平面生成树之间最短翻转序列的结构性质

Oswin Aichholzer 算法与理论研究所，格拉茨技术大学，奥地利

Joseph Dorfer 算法与理论研究所，格拉茨技术大学，奥地利

Peter Kramer
计算机科学系，布伦瑞克工业大学，德国

Christian Rieck
卡塞尔大学数学研究所，德国

Birgit Vogtenhuber
格拉茨技术大学算法与理论研究所，奥地利

arXiv:2603.05205v1 [cs.CG] 5 Mar 2026

摘要

我们研究平面上处于凸位置的点集上的无交叉生成树重构问题。翻转 是一种重构步骤，即将树的一条边替换为另一条边，从而再次得到一棵无交叉生成树。对于给定的点集 P，翻转图 的每个顶点对应 P 上的一棵无交叉生成树，而任意两棵仅相差一次翻转的树之间都有一条边。两棵树之间的翻转距离是它们在翻转图中的最短路径长度，也就是将一棵树变为另一棵树所需的最少翻转次数。

我们研究最短翻转序列的结构性质。多年来，人们针对这些序列提出了若干猜想。广为流传的快乐边猜想 认为，初始树和目标树共有的任何边都不会在最短翻转序列中被翻转。此外，停车猜想 断言，在最短翻转序列中，每条边要么翻转为一条目标边，要么停放在凸包上。值得注意的是，快乐边猜想可由停车猜想推出。还有人猜想，一条边不会被再次停放；也就是说，翻转产生的任何边要么是目标边，要么随后会被翻转为目标边。从本质上说，近期所有翻转算法都依赖于这三个猜想及其所蕴含的性质。

本文中，我们否证了停车猜想和重新停车猜想。具体而言，我们证明，满足停放性质的翻转序列有时所需的翻转次数会比最短翻转序列多出线性数量；我们还给出了这样的例子：任何最短翻转序列都会多次翻转某些边。与此相反，我们找出了若干使停放性质和重新停车性质成立的条件；这些条件意味着相容翻转满足重新停车性质，并且算法可以利用它们来获得较短的翻转序列。

2012年 ACM 学科分类计算理论 → 计算几何

关键词与短语生成树、凸位置、无交叉、翻转图、快乐边、停车猜想

资助*Joseph Dorfer*：奥地利科学基金（FWF）10.55776/DOC183。*Peter Kramer*：德国学术交流中心（DAAD）奖学金。*Christian Rieck*：由德国研究联合会（DFG，德国研究基金会）资助——522790373。

致谢 本研究始于2025年3月在格拉茨工业大学举办的重构周。我们感谢组织者和所有参与者进行了富有成效的讨论，并营造了非常良好的氛围。

1 引言

重构问题关注的是：一个系统的某个可行状态能否通过一系列允许的操作转变为另一个状态。魔方和数字华容道等经典谜题让这一概念变得直观易懂。以魔方为例，旋转一个面（ 90° 或 180°）就构成一个基本操作，而通过一系列这样的面旋转，可以遍历一个约有 $4.3 \cdot 10^{19}$ 个可达构型的庞大状态空间。状态空间的结构，也就是各个构型如何通过允许的操作相互连接，可以用相应的重构图来形式化描述：顶点表示构型，边则对应单个操作。尽管规模极其庞大，魔方的重构图直径却只有20，因此任何打乱的魔方最多经过20次面旋转就能还原。这一视角的应用远不止于谜题：在组合数学、（计算）几何以及理论计算机科学等领域，重构图都是研究可相互转换的对象族的工具。

几何重构的另一个自然且直观的例子，是凸点集的三角剖分重构。在此背景下，翻转是一种局部操作：将由两个相邻三角形构成的凸四边形的一条对角线替换为另一条，从而得到新的三角剖分。重构图（也称翻转图）是结合子的 一维骨架。这些三角剖分与二叉树一一对应 [36], ，每次翻转恰好对应于相关树中的一次旋转。因此，将一个三角剖分转换为另一个所需的翻转次数，与重组相应二叉树所需的旋转次数相同。由于二叉搜索树是一种基础数据结构，理解并尽量减少这些旋转，对于提升依赖树操作的算法效率至关重要。尽管已知结合子的直径等于 $2n-6$ （当 $n \geq 11$时），但是否存在一种多项式时间算法来计算旋转距离 [34, 36]，仍是一个悬而未决的问题。在一篇最新的预印本中，Dorfer [20] 对此给出了否定答案。

本文研究凸位置点集上的无交叉生成树的重构。无交叉生成树中的一次翻转 会从树中删除一条边并加入另一条边，从而再次得到一棵无交叉生成树。相应的翻转图 以凸集 P 上的 n 个点所构成的所有无交叉生成树为顶点，当且仅当对应的树只相差一次翻转时，两个顶点之间才有一条边。两棵树之间的翻转序列 是翻转图中从表示一棵树的顶点到表示另一棵树的顶点的任意路径，而翻转距离 则是此类路径中最短路径的长度。图1展示了非凸位置下4个点的平面生成树的翻转图。

近期研究主要关注：（1）翻转图的直径，即凸点集上任意一对树之间的最大翻转距离；（2）翻转距离计算的算法方面。二十多年来，直径界的基准结果一直是上界 $2n-4$ [11] 和下界 $\lfloor \frac{3n}{2} \rfloor - 5$ [24]。后来，直径上界改进为 $2n - \log(n)$ [1], ，不久后又改进为 $2n - \sqrt{n}$ [18]。在 [17], ，研究者证明直径上界为 $\approx 1.96n$，突破了长期存在的 $2n$ 障碍。目前已知的最佳界为：Bjerkevik、Dorfer、Kleist、Ueckerdt和Vogtenhuber给出的下界$(\frac{11}{7} - o(1))n$ [13], ，以及Bjerkevik、Kleist、Ueckerdt和Vogtenhuber给出的上界 $\frac{5n}{3} - 3$ [14]。

解决上述两个问题的一个潜在有用工具，是起始树与目标树所共有的边集，称为快乐边。所有其他边则称为不快乐边。基于这一概念，以下长期存在的猜想被视为 folklore。

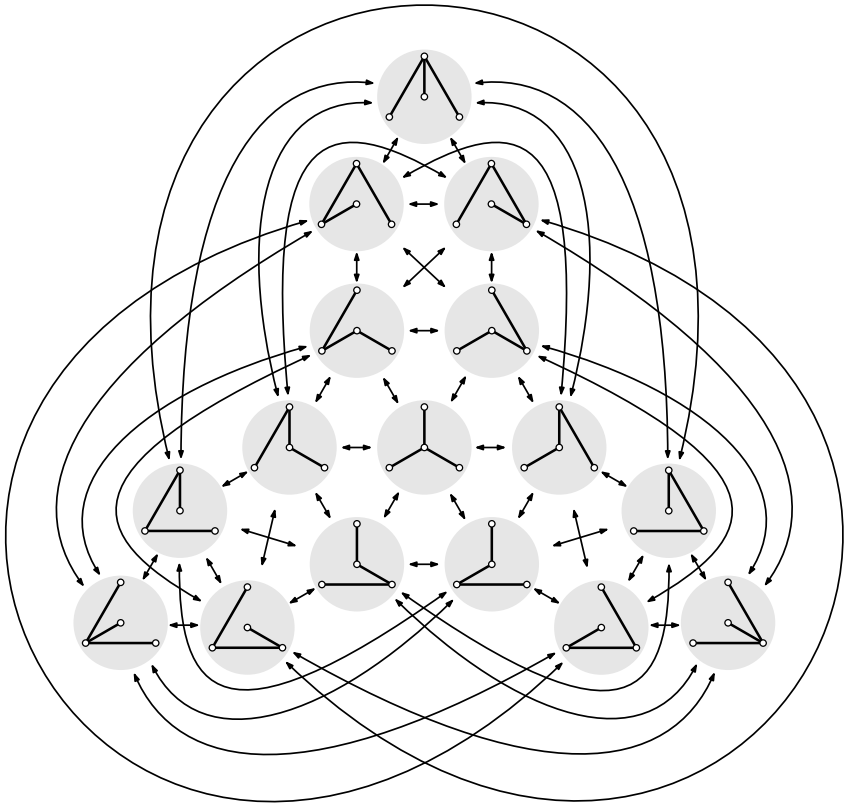


图1 非凸位置中 $n = 4$ 个点的平面生成树翻转图。谜题：你能找出直径的所有 120 种不同实现方式吗?

▶**猜想1** （快乐边猜想）. 设 T_I 与 T_F 为两棵平面生成树，并设 $H = T_I \cap T_F$ 为所有快乐边的集合。则存在一个最短翻转序列 $T_I = T_0$、T_1、.....、T_{k-1}、$T_k = T_F$，使得 $H \subset T_i$ 对每一个 $i \in [0, k]$成立。

上述大多数对直径界的改进都基于遵循这一猜想的路径，也就是说，它们从不翻转快乐边。具体而言， [1] 的作者将其 $2n$ –log(n) 界表述如下：

“.....快乐边不会被翻转。 T_I 的一条不快乐边会翻转一次或两次；如果翻转两次，则先翻转到凸包上，再翻转为 T_F 的一条边。”

此外，他们将沿翻转序列中的某棵树出现、但在起始树和最终树中都不存在的边定义为停车边 ，并猜想只需在凸包上停放即可：

▶ **猜想 2** (停车边猜想)。 对于任意两棵平面生成树 T_I 和 T_F，都存在一条从 T_I 到 T_F 的最短翻转序列，且只使用 $T_I \cup T_F \cup$CH 的边。

尽管这并未被明确表述为一个猜想，但上述引文暗示，始终存在一种最短翻转序列，其中每条不快乐边都恰好翻转一次或两次，这促使我们提出以下猜想。

▶ **猜想3** (重新停车猜想)。 对于任意两棵平面生成树 T_I 和 T_F，都存在一个最短翻转序列，使得每条停车边都翻转为 T_F的一条边。特别是，每条边最多翻转两次。

迄今为止，所有能为直径翻转提供非平凡上界的算法，要么直接将边翻转为最终边，要么先将它们停放在凸包上，再从那里翻转为最终边 [1, 6, 14, 17, 18]；这一事实进一步支持了该猜想。详情请参见表 1。

表 1 自2022年以来用于上界构造的算法及其所使用的猜想的翻转性质。我们突出显示了算法所使用的、已证明并不总是最优的性质所在的单元格。

上界 [参考文献]	翻转类型	猜想1	猜想 2	猜想3
$2n - \log(n)$ [1]	一般翻转	Yes	Yes	Yes
$2n - \sqrt{n}$ [18]	一般翻转	No	Yes	Yes
$1.96n$ [17]	一般翻转	Yes	Yes	Yes
$\frac{5n}{3} - 3$ [14]	一般翻转	Yes	Yes	Yes
$\frac{5n}{3} - 2$ [6]	相容翻转	Yes	Yes	Yes
$\mp 4(n-1)$ [6]	旋转	No	No	Yes

关于这三个猜想的一些初步结果和结构性见解如下：(i) 停车边猜想可直接推出快乐边猜想 [1],；(ii) 当允许的翻转仅限于边滑动 [1] 或旋转 [6]时，快乐边猜想不成立；精确定义见第 2 节。相比之下，(iii) 停车边猜想（因而快乐边猜想）在相容翻转下成立 [6]。对于无限制翻转的情形，这些猜想一直悬而未决。 [14],中给出了目前已知的直径最佳上界，其作者对他们的上界构造作了如下描述：

> “我们注意到，我们为 *[14,* 定理*26*、*arXiv*版本*],*（这是我们最强的上界）构造的翻转序列，具有这两项性质 *[*猜想*1* 和猜想 *2],*，这为这些猜想提供了一些支持。”

1.1 我们的贡献

我们通过揭示不同情形下的结构性见解，推动了这些猜想的研究。这反过来也让我们对凸点集上平面生成树之间最短翻转序列的性质有了新的认识。具体而言，我们得到以下结果。

首先，我们证明了最短翻转序列的一个有用性质：一次翻转要么是最终翻转，要么被移除的边会在序列后续过程中被交叉。此外，我们通过定理 5 证明，停车猜想蕴含重新停车猜想，且猜想3对相容翻转成立。同时，始终存在一种最短翻转序列，其中每次移除凸包边的翻转都会添加一条属于 T_F 的边。

▶ **定理 4** （最终翻转性质）。对于任意两棵平面生成树 T_I 和 T_F，存在一条最短翻转序列，使得每次翻转要么是最终翻转，要么被移除的边会在之后被另一条边交叉。

▶ **定理 5**。考虑点集 T_I 中的任意两棵平面生成树 T_F。 P。**1**。存在一条最短翻转序列，使得任意凸包边至多翻转一次。**2**。在相容情形下，所有边至多翻转两次 （相容重新停放性质）。

此外，我们通过证明某些实例中仅在凸包上引入停车边的算法所生成的序列远非最优，推翻了猜想 2。更确切地说，我们证明：

▶ **定理 6**。 存在一族树$(T_{I,\ k})_{k\in\mathbb{N}}$和$(T_{F,\ k})_{k\in\mathbb{N}}$，它们有 $10k+2$ 个顶点，且仅使用凸包上的停车边的最短翻转序列，其长度至少比可能的最短翻转序列长 k 次翻转。

最后，我们在无限制情形下否定了猜想3，并得到如下结果：

▶ **定理 7.** 存在平面生成树对 T_I与 T_F，以及 T_I与 T'_F，分别位于 22 个点和 32 个点上，使得从 T_I 到 T_F、相应地从T_I到 T'_F 的任意最短翻转序列，都会将同一条边分别翻转 3 次和 4 次。

概要。 我们首先在第 1.2 节概述其他相关工作，并在第 2 节给出一般定义。随后，我们在第 3 节给出正面的结构性结果，再分别在第 4.1 和 4.2 节表明：一般而言，停放性质与重新停车性质对无限制翻转均不成立。

由于篇幅限制，标有 (⋆) 的陈述的证明见附录。

1.2 更多相关工作

平面点集上无交配置的翻转图已针对其他图类得到广泛研究，包括三角剖分 [8, 21, 22, 23, 27, 28, 31, 33, 34, 36], 生成路径 [4, 7, 10, 29, 35], 和匹配 [2, 3, 12, 25, 26]; 另请参阅 Bose 和 Hurtado 撰写的综述 [16]。上述若干研究依赖于与我们针对平面生成树提出的相似结构性观察。

在 [36], 中，作者证明了凸点集三角剖分具有快乐边性质。此外，他们表明，只要条件允许就加入目标三角剖分中的边这一贪心策略，总能生成最短翻转序列。这些见解为若干固定参数可 tract 算法 [15, 19, 30, 32], 奠定了基础，其中翻转距离是参数。反之，对于一般位置的点集三角剖分，快乐边性质并不成立 [16], ；针对一般位置点集 [31, 33], 以及简单多边形 [8]情形的 NP难性证明，关键就在于该性质的反例。

类似地，一般位置点集上完美匹配的翻转距离问题已被证明是 NP 难的 [12]。该证明同样依赖于快乐边性质的反例。当点集被限制为凸点集时，该问题是多项式时间可解的，因为目标匹配中的一条边总是可以贪心地插入 [25]。

尽管快乐边性质并不成立，凸点集上平面路径之间的最短翻转序列仍可在多项式时间内计算 [4]; 该证明的一个关键要素是将快乐边划分为“好的”和“坏的”快乐边，然后分别处理。有时只需考虑凸包上的快乐边即可；例如，对于凸点集上的问题，如平面生成树 [1], 或奇匹配 [2] ，凸包边满足快乐边性质。在这两种情形中，这一观察对于推导相关的固定参数可 tract 算法 [2, 6]起到了关键作用。

除复杂性结果之外，针对快乐边性质的反例已被推广，用以确立凸点集上平面生成树之间滑动距离的二次下界 [9]。

2 预备知识

在本文中，我们用 $P\subset\mathbb{R}^2$ 表示处于凸位置的 n 点集，即满足没有任何一点 $p\in P$ 是其他任意三个点的凸组合的集合 P。对于一

对点 u, $v \in P$，我们用 uv 表示连接这两点的线段。如果 u和 v 在 P的凸包上相邻，则称 uv 为凸包边，并记作 $uv \in \mathrm{CH}$。 P的平面生成树 是由 $n-1$ 条边 $T \subset \binom{P}{2}$ 组成的树，其中各边对应的线段两两不相交，或仅在公共端点处相交。我们用 $T-e$和 $T+e$分别表示删除和插入一条边 e，其他情况则使用标准集合记法。

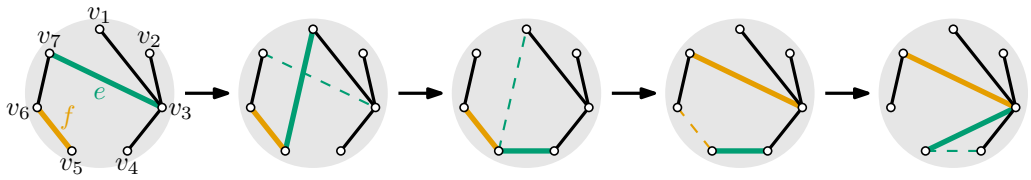


图2 一个翻转序列$\mathcal{S}$的轨迹为$(e,\mathcal{S}) = (e,\ v_1v_5,\ v_4v_5,\ v_3v_5)$，且轨迹$(f,\mathcal{S}) = (f,\ v_3v_7)$。按顺序，这四次翻转分别是交叉、旋转、相容翻转和边滑动。

翻转。在 T 中，翻转会移除一条边 $e \in T$ 并添加一条边 $f \notin T$，使所得结构 $T-e+f$ 再次成为平面生成树。如果初始树 T可由上下文确定，我们就用 $e \to f$ 表示此操作，并记作 e 翻转为 f。我们区分四种类型的翻转；若线段 e 和 f 互不相交或共享一个端点，则翻转 $e \to f$ 是相容的；否则称为交叉。对于 $u, v, w \in P$，形状为 $uv \to uw$ 的相容翻转称为旋转；此外，若 $vw \in T$，则称为滑动。图2展示了所有类型翻转的示例。

翻转图。 P 的平面生成树的翻转图 是一个以所有此类树为顶点的图。当且仅当两棵生成树可以通过一次翻转相互转换时，它们在翻转图中共享一条边。给定两棵树 T_I 和 T_F，从 T_I 到 T_F 的翻转序列 $\mathcal{S}$ 是翻转图中的一条路径，记作一系列树 $T_I = T_0$、T_1、……、T_{k-1}、$T_k = T_F$，其中相邻的树仅相差一次翻转；其长度是该序列中翻转的数量 k。 T_I 和 T_F之间的翻转距离 记为 $d(T_I,\ T_F)$，等于最短序列的长度。此外，边 $T_I \cap T_F$是快乐的，其余所有边都是不快乐边。如果一次翻转引入了一条边 $f \notin T_F$，则通常称其为停车翻转，而 f 称为其停车边。翻转图的直径 表示任意一对树之间的最大翻转距离。

轨迹。 我们引入额外的记号来分析翻转序列。具体而言，我们引入一些结构，不仅可以表示树在翻转操作下发生的变化，还可以表示这些翻转操作如何影响各条边。给定一个翻转序列 $\mathcal{S} = T_0, \ldots, T_k$，我们将边 $e \in T_0$ 的轨迹 定义为该边经翻转后所处位置构成的有序序列；图2给出了一个示例。形式上，这对应于一个边的元组：

$$\mathrm{trace}(e,\mathcal{S}) = \mathrm{trace}(e,T_0,\ldots,T_k) = \begin{cases} (e) & \text{if } k=0, \\ \quad \mathrm{trace}(e,T_1,\ldots,T_k) & \text{if } e \in T_0 \cap T_1, \text{ or} \\ (e) \parallel \mathrm{trace}(f,T_1,\ldots,T_k) & \text{if } \{e,f\} = T_0 \Delta T_1. \end{cases}$$

我们称轨迹长度 为其中翻转的数量，它对应于元素数量减一：轨迹 $(e_1, \ldots, e_\ell)$ 对应于翻转 $e_i \to e_{i+1}$ for $1 \le i < \ell$。注意，$\mathcal{S}$ 的长度恰好等于所有轨迹长度之和。若$\mathcal{S}$ 中的翻转 $e \to f$ 在剩余序列上的 f 轨迹为 (f)，则该翻转是最终的；否则为非最终的。若幸福边 $e \in T_0 \cap T_k$ 的轨迹长度为零，则称其在翻转序列$\mathcal{S}$ 中固定；等价地，

trace(e,$\mathcal{S}$) = (e)。这根据最短翻转序列赋予幸福边两种性质。如果某条幸福边在某个最短翻转序列中是固定的，则称其弱固定；如果它在所有此类序列中都是固定的，则称其强固定。

我们注意到，猜想1–3也可以用轨迹性质来表述：快乐边猜想（猜想1）指出，所有快乐边都是弱固定的。类似地，停车猜想（猜想 2）指出，始终存在一个最短翻转序列，使得其中<样式 id='3'>所有轨迹中除第一个和最终元素之外的元素都位于CH 中；而重新停车猜想（猜想3）指出，存在一个最短翻转序列，使得 T_I 的每条边的轨迹长度至多为二。

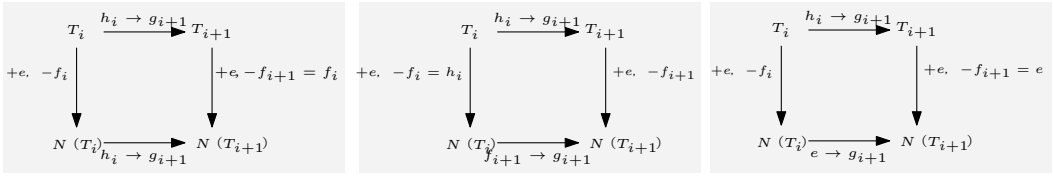
3 结构性质的正面结果

我们推广了 Aichholzer、Dorfer 和 Vogtenhuber 在 [6, 引理 20]中证明的一个性质。他们证明，在每个相容翻转序列中，任何将停车边停放在一条对角线上的停车翻转，都可以替换为将停车边停放在凸包上的停车翻转。实际上，并不要求整个翻转序列都是相容的，只需插入停车边的翻转或移除停车边的翻转中的一个是相容的即可。这一点已经可以从他们的证明中推出。为使论述自洽，我们给出经过调整的证明，并用我们的定义重新表述该结论。

▶**引理 8** ($\star$)。设有一条连接两棵树 T_I 和 T_F 的翻转序列，以及一条边 e，使得 $trace(e) = (e, f, e')$，其中 f 是一条对角线，且翻转 $e \to f$ 或翻转 $f \to e'$ 兼容。则存在一条从 T_I 到 T_F、长度相同的翻转序列，其中 $trace(e) = (e, h, e')$，且 h 是凸包边。任何其他边的轨迹均保持不变。

为了研究重新停车性质，我们首先考察边具有轨迹长度 1 的充分性质，或者等价地，考察边具有更大轨迹长度的必要条件。

▶ **定理 4** (最终翻转性质)。对于任意两棵平面生成树 T_I 和 T_F，存在一条最短翻转序列，使得每次翻转要么是最终翻转，要么被移除的边会在之后被另一条边穿过。



(a) (b) (c)

图 3 关于翻转可能存在差异的交换图。水平箭头表示翻转，因此用翻转记号标注。竖线表示规范化操作，并用集合记号表示，因为它们并非传统意义上的翻转。从左到右：(a) 边 h 和 g 不在与 e相同的环上；(b) 边 f 和 g 与 e位于同一个环上；(c) 包含 e的环上的最后一次翻转。

证明。反设，令 T_I 和 T_F 是两棵树，满足：(1) 它们之间的任意最短翻转序列都包含一条边 e，其轨迹长度大于 1，且 e 不与该翻转序列中的任何其他边相交；(2) T_I 与 T_F 之间的翻转距离为

在所有具有此类最短翻转序列的树对中最小。令 $T_I = T_0, T_1$、.....、 $T_k = T_F$是从 T_I 到 T_F的一个此类翻转序列。不失一般性，我们可以假设 e的轨迹为（e、 f、 e'），且 $e \in T_I = T_0$和 $e' \in T_F = T_k$，否则我们可以考虑一对距离更近的树，从而缩短该翻转序列。换言之，翻转 $e \to f$ 是从$T_I = T_0$ 到 T_1 进行的，而翻转 $f \to e'$ 是最后一次翻转，即从 T_{k-1} 到 $T_k = T_F$。

就我们的目的而言，我们采用一种称为规范化的技术，该技术在 [36, 引理 3]中针对三角剖分引入。此后，这一概念已被修改用于匹配 [25] 和树 [1,命题 18], ，我们在此基础上进一步加以调整。对于树，规范化应用于一条翻转序列，并相对于一条边 e 来定义，该边不被属于此翻转序列的任何其他边所交叉。规范化将一棵树 T映射到另一棵树 N（T），使得映射 N（$\cdot$）仅依赖于原始翻转序列和 e。对于原始翻转序列$\mathcal{S} = T_0$， T_1，.....， T_k 以及一条边 e，我们希望规范化$\mathcal{S}_N = N$（T_0），N（T_1）.....， N（T_{k-1}），N（T_k）具有以下性质：

（1） N（T_0）$= T_0$ 与 N（T_k）$= T_k$。**（2）**对于两棵连续的树 T_i， T_{i+1}，我们有 N（T_i）与 N（T_{i+1}）要么重合，要么仅相差一次翻转。**（3）**轨迹（e，$\mathcal{S}_N$）的长度为1。

注意，在性质（2）中，对于我们的序列 $\mathcal{S}$，不可能出现两个连续的规范化树相同的情况，否则就会得到一个更短的翻转序列，从而产生矛盾。

现在说明如何为关于 e 的翻转序列中的树 T_i 进行这种规范化。如果 e 已经是 T_i的一部分，那么我们只需令 $N(T_i) = T_i$。否则，将 e 添加到 T_i中。此时， T_i包含一个环 c_i，且 e 位于其上。定义 $N(T_i) = (T_i \cup \{e\}) \setminus \{f_i\}$，其中 f_i 要么是从 T_i 到 T_k 的剩余翻转序列中， c_i 上第一个被移除的边；要么，如果不存在这样的边，则令 $f_i = e$。

现在我们论证 $N(\cdot)$ 具有所有所需的性质。对于性质（1），我们注意到 $N(T_0)$ 包含 e，且根据构造 $N(T_0) = T_0$。如果 e属于 T_k，由同样的论证可得 $N(T_k) = T_k$。如果 e 不属于 T_k，则通过添加 e，我们得到所述的平面环 c_k 。由于翻转序列以 T_k 结束，因此在翻转序列之后的任何步骤中， c_k 的边都不会被移除；我们再次移除 e ，得到 $N(T_k)$。由此可知，在这两种情况下， $N(T_k) = T_k$。

对于性质（2），在从 T_i 到 T_{i+1} 的翻转中，我们将被移除的边记为 h_i ，将添加的边记为 g_{i+1}。考虑 $T_i \cup \{e\}$ 中包含 e 的环 c_i 。若 h_i 不属于 c_i，则 $c_{i+1} = c_i$ 且 $f_{i+1} = f_i$。规范化树 $N(T_i)$ 与 $N(T_{i+1})$ 同样因翻转 $h_i \to g_{i+1}$ 而不同。该情形由图 3(a) 中的交换图所示。若 h_i 是 c_i 的一部分，则在规范化中我们有 $h_i = f_i$。此外， $c_i \neq c_{i+1}$，因为我们从 T_i 中移除 h_i 并添加 g_{i+1} 以得到 T_{i+1}。当从 $N(T_i)$ 中移除 f_{i+1} 并添加 g_{i+1} 时，我们得到 $N(T_{i+1})$。换言之，从 T_i 到 T_{i+1} 的翻转 $h_i \to g_{i+1}$ 被变换为从 $N(T_i)$ 到 $N(T_{i+1})$ 的翻转 $f_{i+1} \to g_{i+1}$ ，见图 3(b)。因此我们得出结论：性质（2）成立。

对于性质（3），注意，只要在步骤 k 中，环 c_k 中存在一条之后会被翻转的边 $f_k \neq e$， e 就属于树 N（T_k）。考虑 $\mathcal{S}_N$ 中最后一次为某个 k 向 c_k 添加边的翻转。在这次翻转中， e 从 N（T_k）中被移除，并翻转为环 c_k 中的一条边。由于现在环 c_k 完全由在 $\mathcal{S}$ 的剩余部分中不再被翻转的边组成，性质（3）得证。

借助新的翻转序列 $N(T_0)$, $N(T_1) \ldots, N(T_{k-1})$, $N(T_k)$，我们已确保 e不再是一条被多次翻转、却在翻转序列的任何时刻都未被交叉的边。然而，我们可能在新的翻转序列中引入了另一条具有这些性质的边。如果不存在这样的边，则证明即告完成。因此，在剩余的

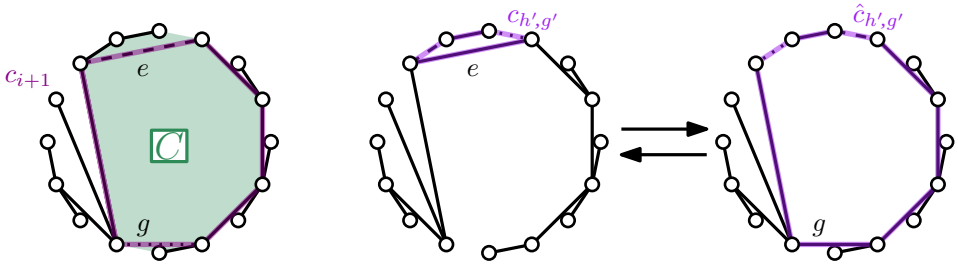
证明中，我们假设又有另一条边被多次翻转，但未被翻转序列 $N(T_0)$，$N(T_1)\ldots,N(T_{k-1})$，$N(T_k)$ 中的任何其他边所交叉。为了能够同时表明定理 4 对所有边成立，我们采用了 [6, 引理 20] 中所使用的技术。

如果一条边 $s=ab$ 包含在某个翻转序列的所有树中，则 s 将该凸点集分成两侧 A 和 B，即按顺时针顺序出现在 a 与 b 之间的顶点，以及按逆时针顺序出现的顶点。顶点 a 和 b 同时计入两侧，即侧 A 和侧 B。对于每棵树，我们得到两棵子树，即 A 和 B上的导出子树。其中一棵树中的每一次翻转都与另一侧的翻转完全独立。因此，我们可以将两个仅翻转两侧之一中的边的有序子序列交错，从而得到具有相同初始树和最终树的有效翻转序列。我们将据此论证：轨迹$(e,\mathcal{S}_N)$中唯一的翻转（记为 $e\to g$）与$\mathcal{S}_N$中的所有后续翻转相互独立。由此即可推出，$e\to g$ 可以作为最后一次翻转来执行。

为此，假设从 $N(T_i)$ 到 $N(T_{i+1})$ 的翻转是移除 e 并插入 g 的最终翻转。我们沿着 $c_{i+1}\setminus\{e,g\}$ 的所有边将凸点集分割开来。每棵树$N(T_i)$ $N(T_{i+1})\ldots,N(T_k)$ 现在都通过沿着 $|c_{i+1}|-1$ 棵树的边进行分割而分裂为 $c_{i+1}\setminus\{e,g\}$（回想一下，对于任意 $c_j=c_{i+1}$，$j\geq i+1$）。所得子集中的这些树的翻转序列彼此独立，可以按任意顺序交错执行。

分割点集后，存在一个特定的顶点子集 C，如图4(a)所示，其中包含 e 和 g 的顶点。根据前面的讨论，我们可以在 C 中执行翻转序列，该序列还执行翻转 $e\to g$，并将其作为所有子集中的翻转序列里的最后一个翻转序列。

接下来还需论证，我们可以将翻转 $e\to g$ 作为 C中的最后一次翻转来执行。为此，我们论证所有其他翻转都可以执行，而与当前树中是否存在 e 或 g 无关。C 有三个部分，其边界由 e 和 g 构成：不包含 g的 e一侧 C_e 、不包含 e的 g 一侧 C_g ，以及 e与 g之间的部分。后一部分恰好包含 c_{i+1}的边。因此，由于 c_{i+1} 的任何边都未被翻转，该部分中除 $e\to g$ 之外不会发生其他翻转。



(a) 环 c_{i+1}的图示。 **(b)** 将$c_{h',g'}$ 与$\hat{}c_{h',g'}$ 相互转换

图4 定理 4 的证明中相关概念的图示。

考虑在 C_e 中进行一次翻转。如果我们想在 C_e 中翻转 $h'\to g'$，那么将 g' 加入 C_e 中涉及的树后，会得到一个环$c_{h',\,g'}$ 其中包含 h' 和 g'。如果 $c_{h',\,g'}$ 既不包含 e，也不包含 g，那么显然，先进行翻转还是后进行翻转 $h'\to g'$，并无区别。如果 e 或 g 中有一个位于 $c_{h',\,g'}$ 中（注意，它们不可能同时位于该环中），那么我们可以论证二者能够相互替换。此时，包含 g'和 h' 的另一种环是 $c_{h',\,g'}$ 与 c_{i+1}的对称差，即 $\hat{}c_{h',\,g'}=c_{h',\,g'}\Delta c_{i+1}$。示例见图4(b)。如果 $c_{h',\,g'}$ 包含边 e，那么 $\hat{}c_{h',\,g'}$ 包含绕行

经由 $c_{i+1}\setminus\{e\}$，其中包括 g。反之，如果 $c_{h',\,g'}$ 包含 g，因此 $c_{i+1}\setminus\{e\}$，那么 $\hat{}c_{h',\,g'}$包含 e。对于 C_g 中的翻转，我们可以对称地进行论证。

为了最终证明该定理，我们总结如下：可以重新排列翻转序列中的翻转，使翻转 $e\to g$（它是一次最终翻转）成为该翻转序列中的最后一次翻转。由于 e 是初始树 T_I 的一条边，且其轨迹长度现在为 1，因此我们可以从序列中移除翻转 $e\to g$，而不改变定理 4 的条件是否成立。因此，省略翻转 $e\to g$ 并相应地将目标树调整为 T_k+e-g 后，我们得到的翻转序列比初始序列短一次翻转。回想一下，我们假设一条被翻转多次、但未被任何其他边交叉的边，是在序列 $N(T_0)$、 $N(T_1)\ldots,N(T_{k-1})$、 $N(T_k)$ 中引入的。因此，我们新的、更短的翻转序列也包含该边。这与我们的假设矛盾：我们假设最初的翻转序列是在所有不满足该定理陈述的序列中，翻转次数最少的序列。◀

我们继续说明，定理4足以解决一般情形下相容翻转序列的重新停车性质，以及 T_I 的凸包边问题。

▶ **定理 5**。 设点集 T_I 中的任意两棵平面生成树 T_F 和 P。**1.** 存在一个最短翻转序列，使得任意凸包边至多被翻转一次 .
2. 在相容情形下，所有边至多被翻转两次 （相容重新停放性质） .

证明。 第一部分由凸包边从不与其他边交叉这一事实以及定理 4 得出。

对于第二部分，我们应用相容翻转满足停放性质这一点 [6,引理 20]。由于存在一个最短翻转序列，仅在凸包上引入停车边，因此我们可以将本定理的第（1）部分应用于凸包上的停车边。于是，每条边至多被翻转两次：一次是停放到凸包上，另一次是进行最终翻转。◀

▶ **推论 9.** 对于任意一对树 T_I、 T_F，都存在一个从 T_I到 T_F 的最短翻转序列，其中每次移除对角线 e 的翻转都会添加以下三者之一：
- 通过最终翻转添加的 T_F 的一条边。凸包上的一条边。一条与 e交叉的停车边。此外，任何移除凸包边的翻转都是最终翻转。

证明。 唯一剩下的情况是，对角线 e 被一条无交叉对角线替换。根据引理 8，存在另一种翻转序列，可用一条凸包边替换 e。◀

4 一般翻转的反例

在上一节中，我们确定了任何针对一般翻转的停车猜想和重新停车猜想反例都必须满足的结构条件。现在，有了这些条件，我们构造特定的树对 T_I 和 T_F。它们不仅满足这些要求，还更进一步，产生了违反这些猜想的最短翻转序列。

通过分类讨论、对翻转序列的进一步观察以及计算机程序的支持，我们验证了所提出的翻转序列确实是可能的最短翻转序列。此外，我们证明了不存在其他满足这些猜想的最短翻转序列。

4.1 停车边性质

在本节中，我们证明定理 6，从而推翻猜想 2。该定理由两部分组成，为方便起见，我们分别陈述并证明这两部分。

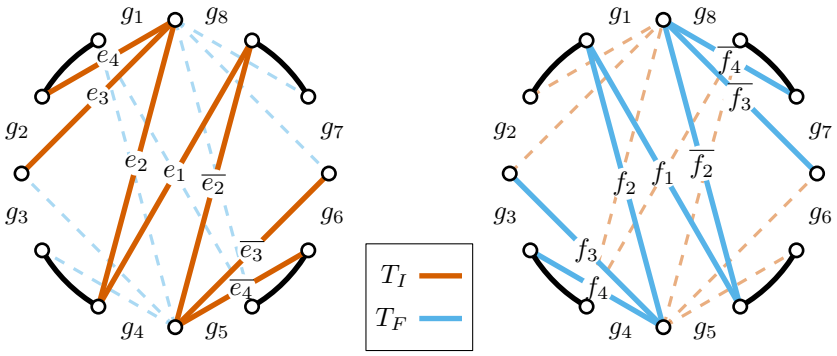


图 5 一个使用 $n = 12$ 个点的停车边猜想反例。

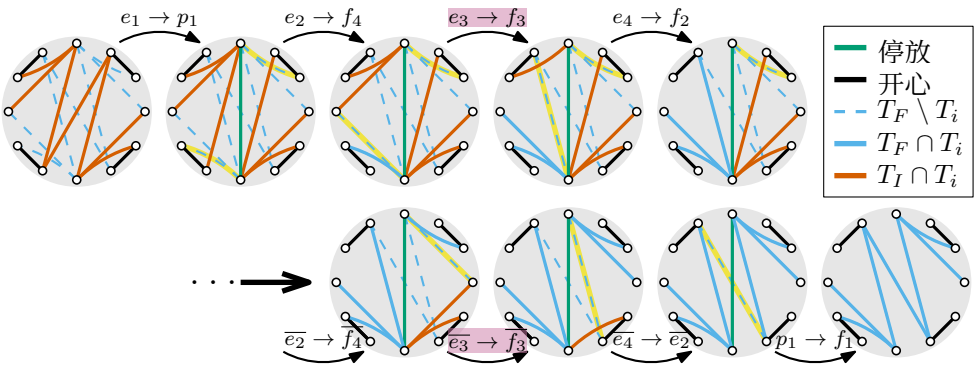
▶ **引理 10** ($\star$). 对于一般翻转以及处于凸位置的 $n \geq 12$ 个点，停车边猜想并不成立。此外，最小反例不一定仅涉及不相容翻转，甚至可能在所有最短翻转序列中都包含边滑动。

证明略述。 考虑图 5 中展示的一对树。我们的目标是将 T_I变换为 T_F。在此过程中，我们将证明，所有最短翻转序列都必然会将一条边停放在对角线上。图 6(a) 展示了一条使用对角停车边 p_1 （绿色）的翻转序列。此外，该序列恰好需要一次停车翻转 $e_1 \to p_1$，而其余所有翻转都是最终翻转。进一步地，由于对称差 $T_I \Delta T_F$ 中的每条边至少被交叉两次，因此，从 T_I 到 T_F 的任何序列，其第一次翻转都必然是停车翻转。因此，图 6(a) 所示的序列确实是最短翻转序列。值得注意的是，这条翻转序列还包含两次边滑动，以紫色标出。

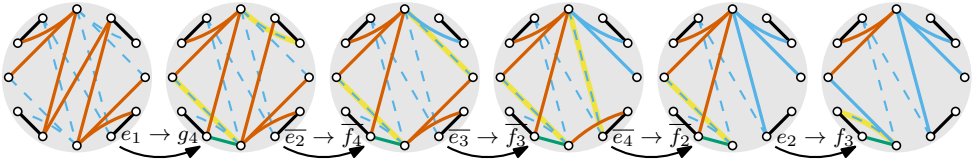
还需证明，任何将边停放到凸包而非 p_1上的停车翻转，至少需要额外进行一次翻转。为此，我们证明：任何以将边停放到凸包上的停车翻转开头的翻转序列，此后若只进行最终翻转，都无法到达 T_F 。图6(b)–6(e)中，我们展示了所有可能的首个翻转（忽略对称情形）：这些翻转会将 $T_I \setminus T_F$ 的一条对角线替换为凸包上的一条边。对于每种此类翻转，图6都展示了一条翻转序列：持续进行完美翻转，直到无法在不再进行停车翻转的情况下添加最终树的任何其他边。需要强调的是，这些序列并非唯一，因为有时可以将多条边翻转为同一条边，也可以按不同顺序进行翻转而得到相同结果。例如，图 7展示了所有以 $e_2 \to g_8$ 开头、其余步骤均为最终翻转的可能序列所构成的图。对于将对角线停放到凸包上的其他情形，同样的推理也适用，因为每种情形都可以用类似的方法进行分析。◀

为了计算图 5 所示示例中的所有相关翻转序列，我们采用了一个简单直接的计算机程序 [5], ，该程序系统地生成并跟踪所有可能的序列。程序以初始树 T_I 的边初始化一个边集，然后以类似深度优先搜索（DFS）的方式运行。对于给定的每个边集，它会考虑所有可能的合法翻转（即翻转后边集仍构成一棵生成树）。对于每次此类翻转，程序执行该翻转；如果这次翻转不是完美翻转，也就是说没有插入最终树的一条边，则将计数器加一。只有当该计数器不

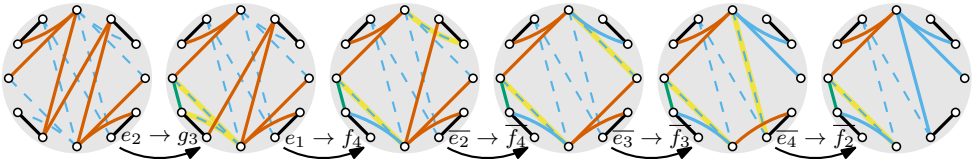
12 平面生成树之间最短翻转序列的结构性质



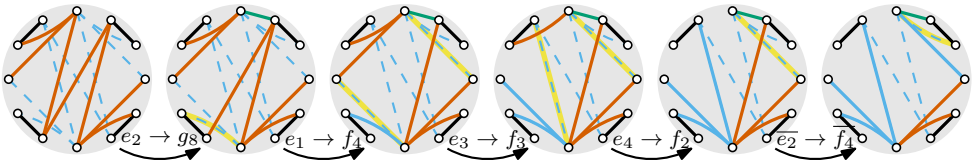
(a) 图 5 所示实例的一个最短翻转序列。两个高亮的翻转是滑移。



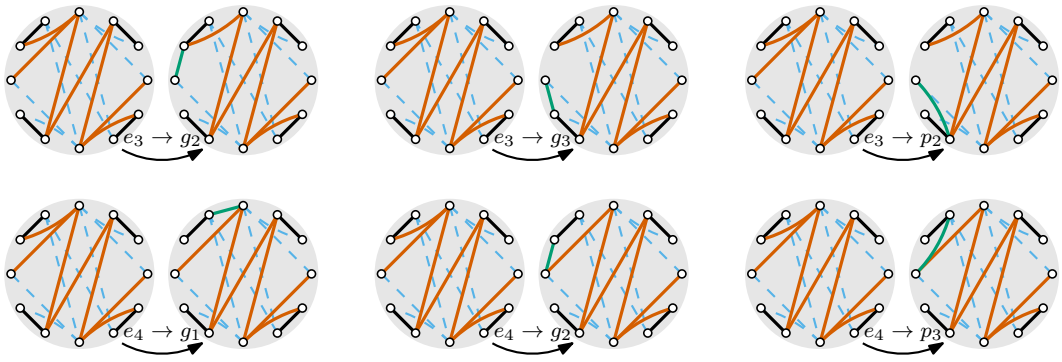
(b) 一个以 $e_1 \to g_4$ 开始的“极大”翻转序列。注意， $e_1 \to g_8$ 与之对称。



(c) 一个以 $e_2 \to g_3$ 开始的“极大”翻转序列。注意， $e_2 \to g_7$ 与之对称。



(d) 以 $e_2 \to g_8$ 开始的“极大”翻转序列。注意， $e_2 \to g_4$ 是对称的。



(e) 停放 e_3 或 e_4 后，经过一次翻转便会到达死端。注意， e_3 和 e_4 的翻转是对称的 .

图6 我们展示了一个最短翻转序列 (a)，其中边1以对角线方式停放；同时还展示了在第一次翻转中选择另一条边对进行停放时，可能实现的最大数量的完美翻转。为清晰起见，突出显示了当前可以插入的 T_F 的边。

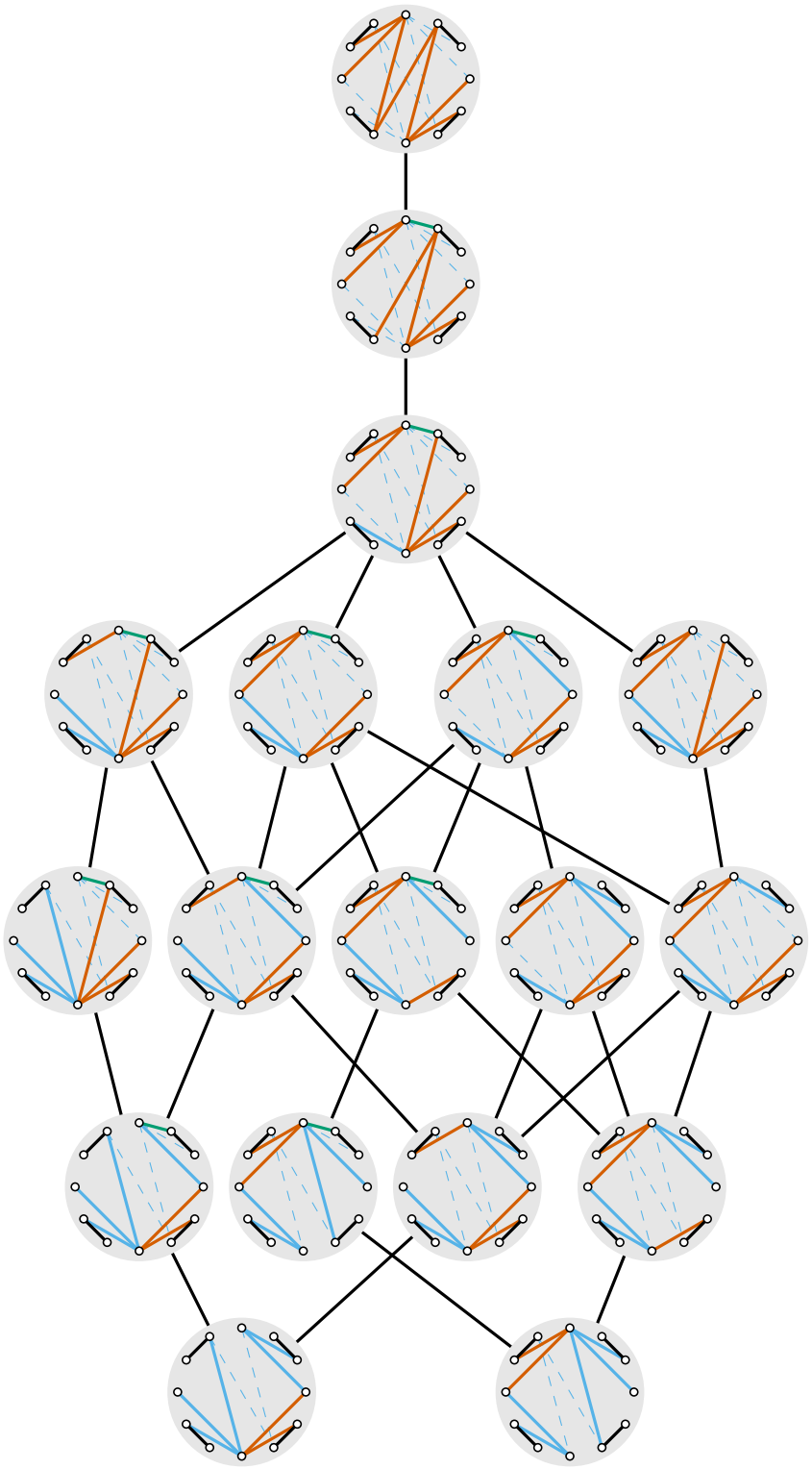


图 7 未进行第二次停车翻转时， $e_2 \to g_8$ 之后可达树构成的子图。

若不超过允许的非完美翻转最大次数（在我们的情况下为一次），程序将继续递归。否则，将撤销该翻转并执行下一个合法翻转，依此类推。每当该集合中的所有边都来自最终树 T_F 时，最短翻转序列的数量计数器就会加一。深度优先搜索完成后，程序终止。

对于图 5 所示的树，程序输出了 20 个长度为 8 的不同翻转序列。通过统计图 12 中描绘的所有最短翻转序列，可以验证这些确实正好是包含停车翻转 $e_1 \to p_1$ 的全部序列。注意，在执行翻转 $e_1 \to p_1$ 后，存在某种左右对称性。接下来必须依次执行翻转 $e_2 \to f_4$、$e_3 \to f_3$和 $e_4 \to f_2$，但它们可以与对称翻转 $e_2 \to f_4$、$e_3 \to f_3$和 $e_4 \to f_2$ 以任意顺序交错进行。由于只有此处的顺序具有灵活性，因此我们得到 $\binom{6}{3} = 20$ 个长度为 8 的不同翻转序列，这正是程序报告的数量。

我们使用程序的一个修改版本生成了相关的翻转图，即所有树，以及从初始树 T_I 出发、通过至多包含一次非完美翻转的翻转序列可以到达的树之间的所有翻转。该图以 118 棵不同的树为节点，它们之间有 214 条边（翻转）。完整的图见附录 C 中的图 12。

通过强制第一次（非完美）翻转必须是翻到凸包上的翻转，我们进一步减少了这些数量。这是可行的，因为为了验证我们陈述的正确性，只需考虑不包含向对角线进行停车翻转的最短翻转序列（回顾一下，第一次翻转必须是停车翻转）。这样得到的图只有 67 棵树和 120 条边，能够清晰地展示所有相关情形，包括引理 10 的证明中详细阐述的情形。两种版本的翻转图均可进行交互式探索 [5]。

为方便读者，图的所有相关组成部分都可在附录 C 的图 12 附近找到；每一种可能的停车翻转都会对应于子图中一个唯一的（除对称情况外）部分，在该部分中无需进行第二次非完美翻转即可继续到达。这些部分见图 7 和图 13–15。

如图 5 所示，将翻转限制在凸包上的停车边，会额外增加一次翻转。接下来，我们将这一结果加以强化，证明在相同限制下，存在需要额外进行 $\Omega(n)$ 次翻转的实例。

▶ **引理 11**。 对于任意 $k \in \mathbb{Z}^+$，都存在两棵定义在 $n \geq 10k + 2$ 个点上的无交叉生成树，使得最短翻转序列需要 $8k$ 次翻转，而仅在凸包边上停放的最短翻转序列至少需要 $9k$ 次翻转。

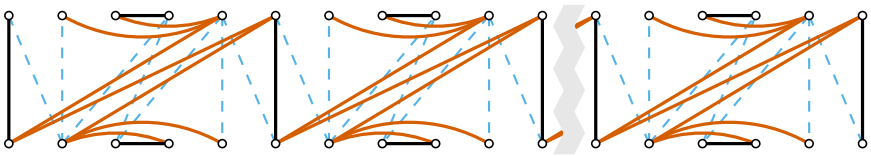


图 8 一个使用 $n = 10k + 2$ 个点构造的停车边猜想反例。

证明。 我们将 k 个树实例 T_I 如图 5 所示，沿幸福边粘合起来。我们的目标结构由 k 个 T_F 实例沿相同的幸福边粘合而成。如果翻转序列中没有幸福边被翻转，这就等同于在每个 k 单独的 T_I 实例中分别使用 个独立的翻转序列；与仅使用凸包上的停车边相比，我们需要在最优翻转序列中进行 次翻转，而不是 次。对所有 T_I 实例求和，可得长度为 $8k$ 的最优翻转序列；而仅使用凸包上的停车边时，最小翻转距离为 $9k$。

接下来，假设对于一个最小反例，所有 $k-1$ 条连续的幸福边都在翻转序列中的某个时刻被移除，并在之后的某次翻转中重新加入。我们论证，除非引入不在凸包上的停车边，否则该翻转序列至少需要 $10k-2$ 次翻转。反设不然。此时，翻转序列会移除并加入幸福边；由于该翻转序列不会引入对角停车边，因此在整个翻转序列中，没有任何边会与幸福边交叉。 [1]中的命题 18 指出，在任何最短翻转序列中，对于每一条被移除的幸福边，之后必定存在某条会与这条幸福边交叉的边。如果不存在这样的交叉边，该命题的证明给出了一种构造翻转序列的方法，使序列至少缩短一次翻转，且不会移除这条幸福边。应用这一结果，我们可以逐步构造翻转序列，使其中不再移除并重新加入某一条幸福边，从而使所得翻转序列比原序列短一次翻转。对所有幸福边逐一应用这一结果，可得到长度至多为 $9k-1$ 的翻转序列，其中没有任何幸福边被翻转，并且仍未使用凸包外的停车边。根据鸽巢原理，必定存在一个 T_I 实例，仅用 8 次翻转且不引入对角停车边，就能将其翻转为一个 T_F 实例，这与引理 10 矛盾。◀

从积极的一面来看，我们表明仅在凸包上停放的翻转序列仍然接近一条最优翻转序列。更精确地说，我们证明一次对角停靠翻转可以用两次凸包停靠翻转来替换。对每一次对角停靠翻转应用一次下述命题，可得到一个不含任何停车翻转的翻转序列，但代价是每次迭代增加一次翻转。

▶**命题 12.** 一个通过非相容翻转添加并移除停靠对角线 d 的翻转序列，可以替换为一个不执行涉及 d 的对角停靠翻转、且仅长一次翻转的翻转序列。特别地，所得翻转序列执行的对角停靠翻转严格少于原始翻转序列。

证明。不失一般性，假设在从 $T_I = T_0$ 到 T_1 的翻转中，边 e 经由一次非相容翻转变为停车边 $f = (v_a, v_b)$，并在从 T_{k-1}到 $T_k = T_F$ f 的翻转中由一次非相容翻转移除，并替换为边 e'。于是， f将点集分成两侧 A 与 B，即从 v_a 到 v_b按顺时针顺序出现的点，以及按逆时针顺序出现的点。我们将 v_a 与 v_b 同时纳入这两个集合。在从 T_1 到 T_{k-1} 的翻转序列中，交换 A 或 B 中边的翻转可以独立于另一侧执行。若从 T_1中移除边 f，我们得到两棵子树，并且在 A中两个连续的凸包顶点 v_i、 v_{i+1} 之间存在一条连接这两棵子树的凸包边。不失一般性， v_i 与 v_a位于同一棵子树中。

我们从 T_0 翻转到 $(T_1 \setminus \{f\}) \cup \{h\}$。接着，翻转序列中从 T_1 到 T_{k-1}且发生在侧边 B 上的任何翻转仍然可以执行。任何添加并移除了一条边、且该边界定了一个包含 f 的环的翻转，现在都包含一条路径：先从 v_a 走到 v_i，到达 A中的子树，然后穿过 h，最后从 v_{i+1} 走到 v_b。

在 B 中的所有翻转都执行完毕，且侧面 B 与 T_{k-1}中的侧面 B 重合后，我们执行一次翻转，移除 h，并添加凸包边 h'，将 B中的两个子树连接起来。接着，我们继续执行侧面 A中的所有翻转。同样，任何涉及包含 f 的环的翻转，此时都包含一条经过侧面 B 和凸包边 h'的路径。执行完 A中的所有翻转后，我们得到树 $T_{k-1} \setminus \{f\} \cup \{h\}$。最后一次翻转中，我们移除 h并添加 e'，得到树 T_k。我们完成了原翻转序列中发生的所有翻转。只有边 e 的轨迹从 (e, f, e') 变为 (e, h, h', e')。因此，所得翻转序列恰好比原翻转序列多一次翻转。◀

我们注意到，由命题 12 的证明得到的翻转序列并不“理想”，因为其中包含一次从一条凸包边到另一条凸包边的非最终翻转。利用定理 4，我们可以重新排列某些翻转，从而得到一个“更理想”的翻转序列。

4.2 重新停放性质

由定理4和引理8，我们推导出重新停车性质的反例所必须满足的必要条件如下：

▶**命题 13.** 假设存在一条最短翻转序列，其中有一条边 e，且 $trace(e) = (e, e_1, e_2, e_3)$ 的长度为 3。若不存在另一条最短翻转序列使得 e 具有更短的轨迹，则必须满足以下条件：

（1） e与 e_3 是对角线。
（2） e与 e_1 相交。（3） e_1 与 e_2 相交。（4） e_2 与 e_3 相交。

证明。 我们逐一讨论这些不同的条件。
(1) 如果 e 是凸包边，那么它尤其不会与任何其他边相交。由定理 4 可知，存在一个最短翻转序列，可以将 e 翻转到其最终位置。如果 e_3 是凸包边，则我们对反向的翻转序列应用定理 4。

(2) 如果 e 和 e_1 不交叉，那么 e_1 可通过一次相容翻转加入。应用引理 8，可以得到一个最短翻转序列，其中 e_1 被替换为一条凸包边。第二步，应用定理 4，得到一个最短翻转序列，其中 e_1 被翻转到其最终位置。因此，我们构造出了一个最短翻转序列，其中 e的轨迹长度为 2。

(3) 如果 e_1 和 e_2 不交叉，我们首先应用引理 8，得到一个最短翻转序列，其中 e_2 被替换为一条凸包边 h_2。现在可以将该翻转序列反向。在新的翻转序列中，e_3 被翻转为一条凸包边。然后，我们应用定理 4，得到一个最短翻转序列，其中 h_2 在反向翻转序列中被翻转到其最终位置，也就是 e。再次将该翻转序列反向后，我们得到一个最短翻转序列，其中 $trace(e) = (e, h_2, e_3)$；尤其是，e的轨迹长度为 2。

(4) 如果 e_2 和 e_3 不相交，我们就反转翻转序列，并按照情况 (2) 进行。 ◀

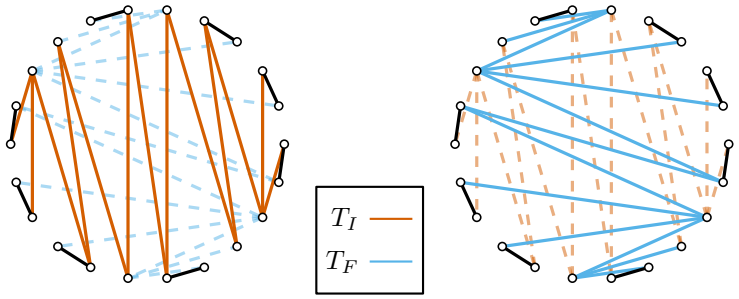


图 9 重新停车性质的一个反例。

图 9 给出了重新停车性质的一个反例。构造此反例时考虑了命题 13 的条件。从 T_I 红色到 T_F 蓝色的任何最短翻转序列都包含一条轨迹长度为 3 的边。图 10 展示了最短翻转序列的示意图。首先，将红色的中央对角线翻转为绿色

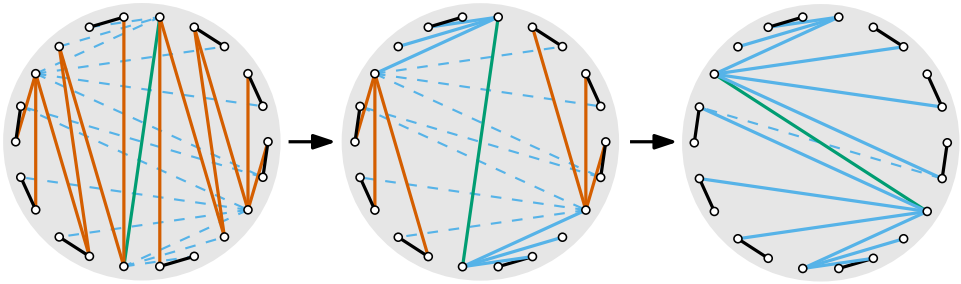


图10 一个关于重新停车性质的反例。

对角线。然后，我们可以进行六次完美翻转，插入来自 T_F的边。之后，再次翻转绿色对角线。我们还可以进行六次额外的完美翻转，之后便可将绿色边翻转到其最终位置。

图10中示意的翻转序列是连续两次涉及绿色边的翻转之间所发生的六次完美翻转的唯一最短翻转序列（忽略顺序）。为验证这一点，我们再次使用了程序 [5]。此处相关的翻转图部分包含 5086 个顶点（这些顶点距离 T_I 至多两次非完美翻转）。因此，这个图的规模似乎相当难以人工检查。正如我们在引理 10 的证明中已经看到的，手工进行分类讨论会过于繁琐，因此这里我们依赖程序。程序共生成了 400 个最短翻转序列，恰好为 $\binom{6}{3}\cdot\binom{6}{3}$，也就是说，这是停车翻转之间的完美翻转可能交错的不同顺序数。

我们甚至将这一构造又向前推进了一步，给出一对树，使得对于每一条最短翻转序列，都存在一条轨迹长度为四的边，也就是说，这条边被重新停放了两次。该例子如图11所示。红色边属于 T_I，蓝色边属于 T_F，而凸包上的黑色边同时属于这两棵树。我们使用与前两个反例相同的程序验证了这一例子。

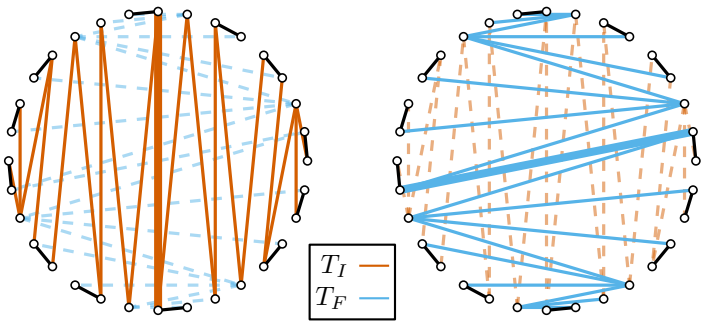


图11 一个包含32个顶点的例子，其中每一条最短翻转序列都包含一条轨迹长度为四的边（以粗边突出显示）。

5 结论与未来工作

我们在一般情形下证明了最短翻转序列不满足停放性质和重新停车性质，并证明当翻转仅限于相容翻转时，重新停车性质成立。这些结果阐明了现有假设的局限，并指出了若干值得探索的未来研究方向。

18 平面生成树之间最短翻转序列的结构性质

- 我们在表 1 中总结了近期关于直径的所有上界，并指出每个上界满足所研究的哪些性质。最值得注意的是，在所有情形下，所有边的轨迹长度上界均为 2。这引出了两个问题：若要进一步改进翻转图直径的上界，是否需要纳入我们的构造？这些反例是否能为构造更紧的下界带来新的见解？通过否定一般翻转下的停车猜想，我们排除了一种此前颇有希望的途径——以此证明快乐边猜想。因此，我们认为，要在该猜想上取得进一步进展，
- 需要全新的方法。我们构造了初始树和最终树，使得对于这些树对，任意最短翻转序列中都有一条边的轨迹长度分别为 3 或 4。我们预计这一构造可以无限次迭代。因此，我们提出以下猜想：
-

▶**猜想 14.** 对于任意常数 c，都存在一对树 T_I 和 T_F，使得 T_I 和 T_F 之间的任意最短翻转序列都包含一条轨迹长度大于 c的边。

- 最近，Bjerkevik、Dorfer、Kleist、Ueckerdt 和 Vogtenhuber 证明了，计算凸点集的平面生成树之间的最短（相容）翻转序列是 NP完全问题 [13] 。这凸显了寻找有效的固定参数可处理算法或近似算法的重要性。能否利用相容翻转的重新停车性质，改进目前已知的 [6] 的固定参数可处理算法？此外，我们关于停放性质和重新停车性质的反例，或许有助于确立更高层次的难解性结果。

参考文献

1 Oswin Aichholzer、Brad Ballinger、Therese Biedl、Mirela Damian、Erik D. Demaine、Matias Korman、Anna Lubiw、Jayson Lynch、Josef Tkadlec 和 Yushi Uno。无交叉生成树的重构。《计算几何期刊》，15(1):224–253，2024。 doi:10.20382/jocg.v15i1a9。2 Oswin Aichholzer、Sofia Brenner、Joseph Dorfer、Hung P. Hoang、Daniel Perz、Christian Rieck 和 Francesco Verciani。几何与组合情形下奇匹配的翻转。载于图绘制与网络可视化研讨会（GD），第12:1–12:18页，2025。doi:10.4230/LIPIcs.GD.2025.12。3 Oswin Aichholzer、Anna Brötzner、Daniel Perz 和 Patrick Schnider。奇匹配中的翻转。《计算几何》，129:102184，2025。 doi:10.1016/J.COMGEO.2025.102184。4 Oswin Aichholzer 和 Joseph Dorfer。求解凸点集上平面生成路径之间最短翻转序列的线性时间算法。几何与拓扑计算，2025。 doi:10.48550/arxiv.2507.02740。5 Oswin Aichholzer、Joseph Dorfer、Peter Kramer、Christian Rieck 和 Birgit Vogtenhuber。https://www.peterkramer.eu/parking/，2026。补充材料。6 Oswin Aichholzer、Joseph Dorfer 和 Birgit Vogtenhuber。平面生成树中的受约束翻转。载于图绘制与网络可视化研讨会（GD），第5:1–5:18页，2025。doi:10.4230/LIPIcs.GD.2025.5。7 Oswin Aichholzer、Kristin Knorr、Wolfgang Mulzer、Johannes Obenaus、Rosna Paul 和 Birgit Vogtenhuber。平面生成路径的翻转。载于算法与计算国际会议及研讨会（WALCOM），第49–60页，2023。 doi:10.1007/978-3-031-27051-2 5。_8 Oswin Aichholzer、Wolfgang Mulzer 和 Alexander Pilz。简单多边形三角剖分之间的翻转距离是NP完全的。《离散与计算几何》，54(2):368–389，2015。doi:10.1007/S00454-015-9709-7。

9 Oswin Aichholzer 和 Klaus Reinhardt。无交叉生成树间滑动的二次距离界。《计算几何》，37(3):155–161，2007。 doi:10.1016/J.COMGEO.2004.12.010。10 Selim G. Akl、Md Kamrul Islam 和 Henk Meijer。论平面路径变换。《信息处理快报》，104(2):59–64，2007。 doi:10.1016/j.ipl.2007.05.009。11 David Avis 和 Komei Fukuda。用于枚举的反向搜索。《离散应用数学》，65(1):21–46，1996。 doi:10.1016/0166-218X(95)00026-N。12 Carla Binucci、Fabrizio Montecchiani、Daniel Perz 和 Alessandra Tappini。匹配翻转是困难问题，2025。 doi:10.48550/arXiv.2503.02842。13 Håvard Bakke Bjerkevik、Joseph Dorfer、Linda Kleist、Torsten Ueckerdt 和 Birgit Vogtenhuber。无交叉生成树的翻转距离：NP难性与改进的界。载于计算几何研讨会（SoCG），2026。即将发表。14 Håvard Bakke Bjerkevik、Linda Kleist、Torsten Ueckerdt 和 Birgit Vogtenhuber。无交叉生成树的翻转。载于离散算法研讨会（SODA），第2313–2325页，2025。 doi:10.1137/1.9781611978322.77。15 Miguel Bosch-Calvo 和 Steven Kelk。简单凸多边形翻转距离问题的改进核。《信息处理快报》，182:106381，2023。 doi:10.1016/J.IPL.2023.106381。16 Prosenjit Bose 和 Ferran Hurtado。平面图中的翻转。《计算几何：理论与应用》，42(1):60–80，2009。 doi:10.1016/j.comgeo.2008.04.001。17 Nicolas Bousquet、Lucas De Meyer、Théo Pierron 和 Alexandra Wesolek。凸几何图中平面树的重构。《离散与计算几何》，第1–34页，2025。 doi:10.1007/s00454-025-00785-8。18 Nicolas Bousquet、Valentin Gledel、Jonathan Narboni 和 Théo Pierron。关于无交叉生成树之间翻转距离的注记。几何与拓扑计算，2023。 doi:10.57717/cgt.v2i1.36。19 Sean Cleary。二叉树之间的受限旋转距离。《信息处理快报》，84(6):333–338，2002。 doi:10.1016/S0020-0190(02)00315-0。20 Joseph Dorfer。凸多边形三角剖分的翻转距离／二叉树的旋转距离是NP完全的。2026。 doi:10.48550/arXiv.2602.22874。21 David Eppstein。翻转图的圆满结局。《计算几何期刊》，1(1):3–28，2010。 doi:10.20382/JOCG.V1I1A2。22 Stefan Felsner、Linda Kleist、Torsten Mütze 和 Leon Sering。翻转图中的彩虹圈。载于计算几何研讨会（SoCG），第38:1–38:14页，2018。 doi:10.4230/LIPICS.SOCG.2018.38。23 Stefan Felsner、Linda Kleist、Torsten Mütze 和 Leon Sering。翻转图中的彩虹圈。SIAM离散数学杂志，34(1):1–39，2020。 doi:10.1137/18M1216456。24 M. Carmen Hernando、Ferran Hurtado、Alberto Márquez、Mercè Mora 和 Marc Noy。凸位置点集的几何树图。《离散应用数学》，93(1):51–66，1999。 doi:10.1016/S0166-218X(99)00006-2。25 M. Carmen Hernando、Ferran Hurtado 和 Marc Noy。无交叉完美匹配图。《图与组合》，18(3):517–532，2002。 doi:10.1007/S003730200038。26 Michael E. Houle、Ferran Hurtado、Marc Noy 和 Eduardo Rivera-Campo。三角剖分图与完美匹配图。《图与组合》，21(3):325–331，2005。 doi:10.1007/S00373-005-0615-2。27 Ferran Hurtado、Marc Noy 和 Jorge Urrutia。三角剖分中的边翻转。载于计算几何研讨会（SoCG），第214–223页，1996。 doi:10.1145/237218.237367。28 Iyad Kanj、Eric Sedgwick 和 Ge Xia。计算三角剖分之间的翻转距离。《离散与计算几何》，58(2):313–344，2017。 doi:10.1007/S00454-017-9867-X。29 Linda Kleist、Peter Kramer 和 Christian Rieck。论平面生成路径翻转图的连通性。载于计算机科学中的图论概念国际研讨会（WG），第327–342页，2024。 doi:10.1007/978-3-031-75409-8_23。

20 平面生成树之间最短翻转序列的结构性质

30 Haohong Li 和 Ge Xia。一个用于计算凸翻转距离的 O(3.82^k) 时间固定参数可解算法。《离散与计算几何》，73(3):702–718，2025。 doi:10.1007/s00454-023-00596-9. 31 Anna Lubiw 和 Vinayak Pathak。点集的两个三角剖分之间的翻转距离是 NP-完全的。《计算几何》，49:17–23，2015。 doi:10.1016/J.COMGEO.2014.11.001. 32 Joan M. Lucas。二叉树旋转距离的改进核大小。《信息处理快报》，110(12-13):481–484，2010。 doi:10.1016/J.IPL.2010.04.022. 33 Alexander Pilz。平面点集三角剖分之间的翻转距离是 APX困难。《计算几何》，47(5):589–604，2014。 doi:10.1016/J.COMGEO.2014.01.001. 34 Lionel Pournin。结合多面体的直径。《数学进展》，259:13–42，2014。doi:10.1016/j.aim.2014.02.035. 35 Eduardo Rivera-Campo 和 Virginia Urrutia-Galicia。凸位置点集的路径图中的哈密顿回路。《计算几何：理论与应用》，18(2):65–72，2001。 doi:10.1016/S0925-7721(00)00026-2. 36 Daniel Dominic Sleator、Robert Endre Tarjan 和 William P. Thurston。旋转距离、三角剖分与双曲几何。载于计算理论研讨会（STOC），第 122–135 页，1986。 doi:10.1145/12130.12143.

第 3 节省略的细节

▶**引理 8** ($\star$)。 设有一条连接两棵树 T_I 和 T_F 的翻转序列，以及一条边 e，使得 $trace(e) = (e, f, e')$，其中 f是一条对角线，且翻转 $e \to f$或翻转 $f \to e'$ 与之兼容。那么存在一条从 T_I 到 T_F 、长度相同的翻转序列，其中 $trace(e) = (e, h, e')$，使得 h是凸包边。任何其他边的轨迹保持不变。

证明。 不失一般性，设 $e \to f$ 是翻转序列中的第一个翻转，而 $f \to e'$ 是最后一个翻转。进一步设 $e \to f$ 是一个相容翻转，否则考察逆序的翻转序列。我们用 v_a 和 v_b 表示 f的两个顶点。

我们根据以下观察对其他翻转重新排序并进行修改。边 f将点集分为两侧： A 和 B，即沿顺时针方向从 v_a 到 v_b 出现的点，以及按逆时针顺序出现的点，其中 v_a 和 v_b 同时包含在 A 和 B中。当 f 属于树时，两侧的翻转可以独立于另一侧执行。注意，任意生成树都包含任意两个顶点之间的唯一路径。特别地，若我们考虑 $T_I = T_0$中从 v_a 到 v_b 的路径 p，可以看出，该路径必须完全包含在由 f诱导的一侧中；不失一般性地，设其位于 B一侧。否则，路径上将存在一条穿过 f的边，从而引入一个从 T_I 到 T_1的非相容翻转。

如果我们添加边 f，所闭合的环完全包含在 B 中。因此， e 位于 B 中。从 T_1 到 T_{k-1} 执行的、发生在 A 中的任何翻转，都可以直接在 T_I中执行。原先包含边 f 的任何环，现在都改为包含路径 p 。我们对翻转进行排序，使得所有发生在 A 中的翻转都从 T_I 开始立即执行。完成所有这些步骤后， A这一侧在 T_F中已与 A这一侧重合，唯独边 f除外。

让我们仔细看看 T_F 中位于 A的部分。这部分由 f 以及 A中的两棵树 T 和 T' 组成（它们的边集可能为空）。由于 T_F 是平面的，凸包边界上必定存在唯一一对相邻的点，其中一个属于 T，另一个属于 T'。我们通过添加这两点之间的凸包边 h，并移除添加 f 时本应移除的那条边，来继续修改后的翻转序列。

现在，我们执行原始翻转序列中 B 的所有翻转。任何涉及 f 并形成环的翻转，现在都会形成一个涉及 h 以及 A 中连接 v_a 和 v_b到 h的路径的环。

最后，我们执行 $h \to e'$ 这一翻转，得到最终树 T_F。 e 的轨迹为（e, h, e'），而 T_I 中的任何其他边的轨迹在这两个翻转序列之间都没有变化。◀

B 第 4 节中省略的细节

▶ **引理 10** ($\star$)。 *</style}*对于凸位置中的 $n \geq 12$个点，停车边猜想对一般翻转并不成立。此外，最小反例不一定局限于不相容翻转，甚至可能在所有最短翻转序列中都包含边滑动。

证明。 考虑图 5 中所示的这对树 T_I， T_F 。在图 6(a) 中，使用对角停车边 p_1 （绿色）的序列恰好需要一次停车翻转， $e_1 \to p_1$，而其余所有翻转都是最终翻转。此外，注意从 T_I 到 T_F 的任意序列，其第一次翻转都必须是停车翻转，因为对称差 $T_I \Delta T_F$ 中的每条边至少被交叉两次。因此，图 6(a) 所示的序列是最短翻转序列。值得注意的是，该翻转序列还包含两次边滑动，并以紫色标出。

22 平面生成树之间最短翻转序列的结构性质

要证明该结论，还需说明，任何将边停放到凸包而非 p_1 的停车翻转，都至少需要多进行一次翻转。为此，我们证明，以将边停放到凸包为起始操作的翻转序列，在此停车翻转之后若只进行最终翻转，就无法到达 T_F。在图 6(b)–6(e) 中，我们列出了所有可能的第一次翻转：移除 $T_I \setminus T_F$ 的一条对角线，并改为添加凸包上的一条边。对于每种情况，我们都展示了一个翻转序列：尽可能长时间地进行完美翻转，直到不再进行另一次停车翻转就无法添加 T_F 的其他边。我们将此类翻转序列称为极大 翻转序列。

图 6 中的翻转序列并未涵盖所有可能出现的情形，因为有时可能有多条边可以翻转为同一条边，或者可以按不同顺序执行多次翻转而得到相同结果。对于第一次翻转为 $e_2 \to g_8$ a 的情形，图 7 展示了描绘所有可能的极大翻转序列的图。对于其他所有将对角线停放在凸包上的情形，也可以用类似的方法论证。由于第一个情形已经凸显出由此产生的分类讨论有多么繁琐，我们将采用两种不同的方法来涵盖所有情形。分类讨论中的所有论证都归结为三个基本论点：(1) 如果 T_F 的一条边被交叉至少两次，那么就不能通过单次翻转将其加入。(2) 如果 T_F 的一条边恰好被交叉一次，那么唯一有可能加入该边的翻转，就是移除与其交叉的那条边的翻转。(3) 如果加入 T_F 的边会在图中形成一个环，我们就不能加入该边。在所有情形中，该环都会包含凸包上的一条停车边。

证明之后，我们将讨论如何通过一个简单的程序 [5] 检查所有情况，并给出一个有向无环图 [5]，其中嵌入了所有可通过 8 次翻转得到的翻转序列，且其中至多一次翻转可以是停车翻转。

回到以 $e_2 \to g_8$ 开始的翻转序列。上方展示了初始树，随后是经过翻转 $e_2 \to g_8$ 得到的树。只有两条边 T_F 没有被交叉两次，即 f_4 和 f_3；它们都被边 e_2 交叉。翻转 $e_1 \to f_3$ 会使树断开，因此唯一有效的翻转是 $e_1 \to f_4$。

对于下一次翻转，可以插入图 T_F 的多条边，因为 f_3 完全没有被交叉，而边 f_4 和 f_3 只被 e_2 交叉。要插入 f_3，可以进行三种翻转：$e_3 \to f_3$、$e_2 \to f_3$ 和 $g_8 \to f_3$。要插入 f_3，可以执行翻转 $e_2 \to f_3$。注意，没有任何翻转可以添加边 f_4，因为 f_4 被 e_2 交叉，而翻转 $e_2 \to f_4$ 会形成一个环。

在第四行的左侧树中，T_F 有三条边被 T_I的一条边交叉。更准确地说，只有 e_4 交叉 f_2，只有 e_2交叉 f_3。同样，只有 e_2交叉 f_4，但直接翻转 $e_2 \to f_4$ 会形成一个环。我们可以按任意顺序执行翻转 $e_4 \to f_2$ 和 $e_2 \to f_3$。翻转 $e_2 \to f_3$之后，翻转 $g_8 \to f_4$ 也变得可行。这样一来，执行翻转 $e_4 \to f_2$、$e_2 \to f_3$和 $g_8 \to f_4$共有三种不同的翻转序列，并且最后一次翻转必须在中间那次翻转之后进行。所有序列最终都得到相同的情形：T_F 的两条边分别被 T_I的两条边交叉两次。

在第四行的第二棵树中，我们目前已添加边 f_4、f_3 和 g_8。T_F 中唯一没有被交叉两次的边是 f_4 和 f_3。只有通过翻转 $g_8 \to f_4$ 才能添加 f_4，而不会形成环。通过翻转 $e_3 \to f_3$ 和 e_3，可以将 f_3 添加到 f_3。

翻转 $g_8 \to f_4$ 后，我们可以继续依次翻转 $e_3 \to f_3$ 和 $e_4 \to f_2$，也可以翻转 $e_3 \to f_3$ 和 $e_4 \to f_3$。这两种情况下，最终都有 T_F 的两条边与 T_I 的两条边相交。若不再进行一次停车翻转，就无法继续添加 T_F 的边。

翻转 $e_3 \to f_3$ 后，还剩下两种可能的翻转，即 $g_8 \to f_4$ 和 $e_4 \to f_3$。这两次翻转无论按何种顺序进行，都能得到有效的翻转序列。两种顺序下，接下来都会出现这样的情况： T_F 中的 f_4 和 f_1 均被 T_I 中的 e_3 和 e_4 穿过。若不再进行停车翻转，就无法继续添加 T_F 的边。翻转 $e_3 \to f_3$ 后，进行 $g_8 \to f_4$ 和 $e_4 \to f_2$ 这两次翻转，也会出现对称的情况。

接下来考虑第四行中的第三棵树。我们注意到，边 f_3 将树分成两部分。两部分中的翻转可以任意交错进行。因此会产生多个结果相同、仅翻转顺序不同的翻转序列；更具体地说，差异在于翻转 $g_8 \to f_4$ 发生的时机。在 f_3 不包含 g_8 的那一侧，有两个翻转序列：先进行 $e_3 \to f_3$，再进行 $e_4 \to f_2$；或者先进行 $e_3 \to f_3$，再进行 $e_4 \to f_2$。这两种序列最终都会使 T_F 的两条边分别被 T_I 的两条边交叉两次。

在第四行的最后一棵树中，我们添加了 T_F 的边 f_3 和 f_4。T_F 中唯有 f_3 和 f_4 没有被交叉两次，它们都被 e_2 交叉一次。唯一可能的非停放翻转是 $e_2 \to f_4$。现在，唯一可能不被交叉两次的边是 f_3，它可以通过两次翻转 $e_3 \to f_3$ 和 $e_3 \rightrightarrows f_3$ 添加。$e_3 \to f_3$ 后面只能接 $e_2 \to f_2$，而 $e_3 \to f_3$ 后面只能接 $e_2 \to f_2$。两种情况下，最终仍会出现 T_F 的两条边被 T_I 的两条边交叉。◀

C 附加图

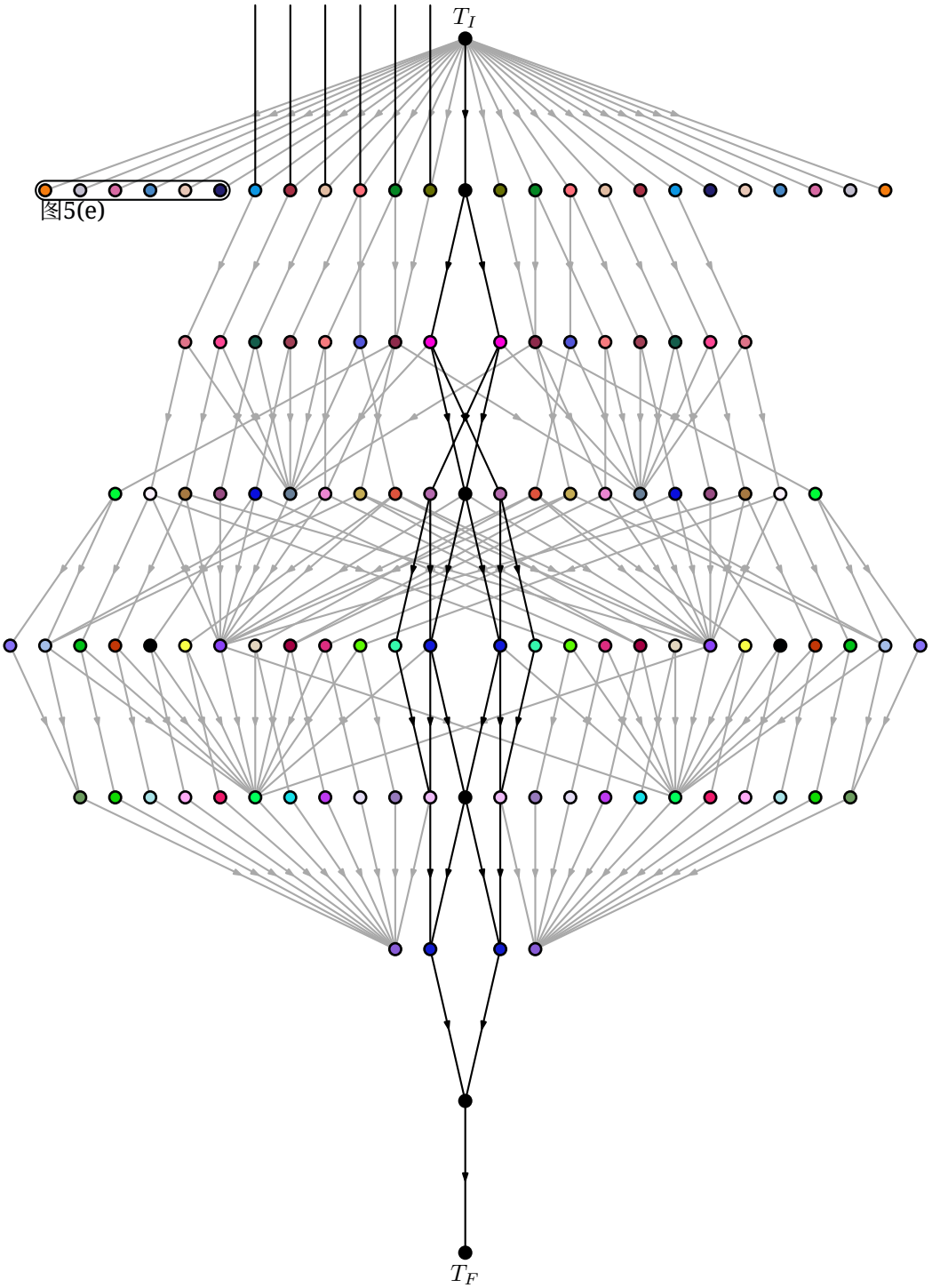


图 12 来自 T_I 的树的可达子图（如图 5 所示），此时最多进行一次停车翻转。黑色边表示执行该翻转不会使目标树 T_F无法由剩余序列到达，节点颜色表示对称树。

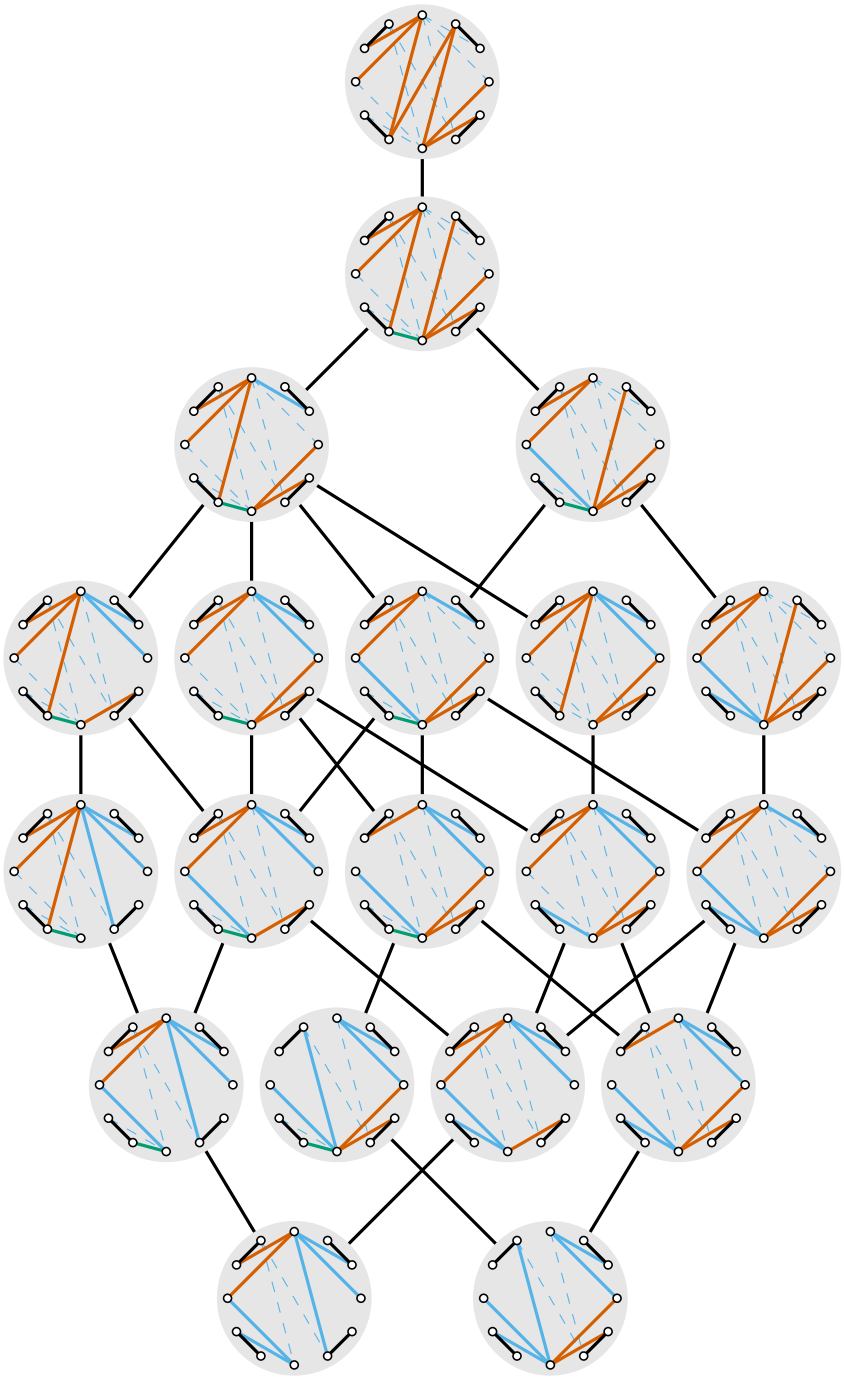


图 13 翻转 $e_1 \to g_4$ 后树的可达子图。

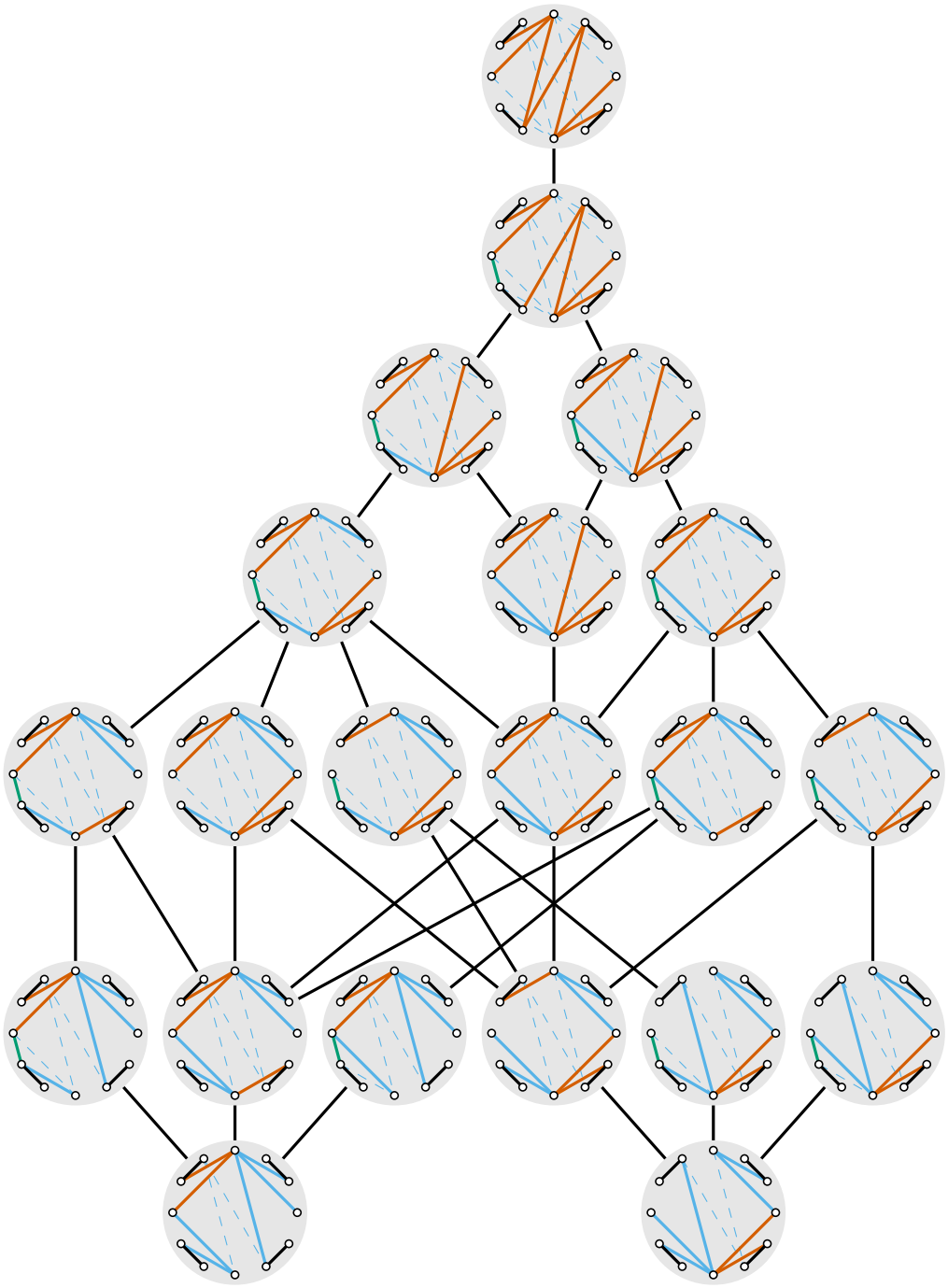
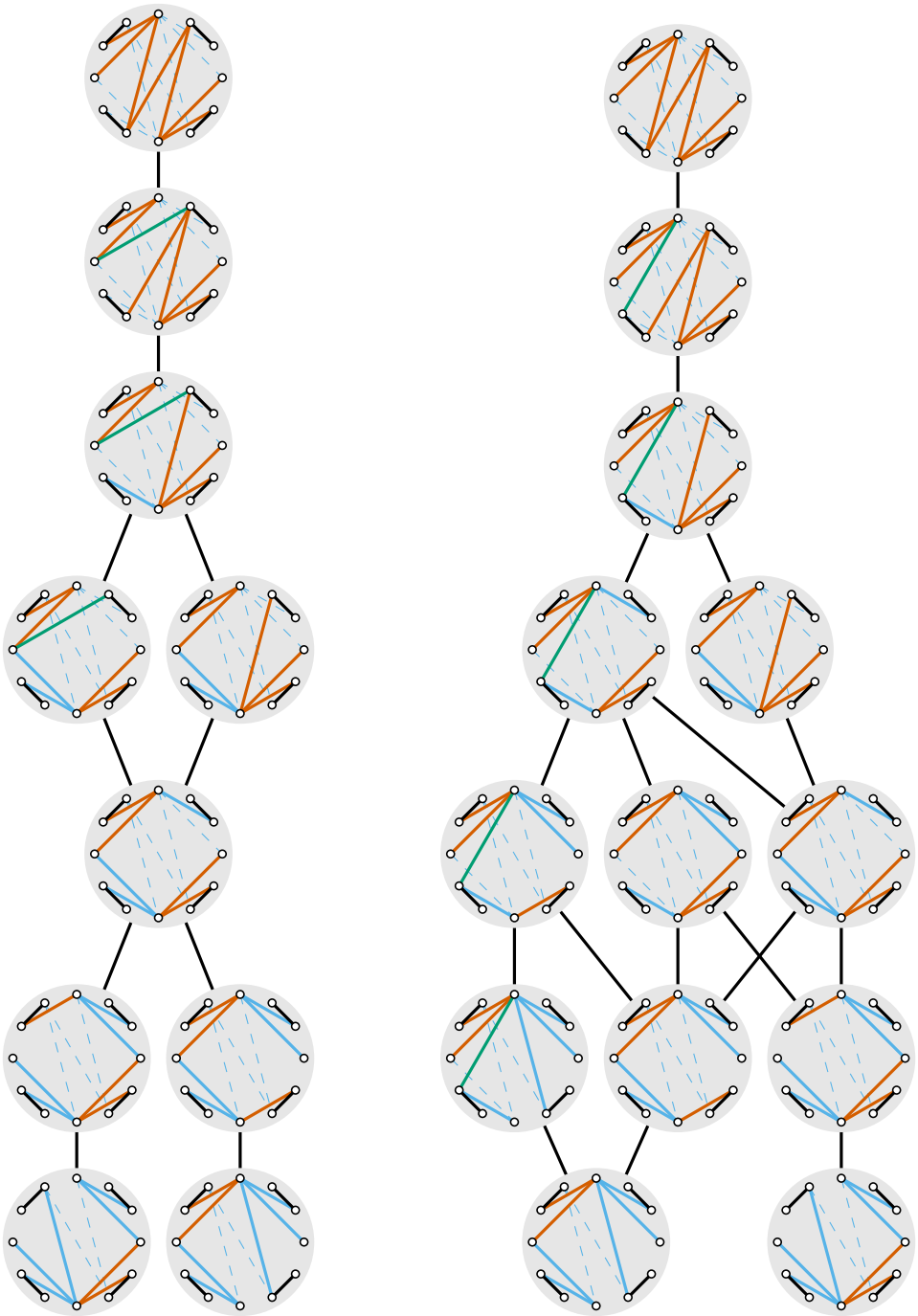


图14 翻转 $e_2 \to g_3$ 后的树的可达子图。



(a) 在翻转 $e_2 \to p_5$ 后的可达树子图，其中 $p_5 = v_4v_{12}$。 **(b)** 在翻转 $e_2 \to p_6$ 后的可达树子图，其中 $p_6 = v_1v_5$。

图1 15 经过两次对角停车翻转后可达树的子图。这些内容与引理 10 的正确性无关，但为完整起见而列出。

28 平面生成树之间最短翻转序列的结构性质

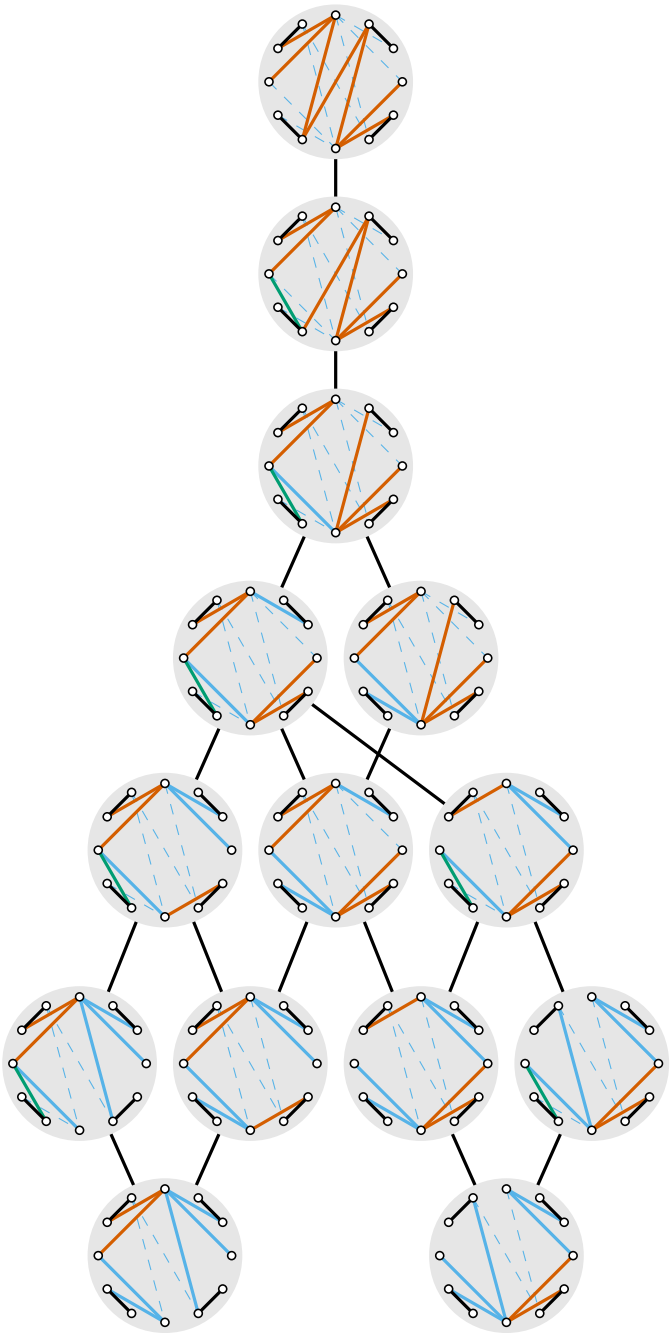


图16 翻转 $e_2 \to p_7$ 后树的可达子图，其中 $p_7 = v_4v_{6^\circ}$ 这与引理 10 的正确性无关，但为完整起见仍予收录。