森林中的公平相关聚类

Katrin Casel
德国柏林洪堡大学
TobiasFriedrich ！德国波茨坦大学哈索·普拉特纳研究所
MartinSchirneck ！
奥地利维也纳大学计算机科学学院
Simon Wietheger
德国波茨坦大学 Hasso Plattner研究所

arXiv:2302.11295v1 [cs.LG] 22 Feb 2023

摘要

近年来，算法公平性研究日益受到关注。这源于人们逐渐认识到，机器学习系统输入数据中的偏差可能导致歧视性输出。对于聚类任务，最核心的公平性概念之一是 Chierichetti、Kumar、Lattanzi 和 Vassilvitskii 在 [NeurIPS 2017] 上提出的形式化定义。若每个簇中敏感属性各类别的分布都与整个输入集中的分布相同，则称该聚类是公平的。这一概念的动机来自各种应用场景：待聚类的对象具有不应被过度或不足代表的敏感属性。关于这一版本公平聚类的大多数研究都聚焦于基于质心的目标。

与此不同，我们探讨这一公平性概念在相关聚类中的适用性。关于由此产生的公平相关聚类问题，现有文献要么提出近似保证较差的近似算法，要么对敏感属性可能的分布施加严格限制（通常只考虑两种类别且比例为 1:1）。我们的目标是了解能否在这两个极端之间取得更好的结果。为此，我们考察受限图类，以便从复杂度角度刻画在哪些敏感属性分布下，这种公平性形式是可处理的。

虽然现有关于公平相关聚类的研究提出了近似算法，但我们关注的是精确解，并探究是否存在可高效求解的实例。对于森林，非公平版本的相关聚类微不足道；但加入公平性后，复杂性呈现出出人意料的丰富图景。我们概述了公平相关聚类从可处理转变为不可处理时所涉及的森林分布和类型。

最令人惊讶的发现是，公平相关聚类之所以困难，并非源于公平性条件过于严格。我们将大部分结果推广到公平性条件较宽松的版本，同样成立。相反，困难的根源似乎在于敏感属性的分布。积极的一面是，我们找到了几种合理且确实可处理的分布。尽管目前只证明了这些分布在森林上的可处理性，但这或许能为更大的图类设计合理的近似算法开辟新途径。

2012年 ACM 学科分类 计算理论 → 图算法分析；社会与专业主题 → 计算/技术政策；计算理论 → 动态规划

关键词和短语 相关聚类、差别影响、公平聚类、宽松公平性

1 引言

在过去十年中，机器学习中的公平性这一概念日益受到关注，例如可参阅 Pessach 和 Schmueli [32]的综述。Feldman、Friedler、Moeller、Scheidegger 和 Venkatasubramanian [26] 根据美国最高法院1971年关于差别影响的一项裁决，对公平性进行了形式化定义。该定义要求，在招聘等决策过程中，既不应明确考虑性别或肤色等敏感属性，

也应确保敏感属性的体现形式按比例分布于决策过程的所有结果中。Feldman等人针对分类任务对这一概念进行了形式化定义。Chierichetti、Kumar、Lattanzi 和 Vassilvitskii [19] 则将这一概念应用于聚类任务。

本文采用基于差别影响的公平性理解方式。形式上，要进行聚类的对象都被赋予一种颜色，以代表某种敏感属性。接着，如果对于每个簇和每种颜色，该簇中此颜色对象的比例与所有顶点中此颜色顶点的总比例相对应，那么这种带颜色对象的聚类就称为公平 。更准确地说，如果一种聚类将对象集合划分为公平子集，那么它就是公平的。

▶ **定义 1** (公平子集)。 设 U 是一个有限对象集，其颜色由函数 $c\colon\ U \to [k]$赋予，其中 $k \in \mathbb{N}_{>0}$ 为某个集合。令 $U_i = \{u \in U \mid c(u) = i\}$表示所有颜色为 i 的对象所构成的集合，其中 $i \in [k]$遍历所有取值。当且仅当对于所有颜色 $i \in [k]$ ，均有 $\frac{|S\cap U_i|}{|S|} = \frac{|U_i|}{|U|}$ 时，集合 $S \subseteq U$ 是公平的。

为了理解这一公平性概念如何影响聚类决策，请看下面的例子。设想机场安保部门希望对旅客进行聚类，为每个群体分配一个潜在风险等级，并采取相应的预防措施。肤色等属性不应影响风险等级的分配。然而，数据中的偏差可能导致某些肤色在某些簇中过多或过少。仅从数据中移除肤色属性可能并不足够，因为它可能与其他属性相关。如果某种肤色在数据中的代表性远低于其他肤色，这类问题尤其容易出现。公平聚类会寻找最优聚类，使每个风险等级下的肤色分布都公平；具体而言，它要求每个簇中的肤色分布大致与所有旅客中的肤色分布相匹配。

Chierichetti等人开创性的公平聚类论文 [19] 将这一公平性概念引入聚类，并研究了以 k-中心和 k-中位数为目标的情形。他们的工作由 Bera、Chakrabarty、Flores 和 Negahbani [11], 进一步拓展；这些研究放宽了公平性约束，要求对每个簇中某种颜色的代表比例设定上界和下界。更准确地说，他们定义了公平集合的以下推广形式。

▶ **定义 2** (宽松公平集)。 对于有限集 U 和着色 $c\colon\ U \to [k]$ 对于某个 $k \in \mathbb{N}_{>0}$，令 $p_i, q_i \in \mathbb{Q}$ 满足：对于所有 $i \in [k]$， $0 < p_i \leqslant \frac{|U_i|}{|U|} \leqslant q_i < 1$ ，其中 $U_i = \{u \in U \mid c(u) = i\}$。集合$S \subseteq U$ 关于 q_i 和 p_i 是宽松公平的，当且仅当对于所有 $i \in [k]$， $p_i \leqslant \frac{|S\cap U_i|}{|S|} \leqslant q_i$ 。

继这些结果之后，这种（宽松）公平性的概念在基于质心的聚类目标方面得到了广泛研究，并取得了许多积极成果。

例如，Bercea 等人 [12] 针对设施选址类问题（如 k-中心和 k-中位数问题）给出了双准则常数因子近似。Bandyapadhyay、Fomin 和 Simonov [7]采用 Schmidt、Schwiegelshohn 和 Sohler [34] 提出的公平核集技术，为许多基于质心的聚类目标给出了常数因子近似；在诸多其他结果中，他们还为欧几里得空间中的公平 k-均值和 k-中位数问题给出了多项式时间近似方案。基于质心的目标的公平性似乎已得到充分理解，因此大多数研究已开始考虑更广义的设定，例如流式处理 [34], 或对群体归属了解不完全的情况 [25]。

相比之下，将这一公平性概念应用于图聚类目标的（正面）研究成果寥寥无几。其中，就公平性而言，研究最为充分的是相关聚类，而相关聚类可以说是研究最为广泛的图聚类目标。对于相关聚类，我们给定一个对象集合上的成对相似度度量，目标是找到一种聚类，使被置于不同簇中的相似对象数量与被置于同一簇中的不相似对象数量之和最小。形式化地说，相关聚类的输入为

相关聚类是一个图 $G = (V, E)$，目标是找到 V 的一个划分 $\mathcal{P}$，使其最小化定义为如下形式的相关聚类代价

$$\mathrm{cost}(G, \mathcal{P}) = |\{\{u, v\} \in \tbinom{V}{2} \setminus E \mid \mathcal{P}[u] = \mathcal{P}[v]\}| + |\{\{u, v\} \in E \mid \mathcal{P}[u] \neq \mathcal{P}[v]\}|. \tag{1}$$

公平相关聚类的任务是找到一个划分，将对象划分为满足公平性的集合，并使相关聚类代价最小。我们强调，这里讨论的是相关聚类的完全无权最小分歧形式。（之所以通常称为完全，是因为每一对对象要么相似，要么不相似，不存在对聚类而言无关紧要的情况。之所以称为无权，是因为两个顶点之间的（不）相似性是二元的。一对相似对象若被置于不同的簇中，或者一对不相似对象若处于同一簇中，都称为一次不一致，因此得名最小化不一致形式。）

有两篇论文似乎分别独立地开始研究公平相关聚类[1]。Ahmadian、Epasto、Kumar 和 Mahdian [2] 分析了公平性约束由某个 α 给定的设置，并要求每个簇中每种颜色的比例最多为 α。对于 $\alpha = \frac{1}{2}$，当存在两种颜色且比例为 1:1 时，这对应于我们的公平性定义，他们获得了 256-近似。对于 $\alpha = \frac{1}{k}$（其中 k 是图中的颜色数量），他们给出了 $16.48k^2$-近似。我们注意到，只有当存在 α^{-1} 种颜色且每种颜色出现次数相同时，他们的所有变体才与我们的公平性概念等价。Ahmadi、Galhotra、Saha 和 Schwartz [1] 针对具有两种颜色且比例为 1: c的实例，给出了 O(c^2)-近似算法。在颜色比例为 1:1的特殊情况下，给定任意一个针对非公平相关聚类的 β-近似算法，他们可以获得 $3\beta + 4$-近似。对于更一般的颜色分布，他们的方法也会显著恶化。对于具有 k 种颜色且比例为 1: $c_2 : c_3 : \ldots : c_k$ 的实例，其中 c_i为正整数，他们针对严格设置给出了 O($k^2 \cdot \max_{2 \leqslant i \leqslant k} c_i$)-近似，针对宽松设置给出了 O($k^2 \cdot \max_{2 \leqslant i \leqslant k} q_i$)-近似[2]。

继这两篇论文之后，Friggstad 和 Mousavi [28]为 1 : 1 颜色比例情形提供了一个因子为 6.18 的近似算法。据我们所知，关于公平相关聚类的最新论文由 Ahmadian 和 Negahbani 撰写，[3] 他们以一种略有不同的方式放宽公平性，并给出了公平相关聚类的近似算法。他们针对颜色分布为 1 : $c_2 : c_3 : \ldots : c_k$ 的情形，给出了比例为 $\mathcal{O}(\varepsilon^{-1} k \max_{2 \leqslant i \leqslant k} c_i)$ 的近似算法，其中 ε 与放宽程度有关（对于我们的宽松公平性定义，大致为 $q_i = (1 + \epsilon) c_i$）。

公平相关聚类的所有这些结果似乎都趋向于考虑两种颜色且比例为 1 : 1 这一非常受限的设定，以获得尚可的近似比。在本文中，我们希望了解这是否不可避免，还是说对于其他（可能更符合现实的）颜色分布，也有望取得更好的结果。为了单独考察公平性的作用，我们考虑相关聚类中的“简单”实例，并研究加入公平性约束后复杂度的增加。在不施加公平性约束的情况下，森林上的相关聚类很容易求解。我们发现，即便进一步假设度为常数或直径有界，将问题限制在森林上的公平相关聚类也会很快变为 NP 难。最令人惊讶的是，这种困难性在很大程度上也适用于宽松公平性，这表明问题的困难并非源于公平性定义过于严格。

[1] 令人困惑的是，这两篇论文的标题都为《公平相关聚类》。[2] 他们的定理声称达到了 O($\max_{2 \leqslant i \leqslant k} q_i$)-近似，但查看证明后可以发现，他们似乎无意中遗漏了 k^2 因子。

另一方面，我们找到了允许采用高效算法的颜色分布。不出所料，这包括比例为 1 : 1 的情形，并可推广到常数种 k 颜色，其分布为 c_1 : c_2 : c_3 : ... : c_k，其中常数 $c_1, \ldots, c_k$。这类分布可用于建模表现形式有限且分布近乎均匀的敏感属性。不那么出人意料的是，我们还发现，从某种意义上说，另一极端情形也具有可解性。我们证明，对于两种颜色且比例为 1 : c 的情形，只要 c 足够大（与顶点总数呈线性关系），森林上的公平相关聚类就能在多项式时间内求解。这种分布可用于建模少数群体严重代表性不足、因而迫切需要公平性约束的情形。尽管我们的结果仅适用于森林，但我们相信它们可以为研究更一般的图类提供起点。我们尤其希望本研究能引起人们对迄今为止遭到忽视的比例为 1 : c 且 c 足够大的分布的关注。

1.1 相关工作

对类似于或等同于相关聚类的聚类目标的研究可以追溯到 20 世纪 60 年代 [10, 33, 37]。Bansal、Blum 和 Chawla [8] 首次将“相关聚类”这一术语用于描述一种聚类目标。值得注意的是，该问题也以“簇编辑”的名称进行研究。关于权重，相关聚类最一般的表述是：对于每一对顶点，考虑两个正实数；如果将对象置于同一个簇中，则将第一个数计入代价；如果将对象置于不同簇中，则将第二个数计入代价 [4]。Bonchi、García-Soriano 和 Gullo [13] 最近出版的书籍全面概述了当前关于相关聚类的研究。

我们关注这样一种特定变体：考虑一个完全图，其边具有 $\{-1,1\}$ 权重，并采用最小分歧目标函数。该版本是 APX-困难的 [16], ，这尤其意味着，除非 P = NP，否则不存在能够实现任意优良近似的算法。相关聚类已知的最佳近似结果是 Cohen-Addad、Lee 和 Newman [20] 最近取得的突破性成果，他们给出了 $(1.994 + \epsilon)$ 的比例。

我们证明，在森林中，最优相关聚类解的所有簇都具有固定大小。在这种情况下，相关聚类与 k平衡划分有关。该问题要求将图划分为 k 个等规模簇，同时尽量减少被划分切断的边数。Feldmann 和 Foschini [27]研究了树上的这一问题，他们的结果与我们的结果有有趣的相似之处。

除了上文已经讨论过的公平相关聚类结果外，我们只知道有三篇论文针对图聚类目标研究了与 Chierichetti 等人 [19] 提出的公平性概念相近的概念。Schwartz 和 Zats [35] 研究了采用最大一致目标函数的不完整公平相关聚类。Dinitz、Srinivasan、Tsepenekas 和 Vullikanti [23] 研究了公平灾害遏制，这是一种涉及公平性的图割问题。他们的问题并非直接属于公平聚类问题，因为他们只要求划分中的一部分（被保留的部分）满足公平性。Ziko、Yuan、Granger 和 Ayed[38] 提出了一种通用的公平聚类启发式方法，但该方法无法为解的质量提供理论保证。

2 贡献

现在，我们概述关于公平相关聚类的研究发现。首先，我们给出若干支撑后续研究的结构性结果。随后，我们介绍针对某些图类和颜色比例的算法及困难性结果。我们还证明，公平聚类的困难性并非源于要求各簇



图1 示例森林：一个大小为 4 的簇与两个大小为 2 的簇具有相同的代价。只有一个大小为 4 的簇时（左图），簇间代价为 0，簇内代价为 4。包含两个大小为 2 的簇时（右图），簇间代价和簇内代价均为 2。

精确再现整个图的颜色分布。本节最后讨论了未来研究可能的方向。

2.1 结构性洞见

这里概述构成我们所有结果基础的结构性洞见。我们首先说明聚类代价、聚类“切断”的边数与图中的边总数之间存在紧密联系。我们将这些“切断”的边数称为簇间代价，以区别于簇内非边的数量，后者称为簇内代价。形式上，簇内代价和簇间代价分别是式 (1) 中相关聚类代价的第一项和第二项。以下引理表明，当所有簇的大小都相同时，只需最小化簇间代价，就足以最小化总代价。这大大简化了相关聚类算法的开发。

▶**引理 3.** 设 P 是 m-边图 G 的顶点划分。令 χ 表示 P 在 G 上产生的簇间代价。如果划分中的所有集合大小均为 d，那么代价$(P) = (d-1)_2 \; n - m + 2\chi$。特别地，如果 G 是一棵树，那么代价$(P) \equiv (d-3)_2 \; n + 2\chi + 1$。

乍看之下，要求所有簇大小相同似乎相当严格。然而，我们将在下文证明，对于二分图，尤其是森林和树，总是存在一种最小成本公平聚类，使得所有簇的大小确实相同。这一性质源于公平性约束对颜色分布的作用，因此是公平相关聚类所特有的。它使我们能够充分利用引理 3，既可用于构造 NP-困难性证明中的归约，也可用于算法方法，因为我们可以将注意力限制在簇大小相同的划分上。

考虑两种颜色，其比例为 1 : 2。那么，任何公平簇都必须至少包含 1 个第一种颜色的顶点和 2 个第二种颜色的顶点，才能满足公平性要求。我们证明，由于边的数量较少，森林的最小代价聚类完全由这样的最小簇构成。簇更大的任何聚类都会产生更高的代价。

▶**引理 4.** 设 F是一个森林，其中有 $k \geqslant 2$ 种颜色，其比例为 $c_1 : c_2 : \ldots : c_k$，且$c_i \in \mathbb{N}_{>0}$ 对所有 $i \in [k]$，最大公约数$(c_1, c_2, \ldots, c_k) = 1$，并且 $\sum_{i=1}^{k} c_i \geqslant 3$。那么，每个最小成本公平聚类中的所有簇大小均为 $d = \sum_{i=1}^{k} c_i$。

如图1所示，引理 4 无法扩展到按 1:1比例划分的两种颜色。事实上，对于森林而言，这种颜色分布是唯一一种允许较大簇的划分具有相同（但不会更小）代价的情况。我们证明了一个比引理 4 稍弱的结论，即始终存在一个最小成本公平聚类，其簇大小由颜色比例决定。我们发现，这一性质不仅对森林成立，对每个二分图也成立。请注意，在一般的二分图中，除了 1:1 之外，还有更多颜色比例会导致这些歧义。

表 1 我们的算法在森林上求解公平相关聚类问题的运行时间，取决于颜色比例。值 p 是任意有理数，且 $n/p - 1$ 为整数； c_1，c_2，……，c_k 是互素的正整数，且可能取决于 n。函数 f 和 g 见定理 23 和 27。

颜色比例	1 : 1	1 : 2	$1 : (n/p - 1)$	$c_1 : c_2 : \ldots : c_k$
运行时间	$O(n)$	$O(n^6)$	$O\left(n^{f(p)}\right)$	$O\left(n^{g(c_1,\ldots,c_k)}\right)$

▶ **引理 5**。 设 $G = (A \cup B, E)$ 是一个二分图，其 $k \geqslant 2$ 种颜色的比例为$c_1 : c_2 : \ldots : c_k$，且对所有 $i \in [k]$ 均有 $c_i \in \mathbb{N}_{>0}$，并且 最大公约数$(c_1, c_2, \ldots, c_k) = 1$。那么，存在一个最小成本公平聚类，使得其所有簇的大小均为 $d = \sum_{i=1}^{k} c_i$。此外，可以在线性时间内将每个包含较大簇的最小成本公平聚类转换为一个所有簇均不超过 d 个顶点的最小成本公平聚类。

总而言之，上述结果表明，颜色类别的比例是决定簇大小的关键参数。如果输入是一个二分图，其顶点以 k 种颜色按 $c_1 : c_2 : \cdots : c_k$的比例着色，那么我们的结果意味着，在不损失最优性的情况下，解可以限制为仅包含大小为 $d = \sum_{i=1}^{k} c_i$的簇，并且每个簇恰好包含 c_i个 i颜色的顶点。基于这些观察，我们在本文中表明，颜色比例也是决定公平相关聚类复杂度的关键参数。一方面，最优解的简单结构限制了搜索空间，并至少对某些实例实现了多项式时间算法。另一方面，这些洞见使我们能够证明，即使对于非常受限的输入类别，问题也已经是困难的。大多数证明的技术部分在于利用聚类成本、边的总数以及聚类所切割的边数之间的联系。

2.2 易处理实例

我们首先讨论算法结果。最简单的情形是有两种颜色，且每种颜色出现的次数相同。我们证明，对于颜色比例为 1 : 1 的二分图，公平相关聚类等价于不同颜色顶点之间的最大二分匹配问题。通过将问题标准地归约为计算最大流，我们得以利用 Chen、Kyng、Liu、Peng、Probst Gutenberg 和 Sachdeva [18] 最近取得的突破。该突破给出了一种运行时间为 $m^{1+o(1)}$ 的算法。

其余结果以森林作为输入，见表 1。我们的主要算法范式是动态规划，这并不令人意外。教科书中的一种方法可以在线性时间内求出森林中的最大匹配，从而解决 1 : 1 的情形。对于一般的颜色比例，我们设计了复杂得多的动态规划程序。我们以颜色比例 1 : 2 为引导示例。该算法分为两个阶段。第一阶段，我们计算一组候选划分，将森林划分为若干连通部分，每部分至多包含 1 个蓝色顶点和 2 个红色顶点。第二阶段，我们将每种划分中的各部分组合成公平簇，并返回由此得到的代价最低的聚类。难点在于这两个阶段并非彼此独立。分别在两个阶段中最小化“割”边的数量并不足够。我们证明，合并所产生的代价还取决于划分中生成的某种类型部分的数量。将这一数量与割的数量一并跟踪，可得到一个运行时间为 $O(n^6)$ 的算法。请注意，只要算法运行时间是多项式级别，我们就没有对其进行优化。

表2 树和一般图上公平相关聚类的复杂度与直径的关系。值 c 是一个正整数，可能取决于 n。

直径	颜色比例	树	一般图
$2, 3$	any	$\mathrm{O}(n)$	NP难
$\geqslant 4$	$1:c$	NP难	NP难

我们将此推广到比例为 $c_1 : c_2 : \cdots : c_k$ 的 k 种颜色。[3] 现在，我们必须考虑顶点划分的所有可能着色，使得每个部分中第 i 种颜色出现的次数至多为 c_i。在组合这些部分时，我们必须确保合并后的着色彼此兼容。由此得到的运行时间为 $\mathrm{O}(n^{g(c_1,\ldots,c_k)})$，其中 g 是某个（显式给出的）多项式。回顾引理 4，最小簇大小为 $d = \sum_{i=1}^{k} c_i$。如果这是一个常数，那么动态规划就在多项式时间内运行。然而，如果颜色数量 k 或某种颜色所占比例随 n 增长，问题就会变得不可处理。等价地说，如果簇的数量虽少于线性级别但仍然非常大，运行时间就会变差。

为减轻这种影响，我们至少针对两种颜色的森林给出一种互补算法。具体而言，考虑颜色比例 $1 : \binom{n}{p} - 1$。此时，最优解有 p 个簇，每个簇的大小为 $d = n/p$。关键观察是，森林中出现次数较少的颜色（例如蓝色）的顶点有 p 个，而任何公平聚类都会将蓝色顶点单独分开。通过切断至多 $p-1$ 条边即可做到这一点，得到一组（子）树，其中每棵树至多包含一个蓝色顶点。为获得聚类，我们将含有红色多余顶点的树分割开，并把这些顶点分配到其余部分。我们跟踪所有 $\mathrm{O}(n^{\mathrm{poly}(p)})$ 个割集和重排的代价，以计算代价最小的方案。总的来说，对于某个关于 p 的多项式，算法的运行时间为 $\mathrm{O}(n^{f(p)})$。总而言之，如果簇的数量 p 为常数，则运行时间为多项式。特别地，考虑整数颜色比例 $1 : c$,[4]，当 $c = \mathrm{O}(1)$ 或 $c = \Omega(n)$ 时，森林上的问题是可解的。接下来我们将证明，具有此类颜色比例的公平相关聚类在树上已是 NP难，因此其难度必然出现在某些中间 c 情形。

2.3 有界直径的二分性

表2展示了有界直径图上的公平相关聚类的复杂度。对于颜色比例为 $1 : c$ 的两种颜色树，我们得到一个二分性。当直径至多为 3 时，可以在 $\mathrm{O}(n)$ 时间内计算出最优聚类；但当直径至少为 4 时，该问题变为 NP难。事实上，这个线性时间算法可以推广到具有任意颜色数量、颜色比例任意的树。

这一方向上的主要结果是：即使在直径至少为 4、两种颜色的比例为 $1 : c$ 的树上，公平相关聚类问题仍然是困难的。该结论通过从强NP难的 3-划分问题归约得到证明。在该问题中，给定正整数 a_1、……、a_ℓ，其中 ℓ 是 3 的倍数，并且存在某个 B，使得 $\sum_{i=1}^{\ell} a_i = B \cdot \frac{\ell}{3}$。任务是将这些数 a_i 划分为三元组，使每个三元组中的数之和都等于 B。即使所有 a_i 都严格介于 $B/4$ 和 $B/2$ 之间，该问题仍然是 NP难的；这确保了如果某个数的子集之和为 B，那么该子集恰好包含三个元素。

[3] 这些 c_i 互素，但相对于 n而言，它们不一定是常数。[4] 在颜色比例为 $1 : c$ 时，c 不一定是常数，但诸如 $2 : 5$ 这样的比例不在讨论范围内。

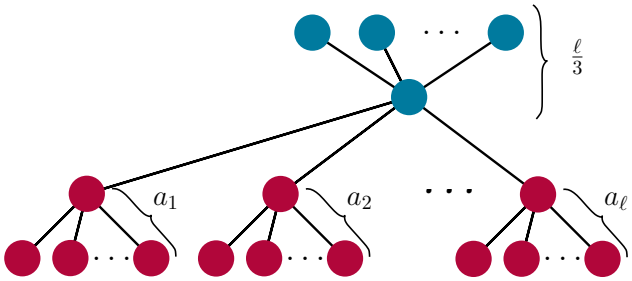

图2 从 3-划分到公平相关聚类的归约中，直径为 4 的树。

我们将该问题建模为一个公平相关聚类实例，如图2所示。我们构造 ℓ 个星形图，其中第 i 个星形图由 a_i 个红色顶点组成；此外，还构造一个由 $\ell/3$ 个蓝色顶点组成的星形图。蓝色星形图和所有红色星形图的中心彼此相连。所得实例中的颜色比例为 $1:B$。引理 4 随即推出，存在一种代价最小的聚类，其中有 $\ell/3$ 个簇，每个簇都包含一个蓝色顶点和 B 个红色顶点。然后，我们应用引理 3 证明：当且仅当每个簇恰好由三个红色星形图（以及任意一个蓝色顶点）组成时，该代价才低于某个阈值，从而解决 3-划分问题。

2.4 最大度

上述归约得到了一棵直径较小、但最大度可以任意大的树。为了证明有界度数情形同样是困难的，我们必须调整归约。结果总结于表3。如果公平相关聚类实例不要求连通，我们可以用最大度为 2 的树森林（即路径森林）来表示 3-划分。输入数字由含有 a_i 个顶点的路径建模。该森林还包含 $\ell/3$ 个孤立蓝色顶点，这同样意味着最优公平聚类必须有 $\ell/3$ 个簇，且每个簇包含 B 个红色顶点。通过设定足够小的代价阈值，我们可以确保公平聚类的代价低于该阈值，当且仅当聚类没有“切断”任何路径边，这对应于对 a_i 的划分。

路径并无特殊之处，只要始终能够构造出具有给定顶点数的树，我们就可以任意限制树的形状。然而，该论证关键依赖于 a_i-路径/树之间不存在边，因此无法推广到连通图。这是由于引理 3 所述的簇间代价与边的总数之间存在密切关系。因此，颜色比例为 $1:c$ 时，单一路径上的公平相关聚类复杂度仍未解决。尽管如此，我们在两种密切相关的设定下证明了树问题的困难性：保持颜色比例为 $1:c$，但将最大度数提高到 5；或者采用单一路径，但包含 $n/2$ 种颜色，且每种颜色恰好由 2 个顶点共享。

对于最大度数为 5 且两种颜色的比例为 $1:c$ 的情形，我们同样可以利用 3-划分的工具。该构造受到 Feldmann 和 Foschini [27] 利用这一问题证明计算所谓的 k-平衡划分具有困难性的启发。我们将其改造为适用于当前设定的构造，其中顶点带有颜色，且各簇必须满足公平性。

对于具有 $n/2$ 种颜色的单一路径，我们从（1-正则双色变体的）单词涂色问题 [24] 归约而来。在该问题中，给定一个单词，其中每个符号

表3 树和森林上的公平相关聚类的困难性取决于最大度。值 c 是一个正整数，可能取决于 n。当颜色比例为 $1:c$ 时，路径（最大度为 2 的树）的复杂度尚未确定。

最大度	颜色比例	树	森林
2	$1:\ c$		NP难
$\geqslant 2$	$n/2$ 颜色，每个 2 个顶点	NP难	NP难
$\geqslant 5$	$1:c$	NP难	NP难

恰好出现两次。任务是为单词中的字母[5] 赋值 0 和 1，使得每个符号的两次出现中恰有一次被赋值为 1，同时使整个单词中连续 0 和 1 的区块数最少。在转化为公平相关聚类时，我们将单词表示为一条路径，将符号表示为颜色。为了保持公平性，必须有两个簇，且每个簇恰好包含每种颜色的一个顶点；这对应于为单词赋予一个 $0/1$ 赋值。

2.5 松弛公平性

有人可能会认为，公平相关聚类即使只针对树类和森林，其困难性也源于严格的公平性条件。毕竟，每个簇中的颜色比例必须与整个图中的颜色比例完全一致。这种看法具有误导性。相反，我们基于 Bera 等人提出的宽松公平性 [11]，将大部分困难性结果推广到松弛公平相关聚类。回顾定义 2。它为每种颜色 i 指定两个有理数 p_i 和 q_i，并允许任何簇中颜色为 i的元素所占比例处于区间 $[p_i, q_i]$内，而不必恰好等于 c_i/d，其中 $d = \sum_{j=1}^{k} c_j$。

核心思路是证明，在某些情形下（但并非所有情形），满足松弛公平性约束的最小代价 解实际上完全 公平。这适用于上述通过从 3-划分规约得到的情形。具体而言，当颜色比例为 $1:c$ 时，松弛公平相关聚类在直径为 4 的树以及路径森林上分别是 NP难。此外，对于具有 $n/2$ 种颜色且每种颜色恰有 2 个顶点的单一路径，困难性的转移是直接的。任何公平性松弛仍要求每个簇中每种颜色都至少有一个顶点，因此依然等价于单词涂色问题。

相比之下，如果存在代价低于任何严格公平解的宽松公平解，那么算法结果就更难推广。此时，我们不再知道最优解中各簇的基数。作为概念验证，我们展示了：稍微调整针对两种颜色、比例为 1:1 的动态规划后，它仍适用于我们所称的 α-宽松公平性。[6] 其中，对于某个参数 $\alpha \in (0,1)$，公平性比例的下限为 $p_i = \alpha \cdot \frac{c_i}{d}$，上限为$q_i = \frac{1}{\alpha} \cdot \frac{c_i}{d}$。我们给出了所需簇大小的上界，该上界取决于 α，足以找到森林的一种良好划分。当然，运行时间现在也取决于 α，但其形式为 $O(n^{h(1/\alpha)})$，其中 h 是某个多项式。特别地，当 α 为常数时，我们得到一个多项式时间算法。正确性证明

[5] 原始表述 [24] 分配颜色，这与油漆车间的类比更加契合。为避免与公平性意义上的颜色混淆，我们在此调整了表述方式。[6] 请勿将此概念与资源分配 [30, 31]中的 α-公平性 混淆。

即使对于 1:1 这一简单情形，也需要穷尽所有情况进行区分。我们相信，这可以推广到更一般的颜色比例，但本工作中并未尝试这样做。

2.6 总结与展望

我们证明了树上的公平相关聚类以及森林上的公平相关聚类都是NP难问题。在常数度或常数直径的树上，该问题仍然是NP难的；而对于某些颜色分布，即使在路径上，它也同样是NP难的。另一方面，当公平簇的最小大小 d 为常数时，我们给出了一种多项式时间算法。对于颜色比例1： c 的情形，当簇的总数为常数时，我们还提供了一种高效算法，这对应于 $c \in \Theta(n)$ 。对于我们的主要算法和困难性结果，我们证明即使放宽公平性约束，它们仍然成立，因此困难性并非源于严格的公平性定义。最终，我们希望从这些证明中获得的见解以及所提出的算法，能够为未来开发用于解决更一般图上公平相关聚类问题的算法提供帮助。特别是，颜色比例为1： c 且 c 非常大的公平性分布，似乎是一个值得未来研究、且可能具有可处理性的有趣类型。

作为推广我们结果的初步尝试，我们为森林上的公平相关聚类给出了一个多项式时间近似方案（PTAS）。未来研究的另一个方向是，界定最优解簇大小的引理 5 是否也适用于二分图。这或许有助于为颜色比例不为 1 : 1 的二分图开发精确算法。

参数化算法是求解更一般实例的另一种方法。考察公平相关聚类的判定版本时，我们的结果可以表述为一种以簇大小 d为参数的 XP算法，因为该问题可在时间 O($n^{g(d)}$) 内求解，其中 g 是某个函数。类似地，以簇的数量为参数，我们也能得到一种 XP算法。我们想知道，对于某些有意义的结构参数，公平相关聚类能否归入固定参数可解问题的 FPT 类。这需要一个例如 $g(d)\cdot \text{poly}(n)$ 的运行时间。对于以解的代价 [15]为参数的簇编辑问题，已有固定参数可解算法。未来研究或许也能为公平变体带来类似结果。对我们的动态规划方法进行自然扩展，有望得到一种以输入图的树宽为参数的算法。不过，这样的解会令人意外，因为据我们所知，即使对于普通的非公平相关聚类[7] 以及相关的最大稠密图划分问题 [22] ，也尚无树宽方法。

最后，公平相关聚类在路径上的表现也颇为有趣。虽然我们借助单词喷漆车间问题证明了某种特定颜色分布下的 NP-困难性，但公平相关聚类在路径上（例如，两种颜色的比例为 1： c ）是否可以高效求解，仍是一个开放问题。不过，我们认为，研究相关的（离散）项链分割问题或许能解答这一问题，参见 Alon 和 West 的研究 [6]。在该问题中，每种颜色类所需的基数都已明确给出，并且通过非构造性方法证明，总存在一种项链划分方案，其割数达到显然的下界。构造性的划分过程或许能为路径上的公平相关聚类带来一些启发。

[7] 更具体地说，目前尚未提出用于完整相关聚类的算法。Xin [36] 给出了一种树宽算法，用于处理不完整的相关聚类，其对象是所有正标记边和负标记边构成的图的树宽。

3 预备知识

这里我们统一技术部分所用的符号，并给出公平相关聚类的形式化定义。

3.1 符号

我们用 $\mathbb{N}$ 表示自然数集合$\{0, 1, 2, \ldots\}$。对于，令 $\{1, 2, \ldots, k\}$，并令 $\mathbb{N}_{>k} = \mathbb{N} \setminus (\{0\} \cup [k])$。我们用 $2^{[k]}$ 表示 $[k]$ 的幂集。用 $\gcd(a_1, a_2, \ldots, a_k)$ 表示 $a_1, a_2 \ldots, a_k \in \mathbb{N}$ 的最大公约数 。

无向图 $G = (V, E)$ 由顶点集合 V 和边集合 $E \subseteq \binom{V}{2} = \{\{u, v\} \mid u, v \in V, u \neq v\}$ 定义。若无特别说明，图 G 的大小指 $n + m$，其中 $n = |V|$ 且 $m = |E|$。若 $m = \binom{n-1}{2}$，则称图为完全图。若图 $G = (A \cup B, E)$ 中 A 和 B 内均没有边，即 $E \cap \binom{A}{2} = E \cap \binom{B}{2} = \emptyset$，则称其为二分图。对于每个 $S \subseteq V$，令 $G[S] = \left(S, E \cap \binom{S}{2}\right)$ 表示由 S 诱导的子图。顶点 $v \in V$ 的度是与该顶点关联的边数，$\delta(v) = |\{u \mid \{u, v\} \in E\}|$。图 $G = (V, E)$ 的度是其所有顶点中的最大度 $\delta(G) = \max_{v \in V} \delta(v)$。图 G 中长度为 k 的路径是一个顶点元组 $(v_1, v_2, \ldots, v_{k-1})$，满足对于每个 $1 \leqslant i < k - 1$，都有 $\{v_i, v_{i+1}\} \in E$。我们只考虑简单路径，即对于所有 $i \neq j$，都有 $v_i \neq v_j$。若对于每一对顶点 u, v，都存在一条连接 u 和 v 的路径，则称图为连通图。两个顶点之间的距离是连接这两个顶点的最短路径的长度，图的直径则是一对顶点之间的最大距离。圈是路径 $(v_1, v_2, \ldots, v_k)$，满足 $v_1 = v_k$，且仅对于其他所有 $i \neq j$ 对有 $v_i \neq v_j$。

森林是没有环的图。连通的森林称为树。在一棵树中，每对顶点之间恰好有一条路径相连。通过选择任意一个顶点$r \in V$作为根，可以将一棵树定根。此时，除根之外的每个顶点 v都有一个父节点，即从 v 到 r的路径上的下一个顶点。所有以 v 为父节点的顶点称为 v 的子节点。没有子节点的顶点称为叶节点。给定一棵有根树 T，用 T_v 表示由 v 及其后代诱导的子树，即满足如下条件的顶点集合：存在一条从 v 开始并在该顶点结束、且不使用通向 v父节点的边的路径。注意，每个森林都是二分图，例如，可将到各自树根距离为偶数的所有顶点置于一侧，其他顶点置于另一侧。

有限集 U 可以通过某个函数 $c : U \to [k]$进行着色，其中 $k \in \mathbb{N}_{>0}$。如果只有两种颜色，即 $k = 2$，为方便起见，我们不再用数字表示，而称它们为红色和蓝色。

对于一个划分 $\mathcal{P} = \{S_1, S_2, \ldots\ldots, S_k\}$，其中 $S_i \cap S_j = \emptyset$ 对于某个集合 $U = S_1 \cup S_2 \cup \ldots \cup S_k$ 中的 $i \neq j$ 以及某个 $u \in U$，我们用 $\mathcal{P}[u]$ 指代满足 $u \in S_i$的集合 S_i。此外，我们定义集合和划分上的着色 这一术语。集合的着色 统计集合中每种颜色的出现次数。

▶ **定义 6** (集合着色)。 设 S 是一个集合，由函数 $c\colon\ S \to [k]$着色。那么， S 的着色是一个数组 C_S，满足 $C_S[i] = |\{s \in S \mid c(s) = i\}|$ 对所有 $i \in [k]$均成立。

划分的着色 统计划分中各集合着色的出现次数。

▶ **定义 7** （划分着色）。 设 U 为一个带颜色的集合，且 P 是U的一个划分。令 $C = \{C_S \mid S \subseteq U\}$ 表示这样一组集合着色：对于其中每种着色，都存在 U的一个子集具有该着色。按照任意固定的顺序，令 C_1、C_2、. . .、C_ℓ 表示 C 的元素。于是，P 的着色是一个数组 $C_\mathcal{P}$，满足 $C_\mathcal{P}[i] = |\{S \in P \mid C_S = C_i\}|$，对所有 $i \in [\ell]$均成立。

3.2 问题定义

为定义公平相关聚类，我们首先给出非公平聚类目标的正式定义。相关聚类接收一组对象的成对相似度度量，旨在尽可能减少被划分到不同簇中的相似对象数量，以及被划分到同一簇中的不相似对象数量。为保持一致性，我们重新表述 Bonchi 等人 [13] 的定义，使得对象之间的成对相似性由图而非显式的二元相似度函数给出。给定一个图 $G = (V, E)$以及 V的一个划分$\mathcal{P}$，相关聚类代价为

$$\mathrm{cost}(G, \mathcal{P}) = |\{\{u, v\} \in \tbinom{V}{2} \setminus E \mid \mathcal{P}[u] = \mathcal{P}[v]\}| + |\{\{u, v\} \in E \mid \mathcal{P}[u] \neq \mathcal{P}[v]\}|.$$

我们将第一个加数称为簇内代价 ψ，第二个加数称为簇间代价 χ。当 G 可根据上下文明确时，我们将其简写为 cost($\mathcal{P}$)。有时，我们会考虑$\mathcal{P}$ 在某个诱导子图上的代价。为此，即使$\mathcal{P}$ 划分的是某个集合 $V' \supseteq V$，我们也允许使用上述相同的代价定义。我们将（非公平）相关聚类定义如下。

相关聚类
输入： 图 $G = (V, E)$。**任务：** 求 V 的一个划分$\mathcal{P}$，使 cost($\mathcal{P}$) 最小。

我们强调，这是相关聚类的完全无权最小分歧形式。它是完全的，因为每一对对象要么相似，要么不相似，在聚类方面不存在无差别的情况。它是无权的，因为两个顶点之间的（不）相似性是二元的。一对被分到不同簇中的相似对象，以及一对被分到同一簇中的不相似对象，都称为不一致，因此称为最小分歧形式。另一种表述是最大一致形式，其目标是最大化不构成不一致的对象对数量。需要注意的是，这两种表述对聚类结果的排序相同，但由于代价函数的表述不同，近似因子可能有所差异。

我们对公平相关聚类问题的定义大致遵循 [2]。公平性将解空间限制为公平划分。如果一个划分中的每个集合都与被划分的全集具有相同的颜色分布，则该划分是公平的。

▶ **定义 8**（公平子集）。设 U 是一个有限的元素集合，其颜色由函数 $c\colon\ U \to [k]$赋予，其中 $k \in \mathbb{N}_{>0}$。令 $U_i = \{u \in U \mid c(u) = i\}$ 表示对于所有 $i \in [k]$而言颜色为 i 的元素集合。那么，某个 $S \subseteq U$ 是公平的，当且仅当对于所有颜色 $i \in [k]$，都有 $\frac{|S \cap U_i|}{|S|} = \frac{|U_i|}{|U|}$。

▶**定义 9**（公平划分）。设 U 是一个有限的元素集合，其颜色由函数$c\colon\ U \to [k]$ 赋予，其中 $k \in \mathbb{N}_{>0}$。那么，划分 $S_1 \cup S_2 \cup \ldots \cup S_\ell = U$是公平的，当且仅当所有集合 S_1、S_2..... S_ℓ都是公平的。

现在我们定义公平相关聚类问题的完全、无权、最小分歧变体。除非另有明确说明，提到（公平）相关聚类时，我们均指这一变体。

公平相关聚类**输入：** 图 $G=(V,\ E)$，颜色 $c\colon\ V\to[k]$。**任务：** 找到一个公平划分$\mathcal{P}$，将 V 的代价($\mathcal{P}$)最小化。

4 结构性洞见

我们在此证明第 2.1 小节概述的结构性结果。最重要的洞见是，在二分图中，尤其是在森林中，总存在一种最小成本公平聚类，使得所有簇的大小都固定。这一性质非常有用，因为它有助于构造困难性证明中的归约，也有助于开发枚举可能聚类方案的算法方法。此外，根据下述引理，这还意味着只需最小化簇间代价，就足以最小化相关聚类代价，从而简化了针对这类实例求解公平相关聚类的算法开发。

▶**引理 3.** 设 P 是 m-边图 G 的顶点划分。令 χ 表示 P 在 G 上产生的簇间代价。若划分中的所有集合大小均为 d，则 $\mathit{cost}(P)=\binom{d-1}{2}\frac{n}{d}-m+2\chi$。特别地，若 G 是一棵树，则 $\mathit{cost}(P)=\binom{d-3}{2}\frac{n}{d}+2\chi+1$。

证明。 注意，在每个 $\frac{n}{d}$ 簇中都有 $\binom{d}{2}$ 个顶点对；若它们之间没有边相连，则每对产生 1 的簇内代价。令簇内总代价为 ψ。由于总共有 m 条边，因此

$$\mathrm{cost}(\mathcal{P})=\chi+\psi=\chi+\frac{n}{d}\cdot\frac{d(d-1)}{2}-(m-\chi)=\frac{(d-1)n}{2}-m+2\chi.$$ ◀

特别地，若 G 是一棵树，则可得 $\mathrm{cost}(\mathcal{P})=\binom{d-3}{2}\frac{n}{d}+2\chi+1$，因为其中有 $m=n-1$。

4.1 森林

我们发现，在森林中，每个最小代价划分里的所有集合，其大小都恰好是满足公平性要求所需的最小值。

▶**引理 4。** 设 F是一个森林，其中有 $k\geqslant 2$种颜色，比例为 $c_1\colon\ c_2\colon\ldots\colon\ c_k$，其中 $c_i\in\mathbb{N}_{>0}$对所有 $i\in[k]$均成立，最大公约数$(c_1,c_2,\ldots,c_k)=1$，且 $\sum_{i=1}^{k}c_i\geqslant 3$。那么，每个最小成本公平聚类中的所有簇大小均为 $d=\sum_{i=1}^{k}c_i$。

证明。设 $d=\sum_{i=1}^{k}c_i$。对于 V的任意聚类 P，要满足公平性，所有簇的大小都必须至少为 d。我们证明，如果该聚类中存在一个簇 S，满足 $|S|>d$，那么通过划分 S可以降低代价。首先注意，为了满足公平性约束，对于某个 $a\in\mathbb{N}_{\geqslant 2}$，我们有$|S|=ad$。考虑通过将 S 划分为 S_1、S_2而得到的新聚类 P′，其中 $S_1\subset S$ 是 S 的一个大小为 d 的任意公平子集，而 $S_2=S\setminus S_1$。注意，对于至多有一个端点位于 S 中的每条边和非边，在两种聚类中的代价相同。令 ψ 表示 P 在 $F[S]$上的簇内代价。对于两个端点都位于 S中的边和非边所产生的代价，我们知道

$$\mathrm{cost}(F[S],\mathcal{P})\geqslant\psi\geqslant\frac{ad(ad-1)}{2}-(ad-1)=\frac{a^2d^2-3ad+2}{2}$$

由于簇大小为 ad，且它是森林的一部分，因此最多包含 $ad-1$ 条边。最坏情况下，$\mathcal{P}'$会割断所有 $ad-1$ 条边。不过，我们得益于较小的簇

大小。我们有

$$\begin{aligned}\operatorname{cost}(F[S],\mathcal{P}')=\chi+\psi &\leqslant ad-1+\frac{d(d-1)}{2}+\frac{(a-1)d\cdot((a-1)d-1)}{2}\\ &=\frac{2d^2+a^2d^2-2ad^2+ad-2}{2}.\end{aligned}$$

因此，$\mathcal{P}'$ 更便宜，便宜的幅度为

$$\operatorname{cost}(F[S],\mathcal{P})-\operatorname{cost}(F[S],\mathcal{P}')\geqslant\frac{2ad^2-2d^2-4ad+4}{2}=ad(d-2)-d^2+2.$$

这一项随 a 增大。由于 $a\geqslant 2$，代入 $a=2$ 后，我们由此得到 $\mathsf{cost}(F[S],\mathcal{P})-\mathsf{cost}(F[S],\mathcal{P}')\geqslant d^2-4d+2$ 的代价下界。对于 $d\geqslant 2$，该下界随 d 增大，并且当 $d>3$ 时为正。这意味着，如果 $d>3$，则没有包含大小超过 d 的簇的聚类具有最小代价；这表明所有最优聚类都只包含大小为 d 的簇。

最后，我们需要讨论 $d=3$ 这种情况，即颜色比例为 1 : 2 或 1 : 1 : 1。此时，d^2-4d+2 的值为 -1。但是，如果我们不任意划分，而是至少保留一条边不被割掉，就会得到正向变化。注意，这意味着少割掉一条边，并多保留一条边，因此 $\mathsf{cost}(T[S],\mathrm{P}')$ 的上界减少 2，所以现在 P 更便宜。因此，假设存在一条边 $\{u,v\}$，使得 $c(u)\neq c(v)$。那么，将 S 划分为 $\{u,v,w\}$ 和 $S\setminus\{u,v,w\}$，并选择某个顶点 $w\in S\setminus\{u,v\}$，使连通分量 $\{u,v,w\}$ 满足公平性，就能得到代价更低的聚类。如果不存在这样的边 $\{u,v\}$，那么 $T[S]$ 就不是连通的。这意味着，当颜色比例为 1 : 1 : 1 时，边数至多为 $3a-3$，因为不同颜色的顶点之间没有边相连，并且每种颜色有 a 个顶点；由于森林结构，每个顶点最多通过 $a-1$ 条边与其他顶点相连。类似地，当颜色比例为 1 : 2 时，边数至多为 $3a-2$。因此，$\mathsf{cost}(T[S],\mathrm{P})$ 的下界增加 1。同时，即使 P' 割掉所有边，它也至多割掉 $3a-2$ 次，因此比预期至少便宜 1。由此，无论如何割边，在这种情况下都有 $\mathsf{cost}(T[S],\mathrm{P}')<\mathsf{cost}(T[S],\mathrm{P})$。◀

注意， 引理 4 未对两种颜色按 1:1的比例的情况作出说明。

4.2 二分图

我们能够将关于树的研究结果部分推广到二分图。我们证明，始终存在一种最小成本公平聚类，其簇大小由颜色比例决定。然而，在二分图中，也可能存在具有更大簇的最小成本聚类。我们先讨论两种颜色按 1:1的比例的情况，然后再推广到其他比例。

▶ **引理 10**。 设 $G=(A\cup B,E)$ 是一个两种颜色按$1:1$的比例构成的二分图。那么，存在一种 G 最小成本公平聚类，其中没有簇包含超过 *2* 个顶点。此外，每一种最小成本公平聚类都可以在线性时间内转换为一种最小成本公平聚类，使得所有簇都至多包含 *2* 个顶点。如果 G 是森林，那么最小成本公平聚类中的任何簇大小都不超过 *4*。

证明。 注意，由于公平性约束，每个公平聚类仅由大小相等的簇组成。我们通过证明以下事实来证明这两个陈述：在每个至少包含 4 个顶点的簇中，总能找到两个顶点，将它们与簇中的其余顶点划分开后，代价不会增加且公平性仍得以保持。

令$\mathcal{P}$ 为一个聚类， $S\in\mathcal{P}$ 为一个包含 $|S|\geqslant 4$的簇。令 $S_A=S\cap A$ 和$S_B=S\cap B$。假设存在 $a\in S_a$ 和 $b\in S_b$，使得 a 和 b 的颜色不同。

于是，将 S 划分为$\{a,b\}$和 $S\setminus\{a,b\}$后得到的聚类 P' 是公平的。现在，我们分析每一对顶点 u、v、$u\neq v$所产生的相关聚类代价如何变化。对于 u 和 v 中至多只有一个顶点属于 S的每一对顶点，代价都不会变化。此外，若$\{u,v\}=\{a,b\}$或$\{u,v\}\subseteq S\setminus\{a,b\}$，代价也不会变化。至多有 $|S_A|-1+|S_B|-1=|S|-2$ 条边，一端位于$\{a,b\}$，另一端位于 $S\setminus\{a,b\}$。这些边在 P' 中被割断，但在 P 中没有，因此最多增加 $|S|-2$的代价。然而，由于二部结构，$S\setminus\{a,b\}$中有 $|S_A|-1$ 个顶点与 a 没有边相连，而 $S\setminus\{a,b\}$中有 $|S_B|-1$ 个顶点与 b没有边相连。这些$|S|-2$ 个顶点在 P 中总共产生 $|S|-2$ 的代价，而在 P'中不产生代价。这抵消了 P 中所有割边的代价，因此划分聚类不会增加代价。

如果不存在 $a\in S_a$ 和 $b\in S_b$ 使得 a 和 b 颜色不同，那么要么是$S_A=\emptyset$，要么是 $S_B=\emptyset$。在这两种情况下，S内部都没有边，因此以任意公平的方式划分聚类都不会增加代价。

通过在任意公平聚类中反复划分较大的簇，我们最终可以得到一个最小代价公平聚类，其中所有簇都恰好由两个顶点组成。

现在，假设 G 是一个森林，并且存在一个最小代价聚类$\mathcal{P}$，其中有某个簇 $S\in\mathcal{P}$，使得对于某个 $a\in\mathbb{N}_{>2}$，$|S|>2a$。考虑一个新的聚类$\mathcal{P}'$，它是通过将 S 划分为$\{u,v\}$ 和 $S\setminus\{u,v\}$ 得到的，其中 u 和 v 是两个颜色不同的任意顶点，它们与 S 中的其他顶点至多有一条边相连。由于森林结构，并且每种颜色都有 $\frac{S}{2}$ 个顶点，这样的两个顶点总是存在。于是，$\mathcal{P}'$仍然是一个公平聚类。注意，对于在 S 中至多有一个端点的每条边和非边，在两种聚类中的代价都相同。令 ψ 表示$\mathcal{P}$在 $G[S]$中的簇内代价。关于两个端点都在 S中的边和非边，我们知道

$$\mathrm{cost}(G[S],\mathcal{P})\geqslant\psi\geqslant\frac{2a(2a-1)}{2}-(2a-1)=2a^2-3a+1$$

因为簇包含 $2a$ 个顶点，且由于森林结构，最多有 $2a-1$ 条边。在最坏情况下，$\mathcal{P}'$ 会割断 2 条边。不过，较小的簇大小对我们有利。我们有

$$\mathrm{cost}(G[S],\mathcal{P}')\leqslant 2+\psi\leqslant 2+1+\frac{2(a-1)(2(a-1)-1)}{2}-(2a-1-2)=2a^2-5a+6.$$

因此，$\mathcal{P}$ 的代价至少比$\mathcal{P}'$, 高 $2a-5$，而根据 $a>2$，这个差值为正。因此，在每个最小成本公平聚类中，所有簇的大小均为 4 或 2。◀

如果图中的颜色比例不是 1 : 1，我们也采用类似的策略。不过，此时我们必须从一个簇中拆分出超过 2 个顶点。为确保聚类成本不增加，我们需要论证，可以从二分图的两侧以某种平衡的方式选取这些顶点。

▶ **引理 5.** 设 $G=(A\cup B,E)$ 是一个二分图，其颜色比例为 $k\geqslant 2$ $c_1:c_2:\ldots:c_k$，其中对所有 $i\in[k]$，$c_i\in\mathbb{N}_{>0}$ 均成立，且 最大公约数$(c_1,c_2,\ldots,c_k)=1$。那么，存在一种最小成本公平聚类，使其所有簇的大小均为 $d=\sum_{i=1}^{k}c_i$。此外，可以在线性时间内将每一种簇较大的最小成本公平聚类转换为一种最小成本公平聚类，使所有簇均包含不超过 d 个顶点。

证明。 由于公平性约束，每个公平聚类仅由大小为 ad 的簇组成，其中 $a\in\mathbb{N}_{>0}$。我们通过证明一个大小至少为$2d$ 的簇可以被拆分，且代价不会增加、公平性仍然得到保持，来证明这些陈述。

设$\mathcal{P}$ 是一个聚类，$S\in\mathcal{P}$ 是一个簇，其中 $|S|=ad$ 对某个 $a\geqslant 2$成立。设 $S_A=S\cap A$，以及 $S_B=S\cap B$，不失一般性，$|S_A|\geqslant|S_B|$。我们的证明分为三个步骤。

16 公平相关聚类

首先，我们证明存在一个公平的 $S \subseteq S$，使得 $|S| = d$ 且 $|S \cap A| \geqslant |S \cap B|$。然后，我们通过将 S 中的顶点替换为 $S_B \setminus S$ 中的顶点，构造一个公平集合 $S \subseteq S$，使得 $|S| = d, |S_A| \geqslant |S_B|$仍然成立，其中 $S_A = S \cap A$ 且 $S_B = S \cap B$，并且还满足 $|\widehat{S}_A| - |\widehat{S}_B| \leqslant |S_A| - |S_B|$。最后，我们证明将 S 划分为 S 和 $S \setminus S$ 不会增加聚类成本。

接着我们注意到，得到的聚类是公平的，因此引理中的结论成立：任何含有超过 d 个顶点的簇的公平聚类，都可以通过反复划分较大的簇，转换为代价至多相同、且仅含有大小为 d 的簇的公平聚类。

对于第一步，假设不存在这样的 $S \subseteq S$，即我们只能从 S_A 中选取 $s < \frac{d}{2}$ 个顶点，而不能选取超过 c_i 个任一颜色 $i \in [k]$ 的顶点。令 s_i 表示对于所有 $i \in [k]$，这 s 个顶点中颜色为 i 的顶点数量。那么，如果 $s_i = 0$，则 S_A 中没有颜色为 i 的顶点，否则我们就可以将相应的顶点加入 S。类似地，如果 $s_i < c_i$，那么 S_A 中颜色为 i 的顶点至多有 s_i 个。如果我们选取 $s_i = c_i$ 个顶点，那么该颜色的 $ac_i = as_i$ 个顶点可能全部都在 S_A 中。因此，$|S_A| \leqslant \sum_{i=1}^{k} as_i = as < \frac{ad}{2}$。这与 $S_A \geqslant S_B$ 矛盾，因为 $|A| + |B| = ad$。于是，存在一个大小为 d 的公平集合 S，使得 $|S \cap S_A| \geqslant |S \cap S_B|$。

现在进行第二步，我们将 S 转换为 S。注意，如果 $|S_A \setminus S| \geqslant |S_B \setminus S|$，那么只需令 $S = S$。否则，我们将 $S \cap S_A$ 中的一些顶点替换为 $S_B \setminus S$ 中颜色相同的顶点。我们需要证明，经过这一步后，从 S_A 中取出的顶点数仍不少于从 S_B 和 $|S_A| - |S_A| \geqslant |S_B| - |S_B|$ 中取出的顶点数。令

$$\delta = |S_B \setminus \widetilde{S}| - |S_A \setminus \widetilde{S}| > 0.$$

回忆一下，$|S_A| \geqslant |S_B|$，因此 $\delta \leqslant |S \cap A| - |S \cap B|$。然后，我们通过将 $\widetilde{S} \cap S_A$ 中的顶点替换为 $S_B \setminus \widetilde{S}$ 中颜色相同的顶点，从 S 构造出 S。如果存在这样的 $\frac{\delta}{2}$ 个顶点，则有 $|S_A \setminus \widehat{S}_A| = |S_B \setminus \widehat{S}_B|$ 和 $|\widehat{S}_A| \geqslant |\widehat{S}_B|$。因此，$\widehat{S}$ 满足要求。

假设不存在这样的 $\frac{\delta}{2}$ 顶点，而我们只能替换 $s < \frac{\delta}{2}$ 顶点。令 s_i 表示对于所有 $i \in [k]$，这些顶点中颜色为 i 的顶点数。根据与上文类似的论证，并且由于在 $S \setminus S$中，每种颜色 i 都只有$(a-1)c_i$ 个顶点，因此我们有

$$|S_B \setminus \widehat{S}| \leqslant \sum_{i=1}^{k} (a-1)s_i = (a-1)s < \frac{(a-1)d}{2}.$$

这与 $|S_B \setminus S| > |S_A \setminus S|$ 矛盾，因为 $|(S_A \cup S_B) \setminus S| = (a-1)d$。因此，总是有足够的顶点来构造 S。

最后，我们通过分析每个顶点对$\{u, v\} \in \binom{V}{2}$的变化，证明将 S 划分为 S 和 $S \setminus S$ 不会增加代价。如果 $u \in S$和 $v \in S$都不成立，则顶点对不受影响。此外，如果$\{u, v\} \subseteq S$或$\{u, v\} \subseteq (S \setminus S)$成立，代价也不会改变。对于其余顶点对，数量至多为

$$|\widehat{S}_A| \cdot |S_B \setminus \widehat{S}_B| + |\widehat{S}_B| \cdot |S_A \setminus \widehat{S}_A| = |\widehat{S}_A| \cdot |S_B| + |\widehat{S}_B| \cdot |S_A| - 2\left(|\widehat{S}_A| \cdot |\widehat{S}_B|\right)$$

将 S 划分为 S 和 $S \setminus S$时被切断的边。同时，还存在

$$|\widehat{S}_A| \cdot |S_A \setminus \widehat{S}_A| + |\widehat{S}_B| \cdot |S_B \setminus \widehat{S}_B| = |\widehat{S}_A| \cdot |S_A| + |\widehat{S}_B| \cdot |S_B| - |\widehat{S}_A|^2 - |\widehat{S}_B|^2$$

一些顶点对彼此不连通，且在$\mathcal{P}'$ 中被分配到不同的簇，但在$\mathcal{P}$中并非如此。因此，$\mathcal{P}$ 比$\mathcal{P}'$ 至少多

$$\begin{aligned}\operatorname{cost}(\mathcal{P}) - \operatorname{cost}(\mathcal{P}') &\geqslant |\widehat{S}_A| \cdot |S_A| + |\widehat{S}_B| \cdot |S_B| - |\widehat{S}_A| \cdot |S_B| - |\widehat{S}_B| \cdot |S_A| \\ &\quad - \left(|\widehat{S}_A|^2 - 2\left(|\widehat{S}_A| \cdot |\widehat{S}_B|\right) + |\widehat{S}_B|^2\right) \\ &\geqslant \left(|\widehat{S}_A| - |\widehat{S}_B|\right) \cdot (|S_A| - |S_B|) - \left(|\widehat{S}_A| - |\widehat{S}_B|\right)^2 .\end{aligned}$$

这是非负的，因为 $|S_A| \geqslant |S_B|$ 和 $|S_A| - |S_B| \leqslant |S_A| - |S_B|$。因此，以这种方式划分一个簇绝不会增加代价。◀

然而，与森林中的情况不同，在二分图上的最小成本公平聚类中，颜色比例并不能限定最大簇大小，它只说明存在一种簇大小有界的最小成本公平聚类。设 $G = (R \cup B, \{\{r, b\} \mid r \in R \wedge b \in B\})$ 是一个完全二分图，满足 $|R| = |B|$，其中 R 中的所有顶点都是红色，B 中的所有顶点都是蓝色。那么，G 中的所有公平聚类代价都相同，包括由单个簇 $S = R \cup B$构成的聚类。这是因为可以采用与引理 10 最后一部分类似的论证：聚类切断的每条边，都恰好由一对不相邻的顶点予以补偿，而这对顶点随后不再属于同一个簇。

5 困难性结果

本节证明了在各种限制条件下，公平相关聚类是 NP难的。

5.1 森林和树

已知最小代价聚类中各簇的固定大小后，我们能够证明，即使将问题限制在某些森林和树的实例上，它仍然出奇地困难。

为了证明在各种假设下公平相关聚类的困难性，我们将其归约到强 NP 完全的 3-划分问题 [29]。

3-划分

输入： $n = 3p$ 其中 $p \in \mathbb{N}$、正整数 a_1，a_2、……、a_n，以及 $B \in \mathbb{N}$ 满足 $\frac{B}{4} < a_i < \frac{B}{2}$，且 $\sum_{i=1}^{n} a_i = pB$。

任务： 判断是否存在将数字 a_i 划分为三元组的方法，使得每个三元组的和都为 B。

我们的第一个归约证明了多种形式的森林都具有困难性。

▶**定理 11.** 公平相关聚类在具有两种颜色且颜色比例为1: c 的森林上是NP困难的。即使任意限制森林中树的形状，只要对于每个 $a \in \mathbb{N}$ 都能构造出一棵具有 a 个顶点的树，该问题仍然是NP困难的。

证明。 我们从 3-划分 归约。对于每个 a_i，我们构造一棵形状任意的树，其中包含 a_i 个红色顶点。此外，我们令其包含 p 个孤立蓝色顶点。注意，

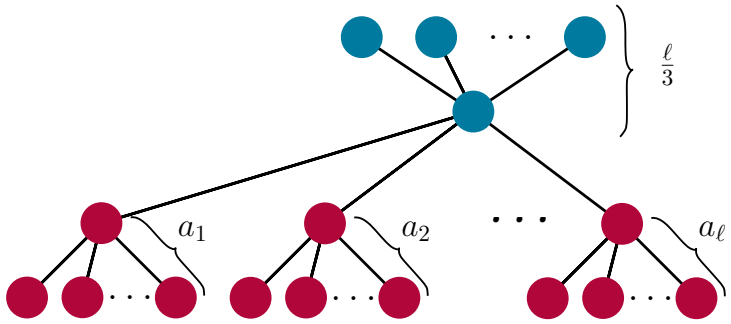

图3 从 3-划分归约到公平相关聚类中直径为 4 的树 N 。符号沿用定理12中的定义。

蓝色顶点与红色顶点的比例为 1 : B。现在我们证明存在一个公平聚类 $\mathcal{P}$ 使得

$$\operatorname{cost}(\mathcal{P}) = p \cdot \frac{B(B+1)}{2} - p(B-3)$$

当且仅当给定实例是 3-划分的肯定实例时。

如果我们有一个 3-划分的肯定实例，那么就存在一个将树的集合划分为大小为p的簇的划分 B。将蓝色顶点任意分配给各自唯一的簇后，我们便得到一个公平划分。由于簇之间没有边，且每个簇由 $B+1$ 个顶点和 $B-3$ 条边组成，因此该划分的代价为 $p \cdot {}^{B}{}_{(B+1)\,2} - p(B-3)$。

对于另一个方向，假设存在一个代价为 ${}^{B}{}_{(B+1)\,2} - p(B-3)$ 的公平聚类。根据引理 4，每个簇恰好由一个蓝色顶点和 B 个红色顶点组成。每个簇需要 ${}^{B}{}_{(B+1)\,2}$ 条边，而图中只有 $p(B-3)$ 条边。因此，仅簇内代价就至少为 $p \cdot {}^{B}{}_{(B+1)\,2} - p(B-3p)$。这意味着簇间代价为 0，即划分没有切断树内部的任何边。由于所有树的大小都大于 $\frac{B}{4}$ 且小于 $\frac{B}{2}$，这意味着每个簇恰好由一个蓝色顶点和三棵未切割的树组成，总计包含 B 个顶点。这样一来，这种聚类便给出了 3-划分的一个解，因此我们的实例是一个肯定实例。

由于构造该图所需的时间关于实例规模为多项式时间，这便得出了我们的困难性结果。◀

请注意，对于路径森林，即最大度为 2 的森林，该困难性尤其成立。

通过下面的定理，我们调整定理 11 的证明，以说明即使图是连通的，该困难性仍然存在。

▶**定理12**。 树上的公平相关聚类 当直径为 *4* 且两种颜色的比例为 1: c 时是 NP-困难的。

证明。 我们从 3-划分 进行归约。对于每个 a_i，我们构造一个包含 a_i 个红色顶点的星形图。此外，我们构造一个包含 p 个蓝色顶点的星形图。将蓝色星形图的中心 v 与所有红色星形图的中心相连，便得到一棵直径为 4 的树。该构造如图3所示。注意，蓝色顶点与红色顶点的比例为 1: B。现在我们证明，存在一个公平聚类$\mathcal{P}$，使得

$$\operatorname{cost}(\mathcal{P}) \leqslant \frac{pB^2 - pB}{2} + 7p - 7$$

当且仅当给定实例是 3-划分的肯定实例时。

如果我们有一个 3-划分的肯定实例，那么就存在一种将星形图集合划分为p个簇的方法，每个簇的大小为 B，且由三个星形图组成。随后，将蓝色顶点任意分配到各自唯一的簇中，我们便得到一个公平划分。我们首先计算簇间代价 χ。如果一条边连接两个蓝色顶点或两个红色顶点，我们分别称其为蓝色边或红色边。如果一条边连接一个蓝色顶点和一个红色顶点，则称其为蓝红边。所有 $p-1$ 蓝色边都会被割断。此外，所有连接 v （蓝色星形图的中心）和红色顶点的边都会被割断，但分配给 v 的三个星形图除外。这会导致额外 $3p-3$ 个割，因此簇间代价为 $\chi = 4p-4$。每个簇都包含 $B+1$ 个顶点和 $B-3$ 条边，但包含 v 的簇有 B 条边。因此，簇内代价为

$$\psi = p\left(\frac{B(B+1)}{2} - B + 3\right) - 3 = \frac{pB^2 - pB}{2} + 3p - 3.$$

将簇内代价与簇间代价相加，即得到所需的代价

$$\mathrm{cost}(\mathcal{P}) = \chi + \psi = \frac{pB^2 - pB}{2} + 7p - 7.$$

对于另一个方向，假设存在一个代价至多为 $\frac{pB^2-pB}{2} + 7p - 7$ 的公平聚类。由于共有 $p(B+1)$ 个顶点，根据引理 4，恰有 p 个簇，每个簇都恰好由一个蓝色顶点和 B 个红色顶点组成。令 a 表示 v 所在簇中的红色中心顶点数。我们证明 $a=3$。为此，令 χ_r 表示被切断的红色边数。此外，我们还切断 $p-1$ 条蓝色边和 $3p-a$ 条蓝红边。因此，该聚类的簇间代价为 $\chi = \chi_r + 4p - a - 1$。对于簇内代价，不存在缺失的蓝色边；由于 v 是唯一一个带有蓝红边的蓝色顶点，因此有 $(p-1)B + B - a = pB - a$ 条缺失的蓝红边。最后，我们需要 $p\cdot\binom{B}{(B-1)/2}$ 条红色边，但图中只有 $pB - 3p$ 条红色边，其中 χ_r 条被切断。因此，至少有 $p\cdot\binom{B}{(B-1)/2} - pB + 3p + \chi_r$ 条缺失的红色边，簇内代价总计为 $\psi \geqslant p\cdot\binom{B}{(B-1)/2} + 3p + \chi_r - a$。由此得到总代价为

$$\mathrm{cost}(\mathcal{P}) = \chi + \psi \geqslant \frac{pB^2 - pB}{2} + 7p + 2\chi_r - 2a - 1.$$

正如我们假设 $\mathrm{cost}(\mathcal{P}) \leqslant \frac{pB^2-pB}{2} + 7p - 7$，我们有 $2\chi_r - 2a + 6 \leqslant 0$，这意味着 $a \geqslant 3$，因为 $\chi_r \geqslant 0$。此外，$\chi_r \geqslant \binom{aB}{4} - (B-a)$，因为每个所选红色中心都至少连接着 $\binom{B}{4}$ 个红色顶点，但总共只有 $B-a$ 个红色顶点可以被放入其中心所在的簇中。因此，我们有 $\binom{aB}{2} - 2B + 6 = \binom{(a-4)}{2}B + 6 \leqslant 0$，从而推出 $a<4$，并证明了我们的 $a=3$ 主张。此外，由于 $a=3$，我们得到 $\chi_r \leqslant 0$，这意味着没有红色边被切断，因此每个红色星形图都完全包含在一个簇中。鉴于每个红色星形图的大小至少为 $\binom{B}{4}$，至多为 $\binom{B}{2}$，这意味着每个簇恰好由三个完整的红色星形图组成，总计包含 B 个红色顶点，因此可得出一个 3-划分实例的解。

由于构造图只需实例规模的多项式时间，且所构造的树的直径为 4，因此这就证明了我们的困难性结果。◀

定理 11 和定理 12 的证明与 [27,定理 2], 的困难性证明思路相同，后者同样通过从 3-划分归约，证明 k-平衡划分问题的困难性。在该问题中，任务是将无色图的顶点划分为 k 个大小相等的簇 [27]。

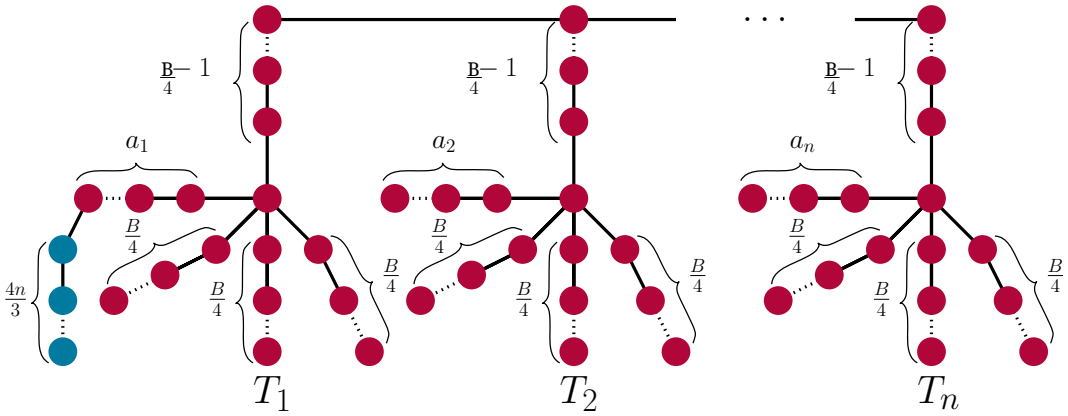

图4 从3-划分到公平相关聚类的归约中最大度为5的树（定理13）。

k-平衡划分

输入： 图 $G=(V,\ E)$，$k\in[n]$。

任务： 找出 V的一个划分 $\mathcal{P}$，使 $|\{\{u,\ v\}\in E\mid\mathcal{P}[u]\neq\mathcal{P}[v]\}|$最小，同时满足 $|\mathcal{P}|=k$且 $|S|\leqslant\lceil\frac{n}{k}\rceil$对所有 $S\in\mathcal{P}$均成立。

k-平衡划分与森林上的公平相关聚类有关，因为根据引理4和引理10，聚类必须将森林划分成等规模簇。因此，在森林上，我们可以将公平相关聚类视为 k-平衡划分的公平变体。根据 [27, 定理 8]，在度为 5 的树上， k-平衡划分是 NP难的。在他们的证明中，Feldmann 和 Foschini [27]从 3-划分归约而来。我们稍微调整他们的构造，将这一结果推广到公平相关聚类。

▶**定理13.** 公平相关聚类 在度数至多为 *5*、两种颜色比例为 $1:c$ 的树上是NP难的。

证明。 我们从 3-划分归约而来；即使限定为 NP-难实例，其中 B 是 4 的倍数，该问题仍然是强 NP 难的，因为我们可以将每个整数乘以 4，从而为任意实例构造一个等价实例。因此，假设给定一个 3-划分 实例，使得 B 是 4的倍数。我们通过将每个 公平相关聚类 的 a_i 表示为一个构件，来构造一张图；对于 $i\in[n]$ ，构件为 T_i。每个构件都有一个中心顶点，它与五条路径的端点相连：一条长度为 a_i的路径、三条长度为 $\frac{B}{4}$ 的路径，以及一条长度为 $\frac{B}{4}-1$的路径。接着，对于 $i\in[n-1]$，我们用一条边连接构件 T_i 和 T_{i+1} 中长度为 $\frac{B}{4}-1$ 的路径的悬挂端点。到目前为止，该构造与 Feldmann 和 Foschini 的构造类似 [27]。我们将迄今添加的所有顶点都染成红色。然后，添加一条由 $\frac{4n}{3}$ 个蓝色顶点组成的路径，并用一条边将其连接到任意一个度为 1 的顶点。所得图如图4所示。

注意，该构造可在多项式时间内完成，并且我们得到一个度为 5 的图。现在我们证明，它存在一个公平聚类 $\mathcal{P}$ ，使得

$$\operatorname{cost}(\mathcal{P})\leqslant\frac{(B-2)n}{2}+\frac{20n}{3}-3$$

当且仅当给定实例是 3-划分的肯定实例时。

假设我们有一个 3-划分的肯定实例。我们割断连接不同构件的边，以及连接 a_i-路径与星形图中心的边。这样，对于每个 $i \in [n]$，我们得到 n 个大小为 B 的连通分量和 1 个大小为 a_i 的连通分量。后者可以在不进行额外割边的情况下合并为 $p = \frac{n}{3}$ 个大小为 B 的簇。接下来，我们割断蓝色顶点之间的所有边，并将每个簇分配一个蓝色顶点。需要注意的是，已经与红色簇连通的蓝色顶点应分配给该簇。通过这种方式，我们得到一个公平聚类，其簇间代价为 $\chi = n - 1 + n + \binom{4n}{3} - 1 = \binom{10n}{3} - 2$；根据引理 3，这给出 $\mathrm{cost}(\mathcal{P}) = (B-2)_2^{\,n} + \binom{20n}{3} - 3$。

对于反方向，设存在一个代价至多为 $(B-2)_2^{\,n} + \binom{20n}{3} - 3$ 的最小成本公平聚类。如 $\sum_{i=1}^{n} a_i = \frac{nB}{3}$，该图由 $\frac{4n}{3} \cdot B$ 个红色顶点和 $\binom{4n}{3}$ 个蓝色顶点组成。根据引理 4，$\mathcal{P}$ 因此由 $\binom{4n}{3}$ 个簇组成，每个簇包含一个蓝色顶点和 B 个红色顶点。因此，$\mathcal{P}$ 必须割断蓝色路径上的 $\binom{4n}{3} - 1$ 条边。此外，$\mathcal{P}$ 必须将红色顶点划分为大小为 B 的集合。根据 [27, 引理 9]，这至少需要 $2n - 1$ 次割边。这将簇间代价限定为 $\chi \geqslant 2n - 1 + \binom{4n}{3} - 1 = \binom{10n}{3} - 2$，从而得到如上所示的相关聚类代价 $(B-2)_2^{\,n} + \binom{20n}{3} - 3$，因此我们知道不会再有其他边被割断。此外，移除后不会留下大小大于 B 的红色连通分量的唯一最小规模边集合，是连接各构件的 $n - 1$ 条边以及连接 a_i 条路径与中心顶点的 n 条边集合 [27, 引理 9]。因此，$\mathcal{P}$ 必须恰好割断这些边。由于没有其他边被割断，a_i 条路径可以组合成大小为 B 的簇，而无需额外割边，因此给定实例必定是 3-划分的肯定实例。◀

5.2 路径

定理 11 表明，即使在路径森林中，公平相关聚类仍然是 NP难。将问题限制为单条连通路径的实例时，它与项链分割问题 [5, 6]密切相关。

离散项链分割

输入： 开放项链 N，表示为一条包含 $n \cdot k$ 颗珠子的路径，每颗珠子属于以下其中一种 t 种颜色，使得对于每种颜色 i，都有 $a_i \cdot k$ 颗该颜色的珠子 一些 $a_i \in \mathbb{N}$。

任务： 切割项链，使所得区间能够划分为 k 个集合，每个集合中每种颜色的珠子数量都相同。

与路径上的公平相关聚类相比，除了命名不同之外，唯一的区别是簇的数量 k 是明确给出的。不过，根据引理4和引理10，对于公平相关聚类，我们也可以隐式地得到这个值。然而，Alon 和 West [6] 并未以构造性方式最小化公平划分所需的割数，而是以非构造性方式证明：如果有 t 种颜色，并且要求划分恰好由 k 个集合组成，且每种颜色的顶点数量相同，那么总能找到一个至多需要 $(k-1) \cdot t$ 次割的划分。因此，这并不能直接帮助我们解决这个优化问题。

此外，路径上的公平相关聚类与单词涂色问题（PPW）的 1-正则双色变体有关。对于 PPW，给定一个单词以及一组颜色，并针对每种符号和颜色规定应按相应颜色着色的符号数量。任务是找到一种满足所有要求的着色，并使相邻字母之间的颜色变化次数最少 [24]。

单词涂色问题（PPW）

输入： 单词 $w = w_1, w_2, \ldots, w_n \in \Sigma^*$、颜色数量 $k \in \mathbb{N}_{>0}$，以及约束函数 $r\colon \Sigma \times [k] \to \mathbb{N}$，使得对于 w 中使用的每个符号 s，$w[s]$ 次出现，我们有 $\sum_{i=}^{k} {}_{1} r(s, i) = w[s]$。

任务： 求一个赋值函数 $f\colon [n] \to [k]$，将颜色赋给 w 中的字母使得对于每个符号 $s \in \Sigma$ 和颜色 $i \in [k]$，着色都满足需求函数，即 $|\{j \in [n] \mid w_j = s \wedge f(j) = i\}| = r(s, i)$。该赋值 f 应使颜色变化的数量 $|\{j \in [n-1] \mid f(j) \neq f(j+1)\}|$ 最小。

例如，设 $w = aabab$ 和 $r(a, 1) = 2$, $r(a, 2) = r(b, 1) = r(b, 2) = 1$。那么，赋值 f 其中 $f(1) = f(2) = f(3) = 1$ 以及 $f(4) = f(5) = 2$，满足了要求，且只有 1 次颜色变化。

每个符号恰好出现两次且有两种 PPW 颜色、每种颜色都要求包含每个符号各一次的 PPW 实例，称为*1*-正则双色，并已证明其为 NP 难，甚至是 APX-困难 [14]。据此，我们证明了即使在路径上，公平相关聚类也是 NP 难的。

▶**定理 14**。 公平相关聚类在路径上是NP难的，即使限定为每种颜色恰好有 *2* 个顶点的实例也是如此。

证明。 我们从 1-正则双色PPW归约。设 $w = s_1 s_2, \ldots\ldots, s_\ell$。我们用 $\frac{\ell}{2}$种不同的符号表示为 $\frac{\ell}{2}$ 种颜色，并构造一条长度为 ℓ的路径，其中每种符号类型都由一种独特的颜色表示。根据引理 4，任何最优的公平相关聚类解都会将路径划分为两个簇，每个簇都恰好包含每种颜色一次；同时，根据引理 3，使割的数量（簇间代价）最小。由于这恰好等价于将单词中的字母分配给两种颜色并使颜色变化次数最少，因此得到我们的困难性结果。◀

然而，APX-困难性并不能传递，因为尽管割的数量（簇间代价）与相关聚类代价之间存在联系，但这两种度量并不相同。事实上，根据定理 42，公平相关聚类在森林上存在多项式时间近似方案，因此，若路径上的问题是 APX-困难的，就意味着 P = NP。

顺便一提，值得注意的是，对于路径上的每个公平相关聚类实例，我们都可以构造一个等价的 PPW 实例（尽管并非所有实例都是 1-正则双色），方法是用颜色表示符号，并用簇表示 PPW 颜色。

我们注意到，如果颜色数量较少（例如只有两种颜色），或许可以高效地求解路径上的公平相关聚类问题。对于只有两个字母的 PPW，已有一个 NP-困难性结果 [24],，但从这些实例进行归约并不像上文那样简单，因为其要求会导致不公平聚类。

5.3 超越树

根据定理12，即使图是直径为 4 的树，公平相关聚类仍然是 NP难的。这里我们将证明，如果允许图中包含环，那么直径为 2 时问题就已经是 NP难的。此外，这也恰好形成鲜明对比：直径为 2 的树上的公平相关聚类问题可以在线性时间内求解，具体见第6.1小节。

▶**定理15.** 公平相关聚类在具有两种颜色、颜色比例为 $1:1$ 的直径为 *2* 的图上是NP难的。

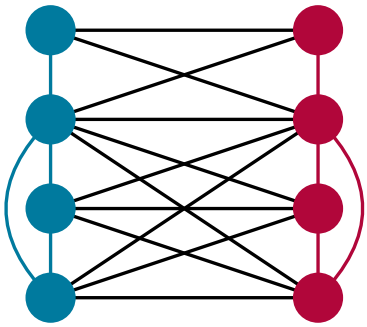

图 5 Ahmadi 等人构造的图 [1]，用于将相关聚类归约为公平相关聚类。蓝色顶点和边对应原图 $G = (V, E)$，红色顶点和边对应其镜像，即 V' 和 E'，黑色边则对应 $\widetilde{E}$。

证明。簇编辑是相关聚类的一种等价表述，在直径为 2 [9]的图上是 NP 难的。此外，Ahmadi 等人 [1] 给出了从相关聚类到公平相关聚类的归约，颜色比例为 1:1。他们证明，可以通过求解图 $G = (V, E)$ 上的公平相关聚类，来求解图 $G' = (V \cup V', E \cup E' \cup E)$ 上的相关聚类，其中前者是后者的镜像。V 中的顶点被染成蓝色，V' 中的顶点被染成红色。形式化地，$\sim V' = \{u' \mid u \in V\}$ 且 $E' = \{\{u', v'\} \mid \{u, v\} \in E\}$。此外，$E$ 将每个顶点~与其镜像顶点以及相邻顶点的镜像相连，即 $E = \{\{u, u'\} \mid u \in V\} \cup \{\{u, v'\} \mid u \in V \wedge v' \in V' \wedge \{u, v\} \in E\}$，见图 5。

注意，若 G 的直径为 2，则 G' 的直径也为 2，理由如下。由于每一对顶点 $\{u, v\} \in \binom{V}{2}$ 的距离至多为 2，且 G的顶点和边均有镜像，因此每一对顶点 $\{u', v'\} \in \binom{V'}{2}$ 的距离也至多为 2。此外，每个顶点与其镜像顶点之间的距离为 1。对于任意一对顶点 $u \in V$，$v' \in V'^{\widetilde{}}$，分两种情况讨论。若 $\{u, v\} \in E$，则 $\{u, v'\} \in E$，因此距离为 1。否则，由于 u 与 v 在 G中的距离至多为 2，存在 $w \in V$ 使得 $\{u, w\} \in E$ 且~$\{v, w\} \in E$。因此，$\{u, w'\} \in E$ 且 $\{w', v'\} \in E'$，所以 u 与 v' 之间的距离至多为 2。

由于直径为 2 的图上的相关聚类是 NP难的，并且 Ahmadi 等人 [1] 的归约在输入图的直径为 2 时会构造出一个直径为 2 的图，因此我们证明了该命题。◀

此外，我们证明，在一般图上，公平相关聚类是 NP难的，即使顶点的颜色使得任何公平聚类中的簇数都不超过 2。这与我们在第6.4小节中的算法形成对比：当簇的最大数量为常数时，该算法可以在多项式时间内求解森林上的公平相关聚类。为此，我们从 NP难的二分问题 [29], 进行归约；该问题是 $k = 2$ k-平衡划分问题的特例。

二分
输入： 图 $G = (V, E)$。**任务：** 找出一个划分 $\mathcal{P}$ $= A\ B\quad V\quad \mid u\ v\ \in E \mid u \in \{,\}$，使其最小化 $\{\{,\}\}$，并满足 $A \wedge v \in B \mid$ 约束条件 。 $|A| = |B|$

▶**定理16。**公平相关聚类在颜色比例为1： c 的双色图上是NP难的，即使 $c = \frac{n}{2} - 1$ 且该图连通。

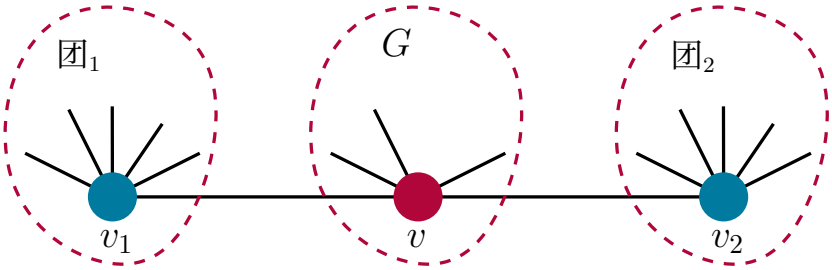

图6 从二分归约到公平相关聚类实例的图，该实例仅包含2个大型簇。中间部分对应于输入图 G，并被着成红色。 $Clique_1$ 和 $Clique_2$ 都是由 $|V|$ 个红色顶点以及各一个蓝色顶点组成的团。

证明。我们从二分归约。设 $G = (V, E)$是一个二分实例，并假设它有偶数个顶点（否则它是一个平凡的否实例）。我们的思路是将 V 中的所有顶点都着为红色，并添加两个团，每个团都由一个蓝色顶点和 $|V|$ 个红色顶点组成，以此确保最小成本公平相关聚类恰好包含两个簇，从而将原图的顶点划分为一个最小代价二分。颜色比例为2： $3|V|$，等于1： $\binom{|V'|}{2} - 1$，其中 V' 是新构造图的顶点集合。我们必须排除最小成本公平相关聚类仅为一个包含整张图的簇这种可能性。为此，我们将新的蓝色顶点 v_1、 v_2 仅连接到一个任意的红色顶点 $v \in V$。我们在图6中展示这一方案。我们首先说明，任何包含两个簇的聚类都比将所有顶点放入同一个簇的代价低。令 $n = |V|$，且 $m = |E|$。设P是一个将所有顶点放入单个簇的聚类。那么，

$$\text{cost}(\mathcal{P}) = \frac{(3n+2)(3n+1)}{2} - \left(m + 2 + 2 \cdot \frac{n(n+1)}{2}\right) = \frac{7n^2}{2} + \frac{7n}{2} - m - 1,$$

由于该簇的大小为 $3n + 2$，因此总计为 $m + 2$ 加上团的边，并且没有边被切断。现在假设我们有一个聚类$\mathcal{P}'$，其簇间代价为 χ'，并且将每个团置于不同的簇中。那么，

$$\begin{aligned}\text{cost}(\mathcal{P}') &= \chi' + 2 \cdot \frac{(\frac{3n}{2}+1)(\frac{3n}{2})}{2} - \left(m - \chi' + \frac{n(n+1)}{2}\right) \\ &= \frac{7n^2}{4} + n - m + 2\chi' \leqslant \frac{9n^2}{4} + n - m + 2,\end{aligned}$$

由于 V 的顶点之间至多有 $\frac{n}{2} \cdot \frac{n}{2}$ 条簇间边，并且从 v 到 v_1 或 v_2中的任一个顶点有一条簇间边，因此 $\chi \leqslant \frac{n^2}{4} + 1$。所以，将所有顶点置于同一个簇中反而更昂贵，增加的代价为

$$\text{cost}(\mathcal{P}) - \text{cost}(\mathcal{P}') \geqslant \frac{7n^2}{2} + \frac{7n}{2} - m - 1 - \left(\frac{9n^2}{4} + n - m + 2\right) = \frac{5n^2}{4} + \frac{5n}{2} - 3$$

比任何包含两个簇的聚类都更大。对于 $n \geqslant 2$，这为正。因此，公平相关聚类总是会返回至少两个簇。此外，由于公平性约束且只有两个蓝色顶点，它恰好生成两个簇。

此外，基于以下原因，它不会从两个团中的任何一个团中切割顶点。由于簇的大小固定，根据引理 3，我们可以着眼于簇间代价来论证，最小成本公平相关聚类只会切割 E 中的边。首先，注意

从两个团中都切割顶点绝非最优，因为只从一个团中切割出差额所需切割的边更少。这也意味着，从该团中切割出的红色顶点至多为 $\frac{n}{2}$ 个；否则，另一个簇中的红色顶点就会超过要求的 $\frac{3n}{2}$ 个。因此，假设从一个团中切割出 $0 < a \leqslant \frac{n}{2}$ 个红色顶点。任何此类解的簇间代价为 $a \cdot (n+1-a) + \chi_E$，其中 χ_E 是在 E 中切割的边数，以便将 V 分成两个大小分别为 $\frac{n}{2}+a$ 和 $\frac{n}{2}-a$ 的簇，从而满足公平划分的要求。我们注意到，不切割团，而是从大小为 $\frac{n}{2}+a$ 的簇中切出 a 个顶点，至多需要切割 $a \cdot \frac{n}{2} + \chi_E$ 条边。由于 $\frac{n}{2} < n+1-a$，这意味着没有最优解会切割这些团。因此，每个最优解都会通过最小代价二分来划分 V。

因此，通过在构造的图上求解公平相关聚类，我们可以在 G 中求解二分问题。此外，构造的图的规模相对于 $|V|$ 为多项式级，因此我们得到了困难性结果。◀

6 算法

第 5 节的结果表明，很难存在一种通用的多项式时间算法来求解树和森林上的公平相关聚类。不过，对于某些类别的实例，我们能够给出高效算法。

6.1 简单情形

首先，我们注意到，当且仅有两种颜色且它们出现的次数相同时，二分图上的公平相关聚类等价于求最大二分匹配问题。这是因为存在一种最小成本公平聚类，使得每个簇的大小均为 2。

▶**定理17**。 在输入图存在一种最小成本公平聚类，且每个簇的基数至多为2的前提下，通过线性时间归约，在比例为1:1 的两种颜色下计算最小成本公平聚类等价于求解最大二分匹配问题。

证明。 设颜色为红色和蓝色。根据假设，存在一种最优聚类，使所有簇的大小至多为 2。由于公平性约束，每个这样的簇恰好由 1 个红色顶点和 1 个蓝色顶点组成。根据引理 3，最低代价由最低的簇间代价达到，即使两个顶点之间存在边的簇数量最大时达到。这正是二分图 $G' = (R \cup B,\ E')$上的匹配问题，其中 R 和 B 分别为红色顶点和蓝色顶点，而$E' = \{\{u,\ v\} \in E \mid u \in R \wedge v \in B\}$。计算出一个最优匹配后，匹配中的每条边定义一个簇，未匹配的顶点则任意打包成公平簇。

对于另一个方向，如果给定二分图匹配的一个实例 $G' = (R \cup B,\ E')$，我们将所有顶点着为 R 红色，将顶点着为 B 蓝色。那么，如上所述，最小成本公平聚类是一个使每个簇中的边数最大化的划分。由于每个顶点恰好属于一个簇，且所有簇都由 R 一个顶点和 B 一个顶点组成，因此这对应于 G' 中的最大二分匹配。◀

根据引理 10，所有二分图都满足定理17的条件。最近的最大流突破 [18] 还给出了一个 $m^{1+o(1)}$时间算法来计算二分图匹配，因此也可用于颜色比例为 1 : 1 的公平相关聚类。对于森林上的公平相关聚类，我们可以做得更好，因为定理17中的归约再次得到一个森林，而标准技术可以在线性时间内求解其中的二分图匹配。为完整起见，我们在此介绍该算法。

26 公平相关聚类

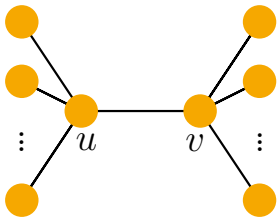

图 7 直径为 3 的每棵树的形状。

▶ **定理18**。 公平相关聚类 在颜色比例为1: 1 的森林上，可在时间 $O(n)$内求解。

证明。我们应用定理17，得到输入图的一个子森林，并需要为其计算最大匹配。我们对每棵树分别运行以下动态规划。我们遍历所有顶点，但只有在已经遍历完某个顶点的所有子节点后，才遍历该顶点（例如，可以使用拓扑排序）。对于每个顶点 v，我们计算以 v 为根的子树中的最大匹配 M_v ，以及假设 v 不在匹配中的情况下，以 v 为根的子树中的最大匹配 M'_v 。我们可以直接得出，M'_v只是每个子节点 u对应的匹配 M_u的并集。除此之外，要么 $M_v = M'_v$，要么在 M_v 中， v 与某个子节点 u之间存在一条边。在后一种情况下，M_v 是{u、 v} M'_u的并集，以及所有子节点 $w \neq u$对应的 M_w的并集。尝试 u 的所有可能取值，并将它们相互比较，同时与 M'_v 进行比较，即可得到 M_v。最后，以 r 为根的树中的最大匹配为 M_r。

每个顶点仅访问一次。如果过程中不直接合并匹配，而只跟踪各自的大小，并在动态规划结束后通过回溯方法找出最大匹配，则每个顶点的时间复杂度与其子节点数量呈线性关系。因此，动态规划可在$O(n)$ 时间内运行。◀

接下来，回顾一下定理12：在直径至少为 4 的树上进行公平相关聚类是 NP难。下一个定理将证明，我们可以高效地解决直径至多为 3 的树上的公平相关聚类问题，因此，除非 P = NP，否则阈值 4 是紧确的。

▶**定理 19**。 公平相关聚类 在直径至多为 *3* 的树上可在时间 $O(n)$内求解。

证明。 直径为 0 或 1 的情况显然，而两种颜色比例为 1: 1 的情况已由定理17处理。因此，假设 $d > 2$ 为公平簇的最小规模。直径为 2 意味着这棵树是一棵星形图。在星形图中，簇间代价等于未与中心顶点置于同一簇中的顶点数。根据引理 4，每个代价最小的聚类都具有最小规模簇。由于在星形图中，所有这些聚类产生的簇间代价均为 $n - d + 1$，根据引理 3，它们的 相关聚类 代价也都相同。因此，输出任意一个包含最小规模簇的公平聚类即可解决该问题。这样的聚类可以在 O(n) 时间内计算得出。

如果我们有一棵直径为 3 的树，它由两个相邻顶点 u，v 组成，并且每个顶点 $w \in V \setminus \{u, v\}$ 都与 u 或 v 相连，而不与其他顶点相连，见图 7。这是因为每个直径为 3 的图都包含一条由四个顶点组成的路径。设中间的两个顶点分别为 u 和 v。这条路径必须是诱导路径，否则该图就不是树。我们可以将其他顶点连接到 u 和 v，而不改变直径；但一旦我们连接

其他位置的一个顶点，直径就会增加。此外，$V \setminus \{u,\ v\}$ 中的顶点之间没有边，否则该图就不是无圈图。

对于聚类，现在有两种可能性，我们分别尝试。要么将 u和 v 放入同一个簇，要么不放在一起。两种情况下，引理 4 都表明，所有簇的大小均为最小值 d。如果 u 和 v 位于同一个簇中，那么所有由大小最小的公平簇组成的聚类，其簇间代价都是 $n-d+2$，因为除了 $d-2$ 个顶点外，其余顶点都必须从 u 和 v中切出。我们在 O(n) 时间内贪心地构造这样的聚类 $\mathcal{P}_1$。如果将 u 和v 放入不同的簇，则通过尽可能多地将它们各自的邻居放入对应的簇中，可以达到最小簇间代价。此后，所有剩余顶点都是孤立的，可用于使这两个簇满足公平性，并在需要时构造更多公平簇。这样的聚类$\mathcal{P}_2$ 也可在 O(n) 时间内计算得到。然后我们返回代价较低的聚类。该聚类是代价最小的公平聚类，因为 u 和 v 要么被放在同一个簇中，要么不放在一起；而在这两种情况下，$\mathcal{P}_1$ 和 $\mathcal{P}_2$ 分别都是代价最小的。◀

6.2 颜色比例 1:2

现在，我们给出适用于森林上的公平相关聚类算法，该算法不要求特定的直径或度。为了解决这些限制较少的实例，首先，我们开发一种算法，用于解决颜色比例为 1:2 的森林上的公平相关聚类问题。

不失一般性，顶点被着色为蓝色和红色，且红色顶点的数量是蓝色顶点数量的两倍。对于大小为 1 的连通分量，如果其中包含的顶点为蓝色，我们称其为 b-分量；如果该顶点为红色，则称其为 r-分量。类似地，对于大小为 2 和 3 的连通分量，我们分别使用 br-分量、rr-分量 和 brr-分量 这些术语。

6.2.1 线性时间尝试

根据引理 4，我们知道，在每个最小成本公平聚类中，每个簇恰好包含 1 个蓝色顶点和 2 个红色顶点。我们的总体思路是采用两个阶段。

在第一阶段，我们以一种方式划分森林 F 的顶点，使得每个簇中至多包含 1 个蓝色顶点和 2 个红色顶点。我们将这种划分称为 划分 F。我们希望采用标准树动态规划，自底向上地收集应属于同一连通分量的顶点；如果继续合并会导致该分量中蓝色顶点超过 1 个或红色顶点超过 2 个，则割去相应的边。我们必须明智地选择割边，但由于最顶层分量中最多只能放置 3 个顶点，因此为了找到割边数量最少的划分，我们只需跟踪有限种可能性。

找到该划分后，我们采用第二阶段：通过合并连通分量，并尽可能少地割去额外的边，找出由该划分组建公平聚类的最佳方式。根据引理 3，具有最小簇间代价的公平划分具有最小的相关聚类代价，因此，这将找到最小成本公平聚类。

遗憾的是，这种方法并没有那么容易奏效。我们发现，第二阶段产生的割数量还取决于 br 和 r 连通分量的数量。

▶ **引理20**。 设 $F=(V,E)$ 是一个有 n个顶点的森林，其中蓝色和红色顶点的比例为$1:2$。假设每个连通分量（按上述含义）中至多有*1*个蓝色顶点和至多*2*个红色顶点。令$\#(br)$ 和$\#(r)$ 分别表示 br连通分量和 r连通分量的数量。那么，割去$\max(0, \frac{\#(br)-\#(r)}{2})$ 条边后，可以合并剩余的连通分量，使得所有簇都恰好由*1*个蓝色顶点和*2*个红色顶点组成。这样的边集可以在n时间内以$O(n)$找到。此外，若割去少于$\max(0, \frac{\#(br)-\#(r)}{2})$ 条边，则无法进行这样的合并。

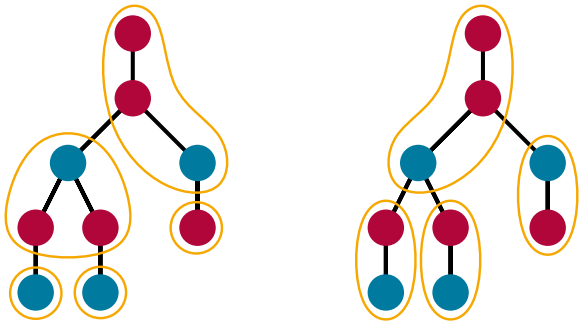

图8 一棵树：采用最小切割数的划分（右）比多切一条边的划分（左）多出 3 个 br-连通分量，少了 1 个 r-连通分量。

证明。 只要可行，就任意将 b-连通分量与 rr-连通分量合并，并将 br-连通分量与 r-连通分量合并。为此无需割边。然后，将剩余的 rr-连通分量拆分，并将拆分后得到的 r-连通分量分别与一个 br-连通分量合并。这样，我们还需多割$\max(0, \frac{\#(br)-\#(r)}{2})$条边，从而得到公平聚类，因为此时每个簇都包含两个红色顶点和一个蓝色顶点。该过程可在$O(n)$时间内完成。

此外，也不存在代价更低的方案。要在不进一步切割的情况下合并每个 br-连通分量，我们需要一个 r-连通分量。共有 $\#(r)$ 个 r-连通分量，而每次切割在移除一个 br-连通分量的同时，至多产生两个 r-连通分量或一个 r-连通分量。因此，需要 $\max(0, \frac{\#(br)-\#(r)}{2})$ 次切割。◀

要使我们的方法奏效，第一阶段必须同时最小化切割数以及 br-连通分量与 r-连通分量之间的差值。然而，这并不容易做到。考虑图8中的这棵树。

在该树中，多切一条割边就会少三个 br-连通分量，多一个 r-连通分量。因此，采用标准树动态规划并不足够：当把这棵树作为某个更大森林或树的子树处理时，我们必须在优化割边数量与优化 br-连通分量和 r-连通分量之间的差值之间作出选择。这里并没有显而易见的答案，因为这一选择取决于图的其余部分会得到多少个 br-连通分量和 r-连通分量。为使我们的方法奏效，我们必须追踪这两种可能性，直至遍历完整个图，这使我们无法实现线性运行时间。

6.2.2 Join子程序

在第一阶段，我们可能会遇到需要跟踪各种子树的多种划分方式的情况。当我们到达这些子树根的父顶点时，就将这些不同的划分方式 Join 在一起。为此，我们给出一个称为 Join 的子程序。我们首先通过以下引理形式化输出，然后介绍各变量的直观含义，最后通过给出算法证明该引理。

▶ **引理 21。** 令 $R_1, R_2, \ldots, R_{\ell_1}$ 对于 $\ell_1 \in \mathbb{N}_{>1}$ 满足 $R_i \in (\mathbb{N} \cup \{\infty\})^{\ell_2}$ 对于 $\ell_2 \in \mathbb{N}, i \in [\ell_1]$ 且 f 为一个可计算函数 $f\colon [\ell_2] \times [\ell_2] \to 2^{[\ell_2]}$。对于 $x \in [\ell_2]$，令

$$A_x = \{M \in ([\ell_2])^{\ell_1} \mid x \in \widehat{f}(M[1], M[2], \ldots, M[\ell_2])\},$$

其中，对于所有 x_1、x_2、$\ldots \in [\ell_2]$

$$\widehat{f}(x_1, x_2) = f(x_1, x_2)$$

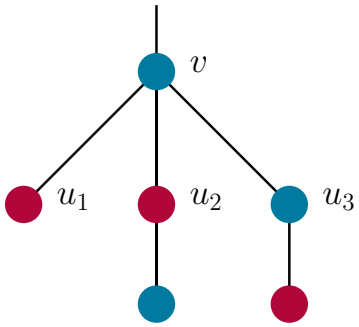

图9 Join 子程序的示例图。

并且对于所有 $2 \leqslant k \leqslant \ell_2$

$$\widehat{f}(x_1, x_2, \ldots, x_k) = \bigcup_{x \in \widehat{f}(x_1, x_2, \ldots, x_{k-1})} f(x, x_k).$$

然后，可以在时间复杂度为 $\mathrm{O}(\ell_1 \cdot \ell_2^2 \cdot T_f)$ 的时间内计算一个数组 $R \in (\mathbb{N} \cup \{\infty\})^{\ell_2}$，使得对于所有 $x \in [\ell_2]$，$R[x] = \min_{M \in A_x} \sum_{i=1}^{\ell_1} R_i[M[i]]$，其中 T_f 是计算 f 所需的时间。

由于之后还会复用该例程，因此其表述比本节所需的更为一般。在这里，对于 1 : 2 的情况，假设我们要合并某个顶点 v 的子节点 $u_1, u_2, \ldots, u_{\ell_1}$ 的分裂。例如，假设 v 有三个子节点，如图9所示。

然后，对于每个子节点 u_i，设有一个数组 R_i，其中 $R_i[x]$ 是对子树 T_{u_i} 进行划分所需的最小切割数，使其恰好比 r 连通分量多出 x 个 br 连通分量。在我们的例子中，假设 v 与其子节点之间的所有边都必须割断。我们看到，对于 $x \neq -1$，$R_1[-1] = 1$ 和 $R_1[x] = \infty$，因为 u_1 的子树唯一可能的划分只割断了通向 v 的边，并且比 br 连通分量多出一个 r 连通分量。此外，我们有 $R_2[1] = 1$（只割断 $\{v, u_2\}$）、$R_2[-1] = 2$（割断 u_2 的两条边），以及对于 $x \notin \{-1,1\}$</style>，有 $R_2[x] = \infty$。最后，注意到 $R_3 = R_2$。

函数 f 返回在合并两种可能性时应更新的索引集合。当一个子节点的子树的一种分裂包含 x_1 个更多的 br-连通分量，而另一个子节点的子树的一种分裂包含 x_2 个更多的 br-连通分量时，这两种分裂的组合包含的 $x_1 + x_2$ -连通分量比 br-连通分量多，而不是 r-连通分量。因此，唯一需要更新的索引是 $f(x_1, x_2) = \{x_1 + x_2\}$。之后，我们需要更新不止一个索引，因此定义 f 返回一个集合，而不是单个索引。注意，根据 f 和$\frown f$的定义，例程放入 $R[x]$ 中的每个值，都对应于从每个数组 R_i 中恰好选择一种分裂，使得 br-连通分量与 r-连通分量之间的差值总和恰好为 x。

在我们的示例中，假设为三个子树分别选择任意一种划分。令 x_i 表示以 u_i 为根、针对 $1 \leqslant i \leqslant 3$ 的子树所选划分中 br 分量与 r 分量的差。然后，Join 为 $x = x_1 + x_2 + x_3$ 设置 $R[x]$。如果有多种方式可以得到索引 x，则将割数最小的方式存储在 $R[x]$ 中。在此示例中，我们有 4 种可能性，即 $x_1 = -1$ 和 x_2、$x_3 \in \{-1, 1\}$。注意，$x_1 = -1$、$x_2 = -1$、$x_3 = 1$ 和 $x_1 = -1$、$x_2 = 1$、$x_3 = -1$ 的结果都为 $x = -1$。因此，只存储两种组合中的一种（即割数较少的一种；在此处，任取一种即可，因为两种变体都意味着 4 次割）。对于结果数组 R，我们有 $R[-3] = 5$、$R[-1] = 4$、$R[1] = 3$，以及针对 $x \notin \{-3, -1, 1\}$ 的 $R[x] = \infty$。请注意，R 中的割数对应于针对 x_i 的相应选择下各子树割数之和。

30 公平相关聚类

现在我们来说明如何计算 Join 子程序。

引理 21 的证明。 该算法以迭代方式运行。假设它已使用前 $i-1$ 个数组找到所有索引对应的最小值，并将它们存储在 R^{i-1}中。随后，它 将 第 i个数组与其中的每个索引 x_1 和 R^{i-1} 中的每个索引 x_2 进行 R_i连接。每次，对于所有索引 $x \in f(x_1, x_2)$，如果 $R^i[x]$ 小于该位置当前的元素，就将其设为 $R^{i-1}[x_1] + R_i[x_2]$。这样，它会尝试将中间解与 R_i 进行所有可能的组合，并针对每个索引追踪能够达到的最小值。形式上，我们在算法 1 中给出该算法。

算法 1 Join 子程序。

输入： $R_1, R_2, \ldots, R_{\ell_1}$，对于 $\ell_1 \geqslant 2$，满足 $R_i \in (\mathbb{N} \cup \{\infty\})^{\ell_2}$，对于 $0 \leqslant i < \ell_1$，以及一个可计算函数 $f\colon [\ell_2] \times [\ell_2] \to 2^{[\ell_2]}$。输出： $R \in (\mathbb{N} \cup \{\infty\})^{\ell_2}$，使得对于所有 $x \in [\ell_2]$, $R[x] =$， $\min_{M \in A_x} \sum_{i=1}^{\ell_1} R_i[M[i]]$ 满足 $A_x = \{M \in ([\ell_2])^{\ell_1} \mid x \in \widehat{f}(M[1], M[2], \ldots, M[\ell_2])\}$,$\widehat{f}(x_1, x_2, \ldots, x_k) = \bigcup_{x \in \widehat{f}(x_1, x_2, \ldots, x_{k-1})} f(x, x_k)$，且 $f(x_1, x_2) = f(x_1, x_2)$。1 $R \leftarrow R_1$2 对于 $i \leftarrow 2$ 到 ℓ_1 执行 3 $R' \leftarrow R$4 对每个 $(x_1, x_2) \in \in ([\ell_2])$ 执行 5 对每个 $x \in f(x_1, x_2)$ 执行 6 $R'[x] \leftarrow \min(R'[x], R[x_1] + R_i[x_2])$7 $R \leftarrow R'$

由于存在嵌套循环，该算法会在 $O(k \cdot \ell^2 \cdot T_f)$ 次迭代后终止。我们通过归纳法证明，每次迭代 i 后， R 都是针对数组 $R_1, \ldots, R_i$ 的 Join 解。第一次迭代只是尝试数组 R_1, R_2 的所有允许组合，并跟踪每个索引对应的最小值，这与我们对 Join 的定义相符。现在假设对于某个 i，该命题成立。注意，只有在存在相应的 $M \in A_x$ 时，我们才会更新值 $R[x]$，因此没有任何值会过小。为了证明没有值过大，任取一个 $x \in [\ell_2]$，令 a 为本次迭代中 $R[x]$ 能取得的实际最小值。令 $j_1, j_2, \ldots, j_{i+1}$ （其中 $\sim x \in f(j_1, j_2, \ldots, j_{i+1})$ ）为取得 a 的索引。于是，存在 $y \in [\ell_2]$，使得在 Join 前 i 个数组后，索引 y 处的值为 $a - R_{i+1}[j_{i+1}]$，且 $y \in f(j_1, j_2, \ldots, j_i)$。根据归纳假设，这意味着 $\widehat{} R[y] \leqslant a - R_{i+1}$。此外，由于 $x \in f(j_1, j_2, \ldots, j_{i+1})$ $\widehat{}$且 $y \in f(j_1, j_2, \ldots, j_i)$，因此我们有 $x \in f(y, j_{i+1})$。所以在本次迭代中，$R[x]$ 被设为不大于 $R[y] + R_{i+1}[j_{i+1}] \leqslant a$。由此，所有值都被正确设置。◀

注意，在与本节相关的 $f(x_1, x_2) = \{x_1 + x_2\}$ 情况下，第 4–6 行的循环计算数组 R 和 R_i 的 (min $+$)-卷积。像算法中那样尝试所有可能的组合，运行时间是二次的。除非推翻 MinConv 猜想，否则无法改进这一点。该猜想指出，对于任意常数 $\varepsilon > 0$ [21]，不存在一个算法能在 $O(n^{2-\varepsilon})$ 时间内计算两个长度为 n 的数组的 (min $+$)-卷积。

6.2.3 跟踪算法

借助 Join 子程序，我们能够构建一个动态规划算法，以比例为 1 : 2 的方式解决具有两种颜色的森林上的公平相关聚类。我们首先介绍如何将该算法应用于树，然后将其推广到森林。

在第一阶段，对于 br-连通分量与r-连通分量数量之间的每种可能差值，我们计算得到该差值的分裂所需的最小割数。在第二阶段，我们找出一种分裂，使第一阶段中被割断的边数与将该分裂转化为公平划分所需的边数之和最小。该和就是该划分的簇间代价，因此根据引理 3，这样便能找到相关聚类代价最小的公平划分。

划分这棵树。

在第一阶段，我们的目标是计算一个数组 D，使得对于所有整数 $-n \leqslant x \leqslant \frac{n}{3}$，$D[x] \subseteq E$，$x = br(T - D[x]) - r(T - D[x])$ 是一个最小规模的边集合，其中$br(T-D[x])$ 和 $r(T-D[x])$ 分别表示 $T-D[x]$ 中 br 连通分量和 r 连通分量的数量。为标记不存在此类集合的情况，我们要求 $D[x] = \mathbb{N}$ 中对应的条目为无穷大。我们通过动态规划填充该数组：对于每个顶点v，以及每种可能的头部 $h \in \{\emptyset, r, b, rr, br\}$，计算一个数组 D_v^h。这里，$D_v^h[x]$ 是一个最小规模的边集合，移除这些边后，根为 v 的子树 T_v 中恰好有 x 个 br连通分量多于 r连通分量。头部 h 表示最顶层分量中的颜色；这一点尤为重要，因为该分量之后可能还会包含来自 T_v 外部的顶点。头部 $h = r$表示包含红色顶点的分量，$h = br$ 表示包含一个蓝色顶点和一个红色顶点的分量，依此类推。如果 v 上方的边被割断，则该分量为空（$h = \emptyset$）。在计算 x时，头部不计作br连通分量或 r连通分量。图10给出了头部如何由子节点的分裂构成的示例。

接下来，我们只展示如何计算 $\Delta_v^h[x] = |D_v^h[x]|$，即获得相应划分所需删除的边集的大小。不过，集合 $D_v^h[x]$ 可通过简单的回溯方法在相同的渐近运行时间内得到。如果 $D_v^h[x] = \mathbb{N}$，则有 $\Delta_v^h[x] = \infty$。我们将所有值初始化为 $\Delta_v^h[x] = \infty$，这表示我们尚不知道删除哪些边后能得到该头部，以及 br类与 r类连通分量之间的差值。然后，对于每个红色叶节点 v，我们设 $\Delta_v^r[0] = 0$ 和 $\Delta_v^\emptyset[-1] = 1$。对于每个蓝色叶节点 v，我们设 $\Delta_v^b[0] = 0$ 和 $\Delta_v^\emptyset[0] = 1$。至此，叶节点的计算便告完成，因为唯一的可能性就是切断叶节点上方的边，或者不切断。现在，假设我们已完成某个顶点 v 的所有子节点 u_1、u_2、……、u_k 的计算。注意，v 至多有两个子节点被置于一个满足 v 的头部中。对于每个可在顶点 v 处形成的头部 $h \in \{\emptyset、\ r、\ b、\ rr、\ br\}$，我们都会尝试获得该头部的所有可能方式。

如果 $h \in \{r, b\}$ 和 $c(v)$ 对应于 h，则这是通过为所有子节点选择 $\emptyset$ 个头节点来完成的。不过，子树的划分并不唯一，因为对于每棵以某个子顶点 u_i 为根的子树，都有一个完整的 $D_{u_i}^\emptyset$ 可能划分数组，其br-连通分量和 r-连通分量的数量各不相同。为了为所有子顶点找到最佳选择，我们使用 Join 子程序；当以 $f(x_1, x_2) = \{x_1 + x_2\}$ 和一个数组列表调用该子程序时，它会返回一个数组R，使得对于所有索引 x $R[x]$，该值都是从每个输入数组中恰好选取一个值，并令所选值的索引之和为 i 时所得的最小值。因此，我们设置 $\Delta_v^h = \text{Join}(\Delta_{u_1}^\emptyset, \ldots, \Delta_{u_k}^\emptyset)$。在此处以及下文中，我们只会在至少有两个数组时调用 Join 子程序。如果只输入一个数组，我们就视同 Join 子程序返回该数组。需要注意的是，这里的索引范围从 $-n$ 到 $\frac{n}{3}$，而 Join 子程序假定索引为正数。因此，我们隐式假设存在一个

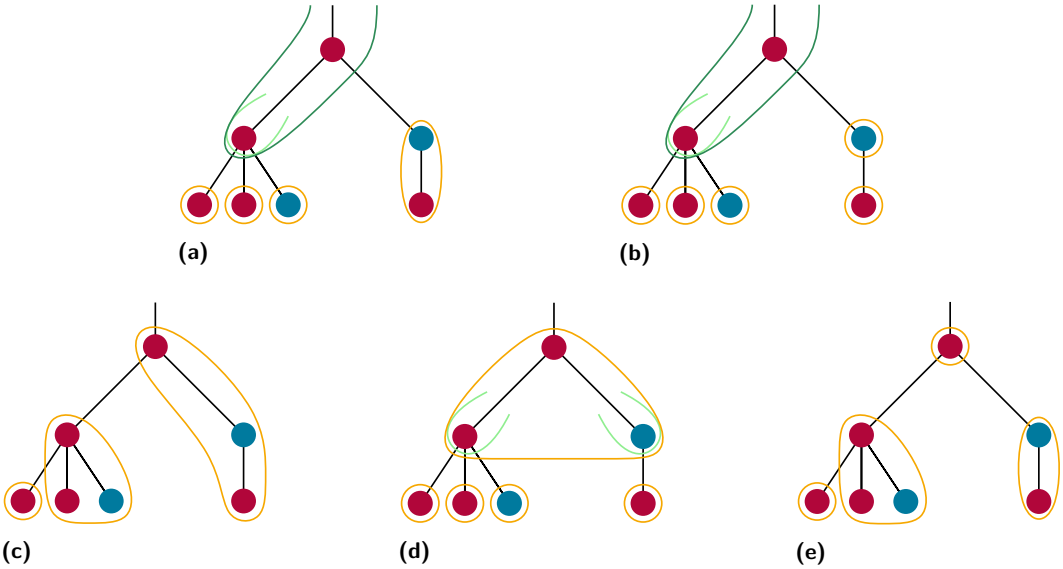

图10 示例子树，展示了获得头节点的多种可能方式。图10a和图10b展示了以 rr为头部（深绿色）的分裂。子节点（浅绿色）的头节点选择没有歧义，因为获得 rr头节点的唯一方式，是为左侧子节点选择 r头节点，为右侧子节点选择$\emptyset$头节点。左侧和右侧两种变体都必须考虑，因为它们的红色连通分量数减去非红色连通分量数所得的差值不同。图10c–10e中的分裂会产生 $\emptyset$头节点，因为它们割断了子树根节点上方的边，因此子树中没有顶点能与子树外的顶点属于同一个连通分量。不过，在这3种分裂中，后续只会考虑图10c和图10d，因为图10e得到的红色连通分量数与非红色连通分量数之差和图10c相同，却多割了一条边。我们注意到，还有其他分裂也会得到$\emptyset$头节点，但这里没有列出。

此处的索引 x 映射到子程序中的索引 $x+n+1$。

如果 $h=br$ 或 $h=rr$ 和 $c(v)$ 都对应于 r，那么所有子节点的头节点都应为 $\emptyset$，但有一个子节点除外，我们将它放入与 v相同的连通分量中。根据 h 和 $c(v)$，它的头部为 $h' \in \{r, b\}$。如果 $h=rr$ 和 $c(v)$ 对应于 R，或者 $h=rb$ 和 $c(v)$ 对应于 b，则有 $h'=r$。否则，$h'=b$。对于所有 $i \in [k]$，我们计算数组 $\Delta'_{u_i} = \mathsf{Join}(\Delta^{\emptyset}_{u_1}, \dots, \Delta^{\emptyset}_{u_{i-1}}, \Delta^{h'}_{u_i}, \Delta^{\emptyset}_{u_{i+1}}, \dots, \Delta^{\emptyset}_{u_k})$，其中指的是 u_i具有非空头部。最后，对于所有 $-n \leqslant x \leqslant \frac{n}{3}$，我们将 $\Delta^h_v[x]$ 设为 $\min_{i \in [k]} \Delta'_{u_i}[x]$。

如果 $h=\emptyset$，那么我们必须尝试 v 所在连通分量的所有不同可能性，并在每种情况下切断 v 上方的边。首先假设我们要将 v 放入一个 brr-连通分量中。那么它必须与两个顶点合并：要么在一个子节点处取一个头部 $h' \in \{br, rr\}$，要么在两个子节点处分别取头节点 $h_1, h_2 \in \{r, b\}$。h', h_1, h_2 的具体选择当然取决于 $c(v)$。对于第一种情况，我们计算数组 $\Delta_{h'} = \mathsf{Join}(\Delta^{\emptyset}_{u_1}, \dots, \Delta^{\emptyset}_{u_{i-1}}, \Delta^{h'}_{u_i}, \Delta^{\emptyset}_{u_{i+1}}, \dots, \Delta^{\emptyset}_{u_k})$。对于第二种情况，我们计算数组 $\Delta_{i,j} = \mathsf{Join}(\Delta^{\emptyset}_{u_1}, \dots, \Delta^{\emptyset}_{u_{i-1}}, \Delta^{h_1}_{u_i}, \Delta^{\emptyset}_{u_{i+1}}, \dots, \Delta^{\emptyset}_{u_{j-1}}, \Delta^{h_2}_{u_j}, \Delta^{\emptyset}_{u_{j+1}}, \dots, \Delta^{\emptyset}_{u_k})$，其中遍历所有满足 $i<j$ 且 $\{v, u_i, u_j\}$ 是一个 brr-连通分量的子节点对 u_i, u_j。现在，我们已将所有通过 v 构成 brr-连通分量的方式所对应的最小切割数，以及数组 $\Delta_{h'}$ 和 Δ_i 中 x 的所有可能性都存储下来；对于 i, j 的所有可能性，情况也一样。不过，v 也可能属于一个 r-、b-、rr-或 br-连通分量。因此，计算 $\Delta^{\emptyset}_v[x]$ 时，我们不仅要在数组 $\Delta_{h'}$ 和 $\Delta_{i,j}$ 中取位置 x 的最小值，还要考虑数组 $\Delta^r_v, \Delta^{br}_v, \Delta^{rr}_v$ 和 Δ^{br}_v。注意，这里必须将 Δ^r_v 中的所有值向左移一位，因为隔离 v 会产生另一个 r-连通分量。因此，写入 $\Delta^r_v[x]$ 的项实际上应放在 $\Delta^r_v[x-1]$ 中。同样，我们必须将 Δ^{br}_v 向右移位，因为这样会在子树顶部产生一个新的 br- 连通分量。最后，只要 v 不是 T 的根，就必须将 $\Delta^{\emptyset}_v$ 中的所有值加一，以反映我们必须在 v 上方额外进行的一次切割。

所有计算完成后，根据 Join 子程序的正确性和归纳论证，所有顶点 v 和头节点 h 对应的 Δ^h_v 均已正确计算。注意，在 Join 子程序中，由于 $f(x_1, x_2)$ 会返回合并两棵子树时的正确索引，$\widehat{f}(x_1, x_2, \dots, x_k)$ 会给出合并 k 棵子树时的正确索引。特别地，$\Delta^{\emptyset}_r$ 是一个数组，其中对每个 $-n \leqslant x \leqslant \frac{n}{3}$，都记录了需要切断的最少边数，使得 br-连通分量的数量恰好比 r-连通分量多 x 个；其中 r 是 T 的根。通过调整 Join 子程序，使其跟踪每个位置上得到最小值的确切组合，我们还可以得到一个数组 D，其中不仅包含边数，还包含需要切断的边集；如果不存在这样的边集，则标记为 $\mathbb{N}$。

在每个节点处，计算这些数组需要 $O(n^5)$ 时间；相比之下，根据引理 21，计算 $O(n^2)$ 个数组 $D_{u,\ w}$ 各自需要 $O(n^3)$ 时间，因为 $\ell_1, \ell_2 \in O(n)$。因此，此阶段的时间复杂度为 $O(n^6)$。

构造公平聚类。

令 D 为第一阶段计算得到的集合。注意，每个边集合 $D[x]$ 都直接给出一种划分，即由 $T - D[x]$ 中的连通分量诱导出的划分。

根据引理20，将 $D[x]$ 给出的划分转化为由 1 个蓝色顶点和 2 个红色顶点组成的集合的聚类，代价最低的方法可以在线性时间内找到，并且会多产生 $\max(0, x)$ 2 个割。因此，我们找到使 $|D[x]| + \max(0, \frac{x}{2})$ 最小的 $-n \leqslant x \leqslant \frac{n}{3}$—我们返回相应的聚类，因为它具有最小的簇间代价。

如果我们在第一阶段跟踪了各类型连通分量的数量，那么每次划分这一阶段只需常数时间，因此整体运行时间由第一阶段主导。

森林。

稍微调整第一阶段后，我们的算法很容易推广到求解两种颜色比例为 1 : 2 的非连通森林上的公平相关聚类。我们对每棵树分别运行上述动态规划。总运行时间仍为 O(n^6)。对于森林中的每棵树 T_i 以及每个 $h \in \{\emptyset,\ r,\ b,\ rr,\ br\}$，令 $\Delta^{\emptyset}_{T_i}$ 表示数组 $\Delta^{\emptyset}_{r}$，其中 r 是树 T_i 的根。为了得到整个森林的划分，而不只是各棵树各自的划分，我们使用这些数组 Δ_{T_i} 和函数 $f(x_1,\ x_2) = \{x_1 + x_2\}$，额外运行一次 Join 子程序。由此得到数组 R，使得对于整片森林而非各棵树，$R[x]$ 是获得一种划分所需的最小切割数，该划分中 x 类连通分量恰好比 r 类连通分量多 br 个。请注意，我们在每棵树中选择 $\emptyset$ 头部，因为这些树彼此不连通；因此，为了找到一种划分，我们暂时无需考虑如何合并不同树上的连通分量，这将在第二阶段完成。随后，第一阶段通过回溯方法输出数组 D，其中包含与 R 对应的边集合。由于额外的子程序调用耗时为 O(n^3)，算法的渐近运行时间不会改变。由此得到以下结果。

▶**定理 22.** 公平相关聚类在两种颜色的比例为1:2 的森林上可以在$\mathrm{O}(n^6)$时间内求解。

6.3 小簇

为了获得一种能够处理更多颜色和不同颜色比例的算法，我们将上一节中针对 1 : 2 颜色比例的做法加以推广。由此得到以下结果。

▶ **定理 23.** 设 F 是一个有 n 个顶点的森林，每个顶点都被染成 $k \geqslant 2$ 种颜色之一。设这些颜色的分布比例为 $c_1 : c_2 : \ldots : c_k$，其中 $c_i \in \mathbb{N}_{>0}$ 对所有 $i \in [k]$ 均成立，且最大公约数$(c_1, c_2, \ldots, c_k) = 1$。那么，在 F上求解公平相关聚类可在$\mathrm{O}(n^{2\,\mathrm{setvars}+\mathrm{setmax}+2} \cdot \mathrm{setvars}^{\mathrm{setmax}})$时间内完成，其中 $\mathrm{setvars} = \prod_{i=1}^{k}(c_i + 1)$，且$\mathrm{setmax} = \sum_{i=1}^{k} c_i$。

算法再次分为两个阶段运行。第一阶段，它会创建一个可能的分裂列表，即这样的划分：对于每种颜色，每个连通分量中该颜色顶点的数量都不超过最小规模公平分量中该颜色顶点的数量。第二阶段，它会检查这些分裂能否合并成一个公平聚类，并从中返回代价最小的那个。我们首先给出在树上解决该问题的算法，然后将其推广到也能处理森林的情形。

森林拆分。

对于 1:2 方法的第一阶段，我们采用了一种动态规划，针对第二阶段重新组装所产生的每种可能代价，记录获得一个分裂所需的最小数量。然而，如果给定一个具有 $k \geqslant 2$ 种颜色、比例为 $c_1 : c_2 : \ldots : c_k$ 的图，那么第二阶段所需的割数并不总能像 1:2 情形中 r-连通分量和 br-连通分量这两种分量类型的数量差那样容易界定。不过，我们发现，只需跟踪获得一个分裂的任意可能着色所需的最小割数即可。

我们首先界定一种划分的着色数量上限。由于在动态规划过程中，我们大多数时候考虑的是 G 的子图的划分，因此还必须计算顶点数少于 n 的划分的所有可能着色。

▶ **引理 24.** 设 U 是一个含有 n 个元素的集合，这些元素被赋予 $k \in \mathbb{N}_{>1}$ 种颜色，并设 $d_1, d_2, \ldots, d_k \in \mathbb{N}$。令 S 为 U 的子集的所有可能划分构成的集合，使得对于每种颜色 i，每个簇中该颜色的顶点至多有 d_i 个。令 C 为 S 中各划分的所有着色构成的集合。那么，$|C| \leqslant (n+1)^{\text{setvars}-1}$，其中 集合变量 $= \prod_{i=1}^{k}(d_i+1)$。

证明。 具有不同着色的集合数量至多为 setvars，因为每个连通分量中颜色为 d_i的顶点数量为 i。因此，仅使用这些集合对划分 $\mathcal{P}$ 进行着色，可由一个大小为 setvars 且取值于 $[n] \cup \{0\}$ 的数组表示，因为每个连通分量出现的次数不超过 n。填充此数组的方法有 $(n+1)^{\text{setvars}}$ 种。然而，由于这些集合着色合起来必须构成一个划分，最后一项由前面 setvars -1 项确定，因此只有 $(n+1)^{\text{setvars}-1}$ 种可能。◀

据此，我们采用一个类似于第 6.2 小节所述的动态规划，但跟踪分裂的所有着色对应的最小割代价。具体如下述引理所示。

▶ **引理 25。** 设 $F = (V, E)$ 为一个森林，其顶点具有 k 种颜色。此外，设 $d_1, d_2, \ldots, d_k \in \mathbb{N}$ 和 S 为所有可能的 V 的集合，这些划分满足：对于 $i \in [k]$，每个簇中颜色为i 的顶点至多有 d_i 个。令 C 为 S 中各划分的所有着色的集合。则在时间复杂度为 $\mathrm{O}(n^{2\,\text{setvars}+\text{setmax}+2} \cdot \text{setvars}^{\text{setmax}})$ 、满足 setvars $= \prod_{i=1}^{k}(d_i+1)$ 且 setmax $= \sum_{i=1}^{k} d_i$ 的条件下，对于所有 $C \in C$，我们都能找到一个最小规模集合 $D_C \subseteq E$，使得$F - D_C$ 中的连通分量构成顶点的划分，并具有着色 C；或者证明不存在这样的集合。

证明。 我们首先描述如何在一棵树上解决该问题 T，然后将该方法推广到森林。我们将满足以下条件的顶点划分称为 i：对于每种颜色 d_i，每个簇中该颜色的顶点数至多为 分裂。

我们采用动态规划，计算 T 中每个顶点为根的所有子树以及所有可能的分裂的着色所对应的集合 D_C。具体而言，我们从叶节点开始计算所有可能的分裂，并逐步向根节点扩展。此时，与当前子树根节点的父节点相连的连通分量尤为重要，因为它是唯一可以由子树外部的顶点扩展的连通分量。我们将该连通分量称为头部。注意，如果根节点与其父节点之间的边被切断，则头部为空。我们不将头部计入分裂的着色中，而是单独明确给出它。形式化地，对于每个 $v \in V$、每个分裂 C 的可能着色，以及头部的每种可能着色 h，我们计算 $D_v^h[C] \subseteq E$：使得 $T_v - D_v^h[C]$ 的连通分量构成具有着色 C 和头部 h 的分裂的最小规模边集合。如果不存在这样的集合，则令 $D_v^h[C] = \mathbb{N}$ 为一个无穷大的集合。

令所有 $D_v^h[C]$ 初始化为 $\mathbb{N}$。然后，对于每个父节点为 w 的叶节点 v，我们设定 $D_v^{h_{c(v)}}[C_\emptyset] = \emptyset$，其中 $h_{c(v)}$ 是连通分量 $\{v\}$ 的着色，而 $C_\emptyset$ 是空集上的划分着色。此外，我们设定 $D_v^{h_\emptyset}[C_{c(v)}] = \{\{v, w\}\}$，其中顶点 v 未被置于头部中，因为通向其父节点的边被切断。由于对于叶节点而言，切断其上方的边或不切断是仅有的两种选择，因此数组的这一部分现已完成。

接下来，假设我们已经完成了某个顶点 v 的所有子节点的计算。对于在顶点 v 处可形成的头部的每一种可能着色 h，我们都尝试所有可能性以获得该着色。

为此，首先假设 h 非空。因此，v 必须放在头部。令 $h_{-c(v)}$ 表示在颜色 $c(v)$处将 h 减一后得到的着色。为了得到头部 h，我们因此必须为以子节点 u_1、u_2、……、u_ℓ 为根的子树选择分裂的着色，使它们各自的头部 h_{u_1}、h_{u_2}、……、h_{u_ℓ} 组合成 $h_{-c(v)}$。A

36 公平相关聚类

着色的组合 C_1、C_2、.....、C_ℓ 指的是具有相应着色的划分 M_1、M_2、.....、M_ℓ 之并集的着色，其定义为数组 C_1、C_2、.....、C_ℓ 的逐元素求和。通常，有多种为子顶点选择头部的方法可以满足这一要求。由于每个头部的大小至多为 setmax，$h_{-c(v)}$ 且包含 v，因此它由少于 setmax 个非空头部组成。可能的头部至多有 setvars 个，而我们需要选择少于 setmax 个子节点，所以，利用这些子节点构造 $h_{-c(v)}$ 的方式至多有 $\binom{n}{\text{setmax}-1} \cdot \text{setvars}^{\text{setmax}-1} < n^{\text{setmax}-1} \cdot \text{setvars}^{\text{setmax}-1}$ 种。将每种方式描述为一个函数 H，该函数为 v 的每个子节点指定一个（可能为空的）头部。即使固定了 H，仍然可能有多种分裂方式。这是因为，即使某个子节点 u 的头部 $H(u)$ 已确定，该子节点 u 的子树仍可能存在多种不同着色的分裂，且这些分裂得到相同的头部。因此，对于每个可能的 H，我们都对所有子节点 u 使用数组 $D_u^{H(u)}$ 调用 Join 子程序，并将集合的基数作为该子程序的输入。为便于阅读，这里用某个向量 C 而非算法描述中 Join 子程序所用的单个数值索引来标记数组。我们默认每种可能的着色都由一个正整数表示。令这些索引以结构化的方式枚举这些向量后，在两种格式之间转换只会额外增加一个 O(n) 的时间因子。

对于 $f(x_1, x_2)$，我们给出一个函数，该函数返回一个仅包含通过组合由 x_1 和 x_2 索引的着色所得到的着色索引的集合，其可在 O(n) 时间内计算。组合着色是指，对于每个集合着色，将两个$\widehat{}$划分着色中的出现次数相加。因此，Join 子程序中定义的 $f(x_1, x_2、\ldots\ldots、x_k)$ 返回由 x_1、x_2、.....、x_k 索引的着色的组合索引。注意，数组至多有 n 个，且每个数组的长度小于 $(n+1)^{\text{setvars}-1}$，因为根据引理 24，不同着色的数量就是这么多。执行 Join 子程序后，根据引理 21，我们得到一个数组 D_H，其中包含通过按照 H 进行划分所能得到的所有可能着色所需的最小割代价。通过略微修改 Join 子程序以采用简单的回溯方法，我们还得到实现该割代价的集合 $D \subseteq E$。最后，对于可能的赋值 H，逐元素地从所有计算得到的数组 D_H 中取最小规模集合，从而完成对 D_v^h 的计算。

如果 h 是空头，也就是说，v 上方的边被割掉，那么 v 会被放入一个连通分量中，该分量要么大小为 setmax，要么其着色对应于某个头部 h'。在第一种情况下，我们按照上文所述的相同方式，尝试所有合适的赋值 H 并使用 Join 子程序，计算数组 D_{full}。在第二种情况下，我们只需取已经填好的数组 $D_v^{h'}$。注意，这两种情况下，我们都必须将数组中的所有值加一，以反映 v上方的边被割掉；但如果 v 是根顶点，则无需如此。此外，我们还必须调整各数组中的值，以反映包含 v 的连通分量不再是头部；但割掉 v 上方的边后，该分量也应计入划分的着色。因此，条目 $D_{\text{full}}[C]$ 实际上存储在 $D_{\text{full}}[C_{-\text{full}}]$ 处，其中 $C_{-\text{full}}$ 是着色 C 减去一个最小大小公平簇的着色。如果不存在这样的条目 $D_{\text{full}}[C_{-\text{full}}]$，我们就假设其值为 ∞。访问数组 $D_v^{h'}$ 时也同理，我们必须从索引中减去着色 h'。对所有 h' 的可能取值，逐元素取这些调整后数组 D_{full} 和 $D_v^{h'}$ 中的最小值，即可得到 $D_v^{\emptyset}$。

根据 Join 子程序的正确性，并且由于我们尝试了在每个顶点构建指定头节点和着色的所有可能性，因此在完成根 r T 处的计算后，我们知道数组 $D_r^{\emptyset}$ 中包含树的一种划分的每种可能着色所需的最小割代价。

对于每个 n 顶点和 setvars 个可能的头节点，我们调用 Join 子程序

至多 $n^{\text{setmax}-1} \cdot \text{setvars}^{\text{setmax}-1}$ 次。每次调用时，最多传入 n 个数组；根据引理 24，可能的着色有 $O(n^{\text{setmax}})$ 种，因此所有这些数组都包含相同数量的元素。因此，每次子程序调用的耗时为 $O(n \cdot (n^{\text{setvars}})^2) = O(n^{2\,\text{setvars}+1})$，所以该算法的耗时为 $O(n^{2\,\text{setvars}+\text{setmax}+2} \cdot \text{setvars}^{\text{setmax}})$，其中还包括 $O(n)$ 的额外因子，用于转换 Join 子程序所需的索引。

当输入图不是树而是森林 F 时，我们对森林中的每棵树应用动态规划。然后，我们还会在森林中所有树的根处，对 $\emptyset$-头部对应的数组运行 Join 子程序。结果数组包含所有可能组合的最小成本解，这些组合来自各棵树的分裂着色，并将该数组作为输出返回。这个额外的子程序不会改变渐近运行时间。◀

根据引理4和引理10，只需考虑将划分作为可能的解，并要求每个簇中颜色为 i 的顶点至多有 c_i 个，对所有 $i \in [k]$ 均如此。因此，对于所有 $i \in [k]$，我们将引理 25 应用于森林 F 和集合 $d_i = c_i$。这样，对于每一种可能的分裂着色，我们都能找到获得具有该着色的分裂所需的最小边集。

构造公平聚类。

令 D 为第一阶段生成的数组，即对于分裂的每种着色 C，$D[C]$ 都是一个规模最小的边集合，使得 $F - D[C]$ 中的连通分量所诱导的划分具有着色 C。在第二阶段，我们必须找到使相关聚类代价最小的分裂。为此，我们逐一判断每个分裂是否具有 可组装性，也就是说，能否合并其簇，使其成为一个公平解，且所有簇的大小都不超过 setmax。在这些分裂中，我们返回第一阶段计算出的簇间代价最小的那个。

以下原因足以说明这一点。首先，注意，判断可组装性只取决于分裂的着色，因此第一阶段只跟踪所有可能的分裂着色，而非所有可能的分裂本身，并无妨碍。其次，在此阶段我们无需考虑进一步的边切割：假设有一个着色为 C_S 的分裂 S，若进一步切割 S 中的 a 条边，得到另一个着色为 $C_{S'}$ 的分裂 S'，就能获得更低的代价。然而，由于我们正确地填充了数组 D，因此存在一个条目 $D[C_{S'}]$ 和 $|D[C_{S'}]| \leqslant |D[C_S]| + a$。既然无论如何都会在求最小值时考虑这个值，就不必再考虑进一步切割这些分裂。第三，根据引理 3，最小簇间代价会给出最小相关聚类代价。合并簇时，第一阶段计算出的簇间代价可能降低，但不会升高。如果代价降低，我们就会高估代价。然而，这种情况意味着两个簇之间存在一条边；而且合并后它们的大小仍不超过 setmax，因此第一阶段也会找到考虑这种情况的另一个分裂。

我们采用动态规划，判定所有可能的 $O(n^{\text{setvars}})$ 分裂着色是否具有可组装性。将划分着色的大小 定义为该划分着色中集合着色的数量（不一定是不同集合着色的数量）。我们从小到大判定所有可能的分裂着色是否具有可组装性。注意，每种这样的着色大小至少为 $^{n}\text{setmax}$。若其大小恰为 $^{n}\text{setmax}$，则其中包含的所有集合着色大小都为 setmax，因此，当且仅当所有集合着色都是公平的，该划分着色才具有可组装性。现在假设我们已找出所有大小恰为 $j \geqslant {}^{n}\text{setmax}$ 的可组装分裂着色。假设一个大小为 $j+1$ 的划分着色 C 具有可组装性。那么，至少有两个来自 C 的集合着色 C_1, C_2 合并在一起。因此，令 C' 为该划分

着色：从 C 中移除集合着色 C_1, C_2，并加入由 C_1 和 C_2 的合并着色得到的集合着色。现在，C' 的大小为 j，且具有可组装性。因此，每个具有 $j+1$ 个连通分量的可组装分裂，都存在一个具有 j 个连通分量的可组装分裂。反过来，若将大小为j 的可组装划分着色中的一个集合着色拆分，就会得到一个大小为 $j+1$ 的可组装划分着色。因此，对于每个大小为 j 的可组装划分着色（数量少于n^{setvars}），尝试拆分其中一个集合着色的每种可能方式（少于 $i \cdot 2^{\text{setmax}}$ 种，因为有 j 个集合着色，且每个集合着色的大小至多为 setmax），即可找出所有大小为 $j+1$ 的可组装分裂着色。因此，计算所有大小为 $j+1$ 的可组装分裂着色需要 O($n^{\text{setvars}} \cdot j \cdot 2^{\text{setmax}}$) 的时间，这意味着第二阶段的 $n -$ $^{n}{}_{\text{setmax}}$ 次迭代总共需要 O($n^{\text{setvars}+2} \cdot 2^{\text{setmax}}$) 的时间。这一时间复杂度由第一阶段的运行时间主导。因此，完整算法的运行时间为 O($n^{2\text{setvars}+\text{setmax}+2} \cdot \text{setvars}^{\text{setmax}}$)，由此得到定理 23。

由此可得一种算法，可在任意森林上求解公平相关聚类问题。不过，其运行时间可能随森林中的颜色比例而变化，并且关于顶点数呈指数级。

6.4 簇数较少

如果图中的颜色分布满足最小成本解中的每个簇大小均为常数，那么上一节介绍的算法可在多项式时间内运行。当簇很大但数量较少时，算法的运行时间最差。针对这种情况，我们提出了另一种算法：若簇数为常数，该算法可在多项式时间内运行。不过，它仅适用于森林中顶点按 $1:c$ 的比例分为两种颜色，且满足某个 $c \in \mathbb{N}$ 的实例。

该算法使用一个子程序，用于计算从树上切割出特定大小的簇所需的最小切割数。引理 26 给出了这个子程序。

▶ **引理 26.** 设 $T=(V,E)$ 是一棵以 $r \in V$ 为根的树，且 $k \in \mathbb{N}$。那么，我们可以计算一个数组 R，使得对于每个 $a_0 \in [n]$ 和 $a = a_1, a_2, \ldots, a_k \in ([n-1] \cup \{0\})^k$，其中 $a_i \geqslant a_{i+1}$ 对于 $i \in [k-1]$ 和 $\sum_{i=0}^{k} a_i = n$ 成立，我们有 $R[a_0, a]$ 是 V 的划分 $P = \{S_0, S_1, \ldots, S_k\}$，且该划分满足 $r \in S_0$，并且对于 $i \in [k]$，$|S_i| = a_i$。计算时间为 $\mathrm{O}((k+3)! \cdot n^{2k+3})$。

证明。 我们给出一种构造，使得 $R[a_0, a]$ 存储的不是划分本身，而是产生的簇间代价。通过一种简单的回溯方法，也可以得到这些划分。

我们采用一种涉及使用 Join 子程序的动态规划。为便于阅读，这里用向量 $a \in [n]^k$ 和 $a_0 \in [n]$ 为数组编索引，而不是像 Join 子程序的算法描述中那样使用单个数值索引。我们默认每个可能的 a_0，a 都由 $[n^{k+1}]$中的某个索引表示。通过让这些索引以结构化的方式枚举这些向量，在两种格式之间进行转换只需额外的 O(k) 时间因子。

从叶节点开始，并沿着所有子节点都已完成计算的顶点继续进行，我们为每个顶点 $v \in V$计算数组 R_v，该数组对其子树 T_v 具有针对 R 所描述的性质。具体而言，对于每个顶点 v，我们执行以下操作。令R_v^0为一个数组，除 $R_v^0[1,$之外的所有索引处都具有 ∞-值，因为这是树 $T[\{v\}]$唯一可能的条目。

如果 v没有子节点，则 $R = R_v^0$。否则，设 v的子节点为 u_1，u_2、……、u_ℓ。然后，我们使用数组 R_v^0、R_{u_1}、R_{u_2}、……、R_{u_ℓ}调用 Join 子程序。我们必须定义 f，使其给出组合子节点的子树划分和 v的所有可能性。对于所有

a_0、a 和 a'_0、a' 的可能取值，回顾 $f((a_0,a),(a'_0,a'))$ 应返回形如 (a''_0,a'') 的索引集合。每个这样的索引描述了 v 与已经考虑的子节点 (a_0,a) 的所有可能性组合，以及下一个子节点 (a'_0,a') 的所有可能性。首先，我们考虑切断 v 与由 (a'_0,a'') 表示的子节点 u 之间的边的情况。然后，我们添加将这两个集合与各自包含 $k+1$ 个簇的所有可能合并方式。由于我们切断了边 $\{u,v\}$，因此有 k 种放置包含 u 的簇（除包含 v 的簇之外）的可能方式，随后还有 $k!$ 种分配其余簇的方式。所有这些方式都被放入集合 $f((a_0,a),(a'_0,a'))$ 中。其次，我们假设边 $\{u,v\}$ 未被切断。那么，包含 v 和 u 的簇必须合并，因此只有 $k!$ 种分配其他簇的方式。特别地，对于以这种方式放入 $f((a_0,a),(a'_0,a'))$ 的所有索引 (a''_0,a'')，我们有 $a''_0 = a_0 + a'_0$。注意，f 可在 $O(k\cdot k!)$ 时间内计算。注意，Join 子程序中定义的 $\sim f(x_1,x_2,\dots,x_\ell)$ 列出了切断组合树的所有可能性，因为它会迭代地组合第一个子节点和顶点 v 的所有可能性，并为所得的树列出与下一个子节点的所有可能组合，依此类推。Join 子程序的运行时间为 $O((k+1)\cdot \left(n^{k+1}\right)^2 \cdot (k\cdot k!)\cdot k)$，即 $O((k+3)!\cdot n^{2k+2})$。因此，子程序的全部 $O(n)$ 次调用所需时间为 $O((k+3)!\cdot n^{2k+3})$。◀

由此，我们能够为双色图给出一个比例为 $1:c$ 的算法；当簇的数量为常数时，也就是说，当 $c\in\Theta(n)$ 时，该算法可在多项式时间内运行。

▶ **定理 27.** 设 F 是一个具有两种颜色、比例为 $1:c$（其中 $c\in\mathbb{N}_{>0}$）的 n-顶点森林，且设 $p=\frac{n}{c+1}$。那么，在 F 上的公平相关聚类可在 $O(n^{p^3+p^2+p})$ 时间内求解。

证明。注意，若每个蓝色顶点对应 c 个红色顶点，则 $p=\frac{n}{c+1}$ 表示蓝色顶点的数量。根据引理 4，任何最小代价聚类都由 p 个簇组成，每个簇恰好包含一个蓝色顶点；根据引理 3，我们知道，只需使这类聚类所割的边数最少即可。所有蓝色顶点都必须被放入不同的簇中。通过割至多 $p-1$ 条边即可将它们分开，因此我们尝试所有大小至多为 $p-1$ 的边子集，共有 $O((p-1)\cdot\binom{n-1}{p-1})$ 个。割去这些边后，我们得到 ℓ 棵树 T_1、T_2、……、T_ℓ，其中有 p 棵恰好包含一个蓝色顶点，其余树则不包含蓝色顶点。若树中有蓝色顶点，就以该顶点为根；否则，就以任意顶点为根。对于每棵树 T_i，令 r_i 表示其中红色顶点的数量。若恰好有 p 棵树，且对所有 $i\in[p]$ 均有 $r_i=c$，则我们就找到了一个最小代价聚类，其中第 i 个簇就是对所有 $i\in[p]$ 而言，树 T_i 的顶点集合。否则，我们必须从树中割下部分，并将其分配给其他簇，以使划分满足公平性。为此，对于每棵树 T_i，我们计算一个数组 R_i，记录割下至多 $p-1$ 个指定大小部分的代价。更具体地说，$R_i[(a_1,a_2,\dots,a_{p-1})]$ 表示分别割下 a_1、a_2、……、a_{p-1} 大小的 $p-1$ 个簇所需的割边数；若不存在 $\sum_{i=1}^{p-1} > r_i$ 所述的割法，则取值为 ∞。只需计算 $R_i[(a_1,a_2,\dots,a_{p-1})]$，其中对 $i\in[p-2]$ 有 $0\leqslant a_i\leqslant a_{i+1}\leqslant n$。

我们利用引理 26 计算这些数组。请注意，此处省略了引理中使用的 a_0，因为在这里它指的是未从树中截断的顶点数。不过，a_0 在 a 上仍有明确的定义，因为所有值的总和等于该树中的顶点数。此外，将所有不含蓝色顶点的树连接到某个新添加的辅助顶点 z，并将以 z 为根的这棵树作为引理 26 的输入，我们将子程序调用次数减少到 $p+1$。随后，对于全红树得到的数组，我们只考虑其中 $a_0=1$ 的条目，因为我们不希望将 z 合并到一个簇中，但该树中除 z 之外的每个顶点都必须合并到另一个簇中。我们将从这棵树得到的数组称为 R_0，将从其他树得到的数组分别称为 R_1、R_2、……、R_p。

请注意，每个公平聚类都可以通过从每个数组R_i中选择一个条目，并将截断部分分配给其他簇来刻画。由于每个数组包含少于 $\binom{n^p}{p}$ 个条目，并且将截断部分分配给各个簇最多有 $(p!)^p$ 种方式，因此总共至多有n^{p^2} 种可能性。对于其中每一种可能性，我们在线性时间内计算其是否形成公平聚类。在这些公平聚类中，我们返回簇间代价最小的一个，该代价通过对从数组 R_i中选出的条目求和来计算。根据引理 3，该聚类具有最小的相关聚类代价。我们得到的总运行时间为

$$\mathrm{O}((p-1)\cdot\binom{n-1}{p-1}\cdot\left((p+1)\cdot\left(n^{p+3}+n^{p^2+p-2}\right)+n^{p^2+1}\right))\subseteq\mathrm{O}(n^{p^3+p^2+p}).$$ ◀

结合定理 23 和 27 的结果可知，对于两种颜色比例为 1 : c 的森林（其中某个 $c\in\mathbb{N}_{>0}$），当簇的大小为常数或大小属于 $\Theta(n)$ 时，均存在多项式时间算法。定理 11 指出，森林上的公平相关聚类是 NP 难问题，因此我们知道，这一困难性存在于这两个极端之间的某处。

7 宽松公平性

公平相关聚类的困难性结果看起来或许是由于公平性的定义过于严格，因为该定义要求森林上的簇具有特定大小。然而，本节将证明，即使放宽公平性要求，我们的结果本质上仍然成立。

7.1 定义

我们采用 Bera 等人提出、并由 Ahmadi 等人用于公平相关聚类的松弛公平性约束 [11] [1]。对于以下定义，给定一个集合U，其颜色由函数 $c\colon U\to k$指定，我们用 $U_i=\{u\in U\mid c(u)=i\}$ 表示颜色为 i 的顶点集合，其中对所有 $i\in[k]$均成立。

▶ **定义 28**（宽松公平集）。设 U 是一个有限元素集，其颜色由函数 $c\colon U\to[k]$ 指定，其中 $k\in\mathbb{N}_{>0}$，并设 p_i、$q_i\in\mathbb{Q}$，且对所有 $i\in[k]$ 都有 $0<p_i\leqslant\frac{|U_i|}{|U|}\leqslant q_i<1$。则，某个 $S\subseteq U$ 关于 q_i 和 p_i 是宽松公平的，当且仅当对于所有颜色 $i\in[k]$，都有 $p_i\leqslant\frac{|S\cap U_i|}{|S|}\leqslant q_i$。

注意，我们要求 p_i 和 q_i 满足：精确公平解也是宽松公平的。此外，我们排除将 p_i 或 q_i 设为 0 的情况，因为这将允许不包含每种颜色的簇，而我们不认为这样的簇是公平的。

▶**定义 29**（宽松公平划分）。设 U 是一个有限元素集，其颜色由函数 $c\colon U\to[k]$ 指定，其中 $k\in\mathbb{N}_{>0}$，并设 p_i、$q_i\in\mathbb{Q}$，且对所有 $i\in[k]$ 都有 $0<p_i\leqslant\frac{|U_i|}{|U|}\leqslant q_i<1$。则，划分 $S_1\cup S_2\cup\ldots\cup S_\ell=U$ 关于 q_i 和 p_i 是宽松公平的，当且仅当所有集合 S_1、S_2、……、S_ℓ 关于 q_i 和 p_i 都是宽松公平的。

松弛公平相关聚类

输入： 图 $G=(V,E)$，$c\colon V\to[k]$，p_i，$q_i\in\mathbb{Q}$，$0<p_i\leqslant\frac{|U_i|}{|U|}\leqslant q_i<1$，$i\in[k]$

着色：，，其中对所有。**任务：** 寻找关于 p_i 和 q_i 的 V 的宽松公平划分$\mathcal{P}$，使代价（$\mathcal{P}$）最小。

虽然我们使用上述定义来得到困难性结果，但对于我们的算法，我们限制了 p_i 和 q_i 的可能性。

▶**定义 30** (α-宽松公平集)。设 U 是一个有限的元素集，由函数$c\colon U \to [k]$ 赋予颜色，其中 $k \in \mathbb{N}_{>0}$，且设 $0 < \alpha < 1$。那么，某个 $S \subseteq U$ 是 α- 宽松公平的，当且仅当它相对于 $p_i = \frac{\alpha|U_i|}{|U|}$ 和 $q_i = \frac{|U_i|}{\alpha|U|}$ 对所有 $i \in [k]$ 都是宽松公平的。◀

▶**定义 31** (α-宽松公平划分)。设 U 是一个有限的元素集，由函数 $c\colon U \to [k]$ 赋予颜色，其中 $k \in \mathbb{N}_{>0}$，且设 $0 < \alpha < 1$。那么，划分 $S_1 \cup S_2 \cup \ldots \cup S_\ell = U$ 是 α-宽松公平的，当且仅当所有集合 S_1、S_2、……、S_ℓ 都是 α-宽松公平的。

α-松弛公平相关聚类

输入： 图 $G = (V, E)$，着色 $c\colon V \to [k]$、$0 < \alpha < 1$。**任务：** 找到 V 的一个 α-宽松公平划分$\mathcal{P}$，使代价($\mathcal{P}$)最小。

7.2 宽松公平性的困难性

路径上精确公平性的困难性结果（见定理 14）可直接推广到宽松公平性情形。这是因为该结果只考虑每种颜色恰好有两个顶点的实例。由于任何宽松公平聚类仍要求每个簇中至少包含每种颜色的一个顶点，这意味着每个宽松聚类要么由单个簇构成，要么由两个簇构成，且每个簇都包含每种颜色的一个顶点。因此，在这些实例中，放宽公平性并无影响。

▶**推论 32**。路径上的松弛公平相关聚类 是NP-困难的，即使仅限于每种颜色恰好有 *2* 个顶点的实例。

我们关于宽松公平性的其他困难性证明基于这样一个想法：可以采用与精确公平性情形类似的构造，并进一步证明，在这些实例中，最小成本解必须满足精确公平性，而不只是宽松公平性。为此，我们需要一个引理，为聚类的簇内代价给出下界。

▶ 引理 33。设 $G = (V, E)$ 是一个 n 个顶点、m 条边的图，P 是 V 的一个划分，其簇间代价为 χ。那么，P 的簇内代价至少为 $\frac{n^2}{2|\mathrm{P}|} - \frac{n}{2} - m + \chi$。如果对于所有簇 $S \in \mathrm{P}$ 都有 $|S| = \frac{n}{|\mathrm{P}|}$，那么 P 的簇内代价恰好为 $\psi = \frac{n^2}{2|\mathrm{P}|} - \frac{n}{2} - m + \chi$。证明。我们首先证明下界。我们采用柯西-施瓦茨不等式：对于任意的 $\ell \in \mathbb{N}$、x_1、x_2、……、x_ℓ 以及 y_1、y_2、……、y_ℓ，都有 $\left(\sum_{i=1}^{\ell} x_i y_i\right)^2 \leqslant \left(\sum_{i=1}^{\ell} x_i^2\right) \cdot \left(\sum_{i=1}^{\ell} y_i^2\right)$。特别地，有 $\left(\sum_{i=1}^{\ell} x_i\right)^2 \leqslant \ell \cdot \sum_{i=1}^{\ell} x_i^2$。注意，我们可以将 P 的簇内代价 ψ 写成

$$\psi = \left(\sum_{S \in \mathcal{P}} \frac{|S| \cdot (|S| - 1)}{2}\right) - (m - \chi) = \frac{1}{2}\left(\sum_{S \in \mathcal{P}} |S|^2\right) - \left(\sum_{S \in \mathcal{P}} \frac{|S|}{2}\right) - m + \chi$$
$$= \frac{1}{2}\left(\sum_{S \in \mathcal{P}} |S|^2\right) - \frac{n}{2} - m + \chi.$$

由柯西-施瓦茨不等式可得 $\sum_{S \in \mathcal{P}} |S|^2 \geqslant \frac{1}{|\mathcal{P}|} \cdot \left(\sum_{S \in \mathcal{P}} |S|\right)^2 = \frac{n^2}{|\mathcal{P}|}$。由此可得，簇内代价至少为 $\psi \geqslant \frac{n^2}{2|\mathcal{P}|} - \frac{n}{2} - m + \chi$。

对于第二个结论，假设 $\mathcal{P}$ 的所有簇大小均为 $|^{n}_{|\mathcal{P}}$。那么，每个簇中都有 $1\,2\cdot|^{n}_{\mathcal{P}|}\cdot\left(|^{n}_{\mathcal{P}|}-1\right)$ 个顶点对。因此，我们有

$$\psi=|\mathcal{P}|\cdot\frac{1}{2}\cdot\frac{n}{|\mathcal{P}|}\cdot\left(\frac{n}{|\mathcal{P}|}-1\right)-(m-\chi)=\frac{n^2}{2|\mathcal{P}|}-\frac{n}{2}-m+\chi.$$ ◀

我们进一步证明，簇大小不等的任何聚类都无法达到引理 33 给出的下界。

▶ **引理 34.** 设 $G=(V,E)$ 是一个 n 个顶点、 m 条边的图，P 是 V 的一个划分，其簇间代价为 χ，且存在一个簇 $S\in P$，使得对于某个 $a\geqslant 0$，有 $|S|=\frac{n}{|\mathcal{P}|}+a$。那么，$P$ 的簇内代价为 $\psi\geqslant\frac{a^2|}{\mathcal{P}2|}|_{\mathcal{P}|-2}+\frac{n^2}{2|}\mathcal{P}|\frac{-\frac{n}{2}-m\pm\chi}{}$。–

证明. 若 $a=0$，则该结论由引理 33 推出。因此，假设 $a>0$。我们将簇内代价写作

$$\psi=\frac{1}{2}\cdot\left(\frac{n}{|\mathcal{P}|}+a\right)\cdot\left(\frac{n}{|\mathcal{P}|}+a-1\right)+\psi_{\text{rest}}$$

其中，ψ_{rest} 是由 $\mathcal{P}\setminus\{S\}$ 产生的簇内代价。将引理 33 应用于 $\mathcal{P}\setminus\{S\}$，可得

$$\begin{aligned}\psi&\geqslant\frac{1}{2}\cdot\left(\frac{n}{|\mathcal{P}|}+a\right)\cdot\left(\frac{n}{|\mathcal{P}|}+a-1\right)+\frac{\left(n-(\frac{n}{|\mathcal{P}|}+a)\right)^2}{2(|\mathcal{P}|-1)}-\frac{n-(\frac{n}{|\mathcal{P}|}+a)}{2}-m+\chi\\&=\frac{n^2}{2|\mathcal{P}|^2}+\frac{an}{|\mathcal{P}|}+\frac{a^2}{2}-\frac{n}{2|\mathcal{P}|}-\frac{a}{2}+\frac{n^2-2n^2/|\mathcal{P}|-2an+n^2/|\mathcal{P}|^2+2a\frac{n}{|\mathcal{P}|}+a^2}{2|\mathcal{P}|-2}\\&\quad-\frac{n}{2}+\frac{n}{2|\mathcal{P}|}+\frac{a}{2}-m+\chi.\end{aligned}$$

将前面的各项通分为 $2|\mathcal{P}|-2$，得到

$$\begin{aligned}\psi&\geqslant\left(\frac{n^2(|\mathcal{P}|-1)}{|\mathcal{P}|^2}+\frac{an(2|\mathcal{P}|-2)}{|\mathcal{P}|}+a^2(|\mathcal{P}|-1)+n^2-\frac{2n^2}{|\mathcal{P}|}-2an+\frac{n^2}{|\mathcal{P}|^2}+\frac{2an}{|\mathcal{P}|}+a^2\right)\\&\quad/(2|\mathcal{P}|-2)-\frac{n}{2}-m+\chi\\&=\left(\frac{n^2|\mathcal{P}|}{|\mathcal{P}|^2}+\frac{2an|\mathcal{P}|}{|\mathcal{P}|}+a^2|\mathcal{P}|+n^2-\frac{2n^2}{|\mathcal{P}|}-2an\right)/(2|\mathcal{P}|-2)-\frac{n}{2}-m+\chi\\&=\left(-\frac{n^2}{|\mathcal{P}|}+a^2|\mathcal{P}|+n^2\right)/(2|\mathcal{P}|-2)-\frac{n}{2}-m+\chi.\end{aligned}$$

然后我们将 $0=-\frac{n^2}{2|}\frac{\mathcal{P}|\cdot}{2|\mathcal{P}|-2}\frac{2|P|-2}{}+\frac{n^2}{2|}\frac{\mathcal{P}|}{}$ 相加，得到

$$\begin{aligned}\psi&\geqslant\left(-\frac{n^2}{|\mathcal{P}|}+a^2|\mathcal{P}|+n^2-\frac{n^2(|\mathcal{P}|-1)}{|\mathcal{P}|}\right)/(2|\mathcal{P}|-2)+\frac{n^2}{2|\mathcal{P}|}-\frac{n}{2}-m+\chi\\&=\frac{a^2|\mathcal{P}|}{2|\mathcal{P}|-2}+\frac{n^2}{2|\mathcal{P}|}-\frac{n}{2}-m+\chi.\end{aligned}$$ ◀

注意到，由于 $|\mathcal{P}|>1$ 且 $a\neq 0$，这意味着此类聚类永远无法达到引理 33 给出的下界。特别地，这意味着，对于森林中最小代价公平聚类的固定簇间代价，所有簇的大小都相等。这样，我们便能够将针对精确公平性得到的一些困难性结果迁移到宽松公平性。

▶ **定理 35**。 对于 $0 < p_1 \leqslant \frac{1}{c+1} \leqslant q_1 < 1$ 和 $0 < p_2 \leqslant \frac{c}{c+1} \leqslant q_2 < 1$的任意选择，在颜色比例为1： c 的森林上，松弛公平相关聚类是NP-困难的。当任意限制森林中树的形状时，只要对于每个 $a \in \mathbb{N}$ 都可以构造一棵具有 a 个顶点的树，该问题仍然是NP-困难的。

证明。我们从3-划分归约。回顾一下，存在 $3p$ 个数值 a_1， a_2，……， a_{3p} 任务是将它们划分为三元组，使每个三元组的和均为 B。我们按如下方式构造一个森林 F。对于每个 a_i，我们构造一棵由 a_i 个红色顶点组成的任意树。此外，我们设置 p个孤立蓝色顶点。注意，蓝色顶点与红色顶点的比例为1: B。现在我们证明，存在一个宽松公平聚类$\mathcal{P}$ 满足

$$\operatorname{cost}(\mathcal{P}) \leqslant p \cdot \frac{B(B+1)}{2} - p(B-3)$$

当且仅当给定实例是 3-划分的肯定实例时。

如果我们有一个 3-划分的肯定实例，那么就存在一种对树集合的划分，将其分为p 个大小为 B的簇。将蓝色顶点任意分配到各自唯一的簇中，由此我们便得到一个完全公平划分，因此它也是宽松公平的。由于簇之间没有边，且每个簇包含 $B+1$ 个顶点和 $B-3$ 条边，该划分的代价为 $p \cdot \frac{B}{(B+1)\,2} - p(B-3)$.

对于另一个方向，假设存在一个宽松公平聚类 $\mathcal{P}$，使得代价(cost($\mathcal{P}$) $\leqslant p \cdot \frac{B}{(B+1)\,2} - p($ $B-3)$。我们证明，该聚类不仅满足宽松公平性，而且恰好满足公平性。注意， $|V| = p(B+1)$ 和 $|E| = p(B-3)$。由于簇间代价 χ 非负，根据引理 33，簇内代价的下界为

$$\psi \geqslant \frac{(p(B+1))^2}{2|\mathcal{P}|} - \frac{p(B+1)}{2} - p(B-3).$$

由于蓝色顶点恰有 p 个，且宽松公平性约束要求每个簇中至少有一个蓝色顶点，因此我们有 $|\mathcal{P}| \leqslant p$。因此，

$$\psi \geqslant \frac{p(B+1)^2}{2} - \frac{p(B+1)}{2} - p(B-3) = p \cdot \frac{B(B+1)}{2} - p(B-3) \geqslant \operatorname{cost}(\mathcal{P}).$$

这意味着 $\mathcal{P}$ 的簇间代价为 0，且 $|\mathcal{P}| = p$。由引理 34 可知，$\mathcal{P}$ 中所有簇恰好由 $B+1$ 个顶点组成。由于这 p 个簇中每个至少含有 1 个蓝色顶点，而蓝色顶点总数为 p 个，因此我们知道每个簇由 1 个蓝色顶点和 B 个红色顶点组成。由于所有树的大小均大于 $\frac{B}{4}$ 且小于 $\frac{B}{2}$，这意味着每个簇恰好由一个蓝色顶点和三棵未切割的树组成，且总计有 B 个顶点。这样，这种聚类便给出了一个 3-划分的解，因此我们的实例是一个肯定实例。

由于构造该图只需关于实例规模的多项式时间，因此这意味着我们的困难性结果。◀

事实上，我们注意到，对于任何允许完全公平解且要求每个簇中至少包含 1 个各颜色顶点的公平性约束，我们都能得到这一困难性结果。将我们的树上直径为 4 的困难性证明进行迁移时，情况也相同。

▶ **定理 36**。 对于任意选择的 $0 < p_1 \leqslant \frac{1}{c+1} \leqslant q_1 < 1$ 和 $0 < p_2 \leqslant \frac{c}{c+1} \leqslant q_2 < 1$，在直径为 4、两种颜色比例为 1： c 的树上进行松弛公平相关聚类是 *NP* 难的。

证明。 我们从 3-划分归约。我们假设 $B^2 > 16p$。这样做是可行的，因为将所有 a_i 和 B 乘以相同的因子（此处为O(p)中的某个值）后，我们仍得到一个等价的 3-划分 实例。对于每个 a_i，我们构造一个由 a_i 个红色顶点组成的星形图。此外，我们构造一个由 p 个蓝色顶点组成的星形图。将蓝色星形图的中心 v 与所有红色星形图的中心相连，即得到一棵直径为 4 的树。注意，蓝色顶点与红色顶点的比例为 1： B。现在我们证明存在一个松弛公平聚类 $\mathcal{P}$ 满足

$$\operatorname{cost}(\mathcal{P}) \leqslant \frac{pB^2 - pB}{2} + 7p - 7$$

当且仅当给定实例是 3-划分的肯定实例时。

如果 3-划分存在肯定实例，那么就存在一种将星形图集合划分为p 个簇的方式，每个簇包含 B 个星形图，每个簇由三个星形图组成。将蓝色顶点任意分配到各自唯一的簇中，由此即可得到一个精确公平划分，因此它也满足宽松公平性。我们先计算簇间代价。若一条边连接两个蓝色顶点或两个红色顶点，我们分别称其为蓝色边或红色边。若一条边连接一个蓝色顶点和一个红色顶点，则称其为蓝红边。所有 $p-1$ 条蓝色边都会被割断。此外，连接 v（蓝色星形图的中心）与红色顶点的所有边都会被割断，唯独分配给 v 的三个星形图除外。这会额外产生 $3p-3$ 个割，因此簇间代价为 $\chi = 4p-4$。每个簇都包含 $B+1$ 个顶点和$B-3$ 条边，唯独包含 v 的簇有 B 条边。簇内代价为

$$\psi = p\left(\frac{B(B+1)}{2} - B + 3\right) - 3 = \frac{pB^2 - pB}{2} + 3p - 3.$$

将簇内代价与簇间代价相加，即可得到所需的代价

$$\operatorname{cost}(\mathcal{P}) = \chi + \psi = \frac{pB^2 - pB}{2} + 7p - 7.$$

对于另一个方向，假设存在一个宽松公平聚类 $\mathcal{P}$ ，使得 $\operatorname{cost}(\mathcal{P}) \leqslant \frac{pB^2-pB}{2} + 7p - 7$。我们证明该聚类不仅是宽松公平的，而且是完全公平的。

为此，我们首先证明 $|\mathcal{P}| = p$。由于每个簇都需要一个 p 蓝色顶点，因此有 $|\mathcal{P}| \leqslant p$。现在，令 χ 表示 $\mathcal{P}$ 的簇间代价。注意， $|V| = p(B+1)$ 和 $|E| = p(B-3) + 3p + p - 1 = p(B+1) - 1$。于是，根据引理 33，我们有

$$\begin{aligned}\psi &\geqslant \frac{(p(B+1))^2}{2|\mathcal{P}|} - \frac{p(B+1)}{2} - (p(B+1)-1) + \chi \\ &= \frac{p^2B^2 + 2p^2B + p^2}{2|\mathcal{P}|} - \frac{3p(B+1)}{2} + 1 + \chi. \end{aligned} \tag{2}$$

注意，下界随 $|\mathcal{P}|$ 的增大而减小。如果 $|\mathcal{P}| \leqslant p-1$，那么

$$\psi \geqslant \frac{p^2B^2 + 2p^2B + p^2}{2(p-1)} - \frac{3p(B+1)}{2} + 1 + \chi.$$

由于簇间代价 χ 为非负数，因此我们由此得到

$$\begin{aligned}\operatorname{cost}(\mathcal{P}) &\geqslant \frac{p^2B^2 + 2p^2B + p^2}{2(p-1)} - \frac{3p(B+1)}{2} + 1 + \chi \\ &\geqslant \frac{p^2B^2 + 2p^2B + p^2}{2(p-1)} - \frac{3p^2B - 3pB + 3p^2 - 3p}{2(p-1)} + \frac{2p-2}{2(p-1)} \\ &\geqslant \frac{p^2B^2 - p^2B - 2p^2 + 3pB + 5p - 2}{2(p-1)}.\end{aligned}$$

然而，我们知道

$$\begin{aligned}\operatorname{cost}(\mathcal{P}) &\leqslant \frac{pB^2-pB}{2}+7p-7\\ &= \frac{p^2B^2-pB^2-p^2B+pB+14p^2-14p-14p+14}{2(p-1)}\\ &= \frac{p^2B^2-pB^2-p^2B+pB+14p^2-28p+14}{2(p-1)}.\end{aligned}$$

因此，$|\mathcal{P}| \leqslant p-1$ 成立的条件是 $-2p^2+3pB+5p-2 \leqslant -pB^2+pB+14p^2-28p+14$，这等价于 $pB^2-16p^2+2pB+33p-16 \leqslant 0$。由于我们假设 $B^2 > 16p$，这始终为假，因此 $|\mathcal{P}| = p$。将其代入方程 2 得到

$$\psi \geqslant \frac{pB^2+2pB+p}{2}-\frac{3p(B+1)}{2}+1+\chi = \frac{pB^2-pB}{2}-p+1+\chi.$$

由于 $\operatorname{cost}(\mathcal{P}) = \chi+\psi$，我们有

$$\frac{pB^2-pB}{2}-p+1+2\chi \leqslant \operatorname{cost}(\mathcal{P}) \leqslant \frac{pB^2-pB}{2}+7p-7, \tag{3}$$

由此得到 $\chi \leqslant 4p-4$。

由于没有两个蓝色顶点位于同一个簇中，蓝色顶点之间的割产生的簇间代价恰为 $p-1$。为了估计被切断的蓝红边的数量，令a 表示位于蓝色中心顶点 v 所在簇中的红色中心顶点数量。于是，$3p$ 条红色边中有 $3p-a$ 条被切断。令 χ_r 表示被切断的红色边的数量。注意，$\chi = p-1+3p-a+\chi_r = 4p-a-1+\chi_r$。

我们证明 $a=3$。由于 $\chi \leqslant 4p-4$，我们有 $\chi_r - a \leqslant -3$，由此可得 $a \geqslant 3$。接下来，我们将 χ_r 界定为 a。设 $\delta \in \mathbb{Z}$ 满足 $B+\delta$ 是包含蓝色中心顶点 v 的簇中的红色顶点数。那么，

$$\chi_r \geqslant \frac{aB}{4}-(B+\delta-a) = \frac{(a-4)B}{4}-\delta+a$$

因为每个红色中心顶点至少与 $\frac{B}{4}$ 个红色叶节点连通，而在 v 的簇中只能容纳其中的 $B+\delta-a$ 个。首先，假设 $\delta \leqslant 0$。这意味着 $\chi_r - a \geqslant (a-4)_4{}^B$。由于我们要求 ~~$\chi_r - a$~~ $\leqslant -3$，因此可得 $a<4$，正如所需。

情形 $\delta \geqslant 1$ 稍微复杂一些。由引理 34，$p = |\mathcal{P}|$，且 $m = n-1 = p(B+1)-1$，我们得到

$$\psi \geqslant \frac{\delta^2|\mathcal{P}|}{2|\mathcal{P}|-2}+\frac{(p(B+1))^2}{2|\mathcal{P}|}-\frac{p(B+1)}{2}-m+\chi = \frac{\delta^2 p}{2p-2}+\frac{pB^2+2pB+p}{2}-\frac{3p(B+1)}{2}+\chi+1.$$

由此可得

$$\frac{\delta^2 p}{2p-2}+\frac{pB^2-pB}{2}-p+2\chi+1 \leqslant \operatorname{cost}(\mathcal{P}) \leqslant \frac{pB^2-pB}{2}+7p-7.$$

由此不等式可推得 $\chi \leqslant 4p-4-\frac{\delta^2 p}{4p-4}$ 和 $\chi_r - a \leqslant -3-\frac{\delta^2 p}{4p-4}$，这意味着 ——

$$\frac{(a-4)B}{4}-\delta \leqslant -3-\frac{\delta^2 p}{4p-4}$$

右侧关于 δ 递减，通过代入情况$\delta \geqslant 1$下的最小值，我们最终得到 $(a-4)_4{}^B \leqslant -2-\frac{p}{4p-4}$。这表明在此处 $a<4$ ~~也必须~~成立。 ——

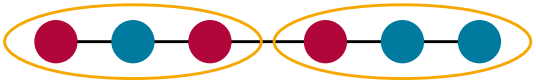

图 11 示例路径，其颜色比例为 1:1，其中存在一种 $_{2\ 3}$-宽松公平聚类，代价为 f 3（橙色线条标出），而代价最低的完全公平聚类的代价为 4。

因此，我们证明了 $a = 3$，这也给出了 $\chi_r = 0$ 和 $\chi = 4p - 4$。所以，我们不仅有 cost($\mathcal{P}$) $\leqslant {}^{pB^2-pB}_{2} + 7p - 7$，而且 cost($\mathcal{P}$) $= {}^{pB^2-pB}_{2} + 7p - 7$。由公式 3 可见，对于 $\chi = 4p - 4$，这恰好达到了引理 33 所确立的下界。因此，根据引理 34，这意味着所有簇都恰好由 1 个蓝色顶点和 B 个红色顶点组成，并且该聚类完全公平。

如 $\chi_r = 0$ 所示，所有红色星形图都是完整的。鉴于每个红色星形图的规模至少为 ${}^{B}_{4}$ 且至多为 ${}^{B}_{2}$，这意味着每个簇恰由三个完整的红色星形图组成，且红色顶点总数为 B，因此可得出 3-划分实例的一个解。由于图的构造只需关于实例规模的多项式时间，且构造出的树的直径为 4，这就得出了我们的困难性结果。◀

在本节的困难性证明中，我们论证了对于所构造的实例，宽松公平但并非完全公平的聚类，其代价会高于完全公平聚类。然而，这种情况并非普遍成立。即使将范围限制在路径和颜色比例为 1:1 的两种颜色上，这一结论也不成立，如图 11 所示。

因此，我们几乎不可能提出一种通用方案，将第 5 节中的所有困难性证明一次性转化到宽松公平性设定中。因此，我们必须像处理定理 35 和 36 那样，分别证明该设定下的困难性结果。我们乐观地认为，其他困难性结果在该设定下仍然成立，尤其是因为定理13的构造与本节采用的构造类似。我们将这些结果的转换留待未来工作。

7.3 宽松公平性的算法

我们还能够将定理 23 的算法结果推广到一种特定的 α-宽松公平性设定。我们利用了这样一点：该算法实际上并不依赖精确公平性，而是依赖于簇大小存在上界这一事实，这使我们能够计算相应的分裂。接下来，我们将证明，对于两种颜色比例为 1:1 的 α-宽松公平性，也存在这样的上界，并据此调整算法。为了计算该上界，我们首先证明引理 37；与引理 4 类似，它对簇大小作出限制，但适用于无色森林。利用该引理和引理 38，我们随后证明，在两种颜色比例为 1:1 的森林中，最小代价 α-宽松公平聚类的簇大小存在上界。

▶ **引理 37**。 设 $F = (V, E)$ 是一个具有 n 个顶点和 m 条边的森林，并且 $P_1 = \{V\}$。此外，设$S \subset V$满足 $4 < |S| \leqslant n - 3$，并设 $P_2 = \{S, V \setminus S\}$。则，代价（$P_1$）$>$ 代价（P_2）。

证明。 由于有 ${}^{n}_{(n-1)\ 2}$ 个顶点对，且有 m 条边，其中没有一条被$\mathcal{P}_1$割开，因此代价($\mathcal{P}_1$) $= {}^{n}_{(n-1)\ 2} - m$。在最坏情况下，$\mathcal{P}_2$ 会割开森林中至多 $n - 1$ 条边，

森林中有一个大小为 $|S|$ 的簇和一个大小为 $n-|S|$ 的簇，因此

$$\begin{aligned}\operatorname{cost}(\mathcal{P}_2) &\leqslant n-1+\frac{(n-|S|)(n-|S|-1)}{2}+\frac{|S|(|S|-1)}{2}-(m-n-1)\\ &=\frac{n(n-1)}{2}+\frac{-2n|S|+|S|^2+|S|}{2}+\frac{|S|^2-|S|}{2}-m+2n-2\\ &=\frac{n(n-1)}{2}-n|S|+|S|^2-m+2n-2.\end{aligned}$$

于是，我们有

$$\operatorname{cost}(\mathcal{P}_1)-\operatorname{cost}(\mathcal{P}_2)\geqslant n|S|-|S|^2-2n+2\geqslant(|S|-2)n-|S|^2+2.$$

注意到，该界限随 n 增大而增大。由于我们有 $n\geqslant|S|+3$ 和 $|S|>4$，因此得到

$$\operatorname{cost}(\mathcal{P}_1)-\operatorname{cost}(\mathcal{P}_2)\geqslant(|S|-2)(|S|+3)-|S|^2+2=|S|-4>0.$$ ◀

了解何时拆分一个簇的代价更低后，我们现在证明，即使对于 α-宽松公平相关聚类，在森林中的最小成本解里，簇大小也存在一个上界。其思路是假设存在一个具有特定大小的簇，然后论证我们可以以某种方式拆分它，从而降低代价并保持 α-宽松公平性。

▶ **引理 38**。设 F 是一个具有两种颜色且比例为 $1:1$ 的森林。设 $0<\alpha<1$，并令 $\hat{\alpha}\in\mathbb{N}$ 为满足 $\frac{2\hat{\alpha}}{\alpha}\in\mathbb{N}$ 且 $\frac{2\hat{\alpha}}{\alpha}>4$ 的最小值。则，如果 P 是 F 上的一个最小成本 α-宽松公平聚类，则对于所有 $S\in P$，都有 $|S|<4\frac{\hat{\alpha}}{\alpha^2}$。

证明。反设不成立，即存在一个簇 S，使得 $|S|\geqslant 4\frac{\hat{\alpha}}{\alpha^2}$。令 b 和 r 分别表示 S 中的蓝色和红色顶点数量，并不失一般性地假设 $b\leqslant r$。由于 $|S|\geqslant 4\frac{\hat{\alpha}}{\alpha^2}$，我们有 $\frac{\alpha}{2}\geqslant\frac{2\hat{\alpha}}{\alpha|S|}$。根据 α-松弛公平性约束，可得 $\frac{b}{|S|}\geqslant\frac{2\hat{\alpha}}{\alpha}$，进而有 $r\geqslant b\geqslant\frac{2\hat{\alpha}}{\alpha}$。

接着，考虑将 $\hat{\alpha}$ 个蓝色顶点和 $\frac{2\hat{\alpha}}{\alpha}-\hat{\alpha}$ 个红色顶点从 S 中分离出来，放入一个新的簇 S_1，由此得到的聚类，并令 $S_2=S\setminus S_1$。注意，我们选择 $\hat{\alpha}$ 时确保这一操作可行，即两个数量都是自然数。由于所有至多有一个端点位于 S 中的边所产生的代价保持不变，而两个端点都位于 S 中的边所产生的代价会降低（如引理 37 所示），因此新聚类的代价低于 $\mathcal{P}$。我们接下来将证明这个新聚类也满足 α-宽松公平性，因此这与 $\mathcal{P}$ 的最优性矛盾。

我们首先证明 S_1 的 α-宽松公平性。对于蓝色顶点，在 S_1 中有 $\frac{\alpha\hat{\alpha}+2\hat{\alpha}-\alpha\hat{\alpha}}{\alpha}=\frac{\alpha}{2}$ 的一部分，这符合 α-松弛公平性约束。对于红色顶点，我们有 $\frac{2\hat{\alpha}-\alpha\hat{\alpha}}{\alpha\hat{\alpha}+2\hat{\alpha}-\alpha\hat{\alpha}}=1-\frac{\alpha}{2}$，由于 $0<\alpha<1$，这符合 α-松弛公平性约束，因此 $1-\frac{\alpha}{2}\geqslant\frac{\alpha}{2}$ 和 $1-\frac{\alpha}{2}=\frac{2\alpha-\alpha^2}{2\alpha}\leqslant\frac{1}{2\alpha}$。

现在我们证明 S_2 的 α-宽松公平性。S_2 中蓝色顶点所占的比例为 $\frac{b-\hat{\alpha}}{r+b-\frac{2\hat{\alpha}}{\alpha}}$，因此我们需要证明该值介于 $\frac{\alpha}{2}$ 和 $\frac{1}{2\alpha}$ 之间。我们首先通过证明 $\frac{\alpha}{2}\cdot\left(r+b-\frac{2\hat{\alpha}}{\alpha}\right)\leqslant b-\hat{\alpha}$，来说明该值至少为 $\frac{\alpha}{2}$。由于 S 是 α-宽松公平的，因此 $r\leqslant\frac{2b}{\alpha}-b$，否则 $\frac{b}{b+r}<\frac{b}{b+\frac{2b}{\alpha}-b}=\frac{\alpha}{2}$。因此，我们有

$$\frac{\alpha}{2}\cdot\left(r+b-\frac{2\hat{\alpha}}{\alpha}\right)\leqslant\frac{\alpha}{2}\cdot\left(\frac{2b}{\alpha}-b+b-\frac{2\hat{\alpha}}{\alpha}\right)=b-\hat{\alpha}.$$

同样，我们通过证明等价命题 $2\alpha(b-\hat{\alpha})\leqslant r+b-\frac{2\hat{\alpha}}{\alpha}$，说明比例至多为 $\frac{1}{2\alpha}$。由于我们假设 $r\geqslant b$，因此有

$$r+b-\frac{2\hat{\alpha}}{\alpha}\geqslant 2b-\frac{2\hat{\alpha}}{\alpha}\geqslant 2\left(b-\frac{\hat{\alpha}}{\alpha}-\left((1-\alpha)b+(\alpha^2-1)\frac{\hat{\alpha}}{\alpha}\right)\right)=2\alpha\left(b-\hat{\alpha}\right).$$

48 公平相关聚类

第二步成立是因为我们假设了 $b \geqslant \frac{\alpha}{\hat{\alpha}+\hat{}\alpha} = \frac{\alpha}{\alpha-\alpha\hat{}+1-\alpha}$，因此有 $(1-\alpha)b + (\alpha^2-1)\frac{\alpha}{\alpha} \geqslant 0$。现在，我们考察 S_2 中红色顶点所占的比例，即 $r - (\frac{2\hat{}\alpha}{\alpha} - \hat{}\alpha)\, r+b-\frac{2\hat{\alpha}}{\alpha}$。我们知道 $r \geqslant \frac{2\hat{}\alpha}{\alpha}$，也就是说，$(1-\alpha)r \geqslant \frac{2\hat{}\alpha}{\alpha} - 2\hat{}\alpha$，换句话说，$r - \left(\frac{2\hat{\alpha}}{\alpha} + \hat{}\alpha\right) \geqslant \alpha r - \hat{}\alpha$。由于 $r \geqslant b$，这意味着

$$r - \left(\frac{2\hat{\alpha}}{\alpha} + \hat{\alpha}\right) \geqslant \frac{\alpha}{2} \cdot \left(r + b - \frac{2\hat{\alpha}}{\alpha}\right)$$

因此 $r - (\frac{2\hat{}\alpha}{\alpha} - \hat{}\alpha)\, r+b-\frac{2\hat{\alpha}}{\alpha} \geqslant \frac{\alpha}{2}$。

仍需证明该比例也至多为 $\frac{1}{2\alpha}$。我们有 $r \geqslant \frac{2\hat{}\alpha}{\alpha} - \hat{}\alpha$，这等价于

$$\left(2\alpha - 1 - \frac{\alpha}{2-\alpha}\right) r \leqslant 4\hat{\alpha} - 2\alpha\hat{\alpha} - \frac{2\hat{\alpha}}{\alpha}.$$

注意到 $2\alpha - 1 - \frac{\alpha}{2-\alpha} = -\frac{2\alpha^2-4\alpha+2}{2-\alpha} = -\frac{2(\alpha-1)^2}{2-\alpha} \leq 0$，并且 $r \leqslant \frac{2b}{\alpha} - b$ 给出 $b \geqslant \frac{r}{\frac{2}{\alpha}-1} = \frac{\alpha r}{2-\alpha}$。由此，上述不等式意味着

$$(2\alpha-1)r - b \leqslant 4\hat{\alpha} - 2\alpha\hat{\alpha} - \frac{2\hat{\alpha}}{\alpha}$$

由此，我们最终得到 $2\alpha \cdot \left(r - \left(\frac{2\hat{}\alpha}{\alpha} - \hat{}\alpha\right)\right) \leqslant r + b - \frac{2\hat{}\alpha}{\alpha}$，即 $r - (\frac{2\hat{\alpha}}{\alpha} - \hat{}\alpha)\, \frac{r+b-2\hat{}\alpha}{\alpha} \leqslant \frac{1}{2\alpha}$。

这证明了 S_1 和 S_2 都是 α-宽松公平的。由于将 S 划分为 S_1 和 S_2 后仍然是α-宽松公平的，且代价更低，这与 S 属于最小代价的 α-宽松公平聚类相矛盾。◀

现在，我们可以调整第 6.3 小节中介绍的算法，以求解具有两种颜色且比例为 1:1 的森林上的松弛公平相关聚类。原算法利用了任何最优解都具有最小大小的公平簇这一性质，而借助引理 38，我们也能够在 α-宽松设置下对簇的大小进行界定。

与原算法类似，我们首先创建一个可能的分裂列表。然而，这些分裂中不仅可以包含一个或两个顶点的连通分量——我们知道，对于两种颜色且比例为 1:1 的精确公平性而言，这样就足够了——每个连通分量还可以包含最多 $4\,\hat{}\frac{\alpha}{\alpha^2}$ 个顶点，其中 $\hat{}\alpha$ 是满足 $\frac{2\hat{}\alpha}{\alpha} \in \mathbb{N}$ 且 $\frac{2\hat{}\alpha}{\alpha} > 4\hat{}$ 的最小自然数，如引理 38 中所定义。在下文中，我们令 $d = \overline{4}\frac{\alpha}{\alpha^2}$ 表示簇的这一最大大小。在第二阶段，算法会检查这些分裂中哪些可以合并为α-宽松公平聚类，并从中返回代价最小的一个。

森林拆分。

为了找到获得每种可能着色划分方式的最优方案，我们只需应用引理 25，并设定 $d_1 = d_2 = d$，因为我们知道，最优解必定存在于每种颜色的顶点数都不超过 d 的簇中。这一阶段的时间复杂度为 $\mathsf{O}(n^{2(d+1)^2+2d+2} \cdot \left((d+1)^2\right)^{2d}) = \mathsf{O}(n^{2d^2+6d+4} \cdot (d+1)^{4d})$。

构造公平聚类。

在第二阶段，我们必须在 $D_r^{\emptyset}$ 中找到一种分裂，使其能够转化为 α-宽松公平聚类，并产生最小的相关聚类代价。由于我们在第一阶段已经记录了各分裂可能的划分着色对应的最小簇间代价，因此

本阶段无需考虑再割掉更多边，因为对于所得分裂的着色，我们已经记录了最小簇间代价。因此，唯一需要考虑的是：一种分裂是否可组装，即能否合并其连通分量，使之成为 α-宽松公平聚类；如果可以，代价最低的合并方式是什么。

关于第一个问题，注意到可组装性 仅取决于该划分的划分着色。因此，第一阶段只跟踪分裂的所有可能划分着色，而不跟踪分裂本身的所有可能情况，并无妨碍。首先，注意分裂的着色本身可能产生一个 α-宽松公平聚类。我们将所有此类划分着色标记为可组装，所需时间为 $O(n^{d^2+1})$。对于其余划分着色，我们采用以下动态规划。

回顾一下，划分着色的大小指的是其中包含的集合着色数量（不一定是不同集合着色的数量）。我们从小到大判断所有可能的划分着色是否可组装。注意，每个划分着色的大小至少为 $\lceil \frac{n}{d} \rceil$。若其大小恰为 $\lceil \frac{n}{d} \rceil$，则不存在两个集合着色，在合并后仍能使大小不超过 d，因为所有其他集合着色的大小至多为 d。因此，在这种情况下，分裂可组装当且仅当它已经是一个 α-宽松公平聚类，所以我们已经正确地标记了这些划分着色。现在，假设我们已经判断了所有大小为 $i \geqslant \lceil \frac{n}{d} \rceil$ 的划分着色是否可组装。我们任取一个大小为 $i+1$ 的划分着色 C，它尚未被标记为可组装。当且仅当将其中至少两个集合着色合并后形成一个 α-宽松公平聚类时，它才可组装。具体而言，当且仅当 C 中存在两个集合着色 C_1、C_2，使得从 C 中移除集合着色 C_1、C_2，并加入由 C_1 和 C_2 的组合着色得到的集合着色后形成的着色 C' 是可组装的，它才可组装。注意，C' 的大小为 i。因此，给定所有大小为 i 的可组装划分着色后，我们可以通过对每个大小为 i 的划分着色尝试将其中一个集合着色拆分为两个的每种可能方式，找出所有大小为 $i+1$ 的可组装划分着色。由于大小为 i 的划分着色至多有 i^{d^2} 个，这需要 $O(i^{d^2} \cdot i \cdot 2^d)$ 的时间。整个动态规划所需时间为 $O(n^{d^2+1} \cdot 2^d) \subseteq O(n^{d^2+d+1})$。

剩下的问题是如何选择能够得到最小代价的组装方式。在精确公平性的算法中，我们不必为此担心，因为我们可以假设相关聚类代价只取决于簇间代价。而在这里情况并非如此，因为 α-宽松公平性允许簇的大小各不相同，因此引理 3 不适用。不过，请回想一下，我们可以将顶点的某个划分$\mathcal{P}$ 的相关聚类代价写成 $\sum_{S \in \mathcal{P}} |S|(|S|-1)/2 + 2\chi$，其中 χ 是簇间代价。因此，代价只取决于簇间代价和各簇的大小，而后者又取决于划分着色。要计算某个分裂的代价，我们采用第一阶段针对 χ计算出的簇间代价。同样，我们忽略合并簇所导致的簇间代价降低，因为由此得到的分裂也会包含在第一阶段生成的数组中。根据柯西-施瓦茨不等式，我们可以看出，合并簇只会增加 $\sum_{S \in \mathcal{P}} |S|(|S|-1)/2$ 的值，因为此时平方项的数量更少、数值却更大。因此，对于本身就是 α-宽松公平的分裂，可获得的最低代价就是该分裂本身的聚类代价。如果某个分裂可组装，但本身并非 α-宽松公平，那么它能够合并成的所有 α-宽松公平分裂的代价总和中的最小值，就是所求的代价。只需在上述动态规划中，除了向下传递可组装性外，也传递该总和的值，并在某个分裂有多个选项时取最小值，即可轻松计算出这一数值。这不会改变渐近运行时间，而第二阶段的运行时间由第一阶段主导。

因此，完整算法的运行时间为 $O(n^{2d^2+6d+4} \cdot (d+1)^{4d})$。

▶**定理 39**。 设 F 是一个有 n 个顶点的森林，顶点分为两种颜色，比例为 1:1。那么，在 F 上的 α-松弛公平相关聚类 可在 $O(n^{2d^2+6d+4} \cdot (d+1)^{4d})$ 时间内求解，其中 $d = 4 \lceil \frac{\alpha}{\alpha^2} \rceil$，且 $\lceil \alpha \rceil \in \mathbb{N}$ 是满足 $2 \lceil \frac{\alpha}{\alpha} \rceil \in \mathbb{N}$ 和 $2 \lceil \frac{\alpha}{\alpha} \rceil > 4$ 的最小值。

我们确信，引理 38 可以推广到任意颜色数量和任意比例的情形，使得最大簇大小可由 α 和颜色比例的某个函数界定。鉴于这一引理在 1 : 1 情形下的复杂性，我们将此问题留作未来工作。如果能够证明这样的界，那么定理 39 中采用的算法方法便适用于任意颜色划分的森林。同样，也可以证明更一般的松弛公平聚类中簇大小的界。作为一种中间方案，我们指出，对于松弛公平相关聚类，可以采用用于 α-松弛公平相关聚类的方法：将 α 设得足够大，以涵盖所有允许的解，并在组装阶段筛除不符合松弛公平性约束的解。此处不详细给出这一过程，因为我们认为，对于这些情况，更有希望的做法是计算最大簇大小的精确上界，并据此执行算法，而不是将问题归约为 α-松弛变体。

8 近似

到目前为止，我们一直致力于在各种实例中求解公平相关聚类的最优解。对于困难问题，常用的一种补救方法是采用近似算法：它们不一定能找到最优解，但能高效地找到近乎最优的解，例如针对（非公平）相关聚类的 2.06-近似算法 [17]。本节将说明，任意采用一种公平聚类都能得到相当接近最优解的近似；而且，若由颜色比例决定的任意公平簇的最小规模增大，这种近似会更加接近最优解。

形式化地说，如果对于每个实例 I，都存在一个可行解集合 $S(I)$ 和一个目标函数 m: $S(I) \to \mathbb{R}_{>0}$，为每个解赋予一个分数，那么该问题就是一个优化问题。根据问题定义，在所有可行解中，分数最高或最低的某个 $S \in S(I)$ 就是最优解。我们将该解的分数称为 $m^*(I)$。例如，对于公平相关聚类，实例由一个带有着色顶点的图给出；顶点的每种公平聚类都是一个可行解；分数为相关聚类代价；目标是使该代价最小。[8] 一个优化问题的 α-近似算法，是指一种算法：对于每个实例 I，它输出一个可行解 $S \in S(I)$，满足 $\frac{1}{\alpha} \leqslant \frac{m(S)}{m^*(I)} \leqslant \alpha$。具体到公平相关聚类，这意味着算法输出的公平聚类，其代价至多为最小聚类代价的 α 倍。

APX 是这样一类问题：它们都允许在 $\alpha \in O(1)$ 内进行 α-近似。多项式时间近似方案（PTAS）是一种算法：对于每个优化问题实例以及参数 $\varepsilon > 0$，它能在 $O(n^{f(\varepsilon)})$ 时间内，针对最小化问题计算一个 $(1+\varepsilon)$-近似，或针对最大化问题计算一个 $(1-\varepsilon)$-近似，其中 f 是一个仅取决于 ε 的可计算函数。我们使用 PTAS 来指代满足以下条件的问题类：

[8] 我们注意到，聚类成本可能为 0，这与定义 m: $S(I) \to \mathbb{R}_{>0}$ 相矛盾。然而，任何代价为 0 的聚类都仅由图的连通分量构成。我们不考虑这些平凡实例。

允许使用 PTAS 的优化问题。若一个优化问题 L 满足：APX 中的每个问题都可以 PTAS-归约到 L，则称其为 APX-困难；也就是说，若 L 存在 PTAS，则 APX 中的每个问题都存在 PTAS。若 L 本身还属于 APX，则称 L 为 APX-完全。根据定义，有 PTAS $\subseteq$ APX。此外，除非 P $=$ NP，否则 PTAS $\neq$ APX。

我们发现，对森林采用<风格 id='1'>任意</风格>公平聚类都能得到良好的近似。

▶ **定理 40**。 设 F 是一个具有 $k \geqslant 2$ 种颜色、颜色比例为 $c_1 : c_2 : \ldots : c_k$ 且 $d = \sum_{i=1}^{k} c_i \geqslant 4$的 n顶点 m边森林。那么，对于 F 上的<样式 id='30'>公平相关聚类</样式>，存在一个 $\frac{(d^2-d)n+2dm}{(d^2-5d+4)n+2dm}$近似算法，其运行时间为 $\mathrm{O}(n)$。

证明。 首先按颜色对顶点排序，然后迭代地将每种颜色的下一个 c_i 顶点 i 加入下一个簇中，我们在线性时间内得到一个公平聚类 $\mathcal{P}$，其簇大小为 d。在最坏情况下，$\mathcal{P}$ 割断所有 m 条边。因此，根据引理 3，我们有

$$\mathrm{cost}(\mathcal{P}) \leqslant \frac{(d-1)n}{2} - m + 2m = \frac{(d-1)n}{2} + m.$$

我们将该代价与最小成本公平聚类 $\mathcal{P}^*$的代价进行比较。根据引理 4，$\mathcal{P}^*$由大小为 d的簇组成。由于森林结构，每个 $\frac{n}{d}$簇至多包含 $d-1$ 条边。因此，至多有 $\frac{n}{d} \cdot (d-1)$条边位于簇内。于是，对于簇间代价，我们有 $\chi \geqslant m - \frac{n}{d} \cdot (d-1) = \frac{n}{d} - n + m$。引理 3 给出

$$\mathrm{cost}(\mathcal{P}^*) \geqslant \frac{(d-1)n}{2} - m + 2\left(\frac{n}{d} - n + m\right) = \frac{(d-5)n}{2} + \frac{2n}{d} + m.$$

因此，$\mathcal{P}$ 给出了公平相关聚类的一个 α-近似，其中

$$\begin{aligned}\alpha &= \left(\frac{(d-1)n}{2} + m\right) \Big/ \left(\frac{(d-5)n}{2} + \frac{2n}{d} + m\right) \\ &= \left(\frac{\left(d^2-d\right)n + 2dm}{2d}\right) \Big/ \left(\frac{\left(d^2-5d+4\right)n + 2dm}{2d}\right) = \frac{\left(d^2-d\right)n + 2dm}{\left(d^2-5d+4\right)n + 2dm}.\end{aligned}$$ ◀

注意，对于 $d \geqslant 4$，α 会随着 d 的变化而递减，并在 $d \to \infty$时收敛到 1。此外，对于$d = 5$，我们得到 $\alpha = \frac{20n+10m}{4n+10m} < 5$。因此，对于 $d \geqslant 5$，我们得到了森林上公平相关聚类的一个 5-近似。对于 $d = 4$，α 会随 $\frac{m}{n}$ 线性变化；而当 d 更小时，它不一定为正，甚至在 $\left(d^2-5d+4\right)n + 2dm = 0$时可能没有定义。这是因为，如果簇非常小，那么在森林中就存在代价几乎为零的解。如果 $d = 2$，也就是说，两种颜色的比例为 1 : 1，那么甚至存在代价为 0 的森林，即所有顶点的度都为 1，且每条边连接两个不同颜色的顶点。此时，割断每条边的解远劣于最优解。如果该因子变为负数或没有定义，这是因为我们用 $\frac{n}{d} - n + m$对最优聚类的簇间代价作了界定，而它可能为负；但簇间代价保证是非负的。

然而，对于树而言，如果簇很小，那么即使是最优解也必须切断一些边，因为簇之间总会存在边。因此，在这种情况下，对于所有可能的 d，我们都能得到良好的近似。请注意，定理 40 的证明实际上并不需要 $d \geqslant 4$，但对于 $d < 4$，近似因子并无实用价值或未定义。如果我们假设森林是一棵树，并代入 $m = n - 1$，情况就会有所不同。

▶ **推论 41**。 设 T 是一棵含有 n 个顶点的树，其 $k \geqslant 2$ 种颜色的比例为 $c_1 : c_2 : \ldots\ldots : c_k$ 和 $d = \sum_{i=1}^{k} c_i$。那么，可以在 T 上将公平相关聚类问题近似到 $\frac{(d^2+d)n-2d}{(d^2-3d+4)n-2d}$ 的比例，且计算时间为 $\mathrm{O}(n)$。

现在，近似因子仍随 d 增大而减小，并在 $d \to \infty$ 时收敛到 1。不过，它对所有 $d \geqslant 2$ 都为正且有定义。对于 $d = 2$，我们得到 $\frac{6n-4}{2n-4} < 3$。因此，我们可以对树上的公平相关聚类实现 3-近似。

尽管如此，我们关于森林的结果足以证明公平相关聚类属于 APX，甚至属于多项式时间近似方案。首先，对于 $d \geqslant 5$，我们针对森林上的公平相关聚类给出了 5-近似算法。如果 $d \leqslant 4$，则根据定理 23，可以在多项式时间内找到该森林上的最小成本公平聚类。因此，森林上的公平相关聚类属于 APX。接下来请回想，最小公平簇大小 d 越大，近似效果就越好。还请回想，对于定理 23，我们的动态规划在值 d 越小时运行时间越短。结合这些结果，我们得到了森林上公平相关聚类的多项式时间近似方案。这与一般图上的公平相关聚类形成对比：在那里，即使是非公平相关性聚类也是 APX-困难的 [16]，因此除非 P = NP，否则不存在多项式时间近似方案。

▶**定理 42.** 森林上的公平相关聚类存在一个多项式时间近似方案。此外，可以在时间$O(n^{\mathsf{poly}(1/\varepsilon)})$内计算出一个$(1+\varepsilon)$-近似公平聚类。

证明。 如果 $d \leqslant 4$，我们可根据定理 23 在多项式时间内找到一个最小成本公平聚类。否则，如果 $\frac{(d^2-d)n+2dm}{(d^2-5d+4)n+2dm} \leqslant 1+\varepsilon$，根据定理 40，只需返回任意一个公平聚类即可。否则，我们有 $d \geqslant 5$ 并且

$$1+\varepsilon < \frac{\left(d^2-d\right)n+2dm}{\left(d^2-5d+4\right)n+2dm} < \frac{\left(d^2-d\right)n}{\left(d^2-5d\right)n} = \frac{d-1}{d-5}.$$

由此可得，$d-5+d\varepsilon-5\varepsilon < d-1$，化简后为 $d < \frac{4}{\varepsilon}+5$。因此，根据定理 23，我们可在 $O(n^{f(\varepsilon)})$ 时间内找到一个最小成本公平聚类，其中 f 是与 n 无关的某个可计算函数。在所有情况下，我们都能找到一个公平聚类，其代价至多为最小相关聚类代价的 $1+\varepsilon$ 倍，且运行时间为 $O(n^{f(\varepsilon)})$，从而得到一个多项式时间近似方案。

要证明 f 实际上有一个关于 $1/\varepsilon$ 的多项式上界，只需考察第三种情况（否则 f 是常数）。界限 $d < \frac{4}{\varepsilon}+5$ 和 $d = \sum_{i=1}^{k} c_i$ 共同意味着，颜色数量 k 相对于 n 是常数。在此条件下，定理 23 中运行时间的指数是关于 d 的多项式，因此也是关于 $1/\varepsilon$ 的多项式。◀

参考文献

1 Saba Ahmadi、Sainyam Galhotra、Barna Saha 和 Roy Schwartz。公平相关聚类。*CoRR*，arXiv:2002.03508，2020年。ArXiv 预印本。网址：https://arxiv.org/abs/2002.03508。2 Sara Ahmadian、Alessandro Epasto、Ravi Kumar 和 Mohammad Mahdian。公平相关聚类。载于第*23*届人工智能与统计学会议（*AISTATS*）论文集，第4195–4205页，2020年。网址：https://proceedings.mlr.press/v108/ahmadian20a.html。3 Sara Ahmadian 和 Maryam Negahbani。公平相关聚类的改进近似算法。*CoRR*，abs/2206.05050，2022年。doi:10.48550/arXiv.2206.05050。4 Nir Ailon、Moses Charikar 和 Alantha Newman。聚合不一致信息：排序与聚类。《*ACM*学报》，55(5):23:1–23:27，2008年。doi:10.1145/1411509.1411513。5 Noga Alon。项链分割。《数学进展》，63(3):247–253，1987年。doi:10.1016/0001-8708(87)90055-7。6 Noga Alon 和 Douglas B. West。博苏克-乌拉姆定理与项链二分。《美国数学会会刊》，98(4):623–628，1986年。doi:10.2307/2045739。

7 Sayan Bandyapadhyay、Fedor V. Fomin 和 Kirill Simonov。关于度量空间和欧几里得空间中公平聚类的核心集及其应用。载于国际自动机、语言与编程研讨会第48届会议（ICALP）论文集，第23:1–23:15页，2021年。doi:10.4230/LIPIcs.ICALP.2021.23。8 Nikhil Bansal、Avrim Blum 和 Shuchi Chawla。相关聚类。《机器学习》，56(1–3):89–113，2004年。doi:10.1023/B:MACH.0000033116.57574.95。9 Lucas Bastos、Luiz Satoru Ochi、Fábio Protti、Anand Subramanian、Ivan César Martins 和 Rian Gabriel S. Pinheiro。高效的簇编辑算法。《组合优化杂志》，31(1):347–371，2016年。 doi:10.1007/s10878-014-9756-7。10 Amir Ben-Dor、Ron Shamir 和 Zohar Yakhini。基因表达模式聚类。《计算生物学杂志》，6(3–4):281–297，1999年。 doi:10.1089/106652799318274。11 Suman K. Bera、Deeparnab Chakrabarty、Nicolas J. Flores 和 Maryam Negahbani。用于聚类的公平算法。载于第33届神经信息处理系统大会（NeurIPS）论文集，第4954–4965页，2019年。12 Ioana Oriana Bercea、Martin Groß、Samir Khuller、Aounon Kumar、Clemens Rösner、Daniel R. Schmidt 和 Melanie Schmidt。关于实质公平聚类的代价。载于2019年组合优化问题近似会议和2019年计算中的随机化会议（APPROX/RANDOM）论文集，LIPIcs第145卷，第18:1–18:22页，2019年。13 Francesco Bonchi、David García-Soriano 和 Francesco Gullo。相关聚类。Morgan & Claypool 出版社，2022年。 doi:10.2200/S01163ED1V01Y202201DMK019。14 Paul Simon Bonsma、Thomas Epping 和 Winfried Hochstättler。单词喷漆车间问题受限实例的复杂度结果。《离散应用数学》，154(9):1335–1343，2006年。 doi:10.1016/j.dam.2005.05.033。15 Sebastian Böcker 和 Jan Baumbach。簇编辑。载于第9届欧洲可计算性会议（CiE）论文集，第33–44页，2013年。 doi:10.1007/978-3-642-39053-1_5。16 Moses Charikar、Venkatesan Guruswami 和 Anthony Wirth。利用定性信息进行聚类。《计算机与系统科学杂志》，71(3):360–383，2005年。 doi:10.1016/j.jcss.2004.10.012。17 Shuchi Chawla、Konstantin Makarychev、Tselil Schramm 和 Grigory Yaroslavtsev。在完全图和完全 k部图上进行相关聚类的近最优线性规划舍入算法。载于第47届计算理论研讨会（STOC）论文集，第219–228页，2015年。doi:10.1145/2746539.2746604。18 Li Chen、Rasmus Kyng、Yang P. Liu、Richard Peng、Maximilian Probst Gutenberg 和 Sushant Sachdeva。近线性时间的最大流与最小费用流。载于第63届计算机科学基础研讨会（FOCS）论文集，第612–623页，2022年。doi:10.1109/FOCS54457.2022.00064。19 Flavio Chierichetti、Ravi Kumar、Silvio Lattanzi 和 Sergei Vassilvitskii。通过公平子集实现公平聚类。载于第 31 届神经信息处理系统大会（NeurIPS）论文集，第5036–5044页，2017年。20 Vincent Cohen-Addad、Euiwoong Lee 和 Alantha Newman。结合 Sherali–Adams 的相关聚类。载于第63届计算机科学基础研讨会（FOCS）论文集，第651–661页。IEEE，2022年。 doi:10.1109/FOCS54457.2022.00068。21 Marek Cygan、Marcin Mucha、Karol Węgrzycki 和 Michał Włodarczyk。关于等价于（min,+）-卷积的问题。《ACM 算法汇刊》，15(1):14:1–14:25，2019年。doi:10.1145/3293465。22 Julien Darlay、Nadia Brauner 和 Julien Moncel。稠密与稀疏图划分。《离散应用数学》，160(16):2389–2396，2012年。 doi:10.1016/j.dam.2012.06.004。23 Michael Dinitz、Aravind Srinivasan、Leonidas Tsepenekas 和 Anil Vullikanti。通过图割问题实现公平灾害遏制。载于第25届人工智能与统计学会议（AISTATS）论文集，第6321–6333页，2022年。网址：https://proceedings.mlr.press/v151/dinitz22a.html。

54 公平相关聚类

24 Thomas Epping、Winfried Hochstättler 和 Peter Oertel。喷漆车间问题的复杂度结果。《离散应用数学》，136:2-3:217–226，2004。 doi:10.1016/S0166-218X(03)00442-6。25 Seyed A. Esmaeili、Brian Brubach、Leonidas Tsepenekas 和 John P. Dickerson。概率公平聚类。载于第34届神经信息处理系统大会（NeurIPS）论文集，第12743–12755页，2020。 doi:10.1145/2783258.2783311。26 Michael Feldman、Sorelle A. Friedler、John Moeller、Carlos Scheidegger 和 Suresh Venkata- subramanian。认证并消除差别影响。载于第21届ACM SIGKDD知识发现与数据挖掘大会（KDD）论文集，第259–268页，2015。 doi:10.1145/2783258.2783311。27 Andreas E. Feldmann 和 Luca Foschini。树的平衡划分及其应用。Algorithmica，71(2):354–376，2015。 doi:10.1007/s00453-013-9802-3。28 Zachary Friggstad 和 Ramin Mousavi。具有全局和局部保证的公平相关聚类。载于2021年算法与数据结构研讨会（WADS）论文集，第414–427页，2021。 doi:10.1007/978-3-030-83508-8_30。29 Michael R. Garey 和 David S. Johnson。《计算机与难解问题：NP完全性理论指南》。W. H. Freeman，1979。30 Jonggyu Jang 和 Hyun Jong Yang。 α受公平性最大化的能量受限小蜂窝网络用户关联。IEEE无线通信汇刊，21(9):7443–7459，2022。 doi:10.1109/TWC.2022.3158694。31 Suchi Kumari 和 Anurag Singh。时变数据通信网络中公平的端到端基于窗口的拥塞控制。国际通信系统杂志，32(11)，2019。 doi:10.1002/dac.3986。32 Dana Pessach 和 Erez Shmueli。机器学习中的公平性综述。《ACM计算调查》，55(3):51:1–51:44，2022。 doi:10.1145/3494672。33 Simon Régnier。关于自动分类问题的若干数学方面。数学与人文科学，82:31–44，1983。34 Melanie Schmidt、Chris Schwiegelshohn 和 Christian Sohler。公平核集与公平 k-均值的流式算法。载于第17届近似与在线算法研讨会（WAOA）论文集，第232–251页，2020。 doi:10.1007/978-3-030-39479-0_16。35 Roy Schwartz 和 Roded Zats。一般图中的公平相关聚类。载于2022年组合优化问题近似大会和2022年计算中的随机化大会（APPROX/RANDOM）论文集，第37:1–37:19页，2022。 doi:10.4230/LIPIcs.APPROX/RANDOM.2022.37。36 Xiao Xin。相关聚类问题的固定参数可处理算法。关键工程材料，474–476:924–927，2011。 doi:10.4028/www.scientific.net/KEM.474-476.924。37 Charles T. Zahn, Jr. 用等价关系近似对称关系。工业与应用数学学会杂志，12(4):840–847，1964。 doi:10.1137/0112071。38 Imtiaz Masud Ziko、Jing Yuan、Eric Granger 和 Ismail Ben Ayed。变分公平聚类。载于第35届 AAAI 人工智能大会（AAAI）论文集，第11202–11209页，2021。 doi:10.1609/aaai.v35i12.17336。