arXiv:2512.25043v1 [cs.CC] 31 Dec 2025

细树验证是 coNP 完全

Alice Moayyedi *

2025年12月31日

摘要

图 G 的 α-*thin tree* T 是一棵生成树，其中每个割 G 在 T 中的边数至多占其边数的 α 比例。*thin tree conjecture* 提出，存在一个函数 f，使得对于任意 $\alpha > 0$，每个 $f(\alpha)$-边连通图都有一棵 α-薄树。除了其独立价值之外，对于给定的 k-边连通图，如果有一种算法能够高效构造一棵 $O(1)/k$-薄树，就能直接得到一个针对非对称旅行商问题（ATSP）的 $O(1)$-近似算法 [GS11]。然而，人们甚至还不知道能否高效地 *verify* 给定的树是 α-薄的。我们证明，判定一棵树的薄度是 coNP 困难的。

1 引言

设 G 是一个无向无权图。我们用 $V(G)$ 表示 G 中顶点的集合，用 $E(G)$ 表示边的集合。给定一个集合 $A \subseteq V(G)$，我们用 $\delta(A)$ 或 $\delta_G(A)$ 表示 G 中恰有一个端点位于 A 中的边的集合。

定义 1。图 G 的子图 S 若对每个割所含的边数至多占其边数的 α 比例，则称为 α-薄的；也就是说，对于 G 的任意割 $C = (A, B)$（其中 $A \subseteq V(G)$，$B = V(G) \setminus A$）：

$$|\delta_S(A)|/|\delta_G(A)| \leq \alpha$$

Goddyn 提出了以下猜想 [God04]:

猜想 1 (薄树猜想)。存在一个函数 f，使得对于任意 $\alpha > 0$，每个 $f(\alpha)$-边连通图都有一棵 α-薄生成树。

除了其他多种推论 [KO23]，之外，如果能够针对一个 k-边连通图构造一棵 $O(1)/k$-薄树，就能直接为标准非对称旅行商问题（ATSP） [GS11] 以及 ATSP 的“瓶颈”版本（其目标是最小化所经过的最大权重边） [AKS21]提供一个 $O(1)$-近似算法。前者已通过其他方法实现，但后者将改进目前已知的最佳算法 [AKS21]。已知在某些图类中可以构造常数因子的薄树 [GS11], ，但对于一般图，尚不知道是否存在任何 α 不依赖于 $n = V(G)$ 的 α-薄树。

事实上，据我们所知，尚无已发表的研究结果确定：验证给定生成树是否为 α-薄的难度。关于这个问题，有人称其尚不知是否存在多项式时间可验证证书 [KO23]，也有人称其为 NP 困难（但未给出证明） [Alg23]。我们证明，下文所述的验证生成树是否为 α-薄的问题是 coNP 完全的。

*戴维·R·切里顿计算机科学学院，滑铁卢大学，加拿大。 anelosima@proton.me

问题 (细树验证问题 (ThinTreeValid))。输入： 一个图 G、其生成树 T 以及一个值 $\alpha \in [0,1]$。输出： 如果 T 是 α-薄的，则输出（是）——也就是说，对于 A 的所有划分 G，

$$|\delta_T(A)|/|\delta_G(A)| \leq \alpha$$

——否则为（否）。

定理 1 (瘦树验证的困难性)。 ThinTreeValid 是 coNP 完全的。

注 1。 所给出的证明并未对 Thin-TreeValid 的近似困难性提供任何保证，也没有暗示任何显而易见的方法来近似 ThinTreeValid；因此，无论是直接证明无法找到边连通度与图的边连通度相差常数因子的生成树，还是寻找这样的生成树，该证明的价值都有限。我们提出这项工作，希望它能间接发挥作用。

2 证明策略

coNP 困难性的证明将依次通过四个问题进行归约，首先从经典问题 MaxCut 的无权版本出发：

问题 (无权最大割问题， MaxCut)。实例：一个（无向、无权）连通图 G和一个整数 k。输出：如果存在 G 的一个划分(A, B) 使得 $|\delta(A)| \geq k$，则输出（是）；否则输出（否）。

MaxCut 众所周知是NP 困难的 [GJS74]。首先，我们将其归约

到这样一个问题：在所有割中寻找具有最大平均权重的割 s:

问题（最大平均割优化问题， MaxAvgCutOpt）。实例：一个带权重的连通图 G，其中权重为 $\mathbf{w}$，满足 $\forall w_e \in \mathbf{w}\ w_e \in \{-1,1\}$。输出：对 G 进行一个非平凡划分（A, B），使得对于 G的所有非平凡划分 A', B'，

$$\left(\sum_{e\in\delta(A)} w_e\right)/|\delta(A)| \geq \left(\sum_{e\in\delta(A')} w_e\right)/|\delta(A')|$$

接下来，我们将其归约至一个自然的判定变体（为便于证明，在多重图上进行） :

问题（最大平均割判定问题， MaxAvgCut）。
实例： 一个带有权重 $\mathbf{w}$ 的连通多重图 G ，满足 $\forall w_e \in \mathbf{w}\ w_e \in \{-1,1\}$，以及一个整数$k \geq -1$。

输出： 若存在 G 的一个非平凡划分(A, B) ，使得

$$\left(\sum_{e\in\delta(A)} w_e\right)/|\delta(A)| \geq k,$$

否则输出 *(NO)*。

最后，我们将此判定问题归约到 ThinTreeValid的补问题：

问题 (细树验证问题的补问题， ThinTreeValid^{C})。实例： 一个连通图 G、 G的一个生成树 T，以及一个值 $\alpha \in [0,1]$。输出： 若 T 是 α-薄的，则输出 *(NO)*；否则输出 *(YES)*。

此外，对于任意图 G 及其子图 T，我们将割 $(A,\ B)$ 的 T-厚度 定义为 $\delta(A)$ 中属于 T 的边所占的比例；即 $|\delta_T(A)|/|\delta_G(A)|$。注意，子图 T 是 G 的 α-薄子图，当且仅当 G 的任意割的 T-厚度都不严格大于 α。

3 证明

注意， ThinTreeValid 属于 coNP；给定一个 ThinTreeValid 实例 $(G,\ T,\ \alpha)$， G 中任何 T-厚度大于 α 的割都能证明该实例是一个（NO）实例，而且这样的割必定存在。因此，只需证明它是 coNP 困难的。

定理 2. $\textsc{MaxCut} \leq_T^P \textsc{MaxAvgCutOpt}$。

证明。 我们将使用以下算法，其中 $A \oplus B$ 表示 A 和 B的对称差:

算法 1: 使用 MaxAvgCutOpt求解 MaxCut。

输入: 一个（无向）连通图 G;
一个整数 k;
一个 maco 子程序（G', $\mathbf{w}$），用于求解 MaxAvgCutOpt。

输出: MaxCut（G, k）。

$C \leftarrow \emptyset$

1 **重复**

对于 $e \in G$，**执行**

$$w_e \leftarrow \begin{cases} -1 & \text{如果 } e \in C \\ 1 & \text{如果 } e \notin C \end{cases}$$

$C' \leftarrow \delta(\text{maco}(G、\mathbf{w}))$

$C_{\text{prev}} \leftarrow C$

$C \leftarrow C \oplus C'$

直到 $\sum_{e \in C'} w_e \leq 0$

如果 $|C_{\text{prev}}| \geq k$，则输出（是）；否则，输出（否）。

注意，除最后一次迭代外，在第 1 行循环的每次迭代中， C' 包含的未与 C 共享的边比与 C共享的边更多。这是因为，当权重为 $\mathbf{w}$ 时， G 中的割的平均权重为正，当且仅当割外的边比割内的边更多；每条割外的边都会使分子增加 1，而每条割内的边都会使分子增加 -1。因此，每次迭代都会增加 C 中的边数，直到首次出现平均权重非正的迭代，此时第 1 行的循环将终止。

因此，除最后一次迭代外，每次迭代中 C 的大小都严格递增；由于 $|C| \leq |E(G)|$，迭代次数至多为 $|V(G)|^2$ 。由于每次迭代都在线性运行时间内完成，并且会调用一次 MaxAvgCutOpt 算法，算法 1 调用 MaxAvgCutOpt子程序的次数有多项式上界，且额外运行时间也有多项式上界。

图的割空间在对称差运算下封闭，因此每个 C produced 实际上都是 G 的一个割。此外，如果存在 G 的一个割 C^*，使得 $|C^*| > |C|$，那么在 MaxAvgCutOpt 实例中， $C^* \oplus C$ 的值将为正（G, $\mathbf{w}$）。因此， MaxAvgCutOpt 子程序不会返回一个值非正的割，算法也不会终止；只有在获得 G 中大小最大的割时，它才会终止。因此，鉴于算法终止时 G 中大小最大的割为 C_{prev} ，如果 G 中存在大小至少为 k 的割，算法就会输出（是）；否则输出（否）。 □

定理 3. $\textsc{MaxAvgCutOpt} \leq_T^P \textsc{MaxAvgCut}$.

证明。 我们使用算法 2。

首先注意，第一行只需通过多项式数量的检查即可完成对 MaxAvgCut 子程序的调用；所有平均权重的形式均为 p/q ，其中 $|p| \leq q \leq n$ 为整数，且 $n = V$（G）。因此，需要检查的权重有 O（n^2）个。此外，任意割的最小可能平均权重为 -1，所以所有输入都是有效的 MaxAvgCut 实例。

下面讨论循环。由于我们在第 2 行的循环中每次迭代都会收缩一对顶点，而第 3 行的循环对于 G 的每条边至多执行一次迭代，因此如果该算法终止，它将在

算法 2： 使用 MaxAvgCut 求解 MaxAvgCutOpt

输入： 一个（无向）连通图 G;
一组定义在 G 上的（整数）权重 $\mathbf{w}$，满足 $\forall w_e \in \mathbf{w}\ w_e \in \{-1, 1\}$;
一个子程序 mac(G', $\mathbf{w}'$, k)，用于求解 MaxAvgCut。

输出： (G, $\mathbf{w}$) 中平均大小最大的割。

1 通过对所有可能的最大平均权重进行穷举搜索，求出图 (G, $\mathbf{w}$) 中割的最大平均权重，并将其设为 k；
在所有可能的最大平均权重中取最大值（即，对每个可能的权重运行 mac(G, $\mathbf{w}$, k)，
mac(G, $\mathbf{w}$, k)，并取使 MaxAvgCut(G, $\mathbf{w}$ k) 为（是）的最大值）。
$G' \leftarrow G$
2 **当** $|V(G')| > 2$ **时执行**
3 **对** (v, u) $\in V(G') \times V(G')$ **执行**
通过在 G' 中收缩顶点 v 和 u 所形成的多重图。
如果 mac(G'', $\mathbf{w}$, k) **那么**
$G' \leftarrow G''$
中断
输出 G 中由与 G' 中各顶点相对应的顶点类所确定的划分。

多项式次调用 MaxAvgCut 子程序，并额外耗费多项式时间。要看出它确实会终止，请注意：如果存在一个最大平均割，将顶点 v 和 u 划分到不同的集合中，则收缩这两个顶点不会降低该图最大平均割的值；因此，在任何包含两个以上顶点的图中，至少存在两个顶点，可以在保持最大平均割大小不变的情况下进行收缩。因此，第 3 行的循环总会以收缩一对顶点结束；最终，G' 由两个顶点组成。

显然，在最后，与 G' 相关联的 G 中的割的平均权重为 k，因为 G' 中唯一的非平凡割的权重为 k，且该割中的边与 G 中相关联割的对应边具有相同的权重。因此，该算法会生成一个平均权重最大的割。 □

定理 4. MaxAvgCut $\leq_m^P$ ThinTreeValid$^{\mathrm{C}}$。

证明。 我们将通过卡普归约来证明这一点：从三元组（G, $\mathbf{w}$ k）构造出一个图 G'、一棵树 T 和一个值 α，使得 ThinTreeValid$^{\mathrm{C}}$ 实例（G', T, α）是一个（YES）实例，当且仅当MaxAvgCut 实例（G, $\mathbf{w}$ k）是一个（YES）实例。

简要来说，我们将构造一个“膨胀”图：原图中的每个顶点对应一个较大的团，而原图中的每条边对应这些团之间的一组边（见图1，不过请注意，实际的顶点数和边数并未按比例准确表示）。随后，我们将构造一棵树 T 和一个阈值 α，使得原图中平均权重至少为 k 的所有割，都对应于新图中的割；这些割在 T 中的边数超过 α比例（从而违反了 T 为 α-薄的性质），反之亦然。直观地说，我们让团之间 T 的边数与原图中相应顶点之间的边权重成比例，并确保这些团足够大，使得将一个团分开的割具有极低的 T-厚度（因此在确定树的薄度时无关紧要，因为树的薄度仅由厚度最大的割决定）。为简化分析，我们将团构造得比满足这一性质所需的大小更大；我们在每个团中生成的顶点数为 $24 \cdot |E(G)|$，但更小的数量也足够，只是会增加证明的复杂性。

取实例 MaxAvgCut（G, $\mathbf{w}$ k）。我们首先使用算法 3 和算法 4 构造一个相关的 G' 和 T。

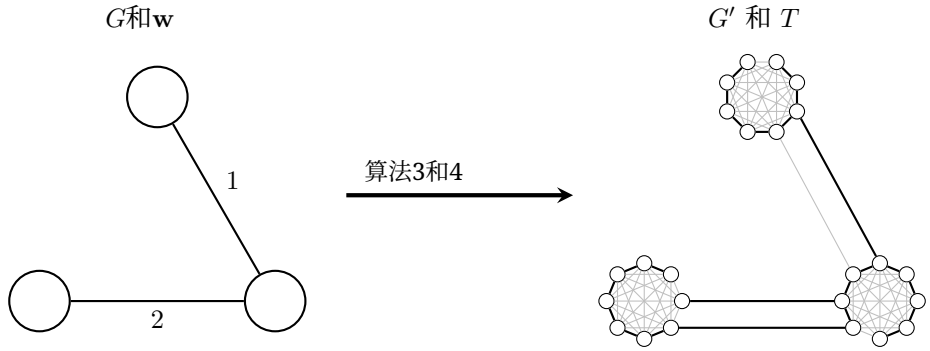


图1： G 和**w** 变换为"膨胀后"的图 G' 和一棵树 T的粗略示意图。

算法 3： G'的构造

输入： 一个连通多重图 G。

令 $m = |E(G)|$。

$G' \leftarrow (\emptyset, \emptyset)$

对于 $v \in V(G)$ **，执行**

  向 G' 添加一个由 $24m$ 个新顶点组成的团 c_v。

对于 $e = (v, w) \in E(G)$ **，执行**

1 在 G' 中选取 3 对互不相同的顶点（v_0, w_0），（v_1, w_1），（v_2, w_2），每对中一个来自 c_v，另一个来自每个 c_w 中尚未与各自团之外顶点有任何关联边的顶点。（也就是说，选择此前未在此过程中作为一对成员被选过的顶点对（通过此过程得到的一对。）

  对于每一对，向 G' 添加边 (v_i, w_i)。将这三条边组成的集合称为 d_e。

算法 4： T 的构造

输入： 一个连通多重图 G;
通过算法 3 从 G 构造的图 G'；
在 G 上的一组整数权重 **w**，使得 $\forall w_e \in \mathbf{w}\ |w_e| \leq 1$。

令 $m = |E(G)|$。

对于每个 $v \in V(G)$，令 c_v 为 G' 中与之关联的团。

对于每个 $e \in E(G)$，令 d_e 为 G' 中与之关联的三条边。

$S \leftarrow (\emptyset, V(G'))$

对 $v \in V(G)$ **执行**

  向 T' 中添加一条路径，连接 c_v的顶点。

对 $e = (v, w) \in E(G)$ **执行**

  从 d_e中的边里选取边，并将其添加到 S $w_e + 2$ 中。

$T \leftarrow S$

1 **当** T 不是一棵树时**执行**

  在 T中找到一个环 C。

  选择一条边 e，它位于 C 中，且属于 G' 的某个团内部。

  从 T 中移除 e。

断言 1. 算法 *3* 和算法 *4* 均可在多项式时间内执行，且对任何符合其约束的输入都能执行。

证明. 显然，算法 3 以及算法 4 中除第 1 行循环之外的所有部分都可在多项式时间内执行；而且显然，两个算法中除算法 3 的第 1 行外的所有行都始终可以执行。

在算法 3 中，注意到任意顶点的度至多为m，因此每个团中至多有 $3m$ 个顶点需要由第 1 行选出；由于 $3m \geq 24m$ 而 $m \geq 1$，每个团中总是有足够的顶点可按这种方式选出，因此第 1 行始终可以执行。

要说明算法 4 第 1 行的循环可在多项式时间内执行，注意到使用任何简单搜索算法都能在 $O(m)$ 时间内找到一个环，并且任何环必然包含一条团内边（ G' 中不存在仅由团间边构成的环，因为任意两条团间边都不共享顶点）。由于 T 中的边数每次迭代都会减少一条，该循环会在多项式时间内终止。

□

断言 2. 算法 *4* 生成的 T 是 G' 的生成树。

证明. 很容易看出，算法 4 生成的 S 是 G' 的生成子图，因为 G 是连通的，并且 G' 的每个团内部都是连通的。具体来说，对于任意两个顶点 $v' \in c_v$ 和 $u' \in c_u$，都可以通过在 G' 中构造一条路径 P' 来连接它们：先取 G 中从 v 到 u 的一条路径 P，再将其中的每条 e 替换为 d_e 中的任意一条边，并补上完成路径所需的团内路径段中的边。因此，生成的 T 也必定是生成子图，因为从环中删除一条边绝不会使任何顶点断开连接（环中的每个顶点仍可沿着该环保持连通，而且不会有其他顶点因此被单独断开）。

□

接下来，我们选取一个合适的 α，以确保 MaxAvgCut 实例（G, $\mathbf{w}$ k）是一个（YES）实例，当且仅当 ThinTreeValid$^{\mathrm{C}}$ 实例（G', T, α）是一个（YES）实例。假设 G' 的一个割 $C' =$（A', B'）保留了所有团内边；也就是说，对于某些顶点 a_0, ..., $a_k \in G$, $A' = \bigcup_{i=0}^{k} c_{a_i}$。将这样的割与 G 的一个割 $C =$（A, B）对应，其中 $A = \{a_0, \ldots, a_k\}$。

引理 1. 存在一个 ϵ，使得对于 $\alpha = (2+k)/3 - \epsilon$，任何平均权重至少为$k$ 的 G 的割 C，都对应于 G' 的一个割 C'，其 T-厚度严格大于 α。同样，任何尊重所有团内边且 T-厚度严格大于 α 的 G'的割 C'，都对应于平均权重至少为 k的 G的一个割 C。

证明。 证明见第 3.1 小节。 □

引理 1 表明，若 MaxAvgCut 实例 (G, $\mathbf{w}$ k) 是一个（YES）实例，则 ThinTreeValid$^{\mathrm{C}}$实例 (G', T, α) 也是一个（YES）实例。

引理 2. 对于按照算法 *4* 构造的 T 和引理 *1* 中的 $\alpha = 2/3 - \epsilon$，任何将团的顶点划分到不同集合中的 G' 的割 C'，其 T-厚度都低于 α。

证明。 证明见第 3.1 小节。 □

引理 2 蕴含：如果 ThinTreeValid$^{\mathrm{C}}$ 实例（G', T, α）是一个（YES）实例，那么 G' 中存在一个 T-厚度严格大于 α 的割，并且该割保留所有团内边（存在某个适当具有 T-厚度的割，而任何分割一个团的割都不可能具有 T-厚度，因此它必然是一个不分割团的割）。根据引理 1，这对应于 G 中平均权重至少为 k的一个割；因此，MaxAvgCut 实例（G, $\mathbf{w}$ k）是一个（YES）实例。

因此，当且仅当 ThinTreeValid$^{\mathrm{C}}$实例 (G', T, α) 是一个（是）实例时， MaxAvgCut 实例 (G, $\mathbf{w}$ k) 才是一个（是）实例；由于映射 (G, $\mathbf{w}$ k) $\mapsto$ (G', T, α) 可以在多项式时间内完成， MaxAvgCut $\leq_m^P$ ThinTreeValid$^{\mathrm{C}}$。

□

定理 1（瘦树验证的困难性）。 ThinTreeValid 是 coNP-完全的 .

证明。 由定理 2、定理 3 和定理 4 可知:

$$\begin{aligned} & \textsc{MaxCut} \leq_T^P \textsc{MaxAvgCutOpt} \leq_T^P \textsc{MaxAvgCut} \leq_m^P \textsc{ThinTreeValid}^{\mathrm{C}} \\ \Rightarrow \quad & \textsc{MaxCut} \leq_T^P \textsc{ThinTreeValid}^{\mathrm{C}} \end{aligned}$$

因此， $\textsc{ThinTreeValid}^{\mathrm{C}}$ 是 NP 困难的；其补问题 ThinTreeValid 因而是 coNP 困难的。由于 ThinTreeValid 属于 coNP，因此它是 coNP 完全的。 □

3.1 其余引理的证明

引理 *1* 的证明。 我们希望 G 中平均权重为 $\geq k$ 的割，对应于 G' 中厚度为 T 的团之间的割。给定的割 $C = (A,\ B)$ 在 G 中对应于 G' 中的割 $C' = (A',\ B')$，其划分之间的边数是前者的 3 倍；并且所有满足 $e \in \delta(A)$ 的 w_e 之和，等于 $\delta(A')$ 中属于 T 的边数减去 $\delta(A)$ 的每条边所对应的 2。因此，

$$\begin{aligned} & \frac{\sum_{e\in\delta(A)} w_e}{|\delta(A)|} \\ = \quad & \frac{|\delta_T(A')| - 2|\delta(A)|}{|\delta(A)|} \\ = \quad & \frac{|\delta_T(A')|}{|\delta_{G'}(A')|/3} - 2 \end{aligned}$$

So:

$$\begin{aligned} & k \leq \left(\sum_{e\in\delta(A)} w_e\right) / |\delta(A)| \\ \Leftrightarrow \quad & (k+2)/3 \leq |\delta_T(A')|/|\delta_{G'}(A')| \end{aligned}$$

然而，我们希望 G' 中平均比例 T 严格大于 α 的任意割，都对应于 G 中平均权重至少 k 的割（$\leq$是 $>$的补问题，而不是 $\geq$）。于是，我们所需的 α 为（$(k+2)/3-\epsilon$，其中 ϵ 足够小，使得 G' 中不存在厚度落在区间（$(k+2)/3-\epsilon$, $(k+2)/3$,]内的割。为了找到一个可行的 ϵ，首先注意到，G' 中任意割的 T 厚度都具有如下形式： p/q，其中 $p,\ q \leq |E(G')|$ 为整数。因此，如果我们取集合

$$S = \{(p,q) : 0 \leq p,q \leq |E(G')|, p/q < (k+2)/3\}$$

并且 $\alpha = \max_{(p,q)\in S}(p/q)$，不可能存在任何 T-厚度介于 α 和 $(k+2)/3$之间的割。我们将ϵ 设为能使 α 等于该值的任意数值。

这保证了，在 G' 中，任何 T-厚度大于 α，且不会将同一团的不同顶点分配到不同划分中的割，都对应于 G 中的一个平均权重至少为k的割；同样，这类 T-厚度至多为 α 的割，其平均权重小于 k。 □

引理 *2* 的证明。 考虑任意将 G' 中的一个团拆分为两部分的割 $C' = (A', B')$。我们将证明，C' 的 T-厚度严格小于 $(k+2)/3 > 1/3$。为此，先考察一个被拆分团的内部边以及所有团间边，再考察其余所有团内边。

考虑某个被它拆分的团 c_v，并将划分中顶点数较少一侧的顶点数记为 a（另一侧有 $24m-a$ 个顶点）。团内边的数量为 $(a)(24m-a)$，并且

其中至多有 $2a$ 条位于 T 中。假设每条团间边都位于 T 中，且其数量达到最大值；这就是 $3m$ 条边。因此，仅包含 c_v 中的边以及所有团间边的 C' 部分的 T-厚度至多为：

$$\frac{3m+2a}{3m+a(24m-a)} \leq \frac{3m+2a}{a(24m-a)} \tag{1}$$

$$\leq \frac{3m}{a(24m-a)} + \frac{2}{24m-a} \tag{2}$$

由于 $a \leq 12m$，我们可以对方程 2 的右项进行界定：

$$\frac{2}{24m-a} \leq \frac{1}{6m}$$

当 $a(24m-a)$ 最小时，左项最大。此时 $a=1$：

$$\frac{3m}{a(24m-a)} \leq \frac{3m}{24m} = \frac{1}{8}$$

因此，这部分 C 的 T-厚度至多为 $\frac{1}{8}+\frac{1}{6m} < \frac{1}{3}$。于是，将一个团一分为二的割，其 T-厚度严格小于 $(k+2)/3$，且至多为上文给出的 $\alpha = (k+2)/3 - \epsilon$。现在我们考虑其他被分割的团内部边的贡献（如果存在）；对于任意这样的团，根据上文的界，该团贡献给 C 的边中，至多有 $1/8$ 部分位于 T。因此，考虑来自其他被分割团的边，不可能使 C 的 T-厚度超过 α。

□

致谢

我们谨向 Elena Grigorescu 致谢，感谢她提供的建议和指导；向 Greg Rosenthal 致谢，感谢他进行了全面的校对和文字编辑；并向 Kaleb Ruscitti 致谢，感谢他提供反馈。

参考文献

[AKS21] Hyung-Chan An、Robert Kleinberg 和 David B. Shmoys。“瓶颈非对称旅行商问题的近似算法”。载于：*ACM* 算法汇刊 17.4（2021 年 10 月）。ISSN：1549-6325。DOI：10.1145/3478537。URL：https://doi.org/10.1145/3478537。[Alg23] Mahtab Alghasi。“结构化图中的组合稀疏树与谱稀疏树”。博士学位论文。滑铁卢大学，2023 年。[GJS74] M. R. Garey、D. S. Johnson 和 L. Stockmeyer。“一些简化的 NP 完全问题”。载于：第六届 *ACM* 年度计算理论研讨会论文集。STOC '74。美国华盛顿州西雅图：计算机协会，1974 年，第 47–63 页。ISBN：9781450374231。DOI：10.1145/800119.803884。URL：https://doi.org/10.1145/800119.803884。[God04] Luis A. Goddyn。我喜欢的一些开放问题。2004 年。URL：https://web.archive.org/web/20240617180800/https://www.sfu.ca/~goddyn/Problems/problems.html（访问日期：2025 年 12 月 15 日）。[GS11] Shayan Oveis Gharan 和 Amin Saberi。有界亏格图上的非对称旅行商问题。2011 年。arXiv：0909。2849 [cs.DS]。URL：https://arxiv.org/abs/0909。2849。[KO23] Nathan Klein 和 Neil Olver。层状族的稀疏树。2023 年。arXiv：2304。07674 [cs.DS]。URL：https://arxiv.org/abs/2304.07674。