Searching on m Bounded Rays Optimally*

Sven Schuierer†

Abstract

We consider the problem of searching on m current rays for a target of unknown location. It is well-known that the optimal strategy to solve this problem achieves a competitive ratio of $C_m = 1 + 2m^m/(m-1)^{m-1}$. We investigate the case where the target is known to be within a fixed distance D of the origin. We present an algorithm to compute the optimal competitive ratio C_m^D that can be achieved by a deterministic on-line strategy.

1 Introduction

Searching for a target is an important and well studied problem in robotics. In many realistic situations the robot does not possess complete knowledge about its environment, for instance, the robot may not have a map of its surroundings, or the location of the target may be unknown [DI94, IK95, Kle92, LOS95, PY89].

Since the robot has to make decisions about the search based only on the part of its environment that it has explored before, the search of the robot can be viewed as an *on-line* problem. One way to judge the performance of an on-line search strategy is to compare the distance traveled by the robot to the length of the shortest path from its starting point s to the target t. The ratio of the distance traveled by the robot to the optimal distance from s to t over all possible locations of the target is called the *competitive ratio* of the search strategy [ST85].

We are interested in obtaining upper and lower bounds on the competitive ratio of searching on m concurrent rays. Here, a point robot is imagined to stand at the origin of m rays and one of the rays contains the target t whose distance to the origin is unknown. The robot can only detect t if it stands on top of it. It can be shown that an optimal strategy visits the rays in cyclic order and increases the step length each time by a factor of $m/(m-1)$ starting with a step length of 1 [BYCR93, Gal80]. The competitive ratio C_m achieved by this strategy is given by $1+2m^m/(m-1)^{m-1}$. If randomization is used, the optimal competitive ratio is given by the minimum of the function $1 + 2a^m/((a-1)\ln a)$, for $a > 1$ [Gal80, KRT97, KMSY94].

*This research is supported by the DFG-Project "Diskrete Probleme", No. Ot 64/8-2.

†Institut für Informatik, Am Flughafen 17, Geb. 051, D-79110 Freiburg, Germany
email: schuiere@informatik.uni-freiburg.de

Searching on m rays has proven to be a very useful tool for searching in a number of classes of simple polygons, such as star-shaped polygons [LOS97], generalized streets [DI94, LOS96], HV-streets [DHS95], and θ-streets [DHS95, Hip94].

Hipke *et al.* consider the maximal reach of a strategy to search on the line if the competitive ratio of the strategy is given [HIKL97]. The *reach* of a strategy X with competitive ratio C is the maximum distance D such that a target placed at a distance D to the origin is still detected by a robot using X. Since C is given, a recurrence equation for the optimal reach can be derived. Using this recurrence equation Hipke *et al.* show that the maximal reach is continuous and strictly monotone in C. This in turn implies that C_2^D is strictly monotone in D and assumes all values in the interval $[3, 9]$.

In this paper we present an algorithm to compute the maximal reach for a given competitive ratio C and arbitrary m—thus, generalizing the results by Hipke *et al.* [HIKL97].

The paper is organized as follows. In the next section we give the basic definitions concerning searching on m rays. In Section 2 we show that an optimal strategy to search on m bounded rays visits the rays in a fixed cyclic order. We also derive a recurrence equation that is satisfied by an optimal strategy. In Section 3 we present an algorithm to compute the strategy with maximal reach for a given competitive ratio C.

2 Searching on m Rays with Competitive Ratio C

We are interested in computing the maximal reach of a strategy for a given competitive ratio C. Let X be a strategy to search on m bounded rays. Strategy X proceeds in *steps*. In each step the robot travels on one ray to a certain distance and, if it does not find the target, then it returns to the origin. Let x_i be the maximal distance to the origin and r_i the ray visited in Step i.

We define J_i as the index of the step in which ray r_i is visited the next time after Step i, that is, $J_i = \min\{j > i \mid r_j = r_i\}$. If there is no $j > i$ with $r_j = r_i$, then we define $J_i = i$. We represent X by the sequence of pairs (x_i, J_i).

Assume that the target is discovered in Step J_k in ray r. By the definition of J_k ray r was visited the last time before Step J_k in Step k and the distance d to the target is greater than x_k. The distance traveled by the robot to discover t is $d + 2\sum_{i=0}^{J_k-1} x_i$. Since the target can be placed arbitrarily close to x_k by an adversary, the competitive ratio of Step k is given by

$$\sup_{d>x_k}(d + 2\sum_{i=0}^{J_k-1} x_i)/d = \sup_{d>x_k} 1 + 2\sum_{i=0}^{J_k-1} x_i/d = 1 + 2\sum_{i=0}^{J_k-1} x_i/x_k.$$

The competitive ratio C_X of X is now given as the supremum of the competitive ratios over all steps.

The first step is a special case that we have not considered yet. If no information about the target is available, then one false move in the beginning may lead to an

arbitrarily large competitive ratio. In order to avoid this problem we assume that a lower bound of one for the distance to the target t is known in advance that is, the target may be placed on any of the m rays somewhere in the interval $[1, D]$.

In order to compute the maximal reach, we first show that an optimal strategy is *periodic*. A strategy is *periodic* if $r_{k+m} = r_k$, for all $0 \leq k \leq n-m$.

Lemma 2.1 *If X is a strategy to search on m rays for a target with competitive ratio C_X and reach D_X, then there is a periodic strategy X^* with $C_{X^*} = C_X$ and $D_{X^*} \geq D_X$.*

Proof: In [LOS98, Lemma 3.2] it is shown that there is a periodic strategy X^* with $C_{X^*} \leq C_X$ and $D_X = D_{X^*}$. If $C_{X^*} = C_X$, then this proves the claim. If $C_{X^*} < C_X$, then we can increase $x^*_{n-m+1}, \ldots, x^*_n$ by an amount of ε until $C_{X^*} = C_X$. This increases the reach of X^*. □

In the following we assume that X is an optimal periodic strategy. It can be easily seen that the competitive simplifies in this case to $C_X = \max_{0 \leq i \leq n-m} 1 + 2\sum_{i=0}^{k+m-1} x_i/x_k$. We now show that the values x_i satisfy a recurrence equation.

Lemma 2.2 *If X is a strategy with maximal reach, then $1 + 2\sum_{i=0}^{k+m-1} x_i/x_k = C_X$, for all $0 \leq k \leq n-m$.*

Proof: The proof is by contradiction. Let $C_X = 1 + 2c$. If $\sum_{i=0}^{k+m-1} x^*_i < cx^*_k$, for some $0 \leq k \leq n-m$, then we can decrease x^*_k by some amount $\varepsilon > 0$ and increase $x^*_{n-m+1}, \ldots, x^*_n$ by ε/m, thus achieving a greater reach. □

In the following let $c_X = (C_X - 1)/2$. Lemma 2.2 implies that the step lengths x_i of an optimal strategy X satisfy the following recurrence equation.

$$\frac{\sum_{i=0}^{k+m-1} x_i}{x_k} = c_X \qquad \text{or} \qquad \sum_{i=0}^{k+m-1} x_i = c_X x_k, \tag{1}$$

for $0 \leq k \leq n-m$. An additional constraint is given by the first time the m-th ray is visited; here, the competitive ratio is given by

$$1 + 2\sum_{i=0}^{m-2} x_i \leq 1 + 2c_X \tag{2}$$

as in steps $0, \ldots, m-2$ the first $m-1$ rays are explored. In fact, if $\sum_{i=0}^{m-2} x_i < c$, then there is a $\lambda > 1$ such that the strategy λX satisfies Equation 1 and $\sum_{i=0}^{m-2} \lambda x_i = c$. If D is the reach of X, then $\lambda D > D$ is the reach of λX. Hence, we can assume that we have equality in Equation 2 for a strategy with maximal reach. If we multiply c_X by $x_{-1} = 1$, then we can view (2) as a special case of (1) for $k = -1$. Hence, we assume in the following that Equation 1 holds for all $-1 \leq k \leq n-m$.

A simple consequence of Lemma 2.2 and the proof of Lemma 2.1 is the following result.

Corollary 2.3 *If $D^*_m(C)$ is the maximal reach of a strategy with competitive ratio C to search on m rays, then D^*_m is a strictly increasing function of C.*

Corollary 2.3 in turn implies that if C_m^D is the minimal competitive ratio to search for a target on m rays that is placed at a distance of at most D, then C_m^D is a strictly increasing function in D.

The linear equation system (1) consists of $n-m+2$ linearly independent equations in the $n+1$ unknowns $(x_0, x_1, \ldots, x_{n-1})$ and D (x_n is irrelevant since x_n does not appear in Equation 1). In addition we are given the $m-1$ equations $x_{n-m+1} = \cdots = x_{n-1} = D$; hence, the $n+1$ solutions $(x_0, x_1, \ldots, x_{n-1}, D)$ are uniquely defined once we are given c_X, and n. We are interested in the question how large D can be for a given c_X such that there is an n and a *positive* solution $(x_0, x_1, \ldots, x_{n-1})$. As this question seems to be rather difficult to answer, we transform Equation 1 into a simpler form.

Lemma 2.4 *The values x_i satisfy the following recurrence equation*

$$x_{k+m-1} - c_X x_k + c_X x_{k-1} = 0, \tag{3}$$

for $0 \le k \le n-m$.

Proof: By Equation 1 we have

$$\sum_{i=0}^{k+m-1} x_i = c_X x_k,$$

for $0 \le k \le n-m$. The same equation also holds for $k-1$. Hence,

$$\sum_{i=0}^{k+m-1} x_i = c_X x_k \qquad \text{and} \qquad \sum_{i=0}^{k+m-2} x_i = c_X x_{k-1}.$$

By subtracting the second equation from the first we obtain Equation 3, for $0 \le k \le n-m$ as claimed. □

Unfortunately, we obtain only $n-m+1$ equations in this way—one too few—and the sequence X is not completely determined anymore by Equation 3 and the $m-1$ additional equations $x_{n-m+1} = \cdots = x_{n-1} = D$.

3 Computing the Optimal Strategy

In this section we present an algorithm to compute the optimal strategy to search on m bounded rays. We assume that we are given the competitive ratio $1+2c$ and we want to compute the maximal *reach* for $1+2c$ [HIKL97]. Note that once we are able to compute the maximum reach, we can easily compute the minimal competitive ratio for a given D by applying binary search. This only increases the running time proportional to the number of bits necessary to represent D.

In the case $m = 2$ it is fairly easy to derive a recurrence equation for the optimal reach [HIKL97]. As we have shown in Section 2 there is a strategy with maximal reach that is periodic and satisfies the equations

$$\sum_{i=0}^{k+m-1} x_i = c\, x_k, \quad (1) \qquad \text{and} \qquad \sum_{i=0}^{m-2} x_i \;=\; c x_{-1} \quad (2),$$

where $x_{-1} = 1$. For $m = 2$, this implies that $x_0 = c$ and x_k is given by

$$x_k = c\, x_{k-1} - \sum_{i=0}^{k-1} x_i,$$

for $k \geq 1$. This determines the strategy completely. For $m > 2$, we still have equality in Equation 2 but now we only obtain the *sum* of the first $m-1$ steps lengths $x_0, \ldots, x_{m-2}$ instead of their values.

Hence, we take a different approach. Let Y be the sequence defined by $y_i = x_{n-i-1}$, for $0 \leq i \leq n-m$. It satisfies recurrence equation

$$y_{k+m} - y_{k+m-1} + \frac{1}{c} y_k \quad = \quad 0, \tag{4}$$

for all $0 \leq k \leq n-m$. However, we only have the initial values for $y_0, \ldots, y_{m-2}$ which are equal to D. Since we need one more initial value we set $\alpha = y_{m-1}/D$, where $0 < \alpha < 1$. Let $Y(c, D, \alpha D) = (y_0, y_1, \ldots)$ be the infinite sequence that is given by Equation 4 and the above initial values. If $c < c_m$, then there is an index n_0 such that y_{n_0} is negative. Lopez and Schuierer show that

$$n_0 \leq 4m^3(3m \log m - \log \varepsilon) + 1 + \frac{2\pi m^{3/2}}{\varepsilon}, \tag{5}$$

where $\varepsilon = \sqrt{m^m/(m-1)^{m-1} - c}$ [LOS98]. We choose n to be the index such that y_n is minimal among $y_0, \ldots, y_{n_0-1}$. The value y_n is now the lower bound on the distance to the target. If we set $x'_k = y_{n-k-1}/y_n$, for $0 \leq k \leq n$, then we obtain a strategy with a lower bound of $x - 1]' = y_n/y_n = 1$ to the target and reach D/y_n, which—by the choice of y_n—is the largest possible reach for a strategy that satisfies Equations 1 and 2 with the above initial values. Unfortunately, we know neither D nor α. However, we can set $D = 1$ since we are going to scale by $1/y_n$ later anyway.

Since we do not know α, we consider the values y_k as numbers over the extended field $\mathbb{R}[\alpha] = \{x + y\alpha \mid x, y \in \mathbb{R}\}$, that is, α is treated as formal parameter. Hence, $y_k = u_k + \alpha\, v_k$, for some values u_k and v_k. On the other hand, $y_k = y_{k-1} - (1/c)\, y_{k-m}$, for $m \leq k \leq n$. This yields two recurrences for u_k and v_k.

$$u_k = u_{k-1} - (1/c)\, u_{k-m}, \qquad \text{and} \qquad v_k = v_{k-1} - (1/c)\, v_{k-m}.$$

The initial values for the sequence $U = (u_0, u_1, \ldots)$ are now given by $u_0 = \cdots = u_{m-2} = 1$ and $u_{m-1} = 0$. The initial values for the sequence $V = (v_0, v_1, \ldots)$ are given by $v_0 = \cdots = v_{m-2} = 0$ and $v_{m-1} = 1$. If we stop after n steps, then Equation 2 should be satisfied, that is, we require

$$\sum_{i=n-m+1}^{n-1} u_i + \alpha_n v_i = c(u_n + \alpha_n v_n) \quad \text{or} \quad \alpha_n = -\frac{\sum_{i=n-m+1}^{n-1} u_i - c\, u_n}{\sum_{i=n-m+1}^{n-1} v_i - c\, v_n}. \tag{6}$$

We obtain the following algorithm.

Algorithm Maximal Reach
Input: The competitive ratio $1 + 2c$ and the number of rays m.
Output: An integer n and a strategy $X = (x_0, \ldots, x_n)$ such that the reach of X is maximal.

1 **if** $c \geq m^m/(m-1)^{m-1}$ **then return** ∞, $x_k = (1 + 1/(m-1))^k$
2 **for** $i \leftarrow 0$ **to** $m-2$ **do** let $u_i \leftarrow 1$, $v_i \leftarrow 0$
3 let $u_{m-1} \leftarrow 0$, $v_{m-1} \leftarrow 1$

4 let $y_{min} \leftarrow 1$, $n_{min} \leftarrow m$, $\varepsilon \leftarrow (m^m/(m-1)^{m-1} - c)^{1/2}$
5 let $n_\varepsilon \leftarrow 4m^3(3m \log m - \log \varepsilon) + 1 + 2\pi m^{3/2}/\varepsilon$
6 **for** $n \leftarrow m$ **to** n_ε **do**
7 let $u_n \leftarrow u_{n-1} - (1/c)\, u_{n-m}$, $v_n \leftarrow v_{n-1} - (1/c)\, v_{n-m}$
8 let $\alpha_n \leftarrow -(\sum_{i=n-m+1}^{n-1} u_i - c\, u_n)/(\sum_{i=n-m+1}^{n-1} u_i - c\, v_n)$
9 **if** $(u_n + \alpha_n v_n < y_{min})$
then let *positive* $\leftarrow$ true
10 **for** $j \leftarrow m$ **to** n **do**
/* Test if all elements are positive */
11 **if** $u_j + \alpha_n v_j \leq 0$ **then** let *positive* $\leftarrow$ false
12 **if** *positive* **then** $y_{min} \leftarrow y_n$, $n_{min} \leftarrow n$
end if

13 let $n \leftarrow n_{min}$
14 **for** $i \leftarrow 0$ **to** $n-1$ **do** let $x_i \leftarrow (u_{n-i-1} + \alpha_n v_{n-i-1})/y_{min}$
15 **return** n, $(x_0, \ldots, x_{n-1}, 1/y_{min})$

We show the correctness of Algorithm *Maximal Reach* in the following two lemmas.

Lemma 3.1 *The competitive ratio of X is $1 + 2c$.*

Proof: First note that because of the test in Step 10 all the elements of X are positive. By the choice of α, U, and V, X obviously satisfies

$$\sum_{i=0}^{m-2} x_i = c \qquad \text{and} \qquad x_{k+m-1} = c(x_k - x_{k-1})$$

for $0 \leq k \leq n-m$ if we set $x_{-1} = 1$.[1] Using induction we see that

$$\sum_{i=0}^{k+m-1} x_i \;=\; x_{k+m-1} + cx_{k-1} \;=\; c(x_k - x_{k-1}) + cx_{k-1} \;=\; cx_k$$

for $0 \leq k \leq n-m$ as claimed. □

[1] Note that y_n is chosen to satisfy $y_n = y_{n-1} - 1/cy_{n-m}$. Since we divide by y_n, this implies that $y_{n-m}/y_n = c(y_{n-1}/y_n - 1)$ or $x_{m-1} = c(x_0 - x_{-1})$.

Lemma 3.2 *The reach of Strategy X is at least as large as the reach of any other strategy with competitive ratio $1 + 2c$.*

Proof: Let $X^* = (x_0^*, x_1^*, x_2^*, \ldots, x_l^*)$ be a strategy with maximal reach for competitive ratio $1 + 2c$. By Equation 5 $l \leq n_\varepsilon$. As we observed previously, X^* satisfies the conditions $\sum_{i=0}^{k+m-1} x_i^* = cx_k^*$, for $0 \leq k \leq l - m$, and $\sum_{i=0}^{m-2} x_i^* = c$. We define a sequence $Y^* = (y_0^*, \ldots, y_l^*)$ by $y_i^* = x_{l-i-1}^*/x_l^*$, for $0 \leq i \leq l$, where we set $x_{-1}^* = 1$. The reach of X^* is $x_{l-1}^* = x_{l-1}^*/x_{-1}^* = (x_{l-1}^*/x_l^*)/(x_{-1}^*/x_l^*) = 1/y_l^*$.

The sequence Y^* satisfies recurrence equation 4. By a simple induction it can be easily seen that $y_k^* = u_k + y_{m-1}^* v_k$. Because of Equations 2 and 6 we obtain that $y_{m-1}^* = \alpha_l$. Hence, Y^* is computed in Step 10 if $k = l$. Let Y be the sequence computed by Algorithm *Maximal Reach*. Since y_n is chosen to be minimal, $y_n \leq y_l^*$. Hence, the reach $1/y_n$ of X is at least as large as the reach of X^*. □

If c consists of b bits, then the time complexity of Algorithm *Maximal Reach* is quadratic in $n_\varepsilon = \sqrt{2^b}$, that is, $\Theta(2^b)$. On the other hand, since $\log D = \Omega(n_\varepsilon)$,[2] the time complexity is quadratic in the size of the output. It is easy to see that the algorithm can be implemented using only $O(m)$ space.

We have implemented Algorithm *Maximal Reach* in Maple. In Figures 1, 2, 3, and 4 the maximal reach of the optimal strategy, the doubling strategy and of the strategy proposed in [LOS98] is shown for $m = 2$ and $m = 3$. In Figure 5 the reach of the optimal strategy for different values of m is shown. and 5a the maximal reach of the optimal strategies for different values of m is shown. The figure illustrates nicely that the logarithm of the maximal reach depends linearly on $1/\varepsilon$. In Figure 5b we compare the maximal reach of the optimal strategy to the maximal reach of the strategy presented in [LOS98] and the strategy given by $x_k = (3/2)^k$ for $m = 3$. It can be seen that the maximal reach of the optimal strategy increases much faster than that of the other two strategies. The figure also shows that the maximal reach of the strategy presented in [LOS98] is a linear function of $1/\varepsilon$ whereas the maximal reach of the other strategy is a logarithmic function of $1/\varepsilon^2$. It should be noted that the lower bound we have presented—which is now an upper bound on the maximal reach—does not fit into the figure as it starts at a value of > 2000 and has a much larger slope.

4 Conclusions

We present an algorithm to compute the strategy with maximal reach for a given competitive ratio and general m. Our algorithm needs time proportional to the size of the output and exponential in the size of the input.

[2] This follows from the fact that there is a strategy with such that $\log D$ is in $\Omega(1/\varepsilon)$—as, for instance, the strategy presented in [LOS98].

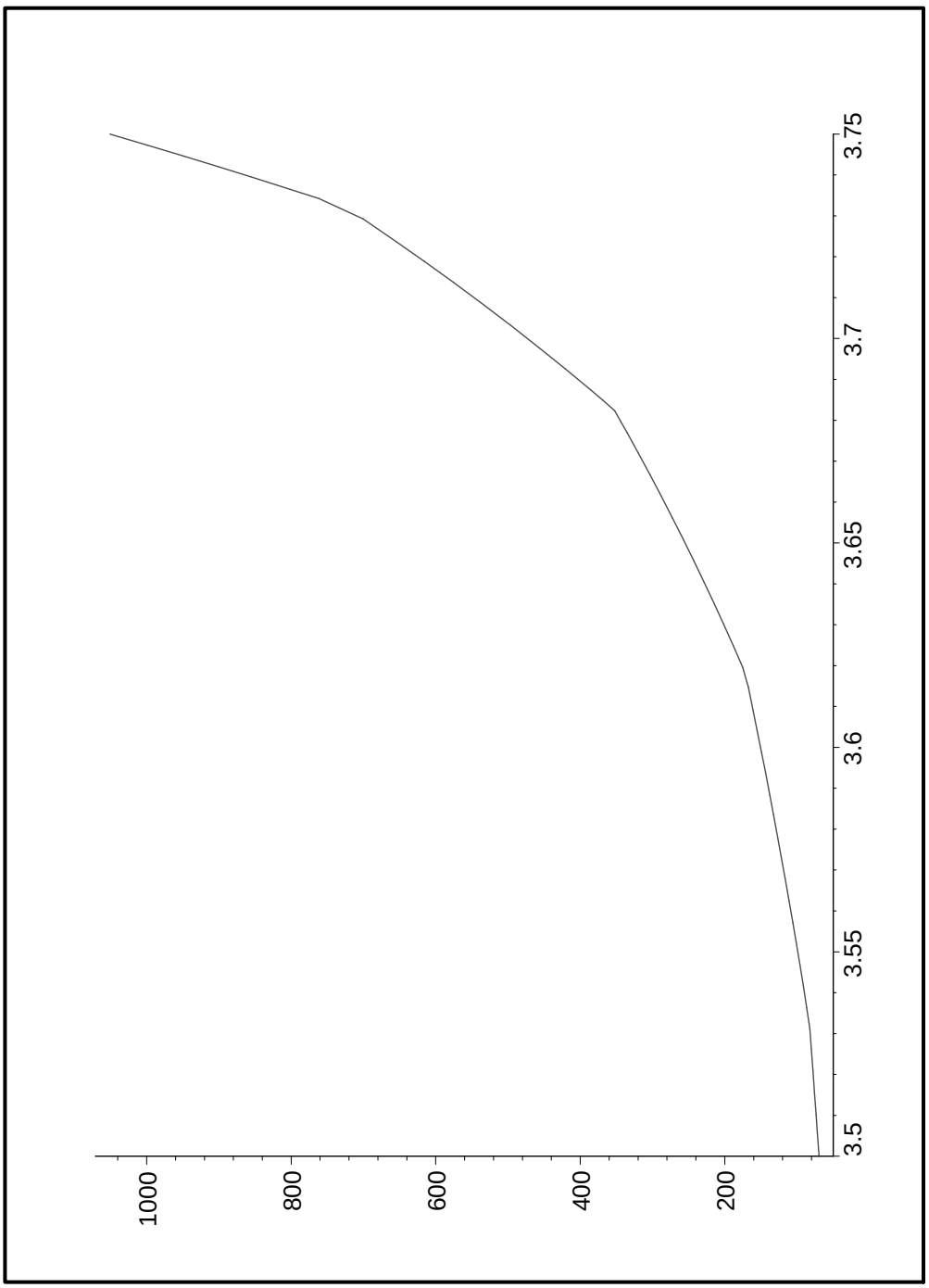

Figure 1: The reach of the optimal strategy for $m = 2$.

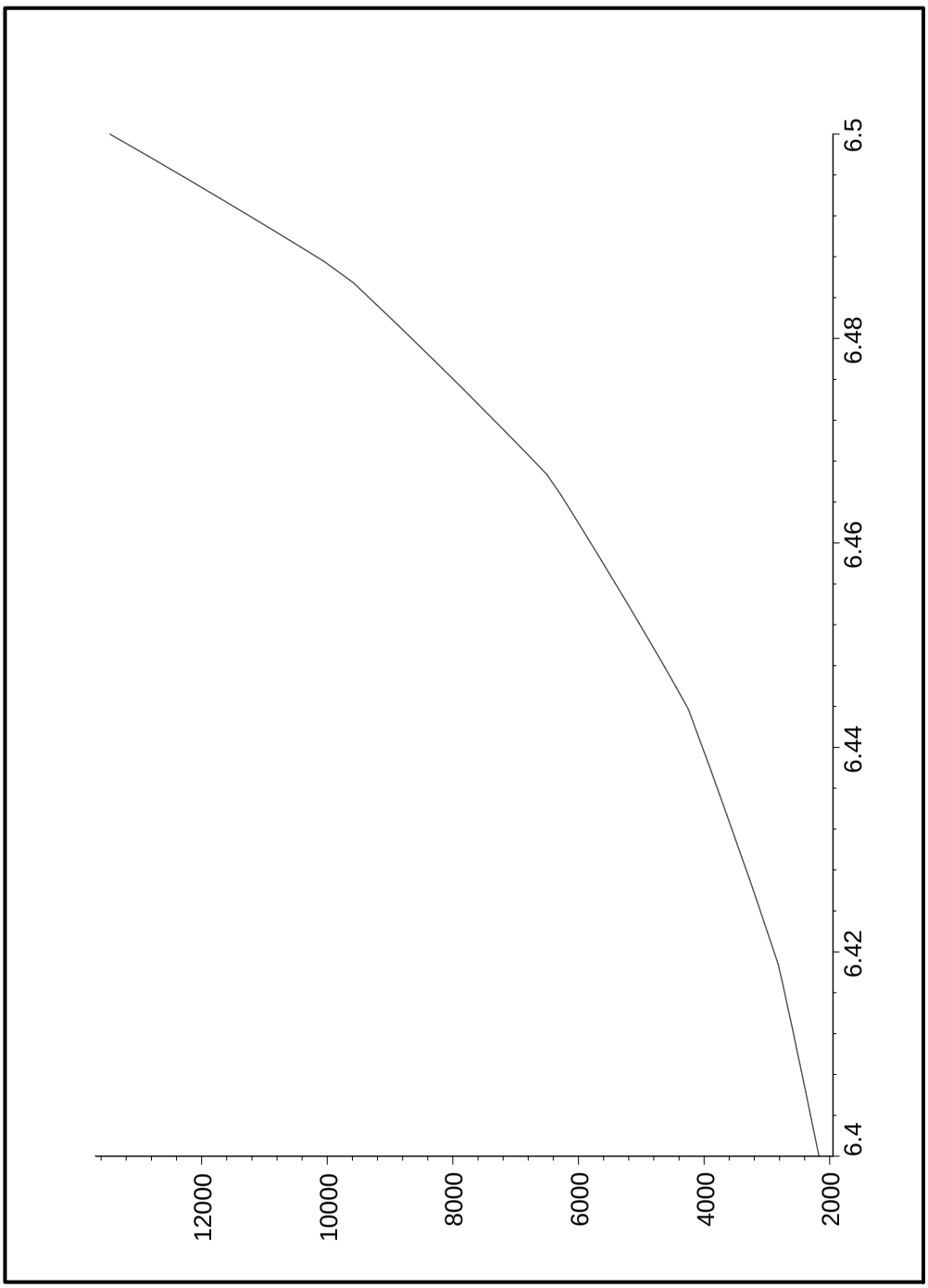

Figure 2: The reach of the optimal strategy for $m = 3$.

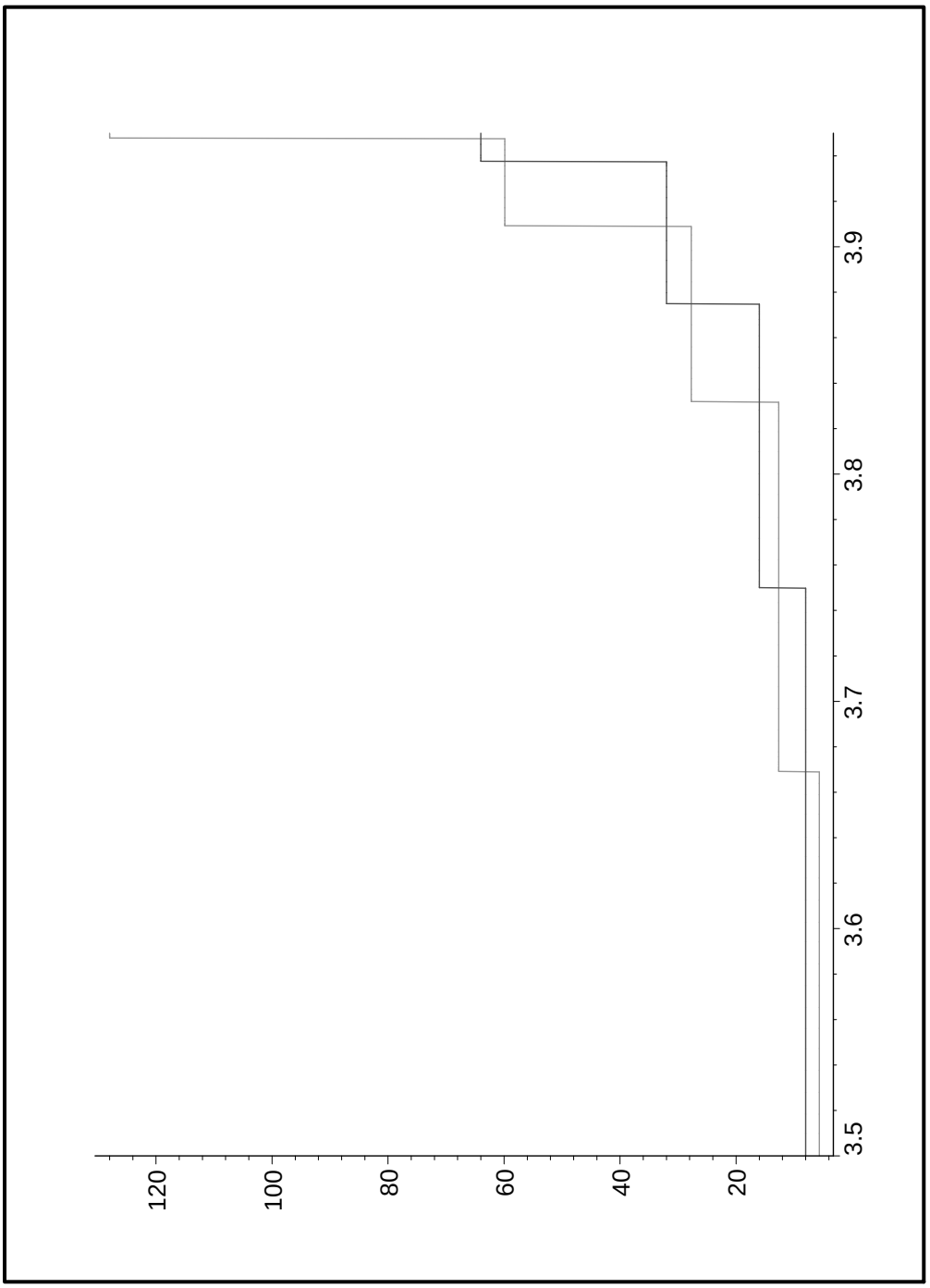

Figure 3: The reach of the doubling strategy and of the strategy proposed in [LOS98] for $m = 2$. The doubling strategy uses the darker line.

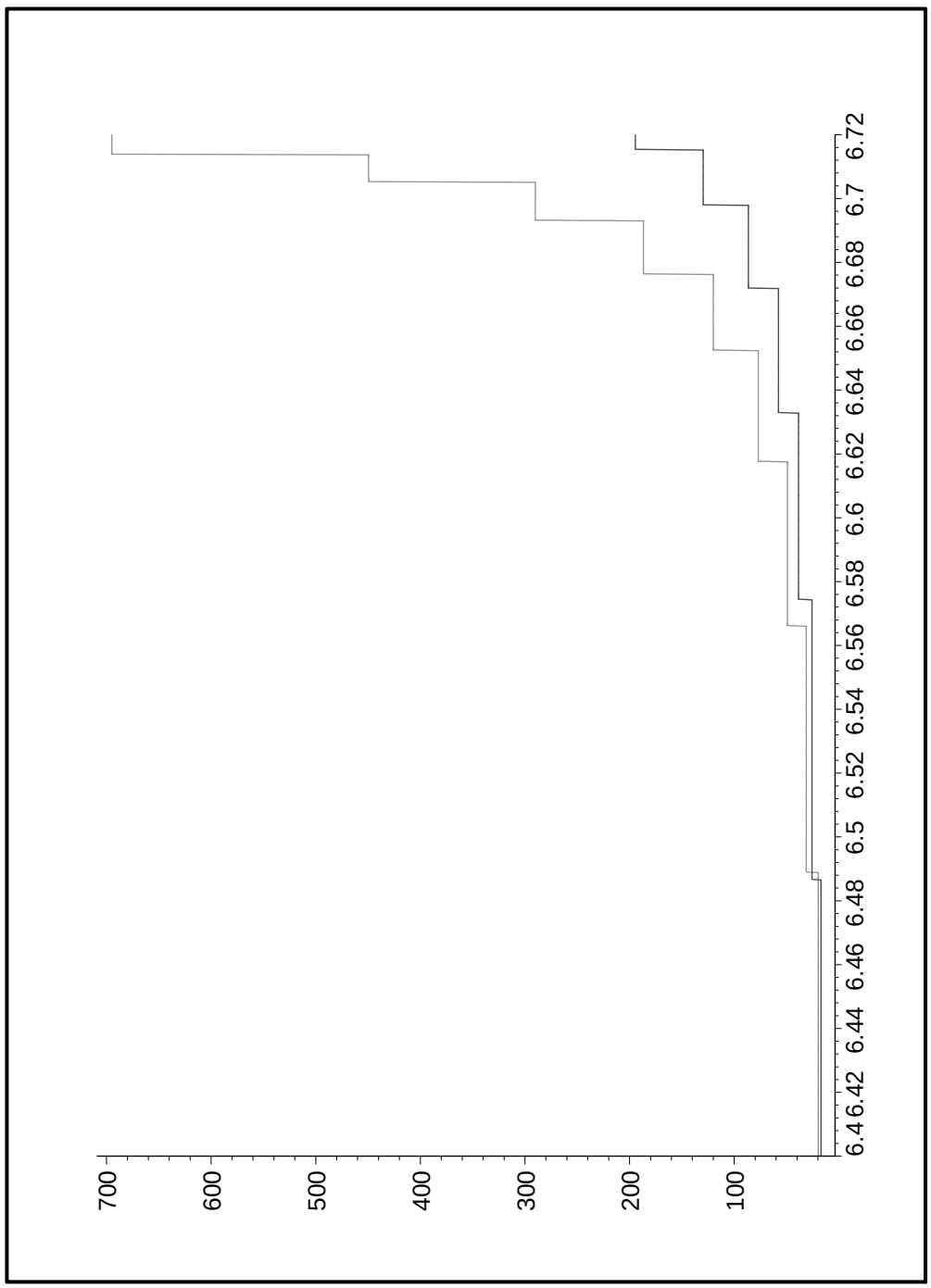

Figure 4: The reach of the doubling strategy and of the strategy proposed in [LOS98] for $m = 3$. The doubling strategy uses the darker line.

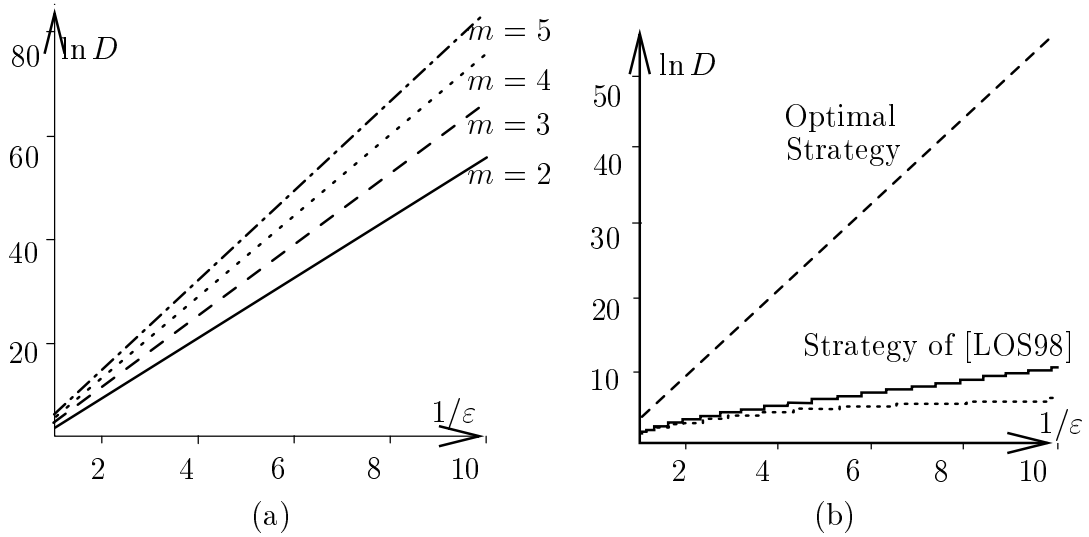

Figure 5: (a) The maximal reach of the optimal strategy for different m. (b) The maximal reach of the optimal strategy for $m = 3$ compared to the maximal reach of the strategy proposed in [LOS98] and the strategy given by $x_k = (3/2)^k$.

References

[BYCR93] R. Baeza-Yates, J. Culberson, and G. Rawlins. Searching in the plane. *Information and Computation*, 106:234–252, 1993.

[DHS95] A. Datta, Ch. Hipke, and S. Schuierer. Competitive searching in polygons—beyond generalized streets. In *Proc. Sixth Annual International Symposium on Algorithms and Computation*, pages 32–41. LNCS 1004, 1995.

[DI94] A. Datta and Ch. Icking. Competitive searching in a generalized street. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 175–182, 1994.

[Gal80] S. Gal. *Search Games*. Academic Press, 1980.

[HIKL97] Ch. Hipke, Ch. Icking, R. Klein, and E. Langetepe. How to find a point on a line within a fixed distance. Informatik-Bericht 220, Fernuni Hagen, November 1997.

[Hip94] Ch. Hipke. Online-Algorithmen zur kompetitiven Suche in einfachen Polygonen. Master's thesis, Universität Freiburg, 1994.

[IK95] Ch. Icking and R. Klein. Searching for the kernel of a polygon: A competitive strategy. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 258–266, 1995.

[Kle92] R. Klein. Walking an unknown street with bounded detour. *Comput. Geom. Theory Appl.*, 1:325–351, 1992.

[KMSY94] M.-Y. Kao, Y. Ma, M. Sipser, and Y. Yin. Optimal constructions of hybrid algorithms. In *Proc. 5th ACM-SIAM Sympos. Discrete Algorithms*, pages 372–381, 1994.

[KRT97] M. Y. Kao, J. H. Reif, and S. R. Tate. Searching in an unknown environment: An optimal randomized algorithm for the cow-path problem. *Information and Computation*, 131(1):63–80, 1997.

[LOS95] A. López-Ortiz and S. Schuierer. Going home through an unknown street. In S. G. Akl, F. Dehne, and J.-R. Sack, editors, *Proc. 4th Workshop on Algorithms and Data Structures*, pages 135–146. LNCS 955, 1995.

[LOS96] A. López-Ortiz and S. Schuierer. Generalized streets revisited. In M. Serna J. Diaz, editor, *Proc. 4th European Symposium on Algorithms*, pages 546–558. LNCS 1136, 1996.

[LOS97] A. López-Ortiz and S. Schuierer. Position-independent near optimal searching and on-line recognition in star polygons. In *Proc. 5th Workshop on Algorithms and Data Structures*, pages 284–296. LNCS 1272, 1997.

[LOS98] A. López-Ortiz and S. Schuierer. The ultimate stragegy to search on m rays? Technical Report 100, Universität Freiburg, Institut für Informatik, www.informatik.uni-freiburg.de/tr/1998/Report100/, May 1998.

[PY89] C. H. Papadimitriou and M. Yannakakis. Shortest paths without a map. In *Proc. 16th Internat. Colloq. Automata Lang. Program.*, volume 372 of *Lecture Notes in Computer Science*, pages 610–620. Springer-Verlag, 1989.

[ST85] D. D. Sleator and R. E. Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28:202–208, 1985.