Information Processing Letters 140 (2018) 13–17

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Reaching a target in the plane with no information

Andrzej Pelc

Département d'informatique, Université du Québec en Outaouais, Gatineau, Québec J8X 3X7, Canada

A R T I C L E I N F O

Article history:
Received 20 October 2017
Received in revised form 10 April 2018
Accepted 10 April 2018
Available online 12 April 2018
Communicated by Marek Chrobak

Keywords:
Exploration
Algorithm
Mobile agent
Speed
Cost
Plane

A B S T R A C T

A mobile agent has to reach a target in the Euclidean plane. Both the agent and the target are modeled as points. At the beginning, the agent is at distance at most $D > 0$ from the target. Reaching the target means that the agent gets at a *sensing distance* at most $r > 0$ from it. The agent has a measure of length and a compass. We consider two scenarios: in the *static* scenario the target is inert, and in the *dynamic* scenario it may move arbitrarily at any (possibly varying) speed bounded by v. The agent has no information about the parameters of the problem, in particular it does not know D, r or v. The goal is to reach the target at lowest possible cost, measured by the total length of the trajectory of the agent.

Our main result is establishing the minimum cost (up to multiplicative constants) of reaching the target under both scenarios, and providing the optimal algorithm for the agent. For the static scenario the minimum cost is $\Theta((\log D + \log \frac{1}{r})D^2/r)$, and for the dynamic scenario it is $\Theta((\log M + \log \frac{1}{r})M^2/r)$, where $M = \max(D, v)$. Under the latter scenario, the speed of the agent in our algorithm grows exponentially with time, and we prove that for an agent whose speed grows only polynomially with time, this cost is impossible to achieve.

1. Introduction

The background and the problem. Finding a target hidden in various environments is a task extensively studied in the literature. When the target is inert, this task is often called *treasure hunt*, and when it can move, it is called *pursuit*. In the latter case, the target is sometimes called the *robber* and the mobile agents chasing it are called *cops*.

The novelty of our formulation of this old problem is twofold. The environment is very simple, it is the Euclidean plane, and the mobile agent that has to find the target has no information whatsoever about the parameters of the problem. We ask what is the minimum cost, measured by the worst-case total length of the trajectory of the agent to reach the target under this complete ignorance.

There is one mobile agent and the target, both modeled as points. At the beginning, the agent is at distance at most $D > 0$ from the target. Reaching the target means that the agent gets at a *sensing distance* at most $r > 0$ from it. When this happens, we say that the agent *senses the target*. The agent has a measure of length and a compass showing the cardinal directions. We consider two scenarios: in the *static* scenario the target is inert, and in the *dynamic* scenario it may move arbitrarily at any (possibly varying) speed bounded by v. The agent has no information about the parameters of the problem, in particular it does not know D, r or v. This total ignorance may occur in many applications. If the target is a lost object, its position and even its distance from the agent may be unknown. If it is a hostile agent able to move, the chasing agent may not know any bound on its speed. Finally, the sensing distance at which the mobile agent can locate the target may also be unknown because it may depend on factors impossible to control. For example, if sensing is visual, the parameter

E-mail address: pelc@uqo.ca.

https://doi.org/10.1016/j.ipl.2018.04.006
0020-0190/

r may depend on the unknown density of the fog, and if it is chemical, it may depend on the unknown strength of the scent emitted by the target.

Our results. Our main result is establishing the minimum cost (up to multiplicative constants) of reaching the target under both scenarios, and providing algorithms for the agent that work at this cost. For the static scenario the minimum cost is $\Theta((\log D + \log \frac{1}{r})D^2/r)$, and for the dynamic scenario it is $\Theta((\log M + \log \frac{1}{r})M^2/r)$, where $M = \max(D, v)$. (The multiplicative constants hidden in the Θ notation are with respect to D, r and M.) Under the latter scenario, the speed of the chasing agent in our algorithm grows exponentially with time, and we prove that for a chasing agent whose speed grows only polynomially with time, this cost is impossible to achieve.

Related work. The problem of searching for a target by one or more mobile agents is an extensively studied task that was investigated under many different scenarios. The environment where the target is hidden may be a graph or a plane, the search may be deterministic or randomized, and the target may be inert or mobile. The book [1] surveys both the search for a fixed target and the related rendezvous problem, where the target and the finder are both mobile and their role is symmetric: they both cooperate to meet. This book is concerned mostly with randomized search strategies. In [12,14] the authors study relations between the problems of treasure hunt (searching for a fixed target) and rendezvous in graphs. By contrast, the survey [4] and the book [3] consider pursuit-evasion games, mostly on graphs, where pursuers try to catch a fugitive trying to escape. The authors of [2] studied the task of finding a fixed point on the line and in the grid, and initiated the study of the task of searching for an unknown line in the plane. This research was continued, e.g., in [9, 11]. The competitive approach to robot navigation and motion planning problems and to exploration and searching was adopted in [5,7,8]. In [13] the authors concentrated on game-theoretic aspects of the situation where multiple selfish pursuers compete to find a target, e.g., in a ring. The main result of [10] is an optimal algorithm to sweep a plane in order to locate an unknown fixed target, where locating means to get the agent originating at point O to a point P such that the target is in the segment OP. In [6] the authors consider the generalization of the search problem in the plane to the case of several searchers. To the best of our knowledge, the problem of efficient search for a fixed or a moving target in the plane, under complete ignorance of the searching agent, has never been studied before.

2. The static scenario

In this section we assume that the target is inert. Our algorithms produce a trajectory of the mobile agent which is a polygonal line whose segments are parallel to the cardinal directions. For any positive real x, the instruction (N, x) (resp. (E, x), (S, x), and (W, x)) has the meaning "go North (resp. East, South, and West) at distance x". Juxtaposition is used for concatenation of trajectories, and $\overline{T}$ denotes the trajectory reverse with respect to trajectory T.

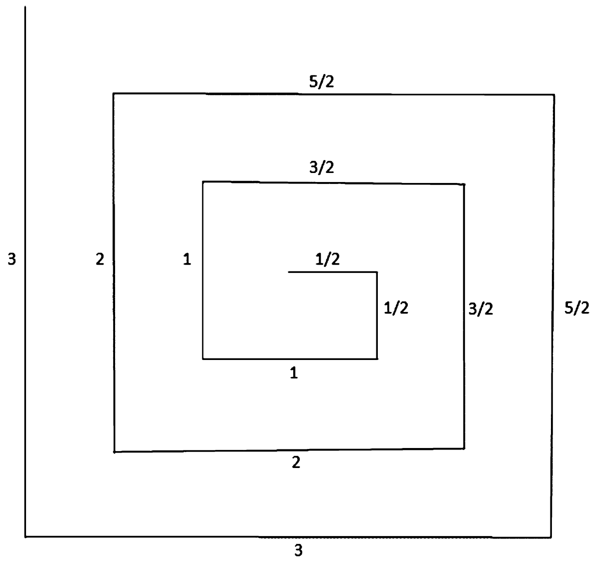

Fig. 1. The spiral $S(2, 1)$.

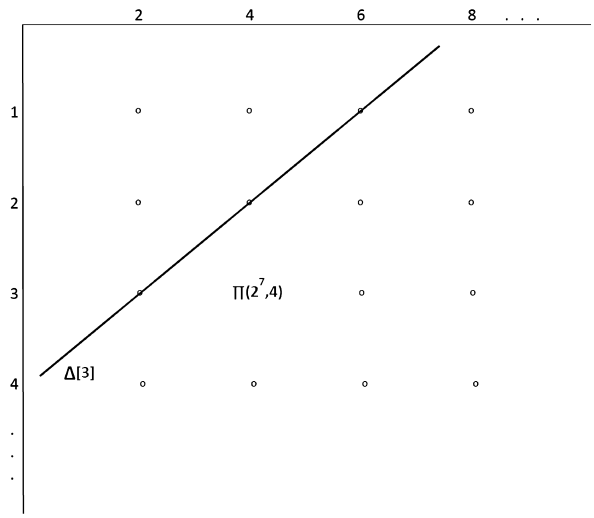

Fig. 2. The matrix A.

For any positive real y, let $Q(y)$ denote the square with side y centered at the starting point of the mobile agent.

For any positive integers k and j, the *spiral* $S(k, j)$ is the trajectory resulting from the following sequence of instructions: $(E, 2^{-j})$, $(S, 2^{-j})$, $(W, 2 \cdot 2^{-j})$, $(N, 2 \cdot 2^{-j})$, $(E, 3 \cdot 2^{-j})$, $(S, 3 \cdot 2^{-j})$, $(W, 4 \cdot 2^{-j})$, $(N, 4 \cdot 2^{-j})$, ..., $(E, (2k+1) \cdot 2^{-j})$, $(S, (2k+1) \cdot 2^{-j})$, $(W, (2k+2) \cdot 2^{-j})$, $(N, (2k+2) \cdot 2^{-j})$, cf. Fig. 1.

Note that, during the traversal of the spiral $S(k, j)$, the mobile agent gets at distance less than 2^{-j} from every point of the square $Q(2k \cdot 2^{-j})$. Denote by $\Pi(k, j)$ the trajectory $S(k, j)\overline{S(k, j)}$.

Consider the infinite matrix A whose rows are numbered by consecutive positive integers and whose columns are numbered by consecutive positive even integers. The term $A(i, j)$ in row i and column j is the trajectory $\Pi(2^{i+j}, j)$, cf. Fig. 2.