arXiv:2301.02569v1 [cs.DS] 6 Jan 2023

稀疏图中的模式查找与计数

BalagopalKomarath ！
IIT Gandhinagar，印度

AnantKumar ！ II
T Gandhinagar，印度

SuchismitaMishra ！ 安德
烈斯·贝略大学，智利

AditiSethia ！ IIT
Gandhinagar，印度

摘要

我们研究在稀疏宿主图中查找和计数小型固定图的算法。在非稀疏情形下，树深度和树宽这两个参数分别在快速常数空间算法和多项式空间算法中发挥着关键作用。我们发现了两个新参数，称为匹配树深度和匹配树宽。我们证明，对于许多模式，在宿主图稀疏时，查找和计数匹配树深度较低且匹配树宽较低的模式，可以比现有算法渐近地更快。作为查找和计数固定大小模式的应用，我们发现了 $\tilde{O}(m^3)$时间 [1]个常数空间算法，可用于处理长度至多为 11 的环；还发现了$\tilde{O}(m^2)$时间的多项式空间算法，可用于处理长度至多为 10 的路径。

2012年 ACM 学科分类 计算理论 → 算法设计与分析；计算理论 → 图算法分析

关键词与短语 子图检测与计数、同态多项式、树宽与树深度、匹配

1 引言

给定简单图 G（称为模式）和 H（称为宿主图），一个基本的计算问题是查找或计数 H 中 G 的出现。 G 在 H 中出现是什么意思？最常见的三种出现概念由映射 $\phi : V(G) \mapsto V(H)$ 来刻画。我们这样定义：

1. 若 $\{u, v\} \in E(G)$ 蕴含 $\{\phi(u), \phi(v)\} \in E(H)$ 且 ϕ 是一一映射，则称ϕ 见证了 H中 G 的一个子图同构。该子图由 ϕ像中的顶点和边构成。-子图的数量 H就是其中这类子图 G' 的数量 H。

2. 若 $\{u, v\} \in E(G)$ 等价于 $\{\phi(u), \phi(v)\} \in E(H)$ 且 ϕ 是一一映射，则 ϕ 见证了 H中 G的一个诱导子图同构。该诱导子图由映像中的顶点及由这些顶点诱导出的全部边构成 ϕ。

3. 如果 $\{u, v\} \in E(G)$ 蕴含 $\{\phi(u), \phi(v)\} \in E(H)$，那么我们称 ϕ 是从G到 H的同态。请注意，与子图同构不同， ϕ 不要求是一一映射。

[1] $\tilde{O}$ 隐藏了与输入规模的对数相关的因子。

2 稀疏图中的模式查找与计数

对于上述任一概念，检测问题显然属于 NP。它们也都是 NP 困难问题 CLIQUE 的直接推广。因此，一般而言，在任何这些概念下，高效地查找或计数模式的算法都不太可能存在。

即使我们将注意力限制在固定模式图上，模式检测与计数问题这一类问题仍然饶有趣味。Williams [21] 证明，寻找三角形的改进算法可用于为诸如 NP-完全问题MAX2SAT设计更快的算法。对于大小为 k 的固定模式图，暴力搜索算法如下：遍历所有 k 个 $V(H)$ 元组，并检查 G 是否出现在 H 在该 k 元组所含顶点上诱导出的子图中。该算法需要 $\theta(n^k)$ 的时间和常数空间。因此，当我们将注意力限制在固定模式上时，就希望在尽可能降低空间使用量的同时，改进这一运行时间。降低运行时间主要有两种方法：将快速矩阵乘法算法用作子程序，以及利用模式图的结构性质。

若 A 是该图的邻接矩阵，那么 Nešet˘ril 和 Poljak [17]证明，可以利用恒等式 $\operatorname{trace}(A^3) = 6\Delta$得到一个 $O(n^\omega)$时间算法来计数三角形，其中 Δ 是该图中三角形的数量，$\omega < 2$为矩阵乘法指数。使用一个简单的归约，他们将这一方法扩展为一个以 $O(n^{k\omega})$时间计数 3k团的算法。他们还证明，可以利用改进后的 k团计数算法来计数任意 k顶点模式。后来，Kloks、Kratsch 和 Müller [13] 展示了如何利用快速矩形矩阵乘法，为计数各种大小的团（而不只是三的倍数）带来类似的运行时间改进。请注意，这些算法带来的改进适用于所有 k顶点模式。也就是说，它们并未利用模式的结构来设计更好的算法。除非指数时间假设不成立，否则寻找一个 k团需要 $n^{\Omega(k)}$时间，因此我们需要利用模式的结构，才能得到显著更快的算法。

对于比团更稀疏的模式，其运行时间甚至可以显著优于基于快速矩阵乘法的（模式查找）算法。关键在于利用模式图的结构。一个 k-游走多项式是一个多项式，其单项式对应于长度为 k 个顶点的游走。例如，游走 (u, v, w, x) 对应于单项式 $x_{uv}x_{vw}x_{wx}$ ，而游走 (u, v, u, v) 对应于单项式 x_{uv}^3 [2]。Williams [22]证明，我们可以通过（1）计算 k-游走多项式，以及（2）检查其中是否含有多线性单项式，来检测图中的 k-路径。我们可以使用一个简单的动态规划算法在线性时间内计算 k-游走多项式，然后在适当的环上对该多项式求值；当随机选取的元素满足 $a^2 = 0$时，就可以以高概率检测出多线性单项式。由此得到一个 $O(2^k(n+m))$-时间算法，用于在具有 k个顶点的 n个顶点、 m条边的宿主图中查找作为子图的路径。

现在我们来考虑稀疏模式计数问题。对于将 k-路径作为子图进行计数，Curticapean、Dell 和 Marx [4] 提出的当前最佳算法 $O(f(k)n^{0.174k+o(k)})$只需 f的时间，其中 f是某个函数。对于固定模式图，Alon、Yuster 和 Zwick [1] 给出了$O(n^\omega)$时间算法，用于计数长度至多为 8 的环子图；该算法结合了快速矩阵乘法与对模式结构的利用。注意，这与计数三角形（3-团）所需的时间相同。

[2] 我们用 uv 表示边 $\{u, v\}$。

事实表明，图同态的概念在上述所有用于寻找和计数非团子图的改进算法中都起着关键作用。具体而言，Fomin、Lokshtanov、Raman、Rao 和 Saurabh [11] 展示了如何利用同态多项式（见定义 17）的高效构造——它是 k-游走多项式的一种推广——高效检测树宽较小的子图。他们的算法可以视为将 Williams 针对 [22] 条 k-路径的算法推广到任意图。类似地，Curticapean、Dell 和 Marx [4]指出，可以从用于高效计数小树宽图的同态的算法，推导出高效的子图计数算法。他们的算法可以视为对 Alon、Yuster 和 Zwick 的环计数算法 [1]的推广。

寻找和计数稀疏宿主图中的模式的算法也得到了研究。设计和分析这些算法时，还考虑了一个额外参数，即 m宿主图中的边数。在最坏情况下， m 可能高达 $\binom{n}{2}$，因此，对于某些 t 而言，一个 $O(n^t)$时间算法和一个 $O(m^{t/2})$时间算法具有相同的渐近时间复杂度。然而，在实际应用中， $m = o(n^2)$是很常见的。例如，如果宿主图表示道路网络，那么 $m = O(n)$，其中常数因子由任一交叉路口处道路数量的最大值决定。在这种情况下，一个$O(m^{t/2})$时间算法在渐近意义下优于一个 $O(n^t)$时间算法。

利用快速矩阵乘法和/或模式的结构参数来获得改进算法这一总体思路，在稀疏宿主图的情形下仍然适用。Eisenbrand 和 Grandoni 利用快速矩阵乘法 [8] 证明了我们可以在k时间内计数 $O(m^{k\omega/6})$-团。Kloks、Kratsch 和 Müller [13]证明了 K_4 子图可以在 $O(m^{(\omega+1)/2})$时间内计数。再次由于 $\omega < 3$，这优于暴力算法给出的 $O(m^2)$时间。利用模式的结构参数，Kowaluk、Lingas 和 Lundell [16]在稀疏宿主图的设置下获得了许多改进算法。例如，他们的方法得到了一种在 $O(m^4)$时间内运行、用于计数P_{10} 作为子图的算法。在本文中，我们得到了一种 $\widetilde{O}(m^2)$时间算法，用于计数 P_{10} （类似的改进见定理12、13、14、15 ）。

我们考虑的计算模型是单位成本 RAM 模型。具体而言，我们可以用常数个字[3]存储宿主图中顶点和边的标签。在该模型中，上文提到的基于快速矩阵乘法和/或树宽的算法需要多项式空间。然而，暴力搜索算法只需使用常数空间，因为它一次只需存储 k 个顶点标签（回想一下，我们将 k 视为常数）。在保持常数空间使用量的同时，我们能获得多大的加速？图参数树深度对于回答这个问题至关重要。众所周知，对于树深度为 d 的模式，我们可以在 $O(n^d)$时间内计数其同态，同时只使用常数空间（关于计数这些同态的算术公式构造，参见 Komarath、Rahul 和 Pandey [14]。这些算术公式可以在常数空间内隐式构造并求值。）。由于除了 k团以外，所有 k顶点模式的树深度都严格小于 k，因此在保持常数空间使用量的同时，这一结果立即带来了相较于暴力搜索运行时间的改进。在本研究中，对于模式图为长度至多为 11 的环的稀疏宿主图，我们改进了基于树深度的算法（见定理2）。

[3] 在图灵机模型或对数代价 RAM 模型中，存储顶点标签需要 $O(\log n)$ 空间。

1.1 与图同态多项式算术电路的联系

这些算法中常用的一个子程序，是 Diaz、Serna 和 Thilikos 提出的一种算法 [6]，它能够高效地计算从小树宽模式到任意宿主图的同态数量。事实上，可以证明，该算法很容易推广为高效构造同态多项式的电路，而不是计数同态。Bläser、Komarath 和 Sreenivasaiah [2]证明，高效构造同态多项式甚至可以在某些情况下用于检测诱导子图。他们还指出，许多更快的诱导子图检测算法，例如 Williams 等人查找四节点子图的算法 [23] 以及 Kowaluk、Lingas 和 Lundell 查找五节点子图的算法 [16]，都可以描述为高效构造这些同态多项式的算法。因此，图同态多项式的算术电路为描述我们已知的几乎所有快速子图查找与计数以及诱导子图查找算法，提供了一个统一框架。我们能否通过找到更高效的同态多项式算术电路构造方法来改进这些算法？遗憾的是，已知对于所构造的这类电路，即不涉及抵消的电路，对于所有模式图，现有构造已是最优的，正如 Komarath、Pandey 和 Rahul 所示 [14]。常数空间算法的情况也类似。目前已知的最佳算法可以表示为分治算法：利用图参数树深度构造小型公式，再对其求值。Komarath、Pandey 和 Rahul[14] 还证明，对于所有模式图，这些算法的运行时间都达到了可能的最佳公式规模。这些算术电路下界构成了我们考虑稀疏宿主图的技术动机，也补充了前文提到的实际动机。

1.2 我们的发现

本文研究在宿主图中查找和计数模式的算法，尤其关注宿主图稀疏时表现良好的算法。我们发现了一些算法，它们（1）严格优于暴力算法；（2）在宿主图稀疏时严格优于已知最佳算法；（3）在宿主图稠密时接近已知最佳算法。我们的算法基于两个新的结构图参数——匹配树深度 和 匹配树宽。（正式定义见 19和21）。我们证明，这些参数可用于分别改进采用常数空间和多项式空间的算法的运行时间。我们的算法总结见表 1。表中，参数 m 表示宿主图中的边数。我们用 mtw 表示模式的匹配树宽，用 mtd 表示模式的匹配树深度。记号$\tilde{O}$省略了关于输入（宿主图）规模的对数多项式因子。

接下来，我们将在本节余下部分阐述这些新参数的重要性，介绍我们的算法、各种图参数之间的关系，以及本文证明的一些结构刻画。

1.2.0.1 树深度与匹配树深度

图的匹配树深度与其树深度密切相关。基于树深度的常数空间算法本质上是在模式图的消去树上执行的分治算法，该算法会对消去树中的每条根到叶路径进行暴力搜索。因此，其时间复杂度为 $O(n^d)$。我们利用了以下事实：

模式	Type	问题	Time	空间	备注
C_k	子图	计数	$\widetilde{O}(m^3)$	$O(1)$	$k \leqslant 11$
P_k	子图	计数	$\widetilde{O}(m^2)$	$\widetilde{O}(m^2)$	$k \leqslant 10$
C_k	子图	计数	$\widetilde{O}(m^2)$	$\widetilde{O}(m^2)$	$k \leqslant 9$
Any	同态	计数	$\widetilde{O}(m^{\lceil mtd/2 \rceil})$	$O(1)$	
Any	同态	计数	$\widetilde{O}(m^{\lceil (mtw+1)/2 \rceil})$	$\widetilde{O}(m^{\lceil (mtw+1)/2 \rceil})$	
C_6	诱导子图	检测	$\widetilde{O}(m^2)$	$\widetilde{O}(m^2)$	
$\overline{P_k}$	诱导子图	检测	$\widetilde{O}(m^{\lceil (k-2)/2 \rceil})$	$\widetilde{O}(m^{\lceil (k-2)/2 \rceil})$	

表 1模式　　稀疏宿主图的计数和检测算法　　图。

消去树是匹配的，这要求每条根到叶路径中的顶点都必须由一个匹配覆盖。这样，算法中的暴力搜索部分就能仅使用 $d/2$ 条边发现路径上的所有 d 顶点。我们用来获得常数空间算法的核心算法如下：

▶ **定理 1.** 设 G是一个具有匹配树深度$(G) = d$的图，那么给定一个作为输入的 m边图 H，我们可以在 $\widetilde{O}(m^{\lceil d/2 \rceil})$时间内以常数空间计数从 G到 H 的同态数量。

众所周知，对于任意 G，G-子图的数量可以表示为从一组相关图到的同态数量的线性组合，这组相关图称为 G的 Spasm。G 的 Spasm 恰好包含所有通过反复合并 G中的独立集而得到的图。（形式化定义见 26 。）尽管 C_{11} 的 Spasm 的树深度有上界 6，但匹配树深度不一定有树深度那么大的上界。我们分析了 C_{10} 和 C_{11} 的 Spasm 中的所有图（共有 501 个），并证明每个图的匹配树深度至多为 6。由此得到以下算法：

▶ **定理 2.** 给定一个有 m 条边的图 H 作为输入，我们可以在 $\widetilde{O}(m^3)$ 时间和常数空间内，统计 C_k（其中 $k \leqslant 11$）作为子图的数量。

作为比较，暴力算法需要 $O(m^6)$ 时间和常数空间；基于树深度的算法需要 $O(n^6)$ 时间和常数空间。

从我们用于计数 C_{11} 的算法证明中可以看出，即使对于相对较小的模式，一个模式的 Spasm 也可能包含大量图。因此，若能有对匹配树深度给出上界的定理，将会很有帮助。遗憾的是，与树深度不同，性质 $mtd(G) \leqslant k$ 甚至不是子图闭合的。例如，可以证明 $mtd(K_4 - e) = 3$，但 $mtd(C_4) = 4$。不过，关于匹配树深度仍然可以得出有趣的结构性观察。下面的定理根据树深度给出了匹配树深度的上界。

▶ **定理 3.** 对于任意图 G，匹配树深度$(G) \leqslant 2 \cdot$ 树深度$(G) - 2$。

定理 3 意味着，与基于树深度的算法相比，我们在定理 1 中用于同态计数的常数空间算法，对于 所有 模式的渐近运行速度都更快，其中输入是稀疏宿主图。

以下定理表明，模式的同态计数时间复杂度，下界不低于其所有诱导子图的计数时间复杂度。

▶ **定理 4.** 设 G是一个图，且 G' 是 G的一个连通诱导子图，则：

1. $\mathrm{mtd}(G') \leqslant \mathrm{mtd}(G)$ 当 $\mathrm{mtd}(G)$ 为偶数时。

2. 匹配树深度$(G') \leqslant$ 匹配树深度$(G)+1$ 当匹配树深度(G) 为奇数时。

鉴于匹配树深度的重要性，我们亟须尽可能深入地理解这一结构参数。树深度为 2 的图恰好构成星图这一类图。这也正是匹配树深度为 2 的图的类别。然而，对于图 C_4，我们有 $\mathrm{td}(C_4) = 3$ and $\mathrm{mtd}(C_4) = 4$。因此，确切地了解哪些图的树深度与匹配树深度相等，是一个值得探究的问题。以下定理应被视为对这两个参数之间关系的初步认识。

▶ **定理 5.** 设 G是一个图，满足 树深度$(G) = 3$。那么，匹配树深度$(G) = 3$ 当且仅当 G 是$(C_4, P_6, T_{3,3})$-自由的。

图$T_{3,3}$ 是 $(3, 3)$ 蝌蚪图（见图2）。

1.2.0.2 树宽与匹配树宽

Díaz、Serna 和 Thilikos 基于树宽的动态规划算法 [6]可以增强为输出一个算术电路，用于计算同态多项式。算术电路是一种有向无环图，其中每个内部节点都标记为 $+$ 或 $\times$；每个叶节点都标记为变量或域常数，并且指定了一个输出节点。这种图以自然的方式计算底层域上的多项式。我们发现，通过在匹配树分解上使用动态规划算法，可以减小稀疏宿主图所对应的算术电路规模。我们的核心定理如下：

▶ **定理 6**。设 G是一个图，其$\mathrm{mtw}(G) = t$，则给定一个作为输入的 m-边宿主图 H，我们可以在时间 $\widetilde{O}(m^{\lceil (t+1)/2 \rceil})$内构造一个算术电路，计算从$G$到 H 的同态多项式。

对于匹配树宽与树宽相同的图，同态计数的运行时间比 Díaz、Serna 和 Thilikos 的算法快一个二次量级[6]（针对稀疏图）。因此，若不改进 Díaz、Serna 和 Thilikos 的算法，这也是所能期待的最大改进。 [6]最坏情况如何？下述定理表明，所得算法在稀疏宿主图上不会更差。

▶ **定理 7.** 对于任意图 G，都有 $\mathrm{mtw}(G) \leqslant 2 \cdot \mathrm{tw}(G) + 1$。

遗憾的是，与树宽不同，参数 $\mathrm{mtw}(G)$ 在图的子图偏序下并非单调。我们首先观察到一个明确的图族，其树宽较低而匹配树宽较大。考虑完全二分图 $K_{n,\,n}$ 它有 n个顶点。注意到 $tw(K_{n,n}) = n$。

▶ **命题 8.** $\mathrm{mtw}(K_{n,n}) = 2n - 2$ 对所有 $n > 1$。

下面的观察表明，存在一些超图，其 mtw 小于$K_{n,\,n}$的 mtw。

▶ **观察 9.** 考虑 K_n 的超图 G_n，使得 $V(G) = V(H)$，并且 K_n 的一个划分中存在边 n，从而使大小为 n 的独立集变为一条包含 n 个顶点的路径。注意，尽管 $\mathrm{mtw}(K_{n,n}) = 2n - 2$，但 $\mathrm{mtw}(G) = n$。

我们将展示如何利用关于匹配树宽的结构定理来证明算法上界。例如，要计算 P_{10} 子图的数量，我们只需证明 P_{10} 的 Spasm 中所有图的匹配树宽都很小。 P_{10} 的 Spasm 是一个庞大的集合，包含 300 多个图。确实，可以逐一分析这些图的匹配树宽。不过，如果我们有定理能够免去这类繁琐的工作，就更好了。

我们推导出了关于低匹配树宽值的一些结构定理。匹配树宽为 1 的图恰好是树。我们还证明了 $\mathrm{tw}(C_5) = 2$ 和 $\mathrm{mtw}(C_5) = 3$。

我们使用禁用诱导子式刻画了部分 2-树的匹配树宽（参见定义 24），并在可能的情况下给出完整刻画。我们证明， C_5 恰好是迫使部分 2-树具有更高匹配树宽的障碍。

▶ **定理 10**。 对于任意部分 *2*-树 G，图 G是 C_5-诱导子式自由的，当且仅当$\mathrm{mtw}(G) = 2$。

注意，$\mathrm{tw}(G) = 2$ 给出了用于同态计数的 $O(n^3)$时间算法。即使 $\mathrm{mtw}(G) = 3$，我们也能得到用于同态计数的 $\tilde{O}(m^2)$时间算法，这对于稀疏图而言是一种改进。所有树宽为 2 的图是否都具有至多 3 的匹配树宽？不是。图 X （见图 1 ）的树宽为 2，匹配树宽为 4（参见观察 35）。事实上，我们可以证明， X 恰好是迫使树宽为 2 的图具有匹配树宽 4 的障碍。

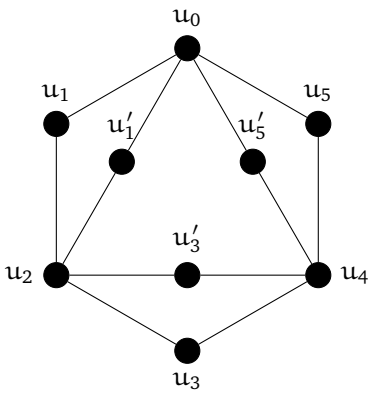

图 1 图 X。

▶ **定理 11**。 对于任意部分 *2*-树 G，图 G 是 X-诱导图子式自由的，当且仅当$\mathrm{mtw}(G) \leqslant 3$。

该定理意味着，所有 X-诱导子式自由、树宽为 2 的模式都存在 $\tilde{O}(m^2)$-时间复杂度的同态计数算法。对于稀疏宿主图，即使与 Curticapean、Dell 和 Marx 提出的基于快速矩阵乘法的算法相比，这也是一项改进。该算法用于计算来自树宽为 2 的图的同态数量，其运行时间为 [9] 。 $O(n^\omega)$由于 P_{10} 的 Spasm 中不包含任何树宽为 4 的图，也不包含任何具有 X-诱导子式的图，我们可以证明，若要对至多含 10 个顶点的所有路径计算子图同构的数量，只需证明 $\tilde{O}(m^2)$的 Spasm 中所有树宽为 3 的图都具有匹配树宽 3，即可得到一个 P_{10}-时间复杂度的算法。这样的图只有 18 个。分析它们的匹配树宽可得以下定理：

▶ **定理 12**。给定一个 m-边图 H 作为输入，我们可以在 $\widetilde{O}(m^2)$-时间内计算 $k \leqslant 10$的 P_k 子图数量。

据我们所知，目前已知最佳的路径计数算法处理 10 个顶点的路径需要 $\Omega(n^4)$ 时间。因此，对于稀疏宿主图，我们的算法有显著改进；对于稠密宿主图，其性能也不逊于目前已知的最佳算法。下面给出这一结果的一个简单推论：

▶ **定理 13.** 给定一个 m-边图 H作为输入，我们可以在 $\widetilde{O}(m^2)$时间内计数长度至多为9 的环。

这些环计数算法在稀疏图上的性能优于 $O(n^\omega)$Alon、Yuster 和 Zwick 给出的、用于计算长度至多为 8 的环的 -时间算法 [1]。

我们还展示了如何利用改进后的同态多项式构造算法，加速诱导子图的检测。具体而言，我们证明了以下结果：

▶ **定理 14.** 给定一个含 m 条边的宿主图作为输入，我们可以在 $\widetilde{O}(m^2)$时间内找到一个诱导 C_6，或者报告不存在这样的子图。

该算法的运行时间不差于 $O(n^4)$ 利用 Bläser、Komarath 和 Sreenivasaiah 的技术可以得到的时间算法 [2]。对于稀疏图，我们的算法带来了二次级的改进。我们还证明了以下结果：

▶ **定理 15**。 给定一个具有 m 条边的宿主图作为输入，我们可以在 $\widetilde{O}(m^{(k-2)/2})$ 时间内找到一个诱导 P_k，或者报告不存在这样的对象。

当宿主图稀疏时，这相较于 $O(n^{k-2})$ Bläser、Komarath 和 Sreenivasaiah 提出的时间算法 [2] 也有二次级的改进。这些算法是通过分析匹配树宽 以及由模式导出的一组图的自同构结构而得到的。

从技术角度来看，我们认为我们的算法自然结合了模式~检测与计数算法，这些算法在稀疏宿主图上表现良好，例如用于计数 k-游走的 $O(m)$ 算法、用于计数 k-团的 $\widetilde{O}(m^{k/2})$ 算法，以及 Vassilevska [19] 提出的用于检测诱导 $K_k - e$ 的 $\widetilde{O}(m^{(k-1)/2})$ 时间算法；同时也结合了通过利用结构参数树深度和树宽来改进稠密图运行时间的见解。我们的算法均未使用快速矩阵乘法。这类算法称为组合算法，在学界也普遍受到关注。

1.3 相关工作

计数诱导子图的算法与我们研究的问题相关，但我们不考虑任何此类算法。这个问题似乎难得多。Floderus、Kowaluk、Lingas 和 Lundell [9]猜想，对于足够大的 k，计数任意 k 个顶点的模式图的诱导子图，其难度与计数 k 团一样。已有多项研究考察了子图计数的参数化复杂性（参见 [5, 4, 18,10, 7]），其首要目标是根据结构图参数，划分出容易与困难问题的二分。一些研究也考察了对宿主图的限制，例如 d-退化性 [3]。参数化复杂性的论文主要关注一族模式（例如 k-路径、 k-环或 k-团）的指数增长率，而非像本文这样关注小图的精确指数。

2 预备知识

我们研究简单图。关于图论基本定义，读者可参阅 Douglas West 的教科书 [20] 。对于一些众所周知的图，我们采用以下常用记号： P_k 表示 k 个顶点的路径， C_k 表示 k-环， K_k 表示 k-团， $K_k - e$ 表示缺少一条边的 k-团，$K_{m,n}$ 表示完全二分图。 k-星是一个有 $(k+1)$ 个顶点的图，其中一个顶点 u 与顶点 v_1、……、v_k 相邻，且没有其他边。对于某个 k，星图是一个 k-星。对于图 G 和 $S \subseteq V(G)$，我们用 $G[S]$ 表示由 G 中 S里的顶点所诱导的子图。

▶ **定义 16.** 给定两个图 G 和 H，从 G到 H 的图同态是一个映射$\phi : V(G) \to V(H)$，满足：若 $uv \in E(G)$，则 $\phi(u)\phi(v) \in E(H)$。

我们用 $\mathrm{Hom}(G, H)$ 表示从 G到 H的所有同态的集合。

▶ **定义 17.** 给定两个图 G 和 H，同态多项式是与之关联的多项式 $\mathrm{Hom}_G[H]$，其单项式与从 G到 H的同态之间存在一一对应关系。我们定义：

$$\mathrm{Hom}_G[H] = \sum_{\phi \in \mathrm{Hom}(G,H)} \prod_{u \in V(G)} y_{\phi(u)} \prod_{\{u,v\} \in E(G)} x_{\{\phi(u),\phi(v)\}}$$

注意， H 含有一个与 G同构的子图，当且仅当 P_G 含有一个多线性单项式。

我们称图 G'是 $(G_1, \ldots\ldots, G_m)$-自由的，如果对于任意 i， G'的任何诱导子图都不同构于G_i。我们用 $\overline{G}$表示图 G的补图。我们有 $V(G) = V(\overline{G})$，且 G的迈恰好是 G 的非边，反之亦然。

我们假设所有模式图都是连通的。由于我们的主要算法都基于同态计数，这并不会损失一般性，因为从一个不连通模式出发的同态数量，恰好等于从其各个连通分量出发的同态数量之积。

▶ **定义 18。** 连通图 G 的消去树 (T, r) 是以 $r \in V(G)$为根的一棵树，其中 r 的子树递归地是图 $G \setminus r$的各个连通分量的消去树。空图（没有顶点或边）的消去树为空树。消去树 (T, r) 的深度定义为 T 中所有根到叶路径上的顶点数的最大值。图 G 的树深度 记作 $\mathrm{td}(G)$，是 G所有可能的消去树中最小的深度。

直观而言，树深度衡量给定图与星图的接近程度；星图恰好是树深度为 2 的连通图。我们引入一个相关概念，称为匹配树深度，它似乎有助于设计用于在稀疏宿主图中查找或计数模式的算法。

▶ **定义 19。** 图 G的匹配消去树 是一棵消去树，且对任意根到叶路径 $(v_1, \ldots, v_k)$，以下条件均成立：

- 若 k 为偶数，则 $v_1v_2, v_3v_4, \ldots, v_{k-1}v_k$ 是 G中的一个匹配。若 k 为奇数，则存在某个 i，使得 $E' = \{v_1v_2, \ldots, v_{i-1}v_i, v_iv_{i+1}, \ldots, v_{k-1}v_k\}$和$E' \subseteq E(G)$。我们有，$E' \setminus \{v_{i-1}v_i, v_iv_{i+1}\}$ 是在 $(k-3)/2$ 个顶点上的一个匹配。

图 G 的匹配树深度 （记作$\mathrm{mtd}(G)$）是 G所有可能的匹配消去树中深度的最小值。

匹配树深度始终有限（见命题 25）。

▶ **定义 20.** 图 G 的树分解是一个二元组 $(T, B(t)_{t\in T})$，其中 T 是一棵树，而 $B(t)$（称为一个袋）是 G 的顶点子集的集合，每个节点$t \in T$对应一个子集。

- （连通性性质）对于所有 $v \in V(G)$，都存在一个节点 $t \in V(T)$，使得 $v \in B(t)$，并且所有包含 v 的此类节点 t 在 T 中构成一个连通分量。
- （边性质）对于所有 $e \in E(G)$，都存在一个节点 $t \in V(T)$，使得 $e \subseteq B(t)$。

树分解的宽度 (T, B) 定义为最大袋大小减一，即最大值$_{t\in T} |B(t) - 1|$。图 G 的树宽 $tw(G)$ 是 G 所有可能的树分解中宽度的最小值。

直观地说，树宽衡量给定图与树的接近程度，而树恰好是树宽为 1 的图。我们引入一个与树宽密切相关的概念，称为匹配树宽；它似乎有助于为稀疏宿主图设计动态规划算法。

▶ **定义 21**。图 G 的匹配树分解 是这样一种树分解：对于树分解中的每个袋，由该袋中的顶点在 G 中诱导出的子图，要么有一个完美匹配，要么有一个匹配，其中恰有一个袋中的顶点 v 未匹配，且 v 与匹配中的某个顶点相邻。我们称这样的袋为匹配的。图 G、$mtw(G)$ 的匹配树宽 是 G 的所有可能匹配树分解中可达到的最小宽度。

所有图的匹配树宽都是有限的（参见命题31）。这并不像树宽那样显而易见，因为包含图中所有顶点的单袋树分解未必是匹配的。

如果没有任何一个袋 B 是另一个袋的子集，我们就称树分解为约化的。给定任意树分解 T，我们可以得到一个约化树分解 T'，使得 T' 的宽度至多为 T 的宽度。此外，T' 中的所有袋也都是 T 中的袋。这意味着，如果 T 是匹配的，那么 T' 也匹配。

树宽有若干种等价刻画。下面我们陈述本文所用的几种。

▶ **定义 22**。*A* k-树是通过从一个 $(k+1)$-团开始，并反复添加一个与现有 $(k+1)$-团中恰好 k 个顶点相连的顶点而构成的图。图 G 的部分图是通过从 G 中删除边而得到的图。所有树宽至多为 k 的图，恰好构成部分 k-树这一类图。

对于任意 k-树，我们可以如下构造一个标准树分解 T：根袋 T 包含初始 $(k+1)$-团中的顶点。设 S 是该团的一个 k 元子集，且将一个新顶点 v 与 S 中的所有顶点相连，从而将其添加到 k-树中。然后，我们向 T 添加一棵子树，该子树将是以 $S \cup \{v\}$ 作为起始 $(k+1)$-团构造出的 k-树的标准树分解。

图 G 的弦完成 是一个超图 G'，它是 G 的超图，且 G' 不含长度大于 3 的诱导环。使最大团的大小最小的弦完成称为最小弦化补全。图 G 的树宽是其最小弦化补全中最大团的大小。

从 u 到 v 的两条路径 P_1 和 P_2 内部不相交，当且仅当 P_1 和 P_2 没有任何公共内部顶点。

若对任意 $x \in V(G)$，$G-x$ 均连通，则图 G 是*2*-连通 或双连通的。等价地，对于 G 中任意两个顶点，G 中至少存在 2 条内部不相交路径。

▶ **定义 23.** 串并联图是一个三元组 (G, s, t)，其中 s 和 t 是 G 中的顶点。该类图递归定义如下：

- 边 $\{s, t\}$ 是串并联图。（串联组合）若 (G_1, s_1, t_1) 和 (G_2, s_2, t_2) 是串并联图，则将 s_2 与 t_1 识别后得到的图也是串并联图。（并联组合）若 (G_1, s_1, t_1) 和 (G_2, s_2, t_2) 是串并联图，则将 s_1 与 s_2 以及 t_1 与 t_2 分别识别后得到的图也是串并联图。

一个图的树宽至多为2 当且仅当其所有双连通分量都是串并联图。

▶ **定义 24。 </style}如果图 可以通过删除边/顶点或收缩边从图 得到，则称图 是图 的图子式 。**（收缩操作将图中的两个相邻顶点 u 和 v 合并，并删除边 (u, v)。）如果图 G 可以通过收缩图 G' 的一个诱导子图中的边得到，则称其为图 G'的诱导子式。

如果图 G' 不是图 G的诱导子式（分别为图子式），则称图 G是 G'-诱导子式-free（G'-图子式-free）的。

边细分是一种删除边 (u, v) 并添加一个新顶点w 以及边 (u, w) 和 (w, v)的操作。通过一系列边细分从图 G 得到的图 G' 称为图 G的细分 。

2.1 图的表示

我们假设基本图操作具有以下时间复杂度。任何满足这些要求的表示都足够。

- ~给定 u 和 v，可以在 $O(1)$时间内检查 uv是否为一条边。
- ~可以在 $O(m)$时间内遍历所有按 x 排序的边 $xy \in E(H)$，其中 m是 H中的边数。

以下邻接表表示满足这些要求。为了表示一个图 H，我们存储一棵红黑树 T，其中包含 H 的非孤立顶点，并按照顶点的标签排序。考虑这棵树中对应于顶点 u的一个节点。该节点存储另一棵红黑树 T_u，其中存储了 H中 u 的所有邻居。现在，要检查 uv 是否为一条边，我们先在 T 中查找 u；如果找到了 u，再在 T_u 中查找 v。我们可以通过对 T 进行中序遍历，按 x 的顺序遍历所有边 xy；对于其中的每个节点 u，再对 T_u进行中序遍历。

如果模式不包含任何孤立顶点，那么我们也可以忽略宿主图中的孤立顶点。如果模式为 $G = G' + v$，其中 v 是一个孤立顶点，G' 是任意图，那么从 G 到 H 的同态数量可通过将

从 G' 到 H 的同态数量乘以 n 得到，其中 n 是 H 中顶点的数量。只需在数据结构中存储 H 的顶点数量即可计算得出。

3 匹配树深度

本节介绍在稀疏宿主图上以常数空间高效计数同态和子图的算法。本节的核心定理如下。

▶ **定理 1.** 设 G是一个具有 $\mathrm{mtd}(G) = d$ 的图，则给定一个作为输入的 m 边图 H，我们可以在 $\widetilde{O}(m^{\lceil d/2 \rceil})$ 时间和常数空间内计数从 G 到 H 的同态数量。

证明。 算法见算法 1。我们可以通过调用 COUNT-HOM-MTD(G, E, r, H, ϕ)来计算所需结果，其中 E 是 G 的一棵深度为 d的消去树，r 是 E中的根顶点，而 ϕ 是空同态。为便于说明，我们假设 E 中的每条根到叶路径都包含偶数个顶点。根到叶路径包含奇数个顶点的情况可类似处理。

我们假设宿主图 H 采用对称邻接表表示。这主要是为了确保我们能够遍历 xy 中 H按 x 排序的所有边（第 3 行）。

首先，我们证明该算法是正确的。我们断言，调用 COUNT-HOM-MTD(G, E, v, H, σ)（参数按算法中的指定设置）会返回从 G_v 到 H 且扩展 σ的同态数量。我们通过对 E中 v 的高度进行归纳法来证明这一点。由于 v 是一个顶层节点，基本情形是高度为 2。在这种情况下，G_v 是一个星图，很容易看出算法能够正确运行。现在我们证明归纳情形。变量 t 计算最终答案。对于图 H 中的顶点 x，记 $s_{u,x}$ 为从 G_u到 H且扩展 $\tau = \sigma \cup \{v \mapsto x\}$的同态数量。注意，由于 $uv \in E(G)$，要扩展 τ，必须将顶点 u 映射到满足 $xy \in E(H)$的某个 y。因此，遍历所有这样的 y 就足够了。注意，我们可以将 t 计算为 $\sum_\tau \prod_u s_{u,x}$。但是，这需要存储 $|V(G)||V(H)|$ 个变量。通过按 x的顺序遍历 H 的边，我们可以重复使用同一个 s_u 来处理不同的 x，而不必为每个 x分别保留一个 $s_{u,x}$。现在，我们证明主循环以一个 x结束后，s_u 能够正确计算出 $s_{u,x}$。根据归纳假设，变量 c_w 是从 G_w 到 H 且扩展 $\sigma' = \sigma \cup \{v \mapsto x, u \mapsto y\}$的同态数量。因此，我们有 $s_{u,x} = \sum_y \prod_w c_w$。第 12 行将其正确计算到 s_u中。第 15 行在处理完一个 x 后正确更新 t。最后，在处理下一个 x之前，我们将 s_u重置为 0。

现在，我们证明该算法的运行时间和空间使用量。注意，递归深度受消去树深度的限制 E，并且每一层递归只存储常数个变量。因此，空间使用量为常数。第 3 行中的主循环执行 $2m$ 次迭代。内部循环的迭代次数均为常数。因此，递归调用仅执行 $O(m)$ 次。每一层中，我们处理消去树的两层。因此，总运行时间为 $t(d) \leqslant O(m)t(d-2) + \widetilde{O}(1) = \widetilde{O}(m^{d/2})$。

算法 1 COUNT-HOM-MTD(G, E, v, H, σ)

要求： G- 模式图
要求： E - G 的匹配消去树
要求： v - E 中的一个顶顶点
要求： H- 宿主图
要求： σ- 从 v 的祖先到 H 的部分同态

```
t ← 0
s_u ← 0 对于所有 v 的子节点 u
对于所有 按 x 排序的边 xy ∈ E(H) 执行
    对于 E 中 v 的所有子节点 u 执行
        σ′ ← σ ∪ {v ↦ x, u ↦ y}
        如果 σ′ 是无效同态则
            继续
        结束条件判断
        对于所有子节点 w 的 u 执行
            c_w ← COUNT-HOM-MTD(G, E, w, H, σ′)
        结束循环
        s_u ← s_u + ∏_w c_w
    结束循环
    如果 xy 是 x上的最后一条边，则
        t ← t + ∏_u s_u
        s_u ← 0 对所有 u
    结束条件判断
结束循环
返回 t
```

◀

关于匹配树深度的一个基本问题是，它对所有图是否都有限。如下命题所示，答案是肯定的。

▶ **命题 25.** 对于任意图 G，mtd(G) 至多为 *1* + 中最小极大匹配所含顶点的数量。 G

证明。 我们将 V(G) 划分为一个极大匹配 $M = (v_1w_1, \ldots, v_mw_m)$ 以及一个独立集 $\{u_1, \ldots, u_k\}$。匹配消去树 T 对于 G 可以按如下方式构造：依次将顶点 $v_1, w_1, \ldots, v_m, w_m$ 放在一条根到叶路径上。我们称这条路径为脊柱。对于每个顶点 u_i，将脊柱中与 u_i 在 G中相邻的最低顶点设为其父节点。显然， T 是匹配消去树，其深度至多为 1 +中的顶点数量 M。

◀

尽管命题 25证明了匹配树深度的一个上界，但从算法视角来看，它并不太有用，因为很容易看出，存在一个 $O(m^{k+1})$ 时间、常数空间算法，可以对最大匹配含有 k 条边的模式进行计数。例如， C_{11} 中最小的极大匹配有 4 条边。不过，下面给出的定理 2 的证明表明，我们可以做得更好。该算法基于一种众所周知的技术，即将子图计数表示为同态计数的线性组合。

14 在稀疏图中寻找并计数模式

▶ **定义 26.** 令 $\mathcal{I}$ 为图 G中所有独立集的集合。对于某个 $I \in \mathcal{I}$，合并(G, I) 是通过合并 I 的顶点得到的图。于是

$$\mathsf{Spasm}(G) = \{G\} \cup \bigcup_{I \in \mathcal{I}} \mathsf{Spasm}(\mathsf{Merge}(G, I))$$

对于任意模式图 G，从 G 到宿主图 H 的子图同构数量，恰好是从 SpasmG到 H的所有可能图同态的线性组合。也就是说，存在常数 $\alpha_{G'} \in \mathbb{Q}$，使得：

$$\mathsf{Sub}_G[H] = \sum_{G'} \alpha_{G'} \mathsf{Hom}_{G'}[H] \tag{1}$$

其中 G' 遍历 $\mathsf{Spasm}(G)$中的所有图。许多作者都使用这个等式来对子图进行计数（例如，参见 [1, 4]）。

▶ **定理 2.** 给定一个以 m 条边的图 H 作为输入，我们可以在 $\widetilde{O}(m^3)$-时间和常数空间内，计数作为子图的 C_k 的数量，其中$k \leqslant 11$。

证明。 我们分析了$\mathsf{Spasm}(C_{11})$ 和 $\mathsf{Spasm}(C_{10})$ 中的所有图，并得出结论：每个图的匹配树深度至多为 6。包含所有这些图及其对应匹配消去树的 PDF 文件可在此处找到（https://github.com/anonymous1203/Spasm）。要说明对于较小的环存在所述算法，只需注意 $\mathsf{Spasm}(C_k) \subset \mathsf{Spasm}(C_{k+2})$ 对于 $k \geqslant 3$。◀

树深度为 2 的图类恰好是星图。这些图具有一个 $O(n^2)$-时间、常数空间的基于树深度的算法。这类图也正是匹配树深度为 2 的图。很容易看出，星图的同态数量也可以利用以下观察，在 $O(m)$-时间和常数空间内进行计数。设d_1、.....、 d_n 是图中顶点的度。那么，计数结果由表达式$\sum_i d_i^k$给出。注意，对于稀疏图而言，这是渐近意义上的改进。下面我们将证明，对于所有模式，这种渐近改进都存在。首先，我们需要一种受限形式的消去树。

▶ **定义 27。** 如果对于 T 中的每个节点 u 以及 T中的u 的子节点 v， T_v中都存在某个节点与 u 相邻，则消去树 T 是连通的

现在，我们将证明连通消去树是最优的。

▶ **引理 28。** 每个连通图 G都有一棵深度为 $\mathsf{td}(G)$ 的连通消去树。

证明。我们说明如何从消去树T构造一个连通消去树 T'，且不增加其深度。首先令 $T' = T$。假设存在某个节点 u 属于 T'，但不满足该性质（否则，证明完毕）。那么， T' 中存在 u 的一个子节点 v，使得 G 中没有任何边连接 u 与 T'_v中的任何节点。令 w 为 T'_v 中的一个节点，满足w 与 u的某个真祖先 x 相邻。这样的 w 必定存在，因为 G 是连通的。现在，在 T'中，移除子树 T'_v，并将其设为节点 x的子树。重复此过程，直到 T' 中的所有节点都满足所需性质。该过程必定终止，因为每一步都会使违反该性质的节点数至少减少一个。该过程不会增加T' 的深度，因为唯一的改动是将一棵子树向上移，使其成为其父节点的某个真祖先的子树。◀

▶ **定理 3.** 对于任意图 G，$\mathrm{mtd}(G) \leqslant 2 \cdot \mathrm{td}(G) - 2$。

证明。 我们从一棵连通消去树开始，T 其深度为 d，G 并展示如何从 T构造一棵深度为 G 的匹配消去树。我们对 d使用归纳法。对于 $d = 2$，该树已经是匹配消去树，且深度为 $2 = 2 \cdot 2 - 2$。

我们的构造采用迭代式自顶向下方法。每次迭代都会确保消去树中当前最顶端的节点在图 G 中与其所有子节点相邻，并确保消去树连通。

每次迭代包含两个阶段，并反复执行，直到满足所需性质。第一阶段确保根节点 r 在 G 中与 T 的所有子节点相邻。如果 r 有一个子节点 v，它在 G 中不与 r 相邻，那么由于 T 连通，T_v 中存在某个节点 w，使得 $rw \in E(G)$。接着，我们在 T 中将 w 设为 r 的子节点，从 T_v 中删除 w，并将 w 的所有子节点设为 w 的父节点的子节点。所得树是一棵深度至多为 $d+1$ 的消去树。此阶段结束后，根节点在 G 中与其所有子节点相邻，树的深度至多增加一层。不过，它可能并不连通。

在第二阶段，我们使用引理 28 中的构造，使树变得连通，同时不增加深度。注意，该构造会保留根原有的子节点不变，并可能为 r 添加新的子节点，这些子节点与 r 在 G中不相邻。假设在第二阶段，添加了一个新节点 u 作为 r 的新子节点。以 u为根的子树高度至多为 $d-1$。因此，通过将 u连接到 r所得到的树 (r, T_u) 的高度至多为 d。现在，我们可以对所有此类 u 对应的树 (r, T_u) 执行阶段 1，而不使深度超过 $d+1$。这个过程最终必定会终止，因为每次我们都会为根至少添加一个新的子节点。

在这一轮迭代结束时，考虑 r的一个孙节点 x。如果它是叶节点，由于这棵树是连通的，x 在 G 中必定与它在 T 中的父节点相邻，因此问题解决。否则，子树 T_x 是一棵深度至少为 2 且至多为 $d-1$ 的连通树。因此，根据归纳假设，我们得到 T_x 是一棵深度至多为 $2(d-1)-2$的匹配消去树。这意味着，原树被转换为一棵深度至多为 $2 + 2(d-1) - 2 = 2d - 2$ 的匹配消去树，符合要求。◀

▶ **推论29。** 假设 G 的树深度为 d。给定一个含 m条边的宿主图作为输入，我们可以在 $\tilde{O}(m^{d-1})$时间内，以常数空间计算从 G 到该宿主图的同态数量。

请注意，对于稀疏宿主图上的所有模式，上述算法在渐近意义上都优于基于树深度的算法。

现在我们证明，诱导子图的匹配树深度不会比包含它的图大很多。换言之，通过获得图的任意诱导子图的匹配树深度下界，我们可以得到该图匹配树深度的下界。我们对连通子图证明该定理。但请注意，非连通图的匹配树深度等于其各连通分量的匹配树深度的最大值。

▶ **定理 4。** 设 G是一个图，且 G' 是 G的一个连通诱导子图，则：

1. $\mathrm{mtd}(G') \leqslant$mtd(G) 如果mtd(G) 为偶数。
2. $\mathrm{mtd}(G') \leqslant \mathrm{mtd}(G) + 1$ 若 $\mathrm{mtd}(G)$ 为奇数。

证明。 我们从一棵 T 深度为偶数的匹配消去树（奇数情形类似）d 出发，针对 G 构造一棵 G′的匹配消去树。首先，删除消去树中属于 G 但不属于 G′的所有节点。若 u 中某个节点 T 的父节点 v 已被删除，则将 u 设为 v 中离 T最近且仍在其中的祖先的子节点。由此得到的最终森林是一棵树 T′，因为 G′是连通的。不失一般性，我们假设 T′是一棵连通消去树。

假设 r′是 T′的根。现在我们将把 T′修改为匹配消去树。首先，我们将分析 T′中的路径，这些路径对应于 T中长度为偶数的根到叶路径。如果 T′在这条路径上未匹配，那么存在一个最靠近 r′的节点 u，使得 u 在G′中与 T′中的某个子节点 v 不连通。只有在以下情况下才有可能： u 在 T 中匹配到 u 的某个子节点 w，或者匹配到其父节点 w，且 T中的 w不属于 G′。因此，我们有 $\mathrm{dist}_{T'}(r',u)+\mathrm{depth}(T'_v)<\mathrm{depth}(T)$，这意味着我们可以将经过边 uv的任意路径长度增加 1。由于 T′是连通的，现在可以应用定理 3 证明中的变换，将 u 与 T'_v中的一个后代匹配。深度仍然至多为 d，因为这种变换使深度最多增加 1。实际上，在树的这个分支中，通过向上提取一个后代而增加的深度，会由以下事实抵消：未匹配顶点是通过删除该分支中的一个顶点而引入的。

我们可以迭代地应用上述构造，使 T′成为一棵匹配消去树，同时保持 T′连通。然而，应用上述变换可能会在 T′中引入一个节点 u，该节点没有与子节点 v 匹配，因为 v 的父节点 w 在树中被上移，以便与其他某个顶点匹配。这样的 u 也满足不等式 $\mathrm{dist}_{T'}(r',u)+\mathrm{depth}(T'_v)<\mathrm{depth}(T)$。为什么？树中任何一条先前同时经过 u 和 v 的路径，都必须经过 w。但是，根据上一段的论证， w 被上移这一事实意味着这些路径的长度严格小于 depth (T) 。而将 w 移到路径中更靠前的位置，不会增加路径长度。

对于 T中长度为奇数的根到叶路径，顶点 uvw上可能存在一个匹配的 P_3，使得 v 不在 G′中。同样，在这种情况下，根据相同的论证，这些变换会使深度最多增加到 d+1 （在这种情况下，我们可能需要上移两个不同的顶点，以便将u 和 w匹配。）。当 d 为偶数时，将这类路径的长度增加 1 不会增加树的深度。当 d 为奇数时，将这类路径的长度增加 1 会使深度最多增加 1。◀

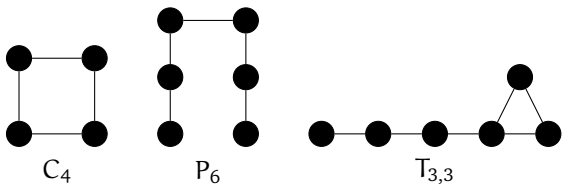

图 2 mtd(的禁用图。) ⩽ 3

定理 3 意味着所有具有 td(G) = 3 的图 G 也具有 mtd(G) ⩽ 4。容易看出，图 2 中给出的 C_4、 P_6 和 $T_{3,3}$ 具有 td(.) = 3 和 mtd(.) = 4。根据定理 4，这些图的某些超图也可能具有 mtd(.) = 3。然而，在下面的定理中，我们将证明这不可能发生，并且具有 td(G) = mtd(G) = 3 的图 G 恰好是那些以这些图作为诱导子图加以禁用的图。

▶ **定理 5.** 设 G是一个图，使得 td(G) = 3。那么，当且仅当 G 是$(C_4, P_6, T_{3,3})$-自由的时，mtd(G) = 3 。

证明。 （“若”方向的证明）设 G 是一个连通图，且 $td(G) = 3$ 满足G 是 $(C_4, P_6, T_{3,3})$-自由的。我们将从其消去树构造一个深度为 3 的 G 匹配消去树。

考虑 T 的一个深度为 3 的消去树 G，其根为某个顶点 r。令 $\{v_1, v_2, \ldots\ldots v_k\}$ 为 r 在 T 中的子节点。注意，对于任意 v_i 的子节点 c（对任意 i），如果 c 在 G 中与 v_i 不相邻，那么它必定在 G 中与 r 相邻（否则，G 就不连通）。我们将所有这类在 G中与 v_i 不相邻的 c，在消去树 T中改为 r 的子节点，而非 v_i的子节点。注意，这既不会违反消去树的任何性质，也不会增加消去树的深度。此外，完成这一修改后，我们还可以进一步假设，v_i 在 T 中的所有子节点都与 v_i 在 G中相邻。

现在，如果对所有 $i \in [k]$都有 $rv_i \in E(G)$，那么 T 本身就是一棵匹配消去树，问题得证。假设存在一个 i 使得 $rv_i \notin E(G)$。由于 G 连通，从 r 到 v_i 必定存在一条路径。由于 T 的深度为 3，这条路径必定是一条 P_3，因此 r 和 v_i 至少有一个公共邻点。r 和 v_i 的任意公共邻点 u 都必定是 v_i 在 T中的子节点。此外，如果 r 和 v_i 有两个公共邻点，例如 u_1 和 u_2，那么 ru_1vu_2r 是一个诱导的 C_4。因此，r 和 v_i 必定恰好有一个公共邻点。现在我们考虑以下情况：

- r恰有一个子节点 v_1。

  设 c_1、c_2、……、c_k 是 v_1的子节点。回想一下，$rv_1 \notin E(G)$，并且它们在 G中恰有一个公共邻点，记为 c_1。接着，我们按如下方式构造匹配消去树 T'：T' 的根是 c_1，顶点 r 和 v_1 成为 c_1的子节点，T'中v_1 的子节点是 $\{c_2, 、\ldots\ldots、 c_k\}$。树 T' 是 G的一棵深度为 3 的合法匹配消去树。

- r 有多个子节点，记为 $\{v_1, 、\ldots\ldots、 v_\ell\}$，其中 $\ell > 1$满足某个条件。

  我们将其分为两种情况。

  - （r恰好不与它的一个子节点相邻，记为 v_1）

    顶点 v_1 不是 T 中的叶节点，因为 G 是连通的。如果 v_1 恰有一个子节点，那么，由于它必定是 r 和 v_1的公共邻点 c，我们交换 c 和 v_1，这不会增加 T的深度。否则，假设 v_1 至少有两个子节点，即$\{c_1, c_2, 、\ldots\ldots、 c_k\}$（其中与 r的公共邻点是 c_1）。现在我们论证，r的所有其他子节点 $\{v_2, 、\ldots\ldots、 v_l\}$必定是 T中的叶节点。否则，假设 v_2 有一个子节点 c_3，那么我们会遇到一个诱导的 P_6，即 $(c_2, v_1, c_1, r, v_2, c_3)$（如果 $rc_3 \notin E(G)$），或一个诱导的T_3，3（如果 $rc_3 \in E(G)$），这就产生了矛盾。因此，$\{v_2, 、\ldots\ldots、 v_l\}$都是叶顶点。现在，我们可以将 T改为以 c_1而非 r为根的 T'。T'中 c_1的子节点恰为 r和 v_1；r的子节点恰为所有叶节点 $\{v_2, 、\ldots\ldots、 v_l\}$；而 v_1的子节点则完全相同，只是去掉了 c_1，也就是 $\{c_2, 、\ldots\ldots、 c_k\}$。不难看出，$T'$是一棵深度为 3 的匹配消去树。

  - （r与其至少两个子节点不相邻，例如 v_1 和 v_2，也可能还有更多）

    由于 G的连通性，顶点 v_1、v_2 不能是 T中的叶节点。如果 v_1 恰有一个子节点 c，那么 c 必定是 r 和 v_1的公共邻点。我们在T中交换 v_1 和 c。这样，要么归入前一种情况，要么我们可以假设 v_1 有多个子节点。设 c_1、c_2 是 v_1的两个子节点，其中 c_1 是 r 和 v_1的公共邻点。于是我们得到一个诱导 P_6，即 $(c_2, v_1, c_1, r, c, v_2)$，其中 c 是 r 和 v_2的公共邻点。（这样的 c 必定存在，因为（r、$v_2 \notin E(G)$）。

（“仅当”方向的证明）假设 G 是一个满足$mtd(G) = 3$的连通图，且 $td(G) = 3$ 。我们将证明 G 不能包含 C_4、 P_6或$T_{3、3}$作为诱导子图。我们将证明， G的任意连通诱导子图 G' 的匹配树深度至多为 3。

设 T 是 G 的深度为 3 的匹配消去树。设 r 是 T 的根。如果 r 不在G'中，那么由于 G' 连通， G' 中的所有顶点必定来自 T 中 r 的同一个子树。在这种情况下，该子树见证了 G'的一棵深度至多为 2 的匹配消去树。如果r 在 G'中，那么可以通过从 T 中删除一些第 1 层和第 2 层顶点来得到 G'。我们将从 T 为 G' 构造一棵匹配消去树。如果某个第 2 层顶点不在 G'中，我们也只需将其从 T 中删除。如果第 1 层顶点 v 不在 G'中，那么对于每个同时也是 T 中 v 的子节点的 $u \in V(G')$，由于 G' 连通， G' 中必定存在一条边 ru。因此，我们将 u 设为 G'的匹配消去树中 r 的子节点。◀

根据定理 1 和定理 3，我们可以得出结论：每个 G（其中 $td(G) = 3$ ）都有一种用于对从 G 到 m-边稀疏宿主图的同态进行计数的 $\widetilde{O}(m^2)$ 算法。因此，在 G 中存在作为诱导子图的 C_4、 P_6 或 $T_{3、}$ 3，并不会影响算法 1 的运行时间。如果我们考虑满足 $td(G) = 4$ 的模式 G ，那么有 P_8 这样的例子，其中 $td(P_8) = 4$ 且 $mtd(P_8) = 5$。因此，我们只能得到一种用于对从 P_8出发的同态进行计数的$\widetilde{O}(m^3)$ 算法。若能证明适用于更高树深度（例如 4 ）的、类似定理 5 的定理，将会很有意思。不过，我们甚至还不知道树深度为 4 [12] 时禁用诱导子图的确切集合，因此这似乎很困难。

4 匹配树宽

本节介绍如何利用多项式空间，在稀疏宿主图上高效地计算同态和子图数量的算法。本节的核心定理如下。

▶ **定理 6**。 设 G是一个图，其$mtw(G) = t$，则给定一个 m-边宿主图 H 作为输入，我们可以在 $\widetilde{O}(m^{\lceil (t+1)/2 \rceil})$时间内构造一个计算从G到 H 的同态多项式的算术电路。

证明。 设 T是 G的一个匹配树分解。我们任意固定 G的边和顶点到 T 中各袋的一个指派，使得每个顶点和边都恰好被指派给一个包含它的袋。设 B 是 T中的一个袋。我们将 M 中 B 的匹配匹配视为边的序列 $(a_1b_1, \ldots, a_kb_k)$，并任意为它们排序。给定宿主图 H中的边 u_1v_1， ...， u_kv_k， 使得 $\sigma(a_i) = u_i$ 且 $\sigma(b_i) = v_i$ 是 B的顶点上的有效的部分同态，我们定义以下单项式：

$$\mathsf{EdgeMon}(B, u_1v_1, \ldots, u_kv_k) = \prod_e x_{\sigma(e)}$$

$$\mathsf{VertexMon}(B, u_1v_1, \ldots, u_kv_k) = \prod_v y_{\sigma(v)}$$

其中 e和 v遍历分配给 B的所有边和顶点。我们还将 $\mathsf{Mon}(B, u_1v_1, \ldots, u_kv_k)$ 定义为 $\mathsf{EdgeMon}(B, u_1v_1, \ldots, u_kv_k)$ 与 $\mathsf{VertexMon}(B, u_1v_1, \ldots, u_kv_k)$的乘积。

设 B和 B' 是 T 中的袋，且 B' 是 B的父节点。我们任意排列这些顶点，得到 $B \cap B'$ 中顶点的序列 $X(B \cap B')$ （我们可以假设这些顶点的标签从 $[|V(G)|]$ 开始，并按递增顺序排列）。我们定义 $\mathsf{MapGate}(B, B', u_1, 。 \ldots,$

$u_k)$ 其中 $X(B \cap B') = (a_1, \ldots, a_k)$ 是与部分同态 $\sigma(a_i) = u_i, 1 \leqslant i \leqslant k$ 对应的命名门。

算法 2 构造所需的电路。注意，第 9、20、36 行循环中的所有操作都在 $O(\log n)$ 时间内运行。循环本身执行 $O(m^{\lceil (t+1)/2 \rceil})$ 次迭代，因为任意一个匹配匹配在 $t+1$ 个顶点上至多包含 $\lceil (t+1)/2 \rceil$ 条边。第 2、7和 18行的循环执行 $O(1)$ 次迭代，因为模式图 G是固定的。注意，我们遍历的是对应于边的对 uv ，而不是边 $\{u, v\}$ ，因为顺序决定了同态。

当 MapGate(.) gate 的哈希表查找失败时，如果该门出现在左侧，我们就以初始值将其添加到该表中；如果出现在右侧，则将其替换为 0。现在，我们利用解析树证明该电路的正确性。注意，电路中唯一的+ 门是 MapGate(. . .)。在此证明中，我们可以将 r 视为 MapGate(R, ϕ)，其中 R 是 T 中以空袋为父节点的根袋。因此，最终多项式的单项式对应于构建解析树时在这些门处所作的选择。对于 T中父节点为 B'、包含顶点 $x_1, \ldots, x_k$ 的任意袋 B，门 MapGate$(B, B', x_1, \ldots, x_k)$ 的输入对应于在 B的顶点上定义的有效部分同态，并且对所有 $a_i \in X(B \cap B')$，这些同态都将 a_i 映射到 x_i 。我们将这些门与袋 B对应起来。现在，给定从 G 到 H的同态 σ ，我们通过在每个这样的门处选取 σ 在该袋顶点上的限制，来构造它的解析树。我们需要证明，这种选择与每个门处的像 $x_1, \ldots, x_k$ 一致。事实上，在根门处这显然成立（列表为空）。对于任意的 $a \in V(G)$，考虑 a 首次出现的最顶层袋 B。在 B处选取的部分同态中，我们可以自由地将顶点 a 的像设为 σ中的对应像。根据构造，向下移动到对应于包含 a 的袋的子门时，这一选择会继续传递（参见第 15、30 行和第 42 行）。此外，如果 B 的某个子袋不包含顶点 a，那么顶点 a 也不会出现在该子树中。容易看出，这棵解析树计算出的单项式是正确的。由于该电路是单调的，这就证明了多项式中包含所有对应于同态的单项式。对于反方向，我们需要说明多项式中只包含对应于同态的单项式。事实上，任意解析树都对应于在每个门处依次选择部分同态。我们将论证这些部分同态必定彼此一致，因此可以合并为一个有效同态。这是因为 T 是一个树分解，因此任意 $a \in V(G)$ 都必须出现在 T的一棵（连通）子树中。电路的构造保证，一旦在解析树中确定了顶点 a 的像，它就会正确地传递到所有定义域包含 a 的部分同态中。此外，由于 G 中的每个顶点和边都唯一地分配给一个袋，它们的像在单项式中恰好出现一次。至此，证明完成。

◀

▶ 备注30。此前一位匿名审稿人对本文早期稿件的评论指出，图参数广义超树宽度（记为 ghw）可能会产生相似的运行时间。事实上，我们已验证，存在 $\tilde{O}(m^{ghw})$-时间算法，且 $ghw \leqslant \lceil (mtw+1)/2 \rceil$与最坏情况下的运行时间一致。据我们所知，尚未有人在针对小模式的快速算法这一背景下分析参数 ghw。它主要用于为树宽高而 ghw 低的超图设计高效算法。我们认为，分析 mtw 也很有用，因为它有时比 ghw 更精细。也就是说，$ghw = k$ 可能包含来自 $mtw = 2k - 2$ 两个类别的图，以及

算法 2 计算同态$_{G}[H]$

1: 令 T 为 G的近完美树分解。 **2:** 对 T 中的每个袋 B执行 **3:** 对 T 中 B的每个子节点 B' 执行 **4:** 初始化空哈希表 $T_{(B,B')}$。 **5:** 结束循环 **6:** 结束循环 **7:** 对 T 中的每个非根叶袋 B执行 **8:** 令 T中 B 的父节点为 B'。 **9:** 对 $(u_1v_1, \ldots\ldots u_kv_k) \in E(H)^k$ 执行 **10:** 令 $\sigma(a_i) = u_i$, $\sigma(b_i) = v_i$为 σ。 **11:** 如果 σ 不是从 G到 H 的有效部分同态，则 **12:** 跳过本次迭代 **13:** 结束条件 **14:** 令 σ中 $X(B \cap B')$ 内顶点的像为 $x_1, \ldots, x_{k'}$。 **15:** **MapGate**$(B, B', x_1, \ldots, x_{k'}) +=$ **Mon**$(B, u_1v_1, \ldots, u_kv_k)$**16:** 结束循环 **17:** 结束循环 **18:** 按自底向上顺序，对 T 中的每个非根袋 B执行 **19:** 令 $M = (a_1b_1, \ldots, a_kb_k)$。 **20:** 对 $(u_1v_1, \ldots\ldots u_kv_k) \in E(H)^k$ 执行 **21:** 令 $\sigma(a_i) = u_i$, $\sigma(b_i) = v_i$为 σ。 **22:** 如果 σ 不是从 G到 H 的有效部分同态，则 **23:** 跳过本次迭代 **24:** 结束条件 **25:** 令 T中 B 的父节点为 B'。 **26:** 令 T 中 B 的子节点为 $B_1, \ldots, B_s$。 **27:** 令 σ 中 $X(B \cap B')$ 内顶点的像为 $x_1, \ldots, x_{k'}$。 **28:** 令 σ中 $X(B \cap B_i)$ 内顶点的像为 $w_{i1}, \ldots, w_{ik_i}$。 **29:** **MapGate**$(B, B', x_1, \ldots, x_{k'}) +=$**30:** **Mon**$(B, u_1v_1, \ldots, u_kv_k) \prod_{i=1}^{s}$ **MapGate**$(B, B_i, w_{i1}, \ldots, w_{ik_i})$**31:** 结束循环 **32:** 结束循环 **33:** 令 T 中的根为 B**34:** 令 T 中 B 的子节点为 $B_1, \ldots, B_s$。 **35:** $r \leftarrow 0$**36:** 对 $(u_1v_1, \ldots\ldots u_kv_k) \in E(H)^k$ 执行 **37:** 令 $\sigma(a_i) = u_i$, $\sigma(b_i) = v_i$为 σ。 **38:** 如果 σ 不是从 G到 H 的有效部分同态，则 **39:** 跳过本次迭代 **40:** 结束条件 **41:** 令 σ中 $X(B \cap B_i)$ 内顶点的像为 $w_{i1}, \ldots, w_{ik_i}$。 **42:** $r +=$ **Mon**$(B, u_1v_1, \ldots, u_kv_k) \prod_{i=1}^{s}$ **MapGate**$(B_i, B, w_{i1}, \ldots, w_{ik_i})$**43:** 结束循环 **44:** 返回 r

$\mathrm{mtw} = 2k - 1$。

首先，我们对 mtw 作出一些基本观察。

正如我们将树宽与树深度联系起来一样，也可以将它与匹配树深度联系起来。

▶ **命题 31**。 对于任意图 G，都有 $\mathrm{mtw}(G) \leqslant \mathrm{mtd}(G) + 1$。

证明。 令 E 为 G的最优匹配消去树。我们按如下方式从 T 构造 G的匹配树分解 E：对于 E 中从最左路径到最右路径的每一条从根到叶节点的路径，构造包含这些路径上所有顶点的袋。然后，通过在袋 B （对应于通往叶节点 u的路径）与 B'（对应于通往叶节点 v的路径）之间添加一条边，将这些袋连接成一条路径，当且仅当 v 是 E 中从 u 左到右排列叶节点时的下一个叶节点。 ◀

现在，我们证明匹配树宽不会比树宽大很多。

▶ **定理 7.** 对于任意图 G，都有 $\mathrm{mtw}(G) \leqslant 2 \cdot \mathrm{tw}(G) + 1$。

证明。 设 T 是图 G的一个树分解，其宽度为 k。我们将描述一种将 T 转换为匹配树分解的过程。构造采用自顶向下的方式。最终的树将具有这样的性质：每个非叶袋中的顶点在 G中诱导出一个完美匹配。

令 B_r 为 T中的根袋。以下过程将应用于 B_r。

1. 在 M中找到一个极大匹配 B_r。

2. 对于每个 $v \in B_r \setminus M$ 使得 $N_G(v) \subseteq B_r$。注意，由于 M 也是一个极大匹配 $N_G(v) \subseteq V(M)$ 。我们从 v 中删除 B_r ，并添加一个叶袋 B_v 作为 B_r的子节点。袋 B_v 将包含 v 以及 M中的顶点。由于 v 与M中的至少一个顶点相邻，因此该袋是匹配的。

3. 对于每个 $v \in B_r \setminus M$ 满足 $N_G(v) \nsubseteq B_r$。我们可以找到一个 u 满足 $u \notin B_r$ 且 $u \in N_G(v)$。我们选择这样一个 u ，它位于袋 B中，且与 B_r在 T中的距离最小。我们将 u加入 B_r 以及从 B到 B_r在 T中的路径上的所有袋。然后我们修改$M = M \cup \{uv\}$。

注意，步骤 2 和步骤 3 中的每次迭代都会使 B_r 中的未匹配顶点数减少 1，并且只会向 T 添加已匹配的叶袋。此外，如果 B_r 最初有 x 个顶点，那么经过此过程后，它最多有 $2x$ 个顶点，因为对于原来的每个 x 个顶点，我们至多添加一个与之对应的顶点。因此，修改后的 B_r 的大小至多为 $2k + 2$ ，且由 B_r 中的顶点诱导的图具有完美匹配。

现在，考虑任意一个袋 B，使得它的所有祖先都是具有完美匹配的袋。令 B_p 为 B在 T 中的父节点，并令 M 为 B_p中顶点上的完美匹配。现在，我们对 B应用以下过程。

1. 对于每个 $v \in B \cap B_p$，令 u 为 v 在 M中的配对项。如果 u 在 B中，那么我们也将 v 与 u 在 B 中配对。否则，我们将 u 加入 B，并将 v 与 u 在 B中配对。这不会违反树分解的任何性质。注意，如果在根袋的过程第 3 步中将 v 加入 B，那么它的配对项 u 也必须在 B中。

2。 对于 $v \in B \setminus B_p$，我们应用根袋流程中所述的步骤 2 和 3。这些步骤只会修改以 T 为根的 B子树中的袋，并且只会向 T 叶节点

袋添加匹配。此外，在此步骤结束时，袋 B 中的顶点所诱导的图具有完美匹配。

注意，袋 B 的大小最多可以从其原始大小增加一倍，因为对于袋中原有的每个顶点，我们至多添加一个顶点。对于所有新添加的叶袋，这一点同样成立。因此，我们构造出了一个宽度至多为 $2k+1$ 的匹配树分解。◀

现在，我们展示一个具体的图族，该图族中树宽与匹配树宽之间的关系接近最坏情况。

▶ **命题 8**。$\mathrm{mtw}(K_{n,n}) = 2n-2$ 对于所有 $n > 1$。

证明。令 T 仅为一条边，其顶点集为 $\{K_{n,n} \setminus \{a\}, K_{n,n} \setminus \{a'\}\}$，其中 a 和 a' 属于同一部分。容易验证，T 是K_n的匹配树分解，n。因此 $mtw(K_{n,n}) \leqslant 2n-2$。

令 T 为 $K_{n,n}$ 的任意匹配树分解。以一个叶袋为根，设其为 X，并令 X_1 为 X 的唯一子节点。如果 $X \subseteq X_1$，那么我们可以从 T 中删除 X。因此，存在 $K_{n,n}$ 的一个顶点 $u \in X$，使得 $u \notin X_1$。所以 $N(u) \subseteq X$。不失一般性，假设 $u \in A$。于是，我们得到 $B \subseteq X$。现在，为了匹配 B 中的 n 个顶点，我们至少需要从袋 X 中取出 A 的 $n-1$ 个顶点。因此 $mtw(K_{n,n}) \geqslant 2n-2$。◀

容易看出 $\mathrm{tw}(K_{n,n}) = n$。因此，其匹配树宽仅比最坏情况下的 $2n+1$ 小 3。

现在我们使用算法 2 和定义 26来统计稀疏宿主图中环里的路径。我们不再分析 $\mathrm{Spasm}(P_{10})$ 中的所有图，而是完整刻画树宽为 2 的图的匹配树宽。这将简化证明许多小模式图的算法上界所需的分类讨论。

4.1 部分 2-树的匹配树宽

本节研究部分 2-树的匹配树宽。总结见表 2。根据定理 7，我们有 $\mathrm{mtw}(G) \leqslant 5$ 当 $\mathrm{tw}(G) = 2$，由此得到最后一行。图 Y（见图 4）满足 $\mathrm{tw}(Y) = 2$ 且 $\mathrm{mtw}(Y) = 5$。但是 Y 也是 Z 的一个诱导子图（见图 5），满足 $\mathrm{tw}(Z) = 2$ 且 $\mathrm{mtw}(Z) = 4$。因此，这种情况下不适用禁诱导 minor 子图刻画。现在我们证明其余两种刻画。

mtw $\leqslant$	禁用诱导子式
2	C_5
3	X
4	不适用。
5	无。

表2 部分 2-树的匹配树宽。

▶ **定理 10.** 对于任意部分 *2*-树 G，图 G是 C_5-诱导子式自由的，当且仅当$\mathrm{mtw}(G) = 2$。

证明。（“若”方向的证明）反设存在部分 2-树 G，使得 $\mathrm{mtw}(G) = 2$，且 G含有一个 C_5-诱导子式。令 T为 G的宽度为 2 的匹配树分解。由于 C_5 是一个诱导子式，我们可以通过删除顶点和收缩边从 G得到 C_5。对于所有被删除的 v，从 T的所有袋中删除 v。类似地，对于所有被收缩的边 uv，在 T的所有袋中，将 u 和 v 一致地替换为 u 或 v之一。由此得到 C_5的一个（不一定是匹配的）树分解 T'。不失一般性，假设 T'是一个约化树分解。由于 T的宽度至多为 2，T'中的任意一个袋至多包含 3 个顶点。如果 T' 中的一个袋包含 3 个顶点，那么它们已经构成一个 P_3，因为它在 T中必定也是如此。T'中不可能有只含一个顶点的袋，因为它是约化的，且C_5 是连通的。我们断言，T' 中大小为 2 的袋必须包含某些 u 和 v，使得$uv \in E(C_5)$。

▷ 命题 32. 设 u 和 v 是 C_5 中两个不相邻的顶点。那么，树分解 T'不可能包含袋 $\{u, v\}$。

证明。 反设 T'包含袋 $B = \{1, 3\}$。其他情况均对称。以该袋为根，将树 T'定根。现在我们分析各种情况。

1. (B 只有一个子节点）由于 1 和 3 都与其他顶点相邻，B的子节点必须同时包含 1 和 3，这与 T'是约化的这一事实矛盾。
2. (B有多个子节点）我们将此情况细分为若干子情况。
   a. (边 15 和 34 分别由 B的不同子树（例如 T_1 和 T_2）覆盖）由于$45 \in E(C_5)$，这条边必须被覆盖。覆盖 45 的袋必须有一条路径，沿着该路径可从 5 到达 T_1 中覆盖 15 的袋，这是由 5 的连通性所致。类似地，该袋必须有一条路径，沿着该路径可从 4 到达 T_2 中覆盖 34 的袋，这是由 4 的连通性所致。但是，这是不可能的，因为 B只包含 1 和 3。
   b. （边 15 和 34 被 B 的同一子树覆盖。）此情况可进一步分为多个子情况。
      i. （袋 B，以及包含 15 和 34 的袋在 T'中位于同一路径上。）假设该路径从 B 到包含 3 和 4 的袋，并经过包含 1 和 5 的袋 B'。另一种情况类似。则袋 B'必须包含 3，以保持 3 的连通性。现在有 $B' \supset B$，矛盾。
      ii. （包含 15 和 34 的袋有一个公共祖先，且该祖先是 B 的真后代。）令 B' 为这个公共祖先。袋 B'必须包含 3，因为它位于从 B 到包含 3 和 4 的袋的路径上。此外，袋 B' 必须包含 1，因为它位于从 B 到包含 1 和 5 的袋的路径上。但是，这样一来 $B' \supseteq B$，矛盾。

◀

这就完成了“若”方向的证明。

（“仅当”方向的证明）我们用反证法。假设 G是顶点数最少的反例。

▷ 断言 33. 图 G是 2-连通的。

证明。假设 G 有一个割点 v。删除 v后，我们得到较小图 G_1，……，G_m，其中某个 $m > 1$。G 中的任何环也是某个 i中的环，且该图为 G_i。由于 G 是极小的，每个 G_i 都有一个宽度至多为 2 的匹配树分解，记作 T_i。对于所有 i，令 B_i 为 T_i 中的一个袋，且该袋包含 v。对所有 $1 < i \leqslant m$，在 B_1 与 B_i 之间添加边。这样便得到 G的一个宽度为 2 的匹配树分解，矛盾。◀

我们取一个 2-树 G'，它是 G 的超图，并且与 G 具有相同的顶点集。

▷ 断言 34。设 uv 是 G' 中但不在 G 中的一条边。那么 u 和 v 在 G 中有一个公共邻点。

证明。由于 G 是 2-连通的，u 和 v 之间在 G 中存在两条内部不相交路径 $P = uu_1 \cdots u_k v$ 和 $P' = uv_1 \cdots v_\ell v$。我们不妨假设 P 和 P' 是 G 中的诱导路径。如果 k 或 ℓ 为 1，那么 u 和 v 有一个公共邻点。因此我们假设 k 和 ℓ 至少为 2。现在 PP' 是一个长度至少为 6 的环，并且它必定有一条弦（否则，这就是 G 中一个 C_5-诱导图子式）。因此，存在某个 i 和 j，使得 u_i 与 v_j 相邻。P、P' 上的边连同这条弦构成 G' 中的一个 K_4-图子式，矛盾。◀

令 T 为 G' 的标准树分解。这是 G 的匹配树分解。我们证明它也是 G 的匹配树分解。只需证明，在 G' 中构成三角形的集合 $\{u, v, w\}$ 会导出 G 中的一个 P_3 或一个三角形。反设 v（例如）并非同时与 u 和 w 相邻。由断言 34 可得顶点 x，它是 u 和 v 在 G 中的公共邻点；以及顶点 y，它是 v 和 w 在 G 中的公共邻点。由于 G' 不含 K_4 图子式，因此有 $x \neq y$、$wx \notin E(G)$、$xy \notin E(G)$和 $uy \notin E(G)$。若 $uw \in E(G)$，则 $uwyvxu$ 是 G 中的一个诱导 C_5，矛盾。因此 $uw \notin E(G)$，并且由断言 34 可得顶点 z，它是 u 和 w 在 G 中的公共邻点。同样，由于 G' 不含 K_4 图子式，因此有 $z \neq x$、$z \neq y$、$zx \notin E(G)$、$zv \notin E(G)$和 $zy \notin E(G)$。于是 $uzwyvxu$ 是 G 中的一个诱导 C_6，矛盾。证明完毕。◀

现在我们证明图 X（见图 1）是一个匹配树宽大于 3 的部分 2-树。

► **引理 35**。 $mtw(X) \geqslant 4$

证明。考虑图 X 的一个约化匹配树分解 T。反设其宽度严格小于 4。考虑树 T 中的一个叶袋 B_l。袋 B_l 必须包含 u_0、u_2 或 u_4，因为 X 中的所有边都与这些顶点之一关联。所有情况都是对称的，因此不失一般性，假设 B_l 是 $\{u_0, u_1\}$的超集。令树 T 以 B_l 为根。由于 u_0 和 u_1 都不是悬挂的，且 T 是约化的，因此 B_l 必须至少包含另一个顶点。令 B_p 为与 B_l相邻的袋。我们声称 $\{u_0, u_2\}$ 或 $\{u_0, u_4\}$ 包含在 B_p中。若 $u_0 \notin B_p$，则 $\{u_1, u_1', u_5, u_5'\} \subseteq B_l$，因此 $|B_l| \geqslant 5$。假设 $u_1 \in B_p$。由于T 是约化的，这意味着 B_l 中有某个不同于 u_0 和 u_1 的顶点，但该顶点不在 B_p中。若此顶点是 u_2 或 u_4，则 $|B_l| \geqslant 5$。若它是其他顶点之一，则 u_2 或 u_4 在B_l 中，且 B_p。若 $u_1 \notin B_p$，则 B_l 必须包含 u_2，因此 B_p 也必须如此。现在，不失一般性地假设 u_0 和 u_2 都在袋 B_p中。若 B_p 还包含 u_4，那么结论成立，因为匹配袋的大小至少为 5。

设 B_c 是 T 中的 B_p 的一个后代袋，且 B_c 是 T 中包含 u_4的子树的根。同样，如果 $\{u_0, u_2, u_4\} \subseteq B_c$，那么 $|B_c| \geqslant 5$，问题便得以解决。因此不失一般性地假设

$u_2 \notin B_c$。如果 $u_0 \notin B_c$，那么 $B_c \supseteq \{u_5, u_5', u_3, u_3'\}$，问题便得以解决。这是因为这些顶点是 u_4 与 u_0 的公共邻点，或是 u_4 与 u_2 的公共邻点；而 B_c 是 T 中包含 u_4 的子树的根袋。因此 u_0 也在 B_c 中。但是，B_c 还包含 u_3 和 u_3'，而 B_c 必须至少再包含一个顶点，才能与 u_0 相匹配。◀

现在我们证明，禁止 X-诱导图子式（对于部分 2-树）恰好得到具有 mtw(的图类) $\leqslant 3$。

▶ **定义 36.** 图 G 的一个最小弦完成 G 称为特殊的，如果 G 中不存在大小为3 的˜独立集在 G 中诱导出一个团。我们用 *smcc* 表示特殊最小弦完成。

图的树宽与其 smcc 的树宽相同（如果存在 smcc）。

▶ **定理 11.** 对于任意部分 *2*-树 G，图 G 是 X-诱导图子式自由的，当且仅当$\mathrm{mtw}(G) \leqslant 3$。

证明。（“若”方向的证明）设 G 是一个具有宽度 3 的匹配树分解 U 的图。反设 G 有一个 X-诱导图子式。于是，我们也可以通过收缩 G 的一个诱导子图 X' 中的若干边得到 X。设 X' 的树分解 T' 是从 U 得到的，即从所有袋中删去集合 S 中的顶点。由于 X 可以通过收缩 X' 的若干边得到，它将包含三个顶点，例如u_0、u_2、u_4，它们构成一个独立集，并且存在从 u_i 到 u_{i+2} [4] 的内部顶点不相交路径 Q_{i+1}、Q'_{i+1}。我们定义映射 $f : V(X') \mapsto V(X)$，使得对于 $i \in \{0, 2, 4\}$，f，它将路径 Q_{i+1} 中的所有内部顶点映射到 u_{i+1}，并将 Q'_{i+1} 映射到 u'_{i+1}。映射 f 对应于我们用来从 X' 得到 X 的边收缩。设 X 的树分解 T 是通过对所有袋中的所有顶点应用 f，从 T' 得到的。我们可以使用标准技术从 T 中删除大小为 0 和 1 的袋。

现在我们修改 T，从而为 X 得到一个宽度 3 的匹配树分解，由此推出矛盾。T中的每个袋 B 在 T'中都有一个对应的袋 P。注意，如果 X'中的 uv 是一条边，那么只要满足 $f(u) \neq f(v)$，X中的 $f(u)f(v)$ 就是一条边。如果 B的大小为 4，那么由于 U 是一个匹配树分解，袋 P 也必须是匹配的。因此，如果 T'中的所有袋大小均为 4，问题便已解决。否则，T中存在一个大小小于 3 的袋 B。我们以 B为根，将树 T 定根，并以自顶向下的方式修改 T。为了证明这一过程的正确性，我们需要以下命题。

▷ 命题 37. 设 T 中的某个袋 B' 大小为 3，则 B' 中的顶点无法在 X中构成一个大小为 3 的独立集。

证明。 如果对应的袋 P 到 B' 在 U 的大小为 4，那么由于 U 是一个匹配树分解，袋 P 是匹配的。因此 B' 是通过删除一个顶点或收缩一条边得到的。这两种操作都会使袋中至少保留一条边。

如果对应的袋 P 大小为 3，那么由于 U 是一个匹配树组合，由 B' 诱导的子图以 P_3 为子图。◀

因此，我们可以假设如下：

[4] 本证明中的所有指标 i 对于 u_i 和 u'_i 均按模 6 计算。

26 在稀疏图中查找并计数模式

1. 每个大小为 4 的袋都被匹配。 T of size 4 is matched.

2. 中的每个袋 T 大小为 3，且至少包含一条边。

3. 中不存在 T 大小为 0 或 1 的袋。

现在，我们从根开始处理一个袋 B，并假设 B 的所有祖先都已匹配。我们将该过程分为两种情况。

(|B| = 3) 令 $B = \{u, v, w\}$, uv 是一条边，且 w 与 u 或 v 不相邻。如果 w 出现在 B 的父节点中，那么它会与某个 w' 匹配。我们也将 w' 添加到 B 。如果 w 未出现在父节点中，那么我们在 T 中选择 B 的一个最近后代 B′，使其包含 w 的一个邻点 w'。从 B 到 B′ 的路径上的每个袋都包含 w。我们将 w' 添加到这条路径上的每个袋中。现在，袋 B 已匹配。此外，我们修改过的其他每个袋（位于从 B 到 B′ 的路径上）现在要么大小为 3 且至少包含一条边，要么大小为 4 且已匹配。因此，所有性质都得以保持。

(|B| = 2) $B = \{u, v\}$和 uv 不是一条边。如果 u 出现在 B在 T中的父节点的袋中，那么它会在那里与某个 u' 匹配。我们将 u' 加入 B。否则，我们选择 T 中 B 的一个最近后代 B′，其中包含 u 的一个邻点 u' 。我们将 u' 加入从 B到 B′的路径上的所有袋中。B现在是一个大小为 3 且至少包含一条边的袋。此外，我们修改的其他每个袋现在要么大小为 3 且至少包含一条边，要么大小为 4 且已匹配。现在，如果 B已匹配，我们就完成了。否则，我们对 B应用前一种情况。

（“仅当”方向的证明）现在我们证明一系列引理，以证明该定理。

► **引理38**。 设 G 是一个图，且 $tw(G) \leqslant 2$， G是 G的一个 *smcc*。如果存在三条˜从顶点 u 到 G中的 v 的内部不相交路径，那么 uv 是 G中的一条边

证明。反设 u 与 v 不相邻。设 Q_1、Q_2、Q_3 是从 u 到 v 的三条内部不相交路径。设 Q_1'、Q_2' 分别是 u 到 v 在 $\tilde{G}[V(Q_1)]$ 和 $\tilde{G}[V(Q_2)]$ 中的最短路径。注意，它们的长度都至少为 3。由于 $\tilde{G}$ 是弦图，由 Q_1' 和 Q_2' 得到的环有一条弦，记为 xy。注意 $\{x, y\} \cap \{u, v\} = \emptyset$。不失一般性，设 x 在 Q_1' 中。因此， y 在 Q_2' 中。所以， Q_1'、Q_2' 和 Q_3˜ 在 G 中构成 K_4 的一个图子式。这与假设矛盾。◄

► **引理 39.** 设 G 是一个不含 X-诱导图子式的连通图，且树宽为2。那么，存在一个 G的 *smcc* G。

证明。 反设存在一个极小反例 G，其 树宽(G) ⩽为 2，且 X不含诱导图子式，但没有 s mcc。由于 G是极小的，它必定是双连通的；根据树宽为 2 的图的刻画，我们可以假设 G 是串并联图。我们将利用串并联图刻画来证明 G不存在。

设 s 和 t 分别是 G的源点和终端点。由于 G 是反例，因此它不是一条边。假设 (G, s, t) 是由 (G_1, s_1, t_1) 与 (G_2, s_2, t_2) 串联组合而成（t_1 与s_2重合）。由于 G 是一个极小反例，因此存在 G_1 的 smcc $\tilde{G_1}$，以及 G_2的 smcc $\tilde{G_2}$。令 $\tilde{G}$˜ ˜ ˜为通过将 t_1 与 s_2识别而由 G_1 和 G_2 得到的图。图 G是G的一个 smcc，这就产生了矛盾。因此，图 (G, s, t) 是由严格较小的图 (G_1, s_1, t_1) 和 (G_2, s_2, t_2)并联组合而成的。

▷ 断言 40。设 G是两个较小图 G_1 和 G_2的并联组合。此外，假设 G_1 是由两个或更多图 G_i'（$1 \leqslant i \leqslant m$）串联组合而成。那么，对于所有 $1 \leqslant i \leqslant m$，如果 G_i' 不是一条边，那么 $s_i't_i'$ 也不是一条边。

证明。假设存在 $1 \leqslant i \leqslant m$ 使得 G_i' 不是一条边，但 $s_i't_i'$ 是 G_i'中的一条边。因此，G_i' 的阶至少为 3。存在一个 smcc G_i，它属于从 G 中删除 G_i' 的所有内部顶点后得到的图（因为 G 是一个最小反例）。同样，存在一个 smcc $\tilde{G}_i'$，它属于 G_i'。定义 $\tilde{G}$为由 G_i 和 $\tilde{G}_i'$ 通过识别边 $s_i't_i'$而得到的图。图 $\tilde{G}$是 G的一个 smcc。◀

注意，G_1 和 G_2 中至少有一个不是边。现在，我们将证明分为两个穷尽的情形：

- （情形 1）G_1 或 G_2 是边。

  不失一般性，设 G_2 是边。因此，s_1t_1 不是G_1 中的边（否则 $G_1 = G$）。设 (G_1, s_1, t_1) 是 (G_3, s_3, t_3)与 (G_4, s_4, t_4)的并联组合。由于 $s_1t_1 \notin E(G_1)$，我们有 $s_3t_3 \notin E(G_3)$ 和 $s_4t_4 \notin E(G_4)$。因此，G_3 和 G_4 都至少有一个内部顶点。令 G_3' 为通过添加边 s_3t_3 从G_3 得到的图，令 G_4' 为通过添加边s_4t_4从 G_4 得到的图。由于 G 是一个最小反例，存在 G_3' 的 smcc G_3' 以及G_4'的 smcc G_4'。注意，G 是通过识别边 s_3t_3 和 s_4t_4由 G_3' 与 G_4' 得到的。因此，通过识别这些边由 $\tilde{G}_3'$ 与 $\tilde{G}_4'$ 得到的图 $\tilde{G}$ 是 G的一个 smcc。这就产生了矛盾。

  我们可以假设，(G_1, s_1, t_1)是较小图 (G_1', s_1', t_1')、(G_2', s_2', t_2')、……、(G_m', s_m', t_m')的串联组合，其中 $m \geqslant 2$满足：对于所有 $2 \leqslant i \leqslant m$，顶点$t_{i-1}'$ 与 s_i' 重合；对于所有 $1 \leqslant i \leqslant m$，$(G_j', s_j', t_j')$ 要么是一条边，要么是两个较小图的并联组合。

  注意 G 不是一个环，因为 G 是一个反例。因此，存在某个 i，使得图G_i' 不是一条边。因此，根据命题40，我们有$s_i't_i'$ 不是一条边。

  ▷ 断言41. 不存在满足以下条件的 $1 \leqslant j(\neq i) < k(\neq i) \leqslant m$：$s_j't_j'$、$s_k't_k'$ 不是边。

  证明. 假设对于某个 $1 < j(\neq i) < k(\neq i) \leqslant m$，$s_j't_j'$、$s_k't_k'$ 不是边。不失一般性，我们可以假设 $i < j$。因此，G_i' 是两个较小图的并联组合。于是，存在两条内部不相交的路径 Q_1、Q_1'（每条长度至少为 3），在 G_i' 中从 s_i' 到 t_i'。类似地，存在两条内部不相交的路径 Q_2、Q_2'（每条长度至少为 3），在 G_j' 中从 s_j' 到 t_j'。此外，存在两条内部不相交的路径 Q_3、Q_3'（每条长度至少为 3），在 G_k' 中从 s_k' 到 t_k'。另外，存在一条路径从 t_i' 到 s_j'，经过 G_{i+1}'、……、G_{j-1}'；还有一条路径从 t_j' 到 s_k'，经过 G_{j+1}'、……、G_{k-1}'。此外，我们有一条路径从 t_k' 到 s_i'，经过 G_{k+1}'、……、G_m'、s_mt_1'、G_1'、……、G_{i-1}'。这构成了 X 的一个细分。这是一个矛盾。因此，上述断言成立。◀

  ▷ 断言 42. 存在一个 G_i'，使得 $s_i't_i'$ 是补全中的一条边。

  证明。由于s_i' 与 t_i'不相邻，(G_i', s_i', t_i') 是两个较小图的并联组合（由断言40），设为 (G_3, s_3, t_3) 和 (G_4, s_4, t_4)。令 G'为由以下图得到的图

G通过收缩边 st。这是一个 X-诱导图子式自由图。G的极小性表明， G' 有一个 smcc，设为 $\tilde{G}'$。由引理38， s'_i 在 $\tilde{G}'$中与 t'_i 相邻。注意，由 G'_i 在 $\tilde{G}'$ 中的顶点所诱导的图是 G'_i 的一个 smcc，其中 $s'_it'_i$ 是一条边。

◀

根据断言41，以下情形现已穷尽。

- （情形1a）对于所有 $1 \leqslant j(\neq i) \leqslant m$， $s'_it'_j$ 都是边。

  ~根据上述断言，存在一个包含 G'_i 的 smcc G'_i，其中 $s'_it'_i$ 是一条边。我们按如下方式构造一个包含 G 的 smcc ~：它包含所有 G'_k 对应的 smcc G'_k 以及边 st。然后，对于所有 $1 \leqslant j(\neq i) \leqslant m$，我们添加边 ss'_i 和 $s'_it'_j$。这些边确保不存在经过 G_1 和 G_2 的无弦环。这个 smcc 导出了矛盾。

- （情形1b）存在 $1 \leqslant j(\neq i) \leqslant m$，使得 s'_i 与 t'_j不相邻。

  不失一般性，我们可以假设 $i < j$。根据断言 42，存在 G'_i 和 s'_j 的一个 ~ ~ ~ ~smcc G'_i, G'_j，分别满足 $s'_it'_i \in E(G'_i)$ 和 $s'_jt'_j \in E(G'_j)$。我们~按如下方式构造 G 的一个 smcc：对所有 G'_k，将 smcc G'_k 纳入其中，并加入边 st。然后，对所有 $1 \leqslant k < i$加入边 $t'_is'_k$，对所有 $i < k < j$加入边 st'_k，并对所有$j < k \leqslant m$加入 s'_jt 和 $s'_jt'_k$。这个 smcc 导致矛盾。

上述构造如图 3所示。黑色的边是 G中的边。红色的边是为构造 smcc 而添加的。

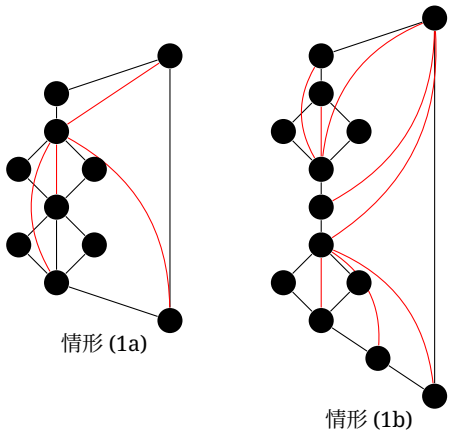

图3 情形 1 中的 smcc 构造。

- （情况2） G_1 和 G_2 都不是边。假设 G_1 和 G_2 都是较小图的并联组合。因此，在 G_1中存在两条从 s_1 到 t_1 的内部不相交路径 P、 P′；在 G_2中存在两条从 s_2 到 t_2 的内部不相交路径 Q、Q′。令 G' 为通过向 G_1 添加 Q得到的图。由于 G 是一个最小反例， G' 有一个 smcc $\tilde{G}'$。同样，P、 P′、 Q 是三条从 s_1 到 t_1且彼此内部不相交的路径。根据引理38， s_1t_1 是 $\tilde{G}'$中的一条边。由 $V(G_1)$ 在 $\tilde{G}'$ 中诱导的图 $\tilde{G}_1$ 是一个 smcc，且 s_1t_1 是 G_1中的一条边 ~ ~。类似地，可以证明 G_2 存在一个 smcc G_2，使得 s_1t_2 是 G_2中的一条边~ ~ ~。将 G_1 和 G_2 中的边 s_1 与 s_2、边 t_1与 t_2 分别识别后得到的图，是 G的一个 smcc。

我们可以假设 G_1 或 G_2 至少有一个是图的串联组合。我们反复将 G_1 和 G_2 分解为串联组合的图，直到无法继续分解为止。也就是说，所有单独的组成部分都是并联组合或一条边。令 (G_1, s_1, t_1) 依次为(G'_1, s'_1, t'_1)、(G'_2, s'_2, t'_2)、……、(G'_m, s'_m, t'_m) 的串联组合，并令 (G_2, s_2, t_2) 依次为 $(G'_{m+1}, s'_{m+1}, t'_{m+1})$、$(G'_{m+2}, s'_{m+2}, t'_{m+2})$、……、$(G'_{m+\ell}, s'_{m+\ell}, t'_{m+\ell})$ 的串联组合。

▷ 断言 43. 存在一个 $1 \leqslant i \leqslant m+\ell$，使得 $s'_i t'_i$ 是一条边。

证明。 反设 s'_i 与 t'_i不相邻，$1 \leqslant i \leqslant m+\ell$对所有 (G'_1, s'_1, t'_1) 均成立。因此，图 s'_1 是两个较小图的并联组合。因此，存在两条内部不相交的路径，每条长度至少为 3，从 s'_1 到 t'_1，且位于 G'_1 中。类似地，存在两条内部不相交的路径，每条长度至少为 3，从 s'_i 到 t'_i，且位于 G'_i中，对所有 $1 \leqslant i \leqslant m+\ell$均成立。由于 $m+\ell > 2$，我们得到一个 X-诱导图子式，它位于 G中，矛盾。 ◀

因此，存在一个 $1 \leqslant i \leqslant m+\ell$，使得 $s'_i t'_i$ 是一条边。根据断言 40，我们可以得出 G'_i 是一条边。不失一般性，我们可以假设 $1 \leqslant i \leqslant m$。注意，$(G \setminus \{s'_i t'_i\}, s'_i, t'_i)$ 是通过将 G'_{i-1}、G'_{i-2}、……、G'_1、G_2、G'_m、G'_{m-1}、……、G'_{i+1} 作串联组合，并分别将 s'_{i-k}、s'_1、t_2、t'_j 与 t'_{i-k+1}、s_2、t'_m 和 s'_j 识别得到的图，其中 $1 \leqslant k < i$ 且 $i < j \leqslant m$。同样，图 (G, s'_i, t'_i) 是将 $(G \setminus \{s'_i t'_i\}, s'_i, t'_i)$ 与一条边作并联组合得到的。现在，我们可以利用情形（1）得出矛盾。因此，最小反例不存在。

◀

▶ **引理 44.** 设 G 是一个树宽为 k 的图，T 是 G 的一个树分解，且 T 中大小为 $k+1$ 的袋在 G中均不是独立集。那么，$mtw(G) \leqslant 2tw(G) - 1$。

证明。 注意，在定理7的证明中，该构造会得到一个宽度为 $2tw(G) - 1$ 的匹配树分解，前提是每个大小为 $tw(G) + 1$ 的袋都至少含有一条边。 ◀

根据引理 39，每个 G满足 $tw(G) = 2$ 且 G 是 X-图子式自由的图都有一个 smcc。现在，T 的 $\tilde{G}$一个标准树分解同时也是 G的树分解，并且具有一项附加性质：每个大小为 3 的袋都至少含有一条边。我们应用引理 44 为 G构造一个宽度为 3 的匹配树分解。 ◀

为完成对部分 2-树的匹配树宽的刻画，我们证明以下引理。

▶ **引理 45.** $mtw(Y) \geqslant 5$

证明。证明与引理 35 的证明类似。我们还会用到这样一个事实：u_1、u_2和 u_3 没有任何公共邻点。反设 Y 存在一个宽度严格小于 5 的约化匹配树分解 T。设 B_l 是 T中的一个叶节点。我们先说明，B_l 必须包含 u_1、u_2或 u_3中的一个。否则，它只包含其他边的某个子集，例如 u_4u_5（其他情况对称）。由于这些顶点不是悬挂顶点，B_l 在 T中的邻点（记作 B_p）将是 B_l的超集。设 B_l包含 u_1 和 u_4（其余情况对称）。由于 T 是约化的，且宽度小于 5，袋 B_p 必须同时包含 u_1 和 u_2（或者 u_1 和 u_3，这是对称的情况）。以 B_l为根对 T定根。

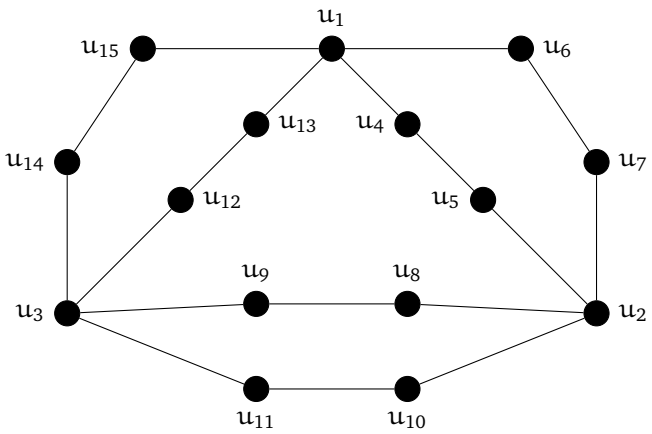

图 4 图 Y。

如果 $u_3 \in B_p$，我们就完成了，因为 $|B_p| \geqslant 6$。设 B_c 是 B_p 中包含 u_3的最近后代。不失一般性，假设 $u_2 \notin B_c$。u_1 和 u_2 不可能都不在 B_c中。否则，u_3 的所有邻居都必须在 B_c 中，这将意味着 $|B_c| \geqslant 6$。因此，u_1 和 u_3 都在 B_c 中（另一种情况对称）。它还必须包含一个与 u_1 相邻的顶点 v，因为该分解是匹配的。考虑 T中从 B_p 到 B_c 的路径 P。如果 u_8（u_{10}）没有出现在这条路径上，那么 u_9（分别为 u_{11}）就必须出现在这条路径上的所有袋中。因此，为了与 u_9（或 u_{11}）匹配，B_p 必须包含 u_3。所以，u_8 和 u_{10} 必须出现在这条路径上。

令 B_1 为 P 中第一个同时出现 u_8 和 u_{10} 的袋。我们将证明分为两种情况。

（u_8 和 u_{10} 都出现在 B_1 中）如果 $u_2 \notin B_1$，那么 $|B_1| \geqslant 6$，结论成立。否则，$B_1 = \{u_1, v, u_2, u_8, u_{10}\}$，其中 v 是 u_1 的某个邻点。令 B_2 为从 B_1 到 B_c 中第一个出现边 u_8u_9 或 $u_{10}u_{11}$ 的袋。于是，要么 $B_2 = \{u_1, v, u_2, u_{10}, u_8, u_9\}$（$u_{10}u_{11}$ 尚未出现），要么 $B_2 = \{u_1, v, u_{10}, u_{11}, u_8, u_9\}$（两条边同时出现），结论成立。

（B_1中缺少 u_8 或 u_{10} 中的一个）这意味着 B_1中有 u_9 或 u_{11}。我们有 $B_1 = \{u_1, v, u_2, u_{10}, u_9, u_3\}$或 $B_1 = \{u_1, v, u_2, u_8, u_{11}, u_3\}$，因此得证。◀

图 5 是 Y 的一个匹配树宽更小的超图。

4.2 使用匹配树宽查找并计数子图和诱导子图

▶ **定理 12**。给定一个以 H 为输入的 m-边图，我们可以在 $\tilde{O}(m^2)$-时间内计数满足 $k \leqslant 10$的 P_k 子图数量。

证明。在$\mathsf{Spasm}(P_{10})$中有 300 多个图。我们已验证它们的mtw 都至多为 3。为尽量减少工作量，我们可以筛除所有满足以下条件的图：其 Spasm 中的图具有$\mathsf{tw}(G') = 1$ 或$\mathsf{tw}(G') = 2$，且 G' 是 X-诱导子式禁用图。注意，由于 X 有 9 个顶点和 12 条边，因此它不可能是$\mathsf{Spasm}(P_{10})$中任何图的诱导子式。此外，树宽为 4 的禁用子式也都不可能出现在$\mathsf{Spasm}(P_{10})$中。因此，我们只需分析$\mathsf{Spasm}(P_{10})$中树宽为 3 的图。这样的图只有 18 个

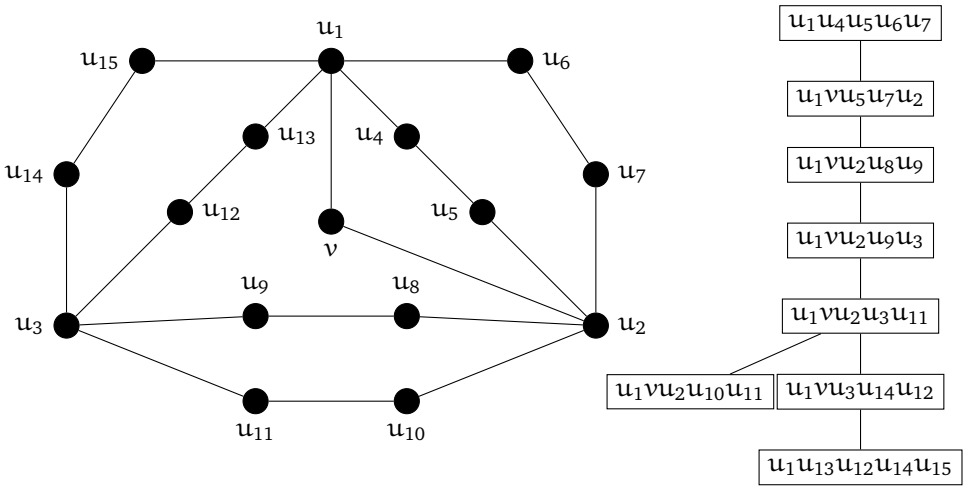

图 5 图 Z 及其宽度 4 的匹配树分解。

这些图。它们列在与本文相关的代码仓库中的一个 PDF 文件里（ https://github.com/anonymous1203/Spasm）。

由于 $\mathsf{Spasm}(P_k) \subseteq \mathsf{Spasm}(P_{k+1})$ 对所有 $k \geqslant 2$，我们可以对所有顶点数少于 10 的路径作出相同的断言。现在我们使用等式 1来计算结果。◀

▶ 注释 46。由于 $K_5 \in \mathsf{Spasm}(P_{11})$，且 K_5 的树宽为 4，上述方法无法为 P_k 得到一个 $\widetilde{O}(m^2)$ 算法，其中 $k \geqslant 11$。

我们注意到，上述证明还给出了 $\widetilde{O}(m^2)$ 时间算法，用于计数 $\mathsf{Spasm}(P_{10})$中的任意模式。这是因为，若 $G \in \mathsf{Spasm}(P_{10})$，则 $\mathsf{Spasm}(G) \subseteq \mathsf{Spasm}(P_{10})$。在下述定理中，我们指出 $\mathsf{Spasm}(P_{10})$中的一个重要图类。

▶ **定理 13.** 给定一个 m-边图 H作为输入，我们可以在 $\widetilde{O}(m^2)$-时间内统计长度至多为9 的环的数量。

证明. 我们注意到 $\mathsf{Spasm}(C_k) \subset \mathsf{Spasm}(P_{10})$ 对于 $k \leqslant 9$。 ◀

▶ 注释 47. 由于 $K_5 \in \mathsf{Spasm}(C_{10})$，且 K_5 的树宽为 4，上述方法无法为 C_k （其中 $k \geqslant 10$）提供一个 $\widetilde{O}(m^2)$ 算法。

我们还可以利用为同态多项式高效构造算术电路的方法来检测小型诱导子图。暴力搜索可以在 C_6 时间内，在一个 m-边宿主图中找到，甚至统计 $O(m^3)$ 作为诱导子图的数量。Bläser、Komarath 和 Sreenivasaiah [2] 证明，高效计算同态多项式可以加速诱导子图的检测。例如，他们的技术可用于证明，在一个 C_6 -顶点宿主图中检测诱导 n可以在 $O(n^4)$ 时间内完成。本节中，我们推导出一种 $\widetilde{O}(m^2)$ 算法，用于在一个 C_6 -边宿主图中检测诱导 m。该算法是 Bla¨ser、Komarath 和 Sreenivasaiah [2]所提出算法的自然类比。

现在，我们给出一些定义，以便理解同态多项式如何用于诱导子图检测。

▶ **定义 48.** 我们将模式图G在 n 个顶点的宿主图上的诱导子图同构多项式记为 Ind_G，定义如下。该多项式的变量为 y_v和 $x_{\{u,v\}}$，其中所有 $u,v \in [n]$ 均满足此条件。

$$\mathsf{Ind}_G = \sum_{G'} \prod_{v} y_v \prod_{e} x_e \prod_{f} (1 - x_f)$$

其中， G' 遍历与 G同构的 K_n 的所有（不一定是诱导的）子图， v 遍历 G'的顶点， e 遍历 G'的边，而 f 遍历 K_n 中连接 G' 的顶点、但不属于 G'的边。

我们用 $\mathsf{Ind}_G(E(H))$表示将 H 中的邻接关系代入边变量后得到的多项式。注意，$\mathsf{Ind}_G(E(H))$ 的单项式是 $|V(G)|$个顶点变量的乘积，并对应于与 G同构的 H 的诱导子图。此外，这些单项式的系数均为 1，因为对于任意给定的 k 个顶点，至多只有一个与 G 同构的诱导子图。诱导子图多项式与子图多项式之间的关系如下：

$$\mathsf{Ind}_G(E(H)) = \sum_{G'} (-1)^{|E(G')-E(G)|} \#\mathsf{Sub}_G[G']\mathsf{Sub}_{G'}[H](x_e = 1) \tag{2}$$

其中， G' 遍历所有 k个顶点的、包含 G 的超图，且 $\#\mathsf{Sub}_G[G']$ 表示 G 在 G'中作为子图出现的次数。我们使用替换 $(x_e = 1)$ 表示将多项式中的所有边变量都替换为 1。许多作者曾使用这一方程的变体进行诱导子图检测（参见 [23, 2, 16]）。

下面简要介绍如何使用同态多项式检测诱导子图同构（更详细的说明见 [2] ）。注意，当且仅当 $v_1, \ldots, v_k$ 诱导出一个 G时，$\mathsf{Ind}_G(E(H))$ 才含有单项式 $x_{v_1} \cdots x_{v_k}$。因此，要检查 G是否作为诱导子图出现，只需测试 $\mathsf{Ind}_G(E(H))$ 是否非零。此外，每个单项式的系数均为 1，因为一个 k 顶点的子图至多包含一个与 G同构的诱导子图。因此，判断 H 是否包含一个与 G 同构的诱导子图，可以归约为判断 $\mathsf{Ind}_G(E(H))$ 模 2 是否非零。在特征为 2 的环上计算的优点是，它会消去等式 2 右侧所有满足 $\#\mathsf{Sub}_G[G']$ 为偶数的 $\mathsf{Sub}_{G'}[H](x_e = 1)$。然而，我们尚无高效计算子图多项式的方法。这里，我们利用如下观察：$\mathsf{Sub}_{G'}[H](x_e = 1)$ 等于 $\frac{1}{\#\mathsf{Aut}(G')}\mathsf{Hom}_{G'}[H](x_e = 1)$ 的多线性部分。因此，要测试 $\mathsf{Sub}_{G'}[H](x_e = 1)$ 模 2 是否非零，只需测试 1 $\#\mathsf{Aut}(G')\mathsf{Hom}_{G'}[H](x_e = 1)$ 是否含有系数为奇数的多线性项。

为了检查是否存在系数为奇数的多线性项，我们可以随机代入满足方程 $x^2 = 0$ 的元素，这些元素取自 $\mathbb{Z}_2$ [15]。需要强调的是，我们只将这些元素代入顶点变量。边变量始终都替换为 0 或 1。使用特征为 2 的环还带来了另一个问题。现在，对于自同构数量为偶数的图，我们无法通过求出同态多项式再除以自同构的数量来计算 $\frac{1}{\#\text{自同构}(G')}\mathsf{Hom}_{G'}[H](x_e = 1)$。解决办法是针对每个这样的 G'，计算一个多项式，使其在 $\mathsf{Hom}_{G'}[H](x_e = 1)$ 的多线性部分中避开这些自同构，这样就无需进行除法，同时还要谨慎避免引入额外的多线性项。这正是下述证明的关键。

▶ **定理 14.** 给定一个具有 m 条边的宿主图作为输入，我们可以找到一个诱导的 C_6，或者报告不存在这样的结构，耗时为 $\widetilde{O}(m^2)$。

证明. 我们将说明如何计算多项式，使其多线性部分与 $\mathrm{Hom}_{G'}[H](x_e = 1)$ 相同，并且对于所有包含 C_6 且含有 C_6 奇数个副本的 6 顶点超图，所有单项式的系数均为奇数。完整列表见图 6。

这些计算需要针对每个这样的 2 对算法稍作修改 G'。我们考虑 C_6的情况。 $\mathrm{Hom}_{C_6}[H](x_e = 1)$ 中的每个多线性单项式的系数都是 12。通过确保多项式中只包含满足以下条件的同态 σ： 其中 $\sigma(2) = \min(\sigma(2), \sigma(3), \sigma(5), \sigma(6))$ ， 我们可以确保所有 C_6 的子图 H 都恰好被计数三次，每次对应 $\{\sigma(1), \sigma(4)\}$的一种选择。算法处理袋2365（图 6）时，可以在第 22 行执行这项检查。注意，在算法中，我们只需要遍历 H中存在的边。这是因为其他边变量无论如何都会被替换为 0，这些单项式也必然会消失。这对于确保我们的算法保持 $\tilde{O}(m^2)$ 而非 $\tilde{O}(n^4)$至关重要。

我们再考虑列表中的另一个情形。图 6 中第一行第三列的图有四个自同构（水平翻转和垂直翻转）。通过确保算法 2 中的第 22 行也会检查 $\sigma(3) < \sigma(6)$ （以阻止水平翻转）和 $\sigma(2) < \sigma(4)$（以阻止垂直翻转），我们可以确保这些子图在多项式中恰好计数一次。算法处理袋1634和1234时，分别可以执行这些检查。

图 6 展示了匹配树分解，以及可用于应用算法 σ 的约束；通过这些约束，可以计算所有这些多项式。

◀

▶ **定理 15。** 给定一个 m 边宿主图作为输入，我们可以在 $\tilde{O}(m^{(k-2)/2})$ 时间内找到一个诱导 P_k ，或者报告不存在这样的图。

证明。 我们首先证明 $\overline{P_k}$的匹配树宽为 $k-3$。我们只考虑奇数k 的情形（偶数情形类似）。设路径上的顶点为 1、2、.....、 $2j+1$。那么，我们的匹配树分解将有三个袋：一个根袋，仅排除 $\{j, j+2\}$；根袋的一个左子节点，排除 $\{j, j+1\}$；以及根袋的一个右子节点，排除 $\{j+1, j+2\}$。这显然是一个树分解。为说明它是匹配的，在根袋处，我们有匹配 $(1, 2j+1)$， $(2, 2j)$， ....., $(j-1, j+3)$， $(j+1, j+3)$。在左子节点处，我们可以保持其余边不变，并匹配 $(j+2, j-1)$。类似地，在右子节点处，我们匹配 $(j, j+3)$。

由于 $\overline{P_k}$具有两个自同构，因此我们还需要证明，在计算同态多项式时，可以避开其中一个自同构。注意，非恒等自同构 τ 必须具有 $\tau(1) = k$ 以及 $\tau(k) = 1$。因此，在算法2中构造同态多项式时，只要始终确保 $\sigma(1) < \sigma(k)$ ，就可以避免这种情况。

我们知道 $\mathrm{Ind}_{P_k}(E(H)) = \mathrm{Sub}_{P_k}[H](x_e = 1) \pmod 2$。定理得证。 ◀

5 致谢：

S. Mishra 的研究工作部分获得了智利国家研究与发展署（ANID）Fondecyt 博士后资助项目 3220618 的资助。

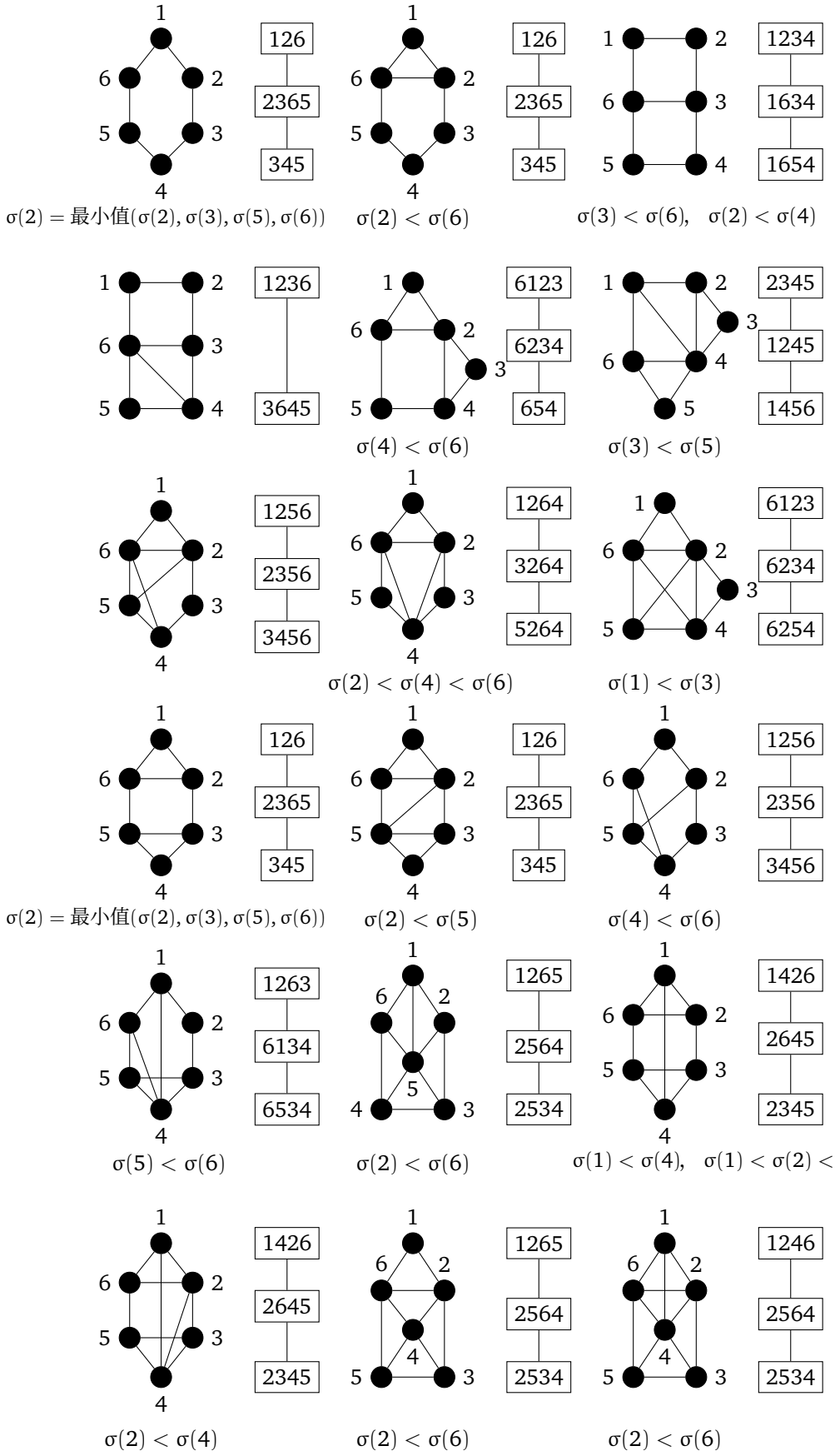
1 2 3 4 5 6
126 2365 345
σ(2) = 最小值(σ(2), σ(3), σ(5), σ(6))
σ(2) < σ(6)
1234 1634 1654
σ(3) < σ(6), σ(2) < σ(4)
1236 3645
6123 6234 654
σ(4) < σ(6)
2345 1245 1456
σ(3) < σ(5)
1256 2356 3456
1264 3264 5264
σ(2) < σ(4) < σ(6)
6123 6234 6254
σ(1) < σ(3)
σ(2) = 最小值(σ(2), σ(3), σ(5), σ(6))
σ(2) < σ(5)
1256 2356 3456
σ(4) < σ(6)
1263 6134 6534
σ(5) < σ(6)
1265 2564 2534
σ(2) < σ(6)
1426 2645 2345
σ(1) < σ(4), σ(1) < σ(2) < σ(6)
1426 2645 2345
σ(2) < σ(4)
σ(2) < σ(6)
1246 2564 2534
σ(2) < σ(6)

图6 检测诱导 C_6

参考文献

1 N. Alon、R. Yuster 和 U. Zwick。寻找并计数给定长度的环。*Algorithmica*，17(3):209–223，1997年3月。 doi:10.1007/BF02523189。 （↑ 2，3，8，14）

2 Markus Bläser、Balagopal Komarath 和 Karteek Sreenivasaiah。图模式多项式。载于 Sumit Ganguly 和 Paritosh K. Pandya（编），*第38届 IARCS 软件技术与理论计算机科学基础年会，FSTTCS 2018，2018年12月11日至13日，印度艾哈迈达巴德*，第122卷，*LIPIcs*，第18:1–18:13页。Schloss Dagstuhl - Leibniz-Zentrum für Informatik，2018年。 doi:10.4230/LIPIcs.FSTTCS.2018.18。 （↑ 4，8，31，32）

3 Marco Bressan 和 Marc Roth。退化图中的精确与近似模式计数：新算法、困难性结果与复杂性二分。载于*2021年 IEEE 第62届计算机科学基础年会 (FOCS)*，第 276–285 页，2022 年。 doi:10.1109/FOCS52979.2021.00036。 （↑ 8）

4 Radu Curticapean、Holger Dell 和 Dániel Marx。同态是计数小子图的良好基础。载于*第49届 ACM SIGACT* 计算理论年会论文集，STOC 2017，第 210–223 页，美国纽约州纽约市，2017 年。计算机协会。 doi:10.1145/3055399.3055502。 （↑ 2，3，8，14）

5 Radu Curticapean 和 Dániel Marx。子图计数的复杂性：只有顶点覆盖数的有界性才重要。载于*IEEE 第55届*计算机科学基础年会，*FOCS 2014*，美国宾夕法尼亚州费城，*2014年10月18日至21日*，第 130–139 页。IEEE计算机学会，2014 年。 doi:10.1109/FOCS.2014.22。 （↑ 8）

6 Josep Díaz、Maria J. Serna 和 Dimitrios M. Thilikos。部分 k-树的 h-着色计数。载于 第7届计算与组合国际年会论文集，CO- COON '01，第 298–307 页，柏林、海德堡，2001 年。施普林格出版社。（↑ 4，6）

7 Julian Dörfler、Marc Roth、Johannes Schmitt 和 Philip Wellnitz。诱导子图计数：一种针对 # w[1]-困难性的代数方法。*Algorithmica*，84(2):379–404，2022 年。doi:10.1007/s00453-021-00894-9。（↑ 8）

8 Friedrich Eisenbrand 和 Fabrizio Grandoni。关于固定参数团和支配集的复杂性。理论计算机科学，326:57–67，2004 年 10 月。doi:10.1016/j.tcs.2004.05.009。 （↑ 3）

9 Peter Floderus、Mirosław Kowaluk、Andrzej Lingas 和 Eva-Marta Lundell。诱导子图同构：某些模式是否比其他模式容易得多？载于 Joachim Gudmundsson、Ju- lián Mestre 和 Taso Viglas 编，计算与组合学，第 37–48 页，柏林、海德尔-堡，2012年。Springer Berlin Heidelberg。 （↑ 7，8）10 Jacob Focke 和 Marc Roth。利用遗传性质计数小型诱导子图。载于 Stefano Leonardi 和 Anupam Gupta 编，*STOC '22*：第 *54* 届 *ACM SIGACT* 计算理论年会，意大利罗马，*2022* 年 *6* 月 *20* 日至 *24* 日，第 1543–1551 页。ACM，2022年。 doi:10.1145/3519935.3520008。 （↑ 8）11 Fedor V. Fomin、Daniel Lokshtanov、Venkatesh Raman、Saket Saurabh 和 B. V. Raghavendra Rao。用于子图查找与计数的更快算法。计算机与系统科学杂志，78(3)：698–706，2012年。 doi:10.1016/j.jcss.2011.10.001。 （↑ 3）12 Archontia C. Giannopoulou 和 Dimitrios M. Thilikos。树深度的障碍。离散数学电子札记，34：249–253，2009年。欧洲组合学、图论与应用会议（EuroComb 2009）。网址：https://www.sciencedirect.com/science/article/pii/S1571065309000821, doi:https://doi.org/10.1016/j.endm.2009.07.041。 （↑ 18）

13 Ton Kloks、Dieter Kratsch 和 Haiko Müller。高效查找与计数小型诱导子图。《信息处理快报》，74(3-4)：115–121，2000年。 doi:10.1016/S0020-0190(00)00047-8。（↑ 2，3）14 Balagopal Komarath、Anurag Pandey 和 C. S. Rahul。图同态多项式：算法与复杂性。CoRR，abs/2011.04778，2020年。网址：https://arxiv.org/abs/2011.04778， arXiv:2011.04778。（↑ 3，4）15 Ioannis Koutis。用于路径与打包问题的更快代数算法。载于自动机、语言与编程国际研讨会，第 575–586 页。Springer，2008年。（↑ 32）16 Miroslaw Kowaluk、Andrzej Lingas 和 Eva-Marta Lundell。通过方程计数和检测小子图。《SIAM 离散数学杂志》，27：892–909，2013年。（↑ 3，4，32）17 Jaroslav Nešetˇril 和 Svatopluk Poljak。论子图问题的复杂性。Commentationes Mathematicae Universitatis Carolinae，026(2)：415–419，1985年。网址：http://eudml.org/doc/17394。（↑ 2）18 Marc Roth、Johannes Schmitt 和 Philip Wellnitz。检测与计数小子图并计算参数化 Tutte 多项式：基于环面网格与 Cayley 图扩展器的下界。载于 Nikhil Bansal、Emanuela Merelli 和 James Worrell 编，《第 48 届自动机、语言与编程国际研讨会（ICALP 2021）》，Leibniz 国际信息学会议论文集（LIPIcs）第 198 卷，第 108:1–108:16 页，德国达格施图尔，2021年。Schloss Dagstuhl – Leibniz 信息学中心。网址：https://drops.dagstuhl.de/opus/volltexte/2021/14177，doi:10.4230/LIPIcs.ICALP.2021.108。（↑ 8）19 Virginia Vassilevska。加权图中路径问题的高效算法。博士论文，卡内基梅隆大学，宾夕法尼亚州匹兹堡 15213，2008年。（↑ 8）20 Douglas B. West。《图论导论》。Prentice Hall，2000年9月。（↑ 9）21 Ryan Williams。最大二元可满足性。算法百科全书，第 507–510 页，2008年。doi:10.1007/978-0-387-30162-4_227。（↑ 2）22 Ryan Williams。以 o*(2k) 时间查找长度为 k 的路径。《信息处理快报》，109(6)：315–318，2009年2月。 doi:10.1016/j.ipl.2008.11.004。（↑ 2，3）23 Virginia Vassilevska Williams、Joshua R. Wang、Ryan Williams 和 Huacheng Yu。以三角形时间查找四节点子图。载于第二十六届 ACM-SIAM 离散算法年会论文集，SODA '15，第 1671–1680 页，美国，2015年。工业与应用数学学会。（↑ 4，32）