arXiv:2305.01367v1 [cs.DS] 2 May 2023

用于线性排列和层次聚类的多层剥离

Yossi Azar* Danny Vainstein†

2023年5月3日

摘要

我们提出了一种新的多层剥离技术，用于对度量空间中的点进行聚类。一种广为人知的非参数目标函数是将度量空间嵌入到更简单、结构化的度量空间中，例如直线（即线性排列）或二叉树（即层次聚类）。度量空间中彼此接近的点应映射到直线或树上彼此接近的点或叶节点；同样，度量空间中相距较远的点，在直线或树上也应相距较远。具体而言，我们研究度量空间上的最大线性排列问题 Hassinand Rubinstein [2001] 以及最大层次聚类问题 Cohen-Addad 等人 [2018]。

我们为这些目标设计了近似方案（对任意常数 $\epsilon > 0$，其近似比为 $1 - \epsilon$）。具体而言，这表明考虑度量空间能够显著改进以往的近似比（最大线性排列为 0.5，最大层次聚类为 0.74）。我们的主要技术称为多层剥离，其过程是递归地剥离那些远离度量空间“核心”的点。当核心成为权重足够密集的度量空间（即平均距离至少为直径的某个常数倍），或相对于其内部目标函数贡献变得可以忽略时，递归便会终止。有趣的是，线性排列情形下的算法比层次聚类情形下的算法复杂得多，并且采用了精细得多的剥离方法。

1 引言

无监督学习在机器学习领域中发挥着重要作用。可以说，最突出的一种无监督学习类型是通过聚类实现的。从抽象层面来看，在这种设定下，我们获得一组数据点，这些数据点之间存在某种成对关系，并通过度量空间来刻画（即距离越近的点越相似）。为了更好地理解数据，目标是在保留原始成对关系的同时，将该空间嵌入一个结构更简单的空间中。在这一领域，一种常见的解是对数据构建平面聚类（或划分）（例如，使用 k 均值算法）。然而，这类解最终无法捕捉所有成对关系（例如，簇内关系）。为了克服这一困难，通常会将度量空间映射到能够捕捉所有成对关系的结构中——在我们的研究中，即映射为线性排列（LA）或层次聚类（HC）。

使用线性排列或层次聚类结构来嵌入空间，这一想法并不新鲜。这类解在实践中已得到广泛应用（例如，参见Citovsky 等人 [2021],Sumengen 等人 [2021],Aydin 等人 [2019], Bateni 等人 [2017], Rajagopalan 等人 [2021]），从理论角度也得到了广泛研究（例如，参见 Dasgupta [2016], Cohen-Addad

*特拉维夫大学计算机科学学院。电子邮箱：azar@tau.ac.il。部分经费由以色列科学基金会提供（资助编号：2304/20）。†特拉维夫大学计算机科学学院。电子邮箱：dannyvainstein@gmail.com。

等人 [2018],Moseley 和 Wang [2017],Charikar 等人 [2006], Feige 和 Lee [2007], Hassin 和 Rubinstein[2001]）。值得注意的是，线性排列类目标最早由 Hansen Hansen [1989]提出，他研究了将图嵌入二维及更高维平面的问题。另一方面，层次聚类类目标的研究由 Dasgupta Dasgupta [2016] 开启，推动了这一富有成果的研究方向，并催生了许多新颖的算法。在实践中，数据往往满足三角不等式（特别是，这保证了：如果点 a 与点相似，也就是距离接近 b 和 c，那么 b 与 c也如此），因此可以用度量来刻画（例如，参见Charikar 等人[2019b], Naumov 等人 [2021], Rajagopalan 等人 [2021]）

我们考虑的第一个目标是最大化线性排列目标。

定义 1.1。 设 $G = (V, w)$ 表示一个度量（具体而言， w 满足三角不等式），其中$|V| = n$。在**最大线性排列问题**中，我们的目标是返回一个一一映射 $y : V \to [n]$，使 $\sum_{i,j} w_{i,j} y_{i,j}$ 最大化，其中 $y_{i,j} = |y_i - y_j|$。

我们考虑的第二个目标是最大化层次聚类目标。

定义 1.2。 设 $G = (V, w)$ 表示一个度量（具体而言， w 满足三角不等式）。在**最大层次聚类问题**中，我们的目标是返回一棵二叉层次聚类树 T，使其叶节点与 V 一一对应。此外，我们希望返回 T，使 $\sum_{i,j} w_{i,j} |T_{i,j}|$ 最大化，其中 T_{ij} 是层次聚类树 T 中以叶节点 i 和 j 的最近公共祖先为根的子树，$|T_{i,j}|$ 是 $T_{i,j}$ 中的叶节点数。

这些目标最初由 Hassin 和 Rubinstein 提出Hassin 和 Rubinstein [2001] 以及 Cohen-Addad 等人提出Cohen-Addad 等人 [2018] （分别）针对非度量情形进行了研究。对于这些（非度量）目标，目前已知的最佳近似比分别为：线性排列目标的 0.5Hassin 和 Rubinstein [2001] 和 0.74（层次聚类目标）Naumov 等人 [2021]）。前者通过随机将数据点二分，随后贪心地排列每个集合而实现；后者则通过近似求解 Balanced Max-2-SAT 问题而实现。

如前所述，实际应用中所考虑的数据往往满足三角不等式。因此，我们的研究成果有两方面的价值。首先，我们提出了一个通用框架，用于处理这类嵌入目标。其次，我们的结果表明，采用这一自然假设可以显著改进此前已知的最佳近似比（从 0.5（线性排列）和 0.74（层次聚类）提升至任意常数 $\epsilon > 0$ 对应的 $1 - \epsilon$）。

我们的结果。 我们给出以下结果。

- 我们设计了一个通用框架，以解决将度量空间嵌入更简单的结构化空间的问题（见算法 1）。随后，我们将该框架具体应用于线性排列和层次聚类问题。扩展讨论请参见“我们的技术”。

- 我们将该框架应用于线性排列问题。在此情形下，我们证明所应用的算法（2）是一个 EPRAS（见定义 3.2）——也就是说，对于任意常数 $\epsilon > 0$，它能给出 $1 - \epsilon$ 近似。

- 我们将框架应用于层次聚类问题。在这种情况下，我们证明所应用的算法（4）是一种 EPRAS（参见定义 3.2），也就是说，对于任意常数 $\epsilon > 0$，它都能给出 $1 - \epsilon$近似。

我们的方法。 我们通用的多层剥离方法见算法1。我们首先检查度量空间的权重是否足够密集（即平均距离是否至少为直径的某个常数倍，或者等价地，度量的加权密度（参见定义 2.1）是否为常数）。如果是，则应用一种专门处理此类实例的算法。对于 LA 问题，我们设计了自己的算法（参见算法 3）。算法 3 利用 Goldreich 等人的一般图划分算法 Goldreich 等人 [1998]，以“猜测”一个最优图划分，从而得到近乎最优的线性排列。对于 HC 问题，我们利用 Vainstein 等人的工作 Vainstein 等人[2021]。

然而，如果度量的权重密度不足，我们会发现其中必定包含一个**核心**——一个由节点构成的子集，几乎包含所有数据点，且其直径显著小于原度量的直径。随后，我们的通用算法会剥离远离核心的数据点（在线性排列中），或不属于核心的数据点（在层次聚类中）。接着，我们嵌入这些被剥离的数据点：在线性排列中，将它们放在线的一端；在层次聚类中，则将它们排列成梯子结构（见定义3.5）。因此，我们只需处理核心（在层次聚类中）或扩展核心（在线性排列中）。

我们再次考虑两种情况：扩展核心内的总权重足够小，此时我们任意嵌入核心；否则，我们对这些数据点诱导出的实例进行递归。我们声称，在每个递归步骤中，扩展核心的密度都会显著增加，直到递归最终结束：要么扩展核心的权重密度足够高，要么扩展核心内的总权重足够小。

我们的证明基于若干个论断。首先，我们考察度量的（扩展）核心与剥离层 o 的比较。由于我们的算法分别嵌入这两个集合，因此需要界定由此导致的目标值损失。我们证明，剥离层 o 内部的权重对目标值的贡献可以忽略不计，而剥离层 o 与（扩展）核心之间的权重则具有显著贡献。因此，为了最大化目标值的增益，剥离这一层是合理的。

虽然上述结果足以界定单个递归步骤中的损失，但这还不够。递归步骤的数量可能不是常数，原则上可能导致误差急剧增大。然而，我们证明每一层的误差都由一个几何序列界定，因此其总量受最深递归步骤的误差支配。由此，我们能够将累积总误差上界控制在一个常数以内，并且该常数可以任意小。

总体而言，虽然这描述了我们的证明技巧，但线性排列目标的算法和分析要稍微复杂一些，因为我们将考虑三组：度量的核心、剥离出的层，以及其余所有点；后者与核心合称为扩展核心。在这种情况下，为了能够证明剥离出一层是合理的，我们必须更积极地选择该层。具体来说，我们将这一层定义为与核心距离足够远的点（而不是像层次聚类的情形那样，将核心以外的任何点都纳入其中）。幸运的是，这个定义的层（见算法 2）符合我们的标准（即通用算法算法 1）。

相关工作。 虽然层次聚类这一概念由来已久，但层次聚类目标相对较新。在其开创性工作中，Dasgupta Dasgupta [2016] 从优化的角度考虑了层次聚类问题。随后，Cohen-Addad 等人 Cohen-Addad 等人 [2018] 首次考虑了我们在本文中采用的目标。他们在工作中证明，众所周知的平均链接算法可达到 2 3 的近似比。随后，Charikar 等人 Charikar 等人[2019a] 通过采用半定规划改进了这一结果，达到了 0.6671 的近似比。最后，Naumov 等人 Naumov 等人 [2021] 通过近似平衡最大 2-SAT 问题，将该结果提高到了 0.74。对于最大 LA 目标，Hassin 和 Rubinstein Hassin 和 Rubinstein [2001] 首次考虑了这一问题。通过采用二分法，然后贪心地排列这些点，Hassin 和 Rubinstein 达到了 0。5 的近似比。我们注意到，

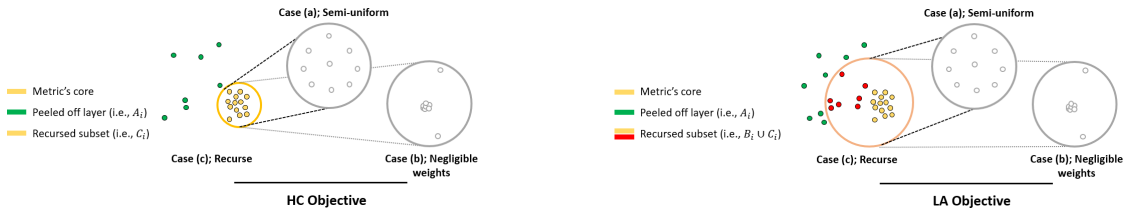


图1：一个递归步骤（情形（c））和两个可能的终止步骤（情形（a）和（b））。黄色点定义了度量的核心。在层次聚类情形中，我们一步就剥离红色和绿色点；而在线性排列情形中，我们必须更加谨慎，只剥离绿色点。

前述结果均适用于任意权重，而我们的主要贡献在于证明：通过假设三角不等式（即基于度量的不相似性权重），我们可以针对这两个目标实现多项式时间近似方案。我们进一步指出，对于基于度量的不相似性权重，特别是 L1度量，Rajagopalan 等人 Rajagopalan 等人 [2021] 通过使用随机割树证明了 0.9 的近似比。

这两个目标最初都是针对其最小化变体进行研究的。最小线性排列问题最早由 Hansen Hansen [1989]研究。Hansen 利用 Leighton 和 Rao Leighton 和 Rao [1999]关于平衡分隔器的研究，将最小线性排列目标近似到 $O(\log^2 n)$ 因子。之后有多项工作改进了这一结果，Charikar 等人 Charikar 等人 [2006]和 Feige 和 Lee Feige 和 Lee [2007] 都利用了 Arora 等人 Arora 等人 [2004] 关于半定规划舍入的新颖研究，并将其与 Rao 和 Richa Rao and Reicha [1998] 的舍入算法相结合，从而证明了一个 $O(\sqrt{\log n} \log \log n)$ 近似结果。关于这类相关目标的更多内容，可参见 Even 等人 [1995], Rao 和 Richa [1998], Seymour [1995],Ravi 等人 [1991]。另一方面，如前所述，最小层次聚类问题由 Dasgupta Dasgupta [2016] 提出，并得到了广泛研究（例如，参见 Dasgupta [2016], Cohen-Addad 等人[2018], Charikar 和 Chatziafratis [2017],Charikar 等人 [2019a], Ahmadian 等人 [2019], Alon 等人 [2020],Vainstein 等人 [2021]）。

与我们工作最相关的是 de la Vega 和 Kenyon 的工作 de la Vega 和 Kenyon [1998]。在他们的工作中，他们针对给定度量的最大割问题提供了一个多项式时间近似方案。该算法首先创建一个克隆图（其中每个原始顶点被克隆若干次，克隆次数取决于其在原度量中的出边权重），使其具有稠密的性质。随后，他们通过应用 de la Vega 和 Karpinski 的算法 de la Vega 和 Karpinski [1998]来解决这一新图中的问题。对于我们的目标（层次聚类和线性排列）而言，这种方法似乎会失败——具体而言，这是因为我们的目标会考虑每个诱导割中的节点数量，而克隆图会增加节点数量，进而增加我们的目标值。因此，对于我们所考虑的目标类型，需要采用更复杂的迭代剥离过程（随后使用更合适的算法终止该过程，这些算法利用了 Goldreich 等人的一般图划分算法 Goldreich 等人 [1998]）。值得一提的是，针对与稠密实例密切相关的目标，也已有广泛研究（例如参见 Kenyon-Mathieu 和 Schudy [2007], Arora 等人 [1999],Karpinski 和 Schudy [2009]）。然而，这类方法似乎仍不够理想，因为我们所考虑的度量不一定是稠密的。

2 多层剥离框架

在定义我们的算法之前，我们需要以下定义。

定义 2.1. 令 $G = (V, w)$ 表示一个度量，$U \subset V$ 表示其节点的一个子集。引入以下记号：(1) 令 $\mathrm{D}_U = \max_{i,j \in U} w_{i,j}$ 表示 U 的**直径**，(2) 令 $\mathrm{W}_U = \sum_{i,j \in U} w_{i,j}$ 表示 U 的

权重之和，(3) 令 $\mathbf{n_U} = |U|$ 表示 U 的 **大小** 以及 (4) 令 $\rho_U = \frac{W_U}{n_U^2 D_U}$ 表示 U 的 **加权密度**[1]。

我们的所有算法都将使用以下简单而实用的结构引理。该引理指出，对于低密度实例，存在一个直径较小的大型节点簇。证明见附录。

引理 2.2. 对于任意度量 $G = (V, w)$，存在一个集合 $U \subset V$，使得 $D_U \leq 4D_V\sqrt{\rho_V}$ 且 $n_U \geq n_V(1 - \sqrt{\rho_V})$。

定义 2.3. 给定一个度量 $G = (V, w)$ 我们将由引理 2.2 保证的 **度量的核心**记为 $U \subset V$。

请注意，只需通过穷举即可找到核心（核心不一定唯一，但我们的算法会任意选择一个）。

在本文中，我们考虑不同的基于度量的目标。为了解决这些目标，我们采用相同的策略：如果实例的加权密度足够高，就应用适用于这类实例的算法。否则，算法会检测度量的核心（这是一个直径较小、包含几乎所有节点的子集），并剥离（随后嵌入）一层远离核心的数据点。然后，算法会处理核心；如果核心足够小（按内部权重衡量），我们就任意嵌入核心并停止。否则，我们递归处理核心。针对这两个目标（LA 和 HC）的算法都将遵循算法 1 中定义的相同结构。

算法 1： 通用算法

若该实例的权重足够密集，**则** // 情形 (a)
  使用 ALG_{d-w} 求解。
else
  令 C 表示度量的核心（定义见2.3）。
  适当地定义要剥离的层 $A \subset V \setminus C$。
  嵌入 A。
  若$W_{V\setminus A}$ 可忽略，**则**任意嵌入 $V \setminus A$ 并返回。 // 情形 (b)
  否则继续对 $V \setminus A$递归处理 // 情况 (c)

我们将算法可能终止的不同情形称为 (a) 和 (b)，将递归步骤称为情形 (c)。此外，我们用 ALG_{d-w} 表示一种辅助算法，它将处理权重足够密集的实例。（这些算法会根据不同的目标而有所不同。）

今后，给定算法 ALG 和度量 G，我们用 $ALG(G)$ 表示该算法返回的嵌入。需要指出的是，在上下文明确时，我们会沿用这一记法，用 $ALG(G)$ 表示该嵌入在相应目标下的取值。等价地，我们将用术语 $OPT(G)$ 表示最优嵌入。

我们的不同算法将以类似的方式定义，因此其分析也将类似。因此，我们引入一种通用方案来分析这类算法。令 k 表示我们的算法执行的递归调用次数。此外，令 G_i 表示算法在步骤 i 中针对 $i = 0, 1, \ldots, k$ 所处理的实例。（即，$G = G_0$ 和 $ALG(G_k)$ 不执行递归步骤，这意味着它以情形 (a) 或 (b) 终止。）我们首先注意到，应用一个简单的平均论证可以得到以下有用的观察结果。

[1]通常，密度是相对于 $\binom{n}{2}$ 定义的。为便于表述，我们选择相对于 n^2 来定义密度；使用前一种定义时，证明仍然相同。

观察 2.4。 如果存在 α_i、 β_i、 $\gamma_i > 0$，使得 $ALG(G_i) \geq \alpha_i + ALG(G_{i+1})$ 且 $OPT(G_i) \leq \beta_i + \gamma_i OPT(G_{i+1})$ 对所有 $i = 0$、…、 $k-1$ 都成立，则

$$\frac{ALG(G)}{OPT(G)} \geq \frac{\sum_{i=0}^{k-1} \alpha_i + ALG(G_k)}{\sum_{i=0}^{k-1} \left(\beta_i \Pi_{j=0}^{i-1} \gamma_j\right) + (\Pi_{i=0}^{k-1} \gamma_i) OPT(G_k)} \geq \min\{\min_i\{\frac{\alpha_i}{\beta_i \Pi_{j=0}^{i-1} \gamma_j}\}, \frac{ALG(G_k)}{(\Pi_{i=0}^{k-1} \gamma_i) OPT(G_k)}\}.$$

我们，为了分析给定的算法，只需设置 α_i、 β_i 和 γ_i 的值，然后进一步分析 $\frac{ALG(G_k)}{OPT(G_k)}$ 在不同终止情形（情形 (a) 和 (b) ）下的近似比。

3 符号与预备知识

我们引入以下记号，以便后续阐述。

定义 3.1。给定度量 $G = (V, w)$、LA 目标的一个解 SOL(G) 以及互不相交的集合 $A, B \subset V$，我们定义：$\text{SOL(G)}|_A = \sum_{i,j \in A} w_{i,j} y_{i,j}$ 以及 $\text{SOL(G)}|_{A,B} = \sum_{i \in A, j \in B} w_{i,j} y_{i,j}$。对于 HC 目标，通过将 $y_{i,j}$ 替换为 $|T_{i,j}|$，可对称地定义这些记号。

我们将使用属于以下算法类别的算法。

定义 3.2. 如果对于任意 $\epsilon > 0$，某算法的期望运行时间为 $f(\frac{1}{\epsilon}) n^{O(1)}$，且其解的值与最优解的值相差不超过因子 $1 - \epsilon$，则该算法称为高效多项式时间随机近似方案（EPRAS）。

我们将频繁使用以下（简单）观察，因此在此列出。

观察 3.3. 给定数值 $\alpha_i \geq 0, \alpha \in (0, \frac{1}{(k+1)})$ 和 $k \in \mathbb{N}$，有：***(1)*** $\Pi_i (1 - \alpha_i) \geq 1 - \sum_i \alpha_i$，***(2)*** $1 + k\alpha < \frac{1}{1-k\alpha} < 1 + (k+1)\alpha$，以及***(3)*** $1 + k\alpha < e^{k\alpha} < 1 + (k+1)\alpha$。

以下事实将在后续证明中发挥作用，因此在此列出。

事实 3.4。 给定一个度量 G，若在 LA 目标下的最优线性排列为 $OPT_{LA}(G)$，且在 HC 目标下的最优层次聚类为 $OPT_{HC}(G)$，则有 $OPT_{LA}(G) \geq \frac{1}{3} n \sum_{i,j} w_{i,j} y_{i,j}$ 以及 $OPT_{HC}(G) \geq \frac{2}{3} n \sum_{i,j} w_{i,j} |T_{i,j}|$。

我们注意到，事实 3.4 中的层次聚类部分已被文献广泛采用（例如，参见 Cohen- Addad 等人 [2018]中的证明）。事实 3.4 中的线性排列部分见于 Hassin 和 Rubinstein [2001]。最后，在层次聚类一节中，我们会用到“梯子”HC 树。下面给出其定义。

定义 3.5。 我们将“梯子”定义为一种 HC 树：在每次切分（或每个内部节点）时，都会将一个单独的数据点与其余部分分开。

4 e 线性排列目标

本节安排如下。我们首先介绍我们的算法（先介绍处理情形 (a) 的算法，随后介绍通用算法）。接着，我们将界定该算法的近似保证（沿用观察2.4的界定方法）。最后，我们将分析该算法的运行时间。

4.1 《算法》的定义

在这里，我们首先将通用算法应用于线性排列问题（我们将其简称为 ALG）。该算法使用一个用于处理情形 (a) 的算法作为子程序。我们将此子程序记为 ALG_{d-w}，并按照 ALG 的定义对其进行定义。

4.1.1 定义 ALG

在这里，我们将通用算法（算法1）应用于线性排列问题。粗略地说，为此我们将要剥离的层定义为所有距离度量的核心“较远”的点的集合， A 。我们还引入一个用于处理密集加权实例的子程序， ALG_{d-w}。

算法 2： 线性排列算法 (ALG)

如果 $\rho \geq \epsilon^6$ **则** 使用 ALG_{d-w} 求解。 // 情形 (a)

else

令 C 表示度量的核心（如引理 2.2所定义）。

令 A 表示与 C的距离为 $\geq \epsilon^2 D_V$ 的所有数据点。

将 A 放在 $V \setminus A$的左侧。任意排列 A 。

如果 $W_{V \setminus A} < \epsilon W_V$ **则**任意排列 $V \setminus A$ 并返回。 // 情形 (b)

否则对 $V \setminus A$递归继续。 // 情况 (c)

集合 $V \setminus \{A \cup C\}$ 将在接下来的证明中频繁使用，因此我们为它单独设定一个记号 .

De定义 4.1. 记 $B = V \setminus \{A \cup C\}$ 其中 A 和 C 的定义与算法 2中相同。

4.1.2 定义 ALG_{d-w}

这里我们将介绍一种算法来处理情形 (a) 类型的实例。在正式定义该算法之前，我们先提供一些直观说明。为此，我们先给出以下定义。

定义 4.2. 考虑 $OPT(G_k)$ 到直线上的嵌入， $[n]$。将 $[n]$划分为 $\frac{1}{\epsilon}$个连续集合，每个集合包含 ϵn个点，并令 P_i^*表示由 $OPT(G_k)$嵌入第 i个连续集合中的点。此外，记由此诱导的度量划分为 $P^* = \{P_i^*\}$。

稍后我们将证明， $OPT(G_k)$的目标值可以由仅基于跨划分集合边所得到的值紧密近似（即任何满足以下条件的 (u, v) 其中 u, v 位于 P^*的不同划分集合中）。尽管$OPT(G_k)$无法通过算法找到，但假设上述结论成立，只需让 ALG_{d-w}猜测划分 P^*即可。事实上，这正是我们将要做的：使用 Goldreich 等人的一般图划分算法。Goldreich 等人 [1998]。

我们将 Goldreich 等人的一般图划分算法记为 Goldreich 等人 [1998] 的$PT(G, \Phi, \epsilon_{err})$。定义 4.10 给出了 Φ和 ϵ_{err}的定义（这些也将由 ALG_{d-w}定义），而定理 4.11 给出了测试器的保证。现在我们可以定义处理权重足够密集实例的算法（算法 3）。

算法 3： 适用于足够密集加权实例的线性排列算法（ALG_{d-w}）

令 $k = \frac{1}{\epsilon}$ 表示划分的大小。

对于 $\{\mu_{j,j'}\}_{j \leq k, j' \leq k, j \neq j'} \subset \{i\epsilon^9 n^2 D_V : i \in \mathbb{N} \wedge i \leq \frac{1}{\epsilon}\}^k$ **执行**

令 $\Phi = \{\epsilon n, \epsilon n\}_{j=1}^k \cup \{\mu_{j,j'}, \mu_{j,j'}\}_{j,j'=1}$。

运行 $PT(G, \Phi, \epsilon_{err} = \epsilon^9)$。令 P 表示输出的划分（若成功）。

令 $\hat{y}$ 表示将 P 依次嵌入到直线上所得的线性排列（并在各划分集合内部任意排列）。

计算 P的值 $\sum_e w_e \hat{y_e}$ 。

返回 $\sum_e w_e \hat{y_e}$ 值最大的划分。

4.2 分析 ALG 的近似比

现在我们已经定义了 ALG，接下来可以分析其近似比。回顾观察2.4 ，只需分析情形 (a)、(b) 的近似比，以及递归步骤造成的总损失（即通过考察 α_i、 β_i 和 γ_i）。

4.2.1 结构引理

回顾一下，我们定义 k 为所使用的递归步骤数， ALG 并且 G_i 是ALG 在递归步骤 i中所应用的实例。进一步回顾，给定 G_i， ALG（G_i） 将该实例划分为A_i， B_i 和 C_i 并且，直观地说，根据引理 2.2 n_{C_i} 包含大多数数据点，而 D_{C_i} 相较于 D_{V_i}较小。

根据 C_i的定义， A_i 可以被视为一组离群点。因此，直观上，将 A_i 与 C_i分开是合理的。为了证明我们的算法的近似比，我们将通过将 A_i 与 C_i分开来表明，事实上，与最优解相比，损失并不大。为此我们将证明， ALG 和 OPT 的值实际上都将大致等于$\frac{1}{2}nW_{A_i,\ C_i}$（这在直觉上是合理的，因为 C_i 的直径较小且包含许多点，而 A_i 是远离该簇的点）。

以下引理讨论两类算法：将 A_i 和 C_i 分开的算法，以及不分开它们的算法。此外，这些引理表明，事实上，根据 A_i 和 C_i的结构性质，若将这两类算法所生成的值限制在由相互权重$W_{A_i,\ C_i}$产生的目标值上，它们的结果近似相等。我们先对分开 A_i 和 C_i的算法所生成的值给出下界。由于篇幅有限，以下证明留至附录。

引理4.3. 给定两个不相交集合 C_i 和 A_i ，以及一个将 A_i 中的所有节点都排在线性排列 y 中 C_i 中所有节点左侧的线性排列，我们可以保证

$$\sum_{c\in C_i, a\in A_i} w_{a,c}y_{a,c} \geq \frac{n_{C_i}}{2}(W_{C_i,A_i} - n_{C_i}n_{A_i}D_{C_i}).$$

由于 C_i 是一个包含大多数数据点的小簇，上述引理可简化为以下推论。

推论 4.4. 给定任意一个将 A_i 中的所有节点都排在线性排列 y 中 C_i 中所有节点左侧的线性排列，我们可以保证

$$\sum_{a\in A_i, c\in C_i} w_{a,c}y_{a,c} \geq \frac{1}{2}nW_{A_i,C_i}(1 - \frac{5\sqrt{\rho}}{\epsilon^2})$$

现在，我们已经为分开 A_i 和 C_i 的算法给出了下界，接下来将为不受此限制的算法给出上界。（注意，我们先处理其中一个不相交集合只有一个数据点的情形，然后再将其推广到两个不相交集合。）

引理 4.5。 给定一个集合 C_i 和一个点 $p \notin C_i$，我们可以保证

$$\sum_{c\in C_i} w_{p,c}y_{p,c} \leq (W_{p,C_i} + n_{C_i}D_{C_i})(n - \frac{n_{C_i}}{2}).$$

现在，我们可以对两组互不相交的点的目标间值给出上界。

引理 4.6. 给定两个不相交集合 C_i 和 A_i 以及任意线性排列 y ，我们可以保证

$$\sum_{c\in C_i, a\in A_i} w_{a,c}y_{a,c} \leq (n - \frac{n_{C_i}}{2})(W_{C_i,A_i} + n_{C_i}n_{A_i}D_{C_i}).$$

由于 C_i 是一个包含大多数数据点的小簇，该引理可简化为以下推论。

推论 4.7。 给定任意线性排列 y，我们可以保证

$$\sum_{a\in A_i, c\in C_i} w_{a,c}y_{a,c} \le \frac{1}{2}nW_{A_i,C_i}(1+\frac{9\sqrt{\rho}}{\epsilon^2}).$$

我们希望证明，ALG 和 OPT 的目标值（以及稍后将定义的一些其他中间值）近似由它们在 $W_{A_i、C_i}$的跨权重上的取值决定。为此，我们首先引入以下结构引理，以帮助解释这一现象。

引理 4.8。 给定一个实例 G 以及由 $ALG(G)$ 定义的集合 A、B 和 C，我们有 $W_A + W_{A,B} \le 2\frac{\sqrt{\rho}}{\epsilon^2}W_{A,C}$。

4.2.2 分析 ALG 的情形 (a) 的近似比

我们首先概述近似比分析。回顾 P^*（定义 4.2）。证明的第一步是说明，与其试图近似 $OPT(G_k)$，不如只考虑相对于 P^*的分区内权重在其上的取值，这样就足够了。更进一步，对于这类权重$w_{u,v}$，它们与 P_i^*和 P_{i+j}^*关联；我们只需假设它们对目标值的贡献（即值 $y_{u,v}$）仅为 $(j-1)\epsilon n$（尽管它可能大到 $(j+1)\epsilon n$）。形式化证明见引理 4.9 （证明推迟至附录）。

接下来，回顾一下，ALG_{d-w} 试图猜测划分 P^*（允许一定的加性误差），并令 P 表示 ALG_{d-w}猜测的划分。注意，如果猜测正确，那么对于 P_i与 P_{i+j}之间任意跨越划分内集合的权重，计入 ALG的目标函数的值至少为 $|P_{i+1}| + \cdots |P_{i+j-1}|$；如果我们还能正确猜出集合大小，那么该值恰好为$(j-1)\epsilon n$（等同于$OPT$的值）。这将在命题4.12中完成。

引理 4.9。 给定集合大小为 ϵn 的平衡线划分，记作 P^*，我们有 e

$$OPT(G_k) \le (1+13\epsilon)\sum_{\substack{1\le i\le k-1\\ 1\le j\le k-i}} W_{P_i^*,P_{i+j}^*}(|P_{i+1}^*| + \cdots + |P_{i+j-1}^*|).$$

在证明命题 4.12 之前，我们先陈述 Goldreich 等人的一般图划分算法的性质。 Goldreich 等人 [1998]。

定义 4.10 (Goldreich 等人 [1998])。 令 $\Phi = \{\lambda_j^{LB}, \lambda_j^{UB}\}_{j=1}^k \cup \{\mu_{j,j'}^{LB}, \mu_{j,j'}^{UB}\}_{j,j'=1}^k$表示一组非负值，使得 $\lambda_j^{LB} \le \lambda_j^{UB}$且 $\mu_{j,j'}^{LB} \le \mu_{j,j'}^{UB}$。我们定义 GP_Φ 为由 n 个顶点构成、具有 k 划分（V_1，……，V_k）且满足以下约束的图 G

$$\forall j: \lambda_j^{LB} \le \frac{|V_j|}{n} \le \lambda_j^{UB}; \quad \forall j,j': \mu_{j,j'}^{LB} \le \frac{W_{V_j,V_{j'}}}{n^2} \le \mu_{j,j'}^{UB}.$$

定理 4.11 (Goldreich 等人 [1998])。 给定输入 $G=(V, w)$，其中 $|V| = n$且 $w: V\times V \to [0,1]$描述图，而 Φ描述所需划分的界限，ϵ_{err}算法 $PT(G, \Phi, \epsilon_{err})$的期望运行时间为[2]

$$\exp\left(\log(\frac{1}{\epsilon_{err}})\cdot(\frac{O(1)}{\epsilon_{err}})^{k+1}\right) + O(\frac{\log\frac{k}{\epsilon_{err}}}{\epsilon_{err}^2})\cdot n.$$

[2]我们指出，原始算法存在错误概率 δ，这会影响运行时间。我们忽略这一误差，并对该算法的期望运行时间进行界定。

此外，若 $G \in \mathcal{GP}_\Phi$ 如定义 *4.10* 所示，则该算法输出的划分满足

- $\forall j:\ \lambda_j^{LB} - \epsilon_{err} \le \frac{|V_j|}{n} \le \lambda_j^{UB} + \epsilon_{err}$,
- $\forall j, j':\ \mu_{j,j'}^{LB} - \epsilon_{err} \le \frac{W_{V_j,V_{j'}}}{n^2} \le \mu_{j,j'}^{UB} + \epsilon_{err}$.

现在我们可以证明命题4.12。

命题 4.12。若 ALG 在情形 *(a)* 下终止，则 $\frac{ALG_{d-w}(G_k)}{OPT(G_k)} - \frac{ALG(G_k)}{OPT(G_k)} \ge 1 - 20\epsilon$.

证明。令 $P = \{P_i\}$ 表示由 $PT(G_k, \Phi, \epsilon_{err})$ 返回的划分，并回想其集合数为 $k = \frac{1}{\epsilon}$ 且 $\epsilon_{err} = \epsilon^9$。首先注意，根据定理 4.11，我们可以保证 $|P_i|$ 相较于 $|P_i^*| = \epsilon n$ 的误差至多为 $|P_i| \ge \epsilon n - \epsilon_{err} n$（这是因为在 Φ 中，我们要求集合大小恰好为 ϵn）。因此

$$ALG_{d-w} \ge \sum_{\substack{1 \le i \le k-1 \\ 1 \le j \le k-i}} W_{P_i, P_{i+j}} (|P_{i+1}| + \cdots + |P_{i+j-1}|) \ge \sum_{\substack{1 \le i \le k-1 \\ 1 \le j \le k-i}} (j-1)(\epsilon n - \epsilon_{err} n) W_{P_i, P_{i+j}}, \tag{1}$$

其中 $W_{P_i,\ P_{i+j}}$ 表示跨越 P_i 与 P_{i+j} 的权重。为便于表述，以下求和时我们将省略下标。

考虑 $W_{P_i、\ P_{i+j}}$ 和 $W_{P_i^*、\ P_{i+j}^*}$ 的割大小之间的差异。它们的差异源于两个误差：(1) PT 算法引入的误差（见定理 4.11）；(2) ALG_{d-w} 为猜测 $OPT(G_k)$ 的划分而引入的误差（见算法 3）。因此，

$$W_{P_i, P_{i+j}} \ge W_{P_i^*, P_{i+j}^*} - \epsilon_{err} n^2 D_V - \epsilon^9 n^2 D_V = W_{P_i^*, P_{i+j}^*} - 2\epsilon^9 n^2 D_V$$

最后一个等号成立是因为 $\epsilon_{err} = \epsilon^9$。将此结果与不等式1 结合可得

$$\begin{aligned} ALG_{d-w} \ge (\epsilon n - \epsilon_{err} n) \cdot \sum (j-1) W_{P_i^*, P_{i+j}^*} - (\epsilon n - \epsilon_{err} n) \cdot 2(\epsilon^9 n^2) D_V \sum (j-1) \ge \\ (\epsilon n - \epsilon_{err} n) \cdot \sum (j-1) W_{P_i^*, P_{i+j}^*} - 2n^3 \epsilon^7 D_V, \end{aligned} \tag{2}$$

其中 最后一个不等式成立是因为 $\epsilon_{err} > 0$、$\sum (j-1)$、$= \sum_{i=1}^{k} \sum_{j=i+1}^{k} (j-1)$ 和 $\le k^3 = \epsilon^{-3}$.

由于我们处于情形 (a)，因此有 $\frac{W}{n^2 D_V} = \rho \ge \epsilon^6$。根据事实3.4 可得 $OPT \ge \frac{1}{3} nW$ 因此，$2n^3 \epsilon^7 D_V$ 可由 $-2n^3 \epsilon^7 D_V \le 2\epsilon n W \le 6\epsilon OPT$ 界定。于是得到 $2n^3 \epsilon^7 D_V \le 6\epsilon OPT(G_k)$。将此结果与不等式2 结合可得

$$ALG_{d-w} \ge (\epsilon n - \epsilon_{err} n) \cdot \sum (j-1) W_{P_i^*, P_{i+j}^*} - 6\epsilon OPT(G_k). \tag{3}$$

另一方面，回想一下，P^* 表示所有集合大小均为 ϵn 的平衡划分。因此，由引理4.9 可得

$$\begin{aligned} OPT(G_k) \le (1 + 13\epsilon) \sum W_{P_i^*, P_{i+j}^*} (|P_{i+1}^*| + \cdots + |P_{i+j-1}^*|) = \\ (1 + 13\epsilon) \sum (j-1)(\epsilon n) W_{P_i^*, P_{i+j}^*} = \epsilon n (1 + 13\epsilon) \cdot \sum (j-1) W_{P_i^*, P_{i+j}^*}. \end{aligned} \tag{4}$$

结合不等式3 和4可得

$$\begin{aligned} ALG_{d-w} \ge \frac{\epsilon n - \epsilon_{err} n}{\epsilon n (1 + 13\epsilon)} OPT(G_k) - 6\epsilon OPT(G_k) = \\ \frac{1 - \epsilon^8}{1 + 13\epsilon} OPT(G_k) - 6\epsilon OPT(G_k) \ge (1 - 20\epsilon) OPT(G_k), \end{aligned}$$

由此完成证明。 □

4.2.3 分析 ALG 情形 (b) 的近似比

利用我们的结构引理，我们将分析在算法于情形 (b) 终止（即 $\rho < \epsilon^6$ 和 $W_{B\cup C} \le \epsilon W_{G_k}$）这一假设下，将 ALG 应用于 G_k 的近似比。完整证明详见附录。

命题 4.13。 如果 ALG 在情形 *(b)* 下终止，则 $\frac{ALG(G_k)}{OPT(G_k)} \ge 1 - 33\epsilon$。

证明概要。 证明遵循以下思路。由于实例的大部分密度集中在度量的核心 C 周围，$OPT(G_k)$ 的目标函数主要来自与 C 相连的权重。由于我们处于情形 (b)，$W_{B\cup C}$ 的权重可以忽略，因此我们将证明，实际上 $OPT(G_k)$ 的目标函数由 $OPT(G_k)|_{A,\ C}$ 定义。此后，我们将证明，对于 $W_{A,\ C}$ 中的权重，最优策略实际上是将 A 放在线的一端，并将 C 放在另一端——幸运的是，这正是 $ALG(G_k)$（近似）所做的——从而近似于 $OPT(G_k)$。□

4.2.4 设置 α_i、β_i 和 γ_i 的值

由于篇幅有限，以下证明将在附录中给出。

命题 4.14。 对于算法应用于 G_i 后定义的 A_i 和 C_i，以及 $\alpha_i = \frac{1}{2}nW_{A,\ C}(1 - \frac{5\sqrt{\rho}}{\epsilon^2})$，有 $ALG(G_i) \ge \alpha_i + ALG(G_{i+1})$。

命题 4.15。 令 $G_i = (V_i, w_i)$ 和 $G_{i+1} = (V_{i+1}, w_{i+1})$ 分别表示由 i 和 $i+1$ 递归步骤定义的实例。此外，令 $\beta_i = \frac{1}{2}nV_iW_{A_i,\ C_i}(1 + \frac{13\sqrt{\rho}}{\epsilon^2})$ 和 $\gamma_i = 1 + 4\sqrt{\rho_i}$。因此，$OPT(G_i) \le \beta_i + \gamma_i OPT(G_{i+1})$。

因此，我们已成功按如下方式设定 α_i、β_i 和 γ_i 的值。

定义 4.16。 我们将值 α_i、β_i 和 γ_i 定义如下

$$\alpha_i = \frac{1}{2}nW_{A_i,C_i}(1 - \frac{5\sqrt{\rho_i}}{\epsilon^2}); \quad \beta_i = \frac{1}{2}nV_iW_{A_i,C_i}(1 + \frac{13\sqrt{\rho_i}}{\epsilon^2}); \quad \gamma_i = 1 + 4\sqrt{\rho_i}. \tag{5}$$

4.2.5 将其整合起来

既然我们已经分析了算法的终止情形（情形 (a) 和 (b)），并且已经设定了 α_i、β_i 和 γ_i 的值，接下来我们将结合这些结果，证明 ALG 的近似比（如观察 3.3 所示）。为此，我们必须界定 $\min\{\ \}$ 和 (G_k) $(\Pi_{i=0}^{k-1}\gamma_i)$ $OPT(G_k)$ 的值。

不过，在此之前，我们首先证明 $\Pi_{j=0}^{i-1}\gamma_j$ 收敛。回想一下，$\gamma_i = 1+4\sqrt{\rho_i}$。下面的引理表明，实例密度（$\rho_i$）以足够快的速率（指数级）增长，从而使 $\Pi_{j=0}^{i-1}\gamma_j$ 收敛。

引理 4.17。 对于所有 $i = 1$、……、$k-1$，我们都能保证 $\rho_{i+1} \ge 4\rho_i$。

证明。 令 V 表示 G_i 的节点集。回顾我们的算法应用于 V 时定义的记号 A、B 和 C（特别地，G_{i+1} 的节点集恰好是 $B \cup C$）。因此，若将 B 中的任意点到其最近的 C 中点的最大距离记为 D_{B-C}，则 $D_{B\cup C} \le 2D_{B-C} + D_C \le 2\epsilon^2 D_V + 4D_V\sqrt{\rho_i}$，其中第一个不等式由三角不等式得出，第二个不等式则是因为 B 定义为所有距离 ϵ^2 以内的点的集合，这些点均距 C 不超过该距离。因此，

$$\rho_{i+1} = \frac{W_{B\cup C}}{n_{B\cup C}^2 \cdot D_{B\cup C}} \ge \frac{W_V}{n_V^2 \cdot D_V}\left(\frac{\epsilon}{2\epsilon^2 + 4\sqrt{\rho_i}}\right) = \rho_i\left(\frac{\epsilon}{2\epsilon^2 + 4\sqrt{\rho_i}}\right), \tag{6}$$

其中等式由 ρ_i 的定义得出，而不等式则是因为 $W_{B\cup C} \geq \epsilon W_V$（这是因为我们处于情况 (c)），$n_{B\cup C} \leq n_V$ 和 $D_{B\cup C} \leq (2\epsilon^2 + 4\sqrt{\rho_i})\, D_V$（如上所述）。由于我们处于情况 (c)，因此可以保证 $\rho_i \leq \epsilon^6$，进而

$$\frac{\epsilon}{2\epsilon^2 + 4\sqrt{\rho_i}} \geq \frac{\epsilon}{2\epsilon^2 + 4\epsilon^3} \geq \frac{1}{3\epsilon}, \tag{7}$$

因为 $\epsilon \leq 10^{-2}$。结合不等式 6 和 7，且由于 $\epsilon < 10^{-2}$ 可得 $\rho_{i+1} \geq \rho_i \left(\frac{\epsilon}{2\epsilon^2+4\sqrt{\rho_i}}\right) \geq \frac{\rho_i}{3\epsilon} \geq 4\rho_i$，由此完成证明。□

现在我们可以证明 $\Pi_{j=0}^{i-1}\gamma_j$ 收敛。

引理 4.18。对于 $\gamma_i = 1 + 4\sqrt{\rho_i}$，我们有 $\Pi_{j=0}^{i-1}\gamma_j \leq 1 + 5\sqrt{\rho_i}$。

证明。注意到 $\Pi_{j=0}^{i-1}(1 + 4\sqrt{\rho_j}) \leq e^{4\cdot\sum_j \sqrt{\rho_j}} \leq e^{4\sqrt{\rho_i}} \leq 1 + 5\sqrt{\rho_i}$，其中，第一个不等式由观察结果 3.3 得出，第二个不等式是因为 $\sqrt{\rho_j}$ 呈指数增长（引理 C.1），第三个不等式再次由观察结果 3.3 结合以下事实得出：$\rho < \epsilon^2$ 且 $\epsilon < 10^{-2}$。□

接下来，我们利用前述引理来界定 $\min_i\{\frac{\alpha_i}{\beta_i\Pi_{j=0}^{i-1}\gamma_j}\}$ 和 $\frac{ALG}{(G_k)\,(\Pi_{i=0}^{k-1}\gamma_i)OPT(G_k)}$.

命题 4.19. 对于 α_i, β_i 和 γ_i 如定义 4.16所述，我们有 $\min_i\{\frac{\alpha_i}{\beta_i\Pi_{j=0}^{i-1}\gamma_j}\} \geq 1 - 23\epsilon$。

证明。首先，我们对 $\frac{\alpha_i}{\beta_i}$ 作界定。根据 α_i 和 β_i 的定义，我们有

$$\frac{\alpha_i}{\beta_i} = \frac{1 - \frac{5\sqrt{\rho_i}}{\epsilon^2}}{1 + \frac{13\sqrt{\rho_i}}{\epsilon^2}} \geq (1 - \frac{5\sqrt{\rho_i}}{\epsilon^2})(1 - \frac{13\sqrt{\rho_i}}{\epsilon^2}) \geq 1 - \frac{18\sqrt{\rho_i}}{\epsilon^2}, \tag{8}$$

其中，第一个不等式由 α_i 和 β_i 的定义得出，其余不等式则由 $\epsilon < 10^2$ 和 $\rho < \epsilon^6$ 得出。

根据引理 4.18，我们可以确定 $\Pi_{j=0}^{i-1}\gamma_j \leq 1 + 5\sqrt{\rho_i}$。结合不等式 8 可得

$$\frac{\alpha_i}{\beta_i\Pi_{j=0}^{i-1}\gamma_j} \geq \frac{1 - \frac{18\sqrt{\rho_i}}{\epsilon^2}}{1 + 5\sqrt{\rho_i}} \geq (1 - \frac{18}{\epsilon^2}\sqrt{\rho_i})(1 - 5\sqrt{\rho_i}) \geq 1 - \frac{23}{\epsilon^2}\sqrt{\rho_i},$$

并且由于 ρ_i 仅增大，而 $\rho_{k-1} \leq \epsilon^6$，我们有 $\min_i\{\frac{\alpha_i}{\beta_i\Pi_{j=0}^{i-1}\gamma_j}\} \geq 1 - \frac{23}{\epsilon^2}\sqrt{\rho_{k-1}} \geq 1 - 23\epsilon$，从而完成证明。□

命题 4.20. 对于 $\gamma_i = 1 + 4\sqrt{\rho_i}$ 我们有 $\frac{ALG}{(G_k)\,(\Pi_{i=0}^{k-1}\gamma_i)OPT(G_k)} \geq 1 - 34\epsilon$。

证明。由命题 4.12 和 4.13 可知，$\frac{ALG}{(G_k)\,OPT(G_k)} \geq 1 - 33\epsilon$。另一方面，由引理 4.18 可知，$\Pi_{i=0}^{k-2}\gamma_i \leq 1 + 5\sqrt{\rho_{k-1}}$。因此，若 $k = 1$，则 $\frac{ALG}{(G_k)\,(\Pi_{i=0}^{k-1}\gamma_i)OPT\,(G_k)} = ALG(G_k)OPT(G_k) \geq 1 - 33\epsilon$。否则，我们有

$$\frac{ALG(G_k)}{(\Pi_{i=0}^{k-1}\gamma_i)OPT(G_k)} \geq \frac{1 - 33\epsilon}{(1 + 4\sqrt{\rho_{k-1}})(1 + 5\sqrt{\rho_{k-1}})} \geq \frac{1 - 33\epsilon}{(1 + 4\epsilon^3)(1 + 5\epsilon^3)} \geq 1 - 34\epsilon,$$

其中第二个不等式成立，是因为 $\rho_{k-1} < \epsilon^6$（因为我们递归到了步骤 k），而后续不等式成立，是因为 $\epsilon < 10^{-3}$ ——由此完成证明。□

最后，我们结合命题 4.19 和4.20，从而界定 ALG 的近似比。

定理 4.21。对于任意度量 G, $\frac{ALG}{(G)\,OPT\,(G)} \geq 1 - 34\epsilon$。

4.3 分析 ALG 的运行时间

考虑 ALG 的定义。我们注意到，在每个递归步骤中，算法都会找到要剥离的层 A，然后进行递归。因此，运行时间由这些递归步骤以及终止情形（即情形 (a) 或情形 (b)）的总和确定。回顾一下，情形 (a) 适用于 ALG_{d-w} 该实例，而情形 (b) 则任意排列该实例。因此，情形 (a) 和 (b) 的界就是 ALG_{d-w} 运行时间的界，该界由引理 4.22 给出（其证明见附录）。

引理 4.22. 给定一个实例 G，$ALG_{d-w}(G)$ 的运行时间至多为 $(\frac{1}{\epsilon^7})^{\frac{1}{\epsilon^2}} \cdot O(n^2)$。

注4.23. 引理 4.17 的一个推论是，递归步骤的数量有上界 $O(\log n)$。证明与引理 C.3 的证明类似，将不等式 $\rho_{i+1} \geq 4\epsilon\sqrt{\rho_i}$ 替换为$\rho_{i+1} \geq 4\sqrt{\rho_i}$ (这由引理4.17可得)。

我们现在可以分析 ALG 的运行时间了。（证明推迟到附录中给出。）

定理 4.24. 算法 ALG 是一个 $EPRAS$（运行时间为 $O(n^2 \log n)$ 加上ALG_{d-w} 的运行时间）。

注 4.25. 我们指出，可以用任何更快的算法替换 ALG_{d-w}，从而提高运行时间，但近似质量会略有下降。

5 e 层次聚类目标

本节的结构如下。我们首先介绍我们的算法（首先是处理情形 (a) 的算法，随后是通用算法）。之后，我们将界定算法的近似保证（遵循观察 2.4 的界定方案）。最后，我们将分析算法的运行时间。

5.1 算法的定义

与线性排列问题类似，我们将首先把通用算法应用于线性排列问题（我们将其简称为 ALG）。该算法使用一种算法作为子程序来处理情形 (a)。我们将这个子程序记为 ALG_{d-w}，并在 ALG 的定义之后对其进行定义。

5.1.1 定义 ALG

这里我们将通用算法（算法 1）应用于层次聚类设置。为此，粗略地说，我们将要剥离的层定义为 A 度量的核心之外的所有点。

算法 4： 层次聚类算法（ALG）

如果 $\rho \geq \epsilon^2$ **那么** 使用 ALG_{d-w}求解该实例。 // 情形 (a)
else
  令 C 表示度量的核心（如引理 2.2所定义）。
  令 $A = V \setminus C$ 表示其余的点。
  将 A 排列成一架（任意的）梯子，并用 T_A表示这棵树。
  如果 $W_C < 16\epsilon \cdot W_V$ **那么** // 情形 (b)
    任意排列 C，并将得到的树记为 T_C。
    将 T_C 的根节点作为 T_A 中最深的内部节点的子节点，然后返回。
  else // 情况 (c)
    对 C 递归处理，并将得到的树记为 T_C。
    将 T_C 的根节点作为 T_A 中最深的内部节点的子节点，然后返回。

注释 5.1. 注意，算法 4 符合通用算法 1 因为 $C = V \setminus A$.

5.1.2 定义 ALG_{d-w}

我们将使用 Vainstein 等人的算法。Vainstein 等人 [2021] 作为 ALG_{d-w}。作为其算法的一部分，他们使用了 Goldreich 等人的一般图划分算法。Goldreich 等人 [1998] 该算法记作 $PT(\cdot)$。由于我们将使用 $PT(\cdot)$ 来为线性排列目标设计自己的算法，因此请读者参阅定义4.10 和定理4.11 ，以深入了解 $PT(\cdot)$ 算法。我们在算法 ALG_{d-w} 5 中重述了 Vainstein 等人所定义的Vainstein 等人 [2021]。

算法 5： 适用于充分密集加权实例的层次聚类算法（ALG_{d-w}）枚举所有具有 $k = \frac{1}{\epsilon}$ 个内部节点的树 T。对于每个这样的 T，执行以下操作：对于 $\{\lambda_i\}_{i\le k} \subset \{i\epsilon^2 n : i \in \mathbb{N} \wedge i \le \frac{3}{\epsilon}\}$，执行以下操作：对于 $\{\mu_{j,j'}\}_{j\le k, j'\le k, j\ne j'} \subset \{i\epsilon^3 n^2 D_V : i \in \mathbb{N} \wedge i \le \frac{9}{\epsilon}\}$，执行以下操作：令 $\Phi = \{\lambda_i, \lambda_i\}_{i=1}^{k} \cup \{\mu_{j,j'}\}_{j,j'=1}^{k}$。运行 $PT(G, \Phi, \epsilon_{err} = \epsilon^3)$。令 P 表示输出的划分（如果运行成功）。根据 T 和 P 计算层次聚类目标值。返回层次聚类目标值最大的划分 P 和树 T。

5.2 分析 ALG 的近似比

既然我们已经定义了 ALG，现在就可以分析它的近似比了。回想一下，根据观察 2.4，只需分析情形 (a)、(b) 的近似比，以及递归步骤产生的总近似损失（即求出 α_i、β_i 和 γ_i）。

5.2.1 分析 ALG 的情形 (a) 的近似比

为了分析 ALG_{d-w} 的近似比，我们首先必须回顾并非全部权重都很小的实例的定义（该定义由 Vainstein 等人给出）。Vainstein 等人 [2021]。

定义 5.2。若存在常数（相对于 n_V）c_0，$c_1 < 1$，使得小于 $c_0 \cdot D_V$ 的权重所占比例至多为 $1 - c_1$，则称度量 G 具有并非全部权重都很小的性质。

以下定理由 Vainstein 等人提出。Vainstein 等人 [2021]。

定理 5.3。对于任意常数 $\xi > 0$ 和任意度量 $G = (V, w)$，若并非所有权重都很小（常数为 c_0 和 c_1），则可以保证 $\frac{ALG_{d-w}(G)}{OPT(G)} \ge 1 - O(\frac{\xi}{c_0 \cdot c_1})$，并且 ALG_{d-w} 的期望运行时间至多为 $f(\frac{1}{\xi}) \cdot n^2$。

将上述定理应用于 $\xi = \epsilon^5$ 我们的度量实例 G_k，可得到命题 5.4（其证明推迟至附录）。

命题 5.4。如果 ALG 在情形 *(a)* 下终止，则 $\frac{ALG_{d-w}(G_k)}{OPT(G_k)} - \frac{ALG(G_k)}{OPT(G_k)} \ge 1 - \epsilon$。

5.2.2 分析 ALG 的情形(b)的近似比

命题 5.5。如果 ALG 在情形 *(b)* 中终止，则 $\frac{ALG}{(G_k)\, OPT(G_k)} \ge 1 - 17\epsilon$。

证明。证明见附录。 □

5.2.3 设置值 α_i、β_i 和 γ_i

由于篇幅有限，以下证明放在附录中。

引理 5.6. 对于算法应用于 G_i 后定义的 A_i 和 C_i，以及 $\alpha_i = n_{V_i}(W_{A_i}+W_{A_i,C_i})(1-\sqrt{\rho_i})$，我们有 $ALG(\overline{G_i}) \geq \alpha_i + ALG(G_{i+1})$。

引理 5.7. 设 $G_i = (V_i, w_i)$ 和 $G_{i+1} = (V_{i+1}, w_{i+1})$ 分别表示由 i 和 $i+1$ 递归步骤定义的实例。此外，令 $\beta_i = n_{V_i}(W_{A_i}+W_{A_i,C_i})$ 和 $\gamma_i = 1+2\sqrt{\rho_i}$。因此，$OPT(\overline{G_i}) \leq \beta_i + \gamma_i OPT(G_{i+1})$。

因此，我们在定义 5.8 中将这些值组合起来。

定义 5.8. 我们定义值 α_i、β_i 和 γ_i 如下

$$\alpha_i = n_{V_i}(W_{A_i} + W_{A_i,C_i})(1-\sqrt{\rho_i}); \quad \beta_i = n_{V_i}(W_{A_i} + W_{A_i,C_i}); \quad \gamma_i = 1+2\sqrt{\rho_i}.$$

5.2.4 综合起来看

既然我们已经分析了算法的终止情形（情形 (a) 和 (b)），并设定了 α_i、β_i 和 γ_i 的值，我们将结合这些结果来证明 ALG 的近似比（如观察2.4所述）。由于篇幅有限，本节的证明将在附录中给出。

命题 5.9。 对于 α_i，β_i 以及 γ_i 如定义 *5.8*中所述，我们有 最小值$_i\{\frac{\alpha_i}{\beta_i \Pi_{j=0}^{i-1}\gamma_j}\} \geq 1-4\epsilon$。

Proposition 5.10. *For* $\gamma_i = 1+2\sqrt{\rho_i}$ *we have* $\frac{ALG(G_k)}{(\Pi_{i=0}^{k-1}\gamma_i)OPT(G_k)} \geq 1-23\epsilon$.

定理 5.11. 对于任意度量 G, $\frac{ALG(G)}{OPT(G)} \geq 1-23\epsilon$.

5.3 分析 ALG 的运行时间

考虑 ALG 的定义。在每个递归步骤中，算法会找出要剥离的层，然后继续递归。因此，运行时间由这些递归步骤的总和以及终止情形（即情形 (a) 或情形 (b)）决定。回想一下，情形 (a) 对实例应用 ALG_{d-w}，而情形 (b) 则任意排列该实例。因此，情形 (a) 和情形 (b) 的界限，只需给出 ALG_{d-w} 运行时间的界限即可，该界限由定理5.3 Vainstein 等人 [2021]给出。在引理C.3 中，我们界定递归步骤的数量，并据此证明定理 5.12 （证明见附录）。

定理 5.12。 该算法 ALG 是一个 *EPRAS*（运行时间为 $O(n^2 \log\log n)$，另加上 ALG_{d-w}的运行时间）。

备注 5.13。 我们指出，可以用任何更快的算法替换 ALG_{d-w} 来缩短运行时间，但近似质量会略有下降。

参考文献

Sara Ahmadian、Vaggos Chatziafratis、Alessandro Epasto、EuiwoongLee、Mohammad Mahdian、Konstantin Makarychev 和 Grigory Yaroslavtsev。《二分并征服：通过最大未切割二分进行层次聚类》。*CoRR*，abs/1912.06983，2019年。

Noga Alon、Yossi Azar 和 Danny Vainstein。层次聚类：0.585 收益近似。载于 Jacob D. Abernethy 和 Shivani Agarwal 编，学习理论会议，*COLT 2020*，*2020*年*7*月*9*日至*12*日，线上活动 *[*奥地利格拉茨*]*，第 125 卷，机器学习研究论文集，第 153–162 页。PMLR，2020年。网址：http://proceedings.mlr.press/v125/alon20b.html。

Sanjeev Arora、David R. Karger 和 Marek Karpinski。针对稠密 NP-hard问题实例的多项式时间近似方案。《计算机系统科学杂志》，58(1):193–210，1999年。doi：10.1006/jcss.1998.1605。网址 https://doi.org/10.1006/jcss.1998.1605。

Sanjeev Arora、Satish Rao 和 Umesh V. Vazirani。扩展图流、几何嵌入与图划分。载于 L´aszl´o Babai（编），第*36*届 *ACM* 计算理论年度研讨会论文集，美国伊利诺伊州芝加哥，*2004*年*6*月*13*日至*16*日，第222–231页。ACM，2004年。doi: 10.1145/1007352.1007355。网址：https://doi.org/10.1145/1007352.1007355。

Kevin Aydin、MohammadHossein Bateni 和 Vahab S. Mirrokni。通过线性嵌入实现分布式平衡划分。《算法》，12(8):162，2019年。doi: 10.3390/a12080162。网址：https://doi.org/10.3390/a12080162。

MohammadHossein Bateni、Soheil Behnezhad、Mahsa Derakhshan、MohammadTaghi Hajiaghayi、Raimon- das Kiveris、Silvio Laanzi 和 Vahab S. Mirrokni。亲和聚类：规模化层次聚类。载于 Isabelle Guyon、Ulrike von Luxburg、Samy Bengio、Hanna M. Wallach、Rob Fergus、S. V. N. Vish- wanathan 和 Roman Garne（编），神经信息处理系统进展 *30*：*2017*年神经信息处理系统年度会议，*2017*年*12*月*4*日至*9*日，美国加利福尼亚州长滩，第6864–6874页，2017年。网址：https://proceedings.neurips.cc/paper/2017/hash/2e1b24a664f5e9c18f407b2f9c73e821-Abstract.html。

Moses Charikar 和 Vaggos Chatziafratis。通过最稀疏割和扩散度量实现近似层次聚类。在第二十八届 *ACM-SIAM*离散算法研讨会论文集，*SODA 2017*，西班牙巴塞罗那，*Porta Fira* 酒店，*1*月*16*日至*19*日，第841–854页，2017年。

Moses Charikar、Mohammad Taghi Hajiaghayi、Howard J. Karlo 和 Satish Rao。*l*用于顶点排序问题的 22 种扩散度量。在第十七届 *ACM-SIAM*离散算法研讨会论文集，*SODA 2006*，美国佛罗里达州迈阿密，*2006*年*1*月*22*日至*26*日，第1018–1027页。ACM Press，2006年。网址 http://dl.acm.org/citation.cfm?id=1109557.1109670。

Moses Charikar、Vaggos Chatziafratis 和 Rad Niazadeh。层次聚类优于平均链接。在第三十届 *ACM-SIAM*离散算法研讨会论文集，*SODA 2019*，圣迭戈，加利福尼亚州，美国，*1*月*6*日至*9*日，第2291–2304页，2019年。

Moses Charikar、Vaggos Chatziafratis、Rad Niazadeh 和 Grigory Yaroslavtsev。欧几里得数据的层次聚类。载于 第*22*届国际人工智能与统计会议（*AISTATS 2019*），*2019*年*4*月*16*日至*18*日，日本冲绳那霸，第2721–2730页，2019b年。网址：http://proceedings.mlr.press/v89/charikar19a.html。

Gui Citovsky、Giulia DeSalvo、Claudio Gentile、Lazaros Karydas、Anand Rajagopalan、Afshin Ros- tamizadeh 和 SanjivKumar。规模化批量主动学习。*CoRR*，abs/2107.14263，2021年。网址：https://arxiv.org/abs/2107.14263。

Vincent Cohen-Addad、Varun Kanade、Frederik Mallmann-Trenn 和 Claire Mathieu。层次聚类：目标函数与算法。载于第二十九届 *ACM-SIAM* 离散算法年度研讨会论文集（*SODA 2018*），美国路易斯安那州新奥尔良，*2018*年*1*月*7*日至*10*日，第378–397页，2018年。

Sanjoy Dasgupta。基于相似度的层次聚类的代价函数。载于 第*48*届 *ACM SIGACT* 计算理论年会论文集，*STOC 2016*，剑桥，美国马萨诸塞州，*2016*年*6*月*18*日至*21*日，第118–127页，2016年。

Wenceslas Fernandez de la Vega 和 Marek Karpinski。MAX-CUT 稠密加权实例的多项式时间近似。 计算复杂性电子 *Colloquium*，（64），1998年。网址 https://eccc.weizmann.ac.il/eccc-reports/1998/TR98-064/index.html。

Wenceslas Fernandez dela Vega 和 Claire Kenyon。度量 MAX-CUT 的随机化近似方案。在第 *39* 届计算机科学基础年会（*FOCS '98*），*1998*年*11*月*8*日至*11*日，美国加利福尼亚州帕洛阿尔托，第 468–471 页。IEEE 计算机学会，1998年。doi：10.1109/SFCS.1998.743497。网址：https://doi.org/10.1109/SFCS.1998.743497。

Guy Even、Joseph Naor、Satish Rao 和 Baruch Schieber。基于扩散度量的分治近似算法（扩展摘要）。载于第*36*届计算机科学基础年会，威斯康星州密尔沃基，美国，*1995*年*10*月*23*日至*25*日，第62–71页。IEEE 计算机学会，1995年。doi: 10.1109/ SFCS.1995.492463。网址 https://doi.org/10.1109/SFCS.1995.492463。

Uriel Feige 和 James R. Lee。最小线性排列问题的改进近似比。*Inf. Process. Lett.*，101(1):26–29，2007年。doi: 10.1016/j.ipl.2006.07.009。URLhttps://doi.org/10.1016/j.ipl.2006.07.009。

Oded Goldreich、Sha Goldwasser 和 Dana Ron。性质测试及其与学习和近似的联系。*J.ACM*，45(4):653–750，1998年。

Mark D. Hansen。平面上几何嵌入的近似算法及其在并行处理问题中的应用（扩展摘要）。载于第*30*届计算机科学基础年会，北卡罗来纳州研究三角园，美国，*1989*年*10*月*30*日至*11*月*1*日，第604–609页。IEEE计算机学会，1989年。doi: 10.1109/SFCS.1989.63542。URLhttps://doi.org/10.1109/SFCS.1989.63542。

Refael Hassin 和 Shlomi Rubinstein。最大线性排列的近似算法。信息处理快报，80(4):171–177，2001年。doi: 10.1016/S0020-0190(01)00159-4。网址 https://doi.org/10.1016/S0020-0190(01)00159-4。

Marek Karpinski 和 Warren Schudy。用于 Gale-Berlekamp 博弈及相关最小化问题的线性时间近似方案。载于 Michael Mitzenmacher 编，第*41*届 *ACM* 计算理论年度研讨会论文集，*STOC 2009*，美国马里兰州贝塞斯达，*2009*年*5*月*31*日至*6*月*2*日，第313–322页。ACM，2009年。doi: 10.1145/1536414.1536458。网址https://doi.org/10.1145/1536414.1536458。

Claire Kenyon-Mathieu 和 Warren Schudy。《如何以较少错误进行排序》。载于 David S. Johnson 和 Uriel Feige 编，第*39*届 *ACM* 计算理论年度研讨会论文集，圣迭戈，加利福尼亚州，美国，*2007*年*6*月*11*日至*13*日，第95–103页。ACM，2007年。doi：10.1145/1250790.1250806。网址：https://doi.org/10.1145/1250790.1250806。

Frank omson Leighton 和 Satish Rao。多商品最大流最小割定理及其在设计近似算法中的应用。《*ACM*学报》，46(6):787–832，1999年。doi：10.1145/331524.331526。网址：https://doi.org/10.1145/331524.331526。

Benjamin Moseley 和 Joshua Wang。《层次聚类的近似界：平均链接法、二分 k 均值和局部搜索》。载于 神经信息处理系统进展 *30*：*2017*年神经信息处理系统年度会议，*2017*年*12*月*4*日至*9*日，美国加利福尼亚州长滩，第 3094–3103 页，2017年。

Stanislav Naumov、Grigory Yaroslavtsev 和 Dmitrii Avdiukhin。基于目标的深度嵌入向量层次聚类。载于第三十五届人工智能 *AAAI* 大会（*AAAI 2021*）、第三十三届人工智能创新应用大会（*IAAI 2021*）、第十一届人工智能教育进展研讨会（*EAAI 2021*），线上活动，*2021*年*2*月*2*日至*9*日，第 9055–9063 页。AAAI出版社，2021年。网址 https://ojs.aaai.org/index.php/AAAI/article/view/17094。

Anand Rajagopalan、Fabio Vitale、Danny Vainstein、Gui Citovsky、Cecilia M. Procopiuc 和 Claudio Gen- tile。《数据流的层次聚类：可扩展算法与近似保证》。载于 Marina Meila 和 Tong Zhang 编，第*38*届国际机器学习会议论文集（*ICML 2021*），*2021*年*7*月*18*日至*24*日，线上活动，第139卷，机器学习研究论文集，第8799–8809页。PMLR，2021年。网址：http://proceedings.mlr.press/v139/rajagopalan21a.html。

Satish Rao 和 Andr´ea W. Richa。关于若干排序问题的新近似技术。载于 HowardJ. Karlo 编，第九届 *ACM-SIAM* 离散算法研讨会论文集，*1998*年*1*月*25*日至*27*日，美国加利福尼亚州旧金山，第211–218页。ACM/SIAM，1998年。网址 http://dl.acm.org/citation.cfm?id=314613.314703。

R. Ravi、Ajit Agrawal 和 Philip N. Klein。近似求解排序问题：单处理器调度与区间图补全。载于 Javier Leach Albert、Burkhard Monien 和 Mario Rodr´ıguez-Artalejo 编，《自动机、语言与编程》第*18*届国际专题讨论会（*ICALP91*），西班牙马德里，*1991*年*7*月*8*日至*12*日，论文集，第计算机科学讲义第510卷，第751–762页。Springer，1991年。doi：10.1007/3-540-54233-7\180。网址https://doi.org/10.1007/3-540-54233-7180。

Paul D. Seymour。分数式打包有向回路。组合学，15(2):281–288，1995年。doi：10.1007/BF01200760。网址https://doi.org/10.1007/BF01200760。

Baris Sumengen、Anand Rajagopalan、Gui Citovsky、David Simcha、Olivier Bachem、Pradipta Mitra、Sam Blasiak、Mason Liang 和 Sanjiv Kumar。将层次凝聚聚类扩展至十亿规模数据集。*CoRR*，abs/2105.11653，2021年。网址：https://arxiv.org/abs/2105.11653。

Danny Vainstein、Vaggos Chatziafratis、Gui Citovsky、Anand Rajagopalan、Mohammad Mahdian 和 Yossi Azar。通过草图进行层次聚类与层次相关聚类。载于 Arindam Banerjee 和 Kenji Fukumizu 编，第*24*届人工智能与统计国际会议，*AISTATS 2021*，*2021*年*4*月*13*日至*15*日，线上活动，第130卷，机器学习研究论文集，第559–567页。PMLR，2021年。网址：http://proceedings.mlr.press/v130/vainstein21a.html。

第 2节的延后证明

引理 *2.2* 的证明。 对于每个节点 v 令 W_v 表示与 v相连的权重之和。令 U_v 表示距离 $2D_V\sqrt{\rho_V}$不超过 v的节点集合。我们证明存在 v使得 $|U_v| \geq n_V(1-\sqrt{\rho_V})$，由此完成证明。

A反设结论不成立。那么，对于每个节点 v，我们都有

$$W_v > (\sqrt{\rho_V} n_V)(2D_V\sqrt{\rho_V}) = 2n_V D_V \rho_V$$

（因为至少有 $n_V\sqrt{\rho_V}$ 个节点与 v 的距离为 $\geq 2D_V\sqrt{\rho_V}$）。对所有 v 求和得到 $2W_V = \sum_v W_v > 2n_V^2 D_V \rho_V = 2W_V$，这就产生了矛盾。□

B 第 4 节的延后证明

引理的证明 4.3。 我们首先注意到，对于任意 $a \in A_i$，都有

$$\sum_{c \in C_i} w_{a,c} y_{a,c} \geq (\min_{c \in C_i}\{w_{a,c}\}) \sum_{c \in C_i} y_{a,c} \geq$$
$$(\min_{c \in C_i}\{w_{a,c}\})(1+2+3+\cdots+n_{C_i}) \geq (\min_{c \in C_i}\{w_{a,c}\})\frac{n_{C_i}^2}{2}, \tag{9}$$

其中第二个 个不等式成立，是因为 y 将所有点都置于 A_i 中所有点的左侧 C_i.

根据三角不等式，对于任意点 $c \in C_i$，我们有 $\min_{c \in C_i}\{w_{a,c}\} + D_{C_i} \geq w_{a,c}$，因此

$$(\min_{c \in C_i}\{w_{a,c}\}) \cdot n_{C_i} \geq \sum_{c \in C_i} (w_{a,c} - D_{C_i}). \tag{10}$$

因此，对所有 $a \in A_i$ 求和

$$\sum_{c \in C_i, a \in A_i} w_{a,c} y_{a,c} = \sum_{a \in A_i} \sum_{c \in C_i} w_{a,c} y_{a,c} \geq$$
$$\sum_{a \in A_i} (\min_{c \in C_i}\{w_{a,c}\})\frac{n_{C_i}^2}{2} \geq \sum_{a \in A_i} \Big(\sum_{c \in C_i} (w_{a,c} - D_{C_i})\frac{n_{C_i}}{2} \Big) =$$
$$\frac{n_{C_i}}{2}(W_{C_i,A_i} - n_{C_i} n_{A_i} D_{C_i}),$$

其中第一个不等式由不等式 9 得出，第二个不等式由不等式 10得出——至此证明完毕。

□

推论的证明 4.4。 我们从引理 4.3开始

$$\sum_{a \in A, c \in C} w_{a,c} y_{a,c} \geq \frac{n_C}{2}(W_{A,C} - n_A n_C D_C). \tag{11}$$

回顾根据 A 和 C 的定义，集合 A 与 C 之间的所有权重至少为 $\epsilon^2 D_V$，因此 $W_{A,C} \geq n_A n_C D_V \epsilon^2$。此外，根据引理 2.2，我们可以保证 $D_C \leq 4\sqrt{\rho} D_V$，因此

$$n_A n_C D_C \leq n_A n_C 4\sqrt{\rho} D_V \leq \frac{4\sqrt{\rho}}{\epsilon^2} W_{A,C}. \tag{12}$$

将不等式 11 和 12 与 $n_C \geq (1-\sqrt{\rho})n$ 这一事实结合可得

$$\sum_{a \in A, c \in C} w_{a,c} y_{a,c} \geq \frac{n_C}{2}(W_{A,C} - n_A n_C D_C) \geq$$
$$\frac{n_C}{2} W_{A,C}(1 - \frac{4\sqrt{\rho}}{\epsilon^2}) \geq \frac{1}{2} n W_{A,C}(1-\sqrt{\rho})(1 - \frac{4\sqrt{\rho}}{\epsilon^2}).$$

最后，由于 $\epsilon < 1$我们有

$$\sum_{a \in A, c \in C} w_{a,c} y_{a,c} \geq \frac{1}{2} n W_{A,C}(1-\sqrt{\rho})(1 - \frac{4\sqrt{\rho}}{\epsilon^2}) \geq \frac{1}{2} n W_{A,C}(1 - \frac{5\sqrt{\rho}}{\epsilon^2}),$$

从而完成证明。

□

引理证明*4.5*。 我们首先观察到

$$\sum_{c\in C_i} w_{p,c}y_{p,c} \le (\max_{c\in C_i}\{w_{p,c}\}) \sum_{c\in C_i} y_{p,c} \le$$
$$(\max_{c\in C_i}\{w_{p,c}\})(\sum_{i=0}^{n_{C_i}-1}(n-n_{C_i}+i)) = (\max_{c\in C_i}\{w_{p,c}\})(n\cdot n_{C_i} - \frac{n_{C_i}(n_{C_i}+1)}{2}) \le \tag{13}$$
$$(\max_{c\in C_i}\{w_{p,c}\})(n\cdot n_{C_i} - \frac{n_{C_i}^2}{2}),$$

其中，第二个不等式源于这样一个事实：要使 $\sum_{c\in C_i} y_{p,c}$ （即所有权重均为 1 时的目标间值）最大，必须将 p 放在线性排列的一端，并将 C_i 放在另一端。另一方面，对于每个 $c \in C_i$，根据三角不等式，我们有 $w_{p,c}+D_m \ge (\max_{c\in C_i}\{w_{p,c}\})$，因此，

$$n_{C_i}\cdot(\max_{c\in C_i}\{w_{p,c}\}) \le \sum_{c\in C_i}(w_{p,c}+D_{C_i}). \tag{14}$$

结合不等式13和14可得

$$\sum_{c\in C_i} w_{p,c}y_{p,c} \le (\max_{c\in C_i}\{w_{p,c}\})(n\cdot n_{C_i} - \frac{n_{C_i}^2}{2}) \le$$
$$\sum_{c\in C_i}(w_{p,c}+D_{C_i})(n-\frac{n_{C_i}}{2}) = (W_{p,C_i}+n_{C_i}D_{C_i})(n-\frac{n_{C_i}}{2}),$$

由此得证。

□

引理的证明 *4.6*。 对所有点应用引理 4.5 即可得出 $a \in A_i$。 □

推论的证明 *4.7*。 我们从引理 4.6 开始

$$\sum_{a\in A, c\in C} w_{a,c}y_{a,c} \le (n-\frac{n_C}{2})(W_{A,C}+n_An_CD_C). \tag{15}$$

回顾一下，根据 A 和 C 的定义，集合 A 与 C 之间的所有权重至少为 $\epsilon^2 D_V$，因此 $W_{A,C} \ge n_An_CD_V\epsilon^2$。此外，根据引理 2.2，可以保证 $D_C \le 4\sqrt{\rho}D_V$，因此

$$n_An_CD_C \le n_An_C4\sqrt{\rho}D_V \le \frac{4\sqrt{\rho}}{\epsilon^2}W_{A,C}. \tag{16}$$

合并不等式15 和16，并结合 $n_C \ge (1-\sqrt{\rho})n$ 可得

$$\sum_{a\in A, c\in C} w_{a,c}y_{a,c} \le (n-\frac{n_C}{2})(W_{A,C}+n_An_CD_C) \le$$
$$(n-\frac{n_C}{2})W_{A,C}(1+\frac{4\sqrt{\rho}}{\epsilon^2}) \le \frac{1}{2}nW_{A,C}(1+\sqrt{\rho})(1+\frac{4\sqrt{\rho}}{\epsilon^2}) \le \frac{1}{2}nW_{A,C}(1+\frac{9\sqrt{\rho}}{\epsilon^2}).$$

最后一个不等式成立，因为 $\rho < 1$ 和 $\epsilon < 1$ ——由此证明得证。 □

引理的证明 4.9。 我们首先注意到 $OPT(G_k)$可以改写为

$$OPT(G_k) = \sum_{\substack{1\le i\le k-1\\ 1\le j\le k-i}} OPT(G_k)|_{P_i^*,P_{i+j}^*} + \sum_{1\le i\le k} OPT(G_k)|_{P_i^*}.$$

为便于表述，以下求和时省略下标。由于$|P_i^*| \le \epsilon n$我们有 $\sum_i OPT(G_k)|_{P_i^*} \le \sum_i \epsilon n W_{P_i^*} \le \epsilon n W$ 结合事实 3.4可知 $\sum_i OPT(G_k)|_{P_i^*} \le 3\epsilon OPT(G_k)$。因此

$$OPT(G_k) = \sum OPT(G_k)|_{P_i^*,P_{i+j}^*} + \sum OPT(G_k)|_{P_i^*} \le$$
$$\sum OPT(G_k)|_{P_i^*,P_{i+j}^*} + 3\epsilon OPT(G_k) \Rightarrow OPT(G_k) \le \frac{1}{1-3\epsilon}\sum OPT(G_k)|_{P_i^*,P_{i+j}^*}.$$

另一方面，任何跨越 P_i^*和 P_{i+j}^*的权重对目标函数的贡献至多为 $(j+1)\epsilon n$，因此 $OPT(G_k)|_{P_i^*,P_{i+j}^*} \le W_{P_i^*,P_{i+j}^*}((j+1)\epsilon n) = W_{P_i^*,P_{i+j}^*}(|P_i^*|+\cdots+|P_{i+j}^*|)$。综合起来可得

$$OPT(G_k) \le \frac{1}{1-3\epsilon}\sum OPT(G_k)|_{P_i^*,P_{i+j}^*} \le (1+4\epsilon)\sum W_{P_i^*,P_{i+j}^*}(|P_i^*|+\cdots+|P_{i+j}^*|), \tag{17}$$

其中，最后一个不等式成立是因为 $\epsilon < \frac{1}{12}$ —— $1+4\epsilon\ \sum W_{P_i^*\ P_{i+j}^*}\ |P_i^*|+$。为了完成证明，我们对 $|P_{i+j}^*|$ $|P_i^*|+|P_{i+j}^*| = 2\epsilon n$ $\sum W_{P_i^*\ P_{i+j}^*} \le W$ 值 ( ) ,( ) 进行界定。回顾可知，。进一步注意到 ., 这显然成立，因为每个权重至多被计数一次。因此。

$$(1+4\epsilon)\sum W_{P_i^*,P_{i+j}^*}(|P_i^*|+|P_{i+j}^*|) = (1+4\epsilon)2\epsilon n\sum W_{P_i^*,P_{i+j}^*} \le (1+4\epsilon)2\epsilon nW. \tag{18}$$

<escape>为完成证明，我们利用以下事实： $OPT(G_k) \ge \frac{1}{3}nW$(参见事实 3.4) 以及 ge</escape> t

$$(1+4\epsilon)2\epsilon nW \le (1+4\epsilon)6\epsilon OPT(G_k) \le 7\epsilon OPT(G_k), \tag{19}$$

对于 $\epsilon < 10^{-2}$。结合上述不等式17、 18 和19 可得

$$OPT(G_k) \le (1+4\epsilon)\sum W_{P_i^*,P_{i+j}^*}(|P_{i+1}^*|+\cdots+|P_{i+j-1}^*|) + 7\epsilon OPT(G_k) \Rightarrow$$
$$OPT(G_k) \le \frac{1+4\epsilon}{1-7\epsilon}\sum W_{P_i^*,P_{i+j}^*}(|P_{i+1}^*|+\cdots+|P_{i+j-1}^*|) \le$$
$$(1+13\epsilon)\sum W_{P_i^*,P_{i+j}^*}(|P_{i+1}^*|+\cdots+|P_{i+j-1}^*|),$$

对于 $\epsilon < 10^{-2}$ ——由此完成证明。 □

命题证明 4.13。 我们的证明将包含三个步骤——（1）我们将证明 $OPT(G_k) \le (1+16\epsilon)\ OPT(G_k)|_{A,\ C}$ 并且（2）我们将证明 $OPT(G_k)|_{A,\ C} \le (1+\frac{16\sqrt{\rho}}{\epsilon^2})\ ALG(G_k)|_{A,\ C}$。在步骤（3）中，我们结合这些观察结果并证明该命题。

1. **$OPT(G_k) \le (1+16\epsilon)\ OPT(G_k)|_{A,\ C}$**：为了证明这一点，我们首先证明$W \le 6\epsilon W_{A,\ C}$（并且由于实例的大部分权重都包含在$W_{A,\ C}$中， $OPT(G_k)$的大部分价值将来自这些权重）。

事实上，由于我们处于情形 (b)，因此有 $W_{B\cup C} \le \epsilon W_V$

$$W_{B\cup C} \le \frac{\epsilon}{1-\epsilon}(W_A + W_{A,B} + W_{A,C}) \le 2\epsilon(W_A + W_{A,B} + W_{A,C}),$$

其中第一个不等式成立是因为 $W_V = W_{B\cup C} + W_A + W_{A,\ B} + W_{A,\ C}$ 第二个不等式则是因为 $\epsilon < 10^{-3}$。另一方面，根据引理 4.8，我们有 $W_A + W_{A,\ B} \leq 2\frac{\sqrt{\rho}}{\epsilon^2} W_{A,\ C}$。因此，

$$\begin{gathered} W_A + W_{A,B} + W_{B\cup C} \leq (1+2\epsilon)W_A + (1+2\epsilon)W_{A,B} + 2\epsilon W_{A,C} \leq \\ (1+2\epsilon)(\frac{2\sqrt{\rho}}{\epsilon^2})W_{A,C} + 2\epsilon W_{A,C} \leq \Big((1+2\epsilon)(2\epsilon) + 2\epsilon\Big)W_{A,C} \leq 5\epsilon W_{A,C}, \end{gathered} \tag{20}$$

其中，最后一个不等式成立是因为我们处于情形 (b)，因此 $\rho \leq \epsilon^6$，且由于 $\epsilon < \frac{1}{4}$。综合以上各项可得

$$\begin{gathered} OPT(G_k) = OPT(G_k)|_A + OPT(G_k)|_{A,B} + OPT(G_k)|_{B\cup C} + OPT(G_k)|_{A,C} \leq \\ n(W_A + W_{A,B} + W_{B\cup C}) + OPT(G_k)|_{A,C} \leq 5\epsilon n W_{A,C} + OPT(G_k)|_{A,C} \leq \\ 15\epsilon OPT(G_k) + OPT(G_k)|_{A,C}\,, \end{gathered}$$

其中，第一个不等式只需重新整理 $OPT(G_k)$的各项即可得到，第二个不等式成立是因为所有的$y_{i,\ j} \leq n$。第三个不等式由不等式20 得出，最后一个不等式则由事实3.4得出。重新整理各项可得

$$OPT(G_k) \leq \frac{1}{1-15\epsilon}\ OPT(G_k)|_{A,C} \leq (1+16\epsilon)\ OPT(G_k)|_{A,C}\,, \tag{21}$$

其中，最后一个不等式成立是因为 $\epsilon < 10^{-4}$。

2. $\mathbf{OPT(G_k)|_{A,\ C} \leq (1 + \frac{16\sqrt{\rho}}{\epsilon^2})ALG(G_k)|_{A,\ C}}$：我们通过将推论4.4应用于 ALG的排列（记为 y^{ALG}），并将推论4.7应用于 OPT的排列（记为y），即可得到这两个推论分别给出

$$\begin{aligned} ALG(G_k)|_{A,C} &\geq \frac{1}{2} n W_{A,C}(1 - \frac{5\sqrt{\rho}}{\epsilon^2}), \\ OPT(G_k)|_{A,C} &\leq \frac{1}{2} n W_{A,C}(1 + \frac{9\sqrt{\rho}}{\epsilon^2}). \end{aligned}$$

综合这两个不等式可得

$$OPT(G_k)|_{A,C} \leq (\frac{1+\frac{9\sqrt{\rho}}{\epsilon^2}}{1-\frac{5\sqrt{\rho}}{\epsilon^2}})\ ALG(G_k)|_{A,C} \leq (1+\frac{16\sqrt{\rho}}{\epsilon^2})\ ALG(G_k)|_{A,C}\,, \tag{22}$$

其中，最后一个不等式成立是因为 $\rho < \epsilon^6$ 且 $\epsilon < 10^{-3}$。

3. $\mathbf{OPT(G_k) \leq (1+33\epsilon)ALG}$：我们通过结合不等式 20 和 22

$$\begin{gathered} OPT(G_k) \leq (1+16\epsilon)\ OPT(G_k)|_{A,C} \leq \\ (1+16\epsilon)(1+\frac{16\sqrt{\rho}}{\epsilon^2})\ ALG(G_k)|_{A,C} \leq (1+16\epsilon)(1+\frac{16\sqrt{\rho}}{\epsilon^2})ALG(G_k), \end{gathered}$$

并且由于 $\rho < \epsilon^6$ 和 $\epsilon < 10^{-3}$，我们得到

$$OPT(G_k) \leq (1+35\epsilon)ALG(G_k) \Rightarrow \frac{ALG(G_k)}{OPT(G_k)} \geq 1 - 33\epsilon,$$

由此完成证明。

□

命题的证明 4.14。 由于 $ALG(G_i)$ 递归 $B \cup C$ 我们有 $ALG(G_i)|_{B\cup C}$ =$ALG(G_{i+1})$。因此，

$$ALG(G_i) = ALG(G_i)|_A + ALG(G_i)|_{A,B\cup C} + ALG(G_i)|_{B\cup C} = ALG(G_i)|_A + ALG(G_i)|_{A,B\cup C} + ALG(G_{i+1}) \geq ALG(G_i)|_{A,C} + ALG(G_{i+1}).$$

应用推论 4.4 于 $ALG(G_i)$的排列，得到

$$ALG(G_i)|_{A,C} \geq \frac{1}{2} n W_{A,C}(1 - \frac{5\sqrt{\rho}}{\epsilon^2}).$$

合并这两个不等式，即得证明。 □

命题的证明4.15. 我们首先观察到

$$OPT(G_i) = OPT(G_i)|_{A_i} + OPT(G_i)|_{A_i,B_i} + OPT(G_i)|_{A_i,C_i} + OPT(G_i)|_{B_i\cup C_i}. \tag{23}$$

考虑 $OPT(G_i)|_{B_i\cup C_i} = \sum_{e\in B_i\cup C_i} w_e y_e$。该值 y_e由来自 n_{A_i}和 $n_{B_i\cup C_i}$的节点组成。因此 $OPT(G_i)|_{B_i\cup C_i} \leq n_{A_i} W_{B_i\cup C_i} + OPT(G_{i+1})$，因为 $OPT(G_{i+1})$ 最优地求解了由$B_i \cup C_i$定义的实例。根据事实 3.4，有 $OPT(G_{i+1}) \geq \frac{1}{3} n_{B_i\cup C_i} W_{B_i\cup C_i}$。此外，根据引理 2.2，有 $n_{B_i\cup C_i} \geq \frac{1-\sqrt{\rho_i}}{\sqrt{\rho_i}} n_{A_i}$。综合以上可得

$$OPT(G_i)|_{B_i\cup C_i} \leq n_{A_i} W_{B_i\cup C_i} + OPT(G_{i+1}) \leq \frac{\sqrt{\rho_i}}{1-\sqrt{\rho_i}} n_{B_i\cup C_i} W_{B_i\cup C_i} + OPT(G_{i+1}) \leq (1 + \frac{3\sqrt{\rho_i}}{1-\sqrt{\rho_i}}) OPT(G_{i+1}). \tag{24}$$

接下来考虑 $OPT(G_i)|_{A_i} + OPT(G_i)|_{A_i,\ B_i}$。注意到 $OPT(G_i)|_{A_i} + OPT(G_i)|_{A_i,\ B_i} \leq n_{V_i}(W_{A_i} + W_{A_i,\ B_i})$，因为每条边最多可贡献 n_{V_i}。根据引理 4.8，有 $W_A + W_{A,\ B} \leq 2\frac{\sqrt{\rho}}{\epsilon^2} W_{A,\ C}$。综合以上可得

$$OPT(G_i)|_{A_i} + OPT(G_i)|_{A_i,B_i} \leq n_{V_i}(W_{A_i} + W_{A_i,B_i}) \leq n_{V_i} \cdot 2\frac{\sqrt{\rho}}{\epsilon^2} W_{A,C}. \tag{25}$$

最后，考虑 $OPT\ (G_i)\ |_{A_i,\ C_i}$。由推论4.7应用于 $OPT\ (G_i)$ 可得

$$OPT(G_i)|_{A_i,C_i} \leq \frac{1}{2} n_{V_i} W_{A_i,C_i}(1 + \frac{9\sqrt{\rho}}{\epsilon^2}). \tag{26}$$

我们总体上得到

$$OPT(G_i) = OPT(G_i)|_{A_i} + OPT(G_i)|_{A_i,B_i} + OPT(G_i)|_{A_i,C_i} + OPT(G_i)|_{B_i\cup C_i} \leq n_{V_i} \cdot 2\frac{\sqrt{\rho}}{\epsilon^2} W_{A,C} + \frac{1}{2} n_{V_i} W_{A_i,C_i}(1 + \frac{9\sqrt{\rho}}{\epsilon^2}) + (1 + \frac{3\sqrt{\rho_i}}{1-\sqrt{\rho_i}}) OPT(G_{i+1}) = \frac{1}{2} n_{V_i} W_{A_i,C_i}(1 + \frac{13\sqrt{\rho}}{\epsilon^2}) + (1 + \frac{3\sqrt{\rho_i}}{1-\sqrt{\rho_i}}) OPT(G_{i+1}),$$

其中，第一个等式由等式23 得出，第一个不等式由不等式24、25和26得出。最后，由于 $\rho_i < \epsilon^6$ 且 $\epsilon < \frac{1}{2}$ 可得 $\frac{3\sqrt{\rho_i}}{1-\sqrt{\rho_i}} \leq 4\sqrt{\rho_i}$ 进而得到

$$OPT(G_i) \leq \frac{1}{2} n_{V_i} W_{A_i,C_i}(1 + \frac{13\sqrt{\rho}}{\epsilon^2}) + (1 + 4\sqrt{\rho_i}) OPT(G_{i+1}).$$

由 $\beta_i = \frac{1}{2} n_{V_i} W_{A_i,\ C_i}(1 + \frac{13\sqrt{\rho}}{\epsilon^2})$ 和 $\gamma_i = 1 + 4\sqrt{\rho_i}$ 可得证明。 □

引理的证明 4.8。显然，我们观察到 $W_A \le \frac{1}{2}n_A^2 D_V$和 $W_{A,\ B} \le n_A n_B D_V$(回忆一下，$D_V$表示 V的直径)。由引理 2.2 可得 $n_A,\ n_B \le \sqrt{\rho}n$ 且 $n_C \ge (1-\sqrt{\rho})n$ 因此

$$W_A + W_{A,B} \le D_V n_A(\frac{1}{2}n_A + n_B) \le D_V n_A(1.5\sqrt{\rho}n) \le D_V n_A n_C \cdot \frac{1.5\sqrt{\rho}}{1-\sqrt{\rho}}.$$

最后，我们注意到，根据定义，C A 和 C 之间的所有权重至少为 $\epsilon^2 D_V$，因此$W_{A,\ C} \ge \epsilon^2 n_A n_C D_V$。结合上述不等式可得

$$W_A + W_{A,B} \le \frac{1.5\sqrt{\rho}}{1-\sqrt{\rho}} n_A n_C D_V \le \frac{1.5\sqrt{\rho}}{\epsilon^2(1-\sqrt{\rho})} W_{A,C} \le 2\frac{\sqrt{\rho}}{\epsilon^2} W_{A,C},$$

其中第三个不等式成立，是因为 $\frac{1}{1-\sqrt{\rho}} \le \frac{4}{3}$（因为 $\rho < \epsilon^6$ 和 $\epsilon < \frac{1}{4}$）。□

引理的证明 4.22。考虑算法 ALG_{d-w}。它只有一个循环，该循环调用 $PT(G, \Phi, \epsilon_{err} = \epsilon^9)$，并计算输出划分的值 $\sum w_e \widehat{y}_e$。计算该划分所需的时间为$O(n^2)$。

为了界定运行时间，我们考虑定理 4.11。注意，在我们的情形中，划分集的数量为 $k = \frac{1}{\epsilon}$。因此，$PT(G, \Phi, \epsilon_{err} = \epsilon^9)$ 的运行时间上界为

$$\exp\left(\log(\frac{1}{\epsilon^9}) \cdot (\frac{O(1)}{\epsilon^9})^{\frac{1}{\epsilon}+1}\right) + O(\frac{\log\frac{1}{\epsilon^{10}}}{\epsilon^{18}}) \cdot n.$$

考虑 ALG_{d-w} 内的 for 循环。对于给定的一对 j，j'，每个值 $\beta_{j,\ j'}$都可以有 $\frac{1}{\epsilon^7}$ 个不同的取值。j 和 j' 各自都可以有 $\frac{1}{\epsilon}$ 个取值。因此，该循环会执行 $(\frac{1}{\epsilon^7})^{\frac{1}{\epsilon^2}}$ 次。因此，算法的总运行时间有如下上界：

$$(\frac{1}{\epsilon^7})^{\frac{1}{\epsilon^2}} \cdot \left(\exp\left(\log(\frac{1}{\epsilon^9}) \cdot (\frac{O(1)}{\epsilon^9})^{\frac{1}{\epsilon}+1}\right) + O(\frac{\log\frac{1}{\epsilon^{10}}}{\epsilon^{18}}) \cdot n + O(n^2)\right) = (\frac{1}{\epsilon^7})^{\frac{1}{\epsilon^2}} \cdot O(n^2).$$

□

定理 4.24 的证明。我们首先注意到，情形 (b) 的运行时间被情形 (a) 的运行时间所涵盖，因此可以假设算法在情形 (a) 下终止。

接下来，我们考察每个递归步骤，并注意到其运行时间由找到 A_i 所需的时间决定。为了界定这一运行时间，考虑引理 2.2 的证明，并注意到该证明是算法性的；可以遍历所有点，并检查每个点周围距离为$\le 2D_V\sqrt{\rho_V}$ 的节点数量，整个过程耗时 $O(n^2)$（相对于输入规模为线性时间）。根据备注 4.23，我们可以保证递归步骤的数量为 $O(\log n)$，因此总计为 $O(n^2 \log n)$。

因此，结合引理 4.22 (它给出了情形 (a) 的运行时间上界)，我们得到 ALG的运行时间为 $O(n^2 \log n)$，再加上 ALG_{d-w}的运行时间（即 $f(\frac{1}{\epsilon}) \cdot O(n^2)$），由此得到一个 EPRAS。□

C 第5节的延后证明

命题的证明 5.4。设 V表示 G_k的节点以及 $n_V = |V|$。由于我们处于情形 (a)，因此有$\rho_V \ge \epsilon^2$且由于 $\rho_V = \frac{W_V}{n_V^2 D_V}$我们有 $\frac{W_V}{D_V} \ge n_V^2\epsilon^2$。我们论证该实例的 $c_0 = c_1 = \epsilon^2$并非所有权重都很小。事实上，否则总权重将受以下值限制

$$\frac{W_V}{D_V} < (1-\epsilon^2)\cdot\epsilon^2 \cdot \binom{n_V}{2} + \epsilon^2 \cdot 1 \cdot \binom{n_V}{2} < 2\epsilon^2\binom{n_V}{2} \le \epsilon^2 n_V^2,$$

与我们的假设相矛盾。因此，由定理 5.3 可知，借助 $\xi = O(\epsilon^5)$ 我们可以保证

$$\frac{ALG(G_k)}{OPT(G_k)} \geq 1 - O(\frac{\xi}{c_0 \cdot c_1}) = 1 - \epsilon,$$

由此完成证明。 □

命题 5.5 的证明。回顾一下，$ALG(G_k)$ 的定义是：首先将 A 聚类为一架梯子（记为 T_A），然后以任意方式对 C 进行聚类（记为 T_C），最后将 T_C 作为 T_A 的根节点。因此，根据层次聚类目标的定义，与 A 相连的每个权重都会为目标值增加其权重乘以n_C，从而 $ALG(G_k) \geq (W_A + W_{A,C})n_C$。由引理 2.2可知，$n_C \geq (1 - \sqrt{\rho})n$，因此$ALG(G_k) \geq (1 - \sqrt{\rho})n(W_A + W_{A,C})$。

由于我们处于情形 (b)，因此有 $W_C \leq 16\epsilon W_V$，进而有 $W_A + W_{A,C} \geq (1 - 16\epsilon)W_V$。结合上文的解释，我们得到 $ALG(G_k) \geq (1 - \sqrt{\rho})(1 - 16\epsilon)nW_V$。显然，$OPT(G_k) \leq Wn$。最后，同样由于我们处于情形 (b)，有 $\rho < \epsilon^2$。综上，

$$ALG(G_k) \geq (1 - \sqrt{\rho})(1 - 16\epsilon)OPT(G_k) \geq (1 - \epsilon)(1 - 16\epsilon)OPT(G_k) \geq (1 - 17\epsilon)OPT(G_k).$$

□

引理 5.6 的证明。由于 $ALG(G_i)$ 将 A 放置为一架梯子，并将 C 放在梯子的底部，因此我们有 $ALG(G_i)|_A + ALG(G_i)|_{A,C} \geq n_C(W_A + W_{A,C})$。另一方面，根据引理 2.2，我们有 $n_C \geq n(1 - \sqrt{\rho})$。最后，由于 $ALG(G_i)$ 对 C 进行递归，因此我们有 $ALG(G_i)|_C = ALG(G_{i+1})$。因此

$$ALG(G_i) = ALG(G_i)|_A + ALG(G_i)|_{A,C} + ALG(G_i)|_C \geq$$
$$n(W_A + W_{A,C})(1 - \sqrt{\rho}) + ALG(G_i)|_C = n(W_A + W_{A,C})(1 - \sqrt{\rho}) + ALG(G_{i+1}),$$

由此得证。 □

引理的证明5.7。我们首先注意到

$$OPT(G_i) = OPT(G_i)|_{A_i} + OPT(G_i)|_{A_i,C_i} + OPT(G_i)|_{C_i} \leq n_{V_i}(W_{A_i} + W_{A_i,C_i}) + OPT(G_i)|_{C_i},$$

因为每条边最多只能贡献 n_{V_i}。考虑 $OPT(G_i)|_{C_i} = \sum_{e \in C_i} w_e T_e$。值 T_e 由来自 n_{A_i} 和 n_{C_i} 的节点组成。因此 $OPT(G_i)|_{C_i} \leq n_{A_i}W_{C_i} + OPT(G_{i+1})$，因为 $OPT(G_{i+1})$ 最优地求解了由 C_i 定义的实例。因此 $OPT(G_i) \leq n_{V_i}(W_{A_i} + W_{A_i,C_i}) + n_{A_i}W_{C_i} + OPT(G_{i+1})$。

根据事实 3.4我们有 $OPT(G_{i+1}) \geq \frac{2}{3}n_{C_i}W_{C_i}$。此外，根据引理 2.2，我们有 $n_{C_i} \geq \frac{1-\sqrt{\rho_i}}{\sqrt{\rho_i}}n_{A_i}$。将这些不等式与上述结果结合可得

$$OPT(G_i) \leq n_{V_i}(W_{A_i} + W_{A_i,C_i}) + n_{A_i}W_{C_i} + OPT(G_{i+1}) \leq$$
$$n_{V_i}(W_{A_i} + W_{A_i,C_i}) + \frac{\sqrt{\rho_i}}{1 - \sqrt{\rho_i}}n_c W_C + OPT(G_{i+1}) \leq$$
$$n_{V_i}(W_{A_i} + W_{A_i,C_i}) + (1 + \frac{\frac{3}{2}\sqrt{\rho_i}}{1 - \sqrt{\rho_i}})OPT(G_{i+1}).$$

最后，由于 $\rho_i < \epsilon^2$ 和 $\epsilon < \frac{1}{2}$，我们有 $\frac{\frac{3}{2}\sqrt{\rho_i}}{1-\sqrt{\rho_i}} \leq 2\sqrt{\rho_i}$，因此

$$OPT(G_i) \leq \beta_i + \gamma_i OPT(G_{i+1})$$

对于 $\beta_i = n_{V_i}(W_{A_i} + W_{A_i,C_i})$ 和 $\gamma_i = 1 + 2\sqrt{\rho_i}$。 □

为了证明定理 5.11，我们需要证明 $\Pi_{j=0}^{i-1}\gamma_j$ 收敛。幸运的是，加权密度 ρ_i 增长得足够快，足以确保这一点。本小节的所有证明均推迟至附录。

引理 C.1。 对于所有 $i = 0,1,\ldots,k-1$，我们都可以保证 $\rho_{i+1} \geq 4\rho_i$。

引理 *C.1* 的证明。 首先注意到 $\rho_{i+1} \geq 4\epsilon\sqrt{\rho_i}$，这由以下事实得出： $W_{V_{i+1}} \geq 16\epsilon W_{V_i}$， $n_{V_{i+1}} \leq n_{V_i}$ 以及 $D_{V_{i+1}} \leq 4D_{V_i}\sqrt{\rho_i}$，而这些都由引理 2.2 以及情况 (c) 适用这一点得出。接下来，同样由于情况 (c) 适用，我们有 $\rho_i \leq \epsilon^2$，因此 $4\epsilon\sqrt{\rho_i} \geq 4\rho_i$，从而完成证明。 □

现在我们可以证明 $\Pi_{j=0}^{i-1}\gamma_j$ 收敛。

推论 C.2。 对于 $\gamma_i = 1 + 2\sqrt{\rho_i}$，我们有 $\Pi_{j=0}^{i-1}\gamma_j \leq 1 + 3\sqrt{\rho_i}$。

推论的证明 *C.2*。 注意到

$$\Pi_{j=0}^{i-1}(1 + 2\sqrt{\rho_j}) \leq e^{2\cdot\sum_j\sqrt{\rho_j}} \leq e^{2\sqrt{\rho_i}} \leq 1 + 3\sqrt{\rho_i},$$

其中第一个不等式由观察 3.3得出，第二个不等式是因为 $\sqrt{\rho_j}$ 呈指数增长（引理C.1），第三个不等式则再次由观察 3.3 结合以下事实得出： $\rho < \epsilon^2$ 且 $\epsilon < 10^{-2}$。 □

接下来，我们利用前述引理来界定 $\min_i\{\frac{\alpha_i}{\beta_i\Pi_{j=0}^{i-1}\gamma_j}\}$ 和 $\frac{ALG}{(\Pi_{i=0}^{k-1}\gamma_i)OPT(G_k)}$。

命题的证明*5.9*。 根据 α_i、 β_i 和 γ_i 的定义，我们有

$$\frac{\alpha_i}{\beta_i\Pi_{j=0}^{i-1}\gamma_j} = \frac{1-\sqrt{\rho_i}}{\Pi_{j=0}^{i-1}\gamma_j} \geq \frac{1-\sqrt{\rho_i}}{1+3\sqrt{\rho_i}} \geq 1 - 4\sqrt{\rho_i},$$

其中，第一个等式由 α_i 和 β_i 的定义得出，第一个不等式由推论 C.2及 γ_i的定义得出。因此，由于 ρ_i 仅增不减，

$$\min_i\{\frac{\alpha_i}{\beta_i\Pi_{j=0}^{i-1}\gamma_j}\} \geq 1 - 4\sqrt{\rho_{k-1}} \geq 1 - 4\epsilon,$$

其中，最后一个不等式由 $\rho_{k-1} < \epsilon^2$ 得出——至此证明完毕。 □

命题的证明*5.10*。 如果 $k = 0$ 那么我们没有递归，根据证明，由命题 5.4和5.5即可得出结论。否则

$$\frac{ALG(G_k)}{(\Pi_{i=0}^{k-1}\gamma_i)OPT(G_k)} \geq \frac{1-17\epsilon}{(1+2\sqrt{\rho_{k-1}})(1+3\sqrt{\rho_{k-1}})} \geq 1 - 23\epsilon,$$

其中第一个不等式由推论 C.2 和 γ_{k-1} 的定义可得，第二个不等式成立是因为 $\rho_{k-1} < \epsilon^2$（因为我们递归到了步骤 k）——从而完成证明。 □

定理的证明 *5.11*。 由观察 2.4 以及命题 5.9 和 5.10 . □

引理 C。 **3.** 算法 *4* 执行的递归步骤数有界于 $O(\log\log n)$.

引理 C.3 的证明。 令 ρ_0 表示原始图 $G_0 = (V_0, W_0)$ 的密度，并令 $n_0 = |V_0|$。根据三角不等式，我们有 $W_0 \geq D_{V_0}(n_0 - 1)$。考虑 u, v，其中 $w_{u,v} = D_{V_0}$，以及任意的 $w \neq u, v$，即可看出这一点。根据三角不等式，$w_{u,w} + w_{v,w} \geq D_{V_0}$。对所有 w 求和，并将其加到 $w_{u,v}$ 上，得到 $W_0 \geq D_{V_0}(n_0 - 1)$。因此 $\rho_0 = \frac{W_0}{n_0^2 D_{V_0}} \geq \frac{1}{n_0} - \frac{1}{n_0^2} \geq \frac{1}{2n_0}$。

作为引理证明的副产品 C.1，我们可以保证 $\rho_{i+1} \geq 4\epsilon\sqrt{\rho_i}$。因此 $\rho_0 \leq (\frac{1}{16\epsilon^2})^{2^k-1} \cdot \rho_{k-1}^{2^k}$。另一方面，根据我们算法的定义，如果执行了一个递归步骤，那么 $\rho_i \leq \epsilon^2$。因此，如果我们将 $k-1$ 视为最后一个递归步骤，则有 $\rho_{k-1} \leq \epsilon^2$。

综合以上所有结果可得

$$\frac{1}{2n_0} \leq \rho_0 \leq (\frac{1}{16\epsilon^2})^{2^k-1} \cdot \rho_{k-1}^{2^k} \leq (\frac{1}{16\epsilon^2})^{2^k-1} \cdot (\epsilon^2)^{2^k}.$$

提取 k 可得 $k = O(\log\log n)$。 □

定理的证明 5.12。 首先我们注意到，情形 (b) 的运行时间被情形 (a) 的运行时间所涵盖，因此可以假设算法在情形 (a) 下终止。

接下来，我们考察每个递归步骤，并注意到其运行时间由查找 A_i所需的时间决定。为了界定这一运行时间，请考虑引理 2.2 的证明，并注意到该证明是算法性的；可以遍历所有点，并对每个点检查距离为$\leq 2D_V\sqrt{\rho_V}$的节点数量，整个过程的运行时间为 $O(n^2)$（该时间与输入规模呈线性关系）。根据引理 C.3，可以保证递归步骤的数量为 $O(\log\log n)$，总计为 $O(n^2 \log\log n)$。

因此，结合定理 5.3 (它界定了情形 (a) 的运行时间)，可得 ALG 的运行时间为 $O(n^2 \log\log n)$ 加上 ALG_{d-w}的运行时间（即 $f(\frac{1}{\epsilon^5}) \cdot O(n^2)$），两者合起来构成一个 EPRAS。 □