Online Searching with Turn Cost

Erik D. Demaine[1], Sándor P. Fekete[2], and Shmuel Gal[3]

[1] Laboratory for Computer Science, MIT, Cambridge MA, USA,
edemaine@mit.edu
[2] Department of Mathematical Optimization TU Braunschweig, Braunschweig, Germany,
sandor.fekete@tu-bs.de
[3] Department of Statistics, University of Haifa, Haifa, Israel,
sgal@univ.haifa.ac.il

Abstract. We consider the problem of searching for an object on a line at an unknown distance OPT from the original position of the searcher. This problem is well-studied and has been rediscovered repeatedly: A trajectory achieving optimal competitive ratio of 9 (i.e., finding the object by traveling not more than 9OPT) consists of a geometric series that doubles its search depth at each step. This strategy does not take into account that in practice, there is a cost for changing direction. Another annoying feature is that there is no first step; instead the trajectories start with infinitesimal "wiggling". We show that these problems can be avoided by assuming a fixed turn cost d for changing direction. We describe a strategy that is guaranteed to find the object while traveling at most 9OPT$+2d$, which is optimal in both the competitive ratio 9 and the additive penalty term $2d$. Our argument for upper and lower bound uses an infinite series of Linear Programs, which is interesting in its own right. Using the same approach, we give an analysis for the generalization to star search on m rays (also known as the cow-path problem), for which the resulting generalized bound turns out to be $\left(1+2\frac{m^m}{(m-1)^{m-1}}\right) OPT + m\left(\left(\frac{m}{m-1}\right)^{m-1}-1\right) d$.

MSC Classification: 91A05, 90C05
Keywords: search games, linear search problem, star search, cow-path problem, turn cost, competitive ratio, infinite linear programs.

1 Introduction

Search games. Searching for an object is one of the fundamental issues of everyday life, and also one of the basic algorithmic problem that needs to be mastered in the context of computing [21]. If the object is located in a bounded domain (say, in one of n discrete locations), then the worst-case complexity is obvious: In accordance with our everyday experience, an imaginary "hider" may have placed the object in the very last spot where we are looking for it.

More challenging is the scenario of searching in an unbounded domain. The classical prototype is the *linear search problem* (LSP), which was first proposed by Bellman [7] and, independently, by Beck: An (immobile) object is located on the real line according to a known probability distribution. A searcher, whose maximal velocity is one, starts from the origin O and wishes to discover the object in minimal expected time. It is assumed that the searcher can change the direction of his motion without any loss of time. It is also assumed that the searcher cannot see the object until he actually reaches the point at which the object is located; the time elapsed until this moment is the cost function. Originally, the problem was presented in a Bayesian context, assuming that the location of the object is given by a known probability distribution F, but what can we do if we do not know F? This situation is quite common and a natural approach for dealing with it is to try to find search trajectories that will be effective against all the possible distributions. Can such "universal" trajectories be found?

For this purpose, it is useful to consider a game between a "searcher" S and a "hider" H. As the time necessary for the searcher to locate the object may be arbitrarily high (as the object may be hidden far from the origin), a useful measure for the performance of a search strategy is the *competitive ratio*: This is the supremum of the ratio between the time the searcher actually travels and the time he would have taken if he had known the hiding place. The competitive ratio is a standard notion in the context of online algorithms; see [9, 14] for recent overviews. As the supremum has to be taken over all possible events of a game (or all possible sequences of events in the case of an online problem), it is quite useful to imagine these events chosen by a powerful adversary, who knows the strategy of the searcher. We will focus on a resulting primal-dual modeling further down.

For the linear search problem, the optimal competitive ratio is 9, as was first shown by Beck and Newman [6]: The searcher should alternate between going to the right and to the left, at each iteration doubling his step size. By placing the object at one of the points just beyond a turning point of the searcher, the hider can actually assure that this ratio of 9 is best possible.

The linear search problem has been rediscovered, re-solved, and generalized independently by a number of researchers. One such generalization is the *star search* (first solved by Gal [11]), where the searcher has to locate the object on one of m rays emanating from the origin; thus, the linear search problem is a special case for $m = 2$. See [4, 5] for a rediscovery and some extensions, as well as [20]. More recent results and references can be found in [23], which focuses on an estimate of lower-order terms if there is a known upper bound on the distance to the object.

Geometric trajectories and turn cost. One common feature of optimal trajectories for the linear search problem and variants is the fact that the step size is a geometric series. For the star search, the step size increases by a constant factor of $\frac{m}{m-1}$ at each iteration, and the overall competitive factor works out to be $\left(1 + 2\frac{m^m}{(m-1)^{m-1}}\right)$. Furthermore, it can be shown that under certain assumptions, *any* unbounded optimal search trajectory has to be a geometric sequence [10, 13]; see [2] for details and citations.

While these geometric trajectories are quite elegant from a mathematical point of view, there is a serious downside: As they "start" with an infinite sequence of infinitesimal steps, they are neither practical, nor is the necessary time realistic. (See Figure 1.) So far, the issue of the infinitesimal startup has been avoided, either implicitly or explicitly; e.g., see [23]: "In order to avoid this problem we assume that a lower bound of one for the distance to the target t is known." It should be noted that the "upper" part of the infinite sequence has been dealt with by assuming that an upper bound on the distance to the object is known in advance; see [23, 18].

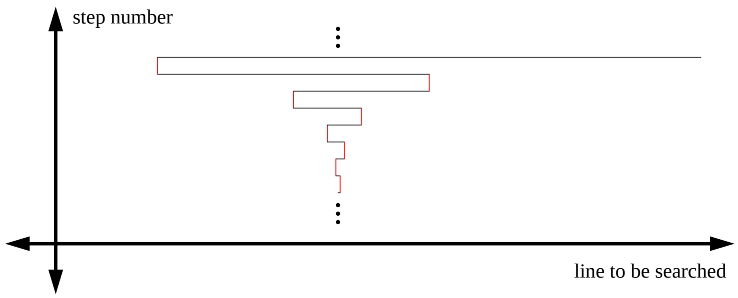

Fig. 1. An optimal search trajectory for the linear search problem: A geometric sequence without a first step.

In this paper, we study a clean and simple way to avoid the problems of geometric sequences without a first step, by assuming a constant turn cost d for changing direction. This assumption is natural and realistic, as any reasonable scenario incurs some such cost for turning. We describe optimal trajectories for this scenario; as it turns out, they are generalizations of the optimal trajectories for the linear search problem without turn cost, and have the same asymptotic behavior as $d \to 0$.

Other related work. Linear search problems occur in various contexts. See [1] for a study of rendezvous search on a line, where the objective of two players is to meet as fast as possible; this turns out to be a double linear search problem. [19] studies randomized strategies for the star search (which is also known as the "cow-path problem", motivated by a cow searching for the nearest pasture.) See [20] for more on the star search, and [15] for parallel searching.

Various types of search problems have been considered in a geometric (mostly two-dimensional) context. Here we only mention [8, 16, 22, 24], and the remarkable paper [17] that shows that online searching in a simple polygon can be performed with a competitive ratio of not more than 26.5.

A good overview on search games can be found in the book [12], and the more recent book [2].

Relatively little work has been done on geometric optimization problems with turn cost. The interested reader may find some recent discussion in the paper [3].

Infinite Linear Programs. A standard tool for computing the values of two-player games is Linear Programming. In fact, studying such games was one of the origins of Linear Programming: An optimal strategy for one player can be interpreted as an optimal primal solution, while an optimal strategy for the second player corresponds to an optimal dual solution.

At first glance, it appears that this tool is not easily applicable to the search games described above, as the game is unbounded. However, we will demonstrate that it makes sense to consider *infinite linear programs*. As it turns out, we can still construct primal and dual solutions, and prove optimality by applying duality. We believe that this tool is useful and applicable in various game-theoretic scenarios, even when there is no clear idea of a possible optimal strategy. We demonstrate this by giving the results of a computational study performed with CPLEX. As a result, we are able to give a tight analysis of the cost of an optimal search strategy in the presence of turn cost.

The paper [18] also uses linear programs for the analysis of the cow-path problem; in particular, it uses these tools for analyzing the scenario where there is a known limit on the distance to the object. Considering turn cost makes our problem different; moreover, we use a different perspective of dealing with the issue of infinitely many constraints by explicitly considering dual variables for establishing a lower bound.

Our results.

- We show that the linear search problem in the presence of turn cost can be characterized by an infinite linear program.
- Using CPLEX, we perform a computational study on the corresponding sequence of linear programs.
- From the computational results, we derive an analytic proof of the optimal strategies of searcher and hider. As a consequence of duality, the resulting bounds are tight: an optimal strategy requires 9OPT+2d, and the optimal search strategy has step size $x_i = d\left(2^i - 1\right)/2$.
- We generalize the above results to the scenario of star search by showing that the searcher can guarantee finding a solution within time

$$\left(1 + 2\frac{m^m}{(m-1)^{m-1}}\right) OPT + m\left(\left(\frac{m}{m-1}\right)^{m-1} - 1\right) d$$

by choosing the strategy

$$x_i = d\left(\left(\frac{m}{m-1}\right)^{i+1} - 1\right)/2.$$

The rest of this paper is organized as follows. In Section 2 we discuss the start of the search and show that the presence of turn cost always forces the existence of a first step with step length bounded from below. In Section 3 we derive an infinite linear program for the value of the game. Section 4 describes the results of a computational study; a clean analysis, with a mathematical proof of optimality of the derived strategies, is given in Section 5. Section 6 describes the extension to star search. Some concluding thoughts are presented in Section 7.

2 The First Move and an Additive Term

In the presence of a positive turn cost d, the hider can make it impossible for the searcher to achieve any competitive ratio at all, by simply placing the object arbitrarily close to the origin, on the side that is not picked first by the searcher. Clearly, this requires a minimum cost of d, regardless of OPT. Moreover, the searcher will be forced to make a second turn if he starts with a too small (or infinitesimal) first step. This increases the minimum cost.

We account for this minimum cost incurred by the turn cost by using an additive term: In the presence of turn cost, we write $c{\cdot}\mathrm{OPT}{+}\lambda d$, where c is the asymptotic competitive ratio without turn cost. As we will see in the following, there is indeed a well-defined optimal value for λ. Determining this critical λ is one of the main objectives of this paper.

It is clear that even in the presence of a fixed additive term, the searcher will not be able to achieve any competitive ratio at all if he uses a large number of steps before first reaching a distance of d from the origin. In particular, it can easily be seen that he is forced to make a first step of length $x_1 = \Omega(d)$. We will use this in the following sections for a more careful analysis.

3 An Infinite Linear Program

Suppose that the searcher carries out a sequence of step lengths $x_1, x_2, \ldots$ from the origin, where $x_1, x_3, \ldots$ are increasing distances to the right, while $x_2, x_4, \ldots$ are increasing distances to the left. In the following, we denote these turn positions by $p_i = (-1)^{i+1} x_i$.

For a given sequence of step lengths x_i, the hider can choose the possible set of locations $y_i = (-1)^{i+1}(x_i + \varepsilon)$ for an arbitrarily small ε. If the object is placed at y_n, the searcher will only encounter it after traveling a distance of $(\sum_{i=1}^{n+1} 2x_i) + x_n + \varepsilon$, and making $n+1$ turns. Note that for arbitrarily small d, this approaches the linear search problem, with a competitive ratio of 9. In order to guarantee this competitive ratio, and an additive cost of λd, λ must be big enough that y_i and the corresponding search trajectories satisfy

$$2x_1 + 2x_2 + \ldots + 2x_{i-2} + 3x_{i-1} + 2x_i + id + \varepsilon \leq 9(x_{i-1} + \varepsilon) + \lambda d. \quad (1)$$

As all x_i are bounded away from zero, and the above condition must hold for any $\varepsilon > 0$, we conclude that

$$2x_1 + 2x_2 + \ldots + 2x_{i-2} + 3x_{i-1} + 2x_i + id \leq 9x_{i-1} + \lambda d. \quad (2)$$

or

$$2x_1 + 2x_2 + \ldots + 2x_{i-2} - 6x_{i-1} + 2x_i + id \leq \lambda d. \quad (3)$$

As λ satisfies all of these linear inequalities, we get the following infinite linear program:

$$
\begin{array}{lllllllr}
 & \min\ \lambda & & & & & & \\
2x_1 & & & & & & + \ \ d & \leq \lambda d \\
-6x_1 + & 2x_2 & & & & & + 2d & \leq \lambda d \\
2x_1 - & 6x_2 + & 2x_3 & & & & + 3d & \leq \lambda d \\
2x_1 + & 2x_2 - & 6x_3 + & 2x_4 & & & + 4d & \leq \lambda d \\
\vdots + & \vdots - & \vdots + & \vdots & & & + \ \vdots & \leq \lambda d \\
2x_1 + & 2x_2 + & \cdots + & 2x_{i-2} - & 6x_{i-1} + & 2x_i + & id & \leq \lambda d \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \leq \lambda d \\
 & & & & & x_i & & \geq \ 0.
\end{array} \tag{4}
$$

4 Solving the Sequence of Linear Programs

Even with sophisticated software, it is impossible to solve an infinite linear program to optimality. However, for each n, the first n linear inequalities describe a relaxation of the overall program. Thus, the resulting optimal value λ_n is a lower bound on the overall λ. Furthermore, the sequence of primal optimal values $x_i^{(n)}$ and dual optimal values $c_i^{(n)}$ should converge, if there is any hope at all for an overall solution.

We have solved a number of these relaxations by using CPLEX 7.1. The computational results are shown in Table 1. Shown are the number of constraints that we used, the optimal value λ_n, the first five primal variables, and the first five dual variables. To allow numerical computation, d was normalized to 1. As each λ_n is a valid lower bound, numbers were truncated, not rounded.

n	λ_n	$x_1^{(n)}$	$x_2^{(n)}$	$x_3^{(n)}$	$x_4^{(n)}$	$x_5^{(n)}$	$c_1^{(n)}$	$c_2^{(n)}$	$c_3^{(n)}$	$c_4^{(n)}$	$c_5^{(n)}$
1	1.0000	0.5000									
2	1.2500	0.1250	.5000				.7500	.2500			
3	1.4166	0.2083	0.3333	0.0000			0.6666	0.2500	0.0833		
4	1.5312	0.2656	0.5625	0.6875	0.0000		0.0625	0.2500	0.0937	0.0312	
5	1.6125	0.3062	0.7250	1.1750	1.3000	0.0000	0.6000	0.2500	0.1000	0.0375	0.0125
6	1.6718	0.3359	0.8437	1.5312	2.2500	2.3750	0.5833	0.2500	0.1041	0.0416	0.0156
7	1.7165	0.3582	0.9330	1.7991	2.9642	4.1607	0.5714	0.2500	0.1071	0.0446	0.0178
8	1.7509	0.3754	1.0019	2.0058	3.5156	5.5930	0.5625	0.2500	0.1093	0.0468	0.0195
9	1.7782	0.3891	1.0563	2.1692	3.913	6.6284	0.5555	0.2500	0.1111	0.0486	0.0208
10	1.8001	0.4000	1.1003	2.3011	4.3031	7.5078	0.5500	0.2500	0.1125	0.0500	0.0218
20	1.9000	0.4500	1.3000	2.9000	5.9000	11.5000	0.5250	0.2500	0.1187	0.0562	0.0265
30	1.9333	0.4666	1.3666	3.1000	6.4333	12.8333	0.5166	0.2500	0.1208	0.05833	0.0281
40	1.9500	0.4750	1.4000	3.2000	6.7000	13.5000	0.5125	0.2500	0.1218	0.0593	0.0289
50	1.9600	0.4800	1.4200	3.2600	6.8600	13.9000	0.5100	0.2500	0.1225	0.0600	0.0293
100	1.9800	0.4900	1.4600	3.3800	7.1800	14.7000	0.5050	0.2500	0.1237	0.0612	0.0303
200	1.9900	0.4950	1.4800	3.4400	7.3400	15.1000	0.5025	0.2500	0.1243	0.0618	0.0307
400	1.9950	0.4975	1.4900	3.4700	7.4200	15.3000	0.50125	0.2500	0.1245	0.0621	0.0310

Table 1. Solutions for a number of linear subsystems.

The table illustrates a number of points:

1. Even small subsystem may have "ugly" solutions, indicating relatively fast increasing effort for trying to establish lower bounds by analyzing subsystems manually.
2. Convergence is rather slow; in fact, it is logarithmic, as doubling the size of the system cuts the remaining error in half. This and the tediousness of the solutions make it rather difficult to give an explicit closed formula for the objective values, which could be used for analyzing the limit. This justifies the use of powerful tools like CPLEX in order to get good estimates quickly.
3. Actually closing the remaining gap and proving that the lower bound of 9OPT+$2d$ is tight requires considering the whole infinite linear program.

We will demonstrate in the following section how the latter point can be carried out analytically.

5 Provably Optimal Strategies

For establishing optimal strategies for the infinite linear program, and thus a pair of optimal strategies, we start by describing a feasible dual solution that yields a lower bound of 2 for the objective value λ.

Consider $c_i = \frac{1}{2^i}$. We will show that with these dual multipliers, the linear combination of the first n coefficients for any variable x_j tends to zero, as n approaches infinity: This coefficient is

$$-\frac{8}{2^{j+1}} + \sum_{i=j}^{n} \frac{2}{2^i} = \frac{1}{2^j}\left(-4 + 2\sum_{i=0}^{n-j}\frac{1}{2^i}\right).$$

Clearly, this tends to zero as n grows.

Similarly, the coefficient of d becomes $\sum_{i=1}^{n}\frac{i}{2^i}$, which tends to 2 as n grows.

Finally, the coefficient of λd on the right hand side of the inequality is the geometric series $\sum_{i=1}^{n}\frac{1}{2^i}$, which tends to one.

In summary: By establishing that all those limits exist, we have justified taking an infinite linear combination with the described dual variables, which yields the bound

$$2d \leq \lambda d.$$

To see that 2 is also an upper bound, and thus an optimal solution for λ, consider $x_j = (2^j - \frac{1}{2})d$. For this particular set, the ith inequality becomes

$$\left(\sum_{h=1}^{i} 2(x_h)\right) - 8x_{i-1} + id \leq \lambda d,$$

which simplifies to

$$2d \leq \lambda d.$$

Clearly, this is satisfied for all i by $\lambda = 2$, implying that these x_i do indeed characterize an optimal strategy for the searcher.

It should be noted that the primal and dual solutions satisfy complementary slackness, as all constraints hold with equality.

We summarize:

Theorem 1
In the presence of turn cost d for the linear search problem, the searcher can guarantee a solution within time 9OPT+$2d$ by choosing the search strategy $x_i = d(2^i - 1)/2$.

6 Star Search

For the problem with no turn cost Gal [12] proved that the sequence of turning points corresponding to an optimal search trajectory has to be cyclic. In the following, we sketch how to generalize the above results to the problem of star search in the presence of turn cost.

As before, we consider a sequence $x_1, x_2, \ldots$ of steps that cycle through the m rays. Suppose we have a turn cost of d_1 on a ray, and d_2 at the origin; set $d = d_1 + d_2$. By picking a hiding spot just beyond one of the turning points, the hider can force the searcher to find the object only after making moves $x_n, \ldots, x_{n+m-1}$ and returning to the ray of x_n, making a total of $(n+m-1)$ turns. This take a time of $\left(2\sum_{i=1}^{n+m-1} x_i\right) + (n+m-1)d + x_n$ instead of the optimal x_n. Without the presence of turn cost, it is well known that the optimal competitive ratio is $\left(1 + 2\frac{m^m}{(m-1)^{m-1}}\right) =: 1 + M$. Therefore, we get the condition

$$\left(2\sum_{i=1}^{n+m-1} x_i\right) + (n+m-1)d + x_n \leq \lambda d + (1+M)x_n$$

or

$$\left(2\sum_{i=1}^{n+m-1} x_i\right) + (n+m-1)d \leq \lambda d + (M)x_n$$

for the additive term λ, if it exists.

Again, this yields an infinite sequence of linear programs. For $m = 3, 4, 5, 6$ and various n up to 1000, we solved these programs by using CPLEX. Making use of the logarithmic convergence of the series, we were able to account for the remaining numerical difficulties and obtain the following solutions:

$$x_i = d\left(\left(\frac{m}{m-1}\right)^i - 1\right)/2$$

and

$$\lambda = m\left(\left(\frac{m}{m-1}\right)^{m-1} - 1\right).$$

We show that this primal strategy and λ satisfy all constraints of the linear program with equality: The left-hand side of inequality n is

$$2\left(\sum_{i=1}^{n} x_i\right) + nd.$$

Using $q := \frac{m}{m-1}$ and the above values for $x_i = d(q^i - 1)/2$ this simplifies to

$$\frac{q^{n+m} - 1}{q - 1} d.$$

On the other hand, the right-hand side

$$\lambda d + \left(2\frac{m^m}{(m-1)^{m-1}}\right) x_{n+m-1}$$

simplifies to

$$\lambda d + \frac{q^{n+1} - 1}{q - 1} d - m\left(q^{m-1} - 1\right) d.$$

Therefore, with $\lambda = m\left(\left(\frac{m}{m-1}\right)^{m-1} - 1\right)$, all constraints are satisfied with equality, regardless of n.

We summarize:

Theorem 2
In the presence of turn cost d for the star search problem on m rays, the searcher can guarantee a solution within time

$$\left(1 + 2\frac{m^m}{(m-1)^{m-1}}\right) OPT + m\left(\left(\frac{m}{m-1}\right)^{m-1} - 1\right) d$$

by choosing the search strategy

$$x_i = d/2\left(\left(\frac{m}{m-1}\right)^{i+1} - 1\right).$$

This is tight; the explicit values of the dual variables c_i can be worked out by considering that the coefficient of x_{j+1} becomes $\left(2\sum_{i=j}^{\infty} c_i\right) + 2\frac{m^m}{(m-1)^{m-1}} c_{j+m-1}$, which must be zero. Subtracting these conditions for x_j and x_{j+1}, we get a recursive formula for c_n.

7 Conclusions

In this paper we have considered the linear search problem in the presence of turn cost. We have shown that this extends the well-studied case without turn cost, and established a performance guarantee of 9OPT$+2d$. We also extended or results to the general star search on m rays (also known as the cow-path problem), and showed that this problem can also be resolved by using an infinite sequence of linear programs.

We believe that our methods and results can be easily extended to various other problems that have been studied; in particular, it should not be too hard to give explicit estimates for the lower order terms, which show up if the distance OPT to the hidden object is known to be bounded by some D: This only requires giving an explicit estimate for the solutions of subsystems of size $n = \Theta(\log D)$.

Just like the cow-path problem extends to various geometric scenarios, we expect that there are also many other problems for which the cost of changing the search direction plays an important role.

Acknowledgments

This research was supported in part by NATO Grants PST.CLG976391 and CRG 972991. Parts of the work were done while Sándor Fekete was visiting MIT, with partial funding by DFG travel grant FE 407/7-1.

References

1. S. Alpern and A. Beck. Pure strategy asymmetric rendezvous on the line with an unknown initial distance. *Operations Research*, **48** (2000), 498–501.
2. S. Alpern and S. Gal. *The Theory of Search Games and Rendezvous*. Kluwer Academic Publishers.
3. E. Arkin, M. Bender, E. Demaine, S. P. Fekete, J. S. B. Mitchell, S. Sethia. Optimal covering tours with turn costs. *Proceedings SODA 01*, ACM-SIAM Symposium on Discrete Algorithms, 2001, 138–147.
4. R. A. Baeza-Yates, J. C. Culberson, and G. J. E. Rawlins. Searching with uncertainty. *Proceedings SWAT 88*, First Scandinavian Workshop on Algorithm Theory. Lecture Notes in Computer Science #318, 1988, 176–189.
5. R. A. Baeza-Yates, J. C. Culberson, and G. J. E. Rawlins. Searching in the plane. *Information and Computation*, **106** (1993), 234–252.
6. A. Beck and D. J. Newman. Yet more on the linear search problem. *Israel Journal of Mathematics*, **8** (1970), 419–429.
7. R. Bellman. An optimal search problem. *SIAM Reviews*, **5** (1963), 274.
8. C. A. Brocker and S. Schuierer. Searching rectilinear streets completely. *Proceedings WADS 99*, Workshop on Algorithms and Data Structures, Lecture Notes in Computer Science, # 1663, Springer 1999, 98–109.
9. A. Fiat and G. J. Woeginger. *Online optimization, the state of the art*. Lecture Notes in Computer Science, # 1442, Springer 1998.
10. S. Gal. A general search game. *Israel Journal of Mathematics*, **12** (1972), 32–45.
11. S. Gal. Minimax solutions for linear search problems. *SIAM Journal on Applied Mathematics*, **27** (1974), 17–30.
12. S. Gal. *Search Games*. Academic Press, New York, 1980.
13. S. Gal and D. Chazan. On the optimality of the exponential functions for some minimax problems. *SIAM Journal on Applied Mathematics*, **30** (1976), 324–348.
14. M. Grötschel, S. Krumke, and J. Rambau. *Online optimization of large scale systems: the state of the art*. Springer 2001.
15. M. Hammar, B. J. Nilsson, and S. Schuierer. Parallel searching on m rays. *Proceedings STACS 99*, Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Computer Science, # 1563, Springer 1999, 132–142.
16. C. Hipke, C. Icking, R. Klein, and E. Langetepe. How to find a point on a line within a fixed distance. *Discrete Applied Mathematics*, **93** (1999), 67–73.

17. F. Hoffmann, C. Icking, R. Klein, and K. Kriegel. The polygon exploration problem. *SIAM Journal on Computing*, **31** (1999), 577–600.
18. P. Jaillet and M. Stafford. Online searching. *Operations Research* **49** (2001), 501–515.
19. 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** (1996), 63–79.
20. O. Kella. Star search—a different show. *Israel Journal of Mathematics*, **81** (1993), 145–159.
21. D. E. Knuth. *Searching and Sorting*. Addison Wesley, 1973.
22. A. Lopez-Ortiz and S. Schuierer. Position-independent near optimal searching and on-line recognition in a star polygon. *Proceedings WADS 97*, Workshop on Algorithms and Data Structures, Lecture Notes in Computer Science, # 1272, 1997, 284–296.
23. A. Lopez-Ortiz and S. Schuierer. The ultimate strategy to search on m rays?, *Proceedings COCOON 98*, International Conference on Computing and Combinatorics, Lecture Notes in Computer Science, # 1449, 1998 75–84.
24. S. Schuierer. Lower bounds in on-line geometric searching. Computational Geometry—Theory & Applications, **18** (2001), 37–53.