arXiv:2305.06067v1 [cs.DC] 10 May 2023

平面中的卵石引导寻宝*

Adri Bhattacharya[1[0000−0003−1517−8779]]、Barun Gorain[2]和 Partha Sarathi Mandal[1[0000−0002−8632−5767]]

[1] 印度古瓦哈提理工学院，印度{a.bhattacharya, psm}@iitg.ac.in[2] 印度比莱理工学院，印度 barun@iitbhilai.ac.in

摘要。 我们研究了在卵石引导下，由移动代理在欧几里得平面中进行寻宝的问题。代理的初始位置和宝藏的位置均被建模为欧几里得平面中的特殊点。宝藏与代理初始位置之间的距离至多为$D > 0$ 。代理拥有精确指南针，但其移动速度由对手控制。因此，代理无法测量在给定时间内行进的距离。代理只有到达宝藏的确切位置时才能找到宝藏。寻宝的代价定义为代理找到宝藏之前所行进的总距离。代理事先不知道宝藏的位置，也不知道 D的值。预言机知道宝藏的位置和代理的初始位置，并放置若干卵石来引导代理前往宝藏。一旦决定沿某个指定的角方向移动，代理只有在遇到卵石或特殊点时，才能决定改变移动方向。

本文提出以下核心问题：
“给定 $k \geq 0$，如果预言机放置的卵石不超过 k 个，最廉价的寻宝算法是什么？”
我们证明，对于 $k = 1$，不存在任何能够以有限代价找到宝藏的寻宝算法。我们证明，对于 $k = 2$，存在一种代价为 $O(D)$ 的算法。对于 $k > 8$，我们设计了一种算法，使用 k 个卵石，以代价 $O(k^2) + D(\sin\theta' + \cos\theta')$ 找到宝藏，其中 $\theta' = \frac{\pi}{2^{k-8}}$。第二个结果表明，对于足够大的 D，存在一种代价可任意接近 D 的算法。

关键词寻宝、移动代理、卵石、欧几里得平面、确定性算法。

1 引言

寻宝问题是指移动代理在未知环境中寻找静止目标的任务。未知环境可以建模为

本文作者感谢 Yoann Dieudonné 博士提供的宝贵意见和建议。

网络或平面。移动代理最初被放置在未知环境中的某个点上，必须找到一个静止目标，称为宝藏。目标或宝藏可能是在洞穴中迷路的矿工。洞穴可能不适合人类进入搜寻失踪的矿工，也可能被有毒水域淹没，因此必须尽快找到此人。在计算机科学应用中，软件代理必须访问由局域网连接的计算机，以找到受恶意软件影响的计算机。

本文研究了欧几里得平面中的寻宝问题，采用一种非常弱的场景假设，即移动代理所掌握的知识和所具有的控制能力都非常有限。具体而言，智能体事先并不知道宝藏的位置，也不知道自己与宝藏之间的距离。此外，智能体无法控制自身的移动速度，并假设对手能够完全控制智能体的速度。在实际应用中，对于网络中的软件代理而言，智能体的移动速度取决于多种因素，例如网络拥塞情况。对于硬件移动机器人而言，其速度取决于许多机械特性以及环境因素。智能体配备了一个精确指南针，能够帮助智能体旋转并沿指定方向移动。智能体最初被放置在平面上的一个点 P。宝藏 T 与 P 之间的距离最多为 $D > 0$（智能体对此距离并不知情）。只有当智能体到达宝藏的精确位置时，才能找到宝藏。智能体的初始位置被视为一个特殊点，并且每当智能体访问 P 时，都能够检测到该点。

由于无法控制自身的移动速度，智能体一旦决定沿某个角度移动，就必须知道何时停止移动。否则，对手可以任意提高速度，导致智能体最终移动任意大的距离。为了让智能体能够在一定程度上控制自己的移动，预言机在知晓宝藏位置和智能体初始位置的情况下，会在平面上放置一些静止的鹅卵石。我们假设预言机放置鹅卵石时受到以下限制：任意两颗鹅卵石之间必须相隔一个固定距离，也就是说，不能将两颗鹅卵石放得任意近 [3].. 为简单起见，我们假设任意两颗鹅卵石之间至少相隔 1 个单位的距离。智能体只有到达预言机放置鹅卵石的位置时，才能检测到鹅卵石的存在。

这些卵石的摆放有助于智能体控制其移动，并排除任意移动极大距离的可能性。智能体从平面上的某个位置出发，沿特定方向移动；一旦在移动路径上遇到卵石，就会停下或改变方向。因此，智能体的移动算法会指示其沿特定角度 α 移动，直到遇到特殊点（即初始位置 P 或宝藏的位置 T）或碰到卵石。

[3] 如果智能体的感知能力较弱，这一限制是必要的，因为智能体可能无法区分彼此距离很近的两颗鹅卵石。

形式化地说，对于给定的正整数 $k \geq 0$，预言机是一个函数 $f_k : (E \times E) \to E^k$，其中 E 是欧几里得平面上所有点的集合。该函数以两个点作为输入，第一个是智能体的初始位置，第二个是宝藏的位置，并输出平面上的 k 个点，表示在这些 k 点各放置一颗卵石。

本文研究的核心问题是："对于给定的 $k \geq 0$，如果平面上放置的卵石不超过 k 个，寻宝的最低代价是多少？"

1.1 贡献

本文的贡献总结如下。

– 对于 $k = 1$ 个卵石，我们证明了无法设计出一种能够以有限代价找到宝藏的寻宝算法。– 对于 $k = 2$ 个卵石，我们提出了一种算法，其寻宝代价不超过 $4.5D$，其中 D 是智能体初始位置与宝藏之间的距离。– 对于 $k > 8$，我们设计了一种使用 k 个卵石、代价为 $O(k^2) + D(\sin\theta' + \cos\theta')$ 的寻宝算法，其中 $\theta' = \frac{\pi}{2^{k-8}}$。当 D 和 $k \in o(\sqrt{D})$ 的值足够大时，该算法的代价可以任意接近 D；若智能体已知宝藏的位置，D 即为最优解的代价。

1.2 Related Work

移动代理搜索静止目标的任务已在文献中针对各种场景得到了严格研究。底层环境或拓扑结构可以是离散域，也可以是连续域，即图或平面。搜索策略可以是确定性的，也可以是随机化的。Alpern 等人 [2] 的著作讨论了基于搜索静止目标的随机化算法，以及会合问题；在会合问题中，目标和智能体都是动态的，并且双方相互协作以实现会合。Miller 等人[19],和 Ta-Shma 等人[22] 的论文探讨了图中会合问题与寻宝问题之间的相关性。

在直线上搜索静止目标的问题最初由 Beck 等人提出 [3]。他们给出了最优竞争比 9。此后，Demaine 等人 [7] 修改了该问题，规定搜索者每次改变方向都要付出代价。在 [17], 中，作者从搜索博弈的角度综述了搜索问题，其中目标可以是静止的，也可以是移动的。搜索域可以是图、有界域或无界集合。Fricke 等人[15], 将该搜索问题推广到平面上的多个搜索者。

如今，带建议的算法这一范式主要已在网络中得到引入。这些提示信息提升了问题的效率，正如[1,4,5,6,8,10,11,12,13,14,16,18,20]中所讨论的那样。在这一范式下，作者 [13,14] 主要研究了求解问题所需的最少提示信息量

。在 [9,11], 中，研究了带提示问题的在线版本。作者 [5], 考虑了寻宝问题，其中给出了一种最优成本算法，智能体在每次移动后都会获得提示。Pelc 等人 [21], 探讨了在 $O(L)$时间内求解几何地形中的寻宝问题所需的信息量，其中 L 是宝藏与初始点之间的最短路径。Bouchard 等人 [6], 研究了不同类型的初始知识如何影响树网络中的寻宝代价。

与本文最接近的两篇论文是 [18,20]。Pelc 等人 [20], 提出了在平面中解决寻宝问题时代价与信息之间的权衡。他们针对不同范围的视野半径，给出了最优和近乎最优的结果。Gorain 等人 [18], 在带石子的图中提出了一种最优寻宝算法，将提示信息称为 advice。在 [4], 中，作者研究了无向端口标记图中卵石数量与寻宝算法代价之间的权衡。

组织结构：本文组织如下。第 2 节简要介绍放置一定数量的卵石时寻宝问题的可行性。第 3 节分为三个小节：第 3.1 小节概述算法的基本思路，第 3.2 小节介绍卵石放置策略，第 3.3 小节讨论寻宝算法。第 4 节讨论正确性和复杂度。最后，第 5 节阐述可能的后续工作并总结全文。

2 Feasibility of Treasure hunt

在本节中，我们将分别讨论预言机放置一颗和两颗卵石时，寻宝问题的可行性。

定理 1. 不可能设计一种至多使用一颗卵石、且能以有限成本找到宝藏的寻宝算法。

证明。 智能体最初位于 P，而预言机将卵石放置在平面上的某个位置。由于智能体没有关于宝藏位置的任何先验信息，对手可以将宝藏放置在平面上的任意位置。智能体唯一可能的初始指令是从 P 沿某个角度移动。智能体在移动过程中必须遇到一颗卵石，否则，由于它无法感知距离，就会沿着该方向无限远地继续移动。遇到卵石后，存在三种可能性：它可以返回 P，并从 P 沿某个角度移动；也可以返回 P，沿着智能体先前经过的相同路径移动至卵石处；或者从卵石所在位置本身沿某个角度移动。对手可以将宝藏放置在智能体可能经过的路径之外的某个位置。因此，不可能以有限代价找到宝藏。

□

在本部分中，我们讨论当预言机放置两颗卵石时，智能体朝向宝藏移动的卵石放置策略及相应的遍历过程。

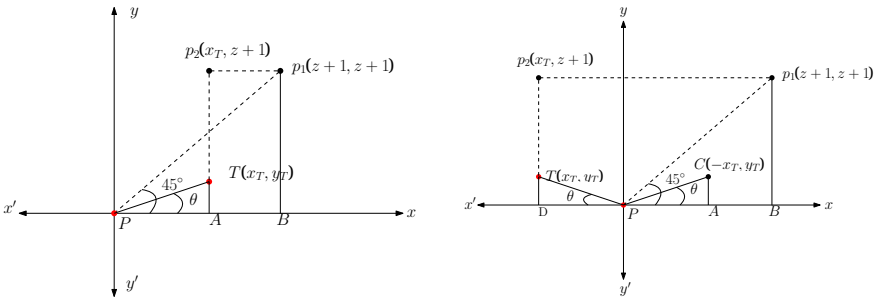
鹅卵石放置：根据宝藏位置，按如下方式放置两颗卵石。设宝藏 T的坐标为 (x_T, y_T)。如果x_T或 y_T中任意一个为正数，则在 $(z+1,\ z+1)$ 处放置一颗卵石，其中 $z = \max\{|x_T|, |y_T|\}$。在 $(x_T,\ z+1)$ 处放置另一颗卵石。否则，如果 x_T和 y_T均为负数，则在 $(1,\ 1)$ 处放置一颗卵石，并在 $(x_T,\ 1)$ 处放置另一颗卵石。

智能体的寻宝过程：智能体最初位于 P，沿着与正 x 轴成 $\frac{\pi}{4}$ 角的方向移动，直到遇到宝藏或卵石（即 p_1）。如果遇到卵石，则智能体从该位置沿着 $\pi - \frac{\pi}{4}$ 角的方向移动，直到遇到宝藏或到达卵石处（即 p_2）。如果遇到卵石（即从 p_2 出发），智能体将进一步沿着 $\frac{\pi}{2}$ 角的方向移动，直到到达宝藏 T。

定理 2. 智能体使用上述算法找到宝藏，代价为 $O(D)$ 。

证明。根据所提出的算法，找到宝藏的代价为路径 $Pp_1 + p_1p_2 + p_2T$（参见图 1），其中 p_1 和 p_2 分别是第一颗和第二颗卵石的位置。令 f_i: $\theta \longrightarrow \mathbb{R}$，其中 $i = 1, \cdots, 5$，是以下各情形对应的代价函数集合，我们分析如下:

1: 如果宝藏位于第一象限，则令 A 和 B 分别为从 T和 p_1作垂线所得的垂足。令 $\angle TPA = \theta$（参见图 1(a)）。因此，$PA = D\cos\theta$和 $AT = D\sin\theta$。现在分以下情形讨论:



(a) 第一象限中的宝藏　　(b) 第二象限中的宝藏

图 1: 宝藏位于平面上半部分时智能体的移动情况

1(a): 当 $x_T \geq y_T$ 时，卵石 p_1和 p_2分别放置在 (x_T+1, x_T+1) 和 (x_T, x_T+1) 处。因此，$PB = D\cos\theta+1$和 $PB = Bp_1$（因为

Δp_1PB是等腰三角形），这意味着 $Pp_1 = \sqrt{2}(D\cos\theta + 1)$。此外，在这种情形下，$p_1p_2 = 1$和 $p_2T = p_2A - TA = D\cos\theta + 1 - D\sin\theta$。因此，总成本为：$\sqrt{2}(D\cos\theta+1)+1+(D\cos\theta+1-D\sin\theta)$，这是关于 D 的线性表达式。

1(b)：当 $y_T > x_T$ 时，卵石 p_1 和 p_2 分别放置在 (y_T+1, y_T+1) 和 (x_T, y_T+1) 处。因此，$Bp_1 = D\sin\theta + 1$ 且 $PB = Bp_1$（因为 Δp_1PB 是一个等腰三角形），这意味着 $Pp_1 = \sqrt{2}(D\sin\theta+1)$。此外，在这种情形下，$p_1p_2 = (D\sin\theta+1) - D\cos\theta$ 且 $p_2T = p_2A - TA = D\sin\theta + 1 - D\sin\theta = 1$。因此，总成本为：$\sqrt{2}(D\sin\theta+1) + (D\sin\theta+1) - D\cos\theta + 1$，这同样是关于 D 的线性表达式。

因此，$f_1(\theta)$ = 最大值{最小值$_{\forall\theta\in[0,2\pi]}$}{$\sqrt{2}(D$ 余弦 $\theta+1) + 1 + (D$ 余弦 $\theta + 1 - D$ 正弦 $\theta)$,$\sqrt{2}(D$ 正弦 $\theta+1) + (D$ 正弦 $\theta+1) - D$ 余弦 $\theta+1\}\}$

2：如果宝藏位于第二象限，令 C 为 T在第一象限的镜像（参见图 1(b)），再令 A 和 B 分别为从 C 和 p_1向垂线所作的垂足。令 $\angle TPD = \theta$，因此 $\angle CPA = \theta$。于是，我们有 $PA = D\cos\theta$和 $AC = D\sin\theta$。下面分情形讨论：

2(a)：当 $|x_T| \geq y_T$时，卵石 p_1 和 p_2 分别放置在 $(-x_T+1, -x_T+1)$ 和 $(x_T, -x_T+1)$。因此，$PB = D\cos\theta + 1$且$PB = Bp_1$（因为 Δp_1PB是等腰三角形），这意味着 $Pp_1 = \sqrt{2}(D\cos\theta+1)$。此外，在这种情形下，$p_1p_2 = D\cos\theta + 1 + D\cos\theta$且 $p_2T = p_2A - TA = D\cos\theta + 1 - D\sin\theta$。因此，总成本为：$\sqrt{2}(D\cos\theta+1) + (D\cos\theta+1-D\sin\theta) + (2D\cos\theta+1)$，它是关于 D的线性表达式。

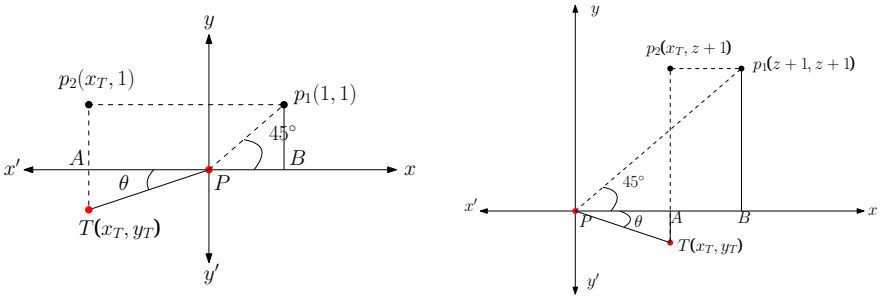
2(b)：当 $y_T > |x_T|$时，卵石 p_1和 p_2分别放置在 (y_T+1, y_T+1) 和 (x_T, y_T+1) 处。因此，$Bp_1 = D\sin\theta+1$且 $PB = Bp_1$（因为Δp_1PB是一个等腰三角形），这意味着 $Pp_1 = \sqrt{2}(D\sin\theta+1)$。我们有 $p_1p_2 = (D\sin\theta+1) + D\cos\theta$，且 $p_2T = p_2A - TA = D\sin\theta + 1 - D\sin\theta = 1$。因此，总成本为 $\sqrt{2}(D\sin\theta+1) + (D\sin\theta+1) + D\cos\theta+1$，这同样是关于 D 的线性表达式。

因此，$f_2(\theta)$ = max{min$_{\forall\theta\in[0,2\pi]}${$\sqrt{2}(D\cos\theta+1) + (D\cos\theta+1-D\sin\theta) + (2D\cos\theta+1)$}{$\sqrt{2}(D\sin\theta+1) + (D\sin\theta+1) + D\cos\theta+1\}$}。

3：如果宝藏位于第三象限，令 A 和 B 分别为从 p_2 和 p_1作垂线所得的垂足。令 $\angle TPA = \theta$（参见图 2(a)），并将卵石 p_1 和 p_2 分别放置在 (1，1) 和 $(x_T$，$-1)$ 处。因此，$Pp_1 = \sqrt{2}$,$p_1p_2 = 1 + D\cos\theta$和 $p_2T = p_2A + AT = 1 + D\sin\theta$。所以，总成本为：$\sqrt{2} + 1 + D\cos\theta + D\sin\theta$，这再次是关于 D的线性表达式。因此，$f_3(\theta)$ = min$_{\forall\theta\in[0,2\pi]}${$\sqrt{2} + 1 + D\cos\theta + D\sin\theta$}。

4：如果宝藏位于第四象限，则令 A 和 B 分别为从 T和 p_1作垂线所得的垂足。令 $\angle TPA = \theta$（参见图 2(b)）。因此，$PA = D\cos\theta$和 $AT = D\sin\theta$。现在我们有以下几种情形：

4(a)：当 $x_T \geq |y_T|$时，卵石 p_1和 p_2 分别放置在 $(x_T+1,\ x_T+1)$ 和 $(x_T,\ x_T+1)$。因此，$PB = D\cos\theta+1$ 和 $PB = Bp_1$（因为 Δp_1PB 是等腰三角形），这意味着 $Pp_1 = \sqrt{2}$（$D\cos\theta+1$）。所以，



(a) 第三象限中的宝藏

(b) Treasure in 4th Quadrant

图 2：当宝藏位于平面的下半部分时智能体的移动情况

我们有 $p_1p_2 = 1$ 和 $p_2T = p_2A + TA = D\cos\theta + 1 + D\sin\theta$。因此，总成本为：$\sqrt{2}(D\cos\theta + 1) + 1 + (D\cos\theta + 1 + D\sin\theta)$，这是关于 D的线性表达式。

4(b)：当 $|y_T| > x_T$ 时，卵石 p_1和 p_2分别放置在 $(-y_T + 1, -y_T + 1)$ 和 $(x_T, -y_T + 1)$ 处。因此，$Bp_1 = D\sin\theta + 1$和$PB = Bp_1$（因为 Δp_1PB是一个等腰三角形），这意味着 $Pp_1 = \sqrt{2}(D\sin\theta + 1)$。因此，我们有 $p_1p_2 = (D\sin\theta + 1) - D\cos\theta$ 和 $p_2T = p_2A + TA = D\sin\theta + 1 + D\sin\theta = 2D\sin\theta + 1$。所以，总成本为：$\sqrt{2}(D\sin\theta + 1) + (D\sin\theta + 1) - D\cos\theta + 2D\sin\theta + 1$，这再次是关于 D 的线性函数。因此，我们有 $f_4(\theta) = \max\{\min_{\forall\theta\in[0,2\pi]}\{\sqrt{2}(D\cos\theta + 1) + 1 + (D\cos\theta + 1 + D\sin\theta) \sqrt{2}(D\sin\theta + 1) + (D\sin\theta + 1) - D\cos\theta + 2D\sin\theta + 1\}\}$

此外，累积代价为 $f_5(\theta) = \max_{\forall i\in\{1,\cdots,5\}}\{f_i(\theta)\}$，其近似为 $4.5D + (\sqrt{2} + 2)$。因此，根据上述所有情形，我们得出结论：代价复杂度关于 D 呈线性增长。

□

3 Improved solution for treasure hunt

在本节中，我们提出了一种更快的算法，该算法至少需要 9 颗卵石才能执行寻宝任务。

3.1 High level idea

在介绍鹅卵石放置算法的具体细节之前，我们先描述其整体思路。直观而言，根据可用卵石的数量，预言机按照第 3.2 节中的描述将平面划分为多个扇区。然后，它确定宝藏所在的扇区编号 m 。

并通过摆放卵石来对这个数字进行“编码”。智能体观察卵石的摆放方式，对这种编码进行“解码”，然后沿着特定的扇区移动，以寻找宝藏。要实现这一设想，还需要克服一些挑战。

扇区检测： 第一个难点在于，如何通过不同的卵石放置方式，使智能体能够区分比特0和比特1。由于智能体无法感知时间和距离，两种不同的卵石放置方式在智能体看来可能完全相同。另一方面，由于智能体对编码整数没有任何先验信息，因此也应以一种特殊的方式规划其移动，使得采用相同的移动策略，在某些情况下能够检测出比特0，而在其他情况下则能够检测出比特1。利用将初始点 P 识别为特殊点的能力，可以克服这一难点。

首先，我们在点 (1,0) 处放置一个卵石 p_1，并在 (1,1) 处放置另外一个固定卵石p_2，在 (2,1) 处放置另一个固定卵石 p_3。其余卵石的放置方式取决于编码中的特定位是 0 还是 1。首先，考虑编码仅包含一个比特0或1的具体情形。其基本思想是在 x轴上为特定卵石 p 设置两个可能的位置，使得智能体从 P出发到达 p，然后从 p 以某个固定角度 α 移动时，在其中一个位置到达 p_2，在另一个位置到达 p_3。智能体无法区分 p_2 和p_3，但从 p_2 沿特定角度 β 移动将到达 P，而从 p_3 出发则将到达 p_1。这两种不同的情形分别对应比特0和1。为了实现这一构想，若编码为1，则将卵石 p 放置在点 (3,0) 处；若编码为0，则将其放置在点 (4,0) 处。这种特定放置方式的优势在于：若将 p 放置在 (3,0) 处，则智能体从 P 移动到 p，随后以反正切 $(\frac{-1}{2})$ 的角度移动，便会到达 p_2；然后再以反正切 (3) 的角度移动，便会到达 P。另一方面，若将p 放置在 (4,0) 处，采用相同的移动方式，智能体则会到达 p_1。因此，它能够将这两种不同的观测结果分别识别为两个不同的比特1和0。（见图3）。

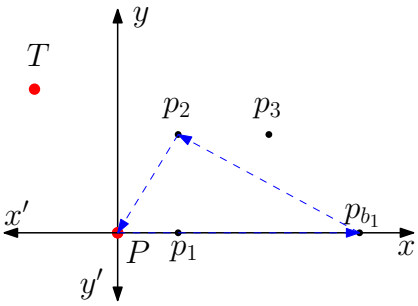


图 3：当第一位为 1 时，预言机放置卵石的位置

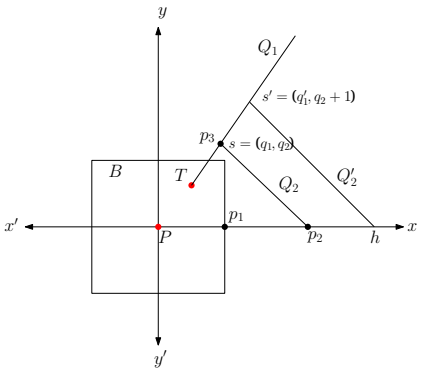


图4：宝藏位于 B 内时的鹅卵石放置

我们将上述思想扩展如下，以实现对任意二进制字符串 μ 的编码。除了卵石 p_1、p_2和 p_3之外，还为 μ 中的每一位额外放置一颗卵石。具体而言，对于 $1 \le i \le |\mu|$，如果第 i位为1，则在$(2i+1,0)$处放置一颗卵石 p_{b_i}；否则，在$(2i+2,0)$处放置 p_{b_i}。从 P出发，朝p_{b_i}方向以反正切 $\left(\frac{-1}{2i}\right)$的角度移动，直到到达一颗卵石，然后以反正切 $\left(\frac{2i+1}{2i-1}\right)$的角度继续移动，智能体最终到达 P或 p_1，具体取决于第 i位是1还是0。

仍有一个需要克服的难点，即智能体如何检测编码的结束。这一点非常重要，因为如果没有指明终止位置，智能体就可能继续移动以寻找更多卵石 p_{b_j}，$j > |\mu|$，并无限距离地持续移动。我们使用另外两颗卵石 p_{t_1} 和 p_{t_2} 来满足终止检测这一特定需求。这两颗卵石 p_{t_1} 和 p_{t_2} 的位置如下。如果二进制字符串 μ 的第 1 位是 1，即将 p_{b_1} 放置在 (3, 0) 处，那么卵石 p_{t_1} 和 p_{t_2} 分别放置在 (4, 1) 和 $(2|\mu| + 6, 0)$ 处。否则，如果第 1 位是 0，则这两颗卵石分别放置在 (5, 1) 和 $(2|\mu| + 7, 0)$ 处。在访问完表示 μ 最后一位的卵石 $p_{|\mu|}$ 后，智能体返回到 P，并像往常一样继续移动以寻找卵石，期望获取二进制字符串的更多位。从 P出发，一旦到达 p_{t_2}，它便以 $\arctan\left(2^{-1}(|\mu|+1)\right)$ 的角度移动，直到到达一颗卵石。注意，这两颗卵石 p_{t_1} 和 p_{t_2} 的放置方式使得角度 $\angle P p_{t_2} p_{t_1} = \arctan\left(2^{-1}(|\mu|+1)\right)$。因此，智能体从 p_{t_2} 以 $\arctan\left(2^{-1}(|\mu|+1)\right)$ 的角度移动，会到达 p_{t_1}；而从 p_{t_1} 以 $\arctan\left(\frac{2(|\mu|+1)+1}{2(|\mu|+1)-1}\right)$的角度移动，则会到达 p_{b_1}。由于上述特定移动会使智能体到达一颗卵石，因此它最初假定自己学到的比特为 0。然而，从 p_{b_1}向西移动时，它会到达另一颗卵石（即卵石 p_1），而不是原点。这一特殊情况向智能体表明编码已经终止。因此，通过这种方式，智能体学到了二进制字符串μ，以及其二进制表示为 μ的整数 Δ。

在扇区内寻找宝藏：在从 T 向 L_{j+1} 作出的垂线的垂足处放置另一颗卵石 p_T（参见图 5）。在学习了 μ_j 的编码后，智能体解码出整数 j，并正确识别出扇区内的两条直线 L_j 和 L_{j+1}，从而帮助智能体确定宝藏的精确位置。

此处在将卵石 p_T 放置在扇区内时会出现一个难题，因为在扇区编号编码过程中，某些已经放置的卵石可能与 p_T的可能位置非常接近（距离为 < 1）。为了解决这一问题，如果宝藏位于左半平面，我们就在正 x 轴上放置用于编码的卵石；如果宝藏位于右半平面，则在负 x 轴上放置这些卵石。为了指示智能体应朝哪个方向移动以找到用于学习编码的卵石，我们在 P处额外放置一颗卵石 p_0。

需要单独处理一些特定情形：如果宝藏位于某个位置 (x, y)，使得 $-1 \le x \le 1$ 和 $y \ge -1$，因为这同样可能导致

在扇区内将 p_T 放置在指定位置时遇到的问题。这些情形的详情将在第 3.2 节解释卵石放置策略时讨论。

3.2 鹅卵石放置

智能体最初位于 P，宝藏位于 T。预言机知道智能体的初始位置 P以及宝藏的位置 $T = (x_T, y_T)$，并在欧几里得平面上放置 k 个卵石。令 B 为由直线 $x = 1$、$x = -1$、$y = 1$和 $y = -1$围成的正方形区域。

根据宝藏的位置，鹅卵石放置分为两种不同的情形。

情况 1：如果 $x_T > 0$ 和 $T \notin B$，那么卵石的放置方式如下。

1. 在 P处放置一颗卵石 p_0 。2. 从智能体的初始位置 P出发，绘制 2^{k-8} 条半直线 $L_0, \cdots, L_{2^{k-8}-1}$，使得 L_0 指向北方，且相邻半直线之间的角度为 $\pi/2^{k-8}$，其中 $i = 0, \cdots, 2^{k-8} - 1$。扇区 S_i 定义为平面中位于 L_i 和$L_{(i+1) \bmod 2^{k-8}}$ 之间的点集，包括 L_i 上的点，但不包括$L_{(i+1) \bmod 2^{k-8}}$上的点。如果对于某个 $j \in \{0, 1, \cdots, 2^{k-8} - 1\}$，有 $T \in S_j$，则按如下方式放置卵石。– 将卵石 p_1 放置在(-1,0)，将 p_2 放置在(-1,-1)，将 p_3 放置在(-2,-1)。– 令 μ_j 为整数 j 的二进制表示，并在前面补上 $\lfloor \log k \rfloor - \lfloor \log j \rfloor$ 个零。如果 $0 \le x_T \le 1$ 且 $y_T > 1$，则 $\mu_j = 0 \cdot \mu_j$，否则$\mu_j = 1 \cdot \mu_j$。对于 $1 \le \ell \le |\mu_j|$，如果 μ_j的第 ℓ位为1，则在$(-2\ell - 1, 0)$处放置一颗卵石，否则在$(-2\ell - 2, 0)$处放置一颗卵石。– 如果 μ_j 的第1位为1，则在(-4,-1)处放置一颗卵石 p_{t_1}，否则在(-5,-1)处放置 p_{t_1}。– 如果 μ_j 的第1位为1，则在$(-2|\mu_j| - 6, 0)$处放置一颗卵石 p_{t_2}，否则在$(-2|\mu_j| - 7, 0)$处放置 p_{t_2}。

3. 如果 $x_T < 0$ 和 $T \notin B$，那么卵石的放置方式如下。对于每个放置在 (m, n) 处的卵石，其中 $m \neq 0$ 或 $n \neq 0$ 在上述情形中，将对应的卵石放置在本情形中的 $(-m, -n)$ 处。此外，在 P 处不放置任何卵石。

4. 如果 μ_j 的第一位是 0，则令 F 为从 T 向 L_j 作垂线所得的垂足；否则，令 F 为从 T 向 L_{j+1} 作垂线所得的垂足。在 F 处放置一颗卵石 p_T（引理 1 确保在这种情况下，卵石之间的距离至少为 1）。

情况 2：如果 $x_T > 0$ 和 $T \in B$，那么卵石按以下方式放置。– 在 (1,0) 处放置一颗卵石 p_1 （参见图4）。– 令 $m_1 = \tan\left(\pi - \text{反正切}(\frac{-1}{2}) - \text{反正切(3)}\right)$且 $m_2 = \tan(\pi - \text{反正切}(\frac{-1}{2}))$。画一条经过 T、斜率为 m_1 的直线 $\overline{Q_1}$，再画一条经过点 (2,0)、斜率为 m_2的直线 Q_2。令 $s = (q_1, q_2)$ 为这两条直线的交点。令 s' 为直线 Q_1上的点。

其坐标为 y，且坐标值为 q_2+1。绘制经过 s' 且与 Q_2 平行的直线 Q_2'。设直线 Q_2' 与 x 轴的交点为 h。另外还放置了两块卵石 p_2 和 p_3，具体如下。

- 如果 $q_2<1$，则将 p_2 放在 h 处，并将 p_3 放在 s'处。
- 否则，将 p_2 放在 (2,0) 处，并将 p_3 放在 s处。

如果 $x_T<0$ 和 $T\in B$，则按如下方式放置卵石。– 将卵石 p_0 放在 P 处，并将 p_1 放在 (-1,0) 处。–令 $m_1=-\tan\left(\pi-\text{反正切}(\frac{-1}{2})-\text{反正切}(3)\right)$，且 $m_2=-\tan(\pi-\text{反正切}\left(\frac{-1}{2}\right))$。作一条经过 T、斜率为 m_1 的直线 Q_1，并作一条经过点 $(-2,0)$、斜率为 m_2 的直线 Q_2。令 $r=(r_1,r_2)$ 为这两条直线的交点。令 r' 为直线 Q_1 上坐标 y 为 r_2+1的点。作与 Q_2 平行且经过 r'的直线 Q_2'。令 n 为直线 Q_2'与 x-轴的交点。

另外还放置了两颗卵石 p_2 和 p_3，具体如下。
- 如果 $r_2<1$，则将 p_2 放置在 n 处，并将 p_3 放置在 r'处。
- 否则，将 p_2 放置在$(-2,0)$处，并将 p_3 放置在 r处。

3.3 Treasure hunt

从 P 出发，智能体借助放置在平面不同位置的卵石寻找宝藏。从较高层次来看，智能体主要执行三项任务：(1) 学习其初始移动的方向；(2) 学习宝藏所在扇区编号的编码方式；(3) 在指定扇区内移动并找到宝藏。

智能体通过观察是否在 P 处放置了卵石，来学习其初始移动的方向。如果放置了卵石，则智能体得知其初始移动方向为西，并在负 x 轴上通过放置卵石来编码扇区编号。否则，它得知其初始移动方向为东，并在正 x 轴上通过放置卵石来编码扇区编号。然后，对于每个 $j=1,2,\cdots$，它沿着特定路径继续移动（具体取决于 j 的值），并学习编码的第 j 位，直到检测到编码终止。具体而言，智能体按照以下顺序从 P 开始移动，从而学习编码的第 j 位。

– 从 P 出发，沿 x 轴移动，直到到达第 $j+1$ 个卵石，– 以 $\arctan(\frac{-1}{2j})$ 的角度移动，并沿此方向继续移动，直到遇到一个卵石– 以 $\arctan(\frac{2j+1}{2j-1})$ 的角度移动，直到找到 P或一个卵石。–如果在上一步中找到 P，则该位为 1。– 如果遇到一个卵石，则沿 x 轴朝 P移动。如果遇到 P，则该位为 0。– 如果在上一步中遇到的是卵石而不是 P，则智能体会得知编码已完成。

Algorithm 1: PEBBLEPLACEMENT

画出从 P 开始的 2^{k-8} 半直线 $L_0, \cdots, L_{2^{k-8}-1}$，其间的角度
两条连续半直线是 $\frac{\pi}{2^{k-8}}$。设扇区~~—~~S_i 为由.....界定的扇区。
半条直线 L_i 和 L_{i+1}，并令 $T \in S_\Delta$, $\Delta \in \{0, 1, \cdots, 2^{k-8}-1\}$

if $x_T \geq 0$ **then**
  如果 $0 \leq x_T \leq 1$ *and* $-1 \leq y_T \leq 1$ **则**
    SquarePlacement(2)
  else
    Place a pebble p_0 at P
    如果 $x_T \leq 1$ *and* $y_T > 1$ **则**
      NonSquarePlacement$(1, 0)$
      将一颗卵石 p_T 放置在从 T 作垂线所得的垂足处 on L_Δ.
    else
      NonSquarePlacement$(1, 1)$
      将一块卵石 p_T 放置在从 T 引出的垂线的垂足处。on $L_{\Delta+1}$.
else
  if $-1 \leq x_T \leq 0$ *and* $-1 \leq y_T \leq 1$ **then**
    Place a pebble p_0 at P.
    SquarePlacement(1)
  else
    if $-1 \leq x_T \leq 0$ *and* $y_T > 1$ **then**
      NonSquarePlacement(2,0)
      将一块卵石 p_T 放置在从 T 作垂线所得的垂足处 on L_Δ.
    else
      NonSquarePlacement(2,1)
      将一块卵石 p_T 放在从 T 引出的垂线的垂足处 on $L_{\Delta+1}$.

Algorithm 2: SQUAREPLACEMENT($count$)

在 $((-1)^{count}, 0)$ 处放置一块卵石 p_1。
if $q_2 < 1$ **then**
  分别将卵石放置在 p_2 处和 p_3 处。
else
  分别将一块卵石 p_2 放置在 $((-1)^{count} \cdot 2, 0)$ 处，并将 p_3 放置在 s 处。

算法3： NonSquarePlacement($count, bit$)

Initially $l = 2$.
将卵石 p_1 放置在 $((-1)^{count}$,0)、 p_2 放置在 $((-1)^{count}, (-1)^{count})$ 以及 p_3 放置在 $((-1)^{count} \cdot 2,\ (-1)^{count})$，分别。
μ_j 为整数 j 的二进制表示，前导 $\lfloor \log k \rfloor - \lfloor \log j \rfloor$ 许多个零 .
$\mu_j = bit.\mu_j$ ▷ 表示将 bit value 与 μ_j连接
如果 $bit = 1$ **则**
    在 $((-1)^{count} \cdot 3,\ 0)$ 处放置一颗卵石。
否则
    在 $((-1)^{count} \cdot 4,\ 0)$ 处放置一颗卵石。
while $l \leq k + 1$ **do**
    if *ℓ-th bit of μ_j is* 1**then**
        在 $((-1)^{count} \cdot (2\ell + 1)$,0) 处放置一颗卵石。
    else
        在 $((-1)^{count} \cdot (2\ell + 2)$,0) 处放置一块卵石。
    $l = l + 1$
如果 *1st bit of μ_j is 1* **则**
    将卵石放置在 p_{t_1} $((-1)^{count} \cdot 4$,$(-1)^{count})$ 和 p_{t_2} 处。
    $((-1)^{count} \cdot (2|\mu_j| + 6),\ 0)$，分别。
else
    将卵石 p_{t_1} 放置在 $((-1)^{count} \cdot 5$,$(-1)^{count})$ 处，并将 p_{t_2} 放置在 $((-1)^{count} \cdot (2|\mu_j| + 7),\ 0)$，分别。

Algorithm 4: AgentMovement

如果在 P 找到卵石，则设置 $angle = \pi$，否则设置 $angle = 0$。
$t = 2$, $\mu = \epsilon$
从与正 x 轴成 $angle$ 角度的方向开始移动。
if *treasure is found* **then**
    Terminate
else
    继续沿着相同的方向移动，直到第 t 个卵石或 the treasure is found.
    如果 *treasure found* **则**
        Terminate
    else
        ℓ=FindBit($t, angle$)
        如果 $\ell \in \{0, 1\}$ **则**
            $\mu = \mu \cdot \ell$.
            $t = t + 1$.
            Go to Step 3
        else
            FindTreasure($\mu, angle$)

算法 5: FindBit$(t, angle)$

以角度 $\pi - \theta_t$ 移动，其中 $\theta_t = \arctan(\frac{-1}{2t})$，直到找到宝藏或遇到卵石 is found.
if *treasure found* **then**
    Terminate
else
    以角度 $\pi - \beta_t$ 移动，其中 $\beta_t =$ 反正切$(\frac{2t+1}{2t-1})$。
    if *treasure found* **then**
        Terminate
    else if P *is found* **then**
        return 1
    否则如果 *a pebble is found at a point other than* P **则**
        如果 $angle = 0$ **则**
            以角度 $\pi + \frac{\pi}{4}$ 移动。 –
        else
            以角度 $\pi - \frac{\pi}{4}$ 移动。 –
        如果 P *is found* **则**
            返回 0
        else
            Continue its movement until P is reached.
            return 2

算法 6: FindTreasure$(\mu, angle)$

设 Δ 为其二进制表示为 μ 的整数。
if $angle = \pi$ **then**
    if $\mu_1 = 0$ **then**
        $val = \Delta$
        SectorTravel(val,1, 2)
    else
        $val = \Delta + 1$
        SectorTravel(val,1, 1)
else
    if $\mu_1 = 0$ **then**
        $val = \Delta$
        SectorTravel$(val, 2, 1)$
    else
        $val = \Delta + 1$
        SectorTravel$(val, 2, 2)$

算法 7: SectorTravel$(val, count, num)$

以角度 $\frac{\pi}{2} + (-1)^{count}\left(\frac{\pi \cdot val}{2^{k-8}}\right)$ 移动，直到找到卵石或宝藏。

if *Treasure found* **then**
  Terminate.
else
  以角度 $\pi + (-1)^{num}\frac{\pi}{2}$ 移动，直到找到宝藏。
  Terminate.

设 μ 为智能体在上述过程中学到的二进制字符串，设 Δ 为其二进制表示为 μ 的整数。如果 μ 的第一位为 1，则智能体从 P 开始沿着 $L_{\Delta+1}$ 移动，直到碰到一块卵石或到达宝藏。一旦到达卵石，如果其初始移动方向为西，智能体就以 $\frac{\pi}{2}$ 的角度改变方向；否则，智能体以 $\frac{3\pi}{2}$ 的角度改变方向。随后，智能体沿当前方向继续移动，直到到达宝藏。

以下引理确保在上述卵石放置策略的情况 1 的步骤 4 中，卵石之间的距离至少为 1。

引理 1。 如果 $T = (x_T, y_T) \in B'$，其中 $B' = \{(x, y) |\ 0 \leq x \leq 1$ 且 $y_T > 1\}$，则垂线 F 在 L_j 上的垂足位于正方形 B 的外部。

证明。设 F 的位置为 (h, k)。设 m_1 为直线 PF 的斜率，m_2 为直线 FT 的斜率（见图 7）。现在，由于 $PF \perp FT$，因此 $m_1 \cdot m_2 = -1$。斜率 $m_1 = \frac{k}{h}$ 为正，因为 $k > 0$ 且 $h > 0$，所以为了满足上述条件，$m_2 = \frac{y_T - k}{x_T - h}$ 必须为负。现在，m_2 在以下情形之一成立时可以为负。

– 情形 1: $y_T > k$ 和 $x_T < h$，–
情况 2: 如果 $y_T < k$ 和 $x_T > h$,

如果情况 1成立，那么点 F 必须位于直线 $x = X_T$ 的右侧，这是不可能的。因此，情况 2 必须成立，即 $1 < y_T < k$ 和 $x_T > h$。这意味着 F 位于 B 的外部。

□

下面通过一个示例来解释该算法的执行过程。

示例 1: 给定 11 颗卵石，预言机将平面划分为 2^{11-8} 个扇区。假设宝藏被放置在扇区 5 中（如图 5 所示）。此外，考虑宝藏的位置位于正方形 B 之外。因此，预言机按照算法 *PebblePlacement 1* 放置卵石，使得智能体在遵循算法 *AgentMovement 4* 后，能够获知其初始移动的方向，并进一步按照以下方式获知宝藏所在扇区编号的编码（即本例中的 101）。

算法 4 的一次迭代定义为一个循环，其中智能体从 P 出发移动并返回 P。在第一次迭代中，最初位于 P 的智能体没有在 P 找到卵石。算法 4 指示

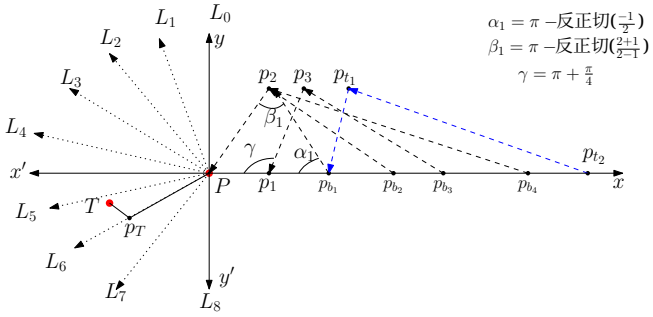


图 5：示例 1 的演示图

智能体朝东移动，直到沿正 x 轴遇到第二颗卵石 p_{b_1}。智能体从 p_{b_1} 以 $\pi -$ 角度，即反正切($\frac{-1}{2}$)的方向移动，直到遇到一颗卵石 p_2。从 p_2 出发，它继续以 $\pi -$ 角度，即反正切 $\binom{2+1}{2-1}$ 的方向移动，直到到达原点 P。因此，第一次迭代完成后（即走过路径 $P \to p_{b_1} \to p_2 \to P$），智能体得知第一位为 1。在第二次迭代中，智能体再次朝东移动，直到到达第三颗卵石 p_{b_2}。智能体从 p_{b_2} 以 $\pi -$ 角度，即反正切 $\binom{-1}{2\cdot 2}$ 的方向移动，直到遇到一颗卵石 p_2；然后从 p_2 继续以 $\pi -$ 角度，即反正切 $\binom{2\cdot 2+1}{2\cdot 2-1}$ 的方向移动，直到到达原点 P。因此，第二次迭代完成后（即走过路径 $P \to p_{b_2} \to p_2 \to P$），智能体再次得知第二位为 1。在第三次迭代中，智能体以类似的方式朝东移动，沿正 x 轴到达第四颗卵石 p_{b_3}。从 p_{b_3} 出发，它继续以 $\pi -$ 角度，即反正切($\frac{-1}{2\cdot 3}$)的方向移动，直到到达卵石 p_3。从 p_3 出发，它以 $\pi -$ 角度，即反正切 $\binom{2\cdot 3+1}{2\cdot 3-1}$ 的方向移动，直到到达一颗卵石 p_1。然后，智能体从 p_1 最后以 $\pi + \frac{\pi}{4}$ 角度移动，直到到达 P。因此，第三次迭代完成后（即走过路径 $P \to p_{b_3} \to p_3 \to p_1 \to P$），智能体得知第三位为 0。在第四次迭代中，智能体以类似的方式到达卵石 p_{b_4}，并从该位置以 $\pi -$ 角度，即反正切 $\binom{-1}{2\cdot 4}$ 的方向移动，直到到达 p_2；然后从 p_2 继续以$\pi -$角度，即反正切 $\binom{2\cdot 4+1}{2\cdot 4-1}$ 的方向移动，直到到达 P。因此，在第四次迭代中（即走过路径 $P \to p_{b_4} \to p_2 \to P$），智能体得知第四位为 1。在第五次迭代中，智能体到达第五颗卵石，即 p_{t_2}（参见图 5）；从p_{t_2}出发，它以 $\pi -$ 角度，即反正切 $\binom{-1}{2\cdot 5}$ 的方向移动，直到遇到一颗卵石 p_{t_1}；从该位置继续以 $\pi -$ 角度，即反正切 $\binom{2\cdot 5+1}{2\cdot 5-1}$ 的方向移动，直到遇到一颗卵石p_{b_1}。接着，智能体从 p_{b_1} 继续以 $\pi + \frac{\pi}{4}$ 角度移动，直到到达P。由于智能体在从 p_{b_1} 出发进行最后一次移动后遇到了卵石 p_1，因此它得知已达到终止条件。由此，智能体得到的二进制字符串为 $\mu = 1101$（假设）。

随后，智能体按照算法 $AgentMovement\ 4$解码得知宝藏位于扇区 5 的某处。此外，由于 $\mu_1 = 1$，智能体接着执行算法 $FindTreasure\ 6$ 和 $SectorTravel\ 7$，最终沿着半直线 L_6 前进并遇到卵石 p_T，再以角度 $\pi + \frac{\pi}{2}$ 移动，最终抵达宝藏 T。

4 复杂度

本节将证明所提出算法的正确性，并给出寻找宝藏所需代价的上界。

以下两个引理证明了当宝藏位于 B 内时算法的正确性，并给出了寻宝代价的上界。

引理 2. 有 *3* 颗卵石且宝藏位于 B 内时，智能体能够成功找到宝藏。

证明。 当宝藏位于正方形 B 内部时，如果宝藏位于 y-轴的左半部分，预言机就会在 P 处放置一颗卵石 p_0。否则，如第 3.2 节中的情况 2 所述，不会在 P 处放置卵石（另请参阅算法 1 的第 3 行和第 14 行）。因此，如果智能体在 P 处发现一颗卵石 p_0，它就会从 P 出发，沿角度 π 移动，即沿负 x-轴移动（参见算法 4 的第 1 行和第 2 行）；否则，如果在 P 处没有发现卵石，智能体就会沿角度 0 移动，即沿正 x-轴移动（参见算法 4 的第 4 行）。现在，根据 P 处是否存在卵石，我们可以得到以下几种情况。

– 在以下位置未找到卵石 P：在这种情形下，智能体沿正x轴移动时，要么找到宝藏并终止算法（参见算法 4 的第 7 行和第 8 行）。否则，智能体会找到一个放置在 (1,0) 处的卵石 p_1，并按照算法 $FindBit$ 5 中的指示忽略它，然后继续移动，直到到达宝藏或遇到一个卵石。如果未找到宝藏，则预言机会将卵石 p_2 放置在 h 或 (2,0) 处（参见算法 2 的第 10 行和第 13 行）。智能体遇到第二个卵石后，将沿着角度为 $\pi - \theta_1$ 的方向移动，其中 $\theta_1 =$ 反正切 $\binom{-1}{2}$，直到找到宝藏或遇到一个卵石。如果找到卵石，则有以下几种情形：

- p_2 放置于 *(2,0)*：在这种情形下，智能体找到了卵石 p_3，位置在 s（参见算法 2 的第 14 行），随后它进一步沿着角度 $\pi - \beta_1$ 移动，其中 $\beta_1 =$ 反正切(3)，并找到了宝藏。
- p_2 放置于 h：在这种情形下，智能体找到位于 s' 的卵石 p_3（参见算法 2 的第 11 行），随后沿着角度 $\pi - \beta_1$（其中 $\beta_1 =$ 反正切(3)）的方向继续移动，并找到了宝藏。

– 发现卵石的位置：P：在这种情形下，智能体沿着负 x轴移动，并执行与上述描述类似的任务。原因在于，卵石以类似的方式放置，只是位于相邻的半轴上（即y轴的左半部分），如上述情形所述（参见第 3.2 节情况 2 中的 $x_T < 0$）。

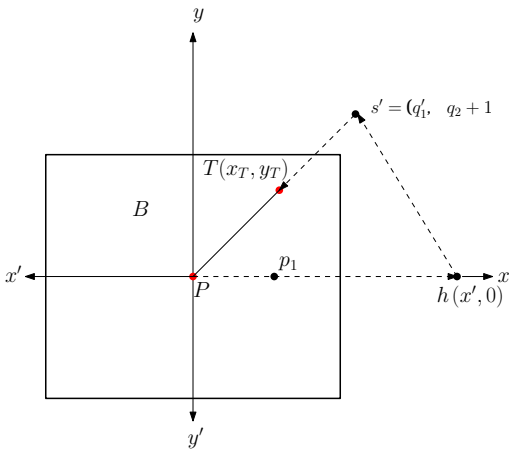


Fig. 6: Traversal of an agent when the treasure is inside th正方形

□

引理 3. 当宝藏位于 B 内部时，从 P 出发的智能体能够成功找到宝藏，代价为 $O(D)$。

证明。宝藏位于 (x_T, y_T)，设 h 的坐标为 $(x', 0)$。预言机将卵石 p_2 和 p_3 放置在最不利的位置，即 h 和 s' 处（其中 $s' = (q_1', q_2 + 1)$，参见图 6 以及算法 2 的第 10 行和第 11 行）。因此，智能体将沿着路径 $Ph \to hs' \to s'T$ 前往宝藏。此次遍历的总成本如下：$x' + \sqrt{(x' - q_1')^2 + (q_2 + 1)^2} + \sqrt{(q_1' - x_T)^2 + (q_2 + 1 - y_T)^2}$（其中 $|Ph| = x'$，$|hs'| = \sqrt{(x' - q_1')^2 + (q_2 + 1)^2}$ 和 $|s'T| = \sqrt{(q_1' - x_T)^2 + (q_2 + 1 - y_T)^2}$）。现在，由于 $|PT| < |Ph| + |hs'| + |s'T|$，即 $D < |Ph| + |hs'| + |s'T|$（因为 $|PT| = D$）。因此，在这种情形下，前往宝藏的代价为 $O(D)$。□

引理 4、引理 5、引理 6 和引理 7 说明，当宝藏位于 B 之外时，算法是正确的。

引理 4。 当宝藏位于 B 之外时，智能体能以代价 $O(j)$成功找到二进制字符串 μ 的第 j 位。

证明。为获得 μ 的第 j 位，智能体的移动如下。

当宝藏位于正方形 B 外部时，若宝藏位于 y 轴的右半部分，预言机便会在 P 处放置一块鹅卵石p_0；否则，不会在 P 处放置鹅卵石。这与第 3.2 节情况 1 中所述一致（参见算法 1 第 6 行）。智能体从 P 开始的移动如下：

- p_0 **发现于** P：在这种情形下，智能体以角度 π移动，即沿负 x 轴移动（参见算法 4 的第 1、2 和 6 行）。此外，它

忽略沿负 x 轴上的前 j 个卵石（参见算法 4 的第 10 行），并继续移动，直到找到宝藏或遇到第 $(j+1)$ 个卵石 p_{b_j} 或 p_{t_1}，这些卵石分别放置在 $(-2j-1,\ 0)$、$(-2j-2,\ 0)$、$(-2j-6,\ 0)$ 或 $(-2j-7,\ 0)$ 处。如果没有找到宝藏，到达这个卵石的代价至多为 $2j+7$。现在，从当前位置出发，指示智能体以角度 $\pi-\theta_j$ 移动（参见算法 $FindBit$5 的第 1 行），其中 $\theta_j=$反正切 $\left(\frac{-1}{2j}\right)$，直到遇到宝藏或卵石 p_2 或 p_3 或 p_{t_2}。如果找到卵石，那么该卵石分别放置在 (-1,- 1) 或 (-2,-1) 或 (-4,-1) 或 (-5,-1) 处，即 p_2、p_3 或 p_{t_2}。因此，从第 $(j+1)$ 个卵石到 p_2 或 p_3或 p_{t_2} 的这段行进代价至多为 $\sqrt{(2j+2)^2+1}$。从这些卵石中的任意一个出发，还会指示智能体沿角度 $\pi-\beta_j$ 移动，其中 $\beta_j=$反正切 $\left(\frac{2j+1}{2j-1}\right)$（参见算法 5 的第 5 行），直到遇到宝藏或卵石，或者以代价 $O(1)$ 到达 P。现在有以下几种情形：

- 如果找到宝藏：在这种情形下，智能体已达到目标，整个过程随之终止。
- 如果找到卵石：在这种情形下，找到的卵石要么是 p_1，要么是 p_{b_1}。在这两种情形中的任意一种下，都会进一步指示智能体沿角度 $\pi+\frac{\pi}{4}$ 或 $\pi-\frac{\pi}{4}$ 移动（参见算法 5 的第 12 行和第 14 行），直到到达 P 或找到一块卵石。因此，我们有两种情形：
  - 如果 P 到达：智能体获知 μ 的第 j 位是 0（参见算法 5 的第 15 和 16 行，以及算法 4 的第 14、15 和 16 行）。因此，获取这一信息所经过的路径为 $P\to p_{b_j}\to p_3\to p_1\to P$。所以，此次遍历的代价至多为 $(2j+2)+\sqrt{(2j)^2+1}+O(1)$，即 $O(j)$。
  - 如果找到卵石：在这种情况下，智能体继续移动，直到到达 P，此时智能体获知终止已实现，即 $(j-1)$ 位是 μ 的终止位。随后，智能体继续移动以执行算法 $FindTreasure$（参见算法 5 的第 18 行，以及算法 4 的第 15 和 20 行）。因此，获取这一信息所经过的路径为 $P\to p_{t_1}\to p_{t_2}\to p_{b_1}\to p_1\to P$。所以，此次遍历的代价至多为 $(2j+7)+\sqrt{(2j+2)^2+1}+O(1)$，即 $O(j)$。
- 如果 P 被到达：在这种情形下，智能体获得的信息是 μ 的第 j 位为 1（参见算法 5 的第 8 行和第 9 行，以及算法 4 的第 14、15 和 16 行）。因此，为获取这一信息所经过的路径为 $P\to p_{b_j}\to p_2\to P$。所以，这次遍历的代价至多为 $(2j+1)+\sqrt{(2j)^2+1}+O(1)$，这同样是 $O(j)$。

– 在 P处未找到卵石：在这种情形下，智能体的移动方式与卵石的放置方式类似。对于每个放置在 (m,n) 处的卵石，其中 $m\neq 0$ 和 $n\neq 0$，在上述情形中，预言机都会将相应的卵石放置在 $(-m,-n)$ 处。同样，获取二进制字符串第 j 位的代价为 $O(j)$。

因此，在每种情形下，找到 μ 的第 j 位的代价为 $O(j)$。

□

引理 5。给定 k 个卵石，且宝藏位于 B 之外，智能体以 $O(k^2)$ 的代价成功找到了二进制字符串 μ。

证明。根据引理 4，智能体成功地以 $O(j)$ 的代价确定了 μ 的第 j 位比特值。现在，由于二进制字符串 μ 的长度为 k，这意味着获取 Γ 的总代价为 $\sum_{j=1}^{k} O(j)$，即 $O(k^2)$。□

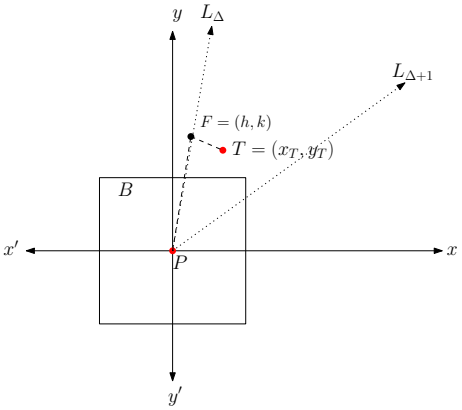


图 7：确定卵石 p_T 的位置

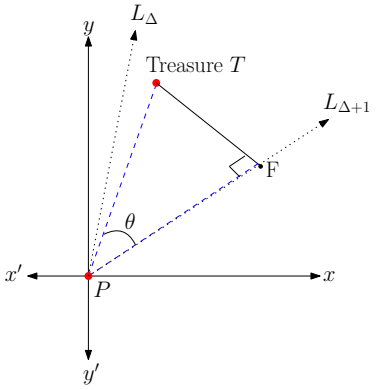


图 8：当 $\mu_1 = 1$ of Γ 时智能体在扇区内的遍历

引理 6. 当宝藏位于 B 之外时，智能体获得二进制字符串 μ 后，通过执行算法 *Find- Treasure* 成功找到宝藏。

证明。算法 *AgentMovement* 4 终止后，智能体使用已获取的二进制字符串 μ 执行算法 *FindTreasure* 6，最终找到宝藏 T（如果此前尚未找到）。

宝藏要么位于区域 $x \geq 0$（即 y 轴的右半部分），要么位于区域 $x \leq 0$（即 y 轴的左半部分）。相应地，预言机将 y 轴的整个左半部分或右半部分划分为 2^{k-8} 个扇区（参见算法 1 的第 1 行），其中一个扇区 S_i 由半直线 L_i 和 L_{i+1} 所限定，且相邻半直线之间的角度为 $\frac{\pi}{2^{k-8}}$。假设宝藏位于扇区 S_Δ 的某处，那么 μ 就是 Δ 的二进制表示。智能体在执行算法 *AgentMovement* 4 后解码出该值 Δ。预言机的全部目标是使智能体沿着半直线 L_Δ 或 $L_{\Delta+1}$ 对齐。智能体沿半直线 L_Δ 或 $L_{\Delta+1}$ 对齐，取决于 μ 的第一位的值，即取决于 μ_1（参见算法 6 的第 3 行），具体如下：

– 情形 $\mu_1 = 0$: 如果在 P 处找到一块卵石（即， $angle = \pi$ 参见算法 4 第 2 行），则指示智能体沿角度 $\frac{\pi}{2} - \frac{\pi\Delta}{2^{k-8}}$ 移动，即沿半直线 L_Δ 移动，直到找到宝藏或一块卵石（参见算法 6 第 4 行和第 5 行，以及算法 7 第 1 行）。否则，如果在 P 处没有找到卵石（即， $angle = 0$ 参见算法 4 第 4 行），则指示智能体沿角度 $\frac{\pi}{2} + \frac{\pi\Delta}{2^{k-8}}$ 移动（参见算法 6 第 11 行和第 12 行，以及算法 7 第 1 行），直到找到宝藏或一块卵石。

- 如果找到宝藏：那么算法将终止，因为我们已经达到了目标（参见算法 7 的第 2 行和第 3 行）。
- 如果找到卵石：进一步指示智能体沿着角度$\pi + \frac{\pi}{2}$ 或 $\pi - \frac{\pi}{2}$ 移动，具体取决于角度 π 或 0（参见算法 7 第 5 行），直到找到宝藏。

– 情形 $\mu_1 = 1$: 如果在 P （即 $angle = \pi$）处发现卵石，则指示智能体沿角度 $\frac{\pi}{2} - \frac{\pi(\Delta+1)}{2^{k-8}}$ 移动，即沿半直线 $L_{\Delta+1}$前进，直到找到宝藏或卵石（参见算法 6 的第 7 行和第 8 行，以及算法 7 的第 1 行）。否则，如果在 P （即 $angle = 0$）处未发现卵石，则指示智能体沿角度 $\frac{\pi}{2} + \pi_{(\Delta+1)\ 2^{k-8}}$ 移动（参见算法 6 的第 14 行和第 15 行，以及算法 7 的第 1 行），直到找到宝藏或卵石。

- 若找到宝藏：则算法终止，因为我们已经达成目标（参见算法 7 的第 2 行和第 3 行）。
- 若找到卵石：随后指示智能体沿角度 $\pi - \frac{\pi}{2}$ 或 $\pi + \frac{\pi}{2}$ 移动，具体取决于角度 π 或 0（参见算法 7 的第 5 行），直至找到宝藏。

因此，在每种情形下，智能体执行算法 $FindTreasure$后都能成功找到宝藏。

□

引理 7. 当宝藏位于 B之外时，智能体获得二进制字符串 μ 后，以代价 $D(\sin\theta' + \cos\theta')$找到宝藏，其中 $\theta' = \frac{\pi}{2^{k'}}$ 且 $k' = k - 8$。

证明。 智能体获得二进制字符串 μ后，执行算法$AgentMovement$ 和 $FindTreasure$，并沿路径 $PF \to FT$从 P出发成功抵达宝藏（参见图 8）。由于 L_Δ与 $L_{\Delta+1}$之间的角度为 $\frac{\pi}{2^{k-8}}$，因此 $\angle FPT$至多为 $\frac{\pi}{2^{k-8}}$（参见图 8（即 θ'（记作），此外 $\angle TFP = \frac{\pi}{2}$（因为若 $\mu_1 = 1$，则 F是 T到 $L_{\Delta+1}$的垂足；否则，若 $\mu_1 = 0$，则 F是 T到 L_Δ的垂足）且 $|PT| \leq D$）。因此，我们有 $PF = D\cos\theta'$和 $FT = D\sin\theta'$。于是，沿扇区 S_Δ从 P移动至 T的代价为 $PF + FT$，即 $D(\sin\theta' + \cos\theta')$。□

结合引理 6 和引理 7，我们得到本节的最终结果，概括如下定理。

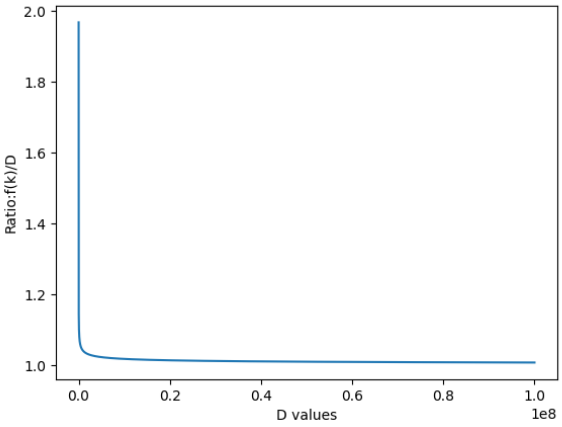


图 9：该曲线表示不同 D 值下 $^{f}{}_{(k)\ D}$ 的比率。

定理 3. 给定 k 个卵石，从 P 出发的智能体能够成功找到宝藏，其代价为 $O(k^2) + D(\sin\theta' + \cos\theta')$，其中 $\theta' = \frac{\pi}{2^{k'}}$ 且 $k' = k - 8$。

注释 1. 考虑函数 $f(k) = O(k^2) + D(\sin\theta' + \cos\theta')$，其中 $\theta' = \frac{\pi}{2^{k-8}}$。注意，对于 D、$k \to \infty$ 和 $k \in o(\sqrt{D})$，$^{f}(k)$ 的值为 $_{D} \to 1$。为了证明这一事实，我们绘制了在 $[1000, 100000000]$ 范围内，随着 D 的值不断增大，以及对于 $k = \lfloor D^{\frac{1}{3}} \rfloor$，$^{f}(k)\ _{D}$ 的取值情况。图 9 展示了在上述范围内，针对不同的 D 值以及每个 D 对应的固定 k 值时，$^{f}(k)\ _{D}$ 的取值。该图表明，当 D 的值较大时，$^{f}(k)\ _{D}$ 的值非常接近 1。

5 Conclusion

我们提出了一种寻宝算法，该算法能够在欧几里得平面中使用 $k \geq 9$ 个卵石以 $O(k^2) + D(\sin\theta' + \cos\theta')$ 的代价找到宝藏，其中$\theta' = \frac{\pi}{2^{k-8}}$。证明与之匹配的下界仍然是一个有待未来研究解决的开放问题。值得注意的是，如果智能体具有一定的可视范围，那么即使只使用一块卵石，该问题也会变得非常简单：在从P 到 T 的直线上，选择一个距离 P不超过 r 的位置放置一块卵石，其中 r 是智能体的可视半径。从 P出发，智能体能够看到卵石的位置，于是移动到卵石处，然后继续前进，直到找到宝藏。然而，如果智能体的罗盘不够精确，问题就会变得具有挑战性：也就是说，如果智能体无法准确测量角度。这似乎是一个很好的未来研究方向，可以作为当前工作的扩展。

References

1. Abiteboul, S., Alstrup, S., Kaplan, H., Milo, T., Rauhe, T.：用于祖先查询的紧凑标记方案。《SIAM 计算杂志》35(6)，1295–1309（2006） 2. Alpern, S., Gal, S.：《搜索博弈与会合理论》，第55卷。Springer Science & Business Media（2006） 3. Beck, A., Newman, D.J.：关于线性搜索问题的更多研究。《以色列数学杂志》8(4)，419–429（1970） 4. Bhattacharya, A., Gorain, B., Mandal, P.S.：使用卵石在图中寻宝。载于：《分布式系统的稳定性、安全性与保障：第24届国际研讨会，SSS 2022，法国克莱蒙费朗，2022年11月15–17日，论文集》。第99–113页。Springer（2022） 5. Bouchard, S., Dieudonn´e, Y., Pelc, A., Petit, F.：利用角度提示在平面中进行确定性寻宝。《Algorithmica》82，3250–3281（2020） 6. Bouchard, S., Labourel, A., Pelc, A.：知识对树中寻宝代价的影响。《网络》80(1)，51–62（2022） 7. Demaine, E.D., Fekete, S.P., Gal, S.：带转向代价的在线搜索。《理论计算机科学》361(2-3)，342–355（2006） 8. Dereniowski, D., Pelc, A.：利用提示信息绘制地图。《并行与分布式计算杂志》72(2)，132–143（2012） 9. Dobrev, S., Kr´aloviˇc, R., Markou, E.：利用提示信息进行在线图探索。载于：《结构信息与通信复杂度：第19届国际研讨会，SIROCCO 2012，冰岛雷克雅未克，2012年6月30日–7月2日，修订精选论文19》。第267–278页。Springer（2012） 10. Emek, Y., Fraigniaud, P., Korman, A., Ros´en, A.：利用提示信息进行在线计算。《理论计算机科学》412(24)，2642–2656（2011） 11. Fraigniaud, P., Gavoille, C., Ilcinkas, D., Pelc, A.：利用提示信息进行分布式计算：图着色的信息敏感性。《分布式计算》21，395–403（2009） 12. Fraigniaud, P., Ilcinkas, D., Pelc, A.：利用提示信息进行树探索。《信息与计算》206(11)，1276–1287（2008） 13. Fraigniaud, P., Ilcinkas, D., Pelc, A.：带提示信息的通信算法。《计算机与系统科学杂志》76(3-4)，222–232（2010） 14. Fraigniaud, P., Korman, A., Lebhar, E.：利用简短提示信息进行局部最小生成树计算。载于：第十九届 ACM 并行算法与体系结构年度研讨会论文集。第154–160页（2007） 15. Fricke, G.M., Hecker, J.P., Griego, A.D., Tran, L.T., Moses, M.E.：用于群体机器人的分布式确定性螺旋搜索算法。载于：2016年 IEEE/RSJ 智能机器人与系统国际会议（IROS）。第4430–4436页。IEEE（2016） 16. Fusco, E.G., Pelc, A.：树中提示信息大小与广播时间之间的权衡。载于：第二十届并行算法与体系结构年度研讨会论文集。第77–84页（2008） 17. Gal, S.：搜索博弈。《Wiley 运筹学与管理科学百科全书》（2010） 18. Gorain, B., Mondal, K., Nayak, H., Pandit, S.：在匿名图中进行卵石引导的最优寻宝。《理论计算机科学》922，61–80（2022） 19. Miller, A., Pelc, A.：会合与寻宝中代价和信息之间的权衡。《并行与分布式计算杂志》83，159–167（2015） 20. Pelc, A., Yadav, R.N.：平面寻宝中的代价与信息权衡。arXiv 预印本 arXiv:1902.06090（2019）

21. Pelc, A., Yadav, R.N.: 《几何地形中寻宝的提示信息复杂度》。Information and Computation 281，104705（2021）。22. Ta-Shma, A., Zwick, U.: 《确定性会合、寻宝与强通用探索序列》。载于：离散算法研讨会：第十八届 ACM-SIAM 离散算法年度研讨会论文集。第7卷，第599–608页（2007）。