TheoretiCS

近线性时间内的正交径向绘图

收稿2024年4月16日修订
2025年1月3日录用2025年2月27日发表 2025年4月8日

关键词图绘制、正交径向绘图、拓扑-形状-度量框架

Yi-Jun Chang[a]

a 新加坡国立大学

摘 要。正交绘图是将平面图嵌入网格中的一种表示。在 Tamassia 的开创性工作（《SIAM计算杂志》1987）中，首次建立了可实现为无弯折正交绘图的角度赋值的简单组合刻画，从而可以通过列出所有角的角度，以组合方式描述正交绘图。该刻画减少了对某些几何因素的考量需求，例如边长和顶点坐标，并简化了图绘制算法设计的任务。

Barth、Niedermann、Rutter 和 Wolf（SoCG 2017）为正交径向绘图建立了类似的组合刻画；正交径向绘图是将正交绘图推广到圆柱网格上的一种形式。该刻画的证明是存在性的，并未产生高效算法。随后，Niedermann、Rutter 和 Wolf（SoCG 2019）通过开发二次时间算法解决了这一问题：这些算法既能测试给定的角度赋值是否可实现为无弯折的正交径向绘图，也能构造出这样的绘图。

本文进一步将这些任务的时间复杂度提升至近线性时间。我们基于良序列的概念，为正交径向绘图建立了一种新的刻画。利用这种新刻画，我们设计了一种简单的贪心算法来构造正交径向绘图。

1. 引言

平面图是带有组合嵌入 $G=(V,\ E)$ $\mathcal{E}$的平面图。组合嵌入$\mathcal{E}$为每个顶点 $v \in V$处相邻边指定循环排序$\mathcal{E}(v)$，规定这些边围绕绘图中 v 的逆时针序。正交

本文的初步版本曾发表于 ICALP 2023 [14]。

引用格式 Yi-Jun Chang. 近线性时间的正交径向绘图. TheoretiCS，第 4 卷（2025 年），第 11 篇文章，1-53。

https://theoretics.episciences.org数字对象标识符 10.46298/theoretics.25.11

绘图 是平面图的一种绘图方式，其中 G 的每条边都绘制为由水平和垂直线段组成的序列。例如，图1展示了K_4 的一种正交绘图，其中有四个弯折。或者，也可以将 G 的正交绘图视为将 G嵌入网格，使得 G 的边对应于网格中的内部不相交路径。正交绘图是图绘制领域研究最悠久的绘图风格之一，应用广泛，包括超大规模集成电路设计 [7, 45],、建筑平面图设计 [35],和网络可视化 [5, 24, 28, 32]。

拓扑-形状-度量框架 正交绘图最基本的质量指标之一是弯折的数量。弯折最小化 问题旨在寻找弯折数量最少的正交绘图，过去40年来一直受到广泛研究 [15, 17, 18, 27, 43, 44]。在一项开创性工作中，Tamassia [44] 提出了拓扑-形状-度量 框架来解决弯折数最小化问题。Tamassia 指出，正交绘图可以通过正交表示以组合方式描述；正交表示由对每个{90◦,180◦, 270◦,360◦} 角进行赋值以及指定外面构成。本文将角 定义为给定组合嵌入中关联于同一顶点且连续的两条边组成的边对。具体而言，Tamassia [44] 证明了，当且仅当满足以下两个条件时，正交表示才能实现为零弯折的正交绘图：

(O1) 每个顶点周围的角度和为360◦。(O2) 对于具有 k 个角的每个面，其周围的角度和为：外面的 $(k+2)\cdot 180^\circ$，其他面为$(k-2)\cdot 180^\circ$。

如果满足上述条件（O1）和（O2），则正交表示有效。给定一个有效的正交表示，可以在线性时间内计算出实现该正交表示的正交绘图 [31, 44]。这一结果（形状 → 度量）使我们能够将寻找弯折数最少的正交绘图（拓扑 → 度量）这一任务，归约为概念上简单得多的寻找弯折数最少的有效正交表示（拓扑 →形状）。

通过关注正交表示，我们可以忽略图绘制中的某些几何方面，例如边长和顶点坐标，从而简化算法设计。特别是，给定一个固定的组合嵌入，寻找弯折数最少的正交表示这一任务可以轻松归约为最小费用流计算 [44]。如果不考虑正交表示，而是直接思考正交绘图，就不容易看出这种归约到流计算的方法。

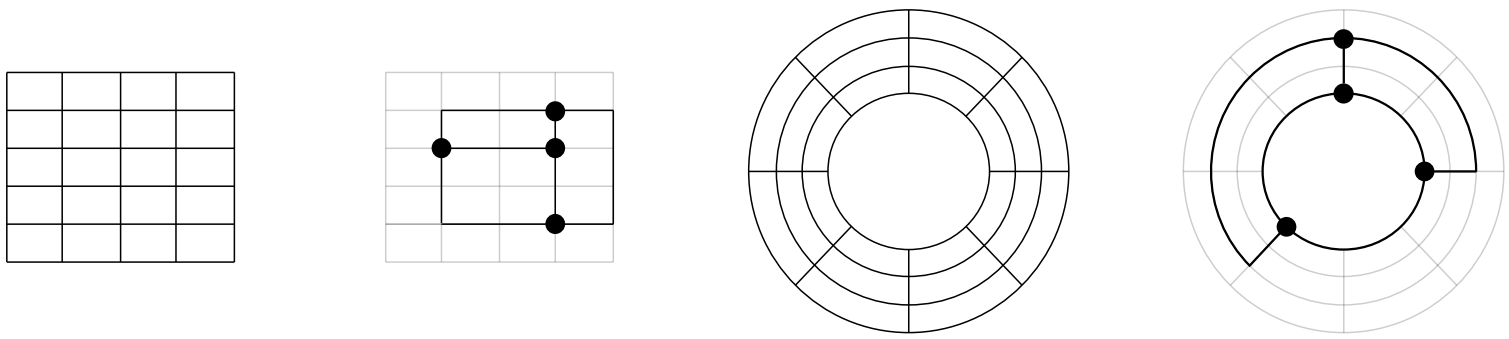

图1。 网格、正交绘图、圆柱网格和正交径向绘图 g.

1.1 正交径向绘图

正交径向绘图 是将正交绘图推广到圆柱网格上的自然方式。圆柱网格的网格线由同心圆以及从圆心向外延伸的直线组成。形式上，正交径向绘图定义为一种平面嵌入，其中每条边都绘制为一系列线段，这些线段要么是以原点为圆心的某个圆上的圆弧，要么是某条经过原点的直线上的线段。我们不允许顶点绘制在原点上，也不允许绘图中的边经过原点。例如，图1展示了 K_4 的一个含有两个弯折的正交径向绘图。

对正交径向绘图的研究源于其在网络可视化中的应用 [4, 25, 47], , 尤其是在交通线路图布局的背景下 [46]。正交径向绘图特别适合用于可视化具有径向线路和环形线路的地铁系统。此类绘图的示例可见于 [38, 41, 42]。

正交径向绘图中有三种类型的面。包含无界区域的面称为外面。包含原点的面称为中心面。其余的面称为正则面。外面和中心面可能是同一个面。

给定一个平面图，正交径向表示被定义为：为每个角赋予一个角度，并指定中心面和外面。Barth、Niedermann、Rutter 和Wolf [2]指出，如果满足以下三个条件，正交径向表示就可以实现为零弯折的正交径向绘图：

（R1）每个顶点周围的角度和为 360◦。

(R2）每个面周围的角度和 s 满足以下条件：该面有 F 个角，且 k .

— $s = (k-2)\cdot 180^\circ$ 如果 F 是正则面。$s = k\cdot 180^\circ$ 如果 F 是中心面或外面之

— 一，但不是两者。$s = (k+2)\cdot 180^\circ$ 如果 F 同时是中心面和外面。

—

(R3) 存在一种参考边的选择，使得正交径向表示中不包含严格单调环。

直观而言，这表明，可以通过考察每个顶点和每个面的角度和，并附加正交径向表示中不存在严格单调环这一要求，类似地刻画那些能够实现为零弯折正交径向绘图的正交径向表示。

严格单调环的定义较为专业，且取决于参考边 $e^\star$ 的选择，因此我们将其正式定义留到后续章节。参考边 $e^\star$ 是外面的轮廓上的一条边，并且必须位于正交径向绘图中使用的最外侧圆弧上。非正式地说，严格单调环的结构类似于一圈向上攀升的台阶，或一圈向下延伸的台阶，因此无法绘制严格单调环。(R1)–(R3) 的必要性直观易见。 [2] 中证明更具挑战性、也更有趣的部分，是说明这三个条件实际上也充分。

1.2 已有方法

期刊论文 [2] 是两项工作的结合 [3, 37]。在第一项工作 [3]，中，对条件 (R1)–(R3) 必要且充分的证明仅具有存在性，也就是说，它并未给出高效算法来检验给定正交径向表示的有效性，也未给出算法来构造实现给定正交径向表示且不含弯折的正交径向绘图。

检查（R1）和（R2）可以用一种直接的方法在线性时间内完成。困难之处在于设计一种高效算法来检查（R3）。检查所有环的最朴素方法需要指数时间。第二项工作 [37] 通过展示一种 $O(n^2)$ 时间算法，解决了这一空白，该算法用于判定给定参考边 $e^\star$是否存在严格单调环，其中 n 是输入图中的顶点数。他们还展示了一种 $O(n^2)$时间算法，用于针对任意给定的不包含严格单调环且具有参考边 $e^\star$ 的正交径向表示，构造无弯折的正交径向绘图。

矩形化第一项工作 [3]中证明的核心思想，是将问题归约到一个更简单的情形：每个正则面都是矩形的。针对这一情形，他们证明了条件 (R1)–(R3) 是必要且充分的，并且在满足 (R1)–(R3) 的前提下，通过归约到流计算，给出了一种高效的绘图算法。

对于任意给定的含有 n 个顶点的正交径向表示，都可以添加 $O(n)$ 条额外的边，将其转化为一种正交径向表示，其中每个普通面都是矩形。 [3] 的证明面临的一大难点是，必须确保添加边后不仅满足 (R1) 和 (R2)，还满足 (R3)。无法高效地检查 (R3) 是否成立，正是 [3]的证明无法直接导出多项式时间算法的原因。

二次时间算法 上述问题已在第二项工作 [37]中得到解决。他们给出了一种 $O(n^2)$时间算法：给定一条固定的参考边，可以在其存在时找到一个严格单调环，

选择参考边 $e^\star$。这立即导出一种 $O(n^2)$时间算法，用于判定给定的、具有固定参考边 $e^\star$ 的正交-径向表示是否存在正交-径向绘图。此外，将这一 $O(n^2)$时间算法与上文讨论的 [3] 证明相结合，可得到一种 $O(n^4)$时间绘图算法。其时间复杂度源于以下事实：矩形化需要进行 $O(n)$次边添加；对于每次边添加，都需要考虑 $O(n)$ 条候选参考边；而为了测试每条候选边的可行性，还需要运行该 $O(n^2)$时间算法，以检验边添加是否会产生严格单调环。

用于寻找严格单调环的 $O(n^2)$时间算法的核心思想是一个结构定理：如果存在严格单调环，那么就存在一个唯一的最外层严格单调环，可以通过从最外层严格单调环上的任意一条边开始执行左优先深度优先搜索来找到它。深度优先搜索算法的耗时为 $O(n)$ 时间。在最外层单调环上猜测一条边，会使时间复杂度增加 $O(n)$ 倍的开销。

利用对 [3], 增广过程的进一步结构性洞察，可以将上述 $O(n^4)$时间绘图算法的时间复杂度降低至 $O(n^2)$ [37]。导致二次时间复杂度的原因是，对于每一次 $O(n)$ 边添加，都需要从新添加的边开始执行一次左优先深度优先搜索，以检验添加该边是否会产生严格单调环。

1.3 我们的新方法

对于有效性检验（检查给定的角度赋值是否会导出严格单调环）和绘图（寻找能够实现给定正交径向表示的几何嵌入），这两种算法在 [2] 中自然都需要 $O(n^2)$ 时间，因为它们都需要执行 $O(n)$ 次左优先深度优先搜索。

在本文中，我们提出了一种用于正交径向绘图的新方法，该方法不基于矩形化和左优先深度优先搜索。对于参考边 $e^\star$ 固定的情况，我们设计了一种简单的 $O(n\log n)$ 时间贪心算法，同时完成有效性检验和绘图。如果参考边 $e^\star$ 未固定，我们的算法需要 $O(n\log^2 n)$ 时间，其中额外的 $O(\log n)$ 因子源于对参考边候选集合进行二分搜索。从较高层次来看，我们的算法尝试以逐块的方式构造正交径向绘图。如果在某个时刻无法取得进展，即当前的部分绘图无法进一步扩展，那么算法便可以识别出一个严格单调环，以证明绘图不存在。

良序列 我们方法的核心是良序列这一概念，下面简要说明。给定一条固定的参考边 $e^\star$，满足 (R1) 和 (R2) 的正交径向表示可以确定一条边 e 是否为垂直边（即 e 被绘制为一段经过原点的直线），或为水平边（即 e 被绘制为以原点为圆心的某个圆上的一段圆弧）。令 E_h 表示水平边的集合，这些边沿顺时针

方向定向，并令 $\mathcal{S}_{\mathsf{h}}$ 表示由 E_{h} 诱导的连通分量集合。注意，每个分量 $S \in \mathcal{S}_{\mathsf{h}}$ 要么是路径，要么是环。

良序列的确切定义较为技术性，因此我们将在后续章节给出。直观而言，良序列是$\mathcal{S}_{\mathsf{h}} = (S_1, S_2, \ldots, S_k)$ 的一个排列，其中 $k = |\mathcal{S}_{\mathsf{h}}|$，它使我们能够设计一种简单的线性时间贪心算法来构造正交径向绘图，并使得 S_1 绘制在圆 $r = k$上， S_2 绘制在圆 $r = k - 1$上，依此类推。

一般而言，即使给定的正交径向表示允许进行正交径向绘图，良序列也可能不存在。在这种情况下，我们证明可以添加虚拟边，将正交径向表示转换为存在良序列的表示。我们将设计一种贪心算法，用于添加虚拟边并构造良序列。每一步中，我们都向当前图添加虚拟垂直边，并将一个新元素 $S \in \mathcal{S}_{\mathsf{h}}$ 追加到序列末尾。如果我们无法找到任何合适的 $S \in \mathcal{S}_{\mathsf{h}}$ 来扩展序列，就可以提取一个严格单调环，以证明不存在正交径向绘图。需要强调的是，该环属于原图，且不使用任何虚拟边。

我们的方法与 [2] 中基于矩形化的方法有一个主要区别：在我们的算法中，添加一条新的虚拟边的代价仅为 $O(\log n)$。正如我们稍后将看到的，在我们的算法中，为了确定要添加的新的虚拟边，只需进行一些简单的局部检查，例如计算角度和；无需执行完整的左优先深度优先搜索来检验新添加的边是否会形成严格单调环。

开放问题 虽然我们给出了一个近乎线性时间的算法，用于解决（形状 $\rightarrow$ 度量）步骤（即从正交径向表示到正交径向绘图），但对于（拓扑 $\rightarrow$ 形状）步骤（从平面图到正交径向表示），目前几乎一无所知。对于给定的平面图，求其最少弯曲正交表示的问题可以轻易归约为计算最小费用流 [44],，但这种归约并不适用于正交径向表示，因为网络流难以处理严格单调环这一概念。能否在多项式时间内计算出平面图的最少弯曲正交径向表示，仍是一个开放问题。

1.4 相关工作

正交绘图是图绘制中的核心课题，综述请见 [22] 。对于最大度数为 4 且未固定组合嵌入的平面图，其正交绘图的弯折数最小化问题是 NP-困难 [26, 27]。若组合嵌入已固定，Tamassia [44] 的拓扑-形状-度量框架可将弯折最小化问题归约为最小费用流计算。Tamassia [44] 的算法耗时为 $O(n^2 \log n)$。后续研究将时间复杂度改进为 $O\left(n^{7/4}\sqrt{\log n}\right)$ [27]，之后又改进为 $O\left(n^{3/2} \log n\right)$ [15]。近期的一项

$O(n \operatorname{poly}\log n)$时间的平面最小费用流算法 [21] 表明，若组合嵌入已固定，则弯折数最小化问题可在 $O(n \operatorname{poly}\log n)$ 时间内求解。

如果组合嵌入未固定，那么只要每条边上的第一个弯折不产生任何代价 [9]，就可以规避 [26, 27] 的 NP-困难性结论；或者，我们也可以将问题限制在某些特殊的平面图类别中。具体而言，已有研究证明，对于最大度数为 3 的平面图，弯折最小化问题可以在多项式时间内求解 [17]。经过一系列改进 [13, 18, 19], ，如今我们知道，可以在 $O(n)$ 时间内计算出最大度数为 3 的平面图的弯折最小化正交绘图 [18]。

拓扑-形状-度量框架 [44]不仅适用于弯折最小化，而且在本质上所有关于正交绘图及其变体的计算问题的图绘制算法中，也隐含地或显式地发挥着作用，例如正交绘图的变形 [8], 、允许顶点度数大于 4 [16, 34, 39], 、限制边的方向 [20, 23], 、绘制簇图 [10], 以及绘制动态图 [11]。

Barth、Niedermann、Rutter 和 Wolf [2] 对正交径向绘图的研究，将拓扑-形状-度量框架 [44] 扩展到适用于圆柱网格。在这项工作 [2],之前，对于路径、环和 theta 图 [30], ，以及图为 3-正则且正交径向表示中的每个正则面都是矩形这一特殊情形 [29]，人们才知道可绘制正交径向表示的组合刻画。

1.5 组织结构

第 2 节讨论本文使用的基本图论术语，回顾前人工作的部分结果 [2], 并陈述我们的主要定理。第 3 节引入良序列的概念，并说明良序列的存在意味着存在一种简单的正交径向绘图算法。第 4 节介绍一种贪心算法：它向给定的、具有固定参考边的正交径向表示中添加虚拟边，使得存在一个覆盖整个图的良序列，并且该序列可以高效计算。第 5 节说明，如果贪心算法失败，则存在一种严格单调环，可据此判定不存在绘图；这种环也可以高效计算。第 6 节说明，我们的结果可以扩展到参考边不固定的情形，但时间复杂度会额外增加一个对数因子。第 7 节通过展示从任意图到双连通简单图的归约，论证将输入图假设为简单且双连通的合理性。最后，第 8 节讨论未来可能的研究方向。

2. 预备知识

全文中，设 $G=(V,\ E)$ 为一个最大度数不超过 4 的平面图，并具有固定的组合嵌入$\mathcal{E}$，具体而言，对于每个顶点 $v \in V$，给定其关联边的循环排序$\mathcal{E}(v)$，以指定这些边的逆时针序

在平面嵌入中围绕 v。正如我们将在第 7 节讨论的，可以假设输入图 G 是简单的且双连通。本节将介绍一些基本的图论术语，并回顾 Barth、Niedermann、Rutter 和 Wolf [2] 的一些结果。

路径与环 除非另有说明，所有边、路径和环均假定为有向的。我们分别用 e、 P 和 C 表示边 e、路径 P 和环 C 的反向。我们允许路径和环包含重复的顶点和边。如果路径或环不含重复顶点，我们称其为简单的。按照 [2]，的说法，如果路径或环满足以下条件，则称其为无交叉：

- 路径或环不包含重复的无向边。图 2 展示了一个例子：环 $C = (v_1, v_5, v_6, v_3, v_4, v_7, v_6, v_5, v_9, v_{10}, v_2)$ 不是无交叉的，因为它沿相反方向两次经过无向边 $\{v_5, v_6\}$。对于路径或环中多次出现的每个顶点 v，路径或环中与 v 相 incident 的边的排列顺序，与这些边在 E (v) 中的顺序相同，或与其相反。图 2 展示了一个例子：路径 $(v_{11}, v_9, v_5, v_1, v_2, v_{10}, v_9, v_8)$ 不是无交叉的，因为它在 v_9 处与自身相交；路径 $(v_8, v_9, v_5, v_1, v_2, v_{10}, v_9, v_{11})$ 是无交叉的，因为路径中与 v_9 相 incident 的边的排列顺序与这些边在 E (v_9) 中的顺序相同。

尽管无交叉路径或无交叉环可能多次经过同一个顶点，但路径或环本身绝不会与自身交叉。对于任意一个面 F，我们将其轮廓的顺时针遍历定义为其面环 C_F。一般而言，面环可能不是简单圈，因为其中可能包含重复的边。例如，图 2 中的环 $C = (v_1, v_5, v_6, v_3, v_4, v_7, v_6, v_5, v_9, v_{10}, v_2)$ 并非简单圈，但它是 F_2 的面环。若假设 G 是双连通的，则 G 的每个面环都必须是简单的无交叉环。

正交径向表示与绘图 角 是与 v 相 incident 的一对有序无向边 (e_1, e_2)，其中 e_2 在逆时针循环次序 $\mathcal{E}(v)$ 中紧接在 e_1 之后。给定一个平面图 $G = (V, E)$ 及其固定的组合嵌入 $\mathcal{E}$，其正交径向表示 $\mathcal{R} = (\phi, F_c, F_o)$ 由以下部分定义：

- 向 G的每个角赋予 ϕ 一个角度 $a \in \{90^\circ, 180^\circ, 270^\circ\}$。将 G 的一个面指
- 定为中心面 F_c。将 G 的一个面指定为外面 F_o。
-

对于 v 只有一条关联边 e的特殊情形，我们将(e, e) 视为一个360° 角。如果考虑双连通图，则不会出现这种情形。

如果正交径向表示 $\mathcal{R} = (\phi, F_c, F_o)$ 可以实现为 G 的零弯折正交径向绘图，且每个角的角度与赋值 ϕ相符、中心面 F_c 包含原点，并且外面 F_o 包含一个无界区域，则称该表示是可绘制的 。

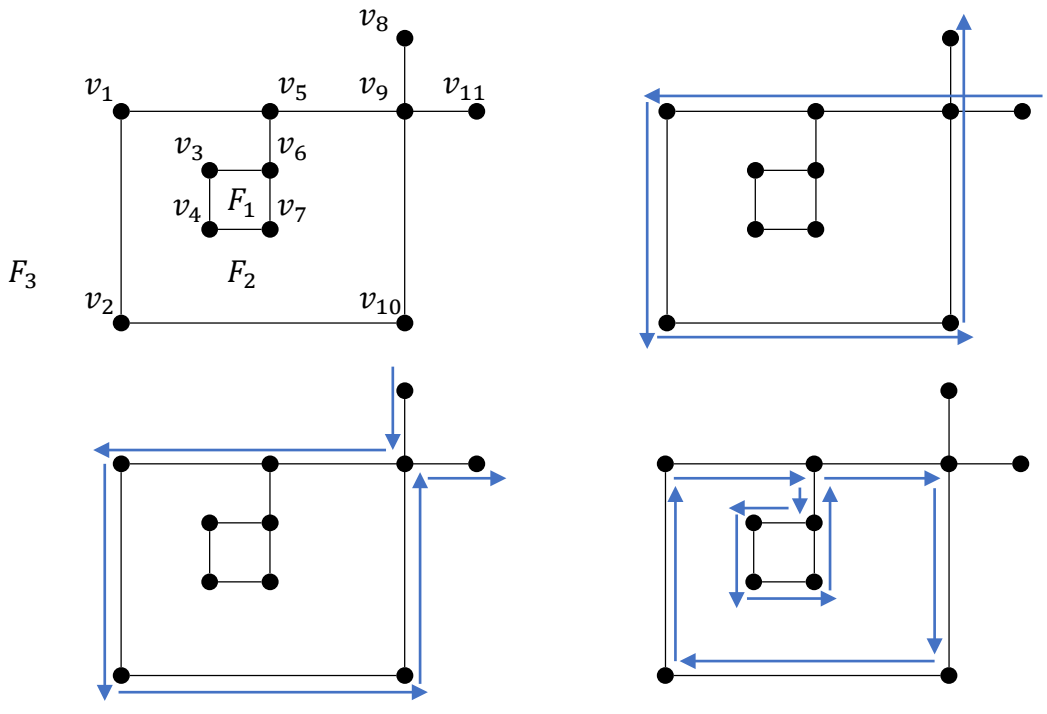

图 2。 一条有交叉的路径、一条无交叉路径和一个面环。

回顾一下，根据正交径向绘图的定义，在零弯折的正交径向绘图中，每条边都绘制为一条经过原点的直线上的线段，或绘制为以原点为圆心的圆上的圆弧。我们还考虑参考边 $e^\star$ 固定的情形。在这种情况下，还要求参考边 $e^\star$ 位于绘图所用的最外侧圆弧上，并沿顺时针方向延伸。如果存在这样的绘图，我们称 $(\mathcal{R}\ e^\star)$ 是可绘制的。图3展示了一个正交径向表示 $\mathcal{R}$ 的绘图示例，其中参考边为 $e^\star = (v_{14}, v_5)$。图中，我们分别用 ∘、∘∘和∘ ∘ ∘ 表示赋予一个角的角度分别为 90∘、180∘和 270∘ 。

已在 [2] 中证明，$(\mathcal{R}\ e^\star)$ 可绘制，当且仅当正交径向表示 $\mathcal{R}$满足 (R1) 和 (R2)，且带有参考边 $e^\star$ 的表示不包含严格单调环。由于可以在线性时间内直接测试是否满足 (R1) 和 (R2)，因此从现在起，除非另有说明，我们假设所考虑的正交径向表示 $\mathcal{R}$ 满足 (R1) 和 (R2)。

组合旋转考虑一条经过 v 且满足 $u \neq w$ 的长度为 2 的路径 $P = (u,\ v,\ w)$ 。给定角度赋值 ϕ, P 可以是 90∘ 左转、直线或 90∘ 右转。我们定义 P 的组合旋转如下。

$$\text{rotation}(P) = \begin{cases} -1, & P \text{ is a } 90^\circ \text{ left turn,} \\ 0, & P \text{ is a straight line,} \\ 1, & P \text{ is a } 90^\circ \text{ right turn.} \end{cases}$$

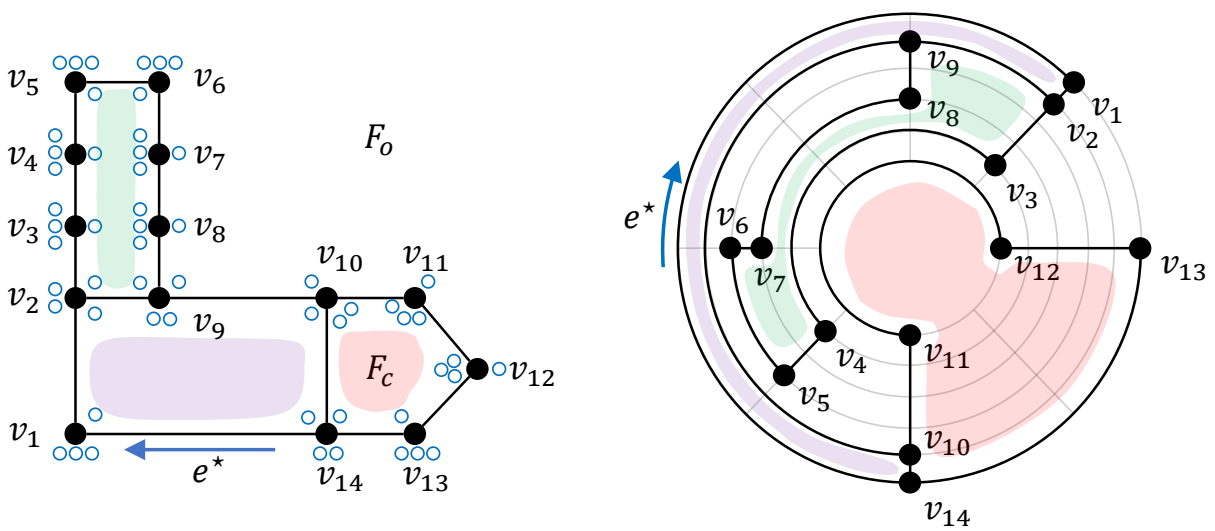

图3. 一个带有参考边的正交径向表示绘图，其中左图中的蓝色小圆表示该表示中的角，这些角在右图中得到了实现。

更正式地说，令 $S = (e_1, \ldots, e_k)$ 表示从 $e_1 = \{u, v\}$ 开始、到 $e_k = \{v, w\}$ 结束的连续子序列，它取自与 v 相 incident 的无向边在循环排序 E (v) 中。于是，$\sum_{j=1}^{k-1} \phi(e_j, e_{j+1}) - 180$度等于在中间顶点 v 处 P 的转向度数，因此 P 的组合旋转为 $\text{rotation}(P) = \left(\sum_{j=1}^{k-1} \phi(e_j, e_{j+1}) - 180\text{度}\right) / 90\text{度}$。

对于 $u = w$ 的特殊情形，$P = (u, v, u)$ 的旋转可能是 180。左转，此时 $\text{rotation}(P) = -2$；也可能是 180。右转，此时 $\text{rotation}(P) = 2$。例如，考虑有向边 $e = (u, v)$，其中 P 先沿着 e 的右侧从 u 到 v，然后沿着 e 的左侧从 v 返回 u。此时，P 被视为 180。左转；类似地，P 被视为 180。右转。特别地，如果 $P = (u, v, u)$ 是面环 C 的子路径，则 P 始终被视为 180。左转，因此 $\text{rotation}(P) = -2$。

对于任意单边路径 P，定义旋转$(P) = 0$。对于任意长度大于 2 的无交叉路径 P，将旋转(P) 定义为其所有长度为 2 的子路径的组合旋转之和。类似地，对于任意长度大于 2 的环 C，将旋转(C)定义为其所有长度为 2 的子路径的组合旋转之和。与基于此概念的 [2]，相同，我们可以将条件 (R2) 重述如下。

(R2′) 对于每个面 F，其面环 C_F 的组合旋转满足以下条件：

$$\text{旋转}(C_F) = \begin{cases} 4, & F \text{ 是正则面，} \\ 0, & F \text{ 是中心面或外面，但不能同时是两者，} \\ -4, & F \text{ 既是中心面又是外面。} \end{cases}$$

例如，考虑图3所示的正交径向表示。路径 $P = (v_{10}, v_{11}, v_{12}, v_{13}, v_{14})$ 的旋转 $(P) = -1$，因为它左转两次，每次 90。，右转一次，每次 90。

度。环 $C = (v_{10}, v_{11}, v_{12}, v_{13}, v_{14})$ 是中心面的面环，其旋转 $(C) = 0$。

新定义与旧定义中 (R2) 的等价性源于这样一个事实：90◦ 左转对应一个 270◦角。如果 F 是一个有 k 个角的正则面，那么在 (R2) 的原始定义中，要求 F 周围的角度总和 s 为 $s = (k-2)\cdot 180^\circ$。由于面环 C_F 按顺时针方向沿着 F 的轮廓遍历，90◦右转的次数减去 90◦ 左转的次数必须恰好为 4。因此，$s = (k-2)\cdot 180^\circ$与旋转$(C_F) = 4$相同，因为在计算旋转(C_F)时，每次 90◦ 右转贡献 $+1$，每次 90◦ 左转贡献 -1。

环的内部区域和外部区域 任意环 C 都会将其余图划分为两部分。如果 C 是面环，则其中一部分为空。相对于这两部分中的一部分而言，C 的方向为顺时针。相对于 C 呈顺时针方向的那一部分连同 C 本身，称为 C 的内部。类似地，相对于 C 呈逆时针方向的那一部分连同 C 本身，称为 C 的外部。特别是，如果顶点 v 位于 C 的内部，那么 v 必须位于 C 的外部。

上述定义与面环的概念一致，因为任意面 F 都位于其面环 C_F的内部。根据上下文，环的内部或外部可以视为子图、顶点集、边集或面集。例如，考虑图 2 所示平面图中的环 $C = (v_1, v_2, v_{10}, v_9, v_5)$。$C$ 的内部是由 v_8、v_{11}以及 C中的所有顶点诱导的子图。C 的外部是由v_3、v_4、v_6、v_7以及 C中的所有顶点诱导的子图。环 C 将面分成两部分：C 的内部包含 F_3，而 C 的外部包含 F_1 和 F_2。

设 C 为一个简单圈，并定向使外面 F_o 位于其外部。按照 [2],，如果中心面 F_c 位于 C的内部，我们称 C 为本质的。否则，我们称 C 为非本质的。以下引理已在 [2]中证明。

引理 2.1 ([2])。假设 *(R1)* 和 *(R2)* 均成立。设 C 是一个简单圈，其方向使外面 F_o 位于其外部，则 C 的组合旋转满足以下条件。

$$\text{rotation}(C) = \begin{cases} 4, & C \text{ is an essential cycle,} \\ 0, & C \text{ is a non-essential cycle.} \end{cases}$$

该引理背后的直觉是：本质环的行为类似于外面的面环或中心面的面环，而非本质圈的行为类似于正则面的面环。

子图 取子图 H of G 时，H 的组合嵌入、角度赋值、中心面和外面自然继承自 G。更准确地说，设 e_1、e_2 和 e_3

是与 v 相 incident 的三条边，它们在循环排序 E（v）中连续出现。若移除 e_2，则新角（e_1, e_3）的角度赋值由 ϕ（e_1, e_2）+ϕ（e_2, e_3）确定。例如，假设 E（v）=（e_1, e_2, e_3），其中 ϕ（e_1, e_2）为 $=90^\circ$，ϕ（e_2, e_3）为 $=180^\circ$，且 ϕ（e_3, e_1）在 G 中为 $=90^\circ$。若 v 在 H 中仅与边 e_1 和 e_3 incident，则 H 中围绕 v 的两个角的角度赋值 ϕ_H 分别为 ϕ_H（e_1, e_3）$=270^\circ$ 和 ϕ_H（e_3, e_1）$=90^\circ$。

G 中的每个面都包含在 H 的恰好一个面中。H 中的一个面可以包含 G 的多个面。若 H 的一个面包含 G 的中心面，则称其为中心面。类似地，若 H 的一个面包含 G 的外面，则称其为外面。

例如，考虑图3所示正交径向表示中由 $\{v_2, v_3, \ldots, v_9\}$ 诱导的子图 H。在 H 中，v_9 只有两条关联边 $e_1=\{v_8, v_9\}$ 和 $e_2=\{v_2, v_9\}$，而 H 中围绕 v_9 的两个角的角度赋值 ϕ_H 将分别为 $\phi_H(e_1, e_2)=90^\circ$ 和 $\phi_H(e_2, e_1)=270^\circ$。$H$ 的外面和中心面相同。

通过参考路径定义方向 对于任意两条边 $e=(u, v)$ 和 $e'=(x, y)$，若无交叉路径 P 的起点为 u 或 v，终点为 x 或 y，且 P 不包含集合 $\{e, e, e', e'\}$ 中的任何边，则称 P 为 e 和 e' 的参考路径。给定边 $e=(u, v)$ 和 $e'=(x, y)$ 的一条参考路径 P，我们如下定义 e' 相对于–e 和—P 的组合方向。

$$\text{direction}(e, P, e') = \begin{cases} \text{rotation}(e \circ P \circ e'), & P \text{ starts at } v \text{ and ends at } x, \\ \text{rotation}(\overline{e} \circ P \circ e') + 2, & P \text{ starts at } u \text{ and ends at } x, \\ \text{rotation}(e \circ P \circ \overline{e'}) - 2, & P \text{ starts at } v \text{ and ends at } y, \\ \text{rotation}(\overline{e} \circ P \circ \overline{e'}), & P \text{ starts at } u \text{ and ends at } y. \end{cases}$$

这里，$P \circ Q$ 表示路径 P 和 Q 的连接。边 e 可视为长度为 1 的路径。在方向(e, P, e') 的定义中，我们允许参考路径 P 只包含一个顶点的情况。若 $v=x$ 和 $u \neq w$，则可以将 P 选为仅由单个顶点 $v=x$ 构成的长度为 0 的路径，此时方向(e, P, e') 就是长度为 2 的路径 (u, v, y) 的组合旋转。我们不考虑 $e=e'$ 或 $e=e'$ 的情况。

考虑图3的正交径向表示中的参考边 $e=(v_{14}, v_1)$。我们根据参考路径 P 的不同选择，测量从 e 出发的 $e'=(v_8, v_9)$ 的方向。如果$P=(v_1, v_2, v_9)$，那么 direction(e, P, e') =rotation$(e \circ P \circ e') - 2 = -1$。如果 $P=(v_{14}, v_{10}, v_9)$，那么我们也有 direction(e, P, e') =rotation$(e \circ P \circ e') = -1$。如果我们选择 $P=(v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8)$，那么得到的 direction$(e, P, e')$ =rotation$(e \circ P \circ e') = 3$ 的值会有所不同。正如我们稍后将讨论的，direction(e, P, e') mod 4 在选择 P [2] 时保持不变。

在方向(e, P, e')的定义中，旋转$(e \circ P \circ e') + 2$ 中的加项 +2 源于这样一个事实：我们实际想要考虑的路径是 $e \circ e \circ P \circ e'$，其中我们在 $e \circ e$处进行了180。右转，这在组合旋转的计算中贡献了 +2 。同样，旋转$(e \circ P \circ e') - 2$ 中的加项 −2 源于这样一个事实：我们实际想要

考虑的路径是 $e \circ P \circ e' \circ e'$，其中我们在 $e' \circ e'$处进行了180。左转。旋转$(e \circ P \circ e')$ 中没有加项，这是因为180。右转 $e \circ e$ 与180。左转 $e' \circ e'$相互抵消。为什么 $e \circ e$ 必须是右转，而 $e' \circ e'$ 必须是左转，稍后将作解释。

有关边方向计算的示例，请参见图4。根据上述公式，可以通过旋转$(e^\star \circ P \circ e') + 2 = 1$ 计算 $e = (u_1, u_2)$ 相对于 $e^\star = (v_1, v_2)$ 以及参考路径 $P = (v_1, v_5, v_4, u_1)$ 的方向，其中加法项+2 源于 $e^\star \circ e^\star$ 处的180。右转。

边方向 设想原点位于南极，在没有弯折的正交径向绘图中，每条边 e 均沿以下四个方向之一绘制：

- e 指向北 方向，当 e 被绘制为一条经过原点的直线的线段，且 e 背离原点时。e 指向南 方向，当 e 被绘制为一条经过原点的直线的线段，且 e 指向原点时。e 指向东 方向，当 e 被绘制为以原点为圆心的圆的圆弧，且方向为顺时针方向时。e 指向西 方向，当 e 被绘制为以原点为圆心的圆的圆弧，且方向为逆时针方向时。

我们称 e 为一条 垂直 边，如果 e 指向北或南。否则，我们称 e 为一条 水平 边。我们说明，只要满足 (R1) 和 (R2)，任意边 e 的方向都由正交径向表示 $\mathcal{R}$ 和参考边 $e^\star$ 唯一确定。

对于参考边 $e^\star$，要求 $e^\star$ 指向东，因此 $e^\star$ 指向西。考虑任意一条既不是 $\overline{e^\star}$ 也不是 $e^\star$的边 e。显然，方向$(e^\star, P, e)$ 的值决定了 e 的方向：如果方向$(e^\star, P, e)$ 模 4 分别等于 0、1、2 或 3，则 e 的方向必定为东、南、西或北。例如，在图3的正交径向表示中，边 $e' = (v_8, v_9)$ 是指向北的垂直边，因为我们计算得出方向$(e^\star, P, e')$ 模 $4 = 3$。

引理 2.2 ([2])。 对于任意两条边 e和 e'，方向$(e,\ P,\ e')$ 模 4 的值与参考路径 P 的选择无关。

上述引理表明，方向$(e^\star,\ P,\ e)$ 模 4 不受参考路径 P 的选择影响，因此，正交径向表示中每条边的方向都是定义明确的，即使在$(\mathcal{R}\ e^\star)$ 可能无法绘制的情况下也是如此。给定参考边 $e^\star$，令 E_h 表示所有朝东方向的水平边组成的集合，并令 E_v 表示所有朝北向的垂直边组成的集合。

水平线段 我们要求，在$(\mathcal{R}\ e^\star)$的绘图中，参考边 $e^\star$ 位于绘图所用的最外侧圆弧上，因此，并非 C_{F_o} 中的每条边都能作为参考边。要确定边 $e \in C_{F_o}$ 是否能作为参考边，我们需要先介绍一些术语。

给定参考边 $e^\star$，北向的垂直边集合 E_v 和东方向的水平边集合 E_h 已固定。令 $\mathcal{S}_\mathsf{h}$ 表示由 E_h 诱导的连通分量集合。每个分量 $S \in \mathcal{S}_\mathsf{h}$ 要么是一条路径，要么是一个环，因此在 $\mathcal{R}$ 的任意绘图中，都存在一个以原点 C 为圆心的圆，使得 S 必须绘制为 C 或其 C圆弧。我们将每个分量 $S \in \mathcal{S}_\mathsf{h}$ 称为水平线段。

每条水平线段 $S \in \mathrm{S}_\mathsf{h}$ 都写作顶点序列 $S = (v_1, v_2, \ldots, v_s)$，其中 s 是 S 中顶点的数量，并满足对每个 $1 \le i < s$，$(v_i, v_{i+1}) \in E_\mathsf{h}$。若 S 是一个环，则还满足 $(v_s, v_1) \in E_\mathsf{h}$，因此 $S = (v_1, v_2, \ldots, v_s)$ 是一个圆周序。若 S 是一个环，我们对索引采用模运算，使得 $v_{s+1} = v_1$。我们用 $\mathrm{N}_{\mathsf{north}}(S)$ 表示满足 $x \in S$ 的垂直边集合 $e = (x, y) \in E_\mathsf{v}$。类似地，$\mathrm{N}_{\mathsf{south}}(S)$ 表示满足 $y \in S$ 的垂直边集合 $e = (x, y) \in E_\mathsf{v}$。我们假设 $\mathrm{N}_{\mathsf{north}}(S)$ 和 $\mathrm{N}_{\mathsf{south}}(S)$ 中的边按其端点在 S中的索引排序。若 S 是一条路径，则该排序为顺序排序；若 S 是一个环，则该排序为圆周排序。以图3中的正交径向表示 R 为例。水平线段 $S = (v_{10}, v_9, v_2)$ 的 $\mathrm{N}_{\mathsf{south}}(S) = ((v_{11}, v_{10}), (v_8, v_9), (v_3, v_2))$，而 $\mathrm{N}_{\mathsf{north}}(S) = ((v_{10}, v_{14}), (v_2, v_1))$。

注意，对于包含 $e^\star$ 的水平线段 $S \in \mathcal{S}_\mathsf{h}$，存在一种绘图，使得$\mathcal{N}_{\mathsf{north}}(S) = \emptyset$ 位于$\mathcal{R}$ 的最外侧圆弧上，这是必要条件。此条件可以在线性时间内轻松检查。

螺旋度 直观地说，方向(e, P, e') 量化了 e' 相对于e 和 P的螺旋度 。遗憾的是，若将方向(e, P, e') 模 4 替换为方向(e, P, e')，引理 2.2 并不成立。 [2] 中的一项关键观察指出，若对 e、 e'和 P的位置增加一些限制，则可以进行这种替换。请看下面的引理。

引 理 2 . 3 ([2]). 设 C和 C' 是本质环，且 C' 位于 C的内部。设 e 是 C上的一条边， e' 是 C'上的一条边。对于位于 C 内部且位于 C'外部的所有路径 P ，值方向(e, P, e') 在参考路径 P的选择下保持不变。

请注意，我们要求参考路径是无交叉的。上述引理中的这一要求至关重要。如果允许 P 是一条并非无交叉的一般路径，那么我们可以选择 P ，使得 P 反复遍历一个非必要的 环许多次，从而可以使方向(e, P, e') 任意大或任意小。

在上述引理中令 $e = e^\star$ 和$\overline{C} = C_{F_o}$，可推知，一旦固定包含 e' 的本质环 C' ，并且只考虑位于 C'外部的参考路径 P ，方向（$e^\star$， P， e'）便确定了。引理的条件得到满足，因为 C_{F_o} 是最外层

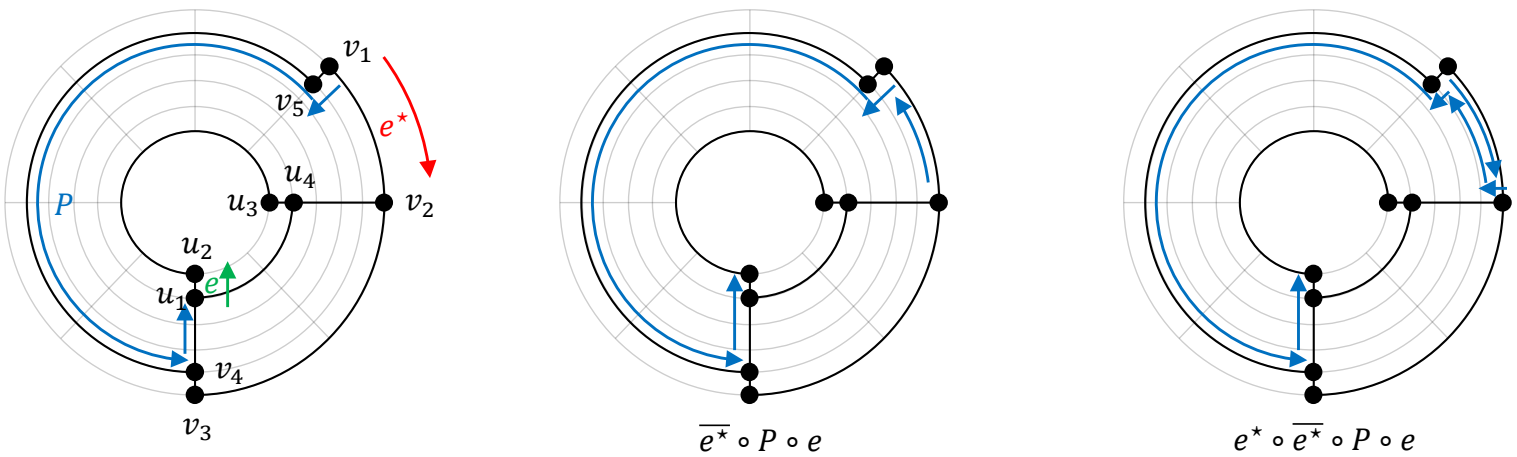

图4. $e^\star$的方向, P, e 的计算。

本质环，因为所有其他本质环都位于 C_{F_o}的内部。我们令$C = C_{F_o}$ 而不是 $\overline{C} = C_{F_o}$ 的原因是， F_o 必须位于 $\overline{C}$的外部。请注意，假设 G是双连通的，确保每个面环都是简单的。

设 C 为一个本质环，并设 e 为 C中的一条边。鉴于上述内容，按照 [2]，我们将 e 相对于 C 的边标签 $\ell_C(e)$定义为 direction$(e^\star, P, e)$ 的值，其中参考路径 P 可以任选，只要它位于 C的外部。对于特殊情形 $e = e^\star$ 和 $C = C_{F_o}$，我们令 $\ell_C(e) = 0$。直观而言，值 $\ell_C(e)$量化了在仅限于 C外部的情况下，e相对于 $e^\star$ 的螺旋度。以图4中的本质环$C = (u_1, u_2, u_3, u_4)$上的边 $e = (u_1, u_2)$为例。由于参考路径 $P = (v_1, v_5, v_4, u_1)$位于 C的外部，我们有 $\ell_C(e) = \text{direction}(e^\star, P, e) = 1$。

我们简要解释一下方向(e, P, e')的方向公式：如前文所述，在方向(e, P, e')的定义中，旋转$(e \circ P \circ e')$中的加性 $+2$ $+2$ 源于这样一个事实：我们实际想要考虑的路径是 $e \circ e \circ P \circ e'$，其中我们在 $e \circ e$中进行了一次 180° 的右转。之所以 $e \circ e$ 必须是右转，是因为引理 2.3 所考察的情形，其中 e 是 C中的一条边。为了确保在沿路径 $e \circ e \circ P \circ e'$从 e 遍历至 e'时始终处于 C 的内部，$e \circ e$ 的 180° 转向必须是右转。方向(e, P, e')的方向公式的其余部分可以类似地解释。

单调环 现在我们可以定义 (R3) 中所使用的严格单调环概念。若本质环 C 的所有边标签 $\ell_C(e)$ 均为非负，或其所有边标签 $\ell_C(e)$ 均为非正，则称其为单调 。设 C 为一个单调的本质环。若 C包含至少一个正边标签，则称 C 为递增。若 C 包含至少一个负边标签，则称 C 为递减。递减环和递增环统称为严格单调环。

直观地说，递增环就像一圈向下的楼梯，而递减环就像一圈向上的楼梯，因此无法绘制。 [2] 已证明，$(\mathcal{R}\ e^\star)$ 是

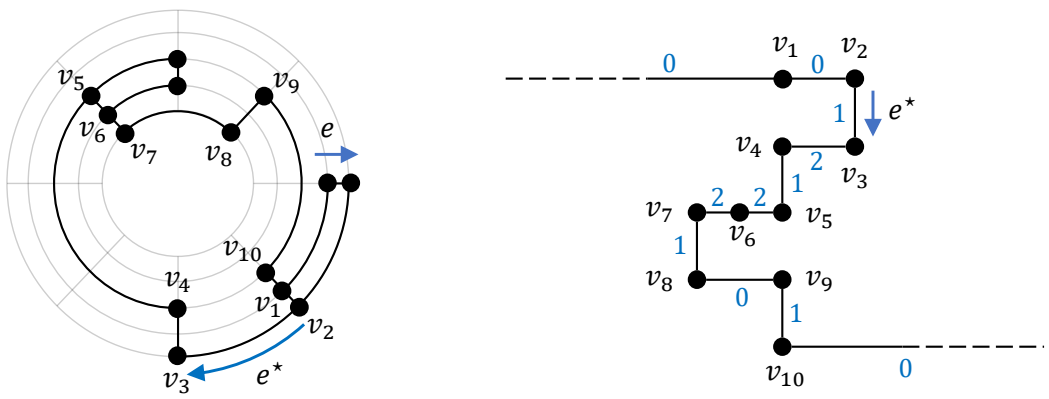

图 5. 将参考边改为 e 会得到一个严格单调环。

可绘制的，当且仅当其中不包含严格单调环。再次提醒，在整篇论文中，除非另有说明，我们都假设给定的正交径向表示已经满足（R1）和（R2）。

引理 2.4 ([2])。若正交径向表示$\mathcal{R}$具有固定参考边 $e^\star$，满足$\mathcal{N}_{\text{north}}(S) = \emptyset$ 对于包含 $e^\star$ 的水平线段 $S \in \mathcal{S}_{\text{h}}$，则当且仅当它不包含严格单调环时，该表示可绘制。

一般而言，$(\mathcal{R}\ e^\star)$ 是否可绘制取决于参考边 $e^\star$ 的选择。例如，在图 5 中，尽管$(\mathcal{R}\ e^\star)$ 可绘制，$(\mathcal{R}\ e)$ 却不可绘制，因为本质环 $C = (v_1, v_2, \ldots, v_{10})$ 是递增的。以 e 为参考边时，环 C 上的所有边标签均为非负，且至少有一个为正。

现在我们可以陈述主要结果了。

定 理 2 . 5. 对于给定的双连通简单图的正交径向表示 $\mathcal{R}$，该图有 n 个顶点，并且指定了一条参考边 $e^\star$，满足：对于包含 $e^\star$ 的水平线段 $S \in \mathcal{S}_{\text{h}}$， $= \emptyset\ S)\ _{\text{north}}(\{v28'\!>\!N_{\text{north}}(S) = \emptyset$ 。存在一个运行时间为 $O(n \log n)$ 的算法 $\mathcal{A}$，它输出 $(\mathcal{R}\ e^\star)$ 的绘图，或输出 （R $e^\star$） 的一个严格单调环。

上述定理改进了 [2] 提出的先前算法，其运行时间为 $O(n^2)$ 。如果算法 $\mathcal{A}$ 的输出是一个严格单调环，那么根据引理 2.4，该环可以证明绘图不存在。我们还将上述定理扩展到了参考边未固定的情形。

定 理 2 .6. 存在一个运行时间为 $O(n \log^2 n)$ 的算法 $\mathcal{A}$，用于判定一个有 n 个顶点的双连通简单图的正交径向表示 $\mathcal{R}$ 是否可绘制。如果 $\mathcal{R}$ 可绘制，则 $\mathcal{A}$ 还会计算出 $\mathcal{R}$ 的一个绘图。

定理 2.5 的证明见第 5 节，定理 2.6 的证明见第 6 节。

3. 通过良序列进行正交径向绘图

本节中，我们引入良序列的概念。良序列的存在使我们能够通过简单的贪心算法构造正交径向绘图。直观地说，良序列 $A = (S_1, S_2, \ldots, S_k)$ 是一个水平线段序列，它使我们能够依次安全地放置水平线段：S_1 绘制在圆 $r = k$ 上，S_2 绘制在圆 $r = k - 1$ 上，依此类推。

水平线段序列 令 $A = (S_1, S_2, \ldots, S_k)$ 为任意水平线段序列。一般而言，我们不要求 A 覆盖 S_h 中的所有水平线段。对于每个 $1 \le i \le k$，我们采用以下术语，其中 k 为该序列的长度 A。

— 令 G_i 为由 S_1、S_2、…、S_i 中的水平边，以及所有两个端点均位于 S_1、S_2、…、S_i 中的垂直边所诱导的 G 的子图。令 F_i 为 G_i 的中心面，并令 C_i 为 F_i 的面环。我们将 $\mathrm{N}_{\mathrm{south}}(S)$ 的概念扩展到水平线段序列，具体如下。令 $\mathrm{N}_{\mathrm{south}}(S_1, S_2, \ldots, S_i)$ 为满足 $y \in C_i$ 和 $x \notin C_i$ 的垂直边 $e = (x, y) \in E_{\mathrm{v}}$ 的集合。令 G_i^+ 为由 G_i 中的所有边以及边集 $\mathrm{N}_{\mathrm{south}}(S_1, S_2, \ldots, S_i)$ 所诱导的 G 的子图。令 F_i^+ 为 G_i^+ 的中心面，并令 C_i^+ 为 F_i^+ 的面环。

—

—

注意，对于每条垂直边 $(,)\mathcal{N}(,,\ldots,)$，南端点？在？中恰好出现一次。我们根据南端点？在？的循环排序中的位置，对边 $(,)\mathcal{N}(,,\ldots,)$ 进行循环排序。以图6中的图？为例。在该图中，有6条水平线段，如图6中的阴影部分所示：

$$S_1 = (v_{1,1}, v_{1,2}, v_{1,3}, v_{1,4}), \quad S_2 = (v_{2,1}, v_{2,2}, v_{2,3}), \quad S_3 = (v_{3,1}, v_{3,2}, v_{3,3}, v_{3,4}, v_{3,5}),$$
$$S_4 = (v_{4,1}, v_{4,2}, v_{4,3}), \quad S_5 = (v_{5,1}, v_{5,2}, v_{5,3}, v_{5,4}, v_{5,5}), \quad S_6 = (v_{6,1}, v_{6,2}).$$

对于序列 $(,,\ldots,)$，图6展示了图？和？，对所有？均如此。例如，对于？，我们有：

$$\mathcal{N}_{\mathrm{south}}(S_1, S_2) = ((v_{3,1}, v_{1,1})(v_{3,2}, v_{2,1}), (v_{3,4}, v_{2,3}), (v_{3,5}, v_{1,4})),$$
$$\mathcal{N}_{\mathrm{north}}(S_2) = ((v_{2,1}, v_{1,2}), (v_{2,2}, v_{1,3}))$$
$$C_2 = (v_{1,1}, v_{1,2}, v_{2,1}, v_{2,2}, v_{2,3}, v_{2,2}, v_{1,3}, v_{1,4}),$$
$$C_2^+ = (v_{1,1}, v_{3,1}, v_{1,1}, v_{1,2}, v_{2,1}, v_{3,2}, v_{2,1}, v_{2,2}, v_{2,3}, v_{3,4}, v_{2,3}, v_{2,2}, v_{1,3}, v_{1,4}, v_{3,5}, v_{1,4}).$$

这里，$\mathcal{N}_{\mathrm{south}}(S_1, S_2)$、$C_2$ 和 C_2^+ 是循环排序，而 $\mathcal{N}_{\mathrm{north}}(S_2)$ 是顺序排列，S_2 也是一条路径。

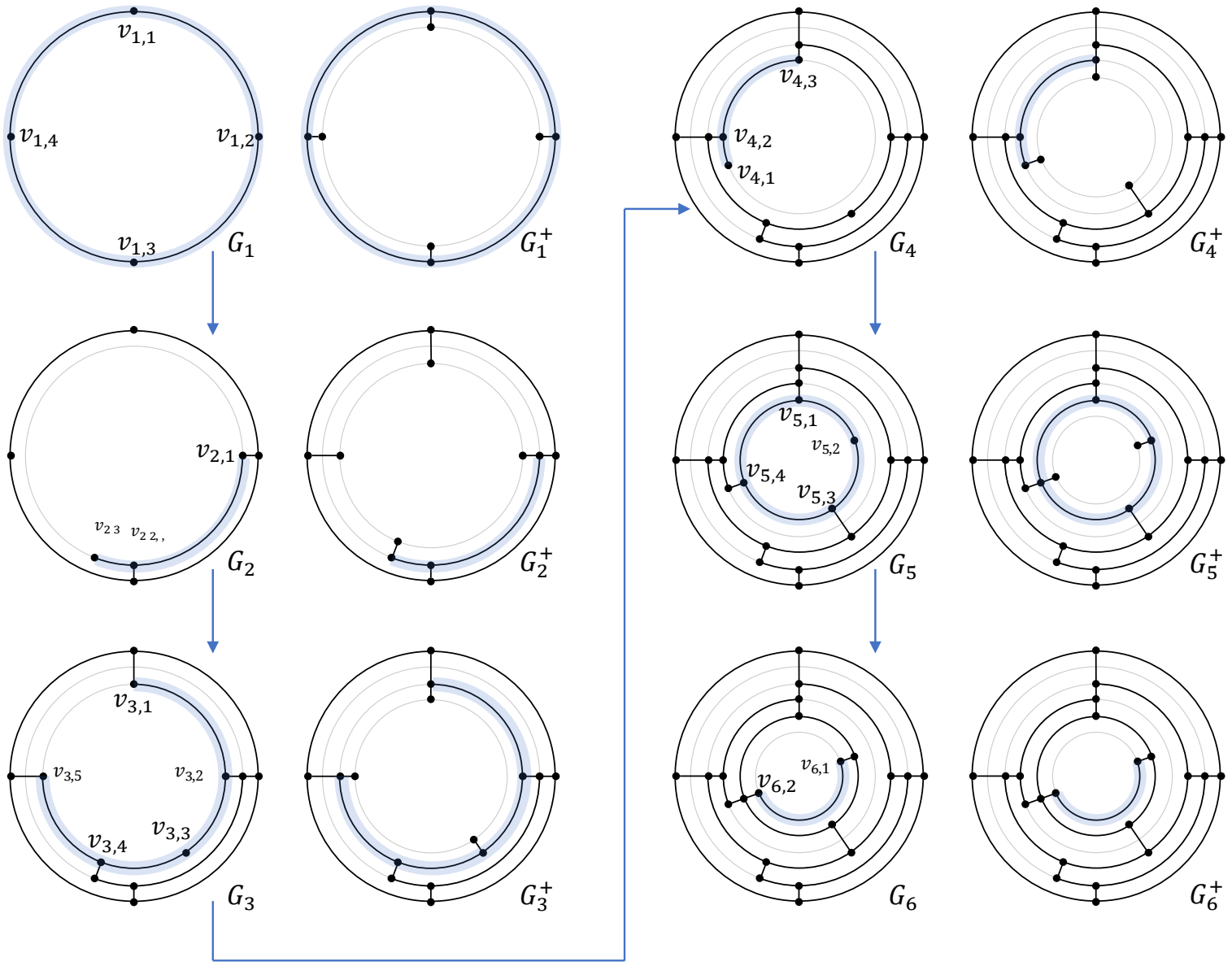

图6。 为良序列构造良好绘图。

良序列我们称水平线段序列 $A=(S_1,S_2,\ldots,S_k)$ 是良的，如果A 满足以下条件。

(S1) S_1 是外面 F_o 的面环的反向，即， $S_1=C_{F_o}$。(S2) 对于每个 $1<i\leq k$，$\mathcal{N}_{\text{north}}(S_i)$均满足以下要求。

- $\mathcal{N}_{\text{north}}(S_i)\neq\emptyset$。如果 S_i 是一条路径，那么 $\mathcal{N}_{\text{north}}(S_i)$是$\mathcal{N}_{\text{south}}(S_1,S_2,\ldots,S_{i-1})$的连续子
- 序列。如果 S_i 是一个环，那么 $\mathcal{N}_{\text{north}}(S_i)=\mathcal{N}_{\text{south}}(S_1,S_2,\ldots,S_{i-1})$。
-

显然，如果 $A=(S_1,S_2,\ldots,S_k)$ 是良好的，那么对每个 $1\leq i<k$，$(S_1,S_2,\ldots,S_i)$ 也同样良好。一般而言，对于给定的 $(\mathcal{R}\ e^\star)$，良序列可能并不存在。特别是，为满足 (S1)，环 C_{F_o} 必须是一条水平线段。图6所示的序列 $A=(S_1,S_2,\ldots,S_6)$是一个良序列。

良好绘图 在本文中，我们使用极坐标系，其中 (r, θ) 是笛卡尔坐标系中由 $x = r\cos\theta$ 和 $y = r\sin\theta$ 给出的点。始终有 $r \geq 0$。设 $A = (S_1, S_2, \ldots, S_k)$ 是由 k 条水平线段组成的良序列。为简化记号，也可将 $\mathrm{N}_{\mathrm{south}}(A) = \mathrm{N}_{\mathrm{south}}(S_1, S_2, \ldots, S_k)$ 写作。设 $(e_1, e_2, \ldots, e_s)$ 为 $\mathrm{N}_{\mathrm{south}}(A)$ 的循环排序，并对每个 $1 \leq j \leq s$，令 $e_j = (x_j, y_j)$，其中 s 是 $\mathrm{N}_{\mathrm{south}}(A)$ 的大小。若绘图满足以下性质，则称其为具有零弯折的正交径向绘图。

(D1) 对于每个 $1 \leq j \leq s$，绘图不会使用 $\{(r, \theta) | 0 \leq r < r_j$ 和 $\theta = \theta_j\}$ 中的任何点，其中我们用 (r_j, θ_j) 表示顶点 y_j 在绘图中的位置。

隐含要求良好绘图必须是平面的，并保留平面图 G_k 的组合嵌入。在图6中，对于每个 $1 \leq i \leq 6$，图 G_i 的绘图都是良序列 $(S_1, S_2, \ldots, S_i)$ 对应的良好绘图。在下面的引理中，我们将证明任何良好绘图都具有以下有利性质。

(D2) 绘图中由 $\theta_1, \theta_2, \ldots, \theta_s$ 给出的 $e_1, e_2, \ldots, e_s$ 的顺时针循环排序，与 $\mathcal{N}_{\mathrm{south}}(A)$ 给出的循环排序相同。

引理 3.1。如果 G_k 的正交径向绘图对应良序列 $A = (S_1, S_2, \ldots, S_k)$ 且满足 *(D1)*，那么该绘图也满足 *(D2)*。

证明。证明示意图见图7。假设 (D2) 不成立，那么可以找到三个指标 a、b 和 c，使得由它们的 θ 坐标确定的顺时针排序 (e_c, e_b, e_a) 与它们在 $\mathcal{N}_{\mathrm{south}}(A) = (e_1, e_2, \ldots, e_s)$ 中的循环排序 (e_a, e_b, e_c) 方向相反。

令 G_k^* 为将 $\mathcal{N}_{\mathrm{south}}(A) = (e_1, e_2, \ldots, e_s)$ 中所有边的南端点合并为一个顶点 v^* 后得到的图。通过将 v^* 放置在原点，并将 $\mathcal{N}_{\mathrm{south}}(A) = (e_1, e_2, \ldots, e_s)$ 中的所有边绘制为直线，扩展给定的 G_k 正交径向绘图，即可得到 G_k^* 的平面绘图。根据 (D1)，G_k^* 的绘图是无交叉的。假设 (D2) 不成立，我们将通过证明该绘图不可能无交叉来导出矛盾，因此 (D2) 必须成立。

对于任意的 $1 \leq i \leq s$ 和 $1 \leq j \leq s$，我们用 $P_{i,\,j}$ 表示从 e_i 出发并在 e_j 结束的 C_k^+ 的子路径。在 G_k^* 中，任何这样的路径都是一个环，因为它始于并终于同一个顶点 v^*。

考虑环 $P_{a,b}$ 在 G_k^* 中。我们关于 $\{e_a, e_b, e_c\}$ 的 θ 坐标的假设意味着，在上述 G_k^* 绘图中，e_c 位于环 $P_{a,b}$ 的内部。现在考虑路径 $P_{c,a}$，它从 e_c 出发并在 e_a 结束。令 v 为 $P_{a,b} - \{v^*\}$ 访问的第一个顶点。由于 $P_{c,a}$ 在 e_a 结束，这样的顶点必然存在。令 e 为 v 的一条关联边，$P_{c,a}$ 经由它进入 v。由于 C_k^+ 是 G_k^+ 的一个面环，v 在 C_k^+ 中的关联边的循环排序必须遵循由 $\mathrm{E}(v)$ 给出的逆时针序，因此 e 必须是环 $P_{a,b}$ 外部的一条边。

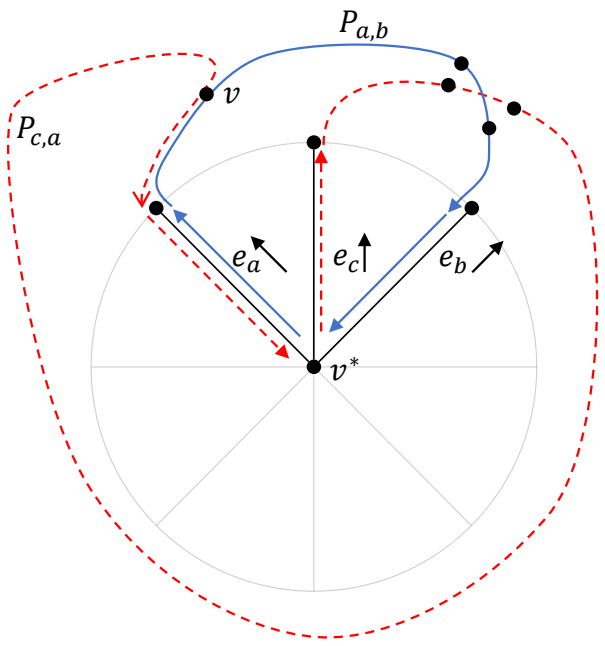

图7. 引理 3.1 证明示意图。

因此，$P_{c、a}$ 的一条边与 $P_{a、b}$ 的一条边必然相交；否则，$P_{c、a}$ 就无法从 $P_{a、b}$ 的内部到达 $P_{a、b}$ 的外部。■

我们给出一种高效算法，对于给定的良序列 $A=(S_1,\ S_2,\ \ldots,\ S_k)$，计算 G_k 的良好绘图。该算法的时间复杂度与 G_k的大小呈线性关系。对于 $G_k = G$这一特殊情形，这给出了一种线性时间算法，可计算实现给定 $(\mathcal{R}\ e^\star)$ 的正交径向绘图。

引理 3.2. 对于给定的良序列 $A=(S_1,\ S_2,\ \ldots,\ S_k)$，可以在 $O\left(\sum_{i=1}^{k}|S_i|\right)$ 时间内构造 G_k 的良好绘图。

证明。 通过对序列 A 的长度 k 进行归纳法来证明该引理。图6展示了证明中所述算法的示意图。

基础情形 对于 $k=1$ 的基础情形，可以如下构造 $G_1=S_1$ 的良好绘图。根据 (S1)，$S_1=C_{F_o}$ 是最外层本质环。令 $\overline{S_1}=(v_1,v_2,\ldots,v_t)$，其中 $t=|S_1|$ 是环 S_1 中顶点的数量。然后，我们可以将 $G_1=S_1$ 绘制在单位圆上，具体做法是对每个 $1\le j\le t$，将 v_j 放在点 $\left(1,-\frac{j}{t}\cdot 2\pi\right)$ 处。负号是因为 S_1 的方向为顺时针方向。由于需要计算这些坐标，绘图的构造需要 $O(|S_1|)$时间。该绘图满足条件 (D1)，因为没有使用任何满足 $0\le r<1$ 的点 (r,θ)。

归纳步骤 在归纳步骤中，假设我们已有 G_{k-1}的良好绘图，我们将通过耗费 $O(|S_k|)$时间，为 S_k中的顶点恰当地指定坐标，从而将该绘图扩展为 G_k的良好绘图。我们选择一个小于

r坐标的数值；这些坐标对应于给定绘图中 G_{k-1} 所有顶点的位置，因此圆 $r = r^*$ 严格包含在给定绘图的 G_{k-1} 的中心面 F_{k-1} 内。令 $S_k = (v_1, v_2, \ldots, v_t)$，其中 $t = |S_k|$ 是 S_k中的顶点数。我们将在圆 $r = r^*$上绘制 S_k 。

步骤 1：给定绘图中具有邻点的顶点设$(e_1, e_2, \ldots, e_s)$ 为$\mathcal{N}_{\text{south}}(S_1, S_2, \ldots, S_{k-1})$的循环排序，并对每个 $1 \le j \le s$，设 $e_j = (x_j, y_j)$，其中 s 是$\mathcal{N}_{\text{south}}(S_1, S_2, \ldots, S_{k-1})$的大小。我们用$(r_j, \theta_j)$表示顶点 y_j 在给定绘图中的位置。

根据 (S2)，$\mathrm{N}_{\text{north}}(S_k)$ 是 $\mathrm{N}_{\text{south}}(S_1, S_2, \ldots, S_{k-1})$ 的子集。对于 $\mathrm{N}_{\text{north}}(S_k)$ 中的每条垂直边 $e_j = (x_j, y_j) \in$，我们将坐标 (r^*, θ_j) 指派给 x_j。根据 (D1)，给定的绘图没有使用 $\{(r, \theta) \mid 0 \le r < r_j$和 $\theta = \theta_j\}$ 中的任何点，因此我们可以将 e_j 绘制为连接 x_j 和 y_j 的直线。

根据引理 3.1，(D2) 可由 (D1) 推出。根据 (D2)，对于我们已绘制的 S_k 中的顶点，也就是 S_k 中在 $\mathcal{N}_{\text{north}}(S_k)$ 中有关联边的顶点集合，其 θ 坐标的顺时针排序遵循水平线段 $S_k = (v_1, v_2, \ldots, v_t)$ 的排序。如果 S_k 是一个环，那么 $(v_1, v_2, \ldots, v_t)$ 可视为循环排序。

步骤 2：两个端点 对于 S_k 是路径的情况，我们按如下方式绘制$S_k = (v_1, v_2, \ldots, v_t)$的两个端点 v_1 和 v_t 。根据 (S2)，在此情况下，$\mathrm{N}_{\text{north}}(S_k)$ 是 $\mathrm{N}_{\text{south}}(S_1, S_2, \ldots, S_{k-1})$ 的一个连续子序列。设 j_1 和 j_2 为相应的索引，使该子序列从 e_{j_1} 开始，到 e_{j_2} 结束。令 $\epsilon = \min_{1 \le j \le s}(\theta_j - \theta_{j-1})/3$。如果 v_1 在 $\mathrm{N}_{\text{north}}(S_k)$ 中没有关联边，则将坐标 $(r^*, \theta_{j_1} + \epsilon)$ 赋给 v_1。同样，如果 v_t 在 $\mathrm{N}_{\text{north}}(S_k)$ 中没有关联边，则将坐标 $(r^*, \theta_{j_2} - \epsilon)$ 赋给 v_t。我们对 ϵ 的选择确保弧度范围 $[\theta_{j_2} - \epsilon, \theta_{j_1} + \epsilon]$ 不与任何 $e_j \in \mathrm{N}_{\text{south}}(S_1, S_2, \ldots, S_{k-1}) \setminus \mathrm{N}_{\text{north}}(S_k)$ 的 θ_j 重叠。

步骤 3：剩余顶点 我们按如下方式绘制 S_k 中剩余的顶点。令 $(v_a, \ldots, v_b)$ 为 S_k 中任意一个由尚未绘制的顶点组成的最长连续子序列。我们只需将这些顶点放置在圆 $r = r^*$ 上的 v_{a-1} 和 v_{b+1} 之间即可完成绘制。形式化地，令 θ_{west} 为 v_{a-1} 所在位置的 θ 坐标，令 θ_{east} 为 v_{b+1} 所在位置的 θ 坐标。对于每个 $a \le j \le b$，将 v_j 的坐标赋值为

$$\left(r^*, \theta_{\text{west}} - (j - a + 1) \cdot \frac{\theta_{\text{east}} - \theta_{\text{west}}}{b - a + 2}\right).$$

一般来说，当 S_k 是一个环且 $|\mathcal{N}_{\text{north}}(S_k)| = 1$ 时，可能出现 $v_{a-1} = v_{b+1}$；此时，$v_{a-1} = v_{b+1}$ 是 S_k 中唯一与 $\mathcal{N}_{\text{north}}(S_k)$ 中唯一一条边关联的顶点。请注意，当底层图 G 是双连通时，不会出现这种情况。对于这种情况，在上述计算中，我们应将 v_{b+1} 位置的θ坐标设为 v_{a-1} 位置的 θ坐标减去 2π 。

绘图的有效性 对于我们构造的 G_k 的绘图，我们验证条件 (D1) 得到满足。考虑 G_k 中任意一个与 $\mathcal{N}_{\text{south}}(S_1, S_2, \ldots, S_k)$ 中的边关联的顶点 v。假设它在我们的绘图中的坐标为 (r_v, θ_v)。为证明条件 (D1) 得到满足，只需验证我们的绘图未使用任何满足 $0 \le r < r_v$ 和 $\theta = \theta_v$ 的点 (r, θ)。对于 v 属于 S_k的情况，我们有 $r_v = r^*$，而我们对 r^* 的选择意味着绘图未使用任何 r值小于 r^*的点。

现在假设 v 不属于 S_k。那么，对于某个 $e_j =$， $v = y_j(x_j, y_j) \in \mathrm{N}_{\text{south}}(S_1, S_2, \ldots, S_{k-1}) \setminus \mathrm{N}_{\text{north}}(S_k)$。注意，只有当 S_k 是一条路径时，这种情况才可能发生。根据归纳假设，给定的 G_{k-1} 绘图不会使用任何满足 $0 \le r < r_v$ 且 $\theta = \theta_v$ 的点 (r, θ)，因此我们只需验证，在绘制水平线段 S_k时，用于绘制 S_k 的圆弧不会与直线 $\{(r, \theta) \mid 0 \le r < r_v$ 和 $\theta = \theta_v\}$ 相交。事实上，这段圆弧的 θ 坐标范围限定在 $[\theta_{j_2} - \epsilon, \theta_{j_1} + \epsilon]$ 内，而我们在步骤 2 中对 ϵ 的选择确保该范围不会与 θ_j 重叠；对于任何 $e_j \in \mathrm{N}_{\text{south}}(S_1, S_2, \ldots, S_{k-1}) \setminus \mathrm{N}_{\text{north}}(S_k)$，情况皆如此，因此不可能发生这种交叉。

运行时间分析 G_1的良好绘图可以在 $O(|S_1|)$时间内构造出来。给定 G_{i-1}的良好绘图，可以在 $O(|S_i|)$时间内构造出 G_i的良好绘图。因此，给定良序列 $A = (S_1, S_2, \ldots\ldots, S_k)$，可以在 $O\left(\sum_{i=1}^{k} |S_i|\right)$时间内构造出 G_k的良好绘图。■

备注 引理3.2中的算法所计算出的绘图使用 k 层（即同心圆）。可以修改该算法，使输出绘图所用的层数最少。其思路很简单：在归纳步骤中处理新的水平线段 S_i 时，不再总是创建新层，而是将 S_i 绘制在最外层的可用层上。更形式化地，对于 S_i，我们定义其层编号 ℓ_i 如下。在基础情形中，令 $\ell_i = 0$。在归纳步骤中，令 $\ell_i = \ell_j + 1$，其中 j 是使下述条件成立的最大指标：存在一条垂直边，其南端点位于 S_i，北端点位于 S_j。根据层编号的定义，任何正交径向绘图至少需要 $\ell^* = 1 + \max_{i \in [k]} \ell_i$层。通过修改引理3.2中的算法，可以构造一个使用 ℓ^* 层的正交径向绘图，具体做法是将 S_i 绘制在圆 $r = 1 - \ell_i/\ell^*$上；这是能够绘制 S_i 的最外层。

4. 构造一个良好序列

给定一个带有参考边 $e^\star$ 的正交径向表示$\mathcal{R}$ of G，其中水平线段 $S^\star \in \mathcal{S}_{\text{h}}$ 满足 $e^\star \in S^\star$ $\mathcal{N}_{\text{north}}(S^\star) = \emptyset$，本节将介绍一种实现以下目标的算法。如果$(\mathcal{R}\ e^\star)$ 可绘图，该算法会向$\mathcal{R}$ 添加虚拟边，从而得到一个良序列 $A = (S_1, S_2, \ldots, S_k)$，使得 $G_k = G$ 存在且能够高效计算；随后可使用上一节中的绘图算法构造$(\mathcal{R}\ e^\star)$ 的绘图。

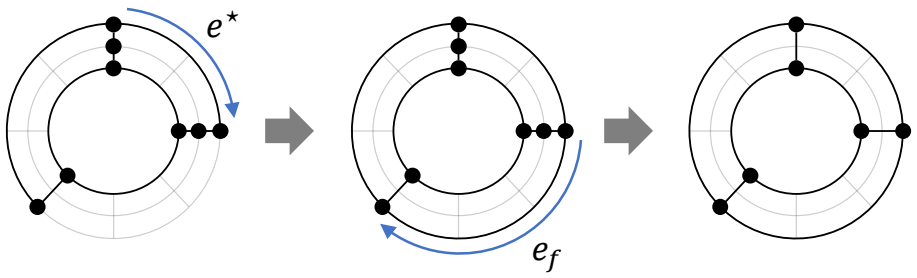

图 8. 预处理步骤。

如果$(\mathcal{R}\ e^\star)$ 不可绘图，该算法会返回 G中的一个严格单调环，以证明$(\mathcal{R}\ e^\star)$ 不可绘图。请回忆，我们要求在$(\mathcal{R}\ e^\star)$的绘图中，将 $e^\star$ 放置在最外侧圆弧上。此类绘图存在的必要条件是$\mathcal{N}_{\text{north}}(S^\star) = \emptyset$。

预处理步骤 1：外面 为确保存在非空良序列，根据 (S1)，要求 $S = C_{F_o}$ 是$\mathcal{S}_{\text{h}}$中的一条水平线段。如果不满足这一要求，我们将在$\mathcal{R}$ 中添加虚拟边以满足该要求。设 $S^\star \in \mathcal{S}_{\text{h}}$ 为包含参考边 $e^\star$的水平线段，并且有 $\mathcal{N}_{\text{north}}(S^\star) = \emptyset$。我们添加一条连接 $S^\star$两个端点的虚拟水平边 e_f，这样 $S^\star$ 与 e_f共同构成外面的新轮廓，并且是$\mathcal{S}_{\text{h}}$中的一条水平线段。图 8 展示了一个示例。

请注意，一般而言，只要添加虚拟边不会破坏 (R1) 或 (R2)，添加虚拟边就不会改变原图中任何已有本质环 C 上的任何边 e 的边标签 $\ell_C(e)$。这是因为，在计算$\ell_C(e)$时， $\ell_C(e)$ 的计算不受参考路径 P 的选择影响，而且总存在一条原图中已有且不包含任何虚拟边的参考路径 P。

预处理步骤 2：平滑。由于我们的目标是找到一个良序列 $A = (S_1, S_2, \ldots, S_k)$，使得 $G_k = G$，因此必须满足 A 包含 $\mathcal{S}_{\text{h}}$ 中的所有水平线段，并且每个顶点 $v \in V$ 都与 $\mathcal{S}_{\text{h}}$中的某条水平线段关联。由于我们假设底层图是双连通的，顶点 $v \in V$ 不与任何水平线段关联的唯一可能情形是：$\deg(v) = 2$，且 v 与两条垂直边 (u, v) 和 (v, w) 关联。我们可以通过平滑来移除任何此类顶点 v，也就是说，将 (u, v) 和 (v, w) 替换为一条垂直边 (u, w)。图 8 展示了一个示例。显然，平滑不会影响 (R $e^\star$) 的可绘性，而且平滑后的图的绘图可以轻松转换为平滑前的图的绘图。从现在起，我们假设每个顶点 $v \in V$ 都与 $\mathcal{S}_{\text{h}}$中的某条水平线段关联，因此我们只需找到一个覆盖所有水平线段的良序列。

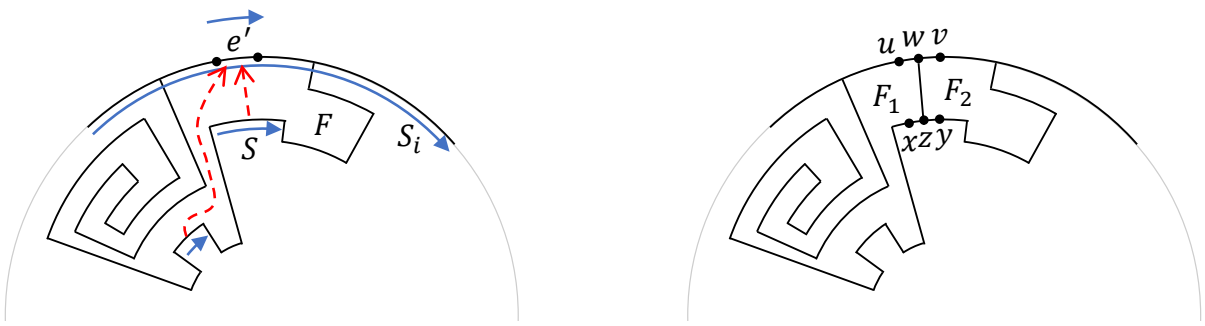

图 9. 在正则面中添加一条虚拟竖直边。

添加虚拟边的资格条件 令 $S \in \mathcal{S}_h$，使得 $\mathcal{N}_{north}(S) = \emptyset$ 且 $S \neq C_{F_o}$。这样的水平线段 S 永远无法加入良序列，因为 (S2) 要求 $\mathcal{N}_{north}(S)$ 非空。为解决这一问题，我们考虑以下将虚拟竖直边关联到此类水平线段 S的资格条件。

设 $A = (S_1, S_2, \ldots, S_k)$ 是一个良序列。设 $S \notin A$ 是一条水平线段，使得 $\mathrm{N}_{north}(S) = \emptyset$。设 F 是满足 S 为 C_F 的子路径的面。如果存在一条边 $e' \in C_F$，使得对于某个 $1 \leq i \leq k$，$e' \in S_i$，且满足以下条件之一：沿着环 C_F，旋转$(e' \circ \cdots \circ S) = 2$，或旋转$(S \circ \cdots \circ e') = 2$，那么我们称 S 可以添加虚拟边。直观地说，条件 $\mathrm{N}_{north}(S) = \emptyset$ 确保添加虚拟边后，可以立即将 S 追加到序列 A 的末尾。

对于 F 为正则面的情况，旋转$(C_F) = 4$，因此旋转$(e' \circ \cdots \circ S) = 2$ 当且仅当旋转$(S \circ \cdots \circ e') = 2$。我们也允许 F 为中心面，在这种情况下，旋转 $(e' \circ \cdots \circ S) = 2$ 和旋转 $(S \circ \cdots \circ e') = 2$ 中至多有一个成立。

我们指出，如果 S 相对于当前良序列 A符合添加虚拟边的条件，那么我们可以添加一条虚拟竖直边 $e_f = (z, w)$，连接 S 中水平边 (x, y) 的中点 z 与水平边 $e' = (u, v)$ 的中点 w。图 9 展示了一个示例。图中，F 是一个正则面，沿着 F 的轮廓有两条水平线段因 $e' \in S_i$而符合添加虚拟边的条件。

我们指出，添加 $e_f = (z, w)$不会破坏 (R1) 和 (R2)。验证 (R1) 很直接。对于 rotation$(S \circ \cdots \circ e') = 2$ 的情况，我们验证 (R2)；另一种情况 rotation$(e' \circ \cdots \circ S) = 2$ 类似。添加 e_f会将 F 分解为两个新面 F_1和 F_2。设 F_1 为面环包含子路径 (u, w, z, x) 的那个面，设 F_2 为面环包含子路径 (y, z, w, v) 的那个面。

我们首先考虑 F_1。F_1 的面环的旋转等于 rotation$(S \circ \cdots \circ e') = 2$ 加上路径 (u, w, z, x) 的旋转，后者也为 2，因为 (u, w, z, x) 由两个右转组成。因此，该面环的旋转为 4，这意味着 F_1 是一个正则面。现在考虑另一个面 F_2。在添加 e_f 之前，F_2 的面环的旋转与 rotation(C_F) 相同。原因是，从 y 到 v 的子路径在 C_F 和 C_{F_2} 中的旋转都为 2，因为我们假设 rotation$(S \circ \cdots \circ e') = 2$。如果 F 是一个正则面，那么旋转之和

对于 F 和 F_2 都是 4，因此 F_2 也是一个正则面。如果 F 是一个中心面，那么对于 F 和 F_2，旋转之和都是 0，因此 F_2 也是一个中心面。

以图10为例：中心面 F 的轮廓中有四条水平线段，由于 $e' \in S_i$ 而符合添加虚拟竖直边的条件。图左侧突出显示的两条水平线段，由于沿环 C_F 旋转$(e' \circ \cdots \circ S) = 2$而符合条件。图右侧突出显示的两条水平线段，由于沿环 C_F 旋转$(S \circ \cdots \circ e') = 2$ 而符合条件。

一种贪心算法 假设 $C_{F_o} \in \mathcal{S}_h$ 且每个 $v \in V$ 都与一条水平线段关联，我们构造良序列的算法如下。我们从平凡的良序列 $A = (S_1)$ 开始，其中 $S = C_{F_o}$，然后反复执行以下两种操作，直到无法再执行为止。

- 找到一条水平线段 $S \in \mathcal{S}_h$，使得将 S 追加到当前序列A 的末尾后得到一个良序列，然后通过将 S 追加到 A 的末尾来扩展 A。找到一条相对于当前良序列 A 符合添加虚拟边条件的水平线段 $S \in \mathcal{S}_h$，然后如上所述，添加一条与 S 关联的虚拟竖直边。

该算法有两种可能的结果。如果我们得到一个覆盖所有水平线段$\mathcal{S}_h$的良序列，那么可以使用引理3.2计算$(\mathcal{R}\ e^\star)$的绘图。否则，算法会停止，并得到一个未覆盖所有水平线段$\mathcal{S}_h$的良序列；此时无法再取得进展。在下一节中，我们将证明可以在原图 G 中找到一个严格单调环。

一种直接的贪心算法实现会在每一步检查所有水平线段，因此需要 $O(n^2)$ 时间。在下面的引理中，我们给出一种更高效的实现，只需 $O(n \log\ n)$ 时间。

引 L I 义 4 .1. 贪心算法可以实现为在 $O(n \log\ n)$ 时间内运行 .

证 明。设 $A = (S_1, S_2, \ldots, S_k)$表示当前的良序列，其初始为空集 $\emptyset$。算法执行期间，我们将循环排序$\mathrm{N}_{\mathrm{south}}(S_1, S_2, \ldots, S_k)$维护为一个循环双向链表。每当将一条路径 $S \in \mathrm{S_h}$ 插入 A时，就通过用$\mathrm{N}_{\mathrm{south}}(S)$替换$\mathrm{N}_{\mathrm{south}}(S_1, S_2, \ldots, S_k)$中的连续子序列$\mathrm{N}_{\mathrm{north}}(S)$来更新该循环双向链表。每当将一个环 $S \in \mathrm{S_h}$插入 A时，有$\mathrm{N}_{\mathrm{south}}(S_1, S_2, \ldots, S_k) = \mathrm{N}_{\mathrm{north}}(S)$，因此将该循环双向链表更新为$\mathrm{N}_{\mathrm{south}}(S)$的循环排序。

水平线段 在整个算法过程中，我们维护一个集合W，其中包含所有水平线段 $S \in \mathrm{S_h}$，使得将 S 添加到 A 后会得到一个良序列。如果 S 是一条路径，那么当 $\mathrm{N}_{\mathrm{north}}(S)$ 是 $\mathrm{N}_{\mathrm{south}}(S_1, S_2, \ldots, S_k)$ 的连续子序列时，就可以将 S 添加到 A。如果 S是一个环，那么当 $\mathrm{N}_{\mathrm{north}}(S) = \mathrm{N}_{\mathrm{south}}(S_1, S_2, \ldots, S_k)$ 时，就可以将 S添加到 A。我们的目标是设计一种合适的

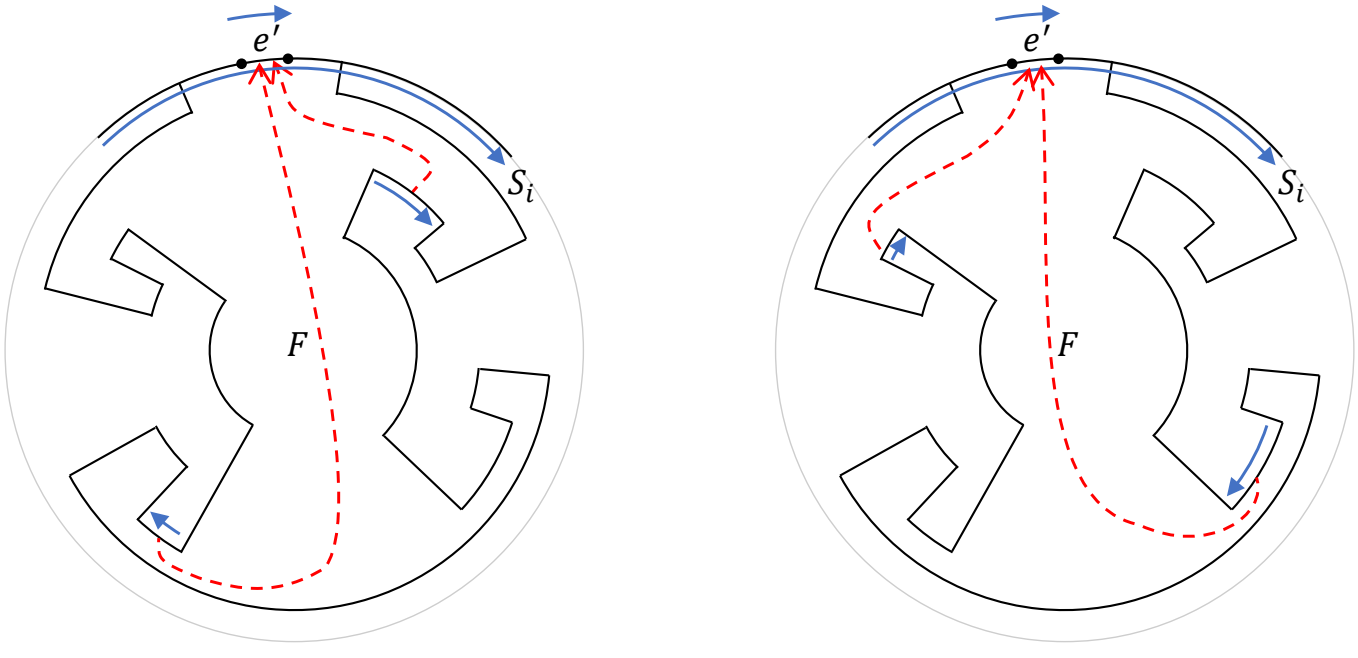

图10。 中心面的轮廓中的合格水平线段。

用于高效检测可添加到 A 的水平线段 S 的数据结构和算法（如果存在这样的线段）。

集合 W 初始化为 $W=\{C_{F_o}\}$，并按如下方式维护。首先，考虑任意满足 $|\mathrm{N}_{\mathrm{north}}(S)|=1$ 的 $S\in \mathrm{S_h}$。令 $\mathrm{N}_{\mathrm{north}}(S)=\{e\}$。如果 S 是一条路径，那么一旦 e 出现在 $\mathrm{N}_{\mathrm{south}}(S_1,S_2,\ldots,S_k)$ 中，我们就将 S 加入 W。如果 S 是一个环，那么一旦 $\mathrm{N}_{\mathrm{south}}(S_1,S_2,\ldots,S_k)=\{e\}$，我们就将 S 加入 W。注意，当 G 是双连通的时，S 是一个满足 $|\mathrm{N}_{\mathrm{north}}(S)|=1$ 的环这一情形不可能发生。

接下来，考虑任意满足 $|\mathrm{N}_{\mathrm{north}}(S)|\geq 2$ 的 $S\in \mathrm{S_h}$。对于任意两条垂直边 e 和 e'，若在排序 $\mathrm{N}_{\mathrm{north}}(S)$ 中 e' 紧接在 e 之后，则我们维护一个指示函数 $X_{e,\,e'}\in\{0,1\}$，使得当 e' 也在 $\mathrm{N}_{\mathrm{south}}(S_1,S_2,\ldots,S_k)$ 中紧接在 e 之后时，$X_{e,\,e'}=1$。初始时，所有 $X_{e,\,e'}$ 均设为 0。每次更新 $\mathrm{N}_{\mathrm{south}}(S_1,S_2,\ldots,S_k)$ 时，我们都会检查并更新所有可能受影响的边所对应的 $X_{e,\,e'}$。例如，如果从 $\mathrm{N}_{\mathrm{south}}(S_1,S_2,\ldots,S_k)$ 中删除一条边 $\tilde{e}$，我们会检查在 $\mathrm{N}_{\mathrm{south}}(S_1,S_2,\ldots,S_k)$ 中紧邻 $\tilde{e}$ 之前和之后的两条边，是否属于同一水平线段 S 对应的 $\mathrm{N}_{\mathrm{north}}(S)$。如果答案为是，则我们将 S 对应的指示函数 $X_{e,\,e'}=1$ 设为 1。根据 (S2)，当且仅当 S 的每个指示函数的值都为 1 时，才能将其插入 A。因此，我们可以通过检查其所有指示函数 $X_{e,\,e'}$ 的总和 X_S，来判断 S 是否应加入 W。每当我们更新 S 对应的指示函数 $X_{e,\,e'}$ 的值时，也会更新并检查这个总和 X_S。

上述数据结构和算法的耗时为 $O(n)$。每次将水平线段 S 插入良序列 $A=(S_1,\ S_2,\ \ldots\ldots,\ S_k)$ 时，都会对循环双向链表 $\mathcal{N}_{\mathrm{south}}(S_1,\ S_2,\ \ldots\ldots,\ S_k)$执行若干次插入和删除操作，而每次更新都会产生 $O(1)$ 次操作。由于更新次数与 $|E_{\mathrm{v}}|=O(n)$呈线性关系，因此总时间复杂度为 $O(n)$。

虚拟边 接下来，我们考虑添加虚拟垂直边的任务。每当将水平线段 $S' \in \mathcal{S}_{\mathsf{h}}$ 插入良序列 $A = (S_1, S_2, \ldots\ldots, S_k)$时，我们都会检查每条边$e' \in S'$，以判断 e' 是否会使某条水平线段 $S \in \mathcal{S}_{\mathsf{h}}$ 符合添加虚拟边的条件（相对于 $(S_1, S_2, \ldots\ldots, S_k, S')$）。对于所有此类 S，我们添加一条与 S 关联的虚拟边 e_f，然后将 S添加到 W，因为添加 e_f会使插入 S后得到的序列成为良序列。

在接下来的讨论中，我们固定一条边 $e' \in S'$，并考虑寻找一条水平线段 $S \in \mathsf{S}_{\mathsf{h}}$ 的任务：如果存在这样一条水平线段 S，则根据 e'，它可以相对于 $(S_1, S_2, \ldots, S_k, S')$ 添加一条虚拟边。设 F 为 $e' \in C_F$ 所在的面。我们只需检查所有满足以下条件的 $S \in \mathsf{S}_{\mathsf{h}}$：$S$ 是 C_F 的子路径，且 $\mathrm{N}_{\mathsf{north}}(S) = \emptyset$。找到这样的 S 并添加一条与 S 相 incident 的虚拟边 e_f 后，面 F 被分为两个面 F_1 和 F_2，边 e' 也被分为两条边 e'_1 和 e'_2，其中 $e'_1 \in C_{F_1}$ 和 $e'_2 \in C_{F_2}$。我们将递归地对 e'_1 和 e'_2 应用该算法。对所有 $e' \in S'$ 应用该算法后，我们可以确保在这一递归过程结束时，无法再添加虚拟边。

针对上述任务，一种直接的算法是检查所有 $S \in \mathcal{S}_{\mathsf{h}}$，使得$S$ 是 C_F 的子路径，并且$\mathcal{N}_{\mathsf{north}}(S) = \emptyset$。在最坏情况下，这种方法需要 $O(n)$ 时间，才能找出一条符合添加虚拟边条件的水平线段 $S \in \mathcal{S}_{\mathsf{h}}$，代价较高。下面我们将介绍一种更高效的算法和数据结构。

正则面 我们首先关注面 F 具有 $e' \in C_F$ 且为正则面的情形。如前所述，我们的任务是找到一条水平线段 $S \in \mathsf{S}_{\mathsf{h}}$，满足 $\mathrm{N}_{\mathsf{north}}(S) = \emptyset$，使得$S$ 是 C_F 的子路径，并且沿环 C_F，$\mathrm{rotation}(e' \circ \cdots \circ S) = 2$ 或 $\mathrm{rotation}(S \circ \cdots \circ e') = 2$，如果存在这样的 S。由于 F是正则面，$\mathrm{rotation}(C_F) = 4$，因此 $\mathrm{rotation}(e' \circ \cdots \circ S) = 2$与 $\mathrm{rotation}(S \circ \cdots \circ e') = 2$ 等价。

我们为面 F 维护以下数据结构，该数据结构包含两个部分：数据结构的第一部分是由边 $C_F = (e_1, e_2, \ldots, e_s)$构成的循环双向链表，其中 s 是 C_F的边数。为了便于计算 C_F的子路径的旋转，我们针对每个$1 \le i \le s$计算并存储 $r_i =$旋转$(e_1 \circ \cdots \circ e_i)$的值。在数据结构的这一部分中，我们将所有满足以下条件的 $S \in \mathsf{S}_{\mathsf{h}}$ 组织成一个桶数组 $B_{\min_i r_i}, \ldots, B_{\max_i r_i}$：$S$ 是 C_F的子路径，且 $\mathrm{N}_{\mathsf{north}}(S) = \emptyset$。如果从 e_1 到 S 的第一条边 e_S 的旋转为 j，则将 S添加到桶 B_j 中。每个桶中的水平线段均按照 e_S 在$C_F = (e_1, e_2, \ldots, e_s)$中的索引进行排序，并组织成双向链表。

查询 我们证明，给定该数据结构，我们可以在 O（1） 时间内找到一条符合添加虚拟边条件的水平线段$S \in \mathcal{S}_{\mathsf{h}}$，前提是存在这样的水平线段。

从 e_i到 e_j的旋转在 $1 \le i \le j \le s$时为 $r_j - r_i$，在 $1 \le j < i \le s$时为 $r_j - r_i + 4$。令 i 为满足 $e' = e_i$的索引。我们的目标是搜索一条满足以下条件的水平线段 S：

旋转$(e' \circ \cdots \circ S) = 2$，因此我们可以将搜索空间限制为满足 $r_j = r_i + 2$ 或 $r_j = r_i - 2$的 e_j，这意味着只需检查两个桶 B_{r_i-2} 和 B_{r_i+2}。

- 考虑桶 B_{r_i-2}，因为对于情形 $1 \le j < i \le s$，从 e_i 到 e_j的旋转为 2，当且仅当 $r_j - r_i + 4 = 2$，这等价于 $r_j = r_i - 2$。因此，这个桶的搜索空间是范围 $[1, i-1]$内的任意索引，所以我们只需检查链表 B_{r_i-2} 的第一个元素，看看其索引是否位于范围 $[1, i-1]$内。同理，对于桶 B_{r_i+2}，我们只需检查链表的最后一个元素，看看其索引是否位于范围 $[i+1, s]$内。

-

搜索可以在 $O(1)$ 时间内完成。

更新 如果找到了所需的水平线段 S，那么如前所述，面 F 将被分成两个面 F_1 和 F_2。如果我们从头开始为这两个面重建上述数据结构，那么最坏情况下重建需要 $O(n)$ 时间，这是我们无法承受的。这里的一个关键观察是，F_1 和 F_2 都可以看作是将 F中旋转为 2 的某个子路径替换为一条旋转为 2 的新路径后得到的结果，这意味着我们仍然可以在 F_1和 F_2中复用相同的旋转值$\{r_i\}$。对于分割后仍属于同一个面的任意两条边 $e_{i'}$和 $e_{j'}$，新面中从 $e_{i'}$ 到 $e_{j'}$ 的旋转仍可使用相同的公式计算，其中 $r_{i'}$ 和 $r_{j'}$ 是相对于旧面 F定义的。由于无需重新计算旋转值，因此可以在 $O(1)$ 时间内，根据给定的循环双向链表 $C_F = (e_1, e_2, \ldots, e_s)$ 计算出两个循环双向链表 C_{F_1} 和 C_{F_2}。

接下来，我们考虑数据结构的第二部分。我们证明，可以在 $O(\min\{|C_{F_1}|, |C_{F_2}|\})$ 时间内计算出 F_1 和 F_2 的两个桶数组，因此我们可以将这部分开销计入较小 的面中的边，每条边的开销为 $O(1)$。由于每条边总共会被计入 $O(\log n)$ 次，更新数据结构所花费的总时间上界为 $O(n \log n)$，如所需。

在 $O(\min\{|C_{F_1}|, |C_{F_2}|\})$时间内，我们可以判断 $|C_{F_1}| \ge |C_{F_2}|$是否成立。不失一般性，假设 $|C_{F_1}| \ge |C_{F_2}|$。我们可以在 $O(|C_{F_2}|) = O(\min\{|C_{F_1}|, |C_{F_2}|\})$时间内，从头开始构建 F_2 的桶数组。通过逐个从链表中删除 F_2 中的水平线段，可以根据给定的 F 桶数组得到 F_1 的桶数组，所需时间为 $O(|C_{F_2}|) = O(\min\{|C_{F_1}|, |C_{F_2}|\})$。

中心面 现在考虑剩余的情况，其中面 F 与 $e' \in C_F$ 是中心面。在这种情况下，两个条件 $\text{rotation}(e' \circ \cdots \circ S) = 2$ 和 $\text{rotation}(S \circ \cdots \circ e') = 2$ 并不等价，因为 $\text{rotation}(C_F) = 0$。然而，我们仍然可以通过考虑这两个条件，采用相同的方法来寻找合格水平线段。

具体来说，这里我们希望找到 S 使得以下任一条件成立：旋转$(e' \circ \cdots \circ S) = 2$ 或 旋转$(S \circ \cdots \circ e') = 2$，来自所有 $S \in \mathcal{S}_{\text{h}}$ 中满足 $\mathcal{N}_{\text{north}}(S) = \emptyset$ 且 S是 C_F的子路径的集合。同样，记作 $C_F = (e_1, e_2, \ldots, e_s)$，令 $e_i = e_{i'}$，并令 e_j为 S中的一条边。那么从 e_i到 e_j的旋转

对于所有 i 和 j，均为 $r_j - r_i$，因此我们只需考虑满足 $r_j = r_i + 2$ 或 $r_j = r_i - 2$ 的 e_j。两个桶 B_{r_i-2} 和 B_{r_i+2} 中的任何水平线段都符合条件，因此搜索仍可在 $O(1)$ 时间内完成。

插入虚拟边后，面 F 将被分成两个面 F_1 和 F_2。与正则面情形不同，我们无法为 F_1 和 F_2重复使用 r值。在这里，两个新面中只有一个$F^* \in \{F_1, F_2\}$ 可以被视为将 F 中旋转为 2 的子路径替换为一条旋转为 2 的新路径所得的结果，因此 F^* 的面环旋转仍为 0，F^* 将成为新的中心面，并且仍可将为 F 计算出的旧 r值用于 F^*。对于另一个新面 $F' \in \{F_1, F_2> \setminus F^*\}$，它是将 F 中旋转为 -2 的子路径替换为一条旋转为 2 的新路径所得的结果，因此 F' 的面环旋转将为 4，F' 将成为正则面，并且无法将为 F 计算出的旧 r值用于 F'。

为了解决上述问题，我们直接从头构建 F' 的数据结构，所需时间与 F'的面环中的边数成线性关系。整个算法中重建的总成本上界为 $O(s) = O(n)$，其中 s 是原图 G中 C_{F_c} 的边数。■

5. 提取严格单调环

本节考虑这样一种情形：上一节中的贪心算法以良序列 $A = (S_1, S_2, \ldots\ldots, S_k)$ 结束，但该序列未覆盖所有水平线段的集合$\mathcal{S}_h$。我们的目标是证明，在这种情况下，可以在 $O(n)$ 时间内计算出原图 G 的一个严格单调环。

我们先介绍本节中将使用的术语。给定一个大小为 k的良序列 $A = (S_1, S_2, \ldots\ldots, S_k)$，令 $(e_1, e_2, \ldots\ldots, e_s)$为$\mathrm{N}_{\mathrm{south}}(A)$的循环排序；对于每个 $1 \le j \le s$，令 $e_j = (x_j, y_j)$，其中 s 是$\mathrm{N}_{\mathrm{south}}(A)$的大小。我们用 $\tilde{G}$表示运行贪心算法后得到的图。也就是说，$\tilde{G}$ 是原图 G 加上贪心算法期间添加的所有虚拟边。G_k 和 G_k^+ 都视为 $\tilde{G}$的子图。对于每个 $1 \le i \le s$，我们用$F_{i,\ i+1}$ 表示 $\tilde{G}$ 中唯一的面 F，使得 C_F 同时包含 e_i和 e_{i+1}。注意，由于 $(e_1, e_2, \ldots\ldots, e_s)$是一个循环排序，因此 $v_{s+1} = v_1$。由于我们假设G是双连通的，因此不可能有 $s = |\mathrm{N}_{\mathrm{south}}(A)| = 1$。我们假设 $A = (S_1, S_2, \underline{\ldots\ldots}, S_k)$和 $\tilde{G}$是贪心算法运行结束后的结果，即 A 无法再扩展为更长的良序列，也无法再向 G 中添加虚拟边。

面类型 考虑面 $F_{i,\ i+1}$，其中 $1 \le i \le s$。我们将 $P_{i \leftarrow i+1}$ 定义为C_{F_i}中从 e_{i+1}开始并在 e_i结束的子路径$_{,\ i+1}$。记作 $P_{i \rightarrow \overline{i+1}} = P_{i \leftarrow i+1}$。我们用 $Z_{i \leftarrow i+1} = (z_1, z_2, \overline{\ldots\ldots})$ 表示这样的数字串，使得 z_l 是 $P_{i \leftarrow i+1}$ 中由前 l 条边组成的子路径的旋转。同样地，令 $Z_{i \rightarrow i+1} = (z_1, z_2, \ldots\ldots)$为这样的数字串，使得$z_l$ 是 $P_{i \rightarrow i+1}$ 中由前 l 条边组成的子路径的旋转。回忆一下，我们定义

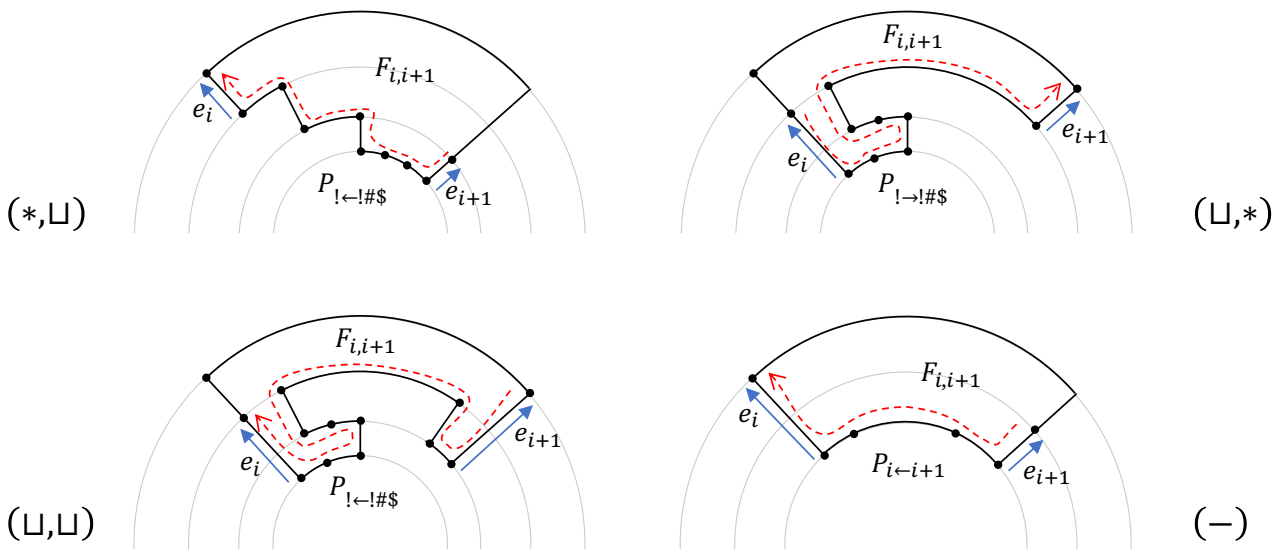

图11. 面类型 $(*, \sqcup)$, $(\sqcup, *)$, $(\sqcup, \sqcup)$, and $(-)$.

旋转（P） $= 0$ 对于任意单边路径 P。我们定义类型 （$*$， ⊔）、（⊔， $*$）、（⊔， ⊔）和（−）如下所述。

- $F_{i,i+1}$ 的类型为（$*, \sqcup$），当且仅当对于某个 $c \geq 1$， $0 \circ 1^c \circ 2$ 是 $Z_{i\leftarrow i+1}$的真前缀。
- $F_{i,i+1}$ 的类型为（$\sqcup, *$），当且仅当对于某个 $c \geq 1$， $0 \circ (-1)^c \circ (-2)$ 是 $Z_{i\rightarrow i+1}$的真前缀。$F_{i,i+1}$ 的类型为（$\sqcup, \sqcup$），当且仅当$F_{i,i+1}$ 同时属于类型（$\sqcup, *$）和类型（$*, \sqcup$）。$F_{i,i+1}$ 的类型为（−），当 $c \geq 1$满足 $Z_{i\leftarrow i+1} = 0 \circ 1^c \circ 2$ 时。

我们强调，由于真前缀要求，任何类型为 $(-)$ 的面都不属于类型 $(*, \sqcup)$、$(\sqcup, *)$ 和 $(\sqcup, \sqcup)$。类型为 $(-)$ 的面也可以定义如下：如果 $F_{i,i+1}$ 的面环中的子路径 $(x_{i+1}, \ldots, x_i)$ 是一条朝向西方的水平直线，则 $F_{i,i+1}$ 属于类型 $(-)$。考虑 $P_{i\rightarrow i+1} = P_{i\leftarrow i+1}$，等价地，对于某个 $c \geq 1$，如果 $Z_{i\rightarrow i+1} = 0 \circ (-1)^c \circ (-2)$，则 $F_{i,i+1}$ 属于类型 $(-)$。———

直观地说，如果 $P_{i\rightarrow i+1}$ 在任何右转之前先左转两次，每次 90◦，且第一次 90◦ 左转发生在 x_i 处，那么面 $F_{i,i+1}$ 属于类型 $(\sqcup, *)$。这两次 90◦ 左转构成一个 ⊔ 形状。类似地，如果 $P_{i\leftarrow i+1}$ 在任何左转之前先右转两次，每次 90◦，且第一次 90◦ 右转发生在 x_{i+1}处，那么面 $F_{i,i+1}$ 属于类型 $(*, \sqcup)$。这两次 90◦ 右转构成一个 ⊔ 形状。

四种面类型的示意图见图11：

- 左上： $Z_{i\leftarrow i+1} = (0, 1, 1, 1, 2, 1, 2, 1, 2)$，因此$F_{i,i+1}$ 属于类型（$*, \sqcup$）。右上： $Z_{i\rightarrow i+1} = (0, -1, -1, -2, -3, -3, -2, -1, -2)$，因此$F_{i,i+1}$ 属于类型（$\sqcup, *$）。左下： $Z_{i\leftarrow i+1} = (0, 1, 2, 1, 0, -1, -1, 0, 1, 1, 2)$ 和 $Z_{i\rightarrow i+1} = (0, -1, -1, -2, -3, -3, -2, -1, 0, -1, -2)$，因此 $F_{i,i+1}$ 属于类型（ $\sqcup, \sqcup$）。右下： $Z_{i\leftarrow i+1} = (0, 1, 1, 1, 2)$，因此$F_{i,i+1}$ 属于类型（−）。

我们旨在通过分析面类型来提取一个严格单调环。例如，如果我们能够证明所有面 $F_{i,\ i+1}$ 的类型均为 （−）和 （$*$， ⊔），且至少有一个面属于 （$*$， ⊔）类型，那么直观地说，就可以找到一个递增环。

结构性质我们分析边 $(e_1, e_2, \ldots\ldots, e_s)$ 及其关联的面 $F_{i,\, i+1}$。以下引理证明了这一直观事实：从参考边 $e^\star$ 沿 G_k^+中的任意无交叉路径 P 到达任意 e_i 时，旋转角必须为 90°。注意，这样的路径 P 必然存在。

引理 5.1. 设 P 是 G_k^+ 中任意一条无交叉路径，它从参考边 $e^\star$ 开始，并在某个 $1 \le i \le s$ 处终止于 e_i。那么 旋转$(P) = 1$。

证 明。图12左侧的绘图展示了证明思路。考虑正交径向表示 R′：将边e_1、e_2、……、e_s 的南端点 x_1、x_2、……、x_s 连接成环 $C = (x_1, x_2, \ldots\ldots, x_s)$，使得从 e_i 到新边 (x_i, x_{i+1}) 的旋转为 90°左转，从 e_i 到新边 (x_i, x_{i-1}) 的旋转为 90°右转，并且 C 的任意子路径都是一条直线。注意，C 是一条水平线段，也是 R′ 中心面的面环。根据 (D1) 和 (D2)，很容易将 G_k 的良好绘图转换为 R′ 的正交径向绘图，因此 R′ 可绘制。由于 C 是一条水平线段，所有边 $e \in C$ 的边标签 $\ell_C(e)$ 必须相同。由于 C 不能是严格单调环，唯一的可能性是 C 是一个单调环，也就是说，对于所有边 $e \in C$，$\ell_C(e) = 0$。现在考虑引理陈述中的路径 P，以及 C 中的边 $e = (x_i, x_{i+1})$。由于 $\ell_C(e) = 0$，因此我们有 rotation$(P \circ e) = 0$。由于 $P \circ e$ 在 x_i 处形成 90°左转，因此 rotation$(P) = 1$。■

在接下来的讨论中，对于任意 $1 \le i \le s$ 和 $1 \le j \le s$，我们用 $P^{\text{out}}_{i \to j}$ 表示从 e_i 开始并在 e_j 结束的 C_k^+。对于特殊情形，$j = i + 1$、$P^{\text{out}}_{i \to j}$ 是 F_i 的面环的一条子路径，$_{i+1}$。下面的引理证明了一个直观事实：$P^{\text{out}}_{i \to j}$ 的旋转为 2。

引 理 5 . 2. 对于任意满足 $i \ne j$ 的 $1 \le i \le s$ 和 $1 \le j \le s$，有 旋转$(P^{\text{out}}_{i \to j}) = 2$。

证 明. 证明示意见图12的中间绘图。考虑正交径向表示 $\mathcal{R}'$，它通过在 G_k^+ 中将边 e_i 和 e_j 的南端点 x_i 和 x_j 连接成水平边 $e' = (x_j, x_i)$而得到，并使路径 $e_j \circ e' \circ e_i$ 连续右转两次，每次 90。。与引理 5.1 的证明类似，通过延伸 G_k 的良好绘图，可以绘出 $\mathcal{R}'$。路径 $P^{\text{out}}_{i \to j}$ 与路径 $e_j \circ e' \circ e_i$ 共同构成一个正则面的面环。根据 (R2)，必须有旋转$(P^{\text{out}}_{i \to j}) = 2$，因为正则面的旋转总和必须为 4。■

下面的引理考察一条从 e_j 到 e_i、且不使用 G_k中任何边的简单路径的旋转。引理表明，根据该路径与 $P^{\text{out}}_{i \to j}$是否共同围住 $\tilde{G}$的中心面，旋转要么为 -2，要么为 2。

引理 5.3。设 $1 \le i \le s$和 $1 \le j \le s$满足 $i \ne j$。设 P是 $\tilde{G}$中的任意一条简单路径，起点为 e_j，终点为 e_i，并满足以下条件。

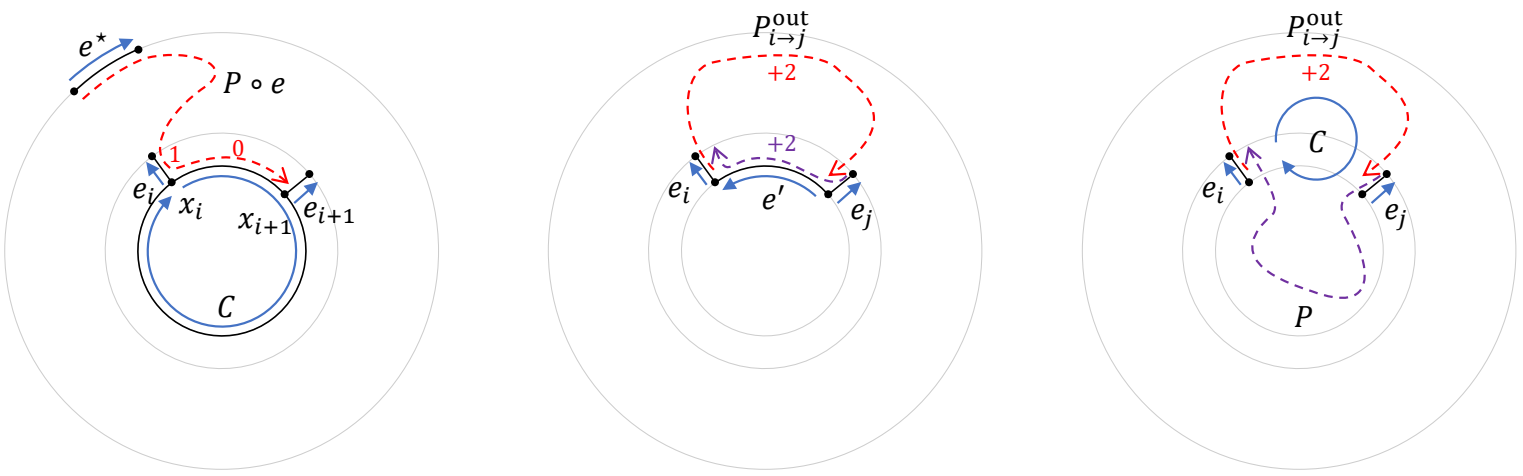

图12。 引理 5.1 至 5.3 的证明示意图。

— P 位于 C_k^+的内部。P不包含 $P^{\text{out}}_{i\to j}\setminus\{x_i、\ y_i、\ x_j、\ y_j\}$
— 中的任何顶点。

令 C为将 P与 $P^{\text{out}}_{i\to j}$结合后得到的环。若中心面位于 C的内部，则旋转$(P) = -2$。否则，旋转$(P) = 2$。

证明。 参见图12右侧的绘图，其中给出了证明示意。考虑由 G_k^+及 P中的所有边诱导的 $\tilde{G}$ 的子图 H。注意到 C 是 H的一个面环。若 $\tilde{G}$ 的中心面位于 C的内部，则 C 是 H中心面的面环，因此根据 (R2)，旋转$(C) = 0$。根据引理 5.2，旋转$(P^{\text{out}}_{i\to j})= 2$，所以必有旋转$(P) = -2$。若 $\tilde{G}$ 的中心面位于 C的外部，则 C 是 H的一个正则面的面环，因此根据 (R2)，旋转$(C) = 4$。因此，引理 5.2 意味着旋转$(P) = 2$。■

根据上述引理及 $j = i+1$和 $P = P_{i\leftarrow i+1}$可知，数列 $Z_{i\leftarrow i+1} = (z_1,\ z_2,\ \ldots)$ 中的最后一个数要么是 -2，要么是 2，具体取决于$F_{i,\ i+1}$是中心面还是正则面。我们稍后将证明，$F_{i,\ i+1}$ 必定是正则面。

引理 5.4. 对任意 $1 \le i \le s$，考虑面$F_{i,\ i+1}$。$F_{i,\ i+1}$的面环 C必定包含一条边 e'，它来自 $A = (S_1,\ S_2,\ \ldots,\ S_k)$ 中的某条水平线段 S_l，且沿环 C有旋转$(e_i \circ \cdots \circ e') = 1$以及旋转$(e' \circ \cdots \circ e_{i+1}) = 1$。

证明. 证明过程示意见图 13 左侧的绘图。考虑路径$P^{\text{out}}_{i\to i+1}$，它是从 e_i 出发并在 e_{i+1}结束的 C 的子路径。根据引理 5.2，旋转$(P^{\text{out}}_{i\to i+1}) = 2$，因此在 $P^{\text{out}}_{i\to i+1}$ 中存在一条边 e'，使得沿路径 $P^{\text{out}}_{i\to i+1}$（等价地，沿环 C）有旋转$(e_i \circ \cdots \circ e') = 1$，且旋转$(e' \circ \cdots \circ e_{i+1}) = 1$。由于 e_i 和 e_{i+1} 是竖直的，这样一条边 e' 必定是水平的。由于 e' 位于 G_k中，e' 属于$A = (S_1, S_2, \ldots, S_k)$中的某条水平线段 S_l。■

将上述引理与无法再添加更多虚拟边这一假设结合起来，我们证明字符串 $Z_{i\leftarrow i+1}$ 和 $Z_{i\to i+1}$ 必须满足某些结构性质。

引理 5.5. 考虑任意 $1 \le i \le s$。如果 $F_{i,\ i+1}$ 的类型为 $(*,\ \sqcup)$，那么除字符串的第一个数字外，字符串 $Z_{i\leftarrow i+1}$ 中的所有数字都至少为 1。如果 $F_{i,\ i+1}$ 的类型为 $(\sqcup,\ *)$，那么除字符串的第一个数字外，字符串 $Z_{i\rightarrow i+1}$ 中的所有数字都至多为 -1。

证明。 证明过程如图 13 右侧的绘图所示。根据不能再添加更多虚拟边这一假设，以及引理 5.4，我们可知不存在满足以下条件的水平线段 $S \in \mathcal{S}_{\mathsf{h}}$ with $\mathcal{N}_{\mathsf{north}}(S) = \emptyset$：

— S 是$C_{F_{i,i+1}}$的子路径。对于每条边 $e \in S$，沿着环$C_{F_{i,i+1}}$，其旋转为 $\mathrm{rotation}(e_{i+1} \circ \cdots \circ e) = 1$ 或
— $\mathrm{rotation}(e \circ \cdots \circ e_i) = 1$。

如果存在这样一条水平线段 $S$with $\mathrm{rotation}(e_{i+1} \circ \cdots \circ e) = 1$，且对所有 $e \in S$均成立，那么由于引理 5.4 中的边 e'，S可以添加一条虚拟边，因为

$$\mathrm{rotation}(e' \circ \cdots \circ e) = \mathrm{rotation}(e' \circ \cdots \circ \overline{e_{i+1}}) + \mathrm{rotation}(\overline{e_{i+1}} \circ \cdots \circ e) = 1 + 1 = 2$$

沿着环 $C_{F_{i,\ i+1}}$。对于剩余的情况，即对所有 $e \in S$，旋转$(e \circ \cdots \circ e_i) = 1$，出于类似的原因，$S$ 也可以添加一条虚拟边，因为引理 5.4 中的边 e'。

现在假设 $F_{i,i+1}$ 的类型为 $(*,\sqcup)$，且字符串 $Z_{i\leftarrow i+1}$ 中某个数 z_l 为 0，并且$l \neq 1$。类型 $(*,\sqcup)$ 保证字符串 $Z_{i\leftarrow i+1}$ 以 $0 \circ 1^{c'} \circ 2$ 开头，其中 $c' \ge 1$。在这个数 2 与上面的数 $z_l = 0$ 之间，$Z_{i\leftarrow i+1}$ 中必定存在一个子串 $2 \circ 1^{c} \circ 0$，其中 $c \ge 1$。对应于子串1^c 的 $P_{i\leftarrow i+1}$ 子路径的反向是一个水平线段 S，满足 $\mathrm{N}_{\mathsf{north}}(S) = \emptyset$，且对所有 $e \in S$，旋转$(e_{i+1} \circ \cdots \circ e) = 1$。由于上面的讨论，这样的水平线段 S 不可能存在。因此，字符串 $Z_{i\leftarrow i+1}$ 中的所有数都必须至少为 1，只有第一个数除外，它始终为 0。

引理第二个陈述的证明类似。假设 $F_{i,\ i+1}$ 属于类型（$\sqcup,\ *$），并且字符串 $Z_{i\rightarrow i+1}$ 中的某个数 z_l 至少为 0，且 $l \neq 1$。那么，对于某个 $c \ge 1$，我们可以找到 $Z_{i\rightarrow i+1}$的子串（-2）$\circ$（-1）$^c \circ 0$，随后得到矛盾，因为与子串（-1）^{c} 对应的水平线段不可能存在。■

直观地说，如果一个面 $F_{i,i+1}$ 属于 $(\sqcup,\sqcup)$ 类型，那么 $P_{i\leftarrow i+1}$ 的中间必定存在一个 ⊓-形状，而且对应于 ⊓-形状中间部分的水平线段必须可以添加一条虚拟边，因此 $F_{i,i+1}$ 不可能属于 $(\sqcup,\sqcup)$ 类型。在下面的引理中，我们结合引理 5.3 和 5.5，正式证明这一直观观察。

引理 5.6。 任取一个 $1 \le i \le s$。假设 $F_{i,i+1}$ 属于类型 $(\sqcup,*)$ 或 $(*,\sqcup)$。那么 $F_{i,i+1}$必须是一个正则面，并且 $F_{i,i+1}$ 不能属于类型 $(\sqcup,\sqcup)$。

证明。 根据引理 5.3，路径 $P_{i\leftarrow i+1}$的旋转为：若 $F_{i,\ i+1}$ 是中心面，则为 -2；若 $F_{i,\ i+1}$ 是正则面，则为 2。假设 $F_{i,\ i+1}$ 的类型为$(*,\ \sqcup)$。那么引理 5.5 表明，$P_{i\leftarrow i+1}$ 的旋转至少为 1，因此 $F_{i,\ i+1}$ 必为正则面，而 $P_{i\leftarrow i+1}$ 的旋转恰好为 2，这意味着字符串 $Z_{i\leftarrow i+1}$ 以数字 2 结尾。因此，如果 $F_{i,\ i+1}$ 也是

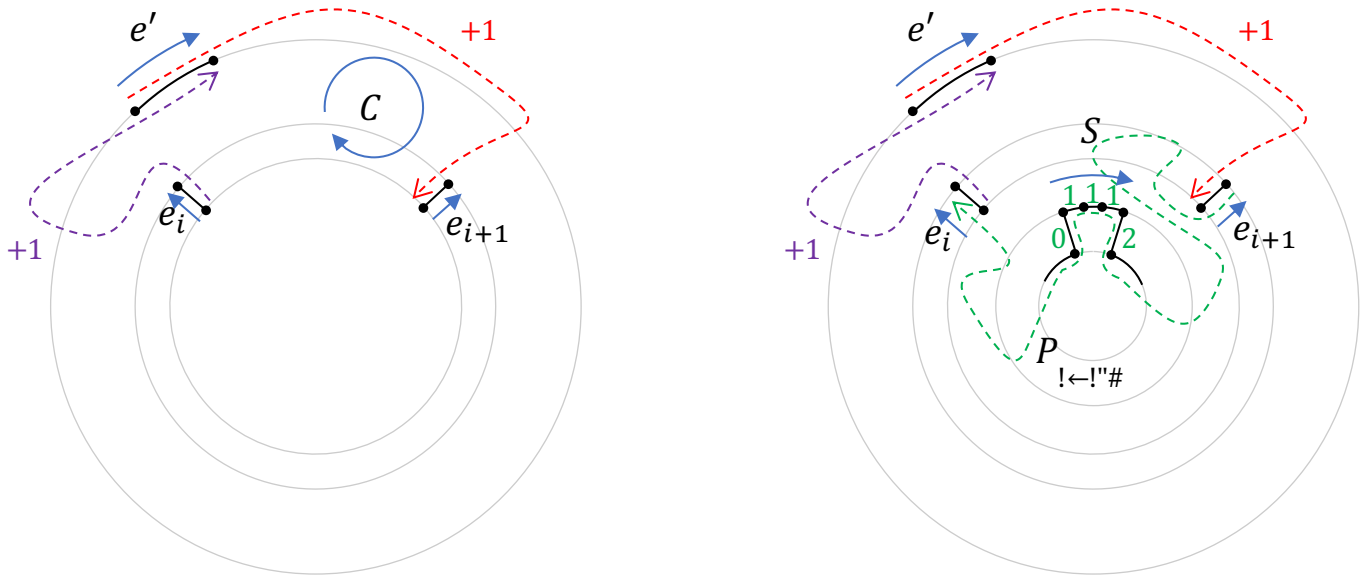

图 13. 引理 5.4 和引理 5.5 的证明示意图。

类型为 (⊔, ∗)，因此 $0 \circ 1^c \circ 2$，对于某个 $c \geq 1$ 将是 $Z_{i \leftarrow i+1}$ 的 严格 后缀，从而违反引理 5.5。因此，$F_{i,\ i+1}$ 不可能是类型 (⊔, ⊔)。

为完成证明，我们只需证明，当 $F_{i,i+1}$ 属于类型 (⊔, ∗) 时，$F_{i,i+1}$ 也必定是正则面。再次由引理 5.3 可知，若 $F_{i,i+1}$ 是中心面，则路径 $P_{i \to i+1} = P_{i \leftarrow i+1}$ 的旋转为 2。这与引理 5.5 矛盾，因为它要求 $P_{i \to i+1}$ 的旋转至多为 -1。因此，$F_{i,i+1}$ 是正则面。■

如上一节所述，我们可以假设 $\tilde{G}$ 中的每个顶点都与一条水平线段关联。考虑与某个 $e_i \in \mathcal{N}_{\text{south}}(A)$ 的南端点 x_i 关联的水平线段 S。根据 (S2)，将 S 添加到当前良序列 A 后未能得到良序列，可能有两个原因。第一种可能是$\mathcal{N}_{\text{north}}(S)$包含一条不属于$\mathcal{N}_{\text{south}}(A)$的边。第二种可能是存在两条边 e 和 e'，使得在$\mathcal{N}_{\text{north}}(S)$ 的排序中，e' 紧接在 e 之后，而在$\mathcal{N}_{\text{south}}(A)$的排序中，$e'$ 并未紧接在 e 之后。我们将证明第二种情况不可能发生。

引理 5.7。设 S 是 $\tilde{G}$ 中任意一条水平线段，e 和 e' 是任意两条边，且在 $\mathcal{N}_{\text{north}}(S)$ 的排序中，e' 紧接在 e 之后。若 e 和 e' 均属于 $\mathcal{N}_{\text{south}}(A)$，则在 $\mathcal{N}_{\text{south}}(A)$ 的循环排序中，e' 也紧接在 e 之后。

证 明。证明示意见图14左侧的绘图。假设引理的陈述不成立。那么，在 $\mathrm{N}_{\text{south}}(A)$ 中存在两条边 e_i 和 e_j，它们带有 $j \neq i+1$ 和一条水平线段 $S \in \mathrm{S}_{\text{h}}$，并且 e_j 在 $\mathrm{N}_{\text{north}}(S)$ 的排序中紧接着 e_i，而所有边 e_{i+1}、e_{i+2}、……、e_{j-1} 都不在 $\mathrm{N}_{\text{north}}(S)$ 中。假设这样的 e_i、e_j 和 S 存在。我们选择这样的 e_i、e_j 和 S，使循环排序 $\mathrm{N}_{\text{south}}(A)$ 中位于 e_i 之后、e_j 之前的边数最少。

我们对 e_i、 e_j 和 S 的选择意味着，对于每条水平线段 S'，只要 $\mathrm{N}_{\mathrm{north}}(S')$ 包含 $(e_{i+1}$、 e_{i+2}、 ……、 $e_{j-1})$ 中的一条边，$\mathrm{N}_{\mathrm{north}}(S')$ 与 $\mathrm{N}_{\mathrm{south}}(A)$ 的交集就必定是 $(e_{i+1}$、 e_{i+2}、 ……、 $e_{j-1})$ 的一个连续子序列；否则，我们就应该选择 S'，而不选择 S。我们根据水平线段 S' 将 $(e_{i+1}$、 e_{i+2}、 ……、 $e_{j-1})$ 划分为若干组，该水平线段与每条边 $e_l = (x_l, y_l)$ 的南端点 x_l 相接。换言之，如果 $(e_{i+1}$、 e_{i+2}、 ……、 $e_{j-1})$ 中两条边的南端点都与同一条水平线段 S' 相接，那么这两条边就属于对应于 S' 的同一组。注意，每组都对应于 $(e_{i+1}$、 e_{i+2}、 ……、 $e_{j-1})$ 的一个连续子序列。

考虑一个组 $(e_a, e_{a+1}, \ldots, e_b)$，并令 S' 为其对应的水平线段，因此 $\mathcal{N}_{\mathrm{north}}(S')$ 与 $\mathcal{N}_{\mathrm{south}}(A)$ 的交集恰好是该组中的边。由于我们假设将 S' 添加到 A 不会产生良序列，因此 $\mathcal{N}_{\mathrm{north}}(S')$ 必须包含一些不属于 $\mathcal{N}_{\mathrm{south}}(A)$ 的边，所以以下至少有一种情况成立：

- e_a 不是 $\mathcal{N}_{\mathrm{north}}(S')$ 的第一个元素，此时面 $F_{a-1,a}$ 属于 $(*, \sqcup)$ 类型，因为 e_a、 S'，以及 $\mathcal{N}_{\mathrm{north}}(S')$ 中紧挨在 e_a 之前的垂直边 e'，构成一个 $\sqcup$-形状，它是一个严格 $Z_{a-1\leftarrow a}$ 的前缀，因为 $e' \neq e_{a-1}$。
- e_b 不是 $\mathcal{N}_{\mathrm{north}}(S')$ 的最后一个元素，此时面 $F_{b,b+1}$ 属于 $(\sqcup, *)$ 类型。

我们令 $i = c_1 < c_2 < \cdots < c_t = j - 1$ 为满足以下条件的所有索引的序列： a 是某个组的最后一条边，或 e_{a+1} 是某个组的第一条边（$t = 4$ 见图14）。我们对 e_i、 e_j 和 S 的选择意味着，$F_{i,i+1} = F_{c_1, c_1+1}$ 必定属于 $(\sqcup, *)$ 类型，因为路径 $P_{i\to i+1}$ 会先在 x_i 处向左转90°，沿着 S 直行，然后在 x_j 处再次向左转90°以进入 e_j。

根据引理5.6， $F_{i、\ i+1} = F_{c_1、\ c_1+1}$ 不可能同时属于 $(*、\ \sqcup)$ 类型，因为这会迫使该面属于 $(\sqcup、\ \sqcup)$ 类型。鉴于上面的讨论，$F_{c_1、\ c_1+1}$ 不可能属于 $(*、\ \sqcup)$ 类型这一事实，迫使 $F_{c_2、\ c_2+1}$ 的类型为 $(\sqcup、\ *)$。类似地，我们可以论证 $F_{c_3、\ c_3+1}$、 $F_{c_4、\ c_4+1}$、……的类型必须为 $(\sqcup、\ *)$，这意味着 $F_{c_t、\ c_t+1} = F_{j-1、\ j}$ 的类型也为 $(\sqcup、\ *)$。证明 $F_{i、\ i+1}$ 属于 $(\sqcup、\ *)$ 类型时所用的同一论证，也可以用来证明 $F_{j-1、\ j}$ 必须属于 $(*、\ \sqcup)$ 类型。因此， $F_{j-1、\ j}$ 属于 $(\sqcup、\ \sqcup)$ 类型，但根据引理5.6这是不可能的，所以引理陈述成立。■

根据引理5.7，对于与某个 e_i 的南端点 x_i 相接的每条水平线段 S， S 必须是一条路径，并且 $\mathcal{N}_{\mathrm{north}}(S)$ 必须包含一条不属于 $\mathcal{N}_{\mathrm{south}}(A)$ 的边 e。在下面的引理中，我们使用引理5.7证明，不可能同时存在一个类型为 $(\sqcup, *)$ 的面和另一个类型为 $(*, \sqcup)$ 的面。

引 理 5 .8. 以下情况之一成立。

- 所有面 $F_{i,i+1}$ 的类型均为 $(-)$ 和 $(\sqcup, *)$，且至少有一个面 $F_{i,i+1}$ 的类型为 $(\sqcup, *)$。
- 所有面 $F_{i,i+1}$ 的类型均为 $(-)$ 和 $(*, \sqcup)$，且至少有一个面 $F_{i,i+1}$ 的类型为 $(*, \sqcup)$。

证 明。我们按照以下规则将 $\mathcal{N}_{\mathrm{south}}(A) = (e_1, e_2, \ldots, e_s)$ 划分为连续子序列： e_i 和 e_{i+1} 属于同一组，当且仅当存在一条水平线段 S'

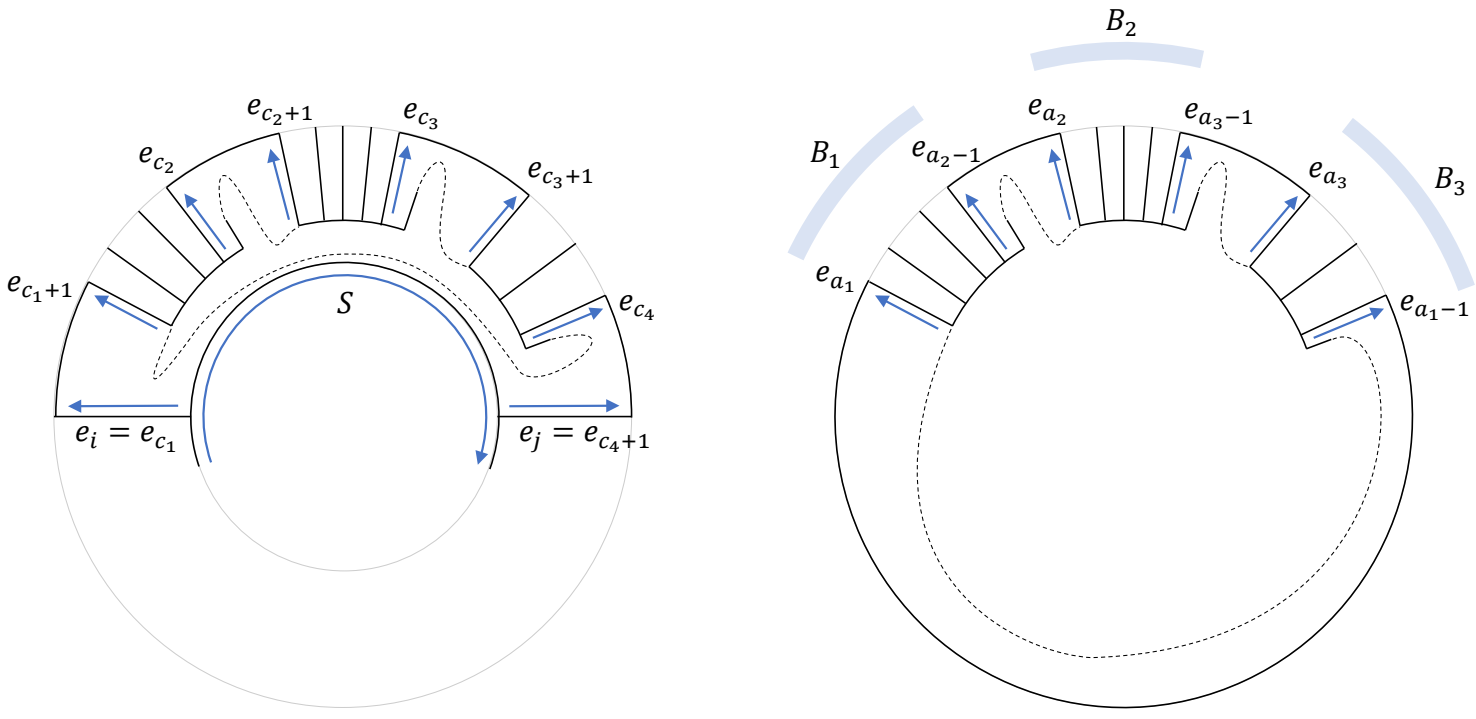

图14。 引理5.7和引理5.8的证明示意图。

使得 e_{i+1} 紧接在 e_i 之后，位于 $\mathcal{N}_{\text{north}}(S')$ 中。每个连续子序列都称为一个组。

设 t 为组的数量，并令 B_1、B_2、……、B_t 表示这些组，按照它们在循环排序 (e_1、e_2、……、e_s) 中的位置进行循环排列。令 a_i 表示满足 B_i 的第一条边为 e_{a_i} 的索引，因此 B_{i-1} 的最后一条边为 e_{a_i-1}。图14右侧的绘图展示了 $t = 3$ 的示例。

令 S' 为与组 B_i 对应的水平线段（即，B_i 是 $\mathcal{N}_{\text{north}}(S')$ 的连续子序列）。如前所述，由引理5.7可知，S' 必须是一条路径，并且$\mathcal{N}_{\text{north}}(S')$ 必须包含一条不属于 B_i 的边。因此，不可能有 $B_i = \mathcal{N}_{\text{north}}(S')$，所以以下情况至少有一种成立：

- 在 $\text{N}_{\text{north}}(S')$ 的顺序排列中，e_{a_i} 的第一条边正前方有一条边 e。由于 e 不在 $\text{N}_{\text{south}}(A)$ 中，F_{a_i-1,a_i} 的类型为 $(*, \sqcup)$，因为 e_{a_i}, S' 和 e 组成一个 $\sqcup$-形。在 $\text{N}_{\text{north}}(S')$ 的顺序排列中，B_i 的最后一条边正后方有一条边 e。由于 e 不在 $\text{N}_{\text{south}}(A)$ 中，$F_{a_{i+1}-1,a_{i+1}}$ 的类型为 $(\sqcup, *)$。

根据引理5.6，对于每个 $1 \le i \le t$，面 $F_{a_i-1、a_i}$ 不可能是类型 $(\sqcup, \sqcup)$。因此，唯一的可能性是它们要么全部是类型 $(*, \sqcup)$，要么全部是类型 $(\sqcup, *)$。

现在考虑任意一个面 $F_{l,\ l+1}$ 它不属于$F_{a_i-1,\ a_i}$ 的形式，其中 $1 \le i \le t$。也就是说，e_l 和e_{l+1} 属于同一组，这意味着它们的南端点都与同一条水平线段 S'相接，并且在顺序排列$\mathcal{N}_{\text{north}}(S')$中，$e_{l+1}$ 紧接在 e_l 之后，因此面$F_{l,\ l+1}$ 必定属于$(-)$类型。■

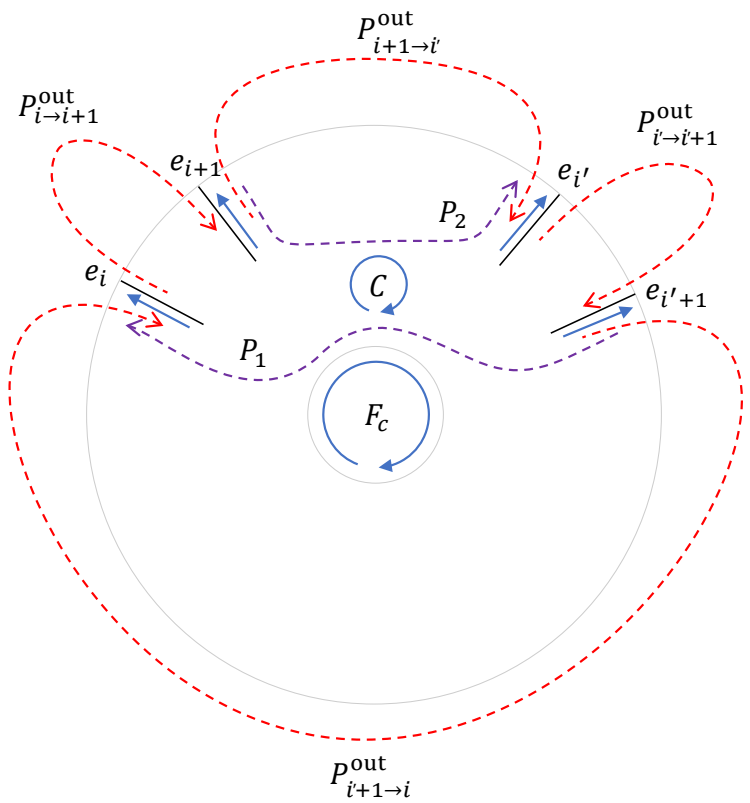

图 15. 引理5.10的证明示意图。

非正式地说，上述引理与引理 5.5 一起意味着，$P_{i\to i+1}$要么全部单调递增，要么 $P_{i\leftarrow i+1}$ 全部单调递增，因此我们应该能够通过考察这些路径中的边，提取出一个严格单调环。在此之前，我们先证明所有面$F_{i,\ i+1}$ 都是彼此不同的正则面。

引 理 5. 9. 对于每个 $1 \le i \le s$，$F_{i,\ i+1}$都是正则面。

证 明。若$F_{i,\ i+1}$属于类型（⊔，∗）或（⊔，∗），则引理5.6表明$F_{i,\ i+1}$是正则面。根据引理5.7，唯一剩下的情况是$F_{i,\ i+1}$ 属于类型（–）。假设$F_{i,\ i+1}$ 属于类型（–），并令 C 为$F_{i,\ i+1}$的面环。那么，rotation(C)等于rotation($P^{\text{out}}_{i\to i+1}$)与rotation($P_{i\leftarrow i+1}$)之和。根据类型（–）的定义，我们知道 rotation($P_{i\leftarrow i+1}$) = 2。根据引理5.2，我们知道rotation($P^{\text{out}}_{i\to i+1}$) = 2。因此，rotation($C$) = 4，所以根据（R2），$C$ 是正则面。■

引理 5.1 0. 任意两个面 $F_{i,\ i+1}$和 $F_{i',\ i'+1}$在 $i \ne i'$ 时均不相同。

证明。假设 $F_{i,\ i+1}$和 $F_{i',\ i'+1}$是同一个面 F。令 C为 F的面环。我们知道，起点为 e_i、终点为 e_{i+1}的 $P^{\text{out}}_{i\to i+1}$，以及起点为 $e_{i'}$、终点为 $e_{i'+1}$的 $P^{\text{out}}_{i'\to i'+1}$，都是 C的子路径。我们可以假设 $e_{i+1} \ne e_{i'}$和 $e_{i'+1} \ne e_i$，否则 C不是简单圈，这意味着底层图不是双连通的。

我们将 P_1 定义为 C 中以 $e_{i'+1}$ 为起点、以 e_i 为终点的子路径，并将 P_2 定义为 C 中以 e_{i+1} 为起点、以 $e_{i'}$为终点的子路径。因此，旋转(C) 是以下旋转之和

四条路径 $P^{\text{out}}_{i\to i+1}$、$P^{\text{out}}_{i'\to i'+1}$、$P_1$ 和 P_2的旋转。根据引理 5.9，F 是一个正则面，因此依据 (R2)，旋转$(C) = 4$。

路径 P_1 和 $P^{\text{out}}_{i'+1\to i}$ 构成一个环。路径 P_2 和 $P^{\text{out}}_{i+1\to i'}$ 构成另一个环。注意，$\tilde{G}$ 的中心面位于这两个环之一的内部。根据对称性，不妨假设中心面位于由 P_1 和 $P^{\text{out}}_{i'+1\to i}$构成的环的内部。如果中心面位于另一个环的内部，则交换 i 和 i'。图 15 展示了一个示例。

根据引理 5.3，取 $P = P_2$ 和 $\overline{P^{\text{out}}_{i+1\to i'}}$，可得 rotation$(P_2) = 2$。类似地，根据引理 5.3，取 $P = P_1$ 和 $P^{\text{out}}_{i'+1\to i}$，可得 rotation$(P_1) = -2$。因此，我们有 rotation$(\overline{P_1}) = 2$ 和 rotation$(P_2) = -2$。应用引理 5.3 时有一个细微问题：P_2可能包含 $P^{\text{out}}_{i+1\to i'} \setminus \{x_{i+1}, y_{i+1}, x_{i'}, y_{i'}\}$ 中的顶点，而 P_1 可能包含 $P^{\text{out}}_{i'+1\to i} \setminus \{x_{i'+1}, y_{i'+1}, x_i, y_i\}$ 中的顶点，因此不满足应用引理 5.3 的条件。可通过选择 i 和 i' 来解决此问题，使得 $F_{i,i+1} = F_{i',i'+1}$ 满足以下条件：循环排序 $(e_1, e_2, \ldots, e_s)$ 中，e_{i+1}之后、$e_{i'}$ 之前的边数最少。这样可确保 P_2 不包含 $(e_{i+2}, \ldots, e_{i'-1})$ 中的任何边及其反向，这意味着 P_2 不可能包含 $P^{\text{out}}_{i+1\to i'} \setminus \{x_{i+1}, y_{i+1}, x_{i'}, y_{i'}\}$ 中的任何顶点。只需能对 P_1和 P_2 中的一个应用引理 5.3 即可，因为根据引理 5.2，我们已知 rotation$(C) = 4$、rotation$(P^{\text{out}}_{i\to i+1}) = 2$ 和 rotation$(P^{\text{out}}_{i'\to i'+1}) = 2$ 。这些旋转数可推出 rotation$(P_2) = -$rotation(P_1)。

根据引理 5.3，$P_{i'\to i'+1}$ 的旋转为 -2，因此字符串 $Z_{i'\to i'+1}$ 的最后一个数字是 -2。注意到 P_1 是路径 $P_{i'\to i'+1}$ 和$\overline{P_1} \neq P_{i'\to i'+1}$的后缀，因此 rotation$(\overline{P_1}) = -2$ 意味着 $Z_{i'\leftarrow i'+1}$ 包含一个不在字符串开头的数字 0。于是，$F_{i',\, i'+1}$不可能属于类型（–）。此外，$F_{i',\, i'+1}$ 也不可能属于类型（⊔、 ∗），因为根据引理 5.5，类型（⊔、 ∗）的一个要求是：字符串 $Z_{i'\to i'+1}$ 中的所有数字（字符串的第一个数字除外）都至多为 -1。引理 5.8迫使$F_{i',\, i'+1}$ 属于类型（∗、 ⊔）。

通过类似地考察路径 $P_{i\leftarrow i+1}$ 的后缀 P_1，我们也可以迫使 $F_{i,\, i+1}$ 为类型 (⊔, ∗)。两种类型 (∗, ⊔) 和 (⊔, ∗) 同时存在，这与引理 5.8 矛盾，因此 $F_{i,\, i+1}$ 和 $F_{i',\, i'+1}$ 不可能是同一个面。■

严格单调环 作为上述引理的推论，所有路径 $P_{i\to i+1}$ 都不能使用属于 G_k 的任何边。尽管 $P_{i\to i+1}$ 的起始边和结束边可能是虚拟边，但 $P_{i\to i+1}$ 的其余边必须出现在原图 G 中。对于所有 $1 \le i \le s$，我们令 H 表示由 $P_{i\to i+1}$ 的所有边的无向版本（第一条边 e_i 和最后一条边 e_{i+1} 除外）所诱导的 G 的子图。现在，我们通过考察 H 的中心面，证明可以找到 G 的一个严格单调环。

引理 5.11. *C 的中心面的面环是 G的严格单调环。*

PROOF. 根据引理 5.9 和 5.10，将路径 $P_{i\to i+1}$ 串联起来形成的环（对所有 $1 \le i \le s$ 均如此），去除第一条边 $\overline{e_i}$ 和最后的各条边 e_{i+1}，将 $\tilde{G}$ 的中心面与

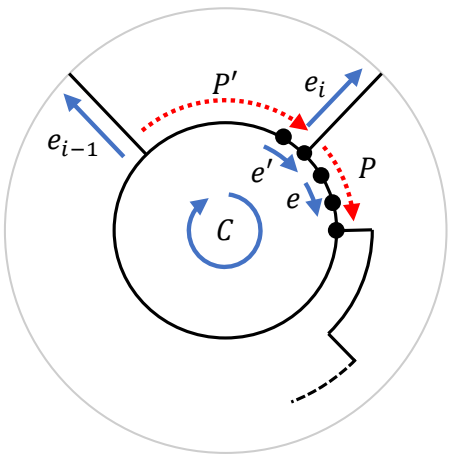

图16。 说明 C是严格单调的。

$\tilde{G}$ 的外面及所有 F_i 面分隔开，$i{+}1$，对所有 $1 \le i \le s$ 均如此。因此，H 的中心面与外面彼此不同，其中 H 的中心面包含 $\tilde{G}$ 的中心面，而 H 的外面包含 $\tilde{G}$ 的外面以及所有 F_i 面，$i{+}1$，对所有 $1 \le i \le s$ 均如此。

设 C 是 H的中心面的面环。我们断言，C 必定是一个简单圈，因此上述讨论表明，C 是 $\tilde{G}$的一个本质环。为说明这一点，考虑任意一条边$e \in C$。由于 e 的无向版本属于 H，因此 e 和 e 中至少有一个是某条路径 $P_{i\to i+1}$中的边，其中 $1 \le i \le s$。对于任意 $1 \le i' \le s$，我们不可能有 $e \in P_{i'\to i'+1}$，因为否则面$F_{i',\ i'+1}$ 将从右侧与 e 相邻，这意味着$F_{i',\ i'+1}$ 包含在 H的中心面中，这是不可能的。因此，对于每个 $e \in C$，都有 $e \in P_{i\to i+1}$，其中 $1 \le i \le s$。由此我们推断，$e \in C$ 和 $e \in C$不可能同时成立，而这迫使 C 成为一个简单圈，因为 C 是一个面环。

–

边标签 接下来，考虑任意 $e \in C$。我们相对于 $\tilde{G}$ 中的参考边 $e^\star$ 计算 $\ell_C(e)$。根据引理 5.8，要么所有面 $F_{i,i+1}$ 都属于类型 (–) 和 (⊔, ∗)，要么所有面 $F_{i,i+1}$ 都属于类型 (–) 和 (∗, ⊔)。我们声称，在第一种情况下，对所有 $e \in C$，$\ell_C(e) \le 0$；在第二种情况下，对所有 $e \in C$，$\ell_C(e) \ge 0$。

假设所有面 $F_{i,i+1}$ 都属于类型 (–) 和 (⊔, ∗)。固定一个 $e \in C$，并选取 $e_i \in \mathrm{N}_{\mathrm{south}}(A)$，使得 $e \in P_{i\to i+1}$。我们通过考虑从 $e^\star$ 到 e_i 的 G_k 的任意无交叉路径$P = (e^\star, \ldots, e_i)$，并考虑 $P_{i\to i+1}$ 的子路径 $P' = e_i \circ \cdots \circ e$，来计算 $\ell_C(e)$。于是，$\ell_C(e) = \mathrm{rotation}(P) + \mathrm{rotation}(P')$。根据引理 5.1，有 $\mathrm{rotation}(P) = 1$。根据引理 5.5，有 $\mathrm{rotation}(P') \le -1$，因为不可能有 $e = e_i$。因此，$\ell_C(e) \le 0$。

假设所有面 $F_{i,\ i+1}$ 都属于类型 (–) 和 (∗, ⊔)。类似地，将引理 5.5 与 $F_{i,\ i+1}$ 是一个正则面这一事实结合起来，我们可得，沿路径 $P_{i\to i+1}$，从 e_i 到 $P_{i\to i+1}$ 的任意中间边 e 的旋转至少为 -1。因此，按照与上文相同的论证，我们可得，对于所有 $e \in C$，$\ell_C(e) \ge 0$。

严格单调性 要完成证明，我们仍需证明 C 是严格 单调的。也就是说，我们需要证明不可能对所有 $e \in C$都有 $\ell_C(e) = 0$。现在假设对所有

$e \in C$均成立。我们的目标是证明，这一假设意味着所有面 $F_{i,\ i+1}$ 都属于 (–) 类型，这与引理5.8矛盾。证明的这一部分如图 16 所示。

再次，我们假设所有面 $F_{i,i+1}$ 都属于类型 (–) 和 (⊔, ∗)。对于另一种情况，即所有面 $F_{i,i+1}$ 都属于类型 (–) 和 (∗, ⊔)，证明类似。考虑任意一个 $e \in C$。我们知道，对于某个 $1 \le i \le s$，有 $e \in P_{i\to i+1}$。我们将 e 延伸为 C 的一条子路径 P，使得 P是 $P_{i\to i+1}$的一条子路径，并且在 C中紧邻 P 之前和紧邻 P之后的边都不属于 $P_{i\to i+1}$。由于 P的所有边 $\tilde{e}$的边标签 ℓ_C（$\tilde{e}$）均为 0，因此沿路径 $P_{i\to i+1}$ 从 e_i 到 P 的任意一条边 $\tilde{e}$ 的旋转必定为 −1。再考虑 $P_{i\to i+1}$ 中紧邻 P之前和之后的两条边，可得一个旋转数序列 0 ∘ $(-1)^c$ ∘ (−2)，它是 $Z_{i\to i+1}$的一个子串。这样的子串对应于绘图中的一个 ⊔-形。

分为两种情况。如果 $F_{i,\ i+1}$ 属于 (–) 类型，那么上述字符串 0∘ $(-1)^c$ ∘ (−2) 必定就是整个字符串 $Z_{i\to i+1}$。如果 $F_{i,\ i+1}$ 属于 (⊔, ∗) 类型，那么根据引理 5.5，上述字符串 0∘ $(-1)^c$ ∘ (−2) 必定是 $Z_{i\to i+1}$ 的一个前缀。无论哪种情况，$P_{i\to i+1}$ 中紧邻 P 之前的边都必须是 e_i。

现在考虑 P 之前紧邻的边 e' in C。这条边 e' 必须是 $P_{i-1\to i}$ 中倒数第二条边，也就是说，e' 紧邻 $P_{i-1\to i}$ 的最后一条边 e_i 之前。用 e 替代 e' 重复上述分析，可得到一条包含 e' 的路径 P'。由于这里 e' 是 $P_{i-1\to i}$ 中倒数第二条边，路径 P' 必须覆盖 $P_{i-1\to i}$ 的所有中间边。换言之，$P_{i-1\to i}$ 必须具有如下结构：经过第一条边 e_{i-1} 后，向左转 90∘ 进入 C，沿着 C 直行，然后从 e' 向左再转 90∘ 到 e_i，因此 $F_{i-1,\ i}$ 属于 (–) 类型。

我们可以对 $F_{i-2,\ i-1}$、$F_{i-3,\ i-2}$ 等重复同样的分析，从而推断它们全都属于类型（–），这与引理 5.8 矛盾，因此不可能对所有 $e \in C$ 都有 ℓ_C（e）$= 0$。所以，C 是严格单调的。■

图17给出了一个提取严格单调环的示例。图中，H 是由阴影区域内的顶点诱导的子图。环 $C =$（v_1、v_2、……、v_5）是 H 的中心面的面环。在此例中，$\mathrm{N}_{\mathrm{south}}$（$A$）=（$e_1$、$e_2$、……、$e_5$）。面 F_5、1、F_1、2 和 F_2、3 属于类型（∗、⊔）。面 F_3、4 和 F_4、5 属于类型（–）。环 C 是严格单调的，因为它是递增的。我们可以计算得出，ℓ_C（（v_1、v_2））$= 1$：先沿无交叉路径 P 从 $e^\star$ 到 e_2，再沿路径 $P_{2\to 3}$ 从 e_2 到（v_1、v_2），因为（v_1、v_2）是 $P_{2\to 3}$ 的一条中间边。第一部分的旋转为 1，第二部分的旋转为 0，因此总旋转为 1。类似地，我们可以计算出，在 C 中每条其余边 e 都有 ℓ_C（e）$= 0$。我们将本节的讨论总结为一个引理。

引理 5.12. 假设贪心算法输出了一个良序列 $A =$（S_1，S_2，. . .，S_k），但该序列未覆盖所有水平线段的集合 $\mathcal{S}_{\mathrm{h}}$。则可以在 O（n）时间内计算出原图 G 中的一个严格单调环。

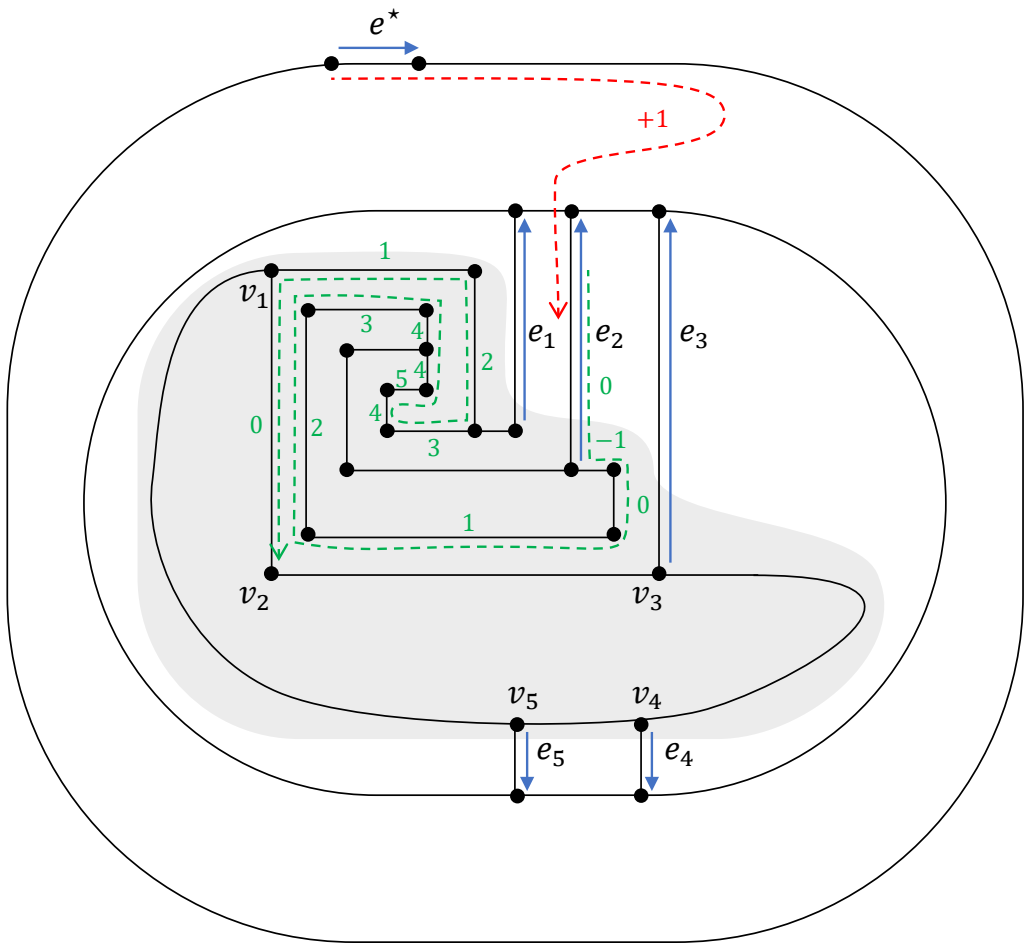

图17. 提取一个严格单调环 $C = (v_1, v_2, \ldots\ldots, v_5)$。

证明。 根据引理 5.11， C的中心面面环是 H的一个严格单调环。计算 H 和 C 可在 $O(n)$ 时间内完成。■

现在我们可以证明定理 2.5。

定理 2.5. (重述) 存在一个 $O(n$对数 $n)$时间算法$\mathcal{A}$，对于给定的正交径向表示$\mathcal{R}$，其中图是一个有 n个顶点的双连通简单图，且具有固定的参考边 $e^\star$，如果$\mathcal{N}_{\text{north}}(S) = \emptyset$，则该算法输出$(\mathcal{R}\ e^\star)$ 的绘图或$(\mathcal{R}\ e^\star)$的严格单调环；这里 $S \in$ 所指的是包含 $e^\star$的水平线段$\mathcal{S}_{\text{h}}$ 。

证明。对于给定输入 $(\mathcal{R}\ e^\star)$，我们运行引理4.1中的 $O(n \log n)$ 时间贪心算法，使得水平线段 $S \in \mathcal{S}_{\text{h}}$ 包含 $e^\star$ 时有 $\mathcal{N}_{\text{north}}(S) = \emptyset$ 。算法可能有两种结果。如果得到一个覆盖所有水平线段 $\mathcal{S}_{\text{h}}$ 的良序列，那么我们可以使用引理3.2计算 $(\mathcal{R}\ e^\star)$ 的绘图。否则，算法会停止并得到一个未覆盖所有水平线段$\mathcal{S}_{\text{h}}$的良序列，且无法再取得进展；此时，根据引理5.12，可以在 $O(n)$时间内计算出一个严格单调环。■

6. 搜索参考边

本节将展示如何把定理 2.5 中针对具有固定参考边 $e^\star$ 的正交径向表示 $\mathcal{R}$ 设计的绘图算法，扩展为处理不具有固定参考边的正交径向表示 $\mathcal{R}$。此扩展会使时间复杂度额外增加$O(\log n)$ 倍。其核心思想是通过二分搜索找到一条参考边 $e^\star$，使得在存在这样的边时，$(\mathcal{R}\ e^\star)$ 可绘制。

在本节中，我们用 $\ell_C^{e^\star}(e)$ 表示以 $e^\star$ 为参考边时，e 相对于 C 的边标签。我们首先证明以下引理，它本质上与 [2,引理 4。2]相同。为完整起见，我们仍给出证明。

引理 6.1. 对于 C_{F_o} 中边 $e_1^\star$ 和 $e_2^\star$ 的任意两种选择，对于任意本质环 C 以及 C 中的任意边 e，我们有 $\ell_C^{e_2^\star}(e) = \ell_C^{e_1^\star}(e) + \ell_{C_{F_o}}^{e_2^\star}(e_1^\star)$。

证明。在证明中，我们不妨假设三条边 e、$e_1^\star$ 和 $e_2^\star$ 互不相同。若 $e_1^\star = e_2^\star$，则等式显然成立，因为 $\ell_C^{e_2^\star}(e) = \ell_C^{e_1^\star}(e)$ 且 $\ell_{C_{F_o}}^{e_2^\star}(e_1^\star) = 0$。若 $e = e_1^\star$，则等式显然成立，因为 $\ell_C^{e_1^\star}(e) = 0$ 且 $\ell_C^{e_2^\star}(e) = \ell_{C_{F_o}}^{e_2^\star}(e_1^\star)$；这是因为用于计算 $\ell_{C_{F_o}}^{e_2^\star}(e_1^\star)$ 的参考路径 P 也可用于计算 $\ell_C^{e_2^\star}(e)$。若 $e = e_2^\star$，则 $\ell_C^{e_2^\star}(e) = 0$ 且 $\ell_C^{e_1^\star}(e) + \ell_{C_{F_o}}^{e_2^\star}(e_1^\star) = 0$，因为$\ell_C^{e_1^\star}(e) = \ell_{C_{F_o}}^{e_1^\star}(e_2^\star)$ 蕴含 $\ell_C^{e_1^\star}(e) = -\ell_{C_{F_o}}^{e_2^\star}(e_1^\star)$。我们还注意到，$e$ 不可能是 $e_1^\star$ 和 $e_2^\star$，因为不存在包含 $e_1^\star$ 或 $e_2^\star$ 的本质环。

寻找合适的参考路径 我们选取一条满足以下条件的路径 P。

1. P 是位于 C外部的一条简单路径，起点为 $e_1^\star$ 或 $e_1^\star$，终点为 e 或 e。2. 一旦 P 离开 C_{F_o}，就不会再访问 C_{F_o} 中的任何顶点。更形式化地说，如果我们将其写为 $P = e_1 \circ e_2 \circ \cdots \circ e_s$，那么要求存在一个索引 i，使得 $e_1 \circ e_2 \circ \cdots \circ e_i$是 C_{F_o} 的一个子路径，并且 $e_{i+1} \circ e_{i+2} \circ \cdots \circ e_s$ 中属于 C_{F_o} 的唯一顶点是其第一个端点。3. P 确实包含 $e_2^\star$ 和 $e_2^\star$。

可以按如下方式找到这样的路径 P（该过程的示意图见图 18）。首先，我们选择任意满足第一个条件的路径 P。为了使其满足其余两个条件，我们令 v 为 P 中所用 C_{F_o} 的最后一个顶点，并以这样的方式分解 P：$P = P_s \circ P_t$，其中 P_s 终止于 v，而 P_t 始于 v。

对于情况 $P_t = \emptyset$，我们知道 e 或 e 位于 C_{F_o}上，因此自然有两种从 $e_1^\star$ 或 $e_1^\star$ 连接到 e 或 e的 C_{F_o} 子路径选择，其中一条不包含 $e_2^\star$ 和 $e_2^\star$，因此这就给出了所需的 P路径。

对于 $P_t \neq \emptyset$ 的情况，我们可以将 P_s 替换为 $e_1^\star$ 到 v 在 C_{F_o} 中的路径，或替换为$e_1^\star$ 到 v 在 C_{F_o} 中的路径，以满足第二个条件，而这两种选择中必有一种满足第三个条件。

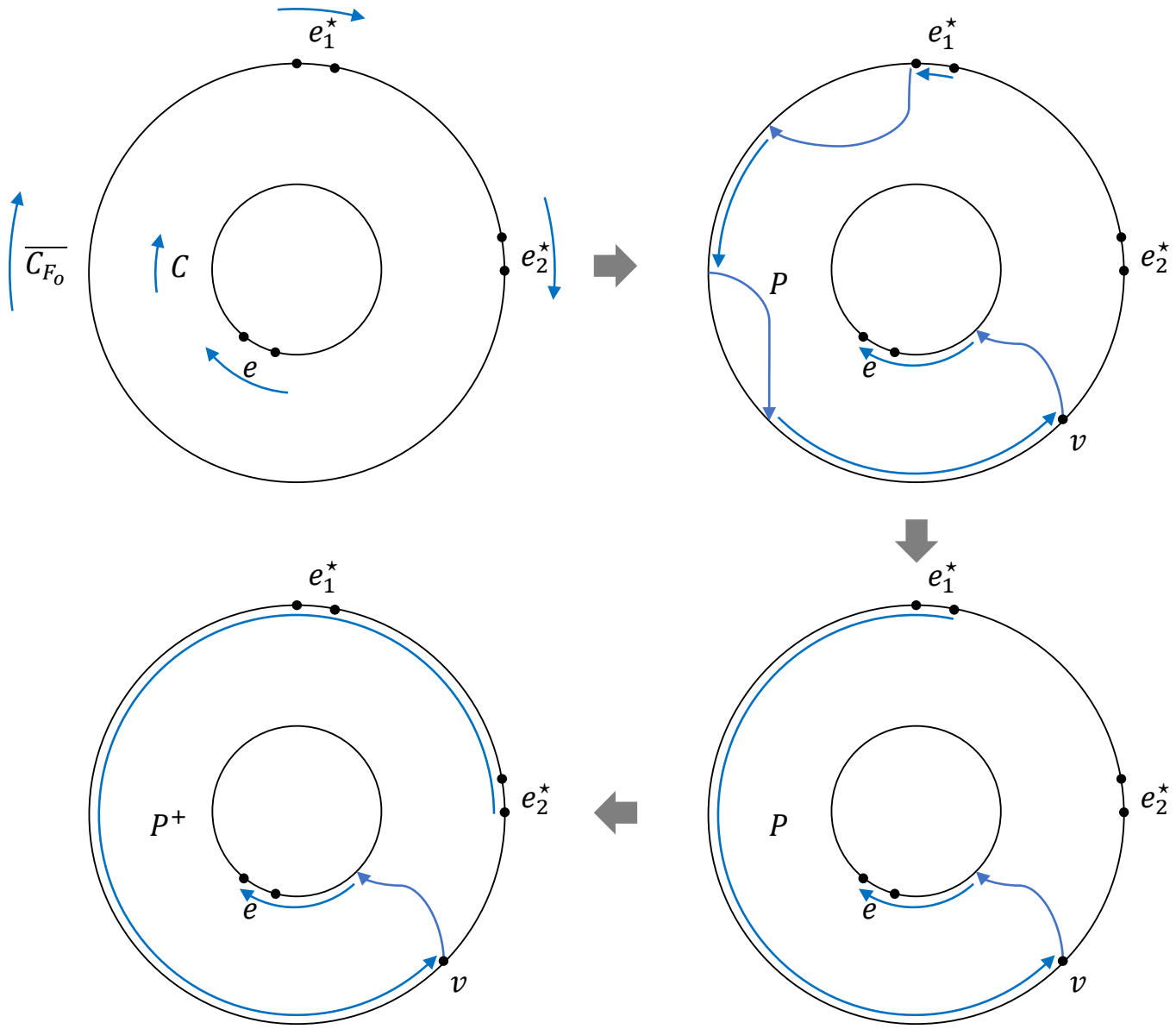

图 18. 引理6.1的证明中寻找合适的参考路径。

令 $P = e_1^\star \circ \cdots \circ \tilde{e}$ 为满足上述条件的路径。其中，$e_1^\star$ 为 $e_1^\star$ 或 $e_1^\star$，而 $\tilde{e}$ 为 e 或 e。令 $P^+ = e_2^\star \circ \cdots \circ e_1^\star \circ \cdots \circ \tilde{e}$ 为一条简单路径，将 P 延伸，使得 $e_2^\star$ 为 $e_2^\star$ 或 $e_2^\star$，且部分 $e_2^\star \circ \cdots \circ e_1^\star$ 位于 C_{F_o} 上（见图 18）。由于上述对 P 的要求，这样的延伸是可能的。

边标签计算 我们可以使用 $P^+ = e_2^\star \circ \cdots \circ e_1^\star \circ \cdots \circ \tilde{e}$（不包括两个端点）作为计算 $\ell_C^{e_2^\star}(e)$时的参考路径。根据方向公式，我们可以写出

$$\ell_C^{e_2^\star}(e) = \text{rotation}(P^+) + 2b_2 - 2b,$$

其中 $b_2 \in \{0, 1\}$ 是用于表示 $e_2^\star = e_2^\star$ 是否成立的指示函数，类似地，$b \in \{0, 1\}$ 是用于表示 $\tilde{e} = e$ 是否成立的指示函数。我们将旋转(P^+) 的计算分成两部分：

$$\text{rotation}(P^+) = \text{rotation}(\tilde{e_2^\star} \circ \cdots \circ \tilde{e_1^\star}) + \text{rotation}(\tilde{e_1^\star} \circ \cdots \circ \tilde{e}).$$

~ ~类似地， $e_2^\star \circ \cdots \circ e_1^\star$ （不包括两个端点）可以用作计算 $\ell_{C_{F_0}}^{e_2^\star}(e_1^\star)$ 的参考路径，因此我们可以推断：

$$\ell_{C_{F_0}}^{e_2^\star}(e_1^\star) = \text{rotation}(\tilde{e_2^\star} \circ \cdots \circ \tilde{e_1^\star}) + 2b_2 - 2b_1,$$

其中， $b_1 \in \{0,1\}$ 是判断 $\tilde{e_1^\star} = e_1^\star$ 的指示函数。我们也可以将 $\ell_C^{e_1^\star}(e)$ 表示为关于旋转$(e_1^\star \circ \cdots \circ \tilde{e})$ 的形式，如下所示：

$$\ell_C^{e_1^\star}(e) = \text{rotation}(\tilde{e_1^\star} \circ \cdots \circ \tilde{e}) + 2b_1 - 2b.$$

结合这些公式，我们得到所需的等式

$$\ell_C^{e_2^\star}(e) = \ell_C^{e_1^\star}(e) + \ell_{C_{F_0}}^{e_2^\star}(e_1^\star),$$

因为所有 $\pm 2b$、 $\pm 2b_1$ 和 $\pm 2b_2$ 都相互抵消。 ■

特别地，如果 $\ell_{C_{F_0}}^{e_2^\star}(e_1^\star) = 0$，那么上述引理意味着，无论 $e^\star = e_1^\star$ 还是 $e^\star = e_2^\star$ ，边标签 $\ell_C(e)$ 都相同。

在下面的讨论中，我们固定一条边 e' in C_{F_0}，并令 I 为所有 e in C_{F_0} 时 $\ell_{C_{F_0}}^{e'}(e)$ 可能取值的范围，则 I 是一个连续的整数序列，且满足 $0 \in I$。对于每个本质环 C，令 I_C 为所有 e in C 时 $\ell_C^{e'}(e)$ 可能取值的范围，则 I_C 也是一个连续的整数序列。

假设当前使用的参考边为 $e^\star$。回顾一下，若对所有 $e \in C$，边标签 $\ell_C^{e^\star}(e)$ 均为非负数，且存在 $e \in C$ 使得 $\ell_C^{e^\star}(e) \geq 1$，则本质环 C 是递增的。于是，引理6.1表明， C 是递增环（以 $e^\star$ 为参考边）当且仅当满足以下条件：

$$\begin{aligned} &\ell_{C_{F_0}}^{e^\star}(e') + \min I_C \geq 0, && \text{若 } |I_C| \geq 2, \\ &\ell_{C_{F_0}}^{e^\star}(e') + \min I_C \geq 1, && \text{若 } |I_C| = 1。 \end{aligned}$$

类似地，根据引理6.1， C 是递减环（以 $e^\star$ 为参考边）当且仅当满足以下条件：

$$\begin{aligned} &\ell_{C_{F_0}}^{e^\star}(e') + \max I_C \leq 0, && \text{if } |I_C| \geq 2, \\ &\ell_{C_{F_0}}^{e^\star}(e') + \max I_C \leq -1, && \text{如果 } |I_C| = 1。 \end{aligned}$$

鉴于上述情况，存在两个仅取决于 I_C 的整数 L_C 和 U_C，使得当且仅当 $\ell_{C_{F_0}}^{e^\star}(e') > U_C$ 时， C 是递增环；并且当且仅当 $\ell_{C_{F_0}}^{e^\star}(e') < L_C$ 时， C 是递减环。因此，当且仅当

$$\ell_{C_{F_0}}^{e^\star}(e') \in [L_C, U_C].$$

特别地，(R3) 得到满足，当且仅当对所有本质环 C，$[L_C, U_C]$ 的交集非空。换言之，当且仅当 $L^* \le U^*$ 时，(R3) 得到满足；其中我们定义 $L^* = \max\ L_C$ 和 $\min\ U_C = U^*$，取最小值和最大值的范围为所有本质环 C。

二分搜索 鉴于给定的正交径向表示 $\mathcal{R}$ 满足 (R1)–(R3)，我们可以使用二分搜索找到一条参考边 $e^\star$，使得 $(\mathcal{R}\ e^\star)$ 可绘制，具体如下。我们在 C_{F_o} 中任取一条边 e'，并计算上文定义的区间 I。我们从区间 I开始二分搜索。在二分搜索的每一步中，我们取当前区间的中位数 x，然后运行定理 2.5 的算法，并选择参考边 $e^\star$，使得$\ell^{e^\star}_C$

$(e') = x$。

如果 $x > U^*$，则存在递增环，算法必须返回一个递增环，因为不存在递减环，这是因为 $x > U^* \ge L^*$。在这种情况下，我们将当前区间的上界更新为 $x - 1$，因为我们得知 $x > U^*$。如果 $x < L^*$，则存在递减环，算法必须返回一个递减环，因为不存在递增环，这是因为 $x < L^* \le U^*$。在这种情况下，我们将当前区间的下界更新为 $x + 1$，因为我们得知 $x < L^*$。如果 $x \in [L^*, U^*]$，则既不存在递增环，也不存在递减环，因此算法将返回 $(\mathcal{R}\ e^\star)$ 的绘图。

寻找合适的参考边在实现上述方法时，还有一个问题需要解决：定理 2.5 的算法要求参考边 $e^\star$ 位于水平线段 $S \in \mathcal{S}_{\mathsf{h}}$ 上，且满足 $\mathcal{N}_{\mathsf{north}}(S) = \emptyset$。等价地，当且仅当 $e^\star \in S$ 使得 S 满足以下某一条件时，选择的参考边$e^\star \in C_{F_o}$ 才符合这一要求。

$S = \overline{C_{F_o}}$ 以及对路径 S中的所有边 e，$\ell_{C_{F_o}}(e) = 0$。

S 是 C_{F_o} 的一个子路径，且满足以下条件。

$$\ell_{C_{F_o}}(e) = \begin{cases} -1 \\ 0, \text{，对于路径 } C_{F_o} \text{中紧邻 } S \text{ 之前的边 } e \text{。对于路径 } S \text{中的所有边 } e, \\ \\ 1 \qquad\qquad e \text{，对于路径 } C_{F_o} \text{中紧邻 } S \text{ 之后的边。} \end{cases}$$

我们声称，对于任意 $x \in \left[L_{C_{F_o}}, U_{C_{F_o}}\right]$，都可以找到一条边 $e^\star$，使得 $\ell^{e^\star}_C(e') = x$，且满足上述要求，因此在二分搜索过程中运行定理 2.5 的算法时，可以将这条边 $e^\star$ 用作参考边。注意，若 $x \notin \left[L_{C_{F_o}}, U_{C_{F_o}}\right]$，那么无论选择哪个满足 $\ell^{e^\star}_C(e') = x$ 的 $e^\star$，环 C_{F_o} 已经是一个严格单调环，因此无需运行定理 2.5 的算法。

引理 6.2. 对于每个 $x \in \left[L_{C_{F_o}}, U_{C_{F_o}}\right]$，都存在一条边 $e^\star \in C_{F_o}$，满足 $\ell^{e^\star}_{C_{F_o}}(e') = x$，使得若将$e^\star$ 用作参考边，则 $e^\star$ 属于一个水平线段 $S \in \mathcal{S}_{\mathsf{h}}$，且满足 $\mathcal{N}_{\mathsf{north}}(S) = \emptyset$。

证 明 . 为简化符号，在本证明中，我们记作 $C = C_{F_o}$。首先，任取一个满足 $\ell_C^{e^\star}(e') = x$ 的 $e^\star \in C$，并将 $e^\star$ 用作参考边。由于 $x \in [L_C, U_C]$，C 不是严格单调环。若对所有 $e \in C$，$\ell_C(e) = 0$ 均成立，则 $S = C$ 本身已经是一个水平线段 $S \in \mathsf{S_h}$，且满足 $\mathrm{N_{north}}(S) = \emptyset$，因此结论成立。否则，$C$ 包含边标签为 -1 和 1 的边。我们选取两条边 $e^- \in C$ 和 $e^+ \in C$，使得 $\ell_C(e^-) = -1$、$\ell_C(e^+) = 1$，并使子路径 P（它属于 C，从 e^- 开始，到 e^+ 结束）的长度最小。我们对 P 的选择意味着，P 中的所有中间边 $\tilde{e}$ 都满足 $\ell_C(\tilde{e}) = 0$，因此它们构成所需的水平线段 S。若 $e^\star \in S$，则结论成立。否则，由于对任意 $\tilde{e} \in S$，$\ell_{C_{F_o}}^{e^\star}(\tilde{e}) = \ell_C(\tilde{e}) = 0$，根据引理6.1，即使将参考边从 $e^\star$ 改为 $\tilde{e}$，所有边标签仍保持不变。因此，即使底层参考边是任意一条 $\tilde{e} \in S$，S 仍是水平线段 $S \in \mathsf{S_h}$，且满足 $\mathrm{N_{north}}(S) = \emptyset$。任何这样的边 $\tilde{e}$ 都满足引理的要求，因为根据引理6.1，$\ell_C^{\tilde{e}}(e') = \ell_C^{e^\star}(e') - \ell_C^{e^\star}(\tilde{e}) = x - 0 = x$，其中 $e = e'$、$e_1^\star = \tilde{e}$ 和 $e_2^\star = e^\star$。■

现在我们准备证明定理 2.6。

定 理 2 .6.（**重述**）存在一个 O（n 对数^2 n）时间算法 $\mathcal{A}$，用于判定一个 n 个顶点的双连通简单图的正交径向表示 $\mathcal{R}$ 是否可绘制。如果 $\mathcal{R}$ 可绘制，则 $\mathcal{A}$ 还会计算出 $\mathcal{R}$ 的一个绘图。

证 明。我们只需说明，对于给定的正交径向表示 $\mathcal{R}$ 可绘制的情况，可以在 O（n 对数^2 n）时间内计算出实现 $\mathcal{R}$ 的正交径向绘图。在这种情况下，利用引理 6.2，上文讨论的二分搜索算法会找到一条参考边 $e^\star$，使得定理 2.5 的 O（n 对数 n）时间算法能够输出实现（R $e^\star$）的正交径向绘图。该绘图也是实现 $\mathcal{R}$ 的正交径向绘图。二分搜索的迭代次数为 O（对数 $|I|$）$= O$（对数 n），因此总体时间复杂度为 O（n 对数^2 n），因为根据定理 2.5，每次迭代的代价为 O（n 对数 n）。■

7. 归约为双连通简单图

本节将说明，不失一般性，我们可以假设输入图是简单且双连通的。具体而言，给定一个平面图 $G = (V, E)$、G 的组合嵌入 $\mathcal{E}$ 以及 $\mathcal{R}$ 的正交径向表示 $(G, \mathcal{E})$，且满足 (R1) 和 (R2)，我们将构造一个双连通简单平面图 $G' = (V', E')$、G' 的组合嵌入 $\mathcal{E}'$ 以及 $\mathcal{R}'$ 的正交径向表示 $(G', \mathcal{E}')$，且满足 (R1) 和 (R2)，使得 $\mathcal{R}$ 可绘制，当且仅当 $\mathcal{R}'$ 可绘制。图19展示了该归约的示意图。此外，该归约只需线性时间，具体如下：

- 构造 G'、$\mathcal{E}'$ 和 $\mathcal{R}'$ 需要线性时间。给定 $\mathcal{R}'$ 的正交径向绘图，可以在线性时间内找到 $\mathcal{R}$ 的正交径向绘图。

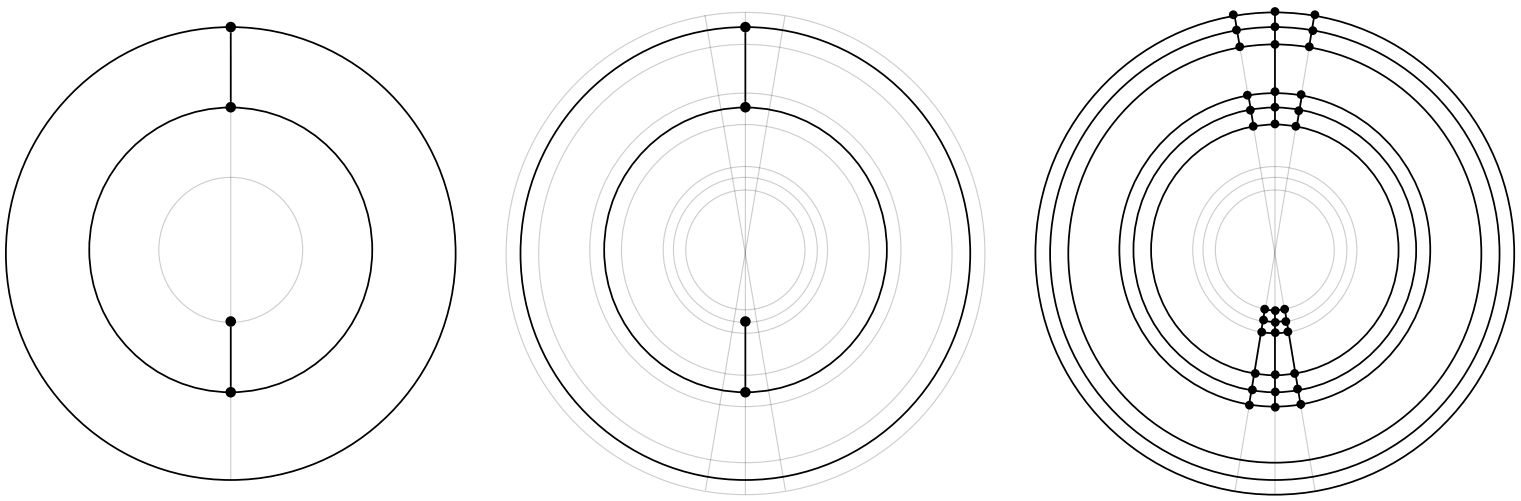

图19。 通过加粗归约为双连通简单图。

归约 在本节中，所有边均为无向边，并且允许图 G包含重边和自环。该归约的思路是通过加粗图来提高连通性。对于每个顶点 $v \in V$，我们将其替换为一个由三条水平线和三条垂直线组成的网格。具体来说，令

$$X_v = \{v_{i,j} \mid i \in \{-1,0,1\} \text{ and } j \in \{-1,0,1\}\} \quad \text{and} \quad V' = \bigcup_{v \in V} X_v.$$

构造边集 E' 时，我们首先在 $v_{i,\ j}$和 $v_{i',\ j'}$之间添加一条边（如果$|i-i'|+|j-j'|=1$），这样 X_v 就成为一个网格图，见图20。

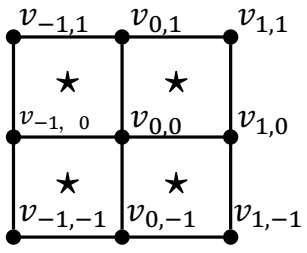

图20。 网格图X_v。

图20 中由 ★标出的网格图中的四个 4-环的内角均设为 90。在 $\mathcal{R}'$ 中，以确保 X_v 必须绘制为网格。我们将网格的四条边界路径写作如下。

$$P^v_{1,0} = (v_{1,1}, v_{1,0}, v_{1,-1}), \qquad P^v_{0,-1} = (v_{1,-1}, v_{0,-1}, v_{-1,-1}),$$
$$P^v_{-1,0} = (v_{-1,-1}, v_{-1,0}, v_{-1,1}), \qquad P^v_{0,1} = (v_{-1,1}, v_{0,1}, v_{1,1}).$$

这四条路径的连接按顺时针方向遍历网格的边界。

我们将与 v 关联的每条边分别对应到 $v_{0,\ 0}$ 的一个不同邻点，使其与给定的逆时针循环次序 E（v）以及给定正交径向表示 $\mathcal{R}$中围绕 v 的角所对应的角度赋值 ϕ 保持一致。参见图 21。由对称性可知，共有五种情况。在第一种情况下，v 仅与一条边 e_1 关联，我们将 e_1 对应到 $v_{1,\ 0}$。在第二种情况下，v 有两条关联边 e_1 和 e_2，满足 ϕ（e_1，e_2）$= 90$。位于 $\mathcal{R}$ 中；我们将 e_1 对应到 $v_{1,\ 0}$，并将 e_2 对应到 $v_{0,\ 1}$。其余情况类似。

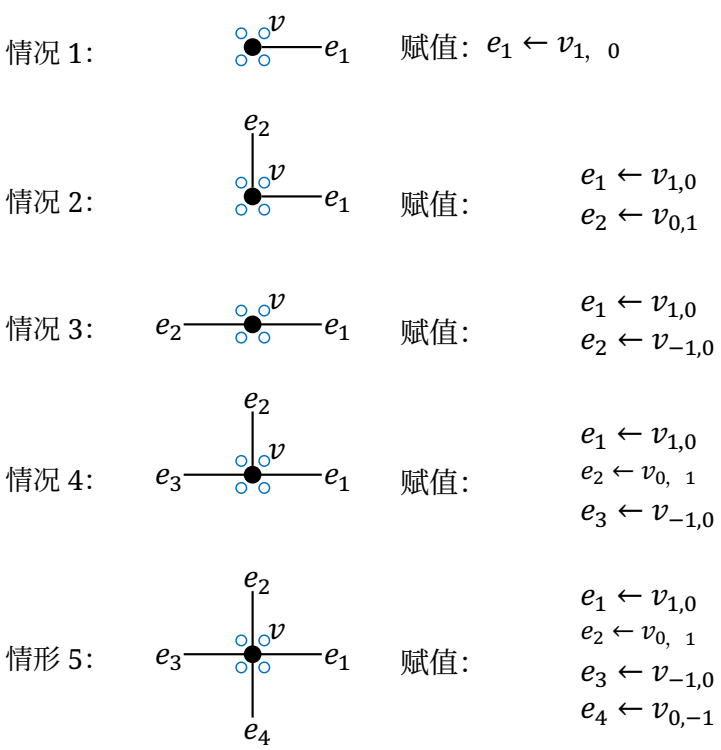

图 21. 将与 v 关联的每条边分别对应到 v_0的一个不同邻点，$_0$。

对于每个 $e = \{u, v\} \in E$，我们向 E' 添加边，以连接 X_u 和 X_v，如下所示。假设在上述赋值中，$e \leftarrow u_{i,j}$ 和 $e \leftarrow v_{k,l}$。然后，我们添加三条边 $e_1 = \{x_1, y_1\}$、$e_2 = \{x_2, y_2\}$ 和 $e_3 = \{x_3, y_3\}$，以连接 X_u 和 X_v，从而连接 $P_{i,j}^{u} = (x_1, x_2, x_3)$ 与 $P_{k,l}^{v} = (y_1, y_2, y_3)$。图 22 展示了 $e \leftarrow u_{1,0}$ 和 $e \leftarrow v_{0,1}$ 时的情况。

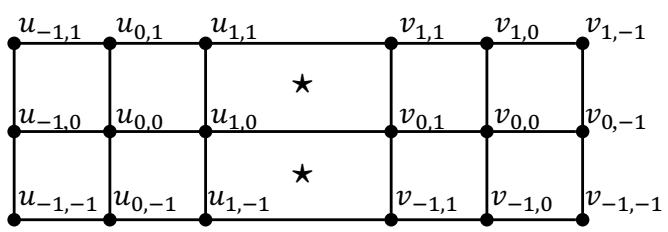

图 22. 向 E' 添加边，以连接 X_u 和 X_v。

添加三条边 e_1、 e_2 和 e_3 会形成两个 4-环，如图 22 中用 ★ 标出的部分所示。类似地，这两个 4-环的内角都设为 90◦，在 $\mathcal{R}'$ 中，以确保它们必须绘制成矩形。

至此，$(G',\mathcal{E}')$的构造完成，它是一个双连通简单平面图。$\mathcal{R}'$ 中所有其余的角都按照以下方式设定：围绕每个顶点的角度和为 360◦。

归约的有效性 为了证明该归约有效，我们需要证明$\mathcal{R}$ 可绘制，当且仅当$\mathcal{R}'$ 可绘制。我们先从$\mathcal{R} \to \mathcal{R}'$这个方向入手。假设给定了一个实现$\mathcal{R}$的正交径向绘图。我们的目标是找到一个实现$\mathcal{R}'$的正交径向绘图。对于每个顶点 $v \in V$，令(r_v, θ_v) 表示它在给定正交径向绘图中的坐标。我们定义

$$B_\epsilon(v) = \{(r,\theta) \mid r \in [r_v - \epsilon, r_v + \epsilon] \text{ and } \theta \in [\theta_v - \epsilon, \theta_v + \epsilon]\}.$$

对于每条边 $e = \{u, v\} \in E$，它被绘制为一条水平线（以原点为圆心的某个圆上的圆弧）或一条垂直线（经过原点的某条直线上的线段）。也就是说， e 的绘图可以由以下两个函数之一表示，参数$(r_e,\theta_{e,1},\theta_{e,2})$ 或$(r_{e,1},r_{e,2},\theta_e)$取某些值：

$$\begin{aligned}
&\{(r,\theta) \mid r = r_e \text{ and } \theta \in [\theta_{e,1}, \theta_{e,2}]\}, && \text{if } e \text{ is drawn as a horizontal line,}\\
&\{(r,\theta) \mid r \in [r_{e,1}, r_{e,2}] \text{ and } \theta = \theta_e\}, && \text{if } e \text{ is drawn as a vertical line.}
\end{aligned}$$

我们将 $B_\epsilon(e)$ 定义如下：

$$B_\epsilon(e) = \begin{cases}
\{(r,\theta) \mid r \in [r_e - \epsilon, r_e + \epsilon] \text{ and } \theta \in (\theta_{e,1} + \epsilon, \theta_{e,2} - \epsilon)\} & \text{if } e \text{ is drawn horizontally,}\\
\{(r,\theta) \mid r \in (r_{e,1} + \epsilon, r_{e,2} - \epsilon) \text{ and } \theta \in [\theta_e - \epsilon, \theta_e + \epsilon]\} & \text{if } e \text{ is drawn vertically.}
\end{cases}$$

通过令 $\epsilon > 0$ 足够小，我们可以确保对于所有 $v \in V$ 和 $e \in E$，集合 $B_\epsilon(v)$ 与 $B_\epsilon(e)$ 均非空且两两不相交。对于每个顶点 $v \in V$，我们在 $B_\epsilon(v)$ 中绘制网格 X_v，方法是绘制以下网格线：

$$\begin{aligned}
&\{(r,\theta) \mid r = r_v + s \cdot \epsilon \text{ and } \theta \in [\theta_v - \epsilon, \theta_v + \epsilon]\}, && \text{for } s \in \{-1, 0, 1\},\\
&\{(r,\theta) \mid r \in [r_v - \epsilon, r_v + \epsilon] \text{ and } \theta = \theta_v + s \cdot \epsilon\}, && \text{for } s \in \{-1, 0, 1\}.
\end{aligned}$$

对于每个 $e = \{u, v\} \in E$，我们通过在 $B_\epsilon(e)$ 中绘制以下三条线，来绘制连接 X_u 和 X_v 的三条边：

$\{(r,\theta) \mid r = r_e + s\cdot\epsilon$和 $\theta \in (\theta_{e,1} + \epsilon, \theta_{e,2} - \epsilon)\}$，当 e 水平绘制时，取 $s \in \in \{-1, 0, 1\}$；若 e 垂直绘制，则取 $s \in \in \{-1, 0, 1\}$，$\{(r,\theta) \mid r \in (r_{e,1} + \epsilon, r_{e,2} - \epsilon)$ 且 $\theta = \theta_e + s\cdot\epsilon\}$。

此绘图的有效性$\mathcal{R}'$源于集合 $B_\epsilon(v)$与 $B_\epsilon(e)$在所有 $v \in V$ 和 $e \in E$上的不相交性。

接下来，我们考虑另一个方向 R $\rightarrow$ R′。假设给定了一个实现 R′ 的正交径向绘图。我们的目标是找到一个实现 R 的正交径向绘图。对于每个顶点$v \in V$，我们将 v 放置在给定的 R′ 绘图中 $v_{0,0}$ 的位置。为了绘制每条边 $e = \{u, v\} \in E$，考虑归约中描述的赋值 $e \leftarrow u_{i,j}$ 和 $e \leftarrow v_{k,l}$。由于归约中描述的 R′ 的角度赋值，路径 $P = (u_{0,0}, u_{i,j}, v_{k,l}, v_{0,0})$ 必须绘制为直线。也就是说，在给定的 R′ 绘图中，P 被绘制为以原点为圆心的某个圆的圆弧，或绘制为经过原点的某条直线的线段。因此，我们可以使用 P 的绘图来嵌入 e，从而得到所需的 R 绘图。

8. 结论

本文提出了一种近线性时间算法，用于判定给定的正交径向表示是否可绘制，改进了此前的二次时间算法 [2]。如果该表示可绘制，我们的算法便会输出一个实现该表示的正交径向绘图。否则，算法会输出一个严格单调环，以证明此类绘图不存在。鉴于拓扑-形状-度量框架在正交绘图中的广泛应用，我们预计，新的正交径向绘图算法将在该领域的未来研究中发挥作用并具有实用价值。

尽管正交绘图领域已有大量研究，但正交径向绘图中基本优化问题的计算复杂度仍有许多未知之处。特别是，寻找能使弯折数量最少的正交径向表示这一问题，目前仅有一种实用算法 [36] 对此进行处理，但该算法没有可证明的性能保证。弯折最小化在多大程度上可通过多项式时间算法求解，仍是一个引人深思的开放问题。[1] 据我们所知，甚至尚不清楚判定给定的平面图是否存在无弯折正交径向绘图能否在多项式时间内完成。

给定一个正交径向表示，我们能否在多项式时间内找到层数最少的正交径向绘图（即同心圆的数量）？ 如第 3 节所述，如果给定一个良序列，我们的算法就能输出层数最少的绘图。对于可能不存在良序列的一般情况，我们的算法无法保证绘图的层数最少，因为添加虚拟边的选择具有一定灵活性，而选择不同的虚拟边会产生不同的良序列。

1 论文 [2] 还将为给定平面图计算最少弯曲正交径向表示列为一个开放问题。在同一篇论文的预印本（arXiv:2106.05734v1） [2], 中，作者声称该问题是 NP-困难的（定理7），似乎已解决了这一问题。然而，这一结果在期刊版本中被删除，表明他们的归约可能存在问题，因此该问题仍未解决。

已有一系列研究根据各种复杂度度量来寻找紧凑正交绘图 [6, 12, 1, 33, 40]。这些研究中提出的思路在多大程度上可以应用于正交径向绘图?

参考文献

[1] MichaelJBannister，DavidEppstein，以及 JosephA Simons。正交紧凑化的不可近似性。图算法与应用杂志 *(JGAA)*，16(3):651–673，2012。 (51)

[2] Lukas Barth，Benjamin Niedermann，I gnazRutter，以及 MatthiasWolf。正交径向图绘制的拓扑-形状-度量框架。离散与计算几何，70(4):1292–1355，2023。 (3–16, 42, 50)

[3] Lukas Barth，Benjamin Niedermann，I gnazRutter，以及 MatthiasWolf。迈向正交径向绘图的拓扑-形状-度量框架。第*33*届国际计算几何研讨会 *(SoCG)*，第77卷，莱布尼茨国际信息学论文集 *(LIPIcs)*，14:1–14:16，达格施图尔，德国，2017。 (4, 5)

[4] HannahBast、Patrick Brosi、和 Sabine Storandt。灵活基础网格上的地铁线路图。第*17*届国际时空数据库研讨会（*SSTD*）论文集，第12–22页，2021年。 (3)

[5] CarloBatini、EnricoNardelli、和 Roberto Tamassia。一种用于数据流图的布局算法。*IEEE*软件工程汇刊，SE-12(4):538–546，1986年。 (2)

[6] Michael A Bekos、CarlaBinucci、GiuseppeDiBattista、WalterDidimo、Martin Gronemann、KarstenKlein、MaurizioPatrignani、和 Ignaz Rutter。关于转弯正则正交表示。图算法与应用杂志（*JGAA*），26(3):285–306，2022年。 (51)

[7] Sandeep N Bhatt 和 Frank Thomson Leighton。一种求解 VLSI图布局问题的框架。计算机与系统科学杂志，28(2):300–343，1984年。 (2)

[8] Therese Biedl，Anna Lubiw，MarkPetrick，以及 Michael Spriggs。正交平面图绘图变形。*ACM*算法汇刊（*TALG*），9(4):1–24，2013年。 (7)

[9] ThomasBl¨asius，IgnazRutter，以及 DorotheaWagner。具有凸弯折代价的最优正交图绘制。*ACM*算法汇刊（*TALG*），12(3):33:1–33:32，2016年。 (7)

[10] Ulrik Brandes，SabineCornelsen，Christian Fieß，和 DorotheaWagner。如何绘制平面图的最小割。计算几何，29(2):117–133，2004年。 (7)

[11] UlrikBrandes 和 DorotheaWagner。少弯折与变更的动态网格嵌入。第九届国际算法与计算研讨会（*ISAAC*）论文集，第 90–99 页。施普林格，1998年。 (7)

[12] Stina SBridgeman，GiuseppeDi Battista，Walter Didimo，GiuseppeLiotta，Roberto Tamassia，以及 Luca Vismara。正交表示的转弯正则性与最优面积绘图。计算几何，16(1):53–93，2000年。 (51)

[13] Yi-Jun Chang 和 Hsu-Chun Yen。平面 3-图的最少弯折正交绘图。第*33*届国际计算几何研讨会（*SoCG*），第77卷，载于莱布尼茨国际信息学论文集（*LIPIcs*），29:1–29:15，德国达格施图尔，2017年。 (7)

[14] Yi-JunChang。近线性时间的正交径向绘图。第*50*届自动机、语言与编程国际研讨会，*ICALP 2023*，*2023*年*7*月*10*日至*14*日，德国帕德博恩，LIPIcs第*261*卷，35:1–35:20。达格施图尔-莱布尼茨信息学中心¨，2023年。 (1)

[15] SabineCornelsen 和 AndreasKarrenbauer。加速弯折最小化。图算法与应用期刊（*JGAA*），16(3):635–650，2012年。 (2, 6)

[16] GiuseppeDi Battista，WalterDidimo，Maurizio Patrignani，以及 MaurizioPizzonia。具有指定尺寸顶点的正交绘图和准向上绘图。第*7*届国际图绘制研讨会（*GD*）会议录，第297–310页，1999年。 (7)

[17] GiuseppeDi Battista，GiuseppeLiotta，和 FrancescoVargiu。螺旋度与最优正交绘图。*SIAM*计算杂志，27(6):1764–1811，1998年。 (2, 7)

[18] WalterDidimo，Giuseppe Liotta，Giacomo Ortali，和 MaurizioPatrignani。平面 3-图的线性时间最优正交绘图。第*31*届*ACM-SIAM*离散算法年会（*SODA*）论文集，第806–825页。工业与应用数学学会，2020年。 (2, 7)

[19] Walter Didimo，GiuseppeLiotta，和 Maurizio Patrignani。二次时间内的最少弯曲正交图绘制。第*26*届国际图绘制与网络可视化研讨会论文集（*GD*），第481–494页。施普林格，2018年。 (7)

[20] WalterDidimo，GiuseppeLiotta，和 MaurizioPatrignani。HV-直角平面性测试的复杂度。第 *22*届图绘制研讨会论文集（*GD*），第343–354页。施普林格，2014年。 (7)

[21] Sally Dong，Yu Gao，GramozGoranci，Yin TatLee，RichardPeng，SushantSachdeva，和 Guanghao Ye。嵌套剖分与内点法相结合：近线性时间内的平面最小费用流。*2022*年*ACM-SIAM*离散算法年会论文集（*SODA*），第 124–153 页。工业与应用数学学会，2022年。 (7)

[22] Christian A Duncan 和 Michael T Goodrich。平面正交绘图与折线绘图算法。Roberto Tamassia 编，《图绘制与可视化手册》，第 8 章。CRC出版社，2013年。(6)

[23] Stephane Durocher、Stefan Felsner、Saeed Mehrabi 和 Debajyoti Mondal。绘制 HV受限平面图。第 11 届拉丁美洲理论信息学研讨会（LATIN）论文集，第 156–167 页。施普林格，2014年。数字对象标识符 (7)[24] Markus Eiglsperger、Carsten Gutwenger、Michael Kaufmann、Joachim Kupke、Michael J¨unger、Sebastian Leipert、Karsten Klein、Petra Mutzel 和 Martin Siebenhaller。以正交风格自动布局 UML类图。《信息可视化》，3(3):189–208，2004年。数字对象标识符 (2)[25] MartinFink、MagnusLechner 和 AlexanderWolff。《同心地铁图》。示意地图绘制研讨会（SMW）论文集，2014年。(3)[26] Michael Forman、Torben Hagerup、James Haralambides、Michael Kaufmann、Frank Thomson Leighton、Antonios Symvonis、Emo Welzl 和 G Woeginger。以高分辨率在平面上绘制图。SIAM计算杂志，22(5):1035–1052，1993年。数字对象标识符 (6, 7)[27] Ashim Garg 和 Roberto Tamassia。一种新的最小费用流算法及其在图绘制中的应用。第 4 届图绘制研讨会（GD）论文集，第 201–216 页，1997年。数字对象标识符 (2, 6, 7)[28] Carsten Gutwenger、Michael J¨unger、Karsten Klein、Joachim Kupke、Sebastian Leipert 和 Petra Mutzel。一种可视化 UML类图的新方法。2003年ACM软件可视化研讨会（SoftVis）论文集，第 179–188 页，2003年。数字对象标识符 (2)[29] Mahdieh Hasheminezhad、S Mehdi Hashemi、Brendan D McKay 和 Maryam Tahmasbi。三次平面图的矩形径向绘图。《计算几何》，43(9):767–780，2010年。数字对象标识符 (7)[30] Mahdieh Hasheminezhad、S Mehdi Hashemi 和 Maryam Tahmasbi。图的正交径向绘图。《澳大拉西亚组合学杂志》，44:171–182，2009年。(7)[31] Min-Yu Hsueh。集成[illegible]的符号布局与压缩。博士论文，加州大学伯克利分校，1980年。(2)

[32] Steve Kieffer、Tim Dwyer、Kim Marriott 和 Michael Wybrow。HOLA：类人正交网络布局。《IEEE 可视化与计算机图形学汇刊》，22(1):349–358，2016年。数字对象标识符 (2)[33] Gunnar W Klau 和 Petra Mutzel。正交网格绘图的最优紧缩。第七届整数规划与组合优化会议（IPCO）论文集，第304–319页。施普林格，1999年。数字对象标识符 (51)[34] Gunnar W.Klau 和 Petra Mutzel。平面图的准正交绘图。技术报告 MPI-I-98-1-013，马克斯·普朗克信息学研究所，萨尔布吕肯，1998年。 (7)[35] Robin S Liggett 和 William J Mitchell。实践中的最优空间规划。《计算机辅助设计》，13(5):277–288，1981年。专题：设计优化。数字对象标识符 (2)[36] Benjamin Niedermann 和 Ignaz Rutter。用于正交径向绘图弯折最小化的整数线性规划。第28届国际图绘制与网络可视化研讨会论文集，第235–249页。施普林格，2020年。数字对象标识符 (50)[37] Benjamin Niedermann、Ignaz Rutter 和 Matthias Wolf。正交径向图绘制的高效算法。第[illegible]届国际计算几何研讨会（SoCG），莱布尼茨国际信息学论文集（LIPIcs）第129卷，53:1–53:14，德国达格施图尔，2019年。数字对象标识符 (4, 5)

[38] Mark Ovenden。《世界地铁地图》。Capital Transport 出版社，2003年。 (3)[39] Achilleas Papakostas 和 Ioannis G Tollis。高阶图的高效正交绘图。《Algorithmica》，26(1):100–125，2000年。数字对象标识符 (7)[40] Maurizio Patrignani。论正交紧凑化的复杂性。《计算几何》，19(1):47–67，2001年。数字对象标识符 (51)[41] Maxwell J Roberts。《贝克之后的地铁地图：Henry Beck 后继者手中的伦敦地铁地图故事》。Capital Transport 出版社，2005年。 (3)[42] Maxwell J Roberts。《地铁地图解读：信息设计探索》。作者设计、出版并发行，2012年。 (3)[43] James A Storer。平面网格上图嵌入的节点成本度量。第12届年度ACM计算理论研讨会（STOC）论文集，第201–210页，1980年。数字对象标识符 (2)[44] Roberto Tamassia。论如何以最少弯折数将图嵌入网格。《SIAM计算杂志》，16(3):421–444，1987年。数字对象标识符 (2, 6, 7)[45] Leslie G Valiant。超大规模集成电路中的通用性考量。《IEEE Transactions on Computers》，100(2):135–140，1981年。数字对象标识符 (2)

[46] Hsiang-YunWu，BenjaminNiedermann，Shigeo Takahashi，Maxwell JRoberts，和 MartinN¨ollenburg。关于交通线路图布局的综述 —— 从设计、机器与人的视角出发。*ComputerGraphics Forum*，39(3):619–646，2020年。(3)

[47] Yingying Xu、Ho-YinChan、和 Anthony Chen。使用混合整数优化自动生成同心圆地铁线路图。*International Journalof GeographicalInformationScience*：1–26，2022年。(3)

本作品依据知识共享署名 4.0 国际许可协议授权。
http://creativecommons.org/licenses/by/4.0/© Yi-Jun Chang。