通过选择性探索 Pachner 图寻找大型反例

Benjamin A. Burton Alexander He 昆士兰大学 昆士兰大学 bab@maths.uq.edu.au a.he@uqconnect.edu.au

2024年3月8日

arXiv:2303.06321v2 [math.GT] 7 Mar 2024

摘要

在三维流形拓扑研究中，我们常常依赖三角剖分普查来指引直觉。然而，如果最小的反例过大，无法收录于普查中，这就可能使我们对猜想产生不恰当的信心。由于三角剖分的数量会随着规模呈超指数增长，因此无法将普查扩展到相对较小的三角剖分之外——目前的普查仅涵盖至 10 个四面体。在本文中，我们展示了如何利用启发式方法对三角剖分进行选择性（而非穷举式）枚举，从而切实可行地搜索大型且难以发现的反例。我们运用这一思路，找到了三个猜想的反例。这些猜想针对某些三维流形提出：单顶点三角剖分是否总有一条“独特”的边，能够帮助我们识别该三维流形。

关键词 计算拓扑、三维流形、三角剖分、反例、启发式方法、实现、帕赫纳变换、双星翻转

致谢 我们感谢审稿人提出的宝贵意见。第二作者获得了澳大利亚政府研究培训计划奖学金的资助。

1 引言

计算几何和拓扑领域中，不乏一些猜想在小规模情形下成立，但由于存在相对较大的反例，最终被证明并非普遍成立。最著名的例子或许是*Hirsch*猜想，该猜想认为，在一个d维、具有 n 个面的多胞体中，任意两个顶点之间都存在一条至多包含 $n-d$条边的路径。已知Hirsch猜想在维数较小的情况下成立（具体而言，当$d \leqslant 3$ [26]时），在面数较少的情况下也成立（具体而言，当 $n \leqslant d+6$ [2]时）。然而，这些小规模情形并不能反映一般情况：Santos证明存在一个具有86个面的43维反例 [38]。

以计算三维流形拓扑中的一个例子为例，考虑塞弗特纤维空间（关于这些三维流形的定义，参见第 2.6 节）。可以用不超过 7 个四面体进行三角剖分的塞弗特纤维空间共有 302 个；对于其中每一个，至少存在一种**最小三角剖分**（即四面体数量尽可能少的三角剖分）采用标准的棱柱与分层构造。这种规律并非总是成立：有一个塞弗特纤维空间，其唯一的（含 8 个四面体的）最小三角剖分由同构签名 iLLLPQcbcgffghhhtsmhgosof 给出（软件包 Regina [9] 可将此字符串还原为三角剖分）；该三角剖分采用的则是一种构件——称为**砖块** B_5——这种构件最早见于 Martelli和Petronio 对四面体数不超过 9 的最小三角剖分所作的普查 [29].[1]

本文的主要目标是在与刚才提到的情形类似的背景下，介绍一种寻找大型反例的技术。不同之处在于，我们不再只研究最小

[1]尽管第一作者 [4] 和 Matveev [31] 后来扩展了普查，但至今尚未发现其他类似 B_5 的装置。

三角剖分的性质，而是考虑任意单顶点三角剖分。这样做的实际动机在于，单顶点三角剖分很容易获得，而通常很难确定某个特定的三角剖分是否为最小三角剖分。

在我们的设定中，最显而易见的反例来源，是对四面体数量不超过某个给定值的所有可能三角剖分进行普查。然而，这只能涵盖小反例，因为随着四面体数量增加，三角剖分的数量会以超指数级增长；截至目前，所有可能的闭三维流形三角剖分普查仅涵盖至 10 个四面体，而这已经包含超过 20 亿个三角剖分，数据量超过 63 GB [4]。

我们如何找到大到无法列入普查的反例？我们展示了一种选择性地（而非穷尽地）枚举三角剖分的方法，使我们得以找到三个相关猜想的大型反例。这三个猜想由 Saul Schleimer 在 2022 年达格斯图尔研讨会低维拓扑空间中的计算与重构上提出。

1.1 这些猜想

上文提到的三个猜想都涉及单顶点三角剖分中边的性质。由于我们的三角剖分只有一个顶点，每条边都实现了一条嵌入闭曲线，因此我们可以考察这样的边是否以某种有趣或有用的方式嵌入。

例如，回想一下，**透镜空间**是由两个实心环面粘合而成的三维流形。识别透镜空间的一种方法是寻找一条**核心曲线**：一条嵌入闭曲线 γ，使得去掉 γ 的开正则邻域后得到的补空间是一个实心环面。我们可以问，透镜空间的单顶点三角剖分是否总包含一条**核心边**（即实现核心曲线的一条边）：

猜想 1.每个透镜空间的单顶点三角剖分都含有一条核心边。

由于实心环面识别在实践中可以高效完成 [13, 6], ，证明猜想 1 就能提供一种新的、相对高效的透镜空间识别方法。透镜空间识别还可用于判定一个三维流形是否为**椭圆的**（即是否具有球面几何）。事实上，Lackenby 和 Schleimer [27] 最近证明，识别椭圆流形属于 NP；他们的证明依赖于以下结果，该结果可视为猜想 1 的一种变体：

定理 (Lackenby 和 Schleimer [27, 定理 9。4])。 设 $\mathcal{M}$ 是一个既非 $\mathbb{R}P^3$ 也非棱柱流形[2], 的透镜空间，并设 $\mathcal{T}$ 是 $\mathcal{M}$的任意三角剖分。那么， $\mathcal{T}$ 的 86th 迭代重心细分包含一列构成核心曲线的边。

另外两个猜想的动机类似：如果某类三维流形的每个单顶点三角剖分都包含一条具有显著性质的边，那么我们就可以利用这样的边来识别该三维流形。

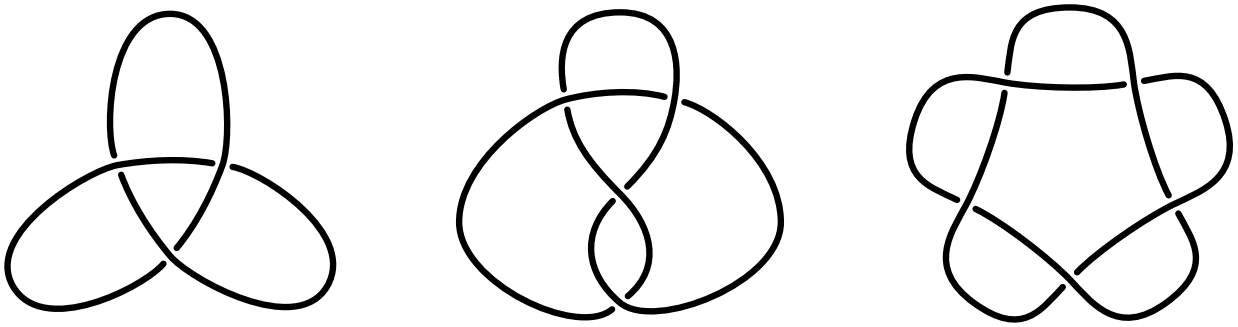
对于第二个猜想，考虑三维球面中的一个纽结 K。 K 的**隧道数** 是指需要添加到 K 上的弧的最小数量，使其补空间成为一个把手体；根据构造，这就是一个纽结不变量。图1所示的三个纽结都是隧道数为一的纽结实例（尽管乍看之下可能并不明显）。

设 $\mathcal{T}$ 是 K 补空间的单顶点理想三角剖分，因此 $\mathcal{T}$ 中的每条边 e 都构成一条弧 α，其端点位于 K 上（且除此之外不与 K 相交）。如果 $K \cup \alpha$ 的补空间是一个亏格2把手体，我们就称 e 为**隧道边**；假设 K 不是平凡纽结，存在隧道边意味着 K 的隧道数为 1。

猜想 2. 设 K 是一个隧道数为 *1* 的纽结。那么， K 的每个单顶点理想三角剖分都含有一条隧道边。

对于第三个猜想，我们考虑**塞弗特纤维空间**。这类三维流形以一种特定方式由圆纤维化（完整定义见第 2.6 节）；我们将每条圆纤维称为**塞弗特纤维**，并将同伦于塞弗特纤维的边称为**塞弗特纤维边**。塞弗特纤维空间之所以重要，是因为它们在闭可定向不可约三维流形的典范环面分解中发挥着核心作用：

[2]Lackenby 和 Schleimer 使用的棱柱流形定义与我们在第 2.6 节给出的定义略有不同。



(a) 三叶结。　　(b) 八字结。　　(c) (5，2) 环面结。

图1：三个隧道数为 1 的纽结。

• JSJ分解（由 Jaco 和 Shalen [22, 24], 以及 Johannson [25]提出）将这类三维流形切分为若干部分，每一部分要么是非环面的，要么是Seifert纤维化的。
• Thurston 的几何化猜想（由 Perel- man [32, 34, 33] 著名地证明）所给出的几何分解，将这类三维流形切分为双曲部分或图流形；图流形正是那些可以沿环面进一步分解为塞弗特纤维空间的三维流形。

小塞弗特纤维空间中不含任何嵌入的双侧不可压缩曲面，这使得它们相对难以处理。

猜想 3. 小塞弗特纤维空间的每个单顶点三角剖分都含有一条塞弗特纤维边。

1.2 使用定向搜索寻找反例

如前所述，我们已经找到了第 1.1 节中列出的三个猜想的反例。本文余下部分将主要聚焦于我们用于寻找这些反例的方法。关键创新是一种启发式方法，用于衡量一个三角剖分距离不含核心边（或者更一般地说，不含满足某种可计算拓扑性质的边）有多远。

要理解为何需要这种启发式方法，不妨考虑三维球面——最简单的透镜空间；对于三维球面，核心边等价于**平凡纽结**的边。在包含至多 10 个四面体的所有三角剖分普查中，有 422533 279 个三维球面的单顶点三角剖分。我们对它们全部进行了测试，发现每个三角剖分至少含有一条核心边。这项工作在 12 个线程上并行运行，墙钟时间略多于 22 小时，但这不包括以下所需的时间：

(1) 首先生成普查；以及 (2) 找出该普查中的所有三维球面（这项工作此前已经完成 [4, 5]）。

（我们还进行了更为穷举的测试——在 12 个并行线程上耗时约 53 个半小时的墙钟时间——结果表明，所有最多含 10 个四面体的单顶点 3-球面实际上都至少有两条 核心边。）

主要结论是，这种穷举搜索既耗费巨大，又未能奏效。相比之下，通过使用我们的启发式方法，有针对性地枚举三角剖分，我们仅用几分钟 墙钟时间便找到了所有反例，而且完全不必预处理大量数据。值得注意的是，我们的所有反例都至少含有 11 个四面体，超出了普查的范围。详细的计算结果请见第5节；在第5节末尾，我们还将展示如何把每个反例转化为一个无限族。

我们将在第3节和第4节讨论帮助我们找到这些例子的关键工具和算法。具体来说，第 4 节介绍我们的启发式方法和定向搜索算法。第3节讨论了若干辅助算法，这些算法本身也可能具有独立的研究价值；特别是，我们在第3.1节提出（并实现）了一种改进的把手体识别算法。

最后，我们将在第 6 节提出一些尚未解答的问题。特别是在第 6.2 节，我们会提及一些其他潜在应用，即利用启发式方法有选择地枚举大型三角剖分。

2 预备知识

本文中，我们称紧 3-流形为**闭**流形，当且仅当其边界为空；若其边界非空，则称为**有界**流形。

2.1 三角剖分

一个（**广义**）三**角剖分** $\mathcal{T}$ 是由有限个四面体组成的集合，其中三角形面可以成对地进行仿射粘合；每个此类面的等价类都是 $\mathcal{T}$ 的一个**面**。 $\mathcal{T}$ 的**大小**记为 $|\mathcal{T}|$，即构成 $\mathcal{T}$ 的四面体数量。 $\mathcal{T}$ 的**边界**由所有未与其他面粘合的面组成。我们允许同一个四面体的面彼此粘合，这意味着我们的三角剖分不必是单纯复形。

由于面的粘合，四面体的许多边和顶点也可能彼此粘合。由此得到的边的等价类称为三角剖分的**边**；同样，顶点的等价类称为三角剖分的**顶点**。三角剖分中边 e 的**度**是与 e 相接的四面体数量，按重数计；这等价于粘合形成 e 的四面体边的数量。三角剖分 $\mathcal{T}$ 的**1-骨架**是仅由 $\mathcal{T}$ 的顶点和边构成的子复形。

如果一个点位于三角剖分的边界上，我们就称它为**边界**点；否则，称它为**内部**点。如果一条边或一个面完全 由边界点构成，我们就称它为**边界**边或边界面；否则，称它为**内部**边或内部面。

广义三角剖分的一个显著特点是，可以得到**单顶点三角剖分**：即所有四面体顶点都被识别为同一个顶点的三角剖分。事实上，每个边界分支至多有一个的紧 3-流形都允许进行单顶点三角剖分；一种证明方法是使用 Weeks [16, 42]首次引入的双四面体“带隧道的三角枕”。

作为一个虽小但颇具说明性的例子，我们仅用一个四面体 Δ 构造实心环面 $\mathcal{T}$ 的单顶点三角剖分。按照图2所示，将 Δ 的顶点标记为 0、1、2 和 3。粘合前面的两个面，使左侧面的顶点 0、1 和 2 分别与右侧面的顶点 1、3 和 0 重合。尽管 $\mathcal{T}$ 不是单纯复形，但按照我们刚才定义的意义，它完全可以作为一个合理的三角剖分。注意， Δ 的四个顶点都被识别为 $\mathcal{T}$ 中的一个边界顶点，而 $\mathcal{T}$ 有三条边（它们全都是边界边），这些边通过识别 Δ 中的以下边得到：

- 边 20、01 和 13 共同构成一条边，如图2中的黑色箭头所示；
- 边 12 和 30 共同构成一条边，如图2中的白色箭头所示；以及
- 边 23 构成最后一条边。

要看出 $\mathcal{T}$ 确实对实心环面进行了三角剖分，一种方法是注意到 $\mathcal{T}$ 的内部面¨构成了一个莫比乌斯带；于是我们可以将整个三角剖分看作这个莫比乌斯带的“加厚”，而后者正是一个实心环面。

一般而言，对于一个三角剖分 $\mathcal{T}$ 要表示一个三维流形，其中的每个点 p $\mathcal{T}$ 都必须是**非奇异的**，也就是说，它具有一个小邻域，该邻域的边界要么是一个圆盘（如果 p 是边界），要么是一个 2-球面（如果 p 是内部）。对于顶点和边的中点而言，这一条件可能不成立。在本文中，我们坚持要求任何一条边都不会与自身反向识别；这确保了边的中点始终是非奇异的。

如果 $\mathcal{T}$ 一个三**维流形**三**角剖分**的所有顶点都不是奇异顶点，我们就称其为三维流形三角剖分，因为在这种情况下， $\mathcal{T}$ 的底层拓扑空间确实是一个三维流形（可能带有边界）。如果底层三维流形 $\mathcal{M}$ 是闭的，我们就称三维流形三角剖分 $\mathcal{T}$ 为**闭**的；如果 $\mathcal{M}$ 有界，我们就称 $\mathcal{T}$ 其为**有界**的。

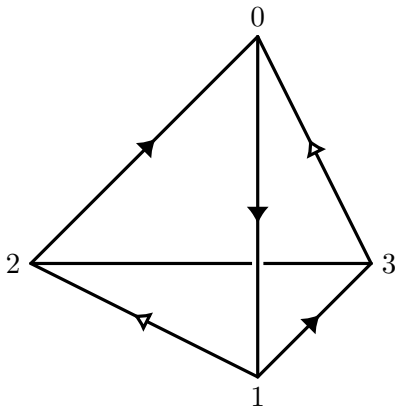


图2：将一个四面体的两个面以“扭转”的方式粘合，可得到实心环面的单顶点三角剖分。

除了三维流形三角剖分外，我们也研究含有**理想**顶点的三角剖分，这意味着该顶点的一个小邻域由一个不同于2-球面的闭曲面围成。在本文中，如果一个三角剖分的**所有**顶点都是理想顶点，我们就称它为**理想**三角剖分。理想三角剖分通常能有效地表示通过截去理想顶点而得到的有界三维流形；这一思想源自 Thurston 对八字结补空间所作的双四面体理想三角剖分 [41, 例1.48]。

为了具体编码一个三角剖分，我们通常为每个四面体指定一个标签，并为其四个顶点排序。若两个三角剖分只在四面体重新标记和／或四面体顶点重新排序后才有所不同，则称它们（**组合意义上**）**同构**。我们可以用一种称为**同构签名**的短字符串来唯一标识任意一个三角剖分同构类，而且这种字符串可以高效计算。同构签名有许多种定义方式；我们采用 [5], 中介绍的定义方式，该方式已在 Regina [6, 9]中实现。

2.2 2-3 变换和3-2 变换

对于任何具有至少两个四面体的单顶点三角剖分 $\mathcal{T}$，我们都可以通过对一个与两个不同四面体相交的（三角形）面执行**2-3 变换**，得到同一三维流形的新单顶点三角剖分；此变换会将这两个四面体替换为三个围绕新边 e连接的四面体，如图3所示。我们称其逆变换为对边 e执行**3-2 变换**。注意，当且仅当一条边是实际与三个不同四面体相交的内部三度边时，才能对其执行3-2 变换。

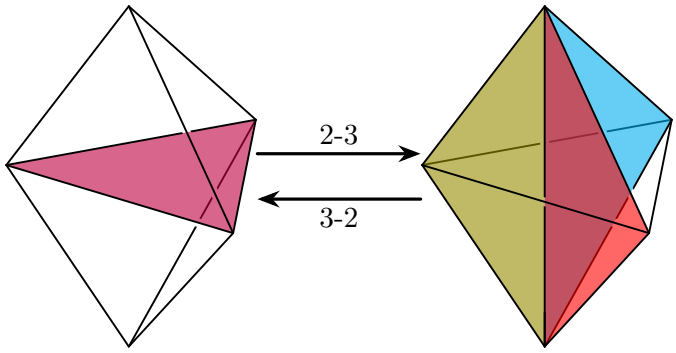


图3：2-3 和 3-2 变换。

有鉴于此，考虑一个闭 3-流形 $\mathcal{M}$。我们可以将 $\mathcal{M}$ 的至少含有两个四面体的单顶点三角剖分视为无限图的节点；当且仅当相应的三角剖分之间可通过 2-3 变换相互转换时，两个节点之间就以无向弧相连。这个图称为 $\mathcal{M}$的**Pachner 图** 。Matveev [30]和皮耶加利尼 [35]各自独立地证明了以下

结果（尽管他们并未用 Pachner 图的术语表述这一结果）：

T定理 4。对于每个闭 *3*-流形 $\mathcal{M}$，$\mathcal{M}$ 的 *Pachner* 图都是连通的。

单顶点理想三角剖分也有类似的结果，这是 Amendola 定理 [1] 的一个特例。Amendola 定理的另一种证明也见于 [37]。

由于我们尤其关注单顶点三角剖分的边，因此有必要强调 2-3 和 3-2 变换会如何影响三角剖分的 1- 骨架。2-3 变换会产生一条新边，但除此之外不会改变 1- 骨架。对边 e 执行 3-2 变换会移除这条边，但除此之外不会改变 1-骨架。

2.3 嵌入曲面

通过考察三维流形所包含的嵌入曲面，我们往往能对其有深入了解。这里我们回顾描述此类曲面的一些标准术语。

在本节中，设 S 表示嵌入在紧致三维流形 $\mathcal{M}$ （可能带有边界）中的一个曲面。若 S 在 $\mathcal{M}$ 中是**适当**嵌入的，则称 $S \cap \partial\mathcal{M} = \partial S$。

如果 S 是一个2-球面，那么如果 S 在 $\mathcal{M}$中围成一个3维球，我们称 S 是**非本质的**；否则称其为**本质**。如果三维流形 $\mathcal{M}$ 包含一个本质2-球面，则称其**可约**；否则称其**不可约**。

嵌入在 $\mathcal{M}$ 中的圆盘 D 是 S 的**压缩圆盘**，如果 $D \cap S = \partial D$。如果曲线 ∂D 围成另一个完全位于 S内的圆盘，则称这样的压缩圆盘 D 是**非本质的**；否则， D 是**本质**的。图4展示了一个非本质压缩圆盘。如果曲面 S 有本质压缩圆盘，则称其**可压缩**；否则称其**不可压缩**。在曲面 S 恰为 $\partial\mathcal{M}$的特殊情况下，请注意， S 的压缩圆盘与 $\mathcal{M}$中的适当嵌入圆盘相同；因此，每当我们处理一个恰当地嵌入的圆盘 D时，也常称 D 为压缩圆盘（无需指定曲面 $\partial\mathcal{M}$）。如果 $\partial\mathcal{M}$ 可压缩，那么我们称 $\mathcal{M}$ 为 ∂-**可约**；否则，称 $\mathcal{M}$ 为 ∂-**不可约**。

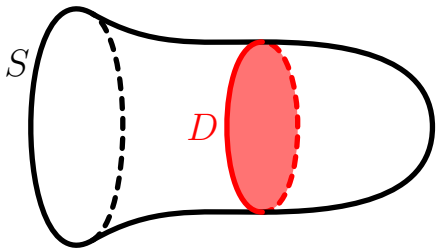


图4：曲面 S的一个非本质压缩圆盘 D 。

如果 $\mathcal{M}$ 包含一个双侧不可压缩曲面，且该曲面既不是 S^2（2-球面），也不是 $\mathbb{R}P^2$ （实射影平面），我们就称 $\mathcal{M}$ 为**足够大**；否则，称 $\mathcal{M}$ 为**小**。如果 $\mathcal{M}$ 不可约、 ∂-不可约且足够大，我们就称 $\mathcal{M}$ 为**Haken** ；否则，称 $\mathcal{M}$ 为**非 Haken**。“小”和“非 Haken”有时会互换使用，因为人们通常只研究不可约且 ∂-不可约的三维流形。

在本段中，假设 S 的边界非空且恰当地嵌入。嵌入在 $\mathcal{M}$ 中的圆盘D 是 S的一个 ∂-**压缩圆盘**，如果存在两条弧 α和 β，使得 $\alpha = D \cap S$、$\beta = D \cap \partial\mathcal{M}$、 $\alpha \cup \beta = \partial D$和 $\alpha \cap \beta = \partial\alpha = \partial\beta$。如果存在 ∂S中的另一条弧 γ，使得 $\alpha \cap \gamma = \partial\alpha = \partial\gamma$，且曲线 $\alpha \cup \gamma$围成另一个完全位于 S内部的圆盘，那么这样的 ∂-压缩圆盘 D 是**非本质的**；否则， D 是**本质的**。图5展示了一个非本质的 ∂-压缩圆盘。如果曲面 S 存在一个本质 ∂-压缩圆盘，则称其为 ∂-**可压缩**；否则，称其为 ∂-**不可压缩**。

最后，假设 S 恰当地嵌入。若存在一种固定 ∂S 的同伦，将 S 映射到 $\partial\mathcal{M}$的一个子曲面上，则称 S 为**边界平行**。当 S 既不是2-球面也不是圆盘时，如果 S 是不可压缩的、 ∂-不可压缩的且不与边界平行，则称其为**本质**。

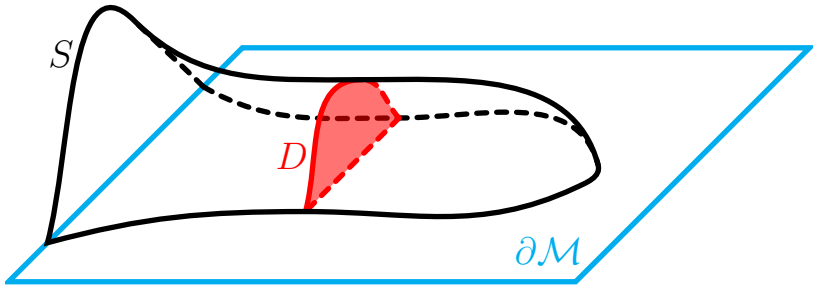


图5：曲面 S的一个非本质 ∂压缩圆盘 D 。

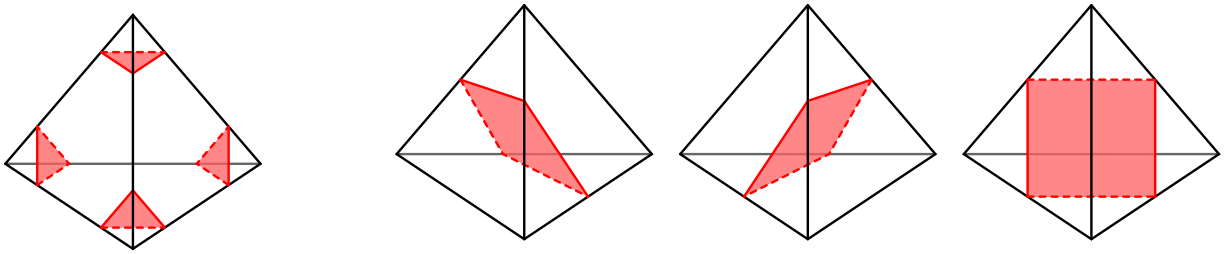
2.4 正规曲面

正如我们将在第3.1节和第3.2节中讨论的，我们的工作依赖于正规曲面理论中的若干结果。在这里，我们仅概述本研究所需的相关思想；更全面的综述见 [18] 和 [21] 。

三维流形三角剖分 $\mathcal{T}$中的（可能不连通的）恰当嵌入曲面 S，若满足以下条件，则称为**正规曲面**：

- S 与 $\mathcal{T}$的每个单形（即顶点、边、三角形或四面体）横截相交；并且
- S 与 $\mathcal{T}$ 的每个四面体 Δ 相交所得部分，是有限个（也可能为空的）圆盘的集合——称为**基本圆盘**[3]——其中每个圆盘都是曲边三角形或四边形，其顶点位于 Δ的不同边上。

在**正规同痕**意义下（即保持 $\mathcal{T}$的每个单形不变的环境同痕），每个基本圆盘都属于图6所示的七种类型之一。



(a) 四种三角形类型。　　(b) 三种四边形类型。

图 6：四面体中的七种基本圆盘类型。

最简单的正规曲面示例是**顶点链接曲面**，它完全由三角形组成。这类曲面总是存在，因此找到它们不会为我们了解底层三维流形带来任何新信息。正因如此，只有包含至少一个四边形片的正规曲面才被视为**非平凡**。

如果 $\mathcal{T}$是一个 n四面体三角剖分，那么我们可以将 $\mathcal{T}$中的任意正规曲面表示为 $\mathbb{R}^{7n}$ 中的**正规坐标**向量，该向量统计每个四面体中每种基本圆盘的数量。反过来，事实证明， $\mathbb{R}^{7n}$ 中对应于 $\mathcal{T}$中正规曲面的向量空间，可以通过一组组合约束（即**四边形约束**）以及一组齐次线性不等式来描述。根据齐次性，这些线性不等式的每个解都是凸多胞体 $\mathcal{P}(\mathcal{T}) \subset \mathbb{R}^{7n}$中某个向量的标量倍数。

给定两个正规曲面 S 和 S'，我们可以将对应的两个正规坐标向量相加。只要仍满足四边形约束，就会得到一个新的正规曲面，称为 S 与 S' 的 **（哈肯）和**，记作 $S + S'$。特别地，我们总可以取

[3]一些作者使用“正规圆盘”这一术语（例如，参见 [21]）。我们将“正规圆盘”一词专门用于指称构成适当嵌入圆盘的正规曲面。

正规曲面 S 与自身相加 k 次，其中 k 是某个正整数；我们将该曲面记作 kS。一个重要的特殊情形是曲面 $2S$，我们称之为 S 的 **双倍曲面**。

正如我们将在第3.1节和第3.2节中看到的，通常只需关注由多胞体 $\mathcal{P}(T)$ 的顶点的最小标量倍数所对应的正规曲面（这里“最小”指使所有坐标均为整数的最小标量倍数）；这类曲面称为**顶点正规曲面**[4]。粗略地说，顶点正规曲面的重要性在于，它们构成了所有正规曲面集合的一个有限且可通过算法枚举的“基”。

2.5 透镜空间

现在我们简要回顾透镜空间，同时说明本文采用的约定。透镜空间有几种等价的定义方式。对我们的目的而言，最方便的方式是将两个实心环面粘合起来构造透镜空间。

我们可以先沿着一个实心环面边界中的非平凡圆 γ 粘上一块加厚圆盘，从而完成任意此类粘合。要完成粘合，只需用一个3维球填充由此得到的2-球面边界。由于用3维球填充的方式只有一种，因此粘合完全由圆 γ 的选取决定。

因此，我们可以通过参数化 γ 的所有可能选择，来参数化所有可能的透镜空间。为此，将每个实心环面视为乘积 $D^2 \times S^1$。在同伦意义下，边界环面 $\partial D^2 \times S^1$ 中每个非平凡圆在普遍覆盖 $\mathbb{R}^2$ 中都提升为一条斜率为有理数或无穷大的直线；我们可以用该斜率来参数化这些圆。这里约定，形如 $\{x\} \times S^1$ 的圆斜率为 ∞，而形如 $\partial D^2 \times \{y\}$ 的圆斜率为 0。

考虑到以上内容，将 γ 的斜率取为 $\frac{p}{q}$ 的构造所得到的三维流形定义为**透镜空间**$L_{p,\ q}$。[5] 这种参数化并不唯一：事实证明，$L_{p,\ q}$与 $L_{p',\ q'}$ 同胚，当且仅当 $|p'| = |p|$ 与 $q' \equiv \pm q^{\pm 1}$（模 p）[3]。有些作者会将退化情形排除在透镜空间的定义之外——最常见的是三维球面（$L_{1,\ 0}$）和/或 $S^2 \times S^1$（$L_{0,\ 1}$）——但我们在此不作此类限制。

2.6 塞弗特纤维空间

现在，我们简要回顾塞弗特纤维空间，同时也借此确立一些记号。关于塞弗特纤维空间的深入讨论有许多来源；这里的大部分内容依据 Hatcher 的《3维流形讲义》 [19], 中的讨论，但在无需完整讨论一般理论的地方做了简化。

我们先构造纤维化实心环面。取圆柱体 $D^2 \times [0,1]$，并将圆盘 $D^2 \times \{0\}$ 和 $D^2 \times \{1\}$ 以 $\frac{2\pi p}{q}$ 旋转粘合，得到一个实心环面，其中 q 是正整数，而 p 是任意一个与 q互素的整数。这个实心环面可按以下方式由圆纤维化:

- 有一条纤维 $\{0\} \times S^1$，由闭合线段 $\{0\} \times [0,1]$得到。
- 其余纤维由拼接形如 $\{x\} \times [0,1]$的 q 段构成。

We 我们将这个实心环面（连同其纤维化）称为具有重数 q 的**纤维化实心环面**。

塞弗特纤维空间是由圆纤维化的一种三维流形，使得每条圆纤维——称为**塞弗特纤维**——都有一个小的实心环面邻域，且该邻域构成一个纤维化实心环面[6]。如果纤维化实心环面邻域的重数为一，我们就称该塞弗特纤维为**正则纤维**；否则，就称其为**例外纤维**[7]。例外纤维彼此孤立，且与流形的边界不相交。

考虑一个紧致塞弗特纤维空间 $\mathcal{M}$，设 k 为 $\mathcal{M}$ 中例外纤维的数量（该数量必然有限）。移除每个例外纤维的一个小实心环面邻域后，我们得到一个

[4]有些作者使用不同的术语。例如，在 [23], 中，顶点正规曲面称为“顶点解”，而“顶点曲面”一词指的是连通的 *two-sided* 正规曲面，它要么是顶点解，要么是顶点解的双倍曲面。[5]按照我们在此采用的约定，这等价于说，$L_{p,\ q}$ 是通过对平凡纽结进行 $\frac{p}{q}$ 德恩手术得到的。[6]有些作者还允许纤维化实心克莱因瓶邻域（例如，参见 [39]）。[7]其他作者对例外纤维也使用各种别称，例如 *multiple fibres* [19] 或 *critical fibres* [39]。

圆丛 $\mathcal{M}'$，其底曲面为 B'；曲面 B' 有 k 个边界分支，对应于我们移除的例外纤维（也可能还有一些边界分支对应于 $\mathcal{M}$ 的边界分支）。设 B 为用圆盘填充 B' 的这 k 个边界分支后得到的曲面；我们称 $\mathcal{M}$ 是 **以 为底的（塞弗特）纤维空间**，并称 B 为**底曲面**。

从此处开始，我们只讨论紧致、连通、可定向且不可约的塞弗特纤维空间。不可约这一限制仅排除了三个例子：

- $S^2 \times S^1$ （2-球面 S^2 与圆 S^1的积），
- $S^2 \tilde{\times} S^1$ （ S^2 与 S^1的扭曲积），以及
- $\mathbb{R}P^3 \# \mathbb{R}P^3$ （两个实射影空间 $\mathbb{R}P^3$的连通和）。

现在我们来谈谈塞弗特纤维空间$\mathcal{M}$ （满足刚才所述限制）中的不可压缩、 ∂-不可压缩曲面[8]。如果嵌入曲面由正则纤维的并组成，我们称其为**竖直**曲面；如果它与所有纤维都横截相交，则称其为**水平**曲面。除同伦外，空间 $\mathcal{M}$ 中的每个不可压缩、 ∂-不可压缩曲面要么是竖直曲面，要么是水平曲面。反过来，每个连通的双侧水平曲面都是不可压缩且 ∂-不可压缩的。然而，对于竖直曲面，这一结论并不完全成立：

- 若一个竖直环面围成一个纤维化实心环面，则它是可压缩的；并且
- 若一个竖直环带从 $\mathcal{M}$ 一个带有乘积纤维化的实心环面中截出一部分，则它是 ∂-可压缩的。

事实证明，这些是仅有的例外；其他所有连通的双侧竖直曲面都是不可压缩的，并且 ∂-不可压缩。

假设除了前面提到的限制之外，塞弗特纤维空间 $\mathcal{M}$ 还是闭的。如第 1.1 节所述，我们尤其关注 $\mathcal{M}$ 为小 的情形（回想第2.3节，这意味着 $\mathcal{M}$ 不含 S^2 或$\mathbb{R}P^2$以外的双侧不可压缩曲面）。[9] 事实证明，除非出现以下任一情况，否则我们总能在 $\mathcal{M}$ 中找到一个不可压缩的竖直环面：

- $\mathcal{M}$ 纤维化于 S^2 之上，且例外纤维至多有三个；或者
- $\mathcal{M}$ 纤维化于 $\mathbb{R}P^2$ 之上，且例外纤维至多有一个。

换言之，如果 $\mathcal{M}$ 是小的，那么它必定以这两种方式之一纤维化（但反过来不成立，因为以这种方式纤维化的三维流形仍可能包含双侧不可压缩的水平曲面）。

考虑 $\mathcal{M}$ 在 $\mathbb{R}P^2$ 上纤维化且至多有一个例外纤维的情形。如果没有例外纤维，那么 $\mathcal{M}$实际上必定与 $\mathbb{R}P^3\#\mathbb{R}P^3$同胚；这是我们此前选择忽略的可约情形之一。如果有一个例外纤维（底曲面仍为 $\mathbb{R}P^2$），那么 $\mathcal{M}$ 就是一个**棱柱流形** [10]。出于种种原因，文献中常将棱柱流形视为一种特殊情形。其中一个原因是，棱柱流形也可以在 S^2 上纤维化，并有三个例外纤维（稍后我们会对此作进一步说明）。由于我们要寻找的是不含塞弗特纤维边的三角剖分，多种塞弗特纤维化的存在对我们的目的而言有些不便，因此我们通常会在讨论中排除棱柱流形。

现在我们考虑这样的情形： $\mathcal{M}$ 以 S^2 为底纤维化，且至多有两个例外纤维。在这种情况下， $\mathcal{M}$ 是一个透镜空间，并且它有无穷多种塞弗特纤维化。因此，在讨论小塞弗特纤维空间时，我们通常也会排除透镜空间。

这就只剩下 $\mathcal{M}$ 以 S^2 为底、恰有三个例外纤维的情形。除了前面提到的棱柱流形外，这些流形都有唯一的塞弗特纤维化。以下构造为这些塞弗特纤维空间提供了一种自然的参数化方式：

[8]在他的《3维流形讲义》中，Hatcher 将这些曲面称为“本质” [19, 第 18]页。我们不采用这一术语，因为按照第2.3节中的定义，我们要求本质曲面不是边界平行的。[9]在塞弗特纤维空间这一特定语境下，有些作者将“小”定义为底曲面为 S^2 且至多有三个例外纤维（例如，参见 [28]）。[10]这一术语与 [39]一致。其他作者对棱柱流形的定义有所不同（例如，参见 [27]）。

构造 5 (底曲面为 S^2 且有三个例外纤维的塞弗特纤维空间**)**。 取圆丛 $P \times S^1$，其中 P 是一个裤子曲面 （即一个去掉三个小圆盘的2球面），并将一个实心环面分别粘到该圆丛的三个环面边界分支上。

我们用斜率来指定这三种粘合方式（类似于第 *2.5* 节中构造透镜空间的方法）。为确保斜率定义明确，固定 $P \times S^1$ 的一个定向。令 b_0、 b_1 和 b_2 表示构成 P 边界分支的圆，固定任意一个 $i \in \{0, 1, 2\}$，并考察环面边界分支 $b_i \times S^1$。这里约定，形如 $b_i \times \{y\}$ 的圆的斜率为 0，而 $b_i \times S^1$ 中的任意纤维（其形式为 $\{x\} \times S^1$）的斜率为 ∞。将实心环面 $\mathcal{T}$ 粘合到 $b_i \times S^1$ 的方式，完全由一个非平凡圆的斜率决定；沿该圆粘合 $\mathcal{T}$ 的子午圆盘，因此我们可以用这个斜率对粘合方式进行参数化。

令 $\frac{\alpha_i}{\beta_i} \in \mathbb{Q}$ （其中 $\beta_i > 0$，且 $\gcd(\alpha_i, \beta_i) = 1$）表示将实心环面 $\mathcal{T}$ 粘合到 $b_i \times S^1$ 时所用的斜率，我们可以扩展圆丛 $P \times S^1$ 上的纤维化，使 $\mathcal{T}$ 成为重数为 β_i 的纤维化实心环面。因此，只要我们通过非整数斜率粘合所有三个实心环面，由此构造得到的 3 维流形便会纤维化于 S^2 之上，且恰有三个例外纤维；本文中我们将这个塞弗特纤维空间记为 $\mathcal{S}\left(\frac{\alpha_0}{\beta_0}, \frac{\alpha_1}{\beta_1}, \frac{\alpha_2}{\beta_2}\right)$。

构造 5 末尾给出的参数化并不唯一；关于何时两个这样的参数化描述相同的塞弗特纤维化，详见 [19] 的命题 2.1。

我们之前还提到，三维流形可能有不止一种 Seifert 纤维化方式。特别是，我们指出，对于底曲面为 S^2 且有三个例外纤维的情形，这种非唯一性只会出现在棱柱流形中。现在我们可以更精确地描述这种非唯一性：除了纤维化为 $\mathbb{R}P^2$ 且有一个例外纤维之外，每个棱柱流形还具有形式为 $\mathcal{S}\left(\frac{1}{2}, -\frac{1}{2}, \frac{\alpha}{\beta}\right)$ 的塞弗特纤维化。

3 关键工具

本节将介绍一些对我们的工作至关重要的新算法和实现。我们预计这些算法也可能适用于其他用途。

3.1 把手体识别

为了找到猜想1和猜想2的反例，我们需要能够识别实心环面和亏格 2 把手体的算法。Regina 实现的实心环面识别算法 [17, 23]在实践中效率极高 [9, 6, 13]。我们将其推广到识别任意亏格的把手体，并改进了 Jaco 和 Tollefson 先前的算法 [23,算法 93]。

下面的定理是 Jaco 和 Tollefson 的推论6.4 [23], 的直接推论，也是我们的把手体识别算法的核心：

定理 6。 设 $\mathcal{T}$ 是一个具有可压缩边界的紧致不可约3维流形的三角剖分。那么 $\mathcal{T}$ 包含一个实现本质压缩圆盘的顶点正规曲面。

请注意，把手体满足该定理的条件。有鉴于此，我们算法的基本策略是反复沿顶点正规压缩圆盘切割；当且仅当这样做最终将$\mathcal{T}$分解为一组 3-球时，三角剖分$\mathcal{T}$才是一个把手体。出于计算方面的考虑，我们对这一方法做了两处修改。

首先，我们将关注范围从圆盘扩展到所有欧拉示性数 $\chi(S)$ 为正的顶点正规曲面 S 。其原因在于， $\chi(S)$ 可以表示为正规坐标中的线性函数，因此我们可以利用线性规划技术。这就得到了一种用于检测非平凡正规球面和圆盘的算法，在实际应用中速度很快（尽管从理论上说，这实际上是实心环面识别和把手体识别的指数时间瓶颈）；详情见 [13] 。

其次，由于沿正规曲面切割可能使四面体数量呈指数级增长，我们改用一种称为**压碎**的技术。压碎最初由 Jaco 和

Rubinstein [21], 在 Casson 早先未发表的工作基础上发展而来，后来又由第一作者 [7]加以改进。压碎非平凡正规曲面有一项关键优势：它保证产生一个四面体数量严格少于原先的新三角剖分。代价是，压碎正规球面或圆盘可能会带来拓扑方面的副作用，详见下述定理（这是对 [7], 中定理 2 的重述，而该定理本身总结了 [21]中的一系列结果）：

定理 7。 设 $\mathcal{T}$ 是一个三维流形三角剖分，其底层三维流形 $\mathcal{M}$ 可定向。设 S 是 $\mathcal{T}$ 中的一个正规 *2*-球面或圆盘。压碎 S 后得到一个三维流形三角剖分 $\mathcal{T}'$，其底层三维流形 $\mathcal{M}'$ 可通过有限次（也可能为零次）以下操作从 $\mathcal{M}$ 得到：

- 撤销连通和；
- 沿适当嵌入圆盘切开；
- 用 *3*-球填充边界 *2*-球；以及
- 删除 *3*-球、*3*-球面、 $\mathbb{R}P^3$、 $L_{3,\ 1}$ 或 $S^2 \times S^1$ 分支。

在本节余下部分，若一个连通可定向三维流形满足以下条件，则称其为**有趣的**：

- 它恰有一个边界分支；并且
- 其第一同调群为 $\mathbb{Z}^g$，其中 g 是边界曲面的亏格。

换句话说，如果三维流形 $\mathcal{M}$ 具有与把手体相同的基本性质，我们就认为它很有研究价值；这意味着我们需要更精妙的技术来判定 $\mathcal{M}$ 是否为把手体。类比于同调球面，也可以称这种有研究价值的三维流形为“同调把手体”。我们的把手体识别算法依赖于定理 7 的以下推论：

命题 8。 设 $\mathcal{T}$ 是一个有研究价值的三维流形的有界三角剖分，并假设 $\mathcal{T}$ 包含一个非平凡的正规2球面或圆盘 S。令 $\mathcal{T}'$ 表示通过压碎S得到的三角剖分。 $\mathcal{T}'$ 的每个分支均为以下两者之一：

- 第一同调群平凡的闭3维流形；或
- （一个.....的有界三角剖分）有研究价值的3维流形。此外， $\mathcal{T}$是把手体，当且仅当 $\mathcal{T}'$ 的每个分支都是3维球面或把手体。

证明。 $\mathcal{T}'$ 的底流形可由 $\mathcal{T}$的底流形经过定理 7 中所列操作的一系列步骤得到。我们对这些操作的次数使用归纳法来证明该命题。

对于基础情形，如果不执行定理 7 中的任何操作，那么 $\mathcal{T}$和 $\mathcal{T}'$ 对应的底三维流形同胚。此时无需证明。

对于归纳步骤，假设我们使用定理 7 中的一系列操作改变 $\mathcal{T}$ 的底三维流形。令 $\mathcal{M}$ 表示所得三维流形的任意一个分支，并令 $\mathcal{M}'$ 表示对 $\mathcal{M}$ 执行定理 7 中任意一个操作后得到的三维流形。只需证明以下两个断言：

(a) $\mathcal{M}'$ 的每个分支要么是第一同调群平凡的闭三维流形，要么是一个有趣的三维流形。(b) 三维流形 $\mathcal{M}$ 是三维球面或把手体，当且仅当 $\mathcal{M}'$ 的每个分支都是三维球面或把手体。

这两项主张的论证取决于 $\mathcal{M}$ 是闭还是有趣，也取决于定理 7 中的哪种操作关联了 $\mathcal{M}$ 和 $\mathcal{M}'$。

我们先讨论 $\mathcal{M}$ 是闭的情况，因为这种情况相对简单。定理 7 中只有两种操作适用于此处：

- 我们可以直接删除 $\mathcal{M}$，此时 $\mathcal{M}'$ 为空。只有当 $\mathcal{M}$ 是 S^3的一个副本、 $L_{3,\ 1}$或 $S^2 \times S^1$时，才会出现这种情况；然而，根据归纳假设，我们可以假定 $\mathcal{M}$ 的第一同调群是平凡的，因此 $\mathcal{M}$ 实际上必定与 S^3同胚。因此，在这种情况下，命题 (a) 和 (b) 显然都成立。

- 我们可以消去一个连通和，此时 $\mathcal{M}'$ 由两个闭分支 $\mathcal{C}_1$ 和 $\mathcal{C}_2$组成。由于

$$H_1(\mathcal{C}_1) \oplus H_1(\mathcal{C}_2) \simeq H_1(\mathcal{C}_1 \# \mathcal{C}_2) \simeq H_1(\mathcal{M}) \simeq 0,$$

由此可得， $\mathcal{C}_1$ 和 $\mathcal{C}_2$ 的第一同调群均为平凡群，因此命题 (a)成立。命题 (b)也成立，因为 $\mathcal{M}$ 是三维球面，当且仅当 $\mathcal{C}_1$ 和 $\mathcal{C}_2$ 都是三维球面。

既然这一点已经解决，证明余下部分将专门讨论 $\mathcal{M}$ 有趣的情形；令 g 表示边界曲面 $\partial\mathcal{M}$的亏格。定理 7中列出的操作里，以下几种相对容易处理：

- 我们可以直接删除 $\mathcal{M}$ ，此时 $\mathcal{M}'$ 为空；这只会在 $\mathcal{M}$ 是一个3维球时发生。在这种情况下，命题 (a)和 (b)显然都成立。
- 如果 $\mathcal{M}$ 有2-球面边界，那么我们可以用一个3维球填充这个2-球面，此时 $\mathcal{M}'$ 是闭的。由于 $\mathcal{M}$ 是有趣的，且其边界亏格为0，因此有 $H_1(\mathcal{M}) = 0$，这意味着 $\mathcal{M}'$ 的第一同调群也平凡；故命题 (a)成立。此外， $\mathcal{M}$ 是一个3维球，当且仅当 $\mathcal{M}'$是三维球面，所以命题 (b)也成立。
- 我们可以消去一个连通和。在这种情况下，由于 $\mathcal{M}$ 恰有一个边界分支， $\mathcal{M}'$有两个分支：一个闭分支 $\mathcal{C}$，以及一个与 $\mathcal{M}$有相同边界的分支 $\mathcal{B}$ 。由于 $H_1(\partial\mathcal{B})$ 的秩为 $2g$，我们知道 $H_1(\mathcal{B})$ 的秩至少为 g （见 [19]的引理 3.5）。因此，由于

$$H_1(\mathcal{B}) \oplus H_1(\mathcal{C}) \simeq H_1(\mathcal{B} \# \mathcal{C}) \simeq H_1(\mathcal{M}) \simeq \mathbb{Z}^g,$$

我们得出 $H_1(\mathcal{B}) \simeq \mathbb{Z}^g$ 和 $H_1(\mathcal{C}) \simeq 0$，从而命题 (a) 成立。此外， $\mathcal{M}$ 是把手体，当且仅当 $\mathcal{C}$ 是三维球面且 $\mathcal{B}$ 是把手体，因此命题 (b) 也成立。[11]

剩下的情况是沿适当嵌入圆盘 D切开；相比其他操作，处理这一剩余操作的论证需要多费一些功夫。

我们先考虑 D 是一个分离圆盘的情形。在这种情况下， $\mathcal{M}'$ 有两个分支 $\mathcal{B}_1$和 $\mathcal{B}_2$，每个分支都只有一个边界分支。对每个 $i \in \{1, 2\}$，令 g_i 表示边界曲面 $\partial\mathcal{B}_i$的亏格；注意到 $H_1(\mathcal{B}_i)$ 的秩至少为 g_i。将 D 收缩到一个点后，我们看到$\mathcal{M}$ 与楔和 $\mathcal{B}_1 \vee \mathcal{B}_2$同伦等价，因此

$$H_1(\mathcal{B}_1) \oplus H_1(\mathcal{B}_2) \simeq H_1(\mathcal{B}_1 \vee \mathcal{B}_2) \simeq H_1(\mathcal{M}) \simeq \mathbb{Z}^g.$$

由此，再结合 $g = g_1 + g_2$这一事实，我们得出 $\mathcal{B}_1$ 和 $\mathcal{B}_2$ 都很有趣，因此命题 (a)成立。命题 (b)也成立，因为 $\mathcal{M}$ 是把手体，当且仅当 $\mathcal{B}_1$ 和$\mathcal{B}_2$ 都是把手体。

剩下的情况是 D 为非分离的情形。在这种情况下， $\mathcal{M}'$ 由一个有界分支构成；与此前所有情形相比，这种情况需要多得多的工作。我们首先注意到， $\mathcal{M}$ 是把手体，当且仅当 $\mathcal{M}'$ 是把手体，因此命题 (b)成立。因此，这个证明的其余部分完全致力于证明 $\mathcal{M}'$ 很有趣，从而验证命题 (a)。第一步是证明 ∂D 是边界曲面 $\partial\mathcal{M}$上的一条非分离曲线，因为这将意味着 $\mathcal{M}'$ 只有一个边界分支。[12]

反过来，假设 ∂D 将 $\partial\mathcal{M}$ 分成两部分 P 和 Q。令 h 表示 P的亏格。由于 D 是一个非分离圆盘，因此在 $\mathcal{M}$ 的内部存在一个闭环 γ ，与 D 恰好相交一次。将 $P \cup D$ 稍微推离边界，得到一个闭的亏格为h 的曲面 E ，它与 γ 恰好相交一次。令 X' 为 $E \cup \gamma$的一个小邻域，并令 $Y' = \mathcal{M} - X'$。 X' 的边界是一个亏格为 $2h$的曲面；令 N 为该曲面的一个小邻域。取 $X = X' \cup N$ 和$Y = Y' \cup N$，得到一对子空间，其内部并集覆盖 $\mathcal{M}$。注意到 $X \cap Y = N$，对 X 和 Y 应用 Mayer-Vietoris 序列，可得以下正合序列：

[11]这里， $\mathcal{M}$ 恰好只有一个边界分支至关重要，因为这排除了 $\mathcal{M}$ 是两个把手体的连通和的情形（否则，即使 $\mathcal{M}$ 本身不是把手体， $\mathcal{M}'$ 也可能是这两个把手体的不交并）。在证明的后面，我们会看到，为什么必须作出更强的假设，即 $\mathcal{M}$ 实际上很有趣。[12]前面我们指出，绝不能产生边界分支多于一个的分支，这一点至关重要。正如我们即将看到的，要在当前情形下证明这一点，需要假设 $H_1(\mathcal{M}) \simeq \mathbb{Z}^g$，这就是为什么我们必须假设 $\mathcal{M}$ 很有趣。

$$\cdots \longrightarrow H_1(N) \xrightarrow{\varphi} H_1(X) \oplus H_1(Y) \xrightarrow{\psi} H_1(\mathcal{M}) \longrightarrow \cdots$$
$$\mathbb{Z}^{4h} \qquad \mathbb{Z}^{2h+1} \oplus H_1(Y) \qquad \mathbb{Z}^g$$

此外，由于 Y 有两个边界分支，一个亏格为 g，另一个亏格为 $2h$，我们知道 $H_1(Y)$的秩至少为 $g+2h$。因此，总的来说， $H_1(X) \oplus H_1(Y)$的秩至少为 $g+4h+1$。但我们也知道 $\mathrm{im}(\psi)$是秩至多为 g 的自由阿贝尔群，因此由正合性可知，$\mathrm{im}(\varphi)$的秩至少为 $4h+1$。这不可能，因为 $H_1(N)$的秩只有 $4h$。

因此，我们得到了所需结论： ∂D 是 $\partial\mathcal{M}$上的一条非分离曲线。这意味着 $\mathcal{M}'$ 恰有一个边界分支，且该边界分支的亏格为 $g-1$。为了完成证明，我们只需验证 $H_1(\mathcal{M}') \simeq \mathbb{Z}^{g-1}$，从而证明 $\mathcal{M}'$ 具有趣味性。

一种方法是考察子空间 X 和 Y 的 Mayer-Vietoris 序列，它们按如下方式构造：

- 设 X 是 $\mathcal{M}$ 的一个子空间，且与 $\mathcal{M}'$同胚。
- 设 Y 是圆盘 D的一个三维球邻域。

我们可以选取 X 和 Y，使得它们的内部覆盖 $\mathcal{M}$，而交集 $N := X \cap Y$是两个互不相交的3维球。这给出以下正合序列：

$$\cdots \longrightarrow H_1(N) \xrightarrow{\varphi_1} H_1(X) \oplus H_1(Y) \xrightarrow{\psi_1} H_1(\mathcal{M}) \xrightarrow{\partial} H_0(N) \xrightarrow{\varphi_0} H_0(X) \oplus H_0(Y) \xrightarrow{\psi_0} H_0(\mathcal{M}) \longrightarrow 0$$
$$0 \qquad H_1(\mathcal{M}') \qquad \mathbb{Z}^g \qquad \mathbb{Z}^2 \qquad \mathbb{Z}^2 \qquad \mathbb{Z}$$

由正合性可知：

- $H_1(\mathcal{M}') \simeq \ker(\partial)$；并且
- $\mathrm{im}(\psi_0)$、$\mathrm{im}(\varphi_0)$ 和 $\mathrm{im}(\partial)$ 的秩均为 1。

因此，我们看到 $H_1(\mathcal{M}') \simeq \mathbb{Z}^{g-1}$，正如所要求的。 □

我们依赖的最后两项是用于识别三维球面 [21, 36, 40] 和三维球的算法（三维球识别是三维球面识别的一个众所周知的变体）。这两种算法的实现均可在 Regina [6, 9]中找到。

以下把手体识别算法现在也已在 Regina [9]中提供：

算法 9 (把手体识别)。 要测试有界三角剖分 $\mathcal{T}$ 是否为把手体，并在是的情况下确定其亏格：

(1) 检查 $\mathcal{T}$ 是否连通、可定向且恰有一个边界分支。如果 $\mathcal{T}$ 不满足其中任何一项条件，则终止并返回 -1 （表示 $\mathcal{T}$ 不是把手体）。

(2) 计算 $\mathcal{T}$ 的边界的亏格 g。
- 在 $g=0$ 的情况下，检查 $\mathcal{T}$ 是否为一个3球。若是，则终止并返回0 （表示 $\mathcal{T}$ 是亏格为0 的把手体）；否则，终止并返回 -1。
- 在 $g>0$ 的情况下，检查 $\mathcal{T}$ 的第一同调群是否为 $\mathbb{Z}^g$。若不是，则终止并返回 -1。

(3) 创建待处理三角剖分列表 $\mathcal{L}$，初始时仅包含 $\mathcal{T}$。当 $\mathcal{L}$ 非空时：

> *(i)* 令 $\mathcal{F}$ 为 $\mathcal{L}$ 中第一个出现的三角剖分。从 $\mathcal{L}$ 中移除 $\mathcal{F}$。*(ii)* 在 $\mathcal{F}$ 中寻找一个非平凡的正规2球面或圆盘 S。如果不存在这样的曲面，则终止并返回 -1。*(iii)* 压碎 S。对于压碎后得到的三角剖分 $\mathcal{R}$ 的每个分支 $\mathcal{C}$：
>
> - 如果 $\mathcal{C}$ 是闭的，则检查 $\mathcal{C}$ 是否为3维球面。如果是，则丢弃 $\mathcal{C}$ 并继续处理 $\mathcal{R}$的下一个分支；否则，终止并返回 -1。
> - 如果 $\mathcal{C}$ 的边界是2维球面，则检查 $\mathcal{C}$ 是否为3维球。如果是，则丢弃 $\mathcal{C}$ 并继续处理 $\mathcal{R}$的下一个分支；否则，终止并返回 -1。
> - 在其他任何情况下，将 $\mathcal{C}$ 添加到列表 $\mathcal{L}$中，然后继续处理 $\mathcal{R}$的下一个分支。

(4) 当 $\mathcal{L}$中不再有三角剖分时，终止并返回 g （表示 $\mathcal{T}$是一个亏格为g的把手体）。

定理 10 （正确性）。 算法 *9* 能正确判断给定的有界三角剖分是否为把手体。

证明。 步骤（2）会排除边界为 2-球面的三维流形，以及不具备研究意义的三维流形。因此，在整个证明过程中，我们可以放心地假设初始三角剖分 $\mathcal{T}$ 表示一个具有研究意义的三维流形，其边界曲面具有正亏格。

我们首先证明该算法会终止。只需证明步骤（3）中的循环不会无限持续。事实上，每当我们从列表 $\mathcal{L}$ 中移除一个三角剖分 $\mathcal{F}$ 时，要么压碎一个非平凡正规曲面 S，要么因为不存在合适的曲面 S 而终止。压碎 S 保证会减少四面体的数量，因此，即使我们将一些三角剖分重新加入列表 $\mathcal{L}$，列表中四面体的总数也必定会减少。这个数量不可能无限减少，所以算法必定会终止。

现在我们证明，该算法能够正确判断输入三角剖分 $\mathcal{T}$ 是否为把手体。关键在于以下不变量：每次我们在步骤（3）中对循环进行一次完整遍历后，该不变量都得以保持： $\mathcal{L}$ 中的每个三角剖分都很有意思，并且 $\mathcal{T}$ 是正亏格把手体，当且仅当 $\mathcal{L}$ 中的每个三角剖分实际上都是正亏格把手体。

为证明该不变量确实得以保持，我们先考虑从 $\mathcal{L}$ 中移除的三角剖分 $\mathcal{F}$ 是正亏格把手体的情形。在这种情况下，定理 6 保证我们会找到一个非平凡的正规 2-球面或圆盘 S，位于 $\mathcal{F}$ 中。接着，根据命题 8，压碎 S 会得到一个新的三角剖分，其各个分支要么是三维球面，要么是把手体。我们舍弃所有三维球面和 3-球，只留下正亏格把手体。这足以说明，当初始三角剖分 $\mathcal{T}$ 是把手体时，不变量始终得以保持。

现在，假设我们从 $\mathcal{L}$ 中移除一个三角剖分 $\mathcal{F}$，而它不是把手体。我们需要检查：如果遍历循环时不终止并返回 -1，不变量是否仍然保持。因此，可以假设我们压碎 $\mathcal{F}$ 中一个非平凡的正规 2-球面或圆盘，从而得到一个新的三角剖分 $\mathcal{F}'$。还可以假设 $\mathcal{F}'$ 的所有闭分支都是三维球面，且 $\mathcal{F}'$ 中所有具有 2-球面边界的连通分支都是 3维球；我们将这些分支全部丢弃。命题 8 告诉我们，剩余的连通分支都很有意义，而且其中至少有一个不是把手体；我们将所有这些有意义的连通分支重新加入 $\mathcal{L}$，从而确保 $\mathcal{L}$ 中仍至少包含一个不是把手体的三角剖分。这表明，当初始三角剖分 $\mathcal{T}$ 不是把手体时，不变量始终保持不变。

总而言之，如果 $\mathcal{T}$是 把手体，那么列表 $\mathcal{L}$ 最终必定变为空；此时我们终止，并返回步骤（2）中计算出的亏格 g；这正确地表明 $\mathcal{T}$ 是亏格为 g 的把手体。另一方面，如果 $\mathcal{T}$不是 把手体，那么 $\mathcal{L}$ 永远不会变空，因此我们必定会在某个时刻终止，并正确地返回 -1。 □

3.2 检测塞弗特纤维边

为了找到猜想 3 的反例，我们需要能够取出小塞弗特纤维空间的单顶点三角剖分中的一条边 e，并判断它是否为塞弗特纤维边。由于很难对此作出确切判断，而且我们的主要目标是找到一个确定不含塞弗特纤维边的三角剖分，因此我们采用一种算法（见算法 17），它只能在一个方向上给出确切答案：证明一条边不是同伦于塞弗特纤维。

在本节中，设 $\mathcal{T}$ 是一个不可约小塞弗特纤维空间的单顶点三角剖分，该空间既不是透镜空间，也不是棱柱流形，并固定 $\mathcal{T}$ 的一条边 e。（回顾我们在第 2.6 节中的讨论， $\mathcal{T}$ 必须以三个例外纤维为底空间进行纤维化，而且这是 $\mathcal{T}$ 上唯一的塞弗特纤维化。）我们的算法的第一步是构造一个三角剖分 $\mathcal{D}$，对应于删去 e 的一个小实心环面邻域后得到的三维流形。接下来，我们的策略是列出一组正规曲面：如果 e 是一条塞弗特纤维边，这些曲面必定会出现在 $\mathcal{D}$ 中；然后检查它们是否全部出现在 $\mathcal{D}$ 中。如果缺少其中任何一个曲面，我们就能确定 e 不是一条塞弗特纤维边。只要使用足够全面的曲面列表，这一策略似乎就有效。

我们采用的关键理论结果如下，它是 Jaco 和 Tollefson 所得的推论 6.8 的一个推论：

命题 11. 设 $\mathcal{F}$ 是一个可定向、不可约且 ∂-不可约的塞弗特纤维空间的有界三角剖分。若 $\mathcal{F}$ 中的一个曲面是本质的，并且是顶点正规曲面或顶点正规曲面的双倍曲面，则称其为良好曲面。

- 如果 $\mathcal{F}$ 有垂直本质环带，那么它就有良好垂直环带。
- 如果 $\mathcal{F}$ 有本质竖直环面，那么它就有良好竖直环面。此外，如果 $\mathcal{F}$ 有多个本质竖直环面（相差同伦），那么它至少有两个良好竖直环面。

证明。首先，假设 $\mathcal{F}$ 有一个本质垂直环带 A。[23] 的推论 6.8 告诉我们，A 同伦于一个正规环带 S，并且对于某个 $k \geqslant 1$，曲面 kS 是若干良好环带和良好环面的和。注意，其中至少有一个加数必须是良好垂直环带。

现在假设 $\mathcal{F}$ 有一个本质竖直环面 A。这一次，[23] 的推论 6.8 告诉我们，A 同伦于一个正规环面 S，并且对某个 $k \geqslant 1$，曲面 kS 是若干良好环面的和。由于 $\mathcal{F}$ 的边界非空，这些良好环面的加项必定都是竖直的。此外，如果 $\mathcal{F}$只包含一个良好竖直环面 V，那么将 V 与自身相加只能得到多个彼此平行的 V；因此在这种情况下，V必定是 $\mathcal{F}$中唯一的本质竖直环面（至多相差同伦）。 □

我们先讨论如何检验 e 是否同伦于一条例外纤维，因为这比正则情形容易得多。如果 e 同伦于一条例外纤维，那么注意到三角剖分 $\mathcal{D}$以一个带有两条例外纤维的圆盘为底纤维化，这意味着 $\mathcal{D}$ 包含一个竖直环带 A，它将 $\mathcal{D}$ 切成一对实心环面。因此，如果我们能证明不存在这样的环带 A，就能确认 e 不同伦于一条例外纤维。以下算法的核心思想即在于此（该算法是算法 17 的一个重要子程序）：

算法 12。 设 $\mathcal{F}$ 是一个可定向不可约3流形的有界三角剖分，且具有一个环面边界分支（没有其他边界分支）。要检验 $\mathcal{F}$ 是否以圆盘为底纤维化且有两个例外纤维：

(1) 创建一个空列表 $\mathcal{A}$。*(2)* 枚举 $\mathcal{F}$中的所有顶点正规曲面。对于每个这样的曲面 S：

- 如果 S 是一个本质压缩圆盘，则终止并返回`false`。
- 如果 S 是一个环带，则将 S 添加到 $\mathcal{A}$。
- 如果 $2S$ 是一个环带，则将 $2S$ 添加到 $\mathcal{A}$。

(3) 对于 $\mathcal{A}$中的每个环带 S，检查沿 S 切开是否会将 $\mathcal{F}$ 分解为一对实心环面。如果存在一个环带能给出这样的分解，则终止并返回无法确定；否则，终止并返回`false`。

命题 13. 如果算法 *12* 返回 `false`，那么输入三角剖分 $\mathcal{F}$ 不可能以带有两个例外纤维的圆盘为底纤维化。

证明。假设输入三角剖分 $\mathcal{F}$ 以带有两个例外纤维的圆盘为底纤维化。我们需要证明该算法不会返回 false。由于 $\mathcal{F}$ 是 ∂-不可约的，算法在步骤（2）中不可能返回 false。因此，只需证明算法会找到一个将 $\mathcal{F}$ 割成一对实心环面的环带，从而在步骤（3）中返回 inconclusive。为此，注意到 $\mathcal{F}$（在同伦意义下）恰有一个垂直本质环带 A，且 A 恰好具有所需性质；于是命题 11 告诉我们，A 要么是顶点正规曲面，要么是顶点正规曲面的双倍曲面，因此算法会在步骤（2）生成的正规曲面中找到 A。

□

现在我们讨论如何检验 e 是否同伦于一条正则纤维。为此，我们依据以下观察：

观察 14. 假设 e 同伦于一个正则纤维。在这种情况下，三角剖分 $\mathcal{D}$ 以圆盘为底纤维化，且有三个例外纤维，这意味着：

(a) 在同伦意义下， $\mathcal{D}$ 恰好包含三个垂直本质环带。每个环带都将一个例外纤维与另外两个分隔开，因此将 $\mathcal{D}$ 切成两部分 $\mathcal{P}_0$ 和 $\mathcal{P}_1$，其中$\mathcal{P}_0$ 是一个实心环面，而 $\mathcal{P}_1$ 以圆盘为底纤维化，且有两个例外纤维。*(b)* 在同伦意义下， $\mathcal{D}$ 恰好包含三个垂直本质环面。每个环面都包围三个例外纤维中的两个，因此将 $\mathcal{D}$ 切成两部分 $\mathcal{Q}_0$ 和 $\mathcal{Q}_1$，其中 $\mathcal{Q}_0$ 以环带为底纤维化，且有一个例外纤维，而 $\mathcal{Q}_1$ 以圆盘为底纤维化，且有两个例外纤维。

根据观察 14 的 (b) 部分，我们可以看出，要排除$\mathcal{D}$中存在垂直本质环面的可能性，就需要能够识别出 $\mathcal{M}$ 以带有一个例外纤维的环带为底纤维化的三维流形。为此，我们依赖这样一个事实：这类三维流形 $\mathcal{M}$ 具有一个垂直本质环带，该环带可将 $\mathcal{M}$ 切割成一个实心环面。下面的算法（这是算法 17 的另一个重要子程序）通过尝试排除这类环带的存在来进行判断：

算法15。 设 $\mathcal{F}$ 是一个可定向不可约三维流形的有界三角剖分，且有两个环面边界分支（没有其他边界分支）。要测试 $\mathcal{F}$ 是否以带有一个例外纤维的环带为底纤维化：

(1) 创建一个空列表 $\mathcal{A}$。*(2)* 枚举 $\mathcal{F}$中的所有顶点正规曲面。对每个这样的曲面 S:

- 如果 S 是一个本质压缩圆盘，则终止并返回假。
- 若 S 是环带，则将 S 添加到 $\mathcal{A}$。
- 若 $2S$ 是环带，则将 $2S$ 添加到 $\mathcal{A}$。

(3) 对于 $\mathcal{A}$中的每个环带 S ，检查沿 S 切开是否会将 $\mathcal{F}$ 分解成一个实心环面。若存在一个环带能产生这样的分解，则终止并返回无法确定；否则，终止并返回`false`。

命题16。若算法*15*返回`false`，则输入三角剖分 $\mathcal{F}$ 不可能以一个例外纤维的环带为底纤维化。

证明。假设输入三角剖分 $\mathcal{F}$ 以一个例外纤维的环带为底纤维化。我们需要证明该算法不会返回 false；证明与命题 13 的证明几乎完全相同。由于 $\mathcal{F}$ 是 ∂-不可约的，该算法在步骤（2）中不可能返回 false。因此，只需证明该算法会找到一个能将 $\mathcal{F}$ 切成单个实心环面的环带，从而在步骤（3）中返回无法确定。为此，注意到 $\mathcal{F}$ 恰有一个垂直本质环带 A（在同伦意义下），并且 A 恰好具有所需性质；于是命题 11 告诉我们， A要么是顶点正规曲面，要么是顶点正规曲面的双倍曲面，因此该算法会在步骤（2）生成的正规曲面中找到 A 。

□

我们终于可以给出检验 e 是否为塞弗特纤维边的算法（算法 17）了。我们已经提到，困难的情形是检验 e 是否同伦于正则纤维；在这种情形下， $\mathcal{D}$ 必须含有若干垂直本质环带和垂直本质环面。有鉴于此，整体策略与前两个算法类似：利用命题 11，尝试排除这些垂直曲面的存在。

算法 17. 设 $\mathcal{T}$ 是一个不可约小塞弗特纤维空间的单顶点三角剖分，该空间既不是透镜空间，也不是棱柱流形。要检验 $\mathcal{T}$ 的边 e 是否为塞弗特纤维边：
(1) 构造一个有界三角剖分 $\mathcal{D}$ ，对应于删去 e 的小实心环面邻域后得到的 3 维流形。*(2)* 创建空列表 $\mathcal{A}$、 $\mathcal{B}$ 和 $\mathcal{L}$。*(3)* 枚举 $\mathcal{D}$ 中所有顶点正规曲面。对每个这样的曲面 S:

- 如果 S 是本质压缩圆盘，则终止并返回`false`。
- 如果 S 是环带，则将 S 添加到 $\mathcal{A}$。
- 如果 $2S$ 是环带，则将 $2S$ 添加到 $\mathcal{A}$。
- 如果 S 是环面，则将 S 添加到 $\mathcal{B}$。
- 如果 $2S$ 是环面，则将 $2S$ 添加到 $\mathcal{B}$。

(4) 如果列表 $\mathcal{A}$ 仍为空，则终止并返回 `false`。*(5)* 对于 $\mathcal{A}$ 中的每个环带 S :

(i) 沿 S 切开，并检查是否得到两个连通分支 $\mathcal{C}_0$ 和 $\mathcal{C}_1$。如果没有，则继续检查 $\mathcal{A}$ 中的下一个环带。*(ii)* 检查 $\mathcal{C}_0$ 和 $\mathcal{C}_1$ 是否为实心环面。

- 如果它们都是实心环面，则终止并返回无法确定。
- 如果 $\mathcal{C}_0$ 是实心环面而 $\mathcal{C}_1$ 不是，则将 $\mathcal{C}_1$ 添加到 $\mathcal{L}$。
- 如果 $\mathcal{C}_1$ 是实心环面而 $\mathcal{C}_0$ 不是，则将 $\mathcal{C}_0$ 添加到 $\mathcal{L}$。

(6) 如果列表 $\mathcal{B}$ 中环面的数量少于两个，则终止并返回`false`。*(7)* 对列表 $\mathcal{L}$中的每个三角剖分运行算法 *12*；如果所有此类三角剖分的输出均为`false` ，则终止并返回`false`。*(8)* 将整数变量 n 初始化为0。*(9)* 对 $\mathcal{B}$中的每个环面 S ：*(i)* 沿 S切开，并检查是否得到两个分支 $\mathcal{C}_0$ 和 $\mathcal{C}_1$。如果没有，则继续处理 $\mathcal{B}$中的下一个环面。*(ii)* 统计 $\mathcal{C}_0$中的边界分支数量 b 。

- 在 $b=1$的情况下，检查以下各项：– $\mathcal{C}_1$ 有2 个边界分支；– 对 $\mathcal{C}_0$ 运行算法 *12*不会得到输出`false`；并且– 对 $\mathcal{C}_1$ 运行算法*15*不会得到输出`false`。如果这些条件均满足，则将 n 增加1。

- 在 $b=2$ 的情况下，请检查：– $\mathcal{C}_1$ 有1 个边界分支；– 对 $\mathcal{C}_0$ 运行算法*15*不会返回`false`；并且– 对 $\mathcal{C}_1$ 运行算法 *12*不会返回`false`。如果这些条件都满足，则将 n 增加1。

(10) 如果 $n<2$，则终止并返回`false`；否则，终止并返回不确定 .

定理18。 如果算法 *17*返回`false`，则输入边 e 不可能是塞弗特纤维边。

证明。 首先注意，如果 e 与一个例外纤维同伦，那么三角剖分 $\mathcal{D}$ 将以圆盘为底纤维化，且有两个例外纤维。在这种情况下，我们可以使用命题 13 的证明中的相同论证，说明算法 17将在步骤 (5) 返回“不确定”。（注意，在这种情况下， $\mathcal{D}$ 没有垂直本质环面，这就是为什么我们需要等到步骤 (6) 才检查列表 $\mathcal{B}$ 是否至少包含两个环面。）

因此，在本证明余下部分中，我们假设 e 同伦于一条正则纤维；此时 $\mathcal{D}$ 以圆盘为底纤维化，且有三条例外纤维。这意味着 $\mathcal{D}$ 是 ∂-不可约的，因此算法在步骤（3）中不会返回 false。

为了说明算法在步骤（4）和（7）中不会返回 false，我们回顾观察 14 的（a）部分： $\mathcal{D}$ 恰有三个垂直本质环带，每个环带都将 $\mathcal{D}$ 切成一块实心环面部分$\mathcal{P}_0$ 和一块以圆盘为底、具有两条例外纤维的纤维化部分 $\mathcal{P}_1$ 。根据命题 11，这些垂直环带中至少有一个是顶点正规曲面，或是某个顶点正规曲面的双倍曲面。因此，在步骤（3）的某个时刻，我们会将这样的一个环带 S 加入列表 $\mathcal{A}$，这意味着算法在步骤（4）中不会返回 false。此外，在步骤（5）的某个时刻，我们会沿着 S 切开，从而将 $\mathcal{D}$ 分解成前面提到的部分 $\mathcal{P}_0$ 和 $\mathcal{P}_1$ 。部分 $\mathcal{P}_1$ 将被加入列表 $\mathcal{L}$，这意味着在步骤（7）的某个时刻，我们会以输入$\mathcal{P}_1$运行算法 12；根据命题 13，在这种情况下，算法 12 不可能输出 false，因此我们的主算法此时也不可能返回 false。

剩下要证明的是，该算法在步骤（6）和（10）中不会返回 false。为此，我们使用观察 14 的 (b) 部分： $\mathcal{D}$ 恰有三个垂直本质环面，每个环面都会将$\mathcal{D}$ 切成两部分：一部分是以环带为底纤维化且有一个例外纤维的 $\mathcal{Q}_0$ ，另一部分是以圆盘为底纤维化且有两个例外纤维的 $\mathcal{Q}_1$ 。根据命题 11，这些垂直环面中至少有两个会作为顶点正规曲面或顶点正规曲面的双倍曲面出现。因此，在步骤（3）的某个时刻，我们会将这两个环面 S_0 和 S_1 加入列表 $\mathcal{B}$，这意味着算法不会在步骤（6）中返回 false。此外，对于每个 $i\in\{0,1\}$，我们都会在步骤（9）的某个时刻沿着 S_i 切开，从而将 $\mathcal{D}$ 分解为前面提到的两部分 $\mathcal{Q}_0$ 和 $\mathcal{Q}_1$ 。随后，我们在 $\mathcal{Q}_0$ 上运行算法15，并在 $\mathcal{Q}_1$上运行算法 12；根据命题 16 和命题 13，这两个子程序都不会

返回 false。因此，对于每个 $i \in \{0,1\}$，我们将变量 n 递增一次，这意味着主算法不会在步骤（10）中返回 false。 □

算法 12、15 和 17 的实现可在 https://github.com/AlexHe98/ triang-counterex 获取。这些实现中我们尚未提及的唯一一点，是我们加入了一种替代方法，用于检验三维流形是否以特定方式 Seifert纤维化，因为有时这样可以避免枚举顶点正规曲面这一计算代价高昂的任务。具体而言，在算法 17 的步骤 (1) 和 (2) 之间，我们的实现会尝试通过组合识别，检测该三角剖分 $\mathcal{D}$ 是否存在塞弗特纤维化；也就是说，我们对 $\mathcal{D}$ 执行 2-3 和 3-2 变换，目的是得到一个标准的塞弗特纤维空间三角剖分，并由 Regina 的 BlockedSFS 类 [9]识别。

3.3 执行 2-3 和 3-2 变换时跟踪边

假设在某个三角剖分 $\mathcal{T}$中，我们已经知道哪些边（例如）是核心边。回顾第 2.2 节，如果我们对 $\mathcal{T}$ 的一条边 e 执行 3-2 变换，生成一个新的三角剖分 $\mathcal{T}'$，那么 $\mathcal{T}$和 $\mathcal{T}'$ 的 1-骨架之间唯一的区别是，$\mathcal{T}'$ 不再包含边 e。因此，原则上我们无需直接计算 $\mathcal{T}'$ 的哪些边是核心边；只需根据我们已知的 $\mathcal{T}$ 相关信息恢复这些信息即可。

反过来，如果我们通过执行 2-3 变换生成新的三角剖分 $\mathcal{T}'$，那么请回想一下，$\mathcal{T}$和 $\mathcal{T}'$ 的 1-骨架之间唯一的区别是，$\mathcal{T}'$ 包含一条新的边 e。在这种情况下，我们需要检查 e 是否为核心边，但应该不必对其他边重新计算。

在实际操作中，情况比较复杂，因为在 Regina 中执行 2-3 和 3-2 变换时，边的编号可能会任意改变。我们的解决方案是采用定制实现来执行 2-3 和 3-2 变换，并在输出中附带边编号如何变化的说明。源代码可在 https://github.com/AlexHe98/triang-counterex 获取。该实现——更广泛地说，这种追踪初等移动如何改变三角剖分的顶点、边或面的编号的思路——或许还有其他应用。

4 使用定向搜索移除坏边

本节讨论我们用于搜索猜想 1、2 和 3 的反例的算法。为了描述这一搜索算法，我们需要以统一的方式处理核心边、隧道边和塞弗特纤维边。为此，我们注意到，这几类边各自都是由一种在环境同痕下保持不变的性质来定义的；令 P 表示任意此类性质。我们称单顶点三角剖分中的一条边 e为**坏边**，如果 e 满足这一性质 P；否则称其为**好边**。在本文余下部分中，每当我们提到好边和坏边时，所指性质P 要么并不重要，要么可以从上下文中明确看出。

用这一术语来说，寻找一个所需反例的任务可以改述为寻找一个没有坏边的三角剖分。我们实现这一目标的策略源于以下观察，这一点我们在第 2.2 节中已经提到：对一条边e执行 3-2 变换时，对 1-骨架的唯一影响就是移除 e。这启发我们找到一种自然方法，将含有一条或多条坏边的三角剖分转变为没有坏边的三角剖分：找到一系列 2-3 和 3-2 变换，使我们能够通过 3-2 变换逐一移除每条坏边。换言之，我们的思路是在 Pachner 图中搜索没有坏边的三角剖分。

如第 1.2 节所述，以定向方式进行这项搜索至关重要（而不是进行暴力搜索）。第 4.1 节和第 4.2 节概述了我们的定向搜索算法。随后，我们将在第 4.3 节和第 4.4 节讨论各种实现细节。

4.1 移除坏边的启发式方法

为了定向搜索不含坏边的三角剖分，我们需要一些启发式方法来衡量当前状态与此类三角剖分之间的距离。现在，我们来说明并定义所

使用的启发式方法。

最显而易见的衡量指标是坏边的数量；显然，搜索时避免增加这一数量似乎是合理的。由于 3-2 变换只会通过移除一条边来改变 1-骨架，我们可以看出，这种变换绝不会增加坏边的数量。然而，2-3 变换会通过创建 一条新边来改变 1-骨架，因此，如果新边恰好是坏边，坏边的数量就会增加；因此，搜索算法中的一项关键子程序是检验 2-3 变换引入的边是否为坏边，从而判断哪些 2-3 变换应当尽量避免。

仅凭这一点并没有特别大的帮助，因为有许多三角剖分的坏边数量相同。为说明一些更精细的启发式方法为何有用，我们再次指出，减少坏边数量的唯一方法是对一条坏边执行 3-2 变换。只有存在一条度为三且实际邻接三个不同四面体的坏边时，这种变换才可行。这表明，我们应当衡量每条坏边的以下数量：

定义 19。设 e 是单顶点三角剖分中的一条边。令 $d(e)$ 表示 e 的度，令 $n(e)$ 表示与 e 相交的不同的四面体数量。

- e 的**度缺陷**记为 $\delta(e)$，其定义为 $|d(e) - 3|$。
- e 的**重数缺陷**记为 $\mu(e)$，其定义为 $d(e) - n(e)$。

注意，度缺陷和重数缺陷始终都非负，并且当且仅当这两个缺陷都等于零时，我们才能对边 e 执行 3-2 变换。因此，为了减少坏边的数量，优先选择坏边的度缺陷和重数缺陷较小的三角剖分，似乎是合理的。这引出了以下定义：

定义 20 （三角剖分的复杂度）。设 $\mathcal{T}$ 是一个至少含有一条坏边的单顶点三角剖分，并令 e_1、……、e_k 表示 $\mathcal{T}$ 中的坏边。$\mathcal{T}$ 的**复杂度**定义为以下四元组

$$\left(k,\ \max_{1 \leqslant i \leqslant k} \mu(e_i),\ \max_{1 \leqslant i \leqslant k} \delta(e_i),\ |\mathcal{T}| \right).$$

我们的搜索算法尝试按照字典序尽可能降低三角剖分的复杂度。这就是为什么坏边的数量 k 会作为首要项出现：我们的首要目标是尽可能减少这个数量。如上所述，我们的次要目标是降低坏边的重数缺陷和度缺陷，希望借此能够通过 3-2 移动移除这些边。我们取所有坏边的最大缺陷（而非例如最小缺陷），因为我们的最终目标是移除所有这些边，而不只是其中一条。

优先降低重数缺陷，而不是度缺陷，这一点至关重要。回顾我们最初解决这个问题的方法，便能最好地说明原因。起初，我们的复杂度完全没有考虑重数缺陷。基于这种天真的复杂度概念，我们发现搜索很容易到达 Pachner 图中的某个区域，在那里坏边的度缺陷接近 0；然而，随后搜索便会陷入枚举大量三角剖分的状态，始终无法将度缺陷降至 0。我们最终意识到，搜索被困在 Pachner 图中的一个区域，其中坏边的重数缺陷等于 2。优先降低重数缺陷，能够促使搜索避开这类区域。

最后，在其他条件都相同的情况下，我们希望反例尽可能小。这就是为什么我们将三角剖分的大小作为复杂度的最后一项。

4.2 目标搜索算法

算法 21. 此算法需要以下输入：

- 一个含有 n 条坏边的单顶点三角剖分 $\mathcal{T}$，其中 $n \geqslant 1$。
- 一个非负整数 x （允许使用的额外坏边数量）。要搜索具有 $n-1$ 条坏边的新三角剖分：*(1)* 对于 $\mathcal{T}$ 中的每条坏边，检查是否可以对这条边执行 *3-2* 变换。如果可以执行此变换，则终止搜索，并返回执行该 *3-2* 变换后得到的三角剖分。

(2) 创建集合 $\mathcal{S}$。此外，创建优先队列 $\mathcal{Q}$，按照复杂度从低到高的顺序存储三角剖分。将 $\mathcal{T}$ 添加到 $\mathcal{S}$ 和 $\mathcal{Q}$ 中。*(3)* 当 $\mathcal{Q}$ 非空时：*(i)* 从 $\mathcal{Q}$ 中移除第一个三角剖分 $\mathcal{F}$（即复杂度最小的三角剖分），并令 m 表示 $\mathcal{F}$ 中坏边的数量。*(ii)* 对于关于边 e 的 *3-2* 变换，如果 e 是一条好边，则称此变换为可执行；对于 *2-3* 变换，令 e 表示此变换产生的新边，如果 e 是一条好边，或者 e 是一条坏边但 $m < n + x$，则称此变换为可执行。对于 $\mathcal{F}$ 上的每个可执行变换，检查集合 $\mathcal{S}$ 中是否已经包含执行此变换后得到的三角剖分 $\mathcal{G}$（模组合等价）。如果没有：

- 将 $\mathcal{G}$ 同时添加到 $\mathcal{S}$ 和 $\mathcal{Q}$中。
- 对 $\mathcal{G}$中的坏边，执行所有可能的 *3-2* 变换序列。对于通过此类变换得到的每个三角剖分$\mathcal{T}'$，如果 $\mathcal{T}'$ 尚未（在组合等价意义下）出现在集合 $\mathcal{S}$ 中，则将 $\mathcal{T}'$ 同时添加到 $\mathcal{S}$ 和 $\mathcal{Q}$中。

- 如果我们能找到由$(m - n + 1)$ 个 *3-2* 变换组成的序列，那么该序列中的最终三角剖分 $\mathcal{T}^*$ 就有 $n - 1$ 条坏边。终止并返回 $\mathcal{T}^*$。

访问 https://github.com/AlexHe98/triang-counterex 查看算法 21 的实现。我们将在第 4.3 节和第 4.4 节中讨论该实现的主要细节。

值得说明的是，为什么我们一旦找到含有$n - 1$ 条坏边的三角剖分，就立即终止搜索，而不是让搜索继续进行，直到它（希望能）找到一个完全没有坏边的三角剖分。这在一定程度上是我们早期研究第 1.1 节中猜想时留下的做法；起初，我们完全不确定（例如）是否存在没有核心边的三角剖分，因此只要能在减少核心边数量方面取得任何进展，我们都乐于接受。我们坚持提前终止搜索的一个原因是，有时我们发现，在逐步减少坏边数量的过程中，灵活地进行调整很有用。这类调整的例子包括：

- 改变我们允许算法 21 使用的额外坏边数量 x（不过，我们尚未遇到过通过这一特定调整获益的例子）；

- 改变步骤（3）中算法 21 使用的并发进程数量（我们将在第 4.3 节中对此作更详细的讨论，并在第 5.1 节和附录 B 中给出示例）；或者
- 甚至改变我们衡量三角剖分复杂度的方式（第 5.3 节给出了一个采用这种方法后取得成效的例子）。

4.3 棘手区域、并发计算与不稳定性

在第 4.1 节中，我们提到了 Pachner 图中的一些区域，其中坏边的度缺陷很低，但重数缺陷等于 2（例如，这类边自然会出现在分层实心环面的核心处）。尽管优先降低重数缺陷确实有助于搜索避开这些区域，但如果搜索还是陷入其中，想要脱身就可能相当困难：唯一的出路是执行会增加 坏边度缺陷的变换，而这完全背离了定向搜索试图优先考虑的方向。

这种现象更为普遍：局部最优的移动可能会使搜索进入 Pachner 图中的一个“棘手”区域（该区域原则上可能是无限的），在此区域中，后续所有可用的移动都无法降低三角剖分的复杂度。在上面讨论的情形中，这会导致搜索陷入枚举大量复杂度相近的三角剖分；正如我们将在第 5.1 节看到的，有些情况下，搜索还可能开始枚举大量复杂度迅速增长的三角剖分。

如果在步骤（3）中一次只处理一个三角剖分，算法 21 尤其容易陷入此类棘手区域。不过，当我们改用多个进程并发处理多个三角剖分时，搜索有时能够避开或脱离这些棘手区域。这背后可能有两个原因：

(1) 使用多个进程会使搜索比纯粹的贪心方法探索得更广；并且

(2) 由于每次运行搜索时，三角剖分插入优先队列的顺序 $\mathcal{Q}$ 可能不同，因此搜索会带有一定的随机性。

或许有更直接的方法能够实现类似的效果；我们的方法已经足够好，而且成本较低（无论是人力投入还是计算复杂性方面）。

一个缺点是，我们无法预先知道应该使用多少个并发进程。事实上，正如我们将在第5节看到的，通常存在一个“最佳点”：运行搜索能够生成具有所需数量坏边的三角剖分 $\mathcal{T}^*$，但如果大幅增加或减少进程数量，搜索就容易陷入棘手区域。

我们的解决方案是实现算法 21，使其定期输出当前处理的三角剖分的复杂度更新，并由人工检查搜索是否在有效降低复杂度方面取得了令人满意的进展。如果没有观察到这样的进展，我们就简单地重新启动搜索，并可视情况调整所用的并发进程数量。尽管需要人工干预并不理想，但对于只需成功解决一次的计算拓扑学难题（例如寻找反例），这并非闻所未闻。另一个例子是 [8] 中将素结普查扩展到最多 19 个交叉的工作。

最后需要指出的是，我们的实现有些不稳定。事实上，我们已经注意到，当使用不同数量的并发进程时，或者仅仅重新运行搜索时，搜索可能会产生不同的结果。我们还遇到过这种不稳定性的另一种表现：即使重构源代码，也可能显著改变搜索结果。我们尚未找到修改算法以使其更加稳定的方法。

4.4 其他实现细节

现在，我们概述算法 21 实现中的其他一些重要细节 .

首先，正如第 3.3 节所述，我们使用 2-3 和 3-2 变换的定制实现，从而尽量减少检查某条边是否为坏边的次数。这一点至关重要，因为就我们的用途而言，检查某条边是否为坏边通常依赖于涉及正规曲面的计算开销较大的技术：

- 要识别核心边或隧道边，我们需要删除该边的一个小正则邻域，并检验所得的三维流形是否分别为实心环面或亏格2把手体。为此，我们使用把手体识别（算法 9）。
- 为了识别塞弗特纤维边，我们依赖算法 17。

其次，由于算法 21 总是将组合等价的三角剖分视为同一类，我们实际上在集合 $\mathcal{S}$ 和优先队列 $\mathcal{Q}$ 中存储同构签名。正如 [5],所述，在探索 Pachner 图时，同构签名是标准且不可或缺的工具。

第三，当优先队列 $\mathcal{Q}$ 中包含多个复杂度相同的三角剖分时，我们按插入顺序确定这些三角剖分的优先级。

最后，在实践中，我们通常将输入变量 x 设为 0，此时算法 21 会丢弃所有坏边数量多于初始输入三角剖分的三角剖分。尽管如此，有时取 $x > 0$ （从而允许坏边数量增加）可能会有帮助；我们将在第 5.3 节稍作讨论。

5 反例

在第 5.1、5.2 和 5.3 节中，我们分别讨论猜想 1、2 和 3 特有的实验细节；尤其是，正如第 1 节所述，我们给出了这三个猜想的反例。此外，我们将在第 5.4 节证明，我们的每个反例都可以转化为一个无限反例族。

本节讨论的所有计算均在一台搭载 Intel Core i5-7200U 处理器的笔记本电脑上运行，该处理器只有两个物理核心，划分为四个逻辑处理器。因此，我们几乎所有反例都能在几分钟内找到，着实令人惊叹

的墙钟时间；即使是我们展示的最耗时的例子（见附录B中的图28），所需墙钟时间也不到34分钟。这主要是因为我们的定向搜索能够将搜索范围缩小到搜索空间中极小的一部分：对于每个成功找到反例的三维流形，我们需要枚举的三角剖分从未超过几千个。

5.1 不含核心边的三角剖分

我们首先讨论三维球面的特殊情形；请回想在这种情况下，核心边等同于非打结边。我们在第 1.2 节中看到，三维球面有 422 533 279 个含 10 个或更少四面体的单顶点三角剖分，而且这些三角剖分全都含有核心边（事实上，至少含有两条核心边）。尽管有这些看似有力的证据，三角剖分 $\mathcal{T}$ 的同构签名为

uLLvQQvLAPvPAQccdfeghhgmklnorsqsstttthsaaggggaaaaaaaanaaagb

结果发现，它是一个含 20 个四面体、不含核心边的三维球面单顶点三角剖分。

为找到这个反例，我们先任意选取了同构签名为 cMcabbgqs 的三角剖分 $\mathcal{T}_2$；这是一个复杂度为 (2,5,4,2) 的三维球面单顶点三角剖分（回想定义 20）。运行算法 21 两次后，我们得到一个不含核心边的三维球面单顶点三角剖分 $\mathcal{T}_0$，含有 22 个四面体；结果汇总于图 7。

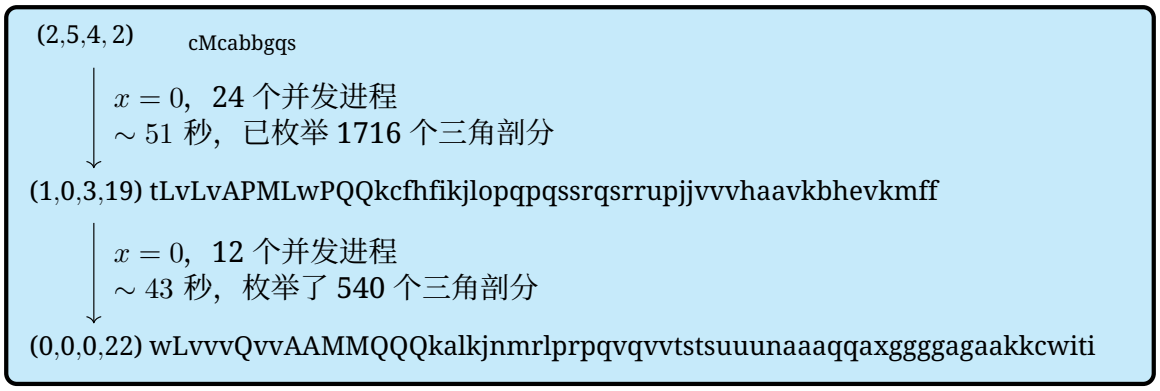


图 7：从 cMcabbgqs（三维球面）中移除核心边。

为了将 $\mathcal{T}_0$ 转化为 20个四面体的例子 $\mathcal{T}$，我们在 Pachner 图上执行了广度优先搜索，但限制为忽略所有会引入核心边的 2-3 变换。这大约花费了 367 秒的墙钟时间。我们还知道，如果不经过至少一个含有 23 个以上四面体的三角剖分，就无法进一步缩小规模（仍然忽略会引入核心边的 2-3 变换）；验证这一点还额外花费了 228 秒的墙钟时间。

值得注意的是，为了从 $\mathcal{T}_2$ 中移除一条核心边，我们尝试使用不同数量的并发进程运行算法 21，但使用约 24 个进程似乎效果最好。具体来说，我们最初尝试使用 12 个进程，但这产生了一个同构签名为

sLvAAvLAzMMQQcdceflkmjmqonprqprrhvrqnkkkksqeekocksf

以及复杂度（1，2，1，18）。该三角剖分中的核心边有 2 的重数缺陷，但度缺陷为 1；这正是我们在第 4.3 节中提到的棘手情况之一。为避免这种情况，我们需要将进程数增加到 24。

有些出乎意料的是，将进程数大幅增加到 24 以上，似乎也会对算法 21 的有效性产生不利影响。例如，使用 36 个进程时，尽管搜索有时能够移除一条核心边，但其表现似乎不那么可靠。相反，我们发现搜索往往会陷入另一种棘手区域：

- 大约 10 秒后，搜索到达一个复杂度为（2， 4， 16，15）的三角剖分。
- 大约 20 秒后，搜索到达一个复杂度为 (2，4，23，22) 的三角剖分。

• 大约 30 秒后，搜索到达一个复杂度为 (2，4，26，25) 的三角剖分。

出现这种情况时，如果允许搜索继续进行，复杂度似乎只会进一步增加。除了三维球面外，我们还在以下透镜空间中找到了没有核心边的单顶点三角剖分：$L_{6,\ 1}$、 $L_{9,\ 2}$、 $L_{11,\ 2}$、 $L_{13,\ 3}$和$L_{16,\ 3}$。详细结果见附录 A。

5.2 没有隧道边的三角剖分

交叉数不超过 7 的十四个素结的隧道数均为 1；这里我们使用它们的 Rolfsen 名称来标识这些纽结。除 5_2 纽结外，我们为所有这些纽结都找到了没有隧道边的单顶点理想三角剖分：

• 对于三叶结（即 3_1 纽结），我们任意选取了同构签名 cPcbbbadu。图 8 总结了运行算法 21 的结果。

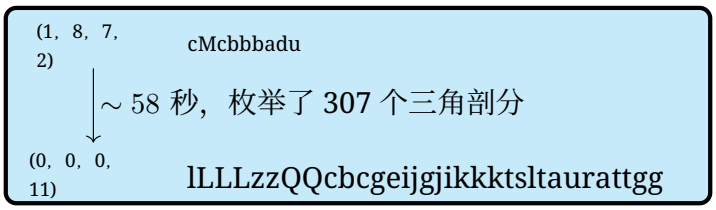


图 8：从 cPcbbbadu（三叶结）中移除隧道边； $x = 0$，8 个进程 es.

•对于八字结（即 4_1 纽结），我们任意选取了同构签名 cPcbbbiht。图 9 总结了两次运行算法 21 的结果。

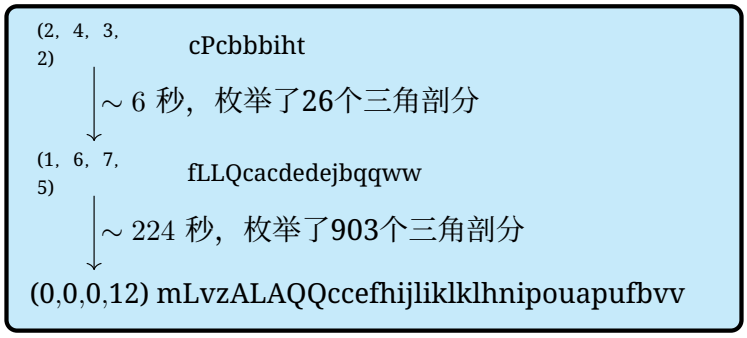


图 9：从 cPcbbbiht（八字结）中移除隧道边； $x = 0$，24 个进程。

• 对于 (5，2) 环面结（即 5_1 纽结），我们任意选取了同构签名 dLQbcccaekv。图10总结了两次运行算法 21 的结果。

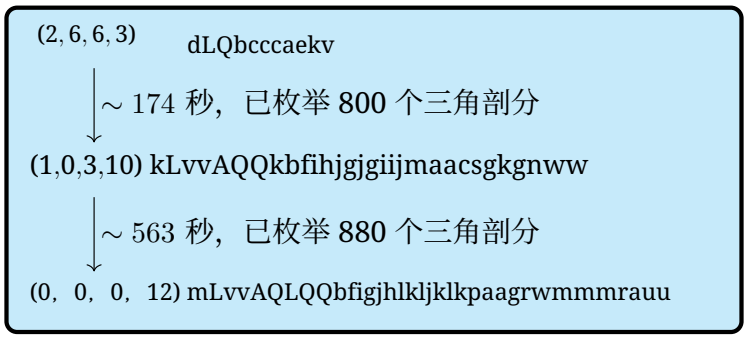


图1　0：从 dLQbcccaekv（(5,2) 环面结）中移除隧道边； $x = 0$，24 个进程。

• 我们还为所有交叉数为 6 或 7 的十个素结找到了没有隧道边的单顶点理想三角剖分。详细结果见附录B。

一个显著的观察是，尽管这里的三角剖分相对较小，而且我们枚举的三角剖分数量也相对较少，但运行时间仍显著长于第 5.1 节中的运行时间。这主要可能是因为识别亏格 2 把手体比识别实心环面更耗时。

5.3 不含塞弗特纤维边的三角剖分

令 $\mathcal{M}$ 表示小塞弗特纤维空间 $\mathcal{S}\left(\frac{1}{2}, 23, -\frac{1}{3}\right)$（回顾构造 5）。我们找到了以下 $\mathcal{M}$ 的 11-四面体单顶点三角剖分 $\mathcal{T}$，结果表明它不含塞弗特纤维边：

```
lLLLLPMQccddfjiihikkkpkrwaaacttvc.
```

为此，我们从同构签名 fLLQcaceeedjkuxkj 开始。使用 $x = 1$ 和 12 个并发进程运行算法 21 四次，得到一个不含塞弗特纤维边的 13-四面体三角剖分 $\mathcal{T}^*$；结果汇总于图 11。随后，我们使用广度优先搜索（与第 5.1 节中的方法类似），将 $\mathcal{T}^*$ 转化为 11-四面体三角剖分 $\mathcal{T}$。

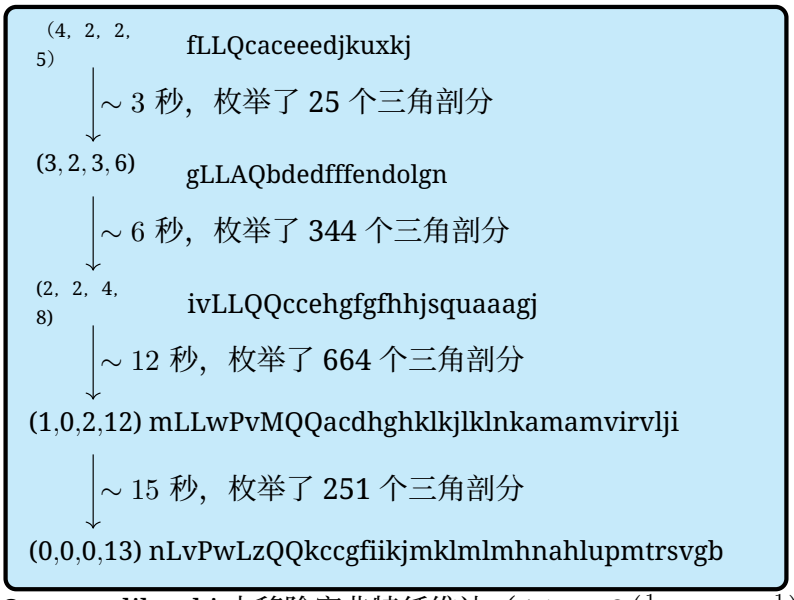


图 11：从 fLLQcaceeedjkuxkj 中移除塞弗特纤维边（$\mathcal{M} = \mathcal{S}\left(\frac{1}{2}, 23, -\frac{1}{3}\right)$）；$x = 1$，12 个进程。

之所以采用 $x = 1$ 而不是 $x = 0$，是因为早期版本的代码如此。算法 17 的早期版本在分析中纳入的正规曲面较少，因此识别好边（在此语境下，指不同伦于塞弗特纤维的边）的效果较差。当时，我们遇到了一些“死胡同”三角剖分，具体来说，每次 2-3 变换都会生成一条我们无法认证为好边的边。我们的短期解决方案是允许搜索在必要时引入额外的坏边。此后，我们强化了算法 17；对于刚才讨论的例子，采用 $x = 1$ 似乎已不再必要。我们没有改回使用 $x = 0$，只是因为没有找到这样做的充分理由。

还值得一提的是，我们遇到过这样一个例子：即使取 $x = 1$，也不足以避免“死路”（即便使用当前版本的算法 17 实现）。具体来说，这发生在同构签名为 fvPQcdecedekrsnrs、复杂度为 (6, 0, 2, 5) 的三角剖分中；它表示庞加莱 $^{\prime}$ 同调球，该空间允许塞弗特纤维化 $\mathcal{S}\left(\frac{1}{2}, 13, -\frac{4}{5}\right)$。当我们将 x 的值增至 2 时，成功到达了一个含有四条塞弗特纤维边的三角剖分，但此后未能取得进一步进展。

类似地，我们确实为另一个小塞弗特纤维空间找到了一个反例，即 $\mathcal{S}\left(\frac{1}{2}, 13, \frac{2}{3}\right)$，记作 $\mathcal{M}'$。具体来说，我们找到了以下 $\mathcal{M}'$ 的 20-四面体三角剖分 $\mathcal{T}'$，其中不含塞弗特纤维边：

```
uLLPvvLQLAPAPQccdfeknlipnomqnrtstssthsipaaliaravlkuaaxxxx。
```

为此，我们从同构签名 gLLMQacdefefjkaknkr 开始，其复杂度为 (4，2，4，6)。然而，我们无法仅通过运行四次算法 21 来移除全部四条塞弗特纤维边。相反，为了移除其中第一条和第三条边，我们需要修改搜索衡量三角剖分复杂度的方式。

为了描述这一修改，考虑 $\mathcal{S}$ 的一个单顶点三角剖分 $\mathcal{M}'$，并令 e_1、.....、e_k 表示 $\mathcal{S}$ 中所有的坏边（在此语境中，“坏”指该边是塞弗特纤维边）。我们可以设想， $\mathcal{S}$ 是在类似于算法 21 的搜索中遇到的某个中间三角剖分，因此背景中存在一个初始输入三角剖分，它有 n 条坏边；此外，还有一个非负整数 x ，使得搜索会舍弃坏边数超过 $n+x$ 条的任何三角剖分。不过，与算法 21 不同，我们暂时不考虑最终目标——移除 $\mathcal{S}$ 中的所有 k 条坏边，而是全力专注于一个中间目标：从 $\mathcal{S}$ 中移除 $k-n+1$ 条坏边（从而将坏边数减少到 $n-1$）。为此，我们按如下方式选出两条（不一定不同的）坏边：

- 首先，按重数缺陷从小到大排列坏边，并将 e_m 取为这一排列中的第 $k-n+1$ 条坏边。
- 接着，按度缺陷从小到大的顺序排列坏边，并令 e_d 为该排序中的第 $k-n+1$ 条坏边。

我们通过以下方式衡量 $\mathcal{S}$ 的复杂度，从而修改算法 21：

$$(\ k,\ \mu(e_m),\ \delta(e_d),\ |\mathcal{S}|\ )\,.$$

图 12 总结了从同构签名为 gLLMQacdefefjkaknkr 的$\mathcal{M}'$ 三角剖分中移除全部四条塞弗特纤维边后的结果；我们使用了 $x=1$ ，并在每个阶段采用 12 个并发进程。在使用修改后的复杂度而非定义 20 中引入的（未修改）复杂度的阶段，图 12 在修改后的复杂度上方列出了未修改的复杂度。我们得到一个没有塞弗特纤维边的 $\mathcal{M}'$ 的 25-四面体三角剖分，随后使用与之前相同的广度优先搜索技术，将其简化为 20-四面体三角剖分 $\mathcal{T}'$ 。

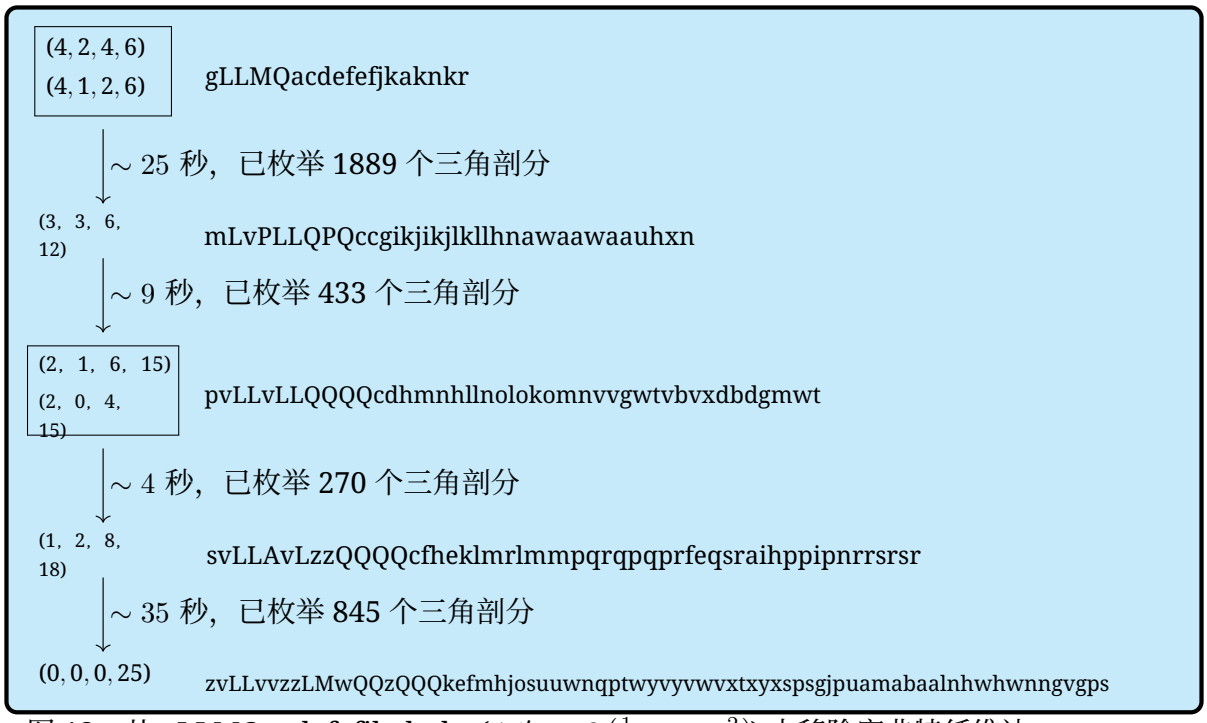


图 12：从 gLLMQacdefefjkaknkr ($\mathcal{M}'=\mathcal{S}\left(\frac{1}{2}、\ 13,\ \frac{2}{3}\right)$) 中移除塞弗特纤维边；$x=1$，12 个进程。方框标示了我们使用修改后的复杂度之处。

5.4 从一个反例到无穷多个

这里，我们证明上述每个反例都可以转化为无穷族。关键在于以下命题：

命题 22. 设 $\mathcal{M}$ 是一个闭3流形，或是一个只有一个正亏格边界分支（且没有其他边界分支）的3流形。在闭情形下，设 $\mathcal{T}$ 是 $\mathcal{M}$ 的单顶点三角剖分；在有边界情形下，设 $\mathcal{T}$ 是 $\mathcal{M}$ 的单顶点理想三角剖分。设 P 是 $\mathcal{T}$ 的边所具有的一种在环境同痕下不变的性质。若 $\mathcal{T}$ 的每条边都满足 P，则称 $\mathcal{T}$ 有趣。如果 $\mathcal{M}$ 有一个有趣的三角剖分，那么 $\mathcal{M}$ 就有无穷多个有趣的三角剖分。

证明。设 $\mathcal{T}$ 是 $\mathcal{M}$ 的一个有趣的三角剖分。固定 $\mathcal{T}$ 的一个四面体 Δ，并令 e 和 f 表示 Δ 的一对对边。我们将 Δ 替换为一个三四面体构件，从而得到 $\mathcal{M}$ 的一个新三角剖分 $\mathcal{T}'$，如图 13 所示。[13] 从 1-骨架来看，我们所做的只是引入两条新边 e' 和 f'，使得 e' 与 e 同伦，且 f' 与 f 同伦。因此，$\mathcal{T}'$ 是 $\mathcal{M}$ 的一个有趣的三角剖分。无限重复这一过程，就得到了 $\mathcal{M}$ 的所需无限族有趣三角剖分。 □

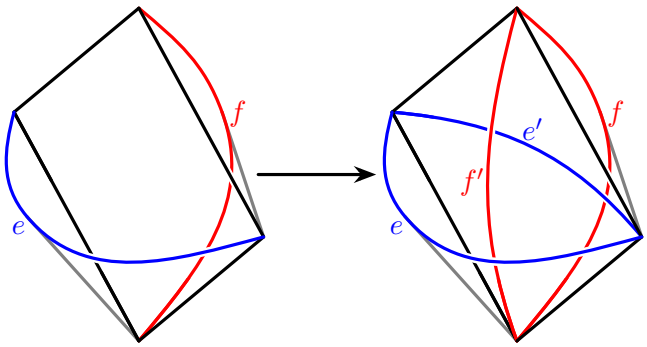


图 13：构造一个新的有趣三角剖分。

将命题 22 与第 5.1、5.2 和 5.3 节中的反例结合，立即得到以下三个定理：

定理 23。 设 $\mathcal{M}$ 是以下透镜空间之一：3-球面、$L_{6,\,1}$、$L_{9,\,2}$、$L_{11,\,2}$、$L_{13,\,3}$，或 $L_{16,\,3}$。那么 $\mathcal{M}$ 有无穷多个不含核心边的单顶点三角剖分。

定理 24。 设 K 是交叉数至多为 7 的素结，但不是 5_2 纽结。那么 K 有无穷多个不含隧道边的单顶点理想三角剖分。

定理 25。 每个小塞弗特纤维空间 $\mathcal{S}\left(\frac{1}{2}, 2, -\frac{1}{3}\right)$ 和 $\mathcal{S}\left(\frac{1}{2}, 3, \frac{2}{3}\right)$ 都有无穷多个不含塞弗特纤维边的单顶点三角剖分。

6 讨论

6.1 未解问题

尽管通过定向搜索，我们在六种不同的透镜空间中找到了没有核心边的三角剖分，但对我们尝试过的所有透镜空间而言，这种方法并非都奏效。作为参考，我们未能为以下透镜空间找到反例：$S^2 \times S^1$，$\mathbb{R}P^3$，$L_{3},1$，$L_{5},1$，$L_{7},2$，$L_{8},3$，$L_{10},3$，$L_{11},3$，$L_{12},5$，$L_{13},5$，$L_{7},1$，$L_{14},3$，$L_{15},4$，$L_{16},7$，$L_{17},5$，$L_{18},5$，$L_{19},7$，$L_{21},8$，$L_{8},1$，以及$L_{13},2$。尽管如此，我们仍怀疑此类反例确实存在。此外，目前没有充分理由相信情况并非如此，因此我们暂且提出以下猜想：

猜想 2 6. 每个透镜空间都存在无穷多个不含核心边的单顶点三角剖分 *s.*

[13] 这种修改是一个初等移动的特例，该移动称为**0-2 移动**或**新月移动** .

对于另外两个我们关注的问题，定向搜索同样并非总能成功。如第 5.2 节所述，我们未能找到一个不含隧道边的 5_2 纽结的单顶点理想三角剖分。我们也未能找到以下小塞弗特纤维空间中不含塞弗特纤维边的三角剖分：$\mathcal{S}\left(\frac{1}{2},\ 13,\ -\frac{2}{3}\right), \mathcal{S}\left(\frac{1}{2},\ 13,\ -\frac{1}{3}\right), \mathcal{S}\left(\frac{1}{2},\ 13,\ -\frac{3}{4}\right), \mathcal{S}\left(\frac{1}{2},\ 13,\ -\frac{4}{5}\right)$，以及$\mathcal{S}\left(\frac{1}{2},\ 13,\ \frac{1}{3}\right)$。尽管如此，我们仍提出以下猜想：

猜想27。 每个隧道数为一的纽结都存在无穷多个不含隧道边的单顶点理想三角剖分。

猜想28。 对于每个既不是透镜空间也不是棱柱流形的不可约小塞弗特纤维空间，都存在无穷多个不含塞弗特纤维边的单顶点三角剖分。

6.2 未来应用

诸如 2-3 和 3-2 变换这样的初等移动，除了本文中的用法之外，还有许多计算应用。最常见的两种是：

(1) 使用初等移动改进三角剖分，这往往是使指数时间计算变得可行的关键；以及 (2) 找到一系列变换，将三角剖分 $\mathcal{T}$变为另一个三角剖分 $\mathcal{T}'$，从而给出 $\mathcal{T}$和 $\mathcal{T}'$ 同胚的计算证明。

找到合适的变换序列可能很困难，因此定向搜索在这些其他情形中也可能有用。

事实上，在这些其他情形中，一个主要难点是，增加四面体数量有时不可避免 [5]:

(1) 存在许多三维球面的三角剖分 $\mathcal{T}$，若要通过 2-3 和 3-2 变换将 $\mathcal{T}$简化为最小三角剖分，就必须经过至少一个比 $\mathcal{T}$多两个四面体的中间三角剖分。还有一个图流形的三角剖分，其简化过程需要三个 额外四面体。

(2) 在最多包含 9 个四面体的最小三角剖分普查中，有许多三维流形需要额外两个四面体，才能找到一系列 2-3 和 3-2 变换，将所有成对的最小三角剖分连接起来。还有一个三维流形——透镜空间 $L_{3,1}$——连接所有最小三角剖分需要三个 额外四面体。

除了不得不增加四面体数量的情况外，有时我们也可能希望 增加这一数量，前提是这样做能在另一项指标上改善三角剖分。其中一项指标是树宽，它很重要，因为已知三维流形拓扑中的许多算法都具有关于树宽的固定参数可处理性 [10, 11, 12, 14, 15]。对于这类算法，降低树宽所带来的益处远远超过增加四面体数量造成的代价。这一点在实际中也有意义，因为正如 Husz´ar 和 Spreer [20, 的推论 20], 所指出的，最小三角剖分（即四面体数量最少的三角剖分）不一定能使树宽最小；这里我们举出两个例子，其中第一个由 Husz´ar 和 Spreer 在 [20]中给出：

• 令 $\mathcal{M}$ 表示庞加莱同调球；如第 5.3 节所述， $\mathcal{M}$ 具有塞弗特纤维化 $\mathcal{S}\left(\frac{1}{2},\ 13,\ -\frac{4}{5}\right)$。 $\mathcal{M}$的唯一最小三角剖分由同构签名 fvPQcdecedekrsnrs 给出；这是一个树宽为 4 的 5 个四面体三角剖分。然而，同构签名 hLALAkcbbeffggqqnnmxkk 给出了 $\mathcal{M}$ 的一个树宽为 2 的 7 个四面体三角剖分。

•在第 1 节中，我们提到存在一个塞弗特纤维空间——准确地说，其纤维化由 $\mathcal{S}\left(\frac{1}{2}, 13, -\frac{9}{11}\right)$给出——它唯一的最小三角剖分具有同构签名 iLLLPQcbcgffghhhtsmhgosof；这是一个含有 8 个四面体、树宽为 4 的三角剖分。不过，我们也可以使用标准的棱柱与分层构造对这个三维流形进行三角剖分；得到的同构签名为 jLAMLLQbcbdeghhiixxnxxjqisj，对应一个含有 9 个四面体、树宽为 2 的三角剖分。

这番讨论的主要启示是，有时我们可能希望或需要增加四面体的数量。如第 1 节所述，增加四面体数量会伴随着三角剖分数量的超指数级增长；即使固定一个三维流形，这种增长也至少是指数级的（其是否实际上为超指数级仍未有定论）。这意味着，穷举搜索技术很快就会遇到问题，不仅运行时间会成为障碍，而且内存管理也会成为问题（后者往往更为重要）。不过，本文的工作表明，采用恰当的启发式方法或许能够绕过这些问题；真正的挑战（目前仍未解决）是在上述情形中实际设计出这样的启发式方法。

参考文献

[1] Gennaro Amendola. 带有嵌入弧的三维流形理想三角剖分的一种演算。Math. Nachr., 278(9):975–994, 2005. doi:10.1002/mana.200310285. [6][2] David Bremner and Lars Schewe. 面数较少的凸多胞体的边图直径界。Experiment. Math., 20(3):229–237, 2011. doi:10.1080/10586458.2011.564965. [1][3] E. J. Brody. 透镜空间的拓扑分类。Ann. Math., 71(1):163–184, 1960. doi:10.2307/1969884. [8][4] Benjamin A. Burton. 为快速枚举三维流形三角剖分而检测顶点链环中的亏格。载于第36届国际符号与代数计算研讨会论文集，第59–66页。美国计算机协会，2011。doi:10.1145/1993886. 1993901. [1, 2, 3][5] Benjamin A. Burton. 闭可定向三维流形三角剖分的 Pachner 图中的化简路径。arXiv:1110.6080, 2011. [3, 5, 21, 27][6] Benjamin A. Burton. 使用 Regina 进行计算拓扑学：算法、启发式方法与实现。载于 Geometry and Topology Down Under，Contemporary Mathematics 第597卷，第195–224页。美国数学会，2013。doi:10.1090/conm/597. [2, 5, 10, 13][7] Benjamin A. Burton. 压碎三维流形三角剖分的新方法。Discrete Comput. Geom., 52:116–139, 2014. doi:10.1007/s00454-014-9572-y. [11][8] Benjamin A. Burton. 接下来的3.5亿个纽结。载于Sergio Cabello和Danny Z. Chen编，第三十六届国际计算几何研讨会（SoCG 2020），Leibniz International Proceedings in Informatics (LIPIcs) 第164卷，第25:1–25:17页。Schloss Dagstuhl– Leibniz-Zentrum f¨ur Informatik，2020。URL: https://drops.dagstuhl.de/opus/volltexte/ 2020/12183, doi:10.4230/LIPIcs.SoCG.2020.25. [21][9] Benjamin A. Burton, Ryan Budney, William Pettersson, 等。Regina：低维拓扑软件。https://regina-normal.github.io, 1999–2022. [1, 5, 10, 13, 18][10] Benjamin A. Burton and Rodney G. Downey. 三角剖分的 Courcelle 定理。组合理论杂志 A 辑，146:264–294, 2017. doi:10.1016/j.jcta.2016.10.001. [27][11] Benjamin A. Burton, Thomas Lewiner, Jo˜ao Paix˜ao, and Jonathan Spreer. 离散莫尔斯理论的参数化复杂度。ACM Trans. Math. Softw., 42(1), 2016. doi:10.1145/2738034.[27][12] Benjamin A. Burton, Cl´ement Maria, and Jonathan Spreer. Turaev-Viro 不变量的算法与复杂度。应用与计算拓扑学杂志，2:33–53, 2018. doi: 10.1007/s41468-018-0016-2. [27].[13] Benjamin A. Burton and Melih Ozlen. 一种用于平凡结识别、具有实验性多项式时间行为的快速分支算法。arXiv:1211.1079, 2012. 即将在 Math. Program. 发表。[2, 10]

[14] Benjamin A. Burton and William Pettersson. 组合拓扑中的固定参数可处理算法。载于Zhipeng Cai, Alex Zelikovsky, and Anu Bourgeois编，计算与组合，第300–311页。Springer International Publishing，2014。doi:10.1007/ 978-3-319-08783-2_26. [27][15] Benjamin A. Burton and Jonathan Spreer. 检测三角剖分中紧角结构的复杂度。载于2013年ACM-SIAM离散算法年会（SODA）论文集，第168–183页。工业与应用数学学会，2013。doi:10.1137/ 1.9781611973105.13. [27][16] Marc Culler, Nathan M. Dunfield, Matthias Goerner, and J effrey R. Weeks. SnapPy：用于研究三维流形几何与拓扑的计算机程序。可从 http://snappy. computop.org 获取（14/08/2022）。 [4][17] Wolfgang Haken. 正规面的理论。Acta Math., 105:245–375, 1961. doi:10.1007/ BF02559591. [10][18] Joel Hass, Jeffrey C. Lagarias, and Nicholas Pippenger. 纽结与链环问题的计算复杂性。J. ACM, 46(2):185–211, 1999. doi:10.1145/301970.301971. [7][19] Allen Hatcher. 基础3-流形拓扑笔记。 https://pi.math.cornell.edu/~hatcher/ 3M/3Mfds.pdf, 2007. [8, 9, 10, 12][20]Krist´of Husz´ar and Jonathan Spreer. 小树宽的三维流形三角剖分。载于Gill Barequet和Yusu Wang编，第三十五届国际计算几何研讨会（SoCG 2019），Leibniz International Proceedings in Informatics (LIPIcs) 第129卷，第44:1–44:20页。Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019。URL: http://drops. dagstuhl.de/opus/volltexte/2019/10448, doi:10.4230/LIPIcs.SoCG.2019.44. [27][21] William Jaco and J. Hyam Rubinstein. 三维流形的0-高效三角剖分。J. Differential Geom., 65:61–168, 2003. doi:10.4310/jdg/1090503053. [7, 11, 13] [22] William Jaco and Peter B. Shalen. 关于不可约且足够大的三维流形的一种新分解定理。载于R. James Milgram编，代数与几何拓扑（第2部分），《纯数学研讨会论文集》第XXXII卷，第71–84页。美国数学会，1978。doi:10.1090/pspum/032.2. [3][23] William Jaco and Jeffrey L. Tollefson. 闭三维流形完全分解的算法。Illinois J. Math., 39(3):358–406, 1995. doi:10.1215/ijm/1255986385. [8, 10, 15][24] William H. Jaco and Peter B. Shalen. 三维流形中的塞弗特纤维空间。Mem. Amer. Math. Soc., 21(220), 1979. doi:10.1090/memo/0220. [3][25] Klaus Johannson. 带边界3-流形的同伦等价，《数学讲义》第761卷。Springer-Verlag，1979。doi:10.1007/BFb0085406. [3][26] Victor Klee. 多面体上的路径。II。Pacific J. Math., 17(2):249–262, 1966. doi:10.2140/pjm. 1966.17.249. [1][27] Marc Lackenby and Saul Schleimer. 椭圆流形的识别。arXiv:2205.08802, 2022. [2,9][28] Tao Li. 在小塞弗特纤维空间中寻找竖直环面的算法。Comment. Math. Helv., 81(4):727–753, 2006. doi:10.4171/cmh/71. [9][29] Bruno Martelli and Carlo Petronio. 复杂度至多为9的三维流形。Experiment. Math., 10(2):207–236, 2001. doi:10.1080/10586458.2001.10504444. [1]

[30] Sergei V. Matveev. 算法拓扑与三维流形分类，数学中的算法与计算第9卷。Springer-Verlag，第二版，2007。doi: 10.1007/978-3-540-45899-9。 [5][31] Sergei V. Matveev. 三维流形图谱。http://www.matlas.math.csu.ru/，访问于2022年11月。 [1][32]Grisha Perelman. 里奇流的熵公式及其几何应用。arXiv:math/0211159，2002。 [3][33] Grisha Perelman. 某些三维流形上里奇流解的有限消亡时间。arXiv:math/0307245，2003。 [3][34] Grisha Perelman. 三维流形上的手术里奇流。arXiv:math/0303109，2003。 [3][35]Riccardo Piergallini. 标准多面体与脊的标准变换。载于*III Convegno Nazionale Di Topologia*：的里雅斯特，*1986*年*6*月*9*日至*12*日：论文集，Rend. Circ. Mat. Palermo (2) Suppl.第18期，第391–414页，1988。doi:11581/244540。 [5][36]J. Hyam Rubinstein. 识别三维球面的算法。载于国际数学家大会论文集，第601–611页。Birkh¨auser，1995。doi:10.1007/978-3-0348-9078-6_54。 [13][37] J. Hyam Rubinstein、Henry Segerman和Stephan Tillmann. 遍历三维流形三角剖分与脊。*Enseign. Math.*，65(1/2)：155–206，2019。doi:10.4171/LEM/65-1/2-5。[6][38] Francisco Santos. Hirsch猜想的一个反例。*Ann. of Math.*，176(1)：383–412，2012。doi:10.4007/annals.2012.176.1.7。 [1][39] Peter Scott. 三维流形的几何结构。*Bull. London Math. Soc.*，15(5)：401–487，1983。doi:10.1112/blms/15.5.401。 [8, 9][40] Abigail Thompson. 薄位置与 S^3的识别问题。*Math. Res. Lett.*，1(5)：613–630，1994。doi:10.4310/MRL.1994.v1.n5.a9。 [13][41] William P. Thurston. 三维几何与拓扑。普林斯顿大学出版社，1997。doi:10.1515/9781400865321。 [5][42] Jeff Weeks. 纽结理论中的双曲结构计算。载于William Menasco和Morwen Thistlethwaite编，《纽结理论手册》，第10章，第461–480页。Elsevier Science，2005。doi:10.1016/B978-044451452-3/50011-3。 [4]

更多没有核心边的三角剖分

在第 5.1 节中，我们提到，除了为三维球面找到没有核心边的三角剖分外，我们还为以下透镜空间找到了这样的三角剖分：$L_{6,\ 1}$、 $L_{9,\ 2}$、 $L_{11,\ 2}$、 $L_{13,\ 3}$，以及$L_{16,\ 3}$。对于这另外五种情况，只需运行算法 21，并使用 $x = 0$ 和 12 个并发进程即可；结果如下：

• 对于$L_{6,\ 1}$，我们以同构签名 dLQbcbchhjs 开始。图 14 总结了运行算法 21 的结果。随后，我们使用广度优先搜索进行简化，得到一个具有同构签名的 12 个四面体的示例

mLvwLLQQQadjljgilkilkknaqlabamaqacv.

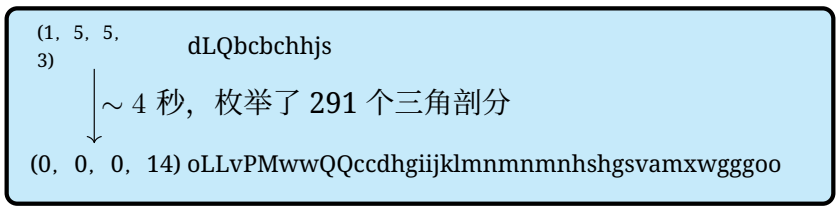


图 14：从 dLQbcbchhjs ($L_{6,\ 1}$) 中移除核心边； $x = 0$，12 个进程。

• 对于$L_{9,\ 2}$，我们从同构签名 dLQbcbchhjw 开始。图15总结了两次运行算法 21 的结果。随后，我们使用广度优先搜索，将其简化为一个包含 13 个四面体的例子，其同构签名为

nvLLMMAwQkcdghghjikmllmmnkpmigoriulkkn.

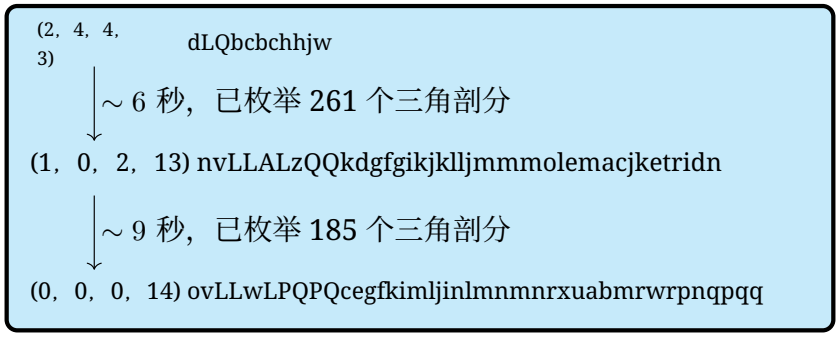


图15：从 dLQbcbchhjw 中移除核心边（$L_{9,\ 2}$）； $x = 0$，12 个进程。

• 对于$L_{11,2}$，我们从同构签名 eLAkbcbddhhjhr 开始。图16总结了两次运行算法 21 的结果。随后，我们使用广度优先搜索进行化简，得到一个具有同构签名的 16四面体示例

qLvwLPLvQQQkadhgkiknopknopponjajcxacsvvocvvvvc.

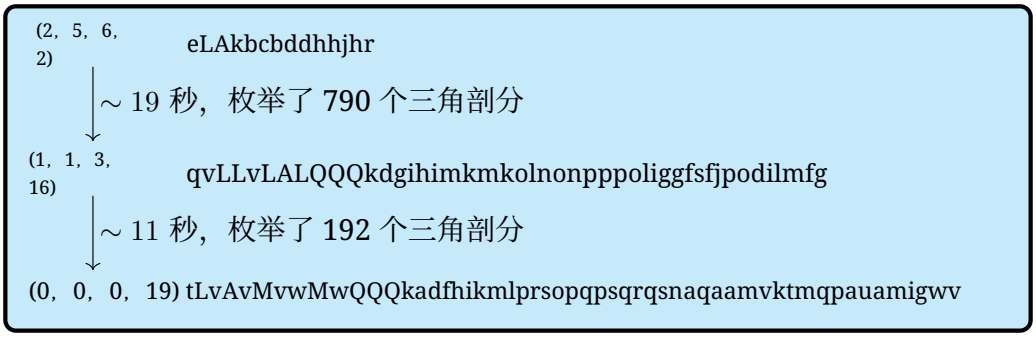


图16：从 eLAkbcbddhhjhr 中移除核心边（$L_{11,\ 2}$）； $x = 0$，12 个进程。

• 对于$L_{13,\ 3}$，我们从同构签名 eLAkbcbddhhjqn 开始。图17总结了两次运行算法 21 的结果。随后，我们使用广度优先搜索进行简化，最终得到一个具有以下同构签名的17四面体示例

rvLLALwwLQQQccdgfkhkoqpqpononpqnkpisnsaaaannngngns.

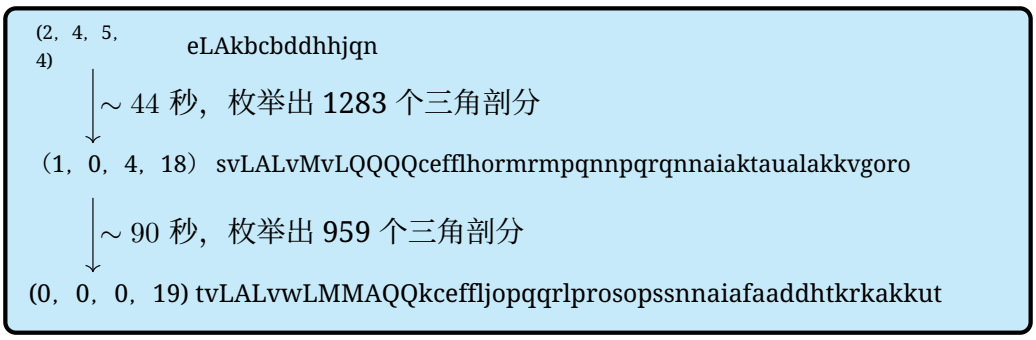


图17：从 eLAkbcbddhhjqn 中移除核心边（$L_{13,\ 3}$）； $x = 0$，12 个进程。

• 对于 $L_{16,\ 3}$，我们从同构签名 fLAMcbcbdeehhjhxj 开始。图 18总结了两次运行算法 21 的结果。随后，我们使用广度优先搜索进行简化，最终得到一个具有以下同构签名的20个四面体的例子

uLvwvAvLzQPQQQcadigooqmoslrprtsprtstnaaoqovfctkfkfqlkfxta。

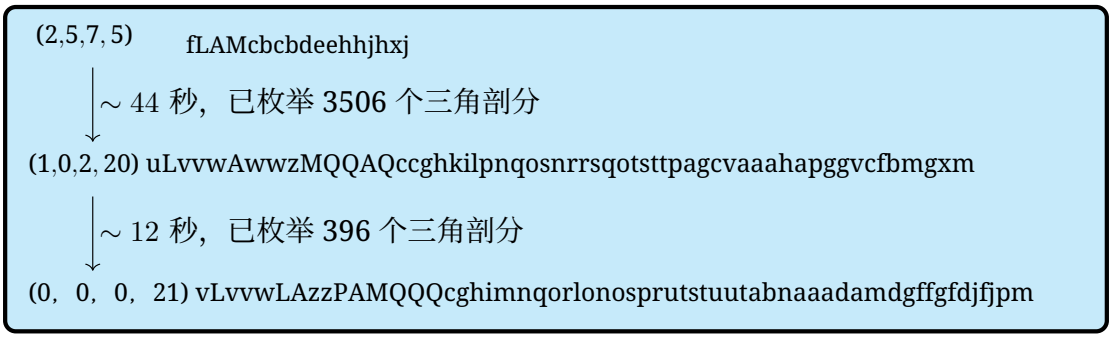


图 18：从 fLAMcbcbdeehhjhxj（$L_{16,\ 3}$）中移除核心边； $x = 0$，12 个进程。

B 更多没有隧道边的三角剖分

在第 5.2 节中，我们提到，对于交叉数不超过 7 的素结，除一个以外，我们都找到了没有隧道边的三角剖分，但没有给出交叉数为 6 和 7 的情况的详细结果。我们在此展示这些补充结果：

•对于 6_1 纽结，我们以同构签名 gLLMQccefeffdfeqldg 为起点。图 19 总结了四次运行算法 21 的结果。

• 对于 6_2 纽结，我们从同构签名 fLLQcbeddeeedgfid 开始。图 20 总结了运行算法 21 四次的结果。

• 对于 6_3 纽结，我们从同构签名 eLPkbcddddcwjb 开始。图 21 总结了运行算法 21 三次的结果。

• 对于 (7，2) 环面结（即 7_1 纽结），我们从同构签名

eLAkbccddaekln。

图 22 总结了运行算法 21 两次的结果。

• 对于 7_2 纽结，我们从同构签名 eLAkbccddmejln 开始。图 23 总结了运行算法 21 三次的结果。

• 对于 7_3 纽结，我们从同构签名 fLLQcbeddeeddgfua 开始。图 24 总结了四次运行算法 21 的结果。

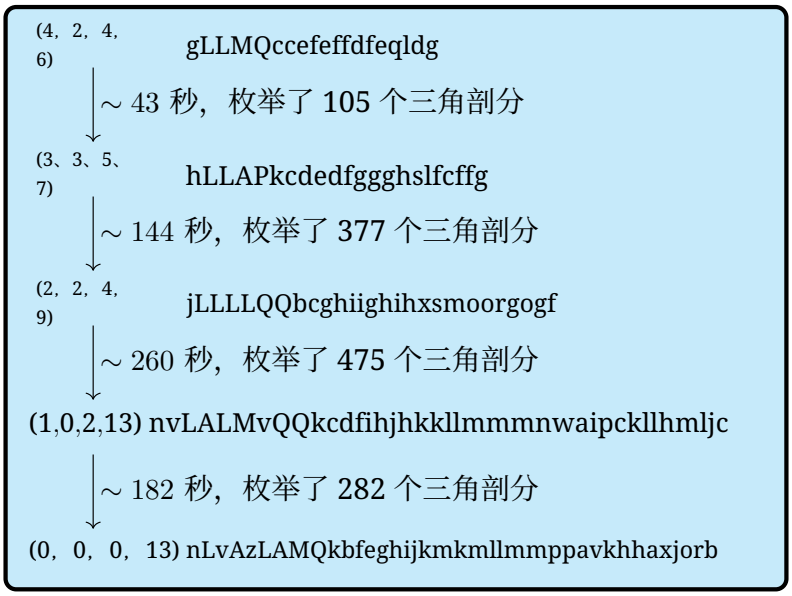


图 19：从 gLLMQccefeffdfeqldg（6_1 纽结）中移除隧道边； $x = 0$，12 个过程。



图 20：从 fLLQcbeddeeedgfid（6_2 纽结）中移除隧道边； $x = 0$，12 个进程。

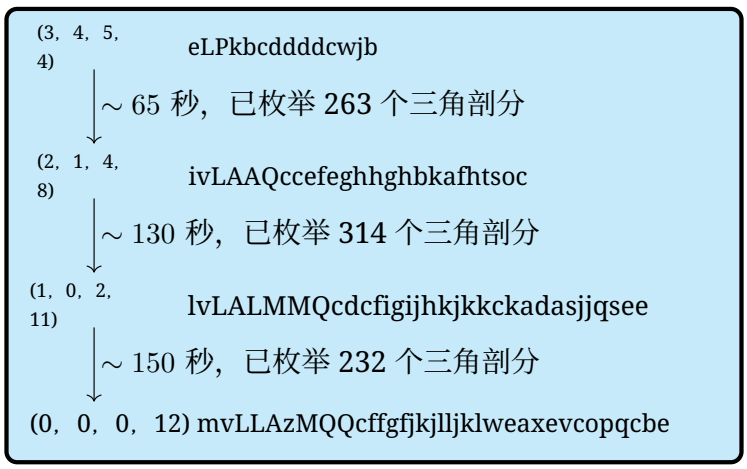


图 21：从 eLPkbcddddcwjb（6_3 纽结）中移除隧道边； $x = 0$，12 个进程。

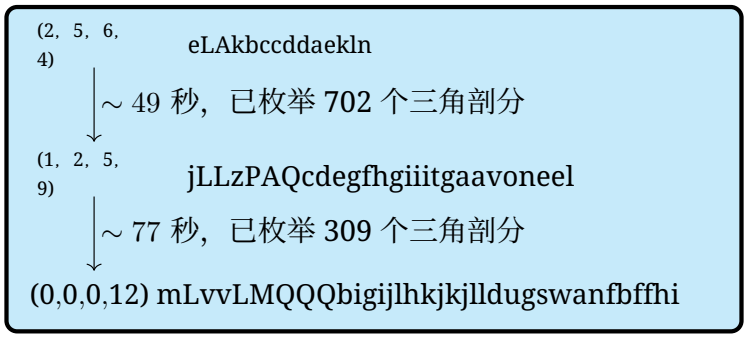
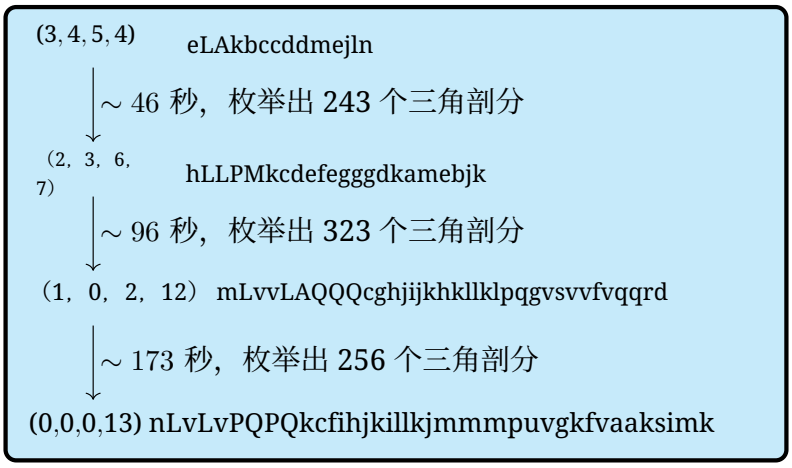


图 22：从 eLAkbccddaekln（((7,2) 环面结)）中移除隧道边； $x = 0$，12 个进程。



Fig图 23：从 eLAkbccddmejln（7_2 纽结）中移除隧道边； $x = 0$，12 个进程 s.



图 24：从 fLLQcbeddeeddgfua（7_3 纽结）中移除隧道边； $x = 0$，12 个进程。

• 对于 7_4 纽结，我们从同构签名 gLLAQbdedfffdwmfarv 开始，其复杂度为 (4,2,4,6)。为了将隧道边的数量减少到 3，我们需要使用 24 个并发进程运行算法 21（12 个进程还不够）。之后，使用 12 个并发进程运行算法 21 三次就足够了。结果汇总于图 25。

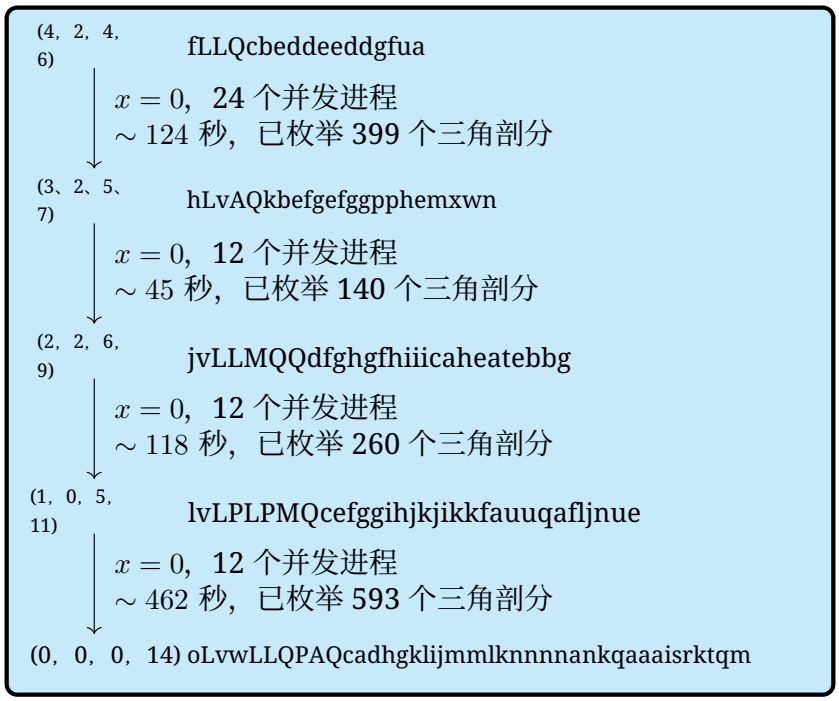


图 25：从 gLLAQbdedfffdwmfarv（7_4 纽结）中移除隧道边。

• 对于 7_5 纽结，我们从同构签名 hLLAMkbedefgggtlftavkb 开始。图 26 总结了运行算法 21 五次的结果。
• 对于 7_6 纽结，我们从同构签名 iLLvQQccdeggfhhhdoldgrgnk 开始。图 27 总结了运行算法 21 四次的结果。
• 对于 7_7 纽结，我们从同构签名 iLLPzQcbcffghghhmomtdirhx 开始，其复杂度为 (4,2,4,8)。要将隧道边的数量减少到 3，我们需要以 24 个并发进程运行算法 21（12 个进程不够）。之后，以 12 个并发进程运行算法 21 三次便足够了。结果总结于图28。

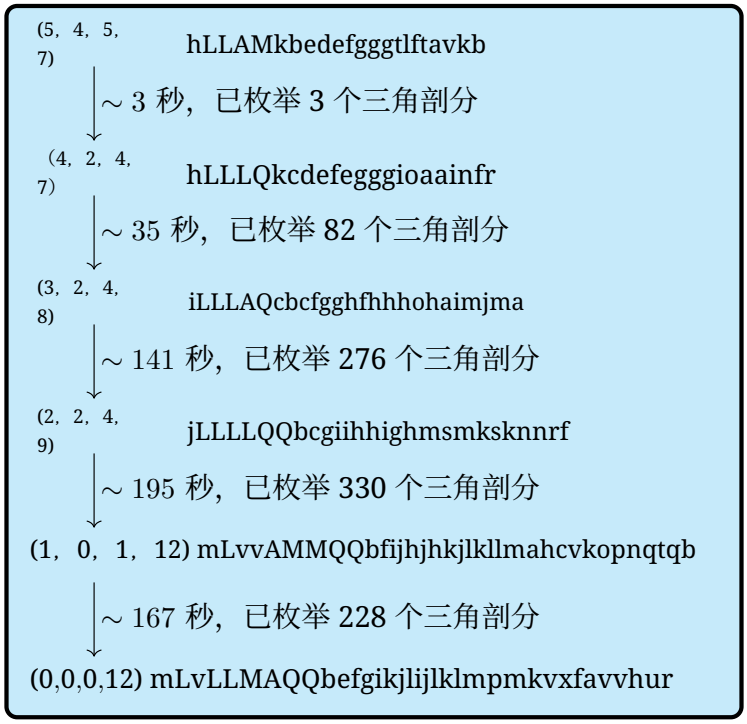


图 26：从 hLLAMkbedefgggtlftavkb（7_5 纽结）中移除隧道边； $x = 0$，12 个进程。

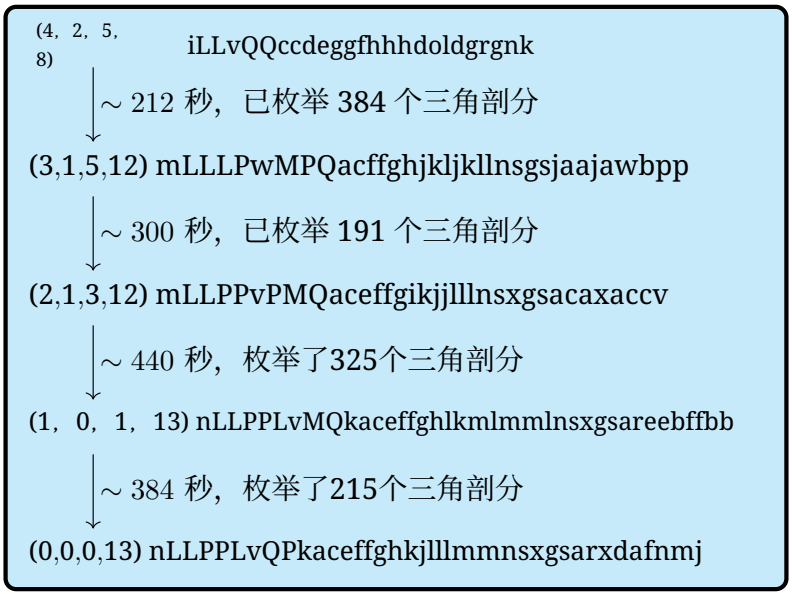


图 27：从 iLLvQQccdeggfhhhdoldgrgnk（7_6 纽结）中移除隧道边； $x = 0$，12 个进程。

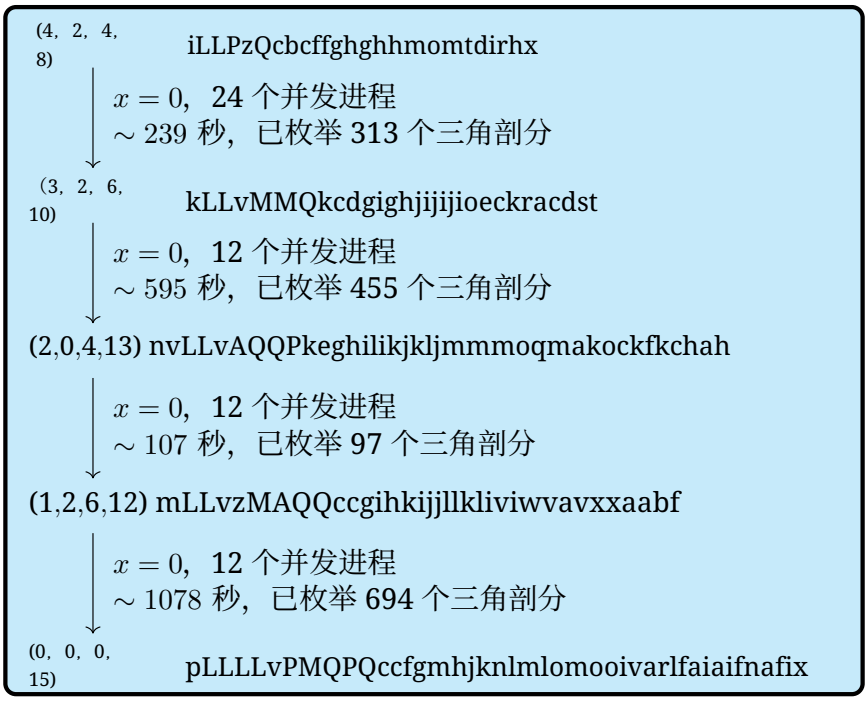


图28：从 iLLPzQcbcffghghhmomtdirhx（7_7 纽结）中移除隧道边。