Doctopic: Algorithms, automata, complexity and games ARTICLE IN PRESS TCS:11297

Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

A general framework for searching on a line [☆,☆☆]

Prosenjit Bose [a], Jean-Lou De Carufel [b,*]

[a] School of Computer Science, Carleton University, Ottawa, Canada
[b] School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Canada

ARTICLE INFO

Article history:
Received 27 April 2017
Accepted 24 August 2017
Available online xxxx
Communicated by L.A. Gasieniec

Keywords:
Searching on a line
Online algorithms
Optimization

ABSTRACT

Consider the following classical search problem: a target is located on a line at distance D from the origin. Starting at the origin, a searcher must find the target with minimum competitive cost. The classical competitive cost studied in the literature is the ratio between the distance travelled by the searcher and D. Note that when no lower bound on D is given, no competitive search strategy exists for this problem. Therefore, all competitive search strategies require some form of lower bound on D.

We develop a general framework that optimally solves several variants of this search problem. Our framework allows us to achieve optimal competitive search costs for previously studied variants such as: (1) where the target is fixed and the searcher's cost at each step is a constant times the length of the step, (2) where the target is fixed and the searcher's cost at each step is the length of the step plus a fixed constant (often referred to as the *turn cost*), (3) where the target is moving and the searcher's cost at each step is the length of the step.

Our main contribution is that the framework allows us to derive optimal competitive search strategies for variants of this problem that do not have a solution in the literature such as: (1) where the target is fixed and the searcher's cost at each step is $\alpha_1 x + \beta_1$ for moving distance x away from the origin and $\alpha_2 x + \beta_2$ for moving back with constants $\alpha_1, \alpha_2, \beta_1, \beta_2$, (2) where the target is moving and the searcher's cost at each step is a constant times the length of the step plus a fixed constant turn cost. Notice that the latter variant can have several interpretations depending on what the turn cost represents. For example, if the turn cost represents the amount of time for the searcher to turn, then this has an impact on the position of the moving target. On the other hand, the turn cost can represent the amount of fuel needed to make an instantaneous turn, thereby not affecting the target's position. Our framework addresses all of these variations.

1. Introduction

Consider the following classical search problem: a target is located on a line at distance D from the origin. Starting at the origin, a searcher must find the target with minimum competitive cost. The classical competitive cost studied in the literature is defined as the ratio between the distance travelled by the searcher and D. This problem and many of its variants

☆ This work was supported by FQRNT and NSERC.

☆☆ A preliminary version of this paper appeared in WALCOM 2016 [7].

* Corresponding author.
E-mail address: jdecaruf@uottawa.ca (J.-L. De Carufel).

http://dx.doi.org/10.1016/j.tcs.2017.08.023

Please cite this article in press as: P. Bose, J.-L. De Carufel, A general framework for searching on a line, Theoret. Comput. Sci. (2017), http://dx.doi.org/10.1016/j.tcs.2017.08.023

have been extensively studied both in mathematics and computer science. For an encyclopaedic overview of the field, the reader is referred to the following books on the area [2,3,19]. Techniques developed to solve this family of problems have many applications in various fields such as robotics, scheduling, clustering, or routing to name a few [9,10,16,22,25,26,28]. In particular, solutions to these problems have formed the backbone of many competitive online algorithms (see [11] for a comprehensive overview).

A *search strategy* for the problem of searching on a line is a function $\mathcal{S}(i) = (x_i, r_i)$ defined for all integers $i \geq 1$. At step i, the searcher travels a distance of x_i on ray $r_i \in \{\text{left}, \text{right}\}$. If he does not find the target, he goes back to the origin and proceeds with step $i+1$. Let D be the distance between the searcher and the target at the beginning of the search. Traditionally, the goal is to find a strategy $\mathcal{S}$ that minimizes the *competitive ratio* $CR(\mathcal{S})$ (or *competitive cost*) defined as the total distance travelled by the searcher divided by D, in the worst case. If D is given to the searcher, any strategy $\mathcal{S}$ such that $\mathcal{S}(1) = (D, \text{left})$ and $\mathcal{S}(2) = (D, \text{right})$ is optimal with a competitive ratio of 3 in the worst case. If D is unknown, a lower bound $\lambda \leq D$ must be given to the searcher, otherwise the competitive ratio is unbounded in the worst case. Indeed, if an adversary places the target at a distance $\epsilon > 0$ from the origin and the first step taken by the algorithm is $\xi > 0$ in the wrong direction, the ratio ξ/ϵ cannot be bounded. Therefore, all competitive search strategies require some form of lower bound on D in order to achieve a constant worst case competitive cost only with respect to D.

Let $Left = \{i \mid r_i = \text{left}\}$ and $Right = \{i \mid r_i = \text{right}\}$. To guarantee that, wherever the target is located, we can find it with a strategy $\mathcal{S}$, we must have $\sup_{i \in Left} x_i = \sup_{i \in Right} x_i = \infty$. We say that $\mathcal{S}$ is *monotonic* if the sequences $(x_i)_{i \in Left}$ and $(x_i)_{i \in Right}$ are strictly increasing. The strategy $\mathcal{S}$ is said to be *periodic* if $r_1 \neq r_2$ and $r_i = r_{i+2}$ for all $i \geq 1$. We know from previous work (see [4,19,27] for instance) that there is an optimal strategy that is periodic and monotonic. Let us say that a strategy is *fully monotonic* if the sequence $(x_i)_{i \geq 1}$ is monotonic and non-decreasing. We can make the following assumption without loss of generality.

Periodic-monotonic assumption. *There exists an optimal search strategy that is periodic and fully monotonic.*

In this paper, we introduce a general framework to resolve several variations of this classical search problem. Our framework allows us to match optimal competitive search costs for previously studied variants. The first one is the classical problem where the target is fixed and the searcher's cost at each step is a constant times the distance travelled. This was first studied by Gal [18,19] and subsequently by Baeza-Yates et al. [4]. Given a lower bound of λ on D, an optimal strategy, sometimes called the *power-of-two strategy*, is the following: $x_i = 2^i \lambda$ $(i \geq 1)$. At step i, if i is even, move to x_i and then return to the origin. If i is odd, move to $-x_i$ and then return to the origin. All known search strategies exhibit this alternating behaviour. The competitive cost of this strategy is 9 in the worst case. With our framework in this setting, our search strategy is $x_i = (i+1)2^i \lambda$, which also has a competitive cost of 9 in the worst case (refer to Lemma 1). Thus, although our framework achieves optimal worst case competitive costs, our strategies are not identical to those in the literature.

Another variant we solve with our framework is where the target is fixed and the searcher's cost at each step is the distance travelled plus a fixed constant (often referred to as the *turn cost*). This was first studied by Alpern and Gal [3, Section 8.4]. They provided a strategy with expected competitive cost $9 + 2t/\lambda$. They left open the question of whether this is optimal. Demaine, Fekete and Gal [14] addressed a deterministic variant of the problem. Their strategy is $x_i = \frac{1}{2}\left(2^i - 1\right)t$ $(i \geq 1)$. The total cost of this strategy is $9D + 2t$ in the worst case. The competitive cost of their algorithm only with respect to D is $9 + 2t/D$. However, as was noted above, the ratio t/D can be unbounded. As such, their strategy is not competitive with respect to only D (by the adversarial argument presented above). However, their search strategy is competitive with respect to the worst case cost of any online search strategy. Notice that when there is a cost of $t > 0$ charged for each turn made by the searcher, in essence $t + D$ is a lower bound on the worst case cost of any online search strategy. This is because in the worst case, any online strategy may start in the wrong direction and have to make at least one turn. Therefore, their search strategy is competitive with respect to $t + D$ as opposed to just D. Surprisingly, with our framework, we prove that when $t/2\lambda \leq 1$, the optimal competitive cost (only with respect to D) is still 9 in the worst case (refer to Lemma 1). When $t/2\lambda \geq 1$, the optimal search cost is

$$\left(9 + 2\frac{\left(\frac{t}{2\lambda} - 1\right)^2}{\frac{t}{2\lambda}}\right) D$$

(refer to Theorem 2). Moreover, our framework allows us to resolve the more general variant where the target is fixed and the searcher's cost at each step is $\alpha_1 x + \beta_1$ for moving distance x away from the origin and $\alpha_2 x + \beta_2$ for moving towards the origin with positive constants $\alpha_1, \alpha_2, \beta_1, \beta_2$ such that $\alpha_1 + \alpha_2 > 0$ (refer to Section 4.1).

The third variant that is encompassed by our framework is one where the target is moving and the searcher's cost at each step is the distance travelled. This was first studied by Gal [19]. Suppose that the searcher travels at speed 1 and the target travels at speed $0 < w < 1$. Note that the searcher must be able to travel strictly faster than the target in order to guarantee the existence of a successful search strategy. Given a lower bound of λ on D, an optimal strategy is $x_i = \left(2\,\frac{1+w}{1-w}\right)^i \lambda$ $(i \geq 1)$. The competitive cost of this strategy is $\frac{(3+w)^2}{(1-w)^3}$ in the worst case. With our framework, we find the strategy

$$x_i = \left(\frac{1+3w}{(1-w)(1+w)}\,i + \frac{1}{1+w}\right)\left(2\,\frac{1+w}{1-w}\right)^i \lambda,$$