arXiv:2304.10028v2 [cs.CG] 8 Jan 2024

最小化移动实体中心的不确定区域大小*

William Evans 和 Seyed Ali Tabatabaee

加拿大温哥华不列颠哥伦比亚大学计算机科学系{will,salitaba}@cs.ubc.ca

摘要。 本文研究在欧几里得空间中，以有界速度移动的对象的 1-中心、质心和 1-中位点问题。我们每单位时间只能获取固定数量（通常为一个）对象的精确位置；对于其他每个对象，其可能位置的集合（称为对象的 *uncertainty region*）会不断扩大，且仅受速度限制约束。因此，对象的中心可能位于多个位置，这些位置构成中心的不确定区域。针对这些中心点问题，我们设计了查询策略，以最小化中心不确定区域的大小，并将其性能与一种最优查询策略进行比较；该策略掌握对象的轨迹，但仍须通过查询来减少其不确定性。对于 $\mathbb{R}^1$ 中 1-中心问题的静态情形，我们给出一种算法，每单位时间查询四个对象；相较于每单位时间进行一次查询的最优算法，该算法具有 1-竞争性。对于 $\mathbb{R}^1$ 中 1-中心问题的一般情形、$\mathbb{R}^d$ 中的质心问题，以及 $\mathbb{R}^1$ 中的 1-中位数问题，我们证明轮询调度算法是可能达到的最佳竞争算法。对于 $\mathbb{R}^d$ 中的质心问题，我们给出一种 $O(\log n)$-竞争算法。此外，对于 $\mathbb{R}^d$ ($d \geq 2$) 中 1-中心问题的一般情形，我们证明没有任何算法能够保证相对于最优算法具有有界竞争比。

关键词： 运动中的数据 · 不确定输入 · 中心点问题 · 在线算法。

1 引言

许多现实世界中的问题，例如控制空中交通和为移动电话提供服务，都涉及移动实体。因此，分析移动对象已成为理论计算机科学领域中一个备受关注的课题。在许多问题中，对象的移动是不可预测的，数据处理必须实时进行。此外，在任意时间点获取某个对象的精确位置通常需要付出成本。因此，在任何时刻获知所有对象的精确位置都是不切实际的。相反，对于每个对象，我们都知道一个潜在位置区域，称为其不确定区域。由于

* 本研究部分得到了NSERC探索资助以及UBC计算、信息与认知系统研究所（ICICS）的资助。

随着对象移动，除非获取其精确位置，否则每个对象的不确定区域会随时间增大。目标是设计经济高效的算法，以高精度分析移动对象。

中心常用于表示给定的点集，并应用于数据聚类和设施选址。对于欧几里得空间中的给定点集，最常见的三种中心如下：

1. **1-中心：** 给定 $\mathbb{R}^d$中的一组 n 点， k-中心 是一组 k 设施位置，使输入点到最近设施的最远距离最小。对于 $k = 1$，它是包含输入点集的最小球的中心。
2. **质心：** 给定 $\mathbb{R}^d$中的一组 n 点， k-质心是 k-均值问题的解，它是一组 k 设施位置，使每个输入点到最近设施的距离平方和最小。1-质心称为质心。给定 $\mathbb{R}^d$ 中的一组 n 加权点，质心 是根据各点质量加权后得到的平均位置。当所有输入点的权重均为单位权重时，质心就是质心。3. **1-中值：** 给定 $\mathbb{R}^d$中的一组 n 点， k-中值是 k 个设施的集合，使输入点到最近设施的平均距离最小。

1.1 模型与定义

给定一个中心函数以及 $\mathbb{R}^d$ 中 n 个移动对象的初始位置集合（$n \geq 2$），其中每个对象的速度均不超过 v，我们考虑如下问题：寻找查询策略（每个时间单位结束时可查询一次），以最小化这样一个度量：在所有查询时刻，指定中心的不确定区域的最大尺寸。我们将区域的尺寸定义为该区域内各点之间的最大两两距离。我们只关注刚刚完成一次查询且某个对象的不确定区域尺寸为 0 时的不确定区域。鉴于对于某些问题实例而言，任何查询策略对应的度量都很大，我们在竞争框架下分析算法的性能，其中竞争因子是我们的算法所得度量与最优算法所得度量之比。我们假设最优算法知道对象的轨迹，但仍必须进行查询，以保持移动对象的不确定区域足够小，从而使其中心的不确定区域也足够小。

在本文的某些部分，我们考虑问题的加权版本，其中每个对象都有一个正权重。不过，我们主要讨论问题的非加权版本；因此，除非另有说明，我们假设所有对象的权重均为单位权重。图 1 展示了不确定位置的少量加权对象的质心问题（质心问题）的加权版本。我们还考虑对象具有不同最大速度的问题版本。不过，除非另有说明，我们假设所有对象的最大速度相同。我们将我们所研究问题的静态情形定义为

一种特殊情形：对象虽然具有最大速度 v，且其不确定区域会相应扩大，但实际上处于静止状态（不会移动）。为了获得更优且更有意义的竞争比，我们可以允许算法在单位时间内查询多个对象。

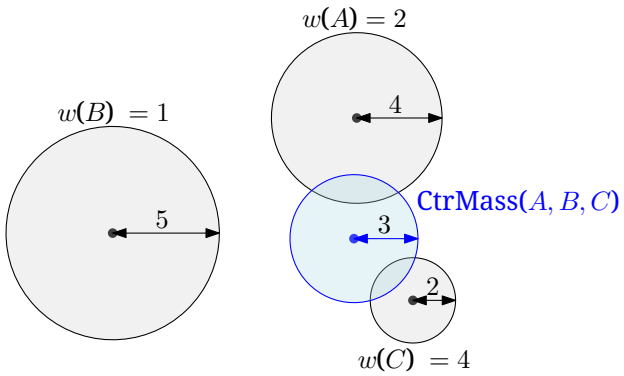

图1：三个加权对象 A、B 和 C 的不确定区域，以及它们质心的可能位置（不确定区域）。

1.2 贡献与组织结构

本文研究欧几里得空间中移动实体的上述中心点问题。我们探讨如何设计竞争性查询策略，以最小化这些不同中心的不确定区域大小。对于 $\mathbb{R}^1$ 中 1-中心问题的静态情形，我们证明存在一种算法，每单位时间查询常数个对象，并且相对于每单位时间进行一次查询的最优算法，该算法是 1-竞争的。对于 $\mathbb{R}^1$ 中移动实体的 1-中心问题，我们给出一种算法，其竞争比为可能达到的最佳值 $O(n)$。对于 $\mathbb{R}^d$ $(d \geq 2)$ 中的这一问题，我们证明没有任何算法能够保证相对于最优算法具有有界竞争比。对于 $\mathbb{R}^d$ $(d \geq 1)$ 中移动实体的质心问题，我们提出一种相对于最优算法为 1-竞争的算法。对于 $\mathbb{R}^d(d \geq 1)$ 中的质心问题加权版本（质心问题），若对象具有不同的最大速度，我们给出一种 $O(\log n)$-竞争算法。对于 $\mathbb{R}^1$ 中移动实体的 1- 中位数问题，我们给出一种算法，其竞争比为可能达到的最佳值 $O(n)$。我们的一些算法采用轮转调度来处理部分实体，并按照固定顺序反复查询该子集中的实体。

本文其余部分组织如下。第2节介绍本文所引入的中心问题，以及具有不确定输入和移动数据的函数计算的背景信息。第3、4和5节研究1-中心问题，

在所提出的模型中，分别是质心问题和 1-中位数问题。最后，第 6 节总结本文提出的结果，并列出一些未解决的问题。

2 背景

k-中心问题及其特殊情况 1-中心问题（也称为最小包围球问题）有多种应用，例如数据分类和设施选址。 k-中心问题已知在欧几里得平面中是 NP-hard 的 [28]。然而，对于任意度量空间中的该问题，都存在多项式时间的 2-近似算法 [17, 20]。欧几里得 1-中心问题始终有唯一解。在固定维度下，该问题存在精确的多项式时间算法 [10, 30] 。当维度不固定时，存在近似因子为 $1+\epsilon$ 的多项式时间近似算法 [31]。在 $\mathbb{R}^d$ ($d \geq 2$) [6] 中，有界速度移动对象的欧几里得 1-中心的速度无界。Durocher [14] 针对有界速度移动对象的欧几里得 k-中心，在 $k \leq 2$ 时提供了有界速度近似，并证明当 $k \geq 3$ 时，该问题不存在有界速度近似。

k-均值问题在数据聚类和设施选址中有着广泛应用。已证明，即使对于 $\mathbb{R}^d$ $k=2$ [13]，该问题也是 NP-hard 的。Lloyd 方法 [27] 被广泛用于求解 k-均值问题目标函数的局部最小值。该方法从输入点的任意 k-聚类开始，并计算各聚类的质心。接着，该方法反复将每个点分配给最近的聚类中心，并重新计算聚类中心。针对欧几里得 k-均值问题，已有研究提出了常数因子近似算法 [1, 24]。该问题目前已知的最佳近似保证为 6.357 [1]。质心是一个唯一的点，表示一组对象的总质量集中所在的位置。质心问题在物理学、天文学和工程学等多个领域都有应用。质心的速度等于每个对象的动量之和除以所有对象的总权重。因此，该速度不超过对象可能达到的最大速度。

k-中位数问题是另一个研究颇多的问题，在数据聚类和设施选址中有着广泛应用。已知该问题在欧几里得平面中是 NP-hard 的 [28]。因此，针对该问题已提出常数因子近似算法 [1, 9]。目前已知该问题最好的近似比为 2.633 [1]。1-中位数问题也称为费马—韦伯问题 [12], ，是 k-中位数问题的一个特例。当输入点不共线 [26] 或输入点的数量为奇数时，欧几里得 1-中位点是唯一的。然而，如果输入点的数量为偶数且这些点共线，那么位于两个中间输入点之间线段上的任意一点都是这些点的一个 1-中位点。按照惯例，此类点集的欧几里得 1-中位点定义为集合中两个中间点的中点。一般而言，欧几里得 1-中位点的精确位置无法计算

当点的数量大于或等于五个时，无法用有理数域上的根式表示 [3]。因此，针对该问题设计了 $(1+\epsilon)$-近似解 [2,11]。欧几里得 1-中位点在 $\mathbb{R}^d$ $(d \geq 2)$ 中会发生不连续移动 [14]。Durocher [14] 为具有有界速度的移动对象的欧几里得 1-中位点提供了有界速度近似，并证明当 $k \geq 2$ 时，欧几里得 k-中位数问题不存在有界速度近似。

具有不确定输入的函数计算一直是多个研究项目的研究主题 [18, 25, 29]。研究人员也对存在不确定性的几何问题进行了研究，并提出了见证算法等方法来解决这些问题 [7]。由于许多涉及移动对象的问题中，移动过程具有不可预测性，因此移动数据的实时处理与不确定性计算密切相关。Kahan [22, 23] 研究了移动数据中的最大值问题、排序问题以及一些几何问题，其目标是降低数据采集成本。此外，针对单位时间内的查询次数受到限制的问题，研究人员开发了竞争性查询策略 [15, 16, 32]。

3 1-中心问题

在本节中，我们考虑移动实体的 1-中心问题。我们首先考虑该问题的一个静态版本，在这个版本中，所有查询恰好都会返回对象的原始位置，尽管我们仍必须通过查询来确认这一点。这种情形下的解决方案比一般情形更简单，因为在一般情形中，对象的移动可能会改变它们在计算中心时的重要性。然而，即使是静态情形，仍然具有挑战性。

3.1 $\mathbb{R}^1$ 中的静态情形

对于 $\mathbb{R}^1$ 中 1-中心问题的静态情形，我们证明存在一种算法，每单位时间查询四个对象，并且相对于每单位时间进行一次查询的最优算法具有 1-竞争性。

我们首先引入一个引理，以证明满足上述保证的算法存在。该引理考虑了无迁移窗口调度 [5] 问题的一个特殊情形，并涉及针轮调度（pinwheel scheduling） [8, 21]的概念。给定一个由正整数组成的多重集 $A = \{a_1, ..., a_n\}$，针轮调度问题要求构造一个关于 $\{1, ..., n\}$ 的无限序列，使得任意整数 $i \in \{1, ..., n\}$ 都至少在该序列的任意连续 a_i 个元素中出现一次。A 的密度定义为 $d(A) = \sum_{i=1}^{n} \frac{1}{a_i}$。可调度性的必要条件是 $d(A) \leq 1$ [21]。此外，$d(A) \leq 0.75$ 是可调度性的充分条件 [19]。基于这些结果，我们在附录 A 中证明了该引理。

引理1. 给定一个正整数多重集 A，其中 $d(A) \leq 2$，可以将 A 划分为三个子集，这些子集本身也是多重集，且每个子集都有一个风车调度。

无迁移窗口调度是窗口调度问题 [4] 的一个受限版本。给定一个正整数多重集 $A = \{a_1, ..., a_n\}$ 和一个正整数 h，窗口调度问题要判断是否可以将 n 个页面安排在 h 条信道上，使得任意时刻每条信道上至多有一个页面，并且第 i 个页面在连续两次出现（可在任意信道上）之间的时间间隔至多为 a_i（$1 \leq i \leq n$）。在无迁移窗口调度中，一个页面的所有出现都必须安排在同一条信道上。在证明下面的定理1时，即使引理1的证明没有利用这种灵活性，允许迁移也不会影响我们对引理1的使用。已有研究表明，可以使用 $d(A) + O(\ln(d(A)))$ 条信道实现窗口调度 [4]。不过，对于引理1所考虑的特殊情形，它给出了更好的保证。

现在，我们可以证明本小节的主要定理了。

定理1。对于 $\mathbb{R}^1$ 中的1-中心问题的静态情形，存在一种算法，每单位时间查询四个对象，并且相对于每单位时间查询一次的最优算法，该算法具有1-竞争比。

证明。不失一般性地，我们假设最大速度 v 为单位值。我们根据对象的位置对其进行排序，并令 x_i 表示第 i 个对象（$x_1 \leq ... \leq x_n$）的位置。我们定义 $f(x, y) = \lfloor |y - x| \rfloor + 1$。为了防止位置为 x 的静止对象的不确定区域超出位置为 y 的点，我们需要在每 $f(x, y)$ 次查询中至少获取一次该对象的精确位置。我们令 b 表示满足 $\sum_{i=1}^{n} \frac{1}{\min(f(x_i, x_n + b),\, f(x_i, x_1 - b))} \leq 1$ 的最小正值。因此，~~考虑到可调度性的必要~~条件 [21]，对于任意值 $b' < b$，没有任何查询策略能够确保每个对象的不确定区域都保持在 $[x_1 - b', x_n + b']$ 内。

$\mathbb{R}^1$ 中的 1-中心的不确定区域大小，等于最大点的不确定区域大小与最小点的不确定区域大小的平均值。给定 $\mathbb{R}^1$中的一组点，最大值问题要求确定集合中最大点的位置，最小值问题要求确定集合中最小点的位置。无论采用何种查询策略，在未来某个时刻，要么最大点的不确定区域会包含 $[x_n, x_n + b]$，要么最小点的不确定区域会包含 $[x_1 - b, x_1]$。如果 $|x_n - x_1| \leq 1$，最大点和最小点的不确定区域的并集将始终包含 $[x_1, x_n]$。否则，这两个不确定区域的并集将始终与 $[x_1, x_1 + 1] \cup [x_n - 1, x_n]$相交，且交集的总大小至少为 1。因此，无论采用何种查询策略，在未来某个时刻，1-中心的不确定区域大小至少为 $\frac{b}{\min(|x_n - x_1|, 1)\ 2}$。对于最优算法，这是 1-中心不确定区域最大大小的下界。

现在我们将说明如何使用单位时间内四次查询，并实现相对于单位时间内仅进行一次查询的最优算法的 1-竞争性。首先，我们展示如何计算 b。我们知道 $0 \leq b \leq n$（因为轮询算法可以确保每个对象的不确定区域始终保持在 $[x_1 - n, x_n + n]$ 以内）。

此外，存在一个索引 $1 \le i \le n$，使得要么 $\lfloor x_i - x_1 + b \rfloor = x_i - x_1 + b$，要么 $\lfloor x_n + b - x_i \rfloor = x_n + b - x_i$（否则，存在一种查询策略，可以确保每个对象的不确定区域在某个 $b' < b$ 的情况下始终保持在 $[x_1 - b', x_n + b']$ 以内）。因此，b 有 $O(n^2)$ 种可能的取值，我们可以通过二分搜索找到 b。此外，我们证明 $f(x_i, x_n + b) \le 2f(x_i, x_n + \frac{b}{2})$。我们有

$$\begin{aligned} f(x_i, x_n + b) &= \lfloor x_n + b - x_i \rfloor + 1 \\ &\le \lfloor x_n + \frac{b}{2} - x_i \rfloor + \lfloor \frac{b}{2} \rfloor + 2 \\ &\le 2\lfloor x_n + \frac{b}{2} - x_i \rfloor + 2 \\ &= 2f(x_i, x_n + \frac{b}{2}). \end{aligned}$$

同理，我们有 $f(x_i, x_1 - b) \le 2f(x_i, x_1 - \frac{b}{2})$。因此，我们推得娱乐主管

$$\sum_{i=1}^{n} \frac{1}{\min(f(x_i, x_n + \frac{b}{2}), f(x_i, x_1 - \frac{b}{2}))} \le \sum_{i=1}^{n} \frac{2}{\min(f(x_i, x_n + b), f(x_i, x_1 - b))} \le 2.$$

因此，根据引理1，我们可以在单位时间内使用三次查询，将所有对象的不确定区域维持在 $[x_1 - \frac{b}{2}, x_n + \frac{b}{2}]$ 以内。我们让第四次查询在位置 x_1 的对象和位置 x_n 的对象之间反复切换。这样，$[x_1, x_n]$ 与最大值和最小值的不确定区域并集之间的交集总大小将至多为 $\min(|x_n - x_1|, 1)$。因此，使用该算法，在未来任意时刻，1-中心的不确定区域大小都将至多为 $b + \min(|x_n - x_1|, 1)$。因此，与单位时间内仅查询一个对象的最优算法相比，所提出的单位时间内查询四个对象的算法具有 1-竞争性。□

3.2 $\mathbb{R}^1$中的一般情形

对于 $\mathbb{R}^1$中1-中心问题的一般情形，我们证明，采用轮转调度算法查询对象，可以相对于最优算法达到最优竞争比。

定理 2。对于 $\mathbb{R}^1$ 中的 1-中心问题，轮转调度算法在未来任意时刻都能将不确定区域的最大尺寸保持在 $O(vn)$ 以内，并且相对于最优算法达到 $O(n)$ 的最优竞争比。

证明。轮询算法能够使每个对象的不确定区域大小保持在 $O(vn)$ 以内，因为它每进行 n 次查询，就会获取每个对象的精确位置一次。对于 $1 \le i \le n$，令 $[s_i, e_i]$ 表示第 i 个对象的不确定区域（s_i 是该不确定区域的起点，e_i 是终点）。在任意时刻，令 m 表示终点最高的对象的索引

（$e_m = \max_{1 \le i \le n} e_i$）。因此，$[s_m, e_m]$ 包含最大值的不确定区域。由此，最大值的不确定区域大小小于或等于 $|e_m - s_m|$，而其上界为 $O(vn)$。同理，在未来任意时刻，最小值的不确定区域大小均不超过 $O(vn)$。因此，采用轮询算法时，1-中心的不确定区域大小（等于最大值不确定区域大小与最小值不确定区域大小的平均值）永远不会超过 $O(vn)$。

对于最优算法，在第一次查询之后的任意时刻，在最后已知精确位置最大的对象和最后已知精确位置最小的对象之间，至少有一个对象不是最后一个被查询的对象；因此，该对象的不确定区域大小将为 $\Omega(v)$。因此，最大值或最小值对应的不确定区域大小至少有一个为 $\Omega(v)$。由此可知，在第一次查询之后的任意时刻，1-中心的不确定区域大小将为 $\Omega(v)$。考虑到轮询算法将不确定区域的最大大小保持在 $O(vn)$ 以内，因此，相对于最优算法，其竞争比为 $O(n)$。

我们证明，$O(n)$ 是相对于最优算法所能达到的最佳竞争比。考虑一个示例，其中 n 个对象最初均位于原点。其中一个对象以 v 的速度沿正方向移动，另一个对象以 v 的速度沿负方向移动。其余对象均保持静止。在此示例中，最优算法每进行两次查询，就能分别获取这两个运动对象的精确位置一次，并将 1-中心的不确定区域大小维持在 $O(v)$ 以内。然而，任何不知道未来对象轨迹的算法，都可能在最初的 $n-2$ 次查询中未能查询这两个运动对象；因此，1-中心的不确定区域的大小可能增大到 $\Omega(vn)$。即使允许算法在每单位时间内查询固定数量的对象（而非仅查询一个对象），这一结论仍然成立。因此，$O(n)$ 是所能达到的最佳竞争比。□

轮询算法将 1-中心的不确定区域的最大大小保持在 $O(vn)$ 内，最优算法也是如此。我们证明，即使对于最优算法，不确定区域的最大大小也可以是 $\Omega(vn)$。证明见附录 B。

命题 1. 对于最优算法，1-中心的不确定区域的最大大小可以为 $\Omega(vn)$ 。

现在，我们给出轮询算法在对象具有不同最大速度的 1-中心问题中的竞争比上界。证明见附录 C。

命题 2。对于 $\mathbb{R}^1$ 中对象具有不同最大速度的1-中心问题，轮询调度算法相对于最优算法的竞争比为 $O(v_M n / v_m)$，其中 v_M 是最高最大速度，v_m 是最低最大速度。

3.3 $\mathbb{R}^d$中的一般情形

对于 $\mathbb{R}^d$ （$d \geq 2$）中的 1-中心问题的一般情形，我们证明，最优算法的不确定区域最大尺寸可能没有上界。以下命题的证明受到 [6] 中定理 2 证明的启发，可见附录 D。

命题 3. 对于最优算法，$\mathbb{R}^d$ $(d \geq 2)$ 中1-中心的不确定区域的最大大小可能没有上界。

接下来，我们将证明，不可能找到一种相较于最优算法表现良好的算法。

定理 3。 对于任何不知道轨迹的算法， $\mathbb{R}^d$ $(d \geq 2)$ 中点的1-中心的不确定区域的最大尺寸，可能比最优算法所得结果大任意倍。

证明。我们给出一个示例，并证明：对于任何不知道对象轨迹的算法，都无法保证其相对于最优算法具有有界的竞争比。尽管我们在 $\mathbb{R}^2$这里解释该示例，但我们也可以在二维以上的空间中考虑相同的示例，其证明过程将非常相似。设 $\mathsf{a} = (0, G)$，$\mathsf{b} = (-2L, 0)$，$\mathsf{c} = (2L, 0)$，$\mathsf{u} = \mathsf{c} - \mathsf{a}$，且 $\mathsf{e} = \mathsf{c} + v(n-3)\hat{}\mathsf{u}$，其中 $n \geq 28$, L 可以任意大（远大于 n 和 v），并且 G 远大于 L，使得对于 $y \geq \frac{G}{3}$，我们有 $\sqrt{100L^2 + y^2} \leq y + v$。我们考虑一个包含 n 个对象的示例，其中一个对象（记为 A）始终位于 a，一个对象（记为 B）始终位于 b，一个对象（记为 E）从 e 出发，始终以 v 的最大速度沿 $\hat{}\mathsf{u}$ 方向移动，而另外 $n-3$ 个对象从 e 出发，以 v 的最大速度沿 $-\hat{}\mathsf{u}$ 方向移动，直到它们在 c 处停止。

对于上述示例，实际的 1-中心始终是 A 和 E 位置的中点。因此，实际 1-中心的 x 坐标始终大于 L。然而，任何不知道对象未来轨迹的算法，都可能在最初 $n-3$ 次查询中未能查询 E。因此，对于任何此类算法，在最初 $n-3$ 次查询之后，c 是最初位于 e 的每个 $n-2$ 对象的一个可能位置，a 是 A 的一个可能位置，而b 是 B 的一个可能位置。因此，1-中心可能位于直线 $x = 0$ 上。因此，对于任何不知道对象未来轨迹的算法，在第（$n-3$）次查询之后，1-中心的不确定区域大小可能会大于 L （其值可以任意大）。

现在，我们只需证明，对于上述示例，最优算法能够将 1-中心的不确定区域的最大尺寸控制在 n 和v 的某个函数范围内。我们考虑这样一种算法：每进行四次查询，就分别查询 A、 B和 E 一次；每进行 $4(n-3)$次查询，就查询每个其他对象一次。因此， A、 B和 E 的不确定区域尺寸永远不会超过 $6v$。此外，每个其他对象的不确定区域尺寸始终为 $O(vn)$。我们将证明，使用该算法，最大值

上述示例中，1-中心的不确定区域尺寸至多为 n 和v 的某个函数。这将证明，对于最优算法也同样成立，因为其效果至少不逊于上述算法。

现在暂且忽略 B，分析其他对象的 1-中心的不确定区域。对于这些对象，最小包围圆的半径至多为不确定区域中相距最远的两个点之间距离的一半。这是因为连接这两点的线段是一个包含所有对象不确定区域的圆的直径。因此，在算法通过上述方式进行 q 次查询（$q \geq 0$）后，最小包围圆的半径至多为 $r_q = \frac{1}{2}(3v + |\mathbf{e} - \mathbf{a}| + qv)$ $\mathbf{c}$ $\mathbf{a}$（）。考虑到在第 q 次查询后，1-中心与 A 的不确定区域中的某个点之间的距离至多为 r_q，1-中心与 $\mathbf{a}$ 之间的距离不超过 $r_q + 3v$。此外，在第 q 次查询后，1-中心与 E 的不确定区域中的某个点之间的距离至多为 r_q。因此，1-中心与 $\mathbf{e} + qv$ˆ$\mathbf{u}$ 之间的距离小于或等于 $r_q + 6v$。于是，在第 q 次查询后，1-中心与 $\mathbf{a}$ 之间的距离至少为 $|\mathbf{e} - \mathbf{a}| + qv - r_q - 6v = r_q - 9v$。

欧几里得空间中一组点的 1-中心位于这些点的凸包内（否则，存在一个超平面将 1-中心与这些点分隔开来，通过将 1-中心向该超平面移动，我们可以减小其到每个点的距离）。因此，1-中心的不确定区域始终位于这些对象的不确定区域的凸包内。由此可推知，在任意时刻，除 B 之外所有对象的 1-中心的不确定区域均位于这些对象的不确定区域的凸包内的一个条带状区域中（见图 2）。在任意时刻，该条带的长度为 $O(vn)$，因为每个对象的不确定区域的大小为 $O(vn)$，且这些对象的不确定区域的中心共线。此外，该条带的宽度至多为 $r_q + 3v - (r_q - 9v) = 12v$。因此，条带状区域的最大大小为 $O(vn)$。因此，使用上述算法，除 B 之外所有对象的 1-中心的不确定区域的最大大小为 $O(vn)$。

令$\mathbf{d}$ 表示条带状区域内某一点的位置。如果$\mathbf{d}$ 的 y 坐标位于 $[\frac{G}{3} + 3v, \frac{G}{2} - 4v]$ 内，那么$\mathbf{d}$ 与不确定区域 A 中任意一点之间的距离至少为 $\frac{G}{2} + v$。此外，在这种情况下，$\mathbf{d}$ 与不确定区域 $\overline{B}$ 中任意一点之间的距离至多为 $\frac{G}{2}$。进一步，容易看出，如果$\mathbf{d}$ 的 y 坐标小于 $\frac{G}{3} + 3v$，那么$\mathbf{d}$ 与 $\underline{B}$ 的不确定区域中任意一点之间的距离，小于$\mathbf{d}$ 与 A 的不确定区域中任意一点之间的距离。如果 $r_q - 9v$（$\mathbf{a}$ 与条带状区域内任意一点之间的最小距离）至少为 $\frac{G}{2} + 5v$，那么条带状区域内任意一点的 y 坐标至多为 $\frac{G}{2} - 4v$，这意味着，除 B 之外所有对象的 1-中心，与包括 B 在内的所有对象的 1-中心完全相同。考虑到 $n \geq 28$，我们有

$$r_q - 9v = \frac{1}{2}|\mathbf{c} - \mathbf{a}| + \frac{v}{2}(n + q) - 9v \geq \frac{G}{2} + \frac{v}{2}(28 + 0) - 9v = \frac{G}{2} + 5v.$$

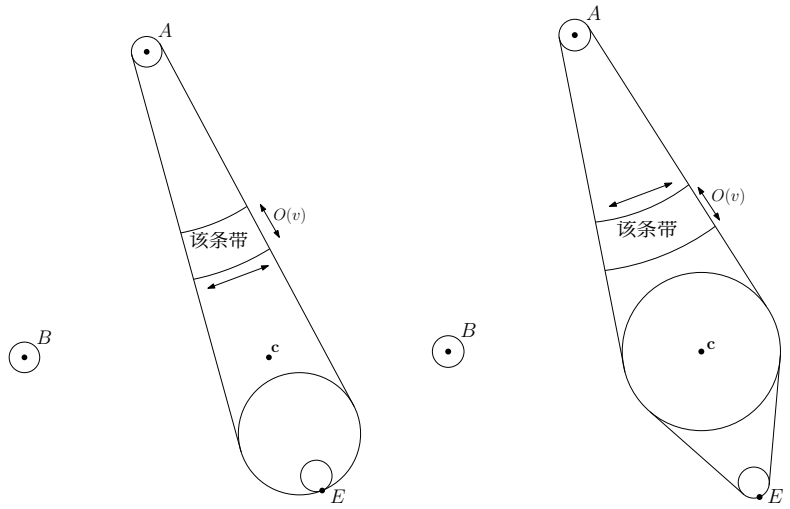

图 2：在定理 3 的证明中给出的示例和算法下，(a) 第 $(n-3)$ 次查询之前和 (b) 第 $(5n-15)$ 次查询之后，对象的不确定区域以及包含除 B 之外所有对象的 1-中心的条带状区域。

因此，使用上述算法，包含 B 在内的所有对象的 1-中心的不确定区域始终位于条带状区域内。因此，对于给定示例，最优算法将 1-中心不确定区域的最大尺寸保持在 $O(vn)$ 以内。□

4 质心问题

我们知道，在任意时刻，每个对象的不确定区域都是一个球，其中心位于该对象最后已知的精确位置。我们引入一个引理，用于求出质心或其特例——质心点——的不确定区域的大小。该引理的证明见附录 E。

引理 2. 给定 $\mathbb{R}^d$ $(d \geq 1)$ 中一组位置不确定的 n 加权对象，其质心的不确定区域是一个球，该球以这些对象的不确定区域中心的加权平均为球心。该球的半径等于这些对象的不确定区域半径的加权平均值。

引理 2 表明，质心的不确定区域大小等价于对象不确定区域大小的加权平均值。现在，对于质心问题，我们证明轮询算法相对于最优算法是 1-竞争的。以下定理的证明见附录 F。

定理 4. 对于 $\mathbb{R}^d$ $(d \geq 1)$ 中的质心问题，轮询调度算法的不确定区域最大大小为 $\Theta(vn)$ ，且轮询算法相对于最优算法是1-竞争的。

对于质心问题，轮询算法相对于最优算法的竞争比为 $\Omega(n)$。通过考虑一个示例即可轻松证明这一点，其中一个对象的权重远大于其他对象的权重。接下来，我们针对具有不同最大速度的对象的质心问题提出一种算法，并证明该算法能够保证相对于最优算法的竞争比为 $O(\log n)$。

定理 5. 对于 $\mathbb{R}^d$ $(d \geq 1)$ 中各对象具有不同最大速度的质心问题，存在一种算法，其竞争比为 $O(\log n)$ ，与最优算法相比。

证明。 我们按照使 $v_1 w_1 \geq \ldots \geq v_n w_n$ 成立的方式对对象进行排序。随后，我们将这些对象划分为 $\lfloor \log n \rfloor + 1$ 个组。对于 $1 \leq i \leq n$，我们将第 i 个对象分配到第 $(\lfloor \log i \rfloor + 1)$ 个组中。现在，我们来描述查询策略。我们以循环方式查询不同组中的对象。此外，在每个组内，我们也以循环方式选择要查询的对象。例如，假设我们有七个对象，将这些对象划分为三个组。我们将第一个对象分配到第一组，将第二个和第三个对象分配到第二组，其余对象分配到第三组。因此，我们的算法在查询对象时会反复遵循以下序列：1，2，4，1，3，5，1，2，6，1，3，7（序列中的各项表示我们的算法所查询的对象）。

上述算法每经过 $\lfloor \log n \rfloor + 1$ 次查询，便能获取包含第 i 个对象（$1 \leq i \leq n$）的组中一个对象的精确位置，且该组至多包含 $2^{\lfloor \log i \rfloor}$ 个对象。因此，每经过 $(\lfloor \log n \rfloor + 1)2^{\lfloor \log i \rfloor}$ 次查询，该算法至少能获取一次第 i 个对象的精确位置。因此，使用该算法，质心不确定区域的最大大小至多为 $\frac{1}{W} \sum_{i=1}^{n} 2((\lfloor \log n \rfloor + 1) 2^{\lfloor \log i \rfloor} - 1) v_i w_i$（根据引理-2），其中 $W = \sum_{i=1}^{n} w_i$ 是所有对象的总权重。

另一方面，为了使质心不确定区域的最大大小保持有限，最优算法必须在某个时刻查询第 n 个对象。此时，质心不确定区域的大小至少为 $\frac{1}{W} \sum_{i=1}^{n-1} 2 i v_i w_i$ （根据引理 2）。我们有

$$\begin{aligned}
\frac{3(\lfloor \log n \rfloor + 1)}{W} \sum_{i=1}^{n-1} 2 i v_i w_i &\geq \frac{\lfloor \log n \rfloor + 1}{W} \sum_{i=1}^{n} 2 i v_i w_i \\
&\geq \frac{1}{W} \sum_{i=1}^{n} 2((\lfloor \log n \rfloor + 1) i - 1) v_i w_i \\
&\geq \frac{1}{W} \sum_{i=1}^{n} 2((\lfloor \log n \rfloor + 1) 2^{\lfloor \log i \rfloor} - 1) v_i w_i.
\end{aligned}$$

因此，对于质心问题，上述算法相对于最优算法的竞争比为 $O(\log n)$。□

5 1-中位点问题

给定 $\mathbb{R}^1$ 中的一组 n 个点，如果 n 为奇数，则 1-中位点是第 $\lceil \frac{n}{2} \rceil$ 小的点；如果 n 为偶数，则是第 $\frac{n}{2}$ 小和第 $(\frac{n}{2}+1)$ 小的点的中点。

定理 6. 对于 $\mathbb{R}^1$ 中的1-中位数问题，轮转调度算法在未来任意时刻都能将不确定区域的最大尺寸控制在 $O(vn)$ 以内，并且相对于最优算法实现了最优竞争比 $O(n)$ 。

证明。（概要）设 s_k （e_k）为所有n 个对象的不确定区域的起始点（结束点）中第 k 小的点。尽管区域 $[s_k, e_k]$ 可能并非任何特定对象的不确定区域，但第 k 小的对象位于该区域内，其大小为 O（vn），因为轮转调度（Round-robin）保证每个对象的不确定区域的大小均不超过 O（vn）。定理的其余部分可由 $[s_k, e_k]$ 大小的下界 Ω（v），以及表明 O（n）是最优可能竞争比的示例得出。（参见附录 G）。□

轮转算法使 1-中位点的不确定区域最大尺寸保持在 $O(vn)$ 以内。我们证明，即使采用最优算法，不确定区域的最大尺寸也可能达到 $\Omega(vn)$。证明见附录 H。

命题 4. 对于 $\mathbb{R}^1$ 中的1-中位数问题，最优算法的不确定区域最大尺寸可能为 $\Omega(vn)$ 。

接下来，我们给出对象具有不同最大速度时，轮转算法针对 1-中位数问题的竞争比上界。证明见附录 I。

命题 5. 对于 $\mathbb{R}^1$ 中对象具有不同最大速度的1-中位数问题，轮询调度算法相对于最优算法的竞争比为 $O(v_M n/v_m)$ ，其中 v_M 为最高最大速度， v_m 为最低最大速度。

现在，我们考虑 $\mathbb{R}^d$ $(d \geq 2)$ 中的 1-中位数问题。以下命题的证明受到 [14] 中定理 5.1 的证明启发，详见附录 J。

命题 6. 对于 $\mathbb{R}^d$ $(d \geq 2)$ 中的 1-中位数问题，最优算法的不确定区域最大规模可能无界。

6 结论

我们得出结论，在最坏情况下，根据先前查询的答案以及对象感知到的位置来调整查询策略，并不能帮助本文讨论的任何中心点问题实现更优的竞争算法。

下面列出一些尚未解决的有趣问题：

– 对于 $\mathbb{R}^1$ 中的 1-中心问题的静态情形，是否存在一种算法，每单位时间查询少于四个对象，并且相对于每单位时间查询一个对象的最优算法具有 1-竞争性；或者是否存在一种算法，每单位时间查询一个对象，并实现竞争比为$o(n)$? – 对于 $\mathbb{R}^1$ 中的 1-中心（1-中位数）问题，当对象具有不同的最大速度时，我们能否找到一种竞争比为$o(v_M n/v_m)$的算法，其中 v_M 是最高最大速度， v_m 是最低最大速度? – 对于 $\mathbb{R}^d$ $(d \geq 2)$ 中的 1-中位数问题，是否存在一种能够保证相对于最优算法具有有界竞争比的算法?

参考文献

1. Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson 和 Justin Ward。通过原始-对偶算法，为 k-均值和欧几里得 k-中位数提供更好的保证。 *SIAM Journal on Computing* 49 4 17 97, 2019 , ( ):FOCS – 。

2. Mihai B ̆adoiu、Sariel Har-Peled 和 Piotr Indyk。通过核集实现近似聚类。载于 *Proceedings of the thirty-fourth annual ACM symposium on Theory of computing* 250 257, 2002 , 第 – 页。

3. Chanderjit Bajaj。几何优化问题的代数次数。 *Discrete & Computational Geometry* 3 2 177 191, 1988 , ( ): – 。 4. Amotz Bar-Noy 和 Richard E Ladner。广播系统中的窗口调度问题。 *SIAM Journal on Computing*, 32(4):1091–1113, 2003。 5. Amotz Bar-Noy、Richard E Ladner 和 Tami Tamir。作为装箱问题受限版本的窗口调度。 *ACM Transactions on Algorithms (TALG)*, 3 3 28 2007 ( ): –es, 。 6. Sergey Bereg、Binay Bhattacharya、David Kirkpatrick 和 Michael Segal。用于维护移动中心的竞争性算法。 *Mobile Networks and Applications* 11 2 177 186, 2006 , ( ): – 。 7. Richard Bruce、Michael Hoffmann、Danny Krizanc 和 Rajeev Raman。具有不确定性的几何计算的高效更新策略。 *Theory of Computing Systems* 38 4 411 423, 2005 , ( ): – 。 8. Mee Yee Chan 和 Francis Y. L. Chin。基于双整数约简的风车问题通用调度器。 *IEEE Transactions on Computers*, 41(06):755–768, 1992。 9. Moses Charikar、Sudipto Guha、 ́Eva Tardos 和 David B Shmoys。k-中位数问题的常数因子近似算法。载于 *Proceedings of the thirty-first annual ACM symposium on Theory of computing* 1 10, 1999 , 第 – 页。

10. George Chrystal。关于构造平面上包围给定 n 个点的最小圆的问题。 *Proceedings of the Edinburgh Mathematical Society*, 3 1885 30 33, 1885 ( ): – 。 11. Michael B Cohen、Yin Tat Lee、Gary Miller、Jakub Pachocki 和 Aaron Sidford。近线性时间求几何中位数。载于 *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing* 9 21, 2016 , 第 – 页。

12. Zvi Drezner 和 Horst W Hamacher。 *Facility location: applications and theory*。Springer Science & Business Media，2004年。

13. Petros Drineas、Alan Frieze、Ravi Kannan、Santosh Vempala 和 V Vinay。通过奇异值分解对大型图进行聚类。 *Machine learning*，56(1)：9–33，2004。14. Stephane Durocher。 *Geometric facility location under continuous motion*。博士论文，不列颠哥伦比亚大学，2006。15. Will Evans、David Kirkpatrick、Maarten L¨offler 和 Frank Staals。针对以不同速度移动的实体的潜在位置的层数最小化的查询策略。载于 *30th European Workshop on Computational Geometry*，2014。16. William Evans、David Kirkpatrick、Maarten L¨offler 和 Frank Staals。针对移动点的潜在位置的层数最小化的竞争性查询策略。载于 *Proceedings of the twenty-ninth annual symposium on Computational geometry*，第155–164页，2013。17. Tom´as Feder 和 Daniel Greene。近似聚类的最优算法。载于 *Proceedings of the twentieth annual ACM symposium on Theory of computing*，第434–444页，1988。18. Tom´as Feder、Rajeev Motwani、Rina Panigrahy、Chris Olston 和 Jennifer Widom。带不确定性的中位数计算。载于 *Proceedings of the thirty-second annual ACM symposium on Theory of computing* 602 607, 2000 ，第 – 页。

19. Peter C Fishburn 和 Jeffrey C Lagarias。风车调度：可实现的密度。 *Algorithmica*，34(1)：14–38，2002。20. Dorit S Hochbaum 和 David B Shmoys。k-中心问题的最佳可能启发式算法。 *Mathematics of operations research*，10(2)：180–184，1985。21. Robert Holte、Aloysius Mok、Louis Rosier、Igor Tulchinsky 和 Donald Varvel。风车：一个实时调度问题。载于 *Proceedings of the 22nd Hawaii International Conference of System Science* 693 702, 1989 ，第 – 页。

22. Simon Kahan。运动中的数据模型。载于 *Proceedings of the Twenty-third Annual ACM Symposium on Theory of computing* 265 277, 1991 ，第 – 页。

23. Simon H Kahan。 *Real-time processing of moving data*。博士论文，华盛顿大学，1991。24. Tapas Kanungo、David M Mount、Nathan S Netanyahu、Christine D Piatko、Ruth Silverman 和 Angela Y Wu。用于 k-均值聚类的局部搜索近似算法。载于 *Proceedings of the eighteenth annual symposium on Computational geometry* 10 18, 2002 ，第 – 页。

25. Sanjeev Khanna 和 Wang-Chiew Tan。带不确定性的函数计算。载于 *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* 171 182, 2001 ，第 – 页。

26. Yaakov Kupitz 和 Horst Martini。广义费马-托里拆利问题的几何方面。 *Bolyai Society Mathematical Studies*，6：55–129，1997。27. Stuart Lloyd。PCM 中的最小二乘量化。 *IEEE transactions on information theory* 28 2 129 137, 1982 ，( )：–。

28. Nimrod Megiddo 和 Kenneth J Supowit。若干常见几何位置问题的复杂性。 *SIAM journal on computing*，13(1)：182–196，1984。29. Simon Aloysius Suyadi。使用查询模型计算不精确输入的函数。硕士论文，不列颠哥伦比亚大学，2012。30. Emo Welzl。最小包围圆盘（球和椭球）。载于 *New results and new trends in computer science* 359 370 1991 ，第 – 页。

Springer，。31. E Alper Yildirim。最小包围球问题的两种算法。 *SIAM Journal on Optimization* 19 3 1368 1391, 2008 ，( )：–。

32. Da Wei Zheng。对移动实体进行查询调度，以确认其中许多实体远离某个区域。硕士论文，不列颠哥伦比亚大学，2020。

引理1的证明

证明。首先，我们证明，对于给定的正整数多重集 B，其中 $d(B) \leq 4$ 3，可以将 B 划分为两个子集 B_1 和 B_2，使得每个子集都具有风车调度。如果 B 包含一个等于 1 的元素，则将该元素分配给 B_1，并将其余元素分配给 B_2。否则，如果 B 在 $\{2,3\}$中至少包含两个元素，则将其中两个元素分配给 B_1，并将其余元素分配给 B_2。否则，如果 B 在 $\{4,5,6\}$中至少包含四个元素，则将其中四个元素分配给 B_1，并将其余元素分配给 B_2。在上述所有情况下，B_1 均可通过轮询算法进行调度，而 B_2也可以调度，因为 $d(B_2) = d(B) - d(B_1) \leq d(B) - \frac{2}{3} \leq \frac{2}{3}$。如果上述情况均不成立，则将 B 中属于 $\{2,3\}$ 的每个元素分配给 B_1，并将 B 中属于 $\{4,5,6\}$ 的每个元素分配给 B_2。每个子集的密度都将小于或等于 0.75。然后，当 $d(B_1) < 0.6$时，我们将 B 中尚未分配的元素逐一添加到B_1中。每个尚未分配的元素都大于或等于 7，因此其对 $d(B_1)$ 的增量小于 0.15。因此，如果 $d(B_1) < 0.6$，将一个尚未分配的元素添加到该子集中，不会使 B_1 的密度超过 0.75。如果我们能够在保持$d(B_1) < 0.6$的同时，将所有尚未分配的元素都添加到 B_1 中，那么 B_1 和 B_2 都可以进行调度。如果在任何时刻，$d(B_1)$变得大于或等于 0.6，我们就将其余尚未分配的元素添加到 B_2中。这样，我们便有 $d(B_1) \leq 0.75$ 和 $d(B_2) = d(B) - d(B_1) \leq d(B) - 0.6 \leq 0.75$；因此，两个子集都可以进行调度。

现在，我们证明，如果 $d(A) \leq 2$，则可以将 A 划分为三个可调度子集 A_1、A_2 和 A_3。如果 A 包含一个等于 1 的元素、两个或更多个属于 $\{2,3\}$ 的元素、四个或更多个属于 $\{4,5,6\}$ 的元素，或五个或更多个等于 7 的元素，则可以选择 A_1，使其可调度且其密度大于或等于 $_2\,{}_3$。这样，我们需要将 $A - A_1$ 划分为两个可调度子集。我们知道这是可能的，因为 $d(A - A_1) \leq \frac{4}{3}$。如果前述情况均不成立，则将 A 中属于 $\{2,3\}$ 的每个元素分配给 A_1，将 A 中属于 $\{4,5,6\}$ 的每个元素分配给 A_2，并将每个等于 7 的元素分配给 A_3。每个子集的密度都将小于或等于 0.75。然后，我们逐个将 A 中未分配的元素添加到当前密度最低的子集中。每个未分配的元素都大于或等于 8。由于 $d(A) \leq 2$，在添加值为 k 的未分配元素之前，密度最低的子集的密度小于或等于 $\frac{2-\frac{1}{k}}{3}$。因此，添加该元素后，该子集的密度将小于或等于 $\frac{2-\frac{1}{k}}{3} + \frac{1}{k} = \frac{2}{3}(1 + \frac{1}{k}) \leq \frac{2}{3}(\frac{1}{2} + \frac{1}{8}) = 0.75$。因此，我们可以将所有未分配的元素添加到这三个子集中，同时不使它们的密度超过 0.75；由此，三个子集都将是可调度的。□

B 命题 1 的证明

证明。考虑一个示例，其中 n 个静态对象位于原点。在至少进行了 $n-1$ 次查询之后的任意时刻，总会至少有一个对象，其精确位置在最近的

截至该时刻的 $n-1$ 次查询中，尚未被最优算法获取。该对象的不确定区域大小至少为 $2v(n-1)$。此外，最大值和最小值的不确定区域大小至少为 $v(n-1)$。因此，对于最优算法，1-中心的不确定区域最大大小为 $\Omega(vn)$。□

C 命题 2 的证明

证明。 轮询算法使 1-中心的不确定区域的最大大小保持在 $O(v_M n)$ 以内。另一方面，最优算法中 1-中心的不确定区域的最大大小为 $\Omega(v_m)$。因此，轮询算法用于 1-中心问题时的竞争比为 $O(v_M n/v_m)$。□

D 命题 3 的证明

证明。 考虑一个示例，其中有三个静态对象，分别位于 $(-2L,\ 0)$、$(2L,\ 0)$ 和 $(0,\ G)$，其中 L 可以任意大，且 G 远大于 L，使得连接 $(0,\ G)$ 与 $(2L,\ -v)$ 的线段是一个圆的直径，而该圆包含 $(-2L,\ 0)$。图3展示了这个示例。如果位于 $(2L,\ 0)$ 的对象不是第一个被查询的对象，那么经过第一次查询后，其不确定区域包含 $(2L,\ 0)$ 和 $(2L,\ -v)$。因此，第一次查询后各对象的不确定区域表明，1-中心可能位于直线 $x=0$ 或直线 $x=L$ 上。因此，1-中心的不确定区域大小至少为 L。类似地，如果位于 $(-2L,\ 0)$ 的对象不是第一个被查询的对象，那么 1-中心的不确定区域大小至少为 L。由此可知，无论哪个对象第一个被查询，1-中心的不确定区域大小都至少为 L，而这可以任意大。因此，对于最优算法，$\mathbb{R}^d$ 中 1-中心的不确定区域的最大大小可能没有上界。□

E 引理 2 的证明

证明。 对于 $1 \le i \le n$，令 $\mathbf{c}_i$ 和 r_i 分别表示第 i 个对象的不确定区域的中心和半径。此外，令 w_i 表示第 i 个对象的权重，$W=\sum_{i=1}^{n} w_i$ 表示所有对象的总权重。我们希望证明，质心的不确定区域是一个以$\mathbf{C}=\frac{1}{W}\sum_{i=1}^{n} w_i\mathbf{c}_i$ 为中心、半径为 $R=\frac{1}{W}\sum_{i=1}^{n} w_i r_i$ 的球。

令 $\mathbf{q}_1,\ldots,\mathbf{q}_n$ 表示 n 个任意点的位置，使得对于 $1\le i\le n$，第 i 个点位于第 i 个对象的不确定区域内。我们证明，这些点的加权平均位于以 $\mathbf{C}$ 为中心、半径为 R 的球内。我们有

$$|\frac{1}{W}\sum_{i=1}^{n} w_i\mathbf{q}_i-\mathbf{C}| = |\frac{1}{W}\sum_{i=1}^{n} w_i(\mathbf{q}_i-\mathbf{c}_i)| \le \frac{1}{W}\sum_{i=1}^{n} w_i|\mathbf{q}_i-\mathbf{c}_i| \le \frac{1}{W}\sum_{i=1}^{n} w_i r_i = R.$$

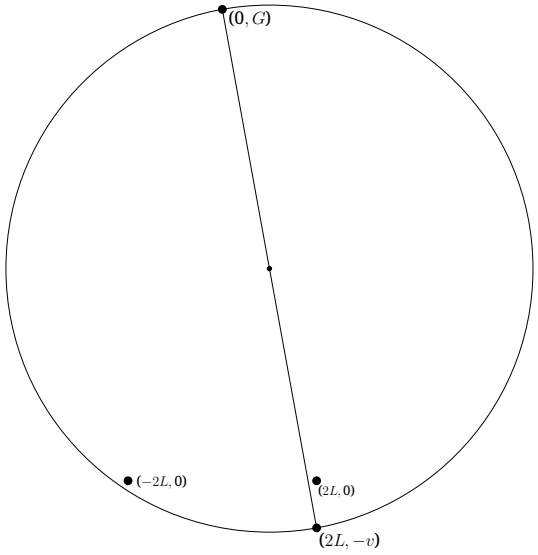

图 3：命题 3 证明中所给示例的示意图。

现在，令 $\mathbf{U}$ 表示以 $\mathbf{C}$ 为中心、半径为 R 的球内任意一点的位置。我们证明，$\mathbf{U}$ 是 n 个点 $\mathbf{u}_1, ..., \mathbf{u}_n$ 的加权平均，使得对于 $1 \leq i \leq n$，$\mathbf{u}_i = \mathbf{c}_i + \frac{r_i}{R}(\mathbf{U} - \mathbf{C})$，因此，第 i 个点位于第 i 个对象的不确定区域内。我们有

$$\frac{1}{W}\sum_{i=1}^{n} w_i \mathbf{u}_i = \frac{1}{W}\sum_{i=1}^{n} w_i(\mathbf{c}_i + \frac{r_i}{R}(\mathbf{U} - \mathbf{C})) = \mathbf{C} + \left(\frac{1}{WR}\sum_{i=1}^{n} w_i r_i\right)(\mathbf{U} - \mathbf{C}) = \mathbf{U}.$$

因此，质心的不确定区域是一个以$\mathbf{C}$为中心、半径为 R 的球。□

F 定理 4 的证明

证明。 在轮询算法执行 n 次查询后的任意时刻，倒数第 i 个被查询的对象（$2 \leq i \leq n$）的不确定区域的大小为 $2v(i-1)$。因此，根据引理 2，质心的不确定区域的大小为 $\frac{1}{n}\sum_{i=1}^{n} 2v(i-1) = v(n-1)$。由此可知，轮询算法下质心不确定区域的最大大小为 $\Theta(vn)$。

对于最优算法，在执行至少 n 次查询后的任意时刻，所有对象的不确定区域中第 i 小的不确定区域（$1 \leq i \leq n$）的大小至少为 $2v$（$i-1$）。因此，根据引理 2，质心的不确定区域大小至少为 v（$n-1$）。由此可知，Round-robin 算法相对于最优算法是 1-竞争的。此外，最优算法下质心不确定区域的最大大小为 Θ（vn）。□

G 定理 6 的证明

引理 3。 无论采用何种查询策略， k 最小（$2 \le k \le n-1$）的移动实体在所有时刻的不确定区域最大尺寸均为 $\Omega(v)$。

证明。 我们通过反证法进行证明。假设存在一个示例，使用算法 A 时，第 k 小的对象的不确定区域的大小永远不会超过 $\frac{v}{2}$。对于 $1 \le i \le n$ 和 $q \ge 0$，令 $s_{i,q}$ ($e_{i,q}$) 等于在 A 执行恰好 q 次查询后，所有对象的不确定区域的起始点（结束点）中第 i 小的点。我们称 $s_{i,q}$ ($e_{i,q}$) 界定了对象 b，如果它等于对象 b 的不确定区域的起始点（结束点）。 $[s_{i,q}, e_{i,q}]$ 表示经过 q 次查询后，第 i 小对象的不确定区域。根据我们的假设，对于所有 q，均有 $e_{k,q} - s_{k,q} \le \frac{v}{2}$。

我们考虑这样的情况：对于某个 $q \ge 0$，我们有 $e_{k,q} - e_{k-1,q} \le \frac{v}{2}$。如果$s_{k+1,q} \le e_{k,q}$，那么在第 $(q+1)$ 次查询之后，我们将至少有 k 个起点小于或等于 $e_{k,q} - v$。此外，我们还将有 $e_{k,q+1} \ge e_{k-1,q} + v$。因此，我们将有 $s_{k,q+1} \le e_{k,q} - v \le e_{k-1,q} - \frac{v}{2} \le e_{k,q+1} - \frac{3v}{2}$。然而，这与使用算法 A 时，第 k 小元素的不确定区域大小永远不会超过 $\frac{v}{2}$ 的假设相矛盾。因此，我们必须有 $s_{k+1,q} > e_{k,q}$。如果第 $(q+1)$ 次查询获取了在经过 q 次查询后被 $e_{k,q}$ 所界定的对象的精确位置，那么该对象的新位置必须大于或等于 $e_{k-1,q} + \frac{v}{2}$，从而使我们有 $e_{k,q+1} - s_{k,q+1} \le \frac{v}{2}$。在这种情况下，我们有$e_{k,q+1} \ge e_{k,q} + \frac{v}{2}$以及 $e_{k,q+1} - e_{k-1,q+1} \le \frac{v}{2}$。如果第 $(q+1)$ 次查询获取的是任何其他对象的精确位置，那么在第 $(q+1)$ 次查询之前，该对象的不确定区域的起始点必须小于或等于$s_{k,q}$（这是满足 $e_{k,q+1} - s_{k,q+1} \le \frac{v}{2}$ 的必要条件）。因此，在这种情况下，$e_{k,q+1} = e_{k,q} + v$。此外，被查询对象的新位置必须大于或等于 $e_{k,q} + \frac{v}{2}$，从而使我们有 $e_{k,q+1} - s_{k,q+1} \le \frac{v}{2}$。因此，我们有 $e_{k,q+1} - e_{k-1,q+1} \le \frac{v}{2}$。由此可知，在这两种情况下，我们都有$e_{k,q+1} - e_{k-1,q+1} \le \frac{v}{2}$, $e_{k,q+1} \ge e_{k,q} + \frac{v}{2}$ 以及 $s_{k+1,q+1} < s_{k+1,q}$。我们可以推断，对于某个 $q' > q$，我们将有 $e_{k,q'} - e_{k-1,q'} \le \frac{v}{2}$ 以及 $s_{k+1,q'} \le e_{k,q'}$。因此，第 k 小元素的不确定区域大小将超过 $\frac{v}{2}$。因此，如果对于某个 $q \ge 0$，我们有 $e_{k,q} - e_{k-1,q} \le \frac{v}{2}$，那么第 k 小元素的不确定区域大小将超过 $\frac{v}{2}$，这与关于算法 A 的假设相矛盾。类似地，我们可以证明，如果对于某个 $q \ge 0$，我们有 $s_{k+1,q} - s_{k,q} \le \frac{v}{2}$，那么在某个时刻，第 k 小元素的不确定区域大小将超过 $\frac{v}{2}$。

现在我们考虑这样的情况：对于某些 $q \ge 0$、 $s_{k,q}$ 和 $e_{k,q}$，它们界定了两个不同的对象。如果由 $e_{k,q}$ 界定的对象不是最后一个被查询的对象，那么至多有 $k-2$ 个终止点小于 $s_{k,q}$；因此，$s_{k,q} \le e_{k-1,q}$。由此，我们得到 $e_{k,q} - e_{k-1,q} \le e_{k,q} - s_{k,q} \le \frac{v}{2}$。如果由 $e_{k,q}$ 界定的对象是最后一个被查询的对象，那么我们在 $e_{k,q}$ 处有一个起始点。在这种情况下，我们有 $s_{k+1,q} - s_{k,q} \le e_{k,q} - s_{k,q} \le \frac{v}{2}$。由此可推得，要么 $e_{k,q} - e_{k-1,q} \le \frac{v}{2}$，要么 $s_{k+1,q} - s_{k,q} \le \frac{v}{2}$。因此，在某个时刻，第 k 小元素的不确定区域的大小将超过 $\frac{v}{2}$，这与关于算法 A 的假设相矛盾。

因此，对于每个 $q \ge 0$、 $s_{k,q}$ 和 $e_{k,q}$ 必须界定同一个对象。此外，我们必须有 $e_{k,q} - e_{k-1,q} > \frac{v}{2}$ 和 $s_{k+1,q} = s_{k,q} > \frac{v}{2}$。由于我们假设

由于 $e_{k,q} - s_{k,q} \leq \frac{v}{2}$，经过 q 次查询后，我们必须确定对象的精确位置位于 $s_{k,q}$ 和 $e_{k,q}$ 所界定的范围内。这意味着 $e_{k,q} - s_{k,q} = 0$，并且如果 $q > 0$，第 q 次查询必须已经确定了对象的精确位置，该位置位于 $s_{k,q}$ 和 $e_{k,q}$ 所界定的范围内。如果经过 q 次查询（$q \geq 0$）后，我们确定了对象 t 的精确位置，就知道 t 位于第 k 小的位置；而且经过第（$q+1$）次查询后，无论查询哪个对象，$e_{k,q} - e_{k-1,q} > \frac{v}{2}$ 和 $s_{k+1,q} - s_{k,q} > \frac{v}{2}$，$t$ 都有可能位于第 k 小的位置。因此，不存在其他对象 t'，使得经过 $q+1$ 次查询后我们能确定其精确位置，并且确定它位于第 k 小的位置。因此，如果算法 A 在任何时候查询了一个初始位置并非第 k 小的对象，那么第 k 小对象的不确定区域大小最终将超过 $\frac{v}{2}$。另一方面，如果算法 A 始终查询同一个对象 t，且该对象的初始位置位于第 k 小，那么其他对象的不确定区域将不断扩大。在这种情况下，考虑到 t 的初始位置位于第 k 小（$2 \leq k \leq n-1$），至少一个其他对象的不确定区域在未来某个时刻会包含 t 的位置。发生这种情况时，t 的位置不一定是第 k 小。因此，第 k 小对象的不确定区域大小在某个时刻将超过 $\frac{v}{2}$。

在所有情况下，我们都证明了使用算法 A 时，第 k 小值的不确定区域大小将超过 $\frac{v}{2}$。这与我们对算法 A 的假设相矛盾。因此，对于每种算法，第 k 小值的不确定区域最大大小都大于 $\frac{v}{2}$。因此，当 $2 \leq k \leq n-1$ 时，无论采用何种查询策略，第 k 小值的不确定区域最大大小均为 $\Omega(v)$。□

定理 6. 对于 $\mathbb{R}^1$ 中的 1-中值问题，轮转调度算法能够在未来任意时刻将不确定区域的最大尺寸保持在 $O(vn)$ 以内，并且相对于最优算法实现了 $O(n)$ 的最优竞争比。

证明。当 n 为奇数时，1-中位数问题与第 $\lceil \frac{n}{2} \rceil$ 小问题相同。此外，当 n 为偶数时，我们将 1-中位点视为第 $\frac{n}{2}$ 小点与第 $(\frac{n}{2}+1)$ 小点的中点。在这种情况下，1-中位点的不确定区域大小等于第 $\frac{n}{2}$ 小点的不确定区域大小与第 $(\frac{n}{2}+1)$ 小点的不确定区域大小的平均值。

轮询算法会将每个对象的不确定区域大小保持在 $O(vn)$ 以内，因为它每经过 n 次查询，就会获取每个对象的精确位置一次。对于 $1 \leq k \leq n$，令 s_k 为所有对象的不确定区域起点中第 k 小的起点。同样，令 e_k 为所有对象的不确定区域终点中第 k 小的终点。因此，$[s_k, e_k]$ 是第 k 小对象的不确定区域。根据轮询算法，我们知道 $e_k - s_k$ 始终为 $O(vn)$，因为该算法会将每个对象的不确定区域大小保持在 $O(vn)$ 以内。因此，第 k 小对象的不确定区域大小为

在未来任意时刻都不超过 $O(vn)$。因此，使用轮询算法时，1-中位点的不确定区域大小永远不会超过 $O(vn)$。

如果 $n = 2$，那么在首次查询之后的任意时刻，其中一个对象的不确定区域大小将为 $\Omega(v)$。因此，1-中位点的不确定区域的最大大小将为 $\Omega(v)$。如果 $n \geq 3$，根据引理 3，无论采用何种查询策略，第 k 小的 $(2 \leq k \leq n-1)$ 的不确定区域的最大大小均为 $\Omega(v)$。因此，对于最优算法，在未来的某个时刻，1-中位点的不确定区域大小将达到 $\Omega(v)$。考虑到轮询算法将不确定区域的最大大小保持在 $O(vn)$ 以内，因此，相对于最优算法，其竞争比为 $O(n)$。

现在，我们证明 $O(n)$ 是相对于最优算法而言所能达到的最佳竞争比。我们考虑以下几种情况：

– 如果 n 是奇数，我们考虑这样一个示例：$\lceil \frac{n}{2} \rceil$ 个对象最初位于原点，$\lfloor \frac{n}{2} \rfloor$ 个对象最初位于 $10vn$。其中一个最初位于原点的对象以 v 的速度朝 $3vn$ 移动，然后停在那里。其余对象保持不动。考虑一种算法，该算法每隔两次查询就获取一次移动对象的精确位置（即使该对象在 $3vn$处停止后也如此），并按循环顺序将其余查询分配给静态对象。该算法将第 $\lceil \frac{n}{2} \rceil$ 小对象的不确定区域大小保持在 $O(v)$ 内。

– 如果 n 为偶数，我们考虑一个示例：$\frac{n}{2}$ 个对象最初位于原点，$\frac{n}{2}$ 个对象最初位于 $10vn$。最初位于原点的对象中，有一个以 v 的速度向 $3vn$ 移动，然后停在那里。此外，最初位于 $10vn$ 的对象中，有一个以 v 的速度向 $7vn$ 移动，然后停在那里。其余对象不移动。我们考虑这样一种算法：每三次查询获取这两个移动对象中每个对象的精确位置一次（即使对象停止移动后也是如此），并按循环顺序将其余查询分配给静态对象。该算法将 $\frac{n}{2}$ 个最小对象和（$\frac{n}{2} + 1$）个最小对象的不确定区域大小维持在 $O(v)$ 以内。

因此，对于所给出的两种情况，最优算法都能将 1-中位点的不确定区域的最大大小保持在 $O(v)$ 以内。然而，任何不了解对象轨迹的算法（例如轮询算法）都可能在前 $\lceil \frac{n}{2} \rceil - 1$ 次尝试中无法查询移动对象；因此，在前 $\lceil \frac{n}{2} \rceil - 1$ 次查询后，第 $\lceil \frac{n}{2} \rceil$ 小的不确定区域的大小可能达到 $\Omega(vn)$。因此，1-中位点的不确定区域的大小可能达到 $\Omega(vn)$。即使允许算法在每个单位时间内查询固定数量的对象（而非一个），这一结论仍然成立。因此，$O(n)$ 是可能达到的最佳竞争比。□

H 命题 4 的证明

证明。考虑一个示例，其中有 n 个静态对象位于原点。在经过至少 $\lfloor \frac{n}{2} \rfloor$ 次查询后的任意时刻，始终至少有 $\lceil \frac{n}{2} \rceil$ 个对象，其精确位置在该时刻之前的最近 $\lfloor \frac{n}{2} \rfloor$ 次查询中尚未被最优算法获取。这些 $\lceil \frac{n}{2} \rceil$ 个对象的不确定区域的终点至少为 $v\lfloor \frac{n}{2} \rfloor$。因此，在所有对象的不确定区域终点中，第 $\lfloor \frac{n}{2} \rfloor + 1$ 小的终点至少为 $v\lfloor \frac{n}{2} \rfloor$。另一方面，所有对象的不确定区域的起点始终小于或等于零。因此，在前 $\lfloor \frac{n}{2} \rfloor$ 次查询后的任意时刻，第 $\lfloor \frac{n}{2} \rfloor + 1$ 小的不确定区域大小为 $\Omega(vn)$。因此，对于最优算法，1-中位点的不确定区域的最大大小为 $\Omega(vn)$。□

I 命题 5 的证明

证明。 轮询算法将 1-中位点的不确定区域的最大尺寸保持在 $O(v_M n)$。另一方面，对于最优算法，1-中位点的不确定区域的最大尺寸为 $\Omega(v_m)$。因此，轮询算法在 1-中位数问题上的竞争比为 $O(v_M n/v_m)$。□

J 命题 6 的证明

证明。 考虑四个静态对象，分别位于 (0,0)、$(0, v)$、$(L,0)$ 和 (L, v)，其中 L 任意大。无论首先查询哪个对象，第一次查询后对象的不确定区域都表明，可能有两个对象在直线 $x = 0$ 上具有完全相同的位置，而另外两个对象在直线 $x = L$上具有不同的位置。在这种情况下，1-中位点将位于直线 $x = 0$上。此外，第一次查询后对象的不确定区域还表明，可能有两个对象在直线 $x = 0$ 上具有不同的位置，而另外两个对象在直线 $x = L$上具有完全相同的位置。在这种情况下，1-中位点将位于直线 $x = L$上。因此，无论首先查询哪个对象，1-中位点的不确定区域大小至少为 L，而该值任意大。因此，对于最优算法，$\mathbb{R}^d$ 中1-中位点的不确定区域最大大小可能是无界的。□