arXiv:2601.20036v1 [cs.CG] 27 Jan 2026

包含许多点的圆盘的计算方面[♈]

Prosenjit Bose[♉]、*Guillermo Esteban*[♌] 和 *Tyler Tuttle*[♉]

摘要。设S 是平面中的一个包含n 个点的点集。我们提出了几种不同的算法，用于寻找S中的一对点，使得任何包含该点对的圆盘都必须包含S中至少cn 个点，其中$c > 0$为某个常数。第一种是随机化算法，对于处于一般位置的点，它能在~~$O(n \log n)$~~期望时间内找到这样一个点对，并且$c = 1/2 - \sqrt{(1+2\alpha)/12}$，对于任意 $0 < \alpha < 1$均成立。第二种算法同样适用于处于一般位置的点，虽然需要二次时间，但将常数c 提高到$1/2 - 1/\sqrt{12} \approx 1/4.7$。第二种算法还可以作为子程序使用，以$O(n^2 \log n)$时间找到使包含该点对的任意圆盘内点数最大化的点对。我们还考虑了该问题的变体。当点集S 处于凸位置时，我们给出一种线性时间算法，能够找到一对点，使得任何经过这两点的圆盘都包含S中至少$n/3$个点。对于另一种变体，即只关注寻找一对点，使得该点对的直径圆盘包含许多点，我们同样给出一种线性时间算法，能够找到一个包含S中至少$n/3$个点的圆盘。最后，我们将前两种算法推广到点集S 中的点使用两种颜色着色的情形。我们还考虑调整这些算法，以解决S 为简单多边形P内部的点集时的相同问题，此时将圆盘的概念替换为测地圆盘。

1 引言

设 S 是平面上的 n 个点的集合。如果点 p ∈S 位于圆盘的边界上或内部，则称该圆盘包含或包围点 p。对于任意两个不同的点 p、q ∈S，我们将 C$_S$(p, q) 定义为满足以下条件的最小整数：任何包围 p 和 q 的圆盘都至少包含 S 中的 C$_S$(p, q) 个点。令 Π(S) 表示 C$_S$(p, q) 在 S 中所有点对 {p, q} 上的最大值，并令 Π(n) 表示所有 n 个点的集合 S 中 Π(S) 的最小值。求 Π(n) 的精确值这一问题由 Neumann-Lara 和 Urrutia 于 1988年 [21] 提出。他们证明，S 中存在一个点对，使得任何包含该点对的圆盘都必须包含 S 中至少 cn 个点，其中 c > 1/60 为某个常数。在接下来约一年时间里，这一结果中的常数因子不断得到改进，最终 Edelsbrunner 等人 [14] 证明，存在一个点对，使得任何包含该点对的圆盘还至少包含 S 中的 n/4.7 个点。最近，Bose 等人研究了将问题中的平面替换为简单多边形 [8] 时的情形。

这些结果都证明了点对的存在性，但没有一个研究如何找到这样一对点的算法问题。这正是本文的重点。用于证明存在性的许多思路，也有助于我们为欧几里得情形和测地情形设计算法。

[♈]这是发表于 LAGOS 2025 论文集中的一篇论文的扩展版本。[♉]卡尔顿大学，jit@scs.carleton.ca，tylertuttle@cmail.carleton.ca[♌]阿尔卡拉大学，g.esteban@uah.es

1.1 以往工作

这一组合问题由 Neumann-Lara 和 Urrutia [21] 于 1988年 提出。他们证明，在 S 中存在一对点 p、q，使得任何包含 p 和 q 的圆盘都至少包含 S 中 $\lceil (n-2)/60 \rceil$ 个点。他们还将 $\Pi(n)$ 定义为所有由 n 个处于凸位置的点组成的集合 S 中， $\Pi(S)$ 的最小值，并证明 $\Pi(n) \geq \lceil (n-2)/4 \rceil$。随后，Bárány 等人 [7] 将下界改进为 $\Pi(n) \geq n/30$。与此同时，Hayward 等人 [17], 也研究了这个问题，并证明 $\lfloor n/27 \rfloor + 2 \leq \Pi(n) \leq \lceil n/4 \rceil + 1$，且 $\Pi(n) = \lceil n/3 \rceil + 1$。对于处于非凸位置的点集，其下界后来由 Hayward [16] 改进为$\Pi(n) \geq \lceil 5(n-2)/84 \rceil$。最后，Edelsbrunner 等人 [14] 证明了

$$\Pi(n) \geq \frac{n-3}{2} - \sqrt{\frac{(n-2)^2-1}{12}} = \left(\frac{1}{2} - \frac{1}{\sqrt{12}}\right)n + O(1) \approx \frac{n}{4.7},$$

这仍然是关于 $\Pi(n)$ 的最佳下界。Ramos 和 Viaña [25], 给出了同一下界的另一种证明，他们证明了更强的结论：总是存在两个点 p 和q，使得任何以p 和q 为边界上两点的圆盘，至少包含$n/4.7$ 个点，并且圆盘外至少有$n/4.7$ 个点。Claverol等人 [11] 利用高阶 Voronoi 图给出了这一事实的另一种证明。

多年来，人们研究了这一问题的各种变体。Akiyama等人 [3] 证明，存在两个点p 和 q ，使得以线段pq 为直径的圆盘至少包含$n/3$ 个点。Bárány等人 [7] 在 $\mathbb{R}^d$中研究了这一问题。具体而言，他们证明：给定 $\mathbb{R}^d$中的一个S 点集，其中包含n 个点，则存在S 的一个子集T，其大小为 $\lfloor (d+3)/2 \rfloor$，使得任何包含T 的 d 维球，都至少包含S中的cn 个点，其中$c > 0$为某个常数。Bárány 和 Larman 进一步将这一结果从球推广到了二次曲面 [6]。此外，[11, 24, 26], 还研究了该问题的双色版本，其目标是找到一对双色点，使得任何包含这对点的圆盘都必须包含许多点。 [24] 的作者还将这一问题的彩色版本推广到了更高维度。

Bose 等人 [8] 将许多这类变体从平面推广到了包含在简单多边形中的点集。他们证明，对于简单多边形 P 中的 S 个 n 点，存在 S 中的两个点，使得任何以这两个点为边界点的测地圆盘都至少包含$n/5$ 个S中的点。对于处于测地凸位置的点集，他们证明存在一对点，使得任何包含这对点的测地圆盘都至少包含$n/3$ 个点，这与欧几里得情形的界相匹配。他们还研究了该问题的直径版本和双色版本，以及同时涉及圆盘内外点的版本。

1.2 我们的结果

本文介绍了多种算法：给定一个由n 个点构成的集合S ，找出一对点，使得任何包含这两个点的圆盘都包含S中常数比例的点。前两种算法针对处于一般位置的点集。其中一种是随机化算法，可在$O(n \log n)$期望时间内，找出一对点p和q，使得任何同时包含p和q 的圆盘都包含S中至少cn 个点，其中任意常数 $0 < c < 1/4.7$。该算法运行时间中的常数取决于c的取值。第二种是二次时间算法，可找出一对点p 和q ，使得任何同时包含p 和q 的圆盘都包含S中至少$n/4.7$个点，

这与 $\Pi(n)$ 的已知最佳下界相匹配。第二种算法还可在$O(n^2 \log n)$时间内，找出使$Cs(p,q)$最大的点对。第三种算法适用于处于凸位置的点集，可找出一对点p 和q ，使得任何同时包含p 和q 的圆盘都包含S中至少$n/3$个点；这是 $\Pi(n)$ 的最优界。第三种算法的运行时间为线性时间。第四种算法针对问题的直径版本，可在$O(n)$时间内找出p 和q，使得以pq 为直径的圆盘包含S 中至少$n/3$个点。最后，我们说明如何修改前两种算法，以适用于问题的双色版本：给定一个集合S ，其中包含$n/2$个红点和$n/2$个蓝点，找出一对红蓝点，使得任何包含这对点的圆盘都包含S中至少cn 个点，其中c为常数。

我们还将讨论如何将这些算法推广到简单多边形中的点集。需要谨慎处理，因为在平面中许多显而易见的性质在这种情形下会变得更加复杂。例如，在平面中，求经过三个点的圆盘中心只需常数时间。而在简单多边形中，这样的圆盘甚至可能不存在。我们将展示如何处理由此产生的所有困难。

2 背景

本节将介绍算法所需的背景知识，包括平面中的情形（欧几里得情形）和简单多边形中的情形（测地情形）。

2.1 欧几里得情形

给定两个点 p 和q，我们用pq表示连接p 和q 的线段，用 $|pq|$表示其长度，用$L(p,q)$表示经过p和q的直线，并用$b(p,q)$表示p和q的平分线。直线$L(p,q)$将平面分成两个开半平面。设$H(p,q)$为沿p到q的方向、位于$L(p,q)$左侧的半平面，设$H(q,p)$为位于$L(p,q)$右侧的半平面。如果点p 位于圆盘D的内部或边界上，我们称圆盘D 包含点p 。如果点p 位于圆盘D的边界上，我们称D 经过点p 。

给定平面上的点集S 以及S中的两个点p 和q ，使得任何包含p和q的圆盘都至少包含S中的k个其他点的最小整数k记作$Cs(p,q)$。使得任何以p 和q 为边界上两点的圆盘都至少包含S中的k 个其他点，且至多包含S中的$n-k-2$个其他点的最小整数k （或者等价地，S中至少有k个点在圆盘˜之外）记作$Cs(p,q)$。

在本文余下部分中，我们假设S 处于一般位置，即S 中任意三点不共线，且S 中任意四点不共圆。

2.2 测地情形

设 P 为一个简单多边形。给定 P 中的一对点 p 和 q，从 p 到 q 的测地线（最短）路径记为 g(p, q)。已知 g(p, q) 是一条折线链 [19]。该路径的长度记为 |g(p, q)|。P 中到 p 和 q 距离相等的所有点的集合称为 p 和 q 的平分线，记为 b(p, q)。在 P 的任何顶点到 p 和 q 的距离都不相等的假设下，已知 b(p, q) 是一条由线段和双曲弧组成的简单分段定义曲线 [4]。p 到 q 的延伸路径记为 l(p, q)，其定义为将 g(p, q) 的第一条和最后一条线段延伸，直到它们分别与 P 的边界适当地相交。延伸路径 l(p, q) 自然地将多边形划分为三部分：位于 l(p, q) 左侧的点集、

位于 $l(p,q)$ 右侧的点集，以及 $l(p,q)$ 本身。如果 u 位于 $g(p,q)$ 的左侧（相应地，右侧），我们就称点 u 位于 $l(p,q)$ 的左侧（相应地，右侧）。

若子集Q 属于P，且对于Q中的任意点对p 和q，测地路径$g(p,q)$都包含于Q，则称Q为测地凸 集。点集S 在P 中的测地凸包 是包含S的最小测地凸集。若S中的每个点都位于S的测地凸包的边界上，则称S 处于测地凸位置 。三个点$p,q,u \in P$上的测地三角形 记作△pqu，它是由$g(p,q)$、$g(q,u)$和$g(u,p)$围成的区域。

以$o \in P$为中心、半径为 $\rho \geq 0$的测地圆盘 是集合$D(o,\rho) = \{p \in P: |g(o,p)| \leq \rho\}$。与平面中的情况类似，若$p$ 满足 $|g(o,p)| \leq \rho$，则称$D(o,\rho)$包含p；若p 满足 $|g(o,p)| = \rho$，则称$D(o,\rho)$经过p。

至少包含三个点的集合 S 称为测地共线，如果存在两个点 p和 q 属于 S，使得 S 是 $g(p,q)$ 的子集。至少包含四个点的集合 S 称为测地共圆，如果存在一个测地圆盘 $D(o,\rho)$，使得 S 包含在其边界上，即对于所有属于 S 的 p，$|g(o,p)| = \rho$。与平面情形一样，对于简单多边形中的任意点集 S，我们作一般位置假设：没有三个点测地共线，没有四个点测地共圆，并且没有两个点到 P的某个顶点距离相等。

2.3 高阶 Voronoi 图

由于我们在欧几里得情形和测地情形中的许多结果都依赖于高阶 Voronoi 图，因此这里简要概述一下。更详细的介绍请参阅 Edelsbrunner [13]。我们将在欧几里得情形中定义这些图，但请注意，经过适当修改后，所有定义都适用于测地情形：将欧几里得距离替换为测地距离，等等。

设S 是平面上的一个点集。点p 的 Voronoi 区域，是平面上所有到p 的距离比到S中任何其他点都近的点的集合。众所周知，p 的 Voronoi 区域非空（因为其中必然包含p）且连通，它构成一个多边形的内部，而该多边形的边是形如$b(p,q)$的平分线线段，其中q 是S中的另一个点。S 的 Voronoi 图是由这些 Voronoi 区域定义的平面剖分，记作$V(S)$。

定义如下。设k 阶 Voronoi 图中，T 是S 的一个子集，且其大小为k。T 的k 阶 Voronoi 区域，是平面上所有到T 中每个点的距离都比到$S \setminus T$中任一点的距离近的点的集合。也就是说，该集合由满足以下条件的点r构成：$|pr| < |qr|$，其中p 是T中的任意点，q 是$S \setminus T$中的任意点。注意，标准 Voronoi 图就是 1 阶 Voronoi 图。我们将k阶 Voronoi 图记作$V_k(S)$。阶数为（$n - 1$）的 Voronoi 图也称为最远点 Voronoi 图。

与 1 阶 Voronoi 图不同，集合 T 的 Voronoi 区域可能为空。不过，与 1 阶 Voronoi 区域一样，任何非空的 k 阶 Voronoi 区域都必须是一个多边形的内部，其边是形如 $b(p,q)$ 的垂直平分线，其中 p 属于 T，而 q 属于 $S \setminus T$。

3 平面算法

本节介绍三种算法，用于在欧几里得平面中找到一对点p 和q，使得$Cs(p,q) \geq cn$，其中$c > 0$。

3.1 计算平分线上的最小权重线段

我们首先介绍一种算法：给定一个包含 n 个点的点集 S，以及 S 中的两个点 p 和 q，该算法可在 O(n log n) 时间内计算$\tilde{C}_S(p,q)$和 $C_S(p,q)$。考虑所有经过 p 和 q 的圆盘。其中有 $n-2$ 个圆盘经过 S 中的第三个点。这些圆盘的圆心都位于 b(p, q) 上，并将直线 b(p, q) 分成 $n-1$ 个开线段，其中两个是无界的。圆心位于同一线段 s 上的两个经过 p 和 q 的圆盘，包含相同的点集。s 的权重定义为以 s 上的点为圆心、经过 p 和 q 的任意圆盘内部的点数。我们将 s 的权重记为 $\omega(s)$。请注意，p 和 q 位于圆盘的边界上，而不在其内部，因此若圆盘内没有其他点，则 $\omega(s)$ 等于 0。还请注意，圆心位于 s 上且经过 p 和 q 的圆，其外部有 $n-\omega(s)-2$ 个点。显然，我们有

$$C_S(p,q) = \min\{\omega(s) : s \text{ is a segment of } b(p,q)\}, \text{ and} \tag{1}$$

$$\tilde{C}_S(p,q) = \min\{\min\{\omega(s), n-\omega(s)-2\} : s \text{ is a segment of } b(p,q)\}. \tag{2}$$

此外，共享一个端点的两条线段s 和s' 的权重恰好相差 1。最后这一观察是我们算法的关键：我们可以沿$b(p,q)$对线段排序，计算其中一条无界线段的权重，然后沿平分线依次高效地计算其余线段的权重。

算法 1 计算$C_S(p,q)$和 $\tilde{C}_S(p,q)$

1：对 S \ {p,q} 中的每个点 x_i，计算经过 p、q 和 x_i 的圆的中心 c_i 2：按 L(p, q) 的方向，沿 b(p, q) 从左到右对 $c_1, c_2, \ldots, c_{n-2}$ 排序 3：$\omega \leftarrow |S \cap H(p,q)|$ 4：$C \leftarrow \omega$ 5：$C \leftarrow \min\{\omega, n-\omega-2\}$ 6：对于 $i \leftarrow 1, 2, \ldots, n-2$，执行 7：如果 x_i 在 L(p, q) 的左侧，则 8：$\omega \leftarrow \omega - 1$ 9：否则 10：$\omega \leftarrow \omega + 1$ 11：结束 if 12：$C \leftarrow \min\{C, \omega\}$ 13：$C \leftarrow \min\{C, \omega, n-\omega-2\}$ 14：结束 for 15：返回 C 和 C

引理 1. 设 S 是平面上的 n 个点的集合，且 p 和 q 是 S 中的点。$\tilde{C}_S(p,q)$和 $C_S(p,q)$的值可在 $O(n \log n)$时间内计算得出。

证明。 参见算法 1。考虑无界线段 s ，它位于 $b(p,q)$ 上，并由 c_1界定。一个经过 p和 q 且中心位于该线段上的圆盘包含 $S \cap H(p,q)$中的所有点。因此，$\omega(s) = |S \cap H(p,q)|$。

假设我们已知线段 s 的权重，并令 s' 为沿$b(p,q)$的相邻线段，且从$L(p,q)$的视角看，s' 位于s的右侧。令x_i为点

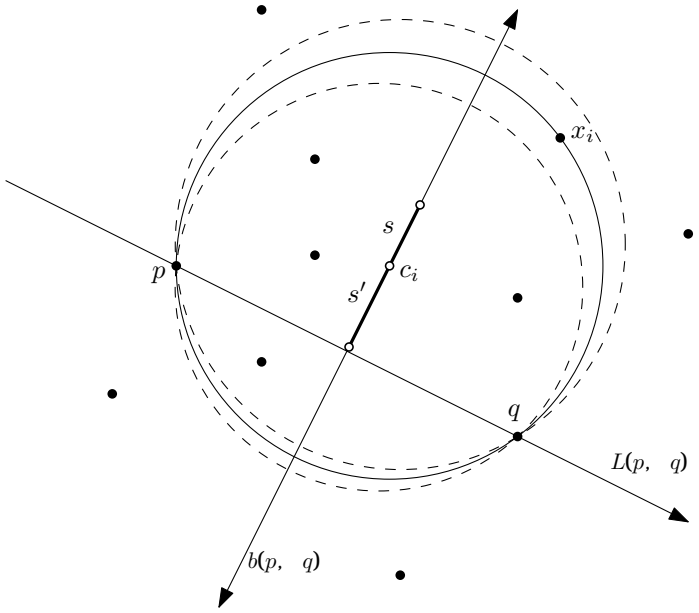

图1：x_i 位于$L(p,q)$的左侧，因此 $\omega(s) = 5$ 和 $\omega(s') = 4$。虚线圆盘是经过p 和q 且圆心位于 s 和s'中的点上的圆盘。

S 中满足 c_i 是 s 与 s′ 的公共端点的点，见图1。若 x_i 位于从 p 指向 q 的直线 L(p, q) 左侧，则 $\omega(s') = \omega(s) - 1$。否则，若 u 位于 L(p,q) 的右侧，则 $\omega(s') = \omega(s) + 1$。因此，我们可以根据 $\omega(s)$ 的值，在常数时间内计算 $\omega(s')$ 的值。显然，算法由此可以正确计算 $C_S(p, q)$ 和 $\tilde{C}_S(p, q)$ 的值。

构造圆心列表$\{c_1, c_2, \ldots, c_{n-2}\}$需要$O(n)$时间，因为在平面上求过三个点的圆的圆心可以在常数时间内完成。沿着$b(p,q)$对这些圆心排序需要$O(n \log n)$时间。计算 ω的初始值——$|S \cap H(p,q)|$——需要$O(n)$时间。最后，循环的每次迭代都需要常数时间，因此整个循环需要$O(n)$时间。该算法的总运行时间主要由圆心排序所需的时间决定。 □

3.2 寻找一对点的随机化算法

现在我们已经有了计算 $C_S(p, q)$ 和 $\tilde{C}_S(p, q)$的算法，下面给出一个算法，用于$\tilde{}$寻找一对{p, q}，使得 $C_S(p, q) \geq cn$，其中 $c > 0$为某个常数：反复均匀随机地选择一对{p, q}$\tilde{}$并计算 $C_S(p, q)$，直到 $C_S(p, q)$的值至少为 cn。注意，这$\tilde{}$也给出了一个算法，用于寻找一对满足 $C_S(p, q) \geq cn$ 的点，因为 $C_S(p, q) \geq C_S(p, q)$。由于——至少在平面上——目前已知的 Π(n)和 Π(n)的下界仅在低阶项上有所差异，因此我们只介绍一种适用于过 p 和 q 的任意圆内部和外部的点的算法。

考虑由 S 中的点对所确定的 $\binom{n}{2}$ 平分线集合。令 S_k 表示至少含有一条权重小于 k 或大于 $n - k - 2$的线段的平分线数量。Ramos 和 Viaña [25]证明了，$S_k \leq 3(k + 1)n - 3(k + 1)(k + 2)$。

定理 1. *L*　设 $0 < \alpha < 1$ 为常数，并设 S 是平面上 n 个点的集合。至少存在

$\alpha\binom{n}{2}$ 对点对 $\{p, q\}$，其中 p、q 属于 S，使得 $\tilde{C}s(p, q) \geq (1/2 - \sqrt{(1+2\alpha)}/12)n + O(1)$。这样的点对可以在 $O(n \log n)$ 期望时间内找到；或者以 $1 - 1/n$ 的概率，在 $O(n\log^2 n)$ 时间内找到。

证明。 如果$S_k < (1-\alpha)\binom{n}{2}$，那么至少有 $\alpha\binom{n}{2}$ 条垂直平分线，其上不存在线段权重小于 k 或大于 $n-k-2$；这意味着存在相应数量的点对 $\{p,q\}$，使得每个经过p和q的圆盘都包含至少 k 个点，且至多包含 $n-k-2$ 个点。

利用 Ramos 和 Viaña 关于S_k的上界，我们注意到，只要 $3(k+1)n - 3(k+1)(k+2) < (1-\alpha)\binom{n}{2}$，就有$S_k < (1-\alpha)\binom{n}{2}$。这是关于$k$的一个二次方程，其较小的解为

$$k < \frac{n-3}{2} - \left(\frac{(1+2\alpha)n^2 - (4+2\alpha)n + 15}{12}\right)^{1/2} = \left(\frac{1}{2} - \sqrt{\frac{1+2\alpha}{12}}\right)n + O(1). \tag{3}$$

较大的解大于 $(n-3)/2$，因此超出了 $\tilde{C}s(p,q)$ 的取值范围。

我们知道，均匀随机选取的点对 $\{p,q\}$ 以概率 α 满足$\tilde{C}s(p, q) \geq (1/2 - \sqrt{(1+2\alpha)}/12)n + O(1)$。根据引理 1，$\tilde{C}s(p, q)$ 的值可在 $O(n \log n)$ 时间内计算。成功之前必须测试的点对数量的期望为 α^{-1}，因此期望运行时间为 $O(\alpha^{-1} n \log n)$。

如果我们选择 $\log 1 \quad {}^{(1-\alpha)^{-1}} \quad 1 - 1 =$ (n) 对，那么没有 成功的概率是 $(1-\alpha)^{\log_{(1-\alpha)^{-1}}(n)}/n$。因此，成功的概率为$/n$，并且以高概率，我们将在$O(n\log^2 n)$ 时间内找到一对点，使得 $\tilde{C}s(p,q) \geq (1/2 - \sqrt{(1+2\alpha)}/12)n + O(1)$。 □

我们选择的 α 值存在一个权衡。如果 α 非常接近 1，这意味着随机选中一对合适点的概率很高，那么圆盘内外点的数量上界就趋近于 0。另一方面，随着 α 趋近于 0，圆盘内外点的数量上界则趋近于 Ramos 和 Viaña 证明的上界$n/4$。。

例如，如果 $c = 1/2$，那么最优的 k 值约为 $\left(\frac{1}{2} - \frac{1}{\sqrt{6}}\right)n \approx n/10.9$。因此，如果我们均匀随机地选择一对点 $\{p,q\}$，那么$Cs(p,q) \geq n/10.9$ 的概率为 1/2。

我们注意到，对每一对点 $\{p,q\}$ 计算 $Cs(p,q)$，即可得到一个简单的蛮力算法，用于寻找使 $Cs(p,q)$ 最大化的点对。由于共有 $\binom{n}{2}$ 对点，因此该算法需要 $O(n^3 \log n)$ 时间。下一节将介绍一种算法，用于寻找使 $Cs(p,q)$ 最大化的点对，其运行时间为 $O(n^2 \log n)$。

观察 1. 给定平面中由 n 个点组成的集合 S，可以在 $O(n^3 \log n)$ 期望时间内，找到一对点 p、$q \in S$，使其在所有点对中达到 $Cs(p, q)$ 或 $\tilde{C}s(p, q)$ 的最大值。

由于 $\Pi(n) \geq n/4.7$，我们可以保证找到一对满足 $Cs(p, q) \geq n/4.7$ 的点；然而，对于给定的点集，可能存在一对点，使得 $Cs(p, q)$ 大至 $n/2$。该点集按如下方式构造：在平面上任意放置两点 p 和 q，然后将每组 $(n-2)/2$ 个点分别放在线 L(p, q) 的两侧，并使其非常靠近平分 p 与 q 的平分线。接着，为避免退化情形，对这些点稍作移动，同时保持点处于一般位置，见图2。任何经过 p、q 和第三个点的圆盘都包含 $(n-2)/2$ 个点。

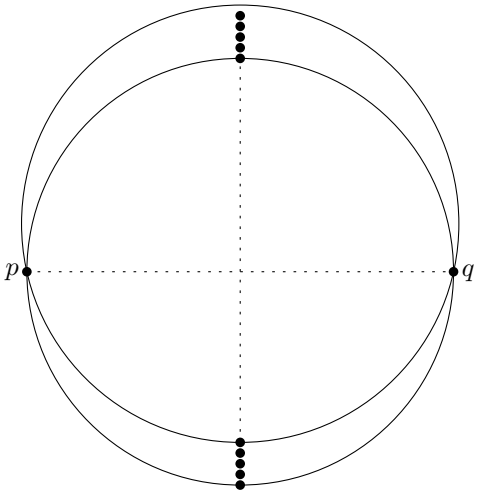

图2：由n个点组成的配置，其中任何经过p和q的圆盘，其内部至多包含$(n-2)/2$个其他点。

3.3 基于高阶 Voronoi 图的二次时间算法

本节介绍一种二次时间算法，给定点集 S 和整数 k，˜判断是否存在一对点满足 C$_S$(p, q) $\geq$ k；若存在，则找出这样一对点。由于我们知道总是存在一对点满足 ˜C$_S$(p, q) $\geq$ n/4.7，因此可以使用该算法（令 k 为 n/4.7），在二次时间内找到这样一对点。无论 α 如何取值，这个常数都优于定理 1 中的算法，但其运行时间为二次，而非 O(n log n)。不过，˜我们也可以使用这个二次算法，比上一节的算法更快地找出使 C$_S$(p, q) 最大的点对。新算法基于如下观察：权重为 k -1 的线段恰好是 k 阶 Voronoi 图的边，[18]，见图3，其中 k $\in\{1,\ldots,$ n $-1\}$。我们还需要 Claverol 等人 [11] 最先提出的两个观察。这些观察表明，如果一条平分线包含一条权重为 k $<$n/2 的线段，那么它也必定包含一条权重为 j 的线段，其中所有 k $\leq$ j $\leq$n/2 都成立。

观察 2。 如果 $b(p,q)$ 的一条无界线段的权重为 j，那么另一条无界线段的权重为 $n-j-2$。此外，由于相邻线段的权重恰好相差1，对于所有满足 $j \leq k \leq n-j-2$的 k，$b(p,q)$中都包含一条权重为 k 的线段。

为了找到一对满足 ˜$C_S(p,q) \geq k$ 的点，我们首先计算第$(k+1)$阶 Voronoi 图和第$(n-k-1)$阶 Voronoi 图。必定存在某一对点，使得第$(k+1)$阶 Voronoi 图的任何边都不是该平分线的一条线段。观察2 意味着该平分线的最小权重至少为$k+1$。

已知，阶数为（$k+1$）的 Voronoi 图的边数为 Θ（nk） [18]。由于k 约为n/4.7，因此该数量为 Θ（n^2）。要找出一条不包含阶数为（$k+1$）的 Voronoi 图中任何一条边的平分线，可以采用暴力方法，遍历阶数为（$k+1$）的 Voronoi 图的各条边，并标记每条边所属的平分线。由于边数如此之多，暴力算法需要二次时间；此外，还需要二次空间来记录哪些平分线已经出现。

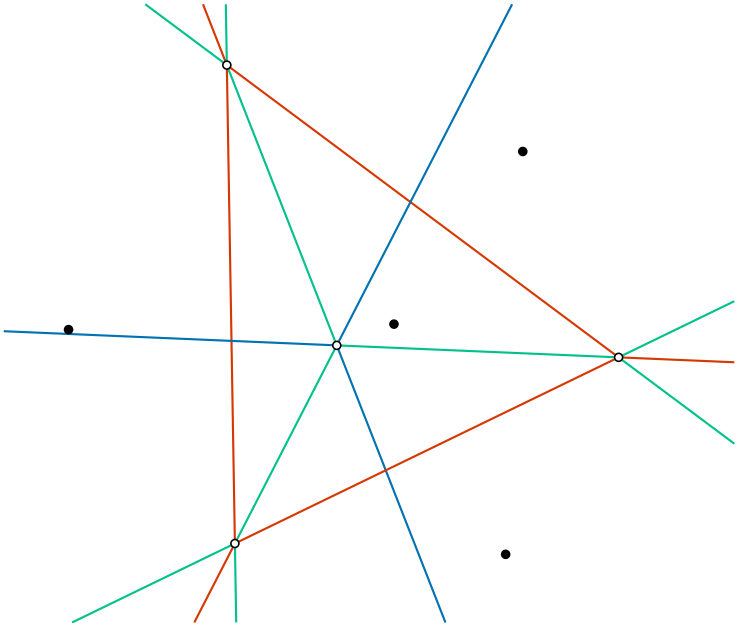

图3：四个点，其一阶 Voronoi 图用红色表示，二阶 Voronoi 图用绿色表示，三阶 Voronoi 图用蓝色表示。阶数为k 的 Voronoi 图的边是权重为$k-1$的线段。

在平面上构造阶数为k 的 Voronoi 图的最快算法由 Chan 等人提出， [10],其运行时间为$O(n\log n+nk)$期望时间。该算法也可用于构造阶数为$(n-k-1)$的 Voronoi 图，所需$O(n\log n+nk)$期望时间。

算法 2 判断是否存在满足 $\tilde{C}_S(p,q)\geq k$ 1: 计算 S 的阶数为 (k + 1) 和 (n − k − 1) 的 Voronoi 图 2: 对 $V_{k+1}(S)\cup V_{n-k-1}(S)$ 的每条边 e 执行 3: 令 {p, q} 为满足 e 是 b(p, q) 的一条线段的点对 4: 标记 { p, q} 5: 结束循环 6: 如果存在未标记的点对 {p,q}，则 7: 返回 {p, q} 8: 否则 9: 返回“无点对” 10: 结束判断

定理 2。 设 S 是平面上一个包含 n 个点的集合。在 $O(n\log n+nk)$期望时间内，我们可以确定是否存在一对 $\{p,q\}$，使得 $C_S(p,q)\geq k$ 或 $C_S(p,q)\geq k$；如果存在这样的一对，我们还可以找出它。

证明。 参见算法 2。首先计算阶数为$(k+1)$和阶数为$(n-k-1)$的 Voronoi 图。若存在一条平分线 $b(p,q)$ ，使得每条线段的权重都大于 k 但小于

n – k – 2，那么 b(p,q) 上不会有任何线段成为 $V_{k+1}(S)$ 或 $V_{n-k-1}(S)$ 的边。反之，根据观察 2，如果 b(p, q) 上有一条权重小于 k 的线段，那么它必定包含一条权重为 k 的线段。同样，如果 b(p, q) 上有一条权重大于 n – k – 2 的线段，那么它必定包含一条权重为 n – k – 2 的线段。因此，如果某条平分线上不包含任何作为 $V_{k+1}(S)$ 或 $V_{n-k-1}(S)$ 的边的线段，那么该平分线上所有线段的权重都严格介于 k 与 n – k – 2 之间。

算法遍历 $V_{k+1}(S)$ 和 $V_{n-k-1}(S)$ 的边，并标记每条边由哪条垂直平分线定义。如果处理完所有边后，每条垂直平分线都已标记，则不存在满足 $Cs(p,q) \geq k$ 的点对 $\{p,q\}$。否则，任意一条未标记的垂直平分线都定义了一个满足 $Cs(p,q) \geq k$ 的点对。

由于共有 $\binom{n}{2}$ 个点对，我们需要 $\Theta(n^2)$ 的空间来记录哪些垂直平分线已被标记。又由于已知 $V_{k+1}(S)$ 和 $V_{n-k-1}(S)$ 的边数为 $\Theta(nk)$，遍历这些边的运行时间由构造图所需的时间主导，即 $\Theta(nk + n\log n)$。

如果需要的是满足 $Cs(p,q) \geq k$ 的点对，则只需跳过 $V_{n-k-1}(S)$ 的计算，只考虑 $V_{k+1}(S)$ 的边即可。 □

结合算法 2 以及始终存在一对 $\{p,q\}$ 满足 $\tilde{Cs}(p,q) \geq n/4.7$，我们得到以下结果。

定理 3. 设 S 是平面上的 n 个点的集合。在 $O(n^2)$ 期望时间内，可以找到一对点 p 和 q，使得 $Cs(p, q) \geq n/4.7$ 或者 $\tilde{Cs}(p, q) \geq n/4.7$.

证明。 运行算法 2，令 $k = n/4.7$。 □

$\tilde{}$对于给定的集合 S， $\Pi(S)$ 的值介于 $n/4.7$ 和 $n/2$ 之间。因此，我们可以$\tilde{}$对 k 进行二分搜索，以确定 $\Pi(S)$ 的值，并找到一对点，使其达到 $\tilde{Cs}(p,q) = \tilde{\Pi}(S)$，调用该算法的次数至多为 $O(\log n)$。

定理 4. 设 S 是平面上的 n 个点的集合。在 $O(n^2 \log n)$ 期望时间内，可以找到一对点 p 和 q，使得 $\tilde{Cs}(p, q)$ 最大。

3.4 凸位置点的算法

本节给出一种线性时间算法，用于在点集 S 处于凸位置时，寻找满足 $Cs(p,q) \geq n/3$ 的一对点。我们假设点集 S 中的点已按其在S的凸包上的顺序给出。我们的算法基于 Hayward 等人 [17] 关于此类点对存在性的证明。以下引理是证明的基础。

引理 2. *[17, 引理 1]* 设 S 为一个点集，p 和 q 为 S 中的点。设 D 为经过 p 和 q 的圆盘。线段 pq 将 D 分为两个区域。如果这两个区域都至少包含 k 个点，那么 $Cs(p, q) \geq k$。

令 $p_1, p_2, \ldots, p_n$ 为沿凸包顺序排列的 S 中的点。注意，对于 i <j，点 p_i 和 p_j 将 S 分成两个集合：$\{p_{i+1}, \ldots, p_{j-1}\}$ 和 $\{p_1, \ldots, p_{i-1}\} \cup \{p_{j+1}, \ldots, p_n\}$。这两个集合的大小分别为 j – i – 1 和 n – (j – i) – 1。因此，给定两个点 p_i 和 p_j，我们可以在常数时间内计算 $H(p_i, p_j)$ 和 $H(p_j, p_i)$ 的大小。

Hayward 等人证明，对于任意一组S 处于凸位置的点，都存在一对点p和q，使得$C_S(p,q) \geq \lceil n/3 \rceil + 1$。这两个点必定位于某个包围圆盘的边界上 [17]。我们的算法只需遍历所有经过 S 中三个点的包围圆盘。下面的引理给出了这些包围圆盘与 S 的最远点 Voronoi 图之间的联系S。

引理 3. 设 S 是平面上的一个点集。设 D 是一个经过三个点的包围圆盘。D的中心是最远点 *Voronoi* 图的一个顶点。反过来，最远点 *Voronoi* 图的每个顶点都是一个经过三个点的包围圆盘的中心。

证明。 设 D 是一个经过三个点p、q和r的包围圆盘。将圆盘稍微向远离p 的方向移动，我们就得到一个包含S 中除p之外所有点的圆盘。因此，D 的中心与p的最远点 Voronoi 区域相邻。同理，D 的中心必定与q 和r的最远点 Voronoi 区域相邻，因而位于最远点 Voronoi 图的一个顶点上。

现在考虑最远点 Voronoi 图的一个顶点x 。由于该顶点位于三条平分线的交点处，设这三条平分线分别为$b(p,q)$、$b(q,r)$和$b(r,p)$，因此它到这三个点的距离相等。考虑以x 为中心且经过p、q和r的圆盘D 。如果S 中有某个点不在这个圆盘内，那么x 就会位于该点的最远点 Voronoi 区域内。但x 是图的一个顶点，所以D 必定是一个包围圆盘。 □

定理 5. 给定平面上处于凸位置的一个含有 n 个点的点集 S，可以在 $O(n)$ 时间内找到一对点 $p,q \in S$，使得 $C_S(p,q) \geq n/3$。

证明。首先，我们计算 S 的最远点 Voronoi 图。然后遍历该图的顶点。每个顶点对应一个经过三个点的包围圆盘，例如 p_i、p_j 和 p_k。这三个点定义了三条直线，我们可以在常数时间内计算 $C_S(p_i, p_j)$、$C_S(p_j, p_k)$ 和 $C_S(p_k, p_i)$ 的下界。引理 2 告诉我们，对于最远点 Voronoi 图的至少一个顶点，我们将找到一对点，使得 $C_S(p, q) \geq \lceil n/3 \rceil + 1$。

计算处于凸位置的点集的最远点 Voronoi 图需要线性时间 [1]。最远点 Voronoi 图有 $O(n)$ 个顶点，因此我们也可以在线性时间内遍历这些顶点，从而得到一个用于处理处于凸位置的点的线性时间算法。 □

3.5 直径圆盘算法

给定平面中的两个点p 和q ，直径圆盘 $B(p,q)$是以线段pq为直径的圆盘。Akiyama等人 [3] 证明，对于任意一个包含n 个点的集合S ，总是存在一对点$\{p,q\}$，使得$B(p,q)$至少包含 $n/3$个S中的点。他们的证明可以直接导出一种算法，因为他们证明了，该点对$\{p,q\}$中的两个点都必须位于S的最小包围圆盘的边界上。

定理 6： 设 S 是平面中包含 n 个点的集合。可以在 $O(n)$时间内找到一对$\{p,q\}$ 点，使得 $B(p, q)$ 包含 S 中至少 $n/3$ 个点。

证明。 首先计算 S 的最小包围圆盘D 。已知这需要线性时间 [20]。如果D的边界上只有两个点p、q，那么$B(p,q) = D$，我们就可以返回 $\{p,q\}$。

算法 3 凸位置中点的算法

```
1: 计算S的最远点 Voronoi 图
2: 对每个顶点v 执行
3:     令p_i、p_j、p_k 为定义v的三个点
4:     如果 min{|j − i|, n − |j − i|} − 1 ≥ n/3 + 1 则
5:         返回{p_i, p_j}
6:     否则如果 min{|k − j|,n − |k − j|} − 1 ≥ n/3 + 1 则
7:         返回 {p_j, p_k}
8:     否则如果 min{|i − k|,n − |i − k|} − 1 ≥ n/3 + 1 则
9:         返回 {p_k,p_i}
10:    结束如果
11: 结束循环
```

否则，D 的边界上有三个点 p、q 和 r。Akiyama等人 [3] 证明，D ⊂B(p, q) ∪B(q, r) ∪B(r, p)，因此这三个直径圆盘中至少有一个包含不少于 n/3 个点。我们可以在 O(n) 时间内计算集合 B(p, q) ∩ D、B(q, r) ∩D 和 B(r, p) ∩D 的大小。 □

3.6 双色点集算法

最后，我们考虑点集为双色点集的问题变体。设 S 是平面上由 n/2 个红点和 n/2 个蓝点组成的点集。我们希望找到一对 {p, q}，使得 p 为红点，q 为蓝点，并且 $C_S(p, q) \geq cn$ 或 $\tilde{C}_S(p, q) \geq cn$，其中 $c > 0$ 为某个常数。我们可以直接使用第 3.1 节中的算法，并调整定理 1 的证明，以表明我们可以均匀随机地选择一对双色点，并以常数概率满足 $C_S(p, q) \geq cn$。

定理 7. 设 $0 < \alpha < 1$ 为常数，S 是平面上由 $n/2$ 个红点和 $n/2$ 个蓝点组成的点集。至少有 $\alpha\binom{n}{2}^2$ 对点 $\{p, q\}$ 属于 S，其中 p 为红点，q 为蓝点，且 $\tilde{C}_S(p, q) \geq (1/2 - \sqrt{(2+\alpha)/12})n + O(1)$。该点对可以在 $O(n \log n)$ 期望时间内找到；或者以 $1 - 1/n$ 的概率，在 $O(n \log^2 n)$ 时间内找到。

证明。 由S中的点定义的 $\binom{n}{2}$ 条垂直平分线中，有$(n/2)^2$ 条是由颜色不同的两个点定义的。回想一下，S_k 表示至少有一条线段的权重小于k 或大于$n - k - 2$的垂直平分线数量。如果 $S_k < (1 - \alpha)(n/2)^2$,，那么至少有 $\alpha(n/2)^2$条红蓝垂直平分线，其线段的权重既不小于k，也不大于$n - k - 2$。 □

我们也可以使用二次时间算法，找出使$C_S(p, q)$或 $\tilde{C}_S(p, q)$达到最大值的红蓝点对$\{p,q\}$。唯一需要进行的修改是只考虑异色边。运行时间不会改变，因为垂直平分线的数量仍为 $\Theta(n^2)$。

定理 8. 设S是平面上由$n/2$ 个红点和$n/2$ 个蓝点组成的集合。给定一个整数k，我们可以在期望时间 $O(n \log n + nk)$ 内，判断是否存在红蓝点对$\{p,q\}$，使得$C_S(p, q) \geq k$或 $\tilde{C}_S(p, q) \geq k$；如果存在这样的点对，我们还能将其找出。

证明。 修改算法2 中的 for 循环，使其仅遍历由不同颜色的点定义的边。同时，仅当这两个点的颜色不同时，才返回未标记点对。该

仍然会计算完整的（$k+1$阶）和（$n-k-1$阶）Voronoi图，因此运行时间不会改变。

□

定理9。设 S 是平面内由 $n/2$ 个红点和 $n/2$ 个蓝点组成的集合。在 $O(n^2)$ 期望时间内，可以找到一对点$\{p, q\}$，使得 p 是红点，q 是蓝点，且 $C(p, q) \geq n/6.8$（或$\tilde{C}s(p, q) \geq n/6.8$）。

定理10。设 S 是平面内由 $n/2$ 个红点和 $n/2$ 个蓝点组成的集合。在 $O(n^2 \log n)$期望时间内，可以找到一对红蓝点$\{p, q\}$，使得 $C(p, q)$或 $\tilde{C}s(p, q)$达到最大值。

4 简单多边形中的算法

本节中，我们考虑在简单多边形的情形下，寻找一对点$\{p,q\}$，使得$Cs(p,q) \geq cn$。本节中，设S 为简单多边形P内的 n 个点组成的集合，其中多边形有m 个顶点，r 个为凹顶点，且$m-r$ 个为凸顶点。

我们的算法计算简单多边形内部的最短路径和按序k 测地 Voronoi 图。给定一个有 m 个顶点和 r 个凹顶点的简单多边形，Aichholzer 等人 [2] 证明，可以计算出一个简单多边形 P'，它包含 P，拥有与 P相同的凹顶点，且总共只有 $O(r)$ 个顶点。在 P 和 P'中，计算最短路径和 Voronoi 图这两个问题是等价的，也就是说，P 中 p 与q之间的最短路径，等于P'中p 与q 之间的最短路径。因此，我们假设 P 已使用该算法进行预处理，耗时 $O(m)$。我们还假设，Guibas 和 Hershberger 的最短路径数据结构 [15] 已针对P'构建完成。该数据结构可以在 $O(\log r)$ 时间内，告知我们P'中任意两点之间最短路径的长度；此外，它还能报告最短路径上的顶点，所需时间与顶点数量成正比。

Pollack、Sharir 和 Rote 的以下引理 [23] 将有助于我们的分析。

引理 4. *[23,引理 1]*设 $\triangle abc$ 是 P 中的一个测地三角形。当 x 沿着 $g(b,c)$变化时，$|g(a, x)|$是关于 $|g(b,x)|$的凸函数，并且 $|g(a, x)| <$最大值$\{|g(a,b)|, |g(a,c)|\}$。

4.1 在简单多边形中计算平分线上的最小权重线段

要调整用于在给定$Cs(p,q)$的情况下计算p和q的算法，面临着若干挑战。即使是在简单多边形中，第一步——计算经过三个点的圆盘的中心——也并非易事。事实上，甚至可能不存在边界上有给定三个点的测地圆盘 [5]。然而，Oh和Ahn [22] 证明，如果存在边界上有给定三个点的测地圆盘，那么该圆盘是唯一的。他们还展示了如何在$O(\log^2 r)$时间内找到该圆盘的中心。他们的算法假设 Guibas 和 Hershberger [15] 的最短路径数据结构已经构建完成。构建这一数据结构需要$O(m)$的预处理时间。

需要注意的一点是，由于我们处理的是一个包含P' 的多边形P，因此必须确保该圆盘的中心确实位于P内部。使用标准的平面点定位数据结构，对每个点需要$O(\log m)$时间，[12]，预处理时间为$O(m)$。这是唯一一个在P'中的结果与在P中的结果不同的操作。

找到经过p、q以及其余点x_i（其中 $1 \leq i \leq n-2$）的测地圆盘的圆心后，我们需要沿平分线对它们排序。在平面上，这仍然是一个简单的任务；但在简单多边形中，如何高效地完成这项工作并不明显。不过，所有这些

圆心都位于$b(p,q)$上。下一个引理表明，我们可以通过按这些点到p （或q）的距离排序，来沿$b(p,q)$对这些点排序。

引理5. 设 p 和 q 是简单多边形P中的两个点。设 a 和 b 是 $b(p,q)$上的两个点，其中 a 位于 b 与 $g(p,q)$ 的中点（即 $g(p,q)$ 与 $b(p,q)$ 的交点）之间。那么 $|g(p,a)| < |g(p,b)|$.

证明。 考虑以 p 为中心、半径为 $|g(p,b)|$ 的圆盘。我们的目标是证明 a 位于该圆盘内。不失一般性，我们可以假设a位于 $\triangle pmb$ 内，其中m 是$g(p,q)$的中点。否则，考虑q 而不是p。

根据引理4，我们知道 $|g(p,a)| < \max\{|g(p,m)|, |g(p,b)|\}$。我们还知道 $|g(p,m)| < |g(p,b)|$。反设如此，则 g(p, b) 与 g(b, q) 的连接将构成从 p 到 q 的最短路径，但最短路径是唯一的。因此 $|g(p,a)| < |g(p,b)|$。 □

唯一还需要的信息是，给定圆心 o 位于延伸路径 $l(p,q)$ 的哪一侧。通过检查路径 $g(p,q)$ 和 $g(p,o)$，并找出它们首次分叉的顶点，Guibas 和 Hershberger 的数据结构可以在 $O(\log r)$ 时间内判断这一点。因此，我们可以将圆心分成两组，一组位于 $g(p,q)$ 的一侧，另一组位于另一侧，然后分别按它们到 p 的距离排序。

引理 6。 设 S 是简单多边形 P 中 n 个点的集合，p 和 q 是 S 中的点。$Cs(p,q)$ 的值可以在 $O(m + n(\log^2 r + \log m + \log n))$ 时间内计算出来。

证明。计算经过 p、q 和 x_i 的测地圆盘的圆心 o_i（其中 $1 \le i \le n-2$），需要 $O(\log^2 r)$ 时间。我们还必须检查该圆心是否位于 P 中，这需要 O(log m) 时间。如果圆心位于 P 中，则在 O(log r) 时间内计算它位于 l(p, q) 的哪一侧以及 $|g(p,o_i)|$。对每个点 x_i（其中 $1 \le i \le n-2$），重复此过程。将圆心分成两组后，由于已经计算出距离 $|g(p,o_i)|$，我们可以在 O(n log n) 时间内将它们排序。最后，点排序完成后，我们可以按照第 3.1 节中的方法继续进行。在从一条线段移动到下一条线段时，判断是否应递增或递减 ω(s) 所需的信息已经计算出来：即 x_i 位于 g(p, q) 的左侧还是右侧。因此，剩余工作需要 O(n) 时间。再加上构建数据结构所需的 O(m) 预处理时间，总时间为 $O(m + n(\log^2 r + \log m + \log n))$。 □

现在我们证明，存在许多点对满足 $Cs(p,q) \ge cn$，其中 $c \approx 1/11.1$。遗憾的是，欧几里得情形中的证明无法推广到测地情形，因为 Ramos 和 Viaña 在证明中使用了这样一种变换的性质：它将 $\mathbb{R}^2$ 中的点 (x,y) 映射到 $\mathbb{R}^3$ 中的点 (x,y,x^2+y^2)。似乎没有简单的方法能将这种映射推广到测地情形。不过，我们可以转而参考 Bose 等人 [8] 的证明。他们也对权重较低的线段所对应的垂直平分线数量给出了上界，但采用了不同的方法。˜我们注意到，在简单多边形中， $\Pi(S)$ 和 $\Pi(S)$ 的下界并不相同。

考虑所有权重为 i 的线段的并集，其中 $0 \le i \le k$，这里 $k \le n-2$ 是某个固定整数。给定的平分线 $b(p,q)$ 上可能有许多权重不超过 k 的线段。两个权重不超过 k 的相邻线段属于同一个连通分量。令 λ_k 为所有平分线上的连通分量总数。Bose 等人 [8] 证明了 $\lambda_k \le 3kn - \frac{5}{2}k^2 - \frac{5}{2}k$。注意

如果 $\lambda_k < \binom{n}{2}$，那么必定至少有一条平分线没有连通分量。换句话说，存在一条平分线，其上没有权重为 k 或更小的线段。这意味着存在一对点满足 $C_S(p,q) \geq k$。

定理 11. 设 $0 < \alpha < 1$ 为一个常数，S 是简单多边形 P 中的 n 个点的集合，P 有 m 个顶点，其中 r 个为凹顶点。以概率 α，均匀随机选取的点对 $\{p,q\} \in S$ 将满足 $C_S(p,q) \geq (3-\sqrt{4+5\alpha})n/5$。

证明。 证明与定理 1的证明类似。如果 $\lambda_k < \alpha\binom{n}{2}$ 那么至少有 $(1-\alpha)\binom{n}{2}$条平分线不包含权重为 k或更小的线段。由于 $\lambda_k \leq 3kn - \frac{5}{2}k^2 - \frac{5}{2}k$，只要满足以下条件，这一结论就成立：

$$k < \frac{3n - \frac{5}{2} - \sqrt{(4+5\alpha)n^2 - (10+5\alpha)n + \frac{25}{4}}}{5} = \left(\frac{3-\sqrt{4+5\alpha}}{5}\right)n + O(1). \qquad \square$$

对于变体——要求任意一个边界上有 k 个点位于内部、且有 个点位于外部的圆盘，其边界经过p 和q ——我们可以将 λ_{n-k} 定义为：所有平分线上的权重大于$n-k-1$的所有线段的并集所含连通分量总数。如果$b(p,q)$不含连通分量，那么任何边界经过p和q的圆盘至多包含$n-k-1$个点。Bose 等人 [9]证明了 $\lambda_{n-k} \leq 4kn - \frac{7}{2}k^2 - \frac{7}{2}k$。

定理 12. 设 $0 < \alpha < 1$ 为一个常数，S 是简单多边形P 内的 n 个点的集合，该多边形有 m 个顶点，其中 r 个为凹顶点。以 α 的概率，均匀随机选取的点对 $\{p,q\} \in S$ 将满足 $\tilde{C}_S(p,q) \geq (7-\sqrt{37+12\alpha})n/12$。

证明。 为了确保存在一条平分线，其上没有权重至多为 k 或至少为$n-k-1$的线段，我们必须有 $\lambda_k + \lambda_{n-k} < \binom{n}{2}$。结合 λ_k和 λ_{n-k}的上界，可得 $\lambda_k + \lambda_{n-k} < 7kn - 6k^2 - 6k$。因此，如果满足 $\lambda_k + \lambda_{n-k} < \binom{n}{2}$，则

$$k < \frac{7n - 6 - \sqrt{(37+12\alpha)n^2 - (72+12\alpha)n + 36}}{12} = \left(\frac{7-\sqrt{37+12\alpha}}{12}\right)n + O(1). \tag{4}$$

$\square$

定理 *13.* 设 $0 < \alpha < 1$ 为一个常数，S 是简单多边形P 内的 n 个点的集合，该多边形有 m 个顶点，其中 r 个为凹顶点。可以在 $O(m + n(\log^2 r + \log m + \log n))$ 期望时间内找到一对点 $\{p,q\} \in S$，使得 $C_S(p,q) \geq (3-\sqrt{4+5\alpha})n/5$；或者找到一对点，使得 $\tilde{C}_S(p,q) \geq (7-\sqrt{37+12\alpha})n/12$。也可以用 $O(m + n\log n(\log^2 r + \log m + \log n))$ 时间，以 $1-1/n$ 的概率找到。

证明。我们知道，以 α 的概率，随机选取的一对点将具有所需性质（根据定理 11 或定理 12）。根据引理 6，可以在 $O(m + n(\log^2 r + \log m + \log n))$ 时间内计算出经过该点对的任意圆盘中点的最小数目和最大数目。在成功之前必须测试的点对数量的期望值为 α^{-1}，因此期望运行时间为 $O(m + n(\log^2 r + \log m + \log n))$。

如果我们选择 $\log_{(1-\alpha)^{-1}} 1 - 1 = 6$ (n) 对，那么没有 成功的概率为 $(1-\alpha)^{\log_{(1-\alpha)^{-1}}(n)}/n$。因此，成功的概率为$/n$。将引理的算法运行$O(\log n)$ 次，耗时为$O(m + n\log n(\log^2 r + \log m + \log n))$。 $\square$

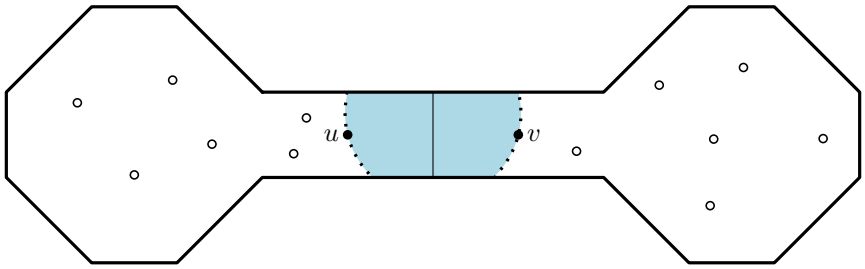

图4：一个简单多边形和两个点u 与v ，它们的平分线由一条权重为 0 的线段构成。没有任何以 u 和v 为边界点的测地圆盘包含该集合中的其他点。

4.2 基于高阶测地 Voronoi 图的算法

在测地情形下，Claverol等人的观察并不适用，因为垂直平分线受多边形边界的限制。如果 $b(p,q)$ 没有权重为 k 的线段，我们不一定能得出结论，认为 $b(p,q)$ 没有权重小于 k 的线段。也可能出现这种情况：$b(p,q)$ 由一条权重为 0 的线段构成，例如见图 4 。若要将蛮力算法应用于测地情形，我们需要构造所有 $1 \leq j \leq k$ 的 j 阶测地 Voronoi 图，并取它们的并集。

Oh和Anh [22] 给出了一种用于简单多边形中 k 阶测地 Voronoi 图的算法，其运行时间为 $O(k^2 n \log n \log^2 r + \min\{rk, r(n-k)\})$。Bose 等人 [8] 证明，在测地情形下$\Pi(n) \geq n/5$，因此，从 $k = 1$ 到 $k = n/5$ 计算 k 阶测地 Voronoi 图，可得到一种运行时间为 $O(n^4 \log n \log^2 r + rn^2)$ 的算法。另一方面，尝试所有 $\binom{n}{2}$ 对并逐一计算 $C_S(p, q)$ 的暴力算法，其运行时间为 $O(m + n^3(\log^2 r + \log m + \log n))$，快一个线性因子。暴力算法还有一个优点：它能找出使 $C_S(p, q)$ 最大化的实际点对，而 k 阶 Voronoi 图方法则不一定能做到这一点。

4.3 测地凸位置中点的算法

与欧几里得情形一样，当 S 中的点处于测地凸位置时，我们假设它们按照沿 S 的测地凸包排列的顺序给出。为了将凸位置中点的算法推广到测地情形，我们需要证明引理2在测地情形下成立。

引理 7. 设 S 是简单多边形 P 中的一个点集，且 p 和 q 是 S 中的点。设 D 是一个经过 p 和 q 的测地圆盘。路径 $g(p,q)$ 将 D 分成两个区域。如果这两个区域都至少包含 k 个点，那么 $C_S(p, q) \geq k$。

证明。设 c 为 D 的中心。考虑 b(p,q) 上另一个位于 l(p,c) 左侧的点 c′ 。设 D′为以 c′ 为中心、半径为 |g(c′,p)| 的圆盘。根据 [8, 引理 1], ，D 中位于 l(p, q) 左侧的任何点都包含在 D′中，因此 D′ 至少包含 S 中的 k 个点。同理可知，以 b(p,q) 上位于 l(p,c) 右侧的点 c″ 为中心、经过 p 和 q 的圆盘 D″ 也必须至少包含 k 个点。因此，任何经过 p 和 q 的圆盘都至少包含 k 个点。□

In the 平面中，使 $C_S(p,q) \geq n/3$ 成立的点对 $\{p,q\}$ 必须是三个点中的两个

位于某个经过三个点的包围圆盘的边界上。这个包围圆盘对应于最远点 Voronoi 图的一个顶点。在简单多边形中，最远点 Voronoi 图可能没有任何顶点。不过在这种情况下，总是存在一对点 $\{p,q\}$ ，使得经过 p 和 q 的直径圆盘，也就是以 $g(p,q)$的中点为中心的圆盘，包含整个集合 S [8]。注意，这样的圆盘以最远点测地 Voronoi 图的一条边为中心。因此，我们计算最远点测地 Voronoi 图；如果它至少有一个顶点，就可以按照第3.4节的方法，以常数时间检查每个顶点。如果它没有任何顶点，则可以以常数时间检查图中每条边所对应的每一对点。

对于一个简单多边形中包含的n 个点的集合，其最远点测地 Voronoi 图可在具有 m个顶点的情况下，于 $O(m+n\log n)$时间内 [27]计算得出。请注意，在这里，使用 Aichholzer 等人的多边形简化并不会改变运行时间，因为计算包含多边形 $O(m)$ 需要 P'时间。已知最远点测地 Voronoi 图有 $O(n)$ 个顶点和边。由于引理 3中给出的最远点测地 Voronoi 图顶点与边界上有三个点的包围圆盘之间的对应关系在简单多边形中也成立，因此我们可以计算最远点测地 Voronoi 图，并使用相同的算法寻找一个点对，使得 $C_S(p,q) \geq n/3$。

定理 14。 给定简单多边形 P 中测地凸位置的 n 个点组成的集合 S，其中 P 有 m 个顶点，可以在 $O(m+n\log n)$ 时间内找到一对点 p、$q \in S$，使得 $C_S(p,q) \geq n/3$。

4.4 直径圆盘算法

与欧几里得情形类似，在简单多边形中的给定点集 S 中，总存在一对点 $\{p,q\}$，使得 $B(p,q)$ 至少包含 S中的 $n/3$ 个点（本节中，$B(p,q)$ 表示以 $b(p,q)$ 为直径的测地圆盘）。Bose 等人 [8] 证明，对于简单多边形中的任意 S 点集（含 n 个点），总存在以下两种情况之一：存在一个包围测地圆盘，其边界上有三个点 p、q、r，且圆盘中心位于 $\triangle pqr$ 内；或者存在一对点 p、q ，使得 $B(p,q)$ 是一个包围测地圆盘。当包围测地圆盘的边界上有三个点时，他们证明，$B(p,q)$、$B(q,r)$ 或 $B(r,p)$ 中至少有一个包含 S 中至少 $n/3$ 个点。

定理 15. 设 S 是简单多边形 P 中由 n 个点构成的集合，P 有 m 个顶点。可以在 $O(m+n\log n)$ 时间内，找到一对点 $p,q \in S$，使得 $B(p,q)$ 包含 S 中至少 $n/3$ 个点。

证明。 首先构造 S 的最远点测地 Voronoi 图和最短路径数据结构。最远点图的每个顶点 v 都对应一个包围测地圆盘，其边界上有三个点 p、q 和r 。利用最短路径数据结构，可以在 $O(\log n)$ 时间内检验 v 是否位于 $\triangle pqr$内部。我们可以在 $O(n\log n)$ 时间内检验图中的所有顶点。如果找到一个顶点 v，使得 v位于 $\triangle pqr$内部，那么圆盘 $B(p,q)$、$B(q,r)$ 或$B(r,p)$ 中必有一个包含至少 $n/3$ 个点。通过显式构造最短路径，我们可以在 $O(\log n+m)$ 时间内求出这些圆盘的中点；随后，计算给定圆盘内点的数量需要 $O(n\log n)$ 时间。

如果没有顶点位于由三个对应点构成的三角形内，那么必定存在一个直径包围圆盘。我们可以通过检查图中的边来找到它。设e 为由两个点p 和q确定的图中的一条边。注意，$B(p,q)$是一个包围圆盘，当且仅当

且$b(p,q)$与e相交；我们可以通过检查e的两个端点是否位于$b(p,q)$的两侧，在$O(\log n)$时间内完成判定。由于图中有$O(n)$条边，因此总耗时为$O(n\log n)$。

□

4.5 双色点集的算法

现在我们考虑点集为双色的这一问题变体。令 S 为简单多边形中的一个点集，其中有$n/2$ 个红点和 $n/2$ 个蓝点。我们要找一对点 $\{p,q\}$，使得p 为红点、q 为蓝点，并且$CS(p,q) \geq cn$，其中c为某个常数 > 0。

定理 *16.* 设 $0 < \alpha < 1$ 为一个常数，且 S 是简单多边形 P 中含有 $n/2$ 个红点和 $n/2$ 个蓝点的点集。P 有 m 个顶点，其中 r 个为凹顶点。S 中至少有 $\alpha\binom{n}{2}^2$ 对点 $\{p, q\}$，满足 p 为红点、q 为蓝点，且 $CS(p, q) \geq (3/5 - \sqrt{(13 + 5\alpha)/50})n + O(1)$。找到这样一对点所需的期望时间为 $O(m + n(\log^2 r + \log m + \log n))$；或者，以 $1 - 1/n$ 的概率在 $O(m + n \log n(\log^2 r + \log m + \log n))$ 时间内找到。

证明。 回想一下， λ_k 表示权重为i的所有线段的并集中所有垂直平分线上的连通分量总数，其中 $0 \leq i \leq k$，且 $\lambda_k \leq 3kn - \frac{5}{2}k^2 - \frac{5}{2}k$。如果 $\lambda_k < (1 - \alpha)\binom{n}{2}^2$，那么至少有 $\alpha\binom{n}{2}^2$ 条红蓝垂直平分线不含任何连通分量，这意味着这些垂直平分线上的每条线段的权重都大于k。使这一结论成立的k最大值为

$$k < \frac{3n - \frac{5}{2} - \sqrt{(\frac{13}{2} + \frac{5}{2}\alpha)n^2 - 15n + \frac{25}{4}}}{5} = \left(\frac{3}{5} - \sqrt{\frac{13 + 5\alpha}{50}}\right)n + O(1).$$

□

与单色情形类似，在测地情形下，问题的内外版本也有一个单独的定理。

定理*17*。设 $0 < \alpha < 1$ 为一个常数，S 是简单多边形P 中由 $n/2$ 个红点和 $n/2$ 个蓝点组成的集合，该多边形有 m 个顶点，其中 r 个为凹顶点。S 中至少有 $\alpha\binom{n}{2}^2$ 对点 $\{p, q\}$，满足 p 为红点、q 为蓝点，且 $\tilde{C}S(p, q) \geq ((7 - \sqrt{43 + 6\alpha})/12)n + O(1)$。这样的点对可在 $O(m + n(\log^2 r + \log m + \log n))$ 期望时间内找到；或者以 $1 - 1/n$ 的概率，在 $O(m + n \log n(\log^2 r + \log m + \log n))$ 时间内找到。

证明。 我们需要 $\lambda_k + \lambda_{n-k} < (1 - \alpha)\binom{n}{2}^2$。我们可以使用 $\lambda_k + \lambda_{n-k} \leq 7kn - 6k^2 - 6k$ 的上界。结合这些结果并求解 k ，得到

$$k < \frac{7n - 6}{12} - \sqrt{\frac{(43 + 6\alpha)n^2 - 84n + 36}{12}} = \left(\frac{7 - \sqrt{43 + 6\alpha}}{12}\right)n + O(1).$$

□

5 结论

我们介绍了几种不同的算法，用于在平面上的一个含有n 个点的点集S 中，寻找一对p和q，使得$CS(p,q) \geq cn$，其中$c > 0$为某个常数。前两种算法适用于处于一般位置的点集。其他算法则针对问题的不同变体：点处于凸位置时、我们只关心寻找一对其直径圆盘包含许多点的点时，以及点集为双色时。

我们还研究了如何将这些算法推广到测地情形：此时S 是简单多边形P内部的一个含有n 个点的点集。

对于处于凸位置的点以及该问题的直径版本，我们的算法达到了已知的 $\Pi(n)$ 值，因此只有找到更快的算法才能改进它们。这样的算法不能使用最远点测地 Voronoi 图，因为构造此类图的最优算法与我们目前的运行时间相当。对于处于一般位置的点，情况则更有探索空间。是否存在一种次二次时间算法，能够保证找到一对满足$Cs(p,q) \geq \Pi(n)$的点？另一个有趣的问题是，如何改进针对给定点集S、找出使$Cs(p,q)$最大的点对的算法。

参考文献

[1] A. Aggarwal、L. J. Guibas、J. B. Saxe 和 P. W. Shor。计算凸多边形 Voronoi 图的线性时间算法。《离散与计算几何》，4:591–604，1989年。[2] O. Aichholzer、T. Hackl、M. Korman、A. Pilz 和 B. Vogtenhuber。保测地线多边形简化。《国际计算几何与应用杂志》，24(4):307–324，2014年。[3] J. Akiyama、Y. Ishigami、M. Urabe 和 J. Urrutia。关于包含最多点的圆。《离散数学》，151(1–3):15–18，1996年。[4] B. Aronov。关于简单多边形中点站点的测地 Voronoi 图。《*Algorithmica*》，4(1):109–140，1989年。[5] B. Aronov、S. Fortune 和 G. T. Wilfong。最远站点测地线 Voronoi 图。《离散与计算几何》，9:217–255，1993年。[6] I. Bárány 和 D. G. Larman。点与椭球的一种组合性质。《离散与计算几何》，5:375–382，1990年。[7] I. Bárány、J. H. Schmerl、S. J. Sidney 和 J. Urrutia。关于欧几里得空间中点与球的一项组合结果。《离散与计算几何》，4(3):259–262，1989年。[8] P. Bose、G. Esteban、D. Orden、R. I. Silveira 和 T. Tuttle。关于包围多个点的测地圆盘。载于 P. Morin 和 E. Oh 编，第*19*届国际算法与数据结构研讨会（*WADS 2025*），莱布尼茨国际信息学会议录（LIPIcs）第349卷，第10:1–10:20页，德国达格施图尔，2025年。Schloss Dagstuhl – Leibniz-Zentrum für Informatik。[9] P. Bose、G. Esteban、D. Orden、R. I. Silveira 和 T. Tuttle。关于包围多个点的测地圆盘。*CoRR*，abs/2506.06477，2025年。[10] T. M. Chan、P. Cheng 和 D. W. Zheng。平面高阶 Voronoi 图的最优算法：非确定性的用处。载于 D. P. Woodruff 编，《*2024*年 *ACM-SIAM* 离散算法研讨会（*SODA 2024*，美国弗吉尼亚州亚历山德里亚，*1*月*7–10*日，*2024*》，第4451–4463页。SIAM，2024年。

[11] M. Claverol、C. Huemer 和 A. Martínez-Moraian。关于包围多个点的圆。《离散数学》，344(10):112541，2021年。[12] M. de Berg、O. Cheong、M. J. van Kreveld 和 M. H. Overmars。《计算几何：算法与应用，第*3*版》。施普林格，2008年。[13] H. Edelsbrunner。组合几何算法，EATCS理论计算机科学专著第10卷。施普林格，1987年。[14] H. Edelsbrunner、N. Hasan、R. Seidel 和 X. J. Shen。始终包围多个点的过两点圆。Geometriae Dedicata，32(1):1–12，1989年。[15] L. J. Guibas 和 J. Hershberger。简单多边形中的最优最短路径查询。计算机系统科学杂志，39(2):126–152，1989年。[16] R. Hayward。关于圆包含问题的一则注记。《离散与计算几何》，4(3):263–264，1989年。[17] R. Hayward、D. Rappaport 和 R. Wegner。关于包含点的圆的一些极值结果。《离散与计算几何》，4(3):253–258，1989年。[18] D.-T. Lee。关于平面中的*k*近邻 Voronoi 图。*IEEE*计算机汇刊，31(6):478–487，1982年。[19] D.-T. Lee 和 F. P. Preparata。存在直线障碍物时的欧几里得最短路径。*Networks*，14(3):393–410，1984年。[20] N. Megiddo。线性规划及相关问题的线性时间算法（r^3 维）。*SIAM*计算杂志，12(4):759–776，1983年。[21] V. Neumann-Lara 和 J. Urrutia。关于平面上点与圆的一项组合结果。《离散数学》，69(2):173–178，1988年。[22] E. Oh 和 H. Ahn。简单多边形中适中规模点集的 Voronoi 图。《离散与计算几何》，63(2):418–454，2020年。[23] R. Pollack、M. Sharir 和 G. Rote。计算简单多边形的测地中心。《离散与计算几何》，4:611–626，1989年。[24] M. N. Prodromou。点与球的一种组合性质：着色版本。《离散与计算几何》，38(4):641–650，2007年。[25] P. Ramos 和 R. Viaña。包围多个点的点、线段与圆的深度：一则注记。《计算几何：理论与应用》，42(4):338–341，2009年。[26] P. Valencia Saravia。圆形覆盖问题。博士论文，墨西哥国立自治大学，2005年。[27] H. Wang。简单多边形中测地最远点 Voronoi 图的最优确定性算法。《离散与计算几何》，70(2):426–454，2023年。