arXiv:2512.13929v1 [cs.CG] 15 Dec 2025

第一离散同调群的快速计算

Jacob Ender KrzysztofKapulkin

2025年12月17日

摘要

我们提出了一种计算图的第一离散同调群的新算法。通过在不同的随机图数据集上测试该算法，我们发现其性能显著优于其他已知算法。

引言

离散同调是在十多年前才被提出的 [BCW14], ，如今已成为多种应用中的重要图不变量。它属于更广泛的离散同伦论领域[BBdLL06,BL05], ，并通过Hurewicz定理 [CK24] 与离散同伦群密切相关，因此能够从拟阵理论、超平面配置和拓扑数据分析等多个角度揭示图的重要信息 [BL05, KK25]。因此，人们投入了大量工作来理解这一不变量，包括：将其与其他图的“同调理论”进行比较 [BGJW19], 、研究其消失性 [BGJW21b], ，以及探讨加速相关软件计算的方法的理论成果 [BGJW21a, GWW25]。离散同调的首次软件计算可追溯到 [BGJW19] ，此后在 [KK24b], 中得到了改进，表明人们可以合理地计算许多实际中遇到的图的第三离散同调群。

本文沿袭 [BGJW19, KK24b], 的工作，专注于计算图的第一离散同调群，并为此提出一种新算法。该算法基于一个简单的观察：图 G 的第一离散同调群，可以改为计算与该图 X_G 相关联的胞腔复形 X_G 的第一（胞腔）同调群。复形 X_G 以 G为其 1-骨架，并将 G 的 3-圈和 4-圈作为其 2-胞腔的边界。 X_G 的胞腔链复形中前两个微分的边界矩阵，通常远小于计算 G离散同调所用链复形中相应微分的矩阵。这一基本观察最初是在离散同伦群的背景下由 [BKLW01], 提出的；得益于离散 Hurewicz定理 [CK24, CKT23], 它也适用于离散同调。

我们的算法始终优于其他现有算法，包括 [BGJW19、KK24b]中使用的算法。这些先前的算法并未针对计算第一同调群进行优化，而是着重于在合理的计算时间和资源范围内，尽可能计算高维同调。我们之所以聚焦于第一离散同调群，是因为在离散同伦论应用于拓扑数据分析时，它是最常用的同调群 [KK25]。具体而言，我们的算法与持久性算法兼容 [ZC05]。

然而，这也引出了一个问题：在更高同调次数下，能否实现类似的改进？在2维中，确定3-圈和4-圈是需要粘合的唯一形状，并不能为更高维度中可能出现的这类形状提供明确的研究方向。更重要的是，将3-圈和4-圈推广到更高维度的方式有太多种，其中一些会导致

2020年数学主题分类：05-04（主要），05C25、55U15（次要）。

可以证明是错误的猜想。寻找一种合适的推广方式，即基本形状猜想，仍然是离散同伦论领域最困难、最重要的开放问题之一。

本文组织如下。第 1 节回顾离散同调所需的理论背景，第 2 节则概述相关算法。我们在第 3 节介绍自己的算法，并在第 4 节讨论该算法与现有算法的运行时间比较。我们算法的伪代码见 A 节；用于运行时间测试的 Julia 实现可在 https://github.com/JacobEnder/DiscreteHomology-Algorithms 获取。

1 离散同调

本节回顾图的离散同调所需的背景知识。离散同调最初由 [BCW14] 在度量空间语境中引入，并通过图的相关滤子进行研究；随后，离散同调几乎完全是在图论语境中得到研究的 [BGJW19, BGJW21b, CK24,KK24b]。

由于我们的算法在域 $\mathbb{Z}/2$ 上效果最佳，我们将只考虑取 $\mathbb{Z}/2$-系数的同调。为此，本文中所有向量空间均取自 $\mathbb{Z}/2$。

我们先回顾图和图映射的定义。非正式地说，我们研究的是简单、无向的自反图及其图同态；形式化定义如下：

定义 1.1。

1. 一个图 G 由集合 G_V及其上的自反对称关系 $\sim \subseteq G_V \times G_V$构成，该关系称为邻接关系。集合 G_V 称为 G的顶点集，而关系 $\sim$ 是 G的边集。我们用 $v \sim w$ 表示 $(v, w) \in \sim$。

2. 对于图 G 和 H，函数 $f\colon G_V \to H_V$是一个图映射，如果 f 保持邻接关系。

例 **1.2 例**。图 I_n的顶点集为 $\{0, 1, \ldots, n\}$，边为 $(i, i+1)$，其中 $0 \le i \le n-$ 1.

文献中已广泛研究了不同的图乘积概念 [IK00,KK24a,GK25]，但为了定义离散同调，我们只需要盒积，现将其引入。

定义 1.3. 给定图 G 和 H，它们的 盒积 $G\square H$ 的顶点集为 $G_V \times H_V$，边是形如 $((u, v),(u', v'))$ 的顶点对，其中 $u = u'$ 和 $v \sim v'$，或者 $u \sim u'$ 和 $v = v'$。

离散同调是图中非退化奇异 n-立方体所构成的链复形的同调。下述定义概括了这一构造的组合方面。

定义 1.4. 1. 离散 n-超立方体 是图 $I_1^{\square n}$，其顶点标记为 $(x_1, \ldots, x_n)$，其中 $x_i \in \{0, 1\}$， $0 \le i \le n$。

2. 图 G 中的一个奇异 n-立方体 是一个图映射 $A\colon I_1^{\square n} \to G$。

3. 对于图 G、整数 $n \ge 1$、奇异 n 立方体 $A\colon I_1^{\square n} \to G$ 和 $1 \le i \le n$，定义 i 阶正面和负面 A如下：

$$\delta_i^+ A(x_1, \ldots, x_{n-1}) := A(x_1, \ldots, x_{n-1}, 1, x_i, \ldots, x_n)$$
$$\delta_i^- A(x_1, \ldots, x_{n-1}) := A(x_1, \ldots, x_{n-1}, 0, x_i, \ldots, x_n).$$

4. 奇异的 n-立方体 $A\colon I_1^{\square n} \to G$ 若对于某个 i，有 $\delta_i^+ A = \delta_i^- A$，则称其为退化的。奇异的 n-立方体若不是退化的，则称为非退化的。

从这里开始，我们只需要简单代数：

定义 1.5. 对于图 G，定义 $C_n(G)$ 为由 G 中的非退化奇异 n-立方体在 $\mathbb{Z}/2$ 上张成的自由向量空间。这些空间构成一个链复形，其微分在奇异n-立方体上的定义为

$$\partial_n(A) := \sum_{i=1}^{n} (-1)^i (\delta_i^- A - \delta_i^+ A).$$

G 的离散同调 是这个链复形的同调，即 $\mathcal{H}_n(G) = \ker(\partial_n)/\mathrm{im}(\partial_{n+1})$。

当然，要证明按上述方式定义了微分的 C_n 确实构成链复形，还需要做一些工作；这一点已在 [BCW14]中得到验证。

2 计算

接下来，按照 [KK24b], , 我们回顾两种现有的用于计算图的离散同调的算法，分别称为立方算法 和边图算法。对于每种算法，我们先简要介绍，再进行复杂度分析。在讨论算法的复杂度时，我们用 n 表示 G的顶点数，用 m表示边数。最早借助计算机进行的离散同调计算见于 [BGJW21b], , 但其中的算法现已被[KK24b]提出的算法取代。

立方算法。 立方算法是一种用于计算任意同调群 $\mathcal{H}_n(G)$ 的算法，因此并未针对专门计算第一同调群进行效率优化。其关键思路是快速生成图中的奇异（$n+1$）立方体：不再考虑任意集合映射并验证其是否为有效的图映射，而是注意到，G 中的奇异 n立方体可以通过检查相应顶点是否由边连接，递归地将两个（$n-1$）立方体配对而构造出来。算法步骤如下。详细的伪代码见 [KK24b]。

1. 生成 0-立方体（顶点）和 1-立方体（边）的列表。这一步的时间复杂度为 $O(1)$。2. 找出非退化的 1-立方体并构造边界算子 ∂_1 的矩阵，然后计算 $\dim(\ker(\partial_1)) = m - \mathrm{rank}(\partial_1)$ 。此秩计算在最坏情况下的时间复杂度为 $O(n^4)$，因此整个步骤的时间复杂度为 $O(n^4)$。3. 对每一对 1-立方体，检查它们是否构成有效的非退化 2-立方体：检查相应的邻接关系，并确保这两个 1-立方体不共享退化方向。这一步的时间复杂度为$O(n^4)$。4. 找出非退化 2-立方体，构造边界算子 ∂_2的矩阵，并计算 $\mathrm{rank}(\partial_2)$。这一步最耗时的部分是通过高斯消元计算 $\mathrm{rank}(\partial_2)$。 ∂_2 的行数为 $O(n^2)$，列数为 $O(n^4)$。因此，这一步的时间复杂度为 $O(n^2 \cdot n^4 \cdot \min(n^2, n^4)) = O(n^8)$。5. 返回 $\dim(\mathcal{H}_1(G)) = \dim(\ker(\partial_1)) - \mathrm{rank}(\partial_2)$。其时间复杂度为 $O(1)$。

整个三次　　代数算法的时间复杂度为 $O(n^8)$，因为其主要耗时在于计算秩(∂_2)。

边图算法。给定一个图 G，其边图 是图 $E(G)$：它的顶点是 G中的边，若两条边在一个 4-圈中互为对边，则这两个顶点之间由一条边连接。在维度 1 中，我们可以不显式构造边图而计算离散同调。相反，只需枚举 G中的某个 4-圈子集。为找到这些 4-圈，我们采用一种可用于构造边图的过程，具体见下文步骤 5。我们可以不显式计算边图，而将找到的圈作为 2-立方体集合，并据此构造边界矩阵。边图算法的伪代码见 A 节。该论文作者提供的这一方法在维度 1 中的实现可见于 https://github.com/JacobEnder/DiscreteHomology-Algorithms/tree/master/edge_graph_homology。该算法的运作方式如下。

1. 对每个顶点 $v \in G_V$，构造其邻域 $N(v)$，即所有与 v 相邻的顶点列表。这一步是 $O(n^2)$。 2. 构建链群 C_0 和 C_1。 C_0 就是顶点集 G_V，而 C_1 是边集 G_E，其中排除顶点的自环。这一步是 $O(1)$。 3. 构造边界算子 ∂_1 的矩阵，并计算 $\dim(\ker(\partial_1))$。在最坏情况下，这一秩计算的复杂度同样是 $O(n^4)$。 4. 为了协助构建 ∂_2 的矩阵，构造一个坐标字典，将每条边映射到其在 ∂_1 中的列索引。这一步是 $O(n^2)$。 5. 按照以下过程找出所有 2-立方体。对于所有 $v \in G_V$，获取邻域 $N(v)$，并考察 w、 $v' \in N(v)$。如果 $v' > w > v$（使用顶点的自然序），那么任何满足 $w' \geq v$ 的 $w' \in N(w) \cap N(v')$ 都会与顶点 v、w、 v'、 w' 构成一个非退化2-立方体。我们还必须考虑其他情况，并在这些情况下调整这些不等式，以确保找到所有必要的 2-立方体。这个过程是 $O(n^4)$。 6. 构造 ∂_2 的矩阵并计算其秩。与立方算法类似，这一秩计算的复杂度为 $O(n^8)$。 7. 返回 $\dim(\mathcal{H}_1(G)) = \dim(\ker(\partial_1)) - \operatorname{rank}(\partial_2)$。这同样是 $O(1)$。

边图算法的最坏情况时间复杂度为 $O(n^8)$，与立方算法相同。然而，在实际应用中，对于更稀疏的图，边图算法比立方算法更快。这是因为立方算法会对所有边对进行朴素搜索，以寻找将 1-立方体配对成 2-立方体的方法；而边图方法则会对每个顶点的邻域进行局部搜索。在完全图这种最坏情况下，这两种过程耗时相同；但对于更稠密的图，边图搜索要快得多。

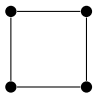
3 胞腔算法

事实证明，第一离散同调群存在一种拓扑实现：可以将一个拓扑空间 X_G 与图 G关联起来，使得G的离散同调与 X_G的（奇异）同调一致。最初，这一结果是在离散同伦 群的背景下由 [BKLW01, Prop. 5.12] 给出的。然而，无论在离散情形还是经典情形下，第一同调群都是第一同伦群的阿贝尔化，因此所需结果可直接得出。

为了构造 X_G，我们回顾一下图中简单圈的定义。

定义 3.1. 对于 $n \geq 3$，图 G 中的一个简单 n 圈是由互不相同的顶点组成的序列（v_0，……， v_{n-1}），满足 $v_i \sim v_{i+1}$对 $0 \leq i \leq n-2$、 $v_{n-1} \sim v_0$ 成立，并且（v_0，……， v_{n-1}）的任何真子序列都不满足这些条件。

例 3.2. 图中的每个 3-圈都是简单的。下图展示了一个简单 4-圈和一个非简单 4-圈的例子。



一个简单 4-圈。



一个非简单 4-圈。

有了简单 n-圈的定义后，我们如下构造所需的 CW 复形 s.

定义 3.3。 固定一个图 G，并定义一个二维 CW 复形 X_G ： 规定其 1-骨架为图 G，并在每个简单 3-圈和 4-圈上粘附一个 2-胞腔。

有了这些，我们现在可以陈述支撑我们算法的定理:

定理 3.4。 对于连通图 G，有 $\mathcal{H}_1(G) \cong H_1(X_G)$，其中 H_1 表示一个拓扑空间的奇异同调。

如上所述，该证明依赖于 Barcelo、Kramer、Laubenbacher 和 Weaver 关于离散同伦群的先前结果。为简洁起见，此处不再详细回顾这些结果，而是直接将不熟悉这些概念的读者引向以下文献中的任意一篇： [BKLW01、BBdLL06、CK24]

证明。 根据 [BKLW01，第 5.12] 号命题（参见 [CK24]），有 $A_1(G) \cong \pi_1(X_G)$，其中 A_1 表示 G的第一离散同伦群。利用离散的 [BCW14，第 4.1]号定理，以及经典的 [Hat02，第 2.A 节]一维 Hurewicz 定理，左侧的阿贝尔化为 $\mathcal{H}_1(G)$，右侧的阿贝尔化为 $H_1(X_G)$，从而证明完毕。 □

胞腔算法

根据定理 3.4，我们可以通过计算 $H_1(X_G)$ 来计算 $\mathcal{H}_1(G)$，因此，我们的算法首先在图中找出 3-圈和 4-圈，然后计算所需的胞腔微分，最后使用高斯消元求出矩阵的秩。由此，我们可以将算法描述如下:

1. 找出图 G 中所有的 3-圈和 4-圈。 2. 创建（稀疏）矩阵 M ： 其列由边、3-圈和 4-圈索引，行由 G 的边和顶点索引，具体如下: 如果顶点 / 边 σ_i 出现在边 / 圈 σ_j 中，则令 $M_{ij} = 1$; 否则，令 $M_{ij} = 0$。因此， M 表示计算 X_G （系数取自 $\mathbb{Z}/2$）的链复形中的前两个微分。 3. 创建一个查找表，记录 M 中对应于边的边界的列索引，以及对应于 3-圈和 4-圈的边界的列索引。分别用 M_1 和 M_2 表示由这两组列索引确定的 M 的子块。 4. 计算 $H_1(X_G) = m - \text{秩}(M_1) - \text{秩}(M_2)$。根据定理 3.4，这等于 $\mathcal{H}_1(G)$。

现在，我们详细讨论该算法的每个步骤。

条目 1：查找 3-圈和 4-圈。算法的第一步是枚举图中所有作为 CW复形 X_G 的 2-胞腔的 3-圈和 4-圈。为列出 3-圈，我们使用 K3 算法 [CN85], 的一种实现，该算法对相邻顶点的邻域取交集，并报告每个三角形一次。此方法的时间复杂度至多为 $O(m^{3/2})$ 或 $O(n^3)$，因为 $m = O(n^2)$。为枚举 4-圈，我们使用 [AKLS23], 的 4-圈枚举算法的多线程实现，该算法枚举 G 中的所有 2-路径（即由顶点组成的三元组 (u, v, w)，其中 $u \sim v$ 且 $v \sim w$），并检查端点匹配的 2-路径，从而找出 G 中的简单 4-圈。其计算复杂度为

此方法为 $O(n^2 + t)$，其中 t 是 G 中简单 4-圈的数量，即 $O(n^4)$，因此该算法为 $O(n^4)$。不过，在通常感兴趣且一般相当稀疏的情况下，这两种算法的运行速度都快得多。K3 算法和 4-圈枚举算法的伪代码可分别见其对应论文。

第 2 项。构建边界矩阵。我们构建胞腔链复形中前两个微分的稀疏边界矩阵 M。我们在 $\mathbb{Z}/2$ 上工作，并将顶点存储为整数。与 X_G 的结构一致， M 的列编码边以及 3-圈和 4-圈的边界。如果列 j 对应一条边，我们就在列 j 中与其端点对应的行索引处插入 1。如果列 j 对应一个 3-圈或 4-圈，我们就在与其组成边对应的行索引处插入 1。所需的顶点和边搜索（在边和圈内部）要求每个单元进行 $O(1)$ 次查找。例如，查找一个 3-圈的组成边需要进行三次查找，每条边一次。这意味着我们必须进行 $O(m + C)$ 次查找，其中 C 是 G 中 3-圈和 4-圈的总数。在最坏情况下， G 中 4-圈的数量为 $O(n^4)$，因此构建 M 同样需要 $O(n^4)$。

条目 3. 维护查找表。构造出 M 后，为了方便起见，我们维护一个查找表，记录哪些列对应于边、3-圈和 4-圈，这样之后就不必再推断或重新计算这些信息。 M 的列按以下顺序排列：对应边的列排在对应 3-圈和 4-圈的列之前。因此，对应于边的边界的列索引介于 $n + 1$ 和 $n + m$之间。同样，对应于 3-圈和 4-圈的列位于索引 $n+m+1$与 M中的最大列索引之间。要创建查找表，我们只需在字典中存储一些固定索引，因此这一步的时间复杂度为常数 $O(1)$。

条目 4. 计算 $\mathcal{H}_1(G)$。计算 X_G 的第一离散同调群，只需直接进行秩计算。对于图 G，将 M 中对应于边界为边的列的子块记为 M_1，将 M 中对应于边界为 3-圈和 4-圈的列的子块记为 M_2。我们利用查找表找出所需的列索引范围，计算 $\mathcal{H}_1(G) = m - \mathrm{rank}(M_1) - \mathrm{rank}(M_2)$。在 $\mathbb{Z}/2$ 上进行秩计算时使用 Modulo2 Julia 软件包；该软件包通过高斯消元以及底层的列相加指令，实现了高度优化的秩计算。秩计算是胞腔算法中开销最大的阶段。再次令 C 表示 G中的简单 3-圈和 4-圈数量，则整个边界矩阵 M 的大小为 $(n + m) \times (m + C)$。在最坏情况下， $n + m$ 为 $O(n^2)$。我们之前看到，在最坏情况下， C 为 $O(n^4)$，所以 $m + C$ 为 $O(n^4)$。一个 $k \times \ell$ 矩阵的高斯消元时间复杂度为 $O(k\ell \cdot \min(k, \ell))$。因此，对于完全图，秩计算的时间复杂度为 $O(n^2 \cdot n^4 \cdot n^2) = O(n^8)$。边界矩阵的构造以及整体胞腔同调计算的伪代码见 A 节。

4 运行时间比较

我们通过在一个实验数据集上测试每种算法，比较细胞方法、立方方法和边图方法计算 1 维离散同调的运行时间。该数据集由 800 个 Erd˝os–R`enyi 图组成（即图 $G(n,\ p)$，包含 n 个顶点，且任意一对顶点之间独立地以固定概率 $p \in [0, 1]$ 连接一条边）。该数据集分为四类：

1. 具有固定边概率 $p = 0.07$ 和随机数量 n 个顶点的 200 个图，其中 $100 \leq n \leq 300$；2. 具有固定边概率 $p = 0.13$ 和随机数量 n 个顶点的 200 个图，其中 $100 \leq n \leq 300$；3. 具有 100 个顶点和随机边概率 $p \in [0.07, 0.13]$ 的 200 个图；4. 具有 300 个顶点和随机边概率 $p \in [0.07, 0.13]$ 的 200 个图。

我们选择相对较低的概率参数 $p \leq 0.13$，是因为更稀疏的图更有可能具有非平凡的第一同调群。对于连接性足够强的图，可以更有效地应用 [KK24b] 的预处理技术。在这里，我们希望关注那些预处理不太可能提高效率的图。

数据集被划分为以下四类，以控制 Erd˝os–R`enyi 图的参数变化可能产生的影响。我们选择了 n和 p 的界限，以确保存在大小各异、相对稀疏和相对稠密的图，同时仍能较大概率保留非平凡的第一同调。在这四组图中的每一组上分别运行三个算法后，我们收集了运行时间结果，并绘制图表进行比较。所有这些图表均可在 https://github.com/JacobEnder/DiscreteHomology-Algorithms/tree/master/results 查看，下面列出了其中一些。第一幅图展示了三个算法各自的运行时间与边概率 p 之间的关系。对于边概率固定的两组图，为便于阅读，我们展示了箱线图。所有时间均以秒为单位测量。

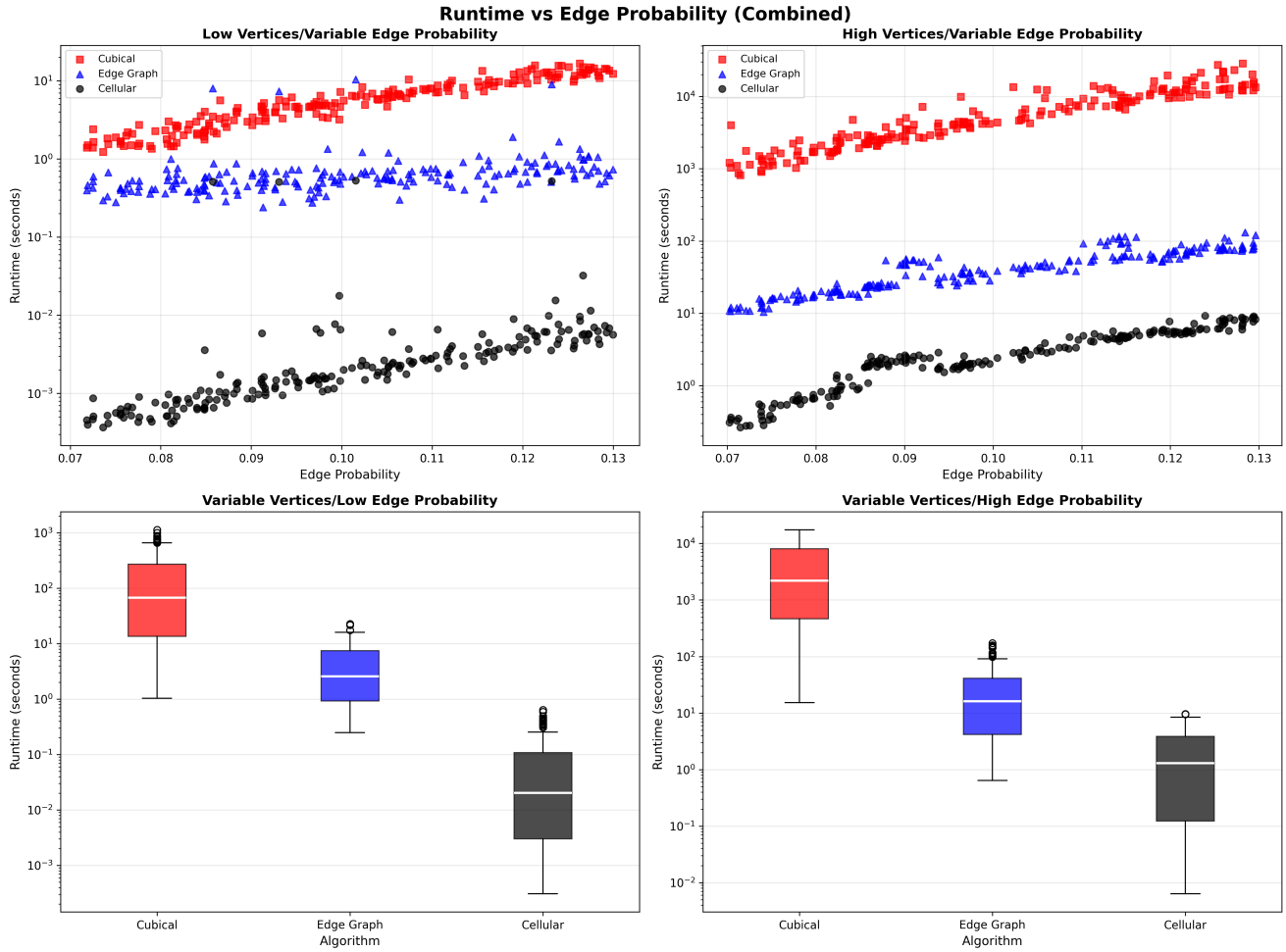


图 1：三个算法各自随边概率变化的运行时间。

可以清楚地看出，三种算法的性能存在明显差异。在这项实验中，立方算法的表现不如边图算法，而边图算法的表现又不如胞腔算法。从图表中可以看出，各算法的运行时间似乎都是边概率 {v1} 的近似相同函数，

只是相差一个常数因子（这一点在图1右上角的图表中最为明显）。这与我们上文进行的复杂度分析一致。我们发现，每种算法的最坏情况渐近复杂度都相同，但常数因子会影响实际计算时间。例如，边图方法必须枚举 G 中的所有 2-立方体，而胞腔算法只枚举简单的 3-圈和 4-圈。这使得胞腔算法的边界矩阵更小，从而带来一定的实际加速效果；尤其当 $\dim(\mathcal{H}_1(G))$ 较高时更是如此。此外，边图算法的实现中存在一些固定的计算开销，而胞腔算法则没有。这些开销包括为 G 中的每个 2-立方体分配稠密向量、拼接稠密向量等。胞腔算法的这类计算开销较少，因此在实际中性能显著提升。当前边图方法的实现可能存在一些可用于缩短运行时间的优化方法，但作者目前尚不了解此类优化。类似的分析也揭示了为何立方算法在实际中的速度甚至比边图算法更慢。

这些结果还有许多图表可在 GitHub 仓库中查看：https://github.com/JacobEnder/DiscreteHomology-Algorithms/tree/master/results。这些图表展示了立方算法、边图算法和胞腔算法分别与 $\dim \mathcal{H}_1(G)$、Erd˝–R`enyi 图的边概率 p 、数量 n，以及图中简单 3-圈和 4-圈总数之间的两两比较。结果目录中还包含一个文件 detailed_results.csv，其中列出了实验数据集中每个图 G 的以下数据：图名称、顶点数、边概率、简单 3-圈数量、简单 4-圈数量、这些数量之和、$\dim \mathcal{H}_1(G)$、三种算法在 G上的运行时间、三种运行时间的两两比值，以及运行速度最快的算法。

本文所述的胞腔算法是数据集中每个图的最快方法，因此最后一列的值始终相同。

一些算法的伪代码

通过 在本节中，$\text{rank}^{\mathbb{Z}/2}$ 指的是 Modulo2 包中的专用秩函数 e.

算法1 使用边图方法计算 $\mathcal{H}_1(G)$ 1：输入：带有邻域字典的图 G 2：输出：$\dim \mathcal{H}_1(G)$ 3：步骤1：构造1-链和1-边界矩阵 4：$\mathcal{C}_1 \leftarrow \{\{u,v\} : (u,v) \in E, u \neq v\}$ {非环边}5：将 $M_1 \in M_{|V|\times|\mathcal{C}_1|}(\mathbb{Z}/2)$ 初始化为零矩阵 6：对于每条边 $e = \{u,v\} \in \mathcal{C}_1$，执行 7：$M_1[u,e] \leftarrow 1$8：$M_1[v,e] \leftarrow 1$9：结束循环 10：步骤2：枚举2-链（方形映射）11：$\mathcal{C}_2 \leftarrow \emptyset$12：对于每个顶点 $v \in V$，执行 13：对于每个满足 $w > v$ 的 $w \in N(v)$，执行 14：对于每个满足 $v' > w$ 的 $v' \in N(v)$，执行 15：对于每个满足 $w' \geq v$ 的 $w' \in N(w) \cap N(v')$，执行 16：{找到方形：$v \sim w, v \sim v', w \sim w', v' \sim w'$}17：将方形 (v,w,v',w') 和 (v,v',w,w') 添加到 $\mathcal{C}_2$ 18：结束循环 19：结束循环 20：添加退化方形 21：结束循环 22：结束循环 23：步骤3：构造2-边界矩阵 24：将 $M_2 \in M_{|\mathcal{C}_1|\times|\mathcal{C}_2|}(\mathbb{Z}/2)$ 初始化为零矩阵 25：对于每个方形 $\sigma = (v,w,v',w') \in \mathcal{C}_2$，执行 26：面 $\leftarrow [\{v,w\},\{v',w'\},\{w,w'\},\{v,v'\}]$27：对于每个面 $f \in$，执行 28：如果 $f \in \mathcal{C}_1$，则 29：$M_2[\text{index}(f), \text{index}(\sigma)] \mathrel{+}= 1$（模2）30：结束条件 31：结束循环 32：结束循环 33：步骤4：计算同调维数 34：$r_1 \leftarrow \text{rank}^{\mathbb{Z}/2}(M_1)$ 35：$r_2 \leftarrow \text{rank}^{\mathbb{Z}/2}(M_2)$ 36：$\dim \mathcal{H}_1 \leftarrow |\mathcal{C}_1| - r_1 - r_2$37：返回 $\dim \mathcal{H}_1$

算法 2 胞腔方法：计算 $\mathcal{H}_1(G)$ 1：输入：具有邻接表表示的图 G 2：输出：$\dim \mathcal{H}_1(G)$ 3：步骤 1：提取图结构 4：$\mathcal{E} \leftarrow \textsc{Edges}(G)$ 5：$\mathcal{T} \leftarrow \textsc{EnumerateTriangles}(G)$ {K3 算法}6：$\mathcal{C} \leftarrow \textsc{EnumerateFourCycles}(G)$ {由 Abboud 等人提出的算法}7：$\mathcal{S} \leftarrow \mathcal{T} \cup \mathcal{C}$ {所有短圈}8：步骤 2：构造边界矩阵 9：$M \leftarrow \textsc{BuildBoundaryMatrix}(V, \mathcal{E}, \mathcal{S})$ 10：步骤 3：提取子矩阵 11：$M_1 \leftarrow M$[顶点行、边列] {大小 $|V| \times |\mathcal{E}|$}12：$M_2 \leftarrow M$[边行、圈列] {大小 $|\mathcal{E}| \times |\mathcal{S}|$}13：步骤 4：计算同调维数 14：$r_1 \leftarrow$秩$_{\mathbb{Z}/2}(M_1)$ 15：$r_2 \leftarrow$秩$_{\mathbb{Z}/2}(M_2)$ 16：$\dim \mathcal{H}_1 \leftarrow |\mathcal{E}| - r_1 - r_2$17：返回 $\dim \mathcal{H}_1$

算法 3 $\textsc{BuildBoundaryMatrix}$：构造稀疏边界算子

1: 输入：顶点集 V，边集 $\mathcal{E}$，短圈 $\mathcal{S} = \mathcal{T} \cup \mathcal{C}$2: 输出：稀疏边界矩阵 $M \in M_{(|V|+|\mathcal{E}|)\times(|V|+|\mathcal{E}|+|\mathcal{S}|)}(\mathbb{Z}/2)$ 3: 构建边索引映射：idx$[e] \leftarrow$ $\mathcal{E}$ 中 e 的位置 4: 初始化坐标列表：$I \leftarrow []$、$J \leftarrow []$、$X \leftarrow []$5: 边列：∂(边) $=$边界顶点 6: 对于 $k = 1$ 到 $|\mathcal{E}|$，执行 7: $(u, v) \leftarrow \mathcal{E}[k]$8: 列 $\leftarrow |V| + k$9: 将 $(u$, 列, 1) 添加到 (I, J, X) 10: 将 $(v$, 列, 1) 添加到 (I, J, X) 11: 结束循环 12: 短圈列：∂(圈) $=$ 边界边 13: 对于 $k = 1$ 到 $|\mathcal{S}|$，执行 14: $\sigma \leftarrow \mathcal{S}[k]$15: 列 $\leftarrow |V| + |\mathcal{E}| + k$16: 边 $\leftarrow \sigma$的组成边 17: 对每条边 $e \in$（属于边）执行 18: 行 $\leftarrow |V| +$idx$[e]$19: 将 (行, 列, 1) 添加到 (I, J, X) 20: 结束循环 21: 结束循环 22: $M \leftarrow \textsc{SparseMatrix}(I, J, X, |V| + |\mathcal{E}|, |V| + |\mathcal{E}| + |\mathcal{S}|)$ 23: 返回 M

参考文献

[AKLS23] A. Abboud、S. Khoury、O. Leibowitz 和 R. Safier， *Listing 4-cycles*，《软件技术与理论计算机科学基础》第 43 届 IARCS 年会，2023 年，第 25 号文章，16 页。MR4695060[BBdLL06] E. Babson、H. Barcelo、M. de Longueville 和 R. Laubenbacher， *Homotopy theory of graphs*，《代数组合杂志》24 (2006)，第 1 期，31–44 页。[BCW14] H. Barcelo、V. Capraro 和 J. A. White， *Discrete homology theory for metric spaces*，《伦敦数学学会会刊》46 (2014)，第 5 期，889–905 页。[BGJW19] H. Barcelo、C. Greene、A. S. Jarrah 和 V. Welker， *Discrete cubical and path homologies of graphs*，《代数与组合》2 (2019)，第 3 期，417–437 页。[BGJW21a] H. Barcelo、C. Greene、A. S. Jarrah 和 V. Welker， *Homology groups of cubical sets with connections*，《应用范畴结构》29 (2021)，第 3 期，415–429 页。[BGJW21b] H. Barcelo、C. Greene、A. S. Jarrah 和 V. Welker， *On the vanishing of discrete singular cubical homology for graphs* **35** 2021 1, 35 54 4196413 ，《SIAM 离散数学杂志》（），第 – 期。MR[BKLW01] H. Barcelo、X. Kramer、R. Laubenbacher 和 C. Weaver， *Foundations of a connectivity theory for simplicial complexes* **26** 2001 1, 97 128 ，《应用数学进展》（），第 – 期。[BL05] H. Barcelo 和 R. Laubenbacher， *Perspectives on A-homotopy theory and its applications*，《离散数学》298 (2005)，第 1–3 期，39–61 页。[CK24] D. Carranza 和 K. Kapulkin， *Cubical setting for discrete homotopy theory, revisited*，《组合数学》160 (2024)，第 12 期，2856–2903 页。MR4881411[CKT23] D. Carranza、K. Kapulkin 和 A. Tonks， *The Hurewicz theorem for cubical homology*，《数学杂志》305 (2023)，第 4 期，第 61 号文章，20 页。MR4661259[CN85] N. Chiba 和 T. Nishizeki， *Arboricity and subgraph listing algorithms* **14** 1985 1, , 210 223 ，《SIAM 计算杂志》（），第 – 期。MR774940[GK25] A. Grenier 和 K. Kapulkin， *Biclosed monoidal structures on the categories of digraphs and graphs*，2025 年。预印本。[GWW25] C. Greene、V. Welker 和 G. Wille， *On the homology of simplicial and cubical sets with symmetries*，2025 年。预印本。[Hat02] A. Hatcher， *Algebraic topology*，剑桥大学出版社，剑桥，2002 年。MR1867354[IK00] W. Imrich 和 S. Klavˇzar， *Product graphs*，Wiley-Interscience 离散数学与优化丛书，Wiley-Interscience，纽约，2000 年。结构与识别，前言作者 Peter Winkler。MR1788124[KK24a] K. Kapulkin 和 N. Kershaw， *Closed symmetric monoidal structures on the category of graphs*，《理论与应用范畴》41 (2024)，第 23 号文章，760–784 页。MR4774716[KK24b] K. Kapulkin 和 N. Kershaw， *Efficient computations of discrete cubical homology*，2024 年。[KK25] K. Kapulkin 和 N. Kershaw， *Data analysis using discrete cubical homology*，2025 年。[ZC05] A. Zomorodian 和 G. Carlsson， *Computing persistent homology* **33** 2005 2, , 249 274 ，《离散与计算几何》（），第 – 期。MR2121296