arXiv:2304.03233v1 [cs.DS] 6 Apr 2023

偏心距最短路径问题的参数化算法*

斯里拉姆·Bhyravarapu[1], 萨蒂亚布拉塔·贾纳[1],劳奎因·卡内什[2],萨凯特·萨拉布[1,3],和沙伊莉·维尔马[1]

[1] 印度金奈 HBNI 数学科学研究所 {sriram, satyabrataj,saket,shailyverma}@imsc.res.in[2] 印度焦特布尔印度理工学院 lawqueen@iitj.ac.in[3] 挪威卑尔根大学

摘要。 给定一个无向图 $G = (V, E)$ 和一个整数 ℓ，偏心距最短路径（ESP）问题要求判断是否存在一条最短路径 P，使得对于每个顶点 $v \in V(G)$，都存在一个顶点 $w \in P$，满足 $d_G(v, w) \leq \ell$，其中 $d_G(v, w)$ 表示 G 中 v 与 w 之间的距离。Dragan 和 Leitert [Theor. Comput. Sci. 2017]研究了该问题的优化版本，即求 ESP 中最小的 ℓ，并证明即使在最大度为 3 的平面二分图上，该问题仍为 NP 难。他们还证明，以 ℓ 为参数时，ESP 是 W[2] 难的。另一方面，Kučera 和 Suchý [IWOCA 2021] 证明，以模块宽度、簇顶点删除集、最大叶数，或不相交路径删除集与 ℓ 的组合作为参数时，ESP 是固定参数可解（FPT）的。该论文还提出一个开放问题：以不相交路径删除集或反馈顶点集为参数时，ESP 是否为 FPT。我们回答了这些问题，并得到了以下结果：

1. ESP 以不相交路径删除集、分裂顶点删除集，或反馈顶点集与 ℓ 的组合参数化时属于 FPT。 2. 以反馈顶点集数量为参数时，存在一个 $(1 + \epsilon)$-因子 FPT 近似算法。 3. ESP 以弦图顶点删除集为参数时是 W[2]-困难的。

关键词： 最短路径，离心率，弦图，分裂图，反馈顶点集，FPT，W[2]-困难性

1 引言

给定图 $G = (V, E)$ 和路径 P，顶点 $v \in V(G)$ 到P的距离 为 $\min\{d_G(v, w) \mid w \in V(P)\}$，其中 $d_G(v, w)$ 是 v与 w之间的距离

*本文的部分研究成果已获准在第 34 届组合算法国际研讨会（IWOCA 2023）上报告

在 G 中。给定图 G 和路径 P，相对于 G 而言， P的离心率记作 $\text{ecc}_G(P)$，定义为 G的每个顶点到 P的最短距离中的最大值。形式化地，$\text{ecc}_G(P) = \max\{d_G(u,P)|u \in V(G)\}$。Dragan 和 Leitert [8] 提出了在给定无向图中寻找离心率最小的最短路径这一问题，称为最小离心率最短路径问题（简称 MESP）。他们发现 MESP 与最小畸变嵌入问题之间存在有趣的联系，并为最小畸变嵌入问题提出了更优的近似算法。MESP 可视为支配路径问题 [9] 的推广；支配路径问题要求找到一条路径，使得图中的每个顶点要么属于该路径，要么与路径上的某个顶点相邻。在 MESP 中，目标是在 G 中找到一条最短路径 P，使得 P 的离心率最小。本文中，我们将 G 中所有最短路径的离心率的最小值称为图 G 的离心率，记作 $\text{ecc}(G)$。MESP 在交通规划、流体运输、水资源管理和通信网络中均有应用。

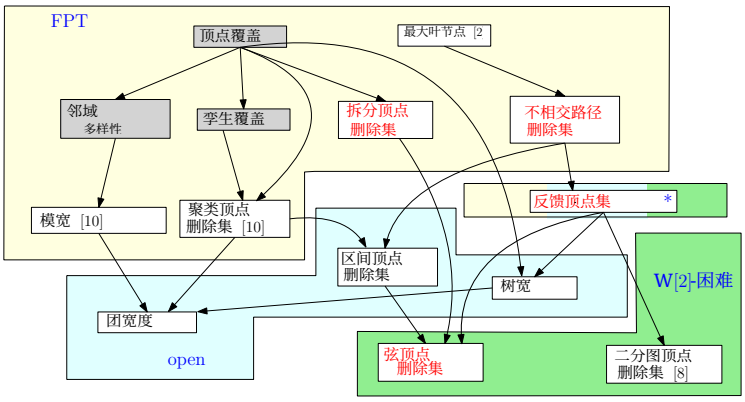

图 1. 本文研究的参数层次结构。箭头从参数 a 指向参数 b 表示 $b \leq f(a)$，其中某个可计算函数为 f。红色参数是本文研究的对象。附加在反馈顶点集上的符号“$*$”表示它与所需离心率结合时是固定参数可处理的。灰色方框表示由已获得的结果所蕴含的结论。

Dragan 和 Leitert [7] 证明，针对 MESP 的快速算法可以导出针对最小线畸变问题的快速近似算法，而特殊图类中低离心率最短路径的存在，则意味着这些图类具有较低的近似界。他们还证明，MESP 在最大度为 3 的平面二分图上是 NP-hard 的。在参数化情形下，他们证明 MESP 对一般图是 W[2]-困难的，并给出了以离心率为参数的 XP 算法。此外，他们设计了运行时间分别为 $O(n^3)$、$O(nm)$ 和 $O(m)$ 的 2-近似、3-近似和 8-近似算法来求解 MESP，其中

n 和 m 分别表示图的顶点数和边数。后一个 8-近似算法采用了双 BFS 技术。2016 年，Birmelé 等人 [1] 通过对双 BFS 过程进行更深入的分析，证明该算法实际上是一个 5-近似算法，并进一步拓展这一思路，得到一个仍以线性时间运行的 3-近似算法。此外，他们研究了 MESP 与 Volké 等人提出的图的层状性之间的联系。层状性 [11] 要求覆盖路径为直径，并据此建立了 MESP 与层状性参数之间的一些紧确界。Dragan 和 Leitert [7]证明，MESP 在距离遗传图上可在线性时间内求解，在弦图和对偶弦图上可在多项式时间内求解。最近，Kučera 和 Suchý [10] 针对一些结构参数研究了 MESP，并给出了以模块宽度、簇顶点删除数（clvd）、最大叶数，或联合参数不相交路径删除数（dpd）和离心率（ecc）为参数的 FPT 算法。我们将 MESP 的判定版本称为离心率最短路径问题（简称 ESP）：该问题要判断是否存在一条最短路径 P，使得对于每个 $v \in V(G)$，v 与 P 之间的距离至多为ℓ。本文进一步拓展了对参数化情形下 MESP 的研究。

1.1 我们的结果与讨论

本文研究 ESP 关于以下结构参数的参数化复杂性：反馈顶点集（fvs）、不相交路径删除集（dpd）、分裂顶点删除集（svd）和弦图顶点删除集（cvd）。我们将该版本称为 ESP/ρ，其中 ρ 是参数。现在，我们正式定义 ESP/fvs + ecc（其他问题也可以类似定义）。

ESP/fvs + ecc **参数：** $k + \ell$**输入：** **</style}一个无向图、一个大小为 的集合(), 满足 是森林，以及一个整数 。问题：** 是否存在 G 中的一条最短路径 P，使得对于每个 $v \in V(G)$，$dist_G(v, P) \leq \ell$?

首先，我们在第3节展示了一个用于 ESP/fvs +离心率的算法，该算法在 $2^{\mathcal{O}(k \log k)} \ell^k n^{\mathcal{O}(1)}$ 时间内运行，其中 ℓ 是图的离心率，k 是反馈顶点集的大小。在第4节，我们设计了一个 $(1+\epsilon)$-因子 FPT 算法用于 ESP/fvs。然后，在第5节，我们设计了一个用于 ESP/dpd 的算法，其运行时间为 $2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$。

此外，我们证明 ESP/svd 存在一个 FPT 算法。随后，我们探讨cvd 上的问题；fvs、svd、clvd 的推广，并证明 ESP/cvd 是 $W[2]$-难的。这些结果分别在第 6 节和第7节中给出。

2 预备知识

本文所讨论的所有图都是有限的、无权的、无向的且连通的。关于标准图论符号，请参阅图论教材。

R. Diestel 的 [5]。关于参数化复杂度术语，我们参阅 Cygan 等人编著的参数化算法一书 [3]。对于 $n \in \mathbb{N}$，我们分别用 $[n]$ 和 $[0, n]$ 表示集合 $\{1, 2, \cdots, n\}$ 和 $\{0, 1, 2, \cdots, n\}$。对于图 $G = (V, E)$，我们用 n 和 m 表示其顶点数和边数。给定整数 ℓ，若存在一个顶点 $u \in V(P)$，使顶点 u 和 v 之间的距离（记为 $d_G(v, u)$）至多为 ℓ，则称路径 P 覆盖顶点 v。图 G 的反馈顶点集是一个集合 $S \subseteq V(G)$，使得 $G - S$ 是无环的。

除反馈顶点集外，我们还考虑了以下结构参数：

1. 不相交路径删除集 (dpd)是一个集合 $S \subseteq V(G)$，使得 $G - S$ 是一组互不相交的路径。2. 分裂顶点删除集 是一个集合 $S \subseteq V(G)$，使得 $G - S$是一个分裂图，其中分裂图 是一种图，其顶点 G 可以划分为两个集合：一个诱导出独立集，另一个诱导出团。3. 弦顶点删除集 是一个集合 $S \subseteq V(G)$，使得 $G - S$是一个弦图；弦图是不含长度至少为 4 的诱导环的图。

给定一个图 G，可以在以 k为参数的 FPT 时间内，找到大小为 k 的反馈顶点集 [3], 、不相交路径删除集 [10],、分裂顶点删除集 [4], 和弦顶点删除集 [2], 。

接下来，我们陈述一些已知结果。

引理1（Dragan 和 Leitert [8]）。如果给定的图 G 包含一条离心率为 ℓ的最短路径，则可以在 $\mathcal{O}(n^{2\ell+2}m)$时间内求解 G 的 *MESP* 问题，其中 m 是 G中的边数。

引理2（Kučera 和 Suchý [10]）。对于任意图 $G = (V, E)$、任意集合 $M \subseteq V$以及任意顶点 $s \in V$，顶点 M 的排列 $\pi = (m_1, \ldots, m_{|M|})$至多有一个满足以下条件：存在一条最短路径 P，使得其首个顶点为 s，P 包含 M的所有顶点，并且 M 中的顶点在 P 上出现的顺序恰好与 π给出的顺序一致。此外，给定 G的预计算距离矩阵后，可以在$\mathcal{O}(|M| \log |M|)$时间内找到排列 π。

3 按反馈顶点集和离心率参数化

在本节中，我们为 ESP/fvs+ecc 设计一个 FPT 算法。本节的主要定理正式表述如下。

定理 1. 存在一个用于 ESP/fvs + ecc 的算法，运行时间为$\mathcal{O}(2^{\mathcal{O}(k \log k)}\ell^k n^{\mathcal{O}(1)})$.

算法概述。给定一个图 G 和一个大小为 k的反馈顶点集 S，我们使用一些归约规则以及两个称为骨架测试和扩展骨架测试的中间问题，将ESP/fvs + ecc归约为辅助图 G' （一个森林）上的“路径问题”（我们称之为彩色路径覆盖）；该辅助图是 $G[V \setminus S]$ 的子图。在第 3.1 节中，我们证明 ESP/fvs +ecc和骨架测试是FPT等价的。接下来，在第3.3节中，我们将骨架测试归约为扩展骨架测试。然后，在第3.4 节中，我们将扩展骨架测试归约为彩色路径覆盖。最后，在第 3.5 节中，我们为彩色路径覆盖设计了一种基于动态规划的算法，其运行时间为 $\mathcal{O}(\ell^2 2^{\mathcal{O}(k \log k)} n^{\mathcal{O}(1)})$。结合归约到中间问题所需的时间，我们得到了所需的 FPT 算法。算法各步骤的流程图如图 2 所示。

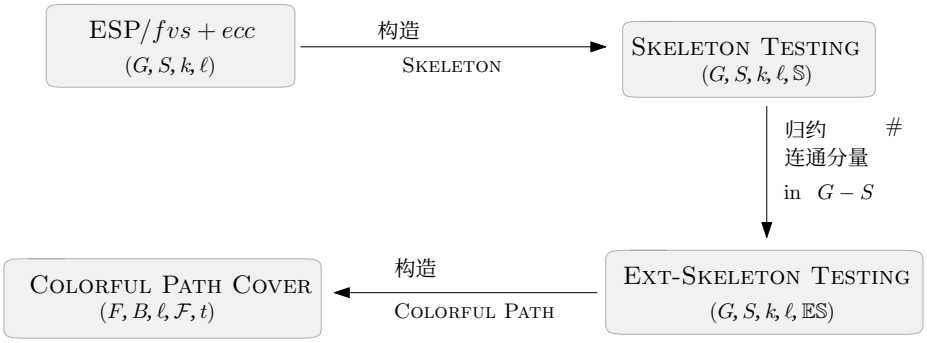

图 2. ESP/fvs + ecc. 算法流程图

3.1 化简为骨架测试

该问题的输入是一个实例（G， S， k， ℓ），其中 $S \subseteq V(G)$ 是 G 中大小为 k 的反馈顶点集。设（G， S， k， ℓ）是一个是实例，并且 P 是一条解路径，即一条最短路径，使得对于每个 $v \in V$（G），存在 $u \in V$（P），满足 d_G（u， v） $\leq \ell$。我们的最终目标是构造这样一条路径 P。为此，我们尝试在时间 $f(k,\ \ell)n^{\mathcal{O}(1)}$ 内尽可能多地获取关于 P 的信息。注意，如果 S 是空集，那么只需知道其端点即可得到 P，因为树中任意两个顶点之间都存在唯一路径。推广这一思路，给定集合 S，我们定义 P 的骨架概念。

定义1（骨架）。 P 的骨架，记为 $\mathbb{S}$，是以下信息集合。

– P的端点，例如 $u, v \in V(G)$.– $S \setminus \{u, v\}$的一个子集，例如 M，由出现在 P上的顶点组成。也就是说， $V(P) \cap (S \setminus \{u, v\}) = M$。

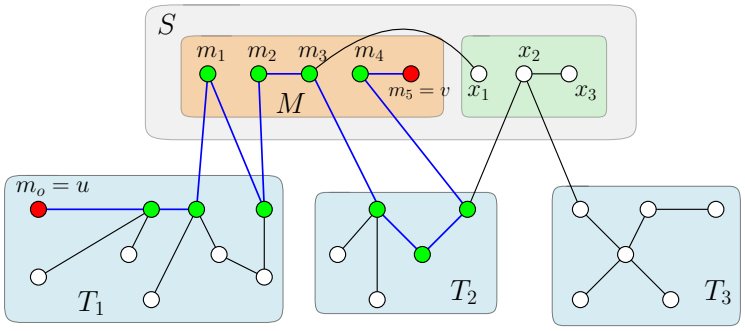

图 3。 P的骨架示例。这里， P是经过绿色内部顶点、连接两个红色顶点 u和 v的最短路径（蓝色边）。对于顶点x_1， x_2和 x_3， $f(x_1) = f(x_2) = 1$， $f(x_3) = 2$， $g(x_1) = 3$， $g(x_2) = g(x_3) = (3,\ 4)$。

– M 的顶点在 P上出现的顺序由排列$\pi = m_1, m_2, \ldots, ^{m_{|M}|$给出。为方便记号，我们用 m_0表示 u， 用$^{m_{|M|+1}}$表示v 。 – 集合 $X = S \setminus M$的距离剖面（f，g）定义如下：函数 $f: X \to [\ell]$ 满足： $f(x)$表示顶点 x到 P的最短距离；函数 $g: X \to \{0, 1, \cdots, |M| + 1, (0, 1), (1, 2) \cdots, (|M|, |M| + 1)\}$满足： $g(x)$存储关于 P上距离 x最近的顶点所在位置的信息。也就是说，如果距离 P最近的顶点属于 $\{m_0, m_1, \ldots, m_{|M|}, m_{|M|+1}\}$， 那么 $g(x)$通过赋予相应的索引来存储该信息。否则，最近的顶点属于某个 $0 \leq i \leq |M|$对应的、位于 m_i, m_{i+1}之间的路径段， $g(x)$则通过赋值$(i, i + 1)$来存储该信息。

骨架的示意图见图3。根据骨架的定义，我们可以得到其上界。

观察1 骨架的数量上界为 $n^2 2^k k! \ell^k (2k+2)^k$.

我们称路径 P 实现 骨架 $\mathbb{S}$ ，如果满足以下条件。

1. $M = S \cap V(P)$、 $X \cap V(P) = \emptyset$， M中顶点的排列在 P中等于 π， P 的端点为 m_0 和$^{m_{|M|+1}}$， 2. 对于每个 $v \in V(G)$，存在一个顶点 $u \in V(P)$，使得 $d_G(u, v) \leq \ell$， 3. 对于每个 $v \in X$，对于所有 $w \in V(P)$， $d_G(v, w) \geq f(v)$（其中 $f(v)$是从 v到 G中 P 上任意顶点的最短距离），以及4. 对于每个 $v \in X$，如果 $g(v) = i$，其中 $i \in [0, |M| + 1]$，则 $d_G(v, m_i) = f(v)$；并且如果 $g(v) = (i, i + 1)$，其中 $i \in [0, |M|]$，则在 P中从 m_i到 m_{i+1}的一条子路径上存在一个顶点 u，使得 $u \notin \{m_i, m_{i+1}\}$ 且 $d_G(u, v) = f(v)$。

现在，给定输入 (G, S, k, ℓ) 和一个骨架 $\mathbb{S}$，我们的目标是测试该骨架能否实现为所需的路径 P。这引出了以下问题。

骨架测试 **参数：** $k+\ell$**输入：** 一个图 G，一个大小为 k 的集合 $S \subseteq V(G)$，使得 $G-S$ 是一片森林，一个整数 ℓ，以及一个骨架 $\mathbb{S}$。**问题：** 在 G 中是否存在一条最短路径 P，使得 $\mathbb{S}$ 得以实现?

下一个引理表明，可以将 ESP/fvs + ecc 归约到骨架测试问题。

引理3。 (G, S, k, ℓ) 是 ESP/fvs + ecc的是实例，当且仅当存在一个骨架 $\mathbb{S}$ 使得 $(G, S, k, \ell, \mathbb{S})$ 是骨架测试的是实例。

证明。对于正向，假设 (G, S, k, ℓ) 是 ESP/fvs + ecc 的是实例，并令 P 为其解路径。令 $M = V(P) \cap S$, $|M| = t$ and$X = S \setminus M$。令 π 为 M 在 P 上的顶点排列。令 $m_0, m_{|M|+1}$ 分别为 P 的第一个和最后一个端点。我们定义 $f\colon X \to [\ell]$，使得对于顶点 $v \in X$，$f(v)$ 是从 v 到路径 P 的最小距离。令 u 为路径 P 上满足 $d(v,u) = f(v)$ 的顶点。注意，f 只赋予集合 $[\ell]$ 中的值。我们定义函数 g, $g\colon X \to \{0, 1, \cdots, t+1, (0,1), (1,2) \cdots, (t, t+1)\}$ 如下：若对于某个 $i \in [0, t+1]$，$u = m_i$，则 $g(v) = i$；若 u 包含在某个 $i \in [0,t]$ 的 m_i 到 m_{i+1} 子路径中，则 $g(v) = (i, i+1)$。注意，g 只赋予集合 $\{0, 1, \cdots, t+1, (0,1), (1,2) \cdots, (t, t+1)\}$ 中的值。观察可知，元组 $(M, X, \pi, m_0, m_{t+1}, f, g)$ 是 P 的一个骨架。因此，$(G, S, k, \ell, \mathbb{S})$ 是骨架测试的一个是实例，其中 $\mathbb{S} = (M, X, \pi, m_0, m_{t+1}, f, g)$】【。

在反向中，假设 $\mathbb{S} = (M, X, \pi, m_0, m_{t+1}, f, g)$ 是一个骨架，使得 $(G, S, k, \ell, \mathbb{S})$ 是骨架测试的一个是实例，并令 P 为其解。注意，对于 ESP/fvs+ecc 的 (G, S, k, ℓ)，P 也是一个解，因为对于每个 $v \in V(G)$，都存在一个顶点 $u \in V(P)$，使得 $d_G(u, v) \leq \ell$。□

观察 1 将骨架的数量上界限定为 $2^{\mathcal{O}(k(\log k + \log \ell))} n^2$。结合引理 3，可推出 ESP/fvs + ecc 和 骨架测试是 FPT等价的。因此，从现在开始，我们将重点讨论 骨架测试。

3.2 骨架测试算法

令 $(G, S, k, \ell, \mathbb{S})$ 为骨架测试的一个实例，其中 $\mathbb{S} = (M, X, \pi, m_0, m_{|M|+1}, f, g)$。我们的算法如下运行。首先，算法通过归约规则执行简单的合理性检查。本质上，它检查骨架 $\mathbb{S}$ 的不同连通分量是否有效。

归约规则1（合理性测试1）返回：若满足以下任一条件，则（$G, S, k, \ell, \mathbb{S}$）不是骨架测试的实例：

1. 对于 $i \in [0, |M|]$, $m_i m_{i+1}$ 是 G 中的一条边，且 $g^{-1}((i, i+1)) \neq \emptyset$。（$g$ 无效。） *2.* 对于一个顶点 $v \in X$，存在一个顶点 $u \in M \cup \{m_0, m_{|M|+1}\}$，使得 $d_G(u, v) < f(v)$。（f无效。）

3. 对于顶点 $v \in X$，$g(v) = i$ 且 $d_G(v, m_i) > f(v)$。（f 无效。） *4.* 对于一个 $i \in [0, |M|]$，$m_i m_{i+1}$ 不是 G 中的一条边，并且在 $G - (S \setminus \{m_i, m_{i+1}\})$ 中不存在从 m_i 到 m_{i+1} 的路径，或者该路径的长度大于 G 中从 m_i 到 m_{i+1} 的最短路径长度。（π 无效。） *5.* 对于 i，$j \in [0, |M|]$，$i < j$，存在 $G - (S \setminus \{m_i, m_{i+1}\})$ 中从 m_i 到 m_{i+1} 的最短路径 P_i，以及 $G - (S \setminus \{m_j, m_{j+1}\})$ 中从 m_j 到 m_{j+1} 的最短路径 P_j，使得如果 $j = i+1$，则 $(V(P_i) \setminus \{m_{i+1}\}) \cap (V(P_j) \setminus \{m_j\}) \neq \emptyset$，否则 $V(P_i) \cap V(P_j) \neq \emptyset$。（$\pi$ 无效——最短路径断言将被违反。） *6.* 对于满足 $m_i m_{i+1} \notin E(G)$ 的 $i \in [0, |M|]$，$g^{-1}(i, i+1) \neq \emptyset$，并且对于 $G - S$ 中的每个连通分量 C，以及对于 $G[V(C)$ 中的每条从 m_i 到 m_{i+1} 的路径 P，存在一个顶点 $u \in g^{-1}(i, i+1)$，使得不存在满足 $d_G(u, v) = f(u)$ 的顶点 $v \in V(P) \setminus \{m_i, m_{i+1}\}$。（$g$ 无效。）

引理 4. 归约规则 *1* 是安全的。

证明。 (1) 的证明基于以下事实：为了保持解的最短路径性质，边 $m_i m_{i+1}$ 必须属于解。这与如下事实矛盾：对于每个 $v \in g^{-1}((i, i+1))$，解路径的 m_i 到 m_{i+1} 子路径上都存在一个顶点 u，使得 $u \notin \{m_i, m_{i+1}\}$ 且 $d_G(u, v) = f(v)$。(2) 的证明基于以下事实：否则，解路径中的函数 g 的性质将遭到破坏。(4) 的证明由解的最短路径性质得出。(3) 和 (6) 的证明显然，因为在这些情形下不可能找到解路径。

(5) 的证明：假设 $(G, S, k, \ell, \mathbb{S})$ 是 Skeleton Testing 的一个是实例，且 P 是其解。对于 $i, j \in [0, |M|], i < j$，在 $G - (S \setminus \{m_i, m_{i+1}\})$ 中存在一条从 m_i 到 m_{i+1} 的最短路径 P_i，并且在 $G - (S \setminus \{m_j, m_{j+1}\})$ 中存在一条从 m_j 到 m_{j+1} 的最短路径 P_j。假设 $j = i+1$，则 $(V(P_i) \setminus \{m_{i+1}\}) \cap (V(P_j) \setminus \{m_j\}) \neq \emptyset$。令 $v^* \in (V(P_i) \setminus \{m_{i+1}\}) \cap (V(P_j) \setminus \{m_j\})$ 为路径 P_i, P_j 上按照路径 P_i 的排列所确定的最后一个相交顶点。注意，P_i 中从 m_i 到 v^* 的距离严格小于 P 上 m_i 到 $m_{i+1}(m_j)$ 的子路径长度，也就是说 $d_{P_i}(m_i, v^*) < d_P(m_i, m_{i+1})$，因为 $v^* \neq m_{i+1} = m_j$，且 P_i 和 m_i 到 m_{i+1} 在 P 上的子路径都是 G 中从 m_i 到 m_{i+1} 的最短路径。类似地，$d_{P_j}(v^*, m_{j+1}) \leq d_P(m_j, m_{j+1})$，因为 P_j 和 m_j 到 m_{j+1} 在 P 上的子路径都是 G 中从 m_j 到 m_{j+1} 的最短路径。这意味着 $d_P(m_i, m_{j+1}) > d_{P_i}(m_i, v^*) + d_{P_j}(v^*, m_{j+1})$。用 m_i 到 v^* 的子路径和 v^* 到 m_{j+1} 的子路径替换 P 中从 m_i 到 m_{j+1} 的子路径，可在 G 中得到一条更短的路径，这与解 P 的最短路径性质矛盾。对于 $j \neq i+1$ 的情形，可以用类似的方法论证。□

对 $G - S$ 的连通分量进行归约： 现在，我们介绍应用于 $G - S$ 中连通分量的标记过程和归约规则。设 P_i 是 P 的一个路径段（子路径），位于 m_i 和 m_{i+1} 之间，且至少包含两条边。进一步，设 P_i^{int} 是从 P_i 中删除 m_i 和 m_{i+1} 后得到的子路径。于是，P_i^{int} 是 $G - S$ 中两个顶点之间的一条路径（也就是说，是森林 $G - S$ 中的一条路径）。这意味着 P 由 S 和森林 $G - S$ 中至多 $k+1$ 条路径组成。设这些路径为 $\mathbb{P} = P_1^{int}$、……、P_q^{int}，其中 $q \leq k+1$。接下来，我们尝试理解森林 $G - S$ 中的这 $k+1$ 条路径。实际上，如果存在

一个连通分量 C 属于 $G-S$，且其中有一个顶点与 S 中的每个顶点都相距很远，那么 C 必须包含 $\mathbb{P}$ 中的一条路径（关键连通分量）。这类连通分量最多有 $k+1$ 个。连通分量包含来自 $\mathbb{P}$ 的路径的另一个原因，是选择一条有助于满足 g 函数所给约束的路径（g-满足条件的连通分量）。接下来，我们给出一个标记 $\mathcal{O}(k)$ 个连通分量的过程，随后证明所有未标记的连通分量都可以安全删除。

标记过程： 令 $\mathcal{C}^*$ 为 $G-S$ 中已标记连通分量的集合。初始时，令 $\mathcal{C}^*=\emptyset$。

– 步骤 1. 如果 $G-(S\cup V(\mathcal{C}^*))$ 中存在一个连通分量 C，其中包含顶点 v，使得对于所有 $m_i\in M$，$d_G(v,m_i)>\ell$，并且对于所有 $u\in X$，$d_G(v,u)>\ell-f(u)$，则将 C 加入 $\mathcal{C}^*$。（标记关键连通分量）– 步骤 2. 对于 $i=0$ 至 $|M|$，按如下方式进行：令 C 为 $G-(S\cup V(\mathcal{C}^*))$ 中某个连通分量，使得存在一条从 m_i 到 m_{i+1} 的路径 P_i，它位于 $G[V(C)\cup\{m_i,m_{i+1}\}]$ 中，是 G 中从 m_i 到 m_{i+1} 的最短路径；并且对于每个顶点 $v\in g^{-1}((i,i+1))$，都存在一个顶点 $u\in V(P_i)\setminus\{m_i,m_{i+1}\}$，使得 $d_G(u,v)=f(v)$。然后，将 C 加入 $\mathcal{C}^*$，并将索引 i 加一。（标记满足 g 的连通分量）

令 $\mathcal{C}_1$ 为在步骤 1中添加到 $\mathcal{C}^*$ 的连通分量集合。现在，我们陈述一些归约规则，算法将按照所陈述的顺序穷尽地应用这些规则。

规约规则 2若 $|\mathcal{C}_1|\geq k+2$，则返回 $(G,S,k,\ell,\mathbb{S})$ 不是骨架测试的实例。

引理 5。归约规则*2* 是安全的。

证明。对于每个连通分量 C 中的 $\mathcal{C}_1$，C 包含一个顶点 v 使得 $d_G(v,m_i)>\ell$，对于所有 $m_i\in M$和 $d_G(v,u)>\ell-f(u)$，对于所有 $u\in X$，这意味着我们必须从解路径中的连通分量 C 添加一条路径作为子路径，使其包含一个覆盖 v的顶点。注意，我们最多可以在解路径中添加 $|M|+1$ 个子路径。因此，$|\mathcal{C}_1|\leq|M|+1$ 如果$(G,S,k,\ell,\mathbb{S})$ 是骨架测试的一个是实例。我们得到所需的界限为 $|M|\leq k$。□

规约规则3 如果存在一个连通分量 C *in* $G-S$ *such that* $C\notin\mathcal{C}^*$，则从 G中删除 $V(C)$。所得实例为 $(G-V(C)\ S,k,\ell,\mathbb{S})$。

引理 6。归约规则 *3* 是安全的。

证明。我们证明，$(G,S,k,\ell,\mathbb{S})$ 是骨架测试的是实例，当且仅当 $(G-V(C)\ S,k,\ell,\mathbb{S})$ 是骨架测试的是实例。回顾 $\mathbb{S}=(M,X,\pi,m_0,m_{|M|+1},f,g)$。在正向，考虑 $(G,S,k,\ell,\mathbb{S})$ 是骨架测试的是实例，并且 P是其解。假设 P 包含一条从 m_i 到 m_{i+1} 的子路径 P^*，使得 P^* 是在 $G[V(C)\cup\{m_i,m_{i+1}\}]$中的一条路径。由于 C不在 $\mathcal{C}^*$中，存在一个连通分量$C'\in\mathcal{C}^*$，使得存在一条从 m_i到 m_{i+1}的路径 P'，该路径位于 $G[V(C')\cup\{m_i,m_{i+1}\}]$,

使得 P' 是 G 中从 m_i 到 m_{i+1} 的最短路径，并且对于每个顶点 $v \in g^{-1}((i, i+1))$，$d_G(v, P' - \{m_i, m_{i+1}\}) = f(v)$。我们将子路径 P^* 替换为子路径 P'。令 P'' 为所得路径。我们断言，P'' 也是 $(G, S, k, \ell, \mathbb{S})$ 的一个解。注意，P'' 满足函数 f 和 g。假设存在一个顶点 $v \in V(G)$，使得 $d_G(v, P'') > \ell$。那么 v 必须属于 C。由于在标记过程的步骤 1 中，C 未被加入 $\mathcal{C}^*$，因此 C 中的每个顶点要么与某个 $m_i, i \in [0, |M|+1]$ 的距离至多为 ℓ，要么对于某个 $u \in X$，$d_G(v, u) \leq \ell - f(u)$。因此，$v$ 被 P'' 上的某个顶点覆盖，矛盾。所以 P'' 也是 $(G, S, k, \ell, \mathbb{S})$ 的一个解，并且 P'' 不包含来自 C 的任何顶点。由于 C 的每个顶点都被覆盖，P'' 也是骨架测试中 $(G - V(C)\ S, k, \ell, \mathbb{S})$ 的一个解。如果 P 不包含来自 C 的任何顶点，那么通过类似的过程，我们可以得到一个针对 $(G, S, k, \ell, \mathbb{S})$ 的解路径 P，其中不包含 C 中的任何顶点。注意，P 也是骨架测试中 $(G - V(C)\ S, k, \ell, \mathbb{S})$ 的一个解。

在反向方向上，假设 $(G - V(C)\ S, k, \ell, \mathbb{S})$ 是骨架测试的一个是实例，并令 P 为其解。由于在标记过程的第 1 步中，C 未被加入 $\mathcal{C}^*$，因此 C 中的每个顶点，要么与某个 $m_i, i \in [0, |M|+1]$ 的距离至多为 ℓ，要么对于某个 $u \in X$，有 $d_G(v, u) \leq \ell - f(u)$，因此 C 中的每个顶点都被 P 上的某个顶点覆盖。这意味着 P 也是 $(G, S, k, \ell, \mathbb{S})$ 的一个解。证明至此完成。□

注意，当归约规则 2 和归约规则 3 不再适用时，$G - S$ 中的连通分量数量至多为 $2(k+1)$。这是因为 $|\mathcal{C}_1| \leq k+1$，并且从 $\mathcal{C}^* - \mathcal{C}_1$ 中的每个连通分量都存在一条路径（属于该解），因此 $|\mathcal{C}^* - \mathcal{C}_1| \leq k+1$。否则，给定实例是一个骨架测试的否实例骨架测试。注意，我们所有的归约规则都可以在 $n^{\mathcal{O}(1)}$ 时间内应用。

3.3 将归约为 骨架测试 到 扩展骨架测试:

设（G、S、k、ℓ、$\mathbb{S}$）是骨架测试的一个归约后实例。也就是说，这是一个不再适用归约规则 1、2 和 3 的实例。这意味着 $G - S$ 中连通分量的数量至多为 $2k+2$。接下来，我们通过添加一个函数 γ 来扩充骨架，该函数记录 $G - S$ 中某个连通分量的索引，此连通分量给出 P 中从 m_i 到 m_{i+1} 的子路径；或者记录$m_i m_{i+1}$是所需路径 P 中的一条边，其中 $i \in [0, |M|]$。

定义 2（增强骨架）。路径 P 的增强骨架 记为 $\mathbb{ES}$，包含 $\mathbb{S}$ 以及 m_i 和 m_{i+1} 之间路径的线段概况，其中 $i \in [0, M]$。设 $C_1, C_2, \ldots, C_q$ 为 $G - S$ 中的连通分量。那么，线段概况由函数 $\gamma\colon [0, |M|] \to [0, q]$ 给出。函数 γ 表示如下：对于每个 $i \in [0, |M|]$，若 $\gamma(i) = 0$，则 m_i, m_{i+1} 这一对应该通过一条边连接在解路径 P 中；否则，若 $\gamma(i) = j$，则在 P 中，从 m_i 到 m_{i+1} 的子路径包含于 $G[V(C_j) \cup \{m_i, m_{i+1}\}]$。此外，称 $\mathbb{ES}$ 丰富了骨架 $\mathbb{S}$。

设 $\mathbb{S}$ 为一个骨架。丰富 $\mathbb{S}$ 的 $\mathbb{ES}$ 数量上界为 $(q+1)^{k+1}$。因此，除非 q 由 k 的一个函数所界定，ℓ，否则这对我们没有帮助。

幸运的是， $G-S$ 中的连通分量数量至多为 $2k+2$，因此 $\mathbb{ES}$ 的数量上界为 $2^{\mathcal{O}(k\log k)}$。

如果路径 P 实现通过 $\mathbb{S}$增强的增强骨架 $\mathbb{ES}$ ，且满足 γ，我们就称该路径 P实现了 $\mathbb{S}$ 。与骨架测试类似，我们可以定义扩展骨架测试，其目标是检验是否存在一条路径，能够实现增强骨架 $\mathbb{ES}$。此外，很容易看出，骨架测试与扩展骨架测试是 FPT等价的，因此我们可以专注于扩展骨架测试。设 $(G, S, k, \ell, \mathbb{ES})$ 是扩展骨架测试的一个实例，其中 $G-S$ 至多有 $2k+2$ 个连通分量。与骨架测试类似，我们首先对扩展骨架测试的实例进行一些合理性检查。

归约规则 4（合理性检查2）如果满足以下任一条件，则返回 $(G, S, k, \ell, \mathbb{ES})$ 是扩展骨架测试的否实例：

1. m_im_{i+1} 是 G 中的一条边，且 $\gamma(i)\neq 0$，（或者） m_im_{i+1} 不是 G 中的一条边，且$\gamma(i)=0$。*2.* 对于一个 $i\in[|M|]$， $\gamma(i)=j\neq 0$ ，并且存在：–在 $G[V(C_j)\cup\{m_i,m_{i+1}\}]$中不存在从 m_i 到 m_{i+1} 的路径，（或者）–在 $G[V(C_j)\cup\{m_i,m_{i+1}\}]$ 中不存在从 m_i 到 m_{i+1} 的路径，该路径同时也是 G中从m_i 到 m_{i+1} 的最短路径，（或者）–不存在一条 P_i 在 $G[V(C_j)\cup\{m_i,m_{i+1}\}]$中的从 m_i 到 m_{i+1} 的路径，该路径同时也是 G 中从 m_i 到 m_{i+1} 的最短路径，并满足以下性质：对于每个顶点 $v\in g^{-1}((i,i+1))$，都存在一个顶点 $u\in V(P_i)\setminus\{m_i,m_{i+1}\}$，使得 $d_G(u,v)=f(v)$。

上述规则的安全性可由定义 2 得出。

3.4 将Ext-骨架测试归约为ColorfulPath-Cover

设（$G,S,k,\ell,\mathbb{ES}$）是 Ext-Skeleton Testing 的一个实例，且规约规则 4 已不再适用。此外，假设 $G-S$ 中的连通分量数量为 $k'\leq 2k+2$，并且 $\gamma:[0,|M|]\to[0,k']$ 是 $\mathbb{ES}$ 中的函数。我们的目标是找到一条路径 P，使其实现 $\mathbb{ES}$。注意，对于一个 $i\in[0,|M|]$，如果 $\gamma(i)=j\neq 0$，那么连接 m_i，m_{i+1} 这一对的相关路径都包含在 $G-S$ 中的连通分量 C_j 内。此外，在 C_j 中所有连接 m_i 和 m_{i+1} 的路径中，只有满足函数 g 的最短路径才是相关路径。因此，我们枚举每一对 m_i,m_{i+1} 在一个路径族 $\mathcal{F}_i$ 中的所有可行路径，并着重寻找一个仅包含来自这个已枚举路径集的子路径的解。注意，现在我们的问题已化简为：在 $G-S$ 中寻找一个路径集 $\mathcal{P}$，该路径集恰好包含每个可行路径族中的一条路径，并覆盖 $G-S$ 中所有距离 S 较远的顶点。接下来，我们将上述讨论形式化。首先，我们描述枚举过程。

对于每个满足 $\gamma(i)=j\neq 0$ 的 $i\in[0,|M|]$，我们按如下方式构造一族 $\mathcal{F}_i$ 可行路径。令 P_i 为 $G[V(C_j)\cup\{m_i,m_{i+1}\}]$ 中的一条路径，使得 (i) P_i 是 G 中从 m_i 到 m_{i+1} 的最短路径，(ii) 对于每个顶点 $v\in g^{-1}((i,i+1)$,

$d_G(v, P_i - \{m_i, m_{i+1}\}) = f(v)$。设 m'_i, m'_{i+1} 分别为 m_i, m_{i+1} 在 P_i 中的邻居。然后我们将 m'_i 添加到 m'_{i+1} 到 $\mathcal{F}_i$ 的子路径中。注意，一个可行路径族 $\mathcal{F}_i$ 满足以下性质：(1) 对所有 $i, i' \in \gamma^{-1}(j)$ $i \neq i'$， $V(\mathcal{F}_i) \cap V(\mathcal{F}_{i'}) = \emptyset$，因为归约规则1的第5项不适用，而且我们只在路径族中添加最短路径。(2) $\mathcal{F}_i$ 恰好包含来自 $G - S$ 中一个连通分量的路径（根据构造）。令 $\mathcal{F}$ 为所有可行路径族的集合。

上述讨论引出了以下问题。

彩色路径覆盖**输入：** 一个森林 F、一个集合 $B \subseteq V(F)$、一个整数 ℓ，以及一个由 t 个两两不相交的可行路径族组成的族 $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_t\}$。
问题： 是否存在一个由 t 条路径组成的集合 $\mathcal{P}$，使得对于每个 $\mathcal{F}_i, i \in [t]$，$|\mathcal{P} \cap \mathcal{F}_i| = 1$，以及每个顶点 $v \in B$，都存在一条路径 $P \in \mathcal{P}$ 和一个顶点 $u \in V(P)$，满足 $d_F(u, v) \leq \ell$?

令 F 为从 $G - S$ 中移除所有满足 $\gamma^{-1}(j) = \emptyset$ 的连通分量 C_j 后得到的森林，也就是说，移除不包含任何有趣路径的连通分量。注意，包含有趣路径的连通分量数量至多为 $2k + 2$。令 $B \subseteq V(F)$ 为未被 S 中的顶点覆盖的顶点集合，也就是说，它包含所有满足以下条件的顶点 $v \in V(F)$：对于所有 $i \in [0, |M| + 1]$ 和 $u \in X$， $d_G(v, m_i) > \ell$，且 $d_G(v, u) > \ell - f(u)$。我们断言，只需在实例 $(F, B, \ell, \mathcal{F})$ 上求解彩色路径覆盖问题，其中 F 至多包含 $2k + 2$ 棵树。以下引理形式化地给出了一个归约，并由此得出：以 k 为参数的 Ext-骨架测试问题与以 k 为参数的彩色路径覆盖问题 FPT等价。

引理 7。$(G, S, k, \ell, \mathbb{ES})$ 是 *Ext*-骨架测试的一个是实例，当且仅当$(F, B, \ell, \mathcal{F})$ 是彩色路径覆盖的一个是实例。

证明。回忆一下， $\mathbb{S} = (M, X, \pi, m_0, m_{|M|+1}, f, g)$和 $\mathbb{ES} = (\mathbb{S}, \gamma)$。在正向方向上，假设$(G, S, k, \ell, \mathbb{ES})$是骨架测试的一个是实例，并令 P为其解。考虑 $G - S$中的一个连通分量 C_j。如果 $\gamma^{-1}(j) = \emptyset$，那么根据 P的性质，不存在 $i \in [0, |M|]$ 使得从 m_i到 m_{i+1}的路径包含于 $G[V(C_j) \cup \{m_i, m_{i+1}\}]$中。否则，对于每个 $i \in \gamma^{-1}(j)$，令 P_i 为 P 中从 m_i 到 m_{i+1}的子路径。令m'_i, m'_{i+1}为 P_i中 m_i, m_{i+1}的邻居，并令 P'_i为 P_i中从 m'_i到 m'_{i+1}的子路径。我们有 P'_i 包含于 C_j 中，并满足函数 g。令$\mathcal{P} = \{P'_i | i \in [0, |M|], V(P'_i) \subseteq V(G - S)\}$。根据 $\mathcal{F}_i$的构造，我们有 $\mathcal{P} \cap \mathcal{F}_i = P'_i$，也就是说，由于归约规则1的条目5， $|\mathcal{P} \cap \mathcal{F}_i| = 1$。由于$P$实现了 $\mathbb{S}$，并且对于每个顶点 $v \in B$， $d_G(v, m_i) > \ell$，对于所有 $i \in [0, |M| + 1]$以及 $d_G(v, u) > \ell - f(u)$，对于所有 $u \in X$。因此，在 $\mathcal{P}$中存在一条路径，该路径包含一个顶点 v'，使得 $d_G(v, v') \leq \ell$。这意味着 $\mathcal{P}$ 是彩色路径覆盖中$(F, B, \ell, \mathcal{F})$的一个解，因此$(F, B, \ell, \mathcal{F})$是彩色路径覆盖的一个是实例。

在反向中，假设（F、B、ℓ、$\mathcal{F}$）是 Col-orful Path-Cover 的是实例，并令 $\mathcal{P}$ 为其解。令 $P'_i = \mathcal{F}_i \cap \mathcal{P}$ 和 m'_i、m'_{i+1} 为其端点，使得 $m_i m'_i$、$m'_{i+1} m_{i+1} \in E(G)$。令 P_i 为包含边 $m_i m'_i$、$m'_{i+1} m_{i+1}$ 和路径 P'_i 的从 m_i 到 m_{i+1} 的路径。当 $\gamma(i) \neq 0$ 时，我们通过连接 $\mathcal{P}$ 中的路径 P_i，并在 $\gamma(i) = 0$ 时加入边 $m_i m_{i+1}$，来构造路径 P。根据 $\mathcal{F}_i$ 的构造，P 满足 M 的函数 g、f、γ 和排列 π。注意，对于每个顶点 $v \in V(G) \setminus B$，$d_G(v, m_i) \leq \ell$，对于某个 $i \in [0, |M|+1]$ 或 $d_G(v, u) \leq \ell - f(u)$，对于某个 $u \in X$。因此，$V(G)$ 中的每个顶点 $\setminus B$ 都由 P 覆盖。显然，P 覆盖 B 中的每个顶点。由于我们只在 $\mathcal{F}$ 中添加最短路径，并且归约规则 4 不适用，因此 P 也是 G 中的最短路径。这意味着 P 是 Ext-Skeleton Testing 中（G、S、k、ℓ、$\mathbb{S}$）的解，因此（G、S、k、ℓ、$\mathbb{S}$）是 Ext-Skeleton Testing 的是实例 □

我们为彩色路径覆盖 问题设计了一种基于动态规划的算法，其参数为 k。由于树的数量至多为$2k+2$，可行路径族的数量为 $|\mathcal{F}| = t$，我们首先猜测 $\mathcal{F}$ 中每棵树对应的可行路径族子集，所需 $\mathcal{O}(k^t)$ 时间。现在，我们可以处理一棵树及其猜测得到的可行路径族。我们先概述算法，然后在第3.6节给出算法。

引理 8. 彩色路径覆盖 问题可在 $\mathcal{O}(\ell^2 \cdot 2^{\mathcal{O}(k \log k)} n^{\mathcal{O}(1)})$ 时间内求解，其中 F是一个包含 $\mathcal{O}(k)$ 棵树的森林。

3.5 彩色路径覆盖算法概述

考虑彩色路径覆盖问题的一个实例 $(T, B, \ell, \mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_t\})$，其中 T是一棵树，$B \subseteq V(T)$，且 $\ell \in \mathbb{N}$ 和 $\mathcal{F}$是可行路径的一个不相交族。目标是找到一个由 t 条路径组成的集合 $\mathcal{P}$，使得对于每个 $\mathcal{F}_i, i \in [t], |\mathcal{P} \cap \mathcal{F}_i| = 1$ 以及每个顶点 $v \in B$，都存在一条路径 $P \in \mathcal{P}$ 和一个顶点$u \in V(P)$，满足 $d_T(u, v) \leq \ell$。

对于顶点 $v \in V(T)$，自底向上动态规划算法会为 v 的每个子节点 w 考虑一个子问题，并从左到右依次处理。为了计算以 v 的某个子节点为根的子树上的部分解，我们区分是否存在一条包含 v 且属于 $\mathcal{P}$ 的路径。为此，我们定义一个变量，用于表示 $\mathcal{P}$ 中是否存在一条包含 v 的路径。如果存在这样的路径，我们就猜测该路径端点所属的区域，其中包括以下几种情形：该路径仅包含顶点 v；该路径包含 v 的父节点，且一个端点属于以 w 为根的子树，或属于 v 的左侧子节点的子树，或属于 v 的右侧子节点的子树；该路径的两个端点分别属于 w 左侧或右侧子节点的子树；一个端点属于以 w为根的子树，而另一个端点属于 w 左侧或右侧子节点的子树。图4展示了这些情形。

在每个节点 v，我们存储 v到 v的子树中最近顶点（设为 w'）的距离，也就是说，存储 v到 $\mathcal{P}$中一条路径上的该顶点的距离。我们这样存储，是希望

w' 能够覆盖未来出现的 B 中的顶点。此外，我们还存储 v 的子树中最远的顶点（设为 w''），该顶点未被 $\mathcal{P}$ 子树中任何已选路径覆盖。我们这样存储，同样是希望未来某个顶点能够覆盖 $w'' \in B$，并使当前的解最终构成一个整体解。

在每个节点 v，我们捕获以下内容的存在性：存在一组 $t' \leq t$ 路径 Y，每个 $\mathcal{F}_i$ 各一条，该组在从 Y 于 $\mathcal{P}$ 中满足最近顶点 w' 和最远顶点 w''（来自 v）的距离条件的路径上，要么包含 v，要么不包含；这些顶点分别位于 Y 上，并且已由Y 覆盖和尚未被覆盖。为得出根节点存在彩色路径覆盖，我们检查是否存在一个由一组 Y 个 t 路径组成的条目，每个 $\mathcal{F}_i$各一条，并且所有未覆盖顶点的最远距离均为零。

3.6 彩色路径-覆盖算法

在本节中，我们针对参数 k，设计一个基于动态规划的 FPT 算法来解决彩色路径覆盖问题。由于树的数量至多为 $2k+2$，且可行路径族的数量为 $|\mathcal{F}| = t$，我们首先在 $\mathcal{O}(k^t)$ 时间内猜测来自 $\mathcal{F}$ 中每棵树的可行路径族子集。现在，我们可以处理一棵树及其猜测的可行路径族。考虑一个彩色路径覆盖问题的实例 $(T, B, \ell, \mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, \ldots \mathcal{F}_t\})$，其中 T 是一棵树，$B \subseteq V(T)$，且 $\ell \in \mathbb{N}$ 和 $\mathcal{F}$ 是可行路径的一个不相交族。目标是找到一个包含 t 条路径的集合 $\mathcal{P}$，使得对于每个 $\mathcal{F}_i$、$i \in [t]$、$|\mathcal{P} \cap \mathcal{F}_i| = 1$ 以及每个顶点 $v \in B$，都存在一条路径$P \in \mathcal{P}$和一个顶点 $u \in V(P)$，使得 $d_T(u, v) \leq \ell$。

首先，我们描述一下算法。算法首先任意选择树 T的一个顶点 $r \in V(T)$ 作为根。下面介绍算法中用到的一些记号。对于顶点 $v \in V(T)$，我们用 $deg(v)$ 表示 v 在 T 中的子节点数。用 T_v表示以 v为根的 T 的子树。此外，v的第 i 个子节点记为 ch_i^v。对于顶点 $v \in T$和 $0 \leq i \leq deg(v)$，我们定义 $T_{v,i}$ 为 T 的子树，其中包含顶点 v 以及以其前 i 个子节点（按索引排列）为根的子树。此外，$T_{\mathsf{ch}_j^v}$ 是 T 的子树，其中包含顶点 v 以及以其第 i个子节点为根的子树。回顾一下，每个顶点 $v \in V(T)$至多属于一个 $\mathcal{F}_i, i \in [t]$。对于每个顶点 $v \in V(T)$，我们按如下方式为 v赋值 $F(v) \in [0, t]$：如果 $v \in V(\mathcal{F}_i)$，则为 $F(v) = i$；否则（当任何 $\mathcal{F}_i, i \in [t]$ 中都不存在包含 v的路径时）为 0。回顾一下，若存在顶点 $u \in V(P)$，使得 $d_G(v, u) \leq \ell$，则路径 P 覆盖顶点 v。对于顶点 $v \in V(T)$，动态规划算法会针对 v的每个子节点考虑相应的子问题。我们按索引排列从左到右处理其子节点；处理到第 i个子节点时，考虑子树 $T_{v,i}$。在定义表项之前，我们先给出表项所用变量的定义。

考虑 $v \in V(T)$ 和 $i \in [deg(v)]$。为了计算子树$T_{v\text{中的}i}$部分解，我们必须区分 v 是否包含在解中的某条路径里。如果它包含在解中的某条路径里，那么我们还要猜测该路径的端点。为此，我们定义变量 tp，它可以取集合 $\{1, 2, \ldots, 11\}$ 中的一个值，其中每个值代表以下定义的不同情形：

– $\mathbf{tp}=\mathbf{1}$：包含 v 的路径不属于解。– $\mathbf{tp}=\mathbf{2}$：顶点 v 本身就是一条包含在解中的路径。– $\mathbf{tp} \in \{\mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{6}\}$：对于所有这些情形，解中都包含 v 的父节点，且满足各自的性质。

- $\mathbf{tp}=\mathbf{3}$：其端点之一就是 v 它本身。
- $\mathbf{tp}=\mathbf{4}$：其一个端点位于 $T_{v,\ i-1}$ $(\neq v)$。
- $\mathbf{tp}=\mathbf{5}$: 其中一个端点位于 $T_{\mathsf{ch}_i^v}$ $(\neq v)$。
- $\mathsf{tp}=\mathbf{6}$：其一个端点位于 $T\mathsf{ch}_j^v$ 中的某个 $j > i$ $(\neq v)$。–$\mathsf{tp}=\mathbf{7}$：其两个端点均位于$T_{v,i-1}$ （至少一个端点不等于 v）。–$\mathsf{tp}=\mathbf{8}$：其一个端点位于 $T_{v,i-1}$，另一个位于 $T\mathsf{ch}_i^v$ $(\neq v)$。–$\mathsf{tp}=\mathbf{9}$：其一个端点位于 $T_{v,i-1}$，另一个位于 $T\mathsf{ch}_j^v$ 中的某个 $j > i(\neq v)$。– $\mathsf{tp}=\mathbf{10}$：其一个端点位于 $T\mathsf{ch}_i^v$，另一个位于 $T\mathsf{ch}_j^v$ 中的某个 $j > i(\neq v)$。–$\mathsf{tp}=\mathbf{11}$：其一个端点位于 $T\mathsf{ch}_j^v$ 中的某个 $j > i$，另一个位于 $T\mathsf{ch}_{j'}^v$ 中的某个 $j' > i$ $(\neq v)$。

我们有一个变量 $(\mathsf{sg})\eta$。其中，sg 表示 η 的符号，可以是 $+$ 或 $-$，而 $(\mathsf{sg})\eta$ 可以取集合 $\{(-)\ell\ -1, \ldots, (-)1, (-)0, (+)0, (+)1, \ldots, (+)\ell\}$ 中的一个值。“–”符号表示 $V(T_{v,i}) \cap B$ 中有尚未覆盖的顶点，需要在后续覆盖；“+”符号表示 $V(T_{v,i}) \cap B$ 中的所有顶点都已覆盖，并且部分解还可以覆盖 $T_{v,i}$ 之外的更多顶点。值 $(-)\eta$ 表示在子树 $T_{v,i}$ 中，满足 u 未被部分解覆盖的顶点 $u \in B$ 到 v 的最大距离为 η。这意味着，$V(T_{v,i}) \cap B$ 中距离 v 至多为 η 的顶点，需要在后续由某些路径覆盖。我们只考虑到 $\ell - 1$ 距离，因为显然，在 $T_{v,i}$ 中，任何距离 v 至少为 ℓ 的顶点，都无法由不属于 $T_{v,i}$ 的路径覆盖。值 $(+)\eta$ 表示子树 $T_{v,i}$ 中顶点 u 的最小距离（u 不必属于 B），且从 u 到 v 的距离为 $\ell - \eta$，并且 u 包含在解中的一条路径里。这意味着，T 中距离 v 至多为 η（距离 u 至多为 ℓ）的 B 中的顶点，可以由解中一条包含 u 的路径覆盖。

在以下定义中，我们给出动态规划程序中的各项。

定义*3.* 对于每个 $v \in V(T)$、$d \in [deg(v)]$，$Y \subseteq [t]$，$tp \in \{1, 2, \ldots, 11\}$，$(sg)\eta \in \{(-)\ell - 1, \ldots, (-)1, (-)0, (+)0, (+)1, \ldots, (+)\ell\}$，我们创- 建一个条目 $c[v, d, Y, tp, (sg)\eta]$ 在我们的表中，当且仅当以下条件成立时，该条目存储 *true*：

– 存在一个由 $|Y|$ 条路径组成的集合 $\mathcal{P}_{v,d}$，使得对于每个 $j \in Y$，$|\mathcal{P}_{v,d} \cap \mathcal{F}_j| = 1$，若 $F(v) \in Y$，则 $\mathcal{P}_{v,d} \cap \mathcal{F}_{F(v)}$ 中的路径应满足 tp 的定义，并且 $\mathcal{P}_{v,d}$ 中的所有路径（可能包含 v 的路径除外）都应包含在子树 $T_{v,d}$ 中。

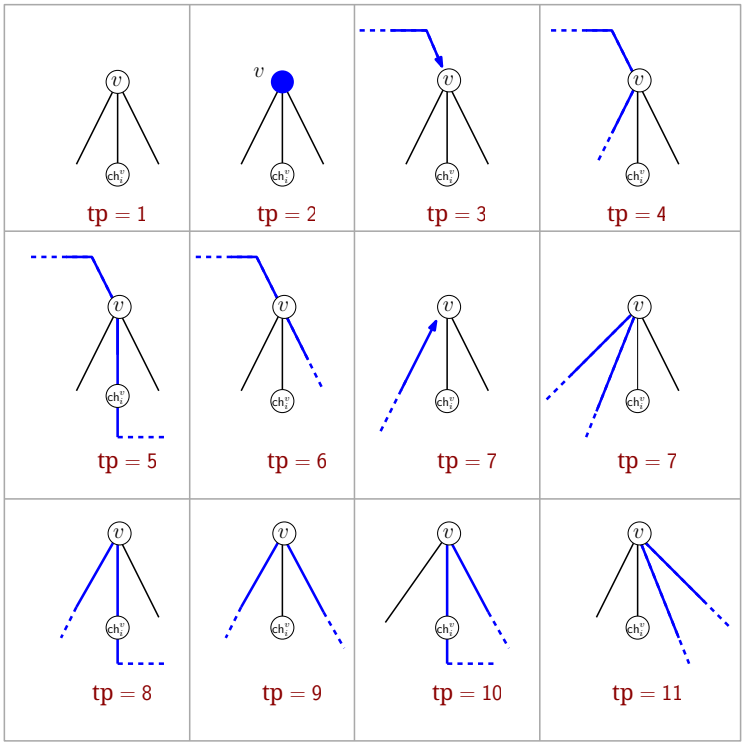

图4. 根据 tp 的值对情形进行说明。“蓝色”表示 $\mathcal{F}$ 中属于解路径一部分的路径。

– 如果sg 是 $-$，那么存在一个顶点 $u \in B$，它在T_v中距离 v 为 η，d 且不被$\mathcal{P}_v$中的任何路径覆盖，d 但将来可以被覆盖（与 v的距离不大于 ℓ）。对于 v 的子树T_v中距离至少为$\eta+1$ 的每个顶点 $u \in B$，d 都存在一条$\mathcal{P}_v$中的路径，d 该路径覆盖 u。

– 如果 sg 是 $+$，那么存在一个顶点 u，位于 $\mathcal{P}_{v,d}$，距离 $\ell-\eta$*from* v *in*$T_{v,d}$。并且 u 可以覆盖（未来）距离 η *from*v不超过该距离的顶点。$\mathcal{P}_{v,d}$ 中的所有其他顶点距离 $\ell-\eta$*from* v至少为该距离。此外，$T_{v,d} \cap B$ 中的所有顶点都由 $\mathcal{P}_{v,d}$覆盖。

否则 $c[v,d,Y,\text{tp }\eta]$ 存储 false。

观察可知，$(T,B,\ell,\mathcal{F} = \{\mathcal{F}_1,\mathcal{F}_2,\ldots,\mathcal{F}_t\})$ 是彩色路径覆盖的一个是实例，当且仅当存在一个被设为 true 的条目$c[r,deg(r)\ \{1,2,\ldots,t\},\text{tp}^*,(+)\eta^*]$, tp$^* \in \{1,2,7,8\}$和 $\eta^* \in [0,\ell]$。

接下来，我们给出用于计算我们的动态规划表中各项的（递归）公式。考虑 $v \in V(T)$ $d \in [deg(v)], Y \subseteq [t], \text{tp} \in 11], (\text{sg})\eta \in \{(-)\ell-1,\ldots,(-)1,(-)0,(+)0,(+)1,\ldots,(+)\ell\}$，我们根据以下情形计算项 $c[v,d,Y,\text{tp},(\text{sg})\eta]$：

叶顶点： v 是一个叶顶点。设置 $c[v,d,Y,\mathsf{tp},(\mathsf{sg})\eta]$ =为 true，当且仅当满足以下条件之一：(a1) $\mathsf{tp}=1$，$(\mathsf{sg})\eta=(-)0$，$v\in B$，或 (a2) $\mathsf{tp}\in\{2,3\}$，$(\mathsf{sg})\eta=(+)\ell$。否则，将该条目设为 false。正确性可由条目的定义直接得出。

非叶顶点： v 不是叶顶点。我们根据 tp 变量考虑以下情形：

Case $\mathsf{tp}=1$： 我们进一步考虑以下情形 es.

(1) 若 $\mathsf{sg}=-$，则当存在 Y 的一个划分 $Y_1\uplus Y_2$，使得以下条件之一成立时，将 $c[v,d,Y,1,(-)\eta]=$ 设为 true：(a1) $c[v,d-1,Y_1,1,(\mathsf{sg})\eta_1]=$ 为 true，$(\mathsf{sg})\eta_1\in\{(-)\eta,\ \ldots\ldots,\ (+)\eta-1\}$，且 (a2) $c[\mathsf{ch}_d^v,\ deg(\mathsf{ch}_d^v)\ Y_2,\ 1,\ (-)\eta-1]=$ 为 true；(b1) $c[v,d-1,Y_1,1,(-)\eta]=$ 为 true，且 (b2) $c[\mathsf{ch}_d^v,\ deg(\mathsf{ch}_d^v)\ Y_2,\ 1,\ (\mathsf{sg})\eta_2]=$ 为 true，$(\mathsf{sg})\eta_2\in\{(-)\eta-1,\ \ldots\ldots,\ (+)\eta\}$。

(1.1) 若 $\mathsf{sg}=+$，则当存在 Y 的一个划分$Y_1\uplus Y_2$，且满足以下条件之一时，令集合 $c[v,d,Y,1,(+)\ell-1]=$ 为 true：(a1) $c[v,d-1,Y_1,1,(\mathsf{sg})\eta_1]=$为 true，$(\mathsf{sg})\eta_1\in\{(-)\ell-1,\ \ldots,(+)\ell-1\}$，且 (a2) $c[\mathsf{ch}_d^v,deg(\mathsf{ch}_d^v)\ Y_2,\ \mathsf{tp}_2,(+)\ell]=$为 true，$\mathsf{tp}_2\in\{2,7,8\}$；或者 (b1) $c[v,d-1,Y_1,1,(+)\ell-1]=$ 为 true，且 (b2)$c[\mathsf{ch}_d^v,deg(\mathsf{ch}_d^v)\ Y_2,\ \mathsf{tp}_2,(\mathsf{sg})\eta_2]=$ 为 true，$\mathsf{tp}_2\in\{1,2,7,8\}$，$(\mathsf{sg})\eta_2\in\{(-)\ell-2,\ \ldots,(+)\ell\}$。

(1.2) 如果 $\mathsf{sg}=+$ 和 $\eta<\ell-1$，那么当存在一个 Y 的划分 $Y_1\uplus Y_2$，使得以下条件之一成立时，集合 $c[v,d,Y,1,(+)\eta]=$ 为真：(a1) $c[v,d-1,Y_1,1,(\mathsf{sg})\eta_1]=$为真，$(\mathsf{sg})\eta_1\in\{(-)\eta,\ \ldots\ldots,\ (+)\eta\}$，并且 (a2)$c[\mathsf{ch}_d^v,\ deg(\mathsf{ch}_d^v)\ Y_2,\ 1,\ (+)\eta+1]=$ 为真；或 (b1) $c[v,d-1,Y_1,1,(+)\eta]=$ 为真，并且 (b2) $c[\mathsf{ch}_d^v,\ deg(\mathsf{ch}_d^v)\ Y_2,\ 1,\ (\mathsf{sg})\eta_2]=$ 为真，$(\mathsf{sg})\eta_2\in\{(-)\eta,\ \ldots\ldots,\ (+)\eta+1\}$。

情形 $\mathsf{tp}=1$ (1) 的正确性：$\mathsf{sg}=-$：考虑(a1) 和(a2)均为真的情形，则存在一个路径集 $\mathcal{P}_1$，其中的路径使(a1)为真，并且存在一个路径集 $\mathcal{P}_2$，其中的路径使(a2) 为真。我们断言，$\mathcal{P}=\mathcal{P}_1\cup\mathcal{P}_2$ 是 $c[v,d,Y,1,(-)\eta]=$为真的一个解。注意，不存在包含 v 的路径属于 $\mathcal{P}$，因为 $\mathsf{tp}=1$ 在 (a1) 中。(a2) 条件意味着，存在一个未覆盖顶点 $u\in T_{\mathsf{ch}_d^v}$，其与 v之间的距离为 η。如果 $\mathsf{ch}_d^v\in\mathcal{P}_2$，那么 u 将被ch_d^v 覆盖，并且 u 不必等待后续顶点来覆盖它。因此，$\mathsf{tp}=1$ 在 (a2) 中。由于 $Y=Y_1\uplus Y_2$，对于每个 $i\in Y$，我们在 $\mathcal{P}\cap\mathcal{F}_i$ 中恰好有一条路径。接下来，注意在$T_{\mathsf{ch}_d^v}$中存在一个顶点 $u\in B$，其与 v 之间的距离为 η，该顶点未被 $\mathcal{P}_1$ 中的路径覆盖，并且 $\mathcal{P}_2$ 也未被覆盖。此外，$(\mathsf{sg})\eta_1\in\{(-)\eta,\ \ldots\ldots,\ (+)\eta-1\}$，$u$ 未被 $\mathcal{P}_1$ 覆盖，因为 $\mathcal{P}_1$ 中的任何顶点至少需要经过 $\eta+1$ 的距离才能到达 u。注意，在T_v 中，与 v 的距离至少为 $\eta+1$ 的每个顶点 d 都被 $\mathcal{P}$中的路径覆盖。当 (b1) 和 (b2) 均为真时，同样的论证也成立。

情形的正确性$\mathsf{tp}=1$ (1.1)： $\mathsf{sg}=+$：由于 $\mathsf{tp}=1$, $v\notin\mathcal{P}$。因此，我们不考虑条目 $c[v,d,Y,1,(+)\ell]$。考虑(a1)和(a2) 为真的情形，则存在一个路径集 $\mathcal{P}_1$，使 (a1)为真，并且存在一个路径集 $\mathcal{P}_2$，使 (a2) 为真。我们断言，$\mathcal{P}=\mathcal{P}_1\cup\mathcal{P}_2$ 是 $c[v,d,Y,1,(+)\ell-1]=$ 为真的一个解。由于 $\mathsf{tp}_2\in\{2,7,8\}$，且 $\mathcal{P}_1$ 中没有路径包含 v，因此不包含 v 的路径不在 $\mathcal{P}$ 中。由于 $Y=Y_1\uplus Y_2$，我们有，对于每个 $i\in Y$，$\mathcal{P}\cap\mathcal{F}_i$ 中恰好有一条路径。接下来，注意到 $T_{\mathsf{ch}_d^v}$中的每个顶点都被 $\mathcal{P}_2$ 覆盖，并且由于 $(\mathsf{sg})\eta_1\in\{(-)\ell-1,\ldots,(+)\ell-1\}$，$T_{v,d-1}$ 中的每个顶点都被 $\mathcal{P}_1$ 或 $\mathcal{P}_2$ 覆盖。根据 (a2)，我们还满足 $(+)\ell$，即

距离至多为 ℓ 的顶点由 $\mathcal{P}_2$ 覆盖。当 (b1) 和 (b2) 为真时，也可用类似的论证。

正确性： $\mathsf{tp} = 1$ **(1.2):** $\mathsf{sg} = +$: 由于 $\eta < \ell - 1$，v 的所有子节点都不能位于 $\mathcal{P}$ 中。因此$\mathsf{tp} = 1$ 在 (a1)和 (a2) 中都成立。考虑(a1)和 (a2)均为真情形，则存在一个路径集 $\mathcal{P}_1$ 使(a1)为真，并且存在一个路径集 $\mathcal{P}_2$ 使 (a2)为真。我们断言，$\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2$ 是 $c[v, d, Y, 1, (+)\eta] =$为真的一个解。由于 $Y = Y_1 \uplus Y_2$，对于每个$i \in Y$， $\mathcal{P} \cap \mathcal{F}_i$中恰好有一条路径。接下来，注意到$T_{\mathsf{ch}_d^v}$中的每个顶点都被 $\mathcal{P}_2$ 覆盖，并且由于 (sg)$\eta_1 \in \{(-)\eta, .., (+)\eta\}$，$T_{v,d-1}$ 中的每个顶点都被 $\mathcal{P}_1$ 或 $\mathcal{P}_2$ 覆盖。根据 (a2)，我们还满足 $(+)\eta$，也就是说，距离至多为 η 的顶点被 $\mathcal{P}_2$覆盖。当 (b1) 和 (b2) 为真时，类似的论证同样成立。

情形 $\mathsf{tp} = 2$: 如果 $F(v) \notin Y$，或 $\mathsf{sg} =$ (–)，或 $\eta \neq \ell$，或 v 不是$\mathcal{F}_{F(v)}$中的路径，则将 $c[v, d, Y, 2,$ (sg)$\eta] =$设为 false。正确性显然成立。否则，如果存在 $Y \setminus \{F(v)\}$的一个划分 $Y_1 \uplus Y_2$，使得以下条件成立，则我们将 $c[v, d, Y, 2, (+)\ell] =$设为 true：(a1) $c[v, d-1, Y_1 \cup \{F(v)\},$ 2, $(+)\ell] =$为 true，且 (a2) $c[\mathsf{ch}_d^v,$ $deg(\mathsf{ch}_d^v)$ $Y_2,$ $\mathsf{tp}_2,$ (sg)$\eta_2] =$为 true，(sg)$\eta_2 \in \{(-)\ell - 1,$ ……, $(+)\ell\}$，$\mathsf{tp}_2 \in \{1, 2, 7, 8\}$。

情形的正确性 $\mathsf{tp} = 2$**：** 假设 (a1) 和 (a2) 为真，则存在一个路径集 $\mathcal{P}_1$ ，其使 (a1)为真，并且存在一个路径集 $\mathcal{P}_2$，其使 (a2) 为真。我们断言，$\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2 \cup \{v\}$ 是$c[v, d, Y, 2, (+)\ell] =$为真的一个解。由于 $Y = Y_1 \uplus Y_2 \uplus \{F(v)\}$，对于每个$i \in Y$，我们恰好在 $\mathcal{P} \cap \mathcal{F}_i$ 中有一条路径。接下来，注意到 $T_{\mathsf{ch}_d^v} \cup T_{v,\ d-1}$ 中的每个顶点都被 $\mathcal{P}$ 覆盖，因为 (sg)$\eta_2 \in \{(-)\ell - 1,$ ……, $(+)\ell\}$，且包含 v 的路径位于 $\mathcal{P}$ 中，因此$T_{v,\ d}$ 中的每个顶点都被 $\mathcal{P}$ 覆盖。情形 $\mathsf{tp} = 3$: 如果 $F(v) \notin Y$，或 $\mathsf{sg} =$ (–)，或 $\eta \neq \ell$，或 $\mathcal{F}_{F(v)}$ 中不存在包含 v、v 的父节点且其某个端点为 v的路径，则设置 $c[v, d, Y, 3,$ (sg)$\eta] =$ false。正确性显然成立。否则，如果存在 $Y \setminus \{F(v)\}$的一个划分 $Y_1 \uplus Y_2$，使得以下条件成立，则我们设置$c[v, d, Y, 3, (+)\ell] =$ true：(a1) $c[v, d-1, Y_1 \cup \{F(v)\},$ 3, $(+)\ell] =$ true，以及 (a2) $c[\mathsf{ch}_d^v,$ $deg(\mathsf{ch}_d^v)$ $Y_2,$ $\mathsf{tp}_2,$(sg)$\eta_2] =$ true，$\mathsf{tp}_2 \in \{1, 2, 7, 8\}$，(sg$\eta_2 \in$ {)(–)ℓ –1，. . . ，(+)ℓ}。正确性可类似于情形 2 进行论证。情形 $\mathsf{tp} = 4$: 如果 $F(v) \notin Y$，或 $\mathsf{sg} =$ (–)，或 $\eta \neq \ell$，或者在 $\mathcal{F}_{F(v)}$ 中不存在包含 v、v 的父节点且其一个端点位于 $T_{v,\ d-1}$ （端点不是 v）的路径，则设置 $c[v, d, Y, 4,$ (sg)$\eta] =$ false。正确性显然成立。否则，如果存在 $Y_1 \uplus Y_2$ 对 $Y \setminus \{F(v)\}$ 的划分，使得以下条件成立，则设置 $c[v, d, Y, 4, (+)\ell] =$ true：(a1) $c[v, d-1, Y_1 \cup \{F(v)\},$ 4, $(+)\ell] =$ true，以及 (a2) $c[\mathsf{ch}_d^v$、$deg(\mathsf{ch}_d^v)$ Y_2、tp_2,(sg)$\eta_2] =$ true、$\mathsf{tp}_2 \in \{1, 2, 7, 8\}$、(sg) $\eta_2 \in$ {(–)ℓ –1、. ..、(+)ℓ}。正确性可通过类似情形 2 的方式论证。

情形 $\mathsf{tp} = 5$: 如果 $F(v) \notin Y$，或 $\mathsf{sg} =$ (–)，或 $\eta \neq \ell$，或在 $\mathcal{F}_{F(v)}$中不存在包含 v、作为 v 的父节点且其一个端点位于 $T_{\mathsf{ch}_d^v}$ 中的路径（端点不是v），则将 $c[v, d, Y, 5,$ (sg)$\eta] =$设为 false。正确性显然成立。否则，如果存在 $Y_1 \uplus Y_2$对 $Y \setminus \{F(v)\}$ 的一个划分，使得以下条件成立，则将 $c[v, d, Y, 5, (+)\ell] =$设为true：(a1) $c[v, d-1, Y_1 \cup \{F(v)\},$ $\mathsf{tp}_1,$ $(+)\ell] =$为 true，$\mathsf{tp}_1 = 6$，以及 (a2) $c[\mathsf{ch}_d^v,$ $deg(\mathsf{ch}_d^v)$ $Y_2 \cup \{F(v)\},$ $\mathsf{tp}_2,$ $(+)\ell] =$为 true，$\mathsf{tp}_2 \in \{3, 4, 5\}$。

情形的正确性 $\mathsf{tp}=5$**:** 假设(a1)和 (a2) 为真，则存在一个路径集 $\mathcal{P}_1$，将(a1) 设为真，并且存在一个路径集 $\mathcal{P}_2$，将 (a2) 设为真。令 $P_1\in\mathcal{P}_1\subseteq\mathcal{F}_F(v)$，并令 $P_2\in\mathcal{P}_1\subseteq\mathcal{F}_F(v)$。注意到，$\mathcal{P}_1^*=(\mathcal{P}_1\setminus\{P_1\})\cup\{P_2\}$ 也会将 (a1) 设为真。我们断言，$\mathcal{P}=\mathcal{P}_1^*\cup\mathcal{P}_2$也是 $c[v,d,Y,5,(+)\ell]=$为真的一个解。由于 $Y=Y_1\uplus Y_2\uplus\{F(v)\}$，且 $\mathsf{tp}_1=6$ 和$\mathsf{tp}_2\in\{3,4,5\}$，并且恰好有一条包含 v 的路径属于 $\mathcal{P}$，因此对于每个 $i\in Y$，在 $\mathcal{P}\cap\mathcal{F}_i$ 中恰好有一条路径。接下来，注意到 $T_{\mathsf{ch}_d^v}\cup T_{v,\ d-1}$ 中的每个顶点都被 $\mathcal{P}$ 覆盖，因此 $T_{v,\ d}$ 中的每个顶点都被$\mathcal{P}$覆盖。

情形 $\mathsf{tp}=6$: 如果 $F(v)\notin Y$，或者 $d=deg(v)$，或者 $\mathsf{sg}=(-)$，或者 $\eta\neq\ell$，或者 $\mathcal{F}_{F(v)}$ 中不存在一条包含 v、v 的父节点且有一个端点位于 $T_{\mathsf{ch}_{d'}^v}, d'>d$（端点不是 v）的路径，则设 $c[v,d,Y,6,(\mathsf{sg})\eta]=$ 为 false。正确性显然成立。否则，如果存在 $Y_1\uplus Y_2$ 对 $Y\setminus\{F(v)\}$ 的一个划分，使得满足以下条件，我们就设 $c[v,d,Y,6,(+)\eta]=$ 为 true：(a1) $c[v,d-1,Y_1\cup\{F(v)\},\ \mathsf{tp}_1,(+)\ell]=$为 true，且 (a2) $c[\mathsf{ch}_i^v,deg(\mathsf{ch}_i^v)\ Y_2,\ \mathsf{tp}_2,(+)\ell]=$ 为 true，其中 $\mathsf{tp}_1,\mathsf{tp}_2=6$。其正确性可仿照情形 5 进行论证。

情形 $\mathsf{tp}=7$:如果 $F(v)\notin Y$，或 $\mathsf{sg}=(-)$，或 $\eta\neq\ell$，或在 $\mathcal{F}_{F(v)}$中不存在包含 v的路径，且其两个端点均在$T_{d,i-1}$ 中（至少一个端点不是 v），则将 $c[v,d,Y,7,(\mathsf{sg})\eta]=$设为 false。正确性显然成立。否则，如果存在 $Y\setminus\{F(v)\}$ 的一个划分 $Y_1\uplus Y_2$，使得以下条件成立，则将$c[v,d,Y,7,(+)\ell]=$设为true：(a1) $c[v,d-1,Y_1\cup\{F(v)\},\ \mathsf{tp}_1,(+)\ell]=$为 true，$\mathsf{tp}_1\in\{7,8\}$，并且 (a2) $c[\mathsf{ch}_d^v,deg(\mathsf{ch}_d^v)\ Y_2,\ \mathsf{tp}_2,(\mathsf{sg})\eta_2]=$为 true，$\mathsf{tp}_2\in\{1,2,7,8\}$，$(\mathsf{sg})\eta_2\in\{(-)\ell-1,\ ...,\ (+)\ell\}$。正确性可用类似于情形 5 的方式论证。

情形 $\mathsf{tp}=8$: 如果 $F(v)\notin Y$，或 $\mathsf{sg}=(-)$，或 $\eta\neq\ell$，或者在 $\mathcal{F}_{F(v)}$中不存在包含 v 且其一个端点在$T_{d,\ i-1}$ 中而另一个端点在 $T_{\mathsf{ch}_d^v}$ 中的路径（端点不是 v），则将 $c[v,d,Y,8,(\mathsf{sg})\eta]=$设为 false。正确性显然成立。否则，如果存在一个 $Y_1\uplus Y_2$ 对 $Y\setminus\{F(v)\}$ 的划分，使得以下条件成立，则将 $c[v,d,Y,8,(+)\ell]=$设为true：(a1) $c[v,d-1,Y_1\cup\{F(v)\},\ \mathsf{tp}_1,(+)\ell]=$为 true，$\mathsf{tp}_1=9$ 以及 (a2) $c[\mathsf{ch}_i^v,\ deg(\mathsf{ch}_i^v)\ Y_2\cup\{F(v)\},\ \mathsf{tp}_2,(+)\ell]=$为 true，$\mathsf{tp}_2\in\{3,4,5\}$，$(\mathsf{sg})\eta_2\in\{(-)\ell-1,\ .....,\ (+)\ell\}$。正确性可类似于情形 5 进行论证。

情形$\mathsf{tp}=9$: 如果 $F(v)\notin Y$，或 $d=deg(v)$或 $\mathsf{sg}=(-)$，或 $\eta\neq\ell$，或者在 $\mathcal{F}_{F(v)}$ 中不存在包含 v的路径，且其一个端点在 $T_{d,i-1}$ 中而另一个端点在$T_{\mathsf{ch}_{d'}^v}$中，对于某个 $d'>d$（端点不是 v），则将 $c[v,d,Y,9,(\mathsf{sg})\eta]=$设为 false。正确性显然成立。否则，如果存在 $Y\setminus\{F(v)\}$的一个划分 $Y_1\uplus Y_2$，使得以下条件成立，则将 $c[v,d,Y,9,(+)\ell]=$设为 true：(a1) $c[v,d-1,Y_1\cup\{F(v)\},\ \mathsf{tp}_1,(+)\ell]=$true，$\mathsf{tp}_1=9$ 以及 (a2) $c[\mathsf{ch}_d^v,deg(\mathsf{ch}_d^v)\ Y_2,\ \mathsf{tp}_2,(+)\ell]=$true，$\mathsf{tp}_2\in\{1,2,7,8\}$，$(\mathsf{sg})\eta_2\in\{(-)\ell-1,\ ..,\ (+)\ell\}$。正确性可以用与情形 5 类似的方式论证。

情形 $\mathsf{tp}=10$**:** 如果 $F(v)\notin Y$，或 $d=deg(v)$，或 $\mathsf{sg}=(-)$，或 $\eta\neq\ell$，或在 $\mathcal{F}_{F(v)}$中不存在包含 v的路径，且其一个端点位于 $T_{\mathsf{ch}_d^v}$，另一个端点对于某个 $d'>d$位于 $T_{\mathsf{ch}_{d'}^v}$（该端点不是 v），则设 $c[v,d,Y,10,(\mathsf{sg})\eta]=$false。正确性显然成立。否则，若存在 $Y\setminus\{$的一个划分 $Y_1\uplus Y_2$，使得满足以下条件，则设 $c[v,d,Y,10,(+)\ell]=$true：(a1) $c[v,d-1,Y_1\cup\{F(v)\},\ \mathsf{tp}_1,(+)\ell]=$true，$\mathsf{tp}_1=11$ 且 (a2) $c[\mathsf{ch}_d^v,\ deg(\mathsf{ch}_d^v)\ Y_2\cup$

$\{F(v)\}$，$\mathsf{tp}_{2'}(+)\ell] = \mathsf{true}$，$\mathsf{tp}_2 = 10$。正确性论证与情形 5 类似。

情形 $\mathsf{tp} = 11$：如果 $F(v) \notin Y$，或 $d = deg(v)$，或 $\mathsf{sg} = (-)$，或 $\eta \neq \ell$，或者在 $\mathcal{F}_{F(v)}$ 中不存在包含 v 的路径，其中一个端点属于某个 $d' > d$ 对应的 $T_{\mathsf{ch}^v_{d'}}$，另一个端点属于某个 $d'' > d$ 对应的 $T_{\mathsf{ch}^v_{d''}}$（端点不是 v），则将 $c[v, d, Y, 11, (\mathsf{sg})\eta] =$ 设为 false。正确性显而易见。否则，如果存在 $Y_1 \uplus Y_2$ 对 $Y \setminus \{F(v)\}$ 的一个划分，使得满足以下条件，则将 $c[v, d, Y, 11, (+)\ell] =$ 设为 true：(a1) $c[v, d-1, Y_1 \cup \{F(v)\}, \mathsf{tp}_{1'}(+)\ell] =$ 为 true，$\mathsf{tp}_1 = 11$，且 (a2) $c[\mathsf{ch}^v_d, deg(\mathsf{ch}^v_d)\ Y_2, \mathsf{tp}_{2'}(+)\ell] =$ 为 true，$\mathsf{tp}_2 \in \{1, 2, 7, 8\}$，$(\mathsf{sg})\eta_2 \in \{(-)\ell - 1, \ldots, (+)\ell\}$。正确性可参照情形 5 进行论证。

至此，我们完成了对用于计算动态规划表所有条目的（递归）公式及其正确性的描述。算法的正确性由（递归）公式的正确性，以及以下事实得出：$(T, B, \ell, \mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_t\})$ 是 Colorful Path-Cover 的是实例，当且仅当存在一个条目 $c[r, deg(r), \mathsf{tp}^*, (+)\eta^*]$，$\mathsf{tp}^* \in \{1, 2, 7, 8\}$ 和 $\eta^* \in [0, \ell]$，其值被设为 true。接下来，我们分析算法的运行时间。回顾一下，$|\mathcal{F}| \leq t$，并且对于每个 $i \in [t]$，都有 $|\mathcal{F}_i| \leq n^2$。注意，我们的每个表条目都可以在 $\mathcal{O}(\ell^2 \cdot 2^{2t} n^{\mathcal{O}(1)})$ 时间内计算得出。条目数上界为 $\mathcal{O}(\ell \cdot 2^t n^{\mathcal{O}(1)})$。

由于我们在 $\mathcal{F}$ 中于 $\mathcal{O}(k^t)$时间内猜出了来自每棵树的可行路径族子集，我们算法的运行时间上界为$\mathcal{O}(\ell^2 \cdot 2^{\mathcal{O}(t \log k)} n^{\mathcal{O}(1)})$。由于 $t \leq k$，我们得到了所需的运行时间。

3.7 定理的证明1

根据引理3，ESP/fvs + ecc 与骨架测试 FPT等价。观察1将骨架的数量上界限定为 $2^{\mathcal{O}(k(\log k + \log \ell))} n^2$。随后，我们证明骨架测试与扩展骨架测试 FPT等价，并且对于每个骨架，至多有 $2^{\mathcal{O}(k \log k)}$ 个扩展骨架。最后，给定一个扩展骨架测试实例，我们在多项式时间内构造一个彩色路径覆盖实例。彩色路径覆盖问题可在 $\mathcal{O}(\ell^2 \cdot 2^{\mathcal{O}(k \log k)} n^{\mathcal{O}(1)})$ 时间内求解，由此完成定理 1 的证明。

4 $(1+\epsilon)$-因子算法，参数化依据为反馈顶点集

定理 2. 对于任意 $\epsilon > 0$，存在一种适用于 ESP/fvs 的 $(1+\epsilon)$-因子近似算法，运行时间为 $\mathcal{O}(2^{\mathcal{O}(k \log k)} n^{\mathcal{O}(1)})$。

我们使用定理1 中的算法，其运行时间为 $\mathcal{O}(2^{\mathcal{O}(k \log k)} \ell^k n^{\mathcal{O}(1)})$。注意，$\ell^k$ 的来源是骨架的数量（观察1）。具体而言，对于函数 f：$X \to [\ell]$，它维护了 S 中未出现在 P上的顶点集合的距离剖面。为了设计一个$(1+\epsilon)$-因子 FPT 近似算法，我们使用 ϵ 将像集 $[\ell]$ 替换为一个固定大小的集合，使得我们能够近似集合中 S

中未出现在 P上的顶点集合，近似因子为$(1+\epsilon)$。其余部分与定理1类似。下面我们详细描述该结果、过程及其正确性。令函数 $f:\ X \to \{\epsilon\ell, \ell\}$ 表示每个顶点到假设解 $x \in X$ 的近似最短距离，该假设解是 P的一个ESP/fvs解。形式化地，

$$f(v) = \begin{cases} \epsilon\ell & \text{if } d_G(v,P) < \epsilon\ell, \\ \ell & \text{if } \epsilon\ell \le d_G(v,P) \le \ell. \end{cases}$$

正确性。假设 P^* 是一条最短路径，其离心率为 ℓ，且 f 是定理 1的证明中定义的函数，并由定理 1中的算法返回。我们证明，对于每个顶点 $v \in V$（G），d_G（v，P）$\le$（$1+\epsilon$）ℓ。注意，对于顶点 $x \in X$，如果 $1 \le d_G$（x，P^*）$< \epsilon\ell$，那么对于 f的正确猜测，f（x）$= \epsilon\ell$ 且 d_G（x，P^*）$< \epsilon\ell$。此外，如果 $\epsilon\ell \le d_G$（x，P^*）$\le \ell$，那么对于 f的正确猜测，f（x）$= \ell$ 且 d_G（x，P^*）$\le \ell$。回顾一下，在算法中，当我们为一个良好函数 γ 构造实例（归约到彩色路径覆盖的实例）时，我们会移除这些顶点，以构造一个彩色路径覆盖实例。我们的假设（或猜测）是，使用 x 可以满足 v 的离心率要求。更明确地说，我们使用以下条件：如果 f（x）$= \epsilon\ell$（分别地，f（x）$= \ell$），那么当 d_G（v，x）$\le \ell$（分别地，d_G（v，x）$\le \epsilon\ell$）时，使用 x 可以满足顶点 v 的离心率要求。现在考虑顶点 $v \in V$（G）$\setminus S$。假设存在一条从 v到 P^*的最短路径，其中不包含来自 S 的任何顶点，那么根据定理1 的算法描述及其正确性，我们得到 d_G（v，P）$\le \ell$。接下来，假设从 v 到 P^*的最短路径包含一个顶点 $x \in X$，那么 d_G（v，x）$+ d_G$（x，P^*）$\le \ell$。因此，对于这样的顶点，当 d_G（x，v）$\le \ell$ 且 d_G（x，P）$< \epsilon\ell$时，我们得到d_G（v，P）$\le d_G$（x，v）$+ d_G$（x，P）$\le \ell + \epsilon\ell =$（$1+\epsilon$）$\ell$；类似地，如果 d_G（x，v）$\le \epsilon\ell$ 且 d_G（x，P）$\le \ell$，那么 d_G（v，P）$\le d_G$（x，v）$+ d_G$（x，P）$\le \epsilon\ell + \ell =$（$1+\epsilon$）$\ell$。这就完成了定理 2 证明的正确性。

5 不相交路径删除集

在本节中，我们设计一个以不相交路径删除集 (dpd) 为参数的 ESP 问题 FPT 算法。

定理 3. 存在一个算法，可在时间$\mathcal{O}(2^{\mathcal{O}(k \log k)} n^{\mathcal{O}(1)})$内求解 ESP/dpd 。

我们使用 ESP/fvs + ecc 的算法（定理 1），其运行时间为 $\mathcal{O}(2^{\mathcal{O}(k \log k)} \ell^k n^{\mathcal{O}(1)})$。请注意，离心率参数 ℓ会出现在运行时间中，这是由于骨架的大小，其中包含 ℓ^k（观察1）项。这个 ℓ^k 项的出现，是因为函数 $f:\ X \to [\ell]$，该函数用于维护集合 S 中不出现在 P 上的顶点的距离剖面。对于 dpd 的情形，我们可以证明存在一个集合 $Q \subseteq [\ell]$，满足 $|Q| \le 2k^2$（引理 9 将在下文证明）。我们可以定义一个函数 $f:\ X \to Q$，用于维护集合 S 中不出现在 P 上的顶点的距离剖面。算法的其余部分与定理 1 中的算法完全相同。因此，我们得到以下结果。

引理 9。 设 (G, S, k) 是 *ESP*/dpd 的一个是实例，且 P 是一个假设解。那么存在一个大小为 $\leq 2k^2$ 的集合 $Q \subseteq [\ell]$，使得对于每个 $w \in S$,$d_G(w, P) \in Q$。此外，可以在 $O(k^2n^2)$ 时间内构造出这样的一个 Q。

证明。令 Q 为按 $Q = \{d_G(x, y)\ d_G(x, y)+1, d_G(x, y)-1\colon\ x, y \in S\}$ 定义的集合。显然，$|Q| \leq 2k^2$ 和集合 Q 可在 $O(k^2n^2)$ 时间内计算出来。现在只需证明，对于 ESP/dpd 问题的任意一个是实例 (G, S, k) 的任意解路径 P，对每个 $w \in S, d_G(w, P)$，均有 $\in Q$。首先，注意到若 $w \in S \cap P$，则 $d_G(w, P) = 0$，且显然值 $0 \in Q$，因为对任意 $x \in S$，$d(x, x) = 0$。现在，对每个 $w \in S \setminus P$，令 P_w 为一条从 w 到 P 的最短路径，并令 $w^* = P \cap P_w$ 和 z 为 P_w 上距离 w^* 最近的、属于 S 的顶点。显然，$z \in S \cap P_w$ 且 $d_G(w, P) = d_G(w, z) + d_G(z, w^*)$。令 $M = P \cap S$。现在考虑以下三种情形。（这些情形见图 5）

情形 1. 如果 $w^* \in S$，那么 $z = w^*$。而值 $d_G(w, P)$ 本质上就是 $d_G(w, w^*)$。由于 $w, w^* \in S$，所以 $d_G(w, w^*) \in \{d_G(x, y)\colon\ x, y \in S\} \subseteq Q$。

情形 2. 如果 $w^* \notin S$且 $w^* \notin N(M)$，那么必定存在某一对 (m_j, m_{j+1}) 的一个子路径 P_j，使得 $w^* \in P_j$。由于 w^* 不是子路径P_j的端点，因此我们有 $d_G(z, w^*) = 1$ 且 $d_G(w, w^*) = d_G(w, z) + 1$。由于 $w, z \in S$，所以$d_G(w, w^*) \in \{1 + d_G(x, y)\colon\ x, y \in S\}$ 它是 Q 的子集。

情形 3. 如果 $w^* \notin S \cap P$和 $w^* \in N(M)$，那么某对 (m_j, m_{j+1}) 必须存在一个子路径 P_j，使得 $w^* \in P_j$。由于 w^* 是子路径 P_j的一个端点，我们有值 $d_G(w, w^*)$ 必须是 $d_G(w, m_j) - 1$ 或$d_G(w, m_{j+1}) - 1$。由于 $w, m_j, m_{j+1} \in S$，所以 $d_G(w, w^*) \in \{d_G(x, y) - 1\colon\ x, y \in S\}$，它是 Q 的一个子集。□

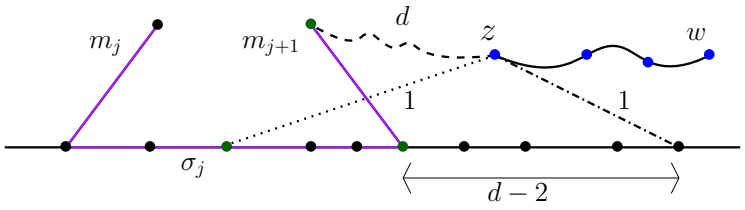

图 5。 引理 9 的证明示意图。虚线、点线和点划线分别描述情形 1、2 和 3。绿色顶点表示相应情形中的顶点 w'。蓝色顶点表示顶点集 Z。 m_j 与 m_{j+1} 之间的子路径表示为 P_j。

6 Split VertexDeletion Set

在本节中，我们设计一个以分裂顶点删除集为参数的 ESP 问题的 FPT 算法。令 (G, S, k, ℓ)为 ESP/svd的是实例， P 为一个假设解，其中 S 是一个分裂顶点删除

集合。我们的主要目标是得到 P。为此，我们的主要思路是将 S 中未出现在 P 中的顶点划分为常数个部分，并确保任何遵循该划分的解（若存在）都能给出一个解。具体而言，我们将 S 中未出现在 P 中的顶点划分为五个互不相交的集合 X，$X_{=2}$，$X_{=3}$，$X_{=4}$ 和 $X_{\geq 5}$，使得 $X_{=i}$，$i \in [4]$ 是 X 中与假设解路径距离恰好为 i 的顶点集合，而 $X_{\geq 5}$ 是与假设解路径距离至少为五的顶点集合。其余过程将在下文明确描述。本节的主要结果如下。

定理4。存在一个用于ESP/svd 的算法，其运行时间为 $O(2^{\mathcal{O}(k \log k)} n^{O(1)})$。

证明。考虑 ESP/svd 的一个实例 (G, S, k)。设 $C \uplus I$ 是 $V(G - S)$ 的划分，使得 C 和 I 分别诱导出一个团和一个独立集。首先，我们描述算法。我们假设 G 是一个连通图，否则它是一个平凡的否实例。算法首先猜测一个集合 $M \subseteq S$，它是 S 与假设解的顶点集的（精确）交集。注意，$|V(P) \cap C| \leq 2$，因为图中的每条最短路径也是图中的一条诱导路径。算法猜测一个大小至多为 2 的集合 $C^* \subseteq V(C)$，它是 C 与假设解的顶点集的（精确）交集。接下来，算法使用引理 2 找出假设解路径上 $M \cup C^*$ 中顶点的（唯一）排列 π。设 $|M \cup C^*| = t$ 和 $\{m_1, m_2, \ldots, m_t\}$ 分别为假设解路径中 $M \cup C^*$ 顶点的猜测排列 π，并设 $X = S \setminus M$。接下来，我们猜测解路径的端点。设 m_0、m_{t+1} 为假设解路径的猜测端点，使得 m_0 是假设解路径上的第一个顶点，而 m_{t+1} 是最后一个顶点（我们也可能得到 m_1、m_{t+1}，使得 $m_0 = m_1$ 或 $m_t = m_{t+1}$，或 $m_0 = m_{t+1}$）。注意，对于正确的猜测 M、C^*、m_0、m_{t+1} 以及排列 m_0、m_1、.....、m_{t+1}，必须满足以下条件，否则我们就丢弃排列 M、C^*、m_0、m_{t+1} 的组合，并进行另一组猜测。如果 $m_i m_j$ 是 G 中的一条边，那么 $j = i + 1$。我们假设最小离心率大于四；否则，我们可以在多项式时间内解决该问题（根据引理1）。

接下来，我们猜测集合 X 的一个划分 $\mathcal{X}$，将其划分为 5个不相交的集合 $X_{=1}$、$X_{=2}$、$X_{=3}$、$X_{=4}$和 $X_{\geq 5}$，使得 $X_{=i}$、$i \in [4]$ 是 S 中与假设解路径的距离恰好为 i 的顶点集合，而 $X_{\geq 5}$ 是与假设解路径的距离至少为五的顶点集合。对于一个顶点 $v \in X_{=i}$，$i \in [4]$（分别为 $X_{\geq 5}$），如果 $d_G(v, m_j) < i$（分别为 $d_G(x, m_j) < 5$），则对于某个 $j \in [0, t+1]$，我们舍弃这样的划分，并猜测 X 的另一个划分。令 $X' = X_{=1} \cup X_{=2} \cup X_{=3} \cup X_{=4} \subseteq X$ 为 X 中满足以下条件的所有顶点的集合：对于每个 $v \in X'$，$d_G(v, m_j) > i$，其中 $v \in X_{=i}$，且对于所有 $j \in [0, t+1]$，$i \in [4]$。在这种情况下，对于 X' 中的每个顶点，假设解中都应存在一个 m_j 中的顶点，使其到 m_{j+1} 子路径来自 $G[(V(G) \setminus S) \cup \{m_j, m_{j+1}\}]$，其中某个 $j \in [0, t+1]$。回顾一下，对于每个 $j \in [0, t]$，如果 $m_j m_{j+1}$ 是 G 中的一条边，那么 $m_j m_{j+1}$ 也是假设解中的一条边。

接下来，我们猜测一个函数 $g:\ X' \to [0,t]$。对于一个顶点 $v \in X'$，$g(v)=j$ 表示所猜测的对(m_j,m_{j+1})，使得存在一个顶点$u \in I$ 位于 m_j 到 m_{j+1} 子路径中，且 $d_G(u,v) = i$，如果 $v \in X_{=i}$。如果函数g 满足以下条件，那么我们舍弃对 g的这一猜测。对于某些 $j \in [0,t]$ 和$v \in g^{-1}(j)$，不存在任何满足以下条件的顶点 $u \in I$：(i) u 同时邻接于 m_j和 m_{j+1}，其中 $(m_j,m_{j+1}) \notin E$；(ii) $d(u,v) = i$，其中 $v \in X_{=i}$对于某些 $i \in [4]$成立。对于每个 $M \subseteq S$、$X = S \setminus M$、$C^* \subseteq C$、m_0、$m_{|M \cup C^*|+1} \in V(G)$，按照上述定义的 $\mathcal{X} = X_{=1} \uplus X_{=2} \uplus X_{=3} \uplus X_{=4} \uplus X_{\geq 5}$ 对 X、X' 的划分，以及如上所述且未根据上述描述被舍弃的函数 $g:\ X' \to [0,t]$，我们执行如下操作：对于每个 i，使得 $m_i m_{i+1}$ 不是 G中的一条边，我们任意选取一个顶点 $u \in I$，使得 u 同时邻接于 m_j 和 m_{j+1}，并且只要 $v \in X_{=i}$ 和 $v \in g^{-1}(j)$同时成立，就有 $d_G(v,u) = i$。至此，算法的描述完毕。对于每个 $j \in [0,t]$，我们通过连接 $m_i m_{i+1}$ 边或 $m_j u m_{j+1}$ 子路径来拼接得到路径 P，并输出该路径。

运行时间：接下来，我们分析算法的运行时间。S 的子集数量为 2^k，这为 M 提供了 2^k 种选择。C^*、m_0、$m_{|M|+1}$ 的选择数量至多为 $\mathcal{O}(n^4)$。为了找到 $M \cup C^*$ 的排列（排序），我们只有一种选择，根据引理2，该选择可以在 $\mathcal{O}(n^{\mathcal{O}(1)})$时间内找到。划分 $\mathcal{X}$ 的选择数量上界为 $\mathcal{O}(5^k)$。函数 g 的选择数量上界为 $\mathcal{O}(k^k)$。因此，我们的算法运行时间为$\mathcal{O}(2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)})$。

正确性：接下来，我们证明算法的正确性。我们将证明，最短路径 P 是 ESP/svd 中 (G, S, k) 的一个解，当且仅当存在一个 $M \subseteq S$, $X = S \setminus M$, $C^* \subseteq C$, m_0, $m_{|M \cup C^*|+1} \in V(G)$、$X, X'$ 的一个划分$\mathcal{X} = X_{=1} \uplus X_{=2} \uplus X_{=3} \uplus X_{=4} \uplus X_{\geq 5}$（如上文所定义）以及一个函数$g$，使得 P 满足 $M, X, C^*, m_0, m_{|M \cup C^*|+1}, \mathcal{X}, X', g$ 的性质。

在正向中，假设 P是 ESP/svd 中针对 (G, S, k) 的一个解。令 $M \subseteq V(P) \cap S$, $X = S \setminus M$, $C^* = V(P) \cap C$。令 π为 $M \cup C^*$在 P上的顶点排列。令 m_0, $m_{|M \cup C^*|+1}$分别为 P的第一个和最后一个端点。令 $\mathcal{X} = X_{=1} \uplus X_{=2} \uplus X_{=3} \uplus X_{=4} \uplus X_{\geq 5}$ 为根据各顶点到 P的距离对 X 进行的划分。令 $X' \subseteq X$ 为 X 中满足以下条件的所有顶点组成的集合：对于每个 $v \in X'$, $d_G(v, m_j) > i, i \in [4]$ 对所有 $j \in [0, t+1]$均成立。为定义函数g，若顶点 $v \in X'$ 存在一条通往 m_{j+1} 的 m_j 中的 u_i 路径，则设 $g(v) = j$。注意到 P满足 $M, X, C^*, m_0, m_{|M \cup C^*|+1}, \mathcal{X}, X', g$的性质。

在反向方向上，假设存在一个 $M \subseteq S$, $X = S \setminus M$,$C^* \subseteq C$, $m_0, m_{|M \cup C^*|+1} \in V(G)$、一个如上定义的 X, X' 的划分 $\mathcal{X} = X_{=1} \uplus X_{=2} \uplus X_{=3} \uplus X_{=4} \uplus X_{\geq 5}$，以及一个函数 g，使得 P 是满足 $M, X, C^*, m_0, m_{|M \cup C^*|+1}, \mathcal{X}, X', g$ 性质的最短路径。我们断言，P 是 ESP/svd 的 (G, S, k) 的一个解。为导出矛盾，假设 P 不是 ESP/svd 的 (G, S, k) 的一个解，并且存在一条最短路径 P_1，它是 ESP/svd 的 (G, S, k) 的一个解，使得 ℓ 是满足以下条件的最小整数：对于所有 $u \in V(G)$，$d_G(u, P_1) \leq \ell$。令 $v \in V(G)$ 满足 $d_G(v, P_1) \leq \ell$ 且 $d_G(v, P) > \ell$。注意，对于 $\mathcal{X}$ 的一个正确猜测，对每个顶点 $u \in X^* = \cup_{i=1}^{4} X_{=i}$，$u$ 到 P 的距离都是最优的；也就是说，对于 (G, S, k) 的任意解 P^*，$d_G(u, P^*) = d_G(u, P)$。因此，$v \notin X^*$。回想一下，$\ell \geq 5$，否则

该算法将利用引理1完全解决该实例。令 P_1^v为从 v 到 P_1的一条最短路径。注意，由于 $G-S$ 是一个分裂图，每条诱导路径的长度至多为3。由于 P_1^v 是一条长度至少为5（$\ell \geq 6$）的最短路径，根据最短路径的诱导性质，我们有存在一个位于 P_v^1上的顶点 $u \in S$，使得 $d_G(u,P) = j \leq 4$。由于 $\mathcal{X}$是一个正确的猜测，因此对于每个 $u \in X_{=i}, i \in [4]$，都有 $d_G(u,P) = i$。根据 $d_G(u,v) \leq \ell - i$（沿着路径 P_1^v）这一观察，我们得到 $d_G(v,P) \leq \ell$，矛盾。因此， P也是(G,S,k)的一个解。这就完成了对我们算法正确性的证明，从而也完成了定理4的证明。□

7 ESP/cvd 的 W[2]-困难性

本节将证明 ESP/cvd 是 W[2]-困难的。该构造基于 [8]，其中作者证明了 ESP/ecc 是 W[2]-困难的。

定理 5. ESP/cvd 是 $W[2]$-困难。

证明。 为了证明 ESP/cvd 是 W[2]-困难的，我们给出一个从支配集（简称 DS）到 ESP/cvd 的多项式时间参数保持归约，其中 DS 以解的大小为参数。对于 DS，给定一个图 G 和一个整数 k，目标是判断是否存在一个大小至多为 k 的集合 $S \subseteq V(G)$，使得对于每个顶点 $v \in V(G)$， $N[v] \cap S \neq \emptyset$。众所周知，DS 以解的大小 k [6] 为参数时是 W[2]-困难的。

考虑一个 DS 实例 (G, k)。首先，我们描述如何构造一个 (H, S, k') 的 ESP/cvd 实例。令 $V(G) = \{v_1, \ldots, v_n\}$。为构造图 H，我们采用以下过程（构造示意见图6）。

1. 初始化 $V(H) = V(G)$，这意味着，我们为 G 中的每个顶点 v_i，在 $V(H)$ 中添加一个与之对应的顶点 v_i。对于每对 $v_i, v_j \in V(H)$ $i \neq j$，我们在 H 中添加一条v_iv_j 边。令 V^* 为此步骤中添加到 $V(H)$ 的顶点集合。2. 添加 k 个集合 $U_1, U_2, \ldots, U_k$，其中 $U_i = \{u_{i1}, u_{i2}, \ldots, u_{in}\}$ 到 $V(H)$。每个$U_i, i \in [k]$ 都是一个包含 n 个顶点的集合。每个集合 U_i 中的顶点 u_{ij}，其中 $i \in [k], j \in [n]$，表示 G 中第 j 个顶点的第 i 个代表元，并且我们为每个顶点添加 k 个代表元。3. 向 $V(H)$ 中添加一个包含 $(k-1)$ 个顶点的集合 $\{z_1, z_2, \ldots, z_{k-1}\}$。对于每个 $i \in [k-1]$，使 $U_i \cup U_{i+1}$ 中的所有顶点都与 z_i 相邻，也就是说，向 $E(H)$ 中添加边集$\{u_{ij}z_i, u_{(i+1)j}z_i : i \in [k], j \in [n]\}$。4. 向 $V(H)$ 中添加四个顶点 s, a, b 和 t，另外还向 $V(H)$ 中添加两个集合 $A = \{a_1, a_2, \ldots, a_k\}$ 和 $B = \{b_1, b_2, \ldots, b_k\}$，每个集合包含 k 个顶点。向 $V(H)$ 中添加 ~ ~ 个边集 $A = \{a_ia_{i+1} : i \in [k-1]\} \cup \{sa_1, a_ka\}$ 和 $B = \{b_ib_{i+1} : i \in [k-1]\} \cup \{tb_1, b_kb\}$。注意，我们利用包含 A 个顶点和边 A 的集合，添加了一条从 ~ s a 到 的长度为 $k+1$ 的路径。同样，我们利用包含 B 个顶点和边 B 的集合，添加了~一条从 b 到 t 的长度为 $k+1$ 的路径。5. 使顶点 a 与 U_1 中的所有顶点相邻，并使顶点 b与 U_k 中的所有顶点相邻。也就是说，向 $E(H)$ 中添加边集 $\{u_{1j}a, u_{kj}b : j \in [n]\}$。

6.对于每个顶点 $u_{ij} \in U_i$、 $i \in [k]$、 $j \in [n]$ 以及每个顶点 $v_t \in N_G[v_j]$，向 $V(H)$ 添加一个包含 k 个顶点的集合$W_{v_t}^{ij} = \{w_{t1}^{ij}, w_{t2}^{ij}, \ldots, w_t^{ij}{}_{(k)}\}$。将边集 $\tilde{W}_{v_t}^{ij} = \{w_{tp}^{ij} w_t^{ij}{}_{(p+1)} \colon p \in [k-1]\} \cup \{u_{ij} w_{t1}^{ij}, w_t^{ij}{}_{(k)} v_t\}$ 中的边添加到 $E(H)$。注意，对于每个 $i \in [k]$、 $j \in [n]$、 $v_t \in N_G[v_j]$，我们使用顶点集 $W_{v_t}^{ij}$ 和边集 $\tilde{W}_{v_t}^{ij}$ 添加了一条长度为 $k+1$ 的从 u_{ij} 到 v_t的路径。直观地说，如果 $v_j v_t$ 是图 G 中的一条边（或者是 $t = j$，因为我们考虑 v_j的闭邻域），那么我们就在步骤2 中添加到集合 U_i、 $i \in [k]$ 中的、分别对应于 v_j 的每个顶点与步骤 1 中添加的对应于顶点 $v_t \in V^*$ 的顶点之间添加一条长度为 $k+1$ 的路径。

7.（为 i个 v_j 顶点的代表元构造梯子，使顶点与 $N[v_j]$ 中对应的顶点相对应。）接下来，我们取上一步添加的 u_{ij}到 v_t 的路径， $u_{ij} \in U_i$、 $i \in [k]$、 $j \in [n]$ 和 $v_t \in N_G[v_j]$，再向这些路径上的顶点添加更多边，并构造梯子结构。具体来说，我们按如下方式进行。对于每个顶点 $u_{ij} \in U_j$、 $i \in [k]$、 $j \in [n]$ 以及每一对顶点 $v_t v_{t^*} \in N_G[v_j]$ ，我们取上一步添加的 u_{ij} 到 v_t 的路径和 u_{ij} 到 v_{t^*}的路径。上一步中，我们有与 u_{ij}到 v_t的路径对应的 $W_{v_t}^{ij} = \{w_{t1}^{ij}, w_{t2}^{ij}, \ldots, w_{tk}^{ij}\}$顶点集，以及与 u_{ij} 到 v_t^* 的路径对应的 $W_{v_{t^*}}^{ij} = \{w_{t^*1}^{ij}, w_{t^*2}^{ij}, \ldots, w_{t^*k}^{ij}\}$顶点集。我们进一步添加边ˆ $W_{v_t v_{t^*}}^{ij} = \bigcup_{p \in [k]} w_{tp}^{ij} w_{t^*}^{ij}{}_{(p)} \cup \{w_{tq}^{ij} w_{t^*}^{ij}{}_{(q+1)} \colon q \in [k-1]\} \cup \{w_{tk}^{ij} v_{t^*}\}$到 $E(H)$。我们称由顶点集 $W_{v_t}^{ij} \cup W_{v_{t^*}}^{ij} \cup \{u_{ij}, v_t, v_{t^*}\}$及边集 $\tilde{W}_{v_t}^{ij} \cup \tilde{W}_{v_{t^*}}^{ij} \cup \{v_t v_{t^*}\}$ 诱导的图 H为 u_{ij} 与v_t之间的梯子， v_{t^*}。

这完成了对图 H的描述。注意， H 中的顶点数量关于 n 和 k是多项式的，并且该构造可以在多项式时间内完成。令 $X = \{z_1, z_2, \ldots z_{k-1}\} \cup \{a, b\}$。显然，$|X| = k+1$。令$k' = k+1$。这完成了对（$H, k'$）的构造的描述。我们断言， G 具有一个大小至多为 k 的支配集，当且仅当 H 具有一条 P，其离心率至多为（$k+1$）。

正确性。 我们证明 (H, S, k') 是 ESP/cvd 的有效实例。首先，我们证明 X 是 H 的弦图删除集。

断言。 $H - X$ 是一个弦图。

证明。设 C 是 $H \setminus X$ 中长度至少为四的一个环。由于 $\{v_1, v_2, \ldots v_n\}$ 在 G 中构成一个团，因此 C 至多包含 V^* 中的两个顶点。现在，在 $H \setminus (X \cup V^*)$ 的每个连通分量中，来自 $U = \cup_{i=1}^{k} U_i$ 的顶点至多有一个。由于 V^* 诱导出一个团，因此不存在包含 U 中多个顶点的诱导环。设 $u_i^j \in U_j$ 是 C 中任意一个顶点。根据我们的构造， u_{ij} 的邻居是 w_{t1}^{ij} 类型的顶点，即从 u_{ij} 到 $v_t \in N_G[v_j]$ 路径上的第一个顶点（步骤 5），或是 $w_{t1}^{i'j}$ 类型的顶点，即从 $u_{i'j}$ 到 $v_t \in N_G[v_j]$ 路径上的第一个顶点（步骤 7）。注意，根据我们在步骤 6 和 7 中对梯子的构造， u_i^j 在 H 中的邻域内任意一对顶点都相邻（这些边通过 ˆY 类型的集合添加到 $E(H)$ 中）。现在，设 $w_{tt'}^{ij}$ 是 u_i^j 在 C 中的一个最近邻居（假设距离为 t'）。如果我们观察

$w_{tt'}^{ij}$ 在 C 中的邻居，由于 C 是一个长度至少为 4 的环，因此唯一的选择是 $w_{t^*t'}^{ij}$ 和 $w_{t\ (t'+1)}^{ij}$。现在，$w_{tt'}^{ij}$ 是 $w_{t^*t'}^{ij}$ 在 C 中的一个邻居，设 w 是 $w_{t^*t'}^{ij}$ 的另一个邻居。显然，$d_H(u_i^j, w) = t'$ 或 $t'+1$。无论是哪种情形，根据我们的构造，$w_{tt'}^{ij}$ 与 w 之间都有一条边。这就导致矛盾。因此，$H \setminus X$ 中不存在长度至少为四的诱导环。因此，我们得出 H 是一个弦图。

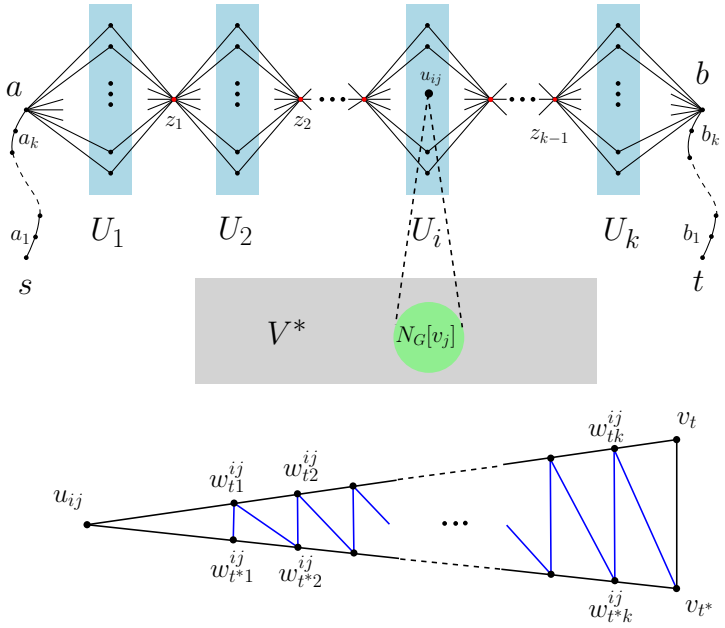

图 6. 从支配集到 ESP/cvd. 定理 5 的证明示意图。

断言。 G 当且仅当 H 存在一条离心率至多为 $(k+1)$ 的最短路径 P 时，拥有一个大小至多为 k 的支配集。

证明。在正向中，假设图 G 有一个大小为 k的支配集 D。令 $D = \{v_{t_1}, v_{t_2}, \ldots, v_{t_k}\}$ 和 $D^* = \{a, u_{1t_1}, z_1, u_{2t_2}, z_2, \ldots, z_{k-1}, u_{kt_k}, b\}$，其中 u_{it_i} 是集合 U_i 中顶点 t_i 的第 i个代表元，该集合是在我们对 H的构造的步骤 2中添加的。注意，根据我们的构造，$P = H[D^*]$是一条诱导路径。由于 D 是 G中的一个支配集，对于每个顶点 $v_{t_i} \in D$，我们有每个顶点 $v_t \in N_G[v_i]$ 在 H中距离u_{it_i} 为 $(k+1)$。根据构造，对于 H中的其余顶点，从 P存在一条长度至多为 $(k+1)$的最短路径。

对于反向，注意每条不包含顶点 a 或 b 的最短路径，其离心率都大于 $k+1$，因为到 s 的距离

到 a（分别是从 t 到 b）的距离恰好为 $k+1$。此外，由于 $d_H(a, b) = 2k$，且任何经过 V^* 的 a 与 b 之间的路径都超过值 $2k$，因此每条最短路径都必须与集合 U_i 中的所有元素相交，并经过所有满足 $i \in [k]$ 和 $i' \in [k-1]$ 的顶点 $z_{i'}$。假设存在一条离心率为 $k+1$ 的最短路径 P，且该路径位于 H 中。这意味着，对于每个顶点 $v \in H$，都存在一个顶点 $u \in P$，使得 $d_H(u, v) \leq (k+1)$。令 v 为 V^* 中任意一个顶点，并设存在一个顶点 $u_{ij} \in P$，使得 $d(v, u_{ij}) = k+1$。这意味着 $v \in N_G[v_j]$。由于 v 是任意选取的，因此集合 $D = \{v_j : u_{ij} \in V(P) \cap U_i, i \in [k]\}$ 是 G 的一个支配集，且大小至多为 k。证明至此完成。

定理 5的证明至此完成。 □

参考文献

1. Etienne Birmelé、Fabien de Montgolfier 和 Léo Planche。最小离心率最短路径问题：一种近似算法及其与 k-层状性问题的关系。载于 组合优化与应用国际会议，第 216–229 页。施普林格，2016 年。2. Yixin Cao 和 Dániel Marx。弦图编辑是固定参数可解的。*Algo- rithmica*，75(1)：118–137，2016 年。3. Marek Cygan、Fedor V Fomin、Łukasz 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。4. Marek Cygan 和 Marcin Pilipczuk。分裂顶点删除遇上顶点覆盖：新的固定参数与精确指数时间算法。*Information Processing Letters*，113(5-6)：179–182，2013 年。5. Reinhard Diestel。图论。数学研究生教材，第 173 卷，2017 年。6. Rodney G Downey 和 Michael R Fellows。参数化复杂性的基础，第 4 卷。施普林格，2013 年。7. Feodor F Dragan 和 Arne Leitert。某些结构化图类中的最小离心率最短路径。载于 计算机科学图论概念国际研讨会，第 189–202 页。施普林格，2015 年。8. Feodor F Dragan 和 Arne Leitert。关于最小离心率最短路径问题。*Theoretical Computer Science*，694：66–78，2017 年。9. Ralph J Faudree、Ronald J Gould、Michael S Jacobson 和 Douglas B West。最小度与支配路径。*Journal of Graph Theory*，84 213，2017 2 202( )：–。10. Martin Kučera 和 Ondřej Such`y。关于结构参数的最小离心率最短路径问题。载于 组合算法国际研讨会，第 442–455 页。施普林格，2021 年。11. Finn Völkel、Eric Bapteste、Michel Habib、Philippe Lopez 和 Chloe Vigliotti。Read 网络与 k-层状图。*arXiv* 预印本 *arXiv:1603.01179*，2016 年。