Complexity of coverage path planning problems for tethered robots in nonconvex environments

Xiao Peng[1], François Schwarzentruber[2], Olivier Simonin[1], Christine Solnon[1]

[1] CITI lab., INSA Lyon, Inria, F-69621 Villeurbanne, France
{xiao.peng,olivier.simonin,christine.solnon}@insa-lyon.fr
[2] Univ Rennes, IRISA, CNRS, France
francois.schwarzentruber@ens-rennes.fr

Mots-clés : *coverage path planning, grid graph, computational complexity.*

1 Introduction

Coverage Path Planning (CPP) is a fundamental problem in robotics, and it has many applications such as automatic floor cleaning, area patrol, or rescue search. CPP can be formulated as follows. Given a planar 2D workspace with convex polygonal obstacles, and the size of a mobile robot, find a shortest coverage cycle that fully covers the workspace and returns to its starting point. To solve this problem, we usually discretize the workspace into a 4-connected grid graph g such that each cell of the grid has the same size as the robot. In this case, if there exists a Hamiltonian cycle in g, we actually have a shortest coverage path as each cell is traversed exactly once. The complexity of deciding of the existence of a Hamiltonian cycle depends on the properties of g: it is in $\mathcal{O}(1)$ for rectangular grids with no obstacles whereas it becomes $\mathcal{NP}$-complete in case of obstacles [2]. Spanning Tree Coverage (STC) may be used to compute an approximate solution in polynomial time [1]: we construct a graph G_4 such that each vertex of G_4 corresponds to a group of 2×2 adjacent cells in g, and edges of G_4 correspond to adjacency relations between these 2×2 cell groups; if each cell of g belongs to exactly one 2×2 cell group, then there exists a Hamiltonian cycle in g if and only if G_4 is connected and, given a spanning tree T of G_4, a Hamitonian cycle in g can be constructed by circumnavigating T, as illustrated in Fig. 1(a).

In this work, we consider CPP for a tethered robot that is anchored by a cable to a fixed base point. Tethered robots are largely deployed in underwater and disaster recovery missions where a tet can provide stable communication links between robots and control center. We study the complexity and introduce algorithms for CPP when adding two constraints related to cables: a limit on the length of the cable (Section 2), and forbidden areas (Section 3).

2 CPP with limited cable length

When the length of the cable is equal to ℓ, some cells may become out of reach because the shortest path from the anchor point in the workspace has a length greater than ℓ. These out-of-reach cells are discarded, and we aim at finding a shortest coverage cycle in g that covers all reachable cells. However, the length of a path in g may be longer than the cable length as the cable is kept taut by a system that pulls on it. Hence, a shortest path in g does not necessarily leads to a shortest cable length, as illustrated in Fig. 1(b). As a consequence, using a breadth-first-search (BFS) to compute a spanning tree in G_4 is not enough to ensure that the corresponding Hamiltonian path in g will never exceed the cable length. We show how to adapt Dijkstra's algorithm to compute a spanning tree in G_4 that minimizes, for each reachable cell c, the cable length between the anchor point and c. This allows us to compute in polynomial time an approximate solution that satisfies the cable length constraint.

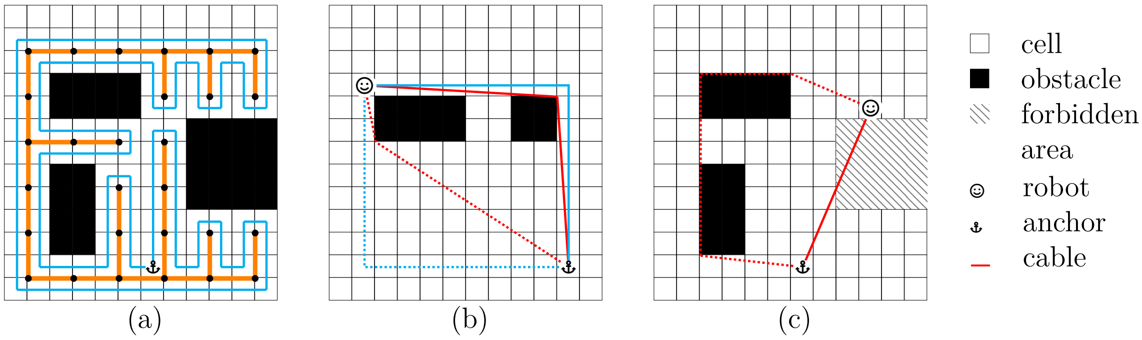

FIG. 1: (a): Example of workspace discretized in a grid g. Vertices of G_4 are black circles, and each of these vertices covers the 4 cells around the circle. In this example, G_4 does not cover the top row of cells. A spanning tree of G_4 is displayed in orange and the cycle in blue is an approximate solution that visits all cells but those in the top row exactly once: cells of this row are visited twice by a forth and back path. (b): Example of paths (in blue) and cable positions (in red). The blue solid path is smaller than the blue dashed path. However, when cables are kept taut, the red solid cable length is longer than the red dashed cable length. (c): the red solid line is an invalid configuration because the tether overlaps the forbidden area. The robot can choose to bend around the obstacles to reach the target with the cable always being inside the workspace.

3 CPP in case of forbidden areas

More generally, the workspace contains forbidden areas where the robot and its cable are forbidden to pass (because passing through these areas may damage the robot or its cable, or because these areas are crossed by humans who may be affected by the presence of a robot or a cable, for example). A main difference between a forbidden area and an obstacle comes from the fact that the cable is blocked by obstacles (it wraps around them), whereas it is not blocked by forbidden areas, as illustrated in Fig. 1(c). We show that the presence of forbidden areas increases the complexity of the problem of finding a spanning tree in G_4: we prove that this problem becomes $\mathcal{NP}$-complete by a reduction from planar 3-SAT [3]. We also use Integer Linear Programming to compute the maximum tree in G_4.

4 Conclusion and perspectives

We proposed a new variant of tethered coverage problem, where the cable has a limited length and the workspace contains forbidden areas. In this new setting, some planning problems become intractable. As future work, we plan to study the parameterized complexity of these problems.

References

[1] Y. Gabriely and E. Rimon. Spanning-tree based coverage of continuous areas by a mobile robot. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, volume 2, pages 1927–1933 vol.2, 2001.

[2] Alon Itai, Christos H Papadimitriou, and Jayme Luiz Szwarcfiter. Hamilton paths in grid graphs. *SIAM Journal on Computing*, 11(4):676–686, 1982.

[3] David Lichtenstein. Planar formulae and their uses. *SIAM Journal on Computing*, 11(2):329–343, 1982.