arXiv:2602.10948v1 [cs.CC] 11 Feb 2026

寻找无孤立顶点的最大公共顶点子图的参数化复杂性

Palash Dey 印度理工学院卡拉格浦尔分校，印度

Anubhav Dhar 马克斯·普朗克信息学研究所，德国

Ashlesha Hota 印度理工学院卡拉格浦尔分校，印度

Sudeshna Kolay 印度理工学院卡拉格浦尔分校，印度

Aritra Mitra 印度理工学院卡拉格浦尔分校，印度

摘要

本文研究最大公共顶点子图问题：给定两个输入图 G_1，G_2 和一个非负整数 h，是否存在一个公共子图 H，包含至少 h 个顶点，且其中没有孤立顶点 H。换言之，H 中的每个连通分量至少包含 2 个顶点。该问题自然源于图论，也是久经研究的最大公共子图问题 [22, 23, 26] 的其他变体之一，同时在计算社会选择领域也有应用 [18, 17, 16]。我们证明该问题是 NP-困难的，并给出一个以 h 为参数的 FPT 算法。接下来，我们研究该问题关于一些常见结构参数的复杂性，包括一个或两个输入图的顶点覆盖数、最大度、树深度、路径宽度和树宽。针对上述结构参数的单独参数化及其组合参数化，我们推导出该问题参数化结果的完整二分定理。这使我们得以深入了解该问题的计算复杂性理论及参数化图景。

2012年 ACM 学科分类

关键词和短语 最大公共子图、公共星打包、FPT、W难性、结构参数

1 引言

在算法和结构图论中，寻找给定两个图的最大公共子图这一问题已得到广泛研究。最著名的变体包括：（i）最大公共诱导子图，即要求最大化公共诱导子图的顶点数；（ii）最大公共边子图，即要求最大化公共子图的边数；（iii）最大公共诱导连通子图，即要求最大化公共连通诱导子图中的顶点数。该问题的一个自然变体是：对于两个输入图 G_1 和 G_2，寻找顶点数最多的公共子图。然而，这个问题有一个平凡解：由 $\min(|V(G_1)|, |V(G_2)|)$ 个孤立顶点构成的图。

显然，在所有公共子图中，这样一个平凡图的顶点数最多 .

本文研究一种自然的变体——最大公共顶点子图问题。该问题要求最大化两个输入图的公共子图中的顶点数，但要求公共子图不含孤立顶点。有趣的是，禁止出现孤立顶点使该问题变得相当复杂。本文证明，最大公共顶点子图问题是 NP-困难的，而且其参数化复杂性呈现出丰富的图景。我们研究了在多种参数选择下最大公共顶点子图问题的参数化复杂性：最大公共子图的大小、最大度、树宽、路径宽度、树深度以及顶点覆盖数。

除了本身是图论中一个有趣且自然的问题外，最大公共顶点子图还可应用于计算社会选择中的图式房屋分配问题。我们给出如下动机。设 $\mathcal{A}$ 为代理人集合，$\mathcal{H}$ 为房屋集合。设 $G_{\mathcal{A}}$ 为一个以 $\mathcal{A}$ 为顶点集的图，其中边表示代理人之间的某种社会关系（例如友谊、相识或合作关系等）。类似地，设 $G_{\mathcal{H}}$ 为一个以$\mathcal{H}$ 为顶点集的图，其中边表示某种“邻近性”（例如地理位置邻近、交通可达等）。每位代理人 $a \in \mathcal{A}$ 必须分配到一所唯一的房屋 $\phi(a) \in \mathcal{H}$。如果代理人 $a \in \mathcal{A}$ 的至少一位邻居 a （在 $G_{\mathcal{A}}$中）被分配到 $\phi(a)$的一所邻居房屋（在 $G_{\mathcal{H}}$中），则该代理人是满意的。最大邻居满意房屋分配问题要求找到一种分配方案，使满意的代理人数最多。尽管这是图式房屋分配中其他已有深入研究的变体的一种自然扩展 [18, 17, 16], ，但该问题与图 $G_{\mathcal{A}}$ 和 $G_{\mathcal{H}}$上的最大公共顶点子图问题完全相同。

我们正式表述最大公共顶点子图问题如下。

最大公共顶点子图

输入： 无向无权图 G_1、 G_2，非负整数 h。**问题：** 是否存在一个至少包含 h 个顶点的图 H，其中没有任何顶点是孤立的，并且 H 是 G_1 和 G_2 的子图?

我们进一步揭示了 H 和 G_1 之间最大公共子图的结构性质。由 H 的各个连通分量的生成树组成的生成森林 F，也是 G_1 和 G_2 的不含孤立顶点且顶点数最多的公共子图。因此，我们可以将范围限制为 G_1 和 G_2的公共森林子图。更进一步， H 的任意最小边覆盖 F' 也会是 G_1 和 G_2的不含孤立顶点的公共子图。回顾一下， H 的边覆盖是其边的一个子集 S，其中 S 的每个顶点 H都至少与一条 S 中的边关联。最小边覆盖是大小最小的边覆盖。已知最小边覆盖会诱导出一组非平凡星 [12, 28], ，其中非平凡星是至少有两个顶点且直径至多为 2 的树。将星森林定义为非平凡星的不相交并。由此，我们可以将搜索范围限制为仅包含作为公共子图出现的星森林。最大公共星森林子图问题（MaxComSubG）的正式表述如下。由于该问题与最大公共顶点子图问题相同，本文将始终聚焦于 MaxComSubG。

最大公共星森林子图（MaxComSubG）

输入： 无向无权图 G_1, G_2，非负整数 h。
问题： 是否存在一个星森林 H，其至少包含 h 个顶点，且其中没有任何一个是孤立的，使得 H 是 G_1 和 G_2 的子图?

1.1 相关工作

子图同构问题询问：是否存在从模式图 H 到宿主图 G 的单射映射ϕ：$V(H) \to V(G)$，使得 $\{u, v\} \in E(H) \implies \{\phi(u)\ \phi(v)\} \in E(G)$。该问题是 NP-困难的 [13]。Kann [22] 指出，最大公共诱导子图、最大公共边子图和最大公共诱导连通子图问题都推广了子图同构问题，因此也是 NP-困难的。本文进一步研究了这些问题的可近似性。研究表明，最大公共诱导子图的近似难度至少与最大团一样高。最大公共诱导连通子图问题属于 NPO PB完全。Kann 还通过 L归约关联了基于节点和基于边的表述，确立了基础性的近似性与困难性结果。Khanna等人 [23] 研究了最大公共子树问题，该问题要求在一组输入树中寻找一棵最大的、作为公共子图出现的树。他们证明，对于有界度树，可以达到 $O\left(^{n}\log\log_{2_n}{}^{n}\right)\log$ 的近似比。对于无界度树，在无标记情形下，已知存在 $O\left(^{n(\log}\log_{2_n}{}^{n^{2}}\right))\log$ 近似算法。当不同标签的数量为 $O(\log^{O(1)} n)$ 时，对于无界度的有标记树，同样的近似保证也成立。此外，Akutsu等人 [1] 研究了一个受限的无序情形，其中公共子树的最大出度有常数上界。Yamaguchi等人 [32] 进一步施加了结构限制，着重研究可处理情形，尤其是具有有界度和树宽且生成树数量为多项式的图。利用这些性质，研究者开发出一种多项式时间算法，用于计算一棵度有界的部分 k-树与一个生成树数量有多项式上界的连通图之间的最大公共连通诱导子图。

在结构方面，Kriege等人 [26] 证明，即使除一个顶点外的所有顶点的度都不超过三，计算两个双连通串并联图（部分2树）的最大公共连通子图仍然是NP-困难的。Hoffmann等人 [15]提出了 k-less子图同构，即在嵌入之前最多可以删除 k 个顶点。他们开发了弱化的过滤不变量，对基于度和路径的约束进行了推广，并将其实现为一种约束规划风格算法。

正如我们所展示的，我们的问题等价于寻找最大公共星森林。图中的星打包问题已得到广泛研究。Prieto 和 Sloper [27] 研究了将图 H 的k 个顶点不相交副本打包到 G中的问题，重点关注 H 为星的情形，$K_{1,\ s}$。他们针对一般的 s 给出了一个二次核，并针对 $s = 2$ 给出了一个线性核及一个$\mathcal{O}(2^{5.301k}k^{2.5} + n^3)$算法。Xi 和 Lin [30] 研究了无爪三次平面图上的最大 P_3 打包问题，并证明即使在这一结构限制下，该问题仍为 NP-困难。在后续工作中，[31], 他们证明了无爪三次图上的 3-星打包问题是 NP 困难的。他们利用结构性质和局部增广论证，设计了一种线性时间近似算法，其覆盖的顶点数至少为$(3n - 8)/4$。Huang 等人 [19]

研究了 k^{+}-星打包问题，并改进了近似保证：利用局部搜索框架，对 $k=2$达到了 $9/5$ ，对一般的 $k \geqslant 2$ 达到了 $(k+2)/2$ 。

1.2 我们的贡献

MaxComSubG 是一个 NP 难问题（由定理 17 得出）。本文给出了 MaxComSubG 的参数化复杂性全景。我们首先研究以公共星森林的规模为参数的问题。定理 12 表明，在这种情况下，MaxComSubG 存在一个 FPT 算法。

接下来，我们研究以输入图的各种结构参数为参数的 MaxComSubG。具体而言，我们考虑最小顶点覆盖规模（即顶点覆盖数）、树深度、路径宽度和树宽等参数。这类结构参数在实际应用中出现的图里通常有界，因此可用于突破其他难以处理的问题所面临的限制 [7, 24]。对于一个图G，$\mathrm{tw}(G) \leqslant \mathrm{pw}(G) \leqslant \mathrm{td}(G)-1 \leqslant \tau(G)$ [24] ，其中 G 的树宽为 $\mathrm{tw}(G)$，路径宽度为 $\mathrm{pw}(G)$，树深度为 $\mathrm{td}(G)$，顶点覆盖数为 $\tau(G)$。除了这些参数，我们还考虑图的最大度 $\Delta(G)$，并探索参数 $\Delta(G)$ 与结构参数 $\mathrm{tw}(G)$、$\mathrm{pw}(G)$、$\mathrm{td}(G)$ 和 $\tau(G)$ 的所有可能组合。

首先，当参数为 (i) 其中一个图的顶点覆盖数时，MaxComSubG 是 W[2]-困难的（定理 15）；(ii) 其中一个图的最大度时，MaxComSubG 是 para-NP-困难的（由定理 17 可得）。事实上，定理 17 表明，即使以两个输入图的最大度之和作为参数，MaxComSubG 仍是 para-NP-困难的。与此相反，我们在定理 20 中证明，以两个输入图的顶点覆盖数之和作为参数时，MaxComSubG 存在 FPT 算法。

这促使我们研究 MaxComSubG 关于其他结构参数的参数化复杂性。在定理 21 中，我们证明，以两个输入图的树深度作为参数时，MaxComSubG 是 para-NP-困难的，因此，以两个输入图的路径宽度或树宽作为参数时也是如此。对于分别以两个输入图的树深度或最大度作为参数时 MaxComSubG 的 para-NP-困难性，我们进一步证明，以两个输入图的树深度与最大度之和作为参数时，MaxComSubG 存在 FPT 算法（推论27）。为此，我们首先给出一个 FPT 算法，在以两个图的连通分量大小作为参数时求解 MaxComSubG（定理 26）。

接下来，我们考察两个输入图的路径宽度与最大度之和，并证明即使输入图是路径宽度至多为 4 的森林，MaxComSubG 仍然是 W[1]-困难的（定理29）。因此，对两个图的树宽和最大度进行参数化时，同样的结论也成立。由此可见，只有当至少一个图的最大度为常数时，才有望存在以路径宽度或树宽为参数的可处理算法。在定理 39 中，我们给出一种 FPT 算法：当参数为两个输入图的树宽，且其中一个图的最大度为常数时，该算法可以求解 MaxComSubG。因此，当两个输入图的路径宽度作为参数，且其中一个图的最大度为常数时，该结论同样成立。

这些结构性结果及其对 MaxComSubG 的影响列于表 1。请注意，所有关于算法和困难性的结果均适用于判定两个输入图是否存在一个大小恰为

h 、不含孤立顶点的公共子图这一问题。

顺带一提，定理 17 表明，即使两个输入图都是最大度有界的平面图，MaxComSubG 仍然是 NP-困难的。我们通过设计一个 EPTAS（定理 40）来补充这一结果。

界 *on both* 图	顶点覆盖 $\leqslant k$	树深度 $\leqslant d$	路径宽度 $\leqslant$ pw	树宽 $\leqslant$ tw	任意的图
任意的 max 度	$f(k)n^{\mathcal{O}(1)}$ 定理 20	NP-困难 对于 $d = 5$ 定理 21	NP-困难 对于路径宽度 $= 4$ 定理 21	NP-困难 对于树宽 $= 1$ 定理 21	NP-困难 定理 21
Max 度 Δ is a 参数	$f(k)n^{\mathcal{O}(1)}$ 定理 20	$f(d,\ \Delta)n^{\mathcal{O}(1)}$ 推论27	W[1]-难 wrt Δ, 针对 pw $= 4$ 定理29	W[1]-难 wrt Δ, 对于树宽 $= 1$ 定理29	NP-困难 对于 $\Delta = 3$ 定理 17
Max 度 Δ is a 常数	$f(k)n^{\mathcal{O}(1)}$ 定理 20	$f(d,\ \Delta)n^{\mathcal{O}(1)}$ 推论27	f(pw $\Delta)n^{\mathcal{O}(\Delta)}$ 定理 39	f(树宽 $\Delta)n^{\mathcal{O}(\Delta)}$ 定理 39	NP-困难 对于 $\Delta = 3$ 定理 17

表 1 不同参数下 MaxComSubG 的可处理性

2 预备知识

记号。我们用 $\mathbb{N}$ 表示自然数集合 {1, 2, . . .}。对于任意整数 ℓ，我们用 $[\ell]$ 表示集合 $\{1, \ldots, \ell\}$。本文中的所有图都是无向、无权且简单的。对于图 G，我们用 $V(G)$ 表示其顶点集合，用 $E(G)$ 表示其边集合。集合 $N_G(v) = \{u \mid \{u, v\} \in E(G)\}$ 表示图 G 中顶点 v 的开邻域。图 G 中顶点 v 的度为 $\deg(v) = |N_G(v)|$。如果图 G 中的每个顶点 $v \in V(G)$ 的度都满足 $\deg_G(v) = k$，则称图 G 为 k-正则图。对于顶点集合 $S \subseteq V$，我们用 $G[S]$ 表示图 G 中由 S 诱导的子图。对于图 G 和顶点 v，令 $G - v$ 表示图$G[V(G)\setminus\{v\}]$。对于非连通图 G，令 cc(G) 表示其连通分量集合 G。如果一个图可以在平面上画出且没有边交叉，则称其为平面图；如果一个图存在一种平面画法，使得所有顶点都位于外面的边界上，则称其为外平面图。如果一个图存在这样的平面嵌入：对于每个顶点，都存在一个至多包含 k 个面的交替序列，其中面和 k 嵌入的顶点交替出现，序列从无界面开始并以该顶点结束，且每对相邻的面和顶点彼此关联，则称该图为 k-外平面图。如果存在映射 ϕ：$V(H) \to V(G)$，使得对于所有 $\{u, v\} \in E(H)$，都有 $\{\phi(u)\ \phi(v)\} \in E(G)$，则称图 H 是图 G 的子图；我们称 ϕ 为 H 到 G 的嵌入。以下结论适用于 MaxComSubG。

▶ **观察 1.** G_1、G_2、h是一个 MaxComSubG 实例，其中 G_2 是一个具有h 个顶点的星森林，当且仅当 G_2 是 G_1的子图时，该实例为是实例。

参数化复杂性。 对于输入规模为 n 且关联参数 k的判定问题，参数化复杂性的目标是设计运行时间为 $f(k) \cdot n^{\mathcal{O}(1)}$的算法，其中 f 是仅依赖于 k [8]的可计算函数。

能够使用此类算法解决的问题称为固定参数可解（FPT）。运行时间为 $f(k)\cdot n^{\mathcal{O}(1)}$ 的算法称为 FPT 算法，相应的运行时间称为 FPT 运行时间 [7, 11]。形式化地，若存在算法 A、可计算函数 $f\colon \mathbb{N}\to\mathbb{N}$以及常数 $c\in\mathbb{N}$，使得对于每个输入 $(x,k)\in\Sigma^*\times\mathbb{N}$，算法 A都能在至多 $f(k)\cdot|(x,k)|^c$的时间内判定 $(x,k)\in L$，则称参数化问题$L\subseteq\Sigma^*\times\mathbb{N}$ 是固定参数可解 。所有此类问题构成的类记为 FPT。接下来，我们定义参数化归约。

▶ **定义 2**（参数化归约）。设 A、$B\subseteq\Sigma^*\times\mathbb{N}$ 是两个参数化问题。从 A 到 B 的参数化归约是一种算法：给定 A 的一个实例（x、k），输出 B 的一个实例（x'、k'），满足：

1.（x，k） 是 A 的是实例，当且仅当 （x'，k'） 是 B 的是实例，**2.** $k'\leqslant g(k)$，其中 g 是某个可计算函数；以及**3.** 该算法的运行时间为 $f(k)\cdot|x|^{\mathcal{O}(1)}$，其中 f 是某个可计算函数。

更多详细信息，请参阅 Cygan 等人 [7]。

结构参数。下面给出树分解、树宽、规范树分解、路径分解、路径宽度、树深度和顶点覆盖的定义。我们先定义树分解、树宽和规范树分解。

▶ 定义 *3*（树分解与树宽）。设 $G(T=(V(T),E(T)),X=\{X_t\}_{t\in V(T)})$ 是一个图。图 G 的树分解是一个二元组 $(T=(V(T),E(T)),X=\{X_t\}_{t\in V(T)})$，其中 T 是一棵树，每个节点 $t\in V(T)$ 都被赋予一个子集 $X_t\subseteq V(G)$，称为袋，并满足以下条件。

1. $\bigcup X_t=V(G)$,$_{t\in V(T)}$ *2.* 对于每条边 $\{x,y\}\in E(G)$，都存在一个 $t\in V(T)$，使得 $x,y\in X_t$，以及 *3.* 对于任意 $v\in V(G)$，由集合 $\{t\mid v\in X_t\}$ 在 T 中诱导出的子图是连通的。

树分解的宽度 是最大值$_{t\in V(T)}|X_t|-1$。 G 的树宽 是 G 的所有树分解中的最小宽度，记作 $\mathrm{tw}(G)$。

▶ **定义 4**（良好树分解）。 如果树分解（T,X） 满足以下条件，则称其为良好树分解： T 是一棵以某个节点 r 为根的树，其中 $X_r=\emptyset$； T 的每个节点至多有两个子节点，且每个节点属于以下类型之一：

▷ **引入节点**：只有一个子节点 t'的节点 t，其中 $X_t\supset X_{t'}$且 $|X_t|=|X_{t'}|+1$。▷ **遗忘节点**：只有一个子节点 t' 的节点 t，其中 $X_t\subset X_{t'}$ 且 $|X_t|=|X_{t'}|-1$。▷ **合并节点**：具有两个子节点 t_1 和 t_2 的节点 t，满足 $X_t=X_{t_1}=X_{t_2}$。▷ **叶节点**： T的叶节点 t，且 $X_t=\emptyset$。

宽度为树宽 tw 的树分解可以转换为具有相同树宽 tw 且包含$\mathcal{O}(\mathrm{tw}\cdot|V(G)|)$个节点 [7] 的良好树分解。关于 k-外平面图 [29]的树宽，已知如下结论。

▶ **命题 5.** k-外平面图的树宽至多为 $3k+1$。

路径宽度是一个与树宽类似的参数，但底层图为路径时例外。树深度则是另一种结构参数。其定义如下。

▶ **定义 6**（路径分解与路径宽度）。 图 G 的路径分解是由一系列 $P = (X_1, X_2, \ldots, X_r)$ 个袋组成的序列，其中 $X_i \subseteq V$ （G） 对每个 $i \in \{1, 2, \ldots, r\}$ 都成立，并满足以下条件：

1. $\bigcup X_i = V(G)$,$i \in [r]$2. $\forall \{u, v\} \in E(G)$, $\exists \ell \in \{1, 2, \ldots, r\}$，使得袋 X_ℓ 同时包含 u 和 v。 *3.* $\forall u \in V(G)$，若对于某个 $i \leqslant k$ 有 $u \in X_i \cap X_k$，则对于所有 j 都有 $u \in X_j$，其中 $i \leqslant j \leqslant k$.

路径分解$(X_1, X_2, \ldots, X_r)$ 的宽度为$\max_{i \in [r]} |X_i| - 1$。图 G 的路径宽度记为$\mathrm{pw}(G)$，它是 G 的路径分解可能具有的最小宽度。

▶**定义 7 (**树深度)。 G 的树深度记为$\mathrm{td}(G)$，定义如下：

$$\mathrm{td}(G) = \begin{cases} 0 & \text{if } G \text{ has no vertices,} \\ \max\limits_{H \in \mathrm{cc}(G)} \{\mathrm{td}(H)\} & \text{if } G \text{ is disconnected,} \\ \min\limits_{v \in V(G)} \{\mathrm{td}(G - v)\} + 1 & \text{if } G \text{ is connected.} \end{cases}$$

最后，我们回顾一下顶点覆盖的定义。

▶ **定义 8 (**顶点覆盖)。 设 $G = (V(G)\ E(G))$ 是一个图。若集合 $S \subseteq V$ 满足图 $G \backslash S$ 不含边，则称其为 G 的顶点覆盖 。

顶点覆盖数记为 $\tau(G)$，它是最小顶点覆盖的大小，即能够覆盖所有边的最小顶点子集。对于任意图 G，这些结构参数满足 $\mathrm{tw}(G) \leqslant \mathrm{pw}(G) \leqslant \mathrm{td}(G) - 1 \leqslant \tau(G)$ [24]。

近似算法。 如果算法 $\mathcal{A}$ 能够在多项式时间内最优地解决问题 $\mathcal{P}$ 的所有实例，则称该算法对问题 $\mathcal{P}$ 是高效的。然而，除非 P= NP，否则我们无法指望为属于 NP-困难的问题类设计高效算法。一个自然的出路是从这三个要求中任选两个。在近似算法领域，我们放宽了对最优性 [29]的要求。

▶**定义 9**。 优化问题的 α-近似算法是一个多项式时间算法，它对该问题的所有实例生成一个解，其值与最优解的值之比不超过 α 。

我们将 α 称为近似比。通常，对于最小化问题， $\alpha > 1$ ；对于最大化问题，$\alpha < 1$ [29]。

▶**定义10 (**EPTAS)。 多项式时间近似方案（*PTAS*）是一族算法 A_ε，其中每个 $\varepsilon > 0$ 都对应一个算法，使得 A_ε 是一个 $(1 + \varepsilon)$近似算法（适用于最小化问题）或一个 $(1 - \varepsilon)$ 近似算法（适用于最大化问题），并且对于每个常数 ε， A_ε 的运行时间关于输入规模是多项式的。

▶**定义 11** (EPTAS)。 高效多项式时间近似方案（*EPTAS*）是一族算法 A_ε，其中每个 $\varepsilon > 0$ 都对应一个算法，使得 A_ε 是一个 $(1+\varepsilon)$ 近似算法（适用于最小化问题）或一个 $(1-\varepsilon)$ 近似算法（适用于最大化问题），并且 A_ε 的运行时间为 $f(1/\varepsilon)\cdot n^{O(1)}$，其中 f 是一个仅依赖于 ε 的可计算函数，n 为输入规模。

3 将公共星森林的大小作为参数

正如我们将在定理 17 中看到的，MaxComSubG 是一个 NP 难问题。我们证明，以公共子图大小 h 为参数的 MaxComSubG 存在一个 FPT 算法。

▶ **定理 12.** 给定一个 MaxComSubG 实例(G_1, G_2, h)，存在一个以 h 为参数的 FPT算法。该算法的运行时间为 $2^{\mathcal{O}(h)}\cdot n$。

证明。 令 H 为 G_1 和 G_2的最大公共星森林子图。我们需要判定 $|V(H)| \geqslant h$是否成立。首先证明以下结果。

▷ 断言 13. $|V(H)| \geqslant h$ 当且仅当以下条件中至少有一个成立。

1. G_1 和 G_2 都存在一个由 $\lceil h/2\rceil$ 条边组成的匹配。**2.** G_1 和 G_2 都存在一个大小恰好为 h的公共星森林子图。

证明。 显然，如果 G_1 和 G_2 都存在大小为 $\lceil h/2\rceil$的匹配（它也是一个星森林），或者它们存在一个大小恰好为 h的公共星森林，那么最大公共星森林的大小至少为 h，即 $|V(H)| \geqslant h$。

现在，我们证明反方向。设 H 是 G_1 和G_2 的一个公共星森林子图，且 $|V(H)| \geqslant h$；并设 G_1 和 G_2 中任一图的最大匹配大小都小于 $\lceil h/2\rceil$。因此，$\lceil h/2\rceil$ 严格大于H的最大匹配大小。于是，H至多有 $\lceil h/2\rceil - 1$ 个星。设 H中的星为 $S_1, \ldots, S_t$，其中 $t \leqslant \lceil h/2\rceil - 1$。对于每个 $i \in [t]$，设 S_i的中心为 c_i，叶节点为 $v_i^{(0)}, v_i^{(1)}, \ldots, v^{(y_i)}{}_i$ ，其中$y_i \geqslant 0$。注意，由 $Q = \{c_1, c_2, \ldots, c_t\}$ 和 $\cup\{v_1^{(0)}, v_2^{(0)}, \ldots, v_t^{(0)}\}$ 所诱导的图在 H中构成一个大小为 t 的匹配，包含 $2t \leqslant 2\lceil h/2\rceil - 2 < h$ 个节点。现在，我们取出 Q 中的所有顶点，并从 $V(H)\setminus Q$ 中取出 $h-2t$ 个顶点，从而得到 $V(H)$ 的一个子集 R，其大小恰好为h。 $H[R]$ 是 G_1 和 G_2 的一个子图，大小为 h。此外，$H[R]$ 是一个星森林，因为它不含孤立顶点；这是因为星 $S_1, \ldots, S_t$的中心，即 $c_1, \ldots, c_t$ 都包含在 R中。◁

令 $S(h)$表示由 h 个顶点构成、不含任何孤立顶点且在同构意义下唯一的所有星森林的集合。

▷ 断言 14。$|S(h)| = 2^{\mathcal{O}(\sqrt{h})}$，且可以在时间 $2^{\mathcal{O}(\sqrt{h})}$内枚举。

证明。整数 $t \geqslant 0$ 的一个划分是由 $[t]$组成的多重集，且这些元素之和为 t。令 $P(h)$ 为 h 的划分集合。我们通过构造一个单射 $\phi: S(h) \to P(h)$，证明 $|S(h)| \leqslant |P(h)|$。设 $H \in S(h)$ 是一个含有 h 个顶点且没有孤立顶点的星森林。令 h 的连通分量大小为 c_1、c_2、……，其中 $c_1 \geqslant c_2 \geqslant \ldots$。注意到 $\sum_i c_i = h$，因此 $\{c_1, c_2, \ldots\ldots\} \in P(h)$。我们定义 ϕ，使得 $H \mapsto \{c_1, c_2, \ldots\ldots\}$。现在，令 $\phi(H_1) = \phi(H_2)$，其中 $H_1, H_2 \in S(h)$。因此，

H_1 和 H_2 的连通分量大小相同。由于 H_1 和 H_2 的所有连通分量都是星，H_1 和 H_2 同构。因此，$H_1 = H_2$。

这证明了 $|S(h)| \leqslant |P(h)|$。现在，由于已知 $|P(h)| \leqslant \exp\left(\pi\sqrt{2h/3 + o(h)}\right) = 2^{\mathcal{O}(\sqrt{h})}$ [3]，我们得到 $|S(h)| = 2^{\mathcal{O}(\sqrt{h})}$。可以使用简单的递归算法在 $\mathcal{O}(h^2 \cdot |P(h)|) = 2^{\mathcal{O}(\sqrt{h})}$ 时间内枚举划分（见附录中的引理 41）。由于 $S(h)$ 包含 $P(h)$ 中不包含 1 的所有划分，因此可以在相同时间内枚举 $S(h)$。◁

现在我们给出以 h 为参数的 MaxComSubG 算法。令 M 为由 $\lceil h/2 \rceil$ 个边的副本组成的图（即大小为 $\lceil h/2 \rceil$ 的匹配）。根据断言 13，当且仅当 $S(h)$ 中存在 G_1 和 G_2 的公共星森林子图 H 时，该实例才是是实例$\{M\}$。

该算法枚举 $S(h)$（断言 14），并且对于所有的 $H' \in S(h) \cup \{M\}$，检查 H' 是否为 G_1 的子图以及 G_2 的子图。由于 H' 是森林，Alon、Yuster 和 Zwick [2] 表明，这可以在 $2^{\mathcal{O}(h)} \cdot n$ 时间内完成。对于某个 $H' \in S(h) \cup \{M\}$，若 H' 是 G_1 和 G_2 的公共星森林子图，则该算法输出 yes；否则输出 no。

该算法的总运行时间为 $2^{\mathcal{O}(\sqrt{h})} \cdot 2^{\mathcal{O}(h)} \cdot n = 2^{\mathcal{O}(h)} \cdot n$。◀

4 输入图的结构参数

本节研究以 G_1 和 G_2 的结构参数为参数的 MaxComSubG。具体而言，我们考察顶点覆盖数、树深度、路径宽度和树宽这几个结构图参数与最大度相结合的情形。回顾一下，若图 G 的树宽为 tw(G)、路径宽度为 pw(G)、树深度为 td(G)，且顶点覆盖数为 $\tau(G)$，则 tw(G) $\leqslant$ pw(G) $\leqslant$ td(G) $- 1 \leqslant \tau(G)$ [24]。

4.1 将顶点覆盖数或最大度作为参数

我们首先证明，当参数取其中一个图的最小顶点覆盖的大小时，MaxComSubG 是 W[2]-困难的。

▶**定理 15.** MaxComSubG 在以两个图的顶点覆盖数的较小值为参数时是 W[2]-困难的。

证明。支配集参数化问题的输入是一个图 G 和一个参数 k，问题是判断 G 是否存在一个大小至多为 k的支配集。以 k 为参数时，支配集是 W[2]-困难的 [7]。我们给出从支配集到 MaxComSubG 的参数化归约。设 (G, k) 是支配集的一个实例，其中 $|V(G)| = n$。不失一般性，我们假设 G 没有孤立顶点。我们的归约构造一个 MaxComSubG 实例 (G_1, G_2, h)。令 $G_1 = G$，$h = n$，并令 G_2 为一个由 k 个星形图组成的不相交集合，每个星形图有 $n - 1$ 个叶节点。G_2 中的第 i 个星形图有一个“中心”c_i，与“叶节点” $v_i^{(1)}$、$v_i^{(2)}$、……、$v_i^{(n-1)}$ 相连。G_2的最小顶点覆盖是所有星形图的中心组成的集合，因此其大小为 k。

▷ 断言 16. (G, k) 是一个支配集是实例，当且仅当 (G_1, G_2, h) 是一个 MaxComSubG 是实例。

证明. 正向。假设 G_1 有一个大小至多为 k 的支配集。令$D = \{u_1, \ldots, u_q\} \subseteq V$为 G的一个最小支配集，其中包含 $q \leqslant k$。因此，存在一个匹配 M，它覆盖 D，由一组边组成，每条边都分别关联于D 中的一个顶点和 D 之外的一个顶点（根据引理 42）。令顶点 u_i 与 $w_i \in V$（G）$\setminus D$ 匹配，其中$i \in [q]$ 属于 M。令 $W = \{w_1, \ldots, w_q\}$。

令 $v \in V(G) \setminus (D \cup W)$。令 b_v 为 $N_G(v) \cap D$ 中任意一个顶点。对于 $u_i \in D$，我们定义 $B_i = \{w_i\} \cup \{v \in V(G) \setminus (D \cup W) \mid b_v = u_i\}$。注意，$1 \leqslant |B_i| \leqslant n - 1$ 和 $B_i \subseteq N_G(u_i)$。

因此，G_1 包含一个由 q 个星组成的生成子图，这些星总计有 n 个节点：第 i 个 u_i 和叶节点 B_i。然而，由于 G_2 含有由 $k \geqslant q$ 个星组成的集合，且这些星有 $n - 1 \geqslant |B_i|$ 个叶节点，因此这样的星集合也作为子图出现在 G_2 中。这证明 (G_1, G_2, h) 是 MaxComSubG 的一个是实例。

反向。现在，令 (G_1, G_2, h) 为 MaxComSubG 的一个是实例。由于$h = n = |V(G_1)|$，这意味着 H 是 G_1 的一个生成子图，且由若干星组成。H 中这些星的中心共同构成一个支配集。令 r 为 H 中星的数量。由于 H 也是 G_2 的子图，因此 $r \leqslant k$（因为 G_2 由 k 个星组成）。所以，(G, k) 是支配集的一个是实例。◁

因此，当参数是其中一个图的最小顶点覆盖的大小时，MaxComSubG 是 W[2]-困难的。◀

接下来，我们研究 MaxComSubG 关于图的最大度的可处理性。我们证明，即使两个输入图都是最大度至多为 3 的平面图，MaxComSubG 仍然是 NP-困难的。

▶**定理 17**。MaxComSubG 即使两个输入图都是平面图且最大度为 3，仍然是 NP-困难的。

在陈述定理 17 的证明之前，我们先陈述 P_3-Factor 问题。P_3 是一个包含 3 个顶点的路径图。

P_3-因子
输入： 无序无权图 G**问题：** G 是否包含一个顶点不相交子图的集合，使其涵盖 G 中的所有节点，并且其中每个子图都同构于 P_3。

Xi 和 Lin [30] 证明了，即使当 G 为无爪的 3-正规 r
平面图时，{v2}-因子问题仍然是 NP-困难的。我们利用这一结果证明定理 17。

证明。我们将 3-正则平面图上的 P_3-Factor 归约到 MaxComSubG。设 G 是 P_3-Factor 的一个实例，它是一个 3-正则平面图。我们可以假设 $n = |V(G)| = 3r$，其中 r 是某个整数（否则 G 是 P_3-Factor 的否实例）。我们按如下方式构造 MaxComSubG 的一个实例 (G_1, G_2, h)。令 $G_1 = G$，令 G_2 为由 $n/3$ 个 P_3 的副本构成的不相交集合，并将目标值设为 $h = n$。注意，G_1 和 G_2 都是平面图，其中 G_1 的最大度为 3，而 G_2 的最大度为 2。

如果 G 是 P_3-Factor 的一个是实例，那么 G_2 是 $G_1 = G$ 的子图，因此 (G_1, G_2, h) 是 MaxComSubG 的一个是实例。反过来，如果 G_1 和 G_2 有一个公共生成星森林子图，那么它必定与 G_2 同构，因为 G_2 本身就是一个星森林。这意味着 $G_1 = G$ 含有 G_2 作为其子图，因此 G 是 P_3-Factor 的一个是实例。◀

▶ 注释 18. 请注意，定理 17 表明，当以其中一个输入图的最大度作为参数时，MaxComSubG 是 para-NP-困难的。

▶ 注释 19. 我们通过设计一个 EPTAS，补充了两个输入图均为平面图时的 NP 困难性结果。对于每个 $0 < \varepsilon < 1$，在最大度不超过 Δ 的平面图上，存在一个用于 MaxComSubG 的 $(1-\varepsilon)$ 近似算法，其运行时间为 $(1/\varepsilon + \Delta)^{\mathcal{O}(1/\varepsilon)} \cdot n^{\Delta+1} \cdot \log n$。详细证明见定理 40。

由于以其中一个图的顶点覆盖数作为参数时，MaxComSubG 是 W[2]-困难的；而以两个输入图的最大度之和作为参数时，它是 para-NP-困难的，因此我们讨论以两个图的顶点覆盖数之和作为参数的情形。我们发现，以 G_1 和 G_2的顶点覆盖数之和作为参数时，MaxComSubG 存在一个 FPT 算法。

▶**定理 20.** 存在一种FPT 算法，用于确定两个输入图 G_1 和 G_2的最大公共星森林子图的大小，参数 k 满足 G_1 和G_2 的顶点覆盖大小均为 $\leqslant k$。该算法的运行时间为 $2^{2^{\mathcal{O}(k)}} \cdot n^{\mathcal{O}(1)}$。

证明。 设 $V_1 = V(G_1)$ 和 $V_2 = V(G_2)$。设 H 是 G_1 和 G_2的一个最大公共星森林子图。该算法分两个阶段运行： 预处理阶段 和整数线性规划阶段。预处理阶段会生成多个整数线性规划实例，每个实例都在整数线性规划阶段求解，最后输出所有实例中的最优解。

预处理阶段。 设 C_1 是 G_1的一个最小顶点覆盖，且 $|C_1| = a \leqslant k$。设$I_1 = V_1 \setminus C_1$。注意，I_1 是 G_1的一个最大独立集。现在，我们创建 2^a 个孪生类，将 I_1划分为这些类，每个 $X \subseteq C_1$对应一个类，具体如下：定义孪生类 $B_X^{(1)} = \{v \in I_1 \mid N_{G_1}(v) = X\}$。设 H_1 是 G_1 的一个子图，且与 H同构。 H_1 中的星数至多为 a，因为每个星都必须包含顶点覆盖 C_1 中的至少一个顶点。我们将这些星分为两类：

▷ I 型星：I 型星的中心是来自 C_1 的一个顶点。▷ II 型星：II 型星的中心是来自 I_1 的一个顶点。

注意，II 型星的每个叶节点都属于 C_1。

我们“猜测”I 型星和 II 型星的数量。设有 p 个 I 型星和q个 II 型星，其中 $p + q \leqslant a$。令 $T_{(1)1}$， $T_2^{(1)}$， ...， $T_{(1)p}$为这些 I 型星，且$L_1^{(1)}$， $L_2^{(1)}$， ...， $L_q^{(1)}$为这些 II 型星。对于 p 和 q，共有 $\mathcal{O}(k^2)$ 种猜测。

现在我们“猜测”I 型星的中心。令 c_1， c_2， ...， $c_p \in C_1$ 分别为$T_{(1)1}$， $T_2^{(1)}$， ...， $T_{(1)p}$的中心。对于 c_1， c_2， ...， $c_p \in C_1$，共有 $\mathcal{O}(a^p) = \mathcal{O}(k^k)$ 种猜测。变量 α_i 表示 $|V(T_i^{(1)})|$ 的值，并且将是我们设计的整数线性规划中的一个变量。

接下来，我们完全“猜测”II 型星。注意，对于任意$X \subseteq C_1$， $B_X^{(1)}$ 中的两个顶点具有相同的邻域，因此彼此不可区分。对于 $L_i^{(1)}$ 、 $i \in [q]$，令 $c_{p+i} \in I_1$ 为其中心。我们猜测 $X \subseteq C_1$，使得 $c_{p+i} \in B_X^{(1)}$。

注意，由于 $B_X^{(1)}$ 中的所有顶点彼此不可区分，我们可以任意选取一个顶点作为 c_{p+i}。接下来，我们猜测 $L_i^{(1)}$ 的确切叶节点集合，该集合是 X的一个子集。对于每个 $i \in q$，有 2^a 种 X 的选择；对于每个 X，有 $2^{|X|}$ 种选择来确定 $L_i^{(1)}$ 的确切叶节点集合。因此，总共至多有 $\sum_{j\in[a]} \binom{a}{|X|} 2^{|X|} \leqslant 3^k$ 种猜测。令 $\alpha_{p+i} \in$ {2、3、.....、$a+1$} 是 $|V(L_i^{(1)})|$ 的值；这将是我们所设计的整数线性规划中的一个常数。此外，对于每个 $X \subseteq C_1$，令 $a_X = |B_X^{(1)} \setminus \{c_{p+i} \mid i \in [q]\}|$ 为 $B_X^{(1)}$ 中对于 $i \in [q]$ 而言不是 $L_i^{(1)}$ 的中心的节点数；这也将是我们所设计的整数线性规划中的一个常数。

对于 C_1 中其余每个既未被猜测为 $T_i^{(1)}$ 的中心、也未被猜测为 $L_{i'}^{(1)}$ 的叶节点的顶点 v，我们猜测它是否是某个 $T_i^{(1)}$ 的叶节点，或者是否不属于 H_1。对于每个 v，有（$p+1$）种选择，因此总共至多有（$p+1$）$^a = \mathcal{O}$（（$k+1$）^{k}）种猜测。完成这一步猜测后，C_1 中的每个顶点要么被唯一地猜测为 $T_{(1)\,i}$ 的中心、$T_i^{(1)}$ 的叶节点或 $L_i^{(1)}$ 的叶节点，要么不属于 H_1。令 $\beta_i = |T_{(1)\,i} \cap C_1|$，这将是我们所设计的整数线性规划中的一个常数。

对于每个 $X \subseteq C_1$ 和 $i \in [p]$，我们用变量 $x_{i,\,X}$ 表示其值为 $|T_i^{(1)} \cap B_X^{(1)}|$；这将是我们子句中的一个变量。我们还会进行合理性检查，如果这些猜测彼此不一致，就舍弃这组猜测。

现在，我们对 G_2 进行类似的“猜测”。令 H_2 为 G_2 中与 H同构的子图。令 G_2 的一个最小顶点覆盖为 C_2，其中 $|C_2| = b \leqslant k$；$I_2 = V_2 \setminus C_2$ 是一个最大独立集。这给出了孪生类 $B_Y^{(2)} = \{v \in I_2 \mid N_{G_2}(v) = Y\}$，对所有 $Y \subseteq C_2$均成立。设有 r 个 I 型星 $T_i^{(2)}$，以及 s 个 II 型星 $L_i^{(2)}$。我们确保 $r + s = p + q$。我们猜测 d_1、d_2、.....、$d_r \in C_2$ 分别是 $T(2)\,1$、$T_2^{(2)}$、.....、$T(2)_r$的中心。变量 γ_i 表示 $|V(T_i^{(2)})|$的值。对于 $L_i^{(2)}$、$i \in [s]$，令$d_{r+i} \in I_2$ 为其中心。我们猜测 $Y \subseteq C_2$，使得 $d_{r+i} \in B_Y^{(2)}$。我们猜测 $L_i^{(2)}$ 在 C_2中的确切叶节点。令 $\gamma_{r+i} \in \{2, 3, \ldots, b+1\}$ 为 $|V(L_i^{(2)})|$的值；对于每个 $Y \subseteq C_2$，令 $b_Y = |B_Y^{(2)} \setminus \{d_{r+i} \mid i \in [s]\}|$。对于 C_2中所有剩余的顶点，我们猜测它是否属于某个 $i \in [r]$ 的$T_i^{(2)}$，或者不在 H_2中。令 $\delta_i = |T(2)_{\,i} \cap C_2|$。对于每个 $Y \subseteq C_2$ 和 $i \in [r]$，我们用变量 $y_{i,Y}$表示其值为 $|T_i^{(2)} \cap B_Y^{(2)}|$。最后，我们对猜测的一致性进行合理性检查。

由于 H_1 和 H_2 彼此同构，因此必然存在一个置换 $\pi : [p+q] \to [r+s]$，使得 $\alpha_i = \gamma_{\pi(i)}$ 对每个 $i \in [p+q]$ 都成立。对于 π，共有 $k!$ 种猜测。

综合起来，共有 $\mathcal{O}\left(\left(k^2 \cdot k^k \cdot 3^k \cdot (k+1)^k\right)^2 \cdot k!\right) = k^{\mathcal{O}(k)}$ 种猜测。对于每一种猜测，我们构造一个整数线性规划并求解，以确定是否存在相应的子图 H_1 和 H_2，并希望最大化其总大小。

整数线性规划阶段。我们固定一个来自预处理阶段的猜测。我们的目标是最大化 $|V(H_1)| = \sum \alpha_i$（等价于 H_2）。由于 $\alpha_{p+1}, \alpha_{p+2}, \ldots, \alpha_{p+q}$ 是常数，我们将目标函数设为最大化 $\sum_{i\in[p]} \alpha_i$。

现在我们来研究确保 H_1 和 H_2 分别是 G_1和 G_2 的星形子图的约束条件。回想一下，对于每个 $X \subseteq C_1$、$x_{i,\,X}$ 是 $|T_i^{(1)} \cap B_X^{(1)}|$的大小。对于 i的两个取值，集合 $T_i^{(1)} \cap B_X^{(1)}$ 两两不相交，且不包含 $L_{i'}^{(1)}$、$i' \in [q]$的任何中心。因此，对所有 i 求和所得的 $x_{i,\,X}$ 不得超过 a_X。类似地，对于每个

$Y \subseteq C_2$，对所有 i 求和所得的 $y_{i,\ Y}$ 不得超过 b_Y。

现在，注意，对于 $i \in [p]$，对于任意 $X \subseteq C_1 \setminus \{c_i\}$，$T_i^{(1)}$ 不包含$B_X^{(1)}$ 中的任何节点；因此，$x_{i,\ X} = 0$。类似地，对于 $i \in [r]$，对于任意 $Y \subseteq C_2 \setminus \{d_i\}$，$T_i^{(2)}$ 不包含 $B_Y^{(2)}$ 中的任何节点；因此，$y_{i,\ Y} = 0$

接下来，我们计算每个 $i \in [p]$对应的 $T_i^{(1)}$ 大小。它在 C_1 中有 β_i 个节点，并且在 I_1 中有$\sum$ $x_{i,\ X}$ 个节点。因此，这两项之和必须等于 α_i.$X \subseteq C_1$类似地，对于每个 $i \in [r]$，δ_i 与$\sum$ $y_{i,\ Y}$之和必须等于 γ_i.$Y \subseteq C_2$

最后，根据 π 的定义，对于每个 $i \in [p+q]$，我们必须有 $\alpha_i = \gamma_{\pi(i)}$。

每个整数线性规划都有 $k_1 = a \cdot 2^a + b \cdot 2^b + p + r = 2^{\mathcal{O}(k)}$ 个变量，因此每个问题都能在 $k^{\mathcal{O}(k_1)}$ $_1 \cdot n^{\mathcal{O}(1)} = 2^{2^{\mathcal{O}(k)}} \cdot n^{\mathcal{O}(1)}$ 时间内求解。由于共有 $k^{\mathcal{O}(k)}$ 个，总运行时间为$k^{\mathcal{O}(k)} \cdot 2^{2^{\mathcal{O}(k)}} \cdot n^{\mathcal{O}(1)} = 2^{2^{\mathcal{O}(k)}} \cdot n^{\mathcal{O}(1)}$。该算法的输出是 $\sum_{i \in [}$ 的最大可能值

α_i 在所有可行猜测中。为完整起见，我们在下文正式给出该整数线性规划。]

$p+q$

最大化 $\sum_{i \in [p]} \alpha_i$

约束条件：

$$\sum_{i \in [p]} x_{i,X} \leqslant a_X \qquad \forall X \subseteq C_1$$

$$\sum_{i \in [r]} y_{i,Y} \leqslant b_Y \qquad \forall Y \subseteq C_2$$

$$x_{i,X} = 0 \qquad \forall i \in [p], X \subseteq C_1 \setminus \{c_i\}$$

$$y_{i,Y} = 0 \qquad \forall i \in [r], X \subseteq C_2 \setminus \{d_i\}$$

$$\alpha_i = \beta_i + \sum_{X \subseteq C_1} x_{i,X} \qquad \forall i \in [p]$$

$$\gamma_i = \delta_i + \sum_{Y \subseteq C_2} y_{i,Y} \qquad \forall i \in [r]$$

$$\alpha_i = \gamma_{\pi(i)} \qquad \forall i \in [p+q]$$

◀

4.2 其他结构参数

首先，由于图的树深度、路径宽度和树宽均不超过最小顶点覆盖的大小，因此当以其中一个输入图的上述参数作为参数时，MaxComSubG 仍然是 W[2]-困难的（定理 15）。另一方面，由于以两个输入图的顶点覆盖大小之和作为参数时，MaxComSubG 存在 FPT 算法，因此我们研究该问题在以两个输入图的 (i) 树深度、(ii) 路径宽度或 (iii) 树宽作为参数时的参数化复杂性。我们证明，对于这些参数，MaxComSubG 是 para-NP-困难的。

▶**定理 21.** MaxComSubG 即使在两个图都是树深度至多为5的不相交树集合时，仍然是 NP-困难问题。

14 求最大公共顶点子图的参数化复杂性

我们先陈述以下强NP困难问题： k-路划分。

k路划分

输入： 一组 n 正整数 a_1、.. .、 a_n，整数 k 和 C，且均以一元形式给出。**问题：** </style}是否可以将划分为个子集、.. .、 B_k，使得对所有$j \in [k]$， $\sum_{i \in B_j} a_i = C$。

k路划分问题不仅是NP-困难的，而且当 k 是参数 [9, 14, 20]时，也是W[1]-困难的。注意，即使整数a_1、. . .、 a_n 至少为 $2k + 10$，该问题仍然是强NP困难的，因为我们只需将 a_1、. . .、 a_n 和 C 乘以 $2k + 10$即可。

证明。 我们从 k路划分问题进行归约。设 a_1、.. .、 a_n、 k 和 C 是 k路划分问题的输入，其中 a_i 是一个值至少为$2k + 10$的偶正整数。不失一般性，我们还假设 $a_1 \geqslant a_2 \geqslant \ldots \geqslant a_n$。

归约后实例的构造。

在我们的归约中，我们构造一个 MaxComSubG 实例，其两个图为 G_1 和 G_2。G_1 应由一组 n 棵树组成，每棵树的深度恰好为 5，而 G_2 应包含多个星。 G_1 和 G_2 应具有完全相同的节点数 n'，且目标值应等于该数目 n'。换句话说，该判定问题要判断 G_1和 G_2 是否具有同构生成星森林。

令 $M = \sum_{i \in [n]} a_i$ 并令 $D = M + 20$。注意， $kC = M$ （否则这就是一个平凡的否实例）。现在我们定义 G_1的构造。对于每个 $i \in [n]$，将树 T_i 加入G_1，定义如下（图 1）。

▷ 构造一个节点 r_i。▷构造 $D - 1$ 个节点$h_{i,}1, h_{i,}2, .. . , h_{i,D-1}$，每个节点都与 r_i相邻。▷构造 k个节点$s_{i,}1, .. . , s_{i,k}$，每个节点都通过一条边与 r_i 相连。▷ 对于每个 $j \in [k]$，进行如下构造。构造 a_i 个节点 $t_i^{(1)}{}_{,j}, t_i^{(2)}{}_{,j}, \ldots, t^{(a_i)}{}_{i,j}$，每个节点都通过一条边与 $s_{i,j}$ 相连。构造 a_i 个节点 $u_i^{(1)}{}_{,j}, u_i^{(2)}{}_{,j}, .. . , u^{(a_i)}{}_{i,j}$，并且对于每个 $\ell \in [a_i]$，用一条边连接 $u_i^{(\ell)}{}_{,j}$和 $t_i^{(\ell)}{}_{,j}$。对于每个 $\ell \in [a_i]$，

-
  - 构造 $2j + 4$ 个节点 $v_{(\ell,\ 1)\,i,\ j}$， $v_{(\ell,\ 2)\,i,\ j}$， ...， $v_{(\ell,\ 2j+4)\,i,\ j}$，它们都与 $u_{(\ell)\,i,\ j}$相邻。

G_1 就是树 T_1，...， T_n 的集合。注意，以 r_i 为根时， T_i 的树深度为 5。此外， T_i 包含的节点数为 $1 + (D - 1) + k + k \cdot a_i + k \cdot a_i + a_i \cdot \sum (2j + 4) = (D + k) + (k^2 + 7k) \cdot a_i$。因此， G_1 总共包含 $\sum_{j \in [k]} n' = n(D + k) + M(k^2 + 7k)$ 个节点，这是该实例输入规模的多项式。

现在我们定义 G_2的构造。它将包含以下星形图（图 2）。

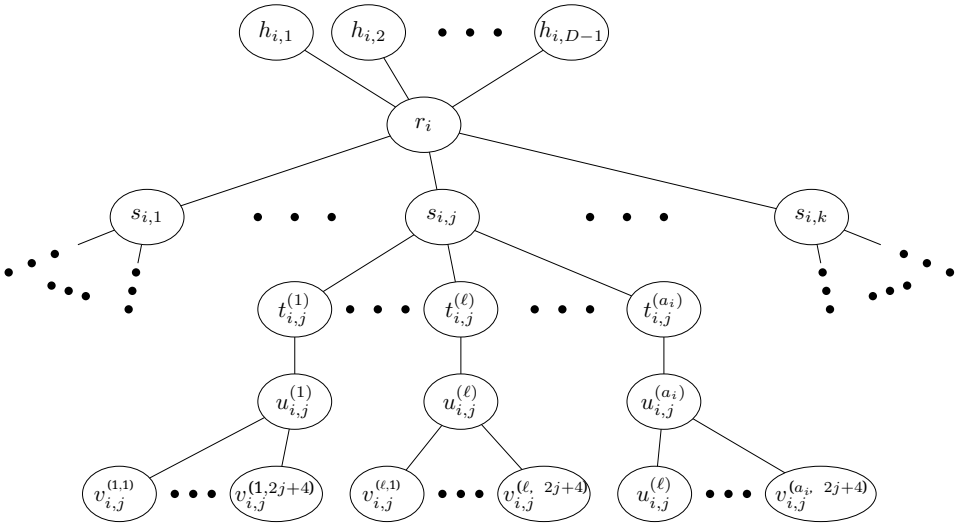


图1 T_i的构造。

▷ 对于每个 $i \in [n]$，构造一个中心为 $\alpha_{i,\ 0}$ 且有 D 个叶节点 $\alpha_{i,\ 1}$，...，$\alpha_{i,\ D}$ 的星形图，每个叶节点都通过一条边连接到$\alpha_{i,\ 0}$；将此星形图称为 P_i。
▷ 对于每个 $i \in [n]$ 和每个 $j \in [k-1]$，构造一个以 β_i 为中心的星，$_{j,\ 0}$ 并带有 a_i 个叶节点$\beta_{i,\ j,\ 1}$，...，$\beta_{i,\ j,\ a_i}$，每个叶节点都通过一条边连接到 $\beta_{i,\ j,\ 0}$；将此星称为 $Q_{i,\ j}$。
▷ 对于每个 $j \in [k]$ 和每个 $\ell \in [C]$，构造一个中心为 $\gamma_{j,\ \ell,\ 0}$ 且有 $2j+5$ 个叶节点$\gamma_{j,\ \ell,\ 1}$、...、$\gamma_{j,\ \ell,\ 2j+5}$ 的星，每个叶节点都通过一条边与 $\gamma_{j,\ \ell,\ 0}$ 相连；将此星称为 $R_{j,\ \ell}$。
▷ 对于每个 $j \in [k]$ 和每个 $\ell \in [M-C]$，构造一个中心为 $\delta_{j,\ \ell,\ 0}$ 且有 $2j+4$个叶节点 $\delta_{j,\ \ell,\ 1}$、...、$\delta_{j,\ \ell,\ 2j+4}$ 的星，每个叶节点都通过一条边与 $\delta_{j,\ \ell,\ 0}$ 相连；将此星称为 $S_{j,\ \ell}$。

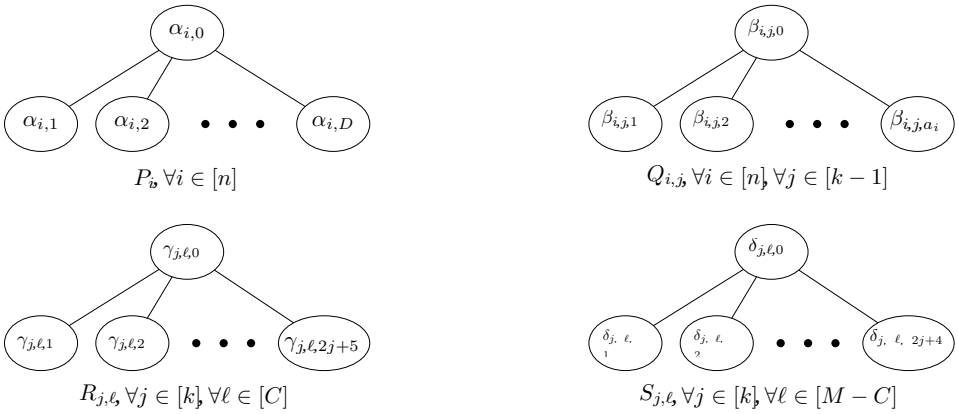


图 2 G_2的构造。

G_2 中的节点总数为 $n(D+1) + \sum_{i\in[n]} (k-1)(a_i+1) + \sum_{j\in[k]} C(2j+6) + (M-C)(2j+5) = n(D+k) + M(k^2+7k) = n'$。我们将目标值设为 n'。这就完成了对该归约的描述。显然，该归约可以在多项式时间内完成。由于 G_2 已经是一个大小为 n'的星森林，根据观察 1，问题归结为判断 G_2 是否为 G_1的子图。

正确性证明：正向。

假设 k-路划分实例是一个是实例。因此，存在 $[n]$ 的一个划分 B_1、.....、B_k，使得对于每个 $j \in [k]$，$\sum_{i \in B_j} a_i = C$。我们证明，这将意味着 G_2 是 G_1的一个子图；因此，根据观察 1，归约后的实例将是一个是实例。

我们定义 $b_i \in [k]$，使得 $i \in B_{b_i}$。令 $p_i = \sum_{i' \in B_{b_i}, i' < i} a_i$ 为索引小于 i 的箱 B_{b_i} 中所有项的总和。对于每个 $j \in [k] \setminus \{b_i\}$，令 $q_{i,j} = \sum a_i$ 为索引小于 a_i 的箱 B_j 外所有项的总和。

我们给出了将星在 G_2 中精确嵌入 G_1 的每棵树 T_i 中的方法（如图 3 所示）。星 P_i 被嵌入 T_i 中，其中 r_i 是中心；除了 $h_{i,1}, \ldots, h_{i,D-1}$ 外，它还包含 s_{i,b_i}。对于所有 $j \in [k] \setminus \{b_i\}$，节点 $s_{i,j}$ 和 $t_i^{(1)}{}_{,j}, \ldots, t^{(a_i)}{}_{i,j}$ 都嵌入星 $Q_{i,j'}$，对每个 $j' \in [k-1]$ 均如此。对于每个 $\ell \in [a_i]$，$R_{b_i,\ell+p_i}$ 都嵌入 T_i 中，其中 $u_i^{(\ell)}{}_{,b_i}$ 是其中心，而 $v_i^{(\ell,1)}{}_{b_i}, , v_i^{(\ell,2)}{}_{b_i}, \ldots, v^{(\ell,2b_i+4)}{}_{i,b_i}$ 和 $t_i^{(\ell)}{}_{,b_i}$ 是叶节点。根据 p_i 的定义，B_{b_i} 中索引最低且满足 $i^* > i$ 的项 i^* 会将 $R_{b_{i^*},\ell+p_{i^*}} = R_{b_i,\ell+(p_i+a_i)}$ 嵌入其中，对每个 $\ell \in [a_{i^*}]$ 均如此。因此，每个 $R_{j',\ell'}$ 都恰好嵌入一个 $T_{i'}$ 中。最后，对于所有 $j \in [k] \setminus \{b_i\}$ 以及所有 $\ell \in [a_i]$，$S_{j,\ell+q_{i,j}}$ 都嵌入 T_i 中，其中 $u_i^{(\ell)}{}_{,j}$ 是其中心，而 $v_i^{(\ell,1)}{}_{,j}, v_i^{(\ell,2)}{}_{,j}, \ldots, v_i^{(\ell,2j+4)}{}_{,j}$ 是叶节点。根据 $q_{i,j}$ 的定义，$[n] \setminus B_j$ 中索引最低且满足 $i^* > i$ 的项 i^* 会将 $S_{b_{i^*},\ell+q_{i^*},j} = S_{b_i,\ell+(q_{i,j}+a_i)}$ 嵌入其中，对每个 $\ell \in [a_{i^*}]$ 均如此。因此，每个 $S_{j',\ell'}$ 都恰好嵌入一个 $T_{i'}$ 中。这就完成了正向的正确性证明。

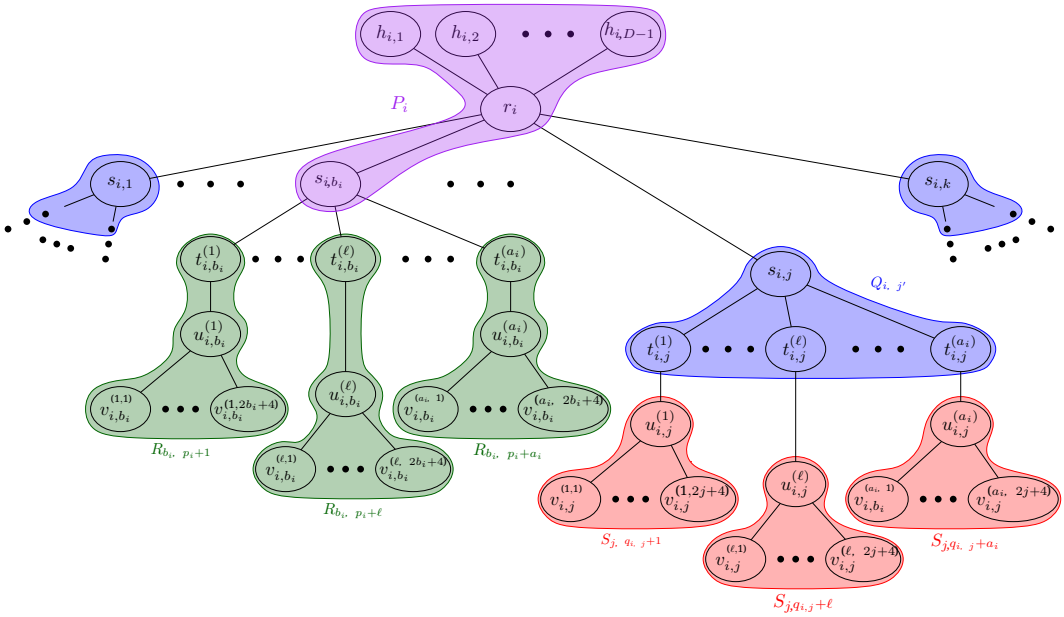


图 3 G_2 在 T_i 中的星嵌入，适用于分支 b_i 和分支 $j \neq b_i$。

正确性证明：反向。

假设归约后的实例是一个是实例，即 G_2 是 G_1 的子图（根据观察 1）。我们通过证明一系列断言，说明 k-路划分实例也是一个是实例。直观而言，这些断言共同证明存在一种嵌入，其形式与正向中得到的嵌入完全相同，即如图 3 所示。

▷ 断言 22. 存在一种将 G_2 嵌入到 G_1 中的方式，使得对于每个 $i \in [n]$，P_i 都被嵌入到 T_i 中，并以 r_i 为中心；除 $h_{i,1}$、……、$h_{i,D-1}$之外，还包含 s_{i,b_i}，其中某个 $b_i \in [k]$。我们称这样的嵌入为 P-良好。

证明。由于 G_2 是 G_1的子图，$\forall i \in [n]$可嵌入到 G_1中。令 $\exists i^* \in [n]$ 满足：对于任意 $i \in [n]$，$\alpha_{i^*},0$ 在 G_1 中的嵌入都不位于 r_i上。在这种情况下，令 w^* 为 $\alpha_{i^*},0$ 在 G_1中的嵌入。注意，G_1 中除 r_i之外的所有顶点的度都至多为$a_1 + 1 < D$。因此，degree(w^*) $< D =$ degree(α_{i^*}, 0)。所以，存在 $\exists j \in [D]$，使得 embedding of α_{i^*},j 不是 w^*的邻居。但是，这样一来，边 {$\alpha_{i^*},0,\alpha_{i^*},j$} 就不会出现在 P_i^*嵌入到 G_1的结果中，这与嵌入的定义矛盾。

这证明了 $\forall i \in [n]\ \exists j \in [n]$，即$\alpha_i,0$在 G_1 中的嵌入是 r_j。不失一般性，我们可以假设α_i, 0被嵌入 r_i中。现在，我们假设 $\exists i^* \in [n], j \in [D-1]$ 使得h_{i^*},j 不包含在 P_{i^*}的嵌入中。由于 $|V(G_2)| = |V(G_1)|$，存在 $\exists w^* \in V(G_2 \setminus P_{i^*})$使得 w^* 被嵌入到h_{i^*},j。由于 $G_2 \setminus P_{i^*}$ 中没有孤立顶点，存在 $\exists x^* \in V(G_2 \setminus P_{i^*})$使得{$x^*, w^*$}是 G_2中的一条边。注意 x^* 并未被嵌入到 r_i，因此边{x^*, w^*}不出现在 G_2 在 G_1中的嵌入中，矛盾。

这证明了存在一个嵌入，使得 $\forall i \in [n]$、P_i 被嵌入到以 r_i 为中心的 T_i中，并且包括 $h_{i,1}$，……，$h_{i,D-1}$。由于 degree($\alpha_{i,0}$) $= D$，将 P_i嵌入 T_i 会包括 r_i的 D 个邻居。其中 $D-1$ 个邻居是 $h_{i,1}$，……，$h_{i,D-1}$。因此，在 $s_{i,1}$，$\cdots s_{i,k}$中恰好还包括一个邻居；我们将其记为s_{i,b_i}。◁

▷命题 23。存在一个将 G_2嵌入 G_1的 P-良好嵌入，使得对于所有 $j \in [k] \setminus \{b_i\}$，节点 $s_{i,j}$和 $t_i^{(1)},j$，……，$t^{(a_i)}i,j$嵌入 $Q_{i,j'}$星，对于每个 $j' \in [k-1]$均如此。将这样的嵌入称为 PQ-良好。

证明。 对于一个 P-良好嵌入 ϕ $i \in [n]$ $\exists j' \in [k-1]\ \forall j \in [k]$ b_i $Q_{i\,j'}$ $s_{i\,j}$ $t_i^{(1)}$ … $t_i^{(a_i}$ i，令其为满足以下条件的最小索引：，\{}、，未被节点，和，j，，），j嵌入。我们称 ϕ的第一个违规索引为 viol(ϕ)。如果不存在 PQ-良好嵌入，那么每个 P-良好嵌入都有一个第一个违规索引。考虑一个嵌入 $\phi^* \in \arg\max_{\phi \text{ is P-good}}\{\text{viol}(\phi)\}$，取任意满足以下条件的嵌入：

第一个违规索引的最大索引。令 i^* =viol(ϕ^*)。由于 ϕ^* 是 P-良好嵌入，r_i 嵌入$\alpha_i,0$ $\forall i \in [n]$（根据断言 22）。此外，根据假设，对于所有 $i \in [i^*-1]$，对于所有 $j \in [k] \setminus \{b_i\}$，节点 $s_{i,j}$和 $t_i^{(1)},j, \ldots, t^{(a_i)}{}_{i,j}$ 嵌入星 $Q_{i,j'}$，对每个 $j' \in [k-1]$ 均如此。令 i' 为满足 $a_{i'} = a_{i^*}$ 的最大索引。图 G_2 有 $(k-1)(i'-i^*+1)$ 个恰有 a_{i^*} 个叶节点的星。令 $X \subseteq V(G_1) \setminus (\{r_i \mid i \in [n]\} \cup \{s_i, b_i \mid i \in [n]\} \cup \{s_{i,j} \mid i \in [i^*-1], j \in [k]\})$ 为 G_1 中嵌入这些星的中心的顶点集合。因此，$|X| = (k-1)(i'-i^*+1)$ 且 $\forall x \in X$，度(x) $\geqslant a_{i^*}$。但是，$V(G_1) \setminus (\{r_i \mid i \in [n]\} \cup \{s_{i,b_i} \mid i \in [n]\} \cup \{s_{i,j} \mid i \in [i^*-1], j \in [k]\})$ 中恰有 $(k-1)(i'-i^*+1)$ 个度至少为 a_{i^*} 的顶点，即顶点集合 {$s_i, j \mid i \in \{i^*, \ldots, i'\}$ $j \neq b_i$}。因此，每个 $Q_{i,j'}$都嵌入到 G_1中，使得其中心嵌入到某个 i'', j 的 $s_{i''}, j$。由于所有这类 $Q_{i,j'}$ 都同构，不失一般性地，可认为 $Q_{i,j'}$ 嵌入到 G_1 中，且其中心

嵌入到$s_{i,\ j}$，其中某个 j。同样，由于 $s_{i,\ j}$ 嵌入了一个 $Q_{i,\ j'}$ 的中心，顶点$t_i^{(1)}{}_{,\ j}$，$\ldots$，$t^{(a_i)}{}_{i,}$ $_j$嵌入了 $Q_{i,\ j'}$ 的叶节点，因为它们是 $s_{i,\ j}$剩余的唯一邻居。这意味着 viol(ϕ^*) $> i'$，这与假设矛盾。◁

▷断言 24. 存在一个 PQ-良好嵌入 G_2在 G_1中，使得对于每个 $\ell \in [a_i]$, $R_{b_i\ \ell+p_i}$ T_i $_{b_i}$ 中心的 $u_i^{(\ell)}$,，且 $v_i^{(\ell,1)}$,,$v_i^{(\ell,2)}$,,,$v^{(\ell,2b_i+4}$),以及 $t_i^{(\ell)}$,作为叶节点，其中 $p_i =$ $\sum_{i'\in[n],i'<i,b_i=b_{i'}}$ $_{b_i}$ $_{b_i}$ $\cdots$ $_{i\ b_i}$ a_{i}。我们称这样的嵌入为 PQR-良好嵌入。◁

证明。对于任意 PQ-良好嵌入，$X = \{h_{i,j} \mid i \in [n], j \in [D-1]\} \cup \{r_i \mid i \in [n]\} \cup \{s_{i,\ j} \mid i \in [n], j \in [k] \setminus \{b_i\}\} \cup \{t_i^{(a)}{}_{,j} \mid i \in [n], j \in [k] \setminus \{b_i\}\ a \in [a_i]\}$中的所有顶点都已嵌入所有 P_i和$Q_{i,j'}$。考虑剩余图 $G_1^R = G_1[V(G_1) \setminus X]$。该图是一个星森林，其中所有星都以某个 $u_i^{(a)}$，$_j$ 为中心。现在，由于 G_2 是 G_1的子图，$\{R_{j,\ell}\} \cup \{S_{j,\ell}\}$中任何具有 q 个叶节点的星都必须嵌入到 G_1^R中具有 q 个叶节点的星。星$R_{b_i,\ell+p_i}$有 $2b_i+5$ 个叶节点，因此必须嵌入到 G_1^R中具有 $2b_i+5$ 个叶节点的星。然而，G_1^R中所有具有 $2b_i+5$ 个叶节点的星，其中心都位于 $u_{i'}^{(a)}$，$_{b_i}$，并具有 $v_{i'\ b_i}^{(a,1)}$,,$v_{i'\ b_i}^{(a,2)}$,,,,,,$v^{(a,2b_i+4}$) $_{i',b_i}$和$t_{i'}^{(a)}$ $_{,b_i}$，其中某个 $a \in [a_i]$满足 $b_{i'} = b_i$。由于这些图彼此同构，我们不失一般性地认为$R_{b_i,\ell+p_i}$ 嵌入到 G_1 中，其中$\gamma_{b_i,\ell+p_i}$,0被嵌入到 $u_{i\ b_i}^{(\ell)}$,。◁

▷ 断言 25. 对于 G_1 中任意一个 PQR 良嵌入 G_2，对每个 $j \in [k]$，令 $B_j = \{i \in [n] \mid b_i = j\}$。那么对所有 $j \in [k]$，$\sum_{i\in B_j} a_i = C$.

证明。 考虑 G_1 中的一个 PQR 良嵌入 G_2。令 $R_{j,\ell}$ 嵌入某个 T_i 中。这意味着 $b_i = j$，因为 T_i 仅将 $R_{b_i,\ell+p_i}$ 嵌入 $\ell \in [a_i]$。因此，

$$\bigcup_{i\in[n],b_i=j} \{R_{b_i,\ell+p_i} \mid \ell \in [a_i]\} = \{R_{j,\ell} \mid j = b_i, \ell \in [C]\}$$

注意，左侧是两两不相交集合的并（由断言 24 可知）。因此，令它们的大小相等，可得:

$$\sum_{i\in[n],b_i=j} a_i = C$$

由此即证。 ◁

这证明了 k-路划分实例是一个是实例。因此，我们完成了 MaxComSubG 的 NP 困难性证明。◀

由于树深度为 5 的树的路径宽度至多为 4，且树宽为 1，即使将两张输入图的树宽或路径宽度作为参数，MaxComSubG 仍然是 para-NP-困难的。

定理 17 和定理 21 表明，即使我们限制两张图的树深度（甚至路径宽度/树宽），或者将两张图的最大度限制为一个较小的常数，MaxComSubG 仍然是 para-NP-困难的。然而，我们发现，当同时将两张输入图的树深度和最大度作为参数时，MaxComSubG 存在 FPT 算法。为证明这一点，我们首先证明定理 26，表明当两张输入图中最大连通

分量的大小作为参数时，MaxComSubG 存在 FPT 算法；随后利用这一结果证明推论27：当以两张输入图的树深度和最大度为参数时，MaxComSubG 存在 FPT 算法。其策略与定理 20 的证明类似。

▶ **定理 26.** MaxComSubG 可由一种FPT 算法求解，运行时间为 $2^{2^{\mathcal{O}(k^2)}} \cdot n^{\mathcal{O}(1)}$，参数为 k，且两个图的每个连通分量至多包含 k 个节点。

证明。 设输入为图 G_1、G_2，我们的目标是最大化公共星森林子图的大小。设 $n = |V(G_1)| + |V(G_2)|$。该 FPT 算法分为两个阶段：预处理阶段 和 整数线性规划阶段。预处理阶段计算整数线性规划所需的值，随后在整数线性规划阶段求解。

预处理阶段。 首先，我们枚举所有由 k 个节点构成的不同有序连通图。这些图记为 C_1、C_2、.....、C_{k_1}。只需枚举 k 个顶点之间边的所有可能子集即可完成此操作。因此 $k_1 \leqslant 2^{\binom{k}{2}}$，且该过程可在$\mathcal{O}(k_1)$ 时间内完成。

令 p_i 为 G_1中 C_i 的副本数，并令 q_i 为 G_2中 C_i的副本数，对所有 $i \in [k_1]$均成立。任何简单搜索算法（如广度优先搜索或深度优先搜索）都能在 $\mathcal{O}(k_1 \cdot k \cdot n)$ 时间内完成此操作。注意，$(p_1, \ldots, p_{k_1}, q_1, \ldots, q_{k_1})$ 完全确定了图 G_1 和 G_2。

接下来，我们枚举集合 $\{0, 1, \ldots, k-1\}^k$ 中的所有元素。令 $S_1, S_2, \ldots, S_{k_2}$（对 $k_2 = k^k$ 而言）为有序元组，其中 $S_\ell = (s_\ell^{(1)}, s_\ell^{(2)}, \ldots, s_\ell^{(k)})$，$s_\ell^{(j)} \in \{0, 1, \ldots, k-1\}$ 对所有 $j \in [k], \ell \in [k_2]$ 均成立。该枚举需要 $\mathcal{O}(k \cdot k_2)$ 时间。

对于 $i \in [k_1]$、$\ell \in [k_2]$，我们将 S_ℓ 到 C_i 的一个实现定义为一组边的子集，这些边在 C_i 中诱导出一个星森林，并且对于每个 $j \in [k]$，恰好有 $s_\ell^{(j)}$ 个具有 j 个叶节点（及 1 个中心）的星出现。接下来，对于每个 $i \in [k_1]$、$\ell \in [k_2]$，我们计算布尔变量 $\text{br}_{i,\ell} \in \{$true, false$\}$。如果存在 S_ℓ 到 C_i 的一个实现，则 $\text{br}_{i,\ell}$ 为 true；否则 $\text{br}_{i,\ell}$ 为 false。由于 $|V(C_i)| \leqslant k$，所有星最多只能有 k 个叶节点，并且星的数量不能超过 k。计算 $\text{br}_{i,\ell}$ 的一个简单算法如下：首先将所有 $\text{br}_{i,\ell}$ 初始化为 false，然后遍历每个 $i \in [k_1]$：对于 C_i 中的每个边子集，检查删除这些边后是否会使 C_i 留下一组星，其中对于 $j \in [k]$，恰好有 $s_\ell^{(j)}$ 个具有 j 个叶节点的星出现。整个过程的运行时间为 $O(k_1 \cdot 2^{\binom{k}{2}} \cdot k) = O(k \cdot k_1^2)$。至此，预处理阶段结束。

整数线性规划阶段。 要解决 MaxComSubG，我们需要从 G_1 和 G_2 中移除一部分边，使剩余边所诱导的子图 H 成为星的集合，并使 H 中的节点数最大化。现在，固定 G_1 的一个连通分量 C。我们将把 H 称为解。令 C 是某个 $i \in [k_1]$ 的副本 C_i。令 $H \cap C$ 中的边诱导出一个星森林，其中具有 $j \in [k]$ 个叶节点的星出现 $s^{(j)}$ 次。必定存在某个 $\ell \in [k_2]$，使得 $S_\ell = (s_\ell^{(1)}, s_\ell^{(2)}, \ldots, s_\ell^{(k)}) = (s^{(1)}, s^{(2)}, \ldots, s^{(k)})$，也就是说，$H \cap C$ 中的边是 S_ℓ 到 C 的一个实现。此外，这样的实现存在当且仅当 $\text{br}_{i,\ell}$ = 为真。

现在我们介绍整数线性规划中的变量。在 p_i C_i 在 G_1 中的出现次数中，我们将整数变量 $x_{i,\ell}$（其中 $i \in [k_1], \ell \in [k_2]$）定义为：使得某个解 H 将 S_ℓ 实现到该连通分量中的连通分量数量。类似地，在 G_2 中，令 $y_{i,\ell}$（其中 $i \in [k_1], \ell \in [k_2]$）表示满足某个解 H 将 C_i 的一个实现 p_i

到该连通分量中的出现次数。这些将是我们所构造的整数线性规划中的唯一变量。

接下来，我们对整数线性规划的约束进行形式化，这些约束应确保变量 $x_{i,\ell}$ 和 $y_{i,\ell}$ 对应于某个有效解。首先，对于 G_1 中 C_i 的所有出现 C，解 H 的边必须诱导出某个 S_ℓ 到 C 中的 $\ell \in [k_2]$ 的实现。因此，对于固定的 i，变量 x_i、ℓ 之和必须恰好等于 p_i；同样地，变量 $y_{i,\ell}$ 之和必须恰好等于 q_i。

接下来需要确保的条件是，仅当 $\mathsf{br}_{i,\ell} =$ 为真时，才允许将 S_ℓ 实现到 C_i 中。因此，对于所有满足 $\mathsf{br}_{i,\ell} =$ 为假的 $i \in [k_1]$、$\ell \in [k_2]$，我们施加约束 $x_{i,\ell} = y_{i,\ell} = 0$。

最后，对于每个 $j \in [k]$，解 H 的边在 G_1 和 G_2 中必须诱导出数量相等的具有 j 个叶节点的星。在 G_1 中 C_i 的任意一次出现中，当 H 的边诱导出该连通分量中的 S_ℓ 的实现时，H 的边所诱导的这类星的数量为 $s_\ell^{(j)}$。因此，H 在 G 中的边所诱导的这类星的出现总数为 $\sum_{i\in[k_1]} \sum_{\ell\in[k_2]} s_\ell^{(j)} \cdot x_{i,j}$。类似地，$i\in[k_1]\ \ell\in[k_2]$在 G 中的边所诱导的这类星的出现总数为 $\sum_{i\in[k_1]} \sum_{\ell\in[k_2]} s_\ell^{(j)} \cdot y_{i,j}$。对于每个 $j \in [k]$，这两个数量必须相等。$i\in[k_1]\ \ell\in[k_2]$这就完成了使这些变量对应于有效解所需的条件。

我们希望最大化这样的解中节点的数量 H。我们已经证明，具有 j 个叶节点的星的出现总次数为 H 是 $\sum_{i\in[k_1]} \sum_{\ell\in[k_2]} s_\ell^{(j)} \cdot x_{i,j}$。每个这样的$i\in[k_1]\ \ell\in[k_2]$星包含 $(j+1)$ 个节点。因此，H 中的节点总数为 $\sum_{j\in[k]} \sum_{i\in[k_1]} \sum_{\ell\in[k_2]} (j+1) \cdot s_\ell^{(j)} \cdot x_{i,j}$。这就是我们要最大化的目标函数。

因此，我们得到一个含有 $k_3 = 2 \cdot k_1 \cdot k_2$ 个变量的整数线性规划。该问题可在时间 $k^{\mathcal{O}(k_3)} \cdot n^{\mathcal{O}(1)} = 2^{2^{\mathcal{O}(k^2)}} \cdot n^{\mathcal{O}(1)}$ 时间 [7] 内求解。这一步的耗时超过了预处理阶段的所有其他步骤。为完整起见，我们在下文给出整数线性规划的正式形式。

$$
\begin{aligned}
&\max \sum_{j\in[k]} \sum_{i\in[k_1]} \sum_{\ell\in[k_2]} (j+1) \cdot s_\ell^{(j)} \cdot x_{i,j} \\
&\qquad \text{subject to:} \\
&\qquad\qquad \sum_{\ell\in[k_2]} x_{i,\ell} = p_i && \forall i \in [k_1] \\
&\qquad\qquad \sum_{\ell\in[k_2]} y_{i,\ell} = q_i && \forall i \in [k_1] \\
&\qquad\qquad x_{i,\ell} = 0 && \forall i \in [k_1], \ell \in [k_2], \text{ with } \mathsf{br}_{i,\ell} = \text{FALSE} \\
&\qquad\qquad y_{i,\ell} = 0 && \forall i \in [k_1], \ell \in [k_2], \text{ with } \mathsf{br}_{i,\ell} = \text{FALSE} \\
&\sum_{i\in[k_1]} \sum_{\ell\in[k_2]} s_\ell^{(j)} \cdot x_{i,j} = \sum_{i\in[k_1]} \sum_{\ell\in[k_2]} s_\ell^{(j)} \cdot y_{i,j} && \forall j \in [k] \\
&\qquad\qquad x_{i,\ell}, y_{i,\ell} \in \mathbb{Z}_{\geqslant 0} && \forall i \in [k_1], \ell \in [k_2]
\end{aligned}
$$

◀

现在，我们已准备好证明以两个输入图的树深度和最大度作为参数的结果。

▶**推论27**。 MaxComSubG 存在一个以 d 和 Δ 为参数的FPT 算法，使得两个输入图的树深度至多为 d ，最大度至多为 Δ 。该算法的运行时间为 $2^{2^{\mathcal{O}(\Delta^{2d})}} \cdot n^{\mathcal{O}(1)}$。

证明。 我们首先证明以下结论。

▷ 断言28. 树深度至多为 d 且最大度为 Δ 的图的任意连通分量的大小至多为 $\Delta^d - 1$。

证明。 我们对 d 进行归纳。对于 $d = 0$，该图必定没有顶点，特别是顶点数至多为 $\Delta^d - 1 = 0$。现在，对于某个 $d \geqslant 1$，假设所有最大度至多为 Δ 且树深度至多为 $d-1$ 的图，其连通分量的大小至多为 $\Delta^{d-1} - 1$。令 G 为一个最大度至多为 Δ 且树深度至多为 d 的图。每个连通分量 C （属于 G ）的最大度至多为 Δ ，且树深度至多为 d。我们证明 $|C| \leqslant \Delta^d - 1$。

根据定义， $d \geqslant \mathsf{td}(C) = \min^{v \in V(C)}\{\mathsf{td}(C - v)\} + 1$。设 $u \in V(C)$ 是满足 $\mathsf{td}(C) = \mathsf{td}(C - u) + 1$ 的顶点。因此， $C - u$ 的每个连通分量的树深度都至多为 $d - 1$ ，且其大小不可能超过 $\Delta^{d-1} - 1$。此外， $C - u$ 中这样的连通分量至多有Δ 个，因为 u 在 C 中的度至多为 Δ。因此$|C - u| \leqslant \Delta \cdot (\Delta^{d-1} - 1) = \Delta^d - \Delta$，从而 $|C| = |C - u| + 1 \leqslant \Delta^d - 1$。◁

因此，定理 26 中的 FPT 算法可以直接给出一个针对参数 Δ 和 d 的 FPT 算法，其运行时间为 $2^{2^{\mathcal{O}(\Delta^{2d})}} \cdot n^{\mathcal{O}(1)}$。◀

推论27促使我们尝试设计以两个输入图的最大度和路径宽度为参数的 MaxComSubG FPT 算法。然而，我们证明，即使两个图的路径宽度都至多为 4，以两个图的最大度为参数时，MaxComSubG 仍是 W[1]-难的。

▶**定理29**。 MaxComSubG 关于 Δ 是 W[1]-难的，其中两个图都是路径宽度至多为 4 的不相交树的集合，且两个图的最大度至多为 Δ。

证明。 我们从以下 W[1]-难问题进行参数化归约：当参数为 k 时的 k 路划分问题。设 $a_1, \ldots, a_n, k$ 和 C 为 k 路划分问题的输入，其中 a_i 是一个正整数。此外，不失一般性，我们假设$a_1 \geqslant a_2 \geqslant \ldots \geqslant a_n$。

归约后的实例的构造。

我们的归约构造了一个 MaxComSubG 实例，其图为 G_1 和 G_2。 G_1 将由一组 n 树组成，每棵树的路径宽度至多为 4，且最大度为 $3k + 7$；而G_2 将包含多个最大度为 $2k + 8$的星。 G_1 和 G_2 的节点数 n'完全相同，目标值将等于数值 n'。

令 $M = \sum_{i \in [n]} a_i$、 $a = \max^{i \in [n]} a_i$、 $D = 2k + 8$ 和 $E = 2k + 6$。注意， $kC = M$（否则它就是一个平凡的否实例）。现在我们定义 G_1 的构造。对于每个 $i \in [n]$，将树 T_i 加入 G_1，其定义如下（图4a）。

22 最大公共顶点子图的参数化复杂性

▷ 构造一个节点 r_i。
▷ 构造 $D-1$ 个节点 $h_{i,\ 1}$， $h_{i,\ 2}$，……， $h_{i,\ D-1}$，它们均与 r_i 相邻。
▷ 构造 k 个节点 $s_{i\ 1}^{(0)}$，， $s_{i\ 2}^{(0)}$，、……、 $s_{i\ k}^{(0)}$，，它们都与 r_i 相邻。
▷ 对于每个 $j \in [k]$，按如下方式构造。构造 $3(a-2)$ 个节点 $s_i^{(1)},_j, s_i^{(2)},_j, \ldots, s_i^{(a-2)},_j, y_i^{(1)},_j, y_i^{(2)},_j, \ldots, y_i^{(a-2)},_j, z_i^{(1)},_j, z_i^{(2)},_j, \ldots, z_i^{(a-2)},_j$。构造 $2a$ 个节点 $t_i^{(1)},_j, t_i^{(2)},_j, \ldots, t_i^{(a)},_j, u_i^{(1)},_j, u_i^{(2)},_j, \ldots, u_i^{(a)},_j$对于每个 $\ell \in [a-2]$，按如下方式构造。

-
  - 在 $s(\ell)\ i,j$ 和 $z(\ell)\ i,j$之间构造一条边。
  - 在 $z(\ell)\ i,j$ 和 $y(\ell)\ i,j$之间构造一条边。
  - 在 $y(\ell)\ i,j$ 和 $s(\ell-1)\ i,j$ 之间构造一条边。
    - 在 $t(\ell)\ i,j$ 和 $s(\ell-1)\ i,j$ 之间构造一条边。
- 在 $t(a-1)\ i,j$ 和 $s(a-2)\ i,j$之间构造一条边在 t
- $(a)\ i,j$ 和 $s(a-2)\ i,j$ 之间构造一条边。对于每个
- $\ell \in [a_i]$，构造如下内容。
  - 在 $u(\ell)\ i,j$ 和 $t(\ell)\ i,j$之间构造一条边。
    - 构造 $2j+4$ 个节点 $v(\ell,1)\ i,j, v(\ell,2)\ i,j, \ldots, v(\ell,2j+4)\ i,j$，每个节点都与 u
- <style id='45'\ >$\ell)\ i,j$相邻。对于每个 $\ell \in [a]\backslash\ [a_i]$，构造如下内容。
  - Construct an edge between $u_{i,j}^{(\ell)}$ and $t_{i,j}^{(\ell)}$.
  - Construct E nodes $v_{i,j}^{(\ell,1)}, v_{i,j}^{(\ell,2)}, \ldots, v_{i,j}^{(\ell,E)}$, each of them adjacent to $u_{i,j}^{(\ell)}$.

G_1 只是由树 T_1、……、T_n组成的集合。 G_1 中度数最高的节点是 r_i （$i \in [n]$），每个节点的度均为 $D-1+k = 3k+7$。

▷ 断言 30： G_1 的路径宽度至多为 4。

证明。 只需证明 T_i 的路径宽度对每个 $i \in [n]$均至多为 4。我们固定一个$i \in [n]$。考虑图 $T_i - r_i$。它有以下连通分量：

▷ 单顶点连通分量 $\{h_{i,\ 1}\}$、$\{h_{i,\ 2}\}$……， $\{h_{i,\ D-1}\}$▷ 对于 $j \in [k]$，
包含这些顶点的连通分量

$$J_{i,j} = \left\{s_{i,j}^{(0)}\right\} \cup \left\{s_{i,j}^{(\ell)}, y_{i,j}^{(\ell)}, z_{i,j}^{(\ell)} \mid \ell \in [a-2]\right\} \cup \left\{t_{i,j}^{(\ell)}, u_{i,j}^{(\ell)} \mid \ell \in [a]\right\}$$
$$\cup \left\{v_{i,j}^{(\ell,1)}, \ldots, v_{i,j}^{(\ell,2j+4)} \mid \ell \in [a_i]\right\} \cup \left\{v_{i,j}^{(\ell,1)}, \ldots, v_{i,j}^{(\ell,E)} \mid \ell \in [a] \setminus [a_i]\right\}$$

显然，$\{h_{i,\ 1}\}$、$\{h_{i,\ 2}\}$……， $\{h_{i,\ D-1}\}$ 的路径宽度均为 0。现在固定 $j \in [k]$。我们给出 $J_{i,j}$ 的一个宽度为 3 的路径分解。定义以下“袋”，每个袋包含 4 个节点。

▷ For $\ell \in [a_i - 1]$, $\ell' \in [2j+4]$, define $\mathcal{X}_{i,j}^{(\ell,\ell')} = \left\{s_{i,j}^{(\ell-1)}, t_{i,j}^{(\ell)}, u_{i,j}^{(\ell)}, v_{i,j}^{(\ell,\ell')}\right\}$
▷ For $\ell = a_i$, $\ell' \in [2j+4]$,

- If $a = a_i$, then define $\mathcal{X}_{i,j}^{(\ell,\ell')} = \left\{s_{i,j}^{(\ell-2)}, t_{i,j}^{(\ell)}, u_{i,j}^{(\ell)}, v_{i,j}^{(\ell,\ell')}\right\}$
- Otherwise, define $\mathcal{X}_{i,j}^{(\ell,\ell')} = \left\{s_{i,j}^{(\ell-1)}, t_{i,j}^{(\ell)}, u_{i,j}^{(\ell)}, v_{i,j}^{(\ell,\ell')}\right\}$

▷ 对于 $\ell \in [a-1]\backslash\ [a_i]$, $\ell' \in [E]$, 定义 $\mathrm{z}^{(\ell,\ell')}{}_{i,j} = \left\{s_{i\ j}^{(\ell-1)},, t_i^{(\ell)},_j, u_i^{(\ell)},_j, v^{(\ell,\ell')}{}_{i,j}\right\}$▷ 对于 $\ell' \in [E]$, 定义 $\mathrm{z}^{(a,\ell')}{}_{i,j} = \left\{s_{i\ j}^{(a-2)},, t_i^{(a)},_j, u_i^{(a)},_j, v^{(a,\ell')}{}_{i,j}\right\}$

▷ For $\ell \in [a-2]$, define $\mathcal{A}_{i,j}^{(\ell)} = \left\{ s_{i,j}^{(\ell-1)}, y_{i,j}^{(\ell)}, z_{i,j}^{(\ell)}, s_{i,j}^{(\ell)} \right\}$

序列 $X_i^{(1,1)}$,j, . . ., X (1,$2j+4$) i,j, $A_i^{(1)}$,j, X (2,1) i,j, . . ., X (2,$2j+4$) i,j, $A_i^{(2)}$,j, X (3,1) i,j, . . ., X (a_i, $2j+4$) i,j, $A^{(a_i)}$ i,j, $Z^{(a_i+1,1)}$ i,j, . . ., $Z^{(a_i+1,E)}$ i,j, $A^{(a_i+1)}$ i,j, $Z^{(a_i+2,1)}$ i,j, . . ., $Z_i^{(a-2,E)}$,j, $A_i^{(a-2)}$,j, $Z_i^{(a-1,1)}$,j, . . ., $Z_i^{(a-1,E)}$,j, $Z_i^{(a,1)}$,j, . . ., $Z_i^{(a,E)}$,j 给出了 $G_1[J_{i,j}]$ 的一个宽度为3的路径分解。

因此， $T_i - r_i$ 具有宽度至多为 3 的路径分解，这可通过拼接其各个连通分量的路径分解实现。因此， T_i 具有宽度至多为 4 的路径分解，这可通过将 r_i 添加到 T_i的路径分解的所有袋中实现。◁

此外， T_i 所包含的节点数为 $1 + (D-1) + k + 3k(a-2) + 2ka + k(a - a_i)E + a_i \cdot \sum(2j+4) = (2ak^2 + 11ak - 3k + 8) - (k^2 + k)a_i$。因此， G_1 总共包含$\sum_{j\in[k]} n' = n(2ak^2 + 11ak - 3k + 8) - (k^2 + k)M$ 个节点，这是该实例输入规模的多项式。

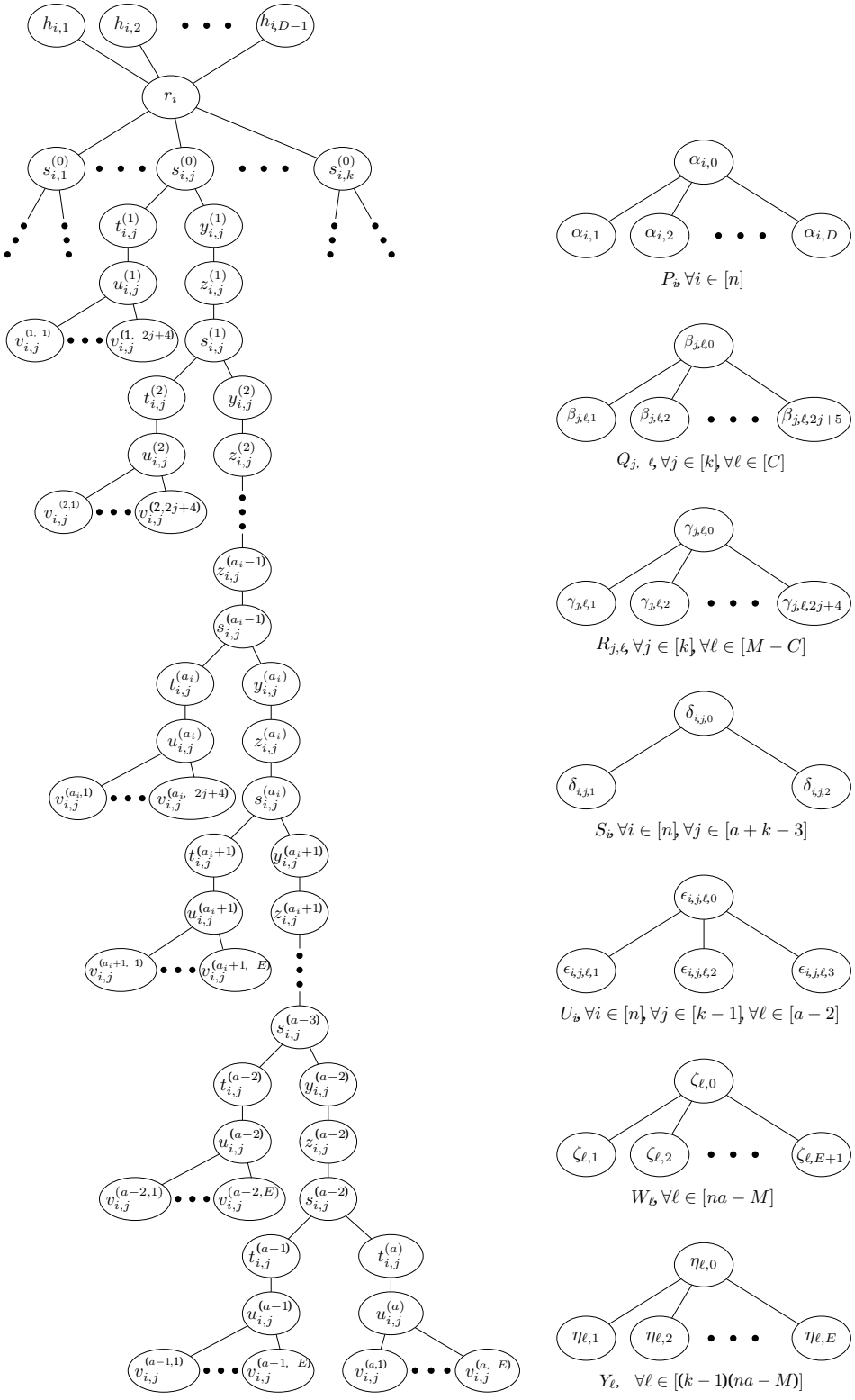
现在定义 G_2的构造。其中将包含以下星形图（图4b）。

▷对于每个 $i \in [n]$，构造一个中心为 α_i,0 的星，并设有 D 个叶节点 α_i,1, . . . , $\alpha_{i,D}$，每个叶节点都通过一条边与 α_i,0 相连；将此星称为 P_i。▷对于每个 $j \in [k]$ 和每个 $\ell \in [C]$，构造一个中心为 $\beta_{j,\ell}$,0 的星，并设有 $2j+5$ 个叶节点 $\beta_{j,\ell}$,1, . . . , $\beta_{j,\ell,2j+5}$，每个叶节点都通过一条边与 $\beta_{j,\ell}$,0 相连；将此星称为 $Q_{j,\ell}$。▷对于每个 $j \in [k]$ 和每个 $\ell \in [M-C]$，构造一个中心为 $\gamma_{j,\ell}$,0 的星，并设有 $2j+4$ 个叶节点 $\gamma_{j,\ell}$,1, . . . ,$\gamma_{j,\ell,2j+4}$，每个叶节点都通过一条边与 $\gamma_{j,\ell}$,0 相连；将此星称为 $R_{j,\ell}$。▷对于每个 $i \in [n]$, $j \in [a+k-3]$，构造一个中心为 $\delta_{i,j}$,0 的星，并设有 2 个叶节点 $\delta_{i,j}$,1 和 $\delta_{i,j}$,2，每个叶节点都通过一条边与 $\delta_{i,j}$,0 相连；将此星称为 $S_{i,j}$。▷对于每个 $i \in [n]$, $j \in [k-1]$ 和每个 $\ell \in [a-2]$，构造一个中心为 $\epsilon_{i,j,\ell}$,0 的星，并设有 3 个叶节点 $\epsilon_{i,j,\ell}$,1、$\epsilon_{i,j,\ell}$,2 和 $\epsilon_{i,j,\ell}$,3，每个叶节点都通过一条边与 $\epsilon_{i,j,\ell}$,0 相连；将此星称为 $U_{i,j,\ell}$。▷对于每个 $\ell \in [na - M]$，构造一个中心为 ζ_ℓ,0 的星，并设有 $E+1$ 个叶节点 ζ_ℓ,1, ζ_ℓ,2, . . . ,$\zeta_{\ell,E+1}$，每个叶节点都通过一条边与 ζ_ℓ,0 相连；将此星称为 W_ℓ。▷对于每个 $\ell \in [(k-1)(na-M)]$, ，构造一个中心为 η_ℓ,0 的星，并设有 E 个叶节点 η_ℓ,1, η_ℓ,2, . . . ,$\eta_{\ell,E}$，每个叶节点都通过一条边与 η_ℓ,0 相连；将此星称为 Y_ℓ。

G_2 中的节点总数为 $n(D+1) + \sum_{j\in[k]} C(2j+6) + \sum_{j\in[k]}(M-C)(2j+5) + 3n(a+k-3) + 4n(k-1)(a-2) + (na-M)(E+2) + (k-1)(na-M)(E+1) = n(2ak^2 + 11ak - 3k + 8) - (k^2 + k)M = n'$。此外，由于 G_2 是由星构成的集合，其路径宽度为 1；而 G_2 中度最高的节点为 $\alpha_{i,0}$，对于 $i \in [n]$，每个节点的度均为 $D = 2k + 8$。我们将目标值设为 n'。显然，该归约可在多项式时间内完成。由于 G_2已经是一个大小为 n'的星森林，我们使用观察 1 来判定 G_2 是否为 G_1的子图。

正确性证明：正向。

假设 k-路划分实例是是实例。因此，存在一个划分 B_1、.....、B_k，将 $[n]$ 划分为若干部分，使得对于每个 $j \in [k]$，$\sum_{i\in B_j} a_i = C$。我们将证明，这将意味着 G_2 是 G_1的子图；因此，根据观察 1，归约后的实例将



(a) T_i的构造。

(b) G_2的构造。

图4 T_i 和 G_2的构造。

是实例。我们定义 $b_i \in [k]$，使得 $i \in B_{b_i}$。令 $p_i = \sum a_i$ 为$i' \in B_{b_i}$中所有索引小于 i、属于箱 B_{b_i} 的$i' < i$项之和。对于每个 $j \in [k] \setminus \{b_i\}$，令 $q_{i,j} = \sum a_{ii' \in [n] \setminus B_j, i' < i}$ 为所有索引小于 a_i、位于箱 B_j 外的项之和。此外，对每个 $j \in [k] \setminus \{b_i\}$，令 $e_i = \sum(a - a_i)$ 和$i' < i f_{i,j} = (k-1)e_i + (a - a_i) \cdot |[j] \setminus \{b_i\}|$。这些项 e_i 和 f_i 将在后续归约中帮助我们记录嵌入 G_1 的 W_ℓ 和 Y_ℓ 的数量。

我们为 G_2 提供了一个精确嵌入，将星嵌入 G_1 的每棵树 T_i 中（如图 5 所示）。星 P_i 被嵌入到 T_i 中，其中 r_i 为中心；除 $h_{i,}1, \ldots, h_{i,D-1}$ 外，它还包含 $s_{i\ b_i}^{(0)}$。对于所有 $\ell \in [a-2]$，节点 $y_{i\ b_i}^{(\ell)}$, , $z_{i\ b_i}^{(\ell)}$, , $s_i^{(\ell)}{}_{,b_i}$ 嵌入星 $S_{i,}\ell$。对于所有 $j \in [k] \setminus \{b_i\}$，节点 $s_i^{(0)}{}_{,j}$, $t_i^{(1)}{}_{,j}$ 和 $y_i^{(1)}{}_{,j}$ 对每个 $j' \in [a+k-3] \setminus [a-2]$ 嵌入星 $S_{i,j'}$。对于所有 $j \in [k] \setminus \{b_i\}$ 以及所有 $\ell \in [a-3]$，节点 $s_i^{(\ell)}{}_{,j}$, $z_i^{(\ell)}{}_{,j}$, $t_i^{(\ell+1)}{}_{,j}$ 和 $y_i^{(\ell+1)}{}_{,j}$ 对每个 $j' \in [k-1]$ 嵌入星 $U_{i,j',\ell}$。对于所有 $j \in [k] \setminus \{b_i\}$，$s_i^{(a-2)}{}_{,j}$, $z_i^{(a-2)}{}_{,j}$, $t_i^{(a-1)}{}_{,j}$ 和 $t_i^{(a)}{}_{,j}$ 对每个 $j' \in [k-1]$ 嵌入星 $U_{i,j',a-2}$。对于每个 $\ell \in [a_i]$，$Q_{b_i,\ell+p_i}$ 被嵌入到 T_i 中，其中 $u_i^{(\ell)}{}_{,b_i}$ 为其中心，$v_{i\ b_i}^{(\ell,1)}$, , $v_{i\ b_i}^{(\ell,2)}$, , , . . . , $v^{(\ell,2b_i+4)}{}_{i,b_i}$ 和 $t_i^{(\ell)}{}_{,b_i}$ 为叶节点。根据 p_i 的定义，B_{b_i} 中索引最小且满足 $i^* > i$ 的项 i^* 对每个 $\ell \in [a_{i^*}]$ 都嵌入 $Q_{b_{i^*},\ell+p_{i^*}} = Q_{b_i,\ell+(p_i+a_i)}$。因此，每个 $Q_{j',\ell'}$ 都恰好嵌入一个 $T_{i'}$。对于所有 $j \in [k] \setminus \{b_i\}$ 以及所有 $\ell \in [a_i]$，$R_{j,\ell+q_{i,j}}$ 被嵌入到 T_i 中，其中 $u_i^{(\ell)}{}_{,j}$ 为其中心，$v_i^{(\ell,1)}{}_{,j}$, $v_i^{(\ell,2)}{}_{,j}$, . . . , $v_i^{(\ell,2j+4)}{}_{,j}$ 为叶节点。根据 $q_{i,j}$ 的定义，$[n] \setminus B_j$ 中索引最小且满足 $i^* > i$ 的项 i^* 对每个 $\ell \in [a_{i^*}]$ 都嵌入 $R_{b_{i^*},\ell+q_{i^*,j}} = R_{b_i,\ell+(q_{i,j}+a_i)}$。因此，每个 $R_{j',\ell'}$ 都恰好嵌入一个 $T_{i'}$。对于所有 $\ell \in [a-a_i]$，$W_{\ell+e_i}$ 被嵌入到 T_i 中，其中 $u^{(\ell+a_i)}{}_{i,b_i}$ 为其中心，$v^{(\ell+a_i,1)}{}_{i,b_i}$, $v^{(\ell+a_i,2)}{}_{i,b_i}$, . . . , $v^{(\ell+a_i,E)}{}_{i,b_i}$ 和 $t^{(\ell+a_i)}{}_{i,b_i}$ 为叶节点。根据 e_i 的定义，B_{b_i} 中索引最小且满足 $i^* > i$ 的项 i^* 对每个 $\ell \in [a_{i^*}]$ 都嵌入 $W_{\ell+e_{i^*}} = W_{\ell+(e_i+a-a_i)}$。因此，每个 $W_{\ell'}$ 都恰好嵌入一个$T_{i'}$。最后，对于所有 $j \in [k] \setminus \{b_i\}$ 以及所有 $\ell \in [a-a_i]$，$Y_{\ell+f_{i,j}}$ 被嵌入到 T_i 中，其中 $u^{(\ell+a_i)}{}_{i,j}$ 为其中心，$v^{(\ell+a_i,1)}{}_{i,j}$, $v^{(\ell+a_i,2)}{}_{i,j}$, . . . , $v^{(\ell+a_i,E)}{}_{i,j}$ 为叶节点。根据$f_{i,j}$ 的定义，$[n] \setminus B_j$ 中索引最小且满足 $i^* > i$ 的项 i^* 对每个 $\ell \in [a_{i^*}]$ 都嵌入 $Y_{\ell+f_{i^*,j}} = Y_{\ell+(f_{i,j}+(k-1)(a-a_i))}$。因此，每个 $Y_{\ell'}$ 都恰好嵌入一个 $T_{i'}$。至此，正向的正确性证明完成。

正确性证明：反向。

假设归约后的实例是一个是实例，即 G_2 是 G_1 的子图（根据观察 1）。我们通过证明一系列断言，说明 k-路划分实例也是一个是实例。直观地说，这些断言共同证明存在一个嵌入，与我们在正向中得到的嵌入相同，即如图 5 所示。

▷ 断言 31. 存在一个将 G_2 嵌入 G_1 的嵌入，使得对于每个 $i \in [n]$，P_i 都嵌入到 T_i 中，且以 r_i 为中心；此外，除 $h_{i,1,\ldots,}h_{i,D-1}$外，它还包含 $s_i^{(0)}{}_{,b_i}$，其中某个 $b_i \in [k]$成立。我们称这样的嵌入为 P-一致。

证明。由于 G_2 是 G_1的子图，$\forall i \in [n]$在 G_1中有一个嵌入。取 $\exists i^* \in [n]$，使得对任意 $i \in [n]$，$\alpha_{i^*,}0$ 在 G_1 中的嵌入都不位于 r_i上。在这种情况下，令 w^* 为 $\alpha_{i^*,}0$ 在 G_1中的嵌入。注意，G_1 中除 r_i之外的所有顶点的度都至多为$a_1 + 1 < D$。因此，$\mathsf{degree}(w^*) < D = \mathsf{degree}(\alpha_{i^*,}0)$。所以，存在 $\exists j \in [D]$，使得 $\alpha_{i^*,j}$ 的嵌入不是 w^*的邻居。但是，这样一来，边 $\{\alpha_{i^*,}0, \alpha_{i^*,j}\}$ 就不会出现在 P_i^*到 G_1的嵌入中，这与嵌入的定义矛盾。

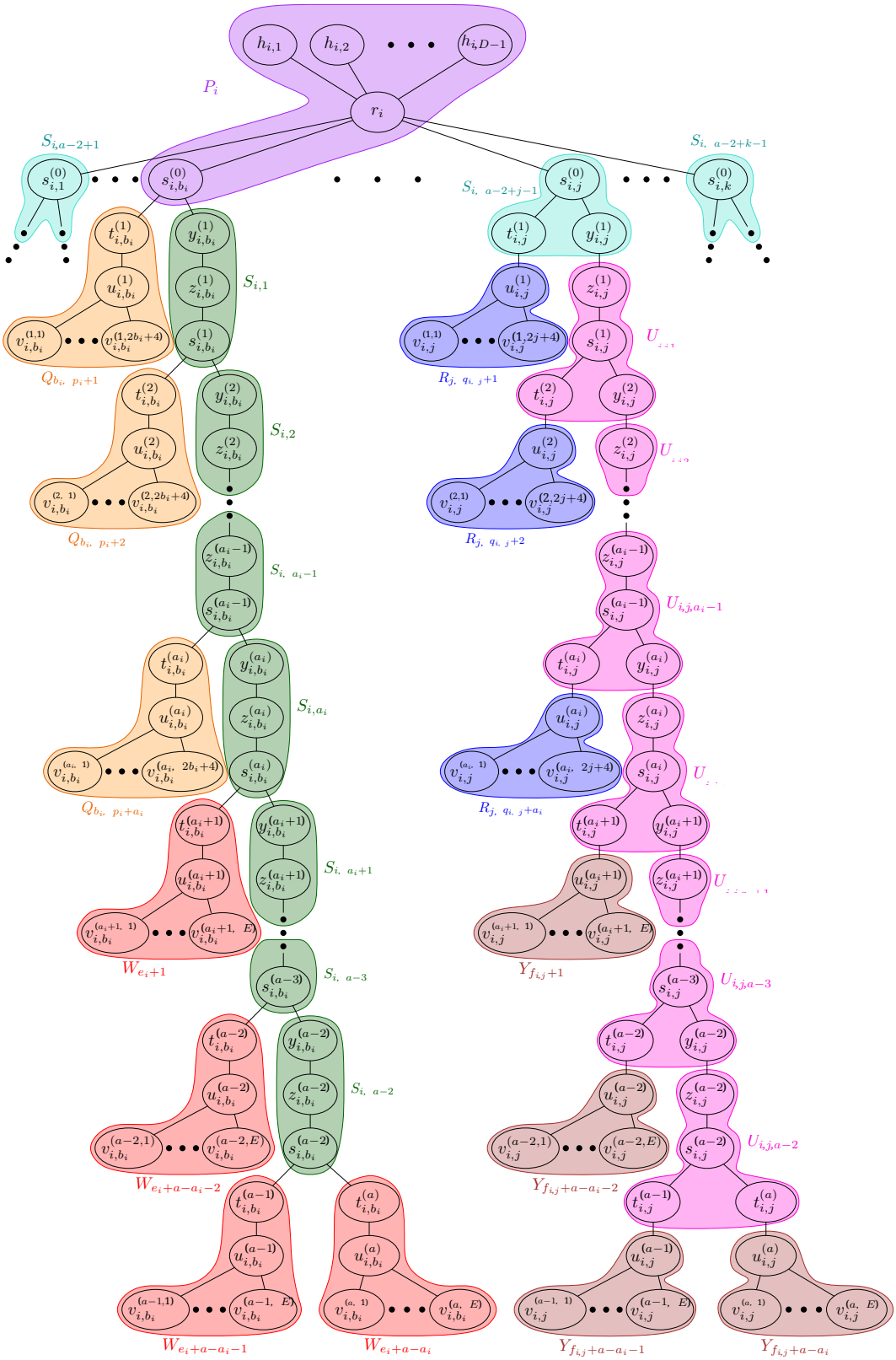


图 5 G_2 的星嵌入到 T_i 中，适用于分支 b_i 和分支 $j \neq b_i$。

这证明了 $\forall i \in [n]\ \exists j \in [n]$，$\alpha_i$,0 在 G_1 中的嵌入是 r_j。不失一般性，我们可以假设 α_i, 0 嵌入在 r_i 中。现在，假设 $\exists i^* \in [n], j \in [D-1]$，使得 $h_{i^*,j}$ 未包含在 P_{i^*} 的嵌入中。由于 $|V(G_2)| = |V(G_1)|$，因此存在 $\exists w^* \in V(G_2 \setminus P_{i^*})$，使得 w^* 嵌入到 $h_{i^*,j}$。由于 $G_2 \setminus P_{i^*}$ 中没有孤立顶点，因此存在 $\exists x^* \in V(G_2 \setminus P_{i^*})$，使得 $\{x^*, w^*\}$ 是 G_2 中的一条边。注意，x^* 未嵌入到 r_i，因此边 $\{x^*, w^*\}$ 不存在于 G_2 在 G_1 中的嵌入中，这就产生了矛盾。

这证明存在一种嵌入，使得 $\forall i \in [n]$, P_i 被嵌入到 T_i 中，其中 r_i 是中心，并且它包含 $h_{i,\,1}$，……，$h_{i,\,D-1}$。由于度($\alpha_{i,\,0}$) $= D$，将 P_i 嵌入 T_i 时会包含 D 个 r_i 的邻居。其中 $D-1$ 个邻居是 $h_{i,\,1}$，……，$h_{i,\,D-1}$。因此，剩余的邻居中恰有一个被包含在内，它属于 $s^{(0)}_{i\,1}$，，…$s^{(0)}_{i\,k}$，；我们将其记为$s^{(0)}_{i\,b_i}$，。◁

▷ 断言 32。存在一个将 G_2 嵌入 G_1 的 P-一致嵌入，使得对于所有 $\ell \in [a-2]$，节点 $z^{(\ell)}_{i\,b_i}$，$y^{(\ell)}_{i\,,\,b_i}$ 和 $s^{(\ell)}_{i\,,\,b_i}$ 嵌入星 $S_{i,\,\ell}$。我们称这种嵌入为 PS*-一致嵌入。

证明。对于一个 P-一致嵌入 ϕ，令 $\ell \in [a-2]$ 为满足以下条件的最小索引：$\exists i \in [n]$ 使得 $S_{i,\,\ell}$ 未被节点 $z^{(\ell)}_{i\,b_i}$，，$y^{(\ell)}_{i\,,\,b_i}$ 和 $s^{(\ell)}_{i\,b_i}$，嵌入。我们称 ℓ 为 ϕ 的第一个违规索引，记作 viol(ϕ)。如果不存在 PS*-一致嵌入，那么每个 P-一致嵌入都有一个第一个违规索引。考虑一个嵌入 $\phi^* \in$，使 ϕ 为 P-一致{viol(ϕ)} be
具有第一个违规索引的最高索引的任意嵌入。令 ℓ^* =违反ϕ^*。由于 ϕ^* 是一个 P-一致嵌入，r_i 嵌入了 $\alpha_{i、\,0}$ $\forall i \in [n]$ （根据断言 31）。此外，根据假设，对于所有 $\ell \in [\ell^* - 1]$，对于所有 $i \in [n]$，节点 $z^{(\ell)}_{i\,b_i}$、$y^{(\ell)}_{i}$、$_{b_i}$ 和 $s^{(\ell)}_{i}$、$_{b_i}$ 嵌入星 $S_{i、\,\ell}$。现在，由于G_2 是 G_1的一个子图，顶点 $y^{(\ell^*)}_{i、\,b_i}$ 必须嵌入来自 G_2的某个顶点 w。$G_1[V(G_1) \setminus (\{r_i \mid i \in [n]\} \cup \{s^{(\ell)}_{i\,,b_i} \mid i \in [n], \ell \in [\ell^*]\})]$ 中 $y^{(\ell^*)}_{i、\,b_i}$ 的唯一邻居是 $z^{(\ell)}_{i\,b_i}$。由于 G_2 中的所有星的中心的度均为 $\geqslant 2$，$y^{(\ell^*)}_{i、\,b_i}$ 无法嵌入一个中心。因此，$z^{(\ell^*)}_{i、\,b_i}$ 嵌入了 G_2中某颗星的中心，该中心以 w 为其叶节点。现在，degree($z^{(\ell^*)}_{i、\,b_i}$) $= 2$，这是 G_2中所有星的中心的最小度。因此，$z^{(\ell^*)}_{i、\,b_i}$ 嵌入了 $\delta_{i'、\,\ell'、}$ 0，来自某个$i' \in [n]$、$\ell' \in [a+k-3]$。由于所有 $S_{i'、\,\ell'}$ 都是同构的，我们不失一般性地假设 $z^{(\ell^*)}_{i、\,b_i}$ 嵌入了 $\delta_{i、\,\ell^*、}$ 0。这意味着 viol(ϕ^*) $> \ell^*$，产生矛盾。◁

▷ 断言 33. 存在一个 G_2 到 G_1 中的 PS*-一致嵌入，使得对于所有$j \in [k] \setminus \{b_i\}$，节点 $s^{(0)}_{i\,,j}$，$t^{(1)}_{i\,,j}$, $y^{(1)}_{i\,,j}$ 嵌入星 $S_{i,j'}$，其中某个 $j \in [a+k-3] \setminus [a-2]$。我们称这样的嵌入为 PS-一致嵌入。

证明. 考虑图 $G_1[V(G_1) \setminus (\{r_i \mid i \in [n]\})]$。这里，$\forall j \in [k] \setminus \{b_i\}$，度($s^{(0)}_{i\,,j}$) $= 2$，并且 $s^{(0)}_{i\,,\,j}$ 的邻居为 $t^{(1)}_{i\,,j}$ 和 $y^{(1)}_{i\,,j}$，二者的度均为 2。现在，由于 G_2 是 G_1的子图，$s^{(0)}_{i\,,j}$ 需要嵌入来自 G_2的一个顶点 w。现在，令 x 为 G_2 中包含 w的星的中心。x 由 $s^{(0)}_{i\,,j}$, $t^{(1)}_{i\,,\,j}$ 和 $y^{(1)}_{i\,,j}$ 中的一个顶点嵌入。若度(x) $\geqslant 3$，则 $\exists w^* \in V(G_2)$，使得 $\{w^*, x\}$ 是 G_2中的一条边，而它未嵌入到 G_1中，这就产生了矛盾。因此，度(x) $= 2$，且 $x = \delta_{i',\ell'}$，其中某个 $i' \in [n], \ell' \in [a+k-3] \setminus [a-2]$。这证明了 $s^{(0)}_{i\,,j}$ 将 G_2中某个 $S_{i',j'}$ 的一个节点嵌入。

注意，有 $n(k-1)$ 个这样的节点 $s^{(0)}_{i\,,\,j}$，它们恰好穷尽了 $n(k-1)$ 个星 $S_{i',\,\ell'}$。由于所有这样的 $S_{i',\,\ell'}$ 都同构，因此不失一般性地，我们可以说 $s^{(0)}_{i\,,\,j}$ 嵌入了一个来自 $S_{i,\,\ell'}$ 的节点。注意，在 $G_1[V(G_1) \setminus (\{r_i \mid i \in [n]\})]$, 中，度($y^{(1)}_{i\,,\,j}$) 为 $= 2$，并且它的两个邻居的度也都是 2。沿用上一段的相同论证，

我们可以看到， $y_i^{(1)}{}_{,j}$ 也嵌入了一个来自 $S_{i',\ell'}$ 的顶点。现在，由于所有 $s_i^{(0)}{}_{,j}$ 恰好穷尽了 $S_{i',\ell'}$，$s_i^{(0)}{}_{,j}$，并且 $y_i^{(1)}{}_{,j}$ 将节点嵌入 G_2 中的同一个星中。这意味着 $\delta_{i,\ell'}$,0 被嵌入 $s_i^{(0)}{}_{,j}$ 或 $y_i^{(1)}{}_{,j}$ 中。如果被嵌入 $s_i^{(0)}{}_{,j}$，那么 $y_i^{(1)}{}_{,j}$ 和 $t_i^{(1)}{}_{,j}$ 将 $S_{i,\ell'}$的叶节点嵌入其中。如果被嵌入 $y_i^{(1)}{}_{,j}$，那么 $s_i^{(0)}{}_{,j}$ 和 $z_i^{(1)}{}_{,j}$ 将 $S_{i,\ell'}$的叶节点嵌入其中。

对于一个 PS*-一致嵌入 ϕ，令 $i \in [n]$ 为满足以下条件的最小索引： $\exists j \in [k] \setminus \{b_i\}, S_{i,\ell'}$ 由节点 $y_i^{(1)}{}_{,j}$，$s_i^{(0)}{}_{,j}$ 和 $z_i^{(1)}{}_{,j}$ 嵌入。我们称 i 为 ϕ的第一个违规索引，记作 viol(ϕ)。如果不存在 PS-一致嵌入，那么每个 PS*-一致嵌入都有一个第一个违规索引。考虑嵌入 $\phi^* \in \arg\max_{\phi \text{ is PS*-consistent}}\{\text{viol}(\phi)\}$，它是具有以下性质的任意嵌入
其第一个违规索引最大。令 $i^* = \text{viol}(\phi^*)$。根据假设，对于所有 $i \in [i^* - 1]$, $\forall j \in [k] \quad b_i \qquad\qquad S_{i\,\ell'} \qquad\qquad \ell' \in [a+k-3] \quad [a-2]$ \ { }，节点 $s_i^{(0)}{}_{,j}$, $y_i^{(1)}{}_{,j}$ 和 $t_i^{(1)}{}_{,j}$ 均可嵌入，其中 \ 。现在，令 $y_{i^*}^{(1)}{}_{,j}$, $s_{i^*}^{(0)}{}_{,j}$ 和 $z_{i^*}^{(1)}{}_{,j}$ 将星 $S_{i^*,\ell}$嵌入。接下来考虑节点 $s_{i^*}^{(1)}{}_{,j}$。由于所有 $S_{i,\ell'}$ 都已被 $s_i^{(0)}{}_{,j}$用尽，节点 $s_{i^*}^{(1)}{}_{,j}$ 会嵌入来自 G_2 中某个星的节点 w，该星至少有 3 个叶节点。令该星的中心为 x。然而，只有 $s_{i^*}^{(1)}{}_{,j}$ 仅有 2 个尚未嵌入的邻居，且这两个邻居的度均为 2。因此，如果其中任何一个将 x, $\exists w^* \in V(G_2)$嵌入，使得 $\{w^*, x\}$ 是 G_2 中一条未嵌入 G_1的边，就会产生矛盾。因此， $s_{i^*}^{(1)}{}_{,j}$ 无法嵌入任何顶点。但这与 G_2是 G_1的子图，以及 $|V(G_2)| = |V(G_1)|$这一事实相矛盾。◁

▷ 命题 34. 存在一个从 G_2 到 G_1 的 PS-一致嵌入，使得对于所有 $j \in [k] \setminus \{b_i\}$、所有 $\ell \in [a-3]$, 节点 $s_i^{(\ell)}{}_{,j}$, $z_i^{(\ell)}{}_{,j}$, $t_i^{(\ell+1)}{}_{,j}$, $y_i^{(\ell+1)}{}_{,j}$ 嵌入为星 $U_{i,j,\ell}$；并且对于 $j \in [k] \setminus \{b_i\}$，节点 $s_i^{(a-2)}{}_{,j}$, $z_i^{(a-2)}{}_{,j}$, $t_i^{(a-1)}{}_{,j}$, $t_i^{(a)}{}_{,j}$ 嵌入为星 $U_{i,j,a-2}$。我们称这样的嵌入为 PSU-一致嵌入。

证明。 若一个 PS-一致嵌入满足：对于 $\forall j \in [k] \setminus \{b_i\}$ $\forall \ell \in [d]$，节点 $s_i^{(\ell)}{}_{,j}$, $z_i^{(\ell)}{}_{,j}$, $t_i^{(\ell+1)}{}_{,j}$, $y_i^{(\ell+1)}{}_{,j}$ 嵌入为星 $U_{i,j,\ell}$，则称其为 PSU$^{(d)}$-一致嵌入。我们通过对 d 使用归纳法来证明该命题。

基本情况： 根据定义，所有 PS-一致嵌入都是 PSU$^{(0)}$-一致嵌入 t.

归纳步骤：考虑一个 PSU$^{(\ell)}$-一致嵌入。顶点 $y_i^{(\ell)}{}_{,j}$ 将星 $U_{i,j,\ell}$中的一个叶节点嵌入其中。现在，考虑顶点 $z_i^{(\ell)}{}_{,j}$。如果该顶点嵌入 G_2中的某个星的中心，则它至少需要 3 个邻居，以便嵌入该星的叶节点。但是，degree($z_i^{(\ell)}{}_{,j}$) $= 2$。因此， $z_i^{(\ell)}{}_{,j}$ 将 G_2中的某个星的叶节点嵌入其中。该星的中心嵌入在 $s_i^{(\ell)}{}_{,j}$中。现在，degree($s_i^{(\ell)}{}_{,j}$) $= 3$，因此 $s_i^{(\ell)}{}_{,j}$只能嵌入某个 i', j', ℓ'对应的 $\epsilon_{i',j',\ell'}$,0。由于所有 $U_{i',j',\ell'}$ 都同构，我们可以不失一般性地说， $s_i^{(\ell)}{}_{,j}$ 嵌入 $\epsilon_{i,j,\ell}$,0。现在， $s_i^{(0)}{}_{,j}$ 有 3 个邻居，它们分别嵌入叶节点$\epsilon_{i,j,\ell}$,1、 $\epsilon_{i,j,\ell}$,2、 $\epsilon_{i,j,\ell}$,3，并将它们嵌入$z_i^{(\ell)}{}_{,j}$, $t_i^{(\ell+1)}{}_{,j}$, $y_i^{(\ell+1)}{}_{,j}$。对所有 i 和 j重复这一论证，我们便得到一个 PSU$^{(l+1)}$-一致嵌入。

这证明了 PSU$^{(a-3)}$-一致嵌入的存在。现在，考虑顶点$z_i^{(a-2)}{}_{,j}$。如果该顶点嵌入了 G_2 中某个星的中心，就至少需要 3 个邻居来嵌入该星的叶节点。但 degree($z_i^{(a-2)}{}_{,j}$) $= 2$。因此， $z_i^{(a-2)}{}_{,j}$嵌入了 G_2中某个星的叶节点。该星的中心嵌入在 $s_i^{(a-2)}{}_{,j}$ 中。现在，degree($s_i^{(a-2)}{}_{,j}$) $= 3$，因此 $s_i^{(a-2)}{}_{,j}$只能嵌入 $\epsilon_{i',j',\ell'}$,0，其中某个 i', j', ℓ'。由于所有 $U_{i',j',\ell'}$ 都同构，我们可以不失一般性地说， $s_i^{(a-2)}{}_{,j}$ 嵌入$\epsilon_{i,j,a-2}$,0。现在， $s_i^{(0)}{}_{,j}$有 3 个邻居，它们分别嵌入叶节点$\epsilon_{i,j,a-2}$,1、 $\epsilon_{i,j,a-2}$,2、 $\epsilon_{i,j,a-2}$,3，而这些叶节点嵌入 $z_i^{(a-2)}{}_{,j}$, $t_i^{(a-1)}{}_{,j}$, $t_i^{(a)}{}_{,j}$。对所有 i 和 j按照这一论证进行推理，我们便得到一个 PSU-一致嵌入。

◁

▷ 断言35. 存在一个 G_2 到 G_1 中的 PSU-一致嵌入，使得对于所有 $\ell \in [a_i]$，$Q_{b_i\ \ell+p_i}$ T_i $_{b_i}$ 以及 $v_i^{(\ell,1)}$. ， $v_i^{(\ell,2)}$. ,, $v^{(\ell,2b_i+4)}$). 和 $t_{\sum_{i'\in[n],i'<i,b_i=b_{i'}}}^{(\ell)}$ $_{b_i}$ $_{b_i}$ $\cdots$ $_{i\ b_i}$ $_{b_i}$. 都被嵌入到 with $u_i^{(\ell)}$. 作为其中心，作为叶节点，其中 $p_i =$ a_i。我们称这种嵌入为 PQSU-一致嵌入。

证明。对于任意 PSU一致嵌入， $X = \{h_{i,j} \mid i \in [n], j \in [D-1]\} \cup \{r_i \mid i \in [n]\} \cup \{s_i^{(\ell)}$, $_j \mid i \in [n], j \in [k] \setminus \{b_i\}\ \ell \in [a-2] \cup \{0\}\} \cup \{y_i^{(\ell)}$ $_{,j} \mid i \in [n], j \in [k] \setminus \{b_i\}\ \ell \in [a-2]\} \cup \{z_i^{(\ell)}$, $_j \mid i \in [n], j \in [k] \setminus \{b_i\}\ \ell \in [a-2]\} \cup \{t_i^{(\ell)}$ $_{,j} \mid i \in [n], j \in [k] \setminus \{b_i\}\ \ell \in [a_i]\}$中的所有顶点都已嵌入所有 P_i, $S_{i,j}$以及$U_{i,j,\ell}$。考虑剩余图 $G_1^R = G_1[V(G_1) \setminus X]$。该图是一个星森林，其中所有星的中心均为某个 $u_i^{(\ell)}$ $_{,j}$ 。现在，由于 G_2 是G_1的子图，$\{Q_{j,\ell}\} \cup \{R_{j,\ell}\} \cup \{W_\ell\} \cup \{Y_\ell\}$中的任何具有 q 个叶节点的星，都必须嵌入G_1^R中具有 q 个叶节点的星。星$Q_{b_i,\ell+p_i}$ 有 $2b_i+5$ 个叶节点，因此必须嵌入G_1^R中具有 $2b_i+5$ 个叶节点的星。然而， G_1^R中所有具有 $2b_i+5$ 个叶节点的星，其中心都在 $u_{i'}^{(\ell)}$ $_{,b_i}$ 中，且$v_{i'\ b_i}^{(\ell,1)}$, , $v_{i'\ b_i}^{(\ell,2)}$, , . . . , $v^{(\ell,2b_i+4)}$) $_{i',b_i}$ 和 $t_{i'}^{(\ell)}$ $_{,b_i}$ ，其中某个 $\ell \in [a_i]$，且 $b_{i'} = b_i$。由于所有这些图都是同构的，我们不失一般性地假设$Q_{b_i,\ell+p_i}$ 嵌入在 G_1中，其中$\beta_{b_i,\ell+p_i}$,0嵌入到 $u_{i\ b_i}^{(\ell)}$, 。◁

▷ 断言36. 对于 G_1中的任意 PQSU一致嵌入 G_2 ，对于每个 $j \in [k]$，令$B_j = \{i \in [n] \mid b_i = j\}$。那么对所有 $j \in [k]$， $\sum a_i = C$.$_{i\in B_j}$

证明。 考虑 G_1中的一个 PQSU一致嵌入 G_2 。令$Q_{j,\ \ell}$ 嵌入某个T_i中。这意味着 $b_i = j$，因为 T_i 只会将$Q_{b_i,\ \ell+p_i}$ 嵌入到 $\ell \in [a_i]$中。因此，

$$\bigcup_{i\in[n],b_i=j} \{Q_{b_i,\ell+p_i} \mid \ell \in [a_i]\} = \{Q_{j,\ell} \mid j = b_i, \ell \in [C]\}$$

注意，左侧是若干两两不相交集合的并（由断言35可知）。因此，令它们的大小相等，可得：

$$\sum_{i\in[n],b_i=j} a_i = C$$

这就证明了该断言。 ◁

这证明了 k-路划分实例是一个是实例。因此，我们证明了即使两个输入图的路径宽度都是常数，MaxComSubG 关于 Δ仍是 W[1]-困难的。◄

定理29还证明了，以两个图的最大度为参数的 MaxComSubG 是 W[1]-困难的，即使这些图的树宽为1也是如此。相比之下，如果将两个图的树宽作为参数，而最大度由常数界定，那么我们可以为 MaxComSubG 设计一个 FPT 算法。

我们的方法基于这样一个观察：任何不含孤立顶点的公共子图都可以分解为一个星森林，其中每颗星都有一个中心顶点，并且可能有若干叶节点。引理 37 中，我们首先设计了一种基于良好树分解的动态规划算法，用于在单个图中寻找最大基数星森林；

30 最大公共顶点子图的参数化复杂性

随后在定理 39 中将其扩展为计算两个图的最大公共子图。从整体上看，袋中的每个动态规划表项都编码了一个部分星森林，该森林由分解中相应子树所处理的顶点诱导的子图构成。对于袋中的每个顶点，我们记录其角色：未使用、叶节点（并记录其中心），或中心（并记录当前连接的叶节点数），从而描述其如何参与部分解。此外，我们还维护一个星形计数向量，用于记录子树内部已完整形成的各尺寸星的数量。

▶**引理 37.** 给定一个具有 n 个顶点、最大度为常数（记作Δ）的图 G，存在一个以树宽 为参数的 FPT 算法，它枚举所有作为 G 的子图出现的星森林 H（在同构意义下唯一），其中 G 的树宽至多为树宽，且 H 的最大度至多为 Δ。该算法的运行时间为 $O\left((\Delta+\text{树宽})^{2(\text{tw}+1)}\cdot\text{树宽}\cdot\Delta\cdot n^{\Delta+1}\cdot\log n\right)$。

证明。令 $V=V(G)$、 $E=E(G)$。令 (T,B) 为 G 的宽度至多为树宽、以 r 为根的良好树分解。对于节点 $t\in V(T)$，令其袋为 $B_t=(v_0,\ldots,v_{k-1})$ $k\leqslant$ 树宽 $+1$，其中袋中的顶点按固定顺序给出。对于节点 $t\in V(T)$，我们用V_t 表示出现在以 t 为根的 T 之子树中各包内的顶点集合，并令$G_t=G[V_t]$。

我们定义函数 $l\colon V(T)\to\mathbb{N}$ 如下。对于根 r，我们设 $l(r)=0$。对于任意节点$t\in V(T)$，我们定义 $l(t)=\text{dist}_T(t,r)$，其中 $\text{dist}_T(t,r)$ 表示树 T中 t与 r之间的距离。令 $L=\max_{t\in V(T)}l(t)$。我们针对 (T,B) 描述一种动态规划算法，该算法按照相对于 l的自底向上方式进行。首先初始化与满足 $l(t)=L$的节点 t对应的所有动态规划状态。然后，对于 $j=L-1,L-2,\ldots,0$，在假设满足 $l(t')>j$的所有节点 t'对应的动态规划状态均已计算完毕的情况下，计算满足 $l(t)=j$的节点 t 对应的动态规划状态。特别地，最后计算根 r 处的动态规划状态（其中 $l(r)=0$）。

状态。 我们维护一个动态规划表 DP，其中状态包含以下几个组成部分：

1. t，树分解 T中的一个节点；2. 一个 掩码 σ，用于描述顶点在 B_t中的角色；3. 一个星形计数向量 $\mathbf{c}=(c_2,c_3,\ldots,c_{\Delta+1})\in[n]^{\Delta}$。

掩码。 对于每个顶点 $v_i\in B_t$，其角色 $\sigma(v_i)$属于以下情况之一：

▷ 未覆盖： v_i 不与部分解中的任何边关联；▷ 中心(d)，对于 $d\in\{2,\ldots,\Delta+1\}$： v_i是一个包含 d个节点的星的中心；▷ 叶节点(u)，对于某个 $u\in B_t$： v_i 是一个星的叶节点，其中心为 u；▷ 叶节点(⊥)： v_i 是一个叶节点，其中心位于 B_t之外。

袋 B_t的掩码是一个元组 $\sigma=(\sigma(v_0)\ldots,\sigma(v_{k-1}))$。

星形计数向量。 对于每个 $d\in\{2,\ldots,\Delta+1\}$，值 c_d表示在 G_t中形成的大小为 d的星的数量。

状态解释。 对于每个节点 $t\in V(T)$、掩码 σ和星形计数向量 $\mathbf{c}$，当且仅当存在一个子图 $H\subseteq G_t$ 满足以下条件时，条目 $\text{DP}[t,\sigma,\mathbf{c}]\in\{0,1\}$被设为 1：

▷ H 是由星构成的顶点不相交并，且每个星的大小至多为 $\Delta+1$；▷ 对于每个 $d\in\{2,\ldots,\Delta+1\}$，其中恰好出现 c_d 个大小为 d 的星；▷ 对于每个顶点 $v_i\in B_t$，其在 H 中的角色恰好由 $\sigma(v_i)$ 给出；▷ 的每条边都是 H 的一条边。

如果不存在这样的子图，我们令 $\mathrm{DP}[t,\sigma,\mathbf{c}]=0$。

初始化（叶节点）。如果 t 是带有袋 $B_t=\emptyset$的 T 的叶节点，则 $\mathrm{DP}[t,(\emptyset),(0,\ldots,0)]=1$。对于其他每个星形计数向量 $\mathbf{c}'\neq(0,\ldots,0)$，我们令 $\mathrm{DP}[t,\emptyset,\mathbf{c}']=0.$

在 DP上进行动态规划。动态规划表沿树 T以自底向上方式填充。对于节点 t，假设与 t 的子节点对应的所有动态规划状态均已计算完成，然后根据 t 的类型（引入、遗忘或合并）更新 $\mathrm{DP}[t,\cdot,\cdot]$。

引入节点。设 t 为一个引入节点，其子节点为 t'，且 $B_t=B_{t'}\cup\{u\}$，其中 $u\notin B_{t'}$。为 B_t固定一个掩码 σ和一个星形计数向量 $\mathbf{c}$。当且仅当以下条件至少有一个成立时，我们令 $\mathrm{DP}[t,\sigma,\mathbf{c}]=1$。

1. **若 u未覆盖**。若 $\sigma(u)=$未覆盖，则 动态规划$[t,\sigma,\mathbf{c}]=1$ 当且仅当存在一个掩码 σ' 对于 $B_{t'}$ 使得 $\sigma'(v)=\sigma(v)$ 对所有 $v\in B_{t'}$ 以及 动态规划$[t',\sigma',\mathbf{c}]=1.$
2. **若 u 是一个叶节点**。假设 $\sigma(u)=$叶节点(v) 对某个 $v\in B_{t'}$ 满足 $\{u,v\}\in E(G)$。则动态规划$[t,\sigma,\mathbf{c}]=1$ 当且仅当存在一个掩码 σ' 和一个向量 $\mathbf{c}'$ 使得
   - ▷ 动态规划$[t',\sigma',\mathbf{c}']=1$,
   - ▷ $\sigma'(w)=\sigma(w)$ 对所有 $w\in B_{t'}\setminus\{v\}$,
   - ▷ 要么
     - $\sigma'(v)=$未覆盖，$c_2=c_2'+1$，并且 $c_j=c_j'$ 对所有 $j\neq 2$，或者
     - $\sigma'(v)=$中心(d)，对于某些 $d\leqslant\Delta$, $c_{d+1}=c_{d+1}'+1$, $c_d=c_d'-1$，以及 $c_j=c_j'$ 对于所有 $j\notin\{d,\ d+1\}$。
3. **若 u是一个中心**。假设 $\sigma(u)=$中心(d) 对于某些 $d\in\{2,\ldots,\ \Delta+1\}$。则动态规划$[t,\sigma,\mathbf{c}]=1$ 当且仅当存在一个掩码 σ' 和一个向量 $\mathbf{c}'$ 使得
   ▷ 动态规划$[t',\sigma',\mathbf{c}']=1$,▷ 令 $S=\{v\in B_{t'}\mid\sigma(v)=$叶节点$(u)\}$，则 S的大小为 $d-1$，其中 $\{u,\ v\}\in E(G)$ 且$\sigma'(v)$ 对所有 $v\in S$,▷ $\sigma(w)=\sigma'(w)$ 均未覆盖，对所有 $w\in B_{t'}\setminus S$,▷ $c_d=c_d'+1$ 以及 $c_j=c_j'$ 对所有 $j\neq d$。

遗忘节点。设 t 是一个遗忘节点，其子节点为 t'，且 $B_t=B_{t'}\setminus\{u\}$。固定一个针对 B_t的掩码 σ和一个星形计数向量 $\mathbf{c}$。当且仅当存在一个针对 $B_{t'}$ 的掩码 σ'，使得 $\mathrm{DP}[t,\sigma,\mathbf{c}]=1$时，我们令其成立：

▷ $\sigma'(v)=\sigma(v)$ 对所有 $v\in B_t$均
成立，并且▷ $\mathrm{DP}[t',\sigma',\mathbf{c}]=1$。

合并节点。设 t 是一个合并节点，其子节点为 t_1 和 t_2，且满足 $B_t=B_{t_1}=B_{t_2}$。在合并节点 t 处，子图 G_t 是两个子图 G_{t_1} 和 G_{t_2} 的并集，它们恰好在袋 B_t 上相交。t_1 处的动态规划状态描述一个部分星森林 $H_1\subseteq G_{t_1}$，而 t_2 处的动态规划状态描述一个部分星森林 $H_2\subseteq G_{t_2}$。为了在 G_t 中获得一个有效解，必须将这两个部分

解合并为 $H = H_1 \cup H_2$。B_t 的顶点是同时出现在两个子图中的唯一顶点，因此也是两个部分解可能发生交互的唯一位置。每个子节点状态都会为每个顶点 $v \in B_t$ 指定一个角色（未覆盖、叶节点，或具有给定大小的中心）。这些角色描述了 v 在该子树内的部分星森林中如何发挥作用。然而，在最终解 H 中，每个顶点在星森林中只能有一个角色：它不能同时是叶节点和中心，也不能连接到两个不同的中心。因此，来自 t_1 和 t_2 的角色分配必须与每个 $v \in B_t$ 的单一全局角色保持一致。接下来我们讨论顶点角色兼容性。

固定掩码 σ、B_t 以及星形计数向量 $\mathbf{c}$。当且仅当存在掩码 σ_1, σ_2、B_{t_1}, B_{t_2} 和星形计数向量 $\mathbf{c}^{(1)}$,$\mathbf{c}^{(2)}$，使得 $\mathrm{DP}[t_1, \sigma_1, \mathbf{c}^{(1)}] = 1$ 和 $\mathrm{DP}[t_2, \sigma_2, \mathbf{c}^{(2)}] = 1$，且满足以下兼容性条件时，我们设 $\mathrm{DP}[t, \sigma, \mathbf{c}] = 1$。

顶点角色兼容性。对于每个顶点 $v \in B_t$，角色对 $(\sigma_1(v), \sigma_2(v))$ 唯一确定 $\sigma(v)$，具体如下。

1. **若 $\sigma_1(v)$ 和 $\sigma_2(v)$ 中有一个未覆盖。**如果一个顶点在某棵子树中未覆盖，该子树就不会在任何星中使用此顶点。因此，在合并后的解中，该顶点的角色完全由另一棵子树决定，因为只有那一侧会为该顶点贡献关联星边。

   a. 如果 $\sigma_1(v)$ 和 $\sigma_2(v)$ 均为未覆盖，则 $\sigma(v)$ =未覆盖。b. 如果其中一个 $\sigma_1(v)$ $\sigma_2(v)$ 未覆盖，另一个为叶节点(u)，则 $\sigma(v)$ = 叶节点(u)。c. 如果其中一个未覆盖，另一个为中心(d)，则 $\sigma(v)$ = 中心(d)。

2. 如果两个角色 $\sigma_1(v)$ $\sigma_2(v)$ 都是叶节点。如果某个顶点在两个子树中都是叶节点，且中心是同一个顶点 u，即 $\sigma_1(v) = \sigma_2(v)$ = 叶节点(u)，那么它必须连接到 B_t 中相同的中心；这意味着两个部分解描述的是同一星形在两个子树中延续的部分，因此该角色仍为 $\sigma(v)$ = 叶节点(u)。如果两个子树将它连接到不同的中心，则角色冲突，无法形成有效的星森林，因此这种组合不兼容。

3. 如果 $\sigma_1(v)$ 和 $\sigma_2(v)$ 都是中心。如果某个顶点 $v \in B_t$ 在两个子状态中都是中心，即 $\sigma_1(v)$ = 中心(d_1) 和 $\sigma_2(v)$ = 中心(d_2)，那么它关联的叶节点可以分布在 G_{t_1} 和 G_{t_2} 中。在组合后的解中，这两个部分星形代表同一星形的不同部分，因此必须合并为一个星。设 $S_1 = \{u \in B_t \mid \sigma_1(u) =$ 叶节点(v)$\}$ 为 B_{t_1} 中该星包含的叶节点，$S_2 = \{u \in B_t \mid \sigma_2(u) =$ 叶节点(v)$\}$ 为 B_{t_2} 中该星包含的叶节点。只有当 σ_1 和 σ_2 满足 $S_1 = S_2$ 时，它们才兼容，即这些星在袋中已有的叶节点上相一致。在这种情况下，设 $q = |S_1|$ 为 v 中已存在于袋内的公共叶节点数量；这些叶节点在两个子树中都会计数，因此不能重复计数。如果两个部分星形的大小分别为 d_1 和 d_2，那么合并后的星大小为 $d = d_1 + d_2 - q - 1$，星形计数向量通过移除两个部分计数并为大小为 d 的合并星增加一个计数来更新。如果 $d \leqslant \Delta + 1$，则设 $\sigma(v)$ = 中心(d)；否则，该组合无效。

4. **如果 $\sigma_1(v)$ 或 $\sigma_2(v)$ 其中一个是叶节点，另一个是中心。**如果一个角色是 叶节点(u)，另一个是 中心(d)，那么这种组合是不相容的，因为在同一个星森林中，一个顶点不能同时充当叶节点和中心。

星形计数更新。对于每个 $d \in \{2, \ldots, \Delta + 1\}$，设 a_d 为满足 $\sigma_1(v)$ = 中心(d) 的顶点 $v \in B_t$的数量，b_d 为满足 $\sigma_2(v)$ = 中心(d) 的顶点数量。令 m_d 表示同时是两棵子树中心，且其两部分星形合并后在合并后的解中形成一个大小为 d 的星的顶点数量。由此可简单地应用

容斥原理，得到结果星形计数向量满足$c_d = c_d^{(1)} + c_d^{(2)} - a_d - b_d + m_d$。我们将展示如何高效计算所有此类 $\mathbf{c}$ 。

我们选取 σ_1, σ_2 ，它们符合顶点角色兼容性中所述的兼容条件。我们令 $\mathsf{C}_1 = \{\mathsf{c} \mid \mathrm{DP}[t_1, \sigma_1, \mathsf{c}] = 1\}$，并类似地令 $\mathsf{C}_2 = \{\mathsf{c}' \mid \mathrm{DP}[t_1, \sigma_1, \mathsf{c}'] = 1\}$。根据引理 44，我们可以在 $O(\Delta n^{\Delta} \log n)$ 时间内计算 $\mathsf{C} = \{\mathsf{c} + \mathsf{c}' \mid \mathsf{c} \in \mathsf{C}_1, \mathsf{c}' \in \mathsf{C}_2\}$。令 $\mathsf{a} = (a_2, a_3, \ldots, a_{\Delta+1})$、$\mathsf{b} = (b_2, b_3, \ldots, b_{\Delta+1})$，且 $\mathsf{m} = (m_2, m_3, \ldots, m_{\Delta+1})$。我们定义 $\mathsf{C}^* = \{\mathsf{c} - \mathsf{a} - \mathsf{b} + \mathsf{m} \mid \mathsf{c} \in \mathsf{C}\}$。若 $\mathsf{c} \in \mathsf{C}^*$，则令 $\mathrm{DP}[t, \sigma, \mathsf{c}]$ 为 1，否则为 0。

令 r 为树分解的根。根据良好树分解的构造，袋 B_r 为空。因此， r 处的任意状态都具有形式 $\mathrm{DP}[r, (\emptyset), \mathbf{c}]$。

由于 $B_r = \emptyset$， H 的所有顶点都严格位于子树内部。因此， H 是 G 中的一个星森林子图。算法输出以下星森林：对于每个$\mathbf{c} = (c_2, \ldots\ldots, c_{\Delta+1}) \in [n]^{\Delta}$，使得 $\mathrm{DP}[r, (\emptyset), \mathbf{c}] = 1$，输出包含以下内容的星森林：对于每个 $i \in \{2, 3, \ldots\ldots, \Delta + 1\}$，其中含有 c_i 个大小为 i 的星。

正确性证明。

我们通过对树分解进行归纳，证明动态规划表满足预期的语义。

▷ 断言 38。对于每个节点 $t \in V(T)$、 B_t 上的掩码 σ 以及星形计数向量 $\mathbf{c}$，当且仅当存在一个子图 $H \subseteq G_t$ 使得 $\mathrm{DP}[t, \sigma, \mathbf{c}] = 1$ 。

1. H 是顶点不相交的星的并，且每个星的大小至多为 $\Delta + 1$，2. 对于每个 $d \in \{2, \ldots, \Delta + 1\}$，大小为 c_d的星恰好有 d个出现在 H中，3. 对于每个顶点 $v \in B_t$，它在 H 中的角色恰好是 $\sigma(v)$，4. 的每条边 H 都属于 G。

证明。我们通过对 $l(t)$进行归纳法来证明该断言，也就是说，按照树分解的自底向上顺序进行。

正确性证明：正向。

基本情况（叶节点）。设 t 是一个叶节点。那么 $B_t = \emptyset$ 和 G_t 是空图。动态规划初始化将 $\mathrm{DP}[t, (\emptyset), (0, \ldots, 0)] = 1$设为相应值，而将所有其他条目设为 0。 G_t 的唯一子图是空图，它显然满足该断言，因为其中没有星，且袋中没有顶点。因此，该结论对叶节点成立。

归纳步骤。假设对于节点 t 的所有子节点，该结论均成立。我们根据 t 的类型证明该结论对其也成立。

引入节点。假设 t 引入了一个顶点 u，即对于其子节点 t'，有 $B_t = B_{t'} \cup \{u\}$ 。任意星森林 $H \subseteq G_t$ 都限制为一个星森林 $H' = H[V_{t'}] \subseteq G_{t'}$。根据归纳假设，这对应于一个状态 动态规划$[t', \sigma', \mathbf{c}'] = 1$。

引入节点的转移恰好考虑了 H 中 u 的三种可能角色：

1。 如果 u 是 未覆盖，那么 $H = H'$ 并且动态规划在$B_{t'}$上保持相同的星数量和角色。

2. 如果 u 是叶节点（v），那么 u 会连接到某个中心 v，并带有 $\{u,v\} \in E(G)$。动态规划转移要么创建一个大小为 2 的新星（如果 v 此前未覆盖），要么扩展一个以 v为中心的现有星，同时更新 c 中对应的条目。3. 如果 u 是中心，那么 u 会成为一棵星的中心，其叶节点从它在 $B_{t'}$中的邻居中选取；动态规划会验证这些顶点此前均未覆盖，并更新星的数量。

因此，每次动态规划转移都会构造从 $G_{t'}$ 到 G_t 的星森林的有效扩展。由此，引入节点的情形下该结论成立。

遗忘节点。设 t 遗忘一个顶点 u，即 $B_t = B_{t'} \setminus \{u\}$。任何星森林 $H \subseteq G_t$ 也是 $G_{t'}$ 中具有相同星形计数的星森林。一旦 $u \notin B_t$，H 中 u 的角色便不再相关。动态规划转移对 u 在子节点状态中的所有可能角色执行“或”操作，同时保留星形计数向量。根据归纳假设，这恰好涵盖了 G_t 中所有有效的星森林。因此，遗忘节点的情形下该结论成立。

合并节点。设 t 是一个合并节点，其子节点为 t_1, t_2 和 $B_t = B_{t_1} = B_{t_2}$。任何星森林 $H \subseteq G_t$ 都可分解为 $H_1 = H \cap G_{t_1}$ 和 $H_2 = H \cap G_{t_2}$，且二者在 B_t 上一致。根据归纳假设，H_1 和 H_2 分别对应状态 $\mathrm{DP}[t_1, \sigma_1, \mathbf{c}^{(1)}] = 1$ 和 $\mathrm{DP}[t_2, \sigma_2, \mathbf{c}^{(2)}] = 1$。顶点角色兼容条件确保袋中顶点的角色在 H_1 和 H_2 之间保持一致。具体而言，中心位于袋中且同时出现在两个子树中的星会被正确合并，而无效组合（例如某个顶点同时作为叶节点和中心）则会被舍弃。星形计数更新会减去两个子树中重复计数且中心位于袋中的星，并加回大小更新后的合并星。因此，向量 $\mathbf{c}$ 恰好记录了 H 中星的数量。

因此，该结论对合并节点成立。根据归纳法，该结论对根 r 成立。因此，动态规划能够正确判定是否存在具有所需大小的星森林。

正确性证明：反向。

假设一个星森林 $H \subseteq G_t$ 满足性质 (1)–(4)。我们证明动态规划 $[t, \sigma, \mathbf{c}] = 1$。

基本情况（叶节点）。如果 t 是叶节点，则 H 必须为空，因此动态规划初始化将 $\mathrm{DP}[t, (\emptyset), (0, \ldots\ldots, 0)] = 1$ 设为相应值。

归纳步骤。假设该结论对子节点 t 的所有子节点都成立。我们根据 t 的类型证明该结论对其成立。

引入节点。设 t 引入 u。令 $H' = H[V_{t'}]$。根据归纳法，H' 对应于 $\mathrm{DP}[t', \sigma', \mathbf{c}'] = 1$。$u$ 在 H 中的作用决定适用哪种引入转移，从而更新角色和计数至 σ，$\mathbf{c}$。因此，$\mathrm{DP}[t, \sigma, \mathbf{c}] = 1$。

遗忘节点。如果 t 遗忘了 u，那么 H 也是 $G_{t'}$中的星森林。根据归纳法，它对应于动态规划$[t', \sigma', \mathbf{c}] = 1$。遗忘转移将 σ'投影到 B_t，因此动态规划$[t, \sigma, \mathbf{c}] = 1$。

合并节点。设 $H_i = H \cap G_{t_i}$对应于 $i = 1, 2$。根据归纳法，它们对应于状态动态规划 $[t_i, \sigma_i, \mathbf{c}^{(i)}] = 1$。由于它们都源自同一个 H，因此它们在 B_t上的角色相容。合并转移将它们合并，并设置动态规划$[t, \sigma, \mathbf{c}] = 1$。

因此，两个方向均成立，证明完毕。 ◁

运行时间。令 $k = \mathsf{tw} + 1$ 表示最大袋大小。对于袋中的每个顶点 v，其角色可以是未覆盖、Centre(d)（其中 $d \in \{2、\ldots\ldots、\Delta + 1\}$），或 Leaf($u$)

for $u \in B_t \setminus \{v\}$ 或 Leaf(⊥)。因此，每个顶点至多有 $1 + \Delta + (k+1) = \Delta + \text{tw} + 3$种可能的角色，每个袋的掩码数量至多为 $(\Delta + \text{tw} + 3)^{\text{tw}+1}$。星计数向量有 Δ 个坐标，每个坐标均不超过 n，因此可能的向量数量为 $O(n^{\Delta})$。所以，每个袋的动态规划状态数量为 $(\Delta + \text{tw} + 3)^{\text{tw}+1} \cdot n^{\Delta}$. Introduce 节点和遗忘节点可以在每个状态的多项式时间（关于 (tw + Δ)）内处理。在合并节点处，最坏情况下，必须将一个子节点的每个状态与另一个子节点的每个状态进行检查，因此该节点的处理时间为 $(\Delta + \text{tw} + 3)^{2(\text{tw}+1)} \cdot \Delta \cdot n^{\Delta} \log n$。由于良好树分解具有 $O(n \cdot \text{tw})$ 个节点，该算法的总运行时间为 $O\big((\Delta + \text{tw} + 3)^{2(\text{tw}+1)} \cdot \text{tw} \cdot \Delta \cdot n^{\Delta+1} \cdot \log n\big)$。◀

现在，我们扩展引理 37 的动态规划算法，以计算两个图的最大公共子图，且该子图不含孤立顶点。

▶**定理 39**。考虑输入为 G_1、G_2、h 的 MaxComSubG 问题，参数为 树宽，其中两个图的树宽均至多为 树宽，且其中一个图的最大度至多为某个常数 Δ。存在一种FPT 算法可求解 MaxComSubG，其运行时间为 $O\big((\Delta + \text{树宽})^{2(\text{tw}+1)} \cdot \text{树宽} \cdot \Delta \cdot n^{\Delta+1} \cdot \log n\big)$，其中 $n = |V(G_1) + V(G_2)|$。

证明。设 H 是 G_1 和 G_2的最大公共星森林子图，则 H 的最大度为 Δ。利用引理 37，我们枚举 G_1 和 G_2 中最大度为 Δ的所有星森林子图，并输出同时作为两者子图出现的最大星森林的大小。◀

5 有界度平面图的 EPTAS

本节研究 MaxComSubG 问题，其中输入图是最大度受某个常数限制的平面图。尽管当两个图都是有界度（由常数 $\Delta = O(1)$ 限制）的平面图时，MaxComSubG 是 NP-困难的（定理 17），我们证明该问题对于此类输入存在 EPTAS。该算法是对 Baker 技术 [4] 的改进：它牺牲最优解的一小部分，将图分解为 k-外平面图（其中 k 较小），然后精确求解该问题。

▶ **定理 40.** 对于每个 $0 < \varepsilon < 1$，都存在一个$(1-\varepsilon)$-近似算法，用于在最大度不超过 Δ 的平面图上求解 MaxComSubG，其运行时间为 $(1/\varepsilon + \Delta)^{O(1/\varepsilon)} \cdot n^{\Delta+1} \cdot \log n$。

证明。我们先给出算法，然后证明其正确性。

近似算法。

设 $0 < \varepsilon < 1$，输入图为 G_1, G_2。考虑 G_1 的一棵广度优先搜索树$\mathcal{T}$。设 Δ 为这些图中的最大度。设 $L_1^{(i)} \subseteq V(G_1)$ 为$\mathcal{T}$中的第 i 层顶点。同样，设 $L_2^{(i)}$ 为 G_2 的某棵广度优先搜索树中的第 i 层顶点。设 $k = \lceil 2/\varepsilon \rceil$。

算法遍历 $r_1, r_2 \in \{0, 1, \ldots, k-1\}$。算法构造图 $G^{(r_1)}{}_1$，该图由 G_1 删除位于层 $L^{(3r_1+2)}{}_1, L^{(3(k+r_1)+2)}{}_1, L^{(3(2k+r_1)+2)}{}_1, L^{(3(3k+r_1)+2)}{}_1, \ldots$中的顶点而得到。类似地，它构造图$G^{(r_2)}{}_2$，该图由 G_2 删除位于层

$L^{(3r_2+2)}$ 2 , $L^{(3(k+r_2)+2)}$ 2 , $L^{(3(2k+r_2)+2)}$ 2 , $L^{(3(3k+r_2)+2)}$ 2 , . . .。 $G^{(r_1)}$ 1 与 $G^{(r_2)}$ 2 是 $3k$-外平面图，且树宽至多为 $9k+1$（由命题 5）。算法现在在 $G^{(r_1)}$ 1 , $G^{(r_2)}$ 1 上求解 MaxComSubG，时间为 O $\left((\Delta+(9k+1))^{2(9k+2)}\cdot(9k+1)\cdot\ \Delta\cdot\ n^{\Delta+1}\cdot\log\ n\right)$（使用定理 39）。算法输出在 r_1, r_2 的所有选择中节点数最大的解。该算法的总运行时间为$k^2\cdot$ O $\left((\Delta+(9k+1))^{2(9k+2)}\cdot(9k+1)\cdot\ \Delta\cdot\ n^{\Delta+1}\cdot\log n\right)=(1/\varepsilon+\Delta)^{\mathcal{O}(1/\varepsilon)}\cdot\ n^{\Delta+1}\cdot\log n$。

正确性：近似因子的保证

设 OPT 为 H of G_1 andG_2的最大公共星森林子图的节点数。设 SOL 为算法输出中的节点数。显然，SOL <OPT，因为该算法在 G_1 和 G_2的一个子图上寻找最大公共子图。

设 S 是 G_1 的子图 H_1 中的一个星，且该星与 H同构；因此， S 是 G_1的星子图。存在 i，使得 $V(S)\subseteq L_1^{(i)}\cup L_1^{(i+1)}\cup L_1^{(i+2)}$，因为一个星不可能跨越四层或更多层。因此，至多有一个 $r_1\in\{0,1,\ldots,k-1\}$使得 $G^{(r_1)_1}$不包含 S的所有顶点；在这种情况下，我们称 r_1 刺入 S。令 s_{r_1} 表示被 r_1刺入的 H_1 中所有星的顶点总数。令 $r_1^*\in\arg\min_{r_1\in\{0,\ldots,k-1\}}\{s_{r_1}\}$ 使数量最小
中星的节点数被 H_1 刺中。令 F_1 为 H 中与被 r_1^* 刺中的星对应的星森林，则 $|F_1|\leqslant\frac{1}{k}\cdot|V(H)|$。此外， $H\setminus F_1$ 是 $G^{(r_1^*)_1}$ 的子图。类似地，存在 $r_2^*\in\{0,\ldots,k-1\}$ 以及 H 中的星森林 F_2，使得 $|F_2|\leqslant\frac{1}{k}\cdot|V(H)|$，且 $H\setminus F_2$ 是 $G^{(r_2^*)_2}$ 的子图。因此， $\bar{H}\setminus(F_1\cup F_2)$ 是 $G^{(r_1^*)_1}$ 和 $G^{(r_2^*)_2}$ 的公共星森林子图。因此，我们有

$$\begin{aligned}\mathsf{SOL}&\geqslant|V(H)\setminus(V(F_1)\cup V(F_2))|\geqslant|V(H)|-|V(F_1)|-|V(F_2)|\\&\geqslant\left(1-\frac{2}{k}\right)\mathsf{OPT}\geqslant(1-\varepsilon)\mathsf{OPT}\end{aligned}$$

◀

6 结论

本文针对 MaxCom-SubG 的参数化复杂性给出了完整的二分定理，其中参数是公共子图大小，或一个或两个输入图的顶点覆盖、最大度、树深度、路径宽度或树宽的组合。未来，研究如何为具有参数化困难性结果的问题设计高效近似方案将十分有趣——类似于我们针对两个输入图均为最大度有界常数的平面图时所设计的 EPTAS。从计算社会选择的角度来看，我们也希望研究输入图来自特殊图类（如单位圆盘图等）时的问题。

参考文献

1 Tatsuya Akutsu、Takeyuki Tamura、Avraham A Melkman 和 Atsuhiro Takasu。关于寻找有界度最大公共子树的复杂性。《理论计算机科学》，590:2–16，2015年。

2 Noga Alon、Raphael Yuster 和 Uri Zwick。颜色编码。美国计算机协会期刊，42(4):844–856，1995年。3 George E Andrews。分拆理论。第2卷。剑桥大学出版社，1998年。4 Brenda S Baker。平面图上NP完全问题的近似算法。美国计算机协会期刊，41(1):153–180，1994年。5 Karl Bringmann。一种用于子集和的近线性伪多项式时间算法。载于第二十八届ACM-SIAM离散算法研讨会论文集，第1073–1084页。工业与应用数学学会，2017年。6 Thomas H Cormen、Charles E Leiserson、Ronald L Rivest 和 Clifford Stein。算法导论。麻省理工学院出版社，2022年。7 Marek Cygan、Fedor V Fomin、Lukasz Kowalik、Daniel Lokshtanov、Dániel Marx、Marcin Pilipczuk、Michał Pilipczuk 和 Saket Saurabh。参数化算法，第5卷。施普林格，2015年。8 Rodney G Downey 和 Michael Ralph Fellows。参数化复杂性。施普林格科学与商业媒体，2012年。9 Jan Dreier、Janosch Fuchs、Tim A Hartmann、Philipp Kuinke、Peter Rossmanith、Bjoern Tauer 和 Hung-Lung Wang。边不相交路径打包的复杂性。载于第14届参数化与精确计算国际研讨会，2019年。10 Nick Fischer。和集、3SUM、子集和：如今终于有了真正的进展！载于2025年ACM-SIAM离散算法年度研讨会（离散算法研讨会）论文集，第4520–4546页。工业与应用数学学会，2025年。11 Jörg Flum 和 Martin Grohe。参数化复杂性理论。施普林格，2006年。12 Tibor Gallai。关于极值点集和边集，《罗兰·厄特沃什命名的布达佩斯大学科学学报》。数学分辑，2:133–138，1959年。13 Michael R Garey 和 David S Johnson。计算机与难解性，第29卷。W. H. Freeman，纽约，2002年。14 Klaus Heeger、Danny Hermelin 和 Dvir Shabtay。截止期限较少的单机调度。载于第18届参数化与精确计算国际研讨会（参数化与精确计算国际研讨会2023），第24–1页。达格施图尔城堡—莱布尼茨信息学中心，2023年。15 Ruth Hoffmann、Ciaran McCreesh 和 Craig Reilly。介于子图同构与最大公共子图之间。载于AAAI人工智能会议论文集，第31卷，2017年。16 Hadi Hosseini、Andrew McGregor、Justin Payan、Rik Sengupta、Rohit Vaish 和 Vignesh Viswanathan。具有相同估值的图式房屋分配。《自主智能体与多智能体系统》，38(2):42，2024年。17 Hadi Hosseini、Andrew McGregor、Rik Sengupta、Rohit Vaish 和 Vignesh Viswanathan。图式房屋分配的紧确近似。arXiv预印本 arXiv:2307.12482，2023年。18 Hadi Hosseini、Justin Payan、Rik Sengupta、Rohit Vaish 和 Vignesh Viswanathan。图式房屋分配。arXiv预印本 arXiv:2301.01323，2023年。19 Zhihua Huang、An Zhang、Mingqi Gao、Jiayi Sun 和 Yong Chen。用于k+-星打包问题的近似算法。《运筹学快报》，59:107249，2025年。20 Klaus Jansen、Stefan Kratsch、Dániel Marx 和 Ildikó Schlotter。重新审视固定数量的箱子的装箱问题。《计算机与系统科学杂志》，79(1):39–49，2013年。21 Ce Jin 和 Hongxun Wu。一种简单的近线性伪多项式时间子集和随机化算法。载于第二届算法简洁性研讨会（算法简洁性研讨会2019），第17–1页。达格施图尔城堡—莱布尼茨信息学中心，2019年。22 Viggo Kann。NP完全优化问题的可近似性。博士论文，斯德哥尔摩皇家理工学院，1992年。23 Sanjeev Khanna、Rajeev Motwani 和 F Frances Yao。最大公共子树问题的近似算法。斯坦福大学计算机科学系，1995年。24 Yasuaki Kobayashi 和 Hisao Tamaki。以顶点覆盖数为参数的树深度。载于第11届参数化与精确计算国际研讨会（参数化与精确计算国际研讨会2016），第18–1页。达格施图尔城堡—莱布尼茨信息学中心，2017年。

25 Dénes König. 关于图及其在行列式理论和集合论中的应用。《数学年刊》，77(4):453–465，1916。26 Nils Kriege、Florian Kurpicz 和 Petra Mutzel。串并联图中的最大公共子图问题。《欧洲组合学杂志》，68:79–95，2018。27 Elena Prieto 和 Christian Sloper。仰望星空。《理论计算机科学》，351(3):437–445，2006。28 Douglas Brent West 等。《图论导论》，第 2 卷。普伦蒂斯·霍尔，上萨德尔河，2001。29 David P Williamson 和 David B Shmoys。《近似算法设计》。剑桥大学出版社，2011。30 Wenying Xi 和 Wensong Lin。关于无爪次立方图中的最大P_3-打包。《组合优化杂志》，41(3):694–709，2021。31 Wenying Xi 和 Wensong Lin。无爪立方图中的最大 3 星打包问题。《组合优化杂志》，47(5):73，2024。32 Atsuko Yamaguchi、Kiyoko F Aoki 和 Hiroshi Mamitsuka。求部分 k-树与生成树数量多项式有界的图的最大公共子图。《信息处理快报》，92(2):57–63，2004。

A 扩展预备知识

我们首先介绍以下枚举给定整数 h 的划分 $P(h)$ 的算法。

▶**引理 41.** 存在一种算法，可以在时间 $2^{\mathcal{O}(\sqrt{h})}$ 内枚举整数 h 的所有划分。

证明。 考虑算法 1 中描述的递归过程。

算法 1 **输入：** h **输出：** $P(h)$

```
过程 划分(多重集 S，整数 h):
    输出 S ∪ {h − ∑(S)}
    对于 i 从 max(S ∪ {1}) 到 (h − ∑(S))/2 执行
        运行划分(S ∪ {i}, h)
    结束循环
结束过程
运行 Partition(∅, h)
```

算法 1 唯一地输出所有划分，并且由于算法的任何分支都不会遇到死路，因此递归树恰好有 $P(h)$ 个叶节点。此外，递归树的高度至多为 h，每次执行耗时 $\mathcal{O}(h)$。因此，整个算法会在 $\mathcal{O}(h^2 \cdot 2^{\mathcal{O}(\sqrt{h})}) = 2^{\mathcal{O}(\sqrt{h})}$ 时间内终止。◀ _

接下来，我们给出关于图的最小支配集的以下结果。

接下来，我们给出关于图的最小支配集的以下结果。

▶**引理 42。** 对于一个没有孤立顶点的图 G，如果 D 是一个最小支配集，那么存在一个匹配 M，覆盖 D，且其中的边分别连接 D 中的一个顶点和 D 外的另一个顶点。

证明。 令 D 为 G的一个最小支配集。考虑由以下边导出的二分图 G'：每条边的一个端点位于 D 中，另一个端点位于 D之外。

首先，我们证明 G' 没有孤立顶点。 D 之外的顶点在 D中至少有一个邻居，因此它们不可能是孤立顶点。现在，设 $v \in D$ 是G'中的一个孤立顶点。 v 在 G 中不是孤立的（因为 G 起初没有孤立顶点）。因此， $N_G(v) \subset D$。所以， $D \setminus \{v\}$ 也是 G的一个支配集，这与 D的极小性矛盾。

现在， D 必定是 G'的最小支配集，因为 G' 的任意支配集也是 G的支配集。此外， D 是 G'的顶点覆盖。由于任意顶点覆盖都是支配集， D 必定是 G'的最小顶点覆盖。根据 Kőnig 定理， [25], 中 G' 的最大匹配 M 的基数为 $|D|$，因此它覆盖了 D。◀

下一个结果是一种广为人知的技术，它利用快速多项式乘法算法求和集，即两个集合中元素两两相加所得的所有和组成的集合。文献中广泛讨论并使用了这类结果 [5, 21, 10]。我们给出著名的快速傅里叶变换（FFT）算法在多项式乘法中的一个特殊应用 [6]。

▶ **命题 43** (FFT)。 给定两个系数属于 $\{0, 1\}$ 且度至多为 t 的多项式 $f(x)$ $g(x)$，存在一种算法，可在 $O(t \log t)$ 时间内计算 $f(x) \cdot g(x)$。

利用这一结果，我们可以得到以下结论。

▶ **引理 44**。 给定两个集合 $A, B \subseteq \{0, 1, \ldots, n\}^d$，存在一种算法，可在 $O(dn^d \log n)$ 时间内计算

$$A \oplus B = \{(a_1 + b_1, a_2 + b_2, \ldots, a_d + b_d) \mid (a_1, b_2 \ldots, a_d) \in A, (b_1, b_2, \ldots, b_d) \in B\}$$

证明。 对于 $\mathbf{s} = (s_1, \ldots, s_d) \in \{0,1,\ldots,2n\}^d$，令 $\gamma_{\mathbf{s}} = \sum_{i \in [d]} s_i \cdot (2n+1)^{i-1}$。

▷ 断言 45. 对于 一个 $= (a_1, \ldots, a_d)$，$\mathbf{b} = (b_1, \ldots, b_d) \in \{0, 1, \ldots, n\}^d$ 且 $\mathbf{c} = (c_1, \ldots, c_d) \in \{0, 1, \ldots, 2n\}^d$，$\gamma_{\mathbf{a}} + \gamma_{\mathbf{b}} = \gamma_{\mathbf{c}}$ 当且仅当 $a_i + b_i = c_i$ 对于每个 $i \in [d]$。

证明。 如果 $a_i + b_i = c_i$ 对于每个 $i \in [d]$，那么

$$\gamma_{\mathbf{a}} + \gamma_{\mathbf{b}} = \sum_{i \in [d]} a_i \cdot (2n+1)^{i-1} + \sum_{i \in [d]} b_i \cdot (2n+1)^{i-1} = \sum_{i \in [d]} (a_i + b_i) \cdot (2n+1)^{i-1} = \gamma_{\mathbf{c}}$$

反过来，假设 $\gamma_{\mathbf{a}} + \gamma_{\mathbf{b}} = \gamma_{\mathbf{c}} = \gamma$。对于所有 $i \in [d]$，$(a_i + b_i)$ $c_i \leqslant 2n$，因此$\gamma <$ $(2n+1)^d$。由于 $\gamma = \sum_{i \in [d]} \beta_i (2n+1)^{i-1}$ 对于 $\beta_i \in \{0, 1, \ldots, 2n\}$，$i \in [d]$ 是唯一的，因此我们必须对每个 $i \in [d]$ 都有 $a_i + b_i = \beta_i = c_i$。 ◁

考虑多项式 $p(x) = \sum_{\mathbf{a} \in A} x^{\gamma_{\mathbf{a}}}$ 和 $q(x) = \sum_{\mathbf{b} \in B} x^{\gamma_{\mathbf{b}}}$。 $p(x)$ 和 $q(x)$ 是系数属于 $\{0, 1\}$ 的多项式，且度至多为 $(2n+1)^d$。根据断言 45，对于 $\mathbf{c} \in \{0, 1, \ldots, 2n\}^d$，$x^{\gamma_{\mathbf{c}}}$ 在 $p(x) \cdot q(x)$ 中有一个非零系数，当且仅当 $\mathbf{c} \in A \oplus B$。因此，可以通过先在 $O((n^d) \log(n^d)) = O(dn^d \log n)$ 时间内求出 $p(x) \cdot q(x)$（命题 43），再取所有满足 x^i 在 $p(x) \cdot q(x)$ 中有非零系数的 i，来构造 $A \oplus B$。◀