Exploring an Unknown Cellular Environment

Christian Icking Thomas Kamphans
Rolf Klein Elmar Langetepe

FernUniversität Hagen, Praktische Informatik VI
D-58084 Hagen, Germany

Abstract

We investigate the exploration problem of a short-sighted mobile robot moving about in an unknown cellular room. In order to explore a cell, the robot must enter it. Once inside, the robot knows which of the 4 adjacent cells exist and which are boundary edges. The robot starts from a specified cell adjacent to the room's outer wall; it visits each cell, and returns to the start. Our interest is in a short exploration tour, that is, in keeping the number of multiple cell visits small. For abitrary environments containing obstacles we provide a strategy producing tours of length $S \leq C + \frac{1}{2}E + H - 3$, where C denotes the number of cells—the area—, E denotes the number of boundary edges—the perimeter—, and H is the number of obstacles.

Key words: Online algorithms, competitive analysis, unknown environment, obstacles.

1 Introduction

Exploring an unknown environment and searching for a target in unknown position are among the basic tasks of autonomous mobile robots. Both problems have received considerable attention in computational geometry and in robotics; see e. g. [3, 5, 7, 10].

Often it is assumed that the robot is equipped with an ideal vision system that provides, in a continuous way, the full visibility polygon of the robot's current position. This is unrealistic in practical situations, since the resolution of any vision system is limited, of course, and, on the other hand, service robots like lawn mowers, vacuum systems, or other cleaning devices need to get close to the parts of the environment they want to inspect and, possibly, to work on.

Therefore, we study in this paper the model of a rather short-sighted robot. We assume that the environment is given by a polygon, P, which consists of square cells on an integer grid and which may contain some impenetrable areas, i. e. obstacles.

The robot starts from a cell, s, adjacent to P's boundary. From there it can enter one of the neighboring cells, and so on. Once inside a cell, the robot knows which of its 4 neighbors exist and which are boundary edges. The robot's task is to visit each cell inside P and to return to the start; see Figure 1 (i) for an example.

This example shows a tour that visits each cell at least once, and we are interested in producing an exploration tour as short as possible.

Even though our robot does not know its environment in advance it is interesting to ask how short a tour can be found in the off-line situation, i. e. when the environment is already known. This amounts to constructing a shortest traveling salesperson (TSP) tour on the cells.

If the polygonal environment contains obstacles, the problem of finding such a minimum length tour is known to be NP-hard [8] and there are some approximation schemes [1, 2, 6, 9].

In a simple polygon without obstacles, the complexity of constructing off-line a minimum length tour seems to be open. There are, however, some results concerning the related Hamiltonian cycle and path problems [4, 12] and approximations [1, 11].

For our online exploration problem we are not aware of any previous work.

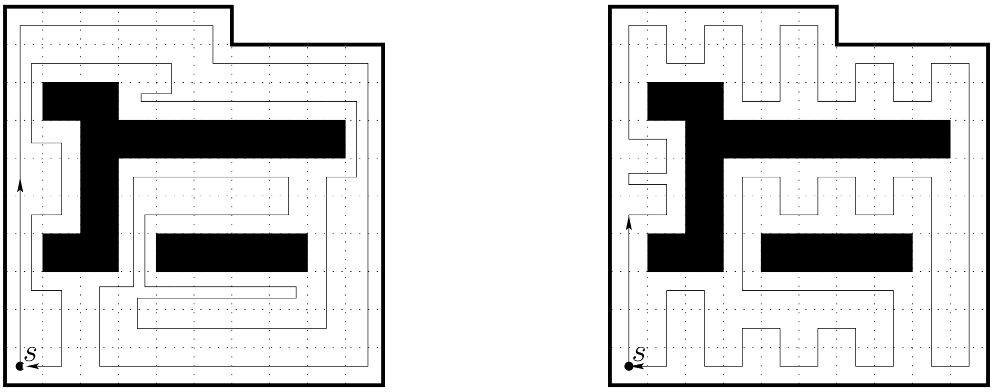

Figure 1: (i) An exploration tour with two obstacles. (ii) A shortest TSP tour for the same scene.

2 Preliminary Considerations

The first question is if the robot is still able to approximate the optimum solution up to a constant factor if it does not know the environment. Surprisingly, there is a quick and rather simple answer.

Theorem 1 *The competitive complexity of exploring an unknown cellular environment with obstacles equals 2.*

Proof. Even if we do not know the environment we can apply depth first search (DFS) to the cell graph.[1] This results in a complete traversal of a tree on C nodes, where C is the number of cells. Such a traversal takes $2C-2$ steps. Since the shortest tour needs at least C steps to visit all cells and to return to s, DFS turns out to be competitive with a factor of 2.

On the other hand, 2 is also a lower bound for the competitive factor of any strategy. To prove this, we construct a special cell graph which depends on the behavior of the strategy. The start position s is situated in a long corridor of width 1. We fix a large number Q and observe how the strategy explores the corridor. Two cases are distinguished.

Case 1: The robot comes back to s some time after having made at least Q and at most $2Q$ steps. At this time, we close the corridor such that there are only two unvisited cells, one at each end, see Figure 2 (i). Let R be the number of cells visited at this point. The robot has already made at least $2R-2$ steps and needs another $2R$ steps to visit the two remaining cells and to return to s while the shortest tour needs only $2R$ steps to accomplish this task.

Case 2: In the remaining case the robot, more or less, concentrates on one end of the corridor. Let R be the number of cells visited after $2Q$ steps. At that time, we add a bifurcation at a cell b just behind the farthest visited cell on the corridor, see Figure 2 (ii). Two paths arise which turn back and run parallel to the long corridor. If the robots comes back to s before exploring one of the two paths, then an argument analogous to Case 1 applies. Otherwise, one of the two paths will eventually be explored till the end at a cell e where it turns out that it is connected to the other end of the first corridor. At this time, the other path is fixed to be a dead end of length R' which closes just one cell after the last visited cell, e'.

---

[1] The nodes of the graph are the cells, two nodes are joined by an edge if their cells lie side by side.

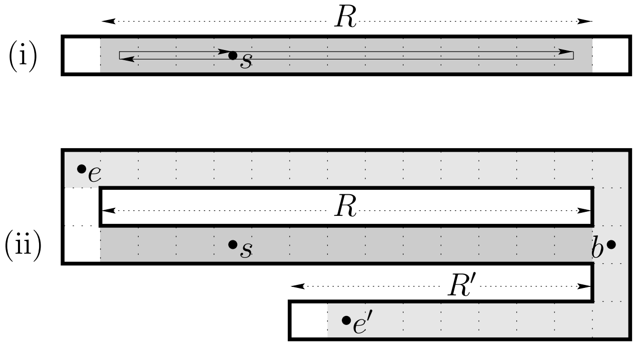

Figure 2: Proving the lower bound of 2 for the competitive factor.

From e, the robot must still walk to the other end of the corridor and visit the dead end, and then return to s. At the end, it will have walked at least four times the length of the corridor, R, plus four times the length of the dead end, R'. The optimal path only needs $2R + 2R'$, apart from a constant number of steps.

In any case the lower bound for f tends to 2 while Q goes to infinity. $\square$

Does this mean that DFS is an optimum exploration strategy? Hardly. There is no reason why one should always visit each cell twice, e. g. in a large empty square, just because some environments require such effort, like lengthy corridors.

In order to differentiate between fleshy and skinny environments we introduce their *perimeter*, E, as a parameter additional to their area, C. Technically, C denotes the number of cells whereas E is the total number of cell edges that appear on the outer wall of P and on the obstacle boundaries.

3 An exploration strategy

Our strategy *CellExplore* uses two modes:

> In the *forward* mode the polygon is explored following the left-hand rule: For every cell that is entered the robot tries to continue its path to an adjacent, unexplored, and unreserved cell, preferring a left turn over a straight step over a right turn, while all cells right to the walked path are reserved for the return path.
>
> Whenever no forward step is possible, the strategy enters the *backward* mode and walks back along the reserved return path until an unexplored and unreserved cell appears adjacent to the actual position. In this case, the forward mode is entered again. If the return path is blocked, the robot walks back using cells from the forward path, see Figure 3 (i) for an example.

Theorem 2 *Strategy CellExplore explores a polygon P with C cells, E edges and H obstacles in $S \leq C + \frac{1}{2}E + H - 3$ steps.*

The proof uses the following observations: For tree polygons, i. e. those whose cell graph is a tree, plain DFS and CellExplore follow exactly the same path of length $S = 2C - 2$ which equals $C + \frac{1}{2}E - 3$ in this case.

Now we construct a new polygon, P', which contains only the cells of P that are first entered in forward mode and certain additional cells such that P' is a tree polygon, see Figure 3 (ii). We note that CellExplore uses essentially identical forward paths in P and P'.

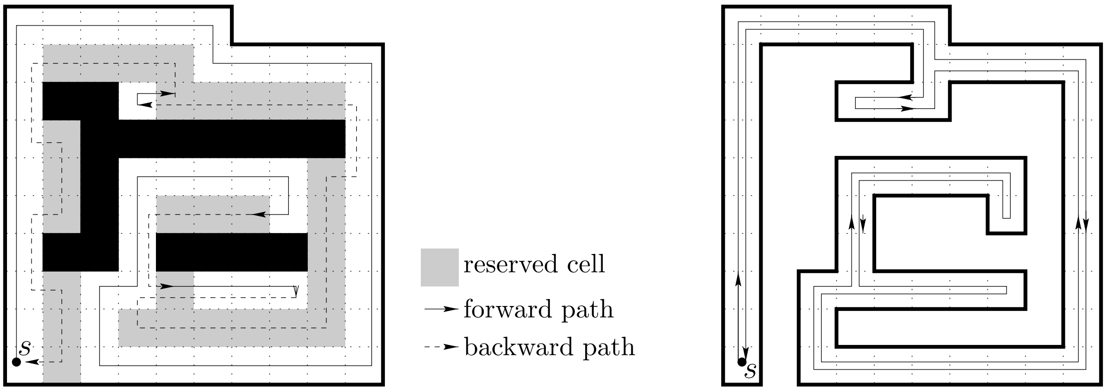

Figure 3: (i) Reserved cells in the scene fr. Figure 1. (ii) The corresponding tree polygon P'.

Starting from P', we successively add the prementioned cells in the order they are visited by the strategy in P and observe how the number of cells, edges and steps change if CellExplore is applied to each of the intermediate polygons. It can be shown that each such elementary transformation provokes only a small change in the exploration path and preserves the desired inequalitity, i. e. $S_{i+1} - (C_{i+1} + \frac{1}{2}E_{i+1}) \leq S_i - (C_i + \frac{1}{2}E_i)$ where C_i, E_i, S_i denote the number of cells, edges and steps before the cell i is added, except for one additional step per obstacle.

For fleshy environments, the value of E is in $O(\sqrt{C})$, so that the number of excess cell visits is substantially smaller than the number of cells. Only in environments which do not contain any 2×2 square of cells does E takes on its maximum possible value of $2(C - H + 1)$, and our upper bound thus becomes $2C - 2$, the cost of applying plain DFS. But in this case one cannot do better, as even the shortest TSP tour uses that many steps. Otherwise, strategy CellExplore is more efficient than DFS.

References

[1] E. M. Arkin, S. P. Fekete, and J. S. B. Mitchell. Approximation algorithms for lawn mowing and milling. Technical report, Mathematisches Institut, Universität zu Köln, 1997.

[2] S. Arora. Polynomial time approximation schemes for Euclidean TSP and other geometric problems. In *Proc. 37th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 2–11, 1996.

[3] X. Deng, T. Kameda, and C. Papadimitriou. How to learn an unknown environment I: The rectilinear case. *J. ACM*, 45(2):215–245, 1998.

[4] H. Everett. Hamiltonian paths in non-rectangular grid graphs. Report 86-1, Dept. Comput. Sci., Univ. Toronto, Toronto, ON, 1986.

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

[6] M. Grigni, E. Koutsoupias, and C. H. Papadimitriou. An approximation scheme for planar graph TSP. In *Proc. 36th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 640–645, 1995.

[7] F. Hoffmann, C. Icking, R. Klein, and K. Kriegel. The polygon exploration problem: A new strategy and a new analysis technique. In *Robotics: The Algorithmic Perspective, Proc. 3rd Workshop Algorithmic Found. Robot.*, pages 211–222. A. K. Peters, 1998.

[8] A. Itai, C. H. Papadimitriou, and J. L. Szwarcfiter. Hamilton paths in grid graphs. *SIAM J. Comput.*, 11:676–686, 1982.

[9] J. S. B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple new method for the geometric k-MST problem. In *Proc. 7th ACM-SIAM Sympos. Discrete Algorithms*, pages 402–408, 1996.

[10] J. S. B. Mitchell. Shortest paths and networks. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 24, pages 445–466. CRC Press LLC, Boca Raton, FL, 1997.

[11] S. Ntafos. Watchman routes under limited visibility. *Comput. Geom. Theory Appl.*, 1(3):149–170, 1992.

[12] C. Umans and W. Lenhart. Hamiltonian cycles in solid grid graphs. In *Proc. 38th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 496–507, 1997.