Competitive Search Ratio of Graphs and Polygons

Rudolf Fleischer [*,a,1], Tom Kamphans [b,2], Rolf Klein [b,2], Elmar Langetepe [b,2] and Gerhard Trippen [a,1]

[a] *Hong Kong University of Science and Technology, Clear Water Bay, Kowloon Hong Kong*

[b] *University of Bonn, Institute of Computer Science I, D-53117 Bonn, Germany*

Abstract

We consider the problem of searching for a goal in an unknown environment, which may be a graph or a polygonal environment. The *search ratio* is the worst-case ratio before the goal is found while moving along some search path, over the shortest path from the start point to the goal, minimized over all search paths. We investigate the problem of finding good approximations to the optimal search path, with or without a priori knowledge of the environment. In the latter case, we are dealing with an online problem. We must compute a good search path while exploring the unknown environment. We present a unified framework that allows us to derive competitive search path algorithms from existing competitive exploration algorithms, if it is possible to modify them in a certain natural way. Our transformation increases the competitive ratio at most by a factor of eight. This expresses the relationship between competitive online exploration and online searching more precisely than before. We apply our framework to searching in trees, (planar) graphs, and in (rectilinear) polygonal environments with or without holes.

Key words: Online motion planning, competitive ratio, searching, exploration, search ratio

1. Introduction

Exploration and searching are fundamental tasks in online motion planning that have attracted a lot of interest during the last decade, see for example the survey by Berman [4]. *Exploration* means that an agent should detect *all* objects inside an unknown environment whereas in *searching* one particular *goal* object has to be detected.

The *competitive ratio* is an established performance measure for online algorithms, see for example [3,7]. In exploration and search problems, the *cost* of an algorithm is typically the length of the path traveled by the agent.

Intuitively, there is a close relationship between online searching and exploration. Since the goal could be hidden anywhere, every search strategy must eventually explore the entire scene. On the other hand, there is a fundamental difference between searching and exploration. In order to achieve a small search ratio, a search strategy should explore the environment in a breadth first search manner, first exploring all locations close to the starting point, then iteratively increasing the search radius. In contrast, an exploration strategy may decide to first follow some long path. For example, for unknown undirected graphs depth first search (DFS) is a 2-competitive online exploration algorithm, but there is no competitive online search algorithm, see [9]. This indicates that online exploration may be an easier problem than online searching.

Koutsoupias et al. [10] first studied how to compute search path approximations offline. A *search path* π explores the entire enviroment. If the goal

* Corresponding author

Email addresses: rudolf@cs.ust.hk (Rudolf Fleischer), kamphans@informatik.uni-bonn.de (Tom Kamphans), rolf.klein@uni-bonn.de (Rolf Klein), langetep@informatik.uni-bonn.de (Elmar Langetepe), trippen@cs.ust.hk (Gerhard Trippen).

[1] The work described in this paper was partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. HKUST6010/01E) and by a grant from the Germany/Hong Kong Joint Research Scheme sponsored by the Research Grants Council of Hong Kong and the German Academic Exchange Service (Project No. G-HK024/02).

[2] This research was supported by the same DAAD/RGC 2003/2004 grant.

becomes visible for the first time while moving along π, we can move towards it. The quality of a search path is determined by a worst-case goal point v that maximizes the ratio between the length of the path to v via π and the shortest path to v, among all goal points v. Koutsoupias et al. called this worst-case ratio the *search ratio* of π. An *optimal search path* has the minimal search ratio among all search paths. They only studied graphs with unit length edges where the goal can only be located at one of the vertices of the graph, and they only studied the offline case, i. e., with full a priori knowledge of the graph. They showed that computing the optimal search ratio offline is an NP-complete problem, and they gave a polynomial time 8-approximation algorithm based on the doubling heuristic, which is also the main ingredient of our approach. The doubling heuristic is a standard technique in online motion planning, see for example [3,2].

Our goal is to approximate the optimal search path in an *unknown* environment by a constant-competitive *online* search algorithm. We propose a general framework how to derive such a competitive online algorithm when there exists a competitive online exploration algorithm for the environment that can be adapted in a certain natural way. Our framework can also be applied to offline exploration approximation algorithms, in which case it yields offline approximation algorithms for the optimal search ratio. The competitive ratio of the search algorithm is only at most four or eight times the approximation factor of the exploration algorithm, depending on whether goals can only be hidden at some finite set of points, like the vertices of a graph, for example, or whether the goal can be hidden anywhere in the environment.

In this extended abstract, all proofs are omitted.

2. Definitions

We want to find a good search path in some given *environment* E. In a graph environmen, edge lengths do not necessarily represent Euclidean distances in some embedding of the graph, even if the graph is planar. In particular, we do not assume that the triangle inequality holds.

The *goal set* $G \subseteq E$ is the set of locations in the environment where the goal might be hidden. For example, if E is a graph $G = (V, E)$, then the goal may either be located anywhere on some edge (*geometric search*), i. e. $G = V \cup E$, or it may only be located at some vertex (*vertex search*), i. e., $G = V$. To *explore* E means to move around in E and eventually see all potential goal positions G. To *search* E means to follow some exploration path in E, the *search path*, until the goal is detected. A path π is a search path if every point in G can be seen from at least one point on π. We assume that search paths always return to their start point. We make the usual assumption that goals must be at least in distance 1 from the start point s. If goals could be arbitrarily close to s, no algorithm could be competitive.

For $d \geq 1$, let $E(d)$ denote the part of E in distance at most d from s. A *depth-d restricted exploration* explores all potential goal positions in $G(d)$. The search path may move outside $E(d)$ as long as it stays within E. Depth-d restricted search is defined similarly.

Agents can either be *blind*, i. e., they can only sense their very close neighborhood, or they can have *vision*, i. e., they can see objects far away if the line of view is not blocked by an obstacle. In a graph environment, an agent with vision can usually see the endpoints of all adjacent edges at a vertex (even if the edges have a curved embedding in the plane). A blind agent standing at a vertex can only sense the outgoing edges of the vertex; it cannot see the length or the endpoint of an edge, and it cannot see the incoming edges [5].

In polygon searching we usually assume the agents to have vision, whereas in graph searching we usually assume the agents to be blind. Note that a blind agent must eventually visit all points in the goal set, wheras an agent with vision can afford to skip some potential goal positions as long as it can see them from somewhere else (maybe far away). For example, when searching a polygon it is sufficient to visit all visibility cuts, i. e., cuts emanating from the edges of reflex vertices.

We assume that agents have perfect memory. They always know a map of the already explored part of E, and they can always recognize when they visit some point for the second time, i. e., they have perfect localization.

Let π be a path in the environment E. For a point $p \in \pi$ let $\pi(p)$ denote the part of π between s and p and $\mathrm{sp}(p)$ the shortest path from s to p in E. We denote the length of a path segment $\pi(p)$ by $|\pi(p)|$. We write $|A|$ for the length of the tour computed by an algorithm A. For a point $p \in E$ let $\mathrm{rise}_\pi(p)$

denote the point $q \in \pi$ from which p is seen for the first time when moving along π starting at s.

The qualitiy of a search path is measured by its *search ratio* $\mathrm{sr}(\pi)$, defined as $\mathrm{sr}(\pi) := \max_{p \in G} \frac{|\pi(q)|+|qp|}{|\mathrm{sp}(p)|}$, where $q = \mathrm{rise}_\pi(p)$. Note that $q = p$ for blind agents. An *optimal search path*, π_{opt}, is a search path with a minimum search ratio $\mathrm{sr}_{\mathrm{opt}}$, i. e., $\mathrm{sr}_{\mathrm{opt}} = \mathrm{sr}(\pi_{\mathrm{opt}})$.

Since the optimal search path seems hard to compute [10], we are interested in finding good approximations of the optimal search path, in offline and online scenarios. We say a search path π is C*-competitive* (with respect to the optimal search path π_{opt}) if $\mathrm{sr}(\pi) \leq C \cdot \mathrm{sr}(\pi_{\mathrm{opt}})$.

3. A General Approximation Framework

In this section we will show how to transform an exploration algorithm, offline or online, for a certain environment into a search algorithm, without losing too much on the approximation factor.

Let E be the given environment and π_{opt} an optimal search path. We assume that, for any point p, we can reach s from p on a path of length at most $\mathrm{sp}(p)$. Note that this is not the case for directed graphs, for example, but it is true for undirected graphs and polygonal environments.

For $d \geq 1$, let *Expl*(d) be a family of depth-d restricted exploration algorithms for E, either online or offline. Let OPT and OPT(d) denote the corresponding optimal offline depth-d restricted exploration algorithms.

Definition 1 *The family Expl*(d) *is* DREP (depth restricted exploration property) *if there are constants* $\beta > 0$ *and* $C_\beta \geq 1$ *such that, for any* $d \geq 1$*, Expl*(d) *is* C_β*-competitive against the optimal algorithm OPT*(βd)*, i. e.,* $|Expl(d)| \leq C_\beta \cdot |OPT(\beta d)|$. □

In the normal competitive framework, $\beta = 1$. Usually, we cannot just take an exploration algorithm *Expl* for E and restrict it to points in distance at most d from s. This way, we might miss useful shortcuts outside of $E(d)$. Even worse, it may not be possible to determine in an online setting which parts of the environment belong to $E(d)$, making it difficult to explore the right part of E.

To obtain a search algorithm for E we use the well-known *doubling strategy.* For $i = 1, 2, 3, \cdots$, we successively run the exploration algorithm *Expl*(2^i), each time starting at s.

Theorem 2 *The doubling strategy based on a DREP exploration strategy is a* $4\beta C_\beta$*-competitive (plus an additive constant) search algorithm for blind agents, and a* $8\beta C_\beta$*-competitive search algorithm for agents with vision.* □

In the next two sections we will apply our framework to various types of environments and agents. The difficult part is always to find good DREP exploration algorithms.

4. Searching Graphs and Polygons

When searching graphs, we assume agents are blind. In the vertex search problem, we assume w.l.o.g. that graphs do not have parallel edges. Otherwise, there can be no constant-competitive vertex search algorithm.

Theorem 3 *For blind agents, there is no constant-competitive online vertex search algorithm for non-planar graphs with unit length edges, directed planar graphs with unit length edges, and undirected planar graphs with arbitrary edge lengths. Further, there is no constant-competitive online geometric search algorithm for directed graphs with unit length edges.* □

Note that non-competitiveness for planar graphs implies non-competitiveness for general graphs, non-competitiveness for unit length edges implies non-competitiveness for arbitrary length edges, and non-competitiveness for undirected graphs implies non-competitiveness for directed graphs.

On trees, DFS is a 1-competitive online exploration algorithm for vertex and geometric search that is DREP; it is still 1-competitive when restricted to search depth d, for any $d \geq 1$. Thus, the doubling strategy gives a polynomial time 4-competitive search algorithm for trees, online and offline. On the other hand, it is an open problem whether the computation of an optimal vertex or geometric search path in trees with unit length edges is NP-complete [10].

We will now give competitive search algorithms for planar graphs with unit length edges and for general graphs with arbitrary length edges. Both algorithms are based on an online algorithm for online tethered graph exploration. In the *tethered* *exploration* problem the agent is fixed to the start point by a restricted length rope. An optimal solution to this problem was given by Duncan et al. [6]. As they pointed out, the algorithm can also be used for depth restricted exploration. We note that it can also be adapted to work on graphs with arbitrary length edges.

Lemma 4 *In planar graphs with unit length edges, Expl(d) is a DREP online vertex exploration algorithm with $\beta = 1 + \alpha$ and $C_\beta = 10 + \frac{16}{\alpha}$.*

Theorem 5 *The doubling strategy based on Expl(d) is a $(104 + 40\alpha + \frac{64}{\alpha})$-competitive online vertex search algorithm for blind agents in planar graphs with unit length edges.* □

For $\alpha = \sqrt{\frac{16}{10}}$, the competitive factor is minimal.

Lemma 6 *In general graphs with arbitrary length edges, Expl(d) is a DREP online geometric exploration algorithm with $\beta = 1+\alpha$ and $C_\beta = 4+\frac{8}{\alpha}$.* □

Theorem 7 *The doubling strategy based on Expl(d) is a $(48 + 16\alpha + \frac{32}{\alpha})$-competitive online geometric search algorithm for blind agents in general graphs with arbitrary length edges.* □

When searching polygons, we assume that agents have vision. The only known algorithm, PE, for the online exploration of a simple polygon by Hoffmann et al. [8] achieves a competitive ratio of 26.5. This algorithm can be adapted for depth restricted exploration.

Lemma 8 *In a simple polygon, $PE(d)$ is a DREP online exploration algorithm with $\beta = 1$ and $C_\beta = 26.5$.* □

Theorem 9 *The doubling strategy based on $PE(d)$ is a 212-competitive online search algorithm for an agent with vision in a simple polygon. There is also a polynomial time 8-competitive offline search algorithm.* □

The problem of efficiently computing an optimal search path in a polygon is still open.

Theorem 10 *For an agent with vision in a simple rectilinear polygon there is a $8\sqrt{2}$-competitive online search algorithm. There is also a polynomial time 8-competitive offline search algorithm.* □

Based on a construction by Albers et al. [1] we can show the following theorem.

Theorem 11 *For an agent with vision in a polygon with rectangular holes there is no constant-competitive online search algorithm.* □

The offline exploration problem is NP-hard, and there is an exponential time 8-approximation algorithm. Thus, we get an approximation factor of 8 for the optimal search ratio. The results of Koutsoupias et al. [10] imply that the offline problem of computing an optimal search path in a known polygon with (rectangular) holes is NP-complete.

References

[1] S. Albers, K. Kursawe, and S. Schuierer. Exploring unknown environments with obstacles. In *Proc. 10th ACM-SIAM Sympos. Discrete Algorithms*, pages 842–843, 1999.

[2] S. Alpern and S. Gal. *The Theory of Search Games and Rendezvous*. Kluwer Academic Publications, 2002.

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

[4] P. Berman. On-line searching and navigation. In A. Fiat and G. Woeginger, editors, *Competitive Analysis of Algorithms*. Springer-Verlag, 1998.

[5] X. Deng and C. H. Papadimitriou. Exploring an unknown graph. *Journal of Graph Theory*, 32:265–297, 1999.

[6] C. A. Duncan, S. G. Kobourov, and V. S. A. Kumar. Optimal constrained graph exploration. In *Proc. 12th ACM-SIAM Symp. Discr. Algo.*, pages 307–314, 2001.

[7] A. Fiat and G. Woeginger, editors. *On-line Algorithms: The State of the Art*, volume 1442 of *Lecture Notes Comput. Sci.* Springer-Verlag, 1998.

[8] F. Hoffmann, C. Icking, R. Klein, and K. Kriegel. The polygon exploration problem. *SIAM J. Comput.*, 31:577–600, 2001.

[9] C. Icking, T. Kamphans, R. Klein, and E. Langetepe. On the competitive complexity of navigation tasks. In H. B. Hendrik I. Christensen, Gregroy D. Hager and R. Klein, editors, *Sensor Based Intelligent Robots*, volume 2238 of *LNCS*, pages 245–258, Berlin, 2002. Springer.

[10] E. Koutsoupias, C. H. Papadimitriou, and M. Yannakakis. Searching a fixed graph. In *Proc. 23th Internat. Colloq. Automata Lang. Program.*, volume 1099 of *LNCS*, pages 280–289. Springer, 1996.