arXiv:2611.31034v1 [cs.CG] 23 Nov 2026

通过双点迹进行在线圆盘击中

周明超 任春英 YingliRan Zhao Zhang*
zmc138575@gmail.com rcy0405@zjnu.edu.cn ranyingli@zjnu.edu.cn hxhzz@sina.com浙江师范大学

摘要

在圆盘的在线击中集问题中，预先给定平面中的一个点集 P，圆盘逐个到达，并且每个圆盘都与 n 中的点相交。每次有新圆盘到达后，算法必须为截至目前所见的所有圆盘维护一个击中集 P。算法只能向 $X \subseteq P$ 中添加点，目标是最小化 X的最终大小 X。Even 和 Smorodinsky 为单位圆盘给出了一种 $O(\log n)$竞争算法，并提出了对于任意半径的圆盘， $O(\log^2 n)$这一界是否可以改进的问题。对于任意半径，我们的随机化算法在每个固定请求序列上的期望选点数为 $O(k(1+\log(n/k)))$，其中 k 是该请求序列的最小击中集大小。这给出了 $O(\log n)$的期望竞争比。对于每个 k，我们证明了最坏情况下期望代价的匹配下界，即使在共线点集上也是如此。算法预先在 P 上采样优先级顺序。对于每个到达的圆盘 D，算法选择当前顺序中 $D \cap P$的第一个点。对于排除固定子式的图中的最短路径球，也成立相同的期望成本界，并且在路径上具有匹配下界。

1 引言

在几何击中集中，目标是从给定点集中选出尽可能少的点，使得一个族中的每个区域都包含至少一个被选中的点。我们研究圆盘的在线阶段。算法预先获得平面中的一组 P 个 n 点。它逐个接收圆盘，每个圆盘都与 P相交。每次有圆盘到达后，算法必须为截至目前所见的圆盘维护一个击中集 $X \subseteq P$，并且只能向 X中添加点。我们将选中点的总数与离线最优值 k进行比较，即整个请求序列的最小击中集大小。

根据对偶性， [2] 中的在线集合覆盖算法给出了 $O(\log^2 n)$ 的竞争比，因为圆盘只能诱导出 P 的多项式数量的不同子集。这是该界限所需的唯一几何事实。Even 和 Smorodinsky [11] 利用单位圆盘的几何性质，获得了 $O(\log n)$ 的竞争比。他们提出了这样一个问题：当圆盘可以具有不同半径时，是否能够改进 $O(\log^2 n)$ 的界限。针对任意半径，在非自适应对手下，我们获得了 $O(\log n)$ 的期望竞争比；该对手会在算法进行随机选择。

当离线最优值较大时，对 k 的依赖关系很重要。例如，当 $k = \Theta(n/\log n)$ 时，下界 $O(k \log n)$ 已经是 $O(n)$，即选择所有点的代价。我们的算法期望选择 $O(k(1+\log(n/k)))$ 个点，在这种情况下得到 $O(k \log \log n)$。对于每个 k，我们都证明了一个匹配的最坏情况下界，即使候选点位于一条直线上也是如此。

*通讯作者。

1.1 相关工作

De 等人 [8] 针对 $[1, M]$ 中的圆盘半径给出了一个确定性 $O((1+\log M)\log n)$ 竞争比。他们还给出了一个 $O(\log n)$-竞争算法，适用于任意半径的情形，条件是给定直线将点与圆盘中心分隔开。Alefkhani 等人 [1] 在候选集由 $(0, N)^2$ 中的所有整数点组成且圆盘位于该方框内时，得到了 $O(\log N)$。对于轴平行正方形，Khan 等人 [14] 在整数坐标假设下给出了 $O(\log N)$ 的竞争比。De 等人 [9] 针对任意有限平面点集和任意大小的正方形，给出了一个 $O(\log n)$ 的竞争比。对于一般的无权范围空间，若 m 范围的大小至多为 d，则 Buchbinder 和 Naor [7] 的集合覆盖界通过对偶性给出了确定性代价 $O(k(1+\log d)(1+\log(m/k)))$。Doron-Arad 和 Naor [10] 表明，即使对于预先知道范围空间的随机化算法，某些范围空间在面对非自适应对手时仍要求 $\Omega(\sqrt{n})$ 的竞争比。

最近，Bhore 等人 [4] 提出了一种优雅的方法，利用 ϵ-网在线阶段对分数命中集进行舍入。后来，作者发现所推导集合系统的浅层单元复杂度界存在错误，因此撤回了该论文。

我们的算法在候选点上采样优先级顺序。我们利用指数时钟，通过候选点子集 [13]上的德洛内图的稀疏性来界定这些顺序的代价。如果某个范围包含两个点且不包含该子集中的其他点，则在这两个点之间连接一条边。我们将这样的点对称为双点迹。对于圆盘，平面支撑意味着该图是平面的 [19]。对于无权图中的球， [5], 中针对单点球的结果表明，该图是宿主图的一个子式。Browne 和 Chang 关于度量支配的工作中也出现了沃罗诺伊单元的类似收缩 [6]。

1.2 我们的贡献

定理 1。 设 P 是平面中的 $n \geq 2$ 个点的集合。存在一种随机在线算法，可以命中与 P相交的任意半径闭圆盘。对于每个固定请求序列，若离线最优值为 $k \geq 1$，则其期望代价为 $O(k(1+\log(n/k)))$。

对于每个 k，都存在匹配下界，即使点集共线也是如此。

定理 2. 对于满足 $1 \leq k \leq n$ 的所有整数 n, k，存在一个由 n 个平面点组成的集合 P，使得对于 P 上的每个随机在线算法，都存在一个离线最优值恰为 k 的固定圆盘请求序列，迫使该算法在期望意义下至少选择 $k(1+\frac{1}{2}\lfloor \text{对数}_2(n/k)\rfloor)$ 个点。

对于圆盘，预处理需要 $O(n^2 \log n)$ 次几何操作。每个请求的最坏情况耗时为 $O(\log^2 n)$（第 5.2 节）。

圆盘的上界由定理 8 得出。如果对于每个 $S \subseteq P$，德洛内图 $G(S)$ 至多有 $\kappa|S|$ 条边，那么该定理保证，对于每个离线最优值为 $k \geq 1$ 的固定请求序列，期望代价为 $O((1+\kappa)k(1+\log(n/k)))$。对于圆盘， $G(S)$ 是平面图，因此 $\kappa = 3$ 即可。

对于最短路径球，将非空候选子集 S 的沃罗诺伊单元收缩后，可得到宿主图的一个子式，其中包含 $G(S)$。因此，排除某个固定子式即可得到所需的线性边数上界。

定理 3（最短路径球）。 固定一个图 F。设 H 是一个有限连通无向图，不包含 F 作为子式，且边长均为正有理数。设 $P \subseteq V(H)$ 为一组 n 候选顶点。存在一种随机在线算法，可命中与 P 相交的最短路径球。对于每个离线最优值为 $k \geq 1$ 的固定请求序列，该算法期望选取 $O(k(1+\log(n/k)))$个顶点，其中常数仅取决于 F。

对于有理半径，预处理的期望时间为多项式时间，且每个请求所需时间关于输入位长度为多项式（第6节）。对于每个 $1 \le k \le n$，在直线上已有匹配下界，因此该成本界对于平面图是紧的。

1.3 技术概述

在 P 上确定一个优先级顺序，并按照该顺序用每个圆盘中的第一个点命中它。如果该顺序以一个最优击中集开头，则该规则至多选择 k 个点。在在线规则固定后，设计问题就是：如何在请求到达之前采样一个优先级顺序，使得该规则在期望意义下选择较少的点。

为了界定采样顺序的代价，固定一个离线击中集。对于集合中的每个点 p，统计在包含 p 的圆盘中可能最先出现的不同点的数量。将这些数量相加即可界定所选点的数量。如果这样的圆盘的第一个点是 $q \neq p$，则它会遗漏 q 之前的每个点。因此，pq 是在加入 p 和 q 后、q 之前的前缀上构成的德洛内图的一条边。q 的不同选择会产生不同的前缀，因此这些边属于不同的图。

为了比较这些图，为每个点分配一个独立的指数时钟，并令 S_t 表示其时钟已在时间 t 前响起的点。当 q 在 t 时响起且早于 p 时，边测试针对的是 $S_t \cup \{p, q\}$。纳入两个端点后，该测试只取决于其他时钟。当两个端点都在 t 前响起时，同一测试会询问 pq 是否是 $G(S_t)$的一条边。这样，我们便可以将每个时刻的加权贡献，界定为单个稀疏图中的期望边数。对于最优大小的一个猜测 s，乘法权重会将加权界转化为期望代价保证，并适用于任何可由至多 s 个点命中的固定序列。我们在请求到达之前，针对依次加倍的猜测采样顺序。当添加一个新的第一个点将超出当前阶段的预算时，在线算法便切换到下一个顺序。

2 预备知识

设 $(P,\mathcal{R})$ 是一个范围空间，其中 P 是 n 候选点集，$\mathcal{R} \subseteq 2^P \setminus \{\varnothing\}$ 是可能范围的族。与 P 相交的圆盘或球 D 会诱导出范围 $D \cap P$。

定义 4（在线击中集）。点集 P 已知。请求 $R_1, \dots, R_\ell \in \mathcal{R}$ 依次到达。算法从 $X_0 = \varnothing$ 开始，在收到 R_i 后维护一个击中集 X_i，使得

$$X_{i-1} \subseteq X_i \subseteq P \qquad \text{and} \qquad X_i \cap R_j \neq \varnothing \quad (1 \le j \le i).$$

最终击中集为 $\mathrm{ALG} = X_\ell$，其代价为 $|\mathrm{ALG}|$。设 OPT 是整个请求序列的最小击中集，并将 $k = |\mathrm{OPT}|$ 记为离线最优值。

定义 5（竞争比）。若对于算法运行前固定的每个非空请求序列，$\mathbb{E}|\mathrm{ALG}| \le ck$，则称随机在线算法相对于非自适应对手是 c-竞争的。期望取自算法的随机选择。

空序列不产生任何代价。对于 $n = 1$，最优做法是在第一个请求中选择唯一的点。我们在第3节中假设 $n \ge 2$。

定义 6（两点迹与德洛内图）。对于 $S \subseteq P$，范围 R 在 S 上的迹为 $R \cap S$。非空迹构成 $\mathcal{R}$ 对 S 的限制。S 中一对不同的点 $\{p,\ q\}$，如果对于某个 $R \in \mathcal{R}$ 有 $R \cap S = \{p,\ q\}$，则称其为 S 上的两点迹。德洛内图 $G(S)$ 的顶点集为 S，并且对于 S [13]上的每个两点迹都有一条边。

该算法给定 κ，使得每个 $S \subseteq P$ 至多有 $\kappa|S|$个两点迹。等价地，

$$|E(G(S))| \le \kappa|S| \qquad (S \subseteq P). \tag{1}$$

3 算法

固定 π 上的一个优先级顺序。当范围 R 到达时，将 R 在 π中的第一个点 first(R,π) 加入击中集。如果最优击中集 O 中的点在 π中排在最前面，该规则只会选取 O中的点，代价至多为 $|O|$。我们的目标是在请求到达之前，抽样一个期望代价较小的顺序。

对于规则选取的每个点 q，令 R 为首次选中该点时对应的请求。在分析中，我们将 q 的代价记到任意一个点 $p \in O \cap R$上。若 q 在 π中先于 p，则写作 $q \prec_\pi p$；并令 $B_\pi(q) = \{r \in P : r \prec_\pi q\}$。由于 q 在 R中排在最前面，因此该范围不包含 $B_\pi(q)$。如果 $p \neq q$，那么 $q \prec_\pi p$ 和$\{p,q\}$ 就是 $B_\pi(q) \cup \{p,q\}$上的一个双点迹。

我们将从 A 的 π 初始段中进行的选择单独计数。这样的点至多有 $|A|$ 个。如果 R 的第一个点位于 A之外，那么 R 不会命中 A，因此其第一个点只能计入 $O \setminus A$中的某个点。

定义 7（负载）。对于 $p \in A$，令$L_{A,\,\pi}(p) = 0$。对于 $p \notin A$，负载 $L_{A,\,\pi}(p)$计算 p 自身以及每个更早的点 $q \notin A$ 的数量，其中这些点可以作为包含 p的范围中的第一个点。因此

$$L_{A,\pi}(p) = 1 + \sum_{\substack{q \notin A \\ q \prec_\pi p}} \mathbf{1}\left[pq \in E\big(G(B_\pi(q) \cup \{p,q\})\big)\right]. \tag{2}$$

对于请求序列的任意击中集 O，该规则至多选择$|A| + \sum_{p \in O} L_{A,\,\pi}(p)$个点（引理 9）。前缀 $B_\pi(q)$会随 q变化，而（1）限制的是单个集合上的边数。为了关联这些前缀，我们使用指数时钟描述一种随机顺序。

固定一个整数 $1 \le s \le n$ 作为 k的猜测值。我们首先描述如何针对 P上给定的正整数权重 $u(p)$生成一个顺序。这些权重由 SampleOrder（算法 2）提供：第一轮中它们全部为 1，并在轮次之间更新。为进行分析，为每个 p分配一个速率为 $\mu_p = u(p)/\sum_{a \in P} u(a)$的独立指数时钟。令 S_t 为其时钟在时间 t之前响铃的点集。以任意顺序先列出 $A = S_s$，再按照响铃时间的顺序追加其余点。由于 $\mathbb{P}[p \in S_s] = 1 - e^{-s\mu_p} \le s\mu_p$，我们有 $\mathbb{E}|S_s| \le s$。

假设 q 在时间 $t > s$ 响铃，而 p 稍后响铃。那么 $B_\pi(q) = S_t \setminus \{q\}$，因此，当且仅当 pq 是 $G(S_t \cup \{p,q\})$ 的一条边时，q 才会计入 p 的负载。固定 t 时，这项检验仅取决于其他时钟。当两个端点都在 t 之前响铃时，同样的检验是在判断 pq 是否为 $G(S_t)$ 的一条边。在每个 t 处比较这些事件，就能利用 $G(S_t)$ 的边数期望来界定 $\mathbb{E}[\sum_{p \in P} \mu_p L_{A,\pi}(p)]$（引理 10）。

我们用这个时钟界来分析算法 1 WeightedOrder，该算法直接对优先级顺序进行采样。固定 P 上的任意顺序。以 $\min\{1,\ s\mu_p\}$ 的概率，独立地将每个点 p 纳入 A；按固定顺序先列出 A，然后每次追加一个剩余点。从剩余集合 M 中，以 $u(p)/\sum_{a \in M} u(a)$ 的概率选取 $p \in M$。

由于 $\min\{1,\ s\mu_p\} \ge 1 - e^{-s\mu_p}$，我们可以将 A 与时钟前缀耦合，使得 $S_s \subseteq A$。在给定 A 的条件下，顺序抽样服从时钟顺序限制在 $P \setminus A$ 上的分布。将额外的点移入前缀不会增加负载。算法 1 给出了具体流程；引理 11 证明了其加权负载界。附录A介绍了随机抽样的精确实现。

算法 1 WeightedOrder

要求： 一个点集 P、一个整数 $1 \le s \le |P|$，以及正整数权重 $u(p)$，其中 $p \in P$

函数 WeightedOrder(P, s, u)
  $W \leftarrow \sum_{p \in P} u(p)$，$A \leftarrow \varnothing$
  对于每个 $p \in P$，**执行**
    以概率 $\min\{1, su(p)/W\}$独立地将 p 添加到 A
  结束 for
  按固定顺序以 π 开始，其中包含 A，并设置 $M \leftarrow P \setminus A$
  当 $M \neq \varnothing$ **时执行**
    以概率 $u(q)/\sum_{a \in M} u(a)$抽取 $q \in M$
    将 q 添加到 π，并从 M中移除它
  结束 while
  返回 (A, π)
结束函数

权重较小的点 $p \in O$ 可能具有较大的负载，但对加权平均的贡献仍然很小。收费界会计算其完整负载。对于给定的猜测 s，SampleOrder 使用乘法权重 [3]。在每一轮中，它调用 WeightedOrder，并根据每个点的估计负载增加其权重。

直接计算 (2) 中的所有负载，需要测试负载和中的每个点对。我们改为从一次抽样 Q中估计所有负载，该抽样从 P 中均匀选取，并且独立于当前点对 (A, π) 以及之前的所有轮次。如果 $Q \notin A$，则对 π 中每个 p 在 $\prec Q$ 之后，测试 $pQ \in E(G(B_\pi(Q) \cup \{p, Q\}))$ 是否成立。这至多需要 $n-1$ 次边测试。对于 $p \in A$，设置 $L(p) = 0$。对于 $p \notin A$，设置

$$\widehat{L}(p) = 1 + n\mathbf{1}\left[Q \in P \setminus A, \quad Q \prec_\pi p, \quad pQ \in E\big(G(B_\pi(Q) \cup \{p, Q\})\big)\right]. \tag{3}$$

如果 $Q \in A$，则对每个 $p \notin A$，估计值都是 1。和 (2) 中的每个 q 以概率 $\prec 1/n$ 抽取，因此，给定 (A, π)，$L(p)$ 是 $L_{A,\,\pi}(p)$ 的无偏估计。对于圆盘，Q 的边测试使用相同的前缀，因此可以一起执行（第 5.2 节）。

在预处理轮次 j 中，令 (A_j, π_j) 表示按权重 u_j 采样得到的点对。这些估计值满足 $\prec 0 \le L_j(p) \le n+1$。以 $D = n+1$ 作为它们的上界，更新

$$u_{j+1}(p) = u_j(p)\big(D + \widehat{L}_j(p)\big), \qquad D = n+1.$$

权重仍为整数。对于每个至多包含 s 个点的固定集合，分析给出了其在 $T = 2n$ 轮次上的平均期望总负载。因此，我们从$\{1, \ldots, T\}$中均匀选取 J，且该选择独立于轮次，并返回 π_J 供在线优先级规则使用。集合 A_j 用于计算负载估计。

算法 2 为猜测 s 采样优先级顺序

函数 SampleOrder(P, s)
  $D \leftarrow n+1$, $T \leftarrow 2n$
  $u_1(p) \leftarrow 1$ 对每个 $p \in P$
  从$\{1, \ldots, T\}$中均匀抽取 J，且独立于以下轮次
  对于 $j = 1, \ldots, T$**执行**
    $(A_j, \pi_j) \leftarrow$ WeightedOrder(P, s, u_j)
    从 P 中均匀抽取 Q
    Set $\widehat{L}_j(p)$ 在 A_j 上执行 $= 0$，并在 $P \setminus A_j$ 上使用 (3)
    $u_{j+1}(p) \leftarrow u_j(p)(D + \widehat{L}_j(p))$ 对每个 $p \in P$，
  结束循环
  返回 π_J
结束函数

对于任何至多可由 s 个点命中的固定序列，返回的顺序所产生的期望代价至多为$(6 + 4\kappa)s(1 + \ln(n/s))$（引理 12）。

由于 k 未知，在线算法在每个阶段使用一个猜测值，从 $s = 1$开始并不断加倍，直到 n。在预处理过程中，我们独立地为各个猜测值 $s < n$ 采样顺序。对于每个这样的 s，预算 B_s 限制了优先级规则在该阶段可选取的不同点的数量。我们令 B_s 至少为前一成本界期望值的四倍。当达到 $s \geq k$ 的阶段时，这一选择将再次进入下一阶段的概率限制为 $1/4$。对于 $s = n$，任意顺序均可，我们设定 $B_n = n$。

算法 3 在各阶段之间保留击中集 X 。在每个阶段内， Y 记录其请求的不同首个点，包括已在 X中的点。每个请求都按照当前顺序进行处理，即使 X 已经命中该请求，这样 $|Y|$ 就能衡量从空击中集开始对这些请求应用优先级规则的代价。如果 $|Y| = B_s$ 且当前请求的首个点位于 Y之外，则该请求启动下一阶段。保留 X，清空 Y，并在新顺序中查找同一请求的首个点。完成此检查后，将当前顺序中该请求的首个点同时加入 X 和 Y。

算法 3 在线击中集 要求：一个包含 $n \geq 2$ 点的集合 P、一个界 κ，以及针对 G（S）的边测试 确保：在每个请求之后维护击中集 X 预处理 1：对于每个 $s = 1$、2、4、.....，且满足 $s < n$，执行 2：$\pi_s \leftarrow$ SampleOrder(P, s)，针对该 s独立执行 3：$B_s \leftarrow \lceil 4(6 + 4\kappa) \rceil s(1 + \lceil \log_2(n/s) \rceil)$ 4：结束循环 5：将 π_n 设为 P 上的任意顺序，并将 $B_n \leftarrow n$ 设为在线阶段 6：$s \leftarrow 1$、$X \leftarrow \varnothing$、$Y \leftarrow \varnothing$ 7：对于每个到达的范围 R，执行 8：$q \leftarrow$ first(R, π_s) 9：如果 $q \notin Y$ 且 $|Y| = B_s$，则 10：$s \leftarrow \min\{2s, n\}$，$Y \leftarrow \varnothing$ 11：$q \leftarrow$ first(R, π_s) 12：结束条件 13：$Y \leftarrow Y \cup \{q\}$，$X \leftarrow X \cup \{q\}$ 14：结束循环

在预处理期间，对范围族的唯一查询是询问两个点是否在 $G(S)$ 中相邻。此时，稀疏性界（1）给出如下保证。

定理 8（稀疏Delaunay图）。假设对每个 $S \subseteq P$ 都成立（1）。对于任意固定请求序列，其离线最优值为 $k \geq 1$，算法 *3* 的期望代价为 $O((1 + \kappa)k(1 + \log(n/k))$。它将每个请求接收为 P 的子集。预处理使用 $O(n^2 \log n)$ 次预言机查询，以判断给定点对是否为 $G(S)$ 的边。其余预处理过程的期望运行时间为多项式时间。

附录A给出了一种变体，其具有相同的期望代价界；除边测试外，预处理的最坏情况时间为多项式时间。

第4节证明这些界。第5节和第6节建立稀疏性，并为圆盘和最短路径球实现所需的查询。

4 分析

我们首先分析将优先级规则应用于固定顺序的代价。

引理 9（收费）。固定一个顺序 π 和 π的一个初始段 A。如果 O 命中序列中的每个请求，则优先级规则所选集合的大小至多为

$$|A| + \sum_{p \in O} L_{A,\pi}(p).$$

证明。将每个选中的点 $q \notin A$ 计入任意一个 $p \in O \cap R$，其中 R 是 q 首次被选中的请求。由于 q 在 R 中排在最前面，该范围不会包含 $B_\pi(q)$，因此也不会包含 A。如果 $p = q$，则式 (2) 中的常数项计入该费用。否则，$q \prec_\pi p$ 和 $R \cap (B_\pi(q) \cup \{p, q\}) = \{p, q\}$，因此 q 对 $L_{A,\pi}(p)$ 贡献一项。计入 p 的不同点贡献不同的项。最多还有 $|A|$ 个选择位于 A 中。 □

对于指数时钟构造，我们界定前缀大小和加权负载 $\mu\cdot L_{A,\ \pi}=\sum_{p\in P}\mu_p L_{A,\ \pi}\ (p)$ 。

引理 10（指数时钟下的加权负载）。 假设（1）成立。固定一个整数 $1\le s\le n$ 和正速率 μ_p，满足 $\sum_{p\in P}\mu_p=1$。令 τ_p 为速率为 μ_p 的相互独立的指数随机变量，并设 $S_t=\{p\in P:\ \tau_p\le t\}$。构造 π：先以任意顺序放置 $A=S_s$，然后按 τ_p 递增的顺序放置其余各点。则 $\mathbb{E}|A|\le s$，且

$$\mathbb{E}[\mu\cdot L_{A,\pi}]\le 1+2\kappa(1+\ln(n/s)). \tag{4}$$

证明。 由于 $\mathbb{P}[p\in S_t]=1-e^{-\mu_p t}\le\mu_p t$，对 p 求和得到 $\mathbb{E}|S_t|\le\min\{t,\ n\}$。特别地，$\mathbb{E}|A|\le s$。

对于不同的 $p,\ q$，令

$$h_{pq}(t)=\mathbb{P}\big[pq\in E(G(S_t\cup\{p,q\}))\big].$$

如果 $\tau_q=t<\tau_p$ 满足 $t>s$，那么 $B_\pi(q)=S_t\setminus\{q\}$，因此仅当定义 $h_{pq}(t)$ 的事件发生时，q 才会对 p 上的负载作出贡献。固定 t 时，该事件与端点时钟独立，因为两个端点都包含在测试集中。对该事件概率、密度 $\mu_q e^{-\mu_q t}$ 以及 $\tau_p>t$ 的概率 $e^{-\mu_p t}$ 的乘积进行积分，即可得到期望贡献。乘以 μ_p 并对有序对求和，便计入了（2）中的指示函数。常数项至多贡献 $\sum_{p\in P}\mu_p=1$，因此

$$\mathbb{E}[\mu\cdot L_{A,\pi}]\le 1+\int_s^\infty a_\mu(t)\,dt,\qquad a_\mu(t)=\sum_{p\ne q}\mu_p\mu_q e^{-(\mu_p+\mu_q)t}h_{pq}(t).$$

如果 $p,\ q\in S_t$，则同一测试会检查 pq 是否为 $G(S_t)$ 的一条边。由于与端点时钟无关，并且每条边被计数两次，可得

$$\sum_{p\ne q}(1-e^{-\mu_p t})(1-e^{-\mu_q t})h_{pq}(t)=2\mathbb{E}|E(G(S_t))|.$$

使用 $\mu_p e^{-\mu_p t}\le(1-e^{-\mu_p t})/t$ 在两个端点处，则由 (1) 和 $\mathbb{E}|S_t|\le\min\{t,\ n\}$，可得

$$a_\mu(t)\le\frac{2\kappa\mathbb{E}|S_t|}{t^2}\le\frac{2\kappa\min\{t,n\}}{t^2}.$$

在 $[s,n]$ 和 $[n,\infty)$ 上积分可得 (4)。 □

我们通过将 WeightedOrder 的前缀与 S_s 配对，把时钟界转移到 WeightedOrder 上。

引理 11（WeightedOrder 的 加权负载）。 假设（1）成立。固定一个整数 $1\le s\le n$ 和正整数权重 $u\ (p)$，并设 $\mu_p=u\ (p)\ /\sum_{a\in P}u\ (a)$。对于算法 *1* 返回的点对（$A,\ \pi$），有 $\mathbb{E}|A|\le s$ 并且

$$\mathbb{E}[\mu\cdot L_{A,\pi}]\le C_s,\qquad C_s=1+2\kappa(1+\ln(n/s)). \tag{5}$$

证明。 令 $\alpha_p=\min\{1,s\mu_p\}$。由于 $\alpha_p\le s\mu_p$，我们有 $\mathbb{E}|A|\le s$。利用独立的均匀随机变量 U_p，将 A 与引理 10 中的 S_s 耦合，取值于 $[0,1]$。若满足 $U_p\le 1-e^{-s\mu_p}$，则将 p 纳入 S_s；若满足 $U_p\le\alpha_p$，则将其纳入 A。不等式 $\alpha_p\ge 1-e^{-s\mu_p}$ 给出 $S_s\subseteq A$。在给定 U_p 的条件下，根据无记忆性，将 $p\notin S_s$ 的剩余时间 τ_p-s 抽样为速率为 μ_p 的相互独立指数随机变量。

将 $A\setminus S_s$ 移入前缀，同时保持 A 之外的相对顺序。A 中的点负载为零，不会对（2）贡献任何项。对于 $p,q\notin A,\ q$ 之前的前缀只能扩大，因此一个范围

在移动后见证某项，也必然在移动前见证该项。因此负载不会增加，（4）仍然适用。

给定 A，$p \notin A$ 的剩余时间 $\tau_p - s$ 是速率为 μ_p 的独立指数随机变量。来自剩余集合 M 的下一个点是 p，其概率为 $u(p)/\sum_{a\in M} u(a)$，与算法 1 完全相同。这证明了 (5)。

□

引理 11 给出了每一轮期望加权负载的界。乘法更新给出了固定击中集上的期望总负载的界。

引理 12 (固定猜测的代价)。设 $s \in [1, n]$ 为一个整数。对于任何至多可由 s 个点命中的固定请求序列，使用算法 *2* 返回的顺序的优先级规则，其期望代价至多为

$$(6 + 4\kappa)s(1 + \ln(n/s)). \tag{6}$$

证明。固定 $U \subseteq P$，其中 $|U| = s$，并记 $W_j = \sum_{p\in P} u_j(p)$ 和 $\mu_j(p) = u_j(p)/W_j$。更新给出 $\ln(u_{T+1}(p)/D^T) = \sum_{j=1}^{T} \ln(1+\widehat{L}_j(p)/D)$。我们将这个固定集合上权重的增长与总权重的增长进行比较。根据算术—几何平均不等式，U 上最终权重的几何平均至多为 W_{T+1}/s。利用 $x/2 \le \ln(1 + x) \le x$（其中 $x = L_j(p)/D \in [0, 1]$）可得

$$\begin{aligned}\frac{1}{2Ds}\sum_{p\in U}\sum_{j=1}^{T}\widehat{L}_j(p) &\le \frac{1}{s}\sum_{p\in U}\ln\frac{u_{T+1}(p)}{D^T} \le \ln\frac{W_{T+1}}{sD^T}\\ &\le \ln\frac{n}{s} + \frac{1}{D}\sum_{j=1}^{T}\mu_j\cdot\widehat{L}_j. \end{aligned}\tag{7}$$

对于最后一个不等式，使用 $W_1 = n$ 和 $W_{j+1}/(DW_j) = 1 + \mu_j \cdot L_j/D$。

在给定前面各轮次的条件下，(A_j, π_j) 的条件分布是固定的，并且 $\widehat{L}_{A_j,\pi_j}$ 对其无偏估计。对 (A_j, π_j) 取平均并应用式 (5) 得到

$$\mathbb{E}[\mu_j \cdot \widehat{L}_j \mid \text{preceding rounds}] \le C_s.$$

在式 (7) 中取期望，利用无偏性，并对 J 取平均（J 在各轮次之外均匀分布），得到

$$\frac{1}{s}\sum_{p\in U}\mathbb{E}L_{A_J,\pi_J}(p) \le \frac{2D}{T}\ln(n/s) + 2C_s.$$

将固定序列的大小至多为 s 的击中集扩展为集合 U，其大小为 s。由非负性和引理 9，期望代价至多为 $\mathbb{E}|A_J|$ 加上 U 上的期望总负载。利用引理 11 中的 $\mathbb{E}|A_J| \le s$ 和 $D \le T$ 可得式 (6)。

□

定理 *8* 的证明。每次请求之后，X 都包含其选定的点以及此前所有选定的点，因此它命中截至目前所见的所有请求。每个阶段都以 $Y = \varnothing$ 和正预算开始。在最后一个阶段，由 $|Y| = B_n = n$ 可推出 $Y = P$，因此不可能再发生阶段变更。

令 s_* 为不小于 k 的最小猜测值。在猜测值低于 s_* 的阶段中所选取的点数，至多等于这些阶段的预算之和。若 $s_* > 1$，令 r 为低于 s_* 的最大猜测值。对于 $s < n$，写作 $g_s = 1 + \lceil \log_2(n/s) \rceil$。更早的猜测值具有 $r/2^i$ 的形式，其中 $i \ge 0$ 为整数，且 $g_{r/2^i} = g_r + i$，因此它们的总预算为

$$O\left((1+\kappa)r\sum_{i\ge 0} 2^{-i}(g_r + i)\right) = O((1+\kappa)rg_r).$$

由于 $r < k \le s_* \le 2r$，这就是 $O((1+\kappa)k(1+\ln(n/k)))$。

固定 $s_* \le s < n$，并在此条件下考虑较小猜测值对应的顺序。若达到阶段 s，则在该条件下，其起始请求及对应后缀均已确定。顺序 π_s 与较小猜测值对应的顺序相互独立。对于该后缀，其击中集大小为 $k \le s$；根据引理 12 及预算选择，采用 π_s 的优先级规则时，条件期望代价至多为 $B_s/4$。仅当该规则会在后缀上选出超过 B_s 个不同的点时，算法才会推进。因此，马尔可夫不等式将推进的条件概率界定为 $1/4$。

最终 X 是所有已达到阶段的最终集合 Y 的并集，每个集合的大小至多为其预算。从 s_* 开始，达到下一阶段的概率至多是达到当前阶段概率的四分之一。由于猜测值至多翻倍且 g_s 不增加，每个后续预算至多是前一个预算的两倍。在最终的转换中同样成立，此时对于最大猜测值 $s < n$，$B_n = n \le 2B_s$。因此，从 s_* 开始的期望代价至多为

$$\sum_{i \ge 0} B_{s_*} 2^i 4^{-i} = 2B_{s_*}.$$

由于 $k \le s_* \le 2k$，这同样是 $O((1+\kappa)k(1+\ln(n/k)))$。

算法 2 的每一轮最多进行 $n-1$ 次边测试。因此，每次猜测进行 $T = 2n$ 轮，共进行 $O(\log n)$ 次猜测，总计使用 $O(n^2 \log n)$ 次边测试。一个权重是 $O(n)$ 个至多为 $2D$ 的整数的乘积，因此包含 $O(n \log n)$ 位。所需的抽样可以在期望多项式时间内精确实现。

□

5 平面圆盘

我们验证圆盘的 (1)，并说明如何测试德洛内邻接，以及如何找到按当前顺序到达的圆盘中的第一个点。记 $D(c,\ \rho) = \{x \in \mathbb{R}^2 : \|x - c\| \le \rho\}$ 为中心为 c、半径为 ρ 的闭圆盘。

5.1 德洛内图的平面性

定理 1 的证明。 固定 $S \subseteq P$，并将 $G(S)$ 的每条边画成一条线段。对于每条边，选择一个包含其端点且不包含 S 中其他点的圆盘。

对于端点不相交的两条边，将它们的圆盘分别记为 $f(x) \le 0$ 和 $g(x) \le 0$，其中 $f(x) = \|x - c\|^2 - \rho^2$ 对于 $D(c, \rho)$，对 g 也同样如此。在第一条线段的端点处，$f \le 0 < g$。在第二条线段的端点处，$g \le 0 < f$。$f - g$ 中的二次项相互抵消，因此这个差是仿射的。它沿第一条线段严格为负，沿第二条线段严格为正。因此，这两条线段不相交。

根据凸性，每条线段都位于其圆盘内，因此不包含 S 的第三个点。特别地，如果两条共享一个端点的线段发生重叠，那么较长的线段将包含较短线段的另一个端点。因此，该绘图是平面的，并且 $|E(G(S))| \le 3|S|$。定理 8 与 $\kappa = 3$ 一起给出了期望代价的成本界。

□

5.2 圆盘的几何查询

每个圆盘请求由其中心和半径指定。我们将坐标上的算术运算、平方根计算和比较运算均按单位成本计数。

预处理期间的边测试。 当一轮采样 $q \notin A$ 时，我们在 π 中测试所有点对 $p,\ q$ 与p在 q 之后。每项测试都询问一个圆盘是否包含该点对，并漏掉 $B_\pi(q)$。以 q 为中心的反演将这些测试转化为针对单个凸包的成员查询。对于 $r \neq q$，定义

$$a_q(r) = \frac{2(r-q)}{\|r-q\|^2}, \qquad H_q = \operatorname{conv}\big(\{0\} \cup \{a_q(r) : r \in B_\pi(q)\}\big).$$

$$\{p,q\} \in E\big(G(B_\pi(q) \cup \{p,q\})\big) \quad \Longleftrightarrow \quad a_q(p) \notin H_q. \tag{8}$$

证明 (8)。 设一个中心为 $q+u$、半径为 ρ 的圆盘包含 $p,\ q$，且不包含该前缀。由于圆盘包含 q，因此有 $b = \rho^2 - \|u\|^2 \geq 0$。在相对于 q 的坐标中，展开圆盘不等式并除以 $\|r-q\|^2$。点 $r \neq q$ 属于该圆盘，当且仅当

$$u \cdot a_q(r) + \tfrac{b}{4}\|a_q(r)\|^2 \geq 1.$$

由于 $b \geq 0$，左侧关于变换后的点是凸的。它在原点处为零，并且在每个变换后的前缀点处小于 1，因此在整个 H_q 中都小于 1。在 $a_q(p)$ 处，该值至少为 1，因此 $a_q(p)$ 位于 H_q 外部。

反之，如果 $a_q(p) \notin H_q$，取一个法向量 v，它对应于一条将 $a_q(p)$ 与 H_q 严格分离的直线，并将其定向为

$$a_q(p) \cdot v > \max_{x \in H_q} x \cdot v \geq 0.$$

令 $u = v/(a_q(p) \cdot v)$，其分母为正，因为 $0 \in H_q$。中心为 $q+u$、半径为 $\|u\|$ 的圆盘满足 $b = 0$，并且 $u \cdot a_q(p) = 1$，因此 p 和 q 都位于其边界上。每个前缀点 r 都满足 $u \cdot a_q(r) < 1$，因而位于该圆盘外部。 □

对于每个 q，预先计算变换后的点，并将它们与原点一起按字典序排序。在采样 q 的轮次中，扫描此列表，根据原点和变换后的前缀点构造上凸包链和下凸包链。第二次扫描按递增的 x 坐标测试查询点。沿着凸包链前进，找出凸包与经过每个查询点的竖直线的交点，并测试该查询点是否属于该交集。对于竖直凸包边，测试其是否属于整个线段；如果凸包是一个点或一条线段，则直接测试其是否属于其中。

每次扫描使用 $O(n)$ 次几何操作，并且对于有理数输入仅使用有理数运算。在每个 $O(\log n)$ 个猜测中，复用经过排序的列表，跨越 $O(n)$ 轮，总计执行 $O(n^2 \log n)$ 次几何操作，其中包括排序。

查找到达圆盘中的第一个点。 对于算法 3 使用的每个优先级顺序（包括 π_n），我们构建一种数据结构，用于返回 P 中满足 P 的圆盘内的第一个点。一次查询的最坏情况时间为 $O(\log^2 n)$。为全部 $O(\log\ n)$ 个优先级顺序构建数据结构需要 $O(n\log^3 n)$ 次几何操作，处于 $O(n^2 \log n)$ 的预处理界内。每个请求最多需要两次查询和 $O(1)$ 次额外操作。附录 B 给出了构造方法及其分析，包括有理数输入的位复杂度。

5.3 Amatching 下界

我们使用 k 个点组，每个点组都有一个嵌套的圆盘请求序列。算法必须先命中当前圆盘，才能获知下一个请求保留其候选点中的哪一半。

定理 *2* 的证明。 固定 $1 \le k \le n$ ，并令 $h = \lfloor \log_2(n/k) \rfloor$。将 k 个点组中的每组 2^h 个点按单位间距放置在水平轴上。对于一个点组内的连续区块，令 a, b 分别表示其最左端点和最右端点的坐标。使用圆盘中心为 $((a+b)/2, 0)$、半径为 $(b-a)/2+1/4$的圆盘。沿水平轴方向，它在每个端点外延伸 $1/4$ ，因此恰好包含其所在点组内该区块的所有点。将一个端点向内移动 δ ，会使圆盘中心移动 $\delta/2$ ，并使半径减少相同的量。因此，子区块对应的圆盘位于完整区块对应的圆盘内部。将各点组分隔开，使包含完整点组的圆盘两两不相交，并将任何剩余的候选点放置在这些圆盘之外。

在每组中，令 I_0 为完整区块。对于 $i = 1$、.....、h，将 I_{i-1} 分成两个长度相等且连续的半区块，并通过公平抛硬币从中选择 I_i 。在请求到达之前抛出所有硬币，各次抛掷彼此独立，且独立于算法的随机选择。按顺序逐组揭示 I_0、.....、I_h 对应的圆盘。每组都可以由 I_h的唯一一个点命中。不同组中的第一个圆盘互不相交，因此该分布中的每个序列都有 $|\mathrm{OPT}| = k$。

固定一个随机化算法，并考察其中一组。令 H_i 表示在请求 i结束时， I_i 中被选取的点数。对于 $i \ge 1$，令 A_i 表示在该请求中首次被选取的 I_i 中的点数。令 A_0 表示在其第一个请求结束时， I_0 中被选取的所有点数，包括该组开始之前已经选取的点。因此 $H_0 = A_0$。在给定截至请求 $i-1$的历史条件下， I_{i-1} 中被选取的 H_{i-1} 个点各自属于 I_i 的概率为 $1/2$。由于 $I_i \subseteq I_{i-1}$，H_i 所计数的点恰好是保留下来的选择和新选取的点。对于 $i \ge 1$,

$$\mathbb{E}H_i = \tfrac{1}{2}\mathbb{E}H_{i-1} + \mathbb{E}A_i,$$

期望取遍随机划分和算法。每个请求都必须被命中，因此 $H_i \ge 1$。对递推式求和可得

$$\sum_{i=0}^{h} \mathbb{E}A_i = \mathbb{E}H_h + \frac{1}{2}\sum_{i=0}^{h-1} \mathbb{E}H_i \ge 1 + \frac{h}{2}.$$

每个点至多贡献给一个 A_i，且各组彼此不相交。对各组求和可得：对序列和算法取期望时，代价至少为 $k(1+h/2)$。因此，存在某个固定序列，使得对算法的随机选择取期望时，代价至少为 $k(1+h/2)$。 □

6 最短路径球

定义 13 （最短路径球）。 设 H 是一个边长为正的有限连通无向图。以 $c \in V(H)$ 为中心、半径为 $r \ge 0$ 的最短路径球为

$$B_H(c, r) = \{x \in V(H) : d_H(c, x) \le r\},$$

其中 d_H 是最短路径距离。

与 P相交的球会诱导请求 $B_H(c,\ r) \cap P$。当 H不含某个固定子式时，证明每个 $G(S)$ 都是 H 的子式，即可得到线性边数上界（1）。

6.1 沃罗诺伊单元与边界

固定一个非空的 $S \subseteq P$。将 H 中的每个顶点分配给其在 S中的最近站点，并按照 S上的固定顺序打破平局。我们证明每个单元都是连通的，并且对于 $G(S)$ 的每条边 pq ， p和 q 的单元相邻。

如果 x 属于 p 的单元，且 y 位于从 x 到 p的最短路径上，那么对于任意 $q \in S$，由三角不等式可得

$$d_H(y,q) - d_H(y,p) \geq d_H(x,q) - d_H(x,p).$$

因此， p 在 y处仍是最近站点。任何在 y 处与 p 并列的站点，在 x处也与 p 并列，因此相同的平局处理规则会将 y 分配给 p。整条路径都位于 p的单元中，这就证明了连通性。收缩各单元、删除自环并合并平行边，得到一个顶点集为 S的简单子式 H_S。

现在令 $B_H(c,\ r)$为一个球，其在 S上的迹为$\{p,\ q\}$。沿着一条从 c到 p的最短路径前进。对于该路径上的任意顶点 x 以及任意站点 $a \in S \setminus p,\ q\}$，

$$d_H(x,a) \geq d_H(c,a) - d_H(c,x) > d_H(c,p) - d_H(c,x) = d_H(x,p),$$

因为 a 在球外，而 p 在球内。从 c 到q的最短路径同样适用这一论证。因此，两条路径都位于 p 和 q的单元并集内。它们共同构成一条从 p到 q 的游走，并且会跨越单元之间的一条边。因此， $pq \in E(H_S)$和 $G(S)$是 H_S的子图。

如果 H 是平面图，那么 H_S也是平面图，并且至多有 $3|S|$ 条边。如果 H 不含 F 作为子式，那么H_S 也不含以 $|V(F)|$ 个顶点构成的完全图作为子式，因为该图包含 F作为子图。Mader 界 [18] 给出 $O_F(|S|)$条边。在这两种情况下， $G(S)$都满足(1)，因此定理 8 给出了定理 3 中的期望代价界。

6.2 预处理与球请求

通过从每个候选点运行 Dijkstra 算法，预先计算每个候选点到 H 的所有顶点的距离。

要测试 $pq \in E(G(S))$，将每个顶点 c 视为中心，并使用使球同时包含 p 和 q 的最小半径。当且仅当球在 S 上的迹为 $\{p,q\}$ 时，

$$\max\{d_H(c,p), d_H(c,q)\} < \min_{a \in S \setminus \{p,q\}} d_H(c,a),$$

其中 $\min \varnothing = +\infty$。不等式是严格的，因为距离等于半径的顶点属于该球。

当球 $B_H(c,r)$ 到达时，将 r 与存储的从 c 到候选点的距离进行比较，并将所得范围传递给算法 3。每个距离都是沿一条简单路径的有理数边长之和，因此具有多项式位长。计算该表、测试边以及使用有理半径处理请求，都只需在其输入位长度上耗费多项式时间。

6.3 路径上的匹配下界

在一条具有单位边长的路径上实现第 5.3 节中的区块，使用偶数索引的顶点作为候选点。将各组依次放置，并将所有额外的候选点放在最后一组之后。对于端点索引为 $a,\ b$ 的区块，其中点 $(a+b)/2$ 是一个顶点，以其为中心、半径为 $(b-a)/2$ 的球恰好包含该区块中的候选点。因此，相同的请求分布具有离线最优值 k，且期望代价下界为

$$k\left(1 + \tfrac{1}{2}\lfloor \log_2(n/k) \rfloor\right)$$

对每个 $1 \leq k \leq n$ 均成立，从而证明了平面图中的紧性。

附录 C 表明，即使在最大度至多为 3 的图中，等半径球也能实现任意范围族。

7 结论

对于任意半径的圆盘，我们的随机在线算法在离线最优值为 k 的每个固定序列上，期望选择 $O(k(1+\log(n/k)))$ 个点，从而得到 $O(\log n)$ 的期望竞争比。对于任意 k，这一成本界都是紧的，即使在共线点集上也是如此。对于排除固定子式的图中的最短路径球，同样的成本界也成立，并且在路径上是紧的。

参考文献

[1] Shanli Alefkhani、Nima Khodaveisi 和 Mathieu Mari。 d维胖对象的在线击中集。载于 Approximation and Online Algorithms（WAOA 2023），Lecture Notes in Computer Science 第 14297 卷，第 134–144 页，2023 年。 doi:10.1007/978-3-031-49815-2_10.[2] Noga Alon、Baruch Awerbuch、Yossi Azar、Niv Buchbinder 和 Joseph Naor。在线集合覆盖问题。SIAM Journal on Computing，39(2):361–370，2009 年。 doi:10.1137/060661946.[3] Sanjeev Arora、Elad Hazan 和 Satyen Kale。乘法权重更新方法：一种元算法及其应用。Theory of Computing，8:121–164，2012 年。 doi:10.4086/toc.2012.v008a006.[4] Sujoy Bhore、Anupam Gupta 和 Amit Kumar。撤回：面向结构化和几何集合系统的改进在线击中集算法。载于 SoCG 2026，LIPIcs 第 367 卷，第 14:1–14:14 页，2026 年。 doi:10.4230/LIPIcs.SoCG.2026.14.[5] Nicolas Bousquet、Wouter Cames van Batenburg、Louis Esperet、Gwenaël Joret、William Lo- chet、Carole Muller 和 François Pirot。排除某个子式的图中的球的打包与覆盖。Combinatorica，41(3):299–318，2021 年。 doi:10.1007/s00493-020-4423-3.[6] Reilly Browne 和 Hsien-Chih Chang。通过保持 minor 的支撑解决单标准度量 r-支配集问题。载于 SoCG 2026，LIPIcs 第 367 卷，第 24:1–24:17 页，2026 年。doi:10.4230/LIPIcs.SoCG.2026.24.[7] Niv Buchbinder 和 Joseph Naor。用于覆盖与打包的在线原始对偶算法。Mathematics of Operations Research，34(2):270–286，2009 年。 doi:10.1287/moor.1080.0363.[8] Minati De、Satyam Singh 和 Csaba D. Tóth。针对有界半径圆盘的在线击中集。载于 ESA 2025，LIPIcs 第 351 卷，第 50:1–50:16 页，2025 年。 doi:10.4230/LIPIcs.ESA.2025.50.[9] Minati De、Satyam Singh 和 Csaba D. Tóth。针对轴对齐正方形的在线击中集。载于 SWAT 2026，LIPIcs 第 370 卷，第 16:1–16:18 页，2026 年。 doi:10.4230/LIPIcs.SWAT.2026.16.[10] Ilan Doron-Arad 和 Joseph Naor。随机在线集合覆盖的紧 $\Omega(\sqrt{m})$ 信息论下界，2026 年。 arXiv:2609.12183.[11] Guy Even 和 Shakhar Smorodinsky。在线击中集与唯一最大着色。Discrete Applied Mathematics，178:71–82，2014 年。 doi:10.1016/j.dam.2014.06.019.[12] Steven Fortune。Voronoi 图的扫描线算法。Algorithmica，2(1–4):153–174，1987 年。 doi:10.1007/BF01840357。

[13] Chaya Keller 和 Shakhar Smorodinsky。线性规模 ε-网和（p，2）定理的新充分条件，2026 年。 arXiv:2507.07269v2.[14] Arindam Khan、Aditya Lonkar、Saladi Rahul、Aditya Subramanian 和 Andreas Wiese。几何集合覆盖与击中集的在线和动态算法。载于 *SoCG 2023*，第 258 卷 *LIPIcs*，第 46:1–46:17 页，2023 年。 doi:10.4230/LIPIcs.SoCG.2023.46.[15] David Kirkpatrick。平面剖分中的最优搜索。*SIAM Journal on Computing*，12(1):28–35，1983 年。 doi:10.1137/0212002.[16] Daniel Lemire。区间内的快速随机整数生成。《*ACM*建模与计算机仿真汇刊》，29(1):3:1–3:12，2019 年。 doi:10.1145/3230636.[17] Gang Liu 和 Haitao Wang。受直线约束的圆盘的无权几何击中集及相关问题。载于 *MFCS 2024*，第 306 卷 *LIPIcs*，第 68:1–68:15 页，2024 年。 doi:10.4230/LIPIcs.MFCS.2024.68.[18] Wolfgang Mader。图的同态性质与平均边密度。《数学年刊》，174(4):265–268，1967 年。 doi:10.1007/BF01364272.[19] Rajiv Raman 和 Saurabh Ray。非穿透区域的平面支撑及其应用。载于*ESA 2018*，第 112 卷 *LIPIcs*，第 69:1–69:14 页，2018 年。 doi:10.4230/LIPIcs.ESA.2018.69.[20] 约翰·冯·诺伊曼。与随机数字相关的各种技术。载于蒙特卡洛方法，《美国国家标准局应用数学丛书》第 12 号，第 36–38 页。美国政府印刷局，1951 年。

人工智能声明

作者在技术讨论中使用了 ChatGPT 5.6 sol 和 ChatGPT 6.1 sol，这些讨论促成了结合指数时钟与乘法权重的方法的开发。这些模型还协助进行了语言编辑和稿件呈现。作者对数学结果、证明以及最终稿件承担全部责任。

使用随机比特进行采样

主算法能够从具有有理概率的有限分布中进行精确采样。要生成 $\{0,\dots,Z-1\}$ 中的均匀整数，生成 $\lceil \log_2 Z \rceil$ 个相互独立的无偏随机比特；如果所得整数小于 Z，则接受该整数，否则重复此过程。在接受的条件下，该整数在目标区间内服从均匀分布。每次尝试成功的概率至少为 $1/2$，因此平均而言少于两次尝试即可成功。这就是拒绝采样 [20, 16]。

对于 WeightedOrder 中的前缀，取 $W = \sum_{p\in P} u(p)$，并在从 $\{0,\dots,W-1\}$ 中均匀抽取的整数小于 $\min\{W, su(p)\}$ 时纳入 p，对各个点进行独立抽样。要选择下一个点，令 M 为剩余集合，并设 $Z = \sum_{p\in M} u(p)$。按固定顺序将 $[0, Z)$ 划分为连续的半开区间，每个剩余点 p对应一个长度为 $u(p)$ 的区间。从 $\{0,\dots,Z-1\}$ 中均匀抽取的整数以概率 $u(p)/Z$ 落入 p 对应的区间。通过扫描累积权重，可以找到包含所抽取整数的区间。Q 和 J 的抽取同样是均匀整数抽取。权重包含 $O(n\log n)$ 位，所需抽样次数为多项式级，因此除边测试外，该方法可在期望多项式时间内实现主算法。

有界时间变体。 为了获得预处理时间的多项式最坏情况界，令 N 为不小于 n 的最小二的幂。算法 4 将前缀概率舍入为 $1/N^2$ 的倍数，并允许在每个位置最多进行 $2\log_2 N$ 次拒绝尝试。如果全部失败，则按固定顺序追加第一个剩余点。

算法 4 FiniteWeightedOrder

函数 FiniteWeightedOrder(P, s, N, u)
  $W \leftarrow \sum_{p\in P} u(p)$, $A \leftarrow \varnothing$
  对于每个 $p \in P$，**执行**
    $b_p \leftarrow \lceil N^2 \min\{1, su(p)/W\}\rceil$
    从$\{0, \ldots\ldots, N^2-1\}$中独立且均匀地抽取 v
    将 p 添加到 A，如果 $v < b_p$
  结束循环
  按固定顺序从 A 的点开始，并设置 $M \leftarrow P \setminus A$
  当 $M \neq \varnothing$ **时执行**
    $Z \leftarrow \sum_{p\in M} u(p)$
    尝试进行至多 $2\log_2 N$ 次均匀抽样，抽取一个 $\lceil \log_2 Z\rceil$ 位整数 v
    接受第一个 $v < Z$（如果有）
    如果接受了整数 v **则**
      取累积权重区间包含 v 的 $q \in M$
    else
      按固定顺序取第一个点 $q \in M$
    结束条件判断
    将 q 添加到 π 中，并从 M 中移除
  结束循环
  返回 (A, π)
结束函数

在 SampleOrder 中，使用此采样器以及 $D = N+1$ 和 $T = 2N$。从$\{1, \ldots, T\}$中均匀地抽取 J，且独立于所有轮次。在每一轮中，从 n 个点和 $N-n$ 个虚拟位置中均匀地抽取 Q，且独立于 (A, π) 以及之前的轮次。将 $\widehat{L}^{\mathrm{fin}}(p) = 0$ 设在 A 上，并且对于 $p \notin A$，设

$$\widehat{L}^{\mathrm{fin}}(p) = 1 + N\mathbf{1}\left[Q \in P \setminus A, \quad Q \prec_\pi p, \quad pQ \in E\big(G(B_\pi(Q) \cup \{p, Q\})\big)\right]. \tag{9}$$

如果 Q 是虚拟位置，则指示变量为零。使用这些估计值进行相同的乘法更新，并返回 π_J。

引理 14。 假设(1) 成立。固定一个整数 $1 \le s \le n$ 和正整数权重 $u(p)$，并令 N 为不小于 n 的最小二的幂。对于算法 *4* 返回的点对$(A、\ \pi)$，其中 $\mu_p = u(p)/\sum_{a\in P} u(a)$，有

$$\mathbb{E}|A| \le s+1, \qquad \mathbb{E}[\mu \cdot L_{A,\pi}] \le 2 + 2\kappa(1 + \ln(n/s)).$$

由此得到的 *SampleOrder* 变体满足代价界(6)，并且除边测试之外，具有多项式最坏情况运行时间。

证明。 令 α_p 为取整后的包含概率。由于 $\alpha_p \le s\mu_p + 1/N^2$，我们有 $\mathbb{E}|A| \le s + n/N^2 \le s+1$。引理 11 证明中的前缀耦合仍然适用，因为 $\alpha_p \ge \min\{1, s\mu_p\}$。

在此前缀之后进行精确的加权采样时，每个点的负载都不高于在 coupledclock 顺序中的负载，因此 (4) 适用。

将算法 4 与使用无限次拒绝尝试的采样器耦合起来：在某个位置的全部 $2\log_2 N$ 次尝试均失败之前，保持前缀和抽样结果完全一致。在此前所有位置都在限制次数内被接受的条件下，在每个位置发生这种失败的概率至多为 $1/N^2$。对至多 n 个位置应用并合界可知，这两个次序以至多 n/N^2 的概率不同。由于加权负载位于 $[0,n]$ 中，期望加权负载之差至多为 $n^2/N^2 \le 1$。这就得到了所声称的负载界。

式 (9) 中的估计量在 (A,π) 条件下是无偏的，并且位于 $[0,D]$ 中。使用这些参数应用乘法权重不等式 (7)，并利用 $D \le T$。记 $g = 1+\ln(n/s)$，引理 12 的论证将期望代价界定为

$$(s+1) + s\left(\frac{2D}{T}\ln(n/s) + 2(2+2\kappa g)\right) \le (6+4\kappa)sg.$$

权重仍然具有 $O(n\log n)$ 个比特。所有前缀抽样、索引 J 以及点或虚拟位置 Q 的样本空间大小都是 2 的幂。其余抽样具有固定的尝试次数上限。因此，除边测试之外，随机比特数和运行时间均由多项式界定。每次猜测有 $T = 2N = O(n)$ 轮，因此在所有猜测中，边测试次数仍为 $O(n^2\log n)$。因此，相同的预算和阶段分析给出了定理 8 中的成本界。

□

B 查找圆盘中的第一个点

我们在二叉树搜索中使用最近邻查询，如 Liu and Wang [17, 引理 8]所述。对于算法 3 所使用的每个优先级顺序（包括 π_n），按该顺序以 P 的点为叶节点构建一棵平衡二叉树。在每个内部节点，构建其子树中点的 Voronoi 图，并对其进行点定位预处理 [12, 15]。对于 m 个站点，构建过程使用 $O(m\log m)$ 次几何操作，而一次最近邻查询需要 $O(\log m)$ 时间。叶节点使用其唯一的站点回答查询。

根节点的点集与圆盘相交。在每个内部节点，找出左子树中距离圆盘中心最近的点。如果该点位于圆盘内，则向左下降；否则向右下降。因此，当前子树始终与圆盘相交。由于左子树中的每个点都排在右子树中的每个点之前，因此最终到达的叶节点就是圆盘中的第一个点。该搜索使用 $O(\log n)$ 次最近邻查询。

每个树级别的图示总共至多包含 n 个站点。其构建过程对于每个阶次使用 $O(n\log^2 n)$ 次几何操作，所有阶次合计为 $O(n\log^3 n)$，符合 $O(n^2\log n)$ 的预处理上界。每个请求至多进行两次搜索：一次在当前阶次中进行；如果阶段发生变化，则在下一阶次中再进行一次。记录每个点最后一次进入 Y 的阶段，以支持常数时间的成员测试，并将 X 存储在布尔数组中。因此，每个请求的最坏情况耗时为 $O(\log^2 n)$。

对于有理点坐标、中心和半径，这些运算可以在关于输入位长度的多项式时间内精确完成。Voronoi 构造中的每个事件坐标，都可以通过对数量恒定的站点坐标进行常数次算术运算和平方根运算得到。

最大度至多为三的球图

给定一个在线击中集实例，我们构造一个具有单位边长的图，并为底集中的每个元素设置一个候选顶点。原始族中的每个范围都是某个球在这些顶点上的迹，且所有球具有相同的半径。

从将每个元素与包含它的范围相连接的二部关联图开始。将每个非孤立顶点替换为一棵有根二叉树，每条关联边对应一个不同的叶节点；孤立顶点保持不变。令 $h \geq 1$ 为树高的一个整数上界。对于关联图的每条边，将相应的叶节点通过一条长度为 $L = 2h+1$ 且含有新的内部顶点的路径连接起来。这些路径连接在不同的叶节点上，因此所得图的最大度至多为三。使用元素树的根作为候选顶点。

将一个范围表示为以其树根为中心、半径为 $R = L + 2h$ 的球。对于范围内的元素，连接路径以及每棵树中至多 h 条边，构成一条连接两个根的路径，其长度至多为 R。对于不在该范围内但位于同一连通分量中的元素，根之间的任意路径都会在元素树和范围树之间交替。由于该元素不属于该范围，这样的路径至少使用三条连接路径，其长度至少为 $3L > R$。

为使宿主图连通，将每个连通分量中的一个顶点通过长度为 $R + 1$ 的路径连接到一棵新二叉树的一个不同叶节点上。对于非单点连通分量，将新路径连接到一条长度为 L 的路径的内部顶点上。对于单点连通分量，则将其连接到该分量唯一的顶点上。连接顶点的度数至多为二，因此最大度仍至多为三。连通分量内部的距离保持不变：任何离开并返回该连通分量的游走都会经过同一个连接顶点。不同的原始连通分量之间的距离仍大于 R。

针对这些球的在线击中集算法会选择候选点根，这些根对应于原实例中的元素。将每个到达范围所对应的球呈现给该算法，可以保持代价、可行性和离线最优值不变。因此，针对非自适应对手的随机化算法竞争比的 $\Omega(\sqrt{n})$ 下界同样适用于这些球实例（第1.1节）。宿主图中的顶点数量可能关于 $|P|$ 呈指数增长，因为其中为每个范围都包含一棵树。