往的研究中, 为了模拟或估计某个变量的边缘分布, 通常需要在 n 个变量上生成完整的赋值, 或至少付出不低于该级别的计算代价. 这一现象引发了一个自然的问题:

这些局部定义的采样或推理问题, 能否以局部的计算代价解决?

然而, 数十年来, 该问题一直未能得到令人满意的解答, 直到最近, 人们在无限自旋系统 (infinite spin system) 中发现了一种新的局部算法 [23], 而这也成为了我们研究的主要灵感来源.

1.2　技术概述

如前所述, 约束的非原子性为现有基于马尔可夫链的算法 (参见文献 [8~10, 12, 13]) 带来了障碍. 另外, 还有另一类采样算法, 我们称为 "基于重采样" (resampling-based) 算法 [4, 24~27]. 这些算法本质上类似于 Moser-Tardos 算法, 其通过对变量进行重采样来修正赋值, 使得最终得到的赋值服从正确的分布. 相比马尔可夫链算法, 基于重采样的方法对解空间连通性的依赖较小. 然而, 为了保证采样结果的正确性, 这类方法必须重采样已经被观察并参与条件化的变量. 这在处理非原子性约束时会引发新的挑战: 算法必须观察大量变量以确认约束是否满足, 但这些变量的重采样可能抵消算法的进展, 甚至使其陷入需要反复修正的无限循环.

为此, 引入了一种新的采样思想, 称为递归边缘采样算法. 这种算法在理念上更接近于基于重采样的方法, 而非传统的马尔可夫链方法. 它利用递归结构, 有效减少了重采样的次数, 从而提升了采样效率. 该算法的灵感来源于 Anand 和 Jerrum [23] 最近提出的一种在无限自旋系统中完美采样的新颖算法, 其中的核心组件就是这样一种能够根据边缘分布抽取自旋的采样算法.

考虑 CSP 公式 $\Phi = (V, \mathcal{Q}, \mathcal{C})$ 上所有解的均匀分布 μ 及其在某个变量 $v \in V$ 上的边缘分布 μ_v (设该变量的取值域为 $Q_v = [q]$).

为了从 μ_v (定义在 $[q]$ 上) 中采样, 一个直观的思路是利用所谓的 "局部均匀性" 性质 (见第 4.1 小节). 该性质基本上说明, 当 Φ 满足某些 LLL 条件时, μ_v 在全变差距离意义下与 $[q]$ 上的均匀分布不会相差太远. 因此, 从 $[q]$ 中均匀采样已经可以得到 μ_v 的一个粗略样本, 而接下来需要做的则是将这种粗糙的采样算法提升到任意精度.

由局部均匀性可知, 存在一个 $\theta < \frac{1}{q}$ 且足够接近 $\frac{1}{q}$, 使得

$$\forall\, x \in [q], \quad \mu_v(x) \geqslant \theta. \tag{4}$$

于是, 边缘分布 μ_v 可分解为

$$q\theta \cdot \mathcal{U} + (1 - q\theta) \cdot \mathcal{D},$$

其中 $\mathcal{U}$ 表示 $[q]$ 上的均匀分布, 而 $\mathcal{D}$ 表示 "溢出" 质量的分布, 其定义为

$$\forall\, x \in [q], \quad \mathcal{D}(x) = \frac{\mu_v(x) - \theta}{1 - q\theta}.$$

由此, 从 μ_v 中采样可以采用以下策略:

- 以概率 $q\theta$, 算法落入局部均匀性区域 (zone of local uniformity), 直接返回一个从 $\mathcal{U}$ 中均匀抽取的样本;
- 以概率 $1-q\theta$, 算法则落入不确定性区域 (zone of indecision), 需要从溢出分布 $\mathcal{D}$ 中抽取样本, 这可以通过构造一个伯努利工厂 (Bernoulli factory) 并利用 μ_v 作为神谕 (oracle) 来实现.

但问题来了, 如果一开始就有了 μ_v 的神谕, 那么为什么从 μ_v 中采样会成为一个问题呢?

上述 "先有鸡还是先有蛋" 的悖论在某种程度上可以通过一个简单的观察得到解决: 如果足够多的其他变量已经被正确采样, 例如采样结果为 X, 那么在满足足够强的 LLL 条件下, 很可能得到的公式 Φ^X 被分解为若干小簇 (cluster), 从而可以利用标准的拒绝采样 (rejection sampling) 方法在 Φ^X 上高效地从 μ_v^X 中采样, 进而整体上从 μ_v 中采样.

因此, 可以对采样策略进行如下修正: 当算法进入 "不确定性区域" 时, 在尝试从溢出分布 $\mathcal{D}$ 中采样之前, 先选择另一个变量 u, 其成功采样可能有助于将公式 Φ 分解为若干小簇. 随后, 算法在变量 u 上递归地调用边缘采样算法, 优先从 u 当前的边缘分布中进行采样.

目前唯一尚未解决的问题在于: LLL 条件并非可自规约的, 即在对某些变量进行固定 (pinning) 后, 该条件可能不再成立. 为了解决这一问题, 借鉴了文献 [11] 中 "冻结" 约束的思想, 用以指导算法如何选择下一个待采样的变量. 具体而言, 将原有的 LLL 条件替换为一个更为精细的不变量 (invariant) 条件, 它在整个算法执行过程中始终保持成立. 该条件一方面保证了每个被选中用于采样的变量都满足与式 (4) 中相同的局部均匀性; 另一方面也确保: 即使在无法再选出其他变量时, 公式仍有较高概率被成功分解.

为了证明递归采样算法的快速收敛性, 文献 [23] 的策略是展示其递归树所诱导的分支过程在给定的最坏边界条件下, 其后代 (offspring) 数期望呈衰减趋势, 然而在本文考虑的情形中, 这并不成立. 相反, 采用了一种平均情况 (average-case) 的分析方法, 并直接根据递归树对采样算法的期望代价进行界定. 更具体地, 我们使用了一种基于证据结构的论证方法 (witness argument). 其核心思路是, 如果递归采样算法的运行时间过长, 则必然出现某种特定的组合结构, 称为 "证据". 我们将证明, 这种证据结构出现的概率非常小, 并通过对于所有可能的证据结构进行并集界 (union bound), 可以推断出 —— 递归采样算法运行时间过长的概率也同样很小.

直接执行以上策略可以得到 $p\Delta^7 \lesssim 1$ 的 LLL 条件, 该条件来源于在上述过程中使用到的名为 $\{2,3\}$-树 ($\{2,3\}$-tree) 的证据结构. 为了获得定理 8 中的 $p\Delta^5 \lesssim 1$ 的改进界, 我们设计了一种称为广义 $\{2,3\}$- 树 (generalized $\{2,3\}$-tree) 的新的组合结构. 在大多数关于采样/计数 LLL 的工作中, 通常考虑两种类型的坏事件: 一种是坏事件是标记变量的赋值未能落入局部均匀区, 另一种是坏事件是当某个约束的大部分变量已被赋值后仍然未被满足 (参见文献 [5,6,9,11]). 在以往的工作中, 这两种坏事件通常被类似地处理, 并使 $\{2,3\}$- 树来加以界定 (参见文献 [28]). 我们得以做出改进的一个关键观察是, 这两类坏事件的密度不同, 这启发了我们设计这种新型组合结构, 以利用这一特性将界限改进至当时最佳.

1.3　文章结构

本文余下内容结构安排如下. 第 2 和 3 节介绍采样算法及相关定义. 第 2 节介绍与约束满足问题 (CSP) 相关的基本记号和定义; 第 3 节正式给出我们提出的递归采样算法. 第 4 和 5 节证明该递归采样算法的正确性. 第 4 节回顾洛瓦兹局部引理及局部均匀性等相关背景知识; 第 5 节在此基础上给出算法的正确性分析, 其中有关伯努利工厂的正确性分析在补充材料中给出. 第 6 和 7 节分析算法的运行效率. 第 6 节概述效率分析的整体框架; 第 7 节补充分析中涉及的广义 $\{2,3\}$- 树结构. 与广义 $\{2,3\}$- 树结构有关的部分技术细节以及有关伯努利工厂的效率分析在补充材料中给出. 第 8 节总结本文的主要贡献, 并提出若干后续研究中可能面临的开放问题.

2　CSP 的记号定义

2.1　CSP 的基本记号

回顾 CSP 公式 $\Phi = (V, \mathcal{Q}, \mathcal{C})$ 的定义. 用 $\Omega = \Omega_\Phi$ 表示 Φ 的所有解的集合, 用 $\mu = \mu_\Phi$ 表示 Ω 上的均匀

分布. 回顾 $\mathcal{P}$ 表示 $\mathcal{Q}$ 上的均匀乘积分布. 对于 $C \subseteq \mathcal{C}$, 记 $\mathsf{vbl}(C) \triangleq \bigcup_{c\in C} \mathsf{vbl}(c)$ 表示 C 中所有约束涉及的变量集合; 对于变量子集 $\Lambda \subseteq V$, 记 $\mathcal{Q}_\Lambda \triangleq \bigotimes_{v\in\Lambda} Q_v$ 表示变量集合 Λ 上的赋值空间. 接下来引入部分赋值 (partial assignments) 的记号.

定义1 (部分赋值)　给定一个 CSP 公式 $\Phi = (V, \mathcal{Q}, \mathcal{C})$, 定义

$$\mathcal{Q}^* \triangleq \bigotimes_{v\in V} (Q_v \cup \{\star, \text{☆}\}),$$

其中 $\star$ 和 ☆ 是两个不在任何 Q_v 中的特殊符号. 每个 $\sigma \in \mathcal{Q}^*$ 称为部分赋值.

在部分赋值 $\sigma \in \mathcal{Q}^*$ 中, 每个变量 $v \in V$ 的分类如下:

- $\sigma(v) \in Q_v$ 表示变量 v 被算法访问并赋值为 $\sigma(v) \in Q_v$;
- $\sigma(v) = \star$ 表示变量 v 被算法访问但尚未赋值为 Q_v 中的某个值;
- $\sigma(v) = \text{☆}$ 表示变量 v 未被算法访问, 因此未赋值为任何值.

此外, 使用 $\Lambda(\sigma) \subseteq V$ 和 $\Lambda^+(\sigma) \subseteq V$ 分别表示部分赋值 $\sigma \in \mathcal{Q}^*$ 中已赋值和已访问的变量集合, 即

$$\Lambda(\sigma) \triangleq \{v \in V \mid \sigma(v) \in Q_v\} \quad \text{和} \quad \Lambda^+(\sigma) \triangleq \{v \in V \mid \sigma(v) \neq \text{☆}\}. \tag{5}$$

给定任意部分赋值 $\sigma \in \mathcal{Q}^*$ 和变量 $v \in V$, 进一步用 $\sigma_{v\leftarrow x}$ 表示通过将 $\sigma(v)$ 替换为 $x \in Q_v \cup \{\star, \text{☆}\}$ 而得到的部分赋值. 一个部分赋值 $\tau \in \mathcal{Q}^*$ 被称为扩展 (extend) 另一个部分赋值 $\sigma \in \mathcal{Q}^*$, 当且仅当满足: $\Lambda(\sigma) \subseteq \Lambda(\tau)$, $\Lambda^+(\sigma) \subseteq \Lambda^+(\tau)$, 并且 σ 与 τ 在所有属于 $\Lambda(\sigma)$ 的变量上相同. 如果一个部分赋值 $\sigma \in \mathcal{Q}^*$ 使得所有扩展 σ 的完整赋值 $\tau \in \mathcal{Q}$ 都满足一个约束 $\in \mathcal{C}$, 则称 σ 满足 (satisfy) 约束 c. 如果一个部分赋值 $\sigma \in \mathcal{Q}^*$ 使得存在一个扩展了 σ 的解 $\tau \in \Omega$, 则称 σ 为可行 (feasible) 的. 给定任何可行的部分赋值 $\sigma \in \mathcal{Q}^*$ 和任意 $S \subseteq V$, 用 σ_S 表示 $\bigotimes_{v\in S} \sigma(v)$, 并用 μ_S^σ 表示在 σ 条件下由分布 μ 对 S 的边缘分布. 对于每个 $\tau \in \mathcal{Q}_S$, 有

$$\mu_S^\sigma(\tau) = \Pr_{X\sim\mu}[X_S = \tau \mid \forall v \in \Lambda(\sigma), X(v) = \sigma(v)].$$

进一步简化记号写作 $\mu_v^\sigma = \mu_v^\sigma$. 对于均匀乘积分布 $\mathcal{P}$, 也使用类似的符号. 对于 $\sigma \in \mathcal{Q}^*$ 和任意事件 $A \subseteq \mathcal{Q}$, 有

$$\mathcal{P}[A \mid \sigma] = \Pr_{X\in\mathcal{Q}}[X \in A \mid \forall v \in \Lambda(\sigma), X(v) = \sigma(v)].$$

2.2　关联超图与依赖图

CSP 公式的结构可以由关联超图与依赖图所定义. 在正式定义这些图前, 先定义一些和图论相关的基本记号.

定义2 (图论基本记号)　给定一个 (超) 图 $H = (V, \mathcal{E})$, 定义以下记号.

- 用 $\mathrm{dist}_H(\cdot,\cdot)$ 表示 H 中的最短路径距离.
- 对任意 $S \subseteq V$, 用 $H[S]$ 表示 H 在 S 上的导出子图 (induced subgraph).
- 用 $\mathrm{Lin}(H)$ 表示 H 的线图 (line graph). $\mathrm{Lin}(H)$ 是一个简单无向图, 其顶点集合是 H 中的 (超) 边集合 $\mathcal{E}$, 任意两条 (超) 边 $u, v \in \mathcal{E}$ 在 $\mathrm{Lin}(H)$ 中相邻当且仅当 $u \cap v \neq \emptyset$.
- 用 $H^2 = (V, E)$ 表示 H 的平方图 (square graph). H^2 是一个简单无向图, 其顶点集合是 H 中的顶点集合 V, 任意两个顶点 $u, v \in V$ 在 H^2 中相邻当且仅当 $\mathrm{dist}_H(u, v) \leqslant 2$.

接下来给出联超图与依赖图的具体定义.

定义3 (关联超图与依赖图)　给定一个 CSP 公式 $\Phi = (V, \mathcal{Q}, \mathcal{C})$.

- 定义它的关联超图 (incidence hypergraph) $H_\Phi = (V, \mathcal{E})$ 如下: 其顶点集是 Φ 中的变量集合, 超边的集合为 $\mathcal{E} = \{\mathsf{vbl}(c) \mid c \in \mathcal{C}\}$. 注意 H_Φ 中可以存在重边.
- 定义它的依赖图 (dependency graph) $G_\Phi = (\mathcal{C}, E)$ 如下: 其顶点集是 Φ 中约束集合, 并且当且仅当对于两个不同的 $c, c' \in \mathcal{E}$, 当 $\mathsf{vbl}(c) \cap \mathsf{vbl}(c') \neq \emptyset$ 时, 在 c 与 c' 之间连接一条边.

对比依赖图 G_Φ 的定义以关联超图 H_Φ 的定义, 可以发现 $G_\Phi = \mathrm{Lin}(H_\Phi)$, 即依赖图是关联超图的线图.

3　边缘递归采样算法

接下来给出用于几乎均匀采样 CSP 公式的解的主要算法, 其中使用了第 2 节定义的符号.

3.1　主采样算法

我们的主采样算法以 CSP 公式 $\Phi=(V,\mathcal{Q},\mathcal{C})$ 为输入, 其中包括值域大小 $q=q_\Phi$、宽度 $k=k_\Phi$、约束度数 $\Delta=\Delta_\Phi$ 和违反概率 $p=p_\Phi$, 这些参数的涵义已在第 1 节中定义. 假设 $n=|V|$ 个变量按照任意顺序枚举为 $V=\{v_1,v_2,\ldots,v_n\}$, 且可通过假设 1 中的满足性判定器访问输入的 CSP 公式 $\Phi=(V,\mathcal{Q},\mathcal{C})$. 同时, 假设给定任何约束 $c\in\mathcal{C}$ (或变量 $v\in V$), 可以检索到其包含的变量 $\mathsf{vbl}(c)$ (或包含其的所有约束 $\{c\in\mathcal{C}\mid v\in\mathsf{vbl}(c)\}$). 主采样算法 (算法 1) 与文献 [6,11] 中的主采样框架相同. 在主采样算法中, 我们维护一个被初始化为空赋值 $\star^V$ 的部分赋值 $X\in\mathcal{Q}^*$.

(1) 在第一阶段, 每一步算法会自适应地按预定顺序选择一个具有足够 "自由度" 的变量 v, 因为在当前赋值 X 下, 该变量不涉及任何容易违反的约束. 然后, 通过子过程 MarginSample, 根据正确的边缘分布 μ_v^X 采样出 $X(v)$ 的值.

(2) 当不再存在具有足够自由度的变量时, 原 CSP 公式被认为已经 "分解" 成为一系列小的子公式, 算法进入第二阶段. 在这一阶段, 通过标准的拒绝采样 (rejection sampling) 子过程, 将第一阶段构建的部分赋值完成为一个均匀随机的解.

设定一个关键的阈值 α 用于表示违反概率, 其定义如下:

$$\alpha=\left(18\mathrm{e}^2q^2k\Delta^2\right)^{-1}, \tag{6}$$

在假设满足式 (3) 中 LLL 条件的情况下, 有 $\alpha>p=p_\Phi$.

为了便于阐述, 假设存在如下的冻结判定器 (frozen oracle), 能够根据当前的部分赋值近似判断一个约束是否变得容易违反.

假设2 (冻结判定器)　存在一个冻结判定器, 使得对于任何部分赋值 $\sigma\in\mathcal{Q}^*$ 和任何约束 $c\in\mathcal{C}$, 该判定器能够区分以下两种情况: $\mathcal{P}[\neg c\mid\sigma]>\alpha$ 和 $\mathcal{P}[\neg c\mid\sigma]<0.99\alpha$, 在此之外的情况则可以任意且一致地 (arbitrarily and consistently) 给出答案, 即在未定义的 $\mathcal{P}[\neg c\mid\sigma]\in[0.99\alpha,\alpha]$ 情况下可以回答为 "是" 或 "否", 但对于相同的 $\sigma_{\mathsf{vbl}(c)}$, 答案始终相同.

显然, 这样的冻结判定器被定理 2 中假设的对 $\mathcal{P}[\neg c\mid\sigma]>\alpha$ 的 FPTAS 所蕴含. 该判定器将在第 6 节中被明确实现. 现在, 基于这样的冻结判定器, 容易违反的约束及其涉及的变量的类定义如下.

定义4 (冻结 (frozen) 和固定 (fixed))　默认假设 2 中的条件. 设 $\sigma\in\mathcal{Q}^*$ 为一个部分赋值.

• 如果约束 $c\in\mathcal{C}$ 被冻结判定器报告为 $\mathcal{P}[\neg c\mid\sigma]>\alpha$, 则称该约束为 σ- 冻结约束. 记 $\mathcal{C}^\sigma_{\mathsf{frozen}}$ 为所有 σ- 冻结约束的集合:

$$\mathcal{C}^\sigma_{\mathsf{frozen}}\triangleq\left\{c\in\mathcal{C}\;\middle|\;\text{约束 } c \text{ 被冻结判定器报告满足 } \mathcal{P}[\neg c\mid\sigma]>\alpha\right\}.$$

• 如果变量 $v\in V$ 在部分赋值 σ 中被访问, 或者 v 在某个 σ- 冻结约束之中, 则称该变量为 σ- 固定变量. 记 V^σ_{fix} 为所有 σ- 固定变量的集合:

$$V^\sigma_{\mathsf{fix}}\triangleq\Lambda^+(\sigma)\cup\bigcup_{c\in\mathcal{C}^\sigma_{\mathsf{frozen}}}\mathsf{vbl}(c).$$

类似的冻结思想也出现在之前关于采样和算法化 LLL 的研究中 (参见文献 [11,29]).

注释2 (冻结/固定决策的单边误差)　根据假设 2 中冻结判定器的性质, 任何满足 $\mathcal{P}[\neg c\mid\sigma]>\alpha$ 的约束 $c\in\mathcal{C}$, 必定属于 $\mathcal{C}^\sigma_{\mathsf{frozen}}$, 且任何涉及该约束的变量 $v\in V$ 必定属于 V^σ_{fix}; 反之, 任何 σ- 冻结约束 $c\in\mathcal{C}^\sigma_{\mathsf{frozen}}$, 必定满足 $\mathcal{P}[\neg c\mid\sigma]\geqslant0.99\alpha$, 且任何未被访问的 σ- 固定变量 $v\in V^\sigma_{\mathsf{fix}}$, 必定在至少一个此类约束中.

算法 1: 主采样算法.

```
输入: 一个 CSP 公式 Φ = (V, 𝒬, 𝒞);
输出: 一个赋值 X ∈ 𝒬;
X ← ☆^V;
for i = 1 to n do
    if v_i 不是 σ-固定的 then
        X(v_i) ← MarginSample(Φ, X, v_i);
X_{V\Λ(X)} ← RejectionSampling(Φ, X, V \ Λ(X));
return X.
```

以下不变量在算法 1 的 for 循环中得到满足 (在引理 2 中正式证明). MarginSample 子程序的正确性由该不变量保证.

条件1 (对于 MarginSample 的不变量) 以下条件适用于输入元组 (Φ,σ,v):

- $\Phi=(V,\mathcal{Q},\mathcal{C})$ 是一个 CSP 公式, $\sigma\in\mathcal{Q}^*$ 是一个可行的部分赋值, $v\in V$ 是一个变量;
- v 不是 σ- 固定的, 且 $\sigma(v)=☆$, 并且对于所有 $u\in V$ 有 $\sigma(u)\in Q_u\cup\{☆\}$;
- 对于所有的 $c\in\mathcal{C}$, 有 $\mathcal{P}[\neg c\mid\sigma]\leqslant\alpha q$.

算法 1 的正确性源自 MarginSample 和 RejectionSampling 子程序的正确性, 它们从正确的边缘分布中进行采样, 这在定理 8 中得到了正式证明. 事实上, 算法 1 中的采样是完美的. 只有在通过蒙特卡罗法实现了假设 2 中的冻结判定器后, 采样才会变得近似.

3.2 拒绝采样子过程

接下来介绍拒绝采样子过程 RejectionSampling. 我们的拒绝采样方法利用了 CSP 公式的固定和分解. 我们在此给出这一部分的定义. 在部分赋值 $\sigma\in\mathcal{Q}^*$ 下, CSP 公式 $\Phi=(V,\mathcal{Q},\mathcal{C})$ 的固定公式记为 $\Phi^\sigma=(V^\sigma,\mathcal{Q}^\sigma,\mathcal{C}^\sigma)$, 它是一个新的 CSP 公式, 满足以下条件.

(1) $V^\sigma=V\setminus\Lambda(\sigma)$, 即去除在部分赋值 σ 下已经被访问过的变量.

(2) $\mathcal{Q}^\sigma=\mathcal{Q}_{V\setminus\Lambda(\sigma)}$, 即去除已访问变量的域.

(3) $\mathcal{C}^\sigma$ 由以下方式从 $\mathcal{C}$ 获得:

- 删除所有已被部分赋值 σ 满足的约束,
- 对于余下的约束, 将 $\Lambda(\sigma)$ 中的变量替换为它们的赋值 $\sigma(v)$.

显然, 对于公式 Φ^σ 的均匀分布 μ_{Φ^σ}, 有 $\mu_{\Phi^\sigma}=\mu^\sigma_{V\setminus\Lambda(\sigma)}$, 即 Φ^σ 的解的均匀分布等于通过部分赋值 σ 对 $V\setminus\Lambda(\sigma)$ 的均匀分布.

回顾定义 3 中的定义. CSP 公式 $\Phi=(V,\mathcal{Q},\mathcal{C})$ 可以表示为一个 (多重) 关联超图 H_Φ, 其中每个变量 $v\in V$ 对应于 H_Φ 中的一个顶点, 每个约束 $c\in\mathcal{C}$ 对应于 H_Φ 中的一个超边 $\mathsf{vbl}(c)$. 我们稍微滥用符号, 写作 $H_\Phi=(V,\mathcal{C})$.

设 $H_i=(V_i,\mathcal{C}_i)$ $(1\leqslant i\leqslant K)$ 表示 H_Φ 中的所有 $K\geqslant 1$ 个连通分量, 而 $\Phi_i=(V_i,\mathcal{Q}_{V_i},\mathcal{C}_i)$ 是它们对应的公式. 显然, $\Phi=\Phi_1\wedge\Phi_2\wedge\cdots\wedge\Phi_K$, 并且 μ_Φ 是所有 μ_{Φ_i} 的乘积. 同时, $S\subseteq V$ 上的均匀分布 μ_S 仅由那些与 S 相交的分量中的 Φ_i 决定.

对于每个 $\sigma\in\mathcal{Q}^*$ 和 $v\in V^\sigma$, 令 $H^\sigma_v=(V^\sigma_v,\mathcal{C}^\sigma_v)$ 表示在 H^σ 中包含顶点/变量 v 的连通分量. 这一定义将在后续中被使用.

用于从边缘分布 μ^σ_S 中采样的拒绝采样算法见算法 2. 该算法的正确性是已知的常识, 这里略去证明.

定理5 对于任何输入 (Φ,σ,S), 如算法 2 所述, RejectionSampling 以概率 1 终止, 并且在终止时, 它返回一个分布为 μ^σ_S 的部分赋值 $X_S\in\mathcal{Q}_S$.

算法 2: RejectionSampling(Φ, σ, S).

输入: 一个 CSP 公式 $\Phi = (V, \mathcal{Q}, \mathcal{C})$, 一个可行部分赋值 $\sigma \in \mathcal{Q}^*$, 以及一个 σ 中未被赋值的子集 $\subseteq V \setminus \Lambda(\sigma)$;
输出: 一个分布为 μ_S^σ 部分赋值 $X_S \in \mathcal{Q}_S$;

```
找出 H_{Φ^σ} 中使得 V_i^σ 与 S 相交的所有连通分量 {H_i^σ = (V_i^σ, C_i^σ) | 1 ⩽ i ⩽ K}, 其中 Φ^σ 表示 Φ 在部分赋值 σ 下的简化;
for 1 ⩽ i ⩽ K do
    repeat
        均匀随机生成 X_{V_i^σ} ∈ Q_{V_i^σ};
    until 所有 C_i^σ 的约束都被 X_{V_i^σ} 满足;
return X_S, 其中 X 是所有 X_{V_i^σ} 的组合 (concatenation).
```

3.3　边缘采样算法

现在介绍我们采样算法的核心部分 ——MarginSample 子程序. 这个过程是一个 "边缘采样算法": 它可以根据变量 $v \in V$ 的边缘分布 μ_v^σ 为其抽取一个随机值. 我们的边缘采样算法受到 Anand 和 Jerrum [23] 最近提出的用于无限自旋系统的采样算法的启发. 对于每个变量 $v \in V$, 假设 Q_v 中的所有值有一个任意的全序 (total order); 使用 $q_v \triangleq |Q_v|$ 来表示变量 v 的域大小, 并固定以下参数:

$$\theta_v \triangleq \frac{1}{q_v} - \eta - \zeta, \quad \theta \triangleq \frac{1}{q} - \eta - \zeta, \quad \text{其中} \begin{cases} \eta = (1 - \mathrm{e}\alpha q)^{-\Delta} - 1, \\ \zeta = (16\mathrm{e}qk\Delta)^{-1}. \end{cases} \tag{7}$$

注意, 由于式 (3) 中的 LLL 条件保证了 $\zeta < \frac{1}{q} - \eta$, 因此 $\theta_v, \theta > 0$. 从边缘分布 μ_v^σ 进行采样的 MarginSample 子程序在算法 3 中给出.

算法 3: MarginSample(Φ, σ, v).

输入: 一个 CSP 公式 $\Phi = (V, \mathcal{Q}, \mathcal{C})$, 一个可行部分赋值 $\sigma \in \mathcal{Q}^*$, 以及一个变量 $v \in V$;
输出: 一个分布为 μ_v^σ 的随机值 $x \in Q_v$;

```
均匀随机地采样实数 r ∈ [0, 1);
if r < q_v · θ_v then                              // r 落入局部均匀性区间
    return Q_v 中的第 ⌈r/θ_v⌉ 个值;
else                                               // r 落入不确定性区间
    return MarginOverflow(Φ, σ_{v←⋆}, v).
```

算法 3 中满足的一个不变量保证了 $\theta_v + \zeta$ 始终是边缘概率的一个下界. 这个下界性质会在本节末尾通过 "局部均匀性" 性质 (推论 1) 得到正式证明.

命题1　假设输入 (Φ, σ, v) 满足条件 1, 则有 $\min_{x \in Q_v} \mu_v^\sigma(x) \geqslant \theta_v + \zeta$.

因此, 下面构造的 "溢出分布" $\mathcal{D}$ 是良定义的 Q_v 上的分布:

$$\forall\, x \in Q_v, \quad \mathcal{D}(x) \triangleq \frac{\mu_v^\sigma(x) - \theta_v}{1 - q_v \cdot \theta_v}. \tag{8}$$

考虑以下的思想实验. 将区间 $[0, 1)$ 划分为 $q_v + 1$ 个子区间 $I_1, I_2, \ldots, I_{q_v}$ 和 I', 其中对于 $1 \leqslant i \leqslant q_v$, $I_i = [(i-1)\theta_v, i\theta_v)$ 是大小为 θ_v 的等长区间; 而 $I' = [q_v \cdot \theta_v, 1)$ 则是剩余部分. 我们将 $\bigcup_{i=1}^{q_v} I_i = [0, q_v \cdot \theta_v)$ 称为 "局部均匀性区间", 将 $I' = [q_v \cdot \theta_v, 1)$ 称为 "不确定性区间". 然后, 从 μ_v^σ 中采样可以通过以下方式模拟. 首先采样一个均匀随机数 $r \in [0, 1)$, 如果 $r < q_v \cdot \theta_v$, 即它落入 "局部均匀性区间", 那么如果 $r \in I_i$, 则返回 Q_v 中的第 i 个值; 否则, 如果 $r \in I'$, 即它落入 "不确定性区间", 则从上述的分布 $\mathcal{D}$ 中抽取一个随机值. 容易验证, 生成的样本是按照 μ_v^σ 分布的. 在假设子程序 MarginOverflow($\Phi, \sigma_{v \leftarrow \star}, v$) 能正确地从 $\mathcal{D}$ 中采样的前提下, 这正是算法 3 所执行的操作.

3.4　从边缘溢出分布中递归采样

MarginOverflow 子程序的目的是从分布 $\mathcal{D}$ 中采样, 而该分布是根据边缘分布 μ_v^σ (如式 (8) 所定义的) 计算得出的.

假设可以访问一个从 μ_v^σ 中采样的神谕 (这种神谕可以通过算法 2 中的 RejectionSampling(Φ, σ, v) 实现). 然后, 从 μ_v^σ 的线性函数 $\mathcal{D}$ 中采样, 可以通过现有的伯努利工厂方法 (Bernoulli factory method) [30~32] 来解决.

这初看起来有些荒谬, 因为如果 μ_v^σ 的这样的神谕是高效的, 我们可以直接使用它来输出一个 μ_v^σ 的样本, 而这正是我们尝试从 $\mathcal{D}$ 中采样的根本原因. 然而, 这种用于从 $\mathcal{D}$ 中采样的伯努利工厂方法可以作为递归的基础: 当在递归过程中, 具有足够 "自由度" 的变量将在其局部均匀性区域内成功采样, 并使得剩余的 CSP 公式被分解为一系列小的公式时, 通过 RejectionSampling(Φ, σ, v) 实现的 μ_v^σ 神谕将变得高效, 并且可以应用伯努利工厂方法来从 $\mathcal{D}$ 中采样.

我们定义了一类变量, 使得它们在递归过程中可以优先被进行采样.

定义5 ($\star$- 影响变量)　令 $\sigma \in \mathcal{Q}^*$ 是一个部分赋值. 令 $H^\sigma = H_{\Phi^\sigma} = (V^\sigma, \mathcal{C}^\sigma)$ 为简化公式 Φ^σ 的超图. 设 H_{fix}^σ 为 H^σ 中由 $V^\sigma \cap V_{\mathsf{fix}}^\sigma$ 所导出的子超图.

- 令 $V_{\star\text{-con}}^\sigma \subseteq V^\sigma \cap V_{\mathsf{fix}}^\sigma$ 为属于 H_{fix}^σ 中包含任何 v 使得 $\sigma(v) = \star$ 的连通分量的顶点集合.
- 令 $V_{\star\text{-inf}}^\sigma \triangleq \{u \in V^\sigma \setminus V_{\star\text{-con}}^\sigma \mid \exists c \in \mathcal{C}^\sigma, v \in V_\star^\sigma : u, v \in \mathsf{vbl}(c)\}$ 为 $V_{\star\text{-con}}^\sigma$ 在 H^σ 中的顶点边界.
- 令 $\mathcal{C}_{\star\text{-con}}^\sigma$ 为 $\mathcal{C}$ 中与 $V_{\star\text{-con}}^\sigma$ 相交的约束集合, 即 $\mathcal{C}_{\star\text{-con}}^\sigma \triangleq \{c \in \mathcal{C} \mid \mathsf{vbl}(c) \cap V_{\star\text{-con}}^\sigma \neq \emptyset\}$.
- 定义 $\mathsf{NextVar}(\sigma)$ 为

$$\mathsf{NextVar}(\sigma) \triangleq \begin{cases} \text{满足 } v_i \in V_{\star\text{-inf}}^\sigma \text{ 中有最小的 } i \text{ 的 } v_i, & \text{如果 } V_{\star\text{-inf}}^\sigma \neq \emptyset, \\ \perp, & \text{否则}. \end{cases} \tag{9}$$

注释3　上述定义的 $\mathcal{C}_{\star\text{-con}}^\sigma$ 在此处未被使用, 但在之后的分析中具有重要意义. 与定义 4 中的一样, V_{fix}^σ 是相对于假设 2 中的冻结判定器定义的.

在之后的第 6.5.1 小节中, 我们给出了一种动态数据结构, 用于高效计算 $\mathsf{NextVar}(\sigma)$.

通过定义 5 中 $\mathsf{NextVar}(\cdot)$ 的构造, 接下来算法 4 给出 MarginOverflow 子程序的描述.

算法 4: MarginOverflow(Φ, σ, v).

输入: 一个 CSP 公式 $\Phi = (V, \mathcal{Q}, \mathcal{C})$, 一个可行部分赋值 $\sigma \in \mathcal{Q}^*$, 以及一个变量 $v \in V$;

输出: 一个分布为 $\mathcal{D} \triangleq \frac{1}{(1-q_v \cdot \theta_v)}(\mu_v^\sigma - \theta_v)$ 的随机值 $x \in Q_v$;

1 $u \leftarrow \mathsf{NextVar}(\sigma)$, 其中 $\mathsf{NextVar}(\sigma)$ 的定义在式 (9) 中给出;

2 **if** $u \neq \perp$ **then**

3 　均匀随机地采样实数 $r \in [0, 1)$;

4 　**if** $r < q_u \cdot \theta_u$ **then** 　　// r 落入局部均匀性区间

5 　　$\sigma(u) \leftarrow Q_u$ 中的第 $\lceil r/\theta_u \rceil$ 个值;

6 　**else** 　　// r 落入不确定性区间

7 　　$\sigma(u) \leftarrow$ MarginOverflow$(\Phi, \sigma_{u \leftarrow \star}, u)$;

　/* 第 4~7 行从 $\mu_u^{\sigma_{u\leftarrow\star}}$ 分布中采样 $\sigma(u)$ 的值; */

8 　**return** MarginOverflow(Φ, σ, v);

9 **else** 　　// 所有非 σ- 固定的变量都和 v 以及 (递归中的) 祖先不连通.

10 　使用补充材料中的访问 RejectionSampling$(\Phi, \sigma, \{v\})$ 神谕的伯努利工厂方法, 来按照分布 $\mathcal{D} \triangleq \frac{1}{(1-q_v \cdot \theta_v)}(\mu_v^\sigma - \theta_v)$ 采样一个随机值 $x \in Q_v$;

　return x.

基本上, 如果一个变量 u 目前有足够的 "自由度" (即 u 不是 σ- 固定的), 并且能够通过当前 σ 下的约束链影响我们在递归中尝试采样的变量 (这些变量通过 $\star$ 标记), 则该变量 u 是一个合适的采样候选变量. 这样的变量由 $\mathsf{NextVar}(\sigma)$ 枚举.

算法 4 的思想很简单. 为了从一个变量 $v \in V$ 的溢出分布 $\mathcal{D}$ 中进行采样: 如果存在另一个候选变量 $u = \mathsf{NextVar}(\sigma)$, 它仍然具有足够的自由度, 则其采样可能是简单的, 并且与在 v 或其祖先处的采样相关, 首先尝试从 u 的边缘值中进行采样 (希望在其局部均匀性区域内, 并通过递归调用来补偿其边缘溢出的影响); 如果没有这样的候选变量可以首先进行采样, 最终将使用伯努利工厂从 $\mathcal{D}$ 中进行采样.

以下不变量由 MarginOverflow 子程序在 MarginSample 子程序中调用时以及 MarginOverflow 本身满足 (在引理 2 中已证明).

条件2 (MarginOverflow 的不变量) 对于输入元组 (Φ, σ, v), 有以下条件:

- $\Phi = (V, \mathcal{Q}, \mathcal{C})$ 是一个 CSP 公式, $\sigma \in \mathcal{Q}^*$ 是一个可行的部分赋值, $v \in V$ 是一个变量;
- $\sigma(v) = \star$;
- 对于所有的 $c \in \mathcal{C}$, 有 $\mathcal{P}[\neg c \mid \sigma] \leqslant \alpha q$.

以下边缘概率下界通过 "局部均匀性" 性质 (推论 1) 得到, 与命题 1 中的推导方式相同, 并在本节最后正式证明.

命题2 假设对于输入 (Φ, σ, v) 满足条件 2, 则有 $\min_{x \in Q_v} \mu_v^\sigma(x) \geqslant \theta_v + \zeta$. 对于 $u = \mathsf{NextVar}(\sigma)$, 如果 $u \neq \perp$, 则也有 $\min_{x \in Q_u} \mu_u^\sigma(x) \geqslant \theta_u + \zeta$.

假设可以访问用于从边缘分布 μ_v^σ 中采样的神谕, 在算法 4 中使用的伯努利工厂是通过现有构造的组合实现的 (具体而言, 是通过文献 [30] 中的减法伯努利工厂 (Bernoulli factory for subtraction) 结合文献 [31] 中的线性伯努利工厂 (linear Bernoulli factory) 和文献 [32] 中的伯努利竞赛 (Bernoulli race) 组合实现的). 该神谕由算法 2 中的 RejectionSampling(Φ, σ, v) 实现.

这一点通过以下引理形式化给出.

引理1 (伯努利工厂的正确性) 假设输入 (Φ, σ, v) 满足条件 2, 则存在一个伯努利工厂, 它访问 RejectionSampling(Φ, σ, v) 作为神谕, 且以概率 1 终止, 并且在终止时返回一个随机值 $x \in Q_v$, 其分布与式 (8) 中定义的 $\mathcal{D}$ 分布一致.

上述引理中伯努利工厂的构造是某种标准的构造, 其具体内容在补充材料中, 该补充材料证明了引理 1 并分析了伯努利工厂的效率.

4 洛瓦兹局部引理的预备知识

著名的洛瓦兹局部引理为 CSP 解的存在提供了一个充分条件.

定理6 [1] 给定一个 CSP 公式 $\Phi = (V, \mathcal{Q}, \mathcal{C})$, 如果它满足以下条件:

$$\exists\, x \in (0,1)^{\mathcal{C}} \text{ s.t. } \mathcal{P}[\neg c] \leqslant x(c) \prod_{\substack{c' \in \mathcal{C} \setminus \{c\} \\ vbl(c) \cap \mathsf{vbl}(c') \neq \emptyset}} (1 - x(c')), \quad \forall\, c \in \mathcal{C}, \tag{10}$$

则

$$\mathcal{P}[\mathcal{C}] \geqslant \prod_{c \in C} (1 - x(c)) > 0.$$

当条件 (10) 满足时, 均匀分布 μ 上的任意事件的概率可以通过乘积分布中该事件的概率来良好地近似.

定理7 (参见文献 [33, 定理 2.1]) 给定一个 CSP 公式 $\Phi = (V, \mathcal{Q}, \mathcal{C})$, 如果式 (10) 成立, 则对于任何由变量子集 $\mathsf{vbl}(\mathcal{A}) \subseteq V$ 上的赋值确定的事件 $\mathcal{A}$, 有

$$\mathcal{P}[\mathcal{A} \mid \mathcal{C}] \leqslant \mathcal{P}[\mathcal{A}] \prod_{\substack{c \in \mathcal{C} \\ \mathsf{vbl}(c) \cap \mathsf{vbl}(\mathcal{A}) \neq \emptyset}} (1 - x(c))^{-1}.$$

以下 “局部均匀性” (local uniformity) 性质是定理 7 的直接推论, 其通过对每个约束 $c \in \mathcal{C}$ 设置 $x(c) = \mathrm{e}p$ 得到 (下界通过 $\mu_v(x) = 1 - \sum_{y \in Q_v \setminus \{x\}} \mu_v(y)$ 计算得到).

推论1 (局部均匀性)　给定一个 CSP 公式 $\Phi = (V, \mathcal{Q}, \mathcal{C})$, 如果 $\mathrm{e}p\Delta < 1$, 则对于任意变量 $v \in V$ 和任意值 $x \in Q_v$, 有以下不等式:

$$\frac{1}{q_v} - \left((1-\mathrm{e}p)^{-\Delta} - 1\right) \leqslant \mu_v(x) \leqslant \frac{1}{q_v} + \left((1-\mathrm{e}p)^{-\Delta} - 1\right).$$

回顾式 (6) 中定义的 α 以及式 (7) 中定义的 θ_v, ζ 和 η. 最后, 给出以下由 “局部均匀性” 性质 (推论 1) 推导得出的推论, 其同时证明了命题 1 和 2.

推论2　对于任意 CSP 公式 $\Phi = (V, \mathcal{Q}, \mathcal{C})$ 和任意部分赋值 $\sigma \in \mathcal{Q}^*$, 如果

$$\mathcal{P}\left[\neg c \mid \sigma\right] \leqslant \alpha q, \quad \forall\, c \in \mathcal{C},$$

则 σ 是可行的, 并且对于任意 $v \in V \setminus \Lambda(\sigma)$ 和任意 $x \in Q_v$, 有

$$\theta_v + \zeta \leqslant \mu_v^\sigma(x) \leqslant \theta_v + 2\eta + \zeta.$$

证明　令 $\Phi^\sigma = (V^\sigma, \mathcal{Q}^\sigma, \mathcal{C}^\sigma)$ 为在部分赋值 σ 下简化后的公式, 其中 $V^\sigma = V \setminus \Lambda(\sigma)$. 我们有

$$\mathcal{P}_{\Phi^\sigma}[\neg c] = \mathcal{P}_\Phi[\neg c | \sigma] \leqslant \alpha q, \quad \forall\, c \in \mathcal{C},$$

这意味着简化后的实例 Φ^σ 的违反概率为 $p_{\Phi^\sigma} \leqslant \alpha q$. 根据在式 (6) 中选择的 α, 仍然有 $\mathrm{e}p_{\Phi^\sigma}\Delta_{\Phi^\sigma} < 1$, 其中 $\Delta_{\Phi^\sigma} \leqslant \Delta$ 是简化实例 Φ^σ 的约束度. 然后, 依据定理 6, Φ^σ 是可满足的, 即 σ 是可行的. 注意到由 μ_{Φ^σ} 在 Φ^σ 解上的边缘分布, 得到的 v 的边缘分布恰好是 μ_v^σ. 根据推论 1, 对于任意 $v \in V^\sigma = V \setminus \Lambda(\sigma)$ 和任意 $x \in Q_v$, 有

$$\mu_v^\sigma(x) \geqslant \frac{1}{q_v} - \left((1-\mathrm{e}\alpha q)^{-\Delta} - 1\right) = \frac{1}{q_v} - \eta = \theta_v + \zeta$$

和

$$\mu_v^\sigma(x) \leqslant \frac{1}{q_v} + \left((1-\mathrm{e}\alpha q)^{-\Delta} - 1\right) = \frac{1}{q_v} + \eta = \theta_v + 2\eta + \zeta.$$

5　采样算法正确性分析

本节证明算法 1 的正确性. 本节中的所有定理都假设式 (6) 和 (7) 中的参数设置, 以及假设 1 和 2 中的判定器设置. 以下是有关算法 1 正确性的主定理.

定理8　对于任何满足式 (3) 的输入 CSP 公式 $\Phi = (V, \mathcal{Q}, \mathcal{C})$, 算法 1 以概率 1 终止, 并在终止时返回 Φ 的一个均匀随机的解.

注释4 (采样的完美性)　注意, 上述定理中的采样是完美的: 算法 1 返回的样本正好服从均匀分布 μ, 该分布覆盖 Φ 的所有解. 在第 6 节中, 假设 2 中假设的冻结判定器将通过蒙特卡罗法实现, 这将进一步推广采样算法, 在满足性判定器(假设 1) 之外不进行任何额外假设, 但会引入一个有界的偏差来影响采样结果.

注释5 (较弱的 LLL 条件)　式 (3) 中的 LLL 条件主要是为了保证算法的效率. 实际上, 定理 8 在一个更弱的 LLL 条件下也成立:

$$2\mathrm{e} \cdot q^2 \cdot p \cdot \Delta < 1.$$

在这个条件下, 存在 α 和 ζ 的选择, 使得它们满足 $p < \alpha < \frac{1}{2\mathrm{e}q^2\Delta}$ 且 $0 < \zeta < \frac{1}{q} - (1-\mathrm{e}\alpha q)^{-\Delta} + 1$. 对于这些参数的任何选择, 原有的分析依然适用, 且算法 1 的正确性与定理 8 中的声明一致.

以下引理保证了算法 3 和 4 的输入分别满足条件 1 和 2 中的不变量条件.

引理2　在执行算法 1 时, 输入为满足式 (3) 的 CSP 公式 $\Phi = (V, \mathcal{Q}, \mathcal{C})$:

(1) 每当调用 MarginSample(Φ, X, v) 时, 条件 1 都会被输入 (Φ, X, v) 满足;

(2) 每当调用 MarginOverflow(Φ, σ, v) 时, 条件 2 都会被输入 (Φ, σ, v) 满足.

在证明这个引理之前, 首先展示这些不变量条件已经足以推导出 MarginSample 的正确性, 这对于主采样算法的正确性至关重要, 因为 RejectionSampling 的正确性是已知的 (见定理 5).

定理9　对于算法 3 和 4, 以下结论成立:

(1) 假设条件 1 成立, MarginSample(Φ, σ, v) 以概率 1 终止, 并且终止时返回一个随机值 $x \in Q_v$, 其分布为 μ_v^σ;

(2) 假设条件 2 成立, MarginOverflow(Φ, σ, v) 以概率 1 终止, 并且终止时返回一个随机值 $x \in Q_v$, 其分布为式 (8) 中定义的 $\mathcal{D} \triangleq \frac{\mu_v^\sigma - \theta_v}{1 - q_v \cdot \theta_v}$.

证明　通过结构归纳 (structural induction) 验证 MarginOverflow 的正确性. 然后 MarginSample 的正确性可以直接得出.

假设 MarginOverflow 在输入 (Φ, σ, v) 上运行, 并且满足条件 2.

基本情况 (base case) 是当 NextVar$(\sigma) = \perp$ 时, MarginOverflow 不会进行进一步的递归调用. 在这种情况下, 基于引理 1 中所述的伯努利工厂的正确性, 假设满足条件 2, 则 MarginOverflow(Φ, σ, v) 以概率 1 终止, 并且返回一个随机值 $x \in Q_v$, 其分布为 $\mathcal{D}$.

对于归纳步骤, 假设 NextVar$(\sigma) = u \in V$. 根据归纳假设, 算法 4 中第 7 和 8 行对 MarginOverflow 的递归调用都以概率 1 终止. 所有其他计算都是有限的. 因此, 根据归纳法, MarginOverflow(Φ, σ, v) 以概率 1 终止.

接下来, 验证采样的正确性. 设 W 表示在算法 4 中第 3~7 行生成的 $\sigma(u)$ 的值. 可以验证, 对于每个 $a \in Q_u$, 有

$$\begin{aligned}\mathbf{Pr}[W = a] &= \mathbf{Pr}[r < q_u \cdot \theta_u] \cdot \frac{1}{q_u} + \mathbf{Pr}[r \geqslant q_u \cdot \theta_u] \cdot \mathbf{Pr}[\mathsf{MarginOverflow}(\Phi, \sigma_{u \leftarrow \star}, u) = a] \\ &= \theta_u + \mu_u^\sigma(a) - \theta_u = \mu_u^\sigma(a),\end{aligned} \tag{11}$$

其中第二个等式由归纳假设 (induction hypothesis) 得出. 故对每个 $a \in Q_u$, 有

$$\begin{aligned}\mathbf{Pr}[\mathsf{MarginOverflow}(\Phi, \sigma, v) = a] &= \sum_{b \in Q_u} \Big(\mathbf{Pr}[W = b] \cdot \mathbf{Pr}[\mathsf{MarginOverflow}(\Phi, \sigma_{u \leftarrow b}, v) = a]\Big) \\ &= \sum_{b \in Q_u} \left(\mu_u^\sigma(b) \cdot \frac{\mu_v^{\sigma_{u \leftarrow b}}(a) - \theta_v}{1 - q_v \cdot \theta_v}\right) \quad \text{(式 (11) 与归纳假设)} \\ &= \frac{\mu_v^\sigma(a) - \theta_v}{1 - q_v \cdot \theta_v},\end{aligned}$$

说明 MarginOverflow(Φ, σ, v) 返回的值遵循分布 $\mathcal{D}$.

因此, 归纳完成, 并证明了在假设条件 2 下 MarginOverflow 的正确性.

接下来, 验证 MarginSample 在条件 1 下的正确性是直接的.

令 (Φ, σ, v) 为 MarginSample 的任意输入, 并满足条件 1. 可以验证, $(\Phi, \sigma_{v \leftarrow \star}, v)$ 满足条件 2, 因此可以应用 MarginOverflow 的正确性, 故 MarginSample(Φ, σ, v) 以概率 1 终止, 且对于每个 $a \in Q_v$, 有

$$\begin{aligned}&\mathbf{Pr}[\mathsf{MarginSample}(\Phi, \sigma, v) = a] \\ &= \mathbf{Pr}[r < q_v \cdot \theta_v] \cdot \frac{1}{q_v} + \mathbf{Pr}[r \geqslant q_v \cdot \theta_v] \cdot \mathbf{Pr}[\mathsf{MarginOverflow}(\Phi, \sigma_{v \leftarrow \star}, v) = a] \\ &= \theta_v + \mu_v^\sigma(a) - \theta_v \\ &= \mu_v^\sigma(a).\end{aligned}$$

这证明了 MarginSample 的终止性和正确性.

接下来验证在引理 2 中声明的不变量. 在此之前, 我们给出在算法 1 中产生的部分赋值序列的正式定义.

定义6 (算法 1 中的部分赋值序列)　令 $X^0, X^1, \ldots, X^n \in \mathcal{Q}^*$ 表示部分赋值序列, 其中 $X^0 = \star^V$, 并且对于每个 $1 \leqslant i \leqslant n$, X^i 是算法 1 中在第 i 次循环迭代后得到的部分赋值 X, 这些赋值出现在第 1~4 行之间的 for 循环内.

首先注意到以下事实.

事实1　对于每个 $1 \leqslant i \leqslant n$, 要么 v_i 是 X^{i-1}- 固定的, 在这种情况下 $X^i = X^{i-1}$; 要么不是, 在这种情况下 X^i 通过在 Q_{v_i} 中为 v_i 赋一个值来扩展 X^{i-1} 得到. 因此, 对于每个 $1 \leqslant i \leqslant n$, 如果 v_i 不是 X^{i-1}- 固定的, 则有 $X^*(v_i) = X^i(v_i)$, 其中 X^* 表示算法 1 的输出.

引理3　对于定义 6 中的 $X^0, X^1, \ldots, X^n$, 以及所有 $0 \leqslant i \leqslant n$, X^i 是可行的, 并且

$$\mathcal{P}\left[\neg c \,\middle|\, X^i\right] < \alpha q, \quad \forall\, c \in \mathcal{C}. \tag{12}$$

证明　只需要证明式 (12) 成立. 然后, X^i 的可行性可以通过推论 2 得到.

固定任意的 $c \in \mathcal{C}$. 回顾假设中的 LLL 条件 (3), 由式 (6) 可得 $\alpha > p$. 然后有

$$\mathcal{P}\left[\neg c \,\middle|\, X^0\right] = \mathcal{P}\left[\neg c \,\middle|\, \star^V\right] = \mathcal{P}\left[\neg c\right] \leqslant p < \alpha.$$

假设 i^* 为最小的使得 $\mathcal{P}\left[\neg c \,\middle|\, X^i\right] \geqslant \alpha$ 的 i. 由于 $\mathcal{P}\left[\neg c \,\middle|\, X^0\right] < \alpha$, 所以有 $i^* \geqslant 1$. 然后 $\mathcal{P}\left[\neg c \,\middle|\, X^{i^*-1}\right] \leqslant \alpha$, 这意味着 $X^{i^*} \neq X^{i^*-1}$. 结合定义 6 和算法 1, 得出 X^{i^*} 通过为 v_{i^*} 在 $Q_{v_{i^*}}$ 中赋一个值来扩展 X^{i^*-1}. 因此,

$$\mathcal{P}\left[\neg c \,\middle|\, X^{i^*}\right] \leqslant \frac{\mathcal{P}\left[\neg c \,\middle|\, X^{i^*-1}\right]}{\min\limits_{x \in Q_{v_{i^*}}} \mathcal{P}\left[v_{i^*} = x \,\middle|\, X^{i^*-1}\right]} < \alpha \left|Q_{v_{i^*}}\right| \leqslant \alpha q. \tag{13}$$

因此, $\alpha \leqslant \mathcal{P}\left[\neg c \,\middle|\, X^{i^*}\right] < \alpha q$. 结合定义 4, 可知 $\mathsf{vbl}(c)$ 中的所有变量都是 X^{i^*}- 固定的, 因此 $\mathsf{vbl}(c)$ 中的变量将在算法 1 中保持不变. 因此, 如果 $i^* < n$, 则对于每个 $v \in \mathsf{vbl}(c)$ 都有 $X^{i^*+1}(v) = X^{i^*}(v)$, 然后 $\alpha \leqslant \mathcal{P}\left[\neg c \,\middle|\, X^{i^*+1}\right] = \mathcal{P}\left[\neg c \,\middle|\, X^{i^*}\right] < \alpha q$. 通过迭代, 可以证明

$$\alpha \leqslant \mathcal{P}\left[\neg c \,\middle|\, X^i\right] < \alpha q, \quad \forall\, i^* \leqslant i \leqslant n.$$

此外, 由于 i^* 是最小的使得 $\mathcal{P}\left[\neg c \,\middle|\, X^i\right] \geqslant \alpha$ 的 $0 \leqslant i \leqslant n$, 所以有

$$\mathcal{P}\left[\neg c \,\middle|\, X^i\right] < \alpha < \alpha q, \quad \forall\, 0 \leqslant i < i^*.$$

因此, 式 (12) 成立.

引理 2(1) 中关于 MarginSample 的条件 1 不变量可以从引理 3 中容易地得到. 为了证明 MarginOverflow 的条件 2 不变量, 需要以下引理.

引理4　假设 (Φ, σ, v) 满足条件 2. 对于任意 $u \in V$, 如果 u 不是 σ- 固定的, 则对于任意 $a \in Q_u \cup \{\star\}$, $(\Phi, \sigma_{u \leftarrow a}, v)$ 和 $(\Phi, \sigma_{u \leftarrow \star}, u)$ 都满足条件 2.

证明　对于 $(\Phi, \sigma_{u \leftarrow \star}, u)$ 和 $(\Phi, \sigma_{u \leftarrow a}, v)$ 的证明是类似的. 因此, 只需要证明对于任意 $a \in Q_u \cup \{\star\}$, $(\Phi, \sigma_{u \leftarrow a}, v)$ 满足条件 2. $\sigma_{u \leftarrow a}$ 的可行性由推论 2 得到. 同时, 也有 $\sigma_{u \leftarrow \star}(v) = \star$.

接下来, 证明对于所有 $c \in \mathcal{C}$, 有 $\mathcal{P}\left[\neg c \mid \sigma_{u \leftarrow \star}\right] \leqslant \alpha q$.

如果 $a = \star$, 则 $\mathcal{P}\left[\neg c \mid \sigma_{u \leftarrow \star}\right] \leqslant \alpha q$ 显然成立, 因为作为一个非 σ- 固定变量, u 必定满足 $\sigma(u) = \star$, 因此将 σ 修改为 $\sigma_{u \leftarrow \star}$ 并不会改变任何约束被违反的条件概率.

对于 $a \in Q_u$ 的情况, 如果 c 是 σ- 冻结的, 则 $u \notin \mathsf{vbl}(c)$, 又因为 u 不是 σ- 固定的, 因此有 $\mathcal{P}\left[\neg c \mid \sigma_{u \leftarrow a}\right] = \mathcal{P}\left[\neg c \mid \sigma\right] \leqslant \alpha q$, 其中不等式是因为 (Φ, σ, v) 满足条件 2; 如果 c 不是 σ- 冻结的, 则 $\mathcal{P}\left[\neg c \mid \sigma\right] \leqslant \alpha$, 因此如同在式 (13) 中计算的那样, 得到

$$\mathcal{P}\left[\neg c \mid \sigma_{u \leftarrow a}\right] \leqslant |Q_u|\, \mathcal{P}\left[\neg c \mid \sigma\right] \leqslant \alpha q,$$

从而完成证明.

引理 2(2) 中关于 MarginOverflow 的条件 2 不变性, 来自引理 2(1) 和 4. 因为在 (Φ, σ, v) 满足条件 1 的情况下, $(\Phi, \sigma_{v\leftarrow\star}, v)$ 满足条件 2. 此外, 在执行 MarginOverflow(Φ, τ, v) 时, 其中 $\tau = \sigma_{v\leftarrow\star}$, 算法只会对于那些不是 τ-固定的顶点 u, 将输入的部分赋值 τ 改为 $\tau_{u\leftarrow a}$, 其中 $a \in Q_u \cup \{\star\}$.

从而, 引理 2 得到了证明.

结合引理 2 和定理 9, 我们证明了在主算法 (算法 1) 中输入的 CSP 满足式 (3) 的 LLL 条件下, MarginSample (算法 3) 的正确性.

RejectionSampling (算法 2) 的正确性已经在定理 5 中建立 (这是标准的).

主采样算法 (算法 1) 的正确性随后也从这两个主要子程序的正确性中得出. 需要注意的是, 这并非直接的, 因为在算法 1 中, 变量是从它们的边缘分布中自适应地选择的. 接下来正式证明, 这种自适应选择随机性的方式并不会影响采样的正确性.

定理 8 的证明 根据引理 5 和 1, 算法 1 以概率 1 终止.

设 $X^0, X^1, \ldots, X^n$ 为在定义 6 中定义的部分赋值序列. 设 X^* 表示算法 1 的输出. 接下来证明, 对于每个 $\sigma \in \Omega$, 都有 $\mathbf{Pr}\left[X^* = \sigma\right] = \mu(\sigma)$.

固定一个任意的解 $\sigma \in \Omega$. 进一步定义部分赋值序列 $\sigma^0, \sigma^1, \ldots, \sigma^n$ 如下. 设 $\sigma^0 = \star^V$. 对于每个 $1 \leqslant i \leqslant n$, 如果 v_i 是 σ^{i-1}- 固定的, 则令 $\sigma^i = \sigma^{i-1}$; 否则, 令 $\sigma^i = \sigma^{i-1}_{v_i\leftarrow\sigma(v_i)}$. 我们声称对于每个 $0 \leqslant i \leqslant n$, 都有

$$\prod_{j=1}^{i} \mathbf{Pr}\left[X^j = \sigma^j \;\middle|\; X^{j-1} = \sigma^{j-1}\right] = \mu_{\Lambda(\sigma^i)}(\sigma_{\Lambda(\sigma^i)}), \tag{14}$$

其中约定当 $i = 0$ 时, 左右两边均等于 1.

接下来通过数学归纳法证明此命题. 基本情况 $i = 0$ 按照约定成立.

对于归纳步骤, 考虑 $i \geqslant 1$. 根据假设 2 中冻结判定器 的一致性, 如果 $X^{i-1} = \sigma^{i-1}$, 则 v_i 在 X^{i-1} 和 σ^{i-1} 中只能同时是固定的或非固定的.

- 如果 v_i 是 σ^{i-1}- 固定的, 则 $\sigma^i = \sigma^{i-1}$, 根据算法 1 中的第 3 和 4 行, 有

$$\mathbf{Pr}\left[X^i = \sigma^i \;\middle|\; X^{i-1} = \sigma^{i-1}\right] = 1.$$

因此,

$$\begin{aligned}\prod_{j=1}^{i} \mathbf{Pr}\left[X^j = \sigma^j \;\middle|\; X^{j-1} = \sigma^{j-1}\right] &= \prod_{j=1}^{i-1} \mathbf{Pr}\left[X^j = \sigma^j \;\middle|\; X^{j-1} = \sigma^{j-1}\right] \\ &= \mu_{\Lambda(\sigma^{i-1})}(\sigma_{\Lambda(\sigma^{i-1})}) \quad (\text{归纳假设}) \\ &= \mu_{\Lambda(\sigma^i)}(\sigma_{\Lambda(\sigma^i)}) \quad (\sigma^i = \sigma^{i-1}).\end{aligned}$$

- 如果 v_i 不是 σ^{i-1}- 固定的, 则根据 MarginSample 的正确性 (由引理 2 和定理 9 保证), 有

$$\mathbf{Pr}\left[X^i = \sigma^i \;\middle|\; X^{i-1} = \sigma^{i-1}\right] = \mu_{v_i}^{\sigma^{i-1}}(\sigma(v_i)).$$

因此, 有

$$\begin{aligned}\prod_{j=1}^{i} \mathbf{Pr}\left[X^j = \sigma^j \;\middle|\; X^{j-1} = \sigma^{j-1}\right] &= \mu_{v_i}^{\sigma^{i-1}}(\sigma(v_i)) \cdot \prod_{j=1}^{i-1} \mathbf{Pr}\left[X^j = \sigma^j \;\middle|\; X^{j-1} = \sigma^{j-1}\right] \\ &= \mu_{v_i}^{\sigma^{i-1}}(\sigma(v_i)) \cdot \mu_{\Lambda(\sigma^{i-1})}(\sigma_{\Lambda(\sigma^{i-1})}) \quad (\text{归纳假设}) \\ &= \mu_{\Lambda(\sigma^i)}(\sigma_{\Lambda(\sigma^i)}) \quad (\text{链式法则}).\end{aligned}$$

至此完成了归纳证明. 式 (14) 得证.

观察到序列 $X_0, X_1, \ldots, X_n, X^*$ 是一个马尔可夫链, 其中最后一步 X^* 是通过 RejectionSampling 从 X^n 构造的. 假设事件 $X^* = \sigma$ 发生. 根据事实 1, 如果 v_i 不是 X^{i-1}- 固定的, 则有 $X^i(v_i) = \sigma(v_i)$. 因此由定义 6 可知, 如果 v_i 是 X^{i-1}- 固定的, 则 $X^i = X^{i-1}$; 否则, $X^i = X^{i-1}_{v_i \leftarrow \sigma(v_i)}$. 因此, 假设 $X^* = \sigma$ 发生, 基于假设 2 中冻结判定器 的一致性, 可以验证 $X^i = \sigma^i$ 对所有 $0 \leqslant i \leqslant n$ 成立. 因此,

$$
\begin{aligned}
\mathbf{Pr}\left[X^* = \sigma\right] &= \mathbf{Pr}\left[(X^* = \sigma) \wedge \left(\bigwedge_{1 \leqslant i \leqslant n} (X^i = \sigma^i)\right)\right] \\
&= \mathbf{Pr}\left[X^* = \sigma \mid \forall 1 \leqslant i \leqslant n, X^i = \sigma^i\right] \cdot \prod_{i=1}^{n} \mathbf{Pr}\left[X^i = \sigma^i \mid \forall 0 \leqslant j < i, X^j = \sigma^j\right] \quad (\text{链式法则}) \\
&= \mathbf{Pr}\left[X^* = \sigma \mid X^n = \sigma^n\right] \cdot \prod_{i=1}^{n} \mathbf{Pr}\left[X^i = \sigma^i \mid X^{i-1} = \sigma^{i-1}\right] \\
&= \mathbf{Pr}\left[X^* = \sigma \mid X^n = \sigma^n\right] \cdot \mu_{\Lambda(\sigma^n)}(\sigma_{\Lambda(\sigma^n)}) \quad (\text{式 } (14)) \\
&= \mu^{\sigma^n}_{V \setminus \Lambda(\sigma^n)}(\sigma_{V \setminus \Lambda(\sigma^n)}) \cdot \mu_{\Lambda(\sigma^n)}(\sigma_{\Lambda(\sigma^n)}) \\
&= \mu(\sigma) \quad (\text{链式法则}),
\end{aligned}
$$

其中第三个等式由序列 $X_0, X_1, \ldots, X_n, X^*$ 的马尔可夫性质得出, 第五个等式由定理 5 得到.

6　采样算法效率分析

本节证明在式 (3) 中的 LLL 条件下, 算法 1 的高效性.

算法 1 假设了能够访问以下两类有关约束 $\mathcal{C}$ 的判定器, 这些判定器都接受一个约束 $c \in \mathcal{C}$ 和一个部分赋值 $\sigma \in \mathcal{Q}^*$ 作为输入.

- $\mathsf{Eval}(c, \sigma)$: 假设 1 中的满足性判定器, 用于判定是否有 $\mathcal{P}[c \mid \sigma] = 1$, 即 c 是否已被 σ 满足;
- $\mathsf{Frozen}(c, \sigma)$: 假设 2 中的冻结判定器, 它区分两种情况 $\mathcal{P}[\neg c \mid \sigma] > \alpha$ 和 $\mathcal{P}[\neg c \mid \sigma] < 0.99\alpha$, 其中 α 是式 (6) 中定义的阈值, 并且在其他情况下提供任意一致的回答.

接下来计算采样算法的复杂度, 主要考虑对两个判定器 $\mathsf{Eval}(\cdot)$ 和 $\mathsf{Frozen}(\cdot)$ 的调用次数以及计算复杂度. 我们证明了以下有关采样算法效率的定理.

定理10　给定一个满足式 (3) 的 CSP 公式 $\Phi = (V, \mathcal{Q}, \mathcal{C})$ 作为输入, 算法 1 对 $\mathsf{Eval}(\cdot)$ 的期望调用次数为 $O(q^2k^2\Delta^9 n)$, 对 $\mathsf{Frozen}(\cdot)$ 的期望调用次数为 $O(k\Delta^6 n)$, 期望计算复杂度为 $O(q^3k^3\Delta^9 n)$.

结合定理 8 中给出的算法 1 的正确性, 这证明了完美采样的主定理 (定理 2) 成立, 因为在定理 2 假设的条件下, 存在估计条件违反概率的 FPTAS, 因此 $\mathsf{Frozen}(\cdot)$ 可在 $\mathrm{poly}(q, k)$ 时间内实现.

注释6 (冻结判定器的蒙特卡罗实现)　通过蒙特卡罗法实现冻结判定器 $\mathsf{Frozen}(\cdot)$. 每当对一个约束 c 和一个部分赋值 σ 求 $\mathsf{Frozen}(c, \sigma)$ 的值时, 我们可以通过独立地测试 $O\left(\frac{1}{\alpha}\log\frac{1}{\delta}\right)$ 次, 判断约束 c 是否被与 σ 在 $\mathsf{vbl}(c)$ 上一致的随机赋值所满足, 从而以概率 $1-\delta$ 区分两种极端情况: $\mathcal{P}[\neg c \mid \sigma] > \alpha$ 和 $\mathcal{P}[\neg c \mid \sigma] < 0.99\alpha$. 进一步应用记忆化 (memoization) 技术, 保证该判定器 的一致性, 满足假设 2 中的要求. 将此时的算法 1 称为算法 $1'$, 并将在之后的第 6.7 小节中正式描述.

这种对 $\mathsf{Frozen}(\cdot)$ 的蒙特卡罗实现引入了有限的偏差, 使得原本完美的采样算法 (算法 1) 转变为一个近似采样算法 (算法 $1'$), 它不再依赖于除了假设 1 中的满足性判定器以外的任何额外假设.

定理11　给定输入 $\varepsilon \in (0, 1)$ 和一个满足式 (3) 的 CSP 公式 Φ, 算法 $1'$ 对 $\mathsf{Eval}(\cdot)$ 的期望调用次数为 $O\left(q^2k^2\Delta^9 n \log\left(\frac{\Delta n}{\varepsilon}\right)\right)$, 期望计算复杂度为 $O\left(q^3k^3\Delta^9 n \log\left(\frac{\Delta n}{\varepsilon}\right)\right)$, 并且输出与在相同输入 Φ 上运行算法 1 的输出之间的全变差距离不超过 ε.

结合定理 8 中所述的算法 1 的正确性, 这证明了本文的主定理 (定理 1).

复杂度的符号表示. 本节采用以下抽象符号表示任何复杂度. 复杂度被正式表示为一个的二元线性函数:

$$t(\underline{\mathbf{x}}, \underline{\mathbf{y}}) = \alpha \cdot \underline{\mathbf{x}} + \beta \cdot \underline{\mathbf{y}} + c, \tag{15}$$

其中, α 表示对 $\mathsf{Eval}(\cdot)$ 的调用次数, β 表示对 $\mathsf{Frozen}(\cdot)$ 的调用次数, γ 表示计算复杂度.

例如, 定理 10 和 11 中的复杂度分别表示为

$$O\left((q^2k^2\Delta^9 n) \cdot \underline{\mathbf{x}} + (k\Delta^6 n) \cdot \underline{\mathbf{y}} + q^3k^3\Delta^9 n\right)$$

和

$$O\left(\left(q^2k^2\Delta^9 n \log\left(\frac{\Delta n}{\varepsilon}\right)\right) \cdot \underline{\mathbf{x}} + q^3k^3\Delta^9 n \log\left(\frac{\Delta n}{\varepsilon}\right)\right).$$

我们指出, 这种表达式仅为符号上的便利, 因为我们希望在分析中同时处理三种不同的复杂度度量. 在整个分析过程中, 仅会对这些函数 $t(\underline{\mathbf{x}}, \underline{\mathbf{y}})$ 进行线性计算. 我们进一步规定:

$$\alpha \cdot \underline{\mathbf{x}} + \beta \cdot \underline{\mathbf{y}} + \gamma \leqslant \alpha' \cdot \underline{\mathbf{x}} + \beta' \cdot \underline{\mathbf{y}} + \gamma' \iff \alpha \leqslant \alpha' \wedge \beta \leqslant \beta' \wedge \gamma \leqslant \gamma'.$$

用 $t(0, 0)$ 来表示式 (15) 中的常数项 γ, 即计算复杂度.

6.1 输入模型和数据结构

除了通过两个在上文中提到的 $\mathsf{Eval}(\cdot)$ 和 $\mathsf{Frozen}(\cdot)$ 这两个判定器的访问外, 算法还可以用以下方式来访问输入的 CSP 公式 $\Phi = (V, \mathcal{Q}, \mathcal{C})$.

- 可以对变量集合 $V = \{v_1, v_2, \ldots, v_n\}$ 和约束集合 $\mathcal{C} = \{c_1, c_2, \ldots, c_m\}$ 中的元素进行随机访问 (random access).
- 对于任意 $c \in \mathcal{C}$, 其对应的 $\mathsf{vbl}(c)$ 可以在 $O(k)$ 时间内检索; 对于任意 $v \in V$, 包含 v 的约束集合 c 可以在 $O(\Delta)$ 时间内检索; 对于任意 $c \in \mathcal{C}$, 与 $\mathsf{vbl}(c)$ 相交的依赖约束集合 $c' \in \mathcal{C}$ 可以在 $O(\Delta)$ 时间内检索.

这些要求可以通过将 CSP 公式 $\Phi = (V, \mathcal{Q}, \mathcal{C})$ 表示为

- 以变量集合和约束集合作为顶点, 约束与变量之间的包含关系作为边的二部图 (bipartite graph);
- Φ 的依赖图 G_Φ (在定义 3 中定义),

并使用邻接表 (adjacency list) 数据结构维护来满足. 对于部分赋值 $\sigma \in \mathcal{Q}^*$ 的维护, 我们想使得 σ 作为参数传递给函数时, 时间复杂度为 $O(1)$. 这一点可以通过将 σ 全局存储为一个大小为 $|V|$ 的由栈组成的数组 (array of stacks), 并在将 σ 作为函数参数传递时传递该数组的指针来解决; 当一个值 x 被赋给 $\sigma(v)$ 时, x 被压入与 v 相关联的栈中; 当函数返回时, 它弹出当前递归层级中已更新的变量所关联的栈. 我们还对部分赋值 $\sigma \in \mathcal{Q}^*$ 维护一个链表 (linked list), 用于存储当前设置为 $\star$ 的变量.

6.2 递归代价树

证明定理 10 的一个关键步骤是分析 $\mathsf{MarginSample}$ 算法 (见算法 3), 该算法调用了递归子程序 $\mathsf{MarginOverflow}$ (见算法 4). 考虑一个满足条件 1 并使得 $\mathsf{MarginSample}(\Phi, \sigma, v)$ 是良定义的输入 (Φ, σ, v). 我们的目标是对以下复杂度进行上界分析.

定义7　令 $\bar{t}_{\mathsf{MS}}(\Phi, \sigma, v)$ 表示 $\mathsf{MarginSample}(\Phi, \sigma, v)$ (见算法 3) 的期望复杂度.

计算 $\mathsf{MarginSample}(\Phi, \sigma, v)$ 涉及两个非平凡的任务: 计算 $\mathsf{NextVar}(\sigma)$ 和伯努利工厂, 这两个任务都出现在 $\mathsf{MarginOverflow}$ (见算法 4) 中.

定义8　令 $\bar{t}_{\mathsf{BF}}(\sigma)$ 表示在满足条件 2 的最坏情况下, 计算伯努利工厂所需要的期望复杂度 (见算法 4 中的第 10 行), 其中 v 存在时定义为 $\bar{t}_{\mathsf{BF}}(\sigma)$; 若不存在这样的 v, 则令 $\bar{t}_{\mathsf{BF}}(\sigma) = 0$. 令 $t_{\mathsf{var}}(\sigma)$ 表示按照式 (9) 的定义确定性计算 $\mathsf{NextVar}(\sigma)$ 的复杂度.

注意到, 定义 8 中的 $\bar{t}_{\mathsf{BF}}(\sigma)$ 是在良定义的输入 (Φ,σ,v) 上计算伯努利工厂的期望复杂度的一个上界. 关于 $t_{\mathsf{var}}(\sigma)$ 和 $\bar{t}_{\mathsf{BF}}(\sigma)$ 的具体上界将在第 6.5.1 和 6.5.2 小节中分别给出. 上述复杂度 $\bar{t}_{\mathsf{MS}}(\Phi,\sigma,v)$, $t_{\mathsf{var}}(\sigma)$ 和 $\bar{t}_{\mathsf{BF}}(\sigma)$ 均以式 (15) 的形式表示. 接下来引入一种叫作递归代价树的组合结构, 其用于将 $\bar{t}_{\mathsf{MS}}(\Phi,\sigma,v)$ 与 $t_{\mathsf{var}}(\sigma)$ 和 $\bar{t}_{\mathsf{BF}}(\sigma)$ 相关联. 对于每个 $v\in V$, 进一步定义 $\mathcal{Q}_v^\star \triangleq Q_v\cup\{\star\}$ 作为包含了 $\star$ 的扩展域.

定义9 (递归代价树)　对于任意的$\sigma\in\mathcal{Q}^*$, 令 $\mathcal{T}_\sigma=(T_\sigma,\rho_\sigma)$, 其中 T_σ 是一棵根节点为 σ 的树, 节点集为 $V(T_\sigma)\subseteq\mathcal{Q}^*$, 而 $\rho_\sigma: V(T_\sigma)\to[0,1]$ 是树中节点的标记, 构造过程如下.

(1) T_σ 的根节点为 σ, 且 $\rho_\sigma(\sigma)=1$, σ 的深度为 0.

(2) 对于 $i=0,1,\ldots$, 对于当前树 T_σ 中深度为 i 的所有节点 $\tau\in V(T_\sigma)$, 进行如下操作:

(a) 如果 $\mathsf{NextVar}(\tau)=\perp$, 则将 τ 作为 T_σ 的叶节点;

(b) 否则, 设 $u=\mathsf{NextVar}(\tau)$, 将 $\{\tau_{u\leftarrow x}\mid x\in Q_u\cup\{\star\}\}$ 作为节点 τ 的 q_u+1 个子节点添加, 并将它们标记为

$$\forall\, x\in Q_u\cup\{\star\},\quad \rho_\sigma(\tau_{u\leftarrow x})=\begin{cases}(1-q_u\cdot\theta_u)\rho_\sigma(\tau), & x=\star,\\ \mu_u^\sigma(x)\cdot\rho_\sigma(\tau), & x\in Q_u.\end{cases}$$

由此构造出的 $\mathcal{T}_\sigma=(T_\sigma,\rho_\sigma)$ 被称为以 σ 为根的递归代价树 (recursive cost tree, RCT).

我们在递归代价树 $\mathcal{T}_\sigma=(T_\sigma,\rho_\sigma)$ 上额外定义以下函数 $\lambda(\cdot)$:

$$\lambda(\mathcal{T}_\sigma)\triangleq\sum_{\tau\in V(T_\sigma)}\left(\rho_\sigma(\tau)\cdot t_{\mathsf{var}}(\tau)\right)+\sum_{T_\sigma\text{中的叶节点}\tau}\left(\rho_\sigma(\tau)\cdot\bar{t}_{\mathsf{BF}}(\tau)\right). \tag{16}$$

需要注意的是, $\lambda(\mathcal{T}_\sigma)$ 与 t_{var} 和 $\bar{t}_{\mathsf{BF}}$ 类似, 也以式 (15) 的形式表示. 通过函数 $\lambda(\mathcal{T}_\sigma)$, 可以给出 MarginSample 的期望复杂度的一个上界.

引理5　对于任何满足条件 1 的输入 (Φ,σ,v), 令 $\sigma^*=\sigma_{v\leftarrow\star}$, 则有如下不等式:

$$\bar{t}_{\mathsf{MS}}(\Phi,\sigma,v)\leqslant(1-q_v\cdot\theta_v)(\lambda(\mathcal{T}_{\sigma^*})+\lambda(\mathcal{T}_{\sigma^*})(0,0))+O(1),$$

其中 $\lambda(\mathcal{T}_{\sigma^*})(0,0)$ 是 $\lambda(\mathcal{T}_{\sigma^*})$ 中的常数项 (即式 (15) 中表示计算复杂度的部分).

接下来, 证明引理 5. 下面给出的 RCT 上的递归关系是容易验证的.

命题3　设 $\sigma\in\mathcal{Q}^*$ 且 $u=\mathsf{NextVar}(\sigma)$. 如果 $u\neq\perp$, 则有

$$\lambda(\mathcal{T}_\sigma)=t_{\mathsf{var}}(\sigma)+(1-q_u\cdot\theta_u)\lambda\left(\mathcal{T}_{\sigma_{u\leftarrow\star}}\right)+\sum_{x\in Q_u}\left(\mu_u^\sigma(x)\cdot\lambda\left(\mathcal{T}_{\sigma_{u\leftarrow x}}\right)\right).$$

然后说明在引理 5 中的复杂度上界适用于 MarginOverflow.

引理6　设 (Φ,σ,v) 是输入给 MarginOverflow (算法 4) 的输入, 且满足条件 2, 令 $\bar{t}_{\mathsf{MO}}(\Phi,\sigma,v)$ 表示 MarginOverflow (Φ,σ,v) 的期望复杂度, 则有

$$\bar{t}_{\mathsf{MO}}(\Phi,\sigma,v)\leqslant\lambda(\mathcal{T}_\sigma)+O(\lambda(\mathcal{T}_\sigma)(0,0)),$$

其中 $\lambda(\mathcal{T}_\sigma)(0,0)$ 是 $\lambda(\mathcal{T}_\sigma)$ 中的常数项.

证明　为了简化表述, 对于任何部分赋值 $\sigma\in\mathcal{Q}^*$, 令 $\gamma(\sigma)=\lambda(\mathcal{T}_\sigma)(0,0)$ 表示 $\lambda(\mathcal{T}_\sigma)$ 中的常数项. 设 $C>0$ 为一个足够大的能够支配 (dominate) 参数传递和 MarginOverflow(Φ,σ,v) 中第 2∼6 行所有计算复杂度的常数. 我们只需要证明

$$\bar{t}_{\mathsf{MO}}(\Phi,\sigma,v)\leqslant\lambda(\mathcal{T}_\sigma)+C\cdot\gamma(\sigma).$$

我们通过对 RCT 进行结构归纳来证明此不等式. 基本情况是, 当 T_σ 仅包含一个根节点时, 在这种情况下 $\mathsf{NextVar}(\sigma)=\perp$, 根据定义 9, 有

$$\lambda(\mathcal{T}_\sigma)=\rho_\sigma(\sigma)\cdot t_{\mathsf{var}}(\sigma)+\rho_\sigma(\sigma)\cdot\bar{t}_{\mathsf{BF}}(\sigma)=t_{\mathsf{var}}(\sigma)+\bar{t}_{\mathsf{BF}}(\sigma).$$

如果 $\mathsf{NextVar}\,(\sigma) = \perp$, 则 $\mathsf{MarginOverflow}(\Phi, \sigma, v)$ 中第 2 行的条件不满足, 因此

$$\bar{t}_{\mathsf{MO}}(\Phi, \sigma, v) \leqslant t_{\mathsf{var}}(\sigma) + \mathbf{E}\left[T_{\mathsf{BFS}}(\Phi, \sigma, v)\right] \leqslant t_{\mathsf{var}}(\sigma) + \bar{t}_{\mathsf{BF}}(\sigma) + C,$$

其中 $T_{\mathsf{BFS}}(\Phi, \sigma, v)$ 表示算法 4 中第 10 行的伯努利工厂的复杂度. 根据定义 8, 对于所有满足条件 2 的 (Φ, σ, v), 有 $\bar{t}_{\mathsf{BF}}(\sigma) \geqslant \mathbf{E}\left[T_{\mathsf{BFS}}(\Phi, \sigma, v)\right]$. 基本情况已证明. 对于归纳步骤, 假设 T_σ 是一个深度大于 0 的树. 因此, 根据定义 9, 存在某个 $u \in V$ 使得 $\mathsf{NextVar}\,(\sigma) = u \neq \perp$. 根据 $\mathsf{MarginOverflow}(\Phi, \sigma, v)$ 中第 3~7 行, 可以验证对于每个 $x \in Q_u$, 在第 8 行时, $\sigma(u) = x$ 的概率为

$$\mathbf{Pr}\left[r < q_u \cdot \theta_u\right] \cdot \frac{1}{q_u} + \mathbf{Pr}\left[r \geqslant q_u \cdot \theta_u\right] \cdot \mathbf{Pr}\left[\mathsf{MarginOverflow}(\Phi, \sigma_{u \leftarrow \star}, u) = x\right] = \mu_u^\sigma(x),$$

其中第一个等式是由于 MarginOverflow 的正确性, 依据定理 9. 根据全期望法则 (law of total expectation), 有

$$\bar{t}_{\mathsf{MO}}(\Phi, \sigma, v) = t_{\mathsf{var}}(\sigma) + (1 - q_u \cdot \theta_u)\bar{t}_{\mathsf{MO}}(\Phi, \sigma_{u \leftarrow \star}, u) + \sum_{x \in Q_u} \left(\mu_u^\sigma(x) \cdot \bar{t}_{\mathsf{MO}}(\Phi, \sigma_{u \leftarrow x}, v)\right) + C. \tag{17}$$

注意到, 根据定义 9(2)(b), 对于每个 $x \in Q_u \cup \{\star\}$, 树 T_σ 中以 $\sigma_{u \leftarrow x}$ 为根的子树正是以 $\tau = \sigma_{u \leftarrow x}$ 为根的树 T_τ. 根据引理 4, 可知 $(\Phi, \sigma_{u \leftarrow x}, v)$ 仍然满足条件 2. 因此, 根据归纳假设, 可得

$$\bar{t}_{\mathsf{MO}}(\Phi, \sigma_{u \leftarrow x}, u) \leqslant \lambda(\mathcal{T}_{\sigma_{u \leftarrow x}}) + C \cdot \gamma(\sigma_{u \leftarrow x}),$$

其中 $\gamma(\sigma_{u \leftarrow x}) = \lambda(\mathcal{T}_{\sigma_{u \leftarrow x}})(0, 0)$ 表示 $\lambda(\mathcal{T}_{\sigma_{u \leftarrow x}})$ 中的常数项. 结合 (17), 有

$$\begin{aligned}
\bar{t}_{\mathsf{MO}}(\Phi, \sigma, v) &\leqslant t_{\mathsf{var}}(\sigma) + (1 - q_u \cdot \theta_u)\left(\lambda(\mathcal{T}_{\sigma_{u \leftarrow \star}}) + C \cdot \gamma(\sigma_{u \leftarrow \star})\right) \\
&\quad + \sum_{x \in Q_u} \left(\mu_u^\sigma(x)\left(\lambda(\mathcal{T}_{\sigma_{u \leftarrow x}}) + C \cdot \gamma(\sigma_{u \leftarrow x})\right)\right) + C \\
&= \lambda(\mathcal{T}_\sigma) + C \cdot \gamma(\sigma) - C(\gamma(t_{\mathsf{var}}(\sigma))) + C \\
&\leqslant \lambda(\mathcal{T}_\sigma) + C \cdot \gamma(\sigma),
\end{aligned}$$

其中等式使用了命题 3, 而 $\gamma(t_{\mathsf{var}}(\sigma)) = t_{\mathsf{var}}(\sigma)(0, 0) \geqslant 1$ 是 $t_{\mathsf{var}}(\sigma)$ 中的常数项, 表示计算 $\mathsf{NextVar}\,(\sigma)$ 的复杂度.

对于满足条件 1 的 (Φ, σ, v), $(\Phi, \sigma_{v \leftarrow \star}, v)$ 满足条件 2, 因此

$$\bar{t}_{\mathsf{MS}}(\Phi, \sigma, v) = (1 - q_v \cdot \theta_v)\bar{t}_{\mathsf{MO}}(\Phi, \sigma_{v \leftarrow \star}, v) + O(1) \leqslant (1 - q_v \cdot \theta_v)(\lambda(\mathcal{T}_{\sigma_{v \leftarrow \star}}) + O(\gamma(\sigma_{v \leftarrow \star}))) + O(1),$$

其中不等式成立是由于引理 6. 引理 5 得证.

6.3　模拟递归代价树的随机路径

给定一个部分赋值 $\sigma \in \mathcal{Q}^*$ 和一个变量 $v \in V \setminus \Lambda(\sigma)$, 定义如下函数:

$$\begin{aligned}
\psi_v^\sigma(\star) &= \frac{1 - q_v \cdot \theta_v}{2 - q_v \cdot \theta_v}, \\
\psi_v^\sigma(x) &= \frac{\mu_v^\sigma(x)}{2 - q_v \cdot \theta_v}, \quad \forall\, x \in Q_v.
\end{aligned} \tag{18}$$

显然, $\psi_v^\sigma(\cdot)$ 是一个在 $\mathcal{Q}_v^\star$ 上的良定义的概率分布. 定义 9 中的递归代价树启发了以下部分赋值的随机过程.

定义10 ($\mathsf{Path}(\sigma)$ 过程)　对于任何部分赋值$\sigma \in \mathcal{Q}^*$, $\mathsf{Path}(\sigma) = (\sigma_0, \sigma_1, \ldots, \sigma_\ell)$ 是一个从初始状态 $\sigma_0 = \sigma$ 开始生成的部分赋值的随机序列, 其生成过程如下. 对于 $i = 0, 1, \ldots,$

(1) 如果 $\mathsf{NextVar}\,(\sigma_i) = \perp$, 则序列在 σ_i 处停止;

(2) 否则, 令 $u = \mathsf{NextVar}\,(\sigma_i) \in V$, 以如下方法从 σ_i 生成部分赋值 $\sigma_{i+1} \in \mathcal{Q}^*$: 随机为 $\sigma(u)$ 赋值 $x \in \mathcal{Q}_u^\star$, 使得

$$\mathbf{Pr}\left[\sigma_{i+1} = (\sigma_i)_{u \leftarrow x}\right] = \psi_u^\sigma(x), \quad \forall\, x \in \mathcal{Q}_u^\star.$$

定义一个随机变量 $\ell(\sigma)$, 表示 $\mathsf{Path}(\sigma) = \left(\sigma_0, \sigma_1, \ldots, \sigma_{\ell(\sigma)}\right)$ 的长度, 其分布由 σ 决定. 如果上下文清晰, 我们简写 $\ell = \ell(\sigma)$ 以及 $\mathsf{Path}(\sigma) = (\sigma_0, \sigma_1, \ldots, \sigma_\ell)$. 显然, $\mathsf{Path}(\sigma)$ 满足马尔可夫性质. 实际上, $\mathsf{Path}(\sigma)$ 可以看作是一个在空间 $\mathcal{Q}^*$ 上的马尔可夫链, 其中任何 σ 满足 $\mathsf{NextVar}(\sigma) = \perp$ 的部分赋值自环概率为 1.

对于任意两个部分赋值 $\tau_1, \tau_2 \in \mathcal{Q}^*$, 定义

$$\chi(\tau_1, \tau_2) \triangleq \prod_{v \in \Lambda^+(\tau_1) \setminus \Lambda^+(\tau_2)} (2 - q_v \cdot \theta_v), \tag{19}$$

其中当 $\Lambda^+(\tau_1) \setminus \Lambda^+(\tau_2) = \emptyset$ 时, 约定 $\chi(\tau_1, \tau_2) = 1$.

为了方便起见, 给定任意部分赋值 $\sigma \in \mathcal{Q}^*$, 令 $\mathsf{Path}(\sigma) = (\sigma_0, \sigma_1, \ldots, \sigma_\ell)$, 用 $\chi(\sigma)$ 表示 $\chi(\sigma_\ell, \sigma)$.

随机过程 $\mathsf{Path}(\sigma)$ 和函数 $\chi(\cdot, \cdot)$ 的意义在于, 它们通过以下函数与 MarginSample 的复杂度相关. 对于任意 $\ell \geqslant 0$ 的序列 $P = (\sigma_0, \sigma_1, \ldots, \sigma_\ell) \in (\mathcal{Q}^*)^{\ell+1}$, 有

$$H(P) \triangleq \sum_{i=0}^{\ell} (\chi(\sigma_i, \sigma_0) \cdot t_{\mathsf{var}}(\sigma_i)) + \chi(\sigma_\ell, \sigma_0) \cdot \bar{t}_{\mathsf{BF}}(\sigma_\ell), \tag{20}$$

其中, $t_{\mathsf{var}}(\cdot)$ 和 $\bar{t}_{\mathsf{BF}}(\cdot)$ 在定义 8 中定义, 并以式 (15) 的形式表达. 回顾式 (16) 中定义的 $\lambda(\mathcal{T}_\sigma)$, 有以下引理.

引理7　设 $\sigma \in \mathcal{Q}^*$ 为一个部分赋值, 则有

$$\mathbf{E}\left[H(\mathsf{Path}(\sigma))\right] \geqslant \lambda(\mathcal{T}_\sigma).$$

结合引理 5 和 7, 可以得到以下推论.

推论3　对于满足条件 1 的任意 (Φ, σ, v), 设 $\sigma^* = \sigma_{v \leftarrow \star}$, 则有

$$\bar{t}_{\mathsf{MS}}(\Phi, \sigma, v) \leqslant (1 - q_v \cdot \theta_v)\mathbf{E}\left[H(\mathsf{Path}(\sigma^*))\right] + O(\mathbf{E}\left[H(\mathsf{Path}(\sigma^*))\right](0, 0)) + O(1),$$

其中 $\mathbf{E}\left[H(\mathsf{Path}(\sigma^*))\right](0, 0)$ 是式 (15) 形式下 $\mathbf{E}\left[H(\mathsf{Path}(\sigma^*))\right]$ 的常数项.

引理 7 的证明　通过对 RCT 的结构归纳来证明该引理. 基本情况是, 当 T_σ 仅包含一个根节点时, 此时 $\mathsf{NextVar}(\sigma) = \perp$, 由定义 9 可得

$$\lambda(\mathcal{T}_\sigma) = \rho_\sigma(\sigma) \cdot t_{\mathsf{var}}(\sigma) + \rho_\sigma(\sigma) \cdot \bar{t}_{\mathsf{BF}}(\sigma) = t_{\mathsf{var}}(\sigma) + \bar{t}_{\mathsf{BF}}(\sigma).$$

同时, 由于 $\mathsf{NextVar}(\sigma) = \perp$, 结合定义 10, 有 $\mathsf{Path}(\sigma) = (\sigma)$, 且 $\chi(\sigma_\ell, \sigma_0) = 1$.

因此, 由式 (20) 可得

$$\mathbf{E}\left[H(\mathsf{Path}(\sigma))\right] = t_{\mathsf{var}}(\sigma) + \bar{t}_{\mathsf{BF}}(\sigma) = \lambda(\mathcal{T}_\sigma),$$

故基本情况得证.

归纳步骤. 假设 T_σ 是深度大于 0 的树. 由定义 9 可知, 存在 $u \in V$ 使得 $\mathsf{NextVar}(\sigma) = u \neq \perp$, 且 $\ell(\sigma) \geqslant 1$. 根据定义 10(2), 有

$$\mathbf{Pr}\left[\sigma_1 = \sigma_{u \leftarrow x}\right] = \psi_u^\sigma(x), \quad \forall x \in \mathcal{Q}_u^\star. \tag{21}$$

进一步地, 由马尔可夫性质可知, 对于任意 $x \in \mathcal{Q}_u^\star$, 在给定 $\sigma_1 = \sigma_{u \leftarrow x}$ 的情况下, 序列 $(\sigma_1, \sigma_2, \ldots, \sigma_{\ell(\sigma)})$ 服从与 $\mathsf{Path}(\sigma_{u \leftarrow x})$ 相同的分布. 此外, 可以验证, 对于任意满足 $\ell \geqslant 1$, $\mathbf{Pr}\left[\mathsf{Path}(\sigma) = P\right] > 0$ 的部分赋值序列 $P = (\tau_0, \tau_1, \ldots, \tau_\ell)$, 有

$$H(P) = (2 - q_u \cdot \theta_u) H((\tau_1, \ldots, \tau_\ell)).$$

因此, 在条件 $\sigma_1 = \sigma_{u \leftarrow x}$ 下, 有

$$\begin{aligned} \mathbf{E}\left[H(\mathsf{Path}(\sigma)) \mid \sigma_1 = \sigma_{u \leftarrow x}\right] &= t_{\mathsf{var}}(\sigma) + (2 - q_u \cdot \theta_u)\mathbf{E}\left[H((\sigma_1, \ldots, \sigma_\ell)) \mid \sigma_1 = \sigma_{u \leftarrow x}\right] \\ &= t_{\mathsf{var}}(\sigma) + (2 - q_u \cdot \theta_u)\mathbf{E}\left[H(\mathsf{Path}(\sigma_{u \leftarrow x}))\right]. \end{aligned} \tag{22}$$

因此, 由全期望法则可得

$$\begin{aligned}\mathbf{E}\left[H(\mathsf{Path}(\sigma))\right] &= \mathbf{Pr}\left[\sigma_1=\sigma_{u\leftarrow\star}\right]\cdot\mathbf{E}\left[H(\mathsf{Path}(\sigma))\mid\sigma_1=\sigma_{u\leftarrow\star}\right]\\&\quad+\sum_{x\in Q_u}\left(\mathbf{Pr}\left[\sigma_1=\sigma_{u\leftarrow x}\right]\cdot\mathbf{E}\left[H(\mathsf{Path}(\sigma))\mid\sigma_1=\sigma_{u\leftarrow x}\right]\right)\\&=\frac{1-q_u\cdot\theta_u}{2-q_u\cdot\theta_u}\cdot\left(t_{\mathsf{var}}(\sigma)+(2-q_u\cdot\theta_u)\mathbf{E}\left[H(\mathsf{Path}(\sigma_{u\leftarrow\star}))\right]\right)\\&\quad+\sum_{x\in Q_u}\left(\frac{\mu_u^\sigma(x)}{2-q_u\cdot\theta_u}\cdot\left(t_{\mathsf{var}}(\sigma)+(2-q_u\cdot\theta_u)\mathbf{E}\left[H(\mathsf{Path}(\sigma_{u\leftarrow x}))\right]\right)\right)\\&=t_{\mathsf{var}}(\sigma)+(1-q_u\cdot\theta_u)\mathbf{E}\left[H(\mathsf{Path}(\sigma_{u\leftarrow\star}))\right]+\sum_{x\in Q_u}\left(\mu_u^\sigma(x)\cdot\mathbf{E}\left[H(\mathsf{Path}(\sigma_{u\leftarrow x}))\right]\right),\end{aligned}\tag{23}$$

其中, 第二个等式由 (21) 和 (22) 推得, 最后一个等式由

$$\frac{1-q_u\cdot\theta_u}{2-q_u\cdot\theta_u}+\sum_{x\in Q_u}\frac{\mu_u^\sigma(x)}{2-q_u\cdot\theta_u}=1$$

推得. 由定义 9(2)(b) 可知, 对于每个 $x\in\mathcal{Q}_u^\star$, T_σ 中以 $\sigma_{u\leftarrow x}$ 为根的子树, 正是以 $\tau=\sigma_{u\leftarrow x}$ 为根的 RCT $\mathcal{T}_\tau=(T_\tau,\rho_\tau)$.

由归纳假设可得

$$\mathbf{E}\left[H(\mathsf{Path}(\sigma_{u\leftarrow x}))\right]\geqslant\lambda(\mathcal{T}_{\sigma_{u\leftarrow x}}).$$

结合 (23), 可得

$$\mathbf{E}\left[H(\mathsf{Path}(\sigma))\right]\geqslant t_{\mathsf{var}}(\sigma_0)+(1-q_u\cdot\theta_u)\lambda(\mathcal{T}_{\sigma_{u\leftarrow\star}})+\sum_{x\in Q_u}\left(\mu_u^\sigma(x)\cdot\lambda(\mathcal{T}_{\sigma_{u\leftarrow x}})\right)=\lambda(\mathcal{T}_\sigma),$$

其中, 最后一个等号由命题 3 推得, 从而证明了该引理成立.

6.4 错误路径的排除

在算法 1 中, 所维护的部分赋值依照随机序列 $X^0,X^1,\ldots,X^n$ 演化, 该随机序列的形式化定义见定义 6. 算法 1 内部调用的 MarginSample (算法 3) 的计算效率在根本上依赖于其输入部分赋值的生成方式, 即按照上述随机序列进行演化.

为方便分析, 定义一个过程 Simulate$(\cdot)$, 其中 Simulate(t) 生成算法 1 所维护的随机部分赋值序列 $X^0,X^1,\ldots,X^n$ 的前缀 $(X^0,X^1,\ldots,X^t)$, 其具体描述见算法 5. 该过程的定义仅用于理论分析, 以便更清晰地刻画算法的行为.

算法 5: Simulate $(1\leqslant t\leqslant n)$.

```
X^0 ← ☆^V;
for i = 1 to t do
    if v_i 不是 X^{i-1}-固定的 then
        均匀随机地采样实数 r ∈ [0,1), 令 b ∈ Q_{v_i} 为唯一满足 ∑_{a<b} μ_{v_i}^{X^{i-1}}(a) ⩽ r < ∑_{a⩽b} μ_{v_i}^{X^{i-1}}(a) 的值;
        X^i ← X^{i-1}_{v_i←b};
    else
        X^i ← X^{i-1};
return (X^0, X^1, ..., X^t).
```

考虑随机过程 Simulate$(t-1)=(X^0,X^1,\ldots,X^{t-1})$, 以及如下定义的中间状态 X_0^t: 若变量 v_t 不是 X^{t-1}- 固定的, 则以概率 $1-q_{v_t}\theta_{v_t}$ 令 $X_0^t=X^{t-1}_{v_t\leftarrow\star}$, 否则令 $X_0^t=X^{t-1}$. 该过程模拟了 MarginOverflow 所接收的输入, 并生成路径 $\mathsf{Path}(X_0^t)=(X_0^t,X_1^t,\ldots,X_\ell^t)$. 形式化地, 对于 $1\leqslant t\leqslant n$, 定义如下随机过程:

$$\begin{aligned}&(X^0,X^1,\ldots,X^{t-1})\leftarrow \mathsf{Simulate}(t-1),\\ &X_0^t\leftarrow\begin{cases}X^{t-1}_{v_t\leftarrow\star}, & v_t \text{ 不是 } X^{t-1}\text{- 固定的且 } r_t=1;\\ X^{t-1}, & \text{否则},\end{cases}\\ &(X_0^t,X_1^t,\ldots,X_\ell^t)\leftarrow \mathsf{Path}(X_0^t),\end{aligned}\tag{24}$$

其中 r_t 独立采样自 $\mathrm{Bern}(1-q_v\theta_v)$.

为分析 MarginSample 的期望复杂度, 我们研究 $\mathbf{E}\left[H(\mathsf{Path}(X_0^t))\right]$ 的上界, 其中 $H(\cdot)$ 由式 (20) 定义. 为此, 首先构造一类特定的 “错误” 路径来作为证据 (witness). 回顾定义 4 中的冻结变量集 $\mathcal{C}^\sigma_{\mathsf{frozen}}$ 以及定义 5 中的边界变量集 $\mathcal{C}^\sigma_{\star\text{-}\mathsf{con}}$. 对于任意 $U\subseteq V$ 和 $E\subseteq\mathcal{C}$, 用 $U\uplus E$ 来表示二者的不交并集 $U\cup E$ 使得符号更为清晰.

定义11 (σ- 错误变量、约束及事件)　令 $t\in[n]$, $U\subseteq V$, $E\subseteq\mathcal{C}$, $\sigma\in\mathcal{Q}^*$ 为部分赋值, $T=U\uplus E$ 为变量和约束的子集, 定义如下:

- 记 $\mathcal{C}^\sigma_{\star\text{-}\mathsf{frozen}}\triangleq\mathcal{C}^\sigma_{\mathsf{frozen}}\cap\mathcal{C}^\sigma_{\star\text{-}\mathsf{con}}$;
- 记 $V^\sigma_\star\triangleq\{v\in V\mid\sigma(v)=\star\}$ 为被赋值为 $\star$ 的变量集合;
- 设 $\mathcal{E}^\sigma_T$ 表示事件 $(U=V^{\sigma_\ell}_\star)\wedge(E\subseteq\mathcal{C}^{\sigma_\ell}_{\star\text{-}\mathsf{frozen}})$, 其中 σ_ℓ 为路径 $\mathsf{Path}(\sigma)$ 末尾的部分赋值;
- 设 $\mathcal{E}^t_T$ 表示事件 $\left(U=V^{X_\ell^t}_\star\right)\wedge\left(E\subseteq\mathcal{C}^{X_\ell^t}_{\star\text{-}\mathsf{frozen}}\right)$, 其中 $(X^0,\ldots,X^{t-1},X_0^t,\ldots,X_\ell^t)$ 按照式 (24) 构造.

直观而言, 集合 $V^\sigma_\star$ 和 $\mathcal{C}^\sigma_{\star\text{-}\mathsf{frozen}}$ 以及事件 $\mathcal{E}^\sigma_T$ 和 $\mathcal{E}^t_T$ 提供了 MarginOverflow 递归过程中的关键信息, 使得任何导致 MarginOverflow 计算效率下降的 “错误” 路径 Path 必然伴随大量 “错误” 变量 $V^\sigma_\star$ 和 “错误” 约束 $\mathcal{C}^\sigma_{\star\text{-}\mathsf{frozen}}$ 的出现, 从而导致某个充分大的 T 使得事件 $\mathcal{E}^\sigma_T$ 和 $\mathcal{E}^t_T$ 发生. 为了进一步刻画路径 $\mathsf{Path}(\sigma)$ 的特性, 首先给出其相关的基本性质, 包括变量和约束属性的单调性, 以及路径长度与 $\mathcal{C}^{\sigma_\ell}_v$, $\mathcal{C}^{\sigma_\ell}_{\star\text{-}\mathsf{con}}$, $V^{\sigma_\ell}_\star$ 和 $\mathcal{C}^{\sigma_\ell}_{\star\text{-}\mathsf{frozen}}$ 规模之间的关系.

回顾第 3.2 小节中的定义, 对于任意 $v\in V^\sigma$, $H^\sigma_v=(V^\sigma_v,\mathcal{C}^\sigma_v)$ 表示变量 v 所处的连通分量, 该分量由约束超图 H^σ 导出而得.

引理8　设 $\sigma\in\mathcal{Q}^*$, 且其导出路径 $\mathsf{Path}(\sigma)=(\sigma_0,\sigma_1,\ldots,\sigma_\ell)$. 则对于任意 $0\leqslant i\leqslant j\leqslant\ell$, 以下性质成立:

- (单调性) $V^{\sigma_i}_\star\subseteq V^{\sigma_j}_\star$, $\mathcal{C}^{\sigma_i}_{\mathbb{P}}\subseteq\mathcal{C}^{\sigma_j}_{\mathbb{P}}$, 其中 $\mathbb{P}$ 取值于集合 $\mathbb{P}\in\{\,\mathsf{frozen},\star\text{-}\mathsf{con},\star\text{-}\mathsf{frozen}\,\}$.

此外, 若 σ 仅有唯一变量 $v\in V$ 满足 $\sigma(v)=\star$, 则有

- (关于 $|\mathcal{C}^{\sigma_\ell}_v|$ 和 $|\mathcal{C}^{\sigma_\ell}_{\star\text{-}\mathsf{con}}|$ 的上界) $|\mathcal{C}^{\sigma_\ell}_v|\leqslant|\mathcal{C}^{\sigma_\ell}_{\star\text{-}\mathsf{con}}|\leqslant\Delta\cdot(|V^{\sigma_\ell}_\star|+|\mathcal{C}^{\sigma_\ell}_{\star\text{-}\mathsf{frozen}}|)$;
- (关于路径长度 $\mathsf{Path}(\sigma)$ 的上界) $\ell\leqslant k\Delta\cdot(|V^{\sigma_\ell}_\star|+|\mathcal{C}^{\sigma_\ell}_{\star\text{-}\mathsf{frozen}}|)$.

以下引理将在引理 8 的单调性性质的证明中被用到.

引理9　给定 $\sigma\in\mathcal{Q}^*$ 且满足 $\mathsf{NextVar}(\sigma)=u\neq\perp$, 则有 $u\notin V^\sigma_{\mathsf{fix}}$.

证明　由 $u=\mathsf{NextVar}(\sigma)\neq\perp$ 以及定义 5 中对 $\mathsf{NextVar}(\sigma)$ 的定义, 可知 $u\in V^\sigma_{\star\text{-}\mathsf{inf}}$. 结合 $V^\sigma_{\star\text{-}\mathsf{inf}}$ 的定义, 进一步得出 $u\notin V^\sigma_\star$. 又由于 $V^\sigma_\star\subseteq V^\sigma\cap V^\sigma_{\mathsf{fix}}$, 从而推导出 $u\notin V^\sigma_{\mathsf{fix}}$, 证毕.

下述引理揭示了在扩展部分赋值 σ 时所满足的基本单调性性质.

引理10　设 $\sigma\in\mathcal{Q}^*$, 且满足 $\mathsf{NextVar}(\sigma)=u\neq\perp$, 对于任意 $a\in Q_u\cup\{\star\}$, 定义 $\tau=\sigma_{u\leftarrow a}$, 则有

$$V^\sigma_\star\subseteq V^\tau_\star,\quad \mathcal{C}^\sigma_{\mathbb{P}}\subseteq\mathcal{C}^\tau_{\mathbb{P}},$$

其中 $\mathbb{P}$ 取值于集合 $\{\mathsf{frozen},\star\text{-}\mathsf{con},\star\text{-}\mathsf{frozen}\}$.

证明　首先, 证明 $V^\sigma_\star\subseteq V^\tau_\star$. 由引理 9 及 $\mathsf{NextVar}(\sigma)=u\neq\perp$ 可知 $u\notin V^\sigma_{\mathsf{fix}}$. 结合定义 4 中 V^σ_{fix} 的定义, 进一步得出 $\sigma(u)\neq\star$. 根据定义 11, 直接可得 $V^\sigma_\star\subseteq V^\tau_\star$.

接下来, 证明 $\mathcal{C}^{\sigma}_{\mathsf{frozen}} \subseteq \mathcal{C}^{\tau}_{\mathsf{frozen}}$. 对于任意约束 $c \in \mathcal{C}^{\sigma}_{\mathsf{frozen}}$, 有 $\mathsf{vbl}(c) \subseteq V^{\sigma}_{\mathsf{fix}}$. 由 $u \notin V^{\sigma}_{\mathsf{fix}}$ 可得 $u \notin \mathsf{vbl}(c)$, 进而有 $\tau_{\mathsf{vbl}(c)} = \sigma_{\mathsf{vbl}(c)}$. 依据假设 2 所述的冻结约束一致性假设, 可推导出 $c \in \mathcal{C}^{\tau}_{\mathsf{frozen}}$, 从而 $\mathcal{C}^{\sigma}_{\mathsf{frozen}} \subseteq \mathcal{C}^{\tau}_{\mathsf{frozen}}$.

然后, 证明 $\mathcal{C}^{\sigma}_{\star\text{-}\mathsf{con}} \subseteq \mathcal{C}^{\tau}_{\star\text{-}\mathsf{con}}$. 对任意 $c \in \mathcal{C}^{\sigma}_{\star\text{-}\mathsf{con}}$, 由定义 5 可知, 存在某个变量 $v \in \mathsf{vbl}(c) \cap V^{\sigma}_{\star\text{-}\mathsf{con}}$. 由于 $v \in V^{\sigma}_{\star\text{-}\mathsf{con}}$, 依据定义, 可知存在变量 v' 使得 $\sigma(v') = \star$, 且 v 与 v' 在 $H^{\sigma}_{\mathsf{fix}}$ 中连通.

接下来, 证明 $H^{\sigma}_{\mathsf{fix}}$ 是 H^{τ}_{fix} 的子超图, 这将进一步推导出 v 和 v' 在 H^{τ}_{fix} 中仍然连通. 此外, 由 $u \notin V^{\sigma}_{\mathsf{fix}}$ 和 $v' \in V^{\sigma}_{\mathsf{fix}}$ 可得 $u \neq v'$, 进而有 $\tau(v') = \sigma(v') = \star$. 结合 v 与 v' 在 H^{τ}_{fix} 中连通, 可推导出 $v \in V^{\tau}_{\star\text{-}\mathsf{con}}$. 结合 $v \in \mathsf{vbl}(c)$, 从而得出 $c \in \mathcal{C}^{\tau}_{\star\text{-}\mathsf{con}}$, 即 $\mathcal{C}^{\sigma}_{\star\text{-}\mathsf{con}} \subseteq \mathcal{C}^{\tau}_{\star\text{-}\mathsf{con}}$.

接下来, 证明 $H^{\sigma}_{\mathsf{fix}}$ 是 H^{τ}_{fix} 的子超图. 首先, 证明

$$V^{\sigma} \cap V^{\sigma}_{\mathsf{fix}} \subseteq V^{\tau} \cap V^{\tau}_{\mathsf{fix}}. \tag{25}$$

显然 $\Lambda^{+}(\sigma) \subseteq \Lambda^{+}(\tau)$. 结合 $\mathcal{C}^{\sigma}_{\mathsf{frozen}} \subseteq \mathcal{C}^{\tau}_{\mathsf{frozen}}$, 可推导出 $V^{\sigma}_{\mathsf{fix}} \subseteq V^{\tau}_{\mathsf{fix}}$. 此外, 有

$$V^{\sigma} \cap V^{\sigma}_{\mathsf{fix}} = (V^{\sigma} \setminus \{u\}) \cap V^{\sigma}_{\mathsf{fix}} \subseteq V^{\tau} \cap V^{\sigma}_{\mathsf{fix}} \subseteq V^{\tau} \cap V^{\tau}_{\mathsf{fix}},$$

其中, 第一个等式由 $u \notin V^{\sigma}_{\mathsf{fix}}$ 得出, 第二个包含关系由 $V^{\sigma} \setminus \{u\} \subseteq V^{\tau}$ 得出, 最后一个包含关系由 $V^{\sigma}_{\mathsf{fix}} \subseteq V^{\tau}_{\mathsf{fix}}$ 得出. 因此, (25) 得证. 进一步地, 设 $\mathcal{C}^{\sigma}_{\mathsf{fix}}$ 和 $\mathcal{C}^{\tau}_{\mathsf{fix}}$ 分别为 $H^{\sigma}_{\mathsf{fix}}$ 和 H^{τ}_{fix} 的超边集合, 则

$$\begin{aligned}
\mathcal{C}^{\sigma}_{\mathsf{fix}} &= \left\{ c^{\sigma} \;\middle|\; (\mathsf{vbl}(c^{\sigma}) \subseteq V^{\sigma} \cap V^{\sigma}_{\mathsf{fix}}) \wedge \left(\sigma \text{ 不满足 } c\right) \right\} \quad (H^{\sigma}_{\mathsf{fix}} \text{ 和 } \mathcal{C}^{\sigma}_{\mathsf{fix}} \text{ 的定义}) \\
&= \left\{ c^{\sigma} \;\middle|\; (\mathsf{vbl}(c^{\sigma}) \subseteq V^{\sigma} \cap V^{\sigma}_{\mathsf{fix}}) \wedge (u \notin \mathsf{vbl}(c)) \wedge \left(\sigma \text{ 不满足 } c\right) \right\} \quad (u \notin V^{\sigma}_{\mathsf{fix}}) \\
&\subseteq \left\{ c^{\tau} \;\middle|\; (\mathsf{vbl}(c^{\tau}) \subseteq V^{\sigma} \cap V^{\sigma}_{\mathsf{fix}}) \wedge (u \notin \mathsf{vbl}(c)) \wedge \left(\tau \text{ 不满足 } c\right) \right\} \quad (u \notin \mathsf{vbl}(c), \text{ 则 } c^{\sigma} = c^{\tau}) \\
&\subseteq \left\{ c^{\tau} \;\middle|\; (\mathsf{vbl}(c^{\tau}) \subseteq V^{\tau} \cap V^{\tau}_{\mathsf{fix}}) \wedge (u \notin \mathsf{vbl}(c)) \wedge \left(\tau \text{ 不满足 } c\right) \right\} \quad ((25)) \\
&\subseteq \left\{ c^{\tau} \;\middle|\; (\mathsf{vbl}(c^{\tau}) \subseteq V^{\tau} \cap V^{\tau}_{\mathsf{fix}}) \wedge \left(\tau \text{ 不满足 } c\right) \right\} \\
&= \mathcal{C}^{\tau}_{\mathsf{fix}} \quad (H^{\tau}_{\mathsf{fix}} \text{ 与 } \mathcal{C}^{\tau}_{\mathsf{fix}} \text{ 的定义}).
\end{aligned}$$

结合式 (25), 可得 $H^{\sigma}_{\mathsf{fix}}$ 是 H^{τ}_{fix} 的子超图.

最终, 由 $\mathcal{C}^{\sigma}_{\mathsf{frozen}} \subseteq \mathcal{C}^{\tau}_{\mathsf{frozen}}$ 及 $\mathcal{C}^{\sigma}_{\star\text{-}\mathsf{con}} \subseteq \mathcal{C}^{\tau}_{\star\text{-}\mathsf{con}}$ 可得

$$\mathcal{C}^{\sigma}_{\star\text{-}\mathsf{frozen}} = \mathcal{C}^{\sigma}_{\mathsf{frozen}} \bigcap \mathcal{C}^{\sigma}_{\star\text{-}\mathsf{con}} \subseteq \mathcal{C}^{\tau}_{\mathsf{frozen}} \bigcap \mathcal{C}^{\tau}_{\star\text{-}\mathsf{con}} = \mathcal{C}^{\tau}_{\star\text{-}\mathsf{frozen}}.$$

由 $\mathsf{Path}(\cdot)$ 的定义及引理 10, 引理 8 所述的单调性性质可通过数学归纳法直接推导得出.

接下来给出两个关键的技术引理, 这对于分析 MarginSample 和 RejectionSampling 的计算效率至关重要. 这两个引理主要提供了关于 "坏" 路径证据的尾界 (tail bound), 表明若按照式 (24) 生成的 "坏" 路径较大, 则其发生的概率呈指数级衰减. 首先, 陈述在分析 MarginSample 过程中所需要的尾界.

引理11　设参数 α 按照式 (6) 定义, 并假设 $8\mathrm{e}p\Delta^3 \leqslant 0.99\alpha$. 令 $1 \leqslant t \leqslant n$, 设 $(X^0, X^1, \ldots, X^{t-1}, X^t_0, X^t_1, \ldots, X^t_\ell)$ 按照式 (24) 生成. 则对于任意整数 $i \geqslant 0$, 有

$$\mathbf{Pr}\left[\left|V_{\star}^{X^t_\ell}\right| + \left|\mathcal{C}_{\star\text{-}\mathsf{frozen}}^{X^t_\ell}\right| \geqslant i\Delta\right] \cdot \mathbf{E}\left[\chi\left(X^t_0\right) \;\middle|\; \left|V_{\star}^{X^t_\ell}\right| + \left|\mathcal{C}_{\star\text{-}\mathsf{frozen}}^{X^t_\ell}\right| \geqslant i\Delta\right] \leqslant 2^{-i}.$$

进一步地, 给出用于分析 RejectionSampling 的尾界.

引理12　假设 $8\mathrm{e}p\Delta^3 \leqslant 0.99\alpha$. 令 $(X^0, X^1, \ldots, X^n) = \mathsf{Simulate}(n)$. 则对于任意整数 $i \geqslant 0$, 有

$$\mathbf{Pr}\left[\left|\mathcal{C}_v^{X^n}\right| \geqslant 2i\Delta^2\right] \leqslant 8\mathrm{e}k \cdot 4^{-i}.$$

引理 11 和 12 的证明采用相似的推理框架, 包括以下两个主要步骤:

(1) 通过对 Simulate 和 Path 的精细分析, 建立 "坏" 变量及其不相交的 "坏" 约束出现的指数级尾界;

(2) 利用一种新提出的组合结构——广义 $\{2,3\}$- 树, 将上述基本尾界提升至引理 11 所要求的形式.

引理 11 和 12 的完整证明涉及较为复杂的技术细节, 因而推迟至第 7 节进行详细阐述.

6.5 MarginSample 的效率分析

接下来证明 MarginSample 算法 (见算法 3) 期望运行时间的上界, 该上界以式 (15) 的形式进行表达. 令 $T_{\mathsf{MS}}(\Phi,\sigma,v)$ 为一个表示在满足条件 1 下, MarginSample(Φ,σ,v) 的复杂度的随机变量. 根据定义 7, $\bar{t}_{\mathsf{MS}}(\Phi,\sigma,v)$ 等于 $T_{\mathsf{MS}}(\Phi,\sigma,v)$ 的期望值, 即 $\bar{t}_{\mathsf{MS}}(\Phi,\sigma,v)=\mathbf{E}\left[T_{\mathsf{MS}}(\Phi,\sigma,v)\right]$.

定理12 假设 $8ep\Delta^3\leqslant 0.99\alpha$. 令 $X^0,X^1,\ldots,X^n$ 为定义 6 中所定义的随机序列. 假设约定当 v_t 为 X^{t-1}-固定时, $T_{\mathsf{MS}}(\Phi,X^{t-1},v_t)=0$. 对于任意的 $1\leqslant t\leqslant n$, 有如下不等式:

$$\mathbf{E}\left[T_{\mathsf{MS}}(\Phi,X^{t-1},v_t)\right]\leqslant O\left(q^2k^2\Delta^9\right)\cdot\underline{\mathbf{x}}+24k\Delta^7\cdot\underline{\mathbf{y}}+O\left(q^3k^3\Delta^9\right),$$

其中期望值是对 X^{t-1} 和 MarginSample 算法的随机性同时取期望.

上述定理中的约定是安全适用的, 因为当 v 不是 σ- 固定时, MarginSample(Φ,σ,v) 从未被主采样算法调用.

为了证明上述定理, 需要为算法中的两个非平凡步骤提供具体的复杂度上界:

(1) 出现在算法 4 第 1 行处的 NextVar(σ) 的调用;

(2) 出现在算法 4 第 10 行处的伯努利工厂的调用.

这两个步骤的复杂度将通过对 $t_{\mathsf{var}}(\sigma)$ 和 $\bar{t}_{\mathsf{BF}}(\sigma)$ (详见定义 8) 的分析得到.

6.5.1 NextVar(σ) 的复杂度

接下来给出计算 NextVar(σ) 的复杂度 $t_{\mathsf{var}}(\sigma)$ (定义 8) 的上界, 该计算位于算法 4 第 1 行处, 并且该复杂度与 $\mathcal{C}^{\sigma}_{\star\text{-con}}$ 的大小 (定义 5) 有关.

假设在第 6.1 小节中所描述的输入模型和数据结构成立, 则得出以下结论.

命题4 对于任意的 $\sigma\in\mathcal{Q}^*$, 计算 NextVar(σ) 至多需要 $|\mathcal{C}^{\sigma}_{\star\text{-con}}|$ 次对 Eval$(\cdot)$ 的调用, $\Delta\,|\mathcal{C}^{\sigma}_{\star\text{-con}}|$ 次对 Frozen$(\cdot)$ 的调用, 并且计算复杂度为 $O\left(k^2\Delta\,|\mathcal{C}^{\sigma}_{\star\text{-con}}|^2+1\right)$, 即

$$t_{\mathsf{var}}(\sigma)\leqslant|\mathcal{C}^{\sigma}_{\star\text{-con}}|\cdot\underline{\mathbf{x}}+\Delta\,|\mathcal{C}^{\sigma}_{\star\text{-con}}|\cdot\underline{\mathbf{y}}+O\left(k^2\Delta\,|\mathcal{C}^{\sigma}_{\star\text{-con}}|^2+1\right).$$

证明 进一步回顾公式 Φ 在 σ 下的固定公式, 即 $\Phi^\sigma=(V^\sigma,\mathcal{Q}^\sigma,\mathcal{C}^\sigma)$, 以及对应关联超图 $H^\sigma=H_{\Phi^\sigma}=(V^\sigma,\mathcal{C}^\sigma)$, 其正式定义见第 3.2 小节, 并在定义 NextVar(σ) 时使用. 计算 NextVar(σ) 的过程在超图 H^σ 上是直接的, 具体步骤如下.

• 在由 $V^\sigma\cap V^\sigma_{\mathsf{fix}}$ 所导出的 H^σ 子超图上, 从 $V^\sigma_\star$ 开始执行深度优先搜索 (depth-first search, DFS), 寻找包含 $V^\sigma_\star$ 的连通分量 $V^\sigma_{\star\text{-con}}$.

• 构造 $V^\sigma_{\star\text{-con}}$ 在 H^σ 中的顶点边界 (vertex boundary). 返回具有最小编号 i 的边界上的顶点 v_i; 若不存在这样的 i, 则返回 $\perp$.

接下来, 验证该过程可以在命题 4 中给定的复杂度下实现. 首先, 考虑检查某个约束 $c\in\mathcal{C}$ 是否属于由 $V^\sigma\cap V^\sigma_{\mathsf{fix}}$ 所导出的 H^σ 子超图的复杂度. 该复杂度的上界为 $\underline{\mathbf{x}}+\Delta\cdot\underline{\mathbf{y}}+O(k\Delta)$, 即一次对 Eval$(\cdot)$ 的调用, Δ 次对 Frozen$(\cdot)$ 的调用, 且计算复杂度为 $O(k\Delta)$.

这是因为, 上述检查与检查 c 是否满足 σ 且 $\mathsf{vbl}(c)\subseteq V^\sigma_{\mathsf{fix}}$ 是等价的. 而检查 c 是否满足 σ 需要一次对 Eval$(\cdot)$ 的调用; 检查是否有 $\mathsf{vbl}(c)\subseteq V^\sigma_{\mathsf{fix}}$ 可以通过枚举所有满足 $\mathsf{vbl}(c)\cap\mathsf{vbl}(c')\neq\emptyset$ 的 $c'\in\mathcal{C}$, 并检索 $\mathsf{vbl}(c')$ 来解决 (该计算复杂度为 $O(k\Delta)$), 然后检查所有这样的 c' 是否为 σ- 冻结的 (该过程总共需要至多 Δ 次对 Frozen$(\cdot)$ 的调用).

进一步分析, 深度优先搜索中需要检查是否属于导出子超图的约束集合, 实际上就是集合 $\mathcal{C}^\sigma_{\star\text{-con}}$ 中的那些约束. 因此, 检查约束是否属于导出子超图的复杂度为 $|\mathcal{C}^\sigma_{\star\text{-con}}|\cdot\underline{\mathbf{x}}+\Delta\,|\mathcal{C}^\sigma_{\star\text{-con}}|\cdot\underline{\mathbf{y}}+O(k\Delta\,|\mathcal{C}^\sigma_{\star\text{-con}}|)$. 这部分是唯一可能访问 Eval$(\cdot)$ 和 Frozen$(\cdot)$ 这两个判定器的地方, 因此可以得出对判定器的查询复杂度上界.

对于其他计算复杂度, 在深度优先搜索中, 已访问的变量和约束集可以通过两个动态数组 (dynamic array) 存储, 一个用于变量, 另一个用于约束. 当需要查询某个变量或约束是否已访问, 或更新其状态时, 我们可以遍历对应的整个数组. 这样在每次查询或更新时, 时间复杂度为当前动态数组大小的线性时间.

需要注意的是, 已访问的约束数最多为 $|\mathcal{C}^{\sigma}_{\star\text{-con}}|$, 因此已访问的变量数最多为 $k\,|\mathcal{C}^{\sigma}_{\star\text{-con}}|$. 因此, 这部分的总计算复杂度为 $O(k^2\,|\mathcal{C}^{\sigma}_{\star\text{-con}}|^2)$.

综上所述, 整体计算复杂度为 $O(k^2\Delta\,|\mathcal{C}^{\sigma}_{\star\text{-con}}|^2+1)$, 其中额外的 $O(1)$ 用于处理 $|\mathcal{C}^{\sigma}_{\star\text{-con}}|=0$ 时的退化情况.

6.5.2 伯努利工厂的复杂度

在此, 给出算法 4 中第 10 行位置的伯努利工厂的期望复杂度上界, 该结果对于分析 MarginSample 算法非常重要.

我们有如下关于伯努利工厂的 $\bar{t}_{\mathsf{BF}}(\sigma)$ (定义 8) 的上界.

命题5 存在常数 $C_0,C_1>0$, 使得以下不等式成立. 设 $1\leqslant t\leqslant n$, 并且令 $\mathsf{Path}(X_0^t)=(X_0^t,X_1^t,\ldots,X_\ell^t)$ 按照式 (24) 的方式生成, 则有

$$\bar{t}_{\mathsf{BF}}(X_\ell^t)\leqslant C_1q^2k^2\Delta^6\left(\left|\mathcal{C}^{X_\ell^t}_{\star\text{-con}}\right|+1\right)(1-\mathrm{e}\alpha q)^{-\left|\mathcal{C}^{X_\ell^t}_{\star\text{-con}}\right|}\cdot\underline{\mathbf{x}}+C_0q^3k^3\Delta^6\left(\left|\mathcal{C}^{X_\ell^t}_{\star\text{-con}}\right|+1\right)(1-\mathrm{e}\alpha q)^{-\left|\mathcal{C}^{X_\ell^t}_{\star\text{-con}}\right|}.$$

注意, 这里 X_ℓ^t 是一个随机变量, 而命题 5 中的上界适用于任何可能的 X_ℓ^t.

接下来, 证明命题 5. 在补充材料的定理 15 中, 对于所有满足条件 2 的 $\sigma\in\mathcal{Q}^*$, 我们证明了以下复杂度上界:

$$\bar{t}_{\mathsf{BF}}(\sigma)=O\left(q^2k^2\Delta^6(|\mathcal{C}_v^{\sigma}|+1)(1-\mathrm{e}\alpha q)^{-|\mathcal{C}_v^{\sigma}|}\cdot\underline{\mathbf{x}}+q^3k^3\Delta^6(|\mathcal{C}_v^{\sigma}|+1)(1-\mathrm{e}\alpha q)^{-|\mathcal{C}_v^{\sigma}|}\right),\tag{26}$$

其中 $\mathcal{C}_v^{\sigma}$ 表示在 Φ^{σ} 中包含变量 v 的连通分量的约束集合, Φ^{σ} 是 Φ 在 σ 下的固定公式.

命题 5 的证明 根据式 (24) 中的构造, X_0^t 满足以下两种情况之一:

(1) (Φ,X_0^t,v_t) 满足条件 2, 并且 v_t 是唯一一个满足 $X_0^t(v_t)=\star$ 的变量;

(2) 对于所有 $u\in V$, 有 $X_0^t(u)\in Q_u\cup\{\star\}$.

对于情况 (1), 根据引理 8, 有 $\left|\mathcal{C}_v^{X_\ell^t}\right|\leqslant\left|\mathcal{C}^{X_\ell^t}_{\star\text{-con}}\right|$, 因为 X_ℓ^t 是由遵循马尔可夫性质的路径 $\mathsf{Path}(X_0^t)=(X_0^t,X_1^t,\ldots,X_\ell^t)$ 生成的. 此外, 根据引理 4 和 Path 的构造, 可以确定条件 2 对于某些 $u\in V$ 总是由 (Φ,X_ℓ^t,u) 满足.

因此, 式 (26) 中的复杂度界限对于 X_ℓ^t 始终成立, 而结合我们刚刚建立的关系 $\left|\mathcal{C}_v^{X_\ell^t}\right|\leqslant\left|\mathcal{C}^{X_\ell^t}_{\star\text{-con}}\right|$, 就可以证明该情况.

对于情况 (2), 根据 Path 的定义, 可知 $\mathsf{Path}(X_0^t)=(X_0^t)$, 并且对于任何 v, 条件 2 不再由 $(\Phi,X_\ell^t,v)=(\Phi,X_0^t,v)$ 满足. 因此, 可以根据定义 8 得出 $\bar{t}_{\mathsf{BF}}(X_\ell^t)=0$.

6.5.3 MarginSample 的复杂度

接下来对 MarginSample 的复杂度进行分析. 首先给出一个在式 (20) 中定义的关于路径的带权函数 H 的上界.

引理13 假设 $8\mathrm{e}p\Delta^3\leqslant 0.99\alpha$, 其中 α 为如式 (6) 中所示的常数. 令 $1\leqslant t\leqslant n$, 并且令 $\mathsf{Path}(X_0^t)=(X_0^t,X_1^t,\ldots,X_\ell^t)$ 按照式 (24) 生成. 存在常数 $C,C'>0$, 使得

$$\mathbf{E}\left[H(\mathsf{Path}(X_0^t))\right]\leqslant Cq^2k^2\Delta^9\cdot\underline{\mathbf{x}}+24k\Delta^6\cdot\underline{\mathbf{y}}+C'q^3k^3\Delta^9.$$

证明 根据式 (20), 只需证明

$$\mathbf{E}\left[\sum_{i=0}^{\ell}\left(\chi(X_i^t,X_0^t)\cdot t_{\mathsf{var}}(X_i^t)\right)\right]\leqslant 24k\Delta^5\cdot\underline{\mathbf{x}}+24k\Delta^6\cdot\underline{\mathbf{y}}+120C_1k^3\Delta^8\tag{27}$$

以及

$$\mathbf{E}\left[\chi(X_\ell^t, X_0^t)\cdot \bar{t}_{\mathsf{BF}}(X_\ell^t)\right] \leqslant 20C_2k^2q^2\Delta^9\cdot \underline{\mathbf{x}} + 20C_3k^3q^3\Delta^9. \tag{28}$$

因此, 根据式 (20), 得到

$$\begin{aligned}
\mathbf{E}\left[(\mathsf{Path}(X_0^t))\right] &= \mathbf{E}\left[\sum_{i=0}^{\ell}\left(\chi(X_i^t, X_0^t)\cdot t_{\mathsf{var}}(X_i^t)\right) + \chi(X_\ell^t, X_0^t)\cdot \bar{t}_{\mathsf{BF}}(X_\ell^t)\right]\\
&= \mathbf{E}\left[\sum_{i=0}^{\ell}\left(\chi(X_i^t, X_0^t)\cdot t_{\mathsf{var}}(X_i^t)\right)\right] + \mathbf{E}\left[\chi(X_\ell^t, X_0^t)\cdot \bar{t}_{\mathsf{BF}}(X_\ell^t)\right]\\
&\leqslant 24k\Delta^5\cdot\underline{\mathbf{x}} + 24k\Delta^6\cdot\underline{\mathbf{y}} + 120C_1k^3\Delta^8 + 20C_2k^2q^2\Delta^9\cdot\underline{\mathbf{x}} + 20C_3k^3q^3\Delta^9\\
&\leqslant (20C_2+24)k^2q^2\Delta^9\cdot\underline{\mathbf{x}} + 24k\Delta^6\cdot\underline{\mathbf{y}} + (120C_1+20C_3)k^3q^3\Delta^9,
\end{aligned}$$

其中在倒数第二个不等式中, 应用了 (27) 和 (28). 通过取 $C = 20C_2 + 24$ 和 $C' = 120C_1 + 20C_3$, 即可得到引理的结论成立.

因此, 只需要证明 (27) 和 (28). 首先, 证明 (27). 我们声称

$$\left|\mathcal{C}_{\star\text{-}\mathsf{con}}^{X_\ell^t}\right| \leqslant \Delta\cdot\left(\left|V_\star^{X_\ell^t}\right| + \left|\mathcal{C}_{\star\text{-}\mathsf{frozen}}^{X_\ell^t}\right|\right). \tag{29}$$

根据式 (24) 中 X_0^t 的构造, 要么 (Φ, X_0^t, v_t) 满足条件 2 且 v_t 是唯一一个使得 $X_0^t(v_t) = \star$ 的变量, 在这种情况下, 式 (29) 可以由引理 8 得出; 要么 X_0^t 满足对于所有 $u\in V$, $X_0^t(u)\in Q_u\cup\{\text{☆}\}$, 此时 $\mathsf{Path}(X_0^t) = (X_0^t)$, $\left|\mathcal{C}_{\star\text{-}\mathsf{con}}^{X^t\ell}\right| = \left|\mathcal{C}_{\star\text{-}\mathsf{frozen}}^{X^t\ell}\right| = \left|V_\star^{X_\ell^t}\right| = 0$, 从而式 (29) 成立. 由此证明了此声称.

令 $L \triangleq \left|V_\star^{X_\ell^t}\right| + \left|\mathcal{C}_{\star\text{-}\mathsf{frozen}}^{X_\ell^t}\right|$. 根据引理 8 中的单调性, 可知 $\mathcal{C}_{\star\text{-}\mathsf{con}}^{X_i^t}\subseteq\mathcal{C}_{\star\text{-}\mathsf{con}}^{X_\ell^t}$ 对于每一个 $0\leqslant i\leqslant\ell$ 成立. 结合式 (29), 得到

$$\left|\mathcal{C}_{\star\text{-}\mathsf{con}}^{X_i^t}\right| \leqslant \left|\mathcal{C}_{\star\text{-}\mathsf{con}}^{X_\ell^t}\right| \leqslant \Delta\left(\left|V_\star^{X_\ell^t}\right| + \left|\mathcal{C}_{\star\text{-}\mathsf{frozen}}^{X_\ell^t}\right|\right) \leqslant \Delta L. \tag{30}$$

然后根据命题 4, 存在常数 $C_1 > 0$, 使得对于每个 $0\leqslant i\leqslant\ell$, 都有

$$\begin{aligned}
t_{\mathsf{var}}(X_i^t) &\leqslant \left|\mathcal{C}_{\star\text{-}\mathsf{con}}^{X_i^t}\right|\cdot\underline{\mathbf{x}} + \Delta\left|\mathcal{C}_{\star\text{-}\mathsf{con}}^{X_i^t}\right|\cdot\underline{\mathbf{y}} + C_1\cdot k^2\Delta\left|\mathcal{C}_{\star\text{-}\mathsf{con}}^{X_i^t}\right|^2 + C_1\\
&\leqslant \Delta L\cdot\underline{\mathbf{x}} + \Delta^2L\cdot\underline{\mathbf{y}} + C_1\cdot k^2\Delta^3L^2 + C_1.
\end{aligned}$$

注意到对于每个 $0\leqslant i\leqslant\ell$, 都有 $\chi(X_i^t, X_0^t)\leqslant\chi(X_\ell^t, X_0^t) = \chi(X_0^t)$. 因此,

$$\mathbf{E}\left[\sum_{i=0}^{\ell}\left(\chi(X_i^t, X_0^t)\cdot t_{\mathsf{var}}(X_i^t)\right)\right] \leqslant \mathbf{E}\left[(\ell+1)\cdot\chi(X_0^t)\cdot\left(\Delta L\cdot\underline{\mathbf{x}} + \Delta^2L\cdot\underline{\mathbf{y}} + C_1\cdot k^2\Delta^3L^2 + C_1\right)\right].$$

根据引理 8 中路径 $\mathsf{Path}(\sigma)$ 长度的上界, 并且考虑到当 X_0^t 满足对所有 $u\in V$ 有 $X_0^t(u)\in Q_u\cup\{\text{☆}\}$ 时, $\ell = 0$, 从而有 $\ell\leqslant kL\Delta$. 因此, 可以分别验证 $\ell > 0$, $(\ell = 0)\wedge(L = 0)$ 和 $(\ell = 0)\wedge(L > 0)$ 三种不同的情况:

$$\begin{aligned}
&(\ell+1)\cdot\chi(X_0^t)\cdot\left(\Delta L\cdot\underline{\mathbf{x}} + \Delta^2L\cdot\underline{\mathbf{y}} + C_1\cdot k^2\Delta^3L^2 + C_1\right)\\
&\quad\leqslant 2kL\Delta\cdot\chi(X_0^t)\cdot\left(\Delta L\cdot\underline{\mathbf{x}} + \Delta^2L\cdot\underline{\mathbf{y}} + C_1\cdot k^2\Delta^3L^2 + C_1\right).
\end{aligned}$$

因此,

$$\begin{aligned}
&\mathbf{E}\left[\sum_{i=0}^{\ell}\left(\chi(X_i^t, X_0^t)\cdot t_{\mathsf{var}}(X_i^t)\right)\right]\\
&\leqslant \mathbf{E}\left[2kL\Delta\cdot\chi(X_0^t)\cdot\left(\Delta L\cdot\underline{\mathbf{x}} + \Delta^2L\cdot\underline{\mathbf{y}} + C_1\cdot k^2\Delta^3L^2 + C_1\right)\right]\\
&\leqslant \mathbf{E}\left[\chi(X_0^t)\cdot\left(L^2\cdot 2k\Delta^2\cdot\underline{\mathbf{x}} + L^2\cdot 2k\Delta^3\cdot\underline{\mathbf{y}} + L^3\cdot 2C_1k^3\Delta^4 + L\cdot 2C_1k\Delta\right)\right].
\end{aligned} \tag{31}$$

令 $j \triangleq \lfloor \frac{i}{\Delta} \rfloor$, $\alpha \in \{1,2,3\}$. 我们有

$$\begin{aligned}
\mathbf{E}\left[\chi(X_0^t)\cdot L^{\alpha}\right] &= \sum_{i\geqslant 0}\left(\mathbf{Pr}\left[L=i\right]\cdot i^{\alpha}\cdot \mathbf{E}\left[\chi(X_0^t)\;\middle|\; L=i\right]\right)\\
&\leqslant \Delta^{\alpha}\sum_{i\geqslant 0}\left((j+1)^{\alpha}\cdot \mathbf{Pr}\left[L=i\right]\cdot \mathbf{E}\left[\chi(X_0^t)\;\middle|\; L=i\right]\right)\\
&\leqslant \Delta^{\alpha}\sum_{i\geqslant 0}\left((j+1)^{\alpha}\cdot \mathbf{Pr}\left[L\geqslant j\Delta\right]\cdot \mathbf{E}\left[\chi(X_0^t)\;\middle|\; L\geqslant j\Delta\right]\right)\\
&= \Delta^{\alpha+1}\sum_{j\geqslant 0}\left((j+1)^{\alpha}\cdot \mathbf{Pr}\left[L\geqslant j\Delta\right]\cdot \mathbf{E}\left[\chi(X_0^t)\;\middle|\; L\geqslant j\Delta\right]\right),
\end{aligned} \tag{32}$$

其中的不等式来自 $\chi(\cdot)$ 的非负性. 根据引理 11, 对于所有 $i\geqslant 0$, 有

$$\mathbf{Pr}\left[L\geqslant i\Delta\right]\cdot \mathbf{E}\left[\chi(X_0^t)\;\middle|\; L\geqslant i\Delta\right] \leqslant 2^{-i}. \tag{33}$$

结合式 (32), 可得

$$\mathbf{E}\left[\chi(X_0^t)\cdot L^{\alpha}\right] \leqslant \Delta^{\alpha+1}\sum_{i\geqslant 0}\left((i+1)^{\alpha}\cdot 2^{-i}\right).$$

注意到 $\sum_{i\geqslant 0}\left((i+1)\cdot 2^{-i}\right)\leqslant 4$, $\sum_{i\geqslant 0}\left((i+1)^2\cdot 2^{-i}\right)\leqslant 12$ 和 $\sum_{i\geqslant 0}\left((i+1)^3\cdot 2^{-i}\right)\leqslant 52$. 因此, 可以得到

$$\mathbf{E}\left[\chi(X_0^t)\cdot L^{\alpha}\right] \leqslant \begin{cases} 4\Delta^2, & 若\ \alpha=1,\\ 12\Delta^3, & 若\ \alpha=2,\\ 52\Delta^4, & 若\ \alpha=3. \end{cases} \tag{34}$$

结合式 (31), 可直接得到式 (27).

接下来, 证明式 (28). 由命题 5 和式 (30) 可知, 存在常数 $C_2, C_3>0$ 使得

$$\begin{aligned}
\bar{t}_{\mathsf{BF}}(X_\ell^t) &\leqslant k^2q^2\Delta^6\left(\left|\mathcal{C}_{\star\text{-}\mathsf{con}}^{X_\ell^t}\right|+1\right)(1-\mathrm{e}\alpha q)^{-\left|\mathcal{C}_{\star\text{-}\mathsf{con}}^{X_\ell^t}\right|}(C_2\cdot \underline{\mathbf{x}}+C_3kq)\\
&\leqslant k^2q^2\Delta^6\cdot(\Delta L+1)(1-\mathrm{e}\alpha q)^{-\Delta L}\cdot(C_2\cdot \underline{\mathbf{x}}+C_3kq).
\end{aligned}$$

由此可得

$$\begin{aligned}
\mathbf{E}\left[\chi(X_\ell^t, X_0^t)\cdot \bar{t}_{\mathsf{BF}}(X_\ell^t)\right] &\leqslant \mathbf{E}\left[\chi(X_0^t)k^2q^2\Delta^6\cdot(\Delta L+1)(1-\mathrm{e}\alpha q)^{-\Delta L}\cdot(C_2\cdot \underline{\mathbf{x}}+C_3kq)\right]\\
&= k^2q^2\Delta^7\cdot \mathbf{E}\left[\chi(X_0^t)(L+1)(1-\mathrm{e}\alpha q)^{-\Delta L}\right]\cdot(C_2\cdot \underline{\mathbf{x}}+C_3kq).
\end{aligned} \tag{35}$$

由式 (6) 可知 $\mathrm{e}\alpha q\leqslant(4\Delta^2)^{-1}$, 因此对于 $\Delta\geqslant 2$, 有 $(1-\mathrm{e}\alpha q)^{-\Delta^2}\leqslant 1.3$. 令 $j\triangleq\lfloor\frac{i}{\Delta}\rfloor$, 则有

$$\begin{aligned}
\mathbf{E}\left[\chi(X_0^t)(L+1)(1-\mathrm{e}\alpha q)^{-\Delta L}\right] &= \sum_{i\geqslant 0}\left(\mathbf{Pr}\left[L=i\right]\cdot \mathbf{E}\left[\chi(X_0^t)\;\middle|\; L=i\right]\cdot(i+1)\cdot(1-\mathrm{e}\alpha q)^{-i\Delta}\right)\\
&\leqslant \sum_{i\geqslant 0}\left((i+1)\cdot 1.3^{i/\Delta}\cdot \mathbf{Pr}\left[L=i\right]\cdot \mathbf{E}\left[\chi(X_0^t)\;\middle|\; L=i\right]\right)\\
&\leqslant \sum_{i\geqslant 0}\left((i+1)\cdot 1.3^{i/\Delta}\cdot \mathbf{Pr}\left[L\geqslant i\right]\cdot \mathbf{E}\left[\chi(X_0^t)\;\middle|\; L\geqslant i\right]\right)\\
&\leqslant 2\Delta\sum_{i\geqslant 0}\left((j+1)\cdot 1.3^{j}\cdot \mathbf{Pr}\left[L\geqslant j\Delta\right]\cdot \mathbf{E}\left[\chi(X_0^t)\;\middle|\; L\geqslant j\Delta\right]\right)
\end{aligned}$$

$$\leqslant 2\Delta^2 \sum_{j \geqslant 0} \left((j+1) \cdot 1.3^j \cdot \mathbf{Pr}\left[L \geqslant j\Delta\right] \cdot \mathbf{E}\left[\chi(X_0^t) \;\middle|\; L \geqslant j\Delta\right] \right),$$

其中等式由全期望法则得出, 不等式由 $\chi(\cdot)$ 的非负性保证. 结合式 (33), 可得

$$\mathbf{E}\left[\chi(X_0^t)(L+1)(1-\mathrm{e}\alpha q)^{-\Delta L}\right] \leqslant 2\Delta^2 \sum_{i \geqslant 0} \left((i+1) \cdot 0.65^i\right) \leqslant 20\Delta^2.$$

结合式 (35), 可得

$$\begin{aligned} \mathbf{E}\left[\chi(X_\ell^t, X_0^t) \cdot \bar{t}_{\mathsf{BF}}(X_\ell^t)\right] &\leqslant k^2q^2\Delta^7 \cdot 20\Delta^2 \cdot (C_2 \cdot \underline{\mathbf{x}} + C_3 kq) \\ &\leqslant 20C_2k^2q^2\Delta^9 \cdot \underline{\mathbf{x}} + 20C_3k^3q^3\Delta^9. \end{aligned}$$

由此得出式 (28), 从而完成该引理的证明.

现在可以证明定理 12, 即第 6.5 小节中的主要定理.

定理 12 的证明　定义集合

$$\begin{aligned} U &\triangleq \left\{\sigma \in \mathcal{Q}^* \;\middle|\; \mathbf{Pr}\left[X^{t-1} = \sigma\right] > 0\right\}, \\ S &\triangleq \left\{\sigma \in \mathcal{Q}^* \;\middle|\; \mathbf{Pr}\left[X^{t-1} = \sigma\right] > 0 \wedge v_t \notin V_{\mathsf{fix}}^{\sigma}\right\}. \end{aligned}$$

对于任意 $\sigma \in S$, 有 $\mathbf{Pr}\left[X^{t-1} = \sigma\right] > 0$ 且 $v_t \notin V_{\mathsf{fix}}^{\sigma}$.

因此, 根据定义 6 中对随机过程 $X^0, X^1, \ldots, X^n$ 的构造, 算法 1 以非零概率调用 $\mathsf{MarginSample}(\Phi, \sigma, v_t)$. 由引理 2 可知, 对于所有 $\sigma \in S$, 输入 (Φ, σ, v_t) 满足条件 1. 对于任意 $\sigma \in \mathcal{Q}^*$, 令 $\gamma(\sigma) = \mathbf{E}\left[H(\mathsf{Path}(\sigma))\right](0,0)$ 表示 $\mathbf{E}\left[H(\mathsf{Path}(\sigma))\right]$ 的常数项.

根据推论 3, 存在常数 $C_1, C_2 > 0$, 使得对于所有 $\sigma \in S$, 有

$$\bar{t}_{\mathsf{MS}}(\Phi, \sigma, v_t) \leqslant (1 - q_{v_t} \cdot \theta_{v_t})(\mathbf{E}\left[H(\mathsf{Path}(\sigma_{v_t \leftarrow \star}))\right] + C_1 \cdot \gamma(\sigma_{v_t \leftarrow \star})) + C_2. \tag{36}$$

而对于所有 $\sigma \in U \setminus S$, 由于 v_t 是 σ- 固定的, 依据约定可得 $\bar{t}_{\mathsf{MS}}(\Phi, \sigma, v_t) = \mathbf{E}\left[T_{\mathsf{MS}}(\Phi, \sigma, v_t)\right] = 0$. 另外, 由于 $\mathbf{E}\left[H(\mathsf{Path}(\sigma))\right]$ 总是非负的, 因此显然有

$$\bar{t}_{\mathsf{MS}}(\Phi, \sigma, v_t) \leqslant \mathbf{E}\left[H(\mathsf{Path}(\sigma))\right] + C_1 \cdot \gamma(\sigma) + C_2. \tag{37}$$

此外, 给定 $\sigma \in U$ 且 $X^{t-1} = \sigma$, 由式 (24) 可知:

- 若 v_t 是 σ- 固定的, 则 $X_0^t = \sigma$;
- 否则, 有

$$X_0^t = \begin{cases} \sigma_{v_t \leftarrow \star}, & \text{以概率 } 1 - q_{v_t} \cdot \theta_{v_t}, \\ \sigma, & \text{以概率 } q_{v_t} \cdot \theta_{v_t}. \end{cases} \tag{38}$$

因此, 若 v_t 是 σ- 固定的, 由式 (37) 及 $X_0^t = X^{t-1} = \sigma$, 有

$$\bar{t}_{\mathsf{MS}}(\Phi, X^{t-1}, v_t) \leqslant \mathbf{E}\left[H(\mathsf{Path}(X_0^t))\right] + C_1 \cdot \gamma(X_0^t) + C_2. \tag{39}$$

若 v_t 不是 σ- 固定的, 结合式 (38) 和全期望法则, 可得

$$(1 - q_{v_t} \cdot \theta_{v_t})(\mathbf{E}\left[H(\mathsf{Path}(\sigma_{v_t \leftarrow \star}))\right] + C_1 \cdot \gamma(\sigma_{v_t \leftarrow \star})) + C_2 \leqslant \mathbf{E}\left[H(\mathsf{Path}(X_0^t))\right] + C_1 \cdot \gamma(X_0^t) + C_2.$$

结合式 (36) 和 $X^{t-1}=\sigma$, 也可推导出式 (39). 综上, 对于任意可能的 $X^{t-1}=\sigma\in U$, 式 (39) 恒成立. 由全期望法则, 进一步得到

$$\mathbf{E}\left[T_{\mathsf{MS}}(\Phi,X^{t-1},v_t)\right]=\mathbf{E}\left[\bar{t}_{\mathsf{MS}}(\Phi,X^{t-1},v_t)\right]\leqslant\mathbf{E}\left[H(\mathsf{Path}(X_0^t))\right]+C_1\cdot\mathbf{E}\left[\gamma(X_0^t)\right]+C_2.$$

根据引理 13, 对于上述构造的 X_0^t, 存在常数 $C_3,C_4>0$, 使得

$$\mathbf{E}\left[H(\mathsf{Path}(X_0^t))\right]\leqslant C_3q^2k^2\Delta^9\cdot\underline{\mathbf{x}}+24k\Delta^6\cdot\underline{\mathbf{y}}+C_4q^3k^3\Delta^9.$$

由此可得

$$\mathbf{E}\left[T_{\mathsf{MS}}(\Phi,X^{t-1},v_t)\right]\leqslant C_3q^2k^2\Delta^9\cdot\underline{\mathbf{x}}+24k\Delta^6\cdot\underline{\mathbf{y}}+(C_1+1)C_4\cdot q^3k^3\Delta^9+C_2.$$

6.6 RejectionSampling 的效率分析

回顾在算法 1 中维护的随机部分赋值序列:

$$X^0,X^1,\ldots,X^n,$$

其形式化定义见定义 6.

在本节中, 我们关注最终获得的部分赋值 $X=X^n$, 即在算法 1 第 2 行中的 for 循环完成 n 次迭代后得到的部分赋值, 并将其作为输入传递给 RejectionSampling (算法 2). 为此, 首先回顾第 3.2 小节中的相关定义.

对于任意部分赋值 $\sigma\in\mathcal{Q}^*$ 及变量 $v\in V^\sigma$, 用 $H_v^\sigma=(V_v^\sigma,\mathcal{C}_v^\sigma)$ 表示超图 H^σ 中包含顶点 (变量) v 的连通分量, 其中 H^σ 是在部分赋值 σ 下由原始 CSP 公式 Φ 简化所得到的 CSP 公式 Φ^σ 的超图表示. 此外, 当 v 在 σ 中已被赋值 (即 $v\in\Lambda(\sigma)$), 规定 $H_v^\sigma=(V_v^\sigma,\mathcal{C}_v^\sigma)=(\emptyset,\emptyset)$ 为空超图.

对于部分赋值 $\sigma\in\mathcal{Q}^*$, 令 $T_{\mathsf{Rej}}(\sigma)$ 为表示运行 RejectionSampling 算法 RejectionSampling$(\Phi,\sigma,V\setminus\Lambda(\sigma))$ 的计算复杂度的随机变量, 其形式如式 (15) 所示. 我们证明如下定理, 该定理给出了在随机部分赋值 $X=X^n$ 下 $T_{\mathsf{Rej}}(X)$ 的期望的一个上界.

定理13 假设 $8\mathrm{e}p\Delta^3\leqslant0.99\alpha$, 其中 α 按照式 (6) 进行设定, 则有

$$\mathbf{E}\left[T_{\mathsf{Rej}}(X)\right]\leqslant128\mathrm{e}k\Delta^4n\cdot\underline{\mathbf{x}}+O(qk\Delta^4n),$$

其中期望是对 X 及 RejectionSampling 算法的随机性共同取的.

证明 令 $\left\{H_i^X=(V_i^X,\mathcal{C}_i^X)\;\middle|\;1\leqslant i\leqslant K\right\}$ 为算法 2 第 1 行处构造的 H^X 中的连通分量. 对于每个 $1\leqslant i\leqslant K$, 定义 $T\left(H_i^X\right)$ 为算法 2 的 repeat 循环过程中由 $H_i^X=(V_i^X,\mathcal{C}_i^X)$ 贡献的计算复杂度. 则总计算复杂度满足

$$T_{\mathsf{Rej}}(X)\leqslant\Delta n\cdot\underline{\mathbf{x}}+O(\Delta n)+\sum_{i\in[K]}T\left(H_i^X\right),\tag{40}$$

其原因如下:

• 第 1 行处的计算复杂度至多为 $n\Delta\cdot\underline{\mathbf{x}}+O(n\Delta)$, 因为该步骤至多进行 $|\mathcal{C}|\leqslant\Delta n$ 次 Eval$(\cdot)$ 调用, 从而贡献 $\Delta n\cdot\underline{\mathbf{x}}$ 这一项, 同时进行深度优先搜索以遍历所有变量和约束, 从而计算 $H_1^X,H_2^X,\ldots,H_K^X$, 其计算复杂度为 $O(\Delta n)$;

• 第 2~5 行处针对每个连通分量 H_i^X 的计算复杂度为 $T\left(H_i^X\right)$.

此外, 对于每个 $v\in V$, 定义 $T\left(H_v^X\right)$ 为满足 $v\in V_i^X$ 的 $1\leqslant i\leqslant K$ 所对应的 $T\left(H_i^X\right)$; 若不存在这样的 i (即 $v\in\Lambda(X)$ 已在 X 中赋值), 则定义 $T\left(H_v^X\right)=0$. 由此可得

$$\sum_{i\in[K]}T\left(H_i^X\right)\leqslant\sum_{v\in V}T\left(H_v^X\right).$$

结合不等式 (40), 可得

$$\mathbf{E}\left[T_{\mathsf{Rej}}(X)\right] \leqslant n\Delta \cdot \underline{\mathbf{x}} + O(n\Delta) + \sum_{v\in V} \mathbf{E}\left[T\left(H_v^X\right)\right]. \tag{41}$$

进一步证明, 对于每个 $v \in V$, 有

$$\mathbf{E}\left[T(H_v^X) \mid X\right] \leqslant \left|\mathcal{C}_v^X\right| \cdot (1-\mathrm{e}\alpha q)^{-\left|\mathcal{C}_v^X\right|} \cdot \underline{\mathbf{x}} + O\left(kq \cdot \left(\left|\mathcal{C}_v^X\right| + 1\right) \cdot (1-\mathrm{e}\alpha q)^{-\left|\mathcal{C}_v^X\right|}\right). \tag{42}$$

对于退化情况 $H_v^X = (\emptyset, \emptyset)$, 结论显然成立. 接下来, 假设 $H_v^X \neq (\emptyset, \emptyset)$.

众所周知, 在算法 2 第 5 行处, 拒绝采样的成功次数的期望值等于 $\mathcal{P}_{H_v^X}[\Omega_{H_v^X}]^{-1}$, 其中 $\Omega_{H_v^X}$ 表示 CSP 公式 H_v^X 的解的集合, 因此 $\mathcal{P}_{H_v^X}[\Omega_{H_v^X}]$ 表示在 H_v^X 关联的 CSP 公式 Φ_v^X 上, 均匀随机赋值 $\sigma \in \mathcal{Q}_{V_v^X}$ 是可满足的概率. 根据定理 6 和引理 3, 可得

$$\mathcal{P}_{H_v^X}\left[\Omega_{H_v^X}\right] \geqslant (1-\mathrm{e}\alpha q)^{\left|\mathcal{C}_v^X\right|}.$$

因此, 期望进行 $(1-\mathrm{e}\alpha q)^{-\left|\mathcal{C}_v^X\right|}$ 轮迭代才能成功采样 V_v^X 的赋值. 而在每次迭代中, 最多进行 $\left|\mathcal{C}_v^X\right|$ 次 $\mathsf{Eval}(\cdot)$ 调用, 且时间复杂度为 $O\left(qk(\left|\mathcal{C}_v^X\right|+1)\right)$. 这证明了式 (42) 成立.

接下来, 对式 (42) 关于 X 的期望进行估计. 设 $\ell \triangleq \lfloor t/(2\Delta^2)\rfloor$. 由式 (6), 可得 $\mathrm{e}\alpha q \leqslant (2\Delta^2)^{-1}$, 从而

$$(1-\mathrm{e}\alpha q)^{-(\ell+1)\Delta^2} \leqslant 2^{\ell+1}.$$

进一步计算, 可得

$$\begin{aligned}
&\sum_{t\geqslant 0}\left((t+1)(1-\mathrm{e}\alpha q)^{-t}\,\mathbf{Pr}\left[\left|\mathcal{C}_v^X\right| = t\right]\right)\\
&\quad\leqslant 2\Delta^2 \sum_{t\geqslant 0}\left(\mathbf{Pr}\left[\left|\mathcal{C}_v^X\right| \geqslant 2\ell\Delta^2\right]\cdot(\ell+1)\cdot(1-\mathrm{e}\alpha q)^{-(\ell+1)\Delta^2}\right)\\
&\quad\leqslant 4\Delta^2 \sum_{t\geqslant 0}\left((\ell+1)\cdot 2^{\ell}\cdot \mathbf{Pr}\left[\left|\mathcal{C}_v^X\right| \geqslant 2\ell\Delta^2\right]\right)\\
&\quad= 4\Delta^4 \sum_{\ell\geqslant 0}\left((\ell+1)\cdot 2^{\ell}\cdot \mathbf{Pr}\left[\left|\mathcal{C}_v^X\right| \geqslant 2\ell\Delta^2\right]\right)\\
&\quad\leqslant 32\mathrm{e}k\Delta^4 \sum_{\ell\geqslant 0}\left((\ell+1)\cdot 2^{-\ell}\right) \quad (\text{引理 } 12)\\
&\quad\leqslant 128\mathrm{e}k\Delta^4.
\end{aligned}$$

因此,

$$\mathbf{E}\left[\left(\left|\mathcal{C}_v^X\right|+1\right)\cdot(1-\mathrm{e}\alpha q)^{-\left|\mathcal{C}_v^X\right|}\right] = \sum_{t\geqslant 0}\left((t+1)(1-\mathrm{e}\alpha q)^{-t}\,\mathbf{Pr}\left[\left|\mathcal{C}_v^X\right| = t\right]\right) \leqslant 128\mathrm{e}k\Delta^4.$$

由此, 存在常数 $C > 0$, 使得式 (42) 的期望满足

$$\mathbf{E}\left[T(H_v^X)\right] \leqslant 128\mathrm{e}k\Delta^4 \cdot \underline{\mathbf{x}} + 128\mathrm{e}Ck^2q\Delta^4.$$

结合式 (41), 定理得证.

6.7　主采样算法的效率分析

现在证明定理 10 和 11.

定理 10 的证明　假设式 (3) 所述的 LLL 条件成立, 则对于式 (6) 中设定的 α, 有 $8\mathrm{e}p\Delta^3 \leqslant 0.99\alpha$, 这正是定理 12 和 13 所要求的参数范围.

算法 1 主要包含以下非平凡的计算代价.

• 第 1 行中的初始化操作, 以及对每个 $1 \leqslant t \leqslant n$, 在第 3 行处检查 v_t 是否为 X^{t-1}- 固定的. 这些操作的总计算代价为

$$\Delta n \cdot \underline{\mathbf{y}} + O(\Delta n),$$

这是因为每个 v_t 需要调用 $\mathsf{Frozen}(\cdot)$ 至多 $|\{c \in \mathcal{C} \mid v_t \in \mathsf{vbl}(c)\}| \leqslant \Delta$ 次, 同时需要 $O(\Delta)$ 计算量来检索所有满足 $v_t \in \mathsf{vbl}(c)$ 的约束 $c \in \mathcal{C}$.

• 在第 4 行处调用 $\mathsf{MarginSample}(\Phi, X^{t-1}, v_t)$. 根据定理 12, 其总计算代价为

$$\sum_{t=1}^{n} \mathbf{E}\left[T_{\mathsf{MS}}(\Phi, X^{t-1}, v_t)\right] \leqslant O\left(q^2k^2\Delta^9 n\right) \cdot \underline{\mathbf{x}} + 24k\Delta^6 n \cdot \underline{\mathbf{y}} + O\left(q^3k^3\Delta^9 n\right).$$

• 在第 5 行处最终调用 $\mathsf{RejectionSampling}(\Phi, X^n, V \setminus \Lambda(X^n))$, 其计算代价由定理 13 给出:

$$\mathbf{E}\left[T_{\mathsf{Rej}}(X^n)\right] \leqslant 128\mathrm{e}k\Delta^4 n \cdot \underline{\mathbf{x}} + O\left(qk\Delta^4 n\right).$$

因此, 算法 1 的总期望复杂度上界为

$$O\left(q^2k^2\Delta^9 n\right) \cdot \underline{\mathbf{x}} + 24k\Delta^6 n \cdot \underline{\mathbf{y}} + O\left(q^3k^3\Delta^9 n\right), \tag{43}$$

定理得证.

接下来, 构造用于近似采样的算法 1′, 并证明定理 11. 给定一个 CSP 公式 $\Phi = (V, \mathcal{Q}, \mathcal{C})$, 其中变量数 $n = |V|$, 以及误差界 $\varepsilon \in (0,1)$, 算法 1′ 执行以下步骤. 参数设定如下:

$$\delta = 0.005 \quad 且 \quad N = \left\lceil \frac{\ln\left(200k\Delta^6 n\varepsilon^{-2}\right)}{0.33\alpha\delta^2} \right\rceil. \tag{44}$$

算法 1′ 运行算法 1, 但将其中的 $\mathsf{Frozen}(\cdot)$ 实现为如下的蒙特卡罗子程序.

• 对于任意约束 $c \in \mathcal{C}$ 和任意部分赋值 $\sigma \in \mathcal{Q}^*$, 重复以下过程 N 次:

– 在 $\mathsf{vbl}(c)$ 上均匀随机生成一个满足 σ 的赋值 $Y \in \mathcal{Q}_{\mathsf{vbl}(c)}$;

– 通过调用 $\mathsf{Eval}(c, Y)$ 检测是否有 $c(Y) = \mathtt{True}$.

• 设 Z 为 N 次试验中 $c(Y) = \mathtt{False}$ 出现的次数, 并返回指示变量 $\mathbb{1}[Z/N > 0.995\alpha]$.

此外, 应用标准的记忆化 (memoization) 技巧, 以确保假设 2 所需要的冻结判定器 $\mathsf{Frozen}(\cdot)$ 的一致性. 每个约束 $c \in \mathcal{C}$ 维护一个确定性动态字典 Dic_c, 用于存储形如 (τ, ans) 的键 - 值对, 其中 $\tau \in \bigotimes_{v \in \mathsf{vbl}(c)}(Q_v \cup \{\star, \text{☆}\})$, $ans \in \{0,1\}$. 每当调用 $\mathsf{Frozen}(c, \sigma)$ 时, 首先在 Dic_c 中查找键 $\sigma_{\mathsf{vbl}(c)}$. 如果找到存储的 $ans \in \{0,1\}$, 则直接返回 ans 作为 $\mathsf{Frozen}(c, \sigma)$ 的结果; 否则, 使用上述蒙特卡罗子程序计算 $ans = I[Z/N > 0.995\alpha] \in \{0,1\}$, 并将键 - 值对 $(\sigma_{\mathsf{vbl}(c)}, ans)$ 插入 Dic_c, 然后返回 ans 作为结果. 利用字典树 (trie) 结构实现 Dic_c, 每次查询或更新的计算复杂度为 $O(k)$. 最终得到的算法即为算法 1′.

定理 11 的证明　在算法 1′ 中, 算法 1 中对冻结判定器 $\mathsf{Frozen}(\cdot)$ 的每一次调用均通过对满足性判定器 $\mathsf{Eval}(\cdot)$ 的 N 次调用来实现, 并伴随 $O(kN)$ 的计算复杂度, 其中 N 按照式 (44) 设定. 根据定理 10, 算法 1 的期望时间复杂度满足式 (43) 的上界. 用

$$\underline{\mathbf{y}} \leftarrow N \cdot \underline{\mathbf{x}} + O(kN)$$

替换后, 算法 1′ 的期望时间复杂度满足以下上界:

$$O\left(q^2k^2\Delta^9 n \log\left(\frac{\Delta n}{\varepsilon}\right) \cdot \underline{\mathbf{x}} + q^3k^3\Delta^9 n \log\left(\frac{\Delta n}{\varepsilon}\right)\right).$$

利用切尔诺夫界 (Chernoff bound), 可以验证, 对于每一次对 $\mathsf{Frozen}(c,\sigma)$ 的调用, 当 $\mathcal{P}[\neg c \mid \sigma] > \alpha$ 和 $\mathcal{P}[\neg c \mid \sigma] \leqslant 0.99\alpha$ 这两种情况时, 发生错误的概率有如下上界:

$$2\exp\left(-\frac{\delta^2}{3}0.99\alpha N\right) \leqslant \frac{\varepsilon}{2M}, \quad \text{其中 } M \triangleq 50k\Delta^6 n\varepsilon^{-1}.$$

由式 (43) 可知, 算法 1 在期望上对 $\mathsf{Frozen}(\cdot)$ 的调用次数至多为 $24k\Delta^6 n$, 从而有 $24k\Delta^6 n \leqslant \frac{\varepsilon M}{2}$. 根据马尔可夫不等式, 算法 1 对 $\mathsf{Frozen}(\cdot)$ 的调用次数超过 M 的概率至多为 $\frac{\varepsilon}{2}$. 结合并集界 (union bound), 可知以至少以 $1-\varepsilon$ 的概率, 所有对 $\mathsf{Frozen}(\cdot)$ 的调用均无误. 因此, 通过算法 1 与 $1'$ 之间的耦合, 可以保证这两种算法在相同输入 CSP 公式 Φ 上的输出的全变差距离至多为 ε.

接下来形式化地重述定理 3 和 4. 其中, 符号 $\tilde{O}(\cdot)$ 隐藏了对数多项式 (polylogarithmic) 因子. 该算法使用算法 3 中作为子程序的 $\mathsf{MarginSample}(\Phi, \star^V, v)$ 的蒙特卡罗方法来从边缘分布 μ_v 中进行采样.

定理14　针对边缘采样和概率推断, 存在以下算法: 对于给定的输入参数 $\varepsilon, \delta \in (0,1)$, CSP 公式 $\Phi = (V, \mathcal{Q}, \mathcal{C})$ (满足式 (3)) 以及变量 $v \in V$, 该算法具有如下性质.

• 用于边缘采样的算法返回一个随机值 $x \in Q_v$, 其分布与 μ_v 的全变差距离至多为 ε, 其期望调用 $\mathsf{Eval}(\cdot)$ 的次数至多为 $O\left(q^2k^2\Delta^9\log(k\Delta/\varepsilon)\right)$, 计算复杂度至多为 $O\left(q^3k^3\Delta^9\log(k\Delta/\varepsilon)\right)$.

• 用于推理的算法返回 $\hat{\mu}_v \in [0,1]^{Q_v}$, 其调用 $\mathsf{Eval}(\cdot)$ 的次数至多为 $\tilde{O}\left(q^3k^2\Delta^9\varepsilon^{-2}\log(1/\delta)\right)$, 计算复杂度至多为 $\tilde{O}\left(q^4k^3\Delta^9\varepsilon^{-2}\log(1/\delta)\right)$, 并满足如下概率保证:

$$\mathbf{Pr}\left[\forall\, x \in Q_v: (1-\varepsilon)\mu_v(x) \leqslant \hat{\mu}_v(x) \leqslant (1+\varepsilon)\mu_v(x)\right] \geqslant 1-\delta.$$

证明　当式 (3) 成立时, 由式 (6) 中 α 的设定可知 $8\mathrm{e}p\Delta^3 \leqslant 0.99\alpha$. 因此, $(\Phi, \star^V, v)$ 满足条件 1. 根据定理 9 和 12, $\mathsf{MarginSample}(\Phi, \star^V, v)$ 能够以期望复杂度

$$O\left(q^2k^2\Delta^9\right)\cdot \underline{\mathbf{x}} + 24k\Delta^6 \cdot \underline{\mathbf{y}} + O\left(q^3k^3\Delta^9\right)$$

生成来自 μ_v 的完美采样. 采用与定理 11 的证明中相同的蒙特卡罗模拟方法来估计 $\mathsf{Frozen}(\cdot)$, 但在本次分析中选取更大的参数

$$N = O\left(\frac{1}{\alpha}\log(k\Delta/\varepsilon)\right) = O\left(q^2k\Delta^2\log(k\Delta/\varepsilon)\right),$$

通过替换 $\underline{\mathbf{y}} \leftarrow N\cdot\underline{\mathbf{x}} + O(kN)$, $\mathsf{MarginSample}(\Phi, \star^V, v)$ 被转化为一个对 μ_v 的近似采样算法, 其与 μ_v 的全变差距离至多为 $\varepsilon/2$, 且其复杂度为

$$O\left(q^2k^2\Delta^9\log(k\Delta/\varepsilon)\right)\cdot\underline{\mathbf{x}} + O\left(q^3k^3\Delta^9\log(k\Delta/\varepsilon)\right).$$

至此, 算法中关于边缘采样部分的证明得以完成.

假设式 (3) 成立, 由推论 1 所述的局部均匀性性质可得, 对于所有 $x \in Q_v$, 有 $\mu_v(x) > \frac{1}{2q}$. 因此, 利用切尔诺夫界, 可以通过 $O(q/\varepsilon^2)$ 次近似采样来估计每个 $\mu_v(x)$ $(x \in Q_v)$, 并使其估计精度在 $1\pm\varepsilon$ 的相对误差范围内, 且成功概率大于 0.9. 期望总计算复杂度为

$$\tilde{O}\left(q^3k^2\Delta^9\varepsilon^{-2}\right)\cdot\underline{\mathbf{x}} + \tilde{O}\left(q^4k^3\Delta^9\varepsilon^{-2}\right). \tag{45}$$

根据马尔可夫不等式, 总计算复杂度超过此上界的概率至多为 0.1. 通过截断 (truncation), 可构造一个固定计算复杂度不超过式 (45) 的算法, 从而保证对于每个 $x \in Q_v$, 该算法以概率至少为 0.8 估计 $\mu_v(x)$ 的值, 并使其精度处于 $1\pm\varepsilon$ 的相对误差范围内. 最终, 满足定理要求的算法可以通过重复上述过程 $O(\log\frac{q}{\delta})$ 次, 并应用中值技巧 (median trick) 来构造.

7　广义 $\{2,3\}$- 树

本节证明引理 11 和 12 所述的尾界, 以此完成本文的所有证明. 这两个尾界基本上表明, 长运行时间的大证据 (witness) 出现的概率呈指数级下降, 它们均通过一种新的组合结构 "广义 $\{2,3\}$- 树" (generalized $\{2,3\}$-tree) 来证明.

广义 $\{2,3\}$- 树是对此前已在文献 [6,11] 中被使用的 $\{2,3\}$- 树的推广. 不同于 $\{2,3\}$- 树对变量和约束的相同处理方式, 广义 $\{2,3\}$- 树对 "坏" 变量和 "坏" 约束进行了区分处理, 因为这两类坏事件的密度不同. 给定一个超图 $H=(V,\mathcal{E})$, 定义 $\mathsf{Lin}\,(H)$ 为 H 的线图, 其顶点集为 $\mathcal{E}$ 中的超边, 两条超边相邻当且仅当它们在 H 中共享至少一个顶点. 令 $\mathrm{dist}_{\mathsf{Lin}(H)}(\cdot,\cdot)$ 表示 $\mathsf{Lin}\,(H)$ 中的最短路径距离.

定义12 (广义 $\{2,3\}$- 树)　给定超图 $H=(V,\mathcal{E})$, 设 $T=U\uplus E$, 其中 $U\subseteq V$ 且 $E\subseteq\mathcal{E}$, 定义 $G(T)$ 为构造于 T 之上的有向图, 使得对于任意 $u,v\in T$, 从 u 到 v 存在一条有向边当且仅当以下至少一个条件时被满足:

- $u,v\in E$, 且 $\mathrm{dist}_{\mathsf{Lin}(H)}(u,v)=2$ 或 3;
- $u\in U$, $v\in E$, 且存在 $e\in\mathcal{E}$ 使得 $u\in e$ 且 $\mathrm{dist}_{\mathsf{Lin}(H)}(v,e)=1$;
- $u\in E$, $v\in U$, 且存在 $e\in\mathcal{E}$ 使得 $v\in e$ 且 $\mathrm{dist}_{\mathsf{Lin}(H)}(u,e)=1$ 或 2;
- $u,v\in U$, 且存在 $e\in\mathcal{E}$ 使得 $u,v\in e$.

若 T 满足以下条件, 则称其为 H 的一个广义 $\{2,3\}$- 树:

(1) 对于所有不同的 $u,v\in E$, 有 $\mathrm{dist}_{\mathsf{Lin}(H)}(u,v)\geqslant 2$;

(2) $G(T)$ 具有一个有根有向生成树 (rooted directed spanning tree).

进一步地, 若 $v\in T$ 是 $G(T)$ 某棵有根有向生成树的根, 则称 v 为 T 的一个根. 对于任意 $v\in V$ 及整数 $r,\ell,t\geqslant 0$, 定义

$$\mathscr{T}_v^{r,\ell}(H)\triangleq\left\{H\text{ 的广义 }\{2,3\}\text{- 树 }T\ \middle|\ (v\text{ 是 }T\text{ 的一个根})\wedge(|T\cap V|=r)\wedge(|T\cap\mathcal{E}|)=\ell\right\},$$

$$\mathscr{T}_v^{t}(H)\triangleq\left\{H\text{ 的广义 }\{2,3\}\text{- 树 }T\ \middle|\ (v\text{ 是 }T\text{ 的一个根})\wedge(|T\cap V|=r+\Delta\cdot|T\cap\mathcal{E}|=t)\right\}.$$

若上下文中超图 H 明确, 我们将 $\mathscr{T}_v^{t}$ 和 $\mathscr{T}_v^{r,\ell}$ 分别简记为 $\mathscr{T}_v^{t}(H)$ 和 $\mathscr{T}_v^{r,\ell}(H)$.

定义 12 中的广义 $\{2,3\}$- 树受到文献 [28] 中关于线图 $\mathsf{Lin}\,(H)$ 的 $\{2,3\}$- 树概念的启发. 我们将这一概念推广到原始超图 H, 以同时刻画 H 中顶点与超边之间的距离关系. 此外, 我们允许在某些广义 $\{2,3\}$- 树 T 中包含 H 的顶点, 前提是所有包含的顶点和超边彼此足够接近. 可以验证, 线图 $\mathsf{Lin}\,(H)$ 中的每棵 $\{2,3\}$- 树也为超图 H 中的一棵广义 $\{2,3\}$- 树. 具体而言, 当定义 12 中的超图 H 为某个 CSP Φ 的关联超图 $H_\Phi=(V,\mathcal{C})$ 时, H_Φ 中的广义 $\{2,3\}$- 树 $T\subseteq V\uplus\mathcal{C}$ 可视为变量和约束的子集.

接下来证明引理 11 和 12. 如第 6.4 小节所述, 这两个引理中的尾界是通过利用广义 $\{2,3\}$- 树对两个 "基本" 尾界进行增强而得出的. 这两个基本尾界涉及 "坏" 变量及互不相交的 "坏" 约束的出现概率. 回顾定义 11 中的坏事件 $\mathcal{E}_T^t$. 以下引理用于证明引理 11, 它在 T 中的约束互不相交的情况下, 为坏事件 $\mathcal{E}_T^t$ 提供了尾界. 该引理表明, 在 LLL 条件约束的范围内, 可能导致 MarginSample 低效的坏事件 $\mathcal{E}_T^t$ 发生的概率极小.

引理14　假设 $8\mathrm{e}p\Delta^3\leqslant 0.99\alpha$, 其中 α 由式 (6) 定义. 设 $1\leqslant t\leqslant n$, 并且 $(X^0,X^1,\ldots,X^{t-1},X_0^t,X_1^t,\ldots,X_\ell^t)$ 依照式 (24) 生成. 对于任意变量与约束的子集 $T=U\uplus E$, 其中 E 中的约束互不相交, 有

$$\mathbf{Pr}\left[\mathcal{E}_T^t\right]\cdot\mathbf{E}\left[\chi(X_0^t)\ \middle|\ \mathcal{E}_T^t\right]\leqslant(8\mathrm{e}k\Delta)^{-|U|}\cdot\left(4\mathrm{e}\Delta^3\right)^{-|E|}. \tag{46}$$

下述引理用于证明引理 12. 它对可能导致 RejectionSampling 低效的一组互不相交约束的坏事件提供了另一个尾界.

引理15　假设 $8\mathrm{e}p\Delta^3\leqslant 0.99\alpha$. 设 $(X^0,X^1,\ldots,X^n)=\mathsf{Simulate}(n)$. 对于任意一组互不相交的约束 $T\subseteq\mathcal{C}$, 有

$$\mathbf{Pr}\left[T\subseteq\mathcal{C}_{\mathsf{frozen}}^{X^n}\right]\leqslant\left(4\mathrm{e}\Delta^3\right)^{-|T|}.$$

这两个引理构成了基本的尾界, 并在补充材料中详细证明. 在本节的剩余部分, 我们在假设引理 14 和 15 成立的前提下, 证明 11 和 12. 为了对 $\mathscr{T}_v^t$ 中广义 $\{2,3\}$- 树的权重和进行约束, 我们需要以下引理.

引理16　设 $H=(V,\mathcal{E})$ 为一个超图, 其中每条超边最多包含 k 个顶点, 并且每个顶点最多与 Δ 条超边相交. 对于任意 $v\in V$ 及 $t>0$, 有

$$\sum_{T\in\mathscr{T}_v^t}\left((8\mathrm{e}k\Delta)^{-|T\cap V|}(4\mathrm{e}\Delta^3)^{-|T\cap\mathcal{E}|}\right)\leqslant 2^{-\lfloor t/\Delta\rfloor-1}\cdot\Delta^{-1}.$$

从现在起, 记 $w_1\triangleq(8\mathrm{e}k\Delta)^{-1}$ 和 $w_2\triangleq(4\mathrm{e}\Delta^3)^{-1}$. 以下定义用于引理 16 的证明, 其灵感来源于 H 的每个超边至多包含 k 个顶点, 并且与至多 Δ 条超边共享顶点的事实.

定义13 ($\{a,b\}$- 标记树)　一个 $\{a,b\}$- 标记树是一棵有根有向树, 其满足如下结论.

- 每个节点的标签为 a 或 b;
- 任何标签为 a 的节点具有权重 w_1x, 而任何标签为 b 的节点具有权重 w_2x^{Δ}. 对于任意 $\{a,b\}$- 标记树 T, 定义 T 的权重为 T 中所有节点的权重的乘积. 稍滥用记号, 记 $a(T)$ 和 $b(T)$ 分别表示树 T 中标签为 a 和 b 的节点数目. 给定 $z\in\{a,b\}$, 定义一棵具有无限节点的树 $\mathcal{T}_z$, 其构造如下:
- $\mathcal{T}_z$ 是一棵 $\{a,b\}$- 标记树, 其根节点标签为 z;
- 每个标签为 a 的节点有 $k\Delta$ 个子节点标记为 a, 并且有 Δ^2 个子节点标记为 b;
- 每个标签为 b 的节点有 $k\Delta^2$ 个子节点标记为 a, 并且有 Δ^3 个子节点标记为 b.

$\mathcal{T}_z$ 的合法子树 (proper subtree) T 是 $\mathcal{T}_z$ 的一个有根有向子树, 且其根节点与 $\mathcal{T}_z$ 的根节点相同. 显然, T 也是一棵 $\{a,b\}$- 标记树. 定义

$$\mathcal{T}_z^t\triangleq\left\{\mathcal{T}_z\text{ 的合法子树 }T\ \middle|\ a(T)+\Delta\cdot b(T)=t\right\}.$$

引理 16 的证明　对于每个 $v\in V$, 我们声称存在一个从 $\mathscr{T}_v^t$ 中的广义 $\{2,3\}$- 树到 $\mathcal{T}_a^t$ 中的 $\{a,b\}$- 标记树的单射 (injection), 使得每个 $T\in\mathscr{T}_v^t$ 被映射到某个满足 $|T\cap V|=a(T')$ 且 $|T\cap\mathcal{E}|=b(T')$ 的 $T'\in\mathcal{T}_a^t$. 回顾定义 $w_1=(8\mathrm{e}k\Delta)^{-1}$ 和 $w_2=(4\mathrm{e}\Delta^3)^{-1}$, 有

$$\sum_{T\in\mathscr{T}_v^t}\left((8\mathrm{e}k\Delta)^{-|T\cap V|}(4\mathrm{e}\Delta^3)^{-|T\cap\mathcal{E}|}\right)\leqslant\sum_{T\in\mathcal{T}_a^t}w_1^{a(T)}w_2^{b(T)}.$$

因此, 为了证明该引理, 只需要证明

$$\sum_{T\in\mathcal{T}_a^t}w_1^{a(T)}w_2^{b(T)}\leqslant 2^{-\lfloor t/\Delta\rfloor-1}\cdot\Delta^{-1}.\tag{47}$$

令 f_1 表示所有 $\mathcal{T}_a$ 的合法子树的权重之和. 类似地, 令 f_2 表示所有 $\mathcal{T}_b$ 的合法子树的权重之和. 根据定义 13, 可以验证

$$\begin{aligned}f_1&=w_1x(1+f_1)^{k\Delta}\cdot(1+f_2)^{\Delta^2},\\ f_2&=w_2x^{\Delta}(1+f_1)^{k\Delta^2}\cdot(1+f_2)^{\Delta^3}.\end{aligned}$$

此外, 记 $[x^m]f_1(x)$ 为 f_1 中 x^m 的系数. 根据定义 13, 也有

$$\forall\, m\geqslant 0,\quad [x^m]f_1(x)=\sum_{T\in\mathcal{T}_a^m}w_1^{a(T)}w_2^{b(T)}.\tag{48}$$

定义

$$h\triangleq x(1+f_1)^{k\Delta}\cdot(1+f_2)^{\Delta^2}.$$

于是有 $f_2 = w_2 h^{\Delta}$, $f_1 = w_1 h$, 并且

$$h = x(1+w_1h)^{k\Delta} \cdot \left(1+w_2h^{\Delta}\right)^{\Delta^2}.$$

应用拉格朗日反演定理 (Lagrange inversion theorem), 对于每个 $m \geqslant 1$, 得到

$$\begin{aligned}[x^m]h(x) &= \frac{1}{m}[u^{m-1}]((1+w_1u)^{k\Delta} \cdot (1+w_2u^{\Delta})^{\Delta^2})^m \\ &= \frac{1}{m}\sum_{i=0}^{\lfloor m/\Delta \rfloor}([u^{m-1-\Delta i}](1+w_1u)^{k\Delta m} \cdot [u^{\Delta i}](1+w_2u^{\Delta})^{\Delta^2 m}).\end{aligned} \tag{49}$$

假设 $m \geqslant 3$ 且 $0 \leqslant i \leqslant \lfloor \frac{m}{2\Delta} \rfloor$, 则有 $m \leqslant 4(m-1-\Delta i)$. 此外, 对于任意满足 $0 < \gamma \leqslant \beta$ 的整数 γ 和 β,

$$\binom{\beta}{\gamma} \leqslant \left(\frac{\mathrm{e}\beta}{\gamma}\right)^{\gamma} \tag{50}$$

成立, 因此有

$$\begin{aligned}(1+w_1u)^{k\Delta m} &= \binom{k\Delta m}{m-1-\Delta i} w_1^{m-1-\Delta i} \\ &\leqslant \left(\frac{\mathrm{e}k\Delta m}{m-1-\Delta i}\right)^{m-1-\Delta i} w_1^{m-1-\Delta i} \quad (\text{式 } (50)) \\ &\leqslant (4\mathrm{e}k\Delta w_1)^{m-1-\Delta i} \quad (m \leqslant 4(m-1-\Delta i)) \\ &= 2^{1+\Delta i - m} \quad \left(w_1 = (8\mathrm{e}k\Delta)^{-1}\right).\end{aligned} \tag{51}$$

类似地, 假设 $m \geqslant 3$ 且 $\lfloor \frac{m}{2\Delta} \rfloor < i \leqslant m$, 则有 $m \leqslant 2\Delta i$. 因此有

$$\begin{aligned}\left(1+w_2u^{\Delta}\right)^{\Delta^2 m} &= \binom{\Delta^2 m}{i} w_2^i \\ &\leqslant \left(\frac{\mathrm{e}\Delta^2 m}{i}\right)^i w_2^i \quad (\text{式 } (50)) \\ &\leqslant (2\mathrm{e}\Delta^3 w_2)^i \quad (m \leqslant 2\Delta i) \\ &= 2^{-i} \quad \left(w_2 = (4\mathrm{e}\Delta^3)^{-1}\right).\end{aligned} \tag{52}$$

结合式 (49), (51) 和 (52), 若 $m \geqslant 3$, 则有

$$[x^m]h(x) \leqslant m^{-1}\left(\sum_{i=0}^{\lfloor m/\Delta \rfloor} 2^{1-m+\Delta i - i}\right) \leqslant 2^{1-\lfloor m/\Delta \rfloor}.$$

对于 $m=1$ 和 $m=2$ 的情形, 可以直接根据式 (49) 验证 $[x^m]h(x) \leqslant 2^{1-\lfloor m/\Delta \rfloor}$. 因此, 由 $f_1 = w_1 h$ 及 $w_1 = (8\mathrm{e}k\Delta)^{-1}$, 得到

$$[x^m]f_1(x) = w_1[x^m]h(x) \leqslant 2^{-\lfloor m/\Delta \rfloor - 1} \cdot \Delta^{-1}.$$

结合式 (48), 式 (47) 立即得证.

接下来, 给出从广义 $\{2,3\}$- 树 $\mathscr{T}_v^t$ 到 $\{a,b\}$- 标记树 $\mathcal{T}_a^t$ 的单射构造, 从而证明引理成立. 给定整数 $r, \ell \geqslant 0$ 及 $z \in \{a,b\}$, 定义

$$\mathcal{T}_z^{r,\ell} \triangleq \left\{\mathcal{T}_z \text{ 的合法子树 } T \;\middle|\; (a(T)=r) \wedge (b(T)=\ell)\right\}.$$

为构造使得每个 $T \in \mathscr{T}_v^t$ 映射到某个满足 $|T \cap V| = a(T')$ 且 $|T \cap \mathcal{E}| = b(T')$ 的 $T' \in \mathcal{T}_a^t$ 的从 $\mathscr{T}_v^t$ 到 $\mathcal{T}_a^t$ 的单射, 只需要对任意整数 $r, \ell \geqslant 0$, 构造从 $\mathscr{T}_v^{r,\ell}$ 到 $\mathcal{T}_a^{r,\ell}$ 的单射.

接下来, 构造从 $\mathscr{T}_v^{r,\ell}$ 到 $\mathcal{T}_a^{r,\ell}$ 的单射. 给定广义 $\{2,3\}$- 树 $T \in \mathscr{T}_v^{r,\ell}$, 由定义 12 知 $G(T)$ 具有以 v 为根的有向生成树. 任意选择这样的生成树 $\phi(T)$. 显然, 对于不同的 $T, T' \in \mathscr{T}_v^t$, 有 $\phi(T) \neq \phi(T')$, 从而 ϕ 是从 $\mathscr{T}_v^{r,\ell}$ 到 $\left\{\phi(T) \,\middle|\, T \in \mathscr{T}_v^{r,\ell}\right\}$ 的单射. 对于 $\phi(T)$ 中的任意节点 u, 有 $u \in V$ 或 $u \in \mathcal{E}$, 且由 $T \in \mathscr{T}_v^{r,\ell}$, 有 $|T \cap V| = r, |T \cap \mathcal{E}| = \ell$. 结合 $\phi(T)$ 为 $G(T)$ 的生成树, 其节点集即为 T, 因此 $\big|\{\phi(T)$ 中标签属于 V 的节点$\}\big| = r$, $\big|\{\phi(T)$ 中标签属于 $\mathcal{E}$ 的节点$\}\big| = \ell$.

给定超图 $H = (V, \mathcal{E})$, 对于任意 $u \in V$, 定义 $V(u) \subseteq V$ 为满足以下条件的顶点集合: 存在超边 $e \in \mathcal{E}$ 使得 $u, u' \in e$. 同时, 定义 $\mathcal{E}(u) \subseteq \mathcal{E}$ 为满足以下条件的超边集合: 存在 $e' \in \mathcal{E}$ 使得 $u \in e'$ 且 $\mathrm{dist}_{\mathsf{Lin}(H)}(e', e) = 1$. 类似地, 对于任意 $e \in \mathcal{E}$, 定义 $V(e) \subseteq V$ 为满足以下条件的顶点集合: 存在超边 $e' \in \mathcal{E}$ 使得 $u \in e'$ 且 $\mathrm{dist}_{\mathsf{Lin}(H)}(e, e') = 1$ 或 2. 同时, 定义 $\mathcal{E}(e) \subseteq \mathcal{E}$ 为满足 $\mathrm{dist}_{\mathsf{Lin}(H)}(e, e') = 2$ 或 3 的超边集合. 此外, 回顾超图 H 的性质, 每条超边至多包含 k 个顶点, 并且每个超边与至多 Δ 条超边共享顶点. 对于任意 $u \in V$ 和 $e \in \mathcal{E}$, 有 $|V(u)| \leqslant k\Delta$, $|\mathcal{E}(u)| \leqslant \Delta^2$, $|V(e)| \leqslant k\Delta^2$, $|\mathcal{E}(e)| \leqslant \Delta^3$.

令 T_v 为一个以 v 为根的无限有向树, 定义如下.

• 每个节点的标签来自 $V \uplus \mathcal{E}$, 且根节点的标签为 v.

• 对于任意 $u \in V \uplus \mathcal{E}$ 和 $u' \in V(u) \uplus \mathcal{E}(u)$, 每个标记为 u 的节点恰有一个子节点标记为 u'. 给定整数 $r, \ell \geqslant 0$, 定义 $\mathsf{T}_v^{r,\ell}$ 为 T_v 的子树集合, 其中的子树 T 满足:

• T 的根与 T_v 的根相同;

• T 中标记属于 V 的节点数为 r, 标记属于 $\mathcal{E}$ 的节点数为 ℓ.

进一步地, 由 T_v 的定义可验证:

• T_v 的根的标签属于 V;

• 若某个节点的标签为 $u \in V$, 则它有至多 $|V(u)| \leqslant k\Delta$ 个子节点的标签属于 V, 且至多 $|\mathcal{E}(u)| \leqslant \Delta^2$ 个子节点的标签属于 $\mathcal{E}$;

• 若某个节点的标签为 $u \in \mathcal{E}$, 则它有至多 $|V(u)| \leqslant k\Delta^2$ 个子节点的标签属于 V, 且至多 $|\mathcal{E}(u)| \leqslant \Delta^3$ 个子节点的标签属于 $\mathcal{E}$.

结合集合 $\mathcal{T}_a$, $\mathcal{T}_a^{r,\ell}$ 和 $\mathsf{T}_v^{r,\ell}$ 的定义, 可以验证 $\left|\mathsf{T}_v^{r,\ell}\right| \leqslant \left|\mathcal{T}_a^{r,\ell}\right|$. 因此, 存在一个从 $\mathsf{T}_v^{r,\ell}$ 到 $\mathcal{T}_a^{r,\ell}$ 的单射 ψ. 此外, 由定义 12 可知, 对于每个 $T \in \mathscr{T}_v^{r,\ell}$ 及其在 $G(T)$ 中的任意一条有向边 (u, u'), 都有 $u \in T \subseteq V \uplus \mathcal{E}$ 且 $u' \in V(u) \uplus \mathcal{E}(u)$. 结合 $\phi(T)$ 是 $G(T)$ 的有根有向生成树的事实, 可得对于 $\phi(T)$ 中的每个节点 u, 其子节点来自 $V(u) \uplus \mathcal{E}(u)$. 因此, $\phi(T)$ 是 T_v 的一棵子树. 此外, 由于 $\phi(T)$ 的根为 v, 且 $\big|\{\phi(T)$ 中标签属于 V 的节点$\}\big| = r$, $\big|\{\phi(T)$ 中标签属于 $\mathcal{E}$ 的节点$\}\big| = \ell$, 可得 $\phi(T) \in \mathsf{T}_v^{r,\ell}$. 形式化地,

$$\left\{\phi(T) \,\middle|\, T \in \mathscr{T}_v^{r,\ell}\right\} \subseteq \mathsf{T}_v^{r,\ell}.$$

回顾 ϕ 是从 $\mathscr{T}_v^{r,\ell}$ 到 $\left\{\phi(T) \,\middle|\, T \in \mathscr{T}_v^{r,\ell}\right\}$ 的单射, 因此它也是从 $\mathscr{T}_v^{r,\ell}$ 到 $\mathsf{T}_v^{r,\ell}$ 的单射. 进一步地, 由于 ψ 是从 $\mathsf{T}_v^{r,\ell}$ 到 $\mathcal{T}_a^{r,\ell}$ 的单射, 故复合映射 $\psi \uplus \phi$ 是从 $\mathscr{T}_v^{r,\ell}$ 到 $\mathcal{T}_a^{r,\ell}$ 的单射. 至此, 引理得证. 令 $\sigma \in \mathcal{Q}^*$ 为只有一个变量 $v \in V$ 满足 $\sigma(v) = \star$ 的部分赋值. 以下引理界定了路径 $\mathsf{Path}(\sigma) = (\sigma_0, \sigma_1, \ldots, \sigma_\ell)$ 的长度 ℓ, 并进一步证明了, 如果在给定 σ_ℓ 后, 坏变量/约束的集合变得过大, 则在 H_Φ 中存在一个大的广义 $\{2,3\}$- 树 T, 使得定义 11 定义的事件 $\mathcal{E}_T^\sigma$ 发生.

引理17 设 $\sigma \in \mathcal{Q}^*$ 是一个部分赋值, 其中恰有一个变量 $v \in V$ 满足 $\sigma(v) = \star$, 并且 $\mathsf{Path}(\sigma) = (\sigma_0, \sigma_1, \ldots, \sigma_\ell)$. 假设 $|V_\star^{\sigma_\ell}| + |\mathcal{C}_{\star\text{-frozen}}^{\sigma_\ell}| \geqslant L$, 其中 $L \geqslant 1$ 是某个整数, 则存在一个以 v 为根的广义 $\{2,3\}$- 树 $T = U \uplus E$, 满足 $\mathcal{E}_T^\sigma$ 发生, 并且 $|U| + \Delta \cdot |E| \geqslant L$.

引理 17 的证明在补充材料中给出. 接下来通过引理 17 来证明引理 11.

引理 11 的证明 当 $i = 0$ 时显然成立. 接下来假设 $i > 0$. 回顾 $V = \{v_1, v_2, \ldots, v_n\}$. 我们声称, 对于每个可能的 σ, 在给定 $X_0^t = \sigma$ 的情况下, 如果 $\left|V_\star^{X_\ell^t}\right| + \left|\mathcal{C}_{\star\text{-frozen}}^{X_\ell^t}\right| \geqslant i\Delta$, 则存在某个 $L \geqslant i\Delta$ 和 $T \in \mathscr{T}_{v_t}^L$, 使得 $\mathcal{E}_T^t$ 发

生. 形式化地,

$$\begin{aligned}&\mathbf{Pr}\left[\left(\left|V_\star^{X_\ell^t}\right|+\left|\mathcal{C}_{\star\text{-frozen}}^{X_\ell^t}\right|\geqslant i\Delta\right)\wedge\left(X_0^t=\sigma\right)\right]\\&\quad\leqslant\mathbf{Pr}\left[\left(\text{存在 } L\geqslant i\Delta, T\in\mathscr{T}_{v_t}^L \text{ 使得 } \mathcal{E}_T^t \text{ 发生}\right)\wedge\left(X_0^t=\sigma\right)\right].\end{aligned}\tag{53}$$

结合并集界, 得到

$$\mathbf{Pr}\left[\left(\left|V_\star^{X_\ell^t}\right|+\left|\mathcal{C}_{\star\text{-frozen}}^{X_\ell^t}\right|\geqslant i\Delta\right)\wedge\left(X_0^t=\sigma\right)\right]\leqslant\sum_{L\geqslant i\Delta}\sum_{T\in\mathscr{T}_{v_t}^L}\mathbf{Pr}\left[\mathcal{E}_T^t\wedge\left(X_0^t=\sigma\right)\right].\tag{54}$$

因此, 有

$$\begin{aligned}&\mathbf{Pr}\left[\left|V_\star^{X_\ell^t}\right|+\left|\mathcal{C}_{\star\text{-frozen}}^{X_\ell^t}\right|\geqslant i\Delta\right]\cdot\mathbf{E}\left[\chi(X_0^t)\;\middle|\;\left|V_\star^{X_\ell^t}\right|+\left|\mathcal{C}_{\star\text{-frozen}}^{X_\ell^t}\right|\geqslant i\Delta\right]\\&=\mathbf{Pr}\left[\left|V_\star^{X_\ell^t}\right|+\left|\mathcal{C}_{\star\text{-frozen}}^{X_\ell^t}\right|\geqslant i\Delta\right]\cdot\sum_\sigma\left(\chi(\sigma)\,\mathbf{Pr}\left[X_0^t=\sigma\;\middle|\;\left|V_\star^{X_\ell^t}\right|+\left|\mathcal{C}_{\star\text{-frozen}}^{X_\ell^t}\right|\geqslant i\Delta\right]\right)\\&=\sum_\sigma\left(\chi(\sigma)\,\mathbf{Pr}\left[\left(X_0^t=\sigma\right)\wedge\left(\left|V_\star^{X_\ell^t}\right|+\left|\mathcal{C}_{\star\text{-frozen}}^{X_\ell^t}\right|\geqslant i\Delta\right)\right]\right)\\&\leqslant\sum_\sigma\chi(\sigma)\left(\sum_{L\geqslant i\Delta}\sum_{T\in\mathscr{T}_{v_t}^L}\mathbf{Pr}\left[\mathcal{E}_T^t\wedge\left(X_0^t=\sigma\right)\right]\right)\\&=\sum_{L\geqslant i\Delta}\sum_{T\in\mathscr{T}_{v_t}^L}\sum_\sigma\left(\chi(\sigma)\,\mathbf{Pr}\left[\mathcal{E}_T^t\wedge\left(X_0^t=\sigma\right)\right]\right),\end{aligned}\tag{55}$$

其中第一个等式来自全期望法则, 第二个等式使用了链式法则, 而不等式则来自 $\chi(\cdot)$ 的非负性和式 (54). 此外, 使用全期望法则, 还有

$$\sum_\sigma(\chi(\sigma)\,\mathbf{Pr}\left[\mathcal{E}_T^t\wedge(X_0^t=\sigma)\right])=\sum_\sigma(\chi(\sigma)\,\mathbf{Pr}\left[X_0^t=\sigma\,|\,\mathcal{E}_T^t\right]\mathbf{Pr}\left[\mathcal{E}_T^t\right])\leqslant\mathbf{Pr}\left[\mathcal{E}_T^t\right]\mathbf{E}\left[\chi(X_0^t)\,|\,\mathcal{E}_T^t\right].\tag{56}$$

因此, 得到

$$\begin{aligned}&\mathbf{Pr}\left[\left|V_\star^{X_\ell^t}\right|+\left|\mathcal{C}_{\star\text{-frozen}}^{X_\ell^t}\right|\geqslant i\Delta\right]\cdot\mathbf{E}\left[\chi(X_0^t)\;\middle|\;\left|V_\star^{X_\ell^t}\right|+\left|\mathcal{C}_{\star\text{-frozen}}^{X_\ell^t}\right|\geqslant i\Delta\right]\\&=\sum_{L\geqslant i\Delta}\sum_{T\in\mathscr{T}_{v_t}^L}\left(\mathbf{Pr}\left[\mathcal{E}_T^t\right]\cdot\mathbf{E}\left[\chi(X_0^t)\;\middle|\;\mathcal{E}_T^t\right]\right)\quad(\text{式 (55) 和 (56)})\\&=\sum_{j\geqslant i}\sum_{r=0}^{\Delta-1}\sum_{T\in\mathscr{T}_{v_t}^{j\Delta+r}}\left(\mathbf{Pr}\left[\mathcal{E}_T^t\right]\cdot\mathbf{E}\left[\chi(X_0^t)\;\middle|\;\mathcal{E}_T^t\right]\right)\\&\leqslant\sum_{j\geqslant i}\sum_{r=0}^{\Delta-1}\sum_{T\in\mathscr{T}_{v_t}^{j\Delta+r}}\left((8\mathrm{e}k\Delta)^{-|T\cap V|}\cdot\left(4\mathrm{e}\Delta^3\right)^{-|T\cap\mathcal{C}|}\right)\quad(\text{引理 14})\\&\leqslant\sum_{j\geqslant i}\sum_{r=0}^{\Delta-1}(2^{-j-1}\cdot\Delta^{-1})\quad(\text{引理 16})\\&=2^{-i}.\end{aligned}$$

接下来, 证明式 (53) 成立. 从而证明定理成立. 给定 X_0^t 的一个可能赋值 σ , 假设 $X_0^t=\sigma$ 且 $\left|V_\star^{X_\ell^t}\right|+\left|\mathcal{C}_{\star\text{-frozen}}^{X_\ell^t}\right|\geqslant i\Delta$. 根据式 (24), 有 $|V_\star^{\sigma_\ell}|+|\mathcal{C}_{\star\text{-frozen}}^{\sigma_\ell}|\geqslant i\Delta$. 我们声称 $\sigma=X_{v_t\leftarrow\star}^{t-1}$. 此外, 对于任意 $v\in V$, 都有 $X^{t-1}(v)\neq\star$, 因为根据式 (24), X^{t-1} 是由 $X^0=\star^V$ 和算法 5 生成的, 且在算法 5 中没有任何顶点被设置为 $\star$. 结合 $\sigma=X_{v_t\leftarrow\star}^{t-1}$ 和式 (24), 得出只有一个变量 $v_t\in V$ 使得 $\sigma(v_t)=\star$. 再结合引理 17 和 $|V_\star^{\sigma_\ell}|+|\mathcal{C}_{\star\text{-frozen}}^{\sigma_\ell}|\geqslant i\Delta$, 得出事件 $\mathcal{E}_T^\sigma$ 对于某个 $L\geqslant i\Delta$ 和 $T\in\mathscr{T}_{v_t}^L$ 发生. 另外, 由 $X_0^t=\sigma$ 和定义 11 中 $\mathcal{E}_T^t$ 和 $\mathcal{E}_T^\sigma$ 的定义, 得出 $\mathcal{E}_T^\sigma$ 恰好是 $\mathcal{E}_T^t$. 因此, 存

在某些 $L \geqslant i\Delta$ 和 $T \in \mathscr{T}_{v_t}^{L}$, 使得事件 $\mathcal{E}_T^t$ 发生, 并且式 (53) 随之成立. 最后, 证明 $\sigma = X_{v_t \leftarrow \star}^{t-1}$, 以完成对式 (53) 和引理的证明.

根据式 (24), 可以得到 $\sigma = X_0^t = X^{t-1}$ 或者 $\sigma = X_0^t = X_{v_t \leftarrow \star}^{t-1}$. 因此, 证明 $\sigma \neq X^{t-1}$ 即可.

利用反证法, 假设 $\sigma = X^{t-1}$. 回顾我们已有的结果, $X^{t-1}(v) \neq \star$ 对于任意 $v \in V$ 都成立. 因此, $\sigma(v) = X^{t-1}(v) \neq \star$ 对于所有的 $v \in V$ 成立. 根据定义 10, 得到 $\mathsf{Path}(\sigma) = (\sigma)$ 且 $\sigma_\ell = \sigma$. 因此, $\sigma_\ell(v) \neq \star$ 对于所有 $v \in V$ 都成立. 结合定义 11, 可以得出结论: $V_\star^{\sigma_\ell} = \emptyset$ 且 $\mathcal{C}_{\star\text{-frozen}}^{\sigma_\ell} = \emptyset$, 这与 $|V_\star^{\sigma_\ell}| + |\mathcal{C}_{\star\text{-frozen}}^{\sigma_\ell}| \geqslant i\Delta$ 的条件矛盾. 因此, $\sigma \neq X^{t-1}$, 进而 $\sigma = X_{v_t \leftarrow \star}^{t-1}$. 至此, 完成了对式 (53) 和引理的证明.

回顾定义 12 中的广义 $\{2,3\}$- 树的定义和第 3.2 小节中 $H_\Phi = (V, \mathcal{C})$ 的定义. 我们有以下引理.

引理18　对于每个 $v \in V$, 存在一个广义的 $\{2,3\}$- 树 $T = \{v\} \uplus E$ 在 H_Φ 中, 其中 v 是它的一个根, 且 $E \subseteq \mathcal{C}_{\text{frozen}}^{X^n}$ 并且 $\Delta^2 |E| \geqslant \left|\mathcal{C}_v^{X^n}\right|$.

引理 18 的证明在补充材料中给出. 接下来展示如何利用引理 18 来证明引理 12.

引理 12 的证明　当 $i = 0$ 时, 结论是平凡的. 在接下来的证明中, 假设 $i > 0$. 回顾第 3.2 小节中 $H_\Phi = (V, \mathcal{C})$ 的定义. 给定整数 $t, r, \ell \geqslant 0$ 和 $v \in V$, 回顾定义 12 中的 $\mathscr{T}_v^t$ 和 $\mathscr{T}_v^{r,\ell}$. 对于任意 $i \geqslant 1$ 和 $v \in V$, 有

$$
\begin{aligned}
\mathbf{Pr}\left[\left|\mathcal{C}_v^X\right| \geqslant 2i\Delta^2\right] &\leqslant \sum_{j \geqslant 2i} \sum_{T \in \mathscr{T}_v^{1,j}} \mathbf{Pr}\left[T \cap \mathcal{C} \subseteq \mathcal{C}_{\text{frozen}}^X\right] \quad (\text{引理 } 18) \\
&\leqslant \sum_{j \geqslant 2i} \sum_{T \in \mathscr{T}_v^{1,j}} (4\mathrm{e}\Delta^3)^{-|T \cap \mathcal{C}|} \quad (\text{引理 } 15) \\
&= 8\mathrm{e}k\Delta \sum_{j \geqslant 2i} \sum_{T \in \mathscr{T}_v^{1,j}} (8\mathrm{e}k\Delta)^{-1} (4\mathrm{e}\Delta^3)^{-|T \cap \mathcal{C}|} \\
&\leqslant 8\mathrm{e}k\Delta \sum_{j \geqslant 2i} \sum_{T \in \mathscr{T}_v^{j\Delta+1}} (8\mathrm{e}k\Delta)^{-|T \cap V|} (4\mathrm{e}\Delta^3)^{-|T \cap \mathcal{C}|} \quad \left(\mathscr{T}_v^{1,j} \subseteq \mathscr{T}_v^{j\Delta+1}\right) \\
&\leqslant 8\mathrm{e}k\Delta \cdot \sum_{j \geqslant 2i} \left(2^{-j-1} \cdot \Delta^{-1}\right) \quad (\text{引理 } 16) \\
&\leqslant 8\mathrm{e}k \cdot 4^{-i}.
\end{aligned} \tag{57}
$$

8　结论与开放问题

本文提出了一种在 LLL 条件下对一般 CSP 解进行均匀采样的算法. 该算法在变量数量方面具有期望近线性的运行时间, 并且在其他局部参数 (包括值域大小 q、约束宽度 k 以及约束度 Δ) 上具有多项式运行时间. 这是首次在 LLL 条件下对常数 q, k 和 Δ 的一般 CSP 解的近线性时间采样算法; 同时也是首次在不对约束度或宽度作任何假设的情况下, 对一般 CSP 解的多项式时间采样算法.

我们采样算法的关键步骤在于构造一个边缘采样算法, 该采样算法能够按照正确的边缘分布抽取变量的取值. 在假设的 LLL 条件下, 这个边缘采样算法是一个局部算法, 其复杂度与 CSP 的整体规模无关, 仅依赖于局部参数 q, k 和 Δ. 这一边缘采样算法揭示了一个发人深省的观点: 在 LLL 条件下, 一个局部定义的采样或推理问题可以以局部成本得到解决.

本文还提出了如下若干开放问题.

• 一个开放问题是, 能否将目前用于采样一般 CSP 的局部引理条件 $p\Delta^5 \lesssim 1$ 改进得更接近下界 $p\Delta^2 \gtrsim 1$. 我们认为, 如果能够消除当前局部引理条件中额外出现的 q 和 k 因子, 将有助于更深入地理解采样局部引理的本质.

• 另一个根本性问题是, 如何将目前针对 CSP 的界限推广到更一般的采样洛瓦兹局部引理, 即适用于非均匀分布和/或不对称条件的情况.

• Anand 和 Jerrum [23] 提出的递归边缘采样算法是一种令人耳目一新的采样新思路. 正如本文所展示的, 这一方法能够解决一个原本难以解决的问题. 未来探索这一新思想在采样局部引理研究中的更多潜在应用将会非常令人兴奋.

• 尽管目前在技术上存在一定障碍, 基于马尔可夫链的算法依然具有若干优势, 如其高效的并行化能力 [34]. 因此, 开发基于马尔可夫链的针对一般 CSP 的采样算法仍具有很高的研究价值. 为此, 或许需要发展一种能够动态投影解空间的新方法, 而该方法本身也可能具有独立意义.

致谢　我们感谢凤维明和吴克文的有益讨论. 尹一通感谢 Vishesh Jain 指出了 robust CSP 的概念.

补充材料　本文的补充材料见网络版 infocn.scichina.com. 补充材料为作者提供的原始数据, 作者对其学术质量和内容负责.

参考文献

1 Erdős P, Lovász L. Problems and results on 3-chromatic hypergraphs and some related questions. Infinite Finite Sets, 1975, 10: 609–628

2 Shearer J B. On a problem of Spencer. Combinatorica, 1985, 5: 241–245

3 Moser R A, Tardos G. A constructive proof of the general Lovász local lemma. J ACM, 2010, 57: 11

4 Guo H, Jerrum M, Liu J. Uniform sampling through the Lovász local lemma. J ACM, 2019, 66: 18

5 Moitra A. Approximate counting, the Lovász local lemma, and inference in graphical models. J ACM, 2019, 66: 10: 25

6 Guo H, Liao C, Lu P, et al. Counting hypergraph colorings in the local lemma regime. SIAM J Comput, 2019, 48: 1397–1424

7 Harris D G. New bounds for the Moser-Tardos distribution. Random Struct Algorithms, 2020, 57: 97–131

8 Feng W, Guo H, Yin Y, et al. Fast sampling and counting k-SAT solutions in the local lemma regime. J ACM, 2021, 68: 40

9 Feng W, He K, Yin Y. Sampling constraint satisfaction solutions in the local lemma regime. In: Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing, 2021. 1565–1578

10 Jain V, Pham H T, Vuong T D. On the sampling Lovász local lemma for atomic constraint satisfaction problems. ArXiv:2102.08342

11 Jain V, Pham H T, Vuong T D. Towards the sampling Lovász local lemma. In: Proceedings of the 2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS), Denver, 2021. 173–183

12 He K, Sun X, Wu K. Perfect sampling for (atomic) Lovász local lemma. ArXiv:2107.03932

13 Feng W, Guo H, Wang J. Improved bounds for randomly coloring simple hypergraphs. In: Proceedings of the 24th International Conference on Randomization and Computation, 2022. 25:1–25:17

14 Galanis A, Goldberg L A, Guo H, et al. Counting solutions to random CNF formulas. SIAM J Comput, 2021, 50: 1701–1738

15 Galanis A, Guo H, Wang J. Inapproximability of counting hypergraph colorings. ACM Trans Comput Theory, 2023, 14: 10

16 Guo H, Jerrum M. A polynomial-time approximation algorithm for all-terminal network reliability. SIAM J Comput, 2019, 48: 964–978

17 Guo H, He K. Tight bounds for popping algorithms. Random Struct Algor, 2020, 57: 371–392

18 Bezáková I, Galanis A, Goldberg L A, et al. Approximation via correlation decay when strong spatial mixing fails. SIAM J Comput, 2019, 48: 279–349

19 Harvey N J A, Vondrák J. An algorithmic proof of the Lovász local lemma via resampling oracles. In: Proceedings of 2015 IEEE 56th Annual Symposium on Foundations of Computer Science, Berkeley, 2015. 1327–1345

20 He K, Wang C, Yin Y. Sampling Lovász local lemma for general constraint satisfaction solutions in near-linear time. In: Proceedings of 2022 IEEE 63rd Annual Symposium on Foundations of Computer Science, 2022. 147–158

21 He K, Wang C, Yin Y. Deterministic counting Lovász local lemma beyond linear programming. In: Proceedings of the 34th ACM-SIAM Symposium on Discrete Algorithms, 2023. 3388–3425

22 Wang C, Yin Y. A Sampling Lovász Local Lemma for Large Domain Sizes. In: Proceedings of the 65th IEEE Symposium on Foundations of Computer Science, 2024. 129–150

23 Anand K, Jerrum M. Perfect sampling in infinite spin systems via strong spatial mixing. SIAM J Comput, 2022, 51: 1280–1295

24 Fill J A, Huber M. The randomness recycler: a new technique for perfect sampling. In: Proceedings of the 41st Annual Symposium on Foundations of Computer Science, Redondo Beach, 2000. 503–511

25 Feng W, Vishnoi N K, Yin Y. Dynamic sampling from graphical models. In: Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, 2019. 1070–1081

26 Jerrum M. Fundamentals of partial rejection sampling. ArXiv:2106.07744
27 Feng W, Guo H, Yin Y. Perfect sampling from spatial mixing. Random Struct Algor, 2022, 61: 678–709
28 Alon N. A parallel algorithmic version of the local lemma. Random Struct Algorithms, 1991, 2: 367–378
29 Beck J. An algorithmic approach to the Lovász local lemma. I. Random Struct Algorithms, 1991, 2: 343–365
30 Nacu Ş, Peres Y. Fast simulation of new coins from old. Ann Appl Probab, 2005, 15: 93–115
31 Huber M. Nearly optimal Bernoulli factories for linear functions. Combin Probab Comput, 2016, 25: 577–591
32 Dughmi S, Hartline J D, Kleinberg R, et al. Bernoulli factories and black-box reductions in mechanism design. In: Proceedings of the 49th ACM Symposium on Theory of Computing, 2017. 158–169
33 Haeupler B, Saha B, Srinivasan A. New constructive aspects of the Lovász local lemma. J ACM, 2011, 58: 28
34 Liu H, Yin Y. Parallelize single-site dynamics up to DoBruschin criterion. J ACM, 2025, 72: 7

Sampling Lovász local lemma for solutions to general constraint satisfaction problems

Kun HE[1], Chunyang WANG[2*] & Yitong YIN[2]

1. *Key Laboratory of Data Engineering and Knowledge Engineering, Renmin University of China, Beijing 100872, China*
2. *State Key Laboratory for Novel Software Technology, New Cornerstone Science Laboratory, Nanjing University, Nanjing 210023, China*

* Corresponding author. E-mail: wcysai@smail.nju.edu.cn

Abstract We give a fast algorithm for sampling uniform solutions of general constraint satisfaction problems (CSPs) in a local lemma regime. Under a given local lemma condition, the algorithm returns an almost uniform satisfying assignment in expected almost linear time. Previously, under similar local lemma conditions, efficient sampling algorithms were only known for the bounded degree case or the almost atomic case, where each constraint is violated by a small number of forbidden local configurations. The key term in our local lemma condition also improves the previously best-known bound for general CSPs and for atomic CSPs, including the special case of k-SAT. Our sampling approach departs from previous fast algorithms for sampling LLL, which were based on Markov chains. A crucial step of our algorithm is a recursive marginal sampler that is of independent interests. Within a local lemma regime, this marginal sampler can draw a random value for a variable according to its marginal distribution, at a cost independent of the size of the CSP.

Keywords sampling, constraint satisfaction problem, Lovász local lemma