View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Carleton University's Institutional Repository

Revisiting the Problem of Searching on a Line*

Prosenjit Bose[1], Jean-Lou De Carufel[1], and Stephane Durocher[2]

[1] School of Computer Science, Carleton University, Ottawa, Canada
[2] Department of Computer Science, University of Manitoba, Winnipeg, Canada

Abstract. We revisit the problem of searching for a target at an unknown location on a line when given upper and lower bounds on the distance D that separates the initial position of the searcher from the target. Prior to this work, only asymptotic bounds were known for the optimal competitive ratio achievable by any search strategy in the worst case. We present the first tight bounds on the exact optimal competitive ratio achievable, parametrized in terms of the given range for D, along with an optimal search strategy that achieves this competitive ratio. We prove that this optimal strategy is unique and that it cannot be computed exactly in general. We characterize the conditions under which an optimal strategy can be computed exactly and, when it cannot, we explain how numerical methods can be used efficiently. In addition, we answer several related open questions and we discuss how to generalize these results to m rays, for any $m \geq 2$.

1 Introduction

Search problems are broadly studied within computer science. A fundamental search problem, which is the focus of this paper, is to specify how a searcher should move to find an immobile target at an unknown location on a line such that the total relative distance travelled by the searcher is minimized in the worst case [3,10,13]. The searcher is required to move continuously on the line, i.e., discontinuous jumps, such as random access in an array, are not possible. Thus, a search corresponds to a sequence of alternating left and right displacements by the searcher. This class of geometric search problems was introduced by Bellman [4] who first formulated the problem of searching for the boundary of a region from an unknown random point within its interior. Since then, many variants of the line search problem have been studied, including multiple rays sharing a common endpoint (as opposed to a line, which corresponds to two rays), multiple targets, multiple searchers, moving targets, and randomized search strategies (e.g., [1,2,3,5,6,7,8,9,12,13,14]).

For any given search strategy f and any given target location, we consider the ratio A/D, where A denotes the total length of the search path travelled by a searcher before reaching the target by applying strategy f, and D corresponds to the minimum travel distance necessary to reach the target. That is, the searcher and target initially lie a distance D from each other on a line, but the searcher

* This research has been partially funded by NSERC and FQRNT.

knows neither the value D nor whether the target lies to its left or right. The *competitive ratio* of a search strategy f, denoted $CR(f)$, is measured by the supremum of the ratios achieved over all possible target locations. Observe that $CR(f)$ is unbounded if D can be assigned any arbitrary real value; specifically, the searcher must know a lower bound $\min \le D$. Thus, it is natural to consider scenarios where the searcher has additional information about the distance to the target. In particular, in many instances the searcher can estimate good lower and upper bounds on D. Given a lower bound $D \ge \min$, Baeza-Yates et al. [3] show that any optimal strategy achieves a competitive ratio of 9. They describe such a strategy, which we call the *Power of Two* strategy. Furthermore, they observe that when D is known to the searcher, it suffices to travel a distance of $3D$ in the worst case, achieving a competitive ratio of 3.

We represent a search strategy by a function $f : \mathbb{N} \to \mathbb{R}^+$. Given such a function, a searcher travels a distance of $f(0)$ in one direction from the origin (say, to the right), returns to the origin, travels a distance of $f(1)$ in the opposite direction (to the left), returns to the origin, and so on, until reaching the target. We refer to $f(i)$ as the distance the searcher travels from the origin during the ith iteration. The corresponding function for the Power of Two strategy of Baeza-Yates et al. is $f(i) = 2^i \min$. Showing that every optimal strategy achieves a competitive ratio of exactly 9 relies on the fact that no upper bound on D is specified [3]. Therefore, it is natural to ask whether a search strategy can achieve a better competitive ratio when provided lower and upper bounds $\min \le D \le \max$.

Given R, the *maximal reach problem* examined by Hipke et al. [10] is to identify the largest bound max such that there exists a search strategy that finds any target within distance $D \le \max$ with competitive ratio at most R. López-Ortiz and Schuierer [13] study the maximal reach problem on m rays, from which they deduce that the competitive ratio $CR(g_{opt})$ of any optimal strategy g_{opt} is at least $1 + 2m^m(m-1)^{1-m} - O(\log^{-2} \rho)$, where $\rho = \max / \min$. When $m = 2$, the corresponding lower bound becomes $9 - O(\log^{-2} \rho)$. They also provide a general strategy that achieves this asymptotic behaviour for a general m, given by $f(i) = (\sqrt{1 + i/m})(m/(m-1))^i \min$. Again, for $m = 2$ this is $f(i) = (\sqrt{1 + i/2})2^i \min$. Surprisingly, this general strategy is independent of ρ. In essence, it ignores any upper bound on D, regardless of how tight it is. Thus, we examine whether there exists a better search strategy that depends on ρ, thereby using both the upper and lower bounds on D. Furthermore, previous lower bounds on $CR(g_{opt})$ have an asymptotic dependence on ρ applying only to large values of ρ, corresponding to having only coarse bounds on D. Can we express tight bounds on $CR(g_{opt})$ in terms of ρ?

Let $g_{opt}(i) = a_i \min$ denote an optimal strategy. Since $D \le \max$, then there must be an n such that $a_n \ge \rho$, i.e., n is the number of iterations necessary to reach the target, so that $g_{opt}(n) \ge \max$. López-Ortiz and Schuierer [13] provide an algorithm to compute the maximal reach for a given competitive ratio together with a strategy corresponding to this maximal reach. They state that the value n and the sequence $\{a_i\}_{i=0}^{n-1}$ for g_{opt} can be computed using binary search, which increases the running time proportionally to $\log \rho$. Can we find a faster algorithm for computing g_{opt}? Since in general, the values $a_0, \ldots, a_{n-1}$ are roots of a polynomial equation of unbounded degree (see Theorem 1), a binary search is equivalent to the bisection method for solving polynomial equations. However, the bisection method is a slowly converging numerical method. Can the computational efficiency be improved? Moreover, given ε, can we bound the number of steps necessary for a root-finding algorithm to identify a solution within tolerance ε of the exact value?

1.1 Overview of Results

We address all of the questions raised above. We characterize g_{opt} by computing the sequence $\{a_i\}_{i=0}^{n-1}$ for the optimal strategy. We do this by computing the number of iterations n needed to find the target and by defining a family of polynomials $p_0, \ldots, p_n$, where p_i has degree $i+1$. We can compute n in $O(1)$ time since we prove that $n \in \{\lfloor \log_2 \rho \rfloor - 1, \lfloor \log_2 \rho \rfloor\}$, where $\rho = \max / \min$. We then show that a_0 is the largest real solution to the polynomial equation $p_n(x) = \rho$. Each of the remaining elements in the sequence $\{a_i\}$ can be computed in $O(1)$ time since we prove that $a_1 = a_0(a_0 - 1)$ and $a_i = a_0(a_{i-1} - a_{i-2})$ for $2 \leq i < n$. This also shows that the optimal strategy is unique. Moreover, as we show in Proposition 1, when no upper bound is known there exist infinitely many optimal strategies for any $m \geq 2$.

We give an exact characterization of g_{opt} and show that $CR(g_{opt}) = 2a_0 + 1$. This allows us to establish the following bounds on the competitive ratio of an optimal strategy in terms of ρ:

$$8\cos^2\left(\frac{\pi}{\lceil \log_2 \rho \rceil + 1}\right) + 1 \leq CR(g_{opt}) \leq 8\cos^2\left(\frac{\pi}{\lfloor \log_2 \rho \rfloor + 4}\right) + 1 \ .$$

López-Ortiz and Schuierer [13] show that $CR(g_{opt}) \to 9$ as $\rho \to \infty$. We show that $g_{opt} \to g_\infty$ as $\rho \to \infty$, where $g_\infty(i) = (2i+4)2^i \min$ has a competitive ratio of 9. We thereby obtain an alternate proof of the result of Baeza-Yates et al. [3]. The strategy g_∞ is a member of the infinite family of optimal strategies in the unbounded case which we describe in Proposition 1.

We assume the Real RAM model of computation, including kth roots, logarithms, exponentiation, and trigonometric functions [15]. The computation of each term a_i in the sequence defining g_{opt} involves computing the largest real root of a polynomial equation of degree $n+1$. We prove that $n+1 \leq 4$ if and only if $\rho \leq 32\cos^5(\pi/7) \approx 18.99761$. In this case the root can be expressed exactly using only the operations $+$, $-$, $\times$, $\div$, $\sqrt{\cdot}$ and $\sqrt[3]{\cdot}$. This implies that if $\max \leq 32\cos^5(\pi/7)\min$, then g_{opt} can be computed exactly in $O(1)$ time ($O(1)$ time per a_i for $0 \leq i \leq n < 4$). In general, when $n+1 \geq 5$, Galois theory implies that the equation $p_n(x) = \rho$ cannot be solved by radicals. Since the corresponding polynomials have unbounded degree, we are required to consider approximate solutions when $\rho > 32\cos^5(\pi/7)$. Therefore, we explain how to find a solution g^*_{opt}, such that $CR(g^*_{opt}) \leq CR(g_{opt}) + \varepsilon$ for a given tolerance ε.

If $n \geq 7\varepsilon^{-1/3} - 4$, we give an explicit formula for a_0. Hence, an ε-approximation can be computed in $O(n) = O(\log \rho)$ time ($O(1)$ time per a_i for $0 \leq i \leq n$). Otherwise, if $32\cos^5(\pi/7) < n < 7\varepsilon^{-1/3} - 4$, we show that a_0 lies in an interval of length at most $7^3\,(n+4)^{-3}$. Moreover, we prove that the polynomial is strictly increasing on this interval. Hence, usual root-finding algorithms work well. Given a_0, the remaining elements of the sequence $\{a_1, \ldots, a_{n-1}\}$ can be computed in $O(n)$ time ($O(1)$ time per a_i for $1 \leq i \leq n$). Finally, we explain how our technique can be generalized to m rays.

2 Searching on a Bounded Line

López-Ortiz and Schuierer [13] showed that there always exists an optimal strategy that is *periodic* and *monotone*. That is, the strategy alternates searching left and right between the two rays and the values in the sequence $\{a_i\}$ are increasing: $i > j$ implies $a_i > a_j$. Thus, it suffices to consider search strategies that are periodic and monotone. Our goal is to identify a strategy f that minimizes

$$CR(f) = \sup_{D \in [\min,\max]} \phi(f, D)/D, \quad \text{where} \quad \phi(f, D) = 2\sum\nolimits_{i=0}^{f^{-1}(D)} f(i) + D$$

denotes the cost incurred by f to find a target at distance D in the worst case and $f^{-1}(D)$ is the smallest integer j such that $f(j) \geq D$.

A simple preliminary strategy is to set $g_0(i) = \max$. The corresponding competitive ratio is

$$CR(g_0) = \sup_{D \in [\min,\max]} (2\max + D)/D = 2\rho + 1 \ .$$

Observe that g_0 is optimal when $\rho = 1$, i.e., when D is known. A second strategy g_1 corresponds to cutting $[\min, \max]$ once at a point $a_0 \min < \rho \min = \max$. Namely, we search a sequence of two intervals, $[\min, a_0 \min]$ and $(a_0 \min, \rho \min] = (a_0 \min, \max]$, from which we define

$$g_1(i) = \begin{cases} a_0 \min & \text{if } 0 \leq i < 1, \\ \rho \min & \text{if } i \geq 1. \end{cases}$$

Therefore, a_0 needs to be chosen such that $CR(g_1)$ is minimized. We have

$$\sup_{D \in [\min, a_0 \min]} \phi(g_1, D)/D = 2a_0 + 1 \quad \text{and} \quad \sup_{D \in (a_0 \min, \rho \min]} \phi(g_1, D)/D = 3 + 2\frac{\rho}{a_0} \ .$$

Hence, to minimize $CR(g_1)$, we must select a_0, where $1 \leq a_0 \leq \rho$, such that $2a_0 + 1 = 3 + 2\rho/a_0$. Therefore, $a_0 = (1 + \sqrt{1 + 4\rho})/2$ and $CR(g_1) = 2 + \sqrt{1 + 4\rho}$. We have that $CR(g_0) \leq CR(g_1)$ if and only if $1 \leq \rho \leq 2$.

In general, we can partition the interval $[\min, \rho \min]$ into $n + 1$ subintervals whose endpoints correspond to the sequence $\min, a_0 \min, \ldots, a_{n-1} \min, \rho \min$, from which we define

$$g_n(i) = \begin{cases} a_i \min & \text{if } 0 \leq i < n, \\ \rho \min & \text{if } i \geq n. \end{cases}$$

Therefore, we must select $a_0, \ldots, a_{n-1}$, where $1 \leq a_0 \leq a_1 \leq \ldots \leq a_{n-1} \leq \rho$, such that $CR(g_n)$ is minimized. We have

$$\sup_{D \in [\min, a_0 \min]} \phi(g_n, D)/D = 2a_0 + 1 \ ,$$

$$\sup_{D \in (a_i \min, a_{i+1} \min]} \phi(g_n, D)/D = 1 + 2 \sum_{k=0}^{i+1} a_k / a_i \quad (1 \leq i \leq n-2),$$

$$\sup_{D \in (a_{n-1} \min, \rho \min]} \phi(g_n, D)/D = 1 + 2 \sum_{k=0}^{n-1} a_k / a_{n-1} + 2\rho / a_{n-1} \ .$$

Hence, the values a_i are solutions to the following system of equations:

$$\sum_{k=0}^{i+1} a_k = a_0 a_i \quad (0 \leq i \leq n-2), \quad \text{and} \quad \sum_{k=0}^{n-1} a_k + \rho = a_0 a_{n-1}. \tag{1}$$

We prove in Theorem 1 that the solution to this system of equations can be obtained using the following family of polynomials:

$$\begin{aligned} p_0(x) &= x \ , \\ p_1(x) &= x(x-1) \ , \\ p_i(x) &= x\left(p_{i-1}(x) - p_{i-2}(x)\right) \qquad (i \geq 2). \end{aligned} \tag{2}$$

We apply (2) without explicitly referring to it when we manipulate the polynomials p_i. Let α_i denote the largest real root of p_i for each i.

Theorem 1. *For all $n \in \mathbb{N}$, the values a_i $(0 \leq i < n)$ that define g_n satisfy the following properties:*

1. *$a_i = p_i(a_0)$,*
2. *a_0 is the unique solution to the equation $p_n(x) = \rho$ such that $a_0 > \alpha_n$, and*
3. *$CR(g_n) = 2a_0 + 1$.*

To prove Theorem 1, we use the following two formulas:

$$p_{n+1}(x) = x p_n(x) - \sum_{i=0}^{n} p_i(x), \text{ and} \tag{3}$$

$$p_n(x) = x^{\lfloor (n+1)/2 \rfloor} \prod_{k=1}^{\lfloor (n+2)/2 \rfloor} \left(x - 4\cos^2(k\pi/(n+2))\right). \tag{4}$$

Equation (3) can be proved by induction on n. Equation (4) is a direct consequence of Corollary 10 in [11] since the p_n's are *generalized Fibonacci polynomials* (refer to [11]). We can deduce many properties of the p_n's from (4) since it provides an exact expression for all the roots of the p_n's. For instance, we have the formula $\alpha_n = 4\cos^2(\pi/(n+2))$.

Proof. 1. We can prove this theorem by induction on i, using (1) and (3).
2. From the discussion preceding Theorem 1 we know that a_0 satisfies $\sum_{k=0}^{n-1} a_k + \rho = a_0 a_{n-1}$. Therefore,

$$\begin{aligned} \rho &= a_0 a_{n-1} - \sum_{k=0}^{n-1} a_k \\ &= a_0 p_{n-1}(a_0) - \sum_{k=0}^{n-1} p_k(a_0) && \text{by Theorem 1-1,} \\ &= p_n(a_0) && \text{by (3).} \end{aligned}$$

Suppose $a_0 < \alpha_n$. Then, by (4), there exists an $i \in \mathbb{N}$ such that $0 \leq i < n$ and $p_i(a_0) < 0$. Hence, $a_i = p_i(a_0) < 0$ by Theorem 1-1. This is impossible since all the a_i's are such that $1 \leq a_i \leq \rho$. Therefore, $a_0 \geq \alpha_n$. Moreover, $a_0 \neq \alpha_n$ since $p_n(a_0) = \rho \geq 1$, whereas $p_n(\alpha_n) = 0$ by the definition of α_n. Finally, this solution is unique since α_n is the biggest real root.
3. This follows directly from the discussion preceding Theorem 1. □

From Theorem 1, the strategy g_n is uniquely defined for each n. However, this still leaves an infinite number of possibilities for the optimal strategy (one for each n). We aim to find, for a given ρ, what value of n leads to the optimal strategy. Theorem 2 gives a criterion for the optimal n in terms of ρ together with a formula that enables to compute this optimal n in $O(1)$ time.

Theorem 2.

1. *For a given ρ, if $n \in \mathbb{N}$ is such that*

$$p_n(\alpha_{n+1}) \leq \rho < p_n(\alpha_{n+2}) \ , \tag{5}$$

then g_n is the optimal strategy and $\alpha_{n+1} \leq a_0 < \alpha_{n+2}$.
2. *For all $n \in \mathbb{N}$,*

$$2^n \leq p_n(\alpha_{n+1}) \leq \rho < p_n(\alpha_{n+2}) \leq 2^{n+2} \ . \tag{6}$$

Notice that the criterion in Theorem 2-1 covers all possible values of ρ since $p_0(\alpha_1) = 1$ and $p_n(\alpha_{n+2}) = p_{n+1}(\alpha_{n+2})$ by the definition of the α_n's.

Proof. 1. Consider the strategy g_n. By Theorem 1-2 and since $p_n(\alpha_{n+1}) \leq \rho < p_n(\alpha_{n+2})$, we have $\alpha_{n+1} \leq a_0 < \alpha_{n+2}$.

We first prove that g_n is better than g_m for all $m < n$. Suppose that there exists an $m < n$ such that g_m is better than g_n for a contradiction. By Theorem 1-2, there exists an a_0' such that $a_0' > \alpha_m$ and $g_m(a_0') = \rho$. Moreover, since g_m is better than g_n by the hypothesis, then $2a_0' + 1 < 2a_0 + 1$ by Theorem 1-3. Therefore,

$$\alpha_m < a_0' < a_0 \ . \tag{7}$$

Also, since $m < n$, then $m+2 \leq n+1$. Thus, since the α_n's are strictly increasing with respect to n, $a_0 \geq \alpha_{n+1} \geq \alpha_i$ for all $m+2 \leq i \leq n+1$. Hence, we find

$$p_m(a_0) \leq p_{m+1}(a_0) \leq p_{m+2}(a_0) \leq \ldots \leq p_{n-1}(a_0) \leq p_n(a_0) \ . \tag{8}$$

But then,

$$\begin{aligned}
\rho &= p_m(a_0') \\
&< p_m(a_0) && \text{by (7) and since } p_m \text{ is increasing on } [\alpha_m, \infty), \\
&\leq p_n(a_0) && \text{by (8)}, \\
&= \rho \ ,
\end{aligned}$$

which is a contradiction. Consequently, g_n is better than g_m for all $m < n$.

We now prove that g_n is better than $g_{n'}$ for all $n' > n$. Suppose that there exists an $n' > n$ such that $g_{n'}$ is better than g_n for a contradiction. By Theorem 1-2, there exists an a'_0 such that $a'_0 > \alpha_{n'}$ and $g_{n'}(a'_0) = \rho$. Moreover, since $g_{n'}$ is better than g_n by the hypothesis, then $2a'_0 + 1 < 2a_0 + 1$ by Theorem 1-3. Therefore,

$$\alpha_n < \alpha_{n'} < a'_0 < a_0 < \alpha_{n+2} \tag{9}$$

since the α_n's are strictly increasing with respect to n, from which $n' = n + 1$. But then,

$$\begin{aligned}
\rho &= p_{n'}(a'_0) \\
&= p_{n+1}(a'_0) \\
&< p_n(a'_0) && \text{by (9) and since } \alpha_n < \alpha_{n+1} < \alpha_{n+2}, \\
&< p_n(a_0) && \text{by (9) and since } p_n \text{ is increasing on } [\alpha_n, \infty), \\
&= \rho \ ,
\end{aligned}$$

which is a contradiction. Consequently, g_n is better than $g_{n'}$ for all $n' > n$.
2. By standard calculus, we can prove that

$$2 \cos^{n+1}(\pi/(n+3)) \geq 1 \tag{10}$$

for all $n \geq 0$. Therefore,

$$\begin{aligned}
2^n &\leq 2^n \, 2 \cos^{n+1}(\pi/(n+3)) && \text{by (10)}, \\
&= \alpha_{n+1}^{(n+1)/2} && \text{by (4)}, \\
&= p_n(\alpha_{n+1}) && \text{this can be proved by induction on } n, \\
&\leq \rho && \text{by (5)}, \\
&< p_n(\alpha_{n+2}) && \text{by (5)}, \\
&= \alpha_{n+2}^{(n+2)/2} && \text{this can be proved by induction on } n, \\
&= 2^{n+2} \cos^{n+2}(\pi/(n+4)) && \text{by (4)}, \\
&\leq 2^{n+2} && \text{since } 0 < \cos(\pi/(n+4)) < 1. \qquad \square
\end{aligned}$$

From (5), there is only one possible optimal value for n. By (6), it suffices to examine two values to find the optimal n, namely $\lfloor \log_2 \rho \rfloor - 1$ and $\lfloor \log_2 \rho \rfloor$. To compute the optimal n, let $n = \lfloor \log_2 \rho \rfloor$ and let $\gamma = 2\cos(\pi/(n+3))$. If $n+1 \leq \log_\gamma \rho$, then n is optimal. Otherwise, take $n = \lfloor \log_2 \rho \rfloor - 1$. By Theorem 2, this gives us the optimal n in $O(1)$ time.

Now that we know the optimal n, we need to compute a_i for each $0 \leq i < n$. Suppose that we know a_0. By (2) and Theorem 1-1, $a_1 = p_1(a_0) = a_0(a_0 - 1)$ and $a_i = a_0(p_{i-1}(a_0) - p_{i-2}(a_0)) = a_0(a_{i-1} - a_{i-2})$ for $2 \leq i < n$. Therefore, given a_0, each a_i can be computed in $O(1)$ time for $1 \leq i < n$. It remains to show how to compute a_0 efficiently. Since g_n is defined by n values, $\Omega(n) = \Omega(\log \rho)$ time is necessary to compute g_n. Hence, if we can compute a_0 in $O(1)$ time, then our algorithm is optimal.

By Theorem 2, for a given n, we need to solve a polynomial equation of degree $n+1$ to find the value of a_0. By Galois theory, this cannot be done by radicals if $n+1 > 4$. Moreover, the degree of the p_n's is unbounded, so a_0 cannot be computed exactly in general. Theorem 3 explains how and why numerical methods can be used efficiently to address this issue.

Theorem 3. *Take ρ and n such that g_n is optimal for ρ.*

1. *Let $a_0^* \in \mathbb{R}$ be such that $\alpha_{n+1} \le a_0 < a_0^* \le \alpha_{n+2}$ and define g_n^* by*

$$g_n^*(i) = \begin{cases} a_0^* \min & \text{if } i = 0, \\ p_i(a_0^*) \min & \text{if } 1 \le i < n, \\ \rho \min & \text{if } i \ge n. \end{cases}$$

   Then $|CR(g_n) - CR(g_n^*)| \le 7^3\,(n+4)^{-3}$.
2. *The polynomial p_n is strictly increasing on $[\alpha_{n+1}, \alpha_{n+2})$ and $|\alpha_{n+2} - \alpha_{n+1}| \le 7^3\,(n+4)^{-3}/2$.*

Proof. 1. Let $a_i^* = p_i(a_0^*)$. We first prove that $CR(g_n^*) = 2a_0^* + 1$. By Theorems 1 and 2-1, there is a $\rho^* \in \mathbb{R}$ such that $p_n(\alpha_{n+1}) \le \rho < \rho^* \le p_n(\alpha_{n+2})$, $p_n(a_0^*) = \rho^*$ and g_n^* is optimal for ρ^*. By Theorem 1 and the discussion preceding it, we have

$$\begin{aligned}
\sup_{D \in [\min, a_0^* \min]} \frac{1}{D}\phi(g_n^*, D) &= 2a_0^* + 1 \ , \\
\sup_{D \in (a_i^* \min, a_{i+1}^* \min]} \frac{1}{D}\phi(g_n^*, D) &= 1 + 2\sum\nolimits_{k=0}^{i+1} \frac{a_k^*}{a_i^*} && (0 \le i \le n-2) \\
&= 2a_0^* + 1 && (0 \le i \le n-2) \ , \\
\sup_{D \in (a_{n-1}^* \min, \rho \min]} \frac{1}{D}\phi(g_n^*, D) &= 1 + 2\sum\nolimits_{k=0}^{n-1} \frac{a_k^*}{a_{n-1}^*} + 2\frac{\rho}{a_{n-1}^*} \\
&< 1 + 2\sum\nolimits_{k=0}^{n-1} \frac{a_k^*}{a_{n-1}^*} + 2\frac{\rho^*}{a_{n-1}^*} \\
&= 2a_0^* + 1 \ .
\end{aligned}$$

This establishes that $CR(g_n^*) = 2a_0^* + 1$. Therefore,

$$\begin{aligned}
&|CR(g_n) - CR(g_n^*)| \\
&= |(2a_0 + 1) - (2a_0^* + 1)| && \text{by Theorem 1-3 and since } CR(g_n^*) = 2a_0^* + 1, \\
&= 2(a_0^* - a_0) \\
&\le 2(\alpha_{n+2} - \alpha_{n+1}) && \text{by the hypothesis and Theorem 2-1,} \\
&= 8\left(\cos^2(\pi/(n+4)) - \cos^2(\pi/(n+3))\right) && \text{by (4).} \\
&\le 7^3\,(n+4)^{-3} && \text{by elementary calculus.}
\end{aligned}$$

2. This is a direct consequence of (4) and Theorem 3-1. $\square$

We now explain how to compute a_0. We know the value of the optimal n. From (4) and Theorem 2-1, n satisfies $n+1 \leq 4$ if and only if $\rho \leq 32\cos^5(\pi/7) \approx 18.99761$. In this case, $p_n(x) = \rho$ is a polynomial equation of degree at most 4. Hence, by Theorem 1-2 and elementary algebra, a_0 can be computed exactly and in $O(1)$ time. Otherwise, let $\varepsilon > 0$ be a given tolerance. We explain how to find a solution g^*_{opt}, such that $CR(g^*_{opt}) \leq CR(g_{opt}) + \varepsilon$.

If $n \geq 7\varepsilon^{-1/3} - 4$, then by Theorem 3, it suffices to take $a_0 = \alpha_{n+2}$ to compute an ε-approximation of the optimal strategy. But $\alpha_{n+2} = 4\cos^2(\pi/(n+4))$ by (4). Hence, a_0 can be computed in $O(1)$ time and thus, an ε-approximation of the optimal strategy can be computed in $\Theta(n) = \Theta(\log\rho)$ time. Otherwise, if $4 \leq n < 7\varepsilon^{-1/3} - 4$, then we have to use numerical methods to find the value of a_0. By Theorem 2-1, we need to solve $p_n(x) = \rho$ for $x \in [\alpha_{n+1}, \alpha_{n+2})$. However, by Theorem 3, $|\alpha_{n+2} - \alpha_{n+1}| < 7^3(n+4)^{-3}/2$ and p_n is strictly increasing on this interval. Hence, usual root-finding algorithms behave well on this problem.

Hence, if $n < 4$ or $n \geq 7\varepsilon^{-1/3} - 4$, then our algorithm is optimal. When $4 \leq n < 7\varepsilon^{-1/3} - 4$, then our algorithm's computation time is as fast as the fastest root-finding algorithm.

It remains to provide bounds on $CR(g_n)$ for an optimal n; we present exact bounds in Theorem 4.

Theorem 4.

1. *The strategy g_0 is optimal if and only if $1 \leq \rho < 2$. In this case, $CR(g_0) = 2\rho + 1$. Otherwise, if g_n is optimal ($n \geq 1$), then*
$$8\cos^2\left(\frac{\pi}{\lceil \log_2 \rho \rceil + 1}\right) + 1 \leq CR(g_n) \leq 8\cos^2\left(\frac{\pi}{\lfloor \log_2 \rho \rfloor + 4}\right) + 1 \ . \quad (11)$$
2. *When* max $\to \infty$*, the best strategy tends toward $g_\infty(i) = (2i+4)2^i$* min *($i \geq 0$) and $CR(g_\infty) = 9$.*

Proof. 1. This is a direct consequence of (6), (4), and Theorems 1 and 2-1.
2. Let g_n be the optimal strategy for ρ. When max $\to \infty$, then $\rho \to \infty$ and then, $n \to \infty$ by (6). Hence, by Theorem 2-1 and (4), $4 = \lim_{n\to\infty} \alpha_{n+1} \leq \lim_{n\to\infty} a_0 \leq \lim_{n\to\infty} \alpha_{n+2} = 4$. Thus, when max $\to \infty$, $a_i = p_i(a_0) = p_i(4) = (2i+4)2^i$ by Theorem 1-1 and (4). Hence, $g_n \to g_\infty$. $\square$

The competitive cost of the optimal strategy is $2a_0 + 1$ by Theorem 1-3. Theorem 4-1 gives nearly tight bounds on $2a_0 + 1$. Notice that when $\rho = 1$, i.e., when D is known, then $2a_0 + 1 = 3$ which corresponds to the optimal strategy in this case. From the Taylor series expansion of $\cos^2(\cdot)$ and Theorem 4-1, we have $CR(g_n) = 9 - O(1/\log^2\rho)$ for an optimal n. This is consistent with López-Ortiz and Schuierer' result (see [13]), although our result (11) is exact.

Letting $\rho \to \infty$ corresponds to not knowing any upper bound on D. Thus, Theorem 4-2 provides an alternate proof to the competitive ratio of 9 shown by Baeza-Yates et al. [3]. From Theorems 2 and 4, the optimal solution for a given ρ is unique. This optimal solution tends towards g_∞, suggesting that g_∞ is the canonical optimal strategy when no upper bound is given (rather than the power of two strategy).

3 Searching on m Bounded Concurrent Rays

For $m \geq 2$, when no upper bound is known, Baeza-Yates et al. [3] proved that the optimal strategy has a competitive cost of $1+2\,m^m/(m-1)^{m-1}$. There exist infinitely many strategies that achieve this optimal cost.

Proposition 1. *All the strategies in the following family are optimal:* $f_{a,b}(i) = (ai+b)\,(m/(m-1))^i \min$, *where* $0 \leq a \leq b/m$ *and* $(m/(m-1))^{2-m} \leq b \leq (m/(m-1))^2$.

Notice that for $m = 2$, when a and b are respectively equal to their smallest allowed value, then $f_{a,b}$ corresponds to the power of two strategy of Baeza-Yates et al. (refer to [3]). Moreover, when a and b are respectively equal to their biggest allowed value, then $f_{a,b} = g_\infty$ (refer to Theorem 4-2). This proposition can be proved by a careful computation of $CR(f_{a,b})$. For a general m, we let g_∞ be the strategy such that a and b are respectively equal to their biggest allowed value.

When we are given an upper bound $\max \geq D$, the solution presented in Section 2 partially applies to the problem of searching on m concurrent bounded rays. In this setting, we start at the crossroads and we know that the target is on one of the m rays at a distance D such that $\min \leq D \leq \max$. Given a strategy $f(i)$, we walk a distance of $f(i)$ on the $(i \bmod m)$-th ray and go back to the crossroads. We repeat for all $i \geq 0$ until we find the target. As in the case where $m = 2$, we can suppose that is the solution is periodic and monotone (refer to Section 2 or see Lemmas 2.1 and 2.2 in [13]).

Unfortunately, we have not managed to push the analysis as far as in the case where $m = 2$ because the expressions in the general case do not simplify as easily. We get the following system of equations by applying similar techniques as in Section 2

$$\begin{aligned} \sum_{k=0}^{i+m-1} a_k &= a_i \sum_{k=0}^{m-2} a_k && (0 \leq i \leq n-m), \\ \sum_{k=0}^{n-1} a_k + (i-(n-m))\rho &= a_i \sum_{k=0}^{m-2} a_k && (n-m+1 \leq i \leq n-1), \end{aligned}$$

for g_n, where

$$g_n(i) = \begin{cases} a_i \min & \text{if } 0 \leq i < n, \\ \rho \min & \text{if } i \geq n. \end{cases}$$

We prove in Theorem 5 that the solution to this system of equations can be obtained using the following family of polynomials in $m-1$ variables, where $\overline{x} = (x_0, x_1, ..., x_{m-2})$ and $|\overline{x}| = x_0 + x_1 + ... + x_{m-2}$.

$$\begin{aligned} p_n(\overline{x}) &= x_n && (0 \leq n \leq m-2) \\ p_{m-1}(\overline{x}) &= |\overline{x}|(x_0 - 1) && \\ p_n(\overline{x}) &= |\overline{x}|(p_{n-(m-1)}(\overline{x}) - p_{n-m}(\overline{x})) && (n \geq m) \end{aligned}$$

In the rest of this section, for all $n \in \mathbb{N}$, we let $\overline{\alpha}_n = (\alpha_{n,0}, \alpha_{n,1}, ..., \alpha_{n,m-2})$ be the (real) solution to the system

$$p_n(\overline{x}) = 0, \quad p_{n+1}(\overline{x}) = 0, \quad \ldots, \quad p_{n+m-2}(\overline{x}) = 0$$

such that

$$0 \leq \alpha_{n,0} \leq \alpha_{n,1} \leq \cdots \leq \alpha_{n,m-2} \tag{12}$$

and $|\overline{\alpha}_n|$ is maximized. Notice that $\overline{\alpha}_n$ exists for any $n \in \mathbb{N}$ since $(0, 0, ..., 0)$ is a solution for any $n \in \mathbb{N}$ by the definition of the p_n's. The proof of the following theorem is similar to those of (3) and Theorem 1.

Theorem 5.

1. *For all $n \in \mathbb{N}$, the values a_i $(0 \leq i < n)$ that define g_n satisfy the following properties.*
   (a) $a_i = p_i(\overline{a})$.
   (b) $\overline{a}$ is a solution to the system of equations

$$p_n(\overline{x}) = \rho, \quad p_{n+1}(\overline{x}) = \rho, \quad \ldots, \quad p_{n+(m-2)}(\overline{x}) = \rho.$$

   (c) $CR(g_n) = 1 + 2|\overline{a}|$.
2. *The strategy g_0 is optimal if and only if $1 \leq \rho \leq m/(m-1)$. In this case, $CR(g_0) = 2(m-1)\rho + 1$.*
3. *For all $n \in \mathbb{N}$, $p_{n+m-1}(x) = p_n(\overline{x}) \sum_{i=0}^{m-2} x_i - \sum_{i=0}^{n+m-2} p_i(\overline{x})$.*
4. *For all $n \in \mathbb{N}$, $p_n\left(g_\infty(0), g_\infty(1), ..., g_\infty(m-2)\right) = g_\infty(n)$.*
5. *For all $0 \leq n \leq m-2$, $\overline{\alpha}_n = (0, 0, ..., 0)$. Moreover, $\overline{\alpha}_{m-1} = (1, 1, ..., 1)$ and $\overline{\alpha}_m = (m/(m-1), m/(m-1), ..., m/(m-1))$.*

4 Conclusion

We have generalized many of our results for searching on a line to the problem of searching on m rays for any $m \geq 2$. Even though we could not extend the analysis of the polynomials p_n as far as was possible for the case where $m = 2$, we believe this to be a promising direction for future research. By approaching the problem directly instead of studying the inverse problem (maximal reach), we were able to provide exact characterizations of g_{opt} and $CR(g_{opt})$. Moreover, the sequence of implications in the proofs of Section 2 all depend on (4), where (4) is an exact general expression for all roots of all equations p_n. As some readers may have observed, exact values of the roots of the equation p_n are not required to prove the results in Section 2; we need disjoint and sufficiently tight lower and upper bounds on each of the roots of p_n. In the case where $m > 2$, finding a factorization similar to (4) appears highly unlikely. We believe, however, that establishing good bounds for each of the roots of the p_n should be possible. Equipped with such bounds, the general problem could be solved exactly on $m > 2$ concurrent rays. We conclude with the following conjecture. It states that the strategy g_n is uniquely defined for each n, it gives a criterion for the optimal n in terms of ρ (and m) and gives the limit of g_n when max $\to \infty$.

Conjecture 1.

1. For all $n \in \mathbb{N}$, the system of equations of Theorem 5-1b has a unique solution $\overline{a}^* = (a_0^*, a_1^*, ..., a_{m-2}^*)$ satisfying (12) and such that $|\overline{a}^*| > |\overline{\alpha}_n|$. Moreover, there is a unique choice of $\overline{a}$ for g_n and this choice is $\overline{a} = \overline{a}^*$.
2. For a given ρ, if $p_n(\overline{\alpha}_{n+m-1}) \leq \rho < p_n(\overline{\alpha}_{n+m})$, then g_n is the best strategy and $|\overline{\alpha}_{n+m-1}| \leq |\overline{a}| < |\overline{\alpha}_{n+m}|$.
3. When max $\rightarrow \infty$, then the optimal strategy tends toward g_∞.
4. For all $n \in \mathbb{N}$, $0 \leq |\overline{\alpha}_n| \leq |\overline{\alpha}_{n+1}| < m^m/(m-1)^{m-1}$ with equality if and only if $0 \leq n \leq m-3$.

References

1. S. Alpern, V. Baston, and S. Essegaier. Rendezvous search on a graph. *J. App. Prob.*, 36(1):223–231, 1999.
2. S. Alpern and S. Gal. *The Theory of Search Games and Rendezvous.* International Series in Operations Research & Management Science. Kluwer Academic Publishers, 2003.
3. R.A. Baeza-Yates, J.C. Culberson, and G.J.E. Rawlins. Searching in the plane. *Inf. & Comp.*, 106(2):234–252, 1993.
4. R. Bellman. Minimization problem. *Bull. AMS*, 62(3):270, 1956.
5. M.A. Bender, A. Fernández, D. Ron, A. Sahai, and S.P. Vadhan. The power of a pebble: Exploring and mapping directed graphs. In *STOC*, pages 269–278, 1998.
6. A. Collins, J. Czyzowicz, L. Gasieniec, and A. Labourel. Tell me where I am so I can meet you sooner. In *ICALP*, pages 502–514, 2010.
7. J. Czyzowicz, D. Ilcinkas, A. Labourel, and A. Pelc. Asynchronous deterministic rendezvous in bounded terrains. *Theor. Comp. Sci.*, 412(50):6926–6937, 2011.
8. Y. Dieudonné and A. Pelc. Anonymous meeting in networks. In *SODA*, pages 737–747, 2013.
9. M. Hammar, B.J. Nilsson, and S. Schuierer. Parallel searching on m rays. *Comput. Geom.*, 18(3):125–139, 2001.
10. C.A. Hipke, C. Icking, R. Klein, and E. Langetepe. How to find a point on a line within a fixed distance. *Disc. App. Math.*, 93(1):67–73, 1999.
11. V. Hoggatt and C. Long. Divisibility properties of generalized fibonacci polynomials. *Fibonacci Quart.*, 12(2):113–120, 1974.
12. E. Koutsoupias, C.H. Papadimitriou, and M. Yannakakis. Searching a fixed graph. In *ICALP*, pages 280–289, 1996.
13. A. López-Ortiz and S. Schuierer. The ultimate strategy to search on m rays? *Theor. Comp. Sci.*, 261(2):267–295, 2001.
14. G. De Marco, L. Gargano, E. Kranakis, D. Krizanc, A. Pelc, and U. Vaccaro. Asynchronous deterministic rendezvous in graphs. *Theor. Comp. Sci.*, 355(3):315–326, 2006.
15. F.P. Preparata and M.I. Shamos. *Computational Geometry - An Introduction.* Springer, 1985.