arXiv:2603.22240v1 [cs.DS] 23 Mar 2026

通过到有界路径宽度的距离研究分量阶连通性的结构化核化分界线

Jakob Greilhuber CISPA亥姆霍兹信息安全中心，德国萨尔布吕肯

Roohani Sharma
基础科学研究所（IBS）离散数学组，韩国大田

摘要

在本文中，我们研究了广泛存在的顶点覆盖（VC）问题的一个经典推广，即分量阶连通性（COC）问题。在 COC 中，给定一个无向图 G、整数 $d \geq 1$ 和 k，目标是确定是否存在一个至多包含 k 个顶点的集合，删除该集合中的顶点后，所得图的每个连通分量都至多包含 d 个顶点。当 $d = 1$ 时，这正是 VC。

本研究受到关于顶点覆盖（VC）的结构参数多项式核化结果的启发。一方面，Jansen & Bodlaender [$TOCS$ 2013] 证明，当参数为到树宽为 1 的图的距离时，VC 存在多项式核；另一方面，Cygan、Lokshtanov、Pilipczuk、Pilipczuk & Saurabh [$TOCS$ 2014] 证明，当参数为到树宽为 2 的图的距离时，VC 不存在多项式核。

Greilhuber & Sharma [$IPEC$ 2024] 证明，对于任意 $d \geq 2$，d-COC 在参数为到路径宽度为 2 的森林的距离时不存在多项式核。这里，d-COC 与 COC 相同，只不过 d 是一个固定常数，并非输入的一部分。我们补充了这一结果，并证明，对于 COC，分界线与顶点覆盖（VC）问题类似：在 VC 中，距离到树宽为 1 的图与距离到树宽为 2 的图，分别对应允许和不允许多项式核化的结构参数化之间的分界线；而在 COC 中，这条分界线则位于距离到路径宽度为 1 的图与距离到路径宽度为 2 的图之间。本研究的主要技术成果是：以到路径宽度为 1 的图的距离加上 d 作为参数时，COC 存在多项式核。

问题 d-COC 也可以表述为针对某个恰当图族 $\mathcal{F}$ 的 $\mathcal{F}$-子式删除问题。围绕 $\mathcal{F}$-子式删除的一个核心问题是：对于哪些图族$\mathcal{F}$和小图封闭图类 $\mathcal{G}$，当参数取为到 $\mathcal{G}$ 的距离时，该问题存在多项式核。以下情形已知存在完整的二分刻画：(1) $\mathcal{F} = \{P_2\}$ [Bougeret 等人，$SIDMA$ 2022],；(2) 当 $\mathcal{F}$ 是由至少三个顶点构成的双连通图组成的有限集合，且其中至少包含一个平面图时 [Bougeret 等人，$STACS$ 2026]。然而，正如 Bougeret 等人明确指出的 [$STACS$ 2026], ，这些结果甚至无法涵盖 2- 连通图删除问题。我们的结果表明，当 $d \geq 2$ 时，以到 $\mathcal{G}$ 的距离为参数的 d-COC 多项式核化的可处理性分界线，与属于类别 (1) 或 (2) 的任何 $\mathcal{F}$-子式删除问题的可处理性分界线都不同。因此，我们的结果表明，在研究以到某个小图封闭图类的距离为参数时多项式核化的二分刻画方面，d-COC 是 $\mathcal{F}$-子式删除问题类中的一个离群问题。

2012年ACM学科分类 计算理论 → 参数化复杂性与精确算法

关键词与短语 核化、分量阶连通性、毛毛虫图、路径宽度、结构参数化

资助 *Jakob Greilhuber*：德国萨尔布吕肯计算机科学研究生院成员。*Roohani Sharma*：获基础科学研究所青年科学家奖学金（IBS-R029-C1）资助。

1 引言

顶点覆盖。顶点覆盖（VC）问题是 Karp 在其开创性著作中提出的 21 个经典 NP 完全问题之一 [41]，也是参数化复杂性中的一个典型问题，可视为用于开发和测试该领域算法技术的范例 [15, 23]。给定一个无向图 G 和一个整数 k，该问题要判断是否存在一个大小至多为 k 的集合 $S \subseteq V(G)$，使得 $G - S$ 的每个连通分量的大小都至多为一。

围绕以解的大小 k为参数时 VC 的核化复杂度，形成了一条尤为丰富的研究脉络。核化（关于该主题的教材式介绍，参见 [23] 或 [15, 第 2]章）是研究 NP 困难问题预处理的一种严谨范式。这里的基本问题是设计多项式时间算法，将参数化问题的一个实例化简为一个等价实例，使其大小和参数均受输入参数的某个函数所界定。黄金标准是输出规模为该参数的多项式函数；在这种情况下，我们称该问题存在多项式核。

针对以解的大小为参数 k 的 VC 核化，催生了丰富多样的技术，包括 Buss 规则 [9], 、皇冠分解 [13, 21], 、（加权）扩张引理 [45], ，以及利用著名 Nemhauser-Trotter 定理的基于线性规划的核 [12, 43, 45, 49], 。这些技术使该问题目前最佳的核至多包含 $2k - c\log k$ 个顶点，其中 c [46]为某个常数。上述结果也启发了针对更广泛问题族的大量推广与扩展。

分量阶连通性。顶点覆盖（VC）的一种自然推广是分量阶连通性（COC）问题：给定一个图 G 以及整数 $d \geq 1$ 和 k，判断是否存在一个大小至多为 k 的集合 $S \subseteq V(G)$，使得从 G 中删除 S 后，所得图的每个连通分量大小都至多为 d。这样的集合 S 称为 G 的 d-coc 集合。当 $d = 1$ 时，这正是顶点覆盖问题。除了是顶点覆盖的一种经典推广之外，COC 也独立地作为一种自然的网络脆弱性度量出现 [30, 42]。本文还关注将 d 固定为某个不属于输入的常数所得到的问题族。具体而言，对于任意整数 $d \geq 1$，问题 d-COC 的输入是一个图 G 和一个整数 k，需要判断 G 是否具有一个大小至多为 k 的 d-coc 集合。对于每个 d [47]，问题 d-COC 都是 NP 完全的。注意，顶点覆盖与 1-COC 完全相同。

有趣的是，许多针对顶点覆盖开发的核化技术，例如 Buss 规则、皇冠分解、（加权）扩张引理，甚至 Nemhauser-Trotter 定理，在 $d \geq 2$ [2, 10, 11, 20, 23, 45, 51] 时也适用于 d-COC。这表明，针对顶点覆盖的解大小参数化所取得的积极核化结果，往往可以推广到以解的大小为参数的 d-COC。

然而，当我们考虑结构参数时，核化的整体格局会发生巨大变化。在这方面，针对顶点覆盖问题最重要的研究之一来自 Jansen & Bodlaender [39]，他们证明了：若以图的反馈顶点集数（fvs）作为参数，顶点覆盖问题存在多项式核；而且可以证明，该参数不大于最小解的大小。参数 fvs 也可以等价地理解为到树宽为 1 的图的（顶点删除）距离。Cygan、Lokshtanov、Pilipczuk、Pilipczuk 和 Saurabh [16] 的研究则补充了这一正向结果，他们证明，若以到

树宽为 2 的图的距离作为参数，顶点覆盖问题不存在多项式核。[1]

与顶点覆盖（VC）形成鲜明对比的是， d-COC 在结构参数化下的表现更不乐观。Greilhuber & Sharma [29] 证明，当 $d \geq 2$ 时， d-COC 关于反馈顶点集参数化时不存在多项式核；更进一步，即使以到路径宽度为 2 的森林的距离为参数，也不存在多项式核。[2] 关于反馈顶点集的核化下界也可由 Jansen 和 Donkers [19]的结果推出。

我们的主要结果。在本文中，我们为 d-COC 确定了一个清晰的结构二分：类似于到树宽为 1 的图的距离与到树宽为 2 的图的距离之分为 VC 的多项式核化划定界限，我们证明，到路径宽度为1 的图与到路径宽度为2 的图 构成了 d-COC 多项式核化的分界线。我们的主要结果表明，以到路径宽度至多为 1 的图的距离为参数的 d-COC 存在多项式核。

我们将这个问题记为 d-COC/pw-1。输入包括一个图 G、一个整数 k，以及一个集合 $M \subseteq V(G)$，其中 $G - M$ 的路径宽度至多为 1。任务是判断 G 是否存在一个大小至多为 k 的 d-coc 集合，参数为 $|M|$。假设调节集 M 作为输入的一部分给出并不失一般性，因为可以在多项式时间内计算出一个调节集，其大小至多是最小调节集的常数倍 [31]。

在我们的主要结果中，我们证明即使问题 COC/d+pw-1 也存在多项式核。这里，问题 COC/d+pw-1 与 d-COC/pw-1 相同，只是现在 d 是输入的一部分，且参数为 $|M| + d$。关于本文所考虑问题的完整形式化定义，请参见第 10 节。为 COC/d+pw-1 给出一个核，远比仅得到 d-COC/pw-1 的结果强得多。例如，对于 d-COC/pw-1 的核，允许运行时间为 $O(|M|^d)$ 或核大小为 $O(d^d \cdot |M|^4)$，但对于 COC/d+pw-1 而言，这完全不可接受。

▶ **主定理 1**。 问题 $\mathrm{COC}/d{+}\mathtt{pw}\text{-}1$ 存在一个具有 $O(d^7|M|^3 + d^6|M|^4)$个顶点的核。对于每个整数 $d \geq 1$ ，问题 $d\text{-}\mathrm{COC}/\mathtt{pw}\text{-}1$ 存在一个具有 $O(|M|^4)$个顶点的核。

主定理 1 肯定地回答了 Greilhuber 和 Sharma [29], 提出的一个公开问题：当参数为到线性森林的删除距离（即每个连通分量都是一条路径的森林）时， d-COC 是否存在多项式核。

小子图删除问题。 核化（及更广泛领域）中最重要的问题族之一是$\mathcal{F}$-子式删除问题族。在此，$\mathcal{F}$ 是一个固定的有限图族。给定一个图 G 和整数 k，$\mathcal{F}$-子式删除问题旨在判定是否存在一个大小至多为 k 的集合 $S \subseteq V$（G），使得 $G - S$ 不含$\mathcal{F}$-子式，也就是说， $G - S$ 不以$\mathcal{F}$ 中的任何图为子式。顶点覆盖、反馈顶点集和平面顶点删除等许多基础问题都属于这一表达能力丰富的问题族（有关$\mathcal{F}$-子式删除的最新详细介绍，参见 [5] ）。问题 d-COC也不例外，同样适用于这一框架。具体而言，当$\mathcal{F}$ $=\{P_2\}$时，顶点覆盖等同于$\mathcal{F}$-子式删除；将 $\mathcal{F}$ 设为 $=\{P_3\}$即可得到问题2-连通图删除；对于 $d \geq 3$，将$\mathcal{F}$ 设为由所有具有 $d+1$ 个顶点的连通图组成的图族即可

[1] 这里提到的所有下界都建立在合理的复杂性假设之上，属于条件性结论。[2] 在 [29] 中给出的结果针对到细分梳状图的距离这一参数；不难看出，这些图是路径宽度至多为 2 的森林。

。就解的大小这一参数而言，当$\mathcal{F}$ 包含一个平面图 [22], 时，已知存在多项式核；此外，还有一系列有影响力的研究致力于理解$\mathcal{F}$-子式删除问题关于结构参数化 [5, 6, 8, 17, 19, 40]的核化复杂度。由此引出的或许最重要的问题是：对于哪些图族$\mathcal{F}$ 和小图封闭图类$\mathcal{G}$ ，以到$\mathcal{G}$ 的距离为参数的$\mathcal{F}$-子式删除问题具有多项式核?

对于某些问题族$\mathcal{F}$，这个问题已经得到了完整解决。根据 Bougeret、Jansen 和 Sau 的结果 [6], ，当$\mathcal{F}$ $=\{P_2\}$时，也就是说，对于顶点覆盖，当且仅当$\mathcal{G}$ 具有有界桥深度时，才存在多项式核。这里，桥深度是一种为这一结果引入的结构参数。我们不详述其形式化定义，只指出任何森林的桥深度都至多为一。当$\mathcal{F}$ $=\{K_3\}$时，也就是说，对于反馈顶点集，当且仅当$\mathcal{G}$ 到森林类的消除距离有界时，才存在多项式核 [17]。最近， [17] 的结果得到了极大推广：如果$\mathcal{F}$ 是一个有限的双连通图集合，其中每个图至少有三个顶点，且至少包含一个平面图，那么当且仅当$\mathcal{G}$ 到$\mathcal{F}$-子式自由图类的消除距离有界时，才存在多项式核 [5]。因此，迄今发现的所有二分定理都围绕着$\mathcal{G}$是否具有有界桥深度，或到$\mathcal{F}$-子式自由图类的有界消除距离展开。

问题族 d-COC 不属于上述任何结果的适用范围。事实上，Bougeret 等人 [5] 明确指出，2-连通图删除的核化复杂度尚不清楚，并提出桥深度是否会影响该问题。正如前文所述，当 $d \geq 2$, d-COC 关于到森林的距离参数化时，不存在多项式核 [19, 29], ，因此，$\mathcal{G}$ 具有有界桥深度显然不足以解决问题。我们的主定理 1 表明，当 $\mathcal{G}$ 是路径宽度至多为 1 的图类时，存在多项式核。该类 $\mathcal{G}$ 到无 $\mathcal{F}$-子式的图类的消除距离并无界。因此，当 $d \geq 2$ 时， d-COC 的二分性必然依赖于另一个迄今尚未确定的参数。据我们所知， d-COC 甚至是首个已知其二分性既不依赖于 $\mathcal{G}$ 具有有界桥深度，也不依赖于 $\mathcal{G}$ 到无 $\mathcal{F}$-子式图类具有有界消除距离的（非平凡）$\mathcal{F}$-子式删除问题。

技术挑战。 在技术层面，我们将本工作的主要结果与其他针对 d-COC结构参数化的多项式核进行比较。Jansen和Pieterse [40]证明，对于每个 d 以及每个常数 η ，问题 d-COC都存在一个以树深度为η 的调节集大小为参数的多项式核。还有其他文献给出了 d-COC的多项式核 [3, 5], ，但事实上，Jansen和Pieterse上述结果已经蕴含了它们所考虑参数的多项式核的存在性。据我们所知，目前尚无其他正向结果。Jansen和Pieterse的核的一个关键特点是，只需能够界定删除调节集后所得图的连通分量数，就足以得到该结果。随后，可以利用树深度至多为 η的图的结构，在至多 η轮迭代中界定这些连通分量的大小。每一轮都涉及将数量有界的顶点（消除森林的根）移入调节集，然后重新界定连通分量的数量。

对于我们的主要结果，关键的技术挑战在于界定这些连通分量的大小。特别是，长路径是路径宽度为一、但树深度无界的图，因此，先前正向结果中采用的迭代方法完全失效。我们的结果是 COC 核化领域的首个成果，它设计了归约规则，能够在移除与参数对应的调节集后，界定图中连通分量的大小

。我们将在第 2 节更详细地解释这一阶段及其挑战。

1.1 进一步结果。

我们以一系列关于 COC 核化的进一步结果补充本文的贡献。尽管这些结果本身也很有趣，但其技术难度低于主定理 1，其中大多数结果可通过将顶点覆盖的已知结果推广到 d-COC，或采用相对标准的技术得到。

关于 d 参数。 首先，我们更详细地考察数值 d 所起的作用。回顾一下，在主定理 1 d 中，该数值要么是常数，要么至少是参数的一部分。我们证明这是不可避免的。定理 9.8 表明，以图的顶点覆盖数 加上解的大小 k 为参数的 COC 不存在多项式核，除非 NP $\subseteq$ coNP/poly（不过，该问题可通过一个简单的分支算法在 FPT 时间内求解，见定理 9.11）。此外，我们在定理 9.1 中证明，仅以到路径宽度为 1 的图的距离为参数时，COC 甚至是 W[1]-难的。因此，当 d 既不是常数，也不属于参数时，就不可能获得类似于主定理 1 的结果。

以到有界度图的距离为参数。接下来，我们利用主定理 1 的结果，在以到最大度至多为二的图类的距离为参数时，得到一个多项式核（对于问题 COC，还需加上 d ）。我们将这一结果与关于平面图上最大度为三的 d-COC 的 NP 困难性结果结合，从而得到一个核化二分定理。这一结果将 Majumdar、Raman 和 Saurabh [48] 关于顶点覆盖的结果推广到了更一般的问题 COC。对于图类 $\mathcal{G}$ 和整数 $d \geq 1$ ，问题 d-COC/dist-to-$\mathcal{G}$ 是指输入中同时给出一个调节集的 d-COC 问题，该调节集用于将图变为 $\mathcal{G}$ 中的图；也就是说，该集合为 $M \subseteq V(G)$，满足 $G - M \in \mathcal{G}$，参数为 $|M|$。问题 COC/d+dist-to-$\mathcal{G}$ 的定义相同，只是 d 是输入的一部分，参数为 $|M| + d$。

▶**定理 1.1.** 当 G 是最大度至多为2 的图类时，问题COC/d+dist-to-G 存在一个多项式核；并且，对于每个 $d \geq 1$，d-COC/dist-to-G 都存在一个多项式核。当 G 是最大度3的平面图类，且 d 为任意正整数时，问题 d-COC/dist-to-G 不存在任何大小的核，除非P $=$ NP。

限制连通分量的数量。 最后，作为主定理 1 的核化过程的一部分，我们需要能够减少 $G - M$ 的连通分量数量；回想一下， M 是输入中给定的调节集。为此，我们谨慎地、但相对标准地推广了 Hols、Kratsch 和 Pieterse 针对顶点覆盖给出的一个结果 [35, 定理 1.3 和 3.12]。我们得到一个预处理程序，可用于定理 4.13 中更一般的一类参数。事实上，对于任意 d，以及任何满足以下条件的图类$\mathcal{G}$ ：(1) 对不交并封闭；(2) 问题d-COC/dist-to-$\mathcal{G}$ 存在多项式核，我们的算法都可用于将 $G - M$ 的连通分量数量减少到 $|M|^{O(1)}$。因此，我们的算法是减少连通分量数量的终极预处理算法。

我们指出，定理 4.13 的结果依赖于 $\mathcal{G}$ 具有有界的极小 d-阻断集。图 G 的一个 d-阻断集是满足以下条件的集合 $X \subseteq V(G)$：G的任何最小 d-coc 集合都不包含 X；若该集合的任何真子集都不是 d-阻断集 [29]，则称这个 d-阻断集是极小的。若存在某个常数 b ，使得图类 $\mathcal{G}$ 具有有界的极小 d-阻断集，则

对于每个图 $G \in \mathcal{G}$，其任何极小 d-阻断集的大小至多为 b。1-阻断集这一概念（有时是隐含地）在许多针对顶点覆盖 [6,8,24,34,35,39],的结构参数化的核化结果中发挥作用；类似的阻断集概念对于一般的 $\mathcal{F}$-子式删除问题也很重要，参见例如 [5]。

1.2 展望。

本文为 COC 提出了若干新的正向核化结果。具体而言，当 $G - M$ 是到路径宽度为 1 的图的调节集时，我们提出了一种新颖的方法来缩小连通分量的规模。缩小连通分量规模这一任务之所以新颖，是因为此前已知的所有关于 d-COC 结构参数化的正向核化结果，都是通过限制连通分量的数量直接得到的。

为 d-COC/dist-to-$\mathcal{G}$ 建立完整二分定理，与其他 $\mathcal{F}$-子式删除问题的已知结果类似，是一个自然的最终目标。然而，我们似乎尚未充分理解这一问题，因而无法证明这样的结果。我们认为，正如顶点覆盖问题的情形，在尝试证明完整二分定理之前，先积累各种正向结果会很有益。尽管如此，我们还是想提出以下猜想

▶ **猜想 1.2.** 对于子式封闭图类 G 和所有常数 $d \geq 1$，问题d-COC/dist-to-G 存在多项式核，当且仅当 G 的极小 d-阻断集有界。

这一猜想似乎合理，因为根据 Greilhuber & Sharma 的一项结果 [29], ，当 $\mathcal{G}$ 对不交并封闭且极小 d-阻断集无界时，问题 d-COC 在以到 $\mathcal{G}$ 的距离为参数的情况下不存在多项式核。此外，该猜想在 $d = 1$ 这一情形下成立，因为 $\mathcal{G}$ 的极小 1-阻断集有界，当且仅当 $\mathcal{G}$ 的桥深度有界 [6]。而且，迄今为止，$\mathcal{G}$的极小 d-阻断集有界，最终总能转化为 d-COC 的多项式核。

2 主定理 1 的技术概述

在本概述中，我们将介绍主定理 1 证明背后的主要思路。重点尤其在于如何缩小 $G - M$ 的连通分量的规模，因为这是我们工作中最具创新性的部分。

一个图的路径宽度至多为一，当且仅当它是一个毛毛虫森林，也就是说，是一个每个连通分量都是毛毛虫图的森林 [1, 引理 2.4]。毛毛虫图是满足以下条件的连通图：删除每个度为一的顶点后，会得到一条（可能为空的）路径。我们首先设定一些便于使用的术语。然后，我们定义一种连通子图的特定打包方式，该方式利用了树中大小至少为 $d + 1$ 的连通子图所具有的紧 Erdős-Pósa 性质 $G - M$。基于这一打包方式，我们识别出一个具有较长脊柱的子毛毛虫（即度为一的顶点所连接的较长中央路径），并可将其替换为一个等价的更小子毛毛虫。这种替换基于一个全新的洞见，我们称之为毛毛虫图的 本质，它使我们能够将毛毛虫图划分为等价类，从而同一等价类中的毛毛虫图可以相互替换。我们证明中的一个有趣特点，是通过与抽象代数的联系，给出了该等价类的小型代表元大小的显式上界。上述结果使我们能够缩短一个 $G - M$毛毛虫图的脊柱长度。为了界定 $G - M$毛毛虫图的数量，我们利用了这样一个事实：

对于所有 $d \geq 1$，毛毛虫图的最小 d-阻塞集的大小至多为二。最后，我们应用一个已知的、针对解的大小参数（加上 d）的核，以确保剩余的约简实例规模较小。

毛毛虫树与脊。对于每个毛毛虫图 C，为方便起见，我们假设给定了 spine(C)，它是 C 的一个固定非空子路径，且 C 中不属于 spine(C) 的每个顶点的度均为一。不属于脊柱的顶点称为悬挂顶点，我们用 pendants(C) 表示 C 的所有悬挂顶点组成的集合。对于 C 的每个悬挂顶点 v，我们将 v 唯一的邻点称为 v 的父顶点。此外，我们假设脊柱具有固定顺序，并将脊柱的一端称为左端点，另一端称为右端点。即使取毛毛虫图 C 的连通子图 C'，这一顺序仍保持不变；所得毛毛虫图 C' 的脊柱是 C 的脊柱限制在 C' 的顶点上所得的部分。若脊顶点 v 比 w 更靠近左端点，则称脊顶点 v 位于脊顶点 w 的左侧；若 v 位于悬挂顶点 w 的父顶点左侧，则称其位于悬挂顶点 w 的左侧。若悬挂顶点 v 的父顶点位于顶点 w 的左侧，则称悬挂顶点 v 位于顶点 w 的左侧。类似地，我们定义顶点/顶点集/子图位于另一个顶点/顶点集/子图左侧或右侧的概念。

现在，我们来讨论如何缩小 $G - M$ 的连通分量。在非常高的层面上，我们的策略是找到 $G - M$ 的一个较大的子毛毛虫，并安全地将其替换为一个更小的毛毛虫图。不幸的是，这是一项相当艰巨的挑战：子毛毛虫的具体结构不仅会以不同方式影响解，而且它还可能与调节集有许多连接，这些连接都需要纳入考虑。

解紧打包。为了确定哪些子毛毛虫可以被其他子毛毛虫替换，我们利用特定的、规模最大的顶点不相交连通图打包，其中每个图至少有$d+1$ 个顶点。这些打包很有用，因为它们体现了如下所述的紧致 Erdős-Pósa 性质。对于图 H，我们将 H 的最小 d-coc 集合的大小记为最优值$_d(H)$。以下引理直接由 Cockayne 等人 $[14,$的定理 $1]$得出。

▶ **引理 2.1**（由 Cockayne 等人的 [14, 定理 1]推出）。设 T 是一棵树，P 是由 T 中每个包含 $d+1$ 个顶点的两两顶点不相交的连通子图组成的最大打包。那么，最优值$_d(T) = |P|$。特别地，任何最优的 d-coc T 集合都必须从 P 的每个图中恰好选取一个顶点。

现在，我们介绍解紧打包的概念，该概念建立在引理 2.1 的基础上。解紧打包是由大小至少为 $d+1$ 的顶点不相交连通图组成的一种特定打包。

▶ **定义 2.2**（解紧打包）。设 C 是一只毛毛虫图，其脊柱（C）$= v_1, \ldots\ldots, v_n$，其中 v_1 是脊柱的左端点，v_n 是脊柱的右端点。C 的一个集合 P 由两两顶点不相交的连通子图组成，且每个子图至少包含 $d+1$ 个顶点；若满足以下条件，则称 P 为解紧打包：

1. $|P| =$最优值$_d(C)$，*2.* 对于每个 $p \in$悬挂顶点(C)，如果 p 的父顶点 p' 属于P中的某个图 H，那么 p 也属于 H；*3.* 存在一个整数 $i \in [0, n]$，使得每个满足 $j \leq i$ 的脊顶点 v_j 都属于P中的某个图，而没有满足 $j > i$ 的脊顶点 v_j 属于某个图；*4.* 对于每个 $H \in P$，包含 H 最右侧脊柱顶点的单元素集合是 H的一个 d-coc集合。

令 F 为毛毛虫森林，$\mathcal{C}$ 为 F 的连通分量集合。那么，当 $F = \bigcup_{C' \in \mathcal{C}} P_{C'}$ 时，打包 P 是 F 的解紧致打包；对于每个 $C' \in \mathcal{C}$，我们都有 $P_{C'}$ 是 C' 的解紧致打包。

8 通过到有界路径宽度的距离对 COC 核化划定分界线

解紧致打包必须满足的性质足以保证每个毛毛虫森林都有且仅有一个解紧致打包。对于任意毛毛虫森林，都可以使用贪心算法轻松计算其解紧致打包（见引理 6.1）。$G - M$ 的解紧致打包对核化十分有用，因为我们知道，任何最优解都必须从打包中的每个图至少选取一个顶点，并且存在一个针对 $G - M$ 的解，使其从每个被打包的图中恰好选取一个顶点。解紧致打包的其他性质还能为我们提供其他有用的保证。例如，如果某个图 $H \subseteq V$（$G - M$）的解紧致打包包含其所有顶点，那么就存在一个针对 H 的最优解，选取 H 的最右侧脊柱顶点。因此，我们对这类图 H 了解得尤其多。

基于这一直觉，我们通过合并解紧打包中的若干个图来构造要替换的子毛毛虫。形式化地，我们引入 α-合并打包的概念，即将解紧打包中每 α 个连续的图合并而成。

▶**定义 2.3** (α-合并打包)。设 C 为一棵毛毛虫图，$\mathcal{P} = \{H_1, \ldots, H_{|\mathcal{P}|}\}$ 为 C 的解紧打包，其中 $\mathcal{P}$ 中的图按照它们沿脊柱出现的顺序排列，使得对于所有 $i \in [1, |\mathcal{P}| - 1]$，图 H_i 位于图 H_{i+1} 的左侧。设 $\alpha \geq 1$ 为一个整数。那么，$\mathcal{P}$ 的 α-合并打包是由 $t = \left\lfloor \frac{|\mathcal{P}|}{\alpha} \right\rfloor$ 个图 $H'_0, \ldots, H'_{t-1}$ 组成的打包，其中对于每个 $i \in [0, t-1]$，H'_i 是由 C 的顶点集 $\bigcup_{j \in [1,\alpha]} V(H_{i \cdot \alpha + j})$ 诱导的子图。

当 F 是毛毛虫森林时，F 的 α-合并打包可以通过取 F 的每个连通分量的 α-合并打包的并集得到。

我们还定义了合并图这一相关概念，并注意到 α-合并打包中的每个图都是合并图。

▶**定义 2.4**（合并图）。设 F 是一个带有解紧打包 $\mathcal{P}$ 的毛毛虫森林。如果对于某个 $\mathcal{P}' \subseteq \mathcal{P}$，$F$ 的诱导子图 C 满足 $V(C) = \bigcup_{C' \in \mathcal{P}'} V(C')$，则称其为合并图 。

与调节集没有连接的子毛毛虫。现在，我们想详细说明如何利用这些打包来替换 $G - M$ 的子毛毛虫。为了获得初步直觉，我们先考虑 $G - M$ 的 α-合并打包中的一个图 C，其中 α 是一个稍后将详细说明的适当大值，并且 C 与调节集没有边。[3] 设 $\hat{C}$ 是 $G - M$ 的连通分量，且 C 是其子图。由于 C 与调节集没有边，我们无需担心调节集对 C 内的解产生显著影响。然而，根据 C 的具体结构，毛毛虫 C 可能会对 C 的其余部分的解产生显著影响。

$\hat{}\hat{}$ 由于 C 位于 C中，我们可以自然地将 C 划分为三部分：第一部分是 $C - C$ 中位于 C左侧的连通分量 L，第二部分是 C，第三部分 R 则是 $C - C$ 中位于 C右侧的连通分量。当然，L、R 可能不存在，但为了便于说明，我们假设它们存在。考虑 G 的某个 d-coc 集合 S，其大小至多为 k。那么，L($S_L = S \cap V$)是 L的一个 d-coc 集合，C($S_C = S \cap V$)是 C的一个 d-coc 集合，而R($S_R = S \cap V$)是 R的一个 d-coc 集合。将 C 替换为另一个图 C'时，我们希望确保存在一个足够小的解 $S_{C'}$，使其适用于 C'，并能与 S_L和 S_R很好地组合。这里，将 C 替换为 C' 是指删除 C、添加 C'，并添加一条边，从

[3] 我们指出，实际的核化并没有专门用于处理这种情况的归约规则。不过，详细阐述这一情况有助于直观理解我们实际采用的归约规则是如何进行的。

从 L 的最右侧脊柱顶点到 C'的最左侧脊柱顶点，再从 R 的最左侧脊柱顶点到 C'的最右侧脊柱顶点添加一条边。

在 $L - S_L$ L 的最右侧脊柱顶点位于一个大小为 x 的连通分量中（若 L 的最右侧脊柱顶点位于 S_L 中，则设为 $x = 0$ ）。因此，这些 x 个顶点并未由 S_L“照料”，确保它们不会落入过大的连通分量是 S_C 的职责。因此， S_C 不仅是 C 的一个 d-coc 集合，甚至还是图 C_x 的一个 d-coc 集合；该图是在 C 的最左侧脊柱顶点上添加 x 个额外悬挂顶点后得到的。若考察图 R，我们会发现类似的现象： C 的最右侧脊柱顶点位于 $C - S_C$ 的某个连通分量中，该分量大小为 y （若选取 C 的最右侧脊柱顶点，则设为 $y = 0$ ）。此时，确保这些 y 个顶点不会造成任何问题是 S_R 的职责。我们可以观察到，在所有大小为 $|S_C|$ 的 C_x 的解中，我们更倾向于采用能使 y 的相应值最小化的解。若 $|S_C| > \mathsf{OPT}_d(C)$，那么我们也可以选择使 $S_{C'}$ 大于 $\mathsf{OPT}_d(C')$；这很容易做到，因为 C' 将有一个大小至多为 $\mathsf{OPT}_d(C') + 1$ 的 d-coc 集合，它选取了 C' 的最左侧和最右侧脊柱顶点。因此，我们主要关注的是：仅考虑图 C_x 的大小为 $\mathsf{OPT}_d(C)$ 的解时，值 y 能有多小。

正是为此目的，我们引入毛毛虫图的本质这一概念。本质是一个可在多项式时间内计算的函数，能够精确刻画^C 中的解如何与图 C其余部分中的解相互作用。

▶**定义 2.5** (毛毛虫图的本质)。 设 C 是一个毛毛虫图，且 C 的解紧打包包含 C的所有顶点。设 C 的脊柱为 v_1，.....， v_n，其中 v_1 是脊柱的左端点， v_n 是右端点。

对于每个整数 $x \geq 0$，定义 C_x 为通过向 v_1 添加 x个额外悬挂顶点、同时保持脊柱不变而从 C 得到的毛毛虫图。对于每个整数 $x \geq 0$ 以及 d-coc 的任意集合 Y，我们将 $\beta_x(Y)$ 定义为：若 $v_n \in Y$，则其值为 0；否则，将 $\beta_x(Y)$ 定义为包含 v_n 的 $C_x - Y$ 的连通分量中的顶点数。对于所有整数 $x \geq 0$，令$\mathcal{D}_x$ 为 C_x的所有最小 d-coc 集合构成的集合。

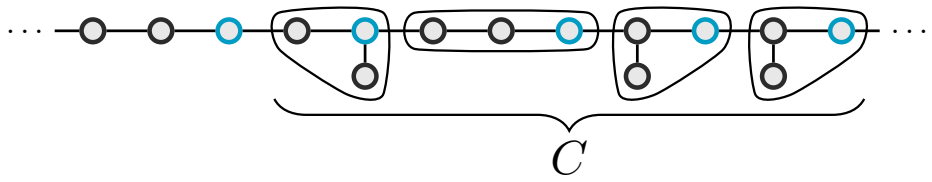
C （相对于 d）的本质是函数 $\gamma\colon \ [0, d+1] \to [0, d+1]$ 其中

$$\gamma(x) = \begin{cases} \min\{\beta_x(Y) \mid Y \in \mathcal{D}_x\} & \text{if } x \leq d \text{ and } \mathsf{OPT}_d(C_x) = \mathsf{OPT}_d(C), \\ d+1 & \text{otherwise.} \end{cases}$$

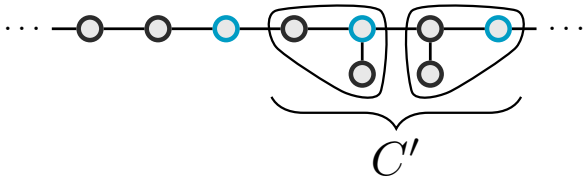
之后，本质的函数复合将发挥重要作用，这也是我们将 $d+1$ 定义为不动点的原因。本质背后的思想是用一个更小的毛毛虫图 C' 替换 C，使其与 C具有相同的本质。图1展示了这样一种替换。该图还展示了具体的选择，这些选择取决于毛毛虫图中位于 C 左侧、且与 C （以及 C'）的最左侧脊顶点处于同一连通分量中的未选择顶点数量。本质的关键性质是，替换是安全的。

▶ 引理 *2.6*。设（G， d， k， M）是 *COC*/d+pw-*1* 的一个实例，且 P是 $G - M$的一个连通合并子图。设 γ是 P的本质，且 P' 是一个本质为 γ的任意毛毛虫图。通过用 P'替换 P，由 G 构造 G'。如果存在一个大小至多为 k的 d-coc 集合 S，它是 G 的集合，则存在一个大小至多为 $k' = k - \mathit{OPT}_d(P) + \mathit{OPT}_d(P')$ 的 d-coc 集合 S'，它是 G' 的集合。

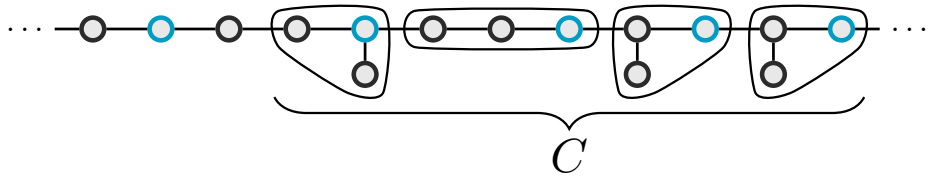
^ 这一替换之所以安全，直观上是：回想我们将 C 拆分为 L、 C、R，无论基于集合 S_L 的值 x 以及 S_C 对应的值 y 是多少，我们都有一个大小适当的解 $S_{C'}$，满足 C'_x，使得 $C' - S_{C'}$ 的最右侧脊柱顶点位于一个大小至多为 y 的连通分量中（若 $y = 0$，则该顶点被选中）。因此，我们发现，$(S \setminus S_C) \cup S_{C'}$ 是 G' 的一个解，其大小至多为 $|S| - \mathsf{OPT}_d(P) + \mathsf{OPT}_d(P')$。



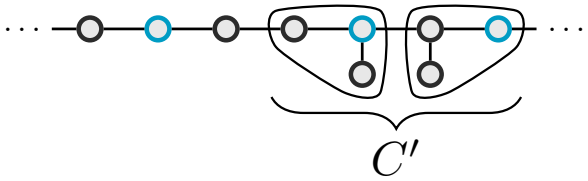
(a) $x = 0$ 时替换前的选择。



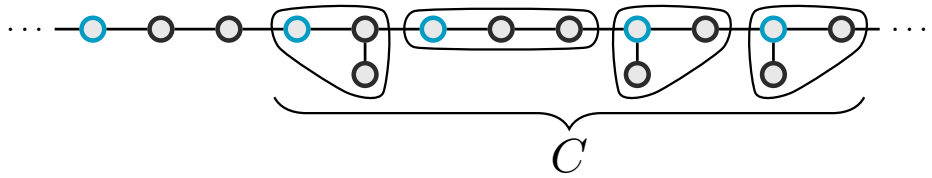
(b) 当 $x = 0$ 时，替换后的选择。



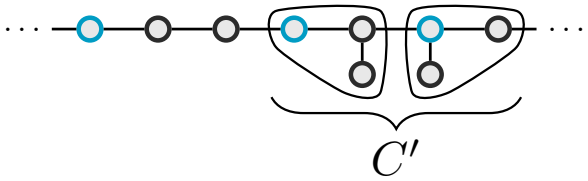
(c) 当 $x = 1$ 时，替换前的选择。



(d) 当 $x = 1$ 时，替换后的选择。



(e) 当 $x = 2$ 时，替换前的选择。



(f) 当 $x = 2$ 时，替换后的选择。

图1 在示例中，用一只具有相同本质 γ 的毛毛虫图 C' 替换子毛毛虫 C。解紧打包 C 和 C' 已突出显示。本质 γ 满足 $\gamma(0) = 0$、$\gamma(1) = 0$、$\gamma(2) = 1$ 和 $\gamma(3) = 3$。对于毛毛虫图中位于 C（或 C'）左侧、且与 C（或 C'）中最左侧脊顶点处于同一连通分量的顶点数 x，我们都在 C 和 C' 中绘出相应的解，该解也见证了本质的取值。

我们已经证明，将一个与调节集没有连接的子毛毛虫 C 替换为另一个具有相同本质的毛毛虫图 C'，是一种安全的操作。然而，我们尚不清楚如何快速找到这样一个小于 C 的毛毛虫图 C'，从而使基于此类操作的归约规则确实能够缩小实例的规模。

计算具有特定本质的毛毛虫图。 请注意，对于任意固定的 d，相对于 d 可能存在的本质数量由 d的某个函数所界定。因此，对于问题d-COC，核可以针对每种可能的本质 γ，预先硬编码具有本质 γ的最小毛毛虫图。然而，我们的结果针对的是问题 COC，其中 d 不是常数。因此，我们必须能够即时计算具有特定本质的小图。现在来说明如何做到这一点。这部分工作与抽象代数有着有趣的联系。注意，毛毛虫图的本质 γ 是非递减的，即如果 $x \leq y$，那么 $\gamma(x) \leq \gamma(y)$，并且满足 $\gamma(0) = 0$ 以及 $\gamma(d+1) = d+1$。牢记这些性质，我们如下定义本质幺半群。[4]

▶**定义 2.7** (本质幺半群)。 设 d 为一个正整数，M_d 为所有非递减函数 $\gamma\colon\ [0, d+1] \to [0, d+1]$ 的集合，其中 $\gamma(0) = 0$，且 $\gamma(d+1) = d+1$。令 $\mathsf{id}^d\colon\ [0, d+1] \to [0, d+1]$ 为恒等函数。那么，$(M_d, \circ, \mathsf{id}^d)$是本质幺半群（关于 d），其中 $\circ$ 表示函数复合操作。

不难验证，这一结构确实是一个以 id^d 为单位元的幺半群。我们证明，对于任意 d，以及 d 的本质幺半群中的任意函数 f，都可以高效地计算至多 d^3 个简单函数，使它们的函数复合为 f。我们将这些简单函数称为基本函数，并在下文中给出定义。

▶ **定义 2.8**（基本函数）。 令 d 为正整数。对于任意 $i \in [1, d]$，令dec_i^d: $[0, d+1] \to [0, d+1]$ 为如下定义的函数:

$$\mathsf{dec}_i^d(x) = \begin{cases} x - 1 & \text{if } x = i, \\ x & \text{otherwise.} \end{cases}$$

令 $\mathsf{inc}^d\colon\ [0, d+1] \to [0, d+1]$ 定义如下:

$$\mathsf{inc}^d(x) = \begin{cases} x + 1 & \text{if } 1 \leq x \leq d, \\ x & \text{otherwise.} \end{cases}$$

（相对于 d 的）基本函数集合为集合 $\{\mathsf{id}^d\} \cup \{\mathsf{dec}_i^d \mid i \in [1, d]\} \cup \{\mathsf{inc}^d\}$。

接下来，我们陈述关于将函数表示为基本函数复合的算法结果。

▶ **引理 2.9.** 存在一种算法，以满足 $f(0) = 0$ 且 $f(d+1) = d+1$ 的非递减函数 $f\colon [0, d+1] \to [0, d+1]$ 为输入，并输出 $g_1, \ldots, g_\ell$，其中 $\ell \leq d^3$，使得 $f = g_\ell \circ g_{\ell-1} \circ \cdots \circ g_1$，且对于所有$i \in [1, \ell]$，函数 g_i 都是相对于 d 的基本函数。该算法的运行时间关于 d 为多项式。

虽然看起来我们无法直接将文献中的一个结果当作黑箱来使用，但我们证明引理 2.9 的方法紧密遵循了

[4] 幺半群是一种代数结构（M，$\oplus$，e），其中 M是一个集合，$\oplus$是一个结合二元运算，M在 $\oplus$下封闭，且 $e \in M$是一个元素，使得 $m \oplus e = m$并且对于所有 $m \in M$，$e \oplus m = m$。

12 通过到有界路径宽度的距离进行 COC 核化的分界线

Higgins [32] 和 Gomes & Howie [28] 在抽象代数文献中的见解。这意味着，我们的论文是这样一个令人振奋的案例：一个看似无关领域（半群理论）中的“旧”结果，突然带来了最先进的算法结果。从代数的角度来看，一个有趣的问题是：哪些函数可以用作$\mathcal{M}_d$的生成元，以及生成任意函数 $f \in \mathcal{M}_d$所需复合的函数的最少数量是多少？解决这个问题或许能缩小我们的核大小。

现在，我们进一步阐述本质幺半群与毛毛虫图之间的联系。事实证明，不仅每个毛毛虫图的本质 γ 都属于本质幺半群，而且对于本质幺半群中的任意函数 f，也都存在一个本质为 f的毛毛虫图。我们认为，所有可能的本质都有一个具有该本质的毛毛虫图，这一事实相当有趣，因为它表明，子毛毛虫与其左侧和右侧的毛毛虫图之间的相互作用可以“复杂到无法想象”。为此，我们证明，幺半群中函数 f_1 和 f_2 的复合操作，与连接本质分别为 f_1 和 f_2的毛毛虫图的脊柱，几乎完全相同。形式化地说，我们证明引理 2.10。

▶**引理 2.10**。设 C_1 是一个毛毛虫图，它具有一个解紧打包，且该打包含有 C_1的所有顶点。设 C_1 的本质为 γ_1，且 C_1 的脊柱为 v_1、……、v_n。类似地，设 C_2 是一个毛毛虫图，它具有一个解紧打包，且该打包含有 C_2的所有顶点； C_2的本质为 γ_2，其脊柱为 w_1、……、w_m。

通过对 C_1 和 C_2 作不交并，并添加一条从 v_n到 w_1的边，构造毛毛虫图 C。C的脊柱为 v_1、..、v_n、$w_1 \ldots$、w_m。于是，C的本质 γ满足 $\gamma = \gamma_2 \circ \gamma_1$。

利用这一事实，假设我们已有以基本函数为本质的毛毛虫图，就可以利用引理 2.9，相对容易地计算出具有特定本质的毛毛虫图。接下来，我们将证明可以轻松计算出这样的毛毛虫图。

▶**引理 2.11**。有一个算法，以任意基本函数 f（相对于某个$d \geq 1$）为输入，并输出一个本质为 f 的毛毛虫图 C；该毛毛虫图具有大小为一的解紧打包（因此只包含 C），且运行时间为关于 d的多项式。

结合引理 2.9 和引理 2.11，并利用引理 2.10，我们可以轻松计算出具有特定本质的小型毛毛虫图。

▶ **引理 2.12**。存在一种算法，以非递减函数 $\gamma\colon [0, d+1] \to [0, d+1]$（满足 $\gamma(0) = 0, \gamma(d+1) = d+1$）作为输入，并输出一个毛毛虫图 C，其本质为 γ，使得 C 的解紧打包 $\mathcal{C}$ 包含 C 的所有顶点，并且该算法可在关于 d 的多项式时间内计算出 $|\mathcal{C}| \leq d^3$。

我们要替换的毛毛虫图应具有大小至少为 $d^3 + 1$ 的解紧打包，因为这样便可以使用引理 2.12 计算出一个本质相同、但解紧打包严格更小的毛毛虫图。因此，着重替换 $G - M$ 的（$d^3 + 1$）合并打包中的毛毛虫图是合理的。

处理与调节集的连接。我们已经看到，如果 C 与调节集没有连接，那么可以快速用一个本质相同但更小的图替换 C。但是，如果 $G - M$ 的连通分量与调节集之间存在边，该怎么办?

不展开详细说明，在实际的归约规则（规约规则 2）中，我们识别出 C $G - M$ 的一个较大的子毛毛虫，并用一个较小的毛毛虫 C' 替换它；这个毛毛虫与 C 具有相同的本质，且 C 还满足一些额外的有用保证，帮助我们处理可能连接到 M 的边。我们还设定 $k' = k -$ 最优值$_d$(C) + 最优值$_d$(C')。该规则安全性的正向可直接由引理 2.6 得出。对于反向，设 S 是

约简图的一个 d-coc 集合，其大小至多为 k'。如果 $N(C) \cap (M \setminus S) = \emptyset$，那么安全性的反向同样可由引理 2.6 得出。否则，C 提供的额外保证便发挥作用。它们确保 C 实际上是某个大得多的毛毛虫 C 的子图。此外，我们还保证存在一个针对 $\hat{C}$ 的最优解 $S\hat{}_C$，它会选取 $N(M \setminus S) \cap V(\hat{C})$ 的所有顶点。现在，设 $\hat{C}'$ 为用 C' 替换 $\hat{C}$ 中的 C 后得到的毛毛虫；显然，$\hat{C}'$ 是约简实例的子图。我们确保每个顶点 $v \in N(C) \cap (M \setminus S)$ 在 $\hat{C}$ 中都有 $d+1$ 个位于 C 左侧的邻居，以及 $d+1$ 个位于 C 右侧的邻居。因此，$\hat{C}'$ 中也有 $d+1$ 个 v 的邻居分别位于 C' 的左侧和右侧。于是，S 必须在 C' 两侧各选取至少一个这样的邻居。这些选出的顶点实际上将 $\hat{C}'$ 分成三部分，其中一部分（记作 $P^{*\prime}$）包含 C'。设 P^* 为图 $P^{*\prime}$，其中用 C 替换了 C'。由此可证明，$(S \setminus V(P^{*\prime})) \cup (V(P^*) \cap S\hat{}_C)$ 是输入图的一个 d-coc 集合，其大小至多为 k，这表明该规则是安全的。

完成核化。 我们可以应用上述归约规则来界定 $G - M$ 的脊柱长度，因为只要 $G - M$ 的 α- 合并打包（对于某个关于参数的多项式值 α）足够大，就总能找到合适的图 C 和 C。然而，$G - M$ 仍可能包含无界数量的连通分量。利用毛毛虫图对于每个整数 d 都存在大小至多为二的最小 d-阻塞集这一性质，可以采用相对标准的技术来界定 $G - M$ 的连通分量数量；这些技术可追溯至 Jansen & Bodlaender [39]。在推论 4.14 中，我们采用一种更复杂的方法，该方法基于扩张引理，并将 Hols 等人 [35] 的结果推广到 COC。

穷尽地应用前述规则后，spine$(G - M)$（由 $G - M$ 的各个连通分量的脊柱组成的图）的大小有一个关于参数的多项式上界。特别地，集合 $S = M \cup V(\text{spine}(G - M))$ 的大小关于 $d + |M|$ 呈多项式级，并且由于 $G - S$ 是独立集，集合 S 是 G 的一个 d-coc 集合。如果 $|S| \leq k$，我们就得到一个是实例；否则，k 的大小也有关于参数的多项式上界，此时可以应用 Xiao [51] 针对参数 $k + d$ 给出的核，从而完成核化过程。

3 预备知识

图。 我们采用标准记号，但为避免歧义，在此对其中一些记号作明确说明。本文所考虑的图均为无向简单图。对于图 G 和顶点集 $X \subseteq V(G)$，$G[X]$ 表示由 X 诱导的图。对于图 G 和集合 $X \subseteq V(G)$，我们也用 $G - X$ 表示图 $G[V(G) \setminus X]$。对于作为 G 子图的 H，我们也用 $G - H$ 表示图 $G - V(H)$。对于（子）图 H，我们用 $|H| = |V(H)|$ 表示。图 G 的最大度记作 $\Delta(G)$。

二分图是一个图 (V, E)，满足存在 V 的一个划分 (A, B)，使得图的所有边都连接 A 中的顶点与 B 中的顶点。我们通常将这样的图记作三元组 (A, B, E)。

对于顶点 v 和图 G，我们将 v 在 G 中的开邻域记为 $N_G(v)$，将闭邻域记为 $N_G[v]$。对于 $X \subseteq V(G)$，我们定义 $N_G(X) = \bigcup_{v \in X} N_G(v) \setminus X$，以及 $N_G[X] = N_G(X) \cup X$。若 H 是 G 的子图，我们定义 $N_G(H) = N_G(V(H))$ 和 $N_G[H] = N_H[V(H)]$。当上下文能够清楚表明所指的图时，我们可以省略下标 G。

给定图 G 和边 $uv \in E(G)$，边 uv 的边收缩操作会生成图 G'：删除 u 和 v，然后添加一个新顶点 w，使其与 $N_G(\{u, v\})$ 中的每个顶点都相邻。若图 H 可以通过以下操作从图 G 得到，则称 H 是 G 的子式：

14 通过到有界路径宽度的距离对 COC 进行核化的分界线

G 删除顶点、删除边和收缩边。

图类与调节集。 图类 $\mathcal{G}$ 是一个（通常无限的）图集合。给定一个图 G， G 的一个到 $\mathcal{G}$ 的调节集是一个集合 $M \subseteq V(G)$，使得 $G - M$ 是$\mathcal{G}$中的一个图。

毛毛虫图。 毛毛虫图 C 是一棵树，删除所有度为一的顶点C 后会得到一条路径。对于毛毛虫图 C， C 的脊柱 是 C的一个子图路径 P，满足 $V(C) \setminus V(P)$中的所有顶点在 C中的度都为一。注意，任何非空毛毛虫图都有一条（不一定唯一的）非空脊柱。此外，给定一个毛毛虫图，可以在多项式时间内轻松计算出一条非空脊柱。因此，我们可以假设所处理的任何毛毛虫图都带有这样一条固定的脊柱。也就是说，对于任意（非空）毛毛虫图 C，spine(C)是 C的一条固定的非空脊柱，我们称 spine(C)为 C的脊柱。

森林 F 是毛毛虫森林，如果 F 的每个连通分量都是毛毛虫。设 F是毛毛虫森林，$\mathcal{C}$ 是 F的连通分量集合。毛毛虫森林 F 的脊柱 spine(F) 是通过对图{spine(C) $\mid C \in \mathcal{C}$}取不交并得到的图。spine(F) 的顶点称为脊顶点。 F的悬挂顶点 是集合 pendants(F)，由 F 中所有不属于 spine(F) 的顶点组成。每个悬挂顶点$v \in$ pendants(F)的度恰好为一，我们将 v 唯一的邻点称为 v的父顶点 。注意，悬挂顶点 v 的父顶点 p 总是一个脊顶点。 F的脊顶点 v 对应的 pendants(v) $= N(v) \cap$ pendants(F)。

为方便起见，给定毛毛虫图 C时，我们还假定脊柱 spine(C) 有一个固定的顺序，也就是说，将脊柱视为路径 $v_1, \ldots, v_t$，并将 v_1 称为 C的左端点，将 v_t 称为 C的右端点。如果脊顶点 v 比脊顶点 w更靠近左端点（右端点），则称 v 位于 w 的左侧（右侧）；如果悬挂顶点 v 的父顶点位于 w的左侧（右侧），则称 w 位于 w的左侧（右侧）。若悬挂顶点 v 的父顶点位于 w 的左侧（右侧），则称悬挂顶点 v 位于顶点 w 的左侧（右侧）。类似地，当 W、W'是 C的子图（或 $V(C)$ 的子集）时，如果W包含一个位于 W'所有顶点左侧的顶点，就称W位于 W'的左侧；如果 W包含一个位于 W'所有顶点右侧的顶点，就称 W位于W'的右侧。此外，若考虑毛毛虫图 C的某个连通子图 H，则 H 的脊柱恰好是将 C的脊柱限制在 H的顶点上所得的部分，并且顺序保持不变。

核化 现在我们介绍参数化复杂性和核化的核心概念。

▶ **定义 3.1**（参数化问题 [23, 定义 1。1]）。 参数化问题是一个语言 $L \subseteq \Sigma^* \times \mathbb{N}$，其中 Σ 是一个固定的有限字母表。对于一个实例 $(x, k) \in \Sigma^* \times \mathbb{N}$， k称为参数。

核是针对参数化问题的特定预处理算法。

▶ **定义 3.2**（核化 [23, 定义 1。3]）。 设 $L \subseteq \Sigma^* \times \mathbb{N}$ 是一个参数化问题。 L 的核化算法（核）是一种算法，它将一个实例（x， k） $\in \Sigma^* \times \mathbb{N}$ 作为输入，并且

- 运行时间关于 $|x| + k$ 为多项式，并且

[5] 否则，包含 v 的毛毛虫图 C 将必须是一个具有两个顶点的路径，并且 spine(C) 将必须为空。

- 输出一个等价实例 (x', k')，其中 $|x'|, k' \leq h(x)$，对应某个可计算函数h。

我们也将 （x'、k'）称为实例 （x、k）的核，并将 h称为核的大小。如果 h是一个多项式函数，那么该核就是多项式的。

在核化领域中，人们经常使用能够缩减实例一部分的较小算法，这类算法称为归约规则。这些规则可以有特定的用途，例如减少连通分量的数量，或减少各个连通分量的大小。如果某条规则输出的是该问题的等价实例，则称其是安全的。当然，如果在核化算法中应用某条规则，那么该规则必须能够在多项式时间内运行。

现在我们介绍复杂性类 FPT，并指出一个众所周知的事实：当且仅当一个参数化问题具有一个（不一定是多项式的）核时，该参数化问题才属于 FPT [23, 定理 1.4]。

▶**定义 3.3** （固定参数可解问题 [23, 定义 1。2]）。 如果存在一个算法，以实例 （x、k） $\in \Sigma^* \times \mathbb{N}$作为输入，并在时间 $f(k) \cdot |x|^{O(1)}$内判定（x、k） $\in L$，其中 $f\colon \mathbb{N} \to \mathbb{N}$是某个可计算函数，则称参数化问题 $L \subseteq \Sigma^* \times \mathbb{N}$属于FPT类（固定参数可解）。

人们猜想，W[1]-困难问题不属于 FPT。类似地，在 NP ⊈coNP/poly 的假设下，已有结果排除了某些问题存在多项式核的可能性。事实上，这些下界通常排除的是多项式压缩。多项式压缩与多项式核类似，但目标问题可以是另一个（非参数化）问题。

▶ **定义 3.4 (**多项式压缩 [23, 定义 1。5])。设 $A \subseteq \Sigma^* \times \mathbb{N}$ 是一个参数化问题，$B \subseteq \Sigma^*$ 是一个非参数化问题。从 A 到 B 的多项式压缩是一种算法：给定问题 A 的实例(I, k)，输出问题 B 的一个等价实例 I'，使得 $|I'| \leq g(k)$，其中 g是某个多项式函数。此外，该算法的运行时间必须关于 $|I| + k$是多项式的。

请注意，排除（到任意目标问题的）多项式压缩也就排除了多项式核。关于这些概念的更多背景，可参见 Cygan 等人编写的参数化算法标准教材 [15]，或 Fomin 等人编写的核化专著 [23]。在我们的困难性证明中，我们使用相应的归约工具。

▶**定义 3.5 (**参数化归约 [15, 定义 13。1])。 从参数化问题 A 到参数化问题 B 的<*strong*>参数化归约</*strong*>是一种算法，其输入为问题 A 的实例。给定问题 A 的实例(I, k)，归约算法会生成 B 的一个等价实例(I', k')。此外， k' 必须由输入参数 k的某个可计算函数所界定，并且该算法的运行时间必须由某个可计算函数 f所确定的 $f(k) \cdot |I|^{O(1)}$所界定。

如果问题$\mathcal{A}$ 是 W[1]-难的，且存在从$\mathcal{A}$ 到$\mathcal{B}$的参数化归约，那么$\mathcal{B}$ 也是 W[1]-难的。类似的归约也可用于排除多项式压缩。

▶ **定义 3.6 (**多项式参数变换 [15, 定义 15。14])。 从参数化问题 A 到参数化问题 B 的<*strong*>多项式参数变换</*strong*>是一种算法，其输入为问题 A 的实例。给定实例(I, k)，它会在关于 $|I| + k$的多项式时间内运行，并输出问题 B 的一个等价实例(I', k')。此外，对于某个多项式函数 g， $k' \leq g(k)$。

如果问题 $\mathcal{A}$ 除非 NP ⊆coNP/poly，否则不存在多项式压缩；并且存在从 $\mathcal{A}$ 到 $\mathcal{B}$的多项式参数变换，那么问题 $\mathcal{B}$ 也除非 NP ⊆coNP/poly，否则不存在多项式压缩。

4 减少连通分量的数量

本节介绍一种通用流程，用于减少 $G - M$ 的连通分量数量，其中 G 是一个图， M 是 G 的一个调节集，而 G 属于某个满足特定性质的图类 $\mathcal{G}$；这些性质将在后文介绍。该流程是 Hols 等人 [35] 针对顶点覆盖问题提出的方法的推广。具体而言，我们的定理 4.13 将 [35, 定理 1.3] 和 [35, 定理 3.12] 推广至 d-COC 问题，而推论 4.14 将 [35, 定理 1.3] 推广至 COC/d+dist-to-$\mathcal{G}$。我们采用他们的方法和证明策略，并通过一些调整使其适用于 d-COC。例如，我们使用扩张引理而非匹配，并作出其他类似调整，以处理顶点覆盖与 d-COC 之间的差异。值得一提的是，这也使得我们采用的方法与 Bhyravarapu 等人 [3] 的策略类似；他们针对 $c \geq d$，给出了一个以 c-COC 集的大小为参数的 d-COC 核。

形式化地说，我们考虑参数化问题 d-COC/dist-to-$\mathcal{G}$，即 d-COC，其输入还包含一个到 $\mathcal{G}$ 的 G 调节集 M。该问题的参数为 $|M|$。注意，我们将该问题视为一个承诺问题，也就是说，保证输入具有这种形式。问题在于，图类 $\mathcal{G}$ 可能是这样一个类：判定某个图是否属于 $\mathcal{G}$ 不可判定，或者至少无法在多项式时间内完成。因此，可能无法足够快速地检查 M 是否确实是 $\mathcal{G}$ 的调节集。不过，如果能够在多项式时间内判定某个图是否属于图类 $\mathcal{G}$ ——当我们将 $\mathcal{G}$ 固定为某个明确的图类时，我们所有的结果都满足这一点——那么也可以去掉这一假设，将该问题视为常规参数化问题。本节陈述的结果也部分适用于问题 COC/d+dist-to-$\mathcal{G}$，其中 d 不是固定常数；但我们首先聚焦于问题 d-COC/dist-to-$\mathcal{G}$，稍后再说明在何种情况下这些结果也能推广到更一般的问题。

现在，我们正式引入 d-阻塞集。顶点覆盖的阻塞集概念在 [34, 35] 和 [6], 中得到了形式化，尽管未必明确提出，但它在许多早期研究中发挥了重要作用（更多细节请参见 [35] ）。最近，Bougeret 等人 [7]也提出了阻塞集这一概念。顶点覆盖的阻塞集概念以自然的方式推广到了 d-COC [29]。

▶ **定义 4.1**（阻塞集 [29, 定义 3]）。设 G 为一个图， $d \geq 1$ 为一个整数。若 V（G）的任意最小 d-coc 集合都不包含 X作为子集，则 X 的子集是一个 d-阻塞集。此外，若 X 的任何真子集都不是 d-阻塞集，则 X 这个 d-阻塞集是极小的。

设 G 为一个图类。若对某个整数 b，任意图 $G \in G$ 的任意极小 d-阻塞集的大小都至多为 b，则称 G 的极小 d-阻塞集大小至多为 b。

对于 d-COC/dist-to-$\mathcal{G}$，其中$\mathcal{G}$ 在取不交并运算下封闭，至关重要的是，$\mathcal{G}$ 中图的最小 d-阻塞集的大小有某个常数上界。否则，在标准计算复杂性理论假设下，不存在多项式核 [29, 定理 4] 。本节将证明，当$\mathcal{G}$ 的最小 d-阻塞集大小至多为 b 时，在满足一些温和的附加条件下，我们可以将 $G - M$的连通分量数量缩减至关于 $|M|$ 的多项式，其指数取决于 b。

现在我们定义块的概念。该概念由 Jansen 和 Bodlaender [39] 提出，用于对以反馈顶点集大小为参数的顶点覆盖进行核化。Hols 等人 [35] 将这一概念推广到问题 1-COC/dist-to-$\mathcal{G}$，其中任意$\mathcal{G}$ 的最小 1-阻断集大小都有界。

▶ **定义 4.2** (块)。 令(G, k, M) 为一个 d-COC/dist-to-G 实例，其中 G 的最小 d-阻断集大小至多为 b。该实例的块集合为 $\mathcal{X} = \{X \in 2^M \mid 1 \leq |X| \leq b\}$。

我们关注块 $X \in \mathcal{X}$，因为我们可以量化一个解中不包含 X 的任何顶点时所造成的不利影响。特别是，如果不取 X 中的任何顶点会使解的大小变得非常糟糕，那么我们可以推断，任何最优解都必须至少包含 X 中的一个顶点。为了量化不将一个块中的顶点纳入解所造成的不利影响，我们引入冲突这一概念；这一概念同样可以追溯到 Jansen 和 Bodlaender 的奠基性工作 [39]。回顾一下，对于图 H，我们将 H 的最小 d-coc 集合的大小记为 $\mathrm{OPT}_d(H)$。

▶ 定义 *4.3*（冲突）。设 (G, k, M) 是 d-COC/dist-to-G 的一个实例，G' 是 $G - M$ 的一个子图，$X \in \mathcal{X}$ 是该实例的一个块。由 X 施加于 G' 上的冲突为 $\mathrm{conf}_d(G', X) = |N_{G'}(X)| + \mathrm{OPT}_d(G' - N_{G'}(X)) - \mathrm{OPT}_d(G')$。

对于块 X 和图 G'，X 对 G' 施加的冲突，是指包含 $N_{G'}$中所有顶点（X）的 G' 的最优解，与 G'的最优解之间的差值。对于顶点覆盖问题，如果一个块 X 中没有顶点被选中，那么任何解都必须选中 X的所有邻居。因此，冲突值直接描述了如果不选取 X 中的任何顶点，那么在 G' 中必须额外付出多少代价。对于 COC 问题，不选取块 X 中的顶点通常并不意味着必须选取 X的所有邻居。尽管如此，我们的定义是顶点覆盖定义的一种自然推广；事实证明，由于如果不选取块 X 中的顶点，我们必须选取 X 中几乎所有的邻居，因此这里提出的冲突概念非常适合我们的问题。

接下来，我们介绍扩张引理，这是我们用来消除连通分量的主要工具。在陈述引理本身之前，我们需要定义q-扩张的概念。

▶ **定义 4.4** (扩张 [15, 第 2 节。4])。 设 (A, B, E) 是一个二分图。对于任意整数 $q \geq 1$，集合 $Q \subseteq E(G)$ 是 A 到 B 的 q-扩张，当且仅当 A 中的每个顶点都恰好与 Q中的 q 条边关联，且 B 中恰好有 $q \cdot |A|$ 个顶点与 Q中的一条边关联。 Q 中边的端点称为（被 Q）饱和的。

现在，我们可以陈述扩张引理本身 f.

▶ **引理 4.5** (扩张引理；由 [15, 引理 2。18])。 设 (A, B, E) 是一个二分图，且 $q \geq 1$ 是一个整数，使得 $|B| \geq q \cdot |A|$，并且 B中没有孤立顶点。于是，存在一个多项式时间算法，能够找到非空集合 $X \subseteq A$ 和 $Y \subseteq B$，使得存在一个 X 到 Y的 q-扩张，Y 中没有顶点在 X 之外有邻点，并且$|Y| = q \cdot |X|$。

所述变体与 [15, 引理 2.18], 中给出的并不完全相同，因为其中陈述的版本无法保证 $|Y| = q \cdot |X|$。不过，检查 [15], 的证明后不难发现，若存在从 X 到 Y 的 q-扩张，我们实际上可以在多项式时间内找到它（使用最大匹配算法）；随后，只需移除 Y 中未被找到的q-扩张饱和的所有顶点，就能得到我们所陈述的引理变体。我们陈述的变体对我们来说更方便，因为否则我们就需要区分 Y 中哪些顶点已饱和，哪些未饱和。接下来，我们定义一个称为冲突图的二分图，并将在其上应用扩张引理。

18 通过到有界路径宽度的距离实现 COC 核化的分界线

▶ **定义 4.6** (冲突图)。 设(G, k, M) 是 d-COC/dist-to-G 的一个实例，其中 G 的极小阻塞集大小有界，且 X 是该实例的块集合。设 C 是 $G - M$的连通分量集合。该实例的冲突图 H 是二分图(X, C, E)，其中 E 包含一条边，当且仅当$\mathsf{conf}_d(C, X) > 0$时，连接顶点 $X \in X$ 与 $C \in C$。

现在，我们给出用于移除连通分量的归约规则。

▶ **归约规则 1**。 设(G, k, M) 是一个 d-COC/dist-to-G 实例，其中 G 的最小 d-阻塞集大小至多为 b。设 X 为该实例的块集合，C 为 $G - M$ 的连通分量。令 $q = (d - 1) \cdot b + 1$，并设(X, C, E) 为该实例的冲突图。

执行以下过程：*1.* 创建冲突图的一个副本 (A', B', E')，令 $k' = k$，并设 G' 为 G 的副本。*2.* 令 I 为图 (A', B', E') 的 B' 中孤立顶点的集合。通过从图中删除 I 来更新 (A', B', E')，并令 $k' = k' - \sum_{C \in I} OPT_d(C)$。同时，从图 G' 中移除 I 的所有连通分量。*3.* 如果 $|B'| < q \cdot |A'|$，则终止并输出实例 (G', k', M)。*4.* 否则，如果 $|B'| \geq q \cdot |A'|$，则对 (A', B', E') 应用扩张引理，以找到非空集合 $X \subseteq A'$ 和 $Y \subseteq B'$，使得存在从 X 到 Y 的 q-扩张。通过删除 Y 和 X 的所有顶点来更新 (A', B', E')，然后转到步骤 *2*。

接下来，我们证明该规则是安全的。

▶**引理4.7**。 归约规则 *1* 是安全的。

证明。设 (G, k, M) 为输入实例，(G', k', M) 为应用归约规则后的输出实例。此外，设 (A, B, E) 为输入实例的冲突图。令 $\mathrm{I} \subseteq B$ 为 $G - M$ 中所有因规则应用而被删除的连通分量组成的集合。注意，只有归约规则的第 2 步会删除 G 中的顶点，因此在规则执行过程中的某个时刻，I 中的每个顶点都是冲突图副本 (A', B', E') 中的一个孤立顶点。我们分别证明安全性的两个方向。

$\Rightarrow$: 假设输入实例是一个是实例，并设 S 是一个 d-coc 集合，其中包含 G，大小至多为 k。那么，S 必须从每个 最优值$_d(C)$ 中至少选取 $C \in \mathcal{I}$ 个顶点。因此， $S \cap V(G')$是输出实例的一个解，其大小至多为 k'。

$\Leftarrow$: 假设输出实例是一个是实例。令 S' 为大小至多为 k' 的 G' 的一个 d-coc 集合，并令 $M' = M \setminus S'$。设 (A', B', E') 为规则应用终止时冲突图的修改副本，即 $|A'| < q \cdot |B'|$，且步骤 3 的 if 条件计算结果为 true。令 $\mathrm{X}' = \mathrm{X} \setminus A'$。接下来证明，对于任意块 $X \in \mathrm{X}'$，我们可以假设 S' 选取 X 中的一个顶点。

▷ 断言 4.8. 存在一个大小至多为 k' 的 G' 的 d-coc 集合，它与 $\mathcal{X}$ 的每个块相交 $'$.

证明。 为实现这一目标，我们对 S' 作如下修改，以构造一个解 $\hat{}S'$。对于每个块$X \in \mathcal{X}'$，如果 X 中尚未有顶点被 S' 选中，则任意选取 X 中的一个顶点。现在，定义 $\hat{}M = M \setminus \hat{}S'$。

令 $\mathcal{D}$ 表示在步骤 4 中被归约规则删除的那些 B 顶点。注意，每个 $C \in \mathcal{D}$ 都是输出图 G'的子图。

我们接下来论证，对于每个 $C \in \mathcal{D}$，都存在某个最小的 d-coc 图 C集合，它选取了 $N(\hat{}M) \cap V(C)$ 的所有顶点。为导出矛盾，假设并非如此。

那么，对于某个 $C \in \mathbf{D}$，不存在选取了 $N(\hat{M}) \cap V(C)$ 中所有顶点的最小 d-coc集合。因此，$N(\hat{M}) \cap V(C)$ 是 C 的一个 d-阻塞集。由于 G 的极小 d-阻塞集的大小至多为 b，图 G 的每个连通分量中的极小 d-阻塞集的大小也至多为 b。因此，我们可以找到 C 的一个极小 d-阻塞集 B，它是 $N(\hat{M}) \cap V(C)$ 的子集，并且有 $|B| \leq b$。然后，对每个顶点 $v \in B$，任意选取 v 的一个邻点 $w \in \hat{M}$，由此得到一个块 $X \subseteq M$，它是冲突图中 C 的邻点。但是，由于 X 中没有顶点被 $\hat{S}'$ 选取，因此 X 必然属于 A'，于是当 C 在步骤 4 中被移除时，X 在冲突图的副本中与 C 有一条边。然而，当 C 从冲突图的副本中被删除时，根据扩张引理的性质，冲突图副本中 C 的所有邻点也都被删除了，这与 X 属于 A' 相矛盾。

因此，对于每个图 $C \in \mathcal{D}$，我们从 $\hat{S}'$ 中移除 $V(C)$，转而选择一个最小的 d-coc集合 C，使其选中 $N(\hat{M}) \cap V(C)$ 的所有顶点。我们将由此得到的选择称为 $\hat{S}''$。

显然，$\hat{S}''$ 是输出实例的一个解，因为我们从解S' 开始，向其中添加更多顶点，然后仅在 $G' - M$ 的连通分量内更改选择，并且所改为的选择既是一个连通分量的解，又能保证 $\hat{}$ 在该分量内选中 M 的所有邻点。

因此，我们现在需要论证 $\hat{S}''$ 的大小至多为 k'。为此，令$\mathcal{X}''$ 为 $\mathcal{X}$，中未选中 X 的块 X 的集合。我们有 $\left|\hat{S}'\right| \leq k' + |\mathcal{X}''|$，因为为了构造 $\hat{S}'$，我们从 $\mathcal{X}''$ 的每个块中至多向 S' 添加一个顶点。

现在，我们需要确保 $\hat{S}''$ 充分小于 $\hat{S}'$。注意，根据步骤 4，存在一个从 $\mathbf{X}''$ 到 D 的 q-扩张。实际上，$\mathbf{X}''$ 的每个块 X 在步骤 4 中的某个时刻都会被一个 q-扩张饱和，而归约规则找到的任何 q-扩张的所有端点随后都会立即从冲突图的副本中删除。固定 $\mathbf{X}''$ 的一个块 X，并注意到 S' 必须选取 X 的邻居中除至多 $b(d-1)$ 个以外的所有邻居。然而，由于 $q = b(d-1) + 1$，这意味着 D 中存在一个连通分量 C，满足 $\mathrm{conf}_d(X, C) > 0$，其中 S' 选取了 $N(X) \cap V(C)$ 的所有顶点。特别地，这意味着 $|S' \cap V(C)| > \mathrm{OPT}_d(C)$。利用 q-扩张，我们可以将 $\mathbf{X}''$ 的各个块单射地映射到 D 的连通分量，使得 S'，因此 $\hat{S}'$ 也不得不从这些分量中各选取一个非最优解。由此可得 $\left|\hat{S}''\right| \leq k'$，因为 $\hat{S}''$ 在 D 的每个分量中都选取了一个最优解。◁

因此，我们可以假设 S' 在每个 $X \in \mathbf{X}'$中至少选择一个顶点。现在，很容易证明，对于每个 $C \in \mathbf{I}$，都存在一个最小的 d-coc集合 S_C，使得 $N(M') \cap V(C) \subseteq S_C$。假设对于某个 $C \in \mathbf{I}$，不存在这样的集合 S_C。那么，集合 $N(M') \cap V(C)$ 是 d的一个阻塞集，并且存在一个大小至多为 b的极小 d阻塞集 $B \subseteq N(M') \cap V(C)$。现在，对于每个 $v \in B$，令 v' 为 B 在 M'中的任意一个邻点，并令 $X = \{v' \mid v \in B\}$。我们知道 $|X| \leq b$，因此 X 是一个块。

此外，由于 B 是一个 d阻塞集，且冲突图中 C 与 X 之间存在一条边，因此 $\mathrm{conf}_d(C, X) > 0$。由于 S' 选择的顶点与 $\mathcal{X}'$中的所有块都有交集，但 S'没有选择 X中的任何顶点，因此必有 $X \in A'$。这与在归约规则执行期间，C 曾是冲突图某个副本中的孤立顶点相矛盾，因此 C不能属于 $\mathcal{I}$。因此，对于每个 $C \in \mathcal{C}$，都存在一个最小的 d-coc 集合 S_C，使其选择 M' 的所有邻点。

最后，我们令 $S = S' \cup \bigcup_{C \in \mathcal{I}} S_C$，它是一个大小至多为 k 的 G 个 d-coc 集合。◀

还要指出的是，在问题定义中，我们有意允许 k 为负整数。否则，上述规则不一定会输出一个有效的 d-COC 实例。

注意， k 为负值显然意味着该实例是一个否实例，因此在实际的核化过程中，可以将这条规则改为输出一个大小为常数的否实例。现在，我们可以陈述关于减少连通分量数量的主要引理了。

▶**引理 4.9**。 设 $\mathcal{G}$ 是一个图类，其最小 d-阻塞集的大小至多为 b。归约规则 *1* 将 d-$\mathrm{COC}/\texttt{dist-to-}\mathcal{G}$ 的一个实例 (G, k, M) 作为输入，并输出一个等价的 d-COC 实例 (G', k')，其中 G' 是通过删除 G 的连通分量得到的子图，$k' \leq k$，且 $G' - M$ 的连通分量数至多为 $((d-1) \cdot b + 1) \cdot |M|^b$。

证明。设 (G, k, M) 是一个 d-COC/dist-to-G 实例。设 (G', k', M) 是归约规则 1 输出的图，并注意到 G', k' 确实满足： G' 是通过删除 $G - M$ 的连通分量从 G 得到的，且 $k' \leq k$。回忆一下，整数 q 被设为 $(d-1) \cdot b + 1$。因此，剩下要证明的只有 $G' - M$ 至多包含 $q \cdot |M|^b$ 个连通分量。

设 (A', B', E') 为归约规则在步骤 3 终止时的冲突图副本。注意，图 G' 中仍然存在的 $G' - M'$ 的任何连通分量，要么是 B' 的一部分（因此从未被规则应用从冲突图中删除），要么是在规则应用的步骤 4 中被删除。规则终止时，我们有 $|B'| < q \cdot |A'|$。此外， $A' \subseteq \mathcal{X}$，其中 $\mathcal{X}$ 是输入实例中块的集合。另一方面，我们还知道，步骤 4 中从冲突图删除（但未从 G 删除）的分量数量恰好为 $|\mathcal{X} \setminus A'| \cdot q$。因此，剩余连通分量的数量至多为 $q \cdot |\mathcal{X}|$。注意 $|\mathcal{X}| \leq |M|^b$，即可完成证明。◀

请注意，我们不能断言输出实例 (G', k', M) 必然也是 d-COC/dist-to-$\mathcal{G}$ 的一个实例，因为不清楚 $G' - M$ 是否属于 $\mathcal{G}$。然而，当 $\mathcal{G}$ 对删除连通分量封闭时，显然 $G' - M$ 属于 $\mathcal{G}$，因此在这种情况下，(G', k', M) 是 d-COC/dist-to-$\mathcal{G}$ 的另一个实例。

接下来，我们要证明，在满足某些条件的前提下，可以在多项式时间内应用归约规则 1，这意味着引理 4.9 可用于核化过程。回想一下，与 Hols 等人类似， [35], 我们将图类$\mathcal{G} + c$ 定义为：对于某个整数c，恰好包含那些存在大小恰为 c、可将其调节至$\mathcal{G}$ 的调节集的图。注意，对于$\mathcal{G}$ 中的一个图，通过添加 c 个孤立顶点，可以构造出$\mathcal{G} + c$ 中的一个图，且这不会改变最小 d-coc 集合的大小。因此，我们有以下结论。

▶ **观察 4.10.** 对于任意整数 $d \geq 1$，如果某个整数 $c \geq 0$下的 $\mathcal{G} + c$ 中的实例可以在多项式时间内求解 d-COC，那么 $\mathcal{G}$ 中的实例也可以在多项式时间内求解 d-COC。

能够快速求解$\mathcal{G} + d$ 中的实例，也有助于我们快速检测$\mathcal{G}$ 中图的阻塞集。

▶ **引理 4.11**。 如果我们能在图 $\mathcal{G} + d$ 的实例上于多项式时间内求解 d-COC，那么对于任意图 $G \in \mathcal{G}$ 和任意集合 $X \subseteq V(G)$，我们都能在多项式时间内判定 X 是否是 G 的 d- 阻塞集。

证明。 首先，利用观察 4.10，在多项式时间内计算 最优值$_d(G)$ （例如，可以通过二分搜索找到一个最小 d-coc 集合，其元素来自 G）。集合 X 是

一个 d-阻塞集，当且仅当 $\mathrm{OPT}_d(G-X)+|X|>\mathrm{OPT}_d(G)$。向 G 添加一个由 d 个新顶点构成的团，并将所添加团的 d 个顶点之一称为 v。用一条边将 v 与 X 中的每个顶点相连。将所得图称为 G'。我们有 $G'\in\mathcal{G}+d$，因此可以在多项式时间内计算 $\mathrm{OPT}_d(G')$。

现在注意，如果 X 不是阻塞集，那么存在某个 G 的最小 d-coc 集合 S，它选取了 X 的所有顶点。这个集合 S 也是 G' 的最小 d-coc 集合，因此我们有 $\mathrm{OPT}_d(G')=\mathrm{OPT}_d(G)$。$G'$ 的任何 d-coc 集合都必须选取我们添加到 G 中的大小为 d 的新团中的一个顶点，或者必须选取 X 的所有顶点。因此，如果 X 是阻塞集，则必有 $\mathrm{OPT}_d(G')>\mathrm{OPT}_d(G)$。这意味着，我们可以通过计算 $\mathrm{OPT}_d(G)$ 和 $\mathrm{OPT}_d(G')$ 并比较其值，在多项式时间内判定 X 是否为 G 的阻塞集。◀

现在我们给出一个条件，在该条件下可以在多项式时间内应用归约规则 1。

▶**引理 4.12**。若 $\mathcal{G}$ 是一个图类，满足以下条件：最小 d-阻塞集的大小至多为 b，且对于任意 $H\in\mathcal{G}$ 和集合 $X\subseteq V(H)$，我们都能在多项式时间内判定 X 是否为一个d-阻塞集，且该集合属于 H，

然后，我们可以在多项式时间内应用归约规则 *1*。

证明。设 $(G,\ k,\ M)$ 是一个 d-COC/dist-to-$\mathcal{G}$ 输入实例，并设 $\mathcal{C}$ 为 $G-M$ 的连通分量集合。由于扩张引理可以在多项式时间内应用，我们只需能够在多项式时间内计算输入实例的冲突图，并且还必须能够在多项式时间内计算连通分量 C 所属 $G-M$的 最优值$_d(C)$。

根据假设，对于任意集合 $X\subseteq V(G-M)$，我们可以在多项式时间内判断 X 是否为 d-$G-M$ 的阻塞集。这使我们能够在多项式时间内计算 $G-M$ 的任意连通分量 C 的最优值$_d(C)$。为此，我们将迭代构造一个最小的 coc-S 集合 d，使其属于 C。首先，遍历 $V(C)$ 的所有顶点 v，并检查 $\{v\}$ 是否为 d- $G-M$ 的阻塞集。如果 $V(C)$ 的每个顶点都构成一个 d-$_d$ 的阻塞集，那么 C 的最优值$_d(C)=0$，算法即告完成。否则，任选一个顶点 $v\in V(C)$，使得 $\{v\}$ 不是 d- $G-M$ 的阻塞集，并令 $S_1=\{v\}$。由于 S_1 不是 d- $G-M$ 的阻塞集，因此存在一个包含 v 的 d-coc 最小集合，于是也存在一个包含 v 的 d-coc 最小集合C。接下来，我们始终保持集合 S_i 是 C 的某个 d-coc 最小集合的子集这一不变式。为了根据 S_i 计算 S_{i+1}，遍历 $V(C)\setminus S_i$ 中的所有顶点 v，并检查 $\{v\}\cup S_i$ 是否为 d- $G-M$ 的阻塞集。如果对于所有 $v\in V(C)\setminus S_i$，$S_i\cup\{v\}$ 都是 d- C 的阻塞集，那么就不存在包含S_i的 d-coc 最小集合 C。但是，由于 S_i 是 C 的某个 d-coc 最小集合的子集，这意味着 $_d(C)=i$。否则，我们找到一个顶点 v，使得 $S_i\cup\{v\}$ 不是 d- $G-M$ 的阻塞集，因此令$S_{i+1}=S_i\cup\{v\}$。由 S_{i+1} 不是 d- $G-M$ 的阻塞集这一事实可直接推出，S_{i+1} 是 C 的某个 d-coc 最小集合的子集；因此它也不是 d- C 的阻塞集。

冲突图的顶点集就是$\mathcal{X}\cup\mathcal{C}$，其中$\mathcal{X}$ 是输入实例中的块集合，而$\mathcal{X}$ 和$\mathcal{C}$ 的大小都关于输入规模呈多项式。因此，为了足够快速地计算冲突图，我们只需能够在多项式时间内判定冲突图中某个 $X\in\mathcal{X}$ 与某个 $C\in\mathcal{C}$ 之间是否存在一条边。这要求我们计算 $\mathrm{conf}_d(X,C)>0$，而只需检查 $N_G(X)\cap V(C)$ 是否为 $G-M$ 的阻塞集，即可在多项式时间内完成此操作。◀

利用关于图类 $\mathcal{G}+d$ 的详细阐述，以及关于遗传图类的一些更简单的观察，我们现在得到了主要结果：一个多项式预处理算法，可用于减少 $G-M$ 的连通分量数量。

▶**定理 4.13**。设 $d \geq 1$ 是一个整数，$\mathcal{G}$ 是一个图类，且 $\mathcal{G}$ 的最小 d-阻塞集大小至多为某个常数 b。此外，假设
(1) d-COC 可在图类 $\mathcal{G} + d$ 上于多项式时间内求解，或者(2) d-COC 可在图类 $\mathcal{G}$ 上于多项式时间内求解，且 $\mathcal{G}$ 是遗传的。

那么，存在一种多项式时间算法，对于给定的 d-COC/dist-to-$\mathcal{G}$ 实例 (G, k, M)，可输出一个等价的 d-COC 实例 (G', k')，其中G' 是通过删除 $G - M$的连通分量而得到的 G 的子图，且 $G' - M$ 的连通分量数至多为 $((d-1) \cdot b + 1) \cdot |M|^b$，并且$k' \leq k$。

证明。首先，考虑 d-COC 可在图类$\mathcal{G} + d$上于多项式时间内求解的情形。那么，该定理可直接由引理 4.9、4.11 和 4.12 得出。

否则，我们有 $\mathcal{G}$ 是遗传的，并且可以在$\mathcal{G}$的实例上以多项式时间求解 d-COC。这也意味着，对于任意 $H \in \mathcal{G}$，我们都可以在多项式时间内计算 $\mathrm{OPT}_d(H)$。于是，对于任意 $H \in \mathcal{G}$ 和 $X \subseteq V(H)$，只需比较 $\mathrm{OPT}_d(H)$ 和 $|X| + \mathrm{OPT}_d(H - X)$，即可判断 X 是否为 H 的阻塞集；这里我们利用 $H - X$ 属于 $\mathcal{G}$ 这一点，得出 $\mathrm{OPT}_d(H - X)$ 也可以在多项式时间内计算。由此，定理可由引理 4.9 和 4.12 得证。◀

请注意，算法要能运行，条件 (1) 和 (2) 确实只需满足其中之一；特别是，如果 d-COC 在 $\mathcal{G} + d$上可在多项式时间内求解，那么$\mathcal{G}$ 不必是遗传的。

我们简要说明定理 4.13 中的前提条件并不苛刻，这可由 Hols 等人 [35] 已经给出的观察得出。若参数化问题不是 FPT，则不可能拥有任何规模的核。另一方面，如果问题 d-COC/dist-to-$\mathcal{G}$ 是 FPT，那么就存在一个多项式时间算法，可在图类 $\mathcal{G} + c$ 上求解 d-COC，其中 c 为任意固定值。从这个意义上说，定理 4.13 的前提条件极为宽松：要使任何核存在，我们必须有多项式时间算法来求解图类 $\mathcal{G} + d$上的 d-COC。此外，当$\mathcal{G}$ 对不交并封闭时，如果$\mathcal{G}$ 不具有有界的极小d-阻塞集，就不可能存在多项式核 [29]。这意味着，对于任何对不交并封闭、且问题 d-COC/dist-to-$\mathcal{G}$ 存在多项式核的图类 $\mathcal{G}$，都可以应用定理 4.13。

目前所述结果仅针对问题 d-COC，其中假设 d 为常数。由于主定理 1 讨论的是问题 COC/d+pw-1，其中 d 不是常数（而是参数的一部分），我们需要确保对于该问题，也能足够快速地应用这条规则。幸运的是，定理 4.13 中关于遗传图类 $\mathcal{G}$ 的条件 (2) 可以很好地推广到这种情形。注意，我们以自然的方式将问题 d-COC/dist-to-$\mathcal{G}$ 的定义推广到问题 COC/d+dist-to-$\mathcal{G}$：将 COC/d+dist-to-$\mathcal{G}$的实例 (G, d, k, M) 视为 d-COC/dist-to-$\mathcal{G}$ 的实例 (G, k, M)。

▶**推论 4.14**。设 $\mathcal{G}$ 是一个遗传图类，b 是常数，并且在 $\mathcal{G}$ 的实例上可以于多项式时间内求解 COC，且对于所有 $d \geq 1$，$\mathcal{G}$ 的极小 d-阻塞集的大小至多为 b。那么，存在一个多项式时间算法，以 COC/d+dist-to-$\mathcal{G}$ 的实例(G, d, k, M)作为输入，并输出一个等价实例 (G', d, k', M)，该实例属于COC/d+dist-to-$\mathcal{G}$，其中

- G' 是通过删除 $G - M$的连通分量得到的 G 的子图，且 $G' - M$ 的连通分量数至多为 $((d-1) \cdot b + 1) \cdot |M|^b$，并且$k' \leq k$。

证明。输入实例的冲突图具有多项式规模。因此，尽管 d 不是常数，仍然可以在多项式时间内计算该实例的冲突图

当我们可以针对任意 $H \in \mathcal{G}$ 在多项式时间内计算 $\mathrm{OPT}_d(H)$，并且可以计算集合 $X \subseteq V(H)$ 是否是图 H （其中 $\mathcal{G}$）的阻塞集时。

对于任意 $H \in \mathcal{G}$，我们可以使用针对 COC 的算法，在多项式时间内轻松计算 $\mathrm{OPT}_d(H)$。此外，由于 $\mathcal{G}$ 是遗传的，给定一个图 H 和 $X \subseteq V(H)$，我们可以通过比较 $\mathrm{OPT}_d(H)$ 和 $|X|+\mathrm{OPT}_d(H-X)$，计算出 X 是否是 H 的阻塞集。因此，我们可以足够快地计算冲突图。扩张引理也能在多项式时间内运行，因此归约规则 1 能在多项式时间内运行。然后，结合引理 4.9，并注意到 $G' - M$ 属于$\mathcal{G}$，即可直接得出该推论，因为$\mathcal{G}$ 是遗传的，因此在删除连通分量后仍然封闭。◀

5 毛毛虫图的极小阻塞集至多有两个元素

我们必须证明毛毛虫森林（它们恰好是路径宽度至多为一的图 [1, 引理 2.4], ）不存在规模较大的极小 d-阻塞集。具体而言，我们证明，与路径的情形（见 [29]）一样，毛毛虫森林的任意极小 d-阻塞集的规模至多为二。为此，我们先重述 [29]的一个简单引理。

▶ **引理 5.1** ([29, 引理 13])。 设 G 是一个图，且 $X \subseteq V(G)$ 是 G的一个 d-阻塞集，其中 $d \geq 1$为某个整数。若存在一个子集 $X' \subseteq X$，使得 G_1 和 G_2 是 $G - X'$的不同连通分量， $V(G_1) \cap X \neq \emptyset$，且 $V(G_2) \cap X \neq \emptyset$，那么 X 不是一个极小的 d-阻塞集。

现在，我们可以着手证明毛毛虫森林的极小 d-阻塞集大小有界。

▶ **定理 5.2**。 设 G 是一个毛毛虫森林， d 为一个正整数，且 $X \subseteq V(G)$ 是 G的一个极小 d-阻塞集。那么， $|X| \leq 2$。

证明。 反设 X 是 G的一个极小阻塞集，且$|X| \geq 3$。那么， X 必定属于 G (否则，由引理 5.1 和 $X' = \emptyset$ 可立即推出矛盾)，因此不失一般性，我们可以假设 G 是连通的。由于 G 是一个毛毛虫图， G 包含脊顶点和悬挂顶点。我们根据 G 所含顶点的类型分情况讨论，并证明假设 $|X| \geq 3$ 会导致矛盾。 X 包含。

情形 1： X **包含一个悬挂顶点** v **及其父顶点** p **的** v。这意味着存在一个最小的 d-coc 集合，同时包含 v 和 p，因为$\{v, p\} \subsetneq X$。令 S 为这样的集合。那么，$S \setminus \{v\}$ 将是一个更小的集合，显然仍然是一个 d-coc 集合，这与 S 具有最小规模矛盾。

情形 2： X 包含两个具有相同父顶点 p 的悬挂顶点 v_1、 v_2 。同样，这意味着存在一个最小的 d-coc 集合 S，同时包含 v_1 和 v_2。令 S 为这样的集合。那么，集合 $(S \setminus \{v_1, v_2\}) \cup \{p\}$ 是一个严格小于 S 的 d-coc 集合。因此，这种情形也不可能发生。

情况 3：中的任意两个顶点都不是同一父顶点的 X **悬挂顶点，且** X **不包含悬挂顶点及其父顶点**。设 G的脊柱为 v_1，……， v_n。的每个顶点 X 要么是顶点 v_i 对应某个 i，要么是顶点 v_i 对应某个 i的悬挂顶点。此外，的任意两个悬挂顶点 X 都没有相同的父顶点。因此，我们可以将的顶点 X 从左

到右排序。令 z_ℓ 为 X中最左侧的顶点，令 z_r 为 X中最右侧的顶点。令 v为 $X \setminus \{z_\ell, z_r\}$ 的一个顶点。此外，令 S_ℓ 为 G 的一个最小 d-coc 集合，其中包含 X 的所有顶点$\setminus\{z_r\}$，并令 S_r 为 G 的一个最小 d-coc 集合，其中包含X 的所有顶点$\setminus\{z_\ell\}$。重要的是， $v \in S_\ell$ 和 $v \in S_r$。我们根据v 是脊柱中的顶点还是悬挂顶点，分别讨论两种不同的情况。

情形 3.1：顶点 v 是一个脊顶点。 在这种情况下，应用引理 5.1 并取 $X' =\{v\}$，即可直接得到矛盾。

情形 3.2：顶点 v 是一个悬挂顶点。令 p 为 v的父顶点，令 G_ℓ 为 $G - \{p\}$中包含 z_ℓ的连通分量，并令 G_r 为 $G - \{p\}$中包含z_r的连通分量。注意， $S'_\ell = (S_\ell \setminus \{v\}) \cup \{p\}$也是一个最小的 d-coc 集合。此外， S'_ℓ 的大小恰好为 $\mathrm{OPT}_d(G_\ell) + \mathrm{OPT}_d(G_r) + 1$，因此 S_ℓ 的大小也相同，且 $\mathrm{OPT}_d(G) =\mathrm{OPT}_d(G_\ell) + \mathrm{OPT}_d(G_r) + 1$。于是，我们必须有： $S_\ell \cap V(G_\ell)$是 G_ℓ 的一个大小为 $\mathrm{OPT}_d(G_\ell)$ 的 d-coc 集合，并且 $S_\ell \cap V(G_r)$是 G_r 的一个大小为 $\mathrm{OPT}_d(G_r)$ 的 d-coc 集合。出于同样的原因，我们知道 $S_r \cap V(G_r)$是 G_r 的一个最小 d-coc 集合，而 $S_r \cap V(G_\ell)$是 G_ℓ的一个最小 d-coc集合。此外，我们知道 S_r 和 S_ℓ 中不包含 p 的悬挂 v' ，除了 v: 否则，移除 v 和 v' 并改为选择 p会严格更优。

现在，我们想更深入地考察解 S_r 。该解包含 v，且其大小最小，因此不可能包含 p。所以， p 位于 $G - S_r$的某个连通分量中。令 C^r 为 $G - S_r$ 中包含 p的连通分量。此外，令$c^r_\ell = |V(C^r) \cap V(G_\ell)|$，并令 $c^r_r = |V(C^r) \cap V(G_r)|$。也就是说， c^r_ℓ 告诉我们 C^r 中的顶点有多少源自图 G_ℓ；同理， c^r_r 告诉我们有多少源自图 G_r。另外，令 c_p 为 p 在 G中的悬挂顶点数。如前所述，S_ℓ 和 S_r 都不可能包含 p 中除 v以外的悬挂顶点，因此 p 的 $c_p - 1$ 个悬挂顶点属于 C^r。

于是， p 的连通分量恰好包含 $1 + c_p - 1 + c^r_\ell + c^r_r$ 个顶点，这表明 $c_p + c^r_\ell + c^r_r \le d$，因为 S_r是一个 d-coc集合。同理，我们将 C^ℓ定义为 $G - S_\ell$中包含 p的连通分量，并定义量 c^ℓ_ℓ和 c^ℓ_r，于是有 $c^\ell_\ell + c^\ell_r + c_p \le d$。

反设 $c^\ell_\ell \le c^r_\ell$，以导出矛盾。令 $\hat{S} = (S_r \cap V(G_r)) \cup (S_\ell \cap V(G_\ell)) \cup \{v\}$。根据我们先前关于 S_r 和 S_ℓ的观察，我们由此知道 $|\hat{S}| =$ 最优值$_d(G)$，并且显然 $\hat{S} \subseteq S$。

剩下要证明的是，$\hat{S}$ 是 G的一个 d-coc 集合。注意，由于 $(S_r \cap V(G_r))$ 是 G_r的一个解，且 $(S_r \cap V(G_\ell))$ 是 G_ℓ的一个解，唯一可能出现的问题是： p，即连接 G_ℓ 和 G_r的顶点，位于一个过大的连通分量中。然而，在 $G - \hat{S}$ 中，顶点 p 所在的连通分量总共有 $c^r_r + c^\ell_\ell + c_p$ 个顶点。我们知道 $c^r_r + c^r_\ell + c_p \le d$，此外，根据 $c^\ell_\ell \le c^r_\ell$ 这一假设，我们现在得到 $c^r_r + c^\ell_\ell + c_p \le d$，这表明 $\hat{S}$ 是 G 的一个 d-coc 集合，并且选取了 X的所有顶点，矛盾。因此， $c^\ell_\ell > c^r_\ell$。

现在，我们通过设定 $S^* = (S_r \cap V(G_\ell)) \cup (S_\ell \cap V(G_r))$，来构造另一个 d-coc 集合 S^*。和上一段一样，我们只需证明 p 不属于 $G - S^*$中规模过大的连通分量。但是， p 属于 G_ℓ的一个连通分量，其中有 c^r_ℓ 个顶点，且有 G_r个顶点。这意味着 p 的连通分量包含 $1 + c_p + c^\ell_r + c^r_\ell < 1 + c_p + c^\ell_r + c^\ell_\ell \le 1 + d$ 个顶点，因此， S^* 确实是 G的一个 d-coc 集合。然而，这个解的大小 S^* 为 $\mathrm{OPT}_d(G_\ell) + \mathrm{OPT}_d(G_r) < \mathrm{OPT}_d(G)$，这就造成了矛盾。◀

6 多项式核

现在我们继续介绍 COC/d+pw-1 问题的核化算法。由于我们可以利用推论 4.14 来减少连通分量的数量，接下来主要的挑战是缩小连通分量的规模。

还要指出的是，在任何归约规则中，我们都不会改变 d 的值。特别是，我们的核本质上是针对问题 d-COC 的一个核，其参数是到路径宽度为 1 的图的调节集大小；此外，运行时间和核大小还是 $|M|+d$ 的多项式。

本节的主要目标是通过缩短 $G-M$ 的脊柱长度，来减小 $G-M$ 连通分量的大小。事实上，一旦我们界定了脊柱长度，就可以使用 Xiao [51] 针对参数 $k+d$ 给出的已知多项式核，完成核化过程。首先，我们回顾树的 d-coc 集合与互不相交的连通子图的最大打包之间的以下关系。

▶ **引理 2.1** (由 Cockayne 等人 [14, 定理 1]推出)。 设 T 是一棵树，P 是由 T 中顶点数均为 $d+1$ 的两两顶点不相交连通子图构成的一个最大打包。那么，最优值$_d(T) = |P|$。特别地， T 的任意一个最优 d-coc 集合都必须从 P 中的每个图恰好选取一个顶点。

我们指出，大小为 $d+1$ 的顶点不相交图的最大打包通常对 COC 很有用，因为它们总能为解的大小提供下界；例如，Casel 等人 [11] 也使用过这种方法。另外请注意，对于 $d=1$，集合 $\mathcal{P}$ 是图 T 的最大匹配。Jansen 和 Bodlaender [39] 的顶点覆盖核也利用了匹配，这启发了我们最初对这些集合产生兴趣。

d-COC 与顶点不相交的连通子图打包问题之间的对偶性，有助于我们论证解。现在我们定义解紧打包的概念：这类打包的大小与最优的 d-coc 集合相同，并具有一些额外的便利性质。

▶ **定义 2.2** (解紧打包)。 设 C 是一棵毛毛虫图，其脊柱$(C) = v_1, \ldots, v_n$，其中 v_1 是脊柱的左端点， v_n 是脊柱的右端点。若集合 P 是 C 的一组两两顶点不相交的连通子图，且每个子图至少有 $d+1$ 个顶点，则称其为解紧打包，当且仅当

1. $|P| =$最优值$_d(C)$，且 *2.* 对于每个 $p \in$悬挂顶点(C)，如果其父顶点 p' 属于图 H 中的 P，那么 p 也属于 H，且 *3.* 存在一个整数 $i \in [0, n]$，使得每个满足 $j \leq i$ 的脊顶点 v_j 都属于 P 中的某个图，而没有满足 $j > i$ 的脊顶点 v_j 属于 P 中的图；且 *4.* 对于每个 $H \in P$，包含 H 最右侧脊柱顶点的单元素集合是 H的一个 d-coc集合。

设 F 为一个毛毛虫森林，C 为 F的连通分量集合。那么，如果对于每个 $C' \in C$，都有 $P_{C'}$是 C'的一个解紧打包，且满足 $F = \bigcup_{C' \in C} P_{C'}$，则打包 P 是 F 的一个解紧打包。

注意，对于任意森林 F 及 F 的解紧打包$\mathcal{P}$，都有 OPT$_d(F) = |\mathcal{P}|$。由于我们打算在核中直接使用解紧打包，因此必须能够快速计算它们。

▶**引理 6.1.** 设 F 为毛毛虫森林。那么， F 有唯一的解紧打包，且可在多项式时间内计算。

26 COC 核化的一条分界线：基于到有界路径宽度的距离

证明。 我们可以分别为每个连通分量 C （属于 F）计算一个解紧打包，然后取这些打包的并集即可。因此，设 C 是一个脊柱为 $v_1, \ldots, v_n$的毛毛虫。我们描述一个简单的递归过程，用于计算 C的一个解紧打包。如果 C 包含少于 $d+1$ 个顶点，只需输出空集；显然，这是 C唯一的解紧打包。

否则，令 i 为使子图 $L_i = C[\bigcup_{j\in[1,i]}\{v_i\}$ ∪的悬挂顶点（v_i）包含至少 $d+1$ 个顶点的最小整数。将 L_i 从 C 中删除，得到毛毛虫图 C'。对 C'递归处理，并令得到的打包为 $\mathcal{P}'$。将 L_i 加入$\mathcal{P}'$，并返回所得打包$\mathcal{P}$。不难验证，这种贪心方法会输出一个解紧打包。

现在我们来证明，实际上 C 的所有解紧打包都包含 L_i。为此，令$\mathcal{P}$ 为 C的任意一个解紧打包。根据解紧打包定义中的性质 1，$\left|\hat{\mathcal{P}}\right| = |\mathcal{P}| > 0$。令 $\hat{L}$ 为 $\hat{\mathcal{P}}$中包含的最左侧图。根据性质 3，该图必须包含 C的连续脊顶点，包括 C最左侧的脊顶点。同样，L_i 包含 C 最左侧的脊顶点。此外，根据性质 2，L_i 和 L 都包含其脊顶点的所有悬挂顶点。如果 L_i 不是 $\hat{}\,\hat{}L$的子图，那么这就与我们对 L_i的顶点集的选择相矛盾。然而，如果 L 是 L_i的真超图，那么 L 包含的脊顶点多于 L_i。这意味着，删除 $\hat{L}$ 的最右侧脊柱顶点后得到的图仍是 L_i的一个（不一定是真）超图。但是，这样一来，性质 4 就不可能对 L成立，因为 L_i 已经包含 $d+1\hat{}$ 个顶点。因此，我们必须有 $L_i = L$。于是，根据归纳法，计算得到的解紧打包是 C的唯一解紧打包。◀

根据引理 6.1，毛毛虫森林 F （脊柱的顺序固定）的解紧打包是唯一的，因此我们可以称其为 F 的解紧打包。就我们的目的而言，有时需要处理远大于解紧打包中那些图的图。为了应对这种情况，同时仍能利用解紧打包的性质，我们引入 α-合并打包。

▶ **定义 2.3** （α-合并打包）。设 C 是一只毛毛虫，$P = \{H_1, \ldots, H_{|\mathcal{P}|}\}$ 是 C 的解紧打包，其中 P 中的图按其沿脊柱出现的顺序排列，使得对所有 $i \in [1, |P| - 1]$，图 H_i 位于图 H_{i+1} 的左侧。设 $\alpha \geq 1$ 是一个整数。那么，P 的 α-合并打包是由 $t = \left\lfloor \frac{|\mathcal{P}|}{\alpha} \right\rfloor$ 个图 $H'_0, \ldots, H'_{t-1}$ 组成的打包，其中对每个 $i \in [0, t-1]$，H'_i 是 C 在顶点集 $\bigcup_{j\in[1,\alpha]} \overline{V}(H_{i\cdot\alpha+j})$ 诱导出的子图。

当 F 是毛毛虫森林时，F 的 α-合并打包可通过取 F各个连通分量的 α-合并打包的并集得到。

直观地说，α-合并打包就是先取得解紧打包$\mathcal{P}$，再将$\mathcal{P}$ 中每 α 个连续的图合并为一个图，从而得到该打包。还要注意，由于我们讨论的是无环图，任何连通子图也是诱导子图，因此，1-合并打包恰好是毛毛虫图或毛毛虫森林的解紧打包。此外，由于解紧打包中的每个图都是连通的，并且解紧打包不包含这样的脊顶点：它不属于任何打包子图，而其右侧的某个脊顶点属于打包子图，因此，对于任意 α，α-合并打包中的每个图也都是连通的。最后请注意，α-合并打包中的每个图自身都有一个大小为 α 的解紧打包，其中包含该图的所有顶点。

接下来，我们定义合并图的概念，并指出每个 α-合并打包中的图都是合并图。

▶**定义 2.4**（合并图）。设 F 为一个带有解紧打包 P 的毛毛虫森林。若某个 F 的诱导子图 C 满足 $V(C) = \bigcup_{C' \in \mathcal{P}'} V(C')$，其中 $\mathcal{P}' \subseteq \mathcal{P}$，则称其为合并图。

我们的归约规则背后的思想是：我们识别毛毛虫图的某个部分 P，对于该部分 $G - M$，我们知道存在一个良好的解；该解选取调节集顶点的所有邻点中自身未被选取的那些顶点。然后，我们希望用一个更小的图替换 P。然而，即使我们识别出了毛毛虫图的这样一个部分 P，也不清楚可以用哪个图来替换它，因为 P 与毛毛虫图其余部分之间的相互作用可能很复杂。为了刻画一个部分与毛毛虫图其余部分之间的相互作用，我们定义毛毛虫图本质的概念。对于任何符合正确形式的毛毛虫图，毛毛虫图的本质都可以在多项式时间内计算得到。其关键思想是，本质描述了该毛毛虫图如何与从其左侧和右侧连接到它的其他毛毛虫图发生相互作用。

▶**定义 2.5**（毛毛虫图的本质）。设 C 为一个毛毛虫图，使得 C 的解紧打包包含 C 的所有顶点。设 C 的脊柱为 v_1、……、v_n，其中 v_1 是脊柱的左端点，v_n 是脊柱的右端点。

对于每个整数 $x \geq 0$，定义 C_x 为从 C 出发，在保持脊柱不变的情况下向 v_1 添加 x个额外悬挂顶点后得到的毛毛虫图。对于每个整数 $x \geq 0$ 以及任意 d-coc 集合 Y（其中 C_x），我们定义 $\beta_x(Y)$：若 $v_n \in Y$，则其值为 0；否则，$\beta_x(Y)$ 定义为包含 v_n的 $C_x - Y$ 的连通分量的顶点数。对于所有整数 $x \geq 0$，令 $\mathcal{D}_x$ 为 C_x的所有最小 d-coc 集合构成的集合。

相对于 d，C的本质是函数 γ：$[0, d+1] \to [0, d+1]$，其中

$$\gamma(x) = \begin{cases} \min\{\beta_x(Y) \mid Y \in \mathcal{D}_x\} & \text{if } x \leq d \text{ and } \mathsf{OPT}_d(C_x) = \mathsf{OPT}_d(C), \\ d+1 & \text{otherwise.} \end{cases}$$

在进一步讨论之前，我们先介绍本质的一些基本性质。函数 $f\colon [0, d+1] \to [0, d+1]$ 是非递减的，意指对于所有满足 $x < y$的$x, y \in [0, d+1]$，均有 $f(x) \geq f(y)$。

▶**引理 6.2.** 设 C 是一个毛毛虫图，使得 C 的解紧打包包含 C的所有顶点，并设 γ 是 C的本质。那么，γ 是一个非递减函数，满足 $\gamma(0) = 0$ 以及 $\gamma(d+1) = d+1$。此外，可以在多项式时间内计算 C 的本质 γ。

证明。 根据本质的定义，我们有 $\gamma(d+1) = d+1$。关于 $\gamma(0)$，注意到 $C_0 = C$，并且存在一个最小的 d-coc 个 C 的集合，它选取了最右侧脊柱顶点，因此 $\gamma(0)$ 确实为0。此外，由于对于任意 $y > x$，都是 C_y 的子图，因此函数 γ 必定是非递减的。

最后，我们需要说明，对于任意 $x \in [0, d]$，都可以在多项式时间内计算 $\gamma(x)$。为此，我们可以在多项式时间内构造 C_x，并在多项式时间内计算C_x的解紧打包。如果该打包比 $\mathsf{OPT}_d(C)$大，那么 $\gamma(x) = d+1$。否则，我们可以通过选取 C_x的解紧打包中每个打包子图的最右侧脊柱顶点，构造 C_x 的一个解 S。不难验证，这是所有最优解中使本质定义里的值 β_x 最小的解。因此，我们可以将 $\gamma(x)$设为$C_x - S$ 中包含 C_x的最右侧脊柱顶点的连通分量的顶点数；若该顶点属于 S，则设为 0。◀

现在，我们正式定义替换毛毛虫图这一操作；这是我们用于缩短脊长的归约规则中的核心操作。

28 通过到有界路径宽度的距离对 COC 进行核化的分界线

▶ **定义 6.3** (替换毛毛虫图)。 令 C_1 为 $G - M$ 的一个连通子图，且包含 $G - M$中至少一个脊顶点。令 C 为 $G - M$中包含 C_1的连通分量。令 C_2 为任意一个毛毛虫图。令 v_ℓ 为 C 中位于 C_1 左侧且最靠右的脊顶点（若不存在这样的顶点，则为 $\perp \notin V(G)$ ），令 v_r 为 C 中位于 C_1 右侧且最靠左的脊顶点（若不存在这样的顶点，则为 $\perp \notin V(G)$ ）。

然后，将 C_1 替换为 C_2 在 G 中的操作由以下过程描述：**1.** 以 G为起点，并删除 C_1。**2.** 将 C_2 添加到图中。**3.** 如果 $v_\ell \neq \perp$，则在 C_2 的最左侧脊柱顶点与 v_ℓ之间添加一条边。**4.** 如果 $v_r \neq \perp$，则在 C_2 的最右侧脊柱顶点与 v_r之间添加一条边。输出所得图。

因此，替换操作对应于一种自然操作：切除毛毛虫图 C_1 ，并将其替换为 C_2。接下来，我们证明，用本质为 γ 的任意毛毛虫图替换本质为 γ 的毛毛虫图，性质良好。

▶ 引理 *2.6*。设 (G, d, k, M) 是 *COC*/d+pw-*1* 的一个实例，且 P 是 $G - M$ 的连通合并子图。设 γ 为 P 的本质， P' 为本质为 γ 的任意毛毛虫图。通过将 P 替换为 P'，从 G 构造 G' 。如果存在一个 d-coc 集合 S ，它是 G 的子集且大小至多为 k，那么就存在一个 d-coc 集合 S'，它是 G' 的子集且大小至多为 $k' = k - \mathit{OPT}_d(P) + \mathit{OPT}_d(P')$。

证明。令 S 为大小至多为 k 的 G 的 d-coc 集合，并令 C 为包含 P 的 $G - M$ 的连通分量。此外，令 v_ℓ 为 C 中位于 P 左侧的最右侧脊柱顶点；若不存在这样的顶点，则令 v_ℓ 为 $\perp \notin V(G) \cup V(G')$。令 P 的最左侧脊柱顶点为 w_1， P 的最右侧脊柱顶点为 w_r；令 P' 的最左侧脊柱顶点为 w'_1， P' 的最右侧脊柱顶点为 w'_r。我们分两种情形讨论。

情形 1： $v_\ell = \perp$ 或 $v_\ell \in S$ 或 $|S \cap V(P)| >$ 最优值 $_d(P)$：令 S'_P 为 P' 的一个最小 d-coc 集合，其中选取了 P' 的最右侧脊柱顶点。注意，根据解紧致打包的性质，以及 P' 有一个包含其所有顶点的解紧致打包这一事实，这样的集合确实存在（而且 P' 必定有这样的解紧致打包，因为本质仅对这类毛毛虫图定义）。那么， $S' = (S \setminus V(P)) \cup \{v_\ell\} \cup S'_P$ 是 G'的一个 d-coc 集合，因为我们选取了 P'的最右侧脊柱顶点，以及（若存在） P'左侧的最右侧脊柱顶点；此外， P'中没有顶点与 M 相邻。此外，由于 S'_P 对于 P' 是最优的，这个集合 S' 至多包含 G' 的 $k' = k -$ 个顶点，即 P 的最优值 $_d$ 与 P' 的最优值 $_d$ 之和 $+$。

情况 2： $v_\ell \neq \perp$ **和** $v_\ell \notin S$ **和** $|S \cap V(P)| =$ 最优值$_d(P)$**：** 我们先总结这一情形的一个基本事实。

▷ 引理 6.4。S 选取了 P 的一个脊顶点，并且在 $G - S$ 中不存在一条从 w_1 到 w_r 、且只经过 P 中顶点的路径。

证明。令 H 为 P 的解紧打包中最左侧的图。若 S 未选择 P 的脊顶点，那么由于 $S \cap V(P)$ 是 P 的一个最小 d-coc 集合， S 恰好选择 H 的一个顶点，且该顶点是悬挂顶点。由于 $|V(H)| \geq d + 1$， S 必须选择 H 的所有邻点；特别地， S 必须选择 v_ℓ，这与 $v_\ell \notin S$ 矛盾。因此， S 选择了 H 的一个脊顶点。此外，在 $G - S$ 中不存在从 w_1 到 w_r 、且仅经过 P 中顶点的路径，因为任何这样的路径都必须包含所选的脊顶点。◁

若 $w_1 \in S$，令 x 为 d。否则，若 $w_1 \notin S$，令 L 为 $G - S$ 中包含 w_1 的连通分量，并令 L_P 为 $P - S$ 中包含 w_1 的连通分量。显然，L_P 是 L 的子图，因此 $|L| = |L_P| + |L - L_P|$。设 $x = |L| - |L_P|$，并注意到 $|L_P| = |L| - x \leq d$，因为 $|L| \leq d$ 且 $x \geq 0$。

令 y 为 0，如果 $w_r \in S$。否则，如果 $w_r \notin S$，令 R 为$G - S$中的 w_r 的连通分量，并令 R_P为 $P - S$中的 w_r 的连通分量。我们有， R_P是R的子图。令 $y = |R_P|$，并注意到我们有 $|R| = y + |R| - |R_P| \leq d$。

▷ 断言 6.5。我们有 $\gamma(x) \leq y$。

证明。考虑图 P_x，它与 P相同，只不过 w_1 多了 x 个悬挂顶点。于是， $S \cap V(P)$ 是 P_x 的一个 d-coc 集合，因为在 $P - S$ 中，每个不包含 w_1 的连通分量大小至多为 d；如果存在一个包含 w_1 的连通分量 L_P，那么它的大小为 $|L_P| = |L| - x$。因此， $S \cap V(P)$ 表明 $\gamma(x) \leq y$。◁

由于 $P_\prime$与 P具有相同的本质，因此存在一个针对 $P_{\prime x}$的最小规模解 $S_{P'}$，使得 w'_r 被选中（若 $y = 0$，则必然如此），或者 w'_r 位于 $P_{\prime x} - S_{P'}$的一个大小至多为 y的连通分量中。令 $S' = (S \setminus V(P)) \cup S_{P'}$。显然有$|S'| \leq k'$，因此还需证明 S' 是 G'的一个 d-coc 集合。为此，我们先收集一些关于路径的其他事实。

▷ 断言 6.6. 不失一般性， $P' - S'$中不存在从 w'_r 到 w'_1 的路径。

证明。 不失一般性， $S_{P'}$不包含 $P_\prime$的任何悬挂顶点；并且由于 $|V(P_\prime)| \geq d + 1 S_{P'}$ 必须包含一个脊顶点，这会破坏这类路径。◁

▷ 断言 6.7. 不存在一条从顶点 $v \notin V(P')$ 到 w'_1（位于 $G' - S'$ 中）的路径，且该路径包含 $P' - \{w'_1\}$ 中的一个顶点。

证明。 假设存在这样一条路径。那么，在到达 w'_1 之前，该路径会先到达 P' 的某个顶点 $w' \neq w'_1$。由于 $P_\prime$ 中只有 w'_1 和 w'_r 这两个顶点可能有 P' 之外的邻点，因此该路径到达的第一个 P' 顶点必定是 w'_r。特别地，它必须通过 w'_r 的邻点 v_r 到达 w'_r，而该邻点是一个不属于 P' 的脊顶点。然而，不经过 v_r 就不存在从 w'_r 到 w'_1 的路径，因为根据断言 6.6，不存在一条仅使用 P' 中顶点、从 w'_r 到 w'_1 的路径。因此，该路径会两次经过 v_r，这是不可能的。◁

由于 w'_1 和 w'_r 是 P' 中唯一邻接到 P'外部顶点的顶点，因此 $G' - S'$ 中唯一可能大小超过 d 的连通分量，是包含 w'_1和 w'_r的那些连通分量。

包含 w'_1的连通分量。假设 $G' - S'$ 的某个连通分量 L'包含 w'_1。注意，这意味着 $w_1 \notin S$；否则，由于 $S_{P'}$ 是 $P_{\prime x} = P_{\prime d}$的一个$d$-coc集合，可以假设 $w'_1 \in S_{P'}$。令 $L'_{P'}$为包含 w'_1 的 $P_\prime - S'$的连通分量。由于 $S_{P'}$是 $P_{\prime x}$的一个 d-coc集合，我们有 $|L'_{P'}| \leq d - x$。我们将构造一个单射函数 δ，为每个顶点 $v \in L' - L'_{P'}$ 指定一个顶点$v' \in L - L_P$。于是，这个函数 δ证明了 $|L'| - |L'_{P'}| \leq |L| - |L_P| = x$，因此也证明了 $|L'| = |L'_{P'}| + |L'| - |L'_{P'}| \leq d - x + x = d$。

令 v 为 $L' - L'_{P'}$的一个顶点。那么，在 $G' - S'$ 中存在一条从 v 到 w'_1 的路径，而这条路径在 $P' - S'$中不存在。

30 通过到有界路径宽度的距离实现 COC 核化的分界线

如果 $v \notin P'$，那么断言 6.7 表明，从 v 到 w_1' 的路径不包含 $P\prime - \{w_1'\}$中的顶点。因此，在 $G - P$中存在一条从 v到 w_1的路径，这表明 v也属于 L。但显然， $v \notin V(L_P)$，因为 $V(L_P) \subseteq V(P)$。因此， $v \in L - L_P$，我们可以设 $\delta(v) = v$。

否则， $v \in P'$。因此，从 v 到 w_1' 的路径包含顶点 w_r'，否则 v 也会属于 $L_{P'}'$。令 $R_{P'}'$ 为 w_r' 在 $P' - S'$ 中的连通分量。由于 v 是任意的， P' 中属于 $L' - L_{P'}'$ 的顶点恰好就是 $R_{P'}'$ 中的顶点。根据本质 γ 的性质以及我们对 $S_{P'}$ 的选择，可得 $|R_{P'}'| \leq y = |R_P|$。此外，由于 v 存在，我们有 $y \geq 1$，因此顶点 $w_r \notin S$。在 $G' - S'$ 中存在一条从 w_r' 到 w_1' 的路径，但根据断言 6.6，在 P' 内不存在从 w_r' 到 w_1' 的路径，因此在 $G - S$ 中必定存在一条从 w_r 到 w_1 的路径。因此， R_P 的每个顶点都属于 L。然而，根据引理 6.4，在 $P - S$ 中不存在从 w_r 到 w_1 的路径，因此 R_P 中没有顶点属于 L_P。由于 $|R_P| \geq |R_{P'}'|$，我们可以将 $R_{P'}'$ 的顶点单射映射到 R_P 的顶点，并满足 δ。

包含以下顶点的连通分量： w_r'。 现在，假设 $G' - S'$ 中存在一个包含 w_r' 的连通分量R'。这意味着 $y \geq 1$，因此 $w_r \notin S$。我们还可以假设 R' 不包含 w_1'，因为我们已经处理了包含 w_1' 的连通分量。令 $R_{P'}'$ 为 $P\prime - S'$ 中 w_r' 所在的连通分量。根据我们对解$S_{P'}$的选择，有$|R_{P'}'| \leq y$。与对连通分量 L' 的证明类似，我们将使用函数 δ，把 $R' - R_{P'}'$ 的每个顶点单射地映射到 $R - R_P$ 的一个顶点。事实上，如果存在这样的函数 δ，那么我们就会得到$|R'| = |R_{P'}'| + |R'| - |R_{P'}'| \leq y + |R| - |R_P| \leq d$。

设 v 是 $R' - R_{P'}'$的一个顶点。那么， v 有一条通往 w_r' 的路径，而这条路径在 $P' - S'$ 中不存在。此外，由于 w_1' 不属于 R'，这条路径不包含 w_1'。因此， v 不可能属于 P'。于是，从 v 到 w_r' 的路径可导出 $G - S$ 中一条从 v 到 w_r 的路径。我们可以映射 $\delta(v) = v$ 来完成证明。◀

为了用一个具有相同本质 γ 的较小图替换一个具有特定本质 γ 的图，我们还必须能够快速计算出本质为 γ 的小图。这项任务极具挑战性，但正如下一个引理所示，这是可行的。

▶ **引理 2.12**。 存在一个算法，以一个非递减函数 $\gamma\colon [0, d+1] \to [0, d+1]$（满足 $\gamma(0) = 0$, $\gamma(d+1) = d+1$）为输入，并输出一个毛毛虫图 C，其本质为 γ，使得 C 的解紧打包 $\boldsymbol{C}$ 包含 C 的所有顶点，并且在关于 d 的多项式时间内得到 $|\boldsymbol{C}| \leq d^3$。

证明推迟到第 7 节；我们还将在该节讨论毛毛虫图的本质与抽象代数之间有趣的联系。这个引理的一个有趣之处在于，对于任何概念上可能的本质，都存在一个具有该本质的小型毛毛虫图。

我们打算利用引理 2.6 和 2.12，将 $G - M$ 的一些子图替换为具有相同本质的更小毛毛虫图；然而，这些连通分量可能与调节集存在的连接会带来一些额外困难。因此，在陈述实际的归约规则之前，我们需要弄清楚 $G - M$ 中哪些部分可以安全地替换，为此我们再次利用图打包。

▶ **引理 6.8**。 设 C 是 $G - M$ 的一个连通子图，$\boldsymbol{P}$ 是 C 的一个两两顶点不相交的连通子图打包，并且 $\boldsymbol{P}$ 中的每个图都包含 C 的至少一个脊顶点。

对于任意整数 $c_1, c_2 \geq 1$，存在一个集合 $\tilde{\boldsymbol{P}} \subseteq \boldsymbol{P}$，满足 $|\tilde{\boldsymbol{P}}| \geq |\boldsymbol{P}| - |M| \cdot (c_1 + c_2)$，使得对于每个 $P \in \boldsymbol{P}$，以及每个 $v \in N(P) \cap M$顶点 v，在 P左侧的 $\boldsymbol{P}$ 中的图里至少有 c_1 个邻居，在 P右侧的 $\boldsymbol{P}$ 中的图里至少有 c_2 个邻居。

证明。 由于 $\mathcal{P}$ 中的每个图都至少有一个脊顶点，因此由脊柱确定了它们之间的自然顺序。对于每个 $v \in M$，我们执行以下标记过程。顶点 v 会标记 c_1 个最靠左的 $\mathcal{P}$ 中与其 v 有边相连的图，并且还会标记 c_2~个最靠右的 $\mathcal{P}$ 中与其 v 有边相连的图。我们的集合$\mathcal{P}$ 由 $\mathcal{P}$中所有未标记的图组成。由于 M中的每个顶点至多标记 c_1+c_2 个 $\mathcal{P}$中的图，因此有 $\left|\tilde{\mathcal{P}}\right| \geq |P| - |M| \cdot (c_1 + c_2)$。

~现在，考虑某个 $P \in \mathcal{P}$。那么，对于每个 $v \in N(P) \cap M$，我们知道 v没有标记P，因此 v 与 c_1 个位于$\mathcal{P}$ 中 P左侧的图，以及 c_2 个位于$\mathcal{P}$ 中 P右侧的图相连。◀

我们可以利用这一事实，找到 (d^3+1)-合并打包中的特定子图。为方便起见，我们定义 $c_{\mathsf{p}} = (d^3+1) \cdot (|M| \cdot 2d+1)$。我们将利用 c_{p}-合并打包。

▶ **引理 6.9**。 设 C 是 $G - M$的一个连通分量，$P_{c_{\mathsf{p}}}$ 是C的 c_{p}-合并打包，$P_{(d^3+1)}$ 是 C的($\{d^3+1)$-合并打包。那么，对于 $P_{c_{\mathsf{p}}}$ 中的每个图 G_1，都存在 G_1 的一个子图 G_2，使得

1. $G_2 \in \mathcal{P}_{(d^3+1)}$，且2. 每个 $v \in N(G_2) \cap M$ 在 d 中有邻点 G_1，这些邻点位于 G_2左侧，3. 每个 $v \in N(G_2) \cap M$ 在 d 中有邻点 G_1，这些邻点位于 G_2右侧。此外，给定 G_2，我们可以在多项式时间内找到 G_1。

证明。 令 G_1 为一个任意的属于$\mathcal{P}_{c_{\mathsf{p}}}$的图。于是，由于 $c_{\mathsf{p}} = (d^3+1) \cdot (|M| \cdot 2d+1)$，可知 G_1 本身由$\cdot\ 2d+1$ 个属于$\mathcal{P}_{(d^3+1)}$的图组成。

将引理 6.8 应用于 $G' - M$ 的分量 C，其中包含这些 $|M| \cdot 2d+1$ 个打包子图，可知存在一个图 G_2 属于$\mathcal{P}_{(d^3+1)}$，使得每个 $v \in N(G_2) \cap M$ 在 G_1 中都有邻点d，这些邻点位于 G_2左侧；并且在 G_1 中都有邻点 d，这些邻点位于 G_2右侧。我们还可以通过遍历构成 G_1 的所有 $|M| \cdot 2d+1$ 个打包子图，并检查它们是否满足所需性质，从而在多项式时间内找到 G_2。◀

现在，我们终于可以陈述所用的归约规则，以减少 $G - M$ 的连通分量中的脊顶点数量。

▶ 规约规则 *2*。设 (G, d, k, M) 是 *COC/*d+pw-*1* 的一个实例。设 X 为该实例的块集合，$P_{c_{\mathsf{p}}}$ 为 c_{p}-合并打包 $G - M$，$P_{(d^3+1)}$ 为 (d^3+1)-合并打包 $G - M$。若 $\left|P_{c_{\mathsf{p}}}\right| > |M|^2 \cdot (|M| + 2(d-1))$，则此规则适用。执行以下步骤。*1.* 对每个块 $X \in X$，令 $P^X_{c_{\mathsf{p}}} = \{\hat{P} \in P_{c_{\mathsf{p}}} \mid conf_d(X, \hat{P}) > 0\}$。对每个 $X \in X$，标记 $P^X_{c_{\mathsf{p}}}$ 中任意的 $min(|M| + 2(d-1)\ \left|P^X_{c_{\mathsf{p}}}\right|)$ 个图。$\hat{}$*2.* 固定一个任意未标记的 $P \in P_{c_{\mathsf{p}}}$。*3.* 固定一个 $\hat{P}$ 的任意子图 P，其中 $P \in P_{(d^3+1)}$，使得每个 $v \in N(P) \cap M$ 在 P 中有 $d\hat{}$个邻居位于 P 左侧，并有 d 个邻居位于 P 右侧。*4.* 计算 P 的本质 γ。*5.* 计算一个图 P_γ，使其具有大小至多为 d^3 、本质为 γ 的解紧打包。*6.* 通过将 P 替换为 P_γ 来构造 G'，并令 $k' = k - OPT_d(P) + OPT_d(P_\gamma)$。*7.* 输出实例 (G', d, k', M)。

接下来，我们论证该规则可以在多项式时间内执行，且这些步骤定义明确。

▶**引理 6.10**。 规约规则 *2* 可以在多项式时间内执行，且每一步都有明确的定义。此外，若规则执行前的实例为(G, d, k, M)，执行后的实例为(G', d, k', M)，则 $G' - M$ 的解紧打包严格小于 $G - M$ 的解紧打包。

证明。 显然，步骤 1 可以在多项式时间内完成，因为我们可以在多项式时间内计算$\mathsf{conf}_d(X, \hat{}P)$，对于每个 $\hat{} P \in \mathcal{P}_{c_\flat}$ 以及 $X \in \mathcal{X}$。

步骤 2 中的图 $\hat{}P$是定义良好的，因为如果该规则适用，则 $|\mathcal{P}_{c_\flat}| > |M|^2 \cdot (|M| + 2(d-1))$；对于每个 $X \in \mathcal{X}$，我们至多标记 $|M| + 2(d-1)$ 个$\mathcal{P}_{c_\flat}$中的图，且 $|\mathcal{X}| \leq |M|^2$。步骤 3 中固定的图 P 由引理 6.9 保证存在。

由于 P 是$\mathcal{P}_{(d^3+1)}$-合并打包的一部分， P 的解紧打包包含 P的所有顶点。因此， P 具有一个本质，我们可以利用引理 6.2 在多项式时间内计算出来。因此，步骤 4 也可以在多项式时间内完成。

通过应用引理 2.12，可以足够快速地执行步骤 5。

显然，步骤 6 和 7 也可以在多项式时间内完成，因为 COC 可以在毛毛虫图上于多项式时间内求解。

要看出该规则会缩小解紧打包的规模，注意图P有一个包含 d^3+1 个图的解紧打包，而 P_γ 的解紧打包仅包含 d^3 个图。 $G - M$ 的解紧打包与$G' - M$的解紧打包相同，区别仅在于将 P的打包替换成 P_γ的打包。因此，该规则应用会缩小解紧打包的规模。◀

现在，我们着手证明该规则是安全的。为此，我们先引入两个进一步的辅助引理。

$\hat{}$▶**引理 6.11.** 设 P 是 $G - M$的两两顶点不相交的连通合并子图的一个打包。对于每个块 $X \in \boldsymbol{X}$，令 $\hat{}\boldsymbol{P}^X = \{H \in \hat{}\boldsymbol{P} \mid \mathsf{conf}_d(H, X) > 0\}$。那么，存在 G 的一个最小 d-coc 集合 S，它与每个块 X都相交，其中 $|\hat{}\boldsymbol{P}^X| \geq |M| + 2(d-1)$。

证明。令 S 为 G 的一个最小 d-coc 集合，并令 X 为一个块，使得 $|\hat{}\mathbf{P}^X| \geq |M| + 2(d-1)$，且假设 $S \cap X = \emptyset$。由于 $|X| \leq 2$， X 至多有 $2(d-1)$ 个邻点未被 S 选中。因此，在 $\hat{}\mathbf{P}^X$ 的图中， X 的所有邻点都被选中。于是，在这些图的每一个中， H 都满足 $S \cap V(H) >$ 最优值$_d(H)$。特别地，由于每个这样的图 H 都由 $G - M$ 的解紧打包 P 中的一些图构成，因此存在 H 的一个子图 ~ ~ ~$H' \in \mathbf{P}$，使得 S 至少选取了 H' 的两个顶点。总之，我们可以找到 P 中若干个顶点不相交的图，使得 S 在每个图中都至少选取两个顶点。这说明 $S \geq |M| +$最优值$_d(G - M)$。因此，选取 M 中所有顶点以及 $G - M$ 的一个最小d-coc 集合的解同样是最优的，并且与所有块相交。◀

▶ **引理 6.12。** 设 $S \subseteq M$ 是一个集合， H 是 $G - M$ 的一个子图，并且对于所有满足 $\mathsf{conf}_d(X, H) > 0$的块 $X \in \boldsymbol{X}$，均有 $S \cap X \neq \emptyset$。那么，存在 H 的一个最小 d-coc 集合 S_H，它选取了 $N(M \setminus S) \cap V(H)$中的所有顶点。

证明。为导出矛盾，假设不存在 H 的最小 d-coc 集合能够选取 $B = N(M \setminus S) \cap V(H)$ 中的所有顶点。那么， B 是 H的一个 d-阻塞集。由于毛毛虫森林的极小 d-阻塞集大小至多为二，我们可找到一个集合 $B' \subseteq B$，它也是 H 和 $|B'| \leq 2$的 d-阻塞集。令 $B' = \{b_1, b_2\}$，我们可以有 $b_1 = b_2$。设 m_1、 m_2 分别为 b_1和 b_2 的邻点，且它们属于 $M \setminus S$。那么 B' 证明了 $\mathsf{conf}_d(H, \{m_1, m_2\}) > 0$。然而，$\{m_1, m_2\}$是一个块，这与我们的假设相矛盾： S命中所有与 H存在非零冲突的块。◀

注意，这两个引理背后的核心思想最初由 Jansen & Bodlaender[39] 用于减少连通分量的数量；不过，我们通过采用解紧打包对其进行了创新，使引理 6.11 适用于并非整个连通分量的图。现在，我们可以完整证明该归约规则的安全性了。

▶**引理 6.13.** 规约规则 *2* 是安全的。

证明。 设 $(G,\ d,\ k,\ M)$ 为应用该规则前的实例，$(G',\ d,\ k',\ M)$ 为应用该规则后的实例。设 P 为被P'替换的毛毛虫图，其中 γ 是 P和 P'的本质。设 C 为包含 P的 $G-M$ 的连通分量， C' 为包含 P'的 $G'-M'$ 的连通分量。我们分别证明安全性的两个方向。

▷ 断言 6.14. 如果输入实例是一个是实例，那么输出实例也是一个是实例。

证明。 假设输入实例是一个是实例，并令 S 为一个 d-coc 集合，其元素属于 G 且大小至多为 k。由引理 2.6 可直接得出，存在一个 d-coc 集合 S'，其大小至多为 k'，对应于 G'。◁

现在我们可以继续证明的反向。正向证明中， P' 没有连接到调节集的边这一点非常方便，因为这让我们可以立即使用引理 2.6。然而，由于 P 可能有这样的边，因此证明这个方向稍微复杂一些。

▷ 断言 6.15. 如果输出实例是一个是实例，那么输入实例也是一个是实例。

证明。 令 S'是一个 d-coc集合， G'其大小至多为 k'。令 $M'=N_G(P)\cap(M\setminus S')$，也就是说，$M'$ 是与之相邻的未选调节集顶点的集合 P。

~ ~ ~情形 1： $M'=\emptyset$：令 $G'=G'-(S'\cap M)$，并令 $G=G-(S'\cap M)$。显然， $S'\cap V(G')$ 是一个~ d-coc集合， G'其大小至多为 $k'-|S'\cap M'|$。此外，由于 $M'=\emptyset$图 P在 G 中没有 ~ ~ ~邻居。因此，我们可以在 G' 中用 P 替换 P'，从而构造出 G。注意， $G'-M=G'-M$ 的解紧打包与 $G-M=G-M$ 的解紧打包~几乎完全相同，唯一的区别是， P 的解紧打包被~ P' 的解紧打包替换。因此， P' 是 $G'-M$ 的连通合并子图。~ ~根据引理 2.6，图 G 因而有一个 d-coc 集合 S，其大小至多为 $k-|S'\cap M'|$。因此，$S=S\cup(S'\cap M)$ 是 G 的一个 d-coc 集合，其大小至多为 k。

情形 2： $M'\neq\emptyset$：设 $\mathcal{P}^{c_p}$是 c_p-合并打包 $G-M$，而 $\mathcal{P}^{(d^3+1)}$是（d^3+1）-^ 合并打包 $G-M$。此外，设 P是$\mathcal{P}^{c_p}$中包含 P 的图，并回想一下， P 实际上是打包$\mathcal{P}^{(d^3+1)}$中的一个图。

由于 M 非空，且 M 的每个顶点在 P 的左侧和右侧都有至少一个邻点，因此 P 中存在一个位于 G 中 ^P 左侧的最右侧脊柱顶点，将该脊柱 v_ℓ顶点记为……；而 P 中存在一个位于 G 中 P 右侧的最左侧脊柱顶点，将该脊柱顶点记为 v_r。此外，将 P 的最左侧脊柱顶点记为 w_1，将P' 的最左侧脊柱顶点记为 w_1'。将 P 的最右侧脊柱顶点记为 w_r，将 P' 的最右侧脊柱顶点记为 w_r'。

令 M 为被归约规则标记的 P^{c_p} 中的图的集合。注意， $G'-M$ 的解紧打包可由 $G-M$ 的解紧打包得到：将 P 的解紧打包替换为 P'的解紧打包。因此，M 中的所有图都是 G'的合并子图。在该归约规则中，图 ^P 未被标记，因此对于每个满足 $\mathrm{conf}_d(X,\hat{P})>0$ 的块 X，我们至少标记了 $\mathrm{P}^X_{c_p}=\{T\in\mathrm{P}_{c_p}\mid\mathrm{conf}_d(X,T)>0\}$ 中的$|M|+2(d-1)$个图，而这些图都不是 ^P。因此，对于每个满足 $\mathrm{conf}_d(X,\hat{P})>0$的块 X，都有 $|\mathrm{P}^X|>|M|+2(d-1)$。现在我们应用

34 通过到有界路径宽度的距离实现 COC 核化的分界线

引理 6.11 于打包$\mathcal{M}$ 和输出实例。这说明我们可以假设 S' 与所有满足 $\mathsf{conf}_d(X, \hat{}P) > 0$的块 X 相交。然后，将引理 6.12 应用于输入实例中的集合$S' \cap M$和 $\hat{}P$，可得存在一个最小的 d-coc集合 $S_{\hat{}P}$，它属于 $\hat{}P$，并选取 $N(M') \cap V(\hat{}P)$中的所有顶点。

为 G 构造一个小型解。回想一下， M' 的每个顶点至少有 P 中 P 左侧的 d个顶点作为邻居，而且至少有 P 中 P 右侧的 d 个顶点。对于任意一个 $m \in M'$，令 L_m 为 m 在 P 左侧、属于 P 的任意 d 个邻居，令 R_m 为 m 在 P 中、位于 P 右侧的任意 d 个邻居。由于未选中 m， S' 必须从 L_m 中至少选一个顶点，并从 R_m 中至少选一个顶点。令 n_ℓ 为 L_m 中 S' 选中的一个顶点，并令 n_r 为 R_m 中 S' 选中的一个顶点。

我们知道， n_ℓ 和 n_r 也都属于 $S_{\hat{}P}$，因为 $S_{\hat{}P}$ 会选中 N 中的所有顶点（$M'\cap V(\hat{}P)$）。S' 与 $S_{\hat{P}}$ 之间的这一共同点，正是我们用来为输入实例构造一个小型解 S 的依据。

设$\hat{}P$的解紧打包为$\{P_1, \ldots, P_t\}$，其中 P_i位于所有满足 $j > i$ 的图$\hat{}P_j$ 的左侧，对任意 i 均如此。回顾一下， P 的解紧打包包含$\hat{}P$的所有顶点。于是，对于$\hat{}P$的任意顶点 v，令 $\mu(v)$为满足 $v \in V(P_{\mu(v)})$的整数。现在我们关注 $P^* = \hat{}P\left[\bigcup_{i\in[\mu(n_\ell)+1,\mu(n_r)-1]} V(P_i)\right]$。换言之，$P^*$是 P的子图$\hat{}$，由 $P_1, \ldots, P_t$中位于包含 n_ℓ的图与包含 n_r的图之间的所有图构成。注意，P^* 包含 P 作为子图。定义 $P^{*\prime}$ 为 G'中对应于 P^* 的图，也就是说，将P^*中的 P 替换为 P'后得到的图。显然， $P^{*\prime}$是 G'的子图。由于 P、P'、P^*都具有包含其所有顶点的解紧打包， $P^{*\prime}$ 也有这样的打包。特别地，$\mathsf{OPT}_d(P^{*\prime}) = \mathsf{OPT}_d(P^*) - \mathsf{OPT}_d(P) + \mathsf{OPT}_d(P')$。

我们令 $S = (S' \setminus V(P^{*\prime})) \cup (V(P^*) \cap S_{\hat{}P})$，现在要证明 S是 d-coc的一个大小至多为 k的 G集合。首先，我们证明 S的大小至多为 k。注意， $S_{\hat{}P} \cap V(P^*)$ 是 P^*的一个最小 d-coc集合，因此 $|V(P^*) \cap S_{\hat{}P}| =$最优值$_d(P^*)$。此外，我们知道$|S' \cap V(P^{*\prime})| \geq$最优值$_d(P^*)$ $-$最优值$_d(P)$ $+$最优值$_d(P')$。由此可直接得出 $|S| \leq k$。因此，要完成证明，只需说明 S 实际上是 G的一个 d-coc 集合。

$\hat{}$ 回想一下， $P_1, \ldots, P_t$是 P的解紧打包中的图，因此，$\hat{}$ P 的任意最小 d-coc 集合都恰好包含每个这样的图 P_i中的一个顶点。利用 $S_{\hat{}P}$是 P的一个最小 d-coc 集合，且它选取了 M'的所有邻点这一事实，我们看到 $S_{\hat{}P}$仅包含 $P_{\mu(n_\ell)}$中的顶点 n_ℓ，这意味着 $P_{\mu(n_\ell)}$中只有顶点 n_ℓ 与 M'中的某个顶点相邻。类似地，$P_{\mu(n_r)}$ 中只有顶点 n_r 与 M'中的某个顶点相邻，并且$S_{\hat{}P}$仅包含 $P_{\mu(n_r)}$中的顶点 n_r。将 P^* 的最左侧脊顶点记为 v^*_ℓ，将 P^* 的最右侧脊柱顶点记为 v^*_r。

现在，考虑 n_ℓ 是悬挂顶点的情况。此时， v^*_ℓ 必须由 $S_{\hat{}P}$ 选中，否则， v^*_ℓ 和 $P_{\mu(n_\ell)}$

</style} 中至少 个未选中的顶点就已经构成一个大小大于 、完全位于 内的 $\hat{}$ 连通分量。出于同样的原因，如果 是悬挂顶点，那么 $\hat{}P$ 就包含 P^* 中的 v^*_r。

根据构造可知， $S \cap V(P^*)$ 是 P^* 的一个 d-coc 集合，且 $S \cap V(G - P^*)$ 是 $G - P^*$ 的一个 d-coc 集合。因此， $G - S$ 中任何大小大于 d 的连通分量都必须同时包含 P^* 和 $G - P^*$ 中的顶点。设 Z 是 $G - S$ 的一个连通分量，其中同时包含 P^* 和 $G - P^*$ 的顶点。由于 $S_{\hat{}P}$ 选中了 $\hat{}P$ 中 M' 的所有邻点，因此 S 选中了 P^* 中所有在 M' 中有邻点的顶点。这意味着，Z 必须包含 v^*_ℓ 或 v^*_r，因为 P^* 中没有其他顶点能在 $G - S$ 中与 P^* 之外的顶点相邻。

假设 v^*_ℓ 不在 S 中，并考虑 $G - S$ 中一条从 v^*_ℓ 向左延伸的路径。如果 n_ℓ 是悬挂顶点，那么就会选中 v^*_ℓ，这与假设不符。否则， $n_\ell \in S$ 是

一个脊顶点，而 n_ℓ 是 $P_{\mu(n_\ell)}$ 中唯一一个在 M' 中有邻点的顶点。因此，这条路径只能包含 $P_{\mu(n_\ell)}$ 中的顶点以及顶点 v^*_ℓ：由于脊顶点 n_ℓ 已被选中，这条路径不能继续向左延伸，也无法到达 M'。类似地，从 v^*_r 出发向右延伸的路径只能包含 $P_{\mu(n_r)}$ 中的顶点以及顶点 v^*_r。由于 Z 中的任意顶点 w 都有一条通往 v^*_ℓ 或 v^*_r 的路径，这说明 Z 的所有顶点都是 $V(P^*) \cup P_{\mu(n_\ell)} \cup P_{\mu(n_r)}$ 的一部分。

现在，令 $\tilde{P} = \hat{P}\left[\bigcup_{i\in[\mu(n_\ell),\mu(n_r)]} V(P_i)\right]$。图 $\tilde{P}$ 与 P^* 类似，但还包含图 $P_{\mu(n_\ell)}$ 和 $P_{\mu(n_r)}$。根据上述详细说明，我们发现 Z 的每个顶点都是 P 的一个 $\tilde{}$ $\tilde{}$ $\tilde{}$ 顶点。此外，根据我们对 S 的选择以及 $n_\ell, n_r \in S\hat{}_P \cap S'$ 这一事实，有 $S\hat{}_P \cap V(P) \subseteq S \cap V(P)$。因此，$Z$ 的顶点也属于 $P - S\hat{}_P$ 中同一个连通 $\tilde{}$ $\tilde{}$ $\tilde{}$ 分量；并且由于 $S\hat{}_P \cap V(P)$ 是 P 的一个 d-coc 集合，我们有 $|Z| \leq d$。◁

结合断言 6.14 和 6.15 即可得出该引理。 ◀

6.1 核化算法

在给出我们的核化算法之前，最后我们陈述 Xiao 关于 COC 的一个结果，其中 COC 以解的大小 k 与整数 d 之和为参数，我们将其记作 COC/d+k。 [51]

▶ **定理 6.16** (由 [51, 定理 2]得出)。 $\mathrm{COC}/d{+}k$ 存在一个核，其顶点数为 $O(dk)$。

现在我们可以描述核化算法，并证明约简实例规模较小。

▶ **主定理 1**。 问题 $\mathrm{COC}/d{+}\mathtt{pw}$-1 存在一个核，其顶点数为 $O(d^7|M|^3 + d^6|M|^4)$。对于每个整数 $d \geq 1$，问题 d-$\mathrm{COC}/\mathtt{pw}$-1 存在一个核，其顶点数为 $O(|M|^4)$。

证明。 显然，我们可以在毛毛虫图上于多项式时间内求解 COC，$\mathcal{G}$ 具有遗传性，并且根据定理 5.2，毛毛虫森林的最小 d-阻塞集的大小至多为二，因此我们可以应用推论 4.14 中的算法来减少连通分量的数量。类似地，根据引理 6.10，我们可以在多项式时间内执行归约规则 2。

设 (G, d, k, M) 为输入实例。若 $|M| = 0$，则 G 是毛毛虫森林，因此我们可以在多项式时间内求解 COC，并根据 (G, d, k, M) 是是实例还是否实例，输出一个平凡的是实例或平凡的否实例。

否则，我们有 $|M| \geq 1$。我们首先反复应用规约规则 2，直到其不再适用。然后，我们应用推论 4.14 的算法（本质上就是归约规则 1）。根据引理 6.10，归约规则 2 的运行次数为多项式级，因为每次应用归约规则 2 都会缩小解紧打包的规模。这两条规则都不会改变值 d 或调节集 M。设 (G_2, d, k_2, M) 为此时的实例。若 $k_2 < 0$，我们输出一个平凡的否实例。否则，令 $\mathcal{C}$ 为 $G_2 - M$ 的连通分量。由推论 4.14 可得 $|\mathcal{C}| \leq ((d-1)\cdot 2+1)\cdot |M|^2 \leq 2d|M|^2$。令 $\mathcal{P}_{c_\mathsf{p}}$ 为 $G_2 - M$ 的 c_p-合并打包，并注意到我们有 $|\mathcal{P}_{c_\mathsf{p}}| \leq |M|^2 \cdot (|M| + 2(d-1)) \leq |M|^3 + 2d|M|^2$，否则鉴于推论 4.14 的算法不会增大解紧打包的规模，规约规则 2 仍然适用。还需回顾，$c_\mathsf{p} = (d^3+1)\cdot(|M|\cdot 2d + 1) \leq 6d^4|M|$。

令 C 为 $\mathcal{C}$ 的任意一个连通分量。注意，C 中不属于某个图 $\mathcal{P}_{c_\mathsf{p}}$ 的脊顶点少于 $c_\mathsf{p}\cdot(d+1)$，否则就可以严格增大 $\mathcal{P}_{c_\mathsf{p}}$ 的规模，

这是不可能的。此外，注意 $\mathcal{P}_{c_\mathsf{p}}$ 中的每个图至多包含 $(d+1)\cdot c_\mathsf{p}$ 个脊顶点。因此， $G_2 - M$ 的脊柱上的顶点数有界，

$$\begin{aligned}(|\mathcal{C}| + |\mathcal{P}_{c_\mathsf{p}}|) \cdot (d+1) \cdot c_\mathsf{p} &\leq (2d|M|^2 + |M|^3 + 2d|M|^2) \cdot (d+1) \cdot 6d^4|M| \\ &\leq 48d^6|M|^3 + 12d^5|M|^4.\end{aligned}$$

现在，令 $M\prime = M \cup V(\text{脊柱}(G_2 - M))$，也就是说，我们通过将$G_2 - M$的脊柱加入调节集来构造 M'。由于 $G_2 - M'$是一个独立集，因此 $M\prime$是 G_2的一个 d-coc集合。因此，当 $k \geq 48d^6|M|^3 + 13d^5|M|^4 \geq |M'|$时，该实例是一个是实例，此时我们输出一个平凡的是实例。否则，我们有 k至多为 $48d^6|M|^3 + 13d^5|M|^4$，因此可以应用定理 6.16 的核，构造一个具有$O(d^7|M|^3 + d^6|M|^4)$个顶点的等价实例，并由我们的核输出。

最后，我们注意到，由于所用的预处理过程都不会改变 d 的值（定理 6.16 中 Xiao 给出的核也满足这一点），该算法也适用于问题 d-COC/pw-1，其中 d 是一个固定常数。◀

7 毛毛虫图的本质与幺半群

现在我们来介绍引理 2.12 的证明。该引理指出，对于任意可能的毛毛虫图本质，我们都能快速计算出一个度较低的小型毛毛虫图，使其具有该本质。有趣的是，利用毛毛虫图的本质，我们可以定义一种幺半群，并用代数术语描述我们要寻找的对象类型。我们先定义本质幺半群。

▶**定义 2.7**（本质幺半群）。设 d 是一个正整数，M_d 是所有非递减函数的集合 γ：$[0, d+1] \to [0, d+1]$，其中 $\gamma(0) = 0$，且 $\gamma(d+1) = d+1$。令 id^d： $[0, d+1] \to [0, d+1]$ 为恒等函数。那么，$(M_d, \circ, \mathsf{id}^d)$ 是本质幺半群（关于 d），其中 $\circ$ 是函数复合操作。

不难验证，$(\mathcal{M}_d, \circ, \mathsf{id}^d)$ 是一个幺半群：函数复合满足结合律，恒等函数显然是非递减的，且 $\mathsf{id}^d(0) = 0$ 且 $\mathsf{id}^d(d+1) = d+1$；由于任意非递减函数的函数复合仍然是非递减的，$\mathcal{M}_d$ 在函数复合下封闭。

关于与毛毛虫图的联系，我们在引理 6.2 中已经论证，每个毛毛虫图的本质都是$\mathcal{M}_d$ 中的一个函数。我们将看到，对于$\mathcal{M}_d$ 中的任意函数 γ，都存在一个以 γ 为本质的毛毛虫图，而且可以快速计算出来。目前，我们暂且不考虑这种与毛毛虫图的联系，而是确立本质幺半群本身的算法性质。

若集合 $\mathcal{M}'_d \subseteq \mathcal{M}_d$ 且 $\mathsf{id}^d \in \mathcal{M}'_d$，则称其为 $(\mathcal{M}_d, \circ, \mathsf{id}^d)$ 的生成元，当且仅当满足以下条件：包含$\mathcal{M}'_d \subseteq \hat{}\mathcal{M}_d$的最小幺半群 $(\hat{}\mathcal{M}_d, \circ, \mathsf{id}^d)$ 满足 $\hat{}\mathcal{M}_d = \mathcal{M}_d$。换句话说，只要有生成元$\mathcal{M}'_d$，就能通过函数复合得到$\mathcal{M}_d$ 中的所有元素。

与$\mathcal{M}_d$ 相关的幺半群在数学文献中已有充分研究。其中，定义在非递减函数 f：$[1, n] \to [1, n]$ 上的幺半群尤为重要，记作$\mathcal{O}_n$ [28, 32, 33, 36–38]。注意，我们的本质幺半群$\mathcal{M}_d$与函数集$\{f \in \mathcal{O}_{d+2} \mid f(1) = 1, f(d+2) = d+2\}$上的幺半群同构。已知$\mathcal{O}_n$可由幂等函数 [28, 33, 36, 37]，或准幂等函数（以及恒等函数）[38]通过复合生成。Kudryavtseva 和 Mazorchuk 的研究中出现过一个与我们的幺半群同构的幺半群 [44, 第 6 节7]，，她们还为其展示了一个小生成元。

[6] 若函数 f 满足 $f = f \circ f$，则称其为幂等的；若其不幂等但 $f \circ f$ 满足该条件，则称其为准幂等的。

接下来，我们定义一组简单的小函数，并将证明它们是本质幺半群的生成元。

▶ **定义 2.8**（基本函数）。设 d 为正整数。对于任意 $i \in [1,d]$，令dec_i^d: $[0,d+1] \to [0,d+1]$ 为如下定义的函数

$$\mathsf{dec}_i^d(x) = \begin{cases} x-1 & \text{if } x = i, \\ x & \text{otherwise.} \end{cases}$$

令 inc^d： $[0,d+1] \to [0,d+1]$ 由下式定义

$$\mathsf{inc}^d(x) = \begin{cases} x+1 & \text{if } 1 \le x \le d, \\ x & \text{otherwise.} \end{cases}$$

（相对于 d 的）基本函数集合为集合 $\{\mathsf{id}^d\} \cup \{\mathsf{dec}_i^d \mid i \in [1,d]\} \cup \{\mathsf{inc}^d\}$。

现在，我们不仅证明基本函数集合是本质幺半群的生成元，而且实际上可以通过复合几个基本函数得到该幺半群中的任意函数。为此，我们首先引入一组高级函数；这些函数仍然相对简单，之后将由基本函数构造得到。

▶ **定义 7.1**（高级函数）。设 d 为正整数。对于任意 $i \in [1,d]$，令inc_i^d: $[0,d+1] \to [0,d+1]$ 为函数

$$\mathsf{inc}_i^d(x) = \begin{cases} x+1 & \text{if } x = i, \\ x & \text{otherwise.} \end{cases}$$

对于任意 $i,j \in [1,d+1]$，满足 $i<j$，定义 $\mathsf{inc}^d_{i,\,j}$： $[0,d+1] \to [0,d+1]$ 为

$$\mathsf{inc}_{i,j}^d(x) = \begin{cases} j & \text{if } i \le x \le j, \\ x & \text{otherwise.} \end{cases}$$

对于任意 $i,j \in [0,d]$，满足 $i<j$，定义 $\mathsf{dec}^d_{i,\,j}$： $[0,d+1] \to [0,d+1]$ 为

$$\mathsf{dec}_{i,j}^d(x) = \begin{cases} i & \text{if } i \le x \le j, \\ x & \text{otherwise.} \end{cases}$$

我们称函数 $\mathsf{inc}_i^d (i \in [1,d])$、$\mathsf{inc}^d_{i,\,j} (i,j \in [1,d+1], i<j)$ 和 $\mathsf{dec}^d_{i,\,j} (i,j \in [0,d], i<j)$ 为相对于 d 的高级函数。

我们用于复合本质幺半群中函数的方法，基于抽象代数文献 [28,33] 中的技巧。回顾一下，$\mathcal{O}_n$ 是定义在域 $[1,n]$ 上的非递减函数幺半群。Higgins [32] 证明，任意函数$g \in \mathcal{O}_n$ 都可以由 $\mathcal{O}_n$ 的 $O(n^2)$ 个缺陷为一的幂等元复合而成。这些函数$g \in \mathcal{O}_n$ 满足 $g = g \circ g$，且只有一个输入值 x 满足 $g(x) \neq x$。这意味着，对于满足 $g(x) \neq x$ 的此类函数 g，必有 $g(x) \in \{x-1,\ x+1\}$。因此，函数 inc_i^d 和 dec_i^d 与这些函数的形式完全相同。尽管 Higgins 给出的结果适用于非递减函数 g，其中 0，$d+1$ 不一定是不动点，但其核心技巧在我们的情形下同样适用。我们在此给出形式化证明，以使论文内容自足，并再次确认这些结果在我们的情形下正确。

然而，我们的基本函数集合不包含函数 inc_i^d，因此首先需要使用基本函数组合出这些函数。为此，我们借鉴 Gomes 和 Howie [28] 在证明引理 2.6 时采用的思路，得到如下引理 7.2。

38 通过到有界路径宽度的距离进行 COC 核化的分界线

▶ **引理 7.2**。 设 d 为正整数，且 $i \in [1,d]$。那么，存在 $\ell \le d$ 个相对于 d 的基本函数 $g_1, \ldots, g_\ell$，使得 $\mathsf{inc}_i^d = g_1 \circ g_2 \cdots \circ g_\ell$。

证明。 令

$$h_1 = \begin{cases} \mathsf{dec}_d^d \circ \cdots \circ \mathsf{dec}_{i+1}^d & \text{if } i+1 \le d, \\ \mathsf{id}^d & \text{otherwise,} \end{cases}$$

$$h_2 = \mathsf{inc}^d,$$

$$h_3 = \begin{cases} \mathsf{dec}_i^d \circ \cdots \circ \mathsf{dec}_2^d & \text{if } i \ge 2, \\ \mathsf{id}^d & \text{otherwise.} \end{cases}$$

注意， h_1 将所有输入 $x \in [i+1,d]$ 恰好减一， h_2 将所有介于 1 和 d 之间的输入增加，且 h_3 将所有 $x \in [2,i]$ 恰好减一。这些函数将其他所有输入映射到自身。令 $h^* = h_3 \circ h_2 \circ h_1$。

任取 $x \in [0,d+1]$。若 $x \in \{0, d+1\}$，则 $h^*(x) = x$，如所需，因为 $0, d+1$ 是所有复合函数的不动点。

若 $1 \le x \le i-1$，则有 $h_1(x) = x$，$(h_2 \circ h_1)(x) = x+1$，且 $h^*(x) = x$。

若 $x = i$，则有 $h_1(x) = i$，$(h_2 \circ h_1)(x) = i+1 = h^*(x)$。若 $i+1 \le x \le d$，则有 $h_1(x) = x-1$，$(h_2 \circ h_1)(x) = x = h^*(x)$。因此，$\mathsf{inc}_i^d = h^*$。 ◀

现在，我们继续采用 Higgins 的方法 [33]，该方法构成了后续观察、推论和引理的基础，直至引理 2.9。首先可以发现，很容易用至多 d 个函数（这些函数要么是基本函数，要么是某些取值下的函数 $\mathsf{inc}_{i'}^d$）复合出函数 $\mathsf{inc}_{i,\,j}^d$ 和 $\mathsf{dec}_{i,\,j}^d$。

▶ **观察 7.3**。 设 d 为正整数。对于任意满足 $i < j$ 的 i、$j \in [1, d+1]$，有 $\mathsf{inc}_{i,\,j}^d = \mathsf{inc}_{j-1}^d \circ \cdots \circ \mathsf{inc}_i^d$。对于任意满足 $i < j$ 的 i、$j \in [0,d]$，有 $\mathsf{dec}_{i,\,j}^d = \mathsf{dec}_{i+1}^d \circ \cdots \circ \mathsf{dec}_j^d$。

我们可以用下面这个简单的算法推论来概括有关高级函数的结果。

▶**推论 7.4**。 设 d 为正整数。存在一个算法，以相对于 d 的高级函数 f：$[0,d+1] \to [0,d+1]$ 为输入，并输出至多 d^2 个相对于 d 的基本函数g_1、……、g_ℓ，使得 $f = g_1 \circ g_2 \cdots \circ g_\ell$。该算法的运行时间关于 d 为多项式。

我们首先关注函数 f，它将任意输入 x 映射到一个至少为 x的值。对于函数 f，令 $\mathrm{fp}(f)$ 表示 f的不动点数量。

▶**引理 7.5**。 设 d 是一个正整数。存在一种算法，输入满足以下条件的非递减函数 f：$[0,d+1] \to [0,d+1]$

1. $f \neq \mathsf{id}^d$，且 *2.* 对所有 $x \in [0,d+1]$，$f(x) \ge x$，并且 *3.* $f(0) = 0$，$f(d+1) = d+1$，作为输入，并输出 $\ell \le d+2$ 个相对于 d 的高级函数 $g_1, \ldots, g_\ell$，使得$f = g_1 \circ \cdots \circ g_\ell$。该算法的运行时间关于 d为多项式。

证明。设 X 是 $[0,d+1]$的一个划分，使得对于所有 $x, y \in [0,d+1]$，x, y属于 X 的同一划分，当且仅当 $f(x) = f(y)$。对于 $X \subseteq [0,d+1]$，我们定义：对所有 $x \in X$，若 $f(x) = y$，则 $f(X) = y$。

现在，从$\mathcal{X}$中移除那些满足 $\min X = f(X)$ 的集合 X，以构造$\mathcal{X}'$，即$\mathcal{X}' = \{X \in \mathcal{X} \mid \min X \neq f(X)\}$。我们将简要说明 $|\mathcal{X}'| \neq \emptyset$。为此，考虑某个

$X \in \mathcal{X}\setminus\mathcal{X}'$，也就是某个集合 $X \in \mathcal{X}$，满足 $f(X) = \min X$。我们论证必有 $|X| = 1$。否则，存在某个 $x \in X \setminus \{\min X\}$。然而，此时 $f(x) = f(X) = \min X < x$，这与对所有定义域元素 y 均有 $f(y) \geq y$ 矛盾。因此，我们有 $|X| = 1$。现在，如果 $\mathcal{X}$ 中的所有集合大小都为一，那么必有 $f = \mathrm{id}^d$。由于情况并非如此，$\mathcal{X}$中的某个集合大小至少为二，并且是$\mathcal{X}'$的一部分。

接下来，我们论证对于任意 $X \in \mathcal{X}'$，都不存在 $\min X = 0$。反设不然。那么，由 $f(0) = 0$可得 $\min X = f(X) = 0$，这与 $X \in \mathcal{X}'$ 矛盾。

现在，按照 f 的值递减排列 $\mathcal{X}'$，即 $\mathcal{X}' = \{X_1, \ldots, X_t\}$，使得对所有 $i < j$，$f(X_i) > f(X_j)$。令 $h = \mathrm{inc}^d_{\min X_t, f(X_t)} \circ \cdots \circ \mathrm{inc}^d_{\min X_1, f(X_1)}$，并注意到，根据上一段的论证，此函数定义良好。

固定某个 $x \in [0, d+1]$。函数 $\mathrm{inc}^d_{j_1,j_2}$ 仅在 $x \in [j_1, j_2 - 1]$ 时改变输入 x。根据我们对 h 的选择，这意味着存在某个集合 X_j，使得如果在 h 的复合中用于 X_j 的函数 $\mathrm{inc}^d_{\min X_j, f(X_j)}$ 改变输入值 x，那么 $\min X_j \leq x < f(X_j)$。

我们将证明 $h(x) = f(x)$。首先，考虑 $f(x) > x$，于是对于某个$X_i \in \mathcal{X}'$，有 $x \in X_i$，且 $f(x) = f(X_i)$。任取一个集合 X_j，满足 $j < i$。由于我们已按 f 的值降序排列$\mathcal{X}'$，因此有 $f(X_j) > f(X_i) = f(x)$。此外，由于 f 是非递减的，我们得到 $\min X_j > x$。因此，在函数复合中用于X_j 的函数 $\mathrm{inc}^d_{\min X_j, f(X_j)}$ 不会改变输入值 x。所以，

$$(\mathrm{inc}^d_{\min X_i, f(X_i)} \circ \cdots \circ \mathrm{inc}^d_{\min X_1, f(X_1)})(x) = \mathrm{inc}^d_{\min X_i, f(X_i)}(x).$$

由于对于所有定义域元素，$f(y) \geq y$，因此有 $\min X_i \leq x \leq f(X_i)$，由此得到 $\mathrm{inc}^d_{\min X_i, f(X_i)}(x) = f(X_i) = f(x)$。现在，考虑满足 $j > i$ 的某个集合 X_j。那么，$f(X_j) < f(X_i)$，因此函数 $\mathrm{inc}^d_{\min X_j, f(X_j)}$ 不会改变输入值$f(X_i)$。由此可得 $h(x) = (\mathrm{inc}^d_{\min X_t, f(X_t)} \circ \cdots \circ \mathrm{inc}^d_{\min X_i, f(X_i)})(x) = f(x)$。

现在，考虑 $f(x) = x$。那么，不存在这样的 $X_j \in \mathcal{X}'$，使得相应的函数 $\mathrm{inc}^d_{\min X_j, f(X_j)}$ 改变输入值 x。事实上，这将意味着 $\min X_j \leq x = f(x) < f(X_j) = f(\min X_j)$。于是，我们会有 $x \geq \min X_j$，但 $f(x) < f(\min X_j)$，这与 f 是非递减的相矛盾。因此，$h(x) = f(x)$，因为这些复合函数都不会改变值 x。

最后，我们需要说明，为构造 h，至多只需复合 $d + 2 - \mathrm{fp}(f)$ 个函数。为此，我们说明，f 的每个不动点 x 要么不属于任何一个 $\mathcal{X}'$ 集合，要么属于一个 $\mathcal{X}'$ 集合，且该集合还包含另一个元素 $x' \in [0, d+1]$，它不是不动点。假设某个 $X \in \mathcal{X}'$ 存在一个不动点 $x \in X$。那么，X 不可能包含除 x 以外的其他不动点，因为 X 的所有元素具有相同的像。此外，如果 $|X| = 1$，那么我们会得到 $\min X = x = f(X)$，这与 $X \in \mathcal{X}'$ 矛盾。因此，X 中还包含某个整数 x'，且它不是不动点。由此，任何 $\mathcal{X}'$ 集合都包含 $[0, d+1]$ 中一个不是不动点的整数，这表明 $\mathcal{X}'$ 的大小至多为 $d + 2 - \mathrm{fp}(f)$。◀

对于满足条件 $f(x) \leq x$ 的函数 f，对定义域中的所有元素 x 均成立，我们可以得到一个类似的引理。证明与前一个证明对称，因此大部分内容相同，但为求完整起见，我们仍在此给出证明。

▶**引理 7.6**。设 d 为正整数。存在一个算法，以非递减函数 f：$[0, d+1] \to [0, d+1]$ 为输入，满足 **1.** $f \neq \mathrm{id}^d$，且**2.** $f(x) \leq x$ 对所有 $x \in [0, d+1]$，且**3.** $f(0) = 0$，$f(d+1) = d+1$，

40 通过到有界路径宽度的距离实现 COC 核化的分界线

以……为输入，并输出 $\ell \leq d+2-\mathsf{fp}(f)$ 高级函数 $g_1, \ldots, g_\ell$（相对于 d），使得 $f = g_1 \circ \cdots \circ g_\ell$。该算法的运行时间关于 d 为多项式。

证明。设 $\mathbf{X}$ 是 $[0, d+1]$ 的一个划分，使得对于所有 x、$y \in [0, d+1]$，x、y 属于 $\mathbf{X}$ 的同一个划分，当且仅当 $f(x) = f(y)$。对于 $X \subseteq [0, d+1]$，我们定义：若对于所有 $x \in X$，$f(x) = y$，则 $f(X) = y$。

现在，从 $\mathbf{X}$ 中移除那些满足 $\max X = f(X)$ 的集合 X，得到 $\mathbf{X}'$，也就是说，$\mathbf{X}' = \{X \in \mathbf{X} \mid \max X \neq f(X)\}$。我们简要说明 $|\mathbf{X}'| \neq \emptyset$。为此，考虑某个 $X \in \mathbf{X} \setminus \mathbf{X}'$，也就是 $\mathbf{X}$ 中某个满足 $f(X) = \max X$ 的集合 $X \in$。我们论证必有 $|X| = 1$。否则，存在某个 $x \in X \setminus \{\max X\}$。然而，此时 $f(x) = f(X) = \max X > x$，这与对于所有定义域元素 y 都有 $f(y) \leq y$ 矛盾。因此，我们有 $|X| = 1$。现在，如果 $\mathbf{X}$ 中所有集合的大小都为一，那么必有 $f = \mathrm{id}^d$。既然并非如此，$\mathbf{X}$ 中就有某个大小至少为二的集合，并且它属于 $\mathbf{X}'$。

接下来，我们证明，对于任意 $X \in \mathcal{X}'$，都不存在 $\max X = d+1$。反设存在。于是，根据 $f(d+1) = d+1$，我们可以得到 $\max X = f(X) = d+1$，这与 $X \in \mathcal{X}'$ 矛盾。

现在，按照 f 的值递增的顺序排列 $\mathcal{X}'$，也就是说，$\mathcal{X}' = \{X_1, \ldots, X_t\}$，使得对所有 $i < j$，都有 $f(X_i) < f(X_j)$。令 $h = \mathsf{dec}^d_{f(X_t), \max X_t} \circ \cdots \circ \mathsf{dec}^d_{f(X_1), \max X_1}$，并注意到，根据上一段的论证，该函数定义良好。

固定某个 $x \in [0, d+1]$。函数 $\mathsf{dec}^d_{j_1, j_2}$ 仅当 $x \in [j_1+1, j_2]$ 时才会改变输入 x。根据我们对 h 的选择，这意味着对于某个集合 X_j，若用于 X_j 在 h 的复合中的函数 $\mathsf{dec}^d_{f(X_j), \max X_j}$ 改变了输入值 x，则有 $f(X_j) < x \leq \max X_j$。

我们将证明 $h(x) = f(x)$。首先，考虑 $f(x) < x$，那么对于某个 $X_i \in \mathcal{X}'$，有 $x \in X_i$，且 $f(x) = f(X_i)$。考虑任意一个满足 $j < i$ 的集合 X_j。由于我们已按 f 的值递增顺序对 $\mathcal{X}'$ 进行了排序，因此有 $f(X_j) < f(X_i) = f(x)$。此外，由于 f 非递减，我们得到 $\max X_j < x$。因此，在函数复合中用于 X_j 的函数 $\mathsf{dec}^d_{f(X_j),\ \max X_j}$ 不会改变输入值 x。所以，

$$(\mathsf{dec}^d_{f(X_i), \max X_i} \circ \cdots \circ \mathsf{dec}^d_{f(X_1), \max X_1})(x) = \mathsf{dec}^d_{f(X_i), \max X_i}(x).$$

由于对于所有定义域元素都有 $f(y) \leq y$，因此我们有 $f(X_i) \leq x \leq \max X_i$，由此可得 $\mathsf{dec}^d_{f(X_i), \max X_i}(x) = f(X_i) = f(x)$。现在，考虑一个满足 $j > i$ 的集合 X_j。那么，$f(X_j) > f(X_i)$，因此函数 $\mathsf{dec}^d_{f(X_j), \max X_j}$ 不会改变输入值 $f(X_i)$。由此可得 $h(x) = (\mathsf{dec}^d_{f(X_t), \max X_t} \circ \cdots \circ \mathsf{dec}^d_{f(X_i), \max X_i})(x) = f(x)$。

现在，考虑 $f(x) = x$。那么，不存在这样的 $X_j \in \mathcal{X}'$，使得对应的函数 $\mathsf{dec}^d_{f(X_j),\ \max X_j}$ 会改变输入值 x。事实上，这将意味着 $f(\max X_j) = f(X_j) < x = f(x) \leq \max X_j$。于是，我们会有 $x \leq \max X_j$ 以及 $f(x) > f(\max X_j)$，这与 f 是非递减的相矛盾。因此，$h(x) = f(x)$，因为这些复合函数都不会改变值 x。

最后，我们需要证明，为构造 h，至多需要复合 $d+2-\mathsf{fp}(f)$ 个函数。为此，我们证明，每个 f 的不动点 x 要么不属于任何形如 $\mathcal{X}'$ 的集合，要么属于一个形如 $\mathcal{X}'$ 的集合，而该集合还包含另一个元素 $x' \in [0, d+1]$，它不是不动点。假设对于某个 $X \in \mathcal{X}'$，存在不动点 $x \in X$。那么，X 不可能包含除 x 之外的其他不动点，因为 X 的所有元素都具有相同的像。此外，如果 $|X| = 1$，那么我们就会得到 $\max X = x = f(X)$，这与 $X \in \mathcal{X}'$ 矛盾。因此，X 中还包含某个整数 x'，它不是不动点。由此可知，任何形如 $\mathcal{X}'$ 的集合都包含 $[0, d+1]$ 中一个不是不动点的整数，这表明 $\mathcal{X}'$ 的大小至多为 $d+2-\mathsf{fp}(f)$。◀

现在，我们可以利用引理 7.5 和 7.6，得到关于复合本质幺半群中函数的主要结果。

▶ **引理 2.9.** 存在一种算法，以非递减函数 $f\colon [0,d+1]\to[0,d+1]$（满足 $f(0)=0$ 且 $f(d+1)=d+1$）为输入，并输出 $g_1,\dots,g_\ell$，其中 $\ell\le d^3$，使得 $f=g_\ell\circ g_{\ell-1}\circ\cdots\circ g_1$，且对于所有 $i\in[1,\ell]$，函数 g_i 相对于 d 是基本函数。该算法的运行时间关于 d 为多项式。

证明。 如果 $f=\mathsf{id}^d$ 我们输出基本函数 id^d。

否则，我们根据 f 构造以下两个函数：f^+ 和 f^-

$$f^+(x)=\begin{cases} f(x) & \text{if } f(x)>x,\\ x & \text{otherwise.}\end{cases}\qquad f^-(x)=\begin{cases} f(x) & \text{if } f(x)<x,\\ x & \text{otherwise.}\end{cases}$$

我们证明 $f=f^+\circ f^-$。如果 $f(x)>x$，这一点显然成立，因为 $f^-(x)=x$。当 $f(x)<x$ 时，注意到由于 f 是非递减的，必有 $f(f(x))\le f(x)$。因此，$f^+(f^-(x))=f^+(f(x))=f(x)$。

我们可以应用引理 7.5 和 7.6，得到用高级函数表示的 f^+ 和 f^- 的函数复合；如果 f^+ 或 f^- 中有一个是 id^d，我们就可以直接省略复合中的该函数。由于每个 $x\in[1,d]$ 都是 f^+ 或 f^-的不动点，并且 $0,d+1$同时也是 f^+ 和 f^- 的不动点，我们发现所用高级函数的总数至多为 $2(d+2)-\mathsf{fp}(f^+)-\mathsf{fp}(f^-)\le d$。根据推论 7.4，我们可以用至多 d^2 个基本函数复合出任意高级函数，证毕。◀

接下来，我们要建立这些函数与所用毛毛虫图之间的联系。具体而言，我们首先证明：对于任意 d 以及相对于 d的任意基本函数，都存在一只本质为该基本函数的简单毛毛虫图。

▶**引理 2.11.** 给定任意基本函数 f（相对于某个$d\ge 1$），存在一个算法，以其为输入，并在关于 d的多项式时间内输出一个本质为 f 的毛毛虫图 C，该图具有大小为一的解紧打包（因此只包含 C）。

证明。 基本函数有三种不同类型：id^d、inc^d和dec_i^d（其中$i\in[1,d]$）。

算法。 令 f 为输入的基本函数。此外，对于任意整数 i，j 满足$i\le j$，令 P_j^i为如下构造的毛毛虫图：取一条包含 v_1，……，v_j 个顶点的路径 j，并在 v_i上添加一个悬挂顶点。图 P_i 的脊柱为 v_1，……，v_j。算法根据 f分三种情况执行。

若 $f=\mathsf{id}^d$，算法只需输出含 $d+1$ 个顶点的路径，此时毛毛虫图的脊柱就是整条路径。若 $f=\mathsf{inc}^d$，算法输出毛毛虫图 P_{d+1}^{d+1}。若 $f=\mathsf{dec}_i^d$，其中某个 $i\in[1,d]$，则算法输出毛毛虫图 P_d^{d-i+1}。容易看出，这些毛毛虫图均可在关于 d的线性时间内计算出来，也可以在线性时间内判断 f 属于哪种基本函数。因此，该算法的总体运行时间关于 d 为线性。

算法的正确性。 接下来还需证明，该算法输出的毛毛虫图确实具有正确的本质。对于任意本质 γ，我们有 $\gamma(d+1)=d+1$ 根据定义，因此我们只需证明，对于介于 f 0 和 d 之间的输入，函数相匹配。

42 通过到有界路径宽度的距离对 COC 进行核化的分界线

$f = \mathrm{id}^d$: 那么，该算法的输出是一条包含 $d+1$ 个顶点的路径 $v_1, \ldots, v_{d+1}$。注意， P 的解紧打包中有一个图包含了 P 的所有顶点。令γ为 P的本质。我们可以选取顶点 v_{d+1}，以证明 $\gamma(0) = 0 = f(0)$。若考虑图 P_d，它是一个 P，其中 v_1 有 d 个额外的悬挂顶点，那么选取v_1 即可证明 $\gamma(d) = d$。

令 $x \in [1, d-1]$。考虑图 P_x，它是通过在 P 最左侧的脊顶点上添加 x 个额外的悬挂顶点得到的。于是， $S = \{v_{d+1-x}\}$ 是 P 的一个 d-coc 集合，并且不存在大小为一、选取一个满足 $j > d+1-x$ 的顶点 v_j 的 d-coc 集合。事实上， $P - S$ 含有一个连通分量，该分量包含 $v_1, \ldots, v_{d-x}$ 和 v_1 的 x 个悬挂顶点，且该连通分量的大小恰为 d。这也表明，不存在大小至多为一且包含一个满足 $j > d+1-x$ 的顶点 v_j 的 d-coc 集合。此外， $P - S$ 含有一个连通分量，该分量包含 $v_{d-x+2}, \ldots, v_{d+1}$，此连通分量的大小恰为 $(d+1) - (d-x+2) + 1 = x$，这说明 $\gamma(x) = x = f(x)$ 。

$f =$ 增加^d：然后，该算法输出 $P = P_{d+1}^{d+1}$ 其脊柱为 v_1、.....、v_{d+1}。再次， P 的解紧打包大小恰好为一，并选择了 P的所有顶点。令 γ 为 P的本质。于是有 γ（0） $= 0 = f$（0），因为 $\{v_{d+1}\}$ 是 d-coc的一个集合 P。

考虑通过向 v_1 添加 d 个悬挂顶点而得到的图 P_d。我们证明 $\mathrm{OPT}_d(P_d) > 1$。任意最小的 d-coc 集合 S 都可以假设选择了 v_1，这是由于 v_1 的 d 个悬挂顶点。然而，如果 S 不选择其他顶点，那么 v_2、 ..... 、 v_{d+1} 以及 v_{d+1} 的悬挂顶点将构成一个大小为 $d+1$ 的连通分量，因此 S 根本不会是一个 d-coc 集合。由于 $\mathrm{OPT}_d(P_d) > \mathrm{OPT}_d(P)$，我们有 $\gamma(d) = d+1 = f(d)$，如 desired。

现在，考虑任意 $x \in [1, d-1]$。那么， $S = \{v_{d+1-x}\}$ 是一个大小为一的 d-coc 集合 P_x。事实上， $P_x - S$包含一个连通分量，其中包含 $v_1, \ldots, v_{d-x}$ 以及 v_1的 x 个悬挂顶点，该分量的大小恰好为 d。这也表明，不存在大小至多为一、选取满足 $j > d+1-x$的顶点 v_j 的 d-coc 集合。 $P_x - S$ 的另一个连通分量包含 $v_{d-x+2}, \ldots, v_{d+1}$，以及 v_{d+1} 额外的悬挂顶点。因此，该连通分量的大小恰好为 $x+1 = \gamma(x) = f(x)$。

$f = \mathrm{dec}_i^d$ **对于某个** $i \in [1, d]$**：** 在这种情况下，输出为 $P = P_d^{d-i+1}$，其脊柱为 $v_1, \ldots, v_d$。同样，不难看出， P的解紧打包中的所有顶点都属于同一个图。对于任意 $x \geq 0$，图 P_x 与 P相同，其中 v_1 获得 x 个额外的悬挂顶点。令 γ 为 P的本质。那么，我们可以选取 v_d，从而得到 $\gamma(0) = 0$。

$i = d$: 接下来，我们要讨论 $i = d$ 的情况，此时 $d - i + 1 = 1$，因此 v_1 在图 P 中有一个悬挂顶点。然后，我们可以选取图 P_d 的 v_1，以表明 $\gamma(d) = d - 1$。事实上， $P_d - \{v_1\}$ 只包含大小为一的连通分量，即 v_1 的悬挂顶点，以及包含 v_2、.....、v_d 且大小为 $d-1$ 的连通分量。对于 $x = d-1$，我们也选取 v_1。由于这会得到与 $x = d$ 的情况基本相同的连通分量，并且 $f(d-1) = d-1$，因此只需说明不存在大小为一且选取 v_2 的解。事实上， v_1 位于 $P_{d-1} - \{v_2\}$ 中一个大小为 $d+1$ 的连通分量内，因为 v_1 在 P_{d-1} 中有 d 个悬挂顶点。因此， $\gamma(d-1) = f(d-1)$。对于任意 $x \in [1, d-2]$，我们可以选取 P_x 的顶点 v_{d-x}，并注意到 $v_{d-x} \neq v_1$。于是，$P_x - \{v_{d-x}\}$ 有一个连通分量，其中包含 v_1、.....、v_{d-x-1}，以及 v_1 的 $x+1$ 个悬挂顶点。该分量的大小为 d，因此不存在大小为 d-coc 、且选取满足 $j > d-x$ 的顶点 v_j 的 P_x 集合。此外， $P_x - \{v_{d-x}\}$ 也有一个包含 v_{d-x+1}、.....、v_d 且大小为 $x = \gamma$ 的连通分量，$x(= f)$ x。

$i < d$**：** 现在，我们考虑 $i < d$的情况。我们已经证明$\gamma(0) = 0$。现在，任取一个 $x \in [1, i-2]$。接着，令 $S = \{v_{d-x}\}$，并观察

that $d-x>1$。图 P_x-S包含一个连通分量，其中包含顶点v_1、……、v_{d-x-1}，并且根据 $d-x-1\geq d-i+1$，还包含 v_{d-i+1}的另一个悬挂顶点，以及 v_1的 x 个额外悬挂顶点。因此，该连通分量的大小恰好为d，这也表明，对于 P_x，不存在大小至多为 1 且选择顶点 v_j并满足 $j>d-x-1$的解。此外，P_x-S 包含一个连通分量，其中包含v_{d-x+1}、……、v_d，其大小恰好为 $x=\gamma(x)=f(x)$。

接下来，考虑 $x=i-1$。在这种情况下，我们再次令 $S=\{v_{d-x}\}=\{v_{d-i+1}\}$。再次注意到，$d-x>1$ 如同 $i<d$。因此，P_x-S 中存在一个连通分量，其中包含顶点 $v_1,\ldots,v_{d-i}$，以及 $i-1$ 个额外的悬挂顶点 v_1。这个连通分量的大小为 $d-1$。因此，我们需要论证不存在选择顶点 v_j 且满足 $j>d-i+1$的解 S'。事实上，$S'=\{v_{d-i+2}\}$ 不是 P_x的 d-coc 集合，因为 P_x-S' 包含一个连通分量，其中包含 $v_1,\ldots,v_{d-i+1}$，以及位于 v_1上的 $i-1$ 个额外悬挂顶点，还有 v_{d-i+1}的额外悬挂顶点，从而得到一个大小为 $d+1$的连通分量。图 P_x-S 还包含一个大小恰为 1 的连通分量，即包含 v_{d-i+1}的悬挂顶点的连通分量。最后，P_x-S 包含一个连通分量，其中包含 $v_{d-x+1},\ldots,v_d$，因此 v_d 位于大小恰为 $x=\gamma(x)=f(x)$ 的连通分量中。

现在，考虑 $x=i$。在这种情况下，我们同样可以令 $S=\{v_{d-x+1}\}$，这意味着我们选择的顶点与前一种情况完全相同，即顶点 $v_{d-i+1}\neq v_1$。从前一种情况的推导显然可知，对于 P_x，不可能存在一个解选择满足 $j>d-x+1$的顶点 v_j。因此，接下来只需说明 S 确实是 P_x的一个 d-coc 集合。事实上，P_x 包含一个连通分量，其中有 v_1、……、v_{d-i}，以及 v_1的 $x=i$ 个额外悬挂顶点，该分量的大小恰好为 d。其余连通分量与我们考虑 $x=i-1$时的情况完全相同，因此特别有，v_d 位于一个大小为 $i-1=x-1=\gamma(x)=f(x)$的连通分量中。接下来，考虑 $x\in[i+1,d-1]$的情况。于是，我们令 $S=\{v_{d-x+1}\}$。再次注意，我们有 $d-x+1>1$。在这种情况下，P_x-S 有一个连通分量，其中包含 v_1、……、v_{d-x}，以及 v_1的 x 个额外悬挂顶点。该连通分量的大小恰好为 d，这也表明，对于任意 $j>d-x+1$，不存在选择 v_j 的最小 d-coc 集合。此外，P_x-S 有一个连通分量，其中包含 v_{d-x+2}、……、v_d，并且与 $d-x+2\leq d-i+1$一样，也包含 v_{d-i+1}的额外悬挂顶点。因此，v_d 位于一个大小恰好为 $x=\gamma(x)=f(x)$ 的连通分量中。

最后，考虑 $x=d$。于是，我们令 $S=\{v_1\}$。在这种情况下，$P_{d+1}-S$ 有一个大小为一的连通分量（包含 v_1的悬挂顶点），以及一个包含 $v_2,\ldots,v_d$和 v_{d-i+1}的额外悬挂顶点的连通分量。这个连通分量的大小恰好为 $d=\gamma(d)=f(d)$。还需说明，不存在选择 v_2的大小为一的解。事实上，在 $P_d-\{v_2\}$ 中，v_1 位于一个大小为 $d+1$ 的连通分量中，因为在该图中 v_1 有 d 个悬挂顶点。

◀

现在，我们证明，将两个毛毛虫图通过它们的脊顶点连接起来，会得到一个毛毛虫图，而其本质恰好是这两个毛毛虫图各自本质的函数复合。

▶**引理 2.10.** 设 C_1 是一个毛毛虫图，其解紧打包包含 C_1的所有顶点。设 C_1 的本质为 γ_1，且 C_1 的脊柱为 $v_1,\ldots,v_n$。类似地，设 C_2 是一个毛毛虫图，其解紧打包包含 C_2的所有顶点，C_2的本质为 γ_2，且其脊柱为 $w_1,\ldots,w_m$。

44 通过到有界路径宽度的距离实现 COC 核化的分界线

通过对 C_1 和 C_2 取不交并，并添加一条从 v_n到 w_1的边，构造毛毛虫图 C。C的脊柱为 $v_1, \ldots, v_n, w_1 \ldots, w_m$。于是，$C$的本质 γ满足 $\gamma = \gamma_2 \circ \gamma_1$。

证明。注意，只需取 C_1 和 C_2的解紧致打包的并集，即可得到 C 的解紧致打包。由此直接可得，$\mathsf{OPT}_d(C) = \mathsf{OPT}_d(C_1) + \mathsf{OPT}_d(C_2)$。此外，由于任意本质都将 $d+1$ 映射到 $d+1$，因此我们有 $\gamma(d+1) = d+1 = \gamma_1(d+1) = \gamma_2(d+1) = \gamma_2 \circ \gamma_1(d+1)$。令 $x \in [0, d]$ 为一个整数，我们分两个不同的步骤证明$\gamma(x) = \gamma_2(\gamma_1(x))$：先证明 $\gamma(x) \leq \gamma_2(\gamma_1(x))$，再证明 $\gamma(x) \geq \gamma_2(\gamma_1(x))$。回想一下，$(C_i)_x$ （其中 $i \in \{1, 2\}$）是图 C_i，其最左侧的脊顶点额外连接了 x 个悬挂顶点。

$\gamma(x) \leq \gamma_2(\gamma_1(x))$：如果 $\gamma_1(x) = d+1$，那么我们也有 $\gamma_2(\gamma_1(x)) = d+1$。此外，如果$\gamma_2(\gamma_1(x)) = d+1$，那么显然有 $\gamma(x) \leq \gamma_2(\gamma_1(x))$。

因此，现在考虑 $\gamma_1(x) \leq d$ 且 $\gamma_2(\gamma_1(x)) \leq d$。如果 $\gamma_1(x) = 0$，令 S_1 为 $(C_1)_x$ 的最小解，其中选取了 v_n；如果 $\gamma_1(x) > 0$，令 S_1 为 $(C_1)_x$ 的解，其中 v_n位于 $(C_1)_x - S_1$ 的一个大小为 $\gamma_1(x)$的连通分量中。类似地，如果 $\gamma_2(\gamma_1(x)) = 0$，令 S_2为 $(C_2)_{\gamma_1(x)}$的最小解，其中选取了 w_m；否则，令 S_2 为 $(C_2)_{\gamma_1(x)}$的解，其中 w_m 位于 $(C_2)_{\gamma_1(x)} - S_2$ 的一个大小为 $\gamma_2(\gamma_1(x))$的连通分量中。我们还有 $|S_1| =$ 最优值$_d(C_1)$以及 $|S_2| =$ 最优值$_d(C_2)$。现在我们声称，$S = S_1 \cup S_2$ 是 C_x 的一个解，并且如果 $\gamma_2(\gamma_1(x)) = 0$，则选取 w_m；而当 $\gamma_2(\gamma_1(x)) > 0$时，$w_m$ 位于 $C_x - S$中一个大小为 $\gamma_2(\gamma_1(x))$的连通分量中。

首先，我们证明 S 是 d-coc 的 C_x 集合。我们可以假设 S_1 和 S_2 只包含 C_1 和 C_2 的脊顶点。因此，S_1 是 $(C_1)_x$ 的一个解，而 S_2 是 C_2 的一个解。于是，$C_x - S$ 中任何过大的连通分量都必须同时包含 $(C_1)_x$ 的两个顶点和 C_2。考虑 $C_x - S$ 中同时包含 C_1 的两个顶点和 C_2的一个连通分量 Z。由于该连通分量必须包含顶点 v_n，因此它恰好包含 C_1 的 x 个顶点。此外，Z 必须包含 w_1。然而，由于 S_2 实际上是图 $(C_2)_{\gamma_1(x)}$ 的一个解，Z至多能包含 C_2 的 x 个顶点，因此连通分量 Z 的总大小至多为 d。由此可知，S 是 d-coc 的 C_x 集合。

只有当 $\gamma_2(\gamma_1(x)) = 0$ 时，才会选中顶点 w_m；这表明在这种情况下也有 $\gamma(x) = 0$。否则，w_m 不会被选中，并且 w_m 位于 $C_x - S$ 的一个连通分量中。我们可以假设 S_2 选取了 C_2 的一个脊顶点，因此，该连通分量恰好就是图 $(C_2)_{\gamma_1(x)} - S_2$中包含 w_m 的那个连通分量，并且该分量的大小为 $\gamma_2(\gamma_1(x))$。因此，由解 S可知，$\gamma(x) \leq \gamma_2(\gamma_1(x))$。

$\gamma(x) \geq \gamma_2(\gamma_1(x))$：显然，若 $\gamma(x) = d+1$，则有 $\gamma(x) \geq \gamma_2(\gamma_1(x))$，因此在证明的其余部分中，我们假设$\gamma(x) \leq d$。于是，存在一个解 S，满足 $|S| =$ 最优值$_d(C_1)$+最优值$_d(C_2)$，对于 C_x，且满足以下条件之一：若 $\gamma(x) = 0$，则 S 选择 w_m；否则，w_m 属于 $C_x - S$ 的一个大小为 $\gamma(x)$ 的连通分量。

我们有，$S \cap V(C_1)$ 是 $(C_1)_x$ 的一个解，其大小为最优值$_d(C_1)$，因为 $(C_1)_x$ 和 C_2都是 C_x的子图，且 S的大小为最优值$_d(C_1)$ +最优值$_d(C_2)$。因此，只有当 $\gamma_1(x) = 0$时，S才能选择 v_n；否则，v_n 属于 $C_x - S$ 的一个大小至少为 $\gamma_1(x)$ 的连通分量。

现在，我们声称，$S \cap V(C_2)$ 必定是 $(C_2)_{\gamma_1(x)}$ 的一个解，其规模为 $\mathsf{OPT}_d(C_2)$。$S \cap V(C_2)$ 的规模为 $\mathsf{OPT}_d(C_2)$，这直接源于以下事实：$|S| = \mathsf{OPT}_d(C_1) + \mathsf{OPT}_d(C_2)$。因此，剩下要做的就是证明 $S \cap V(C_2)$ 确实是 $(C_2)_{\gamma_1(x)}$ 的一个解。我们用矛盾法证明这一性质。假设 $S \cap V(C_2)$ 不是 $(C_2)_{\gamma_1(x)}$ 的解。那么，包含w_1 的 $(C_2)_{\gamma_1(x)}$ 连通分量 Z 的规模大于 d。然而，在 $C_x - S$ 中也存在一个规模与 Z 相同的连通分量，因为 w_1 的连通分量包含 C_1 中的 $\gamma_1(x)$ 个顶点，

而这些顶点的数量恰好等于 w_1 在 $(C_2)_{\gamma_1(x)}$ 中相比于 C_2 多出的悬挂顶点数量。因此， S 根本就不是 C_x 的解。

因此，事实上我们有 $S \cap V(C_2)$ 是 $(C_2)_{\gamma_1(x)}$ 的一个最优解，并且包含 w_m 的 $C_2 - (S \cap V(C_2))$ 的连通分量大小为 $\gamma(x)$。这表明 $\gamma_2(\gamma_1(x)) \leq \gamma(x)$，正如所需。我们得出结论，对于所有 $x \in [0, d+1]$， $\gamma(x) = \gamma_2(\gamma_1(x))$。◀

现在，我们可以轻松地计算出一个具有特定本质的毛毛虫图。

▶ **引理 2.12.** 存在一个算法，以非递减函数 $\gamma\colon [0, d+1] \to [0, d+1]$ 为输入，满足 $\gamma(0) = 0, \gamma(d+1) = d+1$，并输出一个毛毛虫图 C，其本质为 γ，使得 C 的解紧打包 C 包含 C 的所有顶点，并且在关于 d 的多项式时间内完成 $|C| \leq d^3$。

证明。 我们首先利用引理 2.9 找到基本函数 g_1， g_2， $\ldots$， g_ℓ 使得 $\gamma = g_\ell \circ g_{\ell-1} \circ \cdots \circ g_1$，其中 $\ell \leq d^3$。

然后，我们使用引理 2.11 计算本质为 g_i 的毛毛虫图 C_i，对任意 i 均如此。最后，我们取所有这些毛毛虫图的不交并，并对所有 $i \in [1, \ell-1]$，将 C_i 的最右侧脊柱顶点与 C_{i+1} 的最左侧脊柱顶点相连。所得毛毛虫图的脊柱是这样一条路径：从 C_1 的最左侧脊柱顶点开始，到 C_ℓ 的最右侧脊柱顶点结束。

不难确认，这可以在关于 d 的多项式时间内完成，因为引理 2.9 和引理 2.11 中的算法也都在关于 d 的多项式时间内运行。最后，根据引理 2.10，所得毛毛虫图 C 确实具有签名 γ。◀

8 小度数调制器的核化二分定理

现在我们将注意力转向问题 d-COC/dist-to-$\mathcal{G}$，其中 $\mathcal{G}$ 是最大度至多为 c 的图类， c 为某个固定常数。到 $\mathcal{G}$ 的调节集称为度数-c 调制器。对于顶点覆盖，当 $c \leq 2$ 时已知存在多项式核，而当 $c \geq 3$ [48] 时则不太可能存在。

任何最大度至多为 1 的图，其连通分量的大小都至多为 2。因此，对于任意 $d \geq 2$，度数-1 调制器都是一个 d-coc 集合。这意味着当$d \geq 2$时，解大小参数化的核就已经足以解决问题。当 $d = 1$ 时，问题 d-COC 就是顶点覆盖；对于该问题，已知当参数化依据为度数-c 调制器且满足 $c \leq 1$ [39, 48] 时，存在多项式核。另一种论证是，当$c \leq 1$时，度数-c 调制器也是到毛毛虫图类的调节集，因此可以直接应用主定理 1。

为处理 $c \geq 3$ 时度数为 c 的调节集这一情形，我们证明对于任意 $d \geq 1$ 的取值， d-COC 在最大度至多为三的图上是 NP-完全的，因此，以度数为 c 的调节集为参数的任何核都不太可能存在。

因此，尤其是 $c = 2$ 这一情形值得关注。最大度为二的图只包含这样的连通分量：它们要么是路径（因此也是毛毛虫图），要么是环。当 $d \leq 3$ 时，任意 d-coc 集合都会使剩余的连通分量的最大度至多为二，因此，最小度数为 2 的调节集的大小至多为最小 d-coc 集合的大小。当 $d \geq 4$ 时，解的大小与最小度数为 2 的调节集的大小通常无法相互比较。

现在我们证明，当$\mathcal{G}$ 是最大度不超过二的图类时，对于所有$d \geq 1$，问题 d-COC/dist-to-$\mathcal{G}$ 都存在多项式核。事实上，我们证明问题 COC/d+dist-to-$\mathcal{G}$ 也存在多项式核，并通过将$\mathcal{G}$ 定义为连通分量为

46 通过到有界路径宽度的距离进行 COC 核化的分界线

环或毛毛虫图的图类，对结果作了轻微推广。作为第一步，我们证明$\mathcal{G}$ 中的图具有有界的极小d-阻塞集大小。注意，关于环上极小 1-阻断集大小上界的结论此前已由 [48, 中的定理 2]（独立地）证明。

▶**引理 8.1**。设 G 是一个图，其连通分量均为毛毛虫图或环，且$d \geq 1$ 是一个整数。那么，G 的任意极小 d-阻塞集的大小至多为三。

证明。设 X 是 G 的一个阻塞集，且大小至少为四。如果 X 的顶点分属 G的多个连通分量，那么由引理 5.1 结合 $X' = \emptyset$ 可知， X 不是一个极小的 d-阻塞集。否则， X 是G 的单个连通分量的一个阻塞集。因此，不失一般性，我们假设 G 是连通的。如果 G 是毛毛虫图，那么由定理 5.2 可知， X 不是一个极小的 d-阻塞集。

否则， $G = v_1, \ldots\ldots, v_n$ 是一个环。设 v_{λ_1}、 v_{λ_2}、 v_{λ_3}、 v_{λ_4} 是 X 的四个顶点，使得$\lambda_i < \lambda_{i+1}$ 对所有 $i \in [1,3]$ 均成立。那么，若我们选取 $X' = \{v_{\lambda_1}$、 $v_{\lambda_3}\}$，可以看到，从 G 中移除 X' 后会得到两个不同的连通分量，其中一个包含 v_{λ_2}，另一个包含 v_{λ_4}。因此，根据引理 5.1， X 不是极小的。◀

我们指出，引理 8.1 在如下意义下是最优的：对于任意 d ，$\mathcal{G}$中都存在一个大小为三的极小 d- 阻塞集。若 G 是一个包含 $d+2$ 个顶点的环，那么， $V(G)$ 的任意一个恰好包含三个元素的子集都是极小 d-阻塞集，因为 $V(G)$ 的任意一个恰好包含两个元素的子集都是 G的最优解。

现在，我们来构造 COC/d+dist-to-$\mathcal{G}$ 的多项式核。为得到该核，我们采用一个巧妙的技巧，该技巧常用于为顶点覆盖 $[8, 24, 35, 48]$构造多项式核。这个技巧是：减少$G - M$ 的连通分量数量（M 是某个图类的调节集）有时就足以找到一个多项式核，因为随后可以将数量关于参数呈多项式关系的顶点提升到调节集中，从而得到一个已知具有多项式核的问题实例。

▶**引理 8.2**。设 G 为这样一类图：其连通分量为环或毛毛虫图。则，COC/d+dist-to-G 存在多项式核。对于任意 $d \geq 1$ d-COC/dist-to-G，也存在多项式核。

证明。设 (G, d, k, M) 为输入实例。注意，在 G 中的图上，问题 d-COC 可在多项式时间内求解，因为它可在毛毛虫图上通过多项式时间求解，而对于环，多项式时间算法也是显然的（只需选取任意一个顶点，然后对剩余的毛毛虫图求最优解）。此外，图类 G 是遗传的。进一步地，根据引理 8.1，所考虑图类的极小 d-阻塞集大小至多为三。因此，我们可以应用推论 4.14，得到一个等价实例 (G_2, d, k_2, M)，使得 $G_2 - M$至多有 $((d-1) \cdot 3 + 1) \cdot |M|^3$ 个连通分量，并且 $k_2 \leq k$。

接下来，对于 $G_2 - M$ 的每个作为环的连通分量，我们将环上的任意一个顶点移入调节集 M，并将所得实例记为 (G_3, d, k_3, M_3)。这意味着 $|M_3| \leq ((d-1) \cdot 3 + 1) \cdot |M|^3 + |M|$，其关于 $|M| + d$ 是多项式的。现在， $G_3 - M_3$ 的每个连通分量都是毛毛虫图，因此 (G_3, d, k_3, M_3) 是 COC/d+pw-1 的一个实例。我们应用主定理 1，得到 COC/d+pw-1 的一个等价实例 (G_4, d, k_4, M_4)，其中 $|V(G_4)|$ 关于 $|M_3| + d$ 是多项式的，因此 $|V(G_4)|$ 关于 $|M| + d$ 是多项式的。现在，我们只需将 (G_4, d, k_4, M_4) 用作 COC/d+dist-to-G 的输出实例。该预处理算法不会改变 d 的值，因此也适用于 d-COC/dist-to-G。◀

在下界方面，我们利用顶点覆盖在最大度为三的平面图上是 NP 完全的这一事实 [26, 引理 1]，证明对于任意 $d \geq 2$， d-COC 在最大度为三的平面图上也是 NP 困难的。值得注意的是，对于某些 d，这一方向上的结果早已为人所知。例如， $d = 2$ 立方平面图上的 NP 困难性 [50], 以及最大度为三的二分图上的 NP 困难性 [4]均已得到证明。然而，据我们所知，对于 d-COC，当 $d \geq 3$ 时，文献中尚未明确证明其在最大度为三的图上是 NP 困难的。我们通过给出一个简单的多项式时间归约，证明该问题即使在最大度为三的平面图上也很难。

▶ **引理 8.3**。 对于任意 $d \geq 1$, d-COC ，在最大度不超过 *3* 的平面图上是NP完全的。

证明。 显然， d-COC 属于 NP。因此，我们只需证明它在最大度不超过 3 的平面图上是 NP困难的。对于 $d = 1$，该问题就是顶点覆盖，其在最大度为NP3 的平面图上的困难性已知 [26,引理 1]。

对于 $d \geq 2$，我们从最大度为 3 的平面图上的顶点覆盖问题归约。设 (G, k) 为输入实例。对于每条边 $e = uv \in E(G)$，我们将 uv 从 G中移除，并用路径 u， u_e， v_e， v 替换该边，其中 u_e和 v_e 是新顶点，不属于 G。此外，我们向图中添加一条新的路径 $p^1_{u,\ e}$，...， $p^{d-1}_{u\ e,}$ 以及一条新的路径 $p^1_{v,\ e}$，...， $p^{d-1}_{v\ e,}$，并在 $p^1_{u,\ e}$ 与 u_e之间，以及 $p^1_{v,\ e}$ 与 v_e之间添加边。设 G' 为所得图。

我们设定 $k' = k + |E(G)|$，而 d-COC 的输出实例为 (G', k')。注意， $V(G)$ 中的每个顶点在 G 和 G' 中的度都相同，并且 $V(G') \setminus V(G)$ 中的每个顶点的度至多为三。此外， G' 是平面图，因为用于替换边的操作保持平面性。现在我们证明这些实例等价，先证明正向。

▷ 命题 8.4. 若 (G, k) 是顶点覆盖的一个是实例，则 (G', k') 是 d-COC 的一个是实例。

证明。 设 S 是 G 的一个顶点覆盖，其大小至多为 k。我们说明如何构造一个 d-coc 集合 S'，它是 G' 的一个集合，大小至多为 k'。

首先，设定 $S' = S$。现在，遍历 G 的所有边 $e = uv$。边 e 在 G' 中被替换为路径 u, u_e, v_e, v。若 u 属于 S，则将顶点 v_e 加入 S'；否则，将顶点 u_e 加入 S'。显然，集合 S' 的大小至多为 k'。

现在，我们证明 S' 是 G'的一个 d-coc 集合。考虑 $G' - S'$ 的任意一个连通分量 C。如果 C 包含 V （G）中的一个顶点 v，那么 v 就不属于 S。考虑一条边 $e = uv \in E$ （G），由于 v 不属于 S，顶点 $u \in S$，因此顶点 v_e 属于 S'。因此， v 的所有邻点都属于 S'，且 C 的大小恰好为一。

否则， C 不包含 V（G）中的任何顶点。这意味着，对于某条边 $e = uv$ 和某个顶点 $u \in V$ （G）， C 必须包含顶点 $p^1_{u,\ e}$。如果 C 不包含顶点 u_e，那么 C 只包含 $d - 1$ 个顶点 $p^1_{u,\ e}$、.....、 $p^{d-1}_{u\ e}$。否则，如果 C 包含 u_e，那么我们知道 v_e 和 u 都属于 S'，因此 C 恰好包含顶点 $p^1_{u,\ e}$、.....、 $p^{d-1}_{u\ e,}$ 以及顶点 u_e，其大小为 d。◁

我们接下来证明等价关系的反向。

▷ 断言 8.5. 如果 (G', k') 是 d-COC 的是实例，那么 (G, k) 是顶点覆盖的是实例。

证明。 设 S' 是一个 d-coc 集合，其中 G' 的大小至多为 $k' = k+|E(G)|$。对于任意一条边 $e = uv \in E(G)$，我们将该边替换为一条路径 u， u_e， v_e， v在 G'中，使得 u_e和 v_e也有各自新构造的路径

长度为 $d-1$ 的路径连接到它们。注意，如果 S' 包含一条连接到 u_e、长度为$d-1$ 的路径上的顶点，我们可以用 u_e替换所选顶点。类似地，如果 S'包含一条连接到 v_e、长度为 $d-1$的路径上的顶点，我们可以用 v_e替换所选顶点。

因此，不失一般性地， S'不包含长度为 $d-1$的附加路径上的任何顶点。如果 S' 同时包含 u_e 和 v_e，我们可以从解中移除 u_e ，并改为添加顶点u。由于所得集合选中了 u 和 v_e ，这只会产生一个包含 u_e且大小为 d 的连通分量。因此，我们甚至可以假设 S' 至多包含 u_e 和 v_e中的一个。此外， S' 必须至少选中$\{u_e, v_e\}$中的一个顶点，否则 $G'-S'$ 中将存在一个连通分量，其中包含 u_e、 v_e，以及分别连接到 u_e和 v_e的两条长度为$d-1$ 的路径，并且该分量的大小将大于 d。

现在，考虑 S' 不包含顶点 u_e的情况。于是， S' 必须包含 u，否则 $G'-S'$ 中将存在一个连通分量，其中包含顶点 u_e、连接到 u_e的长度为 $d+1$ 的路径以及顶点 u。类似地，如果 S' 不包含 v_e，那么 S'必须包含 v。因此，对于将任意一条边 uv 替换为 u、 u_e、 v_e、 v 的情况，我们可以假设 S' 恰好包含$\{u_e, v_e\}$中的一个顶点，因此这些顶点中至少有一个未被 S'选中。于是，$\{u, v\}$中至少有一个必须被 S'选中。特别地， $S' \cap V(G)$包含 G的每条边中的一个顶点，因此 $S' \cap V(G)$是 G的顶点覆盖。此外， $S' \cap V(G)$的大小至多为 k，因为对于 $e = uv$中的每条边， S' 必须至少选中$\{u_e, v_e\}$中的一个顶点。◁

该引理现在可由断言 6.14 和 6.15 得出。 ◀

现在，我们可以给出以度为 c 的调节集大小为参数的 d-COC 核化二分定理。

▶**定理 1.1**。 当 G 是最大度数至多为2 的图类时，问题COC/d+dist-to-G 存在多项式核；对于每个 $d \geq 1$， d-COC/dist-to-G，也存在多项式核。当 G 是最大度数为3的平面图类，且 d 为任意正整数时，问题 d-COC/dist-to-G 不存在任何大小的核，除非P =≠= NP。

证明。如果 G 是最大度数至多为 2 的图类，那么根据引理 8.2，存在一个多项式核（注意，最大度数至多为 2 的图的连通分量要么是环，要么是路径，而路径是毛毛虫图）。如果 G 是最大度数为 3 的平面图类，那么引理 8.3 表明问题 d-COC 在 G 上是 NP 困难的。特别地，对于 G 中图上的 d-COC 的任意实例（G, k）， （$G, k, \emptyset$）都是 d-COC/dist-to-G 的一个实例，其参数值为 0。若存在一个核（无论大小），那么我们就能在多项式时间内将（G, k）归约为一个常数大小的等价实例，从而能在图类 G 上以多项式时间求解 d-COC。因此，除非 P =NP，否则该问题不存在核。◀

9 参数中 d 的必要性

在主定理 1 和定理 1.1 的正向结果中，我们以到某个图类$\mathcal{G}$ 的距离作为参数，而 d 要么是参数的一部分，要么甚至是常数。这不禁让人提出问题：当仅以 d 或到$\mathcal{G}$的距离作为参数时，是否仍能为 COC 获得多项式核？显然，仅以 d 作为参数并不合适，因为当 d 为常数 1 时，该问题已经是 NP 困难的。因此，本节研究仅以到某个图类$\mathcal{G}$的距离作为参数时，该问题的复杂度。事实证明，以到毛毛虫

森林的距离作为参数的 COC 是 W[1]-困难的。这与主定理 1 形成对比，主定理 1 表明，当 d 也作为参数的一部分时，存在多项式核。

9.1 到毛毛虫森林的距离

现在，我们正式证明 COC/pw-1 的困难性结果；该问题是以到毛毛虫森林的距离为参数的 COC。我们的归约核心与 Ganian 等人 [25, 定理 2] 针对有界度删除问题的归约相同，但有一个关键区别，我们将在后文详述。该归约源自一元多维宽松子集和（UMRSS）。其输入包括一个整数 k、一个由向量组成的集合 $S = \{s_1, \ldots, s_n\}$，其中每个向量 $s_i \in \mathbb{N}^k$、一个目标向量 $t \in \mathbb{N}^k$以及一个整数 k'。所有整数均采用一元编码。任务是判定是否存在一个子集 $S' \subseteq S$，使得 $|S'| \leq k'$，并满足$\sum_{s \in S'} s \geq t$。UMRSS 关于 $k + k'$ [25, 参数化时是 W[1]-困难的（引理 4]）。

▶ **定理 9.1.** COC/pw-1是 W[1]-困难。

证明。我们给出一个从 UMRSS 出发的参数化归约。设 (k, S, t, k') 是 UMRSS 的输入实例。令 $\beta = \max\{\max s \mid s \in S\}$，也就是说，$\beta$ 是集合 S 中任意向量里出现的最大整数。令 $\gamma = (k' + 1) \cdot k \cdot \beta$。我们首先构造一个图，其中对于每个 $1 \leq i \leq k$ 都包含顶点 $u_i^1, \ldots, u_i^{k'+1}$。接下来，对于每个 $s \in S$，我们引入 γ 个顶点 $s_1, \ldots, s_\gamma$。我们令每个顶点 u_i^j 与 $s_1, \ldots, s_\gamma$ 中的 $s[i]$ 个顶点相邻，使得 $s_1, \ldots, s_\gamma$ 中没有任何顶点与两个顶点 u_j^i 相邻。注意，按照我们选择 γ 的方式，这总是可行的。令 $\hat{G}$ 为目前构造出的图。令 $d = \Delta(\hat{G}) + 1$，也就是说，d 是 $\hat{G}$ 的最大度加一。

接下来，对于每个 $s \in S$ 和每个 $1 \leq i < \gamma$，我们向图中添加一个新的星图，它有 $d - 1$ 个叶节点，中心顶点为 $\hat{s}_i$。我们使 s_i 和 s_{i+1} 与 $\hat{s}_i$ 相邻。最后，我们为每个顶点 u_i^j 添加足够多的悬挂顶点，使该顶点总共有 $d + t[i] - 1$ 个邻居。注意，这始终可行，因为 u_j^i 在图 $\hat{G}$ 和 $t[i] \geq 0$ 中至多有 $d - 1$ 个邻居。将所得图称为 G。注意，集合 $M = \{u_i^j \mid i \in [1, k], j \in [1, k' + 1]\}$ 是一个到毛毛虫森林的调节集，因此输出实例的参数为 $k \cdot (k' + 1)$。我们将输出 COC/pw-1 实例的解的大小 ℓ 设为 $(\gamma - 1) \cdot |S| + k'$。于是，输出实例为 (G, d, ℓ, M)。注意，由于输入实例中的所有整数均以一元编码表示，因此该归约可以足够快地完成。现在我们通过两个独立的断言证明该归约的正确性。

▷ 命题 9.2. 若输入实例 (k, S, t, k') 是 UMRSS 的是实例，则 (G, d, ℓ, M) 是 COC/pw-1 的是实例。
证明。令 $S' \subseteq S$ 为满足 $\sum_{s \in S'} s \geq t$ 且 $|S'| \leq k'$ 的向量集合。我们如下构造一个 $V(G)$ 的选择 S^*。对于每个 $s \in S'$，我们为所有 $i \in [1, \gamma]$ 选择顶点 s_i；对于每个 $s \in S \setminus S'$，我们为所有 $i \in [1, \gamma - 1]$ 选择顶点$\hat{s}_i$。注意，所得选择S^* 的大小恰好为 $|S'| \cdot \gamma + |S \setminus S'| \cdot (\gamma - 1) \leq |S| \cdot (\gamma - 1) + k'$。因此，剩下的唯一任务就是论证 $G - S^*$ 没有大小大于 d 的连通分量。

考虑 Z 中任意一个 $G - S^*$ 的连通分量，若 Z 不含顶点 u_j^i，则 Z 必定是一个孤立顶点 $s_{i'}$，或者 Z 含有某个 $s \in S'$ 对应的顶点$\hat{s}_{i'}$。在后一种情况下，由于 $s_{i'}$ 和 $s_{i'+1}$ 都属于 S^*，连通分量 Z 的大小恰好为 d。因此，现在考虑 Z 含有顶点 u_j^i 的情形。我们有，u_j^i 在 G 中恰好有 $d + t[i] - 1$ 个邻居。此外，对于 u_j^i 的每个未被 S^* 选中的邻居，该邻居在 $G - S^*$ 中的度恰好为一。事实上，u_j^i 的邻居要么是 u_j^i 的悬挂顶点，要么是顶点 $s_{i'}$（对某个 i' 而言）；在后一种情况下，除 u_j^i 之外，$s_{i'}$ 的邻居

都属于 S^*。于是，由于 $\sum_{s\in S'} s[i] \geq t[i]$，并且我们知道，对于每个 $s \in S'$， u_j^i 在图 G 的集合 $s_1, \ldots\ldots, s_\gamma$ 中恰好有 $s[i]$ 个邻居，且这些邻居都被 S^* 选中，因此我们看到，至少有 $t[i]$ 个 u_j^i 的邻居被选中。因此，在 $G - S^*$ 中，顶点 u_j^i 至多有 $d-1$ 个邻居，故 Z 的大小至多为 d。◁

▷ 断言 9.3. 若实例 (G, d, ℓ, M) 是 COC/pw-1 的是实例，则 (k, S, t, k') 是 UMRSS 的是实例。

证明。设 $S^* \subseteq V(G)$ 是一个大小至多为 ℓ 的 d-coc 顶点集合 G。考虑任意一个 $s \in S$。那么，对于每个 $i \in [1, \gamma-1]$ 顶点 $\hat{s}_i$，都有 $d-1$ 个悬挂顶点，因此 S^* 必须从 $\hat{s}_i$、s_i 或 $\hat{s}_i$ 的某个悬挂顶点中至少选择一个。特别地，这意味着 S^* 必须至少选择 $(\gamma-1)\cdot|S|$ 个不属于 M 的顶点。因此， S^* 至多选择 k' 个不属于M的顶点，于是存在一个 j，使得 S^* 未选择 u_1^j 、 ..... 、 u_k^j 中的任何顶点。显然， S^* 可以选择 u_i^j，而不是选择 u_i^j 的某个悬挂顶点，因此我们也可以假设 S^* 没有选择 G 中 u_i^j 的任何悬挂顶点。

在继续之前，我们想分析对于任意 $s \in S$， S^* 可能会选取 s_1、.....、s_γ 以及 $\hat{}s_1$、.....、$\hat{}s_{\gamma-1}$ 中的哪些顶点。显然，如果 S^* 选取了某个顶点 $\hat{s}_{i'}$（对于某个 i'）的一个悬挂顶点，我们可以将这个悬挂顶点从 S^* 中移除，改为选取 $\hat{s}_{i'}$。因此，我们可以假设 S^* 不选取顶点 $\hat{s}_1$、.....、$\hat{}s_{\gamma-1}$的任何悬挂顶点。如果 S^* 从 s_1、.....、s_γ、$\hat{s}_1$、.....、$\hat{}s_{\gamma-1}$中选取了 γ个顶点，那么我们可以假设 S^* 恰好选取 s_1、.....、 s_γ，因为 $\hat{s}_1$、.....、$\hat{}s_{\gamma-1}$ 中的顶点均不与 M中的顶点相邻。否则，若 S^* 从 s_1、.....、s_γ、$\hat{s}_1$、.....、$\hat{}s_{\gamma-1}$中选取了 $\gamma-1$个顶点，我们现在要论证，此时 S^* 必须恰好选取顶点 $\hat{}s_1$、.....、$\hat{}s_{\gamma-1}$。否则，假设 S^* 对于某个 i'选取了顶点 $s_{i'}$。那么，对于所有 $j < i'$，$\hat{s}_j$、$\hat{s}_j$的 $d-1$ 个悬挂顶点以及顶点 s_j 在 G中构成一个大小为 $d+1$ 的连通子图。此外，对于所有 $j \geq i'$，$\hat{s}_j$、$\hat{s}_j$的悬挂顶点以及顶点 s_{j+1} 在 G中构成一个大小为 $d+1$ 的连通子图。所有这些子图彼此顶点不相交。这意味着，选取了 $s_{i'}$个顶点的 S^*必须进一步选取 s_1、.....、s_γ 以及 $\hat{s}_1$、.....、$\hat{}s_{\gamma-1}$中的 $\gamma-1$ 个顶点，这与至多选取其中 $\gamma-1$ 个顶点相矛盾。

因此，不失一般性地，我们可以假设对于任意 $s \in S$、 S^* ，要么选择所有顶点 $s_1, \ldots, s_\gamma$，要么选择所有顶点 $\hat{s}_1, \ldots \hat{}s_{\gamma-1}$。我们通过当且仅当 S^* 选择了 γ顶点 $s_1, \ldots, s_\gamma$ 时选择向量 $s \in S$，来构造 UMRSS 的解，并将所得的向量集合称为 S'。根据 ℓ 的选择，我们看到 S' 的大小至多为 k'。现在，考虑任意 $i \in [1,k]$。我们有， u_i^j 在 G 中有 $d+t[i]-1$ 个邻居，而且 u_i^j 以及 u_i^j 在 G 中的悬挂顶点都没有被 S^* 选中。因此， S^* 必须至少选择 $t[i]$ 个邻点，这些邻点对于某个 $s \in S'$ 具有 s_i 的形式。由于 S^* 包含 s_1 中的 $s[i]$ 个顶点 $, \ldots, s_\gamma$，并且当且仅当 $s \in S'$ 时它们是 u_i^j 的邻点，我们确实得到 $\sum_{s\in S'} s[i] \geq t[i]$。◁

根据断言 9.2 和 9.3，归约是正确的，因此定理得证。 ◀

我们的归约和证明与 Ganian 等人 [25]的工作的主要区别在于，他们的归约不适用于问题 COC/pw-1，主要是因为他们用来编码 S 集合的构件不是毛毛虫图（而是有界深度的树）。因此，我们在此的贡献是创建了新的构件来编码 S集合，从而使归约适用于我们的问题。

9.2 顶点覆盖参数化

根据前面的结果，不能从主定理 1 中删除参数 d。然而，即使不将 d 纳入参数，针对不同结构参数仍可能存在多项式核。接下来，我们将证明，即使参数是

顶点覆盖的大小加上<solution size 解的大小，COC 也不具有多项式核（除非 NP ⊆ coNP/poly）。[7] 我们将以顶点覆盖的大小（该顶点覆盖作为输入的一部分给出）加上解的大小为参数化的 COC 记作 COC/k+vc。该结果意味着，对于“几乎所有”正则结构参数化，COC 都不具有多项式核。

我们通过从精确集合覆盖问题进行多项式参数变换，证明了核化下界。该问题的输入由一个全集 $\mathcal{U}$、一个集合族 $\mathcal{F} \subseteq \binom{\mathcal{U}}{t}$（其中某个整数为 t）以及一个整数 k组成。问题是：是否存在一个大小为 k 的子集 $\mathcal{F}' \subseteq \mathcal{F}$，使得 $\mathcal{U}$ 中的每个元素恰好属于 $\mathcal{F}'$中的一个集合？Dell 和 Marx [18] 针对固定常数 t 研究了这个问题，并将其称为完美t-集合匹配。他们证明，除非 NP ⊆coNP/poly [18, 定理 1.2]，否则当 $t \geq 3$ 时，该问题不存在大小为 $O(k^{t-\varepsilon})$ 的压缩，其中任意 $\varepsilon > 0$ 均如此。注意，他们以核的形式表述了这一结果，但其底层方法同样排除了压缩。精确集合覆盖问题的压缩算法也可用于固定 t的情形。由于该问题的任何是实例都必须满足$|\mathcal{U}| = k \cdot t = O(k)$（当 t 固定时），他们的结果意味着，以全集大小 $|\mathcal{U}|$为参数的精确集合覆盖问题（记作精确集合覆盖/U）不存在多项式压缩，除非 NP ⊆ coNP/poly。

▶ **引理 9.4**（由 [18, 定理 2] 推出）。精确集合覆盖/U 除非 NP ⊆coNP/poly，否则不可能进行多项式压缩。

现在，我们以引理 9.4 为起点，进行多项式参数变换。从技术层面来说，我们的归约受到 Giannopoulou 等人 [27] 所做归约的启发。他们从同一个问题出发（固定 t），并证明了三个不同的以顶点覆盖为参数的 $\mathcal{F}$-子式删除问题的压缩下界。我们考虑的问题并非 $\mathcal{F}$-子式删除问题（因为 d 不是常数），但他们构造中的相同整体结构仍可较好地扩展到 COC/k+vc。

为便于阐述，我们先介绍 COC/k+vc 的一个带注释变体。在 A-COC/k+vc 问题中，输入由一个图 G、整数 k、 d、 G的一个顶点覆盖 M以及注释 $A \subseteq \binom{M}{2}$组成。目标是判断是否存在一个大小至多为 k 的集合 $S \subseteq V(G)$，使得 (1) $G - S$ 的每个连通分量的大小至多为 d；(2) 对于每个 $\{u, v\} \in A$，都有 $S \cap \{u, v\} \neq \emptyset$。直观地说，一个解需要同时解决两个问题：图 G上的 d-COC，以及边集为 A的图上的顶点覆盖。参数为 $|M| + k$。

▶**引理 9.5**。问题 A-COC/k+vc 不存在多项式压缩，除非NP ⊆coNP/poly。

证明。设 精确集合覆盖/U 的输入实例为 $(\mathcal{U},\mathcal{F}\ k)$，其中 $\mathcal{F}$中的每个集合大小均为 $t \leq |\mathcal{U}|$。为方便起见，记 $\mathcal{U} = \{u_1, \ldots, u_n\}$。如果我们没有$k = \frac{n}{t}$，显然这是一个否实例，因此我们可以输出一个常数大小的 A-COC/k+vc否实例。

否则，我们有 $k = \frac{n}{t}$，因此 $n = k \cdot t$。现在描述我们的 COC 实例的输出图 G：从空图开始，并初始设置 $A = \emptyset$。对于每个 $i \in [1, n]$和 $j \in [1, k]$，我们创建一个顶点 u_i^j。直观地说，每个顶点 u_i^j 代表全集元素 u_i；然而，出于技术原因，我们为每个这样的代表元设置 k 个副本。对于

[7] 请记住，尽管以解的大小为参数时， d-COC 有多项式核，但问题 COC 是 W[1]-困难的。

52 通过到有界路径宽度的距离对 COC 核化的分界线

所有 $j \in [1,k]$ 集合 $C^j = \{u_i^j \mid i \in [1,n]\}$，以及所有 $i \in [1,n]$ 设置 $R_i = \{u_i^j \mid j \in [1,k]\}$。我们将 C^j 称为构造的第 j 列，将 R_i 称为构造的第 i 行。对于每个 $j \in [1,k]$，我们将顶点 C^j 构成一个团。此外，对于每个 $i \in [1,n]$ 以及每一对顶点 $\{u_i^j, u_i^{j'}\}$（其中 $j \neq j'$），我们将注释 $\{u_i^j, u_i^{j'}\}$ 添加到集合 A 中。设置 $M = \bigcup_{j\in[1,k]} C^j = \bigcup_{i\in[1,n]} R_i$。

现在，对于每个 $j \in [1,k]$ 和每个 $F \in \mathbb{F}$，我们添加一个顶点 v_F^j。我们令 v_F^j 与集合 $\{u_i^j \mid i \in [1,n],\ u_i \notin F\}$ 中的所有顶点相邻。换言之，v_F^j 与 C^j 中不代表集合 F 的元素的所有顶点相邻。我们令 $d = |\mathbb{F}| + t - 1$，且 $k' = n \cdot (k-1)$。我们可以假设 $t \geq 2$，因为当 $t = 1$ 时，输入问题可在多项式时间内轻松求解，因此 $d \geq 1$。A-COC/k+vc 的输出实例为 $(G,\ d,\ k,\ M,\ A)$。注意，集合 M 是 G 的一个顶点覆盖，其大小恰为 $n \cdot k = n \cdot \binom{n}{t} \in O(n^2)$。此外，$k' = n \cdot (k-1) = n \cdot (\binom{n}{t} - 1) \in O(n^2)$。因此，输出实例的参数 $k' + |M|$ 有界于 n 的一个多项式。现在我们证明正确性的正向。

▷ 断言 9.6. 若 $(\mathcal{U},\ \mathcal{F}\ k)$ 是精确集合覆盖/U的一个是实例，则 $(G,\ d,\ k',\ M,\ A)$ 是 A-COC/k+vc的一个是实例。

证明。令 $S = \{F_1,\ \ldots\ldots,\ F_k\} \subseteq \mathcal{F}$ 为 $\mathcal{U}$ 的一个精确集合覆盖，也就是说，这是一个集合族，其中$\mathcal{U}$ 的每个元素恰好出现在 S 的一个集合中。接着，定义

$$S' = \{u_i^j \mid i \in [1,n], j \in [1,k], u_i \notin F_j\}.$$

由于每个全集元素 u_i 恰好属于一个集合 F_j，因此对每个 $i \in [1,n]$，恰好存在一个 j，使得 u_i^j 未被 S' 选中。因此，$|S'| = k'$。对于集合 A 中的注释，注意到这些注释均得到满足，因为每行 R_i 中除恰好一个顶点外，其余顶点都被选中。

接下来还需说明，S' 也是 G 的一个 d-coc 集合。考虑 $G - S'$ 的任意一个连通分量 C。如果 C 的大小为一，则由 $d \geq 1$ 可知结论成立；因此考虑 C 的大小大于一的情形。于是，由于 $G - M$ 是一个独立集，C 必须包含 M 中的一个顶点。令 u_i^j 为 C 中包含的 M 的一个顶点。现在注意到，M 的一个顶点仅与同一列中属于 M 的其他顶点相连，而 $G - M$ 的顶点也仅与同一列中的顶点相邻。因此，C 只能由来自列 C^j 的 M 顶点以及某个 $F \in \mathbb{F}$ 的顶点 v_F^j 组成。注意到，S' 选中了列 C^j 中不对应于集合 F_j元素的所有顶点，因此 C^j 中至多有 t 个顶点未被选中。$v_{F_j}^j$ 的邻域恰好由那些不对应于集合 F_j 元素的顶点组成，因此 $v_{F_j}^j$ 的所有邻居都被 S' 选中。最后，这意味着 C 至多包含 $G - M$ 中的 $|\mathbb{F}| - 1$ 个顶点，因此 C 的大小至多为 $d = |\mathbb{F}| - 1 + t$。◁

现在我们证明正确性的反向。

▷ 断言 9.7. 若 $(G,\ d,\ k,\ M,\ A)$ 是 A-COC/k+vc 的是实例，则 $(\mathcal{U},\ \mathcal{F}\ k')$ 是精确集合覆盖/U 的是实例。

证明。令 S' 为A-COC/k+vc输出实例的一个解。固定任意一个$i \in [1,n]$。由于行 R_i 中每一对不同顶点在 A中都有注释，因此S' 必须选取 R_i中的所有顶点，至多有一个例外。鉴于 $k' = n \cdot (k-1)$，且共有 n 行，我们可以观察到 S' 必须恰好选取 $k-1$ 个 R_i的顶点，因此恰好有一个 j 满足 u_i^j 不属于 S'。

现在，假设存在某一列 C^j，使得 S' 从 C^j中选出的顶点少于 $n - t$ 个。回想一下，C^j 恰好包含 n 个顶点。设 u_i^j 是 C^j中不属于 S'的一个顶点。回想一下，C^j 是 G中的一个团，因此 u_i^j 的连通分量

在 $G - S'$ 中至少包含 $t+1$ 个 C^j的顶点。接下来，考虑任意一个属于 $F \in \mathcal{F}$ 的顶点 v_F^j。由于 v_F^j 在 C^j中有 $n-t$ 个邻点，并且 $|C^j| = n$， v_F^j 在 C^j中有一个未被选中的邻点，因此它属于 $G - S'$ 中 u_i^j 的连通分量。因此， $G' - S'$ 中 u_i^j 的连通分量至少包含 $|\mathcal{F}| + t + 1 > d$ 个顶点，这就产生了矛盾。

如果存在某列 C^j，使得 S' 选取了 C^j 中超过 $n-t$ 个顶点，那么注意到，由于 $k' = n \cdot (k-1)$ $= nk - kt = k(n-t)$，必定存在一列 $C^{j'}$，其中 S' 选取的顶点少于 $n-t$ 个；但这与前一段的结论矛盾。

因此， S' 恰好从每一列 C^j 中选取 $n-t$ 个顶点。对于每个全集元素u_i，又恰好存在一列 C^j，使得 u_i^j 未被选取，因此只需证明 C^j 中未被 S' 选取的那些顶点构成一个 $\mathcal{F}$ 集合。事实上，若如此，则令每个 $j \in [1,k]$ 的 F_j 为 $\{u_i \mid u_i^j \in C^j \setminus S'\}$，并令 $S = \{F_1, \ldots, F_k\}$，则每个元素 $u_i \in \mathcal{U}$ 都恰好出现在 S 的一个集合中，因此该输入实例是一个是实例。

因此，固定任意一个 $j \in [1,k]$，我们将证明 $F_j = \{u_i \mid u_i^j \in C^j \setminus S'\}$ 属于 $\mathcal{F}$。令 u_i^j 为 $C^j \setminus S'$ 的一个顶点，并令 C 为 u_i^j 在 $G - S'$ 中所在的连通分量。回顾一下， S' 从每一列中恰好选取 $n-t$ 个顶点，因此 C 恰好包含 C^j 的 t 个顶点。为导出矛盾，假设 $F_j \notin \mathcal{F}$。考虑任意一个 $F \in \mathcal{F}$。那么，顶点 v_F^j 与 C^j 中所有不代表 F 的 $n-t$ 个顶点均相邻。特别地，由于 $F \neq F_j$，我们有 v_F^j 与 $C^j \setminus S'$ 的一个顶点相邻，因此 v_F^j 是 C 的一部分。所以， C 的大小为 $|\mathcal{F}| + t > d$，这就产生了矛盾。◁

由命题 9.6 和 9.7 可知，该多项式参数变换是正确的，而引理 9.4 则完成了证明。◀

最后，我们需要提供一个从带标注问题 A-COC/k+vc 到 COC/k+vc 的多项式参数变换。为此，我们可以采用引理 8.3 的核心思路。

▶**定理 9.8.** 问题 COC/k+vc 不允许多项式压缩，除非NP $\subseteq$coNP/poly。

证明。我们给出一个从 A-COC/k+vc 出发的多项式参数变换。令(G, k, d, M, A) 为输入实例。注意，我们可以假设 $k, d \leq |V(G)|$，因此这些数值的大小均为输入规模的多项式。

初始设定 $M' = M$。对于每个 $a = \{u, v\} \in A$，我们向图中添加两个顶点 a_u 和 a_v。令 a_v 与 v 相邻， a_u 与 u 相邻，并在 a_u 和 a_v 之间添加一条边。此外，分别向 a_u 和 a_v 添加 $d-1$ 个悬挂顶点。我们还将 a_u 和 a_v 添加到 M' 中。将所得图记为 G'。

显然， M' 是 G' 的顶点覆盖，因为在处理注释 a时，我们将 a_u 和 a_v 加入了该顶点覆盖。令 $k' = k + |A|$。设输出实例为（G'， k'， d， M'）。 M'的大小至多为 $|M| + 2|A| \leq |M| + 2|M|^2 \in O(|M|^2)$（回想一下，我们只允许在 M中的顶点对之间添加注释）。此外， $k' = k + |A| \leq k + |M|^2$。因此，$|M'|+k'$关于 $|M|+k$是多项式有界的。显然，参数变换也能在多项式时间内运行，因为我们只向图中添加了 $2d \cdot |A| \leq 2|V(G)| \cdot |V(G)|^2$ 个新顶点。因此，剩下的就是证明正确性。我们先证明正向。

▷ 断言 9.9. 若 (G, k, d, M, A) 是 A-COC/k+vc 的是实例，则 (G', k', d, M') 是 COC/k+vc 的是实例。

证明。 令 S 为输入实例的一个解，并且初始设定 $S' = S$。对于每个$a = \{u, v\} \in A$ 我们有 S 至少选择了 u 或 v中的一个。如果 S 选择了 u，那么我们将 a_v

加入到 S' 中；否则，如果 S 未选择 u，则将 a_u 加入到 S' 中。所得集合 S' 的大小至多为$k+|A|$，因为我们通过对每个 $a \in A$恰好添加一个顶点来修改 S。

显然， $G'-S'$ 中仅包含 V顶点的每个连通分量（G）的大小至多为 d。因此，考虑 $G'-S'$ 的某个连通分量 C，其中包含一个 V（G'）顶点\ V（G）。每个这样的分量若不只是一个孤立顶点，也会包含一个顶点 a_u 或 a_v，对应某个注释$\{u,v\} \in A$。因此，考虑任意注释$\{u,v\} \in A$。

如果 $u \in S$，那么我们有 $a_v \in S'$，且 $a_u \notin S'$。由于 a_u 的唯一邻点是 a_v、 a_u的悬挂顶点以及顶点 u，因此 $G'-S'$中 a_u 的连通分量只包含 a_u 和 a_u的 $d-1$ 个悬挂顶点。

类似地，如果 $u \notin S$，那么我们有 $a_u \in S'$。此外，由于$\{u,v\} \in A$，我们有 $v \in S$。因此，选中了 a_v 的邻居 v 和 a_u，而$G'-S'$ 中包含 a_v 的连通分量大小至多为 d。◁

▷断言 9.10. 若 (G'， k'， d， M') 是 COC/k+vc 的是实例，则 (G， k， d， M， A) 是 A-COC/k+vc 的是实例。

证明。令 S' 为该输出实例的一个解。对于每个 $a=\{u,v\}$，输出图G' 都包含 a_u、 a_v，以及分别对应于 a_u 和 a_v的 $d-1$ 个悬挂顶点。因此， S' 必须选取 a_u、 a_v，或 a_u 或 a_v的一个悬挂顶点。不难看出，如果 S' 选取了 a_v的一个悬挂顶点，我们可以改为选取 a_v。因此，可以假设 S' 至少包含$\{a_u,a_v\}$中的一个，并且 S'不包含 a_u 或 a_v的任何悬挂顶点。

接下来，假设 S' 同时包含 a_u 和 a_v。在这种情况下，我们可以选取 v 而不是a_v （或者等价地，选取 u 而不是 a_u）。因此，可以假设 S' 在$\{a_u,a_v\}$中恰好选取一个。于是，考虑到$\{a_u,a_v\}$中有一个（设为 a_u）未被选取， S'必须选取顶点 u；否则， a_u、 a_u的 $d-1$ 个悬挂顶点以及顶点 u将导致 $G'-S'$中出现一个大小为 $d+1$ 的连通分量。因此，我们可以看出， $S' \cap V$（G）在A中的每个注释里都选取一个顶点。此外， $S' \cap V$（G）必须是 G 的一个 d-coc 集合，因为 G 是 G'的子图。最后， S' 对每个注释 $a=\{u,v\}$都从$\{a_u,a_v\}$中选取一个顶点，这一事实表明$|S' \cap V$（G）$| \leq k$。◁

结合命题 9.9 和 9.10 以及引理 9.5，即可证明该定理。 ◀

在定理 9.1 中，我们证明了，去掉参数 d 后，问题将变为 W[1]-困难问题。相比之下，定理 9.8 仅排除了多项式压缩。事实证明，这并非我们的结果存在局限：问题 COC/vc（以输入中给定的顶点覆盖大小为参数的 COC）属于 FPT。因此，定理 9.8 的结果无法加强为证明该问题是 W[1]-困难的。接下来，我们介绍用于证明这一点的简单分支算法。

▶**定理 9.11**。COC/vc 属于FPT。

证明。 设输入实例为 (G,d,k,M)。我们对M使用以下颜色进行的所有（非真）着色进行分支： $[0,|M|]$。

令 Γ 为一个固定分支的着色。令 S_1 为满足 $\Gamma(v)=0$的那些顶点 $v \in M$，这表明 v 属于解集。令 $S_2=S_1$。遍历所有 $w \in G-M$。若存在 m_1、 $m_2 \in N(w)$，使得 $\Gamma(m_1) \neq 0$且$\Gamma(m_2) \neq 0$ 并且 $\Gamma(m_1) \neq \Gamma(m_2)$，则将 w 加入 S_2。若 S_2 是一个大小至多为 k 的 d-coc 集 G，则输出“是”。

如果没有任何分支输出该实例是是实例，则输出该实例是否实例。现在我们来证明该算法的正确性。

▷ 命题 9.12. 如果 $(G,\ d,\ k,\ M)$ 是否实例，则算法输出 no。

证明。 由于算法只有在找到一个 d-coc 集合，其中 G 的大小至多为 k 时才会输出 yes，因此，如果输入实例是否实例，算法总会输出 no。 ◁

最后，我们需要证明，当输入实例是是实例时，算法会输出 yes。

▷ 命题 9.13. 如果 $(G,\ d,\ k,\ M)$ 是是实例，则算法输出 yes。

证明。 设 S 是一个 d-coc 集合，其中 G 的大小至多为 k。设 $G-S$中包含 M的顶点的连通分量为 $C_1,\ \ldots\ldots,\ C_t$(显然 $t \leq |M|$)。

现在证明，我们可以假设 S 不会选择 $G-M$ 中满足以下条件的顶点 w：对于某个 i，有$N(w) \setminus S \subseteq V(C_i)$。为导出矛盾，假设存在这样的 $w \in S$。如果$N(w)$，则无需选择 w 。否则，我们可以从$N(w) \setminus S$ 中任意选择一个顶点，而不选择 w。这不会增加连通分量的大小，因为 $N(w) \setminus S$ 中的所有顶点在 $G-S$ 中已经属于同一个连通分量。因此，我们可以假设 $G-M$ 中所有被选中的顶点，在 $C_1, \ldots, C_t$ 的两个不同连通分量中都有未选中的邻点。

显然，存在一种对 M 的着色 Γ，使用颜色 $[0, |M|]$，使得 $S \cap M$ 中的所有顶点都获得颜色 0，而对于所有 $i \in [1, t]$，C_i 中的所有顶点都获得颜色 i 。我们来检查此分支 Γ 的算法。首先，它创建集合 S_1，显然该集合满足 $S_1 = S \cap M$。然后，它通过选取 $G-M$ 中那些具有不同颜色邻点的顶点来创建S_2，这些顶点对应于在 $G-S$ 的不同连通分量中有邻点的顶点。不难确认，我们有 $S_2 \subseteq S$。然而，由于 S 不会选择 $G-M$ 中仅在 $G-S$ 的一个连通分量中有未选中邻点的顶点 w，因此可得 $S = S_2$，算法输出 yes。 ◁

显然，所述算法会终止，运行时间为 $f(|M|) \cdot |V(G)|^{O(1)}$ f或某个可计算函数 f，且根据断言 9.12 和 9.13，该算法是正确的。◀

10 问题定义

本节汇集并正式重述本文所考虑问题的定义。

10.1 分量阶连通性变体

回顾一下，对于整数 $d \geq 1$ 和图 G，d-coc 集合 G 是指满足以下条件的集合 S：$G-S$ 的每个连通分量的大小至多为 d。

分量阶连通性（COC）

输入：图 G，整数 $d \geq 1$，整数 k问题：是否存在一个 d-coc 集合 S，由 G 构成，且满足 $|S| \leq k$?

如果 d 是固定常数，我们则得到问题 d-COC。对于每个固定的 d，我们都有一个不同的问题。

56 通过到有界路径宽度的距离实现 COC 核化的分界线

d-COC
输入：图 G，整数 k问题：是否存在一个 d-coc 集合 S，满足 G 并且具有 $|S| \leq k$?

顶点覆盖（VC）问题与 1-COC 相同。现在，我们继续讨论基本的参数化问题。首先，我们考虑按 $d+k$ 进行参数化。

COC/d+k
输入：图 G，整数 $d \geq 1$，整数 $k \geq 0$参数：
$d+k$问题：是否存在一个 d-coc 集合 S，使得 G 满足 $|S| \leq k$?

接下来，我们考虑结构参数。下文中，$\mathcal{G}$ 始终表示一个固定图类，我们按调节集 M 的大小进行参数化，其中该调节集用于将图变为 $\mathcal{G}$ 中的图。对于 COC 问题，我们还将 d 纳入参数。

COC/d+dist-to-$\mathcal{G}$
输入：图 G，整数 $d \geq 1$，整数 k，集合 M，使得 $G - M \in \mathcal{G}$参数：
$d+|M|$问题：是否存在一个 d-coc 集合 S，它是 G 的集合且满足 $|S| \leq k$?

当我们考虑问题 d-COC 而不是 COC 时，d 是一个常数，因此自然不再是参数的一部分。

d-COC/dist-to-$\mathcal{G}$
输入：图 G，整数 k，集合 M，使得 $G - M \in \mathcal{G}$参数：
$|M|$问题：是否存在一个 d-coc 集合 S，它是 G 的集合且满足 $|S| \leq k$?

主定理 1 中的结果针对问题 COC/d+pw-1，即 COC/d+dist-to-$\mathcal{G}$，其中 $\mathcal{G}$ 固定为路径宽度至多为 1 的图类，也就是毛毛虫森林类。

COC/d+pw-1
输入：图 G，整数 $d \geq 1$，整数 k，集合 M，满足 $G-M$ 是一个毛毛虫森林。
参数：$d+|M|$。问题：是否存在一个 d-coc 集合 S，属于 G 且满足 $|S| \leq k$?

该结果也适用于 d 固定的问题。

d-COC/pw-1
输入：图 G、整数 k、集合 M，满足 $G-M$ 是毛毛虫森林。参数：$|M|$。
问题：是否存在一个 coc - d 集合 S，它是 G 的子集且满足 $|S| \leq k$?

我们还给出了以下问题的结果，其中 d 不属于参数。

COC/vc

输入：图 G、整数 $d \geq 1$、整数 k、集合 M，满足 $G - M$ 是独立集。参数：$|M|$。问题：是否存在一个 coc - d 集合 S，它是 G 的子集且满足 $|S| \leq k$?

COC/k+vc

输入：图 G、整数 $d \geq 1$、整数 k、集合 M，使得 $G - M$ 是一个独立集

参数：$k + |M|$

问题：是否存在一个 d-coc 集合 S，使得 G 满足 $|S| \leq k$?

A-COC/k+vc

输入：图 G、整数 $d \geq 1$、整数 k、集合 M，使得 $G - M$ 是一个独立集，集合 $A \subseteq \binom{M}{2}$。参数：$k + |M|$。问题：是否存在一个大小至多为 k 的集合 $S \subseteq V(G)$，使其是 G 的 d-coc 集合，并且对于每个 $\{u, v\} \in A$，都有 $S \cap \{u, v\} \neq \emptyset$?

COC/pw-1

输入：图 G，整数 $d \geq 1$，整数 k，集合 M使得 $G - M$是毛毛虫图森林

参数：$|M|$

问题：是否存在一个 d-coc 集合 S，由 G 构成，且满足 $|S| \leq k$?

10.2 进一步的问题

为了证明定理 9.1 中的下界，我们将问题归约自 UMRSS；根据 [25, Lemma4]，UMRSS 是 W[1]- 难的。

一元多维宽松子集和（UMRSS）

输入：整数 k，向量集合 $S = \{s_1, .., s_n\}$，其中每个 $s_i \in \mathbb{N}^k$；目标向量 $t \in \mathbb{N}^k$；整数 k'。该实例中的所有整数均采用一元编码。参数：$k + k'$问题：是否存在一个集合 $S' \subseteq S$，使得 $|S'| \leq k'$ 且 $\sum_{s \in S'} s \geq t$?

为了证明定理 9.8 中的核化下界，我们将问题归约自以全集大小为参数的精确集合覆盖，记作精确集合覆盖/U。Dell 和 Marx 的研究给出了该问题的下界 [18]。

精确集合覆盖

输入：全集 $\mathcal{U}$、集合族 $\mathcal{F} \subseteq \binom{\mathcal{U}}{t}$ 对于某个整数 t，整数 k问题：是否存在集合 $\mathcal{F}' \subseteq \mathcal{F}$ 中大小为 k 的集合，使得 $\mathcal{U}$ 中的每个元素恰好属于 $\mathcal{F}'$中的一个集合?

58 通过到有界路径宽度的距离实现 COC 核化的分界线

精确集合覆盖/U输入：全集 $\mathcal{U}$、集合族 $\mathcal{F} \subseteq \binom{\mathcal{U}}{t}$ 对于某个整数 t，整数 k参数：$|\mathcal{U}|$问题：是否存在集合 $\mathcal{F}' \subseteq \mathcal{F}$ 中大小为 k 的集合，使得 $\mathcal{U}$ 中的每个元素恰好属于 $\mathcal{F}'$中的一个集合?

请注意，在我们对精确集合覆盖/U 的定义中，每个集合的大小都相同，为 t，但 t 不是常数，也不是参数的一部分。

我们没有给出以下问题的结果，但仍在此给出它们的定义，因为论文中提到了这些问题。

反馈顶点集输入：图 G，整数 k问题：是否存在一个大小至多为 k 的集合 $S \subseteq V(G)$，使得 $G - S$ 无环?

有界度删除输入：图 G，整数 k、t问题：是否存在一个大小至多为 k 的集合 $S \subseteq V(G)$，使得 $G - S$ 的最大度至多为 t?

在$\mathcal{F}$-子式删除问题中，$\mathcal{F}$ 是一个（通常为有限的）图集合。

$\mathcal{F}$-子式删除
输入：图 G，整数 k问题：是否存在一个大小至多为 k 的集合 $S \subseteq V(G)$，使得 $G - S$ 不包含任何图 $\mathcal{F}$ 作为子式?

参考文献

1 Stefan Arnborg、Andrzej Proskurowski 和 Detlef Seese。单子二阶逻辑、树自动机和禁用子式。载于 Egon Börger、Hans Kleine Büning、Michael M. Richter 和 Wolfgang Schönfeld 编，计算机科学逻辑，第*4*届研讨会，*CSL '90*，海德堡，德国，*1990*年*10*月*1*日至*5*日，论文集，计算机科学讲义第533卷，第1–16页。施普林格，1990年。网址：https://doi.org/10.1007/3-540-54487-9_49，doi:10.1007/3-540-54487-9_49。

2 Samuel Baguley、Tobias Friedrich、Aneta Neumann、Frank Neumann、Marcus Pappik 和 Ziena Zeif。用于 w-分隔集问题的固定参数多目标进化算法。*Algorithmica*，87(4)：537–571，2025年。doi:10.1007/s00453-024-01290-9。

3 Sriram Bhyravarapu、Satyabrata Jana、Saket Saurabh 和 Roohani Sharma。差异决定度：分量阶连通性的结构核化。载于 Neeldhara Misra 和 Magnus Wahlström（编），第*18*届参数化与精确计算国际研讨会，*IPEC 2023*，*2023*年*9*月*6*日至*8*日，阿姆斯特丹，荷兰，LIPIcs第*285*卷，第5:1–5:14页。Schloss Dagstuhl - Leibniz-Zentrum für Informatik，2023年 doi:10.4230/LIPIcs.IPEC.2023.5。

4 Rodica Boliac、Kathie Cameron 和 Vadim V. Lozin。关于计算二分图的解离数和诱导匹配数。*Ars Comb.*，72，2004年。

5 Marin Bougeret、Eric Brandwein 和 Ignasi Sau。结构参数化下命中小图问题的核化二分性。载于 Meena Mahajan、Florin Manea、Annabelle McIver 和 Kim Thang Nguyen（编），第*43*届计算机科学理论方面国际研讨会，*STACS 2026*，法国格勒诺布尔，*2026*年*3*月*9*日至*13*日，第364卷

LIPIcs，第17:1–17:19页。Schloss Dagstuhl - Leibniz-Zentrum für Informatik，2026年。网址：https://doi.org/10.4230/LIPIcs.STACS.2026.17, doi:10.4230/LIPICS.STACS.2026.17。6 Marin Bougeret、Bart M. P. Jansen 和 Ignasi Sau。桥深度刻画了哪些顶点覆盖的子式封闭结构参数化具有多项式核。SIAM离散数学杂志，36(4):2737–2773，2022年。 doi:10.1137/21m1400766。7 Marin Bougeret、Bart M. P. Jansen 和 Ignasi Sau。结构参数化下击中子图问题的核化二分性。载于 Karl Bringmann、Martin Grohe、Gabriele Puppis 和 Ola Svensson（编），第51届自动机、语言与编程国际研讨会，ICALP 2024，2024年7月8日至12日，爱沙尼亚塔林，LIPIcs第297卷，第33:1–33:20页。Schloss Dagstuhl - Leibniz-Zentrum für Informatik，2024年。 doi:10.4230/LIPIcs.ICALP.2024.33。8 Marin Bougeret 和 Ignasi Sau。树深度调节集对在稀疏图之外获得多项式核有多大帮助？Algorithmica，81(10):4043–4068，2019年。 doi:10.1007/s00453-018-0468-8。9 Jonathan F. Buss 和 Judy Goldsmith。P中的非确定性。SIAM J. Comput.，22(3):560–572，1993年。 doi:10.1137/0222038。10 Katrin Casel、Tobias Friedrich、Davis Issac、Aikaterini Niklanovits 和 Ziena Zeif。连通性约束的平衡冠分解。载于 Petra Mutzel、Rasmus Pagh 和 Grzegorz Herman（编），第29届欧洲算法年度研讨会，ESA 2021，2021年9月6日至8日，葡萄牙里斯本（线上会议），LIPIcs第204卷，第26:1–26:15页。Schloss Dagstuhl - Leibniz-Zentrum für Informatik，2021年。 doi:10.4230/LIPIcs.ESA.2021.26。11 Katrin Casel、Tobias Friedrich、Aikaterini Niklanovits、Kirill Simonov 和 Ziena Zeif。组合冠结构以处理脆弱性度量。载于 Édouard Bonnet 和 Pawel Rzazewski（编），第19届参数化与精确计算国际研讨会，IPEC 2024，2024年9月4日至6日，伦敦大学皇家霍洛威学院，英国埃格姆，LIPIcs第321卷，第1:1–1:15页。Schloss Dagstuhl - Leibniz-Zentrum für Informatik，2024年。doi:10.4230/LIPIcs.IPEC.2024.1。12 Jianer Chen、Iyad A. Kanj 和 Weijia Jia。顶点覆盖：进一步的观察与改进。算法杂志，41(2):280–301，2001年。 doi:10.1006/jagm.2001.1186。13 Benny Chor、Mike Fellows 和 David W. Juedes。线性时间内的线性核，或者如何在 $o(n^2)$ 步内节省 k 种颜色。载于 Juraj Hromkovic、Manfred Nagl 和 Bernhard Westfechtel（编），《计算机科学中的图论概念》，第30届国际研讨会，WG 2004，德国巴特洪内夫，2004年6月21日至23日，修订论文，计算机科学讲义第3353卷，第257–269页。施普林格，2004年。网址： https://doi.org/10.1007/978-3-540-30559-0_22, doi:10.1007/978-3-540-30559-0_22。14 Ernest J. Cockayne、Stephen T. Hedetniemi 和 Peter J. Slater。超图中的匹配与横截集、支配以及树中的独立性。组合理论杂志B辑，26(1):78–80，1979年。doi:10.1016/0095-8956(79)90044-3。15 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。16 Marek Cygan、Daniel Lokshtanov、Marcin Pilipczuk、Michal Pilipczuk 和 Saket Saurabh。论宽度损失的困难性。理论计算机系统，54(1):73–82，2014年。 doi:10.1007/s00224-013-9480-1。17 David Dekker 和 Bart M. P. Jansen。通过到森林的消除距离对反馈顶点集进行核化。离散应用数学，346:192–214，2024年。网址： https://doi.org/10.1016/j.dam.2023.12.016, doi:10.1016/J.DAM.2023.12.016。18 Holger Dell 和 Dániel Marx。装箱问题的核化。载于 Yuval Rabani（编），第23届ACM-SIAM离散算法年度研讨会论文集，SODA 2012，日本京都，2012年1月17日至19日，第68–81页。SIAM，2012年。 doi:10.1137/1.9781611973099.6。

60 到有界路径宽度的距离下 COC 核化的分界线

19 Huib Donkers 和 Bart M. P. Jansen。结构参数化下 f-小图无关删除的图灵核化二分。《计算机与系统科学杂志》，119:164–182，2021。 doi:10.1016/j.jcss.2021.02.005。20 Pål Grønås Drange、Markus S. Dregi 和 Pim van 't Hof。顶点完整度与分量阶连通性的计算复杂性。《Algorithmica》，76(4):1181–1202，2016。doi:10.1007/s00453-016-0127-x。21 Michael R. Fellows。膨胀、双赢策略与冠规则：FPT 的一些新方向。载于 Hans L. Bodlaender 编，《计算机科学中的图论概念》，第29届国际研讨会 WG 2003，荷兰埃尔斯佩特，2003年6月19日至21日，修订论文，第2880卷《计算机科学讲义》，第1–12页。施普林格，2003。网址： https://doi.org/10.1007/978-3-540-39890-5_1， doi:10.1007/978-3-540-39890-5_1。22 Fedor V. Fomin、Daniel Lokshtanov、Neeldhara Misra 和 Saket Saurabh。平面 f-删除：近似、核化和最优固定参数算法。载于《第53届 IEEE 计算机科学基础年会》，FOCS 2012，美国新泽西州新不伦瑞克，2012年10月20日至23日，第470–479页。IEEE计算机学会，2012。 doi:10.1109/FOCS.2012.62。23 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。 doi:10.1017/9781107415157。24 Fedor V. Fomin 和 Torstein J. F. Strømme。重新审视顶点覆盖的结构参数化。载于 Pinar Heggernes 编，《计算机科学中的图论概念——第42届国际研讨会》，WG 2016，土耳其伊斯坦布尔，2016年6月22日至24日，修订精选论文，第9941卷《计算机科学讲义》，第171–182页，2016。网址：https://doi.org/10.1007/978-3-662-53536-3_15， doi:10.1007/978-3-662-53536-3_15。25 Robert Ganian、Fabian Klute 和 Sebastian Ordyniak。关于有界度顶点删除问题的结构参数化。《Algorithmica》，83(1):297–336，2021年1月。doi:10.1007/s00453-020-00758-8。26 M. R. Garey 和 David S. Johnson。直角斯坦纳树问题是 NP 完全的。《SIAM应用数学杂志》，32:826–834，1977。27 Archontia C. Giannopoulou、Bart M. P. Jansen、Daniel Lokshtanov 和 Saket Saurabh。命中禁用小图的一致核化复杂度。《ACM算法汇刊》，13(3):35:1–35:35，2017。 doi:10.1145/3029051。28 Gracinda M. S. Gomes 和 John M. Howie。关于某些保序变换半群的秩。《半群论坛》，45(1):272–282，1992。 doi:10.1007/BF03025769。29 Jakob Greilhuber 和 Roohani Sharma。以到细分梳状图的距离为参数的分量阶连通性不存在多项式核。载于 Édouard Bonnet 和 Pawel Rzazewski 编，《第19届参数化与精确计算国际研讨会》，IPEC 2024，2024年9月4日至6日，伦敦大学皇家霍洛威学院，英国埃格姆，第321卷 LIPIcs，第21:1–21:17页。Schloss Dagstuhl - Leibniz-Zentrum für Informatik，2024。 doi:10.4230/LIPIcs.IPEC.2024.21。30 Daniel Gross、Monika Heinig、Lakshmi Iswara、L William Kazmierczak、Kristi Luttrell、John T Saccoman 和 Charles Suffel。图论网络分量顺序连通性模型综述。《WSEAS数学汇刊》，12(9):895–910，2013。31 Anupam Gupta、Euiwoong Lee、Jason Li、Pasin Manurangsi 和 Michal Wlodarczyk。通过分离子集损失树宽。载于 Timothy M. Chan 编，《第三十届 ACM-SIAM 离散算法年会论文集》，SODA 2019，美国加利福尼亚州圣迭戈，2019年1月6日至9日，第1731–1749页。SIAM，2019。 doi:10.1137/1.9781611975482.104。32 Peter M. Higgins。保序映射半群的组合结果。《剑桥哲学学会数学会报》，113(2):281–296，1993。doi:10.1017/S0305004100075964。33 Peter M. Higgins。保序映射半群中的幂等深度。《爱丁堡皇家学会会报：A辑数学》，124(5):1045–1058，1994。 doi:10.1017/S0308210500022502。

34 Eva-Maria C. Hols 和 Stefan Kratsch。用于顶点覆盖核化的更小参数。载于 Daniel Lokshtanov 和 Naomi Nishimura 编，《第12届参数化与精确计算国际研讨会》，IPEC 2017，奥地利维也纳，2017年9月6日至8日，第89卷 LIPIcs，第20:1–20:12页。Schloss Dagstuhl - Leibniz-Zentrum für Informatik，2017。 doi:10.4230/LIPIcs.IPEC.2017.20。35 Eva-Maria C. Hols、Stefan Kratsch 和 Astrid Pieterse。顶点覆盖的消除距离、阻塞集和核。SIAM离散数学杂志，36(3):1955–1990，2022。 doi:10.1137/20m1335285。36 J. M. Howie。某些变换半群中的幂等元乘积。爱丁堡数学学会会刊，17(3):223–236，1971。 doi:10.1017/S0013091500026936。37 J. M. Howie 和 B. M. Schein。保序幂等变换的乘积。伦敦数学学会杂志，s2-7(2):357–366，1973。 doi:10.1112/jlms/s2-7.2.357。38 A. T. Imam、S. Ibrahim、G. U. Garba、L. Usman 和 A. Idris。全保序变换有限半群中的拟幂等元。《代数与离散数学》，35(1):62–72，2023。 doi:10.12958/adm1846。39 Bart M. P. Jansen 和 Hans L. Bodlaender。重新审视顶点覆盖核化——精细参数的上界和下界。《理论计算机系统》，53(2):263–299，2013。doi:10.1007/s00224-012-9393-4。40 Bart M. P. Jansen 和 Astrid Pieterse。结构参数化下命中禁用小图的多项式核。《理论计算机科学》，841:124–166，2020。 doi:10.1016/j.tcs.2020.07.009。41 Richard M. Karp。组合问题之间的可约性。载于 Raymond E. Miller 和 James W. Thatcher 编，《计算机计算复杂性研讨会论文集》，于1972年3月20日至22日在约克敦高地的 IBM Thomas J. Watson研究中心举行，《IBM研究研讨会系列》，第85–103页。Plenum出版社，纽约，1972。网址： https://doi.org/10.1007/978-1-4684-2001-2_9， doi:10.1007/978-1-4684-2001-2_9。42 Lawrence William Kazmierczak。连通性与分量阶连通性之间的关系。博士论文，史蒂文斯理工学院，2003。网址： https://dl.acm.org/doi/10.5555/959796。43 Samir Khuller。算法专栏：顶点覆盖问题。SIGACT News，33(2):31–33，2002。 doi:10.1145/564585.564598。44 Ganna Kudryavtseva 和 Volodymyr Mazorchuk。范畴部分化与逆辫-置换幺半群。《代数与计算国际杂志》，18(6):989–1017，2008。 doi:10.1142/S0218196708004731。45 Mithilesh Kumar 和 Daniel Lokshtanov。用于 l-分量阶连通性的 2lk 核。载于 Jiong Guo 和 Danny Hermelin 编，《第11届参数化与精确计算国际研讨会》，IPEC 2016，丹麦奥胡斯，2016年8月24日至26日，第63卷 LIPIcs，第20:1–20:14页。Schloss Dagstuhl - Leibniz-Zentrum für Informatik，2016。 doi:10.4230/LIPIcs.IPEC.2016.20。46 Michael Lampis。顶点覆盖的阶为 2 k-c log k 的核。信息处理快报，111(23-24):1089–1091，2011。网址： https://doi.org/10.1016/j.ipl.2011.09.003， doi:10.1016/J.IPL.2011.09.003。47 John M. Lewis 和 Mihalis Yannakakis。遗传性质的节点删除问题是 NP 完全的。《计算机与系统科学杂志》，20(2):219–230，1980。 doi:10.1016/0022-0000(80)90060-4。48 Diptapriyo Majumdar、Venkatesh Raman 和 Saket Saurabh。以小度数调制集为参数的顶点覆盖多项式核。《理论计算机系统》，62(8):1910–1951，2018。 doi:10.1007/s00224-018-9858-1。49 Arezou Soleimanfallah 和 Anders Yeo。顶点覆盖的阶为 2k-c 的核。《离散数学》，311(10-11):892–895，2011。 doi:10.1016/j.disc.2011.02.014。

62 通过到有界路径宽度的距离实现 COC 核化的分界线

50 屠建华和杨凤梅。三次图中的顶点覆盖 P_3 问题。 信息处理快报，113(13):481–485，2013年。 doi:10.1016/j.ipl.2013.04.002。**51** 将图分割为有界大小连通分量的线性核。《计算机与系统科学杂志》，88:260–270，2017年。 doi:10.1016/j.jcss.2017.04.004。