arXiv:2302.10110v1 [cs.CC] 20 Feb 2023

时序探索问题的核化

Emmanuel Arrighi ！
卑尔根大学，挪威i
FedorV. Fomin ！卑尔
根大学，挪威
Petra Golovach 卑尔
根大学，挪威
Petra Wolf ！
卑尔根大学，挪威

摘要

我们研究时序图上探索问题的核化。时序图由一组有限的快照图序列 $\mathcal{G} = (G_1, G_2, \ldots\ldots, G_L)$ 构成，这些快照图共享同一顶点集，但边集可能不同。Erlebach 和 Spooner 提出的非严格时序探索问题（简称 NS-TEXP）询问：对于给定的时序图，单个智能体能否访问所有顶点，且智能体经过的边均存在于非严格单调时间步中；也就是说，智能体可以以无限速度沿着某个快照图中的边移动。探索最迟必须在最后一个快照图中完成。该问题的优化变体是 k-arb NS-TEXP 问题，其中智能体的任务是访问时序图中至少 k 个顶点。我们证明，在标准的计算复杂性假设下，无论是 NS-TEXP 还是 k-arb NS-TEXP，都不能在以下标准参数下具有多项式核：顶点数 n、生命周期 L、待访问顶点数 k，以及每个时间步的最大连通分量数 γ；也不能在组合参数 $L + k$、$L + \gamma$ 和 $k + \gamma$ 下具有多项式核。在建立这些下界的过程中，我们还回答了 Erlebach 和 Spooner 留下的几个未解问题。

我们还通过为时序图 $p(\mathcal{G}) = \sum_{i=1}^{L}(|E(G_i)|) - |V(G)| + 1$ 确定一个新参数，开启了对结构核化的研究。通俗地说，该参数衡量时序图的动态程度。我们的主要算法结果是，为更一般的加权 k-arb NS-TEXP 问题构造一个多项式规模（关于 $p(\mathcal{G})$）的核；该问题为顶点赋予权重，任务是寻找一条权重至少为 k 的时序游走。

2012 年 ACM 学科分类 计算理论 → 参数化复杂性与精确算法；计算理论 → 问题、归约与完全性

关键词与短语 时序图、时序探索、计算复杂性、核

1 引言

我们研究时序网络中探索任务和连通性任务的核化。核化，尤其是结构核化，似乎是解决静态图上许多优化问题的有效方法，但至少可以说，它在时序图中的应用效果并不那么理想。对此一个合理的解释是（稍后我们会提供一些证据）：静态图的大多数结构参数，例如树宽、反馈顶点集的大小或顶点覆盖数，在动态环境中似乎并无用处。这引出了以下问题。

> 动态图的何种结构可能对核化算法“有帮助”？

我们提出了一个新的时序图结构参数，用于衡量时序图的“动态性”。我们的主要结果是针对时序图探索问题的一种算法，该算法可生成一个大小关于这一新参数为多项式的核。在

2 时序探索问题的核化

正式陈述问题及我们的结果之前，我们先简要概述时序图和核化。

时序探索。 如今考虑的许多网络都表现出固有的动态行为，例如移动自组织网络中的连通性、社交网络中的关系，或互联网中服务的可访问性。经典图不足以对这些动态行为进行建模，因此所谓的时序图得到了深入研究。一般而言，时序图是随时间变化的图。文献中存在多种模型，用于描述不同类型的动态行为以及时序图的编码方式。在这里，我们将时序图$\mathcal{G}$ 定义为一系列快照 图G_1，G_2.....，G_L 这些图共享同一个顶点集合 V，但可能具有不同的边集E_1，E_2.....，E_L。我们将图 $G = (V, \bigcup_{i=1}^{L} E_i)$ 称为G的底层图。我们可以将时序图$\mathcal{G}$ 看作图 G，其中只有一部分边在某个时间步存在。快照的数量 L 通常称为G的生命周期。由于其在动态系统建模中的重要性，大量经典图问题已被推广到时序图上。关于时序图及其相关组合问题的介绍，请参见 [7, 27, 31, 33]。

图探索是图论领域中的一个基本问题，自 Shannon 在 1951 [34] 中提出以来便受到了广泛研究。Michail 和 Spirakis [32] 将“单个智能体能否探索给定图”这一问题推广到了时序图。他们证明，判断单个智能体能否在给定数量的时间步内访问时序图的所有顶点，同时每个时间步仅经过一条边，是一个 NP困难问题。这一难度促使 Erlebach 和 Spooner 在 [19] 中研究时序探索问题的参数化复杂性。

时序探索问题（简称 TEXP）规定智能体每个时间步只能经过一条边；但在某些情境下，网络变化缓慢得多，远远慢于智能体的移动速度。因此，自然可以假设智能体能够以无限速度移动，并在一个时间步内探索其当前所在的整个连通分量。例如，容延网络 [6]中就会出现这类情境。Erlebach 和 Spooner 在 [17] 将 TEXP 推广为允许智能体以无限速度移动，并将这个问题称为非严格时序探索，简称 NS-TEXP。NS-TEXP 要求判断智能体能否从初始顶点 v 出发，按照以下步骤访问图中的所有顶点。第 1 步，智能体访问包含 v的快照图 G_1 的连通分量 C_1 中的所有顶点。在第$i \in \{2, \ldots, L\}$步，智能体选择探索 G_i的某个连通分量 C_i 中的顶点，前提是该连通分量与智能体在第 $(i-1)$ 步探索的连通分量相交。

转而采用无限探索速度，会彻底改变时序探索问题的性质，因为此时探索所需时间可能远少于顶点数。对于经典图，判断一个图能否以无限速度完成探索，归结为判断其连通性；而对于时序图，NS-TEXP 是 NP 完全的 [17]。Erlebach 和 Spooner 在 [19] 研究了 NS-TEXP 的参数化复杂性，并证明存在以时序图的生命周期 L （等价于必须进行时序探索的时间步数）为参数的固定参数可解算法。他们还研究了 NS-TEXP 的一种优化变体：在图的生命周期内，不必访问所有顶点，但至少要访问 k 个顶点。该变体称为 k-arb NS-TEXP。在 [19] 中，提出了一种以参数 k 求解 k-arb NS-TEXP 的 FPT 算法。在这项研究的长篇版本 [20]，中，作者将以下问题列为开放问题：以每个时间步中连通分量的最大数量 γ 为参数时，NS-TEXP 是否属于 FPT，或至少属于 XP；以及以参数 L 为参数时，k-arb NS-TEXP 是否属于 FPT。

这些开放问题也在 2022年 ICALP 卫星研讨会时序图的算法方面 *V*上提出。在这项工作中，我们将对这两个问题都给出否定答案。

核化。简单来说，核化是一种预处理算法，它通过连续应用各种数据归约规则，尽可能缩小参数化问题的实例规模。核化是参数化复杂性研究的主要领域之一，该领域的许多重要进展都与核化有关。这些进展包括关于哪些问题存在多项式规模核的一般性算法发现，以及在某些复杂性理论假设下排除多项式核的框架。关于该领域的概述，请参阅著作 [22] 。核化（针对静态图）的一种有效方法，是研究各种结构度量（即不同于仅考虑输入总规模或预期解规模的度量）对问题复杂性的影响。这类结构参数化，例如反馈顶点集、顶点覆盖或树深度的最小规模，衡量了实例 [4, 25, 26, 36] 的非平凡程度。

我们的贡献。 我们关注 NS-TEXP 和 k-arb NS-TEXP 问题能否进行核化。在研究这些问题的参数化复杂性时，Erlebach 和 Spooner 在 [19, 20]证明，对于这些问题的某些组合以及诸如顶点数量 n、生命周期 L、需要访问的顶点数量 k和 $L+\gamma$等“标准”参数化，这些问题具有固定参数可解性；其中， γ 表示每个时间步的最大连通分量数，见表 1。首先，我们排除这两个问题针对上述每一种标准参数化都存在多项式核的可能性。此外，即使针对“组合”参数 $L+k$、 $L+\gamma$和 $k+\gamma$，我们的下界仍然成立。在建立核化下界的过程中，我们解决了 Erlebach 和 Spooner 在 [19, 20]提出的、关于 NS-TEMP 和 k-arb NS-TEXP 参数化研究中的两个开放问题。具体而言，我们证明，对于常数值的 $\gamma \geq 5$ ，NS-TEXP 是 NP 完全的；而 k-arb NS-TEXP 针对 L参数化时是 W[1]-困难的。

这促使我们研究探索问题的结构核化。乍看之下，最自然的方向似乎是探索时序图$\mathcal{G}$的底层图结构，但这一方向似乎无法带来新的算法结果。原因在于，我们关于 NS-TEXP 和 k-arb NS-TEXP 的下界对于底层图极为受限的时序图同样成立。具体而言，我们证明，即使底层图是一棵顶点覆盖数至多为 2 的树，这些问题仍然是 NP困难。因此，在这种情况下，合理的结构参数化应当不仅刻画底层图的“静态”结构，还应刻画$\mathcal{G}$中边的“动态”特性。基于这一考虑，我们引入时序图$\mathcal{G}$ =(G_1， G_2，……， G_L)的新参数，

$$p(\mathcal{G}) = \sum_{i=1}^{L}(|E(G_i)|) - |V(G)| + 1.$$

注意，如果 $\sum_{i=1}^{L}(|E(G_i)|) < |V(G)| - 1$，那么底层图不连通，因此无法探索时序图 $\mathcal{G}$ 。参数 $p(\mathcal{G})$ 进一步限制了多次出现的边的边出现次数，因此可以将其视为时序图动态程度的一种度量。我们的主要结果是针对更一般问题 Weighted k-arb NS-TEXP，在参数 $p = p(\mathcal{G})$ 下给出的一个多项式核。在 Weighted k-arb NS-TEXP 中，顶点带有权重，任务是寻找一个时序图，使其访问顶点的权重总和至少达到给定整数 k所指定的 k 。所得核的规模为 $\mathcal{O}(p^4)$，且其顶点数与参数 p 呈线性关系。我们的结果总结于表 1。

参数。	NS-TEXP		k-arb NS-TEXP	
	FPT	核	FPT	核
p	$2^{\mathcal{O}(p)}(nL)^{\mathcal{O}(1)}$	$\mathcal{O}(p^4)^\star$	$2^{\mathcal{O}(p)}(nL)^{\mathcal{O}(1)}$ k [20]中的固定参数可解	$\mathcal{O}(p^4)^\star$
n	$\mathcal{O}^*((2e)^n n^{\log n})$ [20]	不存在多项式核		不存在多项式核
L	$\mathcal{O}^*(L(L!)^2)$ [19]	不存在多项式核	W[1]-困难	不存在多项式核
k	-	-	$\mathcal{O}^*((2e)^k k^{\log k})$ [20]	不存在多项式核
$L+k$	-	-	在 k [20]中为固定参数可解	不存在多项式核
γ	对于 ≤ 2 [20]，属于 P 类，对 ≥ 5 为 NP困难	-	对于 ≤ 2 [20]，属于 P 类，对 ≥ 5 为 NP困难	-
$L+\gamma$	对 L [19]为固定参数可解	无多项式核 对于 $\gamma \geq 6$	$\mathcal{O}(\gamma^L n^{\mathcal{O}(1)})$	无多项式核 对于 $\gamma \geq 6$
$k+\gamma$	-	-	在 k [20]下为固定参数可解	不存在多项式核

表 1 NS-TEXP 和 k-arb NS-TEXP 参数化复杂性的概述。我们的贡献以红色突出显示。声称不存在多项式核的结果依赖于 NP $\not\subseteq$ coNP/poly 假设。这里，对于时序图 $\mathcal{G}$，n 表示顶点数，L 表示生命周期，γ 表示每个时间步的最大连通分量数，而 $p = p$表示$(\mathcal{G})$。对于标有 * 的条目，我们实际上针对更一般的变体 Weighted k-arb NS-TEXP 得到了广义核。

其他相关工作。Michail 和 Spirakis [32] 提出了 TEXP 问题，并证明在不对输入施加任何限制时，该问题是 NP 完全的。他们提出在始终连通假设下研究该问题，该假设要求时序图在每个时间步都连通。Erlebach 等人 [14] 沿用了这一思路，并证明对于始终连通的时序图，计算最早探索调度是 NP困难的，且对于任意 $\epsilon > 0$，无法在近似比为 O($n^{1-\epsilon}$) 的范围内近似求解。他们还证明，在始终连通假设下，对于底层图为平面图、树宽有界或为 $2 \times n$ 网格的时序图，存在次二次时间步的探索调度；对于环及带一条弦的环，则存在线性时间步的探索调度。Alamouti [35] 证明，带有 k 条弦的环可在 O($k^2 \cdot k! \cdot (2e)^k \cdot n$) 个时间步内完成探索。Adamson 等人 [1] 将这一界改进为 O(kn) 个时间步。他们还改进了底层图为平面图或树宽有界的时序图的最坏情况探索时间界。Erlebach 等人 [15] 证明，如果每个快照图都存在一个度有界的生成树，或者允许每步遍历两条边，那么时序图可在 O($n^{1.75}$) 个时间步内完成探索。在 [18], 中，Erlebach 和 Spooner 证明，对于始终连通的时序图，如果每个快照图中底层图至多有 k条边不存在，则总能在 O($kn \log n$) 个时间步内完成探索。同一作者还在 [16] 中证明，如果每个快照图的度有界，则时序图可在 O($n^2 \log n$) 个时间步内完成探索。Bodlaender 和 van der Zanden [3]证明，限制为底层图路径宽度至多为 2 的始终连通时序图时，TEXP 问题仍然是 NP 完全的。Erlebach 和 Spooner [19]从参数化视角研究了 TEXP 问题。基于颜色编码技术，他们给出了一个以 k 为参数的固定参数可解算法，适用于问题 k-arb TEXP 和非严格变体 k-arb NS-TEXP。他们还给出了以时序图的生命周期 L 为参数的固定参数可解算法，适用于 TEXP 和 NS-TEXP 问题。在相应的长篇版本 [20], 中，Erlebach 和 Spooner 进一步研究了每个时间步的最大连通分量数这一参数 γ，并证明当 $\gamma = 1$时，TEXP 是 NP困难的；而当 $\gamma \leq 2$时，NS-TEXP 可在多项式时间内求解。
NS-TEXP 问题

由 Erlebach 和 Spooner [17]提出并加以研究。除此之外，他们还证明了一般问题是 NP 完全的；并且在任意一对顶点之间移动所需的时间步数最多为 2 和 3 的假设下，计算最早探索调度分别无法在$\mathcal{O}(n^{1/2-\epsilon})$ 和$\mathcal{O}(n^{1-\epsilon})$范围内近似。

Casteigts 等人 [8] 对一个相关问题进行了参数化研究：他们研究的是寻找源点与目标点之间的时序路径的问题，该路径在任意中间顶点处等待的连续时间步数不超过 Δ 。Bumpus 和 Meeks [5]研究了一个图探索问题的参数化复杂性；该问题不再要求访问所有顶点，而是要求恰好遍历底层图的所有边一次。他们发现，对于底层图的自然结构参数而言，他们研究的探索问题不允许存在固定参数可解算法。类似地，Kunz、Molter 和 Zehavi [29]在研究寻找时序不相交路径与游走这一问题的参数化复杂性时，针对底层图的结构参数得出了若干困难性结果。该问题由 Klobas 等人 [28]提出。他们证明，以顶点数为参数时，该问题具有 W[1]-困难性；即使在底层图为星图的实例中，这一结论仍然成立。这表明，仅考虑底层图的结构参数不足以为时序图问题设计固定参数可解算法。

2 预备知识

记号。 我们用 $\mathbb{Z}$ 表示数集，用 $\mathbb{N}$ 表示自然数集，用 $\mathbb{N}_{>0}$ 表示正数集，用 $\mathbb{Q}$ 表示有理数集。设 n 为正整数，我们用 $[n]$ 表示集合{1、2、.....、n}。给定向量 $w = (w_1, w_2, ..., w_r) \in \mathbb{Q}^r$，令 $\|w\|_\infty =$ 为最大值 $_{i\in[r]} |w_i|$，且 $\|w\|_1 = \sum_{i\in[r]} |w_i|$。

图。我们将图 $G = (V, E)$ 视为静态无向图。给定图 G，我们用 $V(G)$ 表示 G 的顶点集，用 $E(G)$ 表示 G 的边集，用 $\gamma(G)$ 表示 G 的连通分量数。设 $G = (V, E)$ 为一个图，给定顶点子集 $X \subseteq V(G)$ 和边子集 $E' \subseteq E$，我们定义对 G 的以下操作：$G[X] = (X, \{uv \in E \mid u, v \in X\})$、$G - X = G[V \setminus X]$ 和 $G - E' = (V, E \setminus E')$。我们称图 G 中的游走为顶点与边的交替序列 $W = v_0, e_1, v_1, \ldots, e_r, v_r$，其中 $v_0, \ldots, v_r \in V(G)$、$e_1, \ldots, e_r \in E(G)$，并且对 $i \in [r]$ 有 $e_i = v_{i-1}v_i$；注意，W 可以多次经过相同的顶点和边。不重复顶点和边的游走称为路径。我们称 v_0 和 v_r 是 W 的端点，且 W 是一个 (v_0, v_r) - 游走。我们用 $V(W) \subseteq V(G)$ 表示 G 中被 W 经过的顶点集合，并用 $E(W)$ 表示 G 中属于 W 的边集合。给定游走 $W = v_0, e_1, v_1, \ldots, e_r, v_r$，对于 $0 \le i \le j \le r$，我们称序列 $W' = v_i, e_{i+1}, \ldots, e_j, v_j$ 为 W 的子游走。

时序图。设顶点集合为 V，其上的时序图 G $=(G_1, G_2, \ldots, G_L)$，满足对所有 $t \in [L], V(G_t) = V$。我们称 L 为 G 的生命周期；对于 $t \in [L]$，称 $G_t = (V, E_t)$ 为 G 在时间步 t的快照图。我们也可能将 G_t 称为 G(t)。我们称 $G = (V, E)$（其中 $E = \cup_{t\in[L]}$ ）为 G 的底层图。我们用 V(G) 表示 G 的底层图 G 的顶点集，用 γ(G) $=$ max$_{t\in L} \gamma(G_t)$ 表示 G 的所有快照图中的连通分量数的最大值。若上下文已明确图或时序图，我们分别记作V 和 γ。对于时序图 G $=(G_1, \ldots, G_L)$，我们将边出现总次数定义为 m(G) $= \sum_{i=1}^{L} |E(G_i)|$；若上下文已明确 G，则用 m 表示该值。

接下来，我们将关注智能体在快照图中具有无限速度的时序游走。这类时序游走在 [19] 中称为 非严格时序游走。在 [19]，中，非严格时序游走定义为连通分量的序列。

不过，对我们而言，将非严格时序游走 视为底层图中的单调的游走更为方便。对于底层图 G（G_1，$\ldots, G_L$）中的游走 $W = v_0, e_1, v_1, \ldots, e_r, v_r$，我们称 W是单调的，如果存在 $t_1, \ldots, t_r \in [L]$，使得$1 \leq t_1 \leq \cdots \leq t_r \leq L$，且对每个 $i \in [r]$，都有 $e_{t_i} \in E$（G_{t_i}）。非严格时序游走的定义立即推出以下观察结果。

▶**观察 1.** 给定一个时序图 G 和一个顶点 x，G 存在一条从 x 出发、访问集合 X 中各顶点的非严格时序游走，当且仅当其底层图 G 存在一条单调(x, y)-游走 W，其中某个 $y \in V(G)$ 满足 $X = V(W)$。

本文余下部分主要关注计算问题：寻找能够访问时序图中所有顶点的单调游走。我们也可以将这种游走称为探索调度，或简称探索。

▶**定义 2**（非严格时序探索（NS-TEXP））。
输入：时序图 $G = (G_1, G_2, \ldots, G_L)$，顶点 $v \in V(G)$。
问题：G 中是否存在一条从 v 出发并访问$V(G)$中所有顶点的单调游走?

我们进一步考虑 NS-TEXP 问题的一个更一般的变体：要求存在一条单调游走，至少访问 k 个顶点（而非 $|V|$）。我们将此问题称为 k-arb NS-TEXP。

▶**定义 3** (k-任意非严格时序探索（k-arb NS-TEXP）)。输入：时序图 $G = (G_1, G_2, \ldots, G_L)$，顶点 $v \in V(G)$，整数 k。问题：是否存在一条从 v 出发并至少访问 k 个顶点的单调游走?

我们将通过考虑 k-arb NS-TEXP 的加权版本，进一步推广该问题。我们将此问题称为加权 k-arb NS-TEXP。

▶定义 *4*（加权 k-任意非严格时序探索（加权 k-*arb NS-TEXP*））。输入：时序图 $G = (G_1, G_2, \ldots, G_L)$，正值权重函数 w: $V(G) \to \mathbb{N}_{>0}$，顶点 $v \in V(G)$，整数 k。问题：是否存在一条在 G 中的单调游走 W，使得 W从 v 出发，且 $w(V(W)) = \sum_{u \in V(W)} w(u) \geq k$?
注意，k-*arb NS-TEXP* 是加权 k-*arb NS-TEXP* 的一个特例，其中对每个.....都有 $w(x) = 1$

y $x \in V$。

参数化复杂性与核化。关于该领域的详细介绍，请参阅著作 [10]，另可参阅近期出版的核化专著 [22]。这里我们仅简要介绍一些基本概念。

一个参数化问题是一种语言 $Q \subseteq \Sigma^* \times \mathbb{N}$，其中 Σ^* 是有限字母表 Σ 上所有字符串构成的集合。相应地，Q 的一个输入是一个二元组（I，k），其中 $I \subseteq \Sigma^*$ 且 $k \in \mathbb{N}$；k 是该问题的参数。

若参数化问题 Q 可在某个函数 f 所决定的 $f(k) \cdot |I|^{\mathcal{O}(1)}$ 时间内判定（I，k）$\in Q$，且该函数仅依赖于参数 k，则称其固定参数可解（FPT）。相应地，固定参数可解问题构成参数化复杂性类 FPT。

参数化复杂性理论还提供了在合理的复杂性理论假设下排除固定参数可解算法存在性的工具。为此，引入了参数化复杂性类层次结构

固定参数可解 $\subseteq$ W[1] $\subseteq$ W[2] $\subseteq \cdots \subseteq$ XP

在 [12], 中，并猜想这些包含关系是严格的。证明一个参数化问题不太可能存在 FPT 算法的基本方法，是证明它是 W[1] 或 W[2]-困难的。

对于参数化问题 Q ，数据规约规则（简称归约规则）是一个函数 ϕ: $\Sigma^* \times \mathbb{N} \to \Sigma^* \times \mathbb{N}$，它将 Q 的一个实例 (I, k) 映射为一个等价实例 (I', k')，且该实例属于 Q ，并且 ϕ 可在关于 $|I|$ 和 k 的多项式时间内计算。若以下条件成立，我们称 Q 的两个实例是等价的：$(I, k) \in Q$ 当且仅当 $(I', k') \in Q$。我们将归约规则 ϕ 的这一性质——即将一个实例转换为等价实例——称为该归约规则的安全性。

非正式地说，核化是一种预处理算法，它连续应用各种数据规约规则，以尽可能缩小实例规模。预处理算法以 Q 的一个实例 $(I, k) \in \Sigma^* \times \mathbb{N}$ 作为输入，在关于 $|I|$ 和 k 的多项式时间内运行，并返回 Q 的一个等价实例 (I', k')。预处理算法 $\mathcal{A}$ 的质量由输出的大小衡量。更准确地说，预处理算法 $\mathcal{A}$ 的输出大小是一个函数 $\text{size}_{\mathcal{A}}$: $\mathbb{N} \to \mathbb{N} \cup \{\infty\}$ ，定义如下：

$$\text{size}_{\mathcal{A}}(k) = \sup\{|I'| + k' : (I', k') = \mathcal{A}(I, k), I \in \Sigma^*\}.$$

参数化问题 Q 的核化算法，或简称核，是一种预处理算法$\mathcal{A}$：给定 Q的一个实例（I， k），它能在关于 $|I|$ 和 k 的多项式时间内运行，并返回 Q 的一个等价实例（I'， k'），使得 $\text{size}_{\mathcal{A}}(k) \leq g(k)$，其中 g: $\mathbb{N} \to \mathbb{N}$是某个可计算函数。我们称 $g(\cdot)$ 为核的大小。如果 $g(\cdot)$ 是多项式函数，那么我们就说 Q 存在一个多项式核。

交叉组合。众所周知，每个固定参数可解问题都存在一个核 [13]。然而，在一些合理的复杂性假设下，存在没有多项式核的固定参数可解问题。特别地，我们使用 [2] 中引入的交叉组合技术来证明，除非 NP $\subseteq$ coNP /poly，否则某个参数化问题不存在多项式核。

为了定义交叉组合，我们先给出多项式等价关系的定义。

▶ **定义5**（多项式等价关系）。 集合 Σ^* 上的等价关系R称为多项式等价关系，若满足以下条件：
（**a**）存在一种算法，给定字符串 x、 $y \in \Sigma^*$，可以判定 x 是否与 y 等价，且运行时间关于 $|x| + |y|$为多项式。（**b**）对于任意有限集合 $S \subseteq \Sigma^*$ ，等价关系$\mathcal{R}$ 将 S 中的元素划分为至多（$\max_{x \in S} |x|$）$^{\mathcal{O}(1)}$ 个等价类。

语言到参数化语言的交叉分解定义如下。

▶ **定义6**（交叉组合）。 设 $L \subseteq \Sigma^*$ 是一个语言， $Q \subseteq \Sigma^* \times \mathbb{N}$ 是一个参数化语言。若存在一个多项式等价关系R和一个称为交叉组合的算法A，满足以下条件，则称 L 交叉组合到 Q 。算法A以关于R等价的字符串序列 x_1、 x_2、.....、 $x_t \in \Sigma^*$ 作为输入，运行时间关于 $\sum_{i=1}^{t} |x_i|$为多项式，并输出一个实例（y， k） $\in \Sigma^* \times \mathbb{N}$，使得：

（**a**） $k \leq p$（最大$_{i=1}^{t} |x_i|$ + 对数 t）对于某个多项式 $p(\cdot)$，并且（**b**）
（y， k） $\in Q$ 当且仅当存在至少一个索引 i 使得 $x_i \in L$。

排除多项式核存在性的理论工具建立在以下 Bodlaender 等人的定理之上 [2]。该定理指出，如果某个参数化问题允许将某种 NP 难语言

交叉组合到其中，那么此类问题存在多项式核的可能性极低。

▶ **命题 7** ([2])。 设 $L \subseteq \Sigma^*$ 为一种 *NP* 困难语言。若 L 可交叉组合到一个参数化问题 Q，且 Q 具有多项式核，则 NP $\subseteq$ coNP /poly。

3 下界

NS-TEXP 的下界——受限情形的 NP 难性

[19], 中曾将 NS-TEXP 是否属于固定参数可解，且参数为任意快照图中连通分量的最大数量（即参数为 γ）列为一个开放问题。我们通过证明当 $\gamma \geq 5$ 时 NS-TEXP 为 NP 完全，否定地回答了这个问题。

▶ **定理 8**。 当 $\gamma \geq 5$ 时，*NS-TEXP* 为 *NP* 完全。

证明。 我们将可满足性问题 SAT归约到该问题；该问题要求判断给定的布尔公式是否具有满足赋值。令 $\varphi = \{c_1, c_2, \ldots, c_m\}$ 为定义在变量集合 $X = \{x_1, x_2, \ldots, x_n\}$上的合取范式布尔公式。我们从 φ构造一个时序图$\mathcal{G}$，其中每个快照图包含 4 个或 5 个连通分量，使得$\mathcal{G}$ 存在一条访问 $V(\mathcal{G})$ 中所有顶点的单调游走，当且仅当 φ 是可满足的。

构造的主要思想如下。在 $V(\mathcal{G})$ 中，每个子句对应一个顶点，每个变量 x对应两个顶点，分别对应 x 和 $\neg x$；每个变量 x对应一个控制顶点 $\widehat{x}$，此外，子句集合还对应一个单独的控制顶点 $\widehat{c}$。快照图的序列在包含 4 个和 5 个连通分量之间交替变化。在包含 4 个连通分量的情况下，一个分量收集所有子句顶点，一个分量收集所有变量顶点 x 和 $\neg x$，一个分量收集所有尚未处理的控制顶点 $\widehat{x}$ 和 $\widehat{c}$，另一个分量收集所有已处理控制顶点 $\widehat{x}$。在包含 5 个连通分量的情况下，额外的分量收集某个变量 x 的一个否定文字 $\neg x$ 以及包含 $\neg x$ 的所有子句。在此时间步中，包含 x 的子句被并入变量分量，而该分量仍包含 x。这使我们能够通过探索调度选择一个变量赋值。在下一个时间步，且仅在这个时间步中，控制顶点 $\widehat{x}$ 包含在变量分量中。在之后的所有时间步中，$\widehat{x}$ 包含在收集已处理控制顶点的分量中。由此，控制顶点确保我们在每个由 4 个连通分量组成的快照中都返回包含变量的分量。

现在我们给出一个更正式的构造。对于此构造，我们关注每个快照图的连通分量，而不关注连通分量的实际结构。为简单起见，我们将连通分量定义为顶点集合，并假设每个连通分量都构成一个团。对于子集 $S \subseteq V$，我们用 $K(S)$ 表示 S 中顶点之间所有可能边的集合。因此，$(S, K(S))$ 是一个团。令 $\mathsf{G} = (G_1, G_2, \ldots, G_L)$，其中 $L = 2n + 1$ 是由 $\varphi = \{c_1, c_2, \ldots, c_m\}$ 构造的时序图。我们将 $V(\mathsf{G}) = \{c_1, c_2, \ldots, c_m\} \cup \{x_i, \neg x_i, \widehat{x_i} \mid x_i \in X\} \cup \{\widehat{c}\}$ 定义为。对于 $E(G_1) = K(C_{1,1}) \cup K(C_{1,2}) \cup K(C_{1,3})$，我们设 $C_{1,1} = \{\widehat{c}\} \cup \{\widehat{x_i} \mid x_i \in X\}$，$C_{1,2} = \{c_1, c_2, \ldots, c_m\}$，以及 $C_{1,3} = \{x_i, \neg x_i \mid x_i \in X\}$。起始顶点设为 x_{1}。

对于下一个时间步，我们将快照图的边定义为 $E(G_2) = K(C_{2,1}) \cup K(C_{2,2}) \cup K(C_{2,3}) \cup K(C_{2,4})$，其中 $C_{2,1} = C_{1,1} = \{\widehat{c}\} \cup \{\widehat{x_i} \mid x_i \in X\}$, $C_{2,2} = \{c_1, c_2, \ldots, c_m\} \setminus \{c \in \varphi \mid x_1 \in c \vee \neg x_1 \in c\}$, $C_{2,3} = (\{x_i, \neg x_i \mid x_i \in X\} \setminus \{\neg x_1\}) \cup \{c \in \varphi \mid x_1 \in c\}$，且$C_{2,4} = \{\neg x_1\} \cup \{c \in \varphi \mid \neg x_1 \in c\}$。其直观含义是：如果变量 x_1 被赋值为，探索调度会在访问 $C_{1,3}$ 中的顶点后，访问 $C_{2,3}$ 中的顶点

`true`，否则会在$C_{1,3}$之后访问$C_{2,4}$，前提是 x_1 被赋值为 `false`。注意，从$C_{1,3}$无法到达其他连通分量，因为它与$C_{1,3}$的交集为空。

在下一个时间步中，我们通过让控制顶点 $\widehat{x_1}$ 经过第三个连通分量，强制探索调度返回第三个连通分量。因此，我们定义 $E(G_3) = K(C_{3,1}) \cup K(C_{3,2}) \cup K(C_{3,3})$，其中 $C_{3,1} = (\{\widehat{c}\} \cup \{\widehat{x_i} \mid x_i \in X\}) \setminus \{\widehat{x_1}\}$, $C_{3,2} = \{c_1, c_2, \ldots, c_m\}$，且 $C_{3,3} = \{x_i, \neg x_i \mid x_i \in X\} \cup \{\widehat{x_1}\}$。

在剩余的时间步中，我们交替采用第二和第三个时间步的结构，并加入一个用于收集已处理控制顶点 $\widehat{x_i}$ 的额外连通分量。由于这个连通分量单调增长，它会充当时序游走的汇点，也就是说，一旦进入该连通分量，就无法离开。此外，包含尚未处理的控制顶点的第一个连通分量单调缩小，因此从时序游走的起始顶点无法到达它。其思路是，最后一个控制顶点 $\widehat{c}$ 只会在最后一个时间步离开第一个分量进入变量分量，从而迫使我们不进入已处理控制顶点的汇点分量，进而迫使探索调度在每个奇数时间步 $t \geq 3$ 返回变量分量。我们通过如下方式定义快照图 G_j（对于 $3 < j < 2n+1$），将此形式化。

首先考虑 j 为偶数的情形。我们定义 $E(G_j) = K(C_{j,1}) \cup K(C_{j,2}) \cup K(C_{j,3}) \cup K(C_{j,4}) \cup K(C_{j,5})$，其中 $C_{j,1} = \{\widehat{c}\} \cup \{\widehat{x_i} \mid x_i \in X, i \geq \frac{j}{2}\}$，$C_{j,2} = \{c_1, c_2, \ldots, c_m\} \setminus \{c \in \varphi \mid x_{\frac{j}{2}} \in c \vee \neg x_{\frac{j}{2}} \in c\}$，$C_{j,3} = (\{x_i, \neg x_i \mid x_i \in X\} \setminus \{\neg x_{\frac{j}{2}}\}) \cup \{c \in \varphi \mid x_{\frac{j}{2}} \in c\}$，$C_{j,4} = \{\neg x_{\frac{j}{2}}\} \cup \{c \in \varphi \mid \neg x_{\frac{j}{2}} \in c\}$，以及 $C_{j,5} = \{\widehat{x_i} \mid x_i \in X, i < \frac{j}{2}\}$。

对于 j 为奇数的情况，我们定义 $E(G_j) = K(C_{j,1}) \cup K(C_{j,2}) \cup K(C_{j,3}) \cup K(C_{j,4}) \cup K(C_{j,5})$，其中 $C_{j,1} = \{\widehat{c}\} \cup \{\widehat{x_i} \mid x_i \in X, i > \frac{j}{2}\}$, $C_{j,2} = \{c_1, c_2, \ldots, c_m\}$, $C_{j,3} = \{x_i, \neg x_i \mid x_i \in X\} \cup \{\widehat{x_{\frac{j-1}{2}}}\}$, $C_{j,4} = \emptyset$，以及 $C_{j,5} = \{\widehat{x_i} \mid x_i \in X, i < \frac{j-1}{2}\}$。

最后，对于最后一个时间步 $L = 2n+1$，我们定义 $E(G_L) = K(C_{L,1}) \cup K(C_{L,2}) \cup K(C_{L,3}) \cup K(C_{L,4}) \cup K(C_{L,5})$，其中 $C_{L,1} = \emptyset$, $C_{L,2} = \{c_1, c_2, \ldots, c_m\}$, $C_{L,3} = \{x_i, \neg x_i \mid x_i \in X\} \cup \{\widehat{c}\}$, $C_{j,4} = \emptyset$，以及 $C_{j,5} = \{\widehat{x_i} \mid x_i \in X\}$。

现在来分析一种可能的 $\mathcal{G}$ 探索调度会是什么样。回想一下，对于每个时间步 t，第一连通分量 $C_{t,1}$ 都在单调缩小，也就是说，对于 $t, t' \in [L]$，若满足 $t \leq t'$，则有 $C_{t,1} \supseteq C_{t',1}$。由于探索调度的起始顶点位于第三连通分量中，这意味着：(i) 任何探索调度都无法访问第一连通分量，因为连通分量 $C_{t,1}$ 始终不会与任何连通分量 $C_{t-1,\ell}$ 有非空交集，其中 $1 < t \leq L$，且 $\ell \neq 1$。

此外，回想一下，第五连通分量 $C_{t,5}$ 在单调增长，也就是说，对于 $t, t' \in [L]$，若满足 $t \leq t'$，则有 $C_{t,1} \subseteq C_{t',1}$。因此，(ii) 探索调度无法离开任何第五连通分量。

现在考虑顶点 $\widehat{c}$。对于所有时间步 $t \in [L-1]$，该顶点都包含在第一连通分量 $C_{t,1}$ 中，因此无法从起始顶点到达。$\widehat{c}$ 位于其他连通分量中的唯一时间步是最后一个时间步 $t = 2n+1$，此时 $\widehat{c}$ 与变量顶点一起包含在第三连通分量 $C_{t,3}$ 中。由于探索调度需要到达 $\widehat{c}$，而 $\widehat{c}$ 从未包含在第五连通分量中，根据观察 (ii)，我们绝不能进入第五连通分量。由于 $\{\widehat{x_i} \mid x_i \in X\} \cup \{\widehat{x}\}$ 中的所有元素只会出现在第一、第三或第五连通分量中，根据观察 (i)，(iii) 我们需要在第三连通分量中访问它们。

我们还需要访问 $\{c_1, c_2, \ldots\ldots, c_m\}$ 中的所有元素。假设我们在某个时间步 t，首次访问了某些位于第二个连通分量中的元素。根据$\mathcal{G}$的构造，第二个连通分量只能在奇数时间步从第三个或第四个连通分量到达

的连通分量。请注意，奇数时间步 t 是控制顶点 $\widehat{x_{\frac{t-1}{2}}}$ 位于第三个连通分量中的唯一时间步。根据假设，探索调度会在时间步 t 访问第二个连通分量，因此不会在时间步 t访问$\widehat{x_{\frac{t-1}{2}}}$。但是，观察结果 (iii) 表明，此时 $\widehat{x_{\frac{t-1}{2}}}$无法在剩余的游走中到达。因此，任何探索调度都无法访问任何第二个连通分量，而顶点$\{c_1，\ c_2，\ .....，\ c_m\}$ 必须在偶数时间步于第三个或第四个连通分量中访问。

我们已经观察到，在任何奇数时间步，任何探索调度都必须位于第三个分量中。从这个分量出发，只有通过前往包含某个文字（该文字属于 c_j）的分量，才能访问某个顶点 c_j。在每个偶数时间步 t，可以访问那些包含变量 $x_{\frac{t}{2}}$的文字的子句顶点 c_j。因此，我们可以选择访问包含正文字的那些顶点，或包含负文字的那些顶点。访问所有子句顶点 $\{\bar{c_1}、\ c_2、...$ ...、$c_m\}$ 的探索调度，由此对应于 φ的一个满足赋值。反过来，φ 的一个满足赋值可以给出针对$\mathcal{G}$ 的探索调度：在时间步 x_{2i}前往包含变量 x_i 的满足文字的分量。◀

现在我们将重点转向限制底层图的图类。我们证明，即使将 NS-TEXP 限制在底层图为由一条边连接的两个星形图组成的树（换言之，即深度为二的树）的时序图上，它仍然是NP困难的。此类树的顶点覆盖数为二。作为推论，只需对构造作简单调整即可得到：即使每条边至多出现一次，或至多在一个时间步中不存在，且底层图是一棵树，NS-TEXP 仍然是 NP困难的。

▶**定理 9**。 对于底层图由两个通过一条桥相连的星图构成，或底层图为深度为二的树的时序图，*NS-TEXP* 是 *NP* 完全问题。

证明。 我们从 单调 3-SAT [24, 30], 进行归约。该问题询问：给定一个合取范式布尔公式，每个子句含有三个文字（3-CNF），且每个子句要么只含正文字，要么只含负文字（单调的），是否存在满足赋值。令 $\varphi = \{c_1，\ c_2，\ ..，\ c_m\}$ 为定义在变量集合 $X = \{x_1，\ x_2，\ ..，\ x_n\}$上的单调 3-CNF 布尔公式。我们从 φ构造一个时序图 $\mathcal{G}$，其底层图满足所述限制，并且 $\mathcal{G}$ 存在一条从某个顶点 v 出发、访问 $V(\mathcal{G})$ 中所有顶点的非严格时序游走，当且仅当 φ 可满足。

令 $V(\mathcal{G}) = \{c_1，\ c_2，\ ...，\ c_m\} \cup \{\top，\ \bot\}$。顶点 $\top$ 和 $\bot$分别作为两个星图的中心。我们在底层图中将 $\top$与每个对应于正子句（即只含正文字的子句）的顶点 c_i 相连。同样，我们在底层图中将 $\bot$ 与每个对应于负子句的顶点 c_j 相连。最后，我们用一条边连接 $\top$ 和 $\bot$，并将 $\top$ 设为起始顶点 v。

φ 的变量并非表示为顶点，而是按如下方式编码在时间步中。在 $[2n]$ 的每个奇数时间步，我们只保留 $\top$ 和 $\bot$ 之间的边。在每个偶数时间步 $t \in [2n]$，我们只保留连接 $\top$ 与包含正文字$x_{t/2}$的子句 c_i 的边，以及连接 $\bot$ 与包含负文字$\neg x_{t/2}$的子句 c_j 的边。注意，由于 φ 是单调的，每个子句顶点的度为一；又由于 φ 属于3-CNF，每条与子句顶点相连的边最多出现三次。此外，构造出的底层图由两个星图和一条连接它们的桥组成。

直观地说，在每个奇数时间步 t，我们可以从 $\top$ 移动到 $\bot$，反之亦然，从而选择变量 $x_{(t+1)/2}$ 的赋值。接下来的偶数时间步中，我们便可以访问所有包含在所选赋值下为真的文字的子句。显然，如果 φ 可满足，那么上述策略会产生一条访问

所有顶点的非严格时序游走。反过来，设 W 是 $\mathcal{G}$ 中一条访问了 V（$\mathcal{G}$）所有顶点的非严格时序游走。我们可以据此从 W 为 φ 得到一个变量赋值，方法如下。对于每个偶数时间步 $t \in [2n]$，如果 W 在时间步 t 所属的分量中包含 $\top$，则将 $x_{t/2}$ 赋值为真；如果 W 在时间步 t 所属的分量中包含 $\bot$，则将 $x_{t/2}$ 赋值为假；如果 W 属于其他任意（单顶点）分量，则任意将其赋值为真或假。直观而言，后一种情况对应的是变量 $x_{t/2}$ 的赋值与 φ 的可满足性无关。根据假设，W 访问了所有顶点，因此也访问了所有子句顶点。我们根据对子句顶点的访问情况，从 W 读出变量赋值，显然所得的变量赋值满足 φ。◀

▶**推论 10**。 即使输入时序图中的每条边至多出现一次，*NS-TEXP* 仍然是 *NP* 完全的。

证明。 在定理 9 的构造中，子句顶点与 {⊤、⊥} 之间的边最多出现三次。唯一出现线性次数的边是 {⊤、⊥}。我们通过删除边 {⊤、⊥} 并引入 n 个新顶点 v_1、v_3……、v_{2n-1} 对构造稍作修改，这些新顶点分别通过一条边与 ⊤ 相连，并通过一条边与 ⊥相连。然后，在每个奇数时间步 $t \in [2n]$，我们只让与 v_t 关联的边存在。这样，这些边只会出现一次。容易观察到，如果构造出的图存在一个探索调度，那么就存在一个不会在子句顶点处等待的探索调度。因此，我们会在奇数时间步访问与 ⊤ 和 ⊥ 相连的新顶点。

接下来需要将连接 ⊤ 和 ⊥ 的边替换为与子句顶点相连的边。这些边至多出现三次。同样地，对于原始边出现的每个时间步，我们都将该边替换为一条长度为二的路径。不同于 ⊥ 和 ⊤之间的边，我们无法保证在该边出现的时间步中位于包含与子句顶点相 incident 的边的连通分量中。因此，我们无法保证访问所有与子句顶点相邻的新引入顶点。为了解决这一问题，我们在序列末尾再增加一个时间步。我们将与子句顶点相邻的新引入顶点连接成一个环，该环的边仅在这最后一个时间步中存在。这样，我们便能确保在最后访问所有引入的顶点，同时不会访问其他子句顶点。

请注意，这一修改产生的底层图不再是树。 ◀

▶**推论 11**。 即使底层图是一棵树，并且输入时序图的每条边至多在一个时间步中不存在，*NS-TEXP* 仍是 *NP* 完全。

证明。 我们从定理 9 的证明中的构造开始。回想一下，所构造的时序图的生命周期为 $2n$。现在，我们将每条边替换为一条长度为 $4n$ 的路径（使用新顶点）。我们按如下方式定义这些路径中各条边的出现情况。设 π 是替换边 e的路径。那么，对于每个 t 中 e 存在的时间步，我们令 π 中的所有边都存在。若 e 在时间步 t不存在，则令 t第 $4n - t$条边和第 π 条边不存在，并令 π 中的其余所有边都存在。因此，时间步 π 无法穿过路径 t。由于不存在的边会随着时间步增加而向路径中部移动，我们可以通过在当前不存在的边的一侧等待，避免穿过路径时出现漏洞。

最后，我们需要确保所有新引入的顶点都能被访问，同时保持归约的性质。显然，通过引入的修改，我们无法访问任何额外的子句顶点。此外，⊥ 与 ⊤ 之间路径上的顶点会在第一个时间步被访问。对于通往子句顶点的路径上的顶点，注意到根据构造，我们最终必须访问所有子句顶点，并且可以在与原始构造相同的时间步访问它们

。由于子句顶点的度为一，每个子句顶点都只有一条路径通向它。因此，访问相应的子句顶点时，这条路径上的所有顶点都会被访问。证明至此完成。◀

NS-TEXP 的下界——核

对于组合参数 $\gamma + L$，问题k-arb NS-TEXP（因此也包括 NS-TEXP）存在一个平凡的固定参数可 tractable 算法，因为该参数限定了搜索树的大小，其中 γ 是分支因子， L 是树的深度。在这棵树中，对于每个时间步，我们都考虑从当前连通分量可达的所有连通分量；搜索从包含起始顶点 v的分量开始。

▶ **观察 12**。*k-arb NS-TEXP* 关于参数 $\gamma + L$ 可在 $O(\gamma^L n^{\mathcal{O}(1)})$时间内求解。

相比之下，通过从命中集归约，我们得到：除非 NP $\subseteq$ coNP/poly，否则 NS-TEXP 在相同参数下不存在多项式核。事实上，我们证明了一个更强的结论：如果 γ 是常数，那么在相同假设下， L 不存在多项式核。

▶**定理 13**。除非 *NP* $\subseteq$ *coNP/poly*，否则对于参数 L 取常数的情形，*NS-TEXP* 不存在多项式核。

我们通过从命中集问题进行归约来证明这一结论。

▶ **定义 14**(命中集)。
输入：全集 $U = \{x_1, x_2, \ldots, x_n\}$，集合族 $S = \{S_1, S_2, \ldots, S_m\} \subseteq U$，整数 k。问题：是否存在 $H \subseteq U$，使得 $|H| \leq k$ 且对所有 $1 \leq i \leq m$都有 $H \cap S_i \neq \emptyset$？

文献 [11, 定理 5.3] 证明了，以全集大小为参数的命中集问题不存在多项式核，除非 NP $\subseteq$ coNP/poly。

证明。令 U $= \{x_1, x_2, \ldots, x_n\}$、S $= \{S_1, S_2, \ldots, S_m\} \subseteq$ U，且 k为命中集问题的一个实例。我们构造一个时序图 G，其顶点集为 V(G) = U $\cup$ S $\cup \bigcup_{1 \leq i \leq k}$ U^i，其中 U^{i}是 U 的一个带标记副本，每个元素 x_j 都标记为 x^i_j。

G 的生命周期 L 被划分为 k 个阶段，每个阶段包括选择命中集中的一个元素 $x \in$ U，以及访问集合 S$\mathcal{S}$ 中包含该元素 x的所有集合所对应的顶点。在每个阶段 k，都会使用带标记副本 $\mathcal{U}^i$ 作为控制顶点，以强制探索调度满足某些结构。与定理 8 类似，我们关注的是连通分量中的顶点，而非其结构。因此，我们将连通分量定义为顶点集合，并假设每个连通分量都构成一个团。与定理 8 一样，在整个构造过程中，我们保持连通分量的数量为常数。

现在，我们直观地描述时间步 t 时快照图中每个连通分量所包含的顶点。第一个连通分量 $C_{t,\ 1}$ 包含 $\bigcup_{1 \leq i \leq k} \mathcal{U}^i$ 在时间步 t 的未使用控制顶点。在 $\mathcal{G}$ 的生命周期内，这个分量只会缩小。第二个连通分量 $C_{t,\ 2}$ 包含来自$\mathcal{S}$ 且当前尚未处理的所有元素。最后一个连通分量 $C_{t,\ 6}$ 包含来自$\bigcup_{1 \leq i \leq k} \mathcal{U}^i$ 、已经使用过且在 $\mathcal{G}$ 的生命周期内只会增加的元素。另外三个连通分量用于选取 $\mathcal{U}$ 中的一个元素 x，并在每个 k 阶段处理包含 x 的集合 S。它们的用途如下。在一个阶段开始时，第三个连通分量$C_{t,\ 3}$ 包含$\mathcal{U}$中的所有元素。然后，在一个

阶段中，在每个时间步，都会将一个元素 $x \in \mathcal{U}$ 从第三个分量 $C_{t,3}$ 移到第四个分量 $C_{t,4}$，并将 S 中包含 x 的所有集合一并移过去。下一时间步，x 被移到第五个连通分量 $C_{t+1,5}$，并在那里停留到该阶段结束。现在，我们更正式地定义分量 $C_{t,3}$、$C_{t,4}$ 和 $C_{t,5}$。

每个阶段包含 $(n+1)$ 个时间步。令 $t=(i-1)\cdot(n+1)$，对于 $i \in [k]$，这是第 i 个阶段的第一个时间步。在元素 x 上停留，直到它在接下来的 n 个时间步内离开第三个分量，从而选取 U 中的第 i 个元素。因此，在时间步 t，$C_{t,3} = \mathsf{U} \cup \{x_n^{i-1} \mid \text{当且仅当 } i-1>0\}$，即 U 中的所有元素均可用，且当前没有元素处于已处理状态。添加元素 x_n^{i-1} 的作用稍后会变得清晰。可以将其视为来自上一阶段的悬垂部分。连通分量 $C_{t,4}$ 和 $C_{t,5}$ 为空。对于满足 $j \in [n]$ 的时间步 $t+j$，我们定义 $C_{t+j,3} = \{x_\ell \in \mathsf{U} \mid \ell > j\} \cup \{x_{j-1}^i \mid \text{当且仅当 } j-1>0\}$，$C_{t+j,4} = \{x_j\} \cup \{S \in \mathsf{S} \mid x_j \in S\}$，以及 $C_{t+j,5} = \{x_\ell \in \mathsf{U} \mid \ell < j\} \cup \{x_j^i \mid \text{当且仅当 } j<n\}$。

最后，我们定义最后一个时间步的 $t=k\cdot(n+1)$、$C_{t,1}=\emptyset$、$C_{t,2}=\mathsf{S}$、$C_{t,3}=\mathsf{U} \cup \{x_n^k\}$、$C_{t,4}=\emptyset$、$C_{t,5}=\emptyset$，以及 $C_{t,6} = \bigcup_{1\le i\le k} \mathsf{U}^i \setminus \{x_n^k\}$。将任意 $x \in C_{1,3}$ 设为起始顶点后，我们便完成了该归约的构造。注意，$L = k\cdot(n+1)+1$，其规模为 $\mathsf{O}(n^2)$。

显然，若存在一个大小为 k 的命中集 H，那么固定 H 中元素的顺序，并在 $\mathcal{G}$ 的第 i 个阶段跟随第 i 个元素，就能描述一条穿过 $\mathcal{G}$ 的非严格时序游走；该游走会访问 $\mathcal{S}$ 中的所有元素（因为 H 是一个命中集），以及 $\mathcal{U}$ 中的所有元素（因为我们从分量 $C_{1,3}$ 出发）。所谓沿着图中的元素 x 前进，是指我们总是移动到包含 x 的分量。根据以下观察，我们还会看到 $\mathcal{U}^i$ 中所有满足 $1 \le i \le k$ 的元素。设 x_i 为 H 中第 i 个元素。那么，在第 i 个阶段，我们沿着时序图中的顶点 x_i 前进。这样，我们会在分量 $C_{t+\ell,3}$ 的时间步 $t+\ell$ 看到满足 $\ell < i$ 的顶点 x_ℓ^i，并在分量 $C_{t+\ell'+1,5}$ 的时间步 $t+\ell'+1$ 看到满足 $i \le \ell'$ 的顶点 $x_{\ell'}$；但 x_n^i 除外，它会在阶段 $i+1$ 开始时于第三个分量中被看到。

对于另一个方向，假设 W 是 G 的一个探索调度。注意，由于 W 从分量 $C_{1,3}$ 开始，因此它永远无法到达任何 t 对应的分量 $C_{t,1}$，因为第一个分量只会缩小。此外，如果 W 访问了一个分量 $C_{t,6}$，那么对于所有 $t' > t$，它只能访问分量 $C_{t',6}$，因为第六个分量只会增大。由于元素 x_n^k 仅在最后一步才被处理，因此它永远不会到达第六个分量。因此，在任何访问第六个分量的时序游走中，我们都无法到达 x_n^k。所以，我们只能通过分量 $C_{t,2} - C_{t,5}$ 访问所有顶点。接下来，我们证明，任何探索调度都无法到达第二个分量 $C_{t,2}$。注意，由于我们既不能访问分量 $C_{t,1}$，也不能访问分量 $C_{t,6}$，因此必须在元素从 $\bigcup_{1\le i\le k}\mathsf{U}^i$ 经过分量 $C_{t,3}$ 或 $C_{t,5}$ 时将其“截获”。如果我们漏掉了分量 $C_{t,3}$ 中的一个元素 x_ℓ^i，那么要访问该元素，唯一的选择就是在分量 $C_{t+1,5}$ 中访问它。因此，如果我们访问了任何分量 $C_{t,4}$（从而漏掉了 $C_{t,3}$），就必须在下一步访问分量 $C_{t+1,5}$，否则便无法再访问所有顶点。由于我们只能经由分量 $C_{t-1,4}$ 访问任何第二个分量 $C_{t,2}$，这意味着在探索调度中无法到达任何第二个分量；此外，若不漏掉来自 $\bigcup_{1\le i\le k}\mathsf{U}^i$ 的元素，就不能连续访问两个第四分量 $C_{t,4}, C_{t+1,4}$。由于在一个阶段内第五个分量只会增大，无法连续两次访问第四个分量这一事实意味着，在同一阶段内无法访问任意两个第四分量。因此，我们可以将 U 中出现在 W 的第四个分量里的元素收集到一个集合 H 中。根据前面的论证，可得 $|H| \le k$；并且由于我们无法访问任何第二个分量，因此 H 与 S 中的每个集合 S 都有非空交集。换言之，H 是 S 的一个命中集。◀

NS-TEXP 实例的大小可以由 $n^2 \times L$ 界定，并且对于参数 L 以及参数 n [20]，都存在固定参数可解算法。我们已经证明，在标准复杂性假设下，参数 L 不存在多项式核。下一个结果是通过使用命题 7 将 NS-TEXP 交叉组合到其自身而得到的。

▶ **定理 15.** 除非 *coNP* $\subseteq$*NP/poly*，否则以 n 为参数的 *NS-TEXP* 不存在多项式核。

证明。我们通过从 NS-TEXP 到其自身的交叉组合，利用命题 7 证明这一结论。因此，我们将 ℓ 个 NS-TEXP 实例 $(\mathsf{G}1, v^1), (\mathsf{G}2, v^2) \ldots (\mathsf{G}_\ell, v^\ell)$ 归约为一个 NS-TEXP 实例 $(\mathsf{G}\ v)$，使得 $(\mathsf{G}\ v)$ 是是实例，当且仅当实例 (G_i, v^i) 中至少有一个对于 $1 \leq i \leq \ell$ 是是实例。首先，我们证明可以对实例 (G_i, v^i)，$1 \leq i \leq \ell$ 进行变换，使每个实例都具有相同数量的顶点。令 $n = \max_i |V(\mathsf{G}_i)|$。如果对于某个 $i, |V(\mathsf{G}_i)| < n$，我们向 G_i 添加一个包含 S 个 $n - |V(\mathsf{G}_i)|$ 新顶点的集合，并在 G_i 的快照图序列前添加一个新的快照图，其中包含两个连通分量，一个包含 $\{v^i\} \cup S$，另一个包含其余顶点。对于 G_i 的所有其他快照图，我们将 S 添加为一个连通分量。我们将 G'_i 称为新得到的实例。显然，G'_i 是是实例，当且仅当 G_i 是是实例。因此，以下我们假设所有实例都具有相同数量的顶点；于是，不失一般性，我们可以重新命名顶点集$V(\mathsf{G}_i)$，并假设所有实例具有相同的顶点集和相同的起始顶点 v_1。从现在起，令 $(\mathsf{G}1, v_1), (\mathsf{G}2, v_1) \ldots (\mathsf{G}_\ell, v_1)$ 为顶点集 $V = \{v_1, v_2, \ldots, v_n\}$ 上的 ℓ 个 NS-TEXP 实例。

该构造的思想是利用各个 ℓ 实例共享同一个顶点集，将它们依次拼接起来。随后，我们引入额外的顶点，使得我们能够选择并进入恰好一个实例，并且当且仅当所选实例是一个是实例时，才能访问 V 中的所有顶点。因此，这些额外的顶点确保我们只能访问其中一个实例中的 V 顶点，否则将无法访问所有额外顶点。

现在，我们描述在 G 中实现实例选择的构件。首先，我们介绍如何利用 2 个控制顶点和一个 V的副本，在两个实例 $(\mathsf{G}1, v_1)$ 和 $(\mathsf{G}2, v_1)$ 之间强制进行 V'选择。令 $V' = \{v_1', v_2', \ldots, v_n'\}$ 以及 $x_1, y_2 \notin V$ 为额外顶点。令 $G_1^1, G_2^1, \ldots, G_{L_1}^1$ 为 G_1 的快照序列，令 $G_1^2, G_2^2, \ldots, G_{L_2}^2$为 G_2的快照序列。我们定义 G 如下：$V(\mathsf{G}) = V \cup V' \cup \{x_1, y_1\}$，$L = L_1 + L_2 + 8$，并且$G_1, G_2, \ldots, G_L$是 G 的快照序列，其中在前 $L_1 + 4$ 个快照中，有$E(G_1) = \{\{x_1, v_1\}\}$，$E(G_2) = \{\{x_1, y_1\}\}$，$E(G_{i+2}) = E(G_i^1)$ 对所有 $i \in [L_1]$，$E(G_{L_1+3}) = \{\{v_j, v_j'\} \mid j \in [|V|]\}$，以及 $E(G_{L_1+4}) = \{\{v_j', x_1\} \mid j \in [|V|]\}$。然后，对于剩余的 $L_2 + 4$个快照，我们用 G_2 替代 G_1，重复相同的序列。因此，我们有，$E(G_{L_1+5}) = E(G_1) = \{\{x_1, v_1\}\}$，$E(G_{L_1+6}) = E(G_2) = \{\{x_1, y_1\}\}$，$E(G_{L_1+6+i}) = E(G_i^2)$ 对所有 $i \in [L_2]$，$E(G_{L_1+L_2+7}) = E(G_{L_1+3}) = \{\{v_j, v_j'\} \mid j \in [|V|]\}$，以及 $E(G_{L_1+L_2+8}) = E(G_{L_1+4}) = \{\{v_j', x_1\} \mid j \in [|V|]\}$。构件示意图见图 1。为完成 $(\mathsf{G}\ v)$ 作为 NS-TEXP 实例的构造，我们将起始顶点设为 $v = x_1$。

直观地说，$(\mathcal{G}\ x_1)$中的一次探索过程如下。我们从 x_1开始。然后，在快照 G_1中有两种选择：我们可以前往 v_1，并通过求解实例$\mathcal{G}_1$ 访问 V；或者留在 x_1，并在 G_2中访问 y_1。无论哪种情况，在$\mathcal{G}$中对应于$\mathcal{G}_1$ 的快照之后，我们都必须留在快照 G_{L_1+4}中的 x_1。对于序列的后半部分，我们执行与前半部分互补的操作。如果我们通过$\mathcal{G}_1$, 访问了 V，那么我们就留在 x_1 以访问 y_1；否则，我们前往 v_1，并通过求解$\mathcal{G}_2$来探索 V。

现在我们正式证明：$(\mathcal{G}\ x_1)$是是实例，当且仅当$(\mathcal{G}_1, v_1)$或$(\mathcal{G}_2, v_1)$是是实例。

我们先证明“如果”方向。假设 $(G1, v_1)$ 或 $(G2, v_1)$ 是一个是实例。回忆一下，$V' \cup \{x_1\}$ 在 $G_{L_1+4} = G_{L_1+L_2+8}$ 中构成一个连通分量。令 W' 是 $G_{L_1+4} = G_{L_1+L_2+8}$ 上的一条 (v'_1, v'_n)-游走，使得 $V(W') = V'$。假设 $(G1, v_1)$ 是一个是实例。根据观察 1，存在某个 $u \in V$，使得一条单调的 (v_1, u)-游走 W_1 满足 $V(W_1) = V$。由 W_1，我们按如下方式构造一条单调的 (x_1, y_1)-游走 W，它访问 $V \cup V' \cup \{x_1, y_1\}$ 中的所有顶点：$W = x_1$、x_1v_1、W_1、uu'、u'、$u'v'_1$、W'、v'_nx_1、x_1、x_1y_1、y_1。显然，W 是一条单调游走，并且 $V(W) = V \cup V' \cup \{x_1, y_1\}$。因此，$(G\ x_1)$ 是一个是实例。现在，假设 $(G2, v_1)$ 是一个是实例。根据观察 1，存在某个 $u \in V$，使得一条单调的 (v_1, u)-游走 W_2 满足 $V(W_2) = V$。由 W_2，我们按如下方式构造一条单调的 (x_1, v'_n)-游走 W，它访问 $V \cup V' \cup \{x_1, y_1\}$ 中的所有顶点：$W = x_1$、x_1y_1、y_1、y_1x_1、x_1、x_1v_1、W_1、uu'、u'、$u'v'_1$、W'。显然，W 是一条单调游走，并且 $V(W) = V \cup V' \cup \{x_1, y_1\}$。因此，$(G\ x_1)$ 是一个是实例。

现在继续证明“仅当”方向。假设 $(G\ x_1)$ 是一个是实例。根据观察 1，存在一条单调的 (x_1, u)-游走 W，其中 $u \in V \cup V' \cup \{x_1, y_1\}$，使得 $V(W) = V \cup V' \cup \{x_1, y_1\}$。游走 W 必须在快照 G_2 或快照 G_{L_1+6} 中经过 y_1。假设该游走在快照 G_2 中经过 y_1，那么不难看出，该游走必须在 G_2 中返回 x_1。从那里出发，为了能够到达 V 中的顶点，游走必须在 G_{L_1+5} 中前往 v_1。在时间步 L_1+7 到 L_1+L_2+6 之间，G 在 V 上的表现与 G_2 相同。因此，V 中的顶点由 W 的一个子游走 W_2 访问。根据 G 的构造，W_2 是 G_2 上一条从 v_1 出发的单调游走。因此，$(G2, v_1)$ 是一个是实例。对称地，如果该游走在 G_{L_1+6} 中经过 y_1，那么该游走必须在时间步 3 到 L_1+3 之间的一个子游走 W_1 中经过 V。游走 W_1 对应于 G_1 上的一条单调游走。因此，$(G1, v_1)$ 是一个是实例。

事实上，图1中的构件可以递归嵌套，从而在更多选项中进行选择。为此，我们将图1中包含来自 $V \cup V'$ 的顶点的虚线方框视为两个黑盒；它们具有这样的性质：我们可以分别在快照 G_1 和 G_{L_1+5} 中通过顶点 x_1 进入黑盒，遍历黑盒，然后分别在快照 G_{L_1+4} 和 $G_{L_1+L_2+8}$ 中从黑盒退出到顶点 x_1。关键的观察是，一次探索只能访问第一个黑盒或第二个黑盒，而不能同时访问两者，否则就不会访问顶点 y_1。我们可以利用这一性质，将构件嵌套到自身之中，以便在两个以上的实例中进行选择。图2展示了如何将构件嵌套到自身之中：把构件作为黑盒插入，由此得到一个可在四种不同选择中进行选择的两层构件。对于四个实例，图2中的方框由图1中的虚线方框实现；对于八个实例，这些方框则再次由图1中的构件实现。递归嵌套如何继续很容易看出。递归嵌套会引入 $d2\log(\ell)e$ 个用于选择实例的控制顶点，以及 $|V|$ 个控制变量，作为最高层上的副本 V'。通过递归嵌套这个构件，我们保留了这样一种性质：在每一层上，一次探索只能访问其中恰好一个黑盒，因此总的来说，一次探索只能访问实例 (G_i, v_1) 中恰好一个实例，其中 $1 \le i \le \ell$；否则，对于 $1 \le j \le d2\log(\ell)e$，某个控制变量 y_j 将不会被访问。与两个实例的情形类似，我们可以得到：如果某个实例 (G_i, v^i) 存在探索，其中 $1 \le i \le \ell$，那么 $(G\ v)$ 也存在探索；反过来，如果 $(G\ v)$ 存在探索，那么这次探索会通过恰好进入实例 (G_i, v^i) 中的一个，访问 V 中的顶点，其中 $1 \le i \le \ell$。因此，$(G\ v)$ 是一个是实例，当且仅当存在某个 $1 \le i \le \ell$，使得 (G_i, v^i) 是一个是实例。

◀

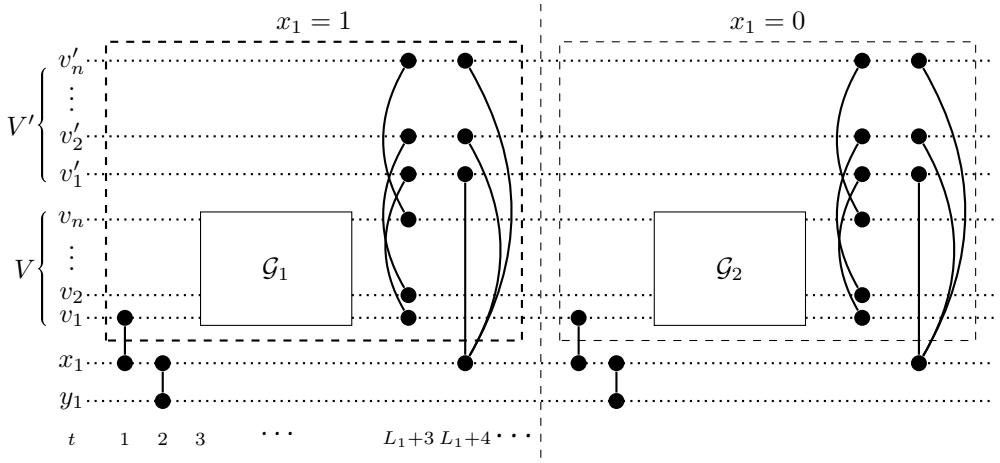

图1 定理 15 的构件：给定两个 NS-EXP 实例（$\mathcal{G}_1$， v_1）和（$\mathcal{G}_2$， v_1），该构造展示了如何将它们连接起来，生成一个实例（$\mathcal{G}$ x_1），使得（$\mathcal{G}$ x_1）是一个是实例，当且仅当（$\mathcal{G}_1$， v_1）或（$\mathcal{G}_2$， v_1）是一个是实例。

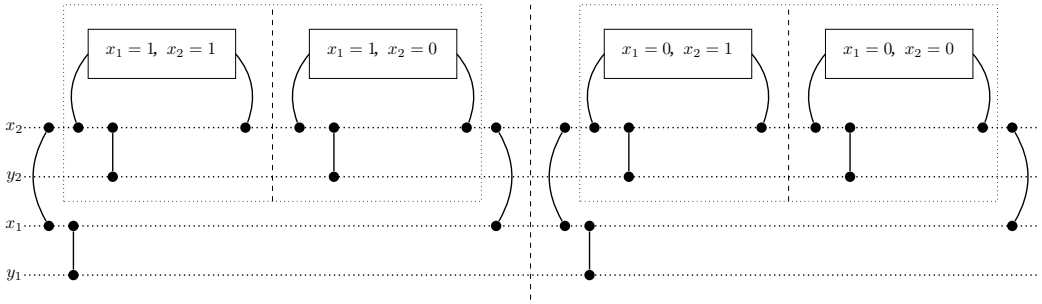

图2 定理15的装置：这展示了如何将图1的构件嵌套到一个两层构件中，从而在四种选择之间进行选择。

k-arb NS-TEXP 的下界

在 [20] 中，针对参数为 k 的问题 k-arb NS-TEXP，给出了一种基于彩色编码的固定参数可解算法。我们将在下文证明，参数化于 L 的 k-arb NS-TEXP 是 W[1]-困难的，从而解决了 [20] 中提出的一个开放问题。这一困难性与问题 NS-TEXP 关于参数 L 的固定参数可解算法形成对比。我们进一步证明，除非 coNP $\subseteq$ NP/poly，否则 k-arb NS-TEXP 关于 $L+k$ 不存在多项式核。该结果基于从问题 k-arb NS-TEXP 到其自身的交叉组合。

▶**定理 16**。 *k-arb NS-TEXP 关于参数 L 是 W[1]-困难的。*

我们通过从多色独立集进行归约来证明该结论。

▶**定义 17** (多色独立集)。输入： k部图 $G=(V_1,V_2,\ldots,V_k,E)$。问题：是否存在一个顶点集合 $S=\{v_1,v_2,\ldots,v_k\}$，使得对于 $1\leq i\leq k$ 有 $v_i\in V_i$，并且 S 是 G 中的独立集？

文献 [10] 证明了，即使在 d正则图（每个顶点的度为 d的图）上，且 d 固定时，多色独立集问题仍然是关于 k参数化的 W[1]-难问题。

证明。令 $G=(V_1,V_2,\ldots,V_k,E)$为一个 d正则图，也是多色独立集问题的一个实例。首先，我们在 E 中对每条边进行细分，引入一组新的顶点 V_E。令 $G'=(V_1,V_2,\ldots,V_k,V_E,E')$为对每条边细分后得到的图。注意，在 G' 中， $\bigcup_{1\leq i\leq k}V_i$ 中的每个顶点只有来自 V_E的邻居。此外，由于 G 是 d正则图，对于任意集合 $H=\{v_1,v_2,\ldots v_k\mid v_i\in V_i,1\leq i\leq k\}$，都有 $|\bigcup_{v\in H}N_{G'}(v)|=kd$，当且仅当 H 是 G中的独立集。为此，注意若 H中的两个顶点 u,v 在 G中相邻，则它们在 G'中有一个共同邻居。由于每个集合 $N_{G'}(v)$的大小为 d （$v\in\bigcup_{1\leq i\leq k}V_i$），等式 $|\bigcup_{v\in H}N_{G'}(v)|=kd$ 仅在集合 $N_{G'}(v)$两两不相交时才对 $|H|=k$ 成立。我们还注意到，每个顶点 $v\in V_E$ 的度为二；并且由于 G 是一个k部图， v 的两个邻居位于不同的集合 $V_i\neq V_j$中。

现在，我们将根据以下思路构造一个非严格时序图$\mathcal{G}$，其基于 G'。在时间步 $t=(i-1)\cdot 2$ 中，我们可以从 V_i 中选择一个顶点，并在下一步访问其在 V_E 中的所有邻居。如果我们以这种方式选择的 k 顶点构成一个独立集，那么我们就能够访问 V_E 中的 kd 个顶点；否则，我们将两次看到 V_E 中的某个顶点。

现在，我们将更加正式地描述该构造。与定理 8 类似，我们关注的是连通分量中的顶点，而不是连通分量的结构。因此，我们将连通分量定义为顶点集合，并假设每个连通分量都构成一个团。对于 $t=0$，令 $C_{0,1}=V_1$ 为一个连通分量，其他每个顶点均为孤立顶点。此外，我们将 $v\in C_{0,1}$ 设为所求时序游走的起始顶点。现在考虑 $t=1$。令 $V_1=\{v_1^1,v_2^1,\ldots,v_{\ell_1}^1\}$。然后，对于 $1\leq j\leq\ell_1$，令 $C_{1,j}=\{v_j^1\}\cup N_{G'}(v_j^1)$ 为一个连通分量。其他每个顶点均为孤立顶点。现在，令 $t=(i-1)\cdot 2$ for$2\leq i\leq k$。然后，我们将 $C_{t,1}=V_i\cup V_{i-1}$ 设为一个连通分量，其他每个顶点均为孤立顶点。对于时间步 $t+1$，令 $V_i=\{v_1^i,v_2^i,\ldots,v_{\ell_i}^i\}$。然后，对于 $1\leq j\leq\ell_i$，我们将 $C_{1,j}=\{v_j^i\}\cup N_{G'}(v_j^i)$ 设为一个连通分量，其他每个顶点均为孤立顶点。

注意到 $L=2k$。接下来，将参数 k' 设为时序游走中需要访问的顶点数。我们设 $k'=|\bigcup_{1\leq i\leq k}V_i|+kd$。

现在假设存在一个多色独立集 H，对应于 G。对于 $1\leq i\leq k$，令 $v_i\in H$ 为包含在 V_i 中的顶点。接下来，在时间步 $2(i-1)$ 和 $2(i-1)$ 跟随顶点 v_i+1，即可访问 $N_{G'}(H)$ 中的所有顶点；根据 H 是一个独立集这一假设，这些顶点共有 kd 个。由于 V_i 和 V_{i+1} 中的顶点在时间

步 $2i$ 中共享一个分量，我们便能够在时间步 $2i$ 将跟随顶点 v_i 改为跟随顶点 v_{i+1}，并进一步访问 $\bigcup_{1\leq i\leq k} V_i$ 中的所有顶点。

对于另一个方向，假设在$\mathcal{G}$ 中存在一条单调游走 W，它至少访问了 $k' = |\bigcup_{1\leq i\leq k} V_i| + dk$个顶点。回顾一下， V_E中的每个顶点至多有两个邻居，并且没有两个邻居属于同一个集合 V_i。因此，在每个奇数时间步中，我们至多可以从 V_E访问 d 个新顶点。由于在每个偶数时间步中， V_E 中的顶点都位于单点连通分量中，因此我们只能在奇数时间步访问 V_E 中的顶点。我们声称，**(*)** 在每个偶数时间步中， t、 W 访问第一个分量。这样一来， W 就会访问 $\bigcup_{1\leq i\leq k} V_i$ 中的所有顶点，并且只需访问 V_E中的 kd 个顶点。假设 **(*)** 不成立。那么， W 在时间步 t处于一个单点连通分量中，并且根据构造，在时间步 $t+1$仍处于一个单点连通分量中。但这样一来， W 在时间步 $t+1$ 不会访问 V_E 中的任何顶点；而我们在奇数时间步中至多只能从 V_E 访问 d 个新顶点（偶数时间步中则一个也不能访问），因此在游走过程中，已不可能访问 V_E 中的 dk 个顶点。现在，收集 W 在奇数时间步访问的 $\bigcup_{1\leq i\leq k} V_i$ 中的顶点，就能得到 G的一个多色独立集。◀

▶ **定理 18**。 除非 *coNP* $\subseteq$ *NP/poly* k*-arb NS-TEXP*，否则其关于 $k+L$参数化时不具有多项式核。

证明。我们通过从 k-arb NS-TEXP 到其自身的交叉组合，并利用命题 7，证明该结果。因此，我们将 ℓ 个实例 (G1, v^1, k_1)、(G2, v^2, k_2).....(G$_\ell$, v^ℓ, k_ℓ) 的k-arb NS-TEXP 归约为一个实例 (G v, k) of k-arb NS-TEXP，使得 (G v, k) 是一个是实例，当且仅当实例 (G$_i$, v^i, k_i) 中至少有一个对于 $1 \leq i \leq \ell$ 是是实例。这里，G$_i$ 是一个时序图， v^i 是起始顶点，而 k_i 是在G$_i$中一次时序游走所要访问的顶点数。

不失一般性，我们可以假设 ℓ 具有 2^x 的形式，并且所有顶点集 $V(\mathcal{G}_i)$ 两两不相交。此外，我们可以假设 $k_1 = k_2 = \cdots = k_\ell$。否则，令$k' =$ 为 max($k_1, \ldots, k_\ell$)。然后，对于第 i 个实例，我们可以添加 $k' - k_i$ 个新顶点，使其在所有快照中都与 v_i 相邻。这样既能确保不改变实例的结构，又能达到 $k' - k_i$ 更多顶点。因此，我们可以设置 $k_i := k'$。进一步地，我们可以假设这 ℓ 个时序图的生命周期 $L_1, L_2, \ldots, L_\ell$ 相同；否则，我们可以通过为每个缺失的时间步复制最后一个快照，为生命周期较短的图补齐。因此，从现在起，我们将生命周期 $L_1 = L_2 = \cdots = L_\ell$称为 L'，并将参数$k_1 = k_2 = \cdots = k_\ell$称为 k'。

现在，我们从（$\mathcal{G}_1, v^1, k_1$）、（$\mathcal{G}_2, v^2, k_2$）.....（$\mathcal{G}_\ell, v^\ell, k_\ell$）构造一个实例（$\mathcal{G}$ v, k），其具有$k = k' + 2\log(\ell) - 1$ 和 $L = L' + \log(\ell)$。

我们定义 V(G) $= \{v_1, v_2, \ldots, v_{2\ell-1}\} \uplus \biguplus_{1\leq i\leq \ell} V$(G$_i$)。随后，我们将 G 的最后一个 L 快照构造为 (G$_i$)$_{1\leq i\leq \ell}$中各快照的并集。形式化地，对于 $1 \leq t \leq L$，我们令E(G($t + \log(\ell)$)) $= \biguplus_{1\leq i\leq \ell} E$(G$_i$($t$))。为了定义最初的 log($\ell$) 个时间步，我们使用 $2\ell - 1$ 新顶点 $v_1, v_2, \ldots v_{2\ell-1}$ 来 V(G)。我们将新顶点排列成一棵平衡二叉树，使得该树的根在时间步 $t = 0$ 与其两个子节点相邻，并且在每个时间步 t（对于 $1 \leq t \leq \ell - 1$），该树第 t 层的节点都与其两个子节点相邻。在时间步 $t = \ell$，该树的 ℓ叶节点 $l_1, l_2, \ldots, l_\ell$分别与这些 ℓ 个实例的起始顶点相邻。更准确地说，对于时间步 $t = \ell$，顶点 l_i 与第 i个实例的起始顶点 v^i 相邻。对于上述未指定的其余情况，每个新引入的顶点 $v_1, v_2, \ldots, v_{2\ell-1}$都是一个孤立顶点。

直观来说，在 ℓ 个时序图的并集前添加的二叉树，使我们能够从中选择一个图。在选择过程中，我们会访问新引入的顶点中的 2 log(ℓ) $- 1$ 个，直到在时间步 ℓ 到达一个实例 $\mathcal{G}_i$。从时间步 ℓ

开始，在 $\mathcal{G}$ 中的时序游走等同于某个 $\mathcal{G}_i$ 中的时序游走，具体取决于游走开始时在二叉树中所做的选择i 。因此，显然，当且仅当我们能在某个图$\mathcal{G}_j$中的时序图上访问 k_j 个顶点时，我们才能在 $\mathcal{G}$ 中的时序游走上访问$k_j + 2\log(\ell) - 1$ 个顶点。◀

4 有界边出现次数的多项式核

在静态图领域，树是一类很好的结构，常常能带来高效算法。基于这一事实，人们成功引入了树宽等结构参数，用于刻画与树的距离，并据此设计高效算法。遗憾的是，我们在定理 9 中看到，即使底层图是一棵顶点覆盖数为 2 的树，我们考虑的问题仍然是 NP困难。因此，本节将引入一个新的结构参数 $p(\mathcal{G})$，直观地刻画 $\mathcal{G}$ 距离这样一种树有多远：其中每条边恰好出现一次。利用这一新参数，我们研究 k-arb NS-TEXP 的复杂度。

我们重新陈述 $p(\mathcal{G})$ 的定义。回顾一下，对于时序图 $\mathcal{G} = (G_1, \ldots, G_L)$，我们记作 $\mathfrak{m} = \sum_{i=1}^{L} |E(G_i)|$。注意，若 $\mathfrak{m} < n - 1$，则$\mathcal{G}$的底层图 G 是不连通的。令 X 为包含源顶点 v 的 G 的连通分量的顶点集合。于是，$\mathcal{G}$ 的 k-arb NS-TEXP 等价于时序图 $\mathcal{G}[X] = (G_1[X], \ldots, G_L[X])$ 上的 k-arb NS-TEXP。因此，我们可以假设底层图是连通的，且 $\mathfrak{m} \geq n - 1$。这使我们能够考虑超保证参数$p(\mathcal{G}) = \mathfrak{m} - n + 1$。我们证明，当底层图连通时，$k$-arb NS-TEXP 关于 $p(\mathcal{G})$ 存在一个多项式核。事实上，我们针对该问题一个更一般的变体给出了一个核：带有顶点权重的 Weighted k-arb NS-TEXP。

我们的核化算法以一个多项式时间算法为起点，该算法用于求解时序图上的加权 k-arb NS-TEXP，其中每条边恰好出现一次，且底层图是一棵树（引理 21）。除该算法外，导出核的关键观察是：输入时序图的底层图存在一个大小有界的反馈边集（引理19）。结合这一观察以及边重复次数有界这一事实，我们证明输入时序图$\mathcal{G}$ 具有某种特定结构。更具体地说，我们证明存在一个大小有界的顶点核心集合$X \subseteq V(\mathcal{G})$，使得剩余图$\mathcal{G} - X$ 的底层图是森林$\mathcal{F}$，并满足以下性质：在$\mathcal{G} - X$中，每条边恰好出现一次，且$\mathcal{F}$ 中的每棵树最多通过两条边与 X 相连（见图3和引理20）。对于$\mathcal{F}$中的每棵树 T，根据其结构以及它与 X的相互作用，我们能够描述探索访问 T中部分顶点的所有可能方式。然后，利用适用于边仅出现一次的树的多项式时间算法，我们可以针对上述每种情况计算 T 中的顶点所贡献的最大权重，并设计一个常数大小的构件来模拟原树 T。因此，该构件保留了 T 中可达顶点的数量及其顶点权重信息，这也是我们需要研究加权版本的 k-arb-NS-TEXP的原因。由此得到若干归约规则（规约规则1-4）。为了完成核的构造，我们证明穷尽地应用这些归约规则后，所得时序图的规模关于 p呈线性增长（命题 28）。最后，利用 Frank 和 Tardos 提出的算法（命题 22），我们可以将所得顶点权重的大小限制在$\mathcal{O}(p^4)$以内。

本节中，我们假设所考虑的时序图具有连通底层图。设 $\mathcal{G} = (G_1, \ldots, G_L)$ 为一个时序图，G 为其底层图。对于 $e \in E(G)$，我们用 $A(e) = \{t\colon e \in E(G_t) \text{ for } 1 \leq t \leq L\}$ 表示，并令 $m(e) = |A(e)|$。

注意， $\mathfrak{m}(\mathcal{G}) = \sum_{e \in E(G)} m(e)$。

根据观察 1，带权 k-arb NS-TEXP 的一个实例（$\mathcal{G}$ w， v， k）是肯定实例，当且仅当底层图 G 对某个 $x \in V(G)$ 存在单调的（v， x）-游走 W，使得 $w(V(W)) \geq k$。略微滥用记号，我们用 $w(W)$ 代替 $w(V(W))$，表示游走所访问顶点的总权重。

我们首先证明，底层图具有大小有界的反馈边集。回忆一下，边的集合 $S \subseteq E(G)$ 是图 G 的一个 反馈边集，如果 $G - S$ 是森林。设 $\mathcal{G} = (G_1, \ldots, G_L)$ 是一个时序图，并设 $p = \mathfrak{m} - n + 1$。我们将底层图 G 中至少出现两次的边的集合记为 R，即当存在互异的 $i, j \in [L]$ 使得 $e \in E(G_i)$ 且 $e \in E(G_j)$ 时， $e \in R$。我们将 R 中的边称为 红色边。我们令 $B = E(G) \setminus R$，并将 B 中的边称为蓝色边。显然，每条蓝色边都在 G_i 中唯一的 $i \in [L]$ 处出现。我们可以得到如下观察。

▶ **引理19**。 设 G 是一个时序图，且 $p = \mathfrak{m} - n + 1$。那么，G 的底层图G 有一个大小至多为 p 的反馈边集 S，使得每条红边都属于 S。
证明。 令 R 为 G 的红边集合。考虑 $q = \sum_{e \in R} m(e) - |R| \geq |R|$。注意， G 恰有 $n - 1 + p - q$ 条边。因此， G 有一个反馈边集 S'，其大小为 $p - q$。于是， $S = S' \cup R$ 是 G 的一个反馈边集，其大小至多为 p，且满足 $R \subseteq S$。 ◀

引理19表明， G 具有一种特殊结构，可用于我们的算法中（见图3） .

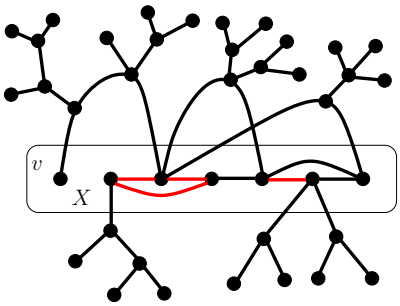

图3 当参数$p = \mathfrak{m} - n + 1$ 有界时，时序图 $\mathcal{G}$ 的底层图 G 的结构。这里 $|X| \leq 4p$。（引理20）

▶ **引理20**。设 G 是一个底层图为 G 的时序图，且 $p = \mathfrak{m} - n + 1$。另设 $v \in V(G)$。那么，存在一个大小至多为 $4p$ 的顶点集合 $X \subseteq V(G)$，使得 *(i)* $v \in X$，*(ii)* 每条红边都属于 $G[X]$，*(iii)* $G - E(G[X])$ 是一片森林，并且 *(iv)* $G - X$ 的每个连通分量 F 都是一棵树，且具有以下性质：每个顶点 $x \in X$ 在 F 中至多有一个邻居，并且

(a) 要么 X 有一个唯一的顶点 x，该顶点在 F中有一个邻居，**(b)** 要么 X 恰好包含两个顶点 x 和 y，它们在 F中有邻居。此外，若 q 是满足 *(b)* 的 F的连通分量数量 $G - X$，则$q \leq 4p - 1$ 并且 $G[X]$ 至多有 $5p - q - 1$ 条边。

证明。根据引理19， G 有一个大小至多为 p 且包含所有红边的反馈边集 S。我们定义 $Y = \{v\}$，其中 $\cup \{x \in V(G) \mid x$ 是 S的一条边的端点$\}$。注意到 $|Y| \leq 2p + 1$。考虑图 G'，它由 G 通过迭代删除不属于 Y的度为一的顶点得到。由于 S 是一个反馈边集，我们得到 $H = G' - S$ 是一个森林，其中每个度至多为一的顶点都属于 Y。众所周知，任何具有 $\ell \geq 2$ 个度为一的顶点（叶）的树，度至少为三的顶点数至多为 $\ell - 2$。

这意味着， Z 中 H 的度至少为三的顶点所组成的集合大小至多为 $2p-1$。我们定义 $X = Y \cup Z$。根据该构造， $|X| \leq 4p$ 以及条件 (i)–(iv) 均得到满足。

令 q 为满足条件 (b) 的 $G - X$ 的连通分量数量。考虑通过以下操作从 $G[X]$ 得到的多重图 G''：对于每个满足条件 (b) 的连通分量 F，其中 x 和 y 是 X 中与 F 的顶点相邻的顶点，将边 xy 添加到 $G[X]$，并在 $xy \in E(G[X])$ 时将其设为重边。根据 (iii)，$G'' - E(G[X])$ 是一棵有 $|X|$ 个顶点的树，因此我们有 $q \leq |X| - 1 \leq 4p - 1$。由于 G 有一个大小至多为 p 的反馈边集， G'' 也具有相同性质。因此， G'' 至多有 $|X| - 1 + p \leq 5p - 1$ 条边。于是， $G[X]$ 至多有 $5p - q - 1$ 条边。◀

根据推论 10，即使时序图中的每条边都只出现一次，NS-TEXP 仍然是 NP 完全的。我们证明，如果每条边恰好出现一次且底层图是一棵树，那么 Weighted k-arb NS-TEXP 可以在多项式时间内求解。如果一条边 e恰好出现一次，那么我们用 $t(e)$ 表示 $A(e)$ 中唯一的元素。

▶**引理 21**。 存在一个运行时间为 $O(nL)$ 的算法，给定一个时序图 $\mathcal{T} = (F_1, \ldots, F_L)$，其底层图 T是一棵树，且对于每个 $e \in E(T)$，都有 $m(e) = 1$，以及一个权重函数 $w\colon V(T) \to \mathbb{Z}_{>0}$和两个顶点 x, $y \in V(T)$，要么找出 T 中单调的(x, y)-游走 W 的最大权重，要么报告不存在这样的游走。

证明。 设$\mathcal{T} = (F_1, \ldots, F_L)$是一个时序图，其底层图 T是一棵树，且 $m(e) = 1$对每个 $e \in E(T)$均成立。设 $x, y \in V(T)\colon V(T) \to \mathbb{Z}_{>0}$是一个权重函数。

若 $y = x$，即 W是一个闭游走，则令 $U = \bigcup_{t \in [L]} V(C_t)$，其中每个 C_t 都是 F_t 中包含 x的连通分量。于是，由于 T 是一棵树，且每条边在$\mathcal{T}$中恰好出现一次， T中最大权重的单调 (x, x)-游走 W恰好经过U 的所有顶点，且 W 的权重为 $w(U)$。由于可以在$\mathcal{O}(n)$时间内求出每个 F_t 的连通分量，因此我们可以构造 U，并在$\mathcal{O}(nL)$时间内计算其权重。

从现在起，我们假设 $y \neq x$。考虑 T中唯一的 (x, y)-路径 $P = v_0, e_1, v_1, \ldots, e_r, v_r$，其中 $x = v_0$且 $y = v_r$。对于任意单调 (x, y)-游走 W, $e_1, \ldots, e_r \in E(W)$，以及$t(e_1) \leq \cdots \leq t(e_r)$。此外， T存在单调 (x, y)-游走，当且仅当 $t(e_1) \leq \cdots \leq t(e_r)$。假设 T 存在单调 (x, y)-游走。

对于 $i \in [r-1]$，如果 $t(e_i) < t(e_{i+1})$，我们称 v_i 是一个连接点 。由于 T 是一棵树，且每条边在$\mathcal{T}$中恰好出现一次，因此 W 是一个单调的（x， y）-游走，并且 $V(W)$ 是包含极大的，当且仅当 $V(W) = \bigcup_{t \in [L]} V(C_t)$ 且 $E(W) = \bigcup_{t \in [L]} E(C_t)$，其中每个 C_t 都是 F_t 的一个连通分量，使得

(i) 对于某个 $i \in [r]$， $t = t(e_i)$，且 C_t 是包含 e_i 的 F_t 的一个连通分量；(ii) 或者 $t < t(e_1)$，且 C_t 是包含 v_0 的 F_t 的连通分量；(iii) 或者对于某个 $i \in [r-1]$， $t(e_i) < t < t(e_{i+1})$，且 C_t 是包含 v_i 的 F_t 的连通分量；(iv) 或者 $t(e_r) < t$，且 C_t 是包含 v_r 的 F_t 的连通分量。

利用这一观察，我们可以求出一条(x, y)-游走的最大权重。首先，我们在 O(n) 时间内找到唯一的 (x, y)-路径 $P = v_0, e_1, v_1, \ldots, e_r, v_r$，它位于 T 中，并检查 $t(e_1) \leq \cdots \leq t(e_r)$ 是否成立。如果后一条件不成立，我们就报告不存在单调的 (x, y)-游走。否则，我们找到满足 (i)–(iv) 的 $t \in [L]$ 对应的 F_t 的连通分量 C_t。然后，构造 $U = \bigcup_{t \in [L]}^{L} V(C_t)$，并返回 $w(U)$。由于每个 F_t 的连通分量都能在 O(n) 时间内找到，因此总运行时间为 O(nL)。◀

为了在我们的核化算法中缩减顶点权重，我们采用 Etscheid 等人提出的方法 [21]，该方法基于 Frank 和 Tardos 的结果 [23]。

22 时序探索问题的核化

▶ **命题 22** ([23])。 存在一种算法，给定一个向量 $\mathsf{w} \in \mathbb{Q}^r$ 和一个整数 N，可在多项式时间内找到一个向量 $\mathsf{w} \in \mathbb{Z}^r$，使得 $k\mathsf{w}k_\infty \leq 2^{4r^3} N^{r(r+2)}$，并且对所有满足 $kbk_1 \leq N-1$ 的向量 $b \in \mathbb{Z}^r$，都有 $\overline{\mathsf{sign}}(\mathsf{w} \cdot b) = \mathsf{sign}(\mathsf{w} \cdot b)$。

事实上，Frank 和 Tardos 算法是强多项式的。

现在我们准备证明本节的主要结果。我们记得，$V(G)$ 的两个子集组成的有序对 (P, Q) 被称为 G 的一个分离，如果 $P \cup Q = V(G)$ 且不存在满足 $x \in P \setminus Q$ 且 $y \in Q \setminus P$ 的边 xy。集合 $P \cap Q$ 是一个分隔集，而 $|P \cap Q|$ 被称为分离 (P, Q) 的阶。如果一个分离的阶为一，则 $P \cap Q$ 中的唯一顶点被称为割点，我们称该顶点为一个分隔集。注意，可能出现 $P \subseteq Q$ 或 $Q \subseteq P$ 的情况；由于可能出现 $P \cap Q$ 不分离 G 的情况，我们对标准记号作了轻微的滥用。我们还记得，G 的一个边割是 $V(G)$ 的一个划分 (P, Q)，而边割集是所有满足 $x \in P$ 和 $y \in Q$ 的边 xy 所组成的集合。

▶ **定理 23**。 以 $p = \mathfrak{m} - n + 1$ 为参数的加权 k-*arb NS-TEXP* 在连通底层图上存在一个大小为 $O(p^4)$ 的核，使得对于输出实例$(G = (G_1, \ldots, G_L)\ w, v, k)$，$G$ 具有 $O(p)$ 个顶点和边，并且 $L \in O(p)$。

证明。设 $(\mathsf{G} = (G_1, \ldots, G_L)\ w, v, k)$ 是加权 k-arb NS-TEXP 的一个实例，并设 $p = \mathfrak{m} - n + 1$。再设 G 为 G 的底层图，并设 R 为红边集合。我们将核化算法描述为一系列归约规则：只要可以应用某条规则，就反复应用这些规则，直至无法继续应用。规则会修改 $\mathsf{G} = (G_1, \ldots, G_L)$ 和 w。每当我们说某条规则删除一个顶点 x 时，这意味着从 G_1、.....、G_L 和 G 中删除 x，以及与其关联的边。同样，每当我们创建一个顶点时，该顶点都会被添加到每个图 G_i（其中 $i \in [L]$ 和 G）。当我们删除或添加一条边时，我们会明确说明 G_i 执行该操作的位置，并假设对 G 执行相应的操作。

为描述第一条规则，我们需要一些辅助记号。设 (P, Q) 是G 的一个阶为一的分离，其割点为 x。若重要，则称 (P, Q) 为重要分离

(i) $G[P]$ 是一棵至少有两个顶点的树，**(ii)** $v \in Q$ 且 $R \subseteq E(G[Q])$，并且**(iii)** P 是满足 (i) 和 (ii) 的包含极大集合。

设 (P, Q) 是 G 的一个重要分离，其割点为 x。令 $a, b \in [L]$。那么，存在 $H = G[P]$ 的一个分离 (P_1, P_2)，其中 x 为割点，使得对于任意 $y \in N_H(x)$，$y \in P_1$ 当且仅当 $a \leq t(xy) \leq b$，此外，(P_1, P_2) 是唯一的。我们用 $\mathsf{T}_{[a,b]}$ 表示时序图 $(G_1[P_1], \ldots, G_L[P_1])$，并且 $T_{[a,b]} = G[P_1]$。注意，$\mathsf{T}_{[a,b]}$ 可能是一个只包含 x的单顶点时序图。我们用 $\mathsf{T}_{(a,b)}$（分别用 $\mathsf{T}_{[a,b)}$和 $\mathsf{T}_{(a,b]}$）表示 $\mathsf{T}_{[a+1,b-1]}$（分别表示 $\mathsf{T}_{[a,b-1]}$ 和 $\mathsf{T}_{[a+1,b]}$），并对底层树采用相应的记号。我们还用 T_a 和 T_a 分别代替 $\mathsf{T}_{[a,a]}$和 $T_{[a,a]}$。给定 G 的一个重要分离 (P, Q)，其割点为 x，我们将$A(Q) = \bigcup_{y \in N_{G[Q]}(x)} A(xy)$记作。若 $i < j$ 且不存在这样的$t \in A(\{v50)$，使得 $i < t < j$，则称 $i, j \in A(Q)$ 是连续的。

注意到，利用标准算法工具，可以在线性时间内找到图的割点和块（参见例如 [9]），因此所有重要分离都可以在线性时间内列出。

给定一个割点 x，使得重要分离 T 的一侧是一棵树，则最优探索可以通过 x 进入并从 T 离开，或在 T的某个顶点结束。在这两种情况下，我们都可以使用引理 21 中的算法计算 T 的最优探索，并用 2 个顶点替换 $T - x$，以模拟这两种选项，从而得到第一条归约规则。

▶ **归约规则 1**。 如果 G 存在一个重要分离(P, Q)，其具有一个割点 x，并且满足以下任一条件：存在 $i \in A(Q)$，使得 $T = T_i$ 至少有四个顶点；存在连续的 $i, j \in A(Q)$，使得 $T = T(i,j)$ 至少有四个顶点；对于 $i = \min A(Q)$，$T = T[1,i)$ 至少有四个顶点；或对于 $j = \max A(Q)$，$T = T(j,L]$ 至少有四个顶点，则对 T 和底层树 T 执行以下操作（见图 4）：

调用引理 21 中的算法，计算 w_1 在 T中单调的 (x, x) 游走的最大权重，并计算 w_2 在 T中单调的 (x, y) 游走的最大权重，其中最大值取遍所有 $y \in V(T)$。令 $t = min_{u \in N_T}(x)\ t(xu)$。删除 G 中 T 的顶点，但保留 x，新建一个顶点 y，使其与 x 在 G_t中相邻，并令 $w(y) = w_1 - w(x)$。如果 $w_2 > w_1$，则新建一个顶点 z，使其与 y 在 G_L 中相邻，并令 $w(z) = w_2 - w_1$。

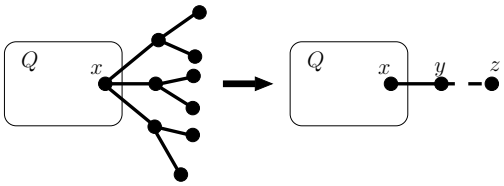

图 4 归约规则 1 的应用。

▷ 断言 24。归约规则 1 是安全的。

断言 24 的证明。 该规则针对 $\mathcal{T}$的四种可能选择进行应用。这些情形的安全性证明几乎完全相同。因此，我们仅给出 $\mathcal{T} = \mathcal{T}(i,j)$ 连续 $i, j \in A(Q)$ 这一情形的证明，并简要讨论其他情形证明中的差异。

记由规则得到的 Weighted k-arb NS-TEXP 实例为 $(\mathcal{G}', w', v, k)$，并令 G' 为$\mathcal{G}'$的底层图。注意，T 有一条边 xu，满足$t(xu) = t$。这意味着 $w_1 \geq w(x) + w(u)$，因此 $w'(y) = w_1 - w(x) > 0$。由于仅当 $w_2 > w_1$时才构造 z，因此有 $w'(z) > 0$。由此，所构造的实例是可行的。还要注意，如果构造了 z，那么在$\mathcal{G}'$中，$L \geq j > t$ 且 $t(xy) < t(yz)$。

假设 (G, w, v, k) 是 Weighted k-arb NS-TEXP 的是实例。根据观察 1，这意味着存在一条单调游走 (v, u) W，其中某个 $u \in V(G)$ 满足$w(W) \geq k$。如果 W 没有进入 $T - x$中的任何顶点，那么 W 就是G' 中的一条单调游走，并且 (G', w', v, k) 是一个是实例。假设情况并非如此，且 W访问了 $T - x$中的某些顶点。我们有，i 和 j 在 $A(Q)$ 中相邻。因此，$i < j$，并且对于每个 $h \in A(Q)$，要么 $h \leq i$，要么 $h \geq j$。由于 x 是一个割点，且对于$t = \min_{u \in N_T(x)} t(xu)$ 以及每个 $t' \in \{t(xu) : u \in N_T(x)\}$，$i < t \leq t' < j$，所以 $V(W) \cap V(T) \setminus \{x\}$ 中的顶点由 W的一个子游走 W'访问，且 W'是 T中的一条游走。此外，如果 $u \notin V(T) \setminus \{x\}$，则$W'$是一条 (x, x)-游走；如果 $u \in V(T) \setminus \{x\}$，则 W'是一条 (x, u)-游走。

假设 W' 是一个 (x, x)-游走。根据 w_1 的定义，$w(W) \leq w_1$，且 $T - x$ 中被 W' 访问的顶点的总权重至多为 $w'(y) \leq w_1 - w(x)$。我们通过将 W' 替换为 x, xy, y, yx, x，在 G' 中构造游走 W。于是，$w'(W) \geq w(W) \geq k$，也就是说，$(\mathcal{G}', w', v, k)$ 是一个是实例。

现在假设 W' 是一个针对 $u \in V(T) \setminus \{x\}$ 的 (x, u)-游走。如果 $w(W') \leq w_1$，那么我们将 W' 替换为 x、xy、y，并得到对于所获得的游走 W，$w'(W) \geq w(W)$。假设 $w(W') > w_1$。那么，$w_2 \geq w(W') > w_1$ 且 $z \in V(G')$。由于 $w'(z) = w_2 - w_1$，我们可以通过将 W' 替换为 x、xy、y、yz、z，从 W 构造出 W，并得到 $w'(W) \geq w(W) \geq k$。这就完成了 $(G'$、w'、v、$k)$ 是一个是实例的证明。

对于相反方向，论证非常相似。假设 (G′、w'、v、k) 是 Weighted k-arb NS-TEXP 的一个是实例，并将由规则⌢ ⌢从 T 创建的树记为 T'。那么，对于某个 $u \in V$（G'），存在一个单调的（v，u）-游走 W，使得 w（W）$\geq k$。
⌢ ⌢如果 W 不进入 y，那么 W 是 G 中的一个单调游走，并且 (G w、v、k) 是一个是实例。⌢假设 W 进入 y。由于 $i < t < j$ 且 x 是 G' 的割点，我们可以假设⌢ ⌢y 被 W 的一个子游走 W' 访问，而该子游走是 T' 中的一条游走。如果 $u \notin V$（T'）\\{x}，我们还可以假设 W' 是一个（x，x）-游走，并且 $W' = x$、xy、y、yx、x，因为如果 z 存在，则 t（yz）$> t$（xy）。⌢我们通过将 W' 替换为 T 中存在的、权重为 w_1 的单调（x，x）-游走 W'，在 G 中构造游走 W；该游走的存在性由 w_1 的选择保证。注意，对于每条边 $e \in E$（W'），$i < t$（e）$< j$。⌢因此，替换后得到的是 G 中的一条单调游走。我们得到 w（W）$= w'$（W）$\geq k$。⌢如果 $u \in V$（T'）\\{x}，那么我们假设：如果 z 不存在，则为 $u = y$ 和 $W' = x$、xy、y；如果 $z \in V$（T'），则为 $u = z$ 和 $W' = x$、xy、y、yz、z。在两种情况下，w'（W'）$= w_2$。我们通过将 W' 替换为 T 中、针对某个 $u \in V$（T）且由 w_2 的定义保证存在的、权重为 w_2 的单调（x，u）-游走 W'，构造⌢W。于是，W 是 G 中的一条单调游走，并且⌢w（W）$= w'$（W）$\geq k$。我们得出 (G w、v、k) 是一个是实例。这就完成了 T $=$ T$^{(i,j)}$ 对连续的 i、$j \in A$（Q）这一情形的证明。

若 T $=$ T$_i$ 对于 $i \in A$(Q)，则为修改证明，我们注意到，若规则构造出 z，则 $i < L$。因此，若规则构造出 z，则在 G′ 中，t(xy) $< t$(yz)。还要注意，对于任意 $u \in N_T$(x)，t (xu) $= i$。这保证我们可以对上文的游走应用交换论证。特别地，不失一般性，我们可以假设，若一条单调游走 W进入 $T - x$，则 V(W) $\cap$ V(T) \ {x} 中的顶点会由W的一条子游走访问。对于 T $=$ T$^{[1,i)}$，其中 $i =$ min A(Q)，我们只需注意到，G 中的任何单调游走只有在 $x = v$ 时才能进入 $T - x$，也就是说，x 是源顶点。最后，对于 T $=$ T$^{(j,L]}$，其中 $j =$ max A(Q)，我们注意到，若 $t = L$，则 T中的任何单调 (x,u)-游走只能使用 G_L的边。因此，在这种情况下，该规则不会生成 z。否则，若生成了 z，则t(xy) $< t$(yz)，我们可以像处理 T $=$ T$^{(i,j)}$ 那样应用相同的论证。证明至此完成。◁

为了陈述接下来的规则，我们定义对算法重要的边割。若图 G 的边割 (P，Q) 满足以下条件，我们称之为重要边割：
(i) 边割集的大小为二，且由端点互不相同的边组成，**(ii)** $G[P]$ 是一棵树，且**(iii)** $v \in Q$ 和 $R \subseteq E(G[Q])$。

给定一个重要边割 (P,Q)，其边割集为 {x_1y_1,x_2y_2}，其中 $x_1, x_2 \in P$，树 $G[P]$ 中存在唯一的 (x_1,x_2)-路径 S。我们称路径 $y_1, y_1x_1, S, x_2y_2, y_2$ 为该边割的骨架。我们还记作 T$_{P,Q}$ $=$ ($G_1[(P \cup \{y_1, y_2\})] - y_1y_2, \ldots, G_L[P \cup \{y_1, y_2\}] - y_1y_2$)。注意，T$_{P,Q}$ 的底层图 T 是一棵树，且 T 中唯一的 (y_1,y_2)-路径就是骨架。

注意，所有重要边割都可以在多项式时间内找到。例如，我们可以考虑所有端点不相同的边对，并检查这对边是否构成某个重要边割的边割集。现在，我们构造三条用于缩短骨干长度的归约规则。下一条规则将删除无关边。

▶ **归约规则 2. </style}若存在一个重要边割 (P, Q)**，其对应的 G 具有骨架 $v_0, e_1, v_1, e_2, v_2, e_3, v_3$，且满足 $t(e_1) > t(e_2)$ 以及 $t(e_3) > t(e_2)$，则通过从 $G_{t(e_2}$ 中删除 e_2 来修改 G 和 G（见图 5）。此外，如果得到的底层图 G 不连通，则删除不包含源顶点 v 的（唯一）连通分量中的顶点。

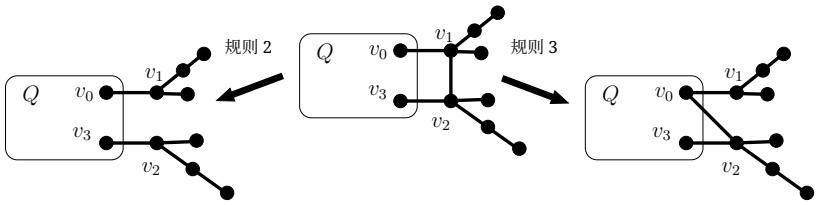

图 5 归约规则 2 和归约规则 3 的应用。

该规则的安全性可直接由以下观察得出：没有单调的 $(v,\ u)$-游走能够使用 e_2，因为任何单调游走只能通过 e_1 或 e_3 进入 P。

下一条规则针对同一 G_t中骨干里的连续边。

▶ **归约规则 3**。如果存在一个重要边割(P, Q)，其对应于 G，并且具有骨架$v_0, e_1, v_1, e_2, v_2, e_3, v_3$，使得 $t(e_1) = t(e_2)$，那么通过从 $G_{t(e_1)}$中删除 e_1 并在 $G_{t(e_1)}$ 中添加 v_0v_2 来修改 G 和 G（见图 *5*）。

▷ 断言 25。归约规则 3 是安全的。

断言 25 的证明。记 (G', w', v, k) 为应用该规则后得到的加权 k-arb NS-TEXP 实例，并令 G' 为 G' 的底层图。假设 $(G\ w, v, k)$ 是加权 k-arb NS-TEXP 的一个是实例。根据观察 1，这意味着存在一条单调的 (v, u)-游走 W，其中 $u \in V(G)$，且 $w(W) \geq k$。我们通过将游走 W' 中每次出现的边 e_1 分别替换为 v_0v_2, v_2, e_2 或 e_2, v_2, v_2v_0，来构造该游走，具体取决于游走穿过 e_1 的方向。这样便得到 G' 中的一条游走；又因为 $t(e_1) = t(e_2)$，所以这是一条单调的 (v, u)-游走。根据构造，$w'(W') \geq w(W) \geq k$，因此 (G', w', v, k) 是一个是实例。反方向的论证几乎相同。区别在于，我们将 G' 中一条游走里每次出现的 v_0v_2 替换为 e_1, v_1, e_2，或者对称地替换为 e_2, v_1, e_1。证明至此完成。◁

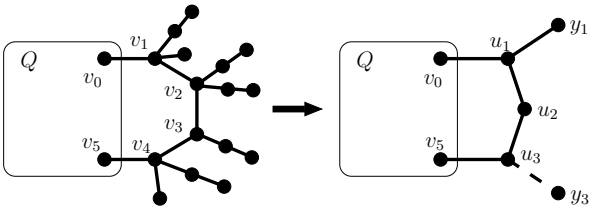

图 6 归约规则 4 的应用。

接下来需要缩短单调骨干。

▶ **归约规则 4**。如果存在一个重要边割 (P, Q)，其骨架为 $G\ v_0, e_1, \ldots, e_5, v_5$，且满足 $t(e_1) < \cdots < t(e_5)$，那么对 $T = T_{P,Q}$ 及其底层树 T（见图 *6*）执行以下操作，以修改 G 和 G：

调用引理 *21* 中的算法，并计算 T 中单调（v_0, v_0）-游走的最大权重 w_1、单调（v_0, v_5）-游走的最大权重 w_2，以及单调 （v_5, v_5）-游走的最大权重 w_3。删除 $V(T) \setminus \{v_0, v_5\}$ 中的顶点，创建新顶点 u_1, u_2, u_3，使 u_1 在 $G_{t(e_1)}$ 中与 v_0 相邻，使 u_2 分别在 $G_{t(e_2)}$ 和 $G_{t(e_4)}$ 中与 u_1 和 u_3 相邻，并使 u_3 在 $G_{t(e_5)}$ 中与 v_5 相邻。

- 令 $w(u_1) = w_1 - w(v_0)$、$w(u_3) = w_3 - w(v_5)$ 和 $w(u_2) = w_2 - w_1 - w_3$。调用引理 *21* 中的算法，并计算 $T - v_5$ 中单调 (v_0, u)-游走的最大权重 w_0，其中最大值取遍所有 $u \in V(T) \setminus \{v_5\}$。创建一个顶点 y_1，使 y_1 在 $G_{t(e_3)}$ 中与 u_1 相邻，并令 $w(y_1) = w_0 - w(v_0) - w(u_1) - w(u_2)$。调用引理 *21* 中的算法，并计算 T 中单调 (v_5, u)-游走的最大权重 w_5，其中最大值取遍所有 $u \in V(T)$。若 $w_5 > w_3$，则创建一个顶点 y_3，使 y_3 在 G_L 中与 u_3 相邻，并令 $w(y_3) = w_5 - w(v_5) - w(u_3)$。

▷ 断言 26。归约规则 4 是安全的。

断言 26 的证明。 记 $(\mathcal{G}', w', v, k)$ 为应用该规则后得到的 加权 k-arb NS-TEXP 实例，并令 G' 为$\mathcal{G}'$的底层图。

首先，我们论证实例 (G', w', v, k) 是可行的，即权重均为正。注意，根据该构造，$w_1 \geq w(v_0) + w(v_1)$。因此，$w'(u_1) > 0$。类似地，$w'(u_3) > 0$。注意，给定 T 中的一个单调 (v_0, v_0)-游走 W_1、一个单调 (v_0, v_5)-游走 W_2 以及一个单调 (v_5, v_5)-游走 W_3，我们可以将它们拼接起来，得到 (v_0, v_5)-游走 W。游走 W 是单调的，$V(W_1) \subseteq V(W)$，且 $V(W_3) \subseteq V(W)$。由于 $t(e_1) < t(e_2) < t(e_4) < t(e_5)$，因此有 $v_2, v_3 \in V(W_2) \setminus (V(W_1) \cup V(W_3))$。这意味着 $w_2 > w_1 + w_3$ 且 $w(u_2)$ 为正。注意，$w_0 \geq w_2 - w(v_5) \geq w(v_0) + w(u_1) + w(u_2) + w(u_3)$。因此，$w(y_1) = w_0 - w(v_0) - w(u_1) - w(u_2) \geq w(u_3) > 0$。最后，我们仅在 $w_5 > w_3$ 时创建 y_3。于是，$w(y_3) = w_5 - w(v_5) - w(u_3) = w_5 - w(v_5) - (w_3 - w(v_5)) = w_5 - w_3 > 0$。此外注意，如果 $t(e_5) = L$，那么 $w_5 = w_3$，因为任意单调 (v_5, u)-游走只能使用 G_L 的边。这意味着，如果规则构造了 y_3，则 $t(e_5) < L$。

假设（$\mathcal{G}\ w, v, k$）是加权 k-arb NS-TEXP 的是实例。根据观察 1，这意味着存在一条单调的（v, u）-游走 W，对于某个 $u \in V$（G），使得 w（W）$\geq k$。如果 W 未使用 T 的任何边，那么 W 是 G' 中的一条单调游走，且（$\mathcal{G}', w', v, k$）是一个是实例。假设情况并非如此，且 W 经过了 $T - \{v_0, v_5\}$ 中的一些顶点。根据 W 的行为，我们分三种情形讨论。

情形 1. 骨架中有一条边未被 W 使用。

如果 $e_5 \notin E$（W），那么，由于 $v \in Q$ 和 W 只能通过 e_1 进入 $T - \{v_0, v_5\}$，不失一般性，可假设 V（W）$\cap V$（T）$\setminus \{v_0, v_5\}$ 中的顶点由 W 的一个子游走 W' 经过，且该子游走是 T 中的一条单调的（v_0, z）-游走，对于某个 $z \in V$（T）。$\frown$如果 $z = v_0$，那么我们通过将 W' 替换为 $v_0, v_0u_1, u_1, v_0u_1, v_0$，在 G' 中构造游走 W。接着，由于根据 w_1 的定义，w（W'）$\leq w_1$，且 w'（u_1）$= w_1 - w'$（v_0），因此有 $\frown$ $\frown w'$（W）$\geq w$（W）$\geq k$。如果 $z \neq v_0$，那么 $z = u$。我们通过将 W' 替换为 $v_0, v_0u_1, u_1, u_1u_2, u_2, u_1u_2, u_1, u_1y_1, y_1$，构造 W。由于 W' 未使用 e_5，根据 w_0 的定义，w（W'）$\leq w_0$。由于 w（y_1）$= w_0 - w$（v_0）$- w$（u_1）$- w$（u_2），可得 w'（W）$\geq w$（W）$\geq k$。在这两种情形下，我们都可以得出（G', w', v, k）是一个是实例。

假设 $e_1 \notin E(W)$。现在我们可以假设，$V(W) \cap V(T) \setminus \{v_0, v_5\}$ 中的顶点由 W' 遍历；W' 是 W 的一个子游走，并且是 T 中某个 $z \in V(T)$ 的单调的 (v_5, z)-游走。$\frown$ 如果 $z = v_5$，那么我们通过将 W' 替换为 $v_5, v_5u_3, u_3, v_5u_3, v_5$，构造 G' 中的游走 W。$\frown$ 接着，因为 $w(W') \leq w_3$ 且 $w'(u_3) = w_3 - w'(v_5)$，我们有 $w'(W) \geq w(W) \geq k$。$\frown$ 如果 $z \neq v_5$，那么 $z = u$。若 $w_5 = w_3$，我们通过将 W' 替换为 v_5, v_5u_3, u_3 来构造 W；若 $w_5 > w_3$，则替换为 $v_5, v_5u_3, u_3, u_3y_y, y_3$。由于 e_1 未被 W' 使用，$w(W') \leq w_5$，并且我们得到 $\frown$，使得 $w'(W) \geq w(W) \geq k$。在这两种情况下，我们都有 (G', w', v, k) 是一个是实例。

现在假设 $e_1, e_5 \in E(W)$。那么，我们可以假设 $V(W)$ 的顶点 $\cap V(T) \setminus \{v_0, v_5\}$ 恰好由 W 中两个互不相交的子游走 W_1 和 W_2 访问，其中 W_1

是 W 的一个 (v_0, v_0)-子游走，而 W_2 是 W 的一个 (v_5, z)-子游走，其中 $z \in V(T)$。接着，我们以与处理 $e_5 \notin E(W)$ 和 $z = v_0$ 完全相同的方式替换 W_1。同样地，以与处理 $e_1 \notin E(W)$ 完全相同的方式替换 W_2。用于替换的子游走互不相交，因此我们得出结论：$(\mathcal{G}', w', v, k)$ 是一个是实例。

情形 2。骨架的边位于 W 和 $u \notin V(T) \setminus \{v_0, v_5\}$ 中。由于 $t(e_1) < \cdots < t(e_5)$，我们可以假设 $V(W) \cap V(T) \setminus \{v_0, v_5\}$ 的顶点由 W 的一个子游走 $\frown W'$ 访问，该子游走是 T 中一条单调的 (v_0, v_5)-游走。然后，我们通过将 W' 替换为 $v_0, v_0u_1, u_1, u_1u_2, u_2, u_2u_3, u_3, u_3v_5, v_5$，在 G' 中构造游走 W。接着，由于 $w(W') \leq w_2$，并根据 u_1, u_2, u_3 的权重定义，我们有 $w'(W) \geq w(W) \geq k$，这意味着 (G', w', v, k) 是一个是实例。

情形 3。骨架的边位于 W 和 $u \in V(T) \setminus \{v_0, v_5\}$ 中。由于骨架的所有边都位于 W 和 $t(e_1) < \cdots < t(e_5)$ 中，我们可以假设，$V(W) \cap V(T) \setminus \{v_0, v_5\}$ 的顶点由 W 的两个子游走 W_1 和 W_2 访问，其中 W_1 是 T 中的单调 (v_0, v_5)-游走，而 W_2 是单调的 (v_5, u)-游走。令 W_3 为 T 中权重最大的单调 (v_5, v_5)-游走。我们不失一般性地假设 W_3 同时是 W_1 和 W_2 的子游走；否则，我们可以分别在这些游走的末尾和开头加入 W_3。注意，对每个 $e \in E(W_3)$，都有 $t(e) = t(e_5)$。于是，由于 $t(e_1) < \cdots < t(e_5)$，且 W_1, W_2 是单调的，我们有 $E(W_1) \cap E(W_2) = E(W_3)$。我们 $\frown$ 通过将 W_1 替换为 $v_0, v_0u_1, u_1, u_1u_2, u_2, u_2u_3, u_3, u_3v_5, v_5$，在 G' 中构造游走 W。如果 $w_5 = w_3$，我们将 W_2 替换为 v_5, u_3v_5, u_3。否则，如果 $w_5 > w_3$，$y_3 \in V(G')$，则将 W_2 替换为 $v_5, u_3v_5, u_3, u_3y_3, y_3$。由于 $E(W_1) \cap E(W_2) = E(W_3)$，$w(V(W_1) \cup V(W_2)) = w(W_1) + w(W_2) - w_3$。根据 u_1, u_2, u_3 和 y_3 的权重定义，我们得到 $\frown w'(W) \geq w(W) \geq k$。因此，$(G', w', v, k)$ 是一个是实例。这就完成了正向情形的证明。

对于相反方向，证明采用类似的论证，只是交换论证的方向相反。按照该规则，记由 T 构造的树为 T'。设 W_1 是 T 中一条权重为 w_1 的单调（v_0，v_0）-游走，W_2 是一条权重为 w_2 的单调（v_0，v_5）-游走，而 W_3 是一条权重为 w_3 的单调（v_5，v_5）-游走。再设 W_1' 是 $T - v_5$ 中一条权重为 w_0 的单调（v_0，z）-游走，并且对某个 $z \in V$（T），W_3' 是 T 中一条权重为 w_5 的单调（v_5，z）-游走。注意，由于 t（v_0u_1）$< t$（u_1u_2）$< t$（u_2u_3）$< t$（u_3v_5），所以 V（W_1）$\cap V$（W_3）$= \emptyset$，并且 V（W_1）$\cap V$（W_3'）$= \emptyset$。还要注意，V（W_1）V（W_3）$\subseteq V$（W_2），否则我们可以将 W_1 和 W_3 与 W_2 拼接起来，从而增加 W_2 的权重。

假设 (G', w', v, k) 是 Weighted k-arb NS-TEXP 的是实例。那么，$\frown$ $\frown$ $\frown$存在一条单调的 (v, u)-游走 W，位于 G' 中，且满足 $w'(W) \geq k$。如果 W没有进入 $T' - \{v_0, v_5\}$，那么 $\frown W$就是 G' 中的一条单调游走，并且 $(G\ w, v, k)$ 是一个是实例。假设 W经过了 $T' - \{v_0, v_5\}$ 中的某些顶点。我们考虑三个情形，它们分别类似于上文的情形 1–3。

情形 1. T'的骨架中存在一条未被 W使用的边。

如果 $u_3v_5 \notin E(\frown W)$，那么不失一般性，我们可以假设 $\frown$ $\frown$ $\frown V(W) \cap V(T') \setminus \{v_0, v_5\}$ 中的顶点由 W 的子游走 W'经过；对于某个 $z \in V(T')$，该子游走是 T' 中的一条单调的 (v_0, z)-游走。如果 $z = v_0$，那么我们通过将 $\frown$ $\frown W' = v_0, v_0u_1, u_1, v_0u_1, v_0$ 替换为 W_1，在 G 中构造游走 W。如果 $z \neq v_0$，那么 $z = u$，并且我们将 W' 替换为 W_1'。在这两种情形下，我们都得到 $w(W) \geq w'(W) \geq k$。因此，$(G\ w, v, k)$ 是一个是实例。

若 $v_0u_1 \notin E(\frown W)$，那么我们可以假设 $\frown V(W) \cap V(T') \setminus \{v_0, v_5\}$ 被某个子游走 W'所访问，该子游走是 T中的单调的（v_5，z）-游走，其中某个 $z \in V$（T'）。如果 $\frown z = v_5$，那么通过将 $W' = v_5$，v_5u_3，u_3，v_5u_3，v_5 替换为 W_3，即可构造出 G 中的游走 W。

如果 $z \neq v_5$，那么 $z = u$，并且我们将 W' 替换为 W'_3。我们有 $w(W) \geq w'(W) \geq k$，且（$\mathcal{G}$ w， v， k）是一个是实例。

假设 v_0u_1， $u_3v_5 \in E($ $W) \cap V(T') \setminus \{v_0, v_5\}$ 中的顶点恰好由 W 的两个互不相交的子游走 W_1 和 W_2 访问，其中 W_1 是 W 的一个 (v_0, v_0)- 子游走，而 W_2 是某个 $z \in V(T')$ 的 (v_5, z)- 子游走。然后，我们以与处理 $u_3v_5 \notin E(W)$ 和 $z = v_0$ 完全相同的方式，用 W_1 替换 W_1。类似地，以与处理 $v_0u_1 \notin E(W)$ 完全相同的方式，用 W_3 或 W'_3 替换 W_2。用于替换的子游走互不相交，因此我们得出结论：(G w, v, k) 是一个是实例。

情况 2。 T'的骨架的边位于 W和 $u \notin V$（T'）\\ {v_0, v_5} 中。在这种情况下，我们可以假设， V（W）$\cap V$（T'）\\ {v_0, v_5} 的顶点由子游走 $W' = v_0, v_0u_1, u_1, u_1u_2, u_2, u_2u_3, u_3, u_3v_5, v_5$ 访问，并通过将 W' 替换为 W_2 来构造 W。然后， w（W）$\geq w'$（W）$\geq k$，且（G w, v, k）是一个是实例。

情况 3。 T'的骨架的边位于 W和 $u \in V$（T'）\\ {v_0, v_5} 中。现在我们可以假设， V（W）$\cap V$（T'）\\ {v_0, v_5} 的顶点由 W 的两个子游走 $W_1 = v_0, v_0u_1, u_1, u_1u_2, u_2, u_2u_3, u_3, u_3v_5, v_5$ 和 W_2 访问，并且以下两种情况之一成立： $y_3 \notin V$（G'）和 $W_2 = v_5, u_3v_5, u_3$，或 $y_3 \in V$（G'）和 $W_2 = v_5, u_3v_5, u_3, u_3y_3, y_3$。我们将 W_1 替换为 W_2，并将 W_2 替换为 W'_3。同样， w（W）$\geq w'$（W）$\geq k$，且（G w, v, k）是一个是实例。至此，反向的证明完成。◁

我们注意到，根据规则 1–4 的定义，可以立即得到以下命题 .

▷ 命题 27. 规则 1–4 不会增加参数 $p = \mathfrak{m} - n + 1$。

我们证明，对规则 1–4 进行穷尽应用后得到的 Weighted k-arb NS-TEXP 实例中，图的规模有界。

▷ 命题 28. 若规则 1–4 不适用于 ($\mathcal{G}$ w, v, k)，则 $|V(\mathcal{G})| \leq 324p$ 且$|E(\mathcal{G})| \leq 326p$。

命题 28 的证明。假设规则 1–4 均不适用于 ($\mathcal{G}$ w, v, k)。我们使用引理20。根据该引理，存在一个顶点集合 $X \subseteq V(G)$，其大小至多为 $4p$，且满足：(i) $v \in X$；(ii) 每条红边都属于 $G[X]$；(iii) $G - E(G[X])$ 是一个森林；(iv) F的每个连通分量 $G - X$ 都是一棵树，并且具有以下性质：每个顶点 $x \in X$ 在 F 中至多有一个邻居，并且

(a) 或者 X 有唯一一个顶点 x，该顶点在 F 中有一个邻居，
(b) 或者 X 恰好包含两个顶点 x，并且 y 在 F 中有邻居。

我们假设 X 是具有这些性质的包含意义下极小集。我们称满足 (a) 或 (b) 的 $G - X$的一个连通分量 F分别为*(a)* 型 或*(b)* 型 分量。根据该引理，如果 q 是 (b) 型分量的数量，那么 $q \leq 4p - 1$ 和 $G[X]$至多有 $5p - q - 1$ 条边。

设 $x \in X$，并考虑具有某个 $r \geq 1$ 作为邻居的 (a) 型分量 $F_1, \ldots, F_r$。令 $P_x = \{x\} \cup \bigcup_{i \in [r]} V(F_i)$，且 $Q_x = (V(G) \setminus P) \cup \{x\}$。由于 X 是满足 (i)–(iv) 的包含意义下极小集，因此 (P_x, Q_x) 是一个重要分离，且 x 是其割顶点。由于无法应用归约规则 1，我们有 $|P_x \setminus \{x\}| \leq 4|A(Q_x)| + 2$，且 $|E(G[P_x])| \leq 4|A(Q_x)| + 2$。在 Q_x 中， x 或与 X 中的一个顶点相邻，或与一个 (b) 型分量相邻。考虑到连接 X 与 (b) 型分量的边恰好出现在一个时间步中，我们有

$$\sum_{x \in X} |A(Q_x)| \leq 2 \sum_{e \in E(G[X])} m(e) + 2q \leq 2p + 2|E(G[X])| + 2q \leq 12p - 2.$$

因此， $G - X$ 的 (a) 型分量 F 中的顶点总数至多为 $56p - 8$，且这些顶点关联的边至多有 $56p - 8$ 条。

考虑 $G - X$ 的一个 (b) 型分量 F。设 x 和 y 是 X 中 F 的邻居，并记 x' 和 y' 为 F 中 x 和 y 的邻居。设 S 为树 F 中的 (x', y')-路径。我们声称 S 至多有 6 个顶点。

证明采用反证法。假设 S 至少有 7 个顶点。注意，(P, Q) 是一个重要边割，其中 $P = V(F)$，且 $Q = V(G) \setminus P$。设 v_0、e_1、……、e_ℓ、v_ℓ 为其骨架。注意 $\ell \geq 8$。由于归约规则 3 不适用，对于每个 $i \in [\ell - 1]$，都有 $t(e_i) \neq t(e_{i+1})$。由于归约规则 2 不适用，对于每个 $i \in [\ell - 2]$，要么 $t(e_i) < t(e_{i+1}) < t(e_{i+2})$，要么 $t(e_i) > t(e_{i+1}) > t(e_{i+2})$，要么 $t(e_i) < t(e_{i+1})$ 且 $t(e_{i+1}) > t(e_{i+2})$。注意，最后一种情况 $t(e_i) < t(e_{i+1})$ 且 $t(e_{i+1}) > t(e_{i+2})$ 只可能对应唯一的 $i \in [\ell - 2]$。由于 $\ell \geq 8$，要么 $t(e_1) < \cdots < t(e_5)$，要么 $t(e_{\ell-4}) > \cdots > t(e_\ell)$。不失一般性，利用对称性，可设 $t(e_1) < \cdots < t(e_5)$。注意，$\{e_1, e_5\}$ 是重要边割 (P', Q') 的割集，其中 P' 是包含 $v_1 = x'$ 的 $F - e_5$ 连通分量的顶点集合，且 $Q' = V(G) \setminus P'$。于是， v_0、e_1、……、e_5、v_5 是相应的骨架。然而，我们可以对 (P', Q') 应用归约规则 4。这与我们关于归约规则均不适用的假设矛盾，并证明 S 至多有 6 个顶点。

令 $z \in V(S)$，并考虑 $F - E(S)$ 的一个包含 z 的连通分量 H。如果 H 至少有一条边，那么 (P, Q)，其中 $P = V(H)$ 且 $Q = (V(G) \setminus V(H)) \cup \{z\}$ 是一个重要分离。回想一下， S 的每条边以及边 xx' 和 yy' 在时序图中都只出现一次。因此， $|A(Q)| \leq 2$。由于无法应用归约规则 1，我们得到 $|V(H)| \leq 4|A(Q)| + 3 \leq 11$ 且 $|E(H)| \leq 10$。由于 S 至多有 6 个顶点，我们得出 F 至多有 66 个顶点，且 F 的顶点与 G 至多 67 条边相邻。

由于 (b) 型分量的总数为 $q \leq 4p - 1$，这些分量中的顶点总数至多为 $264p - 66$，且这些分量的顶点与 G 的至多 $67q$ 条边关联。

综上并利用 $q \leq 4p - 1$, $|X| \leq 4p$ 和 $|E(G)| \leq 5p - q - 1$，我们得到

$$|V(\mathcal{G}| = |V(G)| \leq |X| + (56p - 8) + (264p - 66) \leq 324p$$

and

$$|E(\mathcal{G})| \leq |E(G)| + p \leq |E(G[X])| + (56p - 8) + 67q + p \leq 326p.$$

证明至此完成。 ◁

G 的边数上界使我们能够减小 L的值。

▶ **归约规则 5**。 若存在 $t \in [L]$ 使得 G_t 没有边，则从 G 中删除 G_t。

该规则显然是安全的，因为无边图与 $\mathcal{G}$ 的探索无关。应用该规则后， $L \leq 326p$ 如同 $|E(\mathcal{G})| \leq 326p$。

我们的最后一个目标是缩减权重。为此，我们使用命题 22 中的算法。设 $n = |V(G)|$，并设 $V(G) = \{v_1, \ldots, v_n\}$。定义 $r = n + 1$，并考虑向量 $\mathsf{w} = (w_0, w_1, \ldots, w_n) \in \mathbb{Z}^r$，其中 $w_0 = k$ 且 $w_i = w(v_i)^{\mathsf{T}}$对于 $i \in [n]$。

▶ **归约规则 6**。 对w和 $N = r + 1$ 应用命题 *22* 中的算法，并求出向量$\mathsf{w} = (w_0, \ldots, w_n)$。令 $k := \overline{w_0}$，并令 $w(\overline{v_i}) := w_i$，其中 $i \in [n]$。

为说明该归约规则是安全的，令 $k' = \mathfrak{w}_0$，并令 $w'(v_i) = w_{\overline{i}}$，其中 $i \in [n]$。注意，根据 N 的选择，对于每个向量 $b \in \{-1,\ 0,\ 1\}^r$，我们都有 $\mathsf{sign}(\mathsf{w} \cdot b) = \mathsf{sign}(\mathsf{w} \cdot b)$。这意味着新的权重 $w'(x)$ 和 k' 都是正整数，并且对于每个 $U \subseteq V(G)$，$w(U) \geq k$ 当且仅当 $w'(U) \geq k'$。

根据命题 22，我们得到 $k \leq 2^{4n^3}(n+2)^{(n+1)(n+2)}$，并且对于每个 $x \in V(G)$，$w(x)$也有相同的上界。由于 $|V(G)| = \mathcal{O}(p)$，我们需要 $\mathcal{O}(p^3)$ 位来编码 k 和每个顶点的权重。因此，权重和 k 的编码总位长为 $\mathcal{O}(p^4)$。考虑到 $|V(G)| = \mathcal{O}(p)$、$|E(G)| = \mathcal{O}(p)$ 且$L = \mathcal{O}(p)$，我们得出结论：我们得到了一个规模为 $\mathcal{O}(p^4)$ 的核。

由于可以在多项式时间内找到重要分离和重要边割，且引理 21 中的算法是多项式时间的，因此可以在多项式时间内穷尽应用规则 1–4。显然，归约规则 5 可以在多项式时间内应用。由于命题 22 中的算法是多项式时间的，归约规则 6 需要多项式时间。因此，我们的核化算法的整体运行时间是多项式的。证明至此结束。◀

在 [20], Erlebach 和 Spooner证明了 NS-TEXP 可以在$\mathcal{O}(2^n Ln^3)$ 时间内求解。这一事实结合定理 23，得出以下推论。

▶ **推论 29.** 对于具有连通底层图的时序图，参数为 $p = \mathfrak{m} - n + 1$ 的加权 k-*arb NS-TEXP* 可在 $2^{\mathcal{O}(p)}(nL)^{\mathcal{O}(1)}$ 时间内求解。

证明。给定一个加权 k-arb NS-TEXP 实例 $(\mathsf{G} = (G_1, \ldots, G_L)\ w, v, k)$，我们首先应用定理 23 中的核化算法。然后，我们猜测由探索游走访问的顶点集合 U。由于顶点数为 $\mathsf{O}(p)$，因此有 $2^{\mathcal{O}(p)}$ 种 U的选择。对于每个 U，我们验证 $w(U) \geq k$ 是否成立，以及 $(\mathsf{G}[U] = (G_1[U], \ldots, G_L[U])\ v)$ 是否为 NS-TEXP 的是实例，并使用 Erlebach 和 Spooner 的算法 [20] 进行判断。该算法的总运行时间为 $2^{\mathcal{O}(p)}(nL)^{\mathcal{O}(1)}$。◀

我们指出，使用 Weighted k-arb NS-TEXP 核化算法中的归约规则的简化版本，可以为以 $p = \mathfrak{m} - n + 1$ 为参数的 NS-TEXP 获得一个线性规模的核。

我们在本节最后指出，以边的不出现次数为参数的 Weighted k-arb NS-TEXP 存在一个简单的核。

▶ **观察30**。对于连通底层图，以 $q = |E(G)|L - \mathfrak{m}$ 为参数的 *Weighted k-arb NS-TEXP* 存在一个大小为 $\mathcal{O}(q^4)$ 的核；对于输出实例$(\mathcal{G} = (G_1, \ldots, G_L)\ w, v, k)$，$\mathcal{G}$ 至多有 $q+1$ 个顶点、至多有 $(q-1)L$ 条边，且 $L \leq q$。

证明。设 $(\mathcal{G} = (G_1, \ldots, G_L)\ w, v, k)$ 是 $\textsc{Weighted}\ k\text{-}\textsc{arb}\ \textsc{NS-TEXP}$ 的一个实例，其中 G 是 $\mathcal{G}$ 的底层图。再设 $q = |E(G)|L - \mathfrak{m} = \sum_{t=1}^{L} |E(G) \setminus E(G_t)|$。该核化算法归结为以下几个直接的归约规则。

▶**归约规则 7**。如果存在 $xy \in E(G)$，使得对于每个 $t \in [L]$，都有 $xy \in E(G_t)$，那么就在每个 G_t 中收缩边 xy，并将所得顶点 z_{xy} 的权重设为$w(z_{xy}) := w(x) + w(y)$。此外，如果 $v \in \{x, y\}$，则将源顶点 $v := z_{xy}$。

▶ **归约规则 8**。如果存在 $t \in [L]$ 使得 G_t 是连通的，那么如果 $w(V(\mathcal{G})) \geq k$，则返回一个常数大小的平凡是实例；否则，如果 $w(V(G)) < k$，则返回一个平凡否实例。

注意，穷尽应用归约规则 7 后， G 至多有 q 条边。由于G 是连通的， G 至多有 $q+1$ 个顶点。根据归约规则 8，每个 G_t 至多有 $q-1$ 条边，因为 G 至少有一条边未出现在 G_t中。因此，边的总数至多为 $(q-1)L$。归约规则 8 还保证 $L \leq q$，因为每个 G_t中都缺少 G 的至少一条边。为完成证明，注意到权重可以按照定理 23 的证明中使用命题 22 的完全相同方式进行缩减（参见归约规则 6）。证明至此结束。◀

结论

我们开启了对时序图上探索问题的多项式核研究。我们证明，对于 NS-TEXP、 k-arb NS-TEXP 和 Weighted k-arb-NS- TEXP，除非 NP $\subseteq$coNP/poly，否则对于参数顶点数 n、生命周期 L 和待访问顶点数 k，以及组合参数 $L+k$、 $L+\gamma$ 和 $k+\gamma$，都不存在多项式核，其中 γ是每个时间步的最大连通分量数。事实上，通过一种直接的归约，将每个快照重复足够多次，我们所有不涉及参数 L的困难性结果都适用于严格设置下的 TEXP 和 k-arb TEXP，在这些问题中，时序探索每个时间步至多遍历一条边。我们证明，即使将时序图限制为底层图是深度为二的树，时序探索问题仍然是 NP 困难的。从参数化复杂性的角度来看，这排除了底层图上常见的大多数结构参数。尽管如此，我们还是找到了时序图的一个新参数 $p(\mathcal{G}) = \sum_{i=1}^{L} (|E(G_i)|) - |V(G)| + 1$，它刻画了时序图与每条边恰好出现一次的树有多接近。我们的参数也可以视为时序图稀疏性的一种度量。对于这一参数 $p(\mathcal{G})$，我们为 Weighted k-arb NS-TEXP 得到了一个多项式核。通过简化归约规则，我们可以为 NS-TEXP 得到一个线性规模的核。虽然从 NS-TEXP 到 TEXP 的归约会使参数 $p(\mathcal{G})$ 急剧增大，但我们仍可调整归约规则，为所研究探索问题的严格变体获得多项式核。自然的下一步是评估我们的参数对于时序图上其他问题的适用性。

参考文献

1 Duncan Adamson、Vladimir V. Gusev、Dmitriy S. Malyshev 和 Viktor Zamaraev。某些时态图的更快探索。载于 James Aspnes 和 Othon Michail 编，第*1*届动态网络算法基础研讨会，*SAND 2022*，*2022*年*3*月*28*日至*30*日，线上会议，第 *LIPIcs* 卷第 221 卷，第 5:1–5:10 页。达格施图尔城堡—莱布尼茨信息学中心，2022年。 doi:10.4230/LIPIcs.SAND.2022.5。2 Hans L. Bodlaender、Rodney G. Downey、Michael R. Fellows 和 Danny Hermelin。关于没有多项式核的问题。《计算机与系统科学期刊》，75(8):423–434，2009年。3 Hans L. Bodlaender 和 Tom C. van der Zanden。探索路径宽度较小的始终连通时态图。《信息处理快报》，142:68–71，2019年。 doi:10.1016/j.ipl.2018.10.016。4 树深度调节器对稀疏图之外获得多项式核有多大帮助？Marin Bougeret 和 Ignasi Sau。《算法学》，81(10):4043–4068，2019年。 doi:10.1007/s00453-018-0468-8。5 Benjamin Merlin Bumpus 和 Kitty Meeks。时态图的边探索。《算法学》，第 1–29 页，2022年。

6 Sobin C. C.、Vaskar Raychoudhury、Gustavo Marfia 和 Ankita Singla。容延网络中的路由与数据传播综述。《网络与计算机应用期刊》，67:128–146，2016年。 doi:10.1016/j.jnca.2016.01.002。 7 Arnaud Casteigts、Paola Flocchini、Walter Quattrociocchi 和 Nicola Santoro。时变图与动态网络。《国际并行、涌现与分布式系统期刊》，27(5):387–408，2012年。 doi:10.1080/17445760.2012.668546。 8 Arnaud Casteigts、Anne-Sophie Himmel、Hendrik Molter 和 Philipp Zschoche。在等待时间约束下寻找时态路径。《算法学》，83(9):2754–2802，2021年。 doi:10.1007/s00453-021-00831-w。 9 Thomas H. Cormen、Charles E. Leiserson、Ronald L. Rivest 和 Clifford Stein。《算法导论》第二版。麻省理工学院出版社和麦格劳-希尔图书公司，2001年。 10 Marek Cygan、Fedor V. Fomin、Lukasz Kowalik、Daniel Lokshtanov、Dániel Marx、Marcin Pilipczuk、Michal Pilipczuk 和 Saket Saurabh。《参数化算法》。施普林格，2015年。doi:10.1007/978-3-319-21275-3。 11 Michael Dom、Daniel Lokshtanov 和 Saket Saurabh。通过颜色与标识符得到核化下界。《ACM算法汇刊》，11(2):13:1–13:20，2014年。 doi:10.1145/2650261。 12 Rodney G. Downey 和 Michael R. Fellows。固定参数可解性与完全性。第21届曼尼托巴数值数学与计算会议论文集。《数值数学大会论文集》，87:161–178，1992年。 13 Rodney G. Downey 和 Michael R. Fellows。参数化复杂性。施普林格出版社，纽约，1999年。 14 Thomas Erlebach、Michael Hoffmann 和 Frank Kammer。关于时态图探索。《计算机与系统科学期刊》，119:1–18，2021年。 doi:10.1016/j.jcss.2021.01.005。 15 Thomas Erlebach 、 Frank Kammer 、 Kelin Luo 、 Andrej Sajenko 和 Jakob T. Spooner。每个时间步移动两次会带来不同结果。载于 Christel Baier、Ioannis Chatzigiannakis、Paola Flocchini 和 Stefano Leonardi 编，《第46届自动机、语言与编程国际研讨会，ICALP 2019》，2019年7月9日至12日，希腊帕特雷，第 132 卷 LIPIcs，第 141:1–141:14 页。达格施图尔城堡—莱布尼茨信息学中心，2019年。 doi:10.4230/LIPIcs.ICALP.2019.141。 16 Thomas Erlebach 和 Jakob T. Spooner。度受限时态图的更快探索。载于 Igor Potapov、Paul G. Spirakis 和 James Worrell 编，《第43届数学计算机科学基础国际研讨会，MFCS 2018》，2018年8月27日至31日，英国利物浦，第 117 卷 LIPIcs，第 36:1–36:13 页。达格施图尔城堡—莱布尼茨信息学中心，2018年。 doi:10.4230/LIPIcs.MFCS.2018.36。 17 Thomas Erlebach 和 Jakob T. Spooner。非严格时序探索。载于 Andrea Werneck Richa 和 Christian Scheideler 编，《结构信息与通信复杂性——第27届国际研讨会，SIROCCO 2020》，德国帕德博恩，2020年6月29日至7月1日，论文集，第 12156 卷《计算机科学讲义》，第 129–145 页。施普林格，2020年。 doi:10.1007/978-3-030-54921-3_8。 18 Thomas Erlebach 和 Jakob T. Spooner。缺少 k 条边的时序图探索。《信息学学报》，59(4):387–407，2022年。 doi:10.1007/s00236-022-00421-5。 19 Thomas Erlebach 和 Jakob T. Spooner。参数化时序探索问题。载于《第 1 届动态网络算法基础研讨会》（SAND），第 221 卷 LIPIcs，第 15:1–15:17 页。达格施图尔城堡—莱布尼茨信息学中心，2022年。 doi:10.4230/LIPIcs.SAND.2022.15。 20 Thomas Erlebach 和 Jakob T. Spooner。参数化时序探索问题。CoRR，abs/2212.01594，2022年。 arXiv:2212.01594, doi:10.48550/arXiv.2212.01594。 21 Michael Etscheid、Stefan Kratsch、Matthias Mnich 和 Heiko Röglin。加权问题的多项式核。《计算机与系统科学期刊》，84:1–10，2017年。 doi:10.1016/j.jcss.2016.06.004。 22 Fedor V. Fomin、Daniel Lokshtanov、Saket Saurabh 和 Meirav Zehavi。核化。《参数化预处理理论》。剑桥大学出版社，2019年。

23 András Frank 和 Éva Tardos. 同时丢番图逼近在组合优化中的应用。《Comb.》，7(1):49–65，1987年。 doi:10.1007/BF02579200。24 E. Mark Gold. 根据给定数据进行自动机识别的复杂性。《Inf. Control.》，37(3):302–320，1978年。 doi:10.1016/S0019-9958(78)90562-4。25 Bart M. P. Jansen 和 Hans L. Bodlaender. 重新审视顶点覆盖核化——精细化参数的上界与下界。《计算系统理论》，53(2):263–299，2013年。26 Bart M. P. Jansen、Jari J. H. de Kroon 和 Michal Wlodarczyk. 由消除距离及更小参数化的顶点删除。载于第53届年度ACM SIGACT计算理论研讨会论文集（STOC），第1757–1769页。ACM，2021年。 doi:10.1145/3406325.3451068。27 David Kempe、Jon M. Kleinberg 和 Amit Kumar. 时序网络的连通性与推理问题。《计算机与系统科学期刊》，64(4):820–842，2002年。 doi:10.1006/jcss.2002.1829。28 Nina Klobas、George B. Mertzios、Hendrik Molter、Rolf Niedermeier 和 Philipp Zschoche. 时间中的无干扰游走：时间不相交路径。《Auton. Agents Multi Agent Syst.》，37(1):1，2023年。 doi:10.1007/s10458-022-09583-5。29 Pascal Kunz、Hendrik Molter 和 Meirav Zehavi. 哪些图结构能够高效地寻找时序不相交路径与游走？CoRR，abs/2301.10503，2023年。 arXiv:2301.10503,doi:10.48550/arXiv.2301.10503。30 Wing Ning Li. 双段信道布线是强 NP 完全问题。《Discret. Appl. Math.》，78(1-3):291–298，1997年。 doi:10.1016/S0166-218X(97)00020-6。31 Othon Michail. 时序图简介：算法视角。《Internet Math.》，12(4):239–280，2016年。 doi:10.1080/15427951.2016.1177801。32 Othon Michail 和 Paul G. Spirakis. 时序图中的旅行商问题。《Theor. Comput. Sci.》，634:1–23，2016年。 doi:10.1016/j.tcs.2016.04.006。33 Polina Rozenshtein 和 Aristides Gionis. 时序网络挖掘。载于第25届ACM SIGKDD知识发现与数据挖掘国际会议论文集（KDD），第3225–3226页。ACM，2019年。 doi:10.1145/3292500.3332295。34 Claude E Shannon. 迷宫求解机器的展示。载于《Claude Elwood Shannon论文集》，第681–687页，1993年。35 Shadi Taghian Alamouti. 探索时序环与网格。硕士论文，康考迪亚大学，2020年。36 Johannes Uhlmann 和 Mathias Weller. 以反馈边集为参数的双层平面化。《Theor. Comput. Sci.》，494:99–111，2013年。 doi:10.1016/j.tcs.2013.01.029。