arXiv:2301.02421v1 [cs.DM] 6 Jan 2023

平面有向图中绕行路径的更简单、更快速算法 *

Meike Hatzel† Konrad Majewski‡ Micha-l Pilipczuk§ Marek Soko-lowski¶

摘要

在 Directed Detour 问题中，给定一个有向图 G 和一对顶点s 和 t，任务是判断在 G 中是否存在一条从 s 到 t的有向简单路径，其长度大于 $\mathrm{dist}_G(s, t)$。更一般的参数化变体Directed Long Detour要求寻找一条从 s到 t 的简单路径，其长度至少为 $\mathrm{dist}_G(s, t) + k$，其中给定参数 k。令人惊讶的是，目前仍不清楚 Directed Detour是否能在一般有向图上于多项式时间内求解。然而，对于平面有向图，Wu 和 Wang [在 Networks（'15] ）中提出了一种用于 Directed Detour的 $\mathcal{O}(n^3)$时间算法，而 Fomin 等人 [在 STACS 2022] 中为 Directed Long De-tour给出了一种 $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$时间的 FPT 算法。Wu 和 Wang 的算法依赖于对平面嵌入中短绕行路径可能形态的复杂分析，而 Fomin 等人的算法则基于将问题归约为平面有向图上的 3-Disjoint Paths问题。后一个问题可利用 Schrijver 在 SIAM J. Comp.（'94], ）中提出的代数工具在多项式时间内求解，但所得多项式因子的次数非常高。

本文提出了两种简单的算法：我们展示了如何在平面有向图中，以 $\mathcal{O}(n^2)$ 的时间求解 Directed Detour ，以及以 $2^{\mathcal{O}(k)} \cdot n^4 \log n$ 的时间求解 Directed Long Detour 。两种情况下，我们的思路都是将问题归约为平面有向图中的问题 2-Disjoint Paths，并观察到该问题所得到的实例具有某种拓扑结构，使其适合采用直接的贪心策略。

*本研究是 BOBR 项目（KM、MP、MS）的一部分。该项目获得了欧洲研究委员会（ERC）通过欧盟地平线2020研究与创新计划提供的资助（资助协议编号：948057）。M. Hatzel 获得了联邦教育与研究部（BMBF）的资助，并获得了德国学术交流中心（DAAD）IFI 计划的奖学金。

†日本东京国立情报学研究所（meikehatzel@nii.ac.jp）
‡波兰华沙大学信息学研究所（k.majewski@mimuw.edu.pl）§波兰华沙大学信息学研究所（michal.pilipczuk@mimuw.edu.pl）
¶波兰华沙大学信息学研究所（marek.sokolowski@mimuw.edu.pl）

1 引言

Directed Detour的复杂度状况，或许是图算法领域最令人着迷的开放问题之一。该问题要求判断：给定一个有向图G 和两个终端 s 和 t，是否存在一条从 G s 到 t 的简单路径，其长度严格大于最短路径的长度——也就是说，该路径不是最短路径：$\mathsf{dist}_G(s,t)$。目前仍不清楚该问题是否能在一般有向图上以多项式时间求解。关于相关文献的讨论，参见 Fomin 等人 [3] 的研究。

鉴于目前的情况，研究受限的有向图类上的 Directed Detour 很有意义，或许能找到更多积极成果或有用的见解。基于这一思路，一个特别有充分理由的方向是研究平面有向图类。原因在于，对于平面有向图， k-Disjoint Paths 问题——判定是否存在连接给定 k 对终端的互不相交有向路径——对于每个固定的 k [8]都可以在多项式时间内求解；甚至当以 k [2]为参数时，该问题也是固定参数可解的。这在一般有向图中并非如此，因为该问题对于 $k = 2$ [5]已经是 NP-困难的。这一点可用于 Directed Detour。具体而言，Fomin 等人 [3] 证明，问题的更一般参数化版本—— Directed Long Detour，即寻找一条从 s到t 的简单路径，其长度至少为 $\mathsf{dist}_G(s,\ t) + k$，其中给定参数 k——可以通过（在复杂度上增加 $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ 乘法开销）归约到 3-Disjoint Paths；利用 Schrijver 的算法 [8]，该问题可以在平面有向图上以多项式时间求解。当然，将 $k = 1$设为相应值也适用于基本的 Directed Detour 问题，但更早之前，Wu 和 Wang [9] 已针对这一情形给出了一个运行时间为 $\mathcal{O}$（n^3）的直接算法。

虽然 Fomin 等人的归约 [3] 实际上相当简单，但 Schrijver 的算法 [8] 用于 3-Disjoint Paths平面有向图时却并非如此，因为它依赖于一个复杂的代数框架。具体而言，运行时间上界多项式的次数至少是两位数。另一方面，Wu 和 Wang 的立方时间算法 [9] 也相当复杂，并且依赖于对可能出现的不同平面构型的分析。

在本文中，我们提出两种简单的算法：一种用于 Directed Detour ，另一种用于 Directed Long Detour，二者均适用于平面有向图。下面对它们进行概述。

Th定理 1.1。 Directed Detour 在平面有向图中可在时间 $\mathcal{O}$（n ^{2}）.

定理 1.2。 Directed Long Detour 在平面有向图中可通过蒙特卡洛算法在时间 $2^{\mathcal{O}(k)} \cdot n^4$ 内求解，或确定性地在时间 $2^{\mathcal{O}(k)} \cdot n^4$ 对数 n内求解。

定理1.1 证明中的主要思路是将问题归约为 2-DisjointPaths 问题，大致沿用 Fomin 等人 [3] 的工作中的做法。然而，我们注意到，如果谨慎地进行这种归约，实质上就能得到一个 2-DisjointPaths 实例，其中四个终端中有三个位于同一个面上，例如外面。这类实例很容易求解：其中一条路径——即两个终端都位于外面上的那条路径——可以通过贪心方式选取，从而为另一条路径留出尽可能大的空间。随后，只需进行一次简单的可达性检查，就能找到另一条路径。

对于定理 1.2，与 Fomin 等人 [3], 一样，我们使用 Bez´akov´a 等人 [1] 的结果：即使在一般有向图上，也可以在时间 $2^{\mathcal{O}(k)} \cdot n^2$ 内解决Exact Directed Long Detour 问题——寻找一条长度恰为 $\mathsf{dist}_G(s,t) + k$ 的 s-到-t 路径。这样，在求解 Directed Long Detour 时，我们可以假设不存在长度介于 $\mathsf{dist}_G(s,t) + k$和 $\mathsf{dist}_G(s,t) + 3k$之间的 s-到-t 路径。在这一假设下，定理 1.2 的证明通过结合颜色编码，拓展定理 1.1 背后的基本思路。

我们认为，与 [3, 9], 相比，我们的 Directed (Long) Detour 算法能更简单地解释为何平面有向图上的问题具有可处理性。虽然我们不认为由此获得的见解能直接应用于一般有向图，但我们希望它能为推广到限制较少的拓扑图类提供更好的起点，例如有界亏格有向图，或其底层无向图不含固定子式的有向图。

2 预备知识

图。 本文研究平面有向图 $G = (V(G), E(G))$。对于我们的问题，可以假设输入图是简单的，也就是说，不存在自环，也不存在连接同一方向上两个顶点的重弧。此外，我们假设 G 是弱连通的，也就是说，其底层无向图是连通的。为方便起见，我们设 $n = |V(G)|$ 和 $m = |E(G)|$。由于 G 是平面且简单的，因此有 $m \in \mathcal{O}(n)$。

给定一条弧 $e = uv \in E(G)$，我们称 u 为 e 的尾，称 v 为 e的头。这里，我们认为 e 与 u和 v均关联。顶点序列 $v_1 v_2 \dots v_k$ 称为G 中的一条游走，如果对每个 $i \in \{1, \dots, k-1\}$，$v_i v_{i+1}$ 都是 G中的一条弧。顶点 v_1 称为该游走的起点，而 v_k 是其终点。如果游走中的顶点两两不同，则该游走是一条路径。给定两条游走 W_1、W_2，若 W_1 的终点与 W_2的起点重合，则定义 $W_1 \circ W_2$ 为两条游走的拼接。路径 P的长度 记为 $\mathsf{length}(P)$，即其包含的弧数。给定一条路径 P 及其两个顶点 x 和y，我们用 $P[x \to y]$ 表示 P 的一条子路径，该子路径从 x出发，沿着 P的弧前进，并在 y结束。

给定一对顶点 u，$v \in V(G)$，u 到 v 在 G 中的距离 是 u到 v的最短路径在 G 中的长度，记作 $\mathsf{dist}_G(u, v)$。如果在 G中不存在从 u到 v 的有向路径，则令 $\mathsf{dist}_G(u, v) = +\infty$。如果上下文中已知图 G，则可以在记号中省略G，直接写作 $\mathsf{dist}(u, v)$。

平面嵌入。 给定一个平面图 G，将其嵌入平面是指将 G 的顶点映射到平面上两两不同的点，并将 G 的弧映射到平面曲线，满足：

- 每条弧 $uv \in E(G)$ 都映射为一条平面曲线，其端点与 u 和 v的像重合，并且 G 的其他顶点均不映射到该曲线上的点；并且
- G 的边的像两两内部不相交。

平面嵌入图是一个平面图，连同该图在平面中的一个固定嵌入。这样的嵌入将平面分割成若干区域，称为面。其中一个面是无界的，称为外面。给定一个面 F，其边界 ∂F 可以描述为围成 F 的弧的一个循环序列（假设 G 是连通的）。如果 ∂F 同构于一个简单多边形（忽略弧的方向），则称 F 是简单的。

出于算法设计的需要，我们以组合方式表示平面嵌入：图的每个顶点 v都存储了与其 v关联的所有弧 G 的逆时针顺序。给定一个平面图 G，其组合嵌入可以在线性时间内计算，时间相对于图的

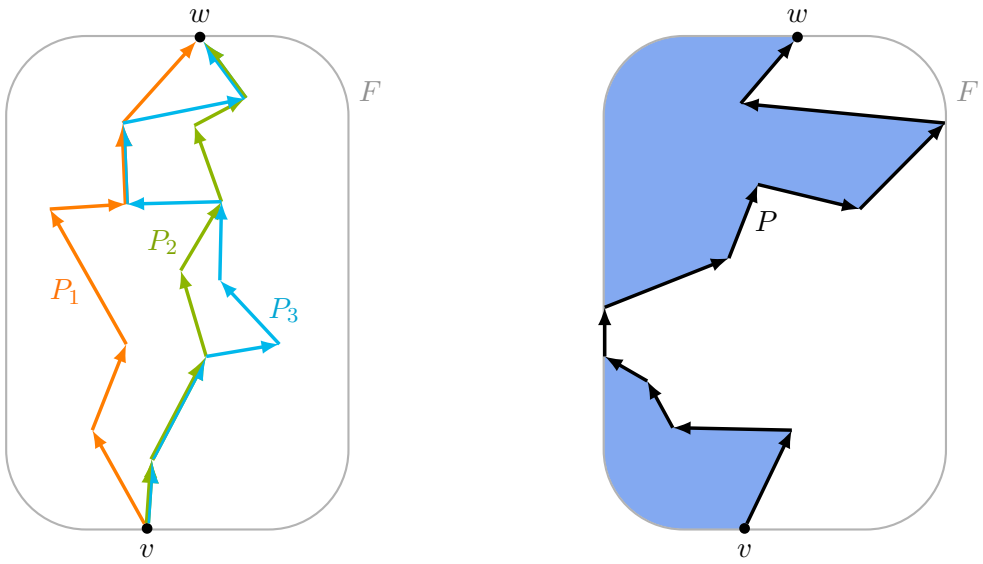

图1：(a) 三条（v，w）锚定路径。我们有 $P_1 \prec_{\text{lex}} P_2 \prec_{\text{lex}} P_3$，$P_1 \prec_{\text{top}} P_2$，$P_1 \prec_{\text{top}} P_3$，但 $P_2 \not\prec_{\text{top}} P_3$。(b) 一条（$v$，$w$）锚定路径 P及其左侧区域 $\Gamma_{\text{L}}(P)$（填充为 蓝色）。

大小 [6]而言。请注意，组合嵌入唯一确定了平面图的各个面，但并未指定嵌入的外面。因此，在我们的算法中，可以任选一个面作为外面。

平面图中的路径比较。 设 G 为一个平面嵌入图，F 为其外面。假设 F 是简单的。若路径 P 位于 G 中，且其起点 v是 F的一个顶点，则称该路径是相对于 F的锚定路径 。此外，若路径 P的起点 v 和终点 w 都是 F的顶点，则称其是相对于 F（v，w）的锚定路径 。现在我们介绍两种比较 G中（双重）锚定路径的方法。以下定义都是标准定义。

定义2.1。 假设在平面嵌入图 G 中，P_1 和 P_2 是关于面 F 的两个不同的 v-锚定路径。若满足以下条件之一，我们称 P_1 按字典序位于 P_2 左侧（记作 $P_1 \prec_{\text{lex}} P_2$）：

- P_1 是 P_2 的前缀；或
- 考虑平面嵌入图 G_v，它由 G 及其平面嵌入构成，并添加一个新顶点 v' 和一条新弧 $v'v$；将其映射到平面上，使得 v' 的像位于 F 外部（即 v' 不被 F 的边界围住）。设 P_1'和 P_2' 是 G_v 中的路径，分别通过将 v' 前置于 P_1 和 P_2 构造而成。设 q 是 P_1'和 P_2' 的最长公共前缀的最后一个顶点，p 是公共前缀上位于 q 之前的顶点，$r_1 \neq r_2$ 是分别在 P_1' 和 P_2'上紧接 q 之后的顶点。于是，弧 pq、qr_1 和 qr_2 围绕 q 按顺时针方向依次嵌入。

直观地说，$P_1 \prec_{\text{lex}} P_2$ 如果在 P_1 与 P_2的公共前缀末尾，路径 P_1在 P_2 的左侧终止或分叉（图 1(a)）。注意，$\prec_{\text{lex}}$ 在以顶点 $v \in V$（F）为起点的所有路径上诱导出一种线性序。因此，对于任何可从 $w \in V$（G）到达的顶点

从 v 在 G 中，我们将从 v 到 w 的字典序最左路径定义为 L_{vw}：它是从 v 到 w 的、关于 $\prec_{\text{lex}}$ 的唯一极小路径。给定 G 的组合嵌入，我们可以在 $\mathcal{O}(n)$ 时间内计算从 v 到 w 的字典序最左路径：只需对 G 运行深度优先搜索，并让每个顶点按照从左到右的顺序考察其所有邻居。

接下来，我们介绍一种在 G中比较路径的限制更严格的方法。给定一条（v, w）锚定路径 P，我们将位于 P左侧的区域 $\Gamma_{\mathsf{L}}(P$定义为平面上的一个区域，其逆时针边界是由 P与 F 边界上从 w 到 v 的逆时针线段拼接而成的闭游走（图1(b)）。注意，如果 $\Gamma_{\mathsf{L}}(P$在内部与 P 的边界相交，那么 F的内部可能是不连通的。

定义 2.2。给定两条(v,w)锚定路径 P_1 和 P_2，如果 P_1 在拓扑上位于 P_2 左侧（记作 $P_1 \prec_{\text{top}} P_2$），则称 $\Gamma_{\mathsf{L}}(P_1) \subsetneq \Gamma_{\mathsf{L}}(P_2)$。

注意，如果 $P_1 \prec_{\text{top}} P_2$，那么 $P_1 \prec_{\text{lex}} P_2$。因此，$\prec_{\text{top}}$ 是所有（v, w）锚定路径集合上的偏序，而将 $\prec_{\text{lex}}$ 限制在这些路径上，便是 $\prec_{\text{top}}$的一个线性扩张。不难看出，从 v 到 w 的字典序最左路径——即 $\prec_{\text{lex}}$ 中的最小（v, w）锚定路径——也是 $\prec_{\text{top}}$中唯一的最小（v, w）锚定路径。因此，如果 v 和 w 都位于 F上，我们就将 L_{vw} 简称为从 v 到 w的最左路径。

类似地，我们定义从 v 到 w 的（字典序）最右路径，记为 R_{vw}，以及位于 $(v$, $w)$-锚定路径 P右侧的区域，记为 $\Gamma_{\mathsf{R}}(P)$。注意，每条 $(v$, $w)$-锚定路径 P都会将图分割为两部分：P左侧的区域 $\Gamma_{\mathsf{L}}(P)$ 和 P右侧的区域 $\Gamma_{\mathsf{R}}(P)$，这两个区域仅在 P处相交。

简化外面。对锚定路径进行字典序比较和拓扑比较，都要求平面嵌入图 G 的外面 F 是简单的。然而，在我们的问题场景中，这一前提可能并不成立。为了解决这一问题，我们将展示如何通过向平面嵌入图添加两条有向弧来简化外面。

设 u和 v是 F的两个顶点。G中外面的 (u,v)-简化记为 Simplify(G,u,v)，它是通过向 G 添加两条弧 f_1、f_2得到的，每条弧都从 u连接到 v。两条弧都嵌入在 G的外面 F中，使得 Simplify(G,u,v)的外面由 f_1和 f_2围成（见图 2）。因此，Simplify(G,u,v)的外面是简单的，且包含两个顶点 u和 v。需要指出的是，Simplify(G,u,v)不一定是简单有向图；不过，这并不会造成问题，因为简化后，图的边数仍然与图的顶点数呈线性关系。

平面图的弧切割。最后，我们描述一种对平面图进行操作的方法，该操作会在嵌入的外面引入新顶点 F。假设 F 是简单的。取两个顶点 $u \in V(F)$，$v \notin V(F)$，二者都与一条弧 $e \in E$ $(G$关联。沿着切割图 e 的操作会生成一个新图 G_e，其生成过程如下（见图 3）：

1. 按逆时针顺序枚举所有与 u关联的弧：e_1，e_2，……，e_k，其中 e_1，e_k $\in E(F)$。
2. 令 $\ell \in \{2,3,\ldots,k-1\}$ 满足 $e_\ell = e$。
3. 从图中移除 u，以及所有与 u关联的弧。

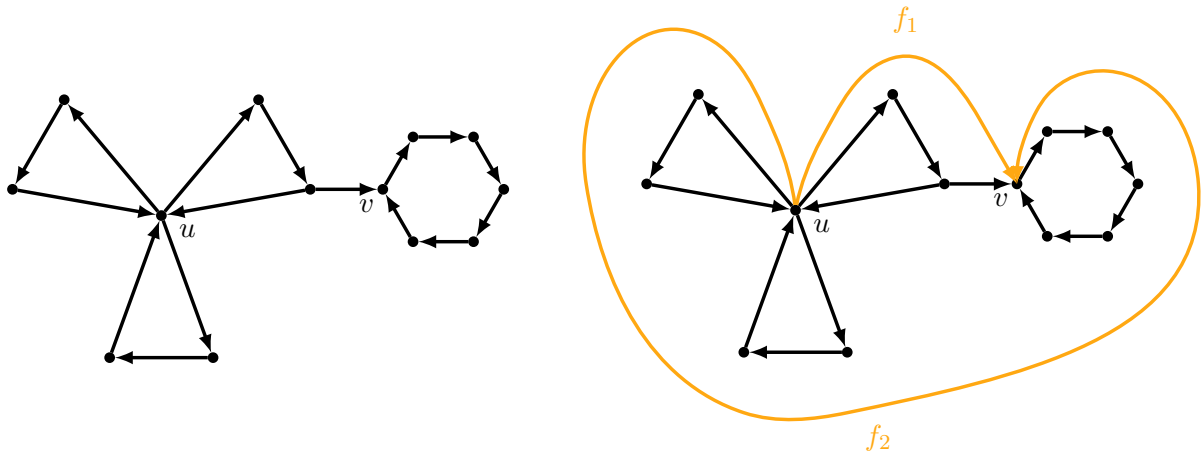

图 2：一个平面有向图示例（左）及其 (u, v)-简化（右）。

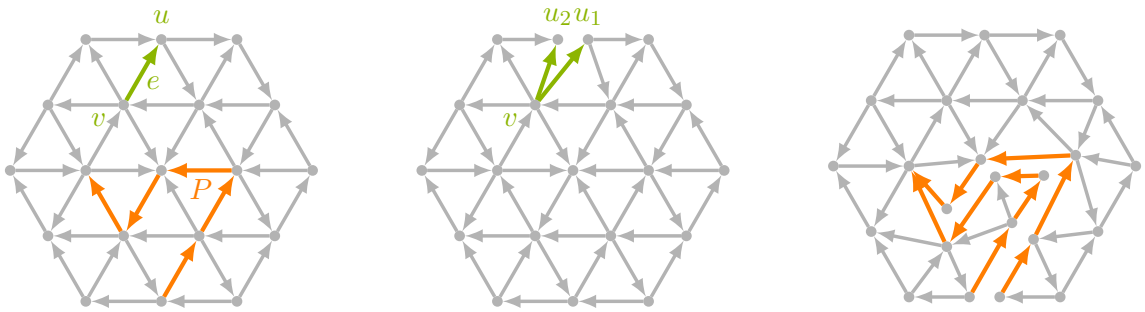

图 3：(a) 一个有向平面图 G。(b) 沿 e 切割 G 后的结果。(c) 沿 P 切割 G 后的结果。

4. 在图中引入两个新顶点 u_1、u_2。对于每个 $i \in \{1, 2, \ldots, \ell\}$，向图中添加一条新弧 e_i^1，它是将弧 e_i 的端点 u替换为 u_1后得到的。同样，对于每个 $i \in \{\ell, \ell+1, \ldots, k\}$，向图中添加一条新弧 e_i^2，它是将弧 e_i 的端点 u 替换为 u_2后得到的。

直观地说，G_e 可以通过将图 G 画在一张纸上，再沿着 e剪开这张纸得到。注意，经过这一操作后，所得的图仍然是平面图（图 3）。此外，F_e 的外面 G_e 是简单的，并且 $V(F) \setminus V(F_e) = \{u\}$ 和$V(F_e) \setminus V(F) = \{u_1, v, u_2\}$。也就是说，$v$ 现在位于新图的外面上。

我们可以将这一过程推广到路径，而不仅仅是边。假设 $P = v_1v_2 \ldots v_k$ 是一条 v_1-接地路径，且除 v_1外，与 $V(F)$ 不相交。那么，沿 P切割图，就意味着依次沿弧 v_1v_2、v_2v_3、……、$v_{k-1}v_k$切割图。由此，我们得到一个平面嵌入图，其中该路径的终点位于外面上。图 3展示了一个例子。

显然，给定一个平面嵌入图 G的组合嵌入，就能在线性于图大小的时间内，计算出沿路径 P 切割 G 后所得图的平面嵌入。因此，我们再次得到一个平面嵌入图。

3 有向绕行

本节证明定理 1.1。

设 G 是一个平面有向图，并设 $s,t \in V(G)$ 是一对顶点。回顾一下，我们假设 G 是弱连通的，并固定 G 的平面嵌入。因此，为了简便起见，从现在起，我们将 G 中的特征（顶点、边、路径等）与其在嵌入下的像等同起来。我们的目标是判断 G 中是否存在一条从 s 到 t 、长度至少为 $\mathrm{dist}_G(s,t) + 1$ 的路径。我们将把这个问题归约为求解一组 2-Disjoint Paths 问题。该归约与 Fomin 等人 [3] 针对Directed Long Detour提出的算法类似。

首先，我们从 s 开始，在 G 中运行广度优先搜索（BFS）。令 L_i 表示该广度优先搜索的第 i 层，也就是说，对于 $i \in \{0,1,\ldots,n\}$， $L_i = \{v \in V(G) \mid \mathrm{dist}_G(s,v) = i\}$。

现在假设实例 (G, s, t) 是一个是实例，并令 P 为一条 s-到-t 路径，以此证明这一点。由于 P 是一条从 s 到 t 的非最短路径，因此存在一个索引 $p \in \{0,1,\ldots,n\}$，使得层 L_p 至少包含 P 的两个顶点。我们选择 p 为满足条件的最小索引。令 $G_{\geqslant p}$ 为从 G 中移除 $L_0 \cup L_1 \cup \ldots \cup L_{p-1}$ 的所有顶点后得到的平面有向图。我们可以假设 $G_{\geqslant p}$ 是弱连通的，否则就舍弃其中所有不包含顶点 t 的弱连通分量。

断言 1. L_p 的所有顶点都位于 $G_{\geqslant p}$ 的同一个面上。

证明. 考虑 L_p 中任意顶点 v，以及 G 中从 s 到 v 的任意最短路径 Q。注意，除了 v 之外， Q 的所有顶点都位于层 L_0， L_1、 ..... 、 L_{p-1} 中，因此在从 G 构造 $G_{\geqslant p}$ 时会被移除。由此可知， v 位于包含 s 的 $G_{\geqslant p}$ 的（唯一）面的边界上。 □

分别用 x 和 y 表示路径 P 上位于第 L_p 层的第一个和第二个顶点。在我们的算法中，我们遍历顶点 y 的所有可能取值。注意， y 的取值决定了 p 的值，因为 $y \in L_p$。观察可知，如果我们正确猜出顶点 y，那么要找到一条从 s 到 t 的非最短路径，只需找到一个顶点 $x \in L_p$ （$x \neq y$）和三条路径 P_{start}、 P_{middle} 和 P_{end}，使得

- $P_{\text{起始}}$ 是图 G 中从 s 到 x 的一条最短路径；并且
- $P_{\text{中间}}$， $P_{\text{终点}}$ 是图 $G_{\geqslant p}$ 中两条内部顶点不相交的路径，分别从 x 到 y，以及从 y 到 t。

注意， y 可能等于 t，此时 P_{end} 是一条仅由单个顶点 t 构成的平凡路径。

一方面，顶点 x 和 y 将 P 分成子路径 P_{start}、 P_{middle} 和 P_{end}，它们满足上述性质。另一方面，如果我们找到一个顶点 x 和路径 P_{start}、 P_{middle}、 P_{end}，使其满足上述条件，那么它们的拼接便构成一个有效解。这是因为路径 P_{start} 由于是最短路径，会依次经过第 L_0、 L_1、 .....、 L_p 层，因此它与 P_{middle} 和 P_{end} 内部顶点不相交。此外， P_{start}、 P_{middle}、 P_{end} 的拼接是一条从 s 到 t 的非最短路径，因为它至少包含两个来自第 L_p 层的不同顶点。

自然的做法是也遍历 x 的所有可能选择，但为了优化运行时间，我们改为执行以下操作：从有向图 $G_{\geqslant p}$ 构造有向图 H，方法是向其中添加一个顶点 x_{super}，并为所有 $u \in L_p \setminus \{y\}$ 添加弧 $x_{\text{super}}u$。

断言 2. H 是一个平面有向图，其隐式嵌入可在线性时间内由 $G_{\geqslant p}$ 的隐式嵌入得到。

证明。只需扩展 $G_{\geqslant p}$ 的隐式嵌入：将 x_{super} 嵌入到唯一一个其边界包含 L_p 的所有顶点的面中（根据断言 1，这样的面存在），并在该面内绘制连接 x_{super} 与 $L_p \setminus \{y\}$ 的顶点的弧。显然，这可以在线性时间内以隐式嵌入的方式完成。

□

注意，从 x_{super}到 y （位于 H 中）以及从 y 到 t （位于 H 中）的内部点不相交路径对(P_1, P_2)，与从某个顶点 $x \in L_p$ （$x \neq y$）到 y （位于 $G_{\geqslant p}$ 中）以及从 y 到 t （位于 $G_{\geqslant p}$中）的内部点不相交路径对$(P_{\text{middle}}, P_{\text{end}})$一一对应。我们还可以假设， H 中同时包含 x_{super}和 y 的面是 H的外面。这样，我们就将原问题化简为至多 n个实例的集合（对顶点 y 的每种选择各对应一个实例），所涉及的问题如下：

PROBLEM A
输入： 一个平面有向图 H和三个顶点 $x, y, t \in V(H)$，其中 x和 y位于 H的外面。

问题： 是否存在两条内部顶点不相交的路径 P_1 和 P_2，使得 P_1 是 x到 y 的路径，且 P_2 是 y到 t 的路径，位于 H中?

因此，要证明定理 1.1 只需证明下面的引理。

引理 3.1。 PROBLEM A 可以在时间 $\mathcal{O}(n)$内求解。

证明。令 $H^\star$ 为 H外面的一个 (y, x)-简化。显然， $H^\star$的外面是一个简单环。此外，$(H^\star, x, y, t)$ 是是实例，当且仅当 (H, x, y, t) 是是实例：$H^\star$ 恰好是带有两条额外弧的图 H，显然这两条弧不可能属于所求的解。

现在假设 $(H^\star, x, y, t)$ 是是实例，并令 (P_1, P_2) 为 $H^\star$中的一对路径，它们构成了 $H^\star$上 PROBLEM A的一个解。 P_1 连接了位于 $H^\star$ 外面F_O 上的 x 和 y （ $H^\star$ 是一个简单环），因此相对于 F_O而言，它是 (x, y)-定基的。因此，它将整个有向图 H 分成两部分：位于 P_1 左侧的区域 $\Gamma_{\mathsf{L}}(P_1)$ 和位于 P_1右侧的区域 $\Gamma_{\mathsf{R}}(P_1)$，两个区域仅在 P_1处相交。由于 P_2 与 P_1内部不相交，它必定完全包含在 $\Gamma_{\mathsf{L}}(P_1)$ 或 $\Gamma_{\mathsf{R}}(P_1)$ 中；不失一般性，假设P_2 包含在后者中。接着，令 $L_{x,y}$ 为 $H^\star$中从 x 到 y 的最左路径。

命题 3. $(L_{x,y}, P_2)$ 也是 PROBLEM A 的有效解。

证明。根据最左路径的定义，我们知道 $\Gamma_{\mathsf{L}}(L_{x,y}) \subseteq \Gamma_{\mathsf{L}}(P_1)$，也就是说，$L_{x,y}$ 不能使用位于 P_1 右侧的任何顶点。由于 P_2 的所有顶点（除 y 外）都位于 P_1 右侧，因此我们得出该命题。

□

类似的论证表明，如果 P_2 包含在 $\Gamma_{\mathsf{L}}(P_1)$ 中，那么 $(R_{x,\,y},\ P_2)$ 是 PROBLEM A 的一个有效解，其中 $R_{x,\,y}$ 是 $H^\star$ 中从 x 到 y 的最右路径。根据上述观察，我们知道，要验证 $(H^\star,\ x,\ y,\ t)$ 是否为是实例，只需找到 $L_{x,\,y}$ 和 $R_{x,\,y}$ 作为 P_1 的候选，并检查对每个候选而言，顶点 t 是否能从 y 到达 $H^\star - (V(P_1) \setminus \{y\})$ 中。通过运行适当的深度优先搜索算法，所有这些检查都能在时间$\mathcal{O}(n)$内完成。

□

4 有向长绕行

本节证明定理 1.2。

给定一个平面有向图 G、两个顶点 s、$t \in V(G)$以及一个整数 $k \in \mathbb{N}$。与第3节一样，我们假设 G 是弱连通的，并固定 G的平面嵌入。我们需要判断在 s到t 之间是否存在一条 G 长度至少为$\mathsf{dist}_G(s,t) + k$的路径。我们将把这个问题归约为求解一组 2-Disjoint Paths 问题的实例（附加一些要求）。

首先，我们运行 Bez´akov´a 等人 [1] 的算法，检查是否存在一条从 s 到 t 的路径，其长度为 $\mathsf{dist}_G(s,t)+l$，其中某个 $l \in \{k, k+1, \ldots, 3k-1\}$ 满足条件。从现在起，我们假设不存在这样的从 s 到 t 的路径。

与第 3 节一样，我们从 s 出发，在 G 中运行广度优先搜索，并令 $L_i = \{v \in V(G) \mid \mathsf{dist}_G(s,v) = i\}$，其中 $i = 0$ 取 1，...，n。对于 $i = 1$ 取 2，...，n，令 $G_{\geqslant i}$ 为从 G 中删除 $L_0 \cup \ldots \cup L_{i-1}$ 中所有顶点后得到的平面有向图。同样，不失一般性，我们假设 $G_{\geqslant i}$ 弱连通。

现在，我们遍历 $p = 1$、2、.....、n 的所有取值，并遍历 x、$y \in L_p$ 的所有选取，其中 $x \neq y$（$(p$、x、$y)$ 最多有 n^2 种选取）。回想第 3 节（第 1 节），x 和 y 都位于 $G_{\geqslant p}$ 的外面上。接下来，令 $G^{x,y}_{\geqslant p}$ 为 $G_{\geqslant p}$ 外面的 (y, x)-简化。现在，我们将搜索满足以下条件的三条路径 P_{start}、P_{middle} 和 P_{end}：

(a) P_{start} 是 G 中从 s 到 x 的最短路径；

(b) P_{middle} 是 $G^{x,y}_{\geqslant p}$ 中长度至少为 $2k$ 的 (x, y)-接地路径；

(c) P_{end} 是 $G^{x,y}_{\geqslant p}$ 中从 y 到 t 的一条路径；

(d) 路径 P_{middle} 和 P_{end} 内部顶点不相交；并且

(e) 若 $P_{\text{end}} \subseteq \Gamma_{\mathsf{R}}(P_{\text{middle}})$，则 $G^{x,y}_{\geqslant p}$ 中不存在长度至少为 k 的 (x, y)-接地路径 P'_{middle}，使得 $P'_{\text{middle}} \prec_{\text{top}} P_{\text{middle}}$；若 $P_{\text{end}} \subseteq \Gamma_{\mathsf{L}}(P_{\text{middle}})$，则不存在满足 $P_{\text{middle}} \prec_{\text{top}} P'_{\text{middle}}$ 的此类路径。

我们指出，在对 $G_{\geqslant p}$ 进行 (y, x)-简化的过程中添加的两条 yx 弧，不可能属于路径 P_{start}、P_{middle}、P_{end} 中的任何一条，因此在接下来的一系列断言中可以放心地忽略它们的存在。

首先，我们证明上述过程实际上等价于在实例 (G, s, t, k) 上求解Directed Long Detour 问题。

断言 4. 假设路径 P_{start}、P_{middle} 和 P_{end} 满足上述性质 (a)–(d)。那么拼接

$$P = P_{start} \circ P_{middle} \circ P_{end}$$

构成了该 Directed Long Detour 问题的有效解。

证明。首先，P 是一条从 s到t 的简单路径。事实上，根据性质（d），P_{middle} 和 P_{end} 内部顶点不相交。此外，由于路径 P_{start} 是 G中从 s到x 的最短路径，因此除了 x外，它不经过 $L_p \cup \ldots \cup L_n = V$（$G_{\geqslant p}$）中的任何顶点，因此 P_{start} 也与 P_{middle} 和 P_{end} 内部顶点不相交。

根据性质（b），P 的长度至少为

$$\mathsf{dist}_G(s,x)+2k+\mathsf{dist}_G(y,t)=\mathsf{dist}_G(s,y)+2k+\mathsf{dist}_G(y,t)\geqslant\mathsf{dist}_G(s,t)+2k,$$

至此，证明完成。 □

断言 5. 如果(G,s,t,k) 是一个是实例，那么存在一个整数 $p\in\mathbb{N}$ 以及顶点$x,y\in L_p$，使得存在路径 P_{start}, P_{middle} 和 P_{end} 满足性质 $(a)-(e)$。

证明。 假设 (G,s,t,k) 是一个是实例。设 P是一条能够证明这一点的从 s 到 t 的路径。若这样的路径有很多条，我们选择最短的一条作为 P。正如我们已经证明的，对于任意 $l\in\{k,k+1,\ldots,3k-1\}$，都不存在长度为 $\mathsf{dist}_G(s,t)+l$ 的从 s 到 t 的路径，因此可以假设路径 P 的长度至少为 $\mathsf{dist}_G(s,t)+3k$。由于 P 是一条非最短的从 s 到 t 的路径，我们可以将 $p\in\{1,2,\ldots,n\}$ 定义为满足以下条件的最小索引：L_p 至少包含路径 P 上的两个顶点。设 x 和 y 分别为路径 P 上位于层 L_p 的第一个和第二个顶点。

根据 p 的定义，子路径 $P[s\to x]$ 是一条从 s 到 x 的最短路径，因此我们可以设 $P_{\text{start}}\coloneqq P[s\to x]$。我们还设 $P_{\text{end}}\coloneqq P[y\to t]$。我们将在证明过程中稍后选择 P_{middle}。

我们注意到，子路径 $P[x\to y]$ 的长度至少为 $2k$：否则，考虑路径 P'，它由一条在 s中从 y 到 G 的最短路径与子路径$P[y\to t]$拼接而成。与断言 4中一样，我们论证 P'是 G 中的一条简单 s到 t路径，且 P'的长度为

$$\begin{aligned}\mathsf{dist}_G(s,y)+\mathsf{length}(P[y\to t])&=\mathsf{dist}_G(s,x)+\mathsf{length}(P[y\to t])\\&=\mathsf{length}(P)-\mathsf{length}(P[x\to y])\\&>(\mathsf{dist}_G(s,t)+3k)-2k=\mathsf{dist}_G(s,t)+k.\end{aligned}$$

因此，P' 也是我们实例 $(G,\ s,\ t,\ k)$ 的一个有效解，且比 P 短，这与假设矛盾。

由于 $P[x\to y]$ 是 G^x 中的一条$(x,\ y)$-接地路径，且与 $P[y\to t]$ 内部不相交，我们可以假设在 G^x 中有 $(x,\ y)_{\geqslant p}$；情形 $P[y\to t]\subseteq\Gamma_{\mathsf{L}}(P[x\to y])$ 则对称。

注意，如果我们令 $P_{\text{middle}}\coloneqq P[x\to y]$，那么性质 $(a)-(d)$ 就成立。若条件 (e) 也成立，则问题解决。否则，我们将 P_{middle} 定义为 $G^{x,y}_{\geqslant p}$ 中一条关于 $\prec_{\text{top}}$ 极小的 (x,y)-接地路径，其长度至少为 $2k$。于是，$P_{\text{middle}}\prec_{\text{top}}P[x\to y]$，因此 P_{middle} 与 $P_{\text{end}}=P[y\to t]$ 内部顶点不相交，因为我们有 $P[y\to t]\subseteq\Gamma_{\mathsf{R}}(P[x\to y])$。

剩下要证明的是，数对 $(P_{\text{middle}},P_{\text{end}})$ 满足性质 (e)。根据 P_{middle} 的定义，我们知道不存在长度至少为 $2k$、且在拓扑上位于 P_{middle} 左侧的 (x,y)-接地路径。现在假设存在一条 (x,y)-接地路径 P'_{middle}，位于 $G^{x,y}_{\geqslant p}$ 中，并且满足 $P'_{\text{middle}}\prec_{\text{top}}P_{\text{middle}}$，且 $\mathsf{length}(P'_{\text{middle}})\in[k,2k-1]$。考虑以下游走 P'：

$$P'=P[s\to x]\circ P'_{\text{middle}}\circ P[y\to t].$$

P' 是一条简单路径，如 $G_{\geqslant p}$ 所示，我们有 $P'_{\text{middle}}\prec_{\text{top}}P_{\text{middle}}\prec_{\text{top}}P[x\to y]$ 和 $P[y\to t]\subseteq\Gamma_{\mathsf{R}}(P[x\to y])$，因此 P'_{middle}与 $P[s\to x]$ 和 $P[y\to t]$均内部顶点不相交。此外，P' 的长度至少为

$$\mathsf{dist}_G(s,x)+\mathsf{length}\left(P'_{\text{middle}}\right)+\mathsf{dist}_G(y,t)=$$
$$\text{距离}_G(s,\ y)+\text{距离}_G(y,\ t)+\text{长度}\left(P'_{\text{middle}}\right)\geqslant\text{距离}_G(s,\ t)+\text{长度}\left(P'_{\text{middle}}\right)\geqslant\text{距离}_G(s,\ t)+k.$$

因此， P' 是实例（G， s， t， k）的有效解。最后，由于长度(P'_{middle}) $<2k \leqslant$ 长度($P[x \to y]$)，路径 P' 比路径 P 短，这与前述结论矛盾。

□

接下来要说明如何找到满足上述条件的路径 P_{start}、 P_{middle} 和 P_{end}。假设我们正确猜出了 p、 x 和 y 的值。根据断言 4，我们知道 P_{start} 可以设为任意一条从 s 到 x 的最短路径，并且我们看到，为了找到所需的路径 P_{middle} 和 P_{end}，可以将范围限制在有向图 $G^{x}_{\geqslant p}{}^{y}$ 中。若一对路径 $(P_{\text{middle}}, P_{\text{end}})$ 满足上文所述的性质 (b) – (e)，则称其对于 $(G^{x}_{\geqslant p}{}^{y}, x, y, t, k)$ 是特殊的。这样，我们便将该 Directed Long Detour 问题的实例简化为求解至多 n^2 个实例（分别对应 x 和 y 的每种取值）中的以下问题（其中 $H := G^{x}_{\geqslant p}{}^{y}$）。

Problem B
输入： 一个平面有向图 H、三个顶点 x， y， $t \in V(H)$，以及一个整数 $k \in \mathbb{N}$，其中 x 和 y 位于 H 的外面。 H 的外面是简单的，且只包含顶点 x 和 y。

问题： 是否存在一对特殊路径 (P_1, P_2)，对应于 (H, x, y, t, k)?

要完成命题1.2 的证明，只需证明以下引理。

引理 4.1。 Problem B 可以通过蒙特卡洛算法在时间 $2^{\mathcal{O}(k)} \cdot n^2$ 内求解，也可以在时间 $2^{\mathcal{O}(k)} \cdot n^2 \log n$内确定性地求解。

证明。 我们采用标准色彩编码技术 [4]的一个变体。将 H 中的每个顶点独立地染成两种颜色之一，例如绿色和蓝色，每种颜色的概率均为 $1/2$。

暂且假设 (H, x, y, t, k) 是一个是实例，并设 (P_1, P_2) 是该实例的一对特殊路径。根据前面的讨论，我们知道 $P_2 \subseteq \Gamma_{\mathsf{L}}(P_1)$ 或者 $P_2 \subseteq \Gamma_{\mathsf{R}}(P_1)$。不失一般性，假设 $P_2 \subseteq \Gamma_{\mathsf{R}}(P_1)$。回想一下，$\mathsf{length}(P_1) \geqslant 2k$。因此，我们可以将 x' 定义为 P_1 上的第 k 个顶点，并将 y' 定义为从 P_1 末端起的第 k 个顶点。于是，$\mathsf{length}(P_1[x \to x']) = \mathsf{length}(P_1[y' \to y]) = k - 1$，并且路径 $P_1[x \to x']$ 和 $P_1[y' \to y]$ 点不相交。

假设颜色的分配方式使得路径 $P_1[x \to x']$上的所有顶点都是绿色，而路径 $P_1[y' \to y]$ 上的所有顶点都是蓝色。此类事件发生的概率为 $(1/2)^{2k}$。令 $\mathcal{G}$ 为随机着色中 H 的绿色顶点集合。注意，我们假设 $V(P_1[x \to x']) \subseteq \mathcal{G}$。

断言 6. $P_1[x \to x']$ 是完全包含在 $\mathcal{G}$中的 H 里按字典序位于最左侧的 x-至-x' 路径。

证明。 假设存在一条从 x 到 x' 的路径 Q，位于 H 中，且完全包含于 $\mathcal{G}$ 中，使得 $Q \prec_{\text{lex}} P_1[x \to x']$。令 $q \in V(H)$ 为顶点，使得路径 $P_1[x \to q]$ 是 P_1 和 Q 的最长公共前缀。此外，令 r 为 $Q[q \to x']$上的第一个顶点（不包括 q），使得 $r \in V(P_1)$。这里， r 定义良好，因为顶点 x' 同时属于 $Q[q \to x']$ 和 P_1。由于 r 位于 Q 上，我们必然有 r 是绿色的。因此，$r \notin V(P_1[y' \to y])$，因为 $P_1[y' \to y]$ 仅由蓝色顶点组成。于是，以下游走 P'_1:

$$P'_1 = Q[x \to r] \circ P_1[r \to y]$$

是 H 中一条长度至少为.....的 (x, y)-接地路径

$$\mathsf{length}\big(P_1[y' \to y]\big) + 1 = k.$$

回想一下， $Q \prec_{\text{lex}} P_1$ 属于 H。根据 r 的定义，我们观察到 $Q[x \to r] \prec_{\text{top}} P_1[x \to r]$。因此，$P_1' \prec_{\text{top}} P_1$。由此可知，路径 P_1' 的长度至少为 k ，且位于 P_1的左侧，这与点对（P_1, P_2）的性质（e）矛盾。 □

注意，路径 $P_1[x \to x']$ 与 H外面上的顶点集合不相交，除了其第一个顶点 x。因此，我们可以将 H^{cut} 定义为沿路径 $P_1[x \to x']$切割H 后得到的平面有向图。接下来，令 $Z \subseteq V$（H^{cut}）表示通过此操作向 H' 中引入的新顶点集合。令 H' 为H^{cut}外面的一个（y, {v14>）-简化。

断言 7. $P_1[x' \to y]$ 是 H' 中所有连接 x' 和 y 且不包含 Z中任何顶点的路径里最左的路径。

证明。 假设存在一条 x'到y 的路径 Q ，它位于 H' 中，不经过 Z ，并满足 $Q \prec_{\text{lex}} P_1[x' \to y]$。设 $q \in V$（H）是 H 中的顶点，使得 $Q[x' \to q]$ 是 Q 和 $P_1[x' \to y]$的最长公共前缀。设 r 是 $Q[q \to y]$ （不包括 q）上的第一个满足$r \in V$（P_1）的顶点。

与断言 6的证明类似，我们证明游走 P_1' 按如下方式定义于 H：

$$P_1' = P_1[x \to x'] \circ Q[x' \to r] \circ P_1[r \to y]$$

是一条长度至少为 k 的简单路径，在 H中按字典序位于 P_1 左侧。这与（P_1, P_2）的性质（e）矛盾。 □

上述观察导出了一个用于 Problem B 的算法。首先，考虑解 (P_1, P_2) 中满足 $P_2 \subseteq \Gamma_{\mathsf{R}}(P_1)$ 的情况。对 H 的顶点随机赋予颜色后，我们依次遍历所有顶点 $u \in V(H) \setminus \{x, y, t\}$，将其作为顶点 x' 的候选。然后，在 H 中寻找从 x 到 u 的最左路径 S，其所有顶点均为绿色。接下来，构造 H^{cut} 和 H'，并在 H' 中寻找从 u 到 y 的最左路径 T，该路径避开了分割向 H' 引入的所有新顶点。最后，令 $P_1 = S \circ T$。接下来只需验证在 H 中是否存在一条从 y 到 t 的路径 P_2，与 P_1内部不相交。所有这些检查都可以通过适当的深度优先搜索在线性总时间内完成。

接下来，我们考虑 $P_2 \subseteq \Gamma_{\mathsf{L}}(P_1)$ 的情形。因此，我们重复上述所有步骤，将 H 中任意的最左路径替换为对应的最右路径。为了获得恒定错误概率，我们将上述整个过程重复 $2^{\mathcal{O}(k)}$ 次。

为了对该算法进行去随机化，我们不再随机地为顶点分配颜色，而是构造并使用一个 (n, $2k$)-通用集 [7]。在我们的情形中，该通用集是由 $V(H)$ 的子集组成的一个族 $\mathcal{U}$，使得对于 $V(H)$ 的任意大小为 $2k$ 的子集 A，族 $\{A \cap U \mid U \in \mathcal{U}\}$ 都包含 A 的所有 2^{2k} 子集。我们将 $\mathcal{U}$ 中的每个元素视为一种单独的着色；也就是说，子集 $A \in \mathcal{U}$ 对应于着色 c，该着色将绿色分配给 A 中的顶点，并将蓝色分配给其余顶点。

我们知道，存在这样一个大小为 $2^{\mathcal{O}(k)} \cdot \log n$ 的族 $\mathcal{U}$，并且可以在 $2^{\mathcal{O}(k)} \cdot n \log n$ [7] 时间内构造出来。我们在算法中只需构造一次，然后不再随机抽取着色，而是遍历所构造的通用集 $\mathcal{U}$ 中的所有元素。根据

$\mathcal{U}$ 的性质，存在一个元素 $U \in \mathcal{U}$，使得 $V(P_1[x \to x']) \subseteq U$ 且 $V(P_1[y' \to y]) \cap U = \emptyset$，因为 $|V(P_1[x \to x']) \cup V(P_1[y' \to y])| = 2k$。这保证了我们的确定性算法的正确性。 □

致谢。作者感谢 Olek ˙Lukasiewicz 指引我们关注 Wu 和 Wang 的研究 [9]。

参考文献

[1] Ivona Bez´akov´a、Radu Curticapean、Holger Dell 和 Fedor V. Fomin。寻找绕行路径是固定参数可解的。*SIAM* 离散数学杂志，33(4):2326–2345，2019年。[2] Marek Cygan、D´aniel Marx、Marcin Pilipczuk 和 Micha˙l Pilipczuk。平面有向点不相交路径问题是固定参数可解的。载于第*54*届 *IEEE* 计算机科学基础年度研讨会（*FOCS 2013*），第197–206页。IEEE计算机学会，2013年。[3] Fedor V. Fomin、Petr A. Golovach、William Lochet、Danil Sagunov、Kirill Simonov 和 Saket Saurabh。《有向图中的绕行路径》。载于第*39*届理论计算机科学国际研讨会（*STACS 2022*），第219卷，*LIPIcs*，第29:1–29:16页。Schloss Dagstuhl — Leibniz 信息学中心，2022年。[4] Fedor V. Fomin、Daniel Lokshtanov、Fahad Panolan、Saket Saurabh 和 Meirav Zehavi。长有向 (s, t)-路径：FPT 算法。信息处理快报，140:8–12，2018年。[5] Steven Fortune、John E. Hopcroft 和 James Wyllie。有向子图同胚问题。理论计算机科学，10:111–121，1980年。[6]John E. Hopcroft 和 Robert Endre Tarjan。高效平面性测试。*ACM* 杂志，21(4):549–568，1974年。[7] Moni Naor、Leonard J. Schulman 和 Aravind Srinivasan。分裂器与近最优去随机化。载于第*36*届计算机科学基础年度研讨会，威斯康星州密尔沃基，美国，*1995*年*10*月*23*日至*25*日，第182–191页。IEEE计算机学会，1995年。[8] Alexander Schrijver。在有向平面图中寻找 k 点不相交路径。*SIAM* 计算机杂志，23(4):780–788，1994年。[9] Bang Ye Wu 和 Hung-Lung Wang。具有正边权的有向图次短路径问题。*Networks*，65(3):205–211，2015年。