arXiv:2602.02642v1 [math.GT] 2 Feb 2026

纤维结的网格图

Paul Leon Itzlinger

摘要

网格图是三维球面中纽结的一种特殊表示，用于定义纽结 Floer 同调的组合版本。Paolo Ghiggini 和 Yi Ni 证明了纽结 Floer 同调能够检测纤维结 [Ghi08; Ni07]。他们的结果特别表明，只有对于纤维结，才存在具有唯一网格状态且其 Alexander 分次为极大的网格图。每个纤维结是否都存在这样的图，仍是一个未解决的问题 [OSS15, 未解决问题 17.1.5]。

在这里，我们研究纤维结是否存在这样的特殊网格图。我们开发了一种高效方法，用于判断给定网格图是否满足更严格的条件：存在唯一一个达到 Alexander 函数上界的网格状态。通过将此方法实现为 Python 软件包，我们为 5397 个交叉数为 ≤ 13 的纤维化素纽结中的 5385 个找到了合适的网格图。

致谢

我非常感谢我的硕士学位论文导师 Vera Vértesi 教授，她将我引入这一课题，其持续的支持在整个项目过程中弥足珍贵。

引言

纽结 Floer 同调是三维流形中链环的一个不变量，由 Ozsváth、Szabó 和 Rasmussen 发现 [OS04b; Ras03]。该课题基于 Heegaard-Floer 同调，这是闭可定向三维流形的一个不变量，使用了辛几何中的工具。纽结 Floer 同调依赖于对伪全纯圆盘的计数。Sarkar 和 Wang 发现了一种在特殊情形下对这些圆盘进行计数的组合方法 [SW10]。基于这些想法，Manolescu、Ozsváth、Sarkar 和 Thurston 发展出三维球面中纽结的纽结 Floer 同调的组合描述 S^3 [Man+07; MOS09]，并由此创立了一种针对 S^3中纽结的独立的纯组合同调理论，称为格点同调 [OSS15]。

对于 S^3中的纽结 K，简单阻塞网格同调 $\widehat{\mathrm{GH}}(K)$与纽结 Floer 同调 $\widehat{\mathrm{HFK}}(K)$ [MOS09]之间存在双分次同构。纽结 Floer 同调是研究纽结拓扑性质的有力工具，可用于计算 Seifert 属，并判定纽结是否为纤维化的。事实上，Ozsváth 与 Szabó 证明，Seifert 属等于使$\widehat{\mathrm{HFK}}_*(K,s) = \bigoplus_d \widehat{\mathrm{HFK}}_d(K,s)$ 非零的极大整数 s [OS04a]。Ghiggini 与 Ni 证明，Seifert 属为 g 的纽结 K 是纤维化的，当且仅当 $\dim \widehat{\mathrm{HFK}}_*(K,g) = 1$ [Ni07; Ghi08]。

由这些结果可得出如下命题。它采用格点同调的形式体系；精确定义将在下文的记号与预备知识一节中回顾。

命题 (I) 若存在表示纽结 K 的网格图 $\mathbb{G}$，使得存在唯一的网格状态 $x \in S(\mathbb{G})$，其 Alexander 分次为极大，则 K 是纤维化的，且具有 Seifert 属 $A(x)$。

证明。 我们采用 [OSS15，注记 4.6.13] 中简单阻塞网格链复形 $\widehat{\mathrm{GC}}(\mathbb{G})$ 的表示。设 n 为 $\mathbb{G}$ 的网格大小，并设 $A(x) = s$。回顾微分 $\hat{\partial}_{\mathbb{X},O_n} : \widehat{\mathrm{GC}}(\mathbb{G}) \to \widehat{\mathrm{GC}}(\mathbb{G})$ 在 x 上的作用

$$\hat{\partial}_{\mathbb{X},O_n}(x) = \sum_{y \in S(\mathbb{G})} \sum_{\{r \in \mathrm{Rect}^\circ(x,y) : r \cap \mathbb{X} = \emptyset, O_n(r) = 0\}} V_1^{O_1(r)} \cdots V_{n-1}^{O_{n-1}(r)} \cdot y.$$

该微分保持 Alexander 分次，因此 $\hat{\partial}_{\mathbb{X},O_n}(x)$ 的一个直和项 $V_1^{O_1(r)} \cdots V_{n-1}^{O_{n-1}(r)} \cdot y$ 满足

$$A(x) = A(V_1^{O_1(r)} \cdots V_{n-1}^{O_{n-1}(r)} \cdot y) = A(y) - \#(r \cap \mathbb{O}).$$

这意味着 $A(y) \geq A(x)$，根据假设，除了 x 本身之外，没有其他网格状态满足这一条件。由于不可能存在一个将 x 与自身相连的矩形，因此

可知 $\hat{\partial}_{\mathbb{X},\,O_n}(x)=0$。 $\widehat{\mathrm{GC}}_*(\mathbb{G},s)=\bigoplus_d \widehat{\mathrm{GC}}_d(\mathbb{G},s)$ 的基由形如 $V_1^{k_1}\cdots V_{n-1}^{k_{n-1}}\cdot y$ 且满足 $A(V_1^{k_1}\cdots V_{n-1}^{k_{n-1}}\cdot y)=s$的元素构成。由此得出，$\widehat{\mathrm{GC}}_*(\mathbb{G},s)$ 由单个元素 x 生成，并且微分 $\hat{\partial}_{\mathbb{X},\,O_n}$ 限制在 $\widehat{\mathrm{GC}}_*(\mathbb{G},s)$上时为零映射。对于 $\widehat{\mathrm{GH}}_*(K,s)=\bigoplus_d \widehat{\mathrm{GH}}_d(K,s)$，可知

$$\widehat{\mathrm{GH}}_*(K,s)=\frac{\ker \hat{\partial}_{\mathbb{X},O_n}\cap \widehat{\mathrm{GC}}_*(\mathbb{G},s)}{\operatorname{im}\hat{\partial}_{\mathbb{X},O_n}\cap \widehat{\mathrm{GC}}_*(\mathbb{G},s)}=\widehat{\mathrm{GC}}_*(\mathbb{G},s).$$

特别地，对于所有 $s'>s$， $\dim(\widehat{\mathrm{GH}}_*(K,s))=1$ 且 $\widehat{\mathrm{GH}}_*(K,s')=0$。由于简单阻塞网格同调与纽结 Floer 同调同构，我们由前述定理得出， K 是纤维化的，且 s 是 K 的 Seifert 属。

□

为简便起见，我们称这类网格状态为极大且唯一。自然会想要探讨命题 I 的逆命题。

问题（i）对于每个纤维化纽结，是否都存在一个具有极大且唯一网格状态的网格图?

为了分析这个问题，我们首先在第 1 节中为 Alexander 函数建立一个上界。随后，我们在第 2 节中提出一种高效方法，用于检验给定的网格图是否存在一个网格状态，使 Alexander 函数取到该上界，并且在此意义下是唯一的。利用这些思路，我们在第 3 节中编写了一个简单的 Python 程序，用来搜索纤维结的合适网格图。最后，我们在第 4 节中作出一些评论，并列出所有尚未解决的、交叉数为 ≤ 13 的纤维化素纽结。

对于一个大小为 n的网格图，共有 $n!$ 个不同的网格状态。因此，使用蛮力方法寻找特定网格状态，需要计算并比较 $n!$个分次。我们在此提出的方法至多需要在一个矩阵上进行 $2n$ 次最小值检测，以判断大小为 n 的网格图是否存在一个不仅是极大且唯一，而且还能使 Alexander 函数达到上界的网格状态。如果该网格状态存在，此方法还会返回这个唯一的网格状态。

记号与预备知识

本文探讨 [OSS15 第 17] 章提出的开放问题 17.1.5。具体而言，我们研究 [OSS15 定义 4.6 中所定义的简化格点同调 $\widehat{\mathrm{GH}}(\mathbb{G})$ 的构造。构造所需的最重要概念列于下文。

• 平面网格图 [OSS15，定义 3.1，以及环面网格图[OSS15，第 3 节2]。我们用 $\mathbb{G}$ 表示环面网格图，并简称其为网格图。我们将环面网格图的平面表示称为平面实现。

• 网格图的网格状态集合 $S(\mathbb{G})$ [OSS15, 定义 4.1。1] 网格状态用于定义简单阻塞网格同调的链复形的生成元 $\widehat{\mathrm{GC}}(\mathbb{G})$。

• 元素 x、$y \in S(\mathbb{G})$之间的矩形集合 $\mathrm{Rect}(x,y)$ [OSS15, 第 4 页。2] 定义微分 $\hat{\partial}_{\mathbb{X},\ O_n}$ 的作用需要用到矩形，参见 [OSS15, 注 4.6。13]。

• 诱导 $\widehat{\mathrm{GH}}(\mathbb{G}) = \bigoplus_{d,\ s\in\mathbb{Z}} \widehat{\mathrm{GH}}_d(\mathbb{G},s)$的双分次结构的亚历山大函数和马斯洛夫函数 $A : S(\mathbb{G}) \to \mathbb{Z}$ 和 $M : S(\mathbb{G}) \to \mathbb{Z}$，参见 [OSS15, 第 4 节。3]。

全文中，所有纽结均假定定义在三维球面 S^3中。

1 Alexander 函数上界

为了定义 Alexander 函数的上界，我们回顾其以绕数表述的定义，参见 [OSS15 第 4 节7]。

定义 1.1。设 $\mathbb{G}$ 是纽结 K的一个网格图，p 是 $\mathbb{G}$ 上的一点，且不位于纽结投影上。绕数 $w(p)$ 是一个整数，用于描述纽结投影绕 p 逆时针绕行的次数。选取 $\mathbb{G}$的一个平面实现后，可以通过从 p 引出一条射线 γ，并计算 γ 与纽结投影的交点数，来求出 p 处的绕数。纽结投影逆时针经过的交点使 $w(p)$ 增加 1，顺时针经过的交点则使 $w(p)$ 减少 1，参见图 1。

定义 1.2. 设 $\mathbb{G}$ 是一个大小为 n 的网格图，则网格上有 $2n$ 个小方格，分别标记为 X 或 O。将所有 $2n$ 个方格的四个角点全部计入，可得到网格上由 $8n$ 个格点组成的元组 $(p_1,\dots,p_{8n})$，其中某些点可能被重复计数。网格状态 $x = \{x_1,\dots,x_n\}$ 上的 *Alexander* 函数由下式给出

$$A(x) = -\sum_i w(x_i) + \frac{1}{8}\sum_{j=1}^{8n} w(p_j) - \left(\frac{n-1}{2}\right).$$

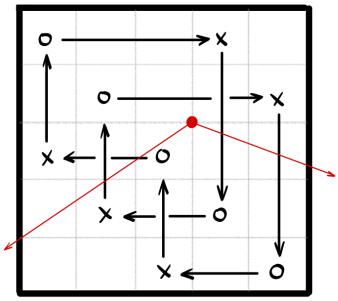

图 1. 对于网格图的这一平面实现，红点 p 处的绕数为 -2。要看出这一点，可以从 p 出发任选一条射线，并按照定义 1.1 中的说明，计算它与纽结投影的带符号交点数。图中画出了两条可能的射线，它们都以顺时针方向与纽结投影相交两次。

定义 1.3. 设 $\mathbb{G}$ 是一个大小为 n 的网格图，且 $x = \{x_1, \dots, x_n\} \in S(\mathbb{G})$。我们将 绕数函数 $A' : S(\mathbb{G}) \to \mathbb{Z}$ 定义为

$$A'(x) = -\sum_{i=1}^{n} w(x_i).$$

$A(x)$ 与 $A'(x)$ 的区别仅在于项 $\frac{1}{8}\sum_{j=1}^{8n} w(p_j) - (n-1)/2$，该项与 x无关。

定义 1.4. 网格图 $\mathbb{G}$的一个平面实现的绕数矩阵是一个 $n \times n$ 矩阵，其元素是 $\mathbb{G}$的格点处的绕数，参见图 2。

固定网格图的一个平面实现后，我们可以从左到右为垂直网格线编号，并从上到下为水平网格线编号。网格状态 x 可以由置换 σ_x 指定，使得 x 中位于第 i条垂直线上的点，处于与第 $\sigma_x(i)$条水平线的交点处。使用这种记法可知，平面实现中第 i条垂直线上的 x 点的绕数，因而是相应绕数矩阵第 i列的第 $\sigma_x(i)$个元素；也就是说，绕数矩阵的每一行和每一列都恰好包含 x的一个绕数。

定义 1.5. 给定一个网格图 $\mathbb{G}$，设 M 为 $\mathbb{G}$ 的某个平面实现的绕数矩阵。记 r_i 为 M 第 i 行的最小值，记 c_i 为 M 第 i 列的最小值。我们称 $r = -\sum_i r_i$ 为 $\mathbb{G}$ 的行数，并且

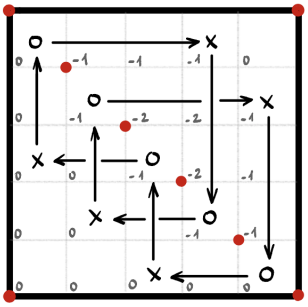

图 2. 三叶结网格图的一个平面实现，其中网格状态以红点表示。绕数矩阵的各项以灰色标出。

$c = -\sum_i c_i$，其 $\mathbb{G}$。

同一网格图的任意两个绕数矩阵之间只相差若干次行和列的循环置换，因此 r 和 c 与所选的平面实现无关。因此，行数和列数给出了以下上界。

$$A'(x) \le \min\{r, c\}$$
$$A(x) \le \min\{r, c\} + \frac{1}{8}\sum_{j=1}^{8n} w(p_j) - (n-1)/2.$$

定义 1.6. 如果网格状态 x 实现了 Alexander 函数的上界，即满足 $A'(x) = \min\{r, c\}$，我们称其为完美。如果 x 满足 $A'(x) = r$，则称其为行完美；如果 $A'(x) = c$，则称其为列完美。

行数和列数使我们能够快速检验某些网格状态是否为极大状态。此外，在下一节的命题 2.4 中，我们将展示一种高效的方法，用于分析网格图是否具有唯一完美网格状态。

例 1.7. 图 2 中的网格状态是该网格图唯一的完美网格状态。事实上，对于图中的绕数矩阵，我们有 $r = c = 6$。紫色圆点所表示的网格状态 x 达到了这个上界，因为 $A'(x) = 6$。要看出其唯一性，请注意，该网格图的任何极大网格状态，其在绕数矩阵 M 第三行中对应的元素都必须位于同一位置，因为这是该行中元素取最小值的唯一位置。

这就迫使第二行和第四行中对应的元素也必须位于给定网格状态中的相同位置。照此继续，第一行以及最后一行中的元素也都随之确定。

2 寻找唯一完美网格状态

在这里，我们提出一种高效的方法，用于分析一个网格图是否具有唯一的完美网格状态。我们从一个技术性定义开始。

定义 2.1。设 $\mathbb{G}$ 为大小为 n的网格图，设 M 为 $\mathbb{G}$ 的一个固定平面实现的绕数矩阵，其列为 c_1、…、c_n。设 x 为由置换 σ描述的网格状态。若存在 (M,σ) 具有一个回路，即存在 M 的若干列 $c_{i_1}\dots c_{i_k}$，并且对每一列都有一个索引 $j_{i_1}\dots j_{i_k}$，使得 $j_{i_r}\neq\sigma(i_r)$和 $M_{j_{i_r},\,i_r}=M_{\sigma(i_r),\,i_r}$ 对所有 $1\le r\le k$均成立。此外，我们要求矩阵索引序列

$$(\sigma(i_1),i_1),(j_{i_1},i_1),(\sigma(i_2),i_2),(j_{i_2},i_2),\dots,(\sigma(i_k),i_k),(j_{i_k},i_k)$$

描述 M中的一个回路，即对所有 $1\le r\le k-1$ 和 $j_{i_k}=\sigma(i_1)$都有 $j_{i_r}=\sigma(i_{r+1})$。

例 2.2。下面的 5×5 矩阵 M 对于置换$\sigma=(1,3,4,2,5)$具有一个回路。前三列包含合适的索引 j_1、j_2、j_3。对于$1\le i\le3$，元素$M_{\sigma(i),\,i}$为红色，而$M_{j_i,\,i}$为蓝色。

$$\begin{pmatrix} -1 & 0 & -1 & 2 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ -1 & -2 & 1 & 1 & -2 \\ -1 & -2 & -1 & 3 & -2 \\ 1 & -1 & 3 & 1 & -3 \end{pmatrix}$$

如下引理所示，回路这一概念很有用。

引理 2.3。设 $\mathbb{G}$ 是一个大小为 n 的网格图，且 M 是某个固定平面实现的绕数矩阵。设 $x\in S(\mathbb{G})$ 由置换 σ 描述。如果 (M,σ) 含有一个回路，则存在另一个网格状态 $y\neq x$，使得 $A(y)=A(x)$。

证明。设 c_{i_1}、…、c_{i_k} 是 M 中索引为 j_{i_1}、…、j_{i_k} 的列，它们为 (M,σ) 构成了回路。为简化记号，假设 $i_1=1$、…、$i_k=k$。于是我们可以定义序列

$$\tau=(j_1,\dots,j_k,\sigma(k+1),\dots,\sigma(n)).$$

根据构造，τ 是一个不同于 σ的置换。事实上，回路条件意味着 j_1、…、j_k 都互不相同，并且对于 $1\le r\le k$，$j_r\neq\sigma(r')$ 和

$r' \in \{k+1, \ldots, n\}$。令 y 为 $\mathbb{G}$ 的网格状态，其网格状态置换为 τ。于是，由假设 $M_{j_r,\ r} = M_{\sigma(r),\ r}$可得结论 $A(y) = A(x)$。 □

现在，我们可以表述并证明一个关于唯一完美网格状态存在性的重要命题。

命题 2.4. 设 $\mathbb{G}$ 是一个具有唯一列完美网格状态的网格图。设 M 是 $\mathbb{G}$ 的一个固定平面实现所对应的绕数矩阵。则 M 中存在一列，其最小元素唯一。对于行完美网格状态，也有类似的结论。

证明。 我们针对列完美网格状态给出证明；行完美的情形类似。设 n 是 $\mathbb{G}$ 的网格大小，并反设 M 的每一列都有多个在该列中取最小值的元素。如果该网格图不允许列完美网格状态，则结论显然成立；否则，存在一个置换 σ，它表示一个列完美网格状态 x。此外，对于 M 的所有列 $1 \le i \le n$，都存在一个索引 $j_i \neq \sigma(i)$，使得 $M_{j_i,\ i} = M_{\sigma(i),\ i}$。如果我们能构造一个不同于 σ 的置换 τ，使得对每个索引 i 都有 $\tau(i) \in \{\sigma(i), j_i\}$，则结论成立。这可以通过算法 1 实现。

算法可以如下进行可视化。首先，将x 中对应于 $\sigma(1)$ 的点，通过一条垂直线连接到网格图中对应于 j_1的点。然后，我们将该点连接到同一条水平线上 x 中唯一的点，即连接到由 $\sigma(t)$ 表示的点，其中 $1 \le t \le n$、$t \neq 1$取某些值；接着，再通过一条垂直线将该点连接到对应于 j_t的点。如此继续下去，可能出现两种情况。如果这条分段线性曲线两次经过 x 中的某个点，就会形成一个回路，该回路确定了一个新的列完美网格状态，具体如引理 2.3 所述。否则，σ 的修改版本在某一步再次成为一个与原置换不同的置换，参见图 3。

□

前面的命题可以扩展为一种高效方法，用于确定网格图 $\mathbb{G}$ 是否具有唯一完美网格状态。固定 $\mathbb{G}$ 的一个平面实现，其绕数矩阵为 M，并计算 r 和 c。如果其中一个值严格更大，则最多只能存在一种类型的完美网格状态，不妨设为列完美网格态。接着，我们检查 M 是否存在某一列具有唯一的最小元素。如果不存在，根据命题 2.4，我们立即知道 $\mathbb{G}$ 不具有列完美网格状态，因而根本不具有完美网格状态。如果存在这样的列，我们保存其唯一的最小元素，然后删除包含该元素的 M的列和行。对于新的约化矩阵，可能出现三种情况：

算法 1新完美网格状态

要求： 一个长度为 n 的置换 σ，以及一个相同长度的列表 J，使得对所有 $1 \le i \le n$，$J[i] \neq \sigma[i]$。

确保： 一个与 σ 不同的新置换 τ，使得在每个索引 i 处，新的置换取值为 $\sigma(i)$ 或 $J(i)$。

$n \leftarrow$长度(σ)
$S \leftarrow$复制(σ)
$S[1] \leftarrow J[1]$ ▷ 将第一个元素替换为来自 J的值
$i \leftarrow 1$ ▷ 中最近一次更改的索引
$changes \leftarrow [\, i \,]$ 已修改索引的列表。
当 为真**时执行**
  $duplicates \leftarrow$ S 的重复元素
  如果 $duplicates = \emptyset$**那么** ▷ 未发现重复项：该置换有效
    $\tau \leftarrow$复制(S)
    返回 τ
  else
    设 t 为 $duplicates$ 中满足 $t \neq i$的唯一元素
  结束如果
  如果 $t \in changes$**则** ▷ 检测到循环：仅修改 t之间的索引
    $start \leftarrow t$ 在 $changes$中的位置
    $loop \leftarrow$ 从位置 $start + 1$ 起 $changes$ 的元素
    $\tau \leftarrow$复制(σ)
    对所有 $k \in loop$ **执行**
      $\tau[k] \leftarrow J[k]$
    结束循环
    返回 τ
  结束条件判断
  $S[t] \leftarrow J[t]$ ▷ 将 S 在 t 处更新为来自 J的值
  $i \leftarrow t$ ▷ 将当前索引更新为最近更改的索引
  将 i 追加到 $changes$
结束 while

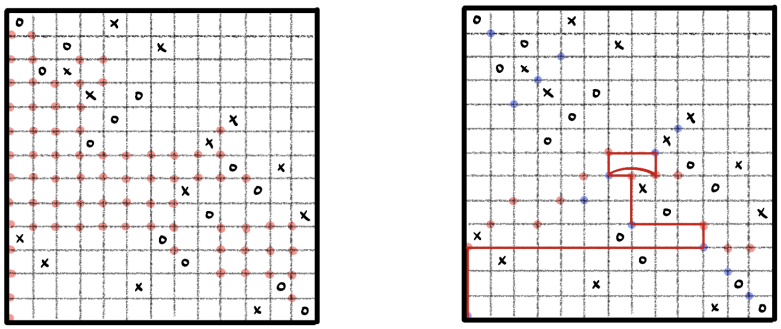

图 3。 命题 2.4 的可视化解释。

左：一个网格图示例，其中绕数矩阵没有任何一列具有唯一的最小值。红点表示该列中绕数矩阵的最小值项。

右：对于每一列，我们选取两个最小值项（蓝色和红色）。蓝点表示一个列完美网格状态。然后我们绘制一条曲线（红色），直到网格状态中的某个点被访问两次。由此曲线形成的闭环告诉我们，可以将哪些蓝点切换为红点，从而创建一个新的列完美网格状态。

- 现在有些列不再包含其原来的最小值作为项。在这种情况下，我们知道 $\mathbb{G}$ 不存在列完美网格状态。

- 对于所有剩余的行，不存在任何一列仅有一个具有最小值的项。如果发生这种情况，根据与命题 2.4 中相同的论证，我们知道 $\mathbb{G}$ 不存在唯一的列完美网格状态。$\mathbb{G}$ 可能存在多个完美网格状态，也可能一个都不存在。

- 同样，至少存在一列，其中有一个唯一的元素，其值为该列原来的最小值。在这里，我们可以像之前一样继续操作：提取这个新的唯一最小元素的原始位置，然后再次缩减矩阵。

这个过程要么在找到唯一的列完美网格状态后停止，要么在经过某次迭代后出现情况1或2时停止。在 $r = c$ 的情况下，我们确实必须对列和行执行此方法。这一过程的一个缺点是，它无法区分存在多个列完美网格状态的网格图与不存在列完美网格状态的网格图。了解这一点会很有意思，因为如果 $\mathbb{G}$ 不存在列完美网格状态，它仍可能有一个极大

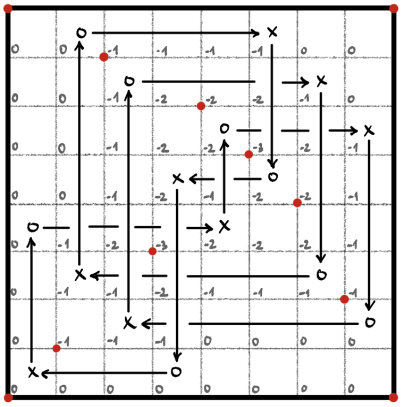

图4. 一个网格图示例，其中的网格状态是极大的且唯一的，但并非完美。

且唯一的网格状态。图4给出了一个例子。

例2.5. 图4中以红点标示的网格状态 x 是极大的且唯一的，但并非完美。事实上，它并不完美，因为 $A'(x) = 13$ 而$r = 14 = c$。注意，对应绕数矩阵的第5行和第6行，其唯一的最小值都位于同一列，因此不可能存在行完美网格状态。类似地，第3列和第4列的唯一最小值也位于同一行。因此，列完美网格状态也不可能存在，而我们看到 x 是极大的。任何其他极大网格状态在绕数矩阵第3列或第4列对应的值，都必须位于第5行；这迫使第2、5、6、7和8列的元素与 x 位于相同位置。现在很容易看出， x 是唯一最大网格态。

虽然当网格状态是完美网格状态时，我们证明极大网格状态唯一性的方法很快，但从前面的例子可以看出，当网格状态并非完美网格状态时，要证明其唯一性就繁琐得多。在图4中，网格状态 x 几乎是完美的，因为 $A'(x) + 1 = \min\{r, c\}$。一般来说， $\min\{r, c\}$ 与 $A'(x)$之间的差异越大，这种方法就越不实用。因此，我们将重点研究具有唯一完美网格状态的纽结网格图。为方便起见，我们引入以下定义。

定义2.6。 给定纽结 K的网格图 $\mathbb{G}$，若存在一个唯一的完美网格状态state，适用于 $\mathbb{G}$，则称 $\mathbb{G}$ 是良好的。

我们不直接处理问题 i，而是将其重新表述为：

问题2.7。 每个纤维结都存在优美网格图吗?

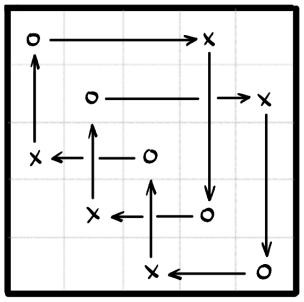

图 5。 三叶结的平面网格图。

3 使用 Python 编写的寻找漂亮网格图的程序

最后一章介绍了 Python 程序*griddiagrams* [Itz26]，该程序基于前几章的思路开发。它旨在通过生成一个大型列表，其中包含代表输入纽结的候选图，来寻找纤维结的优美网格图。这里我们将介绍关键函数；更多文档可参阅源代码。

回顾 [OSS15，第 3.1 节2]，网格移动，即交换和（去）稳定化，是对网格图进行的不会改变底层结型的修改。它们可用于创建同一个纽结的大量图的列表。对于每个网格图，我们希望应用上一节中的高效方法来寻找唯一完美网格状态。虽然这种方法非常蛮力，但大多数所需的计算都可以非常高效地实现。

我们使用由 *https://knotinfo.org* [LM26] 提供的结数据库。该数据库包含 5397 个纤维化素纽结，而且很方便的是，其中所有纽结都已经配备了最小网格图表示。网格图以所谓的 网格表示法给出，它是一个元组列表，用于指定 X-和 O-标记的位置。例如，图 5 中大小为 5 的图将由以下 10 个元组组成的序列给出

$$(1,1),(1,3),(2,2),(2,4),(3,3),(3,5),(4,1),(4,4),(5,2),(5,5).$$

在这种表示法中，网格图的定向未作指定。为了高效地进行列交换，用垂直线段来描述网格图要实用得多。在这种描述中，图 5 由 5 个定向垂直区间表示。

$$(3,1),(4,2),(5,3),(1,4),(2,5).$$

我们称之为 *vertlist* 表示法。Python 程序分两步将网格表示法转换为 vertlist 表示法：首先使用函数 `gridnotation_to_gridlist()` 将网格表示法转换为中间格式，然后使用`vlist()`将此格式转换为 vertlist 表示法。

在这一变换过程中所选择的定向并不重要。在网格图上反转 X 和 O 标记的位置，相当于反转底层纽结的定向。这一操作在函数 `rev()` 中实现，稍后将使用该函数在两种定向下寻找唯一完美网格状态。在 vertlist 表示法中，一个简单的函数 `can_commute()` 就足以寻找可能的列交换。为了高效地执行行交换，我们使用其垂直线段来描述网格图。我们将其称为*horzlist-* 表示法。另一个简单的 Python 函数 `v_to_h()` 可将 vertlist 表示法转换为 horzlist 表示法；我们再次使用 `can_commute()` 来检查相邻行是否可以交换。下一步是获取一个以 vertlist 形式给出的网格图，并计算经过一次行交换或列交换后能够得到的所有网格图的列表。这通过 `c_move()` 完成。由于我们将网格图视为嵌入在环面上，该函数允许在第一条和最后一条垂直线段之间进行列交换，也允许在最高行和最低行之间进行行交换。

使用 vertlist 表示法描述网格图的另一个好处是，计算绕数矩阵十分直接；参见 `w_matrix()`，该函数的灵感来自程序 Gridlink [Cul]。对于一个纽结的矩阵，我们可以使用由命题 2.4 推导出的方法，寻找唯一的行完美和列完美网格状态。函数 `h_type_0_permutation()`[1] 可以找出唯一的行完美网格状态。类似地，我们为唯一的列完美网格状态定义了 `v_type_0_permutation()` 。根据命题 I，唯一完美网格状态的 Alexander 分次与该纽结的 Seifert 属一致。因此，程序还包含 `a_grading()`，这是一个计算网格状态的 Alexander 分次的函数，用于验证代码找到的唯一完美网格状态是否满足这一必要条件。所有这些功能都整合在 `try_permutations()` 中；该函数会尝试为给定的图寻找唯一完美网格状态。

现在，我们已经具备创建第一个广度优先搜索（BFS）算法的所有基础模块，即 `gridstate_finder_commute()`。该算法仅使用交换操作生成大量表示，从而寻找优美网格图。如果算法成功，由于交换操作不会改变网格大小，因此得到的是最小大小的优美网格图。该方法成功为 5397 个纤维化素纽结中的 4346 个找到了优美网格图。

对于其余纽结，我们在进行稳定化后采用类似的方法。任何稳定化都可以通过 X-SW 型稳定化实现，然

[1]“h型-0”表示差值 $r - A'(x)$ 为零的网格状态 x。

交叉数	未解结
≤ 11	$\emptyset$
12	$12n_{79}, 12n_{168}$
13	$13n_{282}, 13n_{917}, 13n_{1279}, 13n_{1281}, 13n_{1413}, 13n_{1826}, 13n_{2915}, 13n_{3089}, 13n_{3904}, 13n_{3932}$

表 1. 按交叉数分类的未解结

后再进行一系列交换 [OSS15, Cor. 3.2.3]。类似地，只需结合交换进行 X-NW 型稳定化即可。函数 `x_nw()` 会对 vertlist 表示法中的网格图在指定线段上执行 X-NW 型稳定化。第二个广度优先搜索算法`gridstate_finder_stab()`随后尝试在进行一次稳定化后找到一幅优美图。该算法明显慢于 `gridstate_finder_commute()`，这很可能是因为稳定化会引入一条长度为一、可以与任意线段交换的线段，从而大幅增加可能的交换次数。尽管如此，它仍成功为其余纽结中的 1039 个找到了优美网格图。还有 12 个交叉数为 ≤ 13 的纤维化素纽结尚未解决。

4 最后的评注与问题

其余交叉数为 ≤ 13 、尚未找到优美网格图的纤维化素纽结列于表 1。

巧合的是，对所有未解结而言，最小网格尺寸均与交叉数重合。在完整数据集中，35% 的纽结的最小网格尺寸为 14 或 15。对于所有这些纽结，均已找到优美网格图；参见图 6 中两个网格大小为 16 的优美网格图示例，即这些是最小网格尺寸为 15 的纽结的例子，代码仅在一次稳定化之后才找到优美网格图。

对于未解结，引入第二次稳定化并未得到任何优美网格图。然而，要恰当地研究经过多次稳定化的网格图，就有必要大幅提高算法的效率。一种可能的方案是采用某种形式的多进程。在本项目中，我们已经使用多进程同时为多个纽结寻找优美网格图。不过，对于最耗时的纽结，

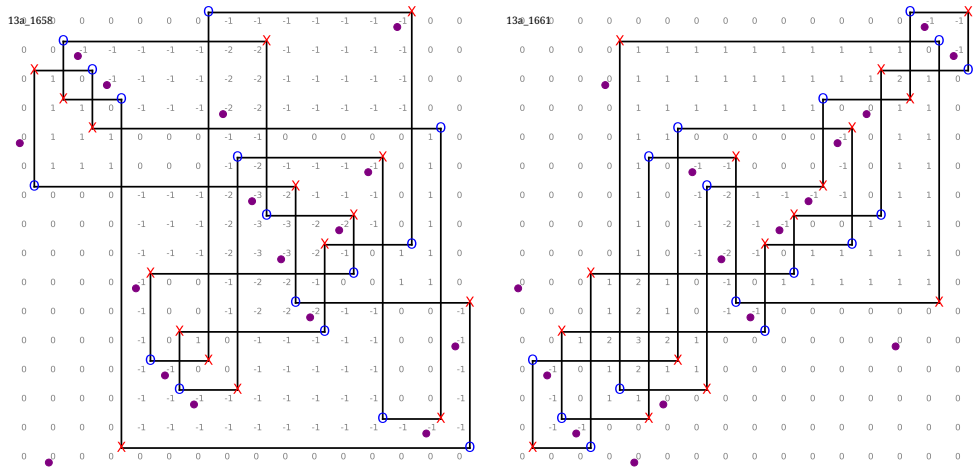

图 6。 两个网格大小为 15 的纤维化素纽结的优美网格图示例，其中进行了稳定化。紫色圆点表示相应的完美网格状态。

并行化的广度优先搜索算法会快得多。创建效率更高的代码版本仍是一项值得探索的任务。目前，代码的功能也有所局限；缺少的功能包括：

- 代码只能执行类型为 $X - NW$ 的稳定化。虽然从理论上讲这已经足够，但引入其他稳定化类型可能会大幅减少所需的交换次数。- 代码不会对图进行反稳定化。- 我们不考虑 2 次或更多次稳定化。

即使不引入反稳定化，并且在最小网格图上至多使用一次稳定化，这个简单的 Python 程序也能够为数据库中的 5397 个纤维化素纽结中的 5384 个找到优美网格图。这引出了如下问题。

问题 4.1。 每个具有唯一最大网格态的网格图的纽结，是否也都具有优美网格图?

这个项目引出了许多更有趣的问题，例如：

- 对于那些只有经过稳定化后才找到优美网格图的纽结，是否存在具有最小网格尺寸的优美网格图？我们没有引入

逆稳定化操作。这可能与以下问题有关：同一大小的一个纽结的所有网格图，是否都能仅通过交换彼此连通，还是必须进行稳定化和逆稳定化。Dynnikov 证明，任意平凡结的网格图都可以在不进行稳定化的情况下，化简为最简单的图 [Dyn06]。然而，程序 Gridlink[Cul] 的文档指出，[Men06] 的附录中有一个纽结的网格图示例，无法在不进行稳定化的情况下化简。

- 假设某个纽结既有一个具有唯一极大网格状态的最小网格图，也有一个优美网格图，那么是否存在最小优美网格图？目前，代码尚未为 9_{42},找到最小优美网格图，不过图4给出了 9_{42} 的一个具有唯一极大网格状态的最小网格图示例。

我们以一系列备注结束本项目。

- 如今，已经为大量纤维化素纽结找到了优美网格图，因此，采用机器学习方法来寻找更高效的优美网格图搜索方法，或许会很有意思。目前还没有衡量一个网格图距离优美网格图有多远的方法；机器学习方法甚至或许能帮助找到这样的度量标准。

- 该软件包会检查唯一完美网格状态的 Alexander 分次是否与输入纽结的三属一致。根据引理I，情况必然如此。对于所有已找到优美网格图的纽结，网格状态的分次都与纽结的亏格一致。这强有力地表明代码得出了正确结果。此外，还可以将所得网格图的同调秩与原纽结的同调秩进行比较，例如使用 Gridlink [Cul], , 以进一步确认计算正确。我们已对若干网格图进行了抽样比较，结果表明同调秩始终正确。

参考文献

[Ras03] Jacob Andrew Rasmussen. *Floer* 同调与纽结补空间. 哈佛大学, 2003年。[OS04a] Peter Ozsváth 和 Zoltán Szabó. “全纯圆盘与亏格界”

.

载于：《几何与拓扑》8.1 (2004), pp. 311–334.[OS04b] Peter Ozsváth 和 Zoltán Szabó。“全纯圆盘与纽结不变量”。载于：《数学进展》186.1 (2004), pp. 58–116.[Dyn06] Ivan Dynnikov。“链环的弧表示：单调化简”。载于：《数学基础》1.190 (2006), pp. 29–76.[Men06] William W Menasco。“关于迭代环面纽结的补遗”。载于：*arXiv* 预印本 *math/0610566* (2006).[Man+07] Ciprian Manolescu 等。“组合链环 Floer 同调”。载于：《几何与拓扑》11.4 (2007), pp. 2339–2412.[Ni07] Yi Ni。“纽结 Floer 同调可检测纤维化纽结”。载于：《数学创刊》170.3 (2007), pp. 577–608.[Ghi08] Paolo Ghiggini。“纽结 Floer 同调可检测亏格一纤维化纽结”。载于：《美国数学杂志》130.5 (2008), pp. 1151–1169.[MOS09] Ciprian Manolescu、Peter Ozsváth 和 Sucharit Sarkar。“纽结 Floer 同调的组合描述”。载于：《数学年刊》(2009), pp. 633–660.[SW10] Sucharit Sarkar 和 Jiajun Wang。“一种用于计算某些 Heegaard Floer 同调的算法”。载于：《数学年刊》(2010), pp. 1213– 1236.[OSS15] Peter S Ozsváth、András I Stipsicz 和 Zoltán Szabó。纽结与链环的格点同调。第 208 卷。美国数学会，2015.[Itz26] Paul Leon Itzlinger。*griddiagrams*：用于处理纽结网格图的 *Python* 程序。2026 年 1 月。网址：https://github.com/pauli tzlinger/griddiagrams.[LM26] Charles Livingston 和 Allison H. Moore。*KnotInfo*：纽结不变量表。网址：knotinfo.org。2026 年 1 月.[Cul] Marc Culler。*Gridlink*：用于处理纽结网格图的图形工具。网址：https://github.com/3-manifolds/gridlink（访问于 2024 年 12 月 15 日）。