

Overview of the knowledge on minimum graph motion planning problem

Stepan Yurtsiv, 246437

May 28, 2022

Problem description

We are given a connected, undirected graph G with n vertices. One of the vertices s contains a mobile *robot*. Each of several other vertices contains a single moveable *obstacle*. The robot and the obstacles may only reside at vertices, although they may be moved across edges. A vertex can only contain a single object at a given time (robot/obstacle). In one step, we may either move the robot or one of the obstacles from its current position v to an adjacent vertex that is not occupied by the robot or an obstacle (in other words a *hole*). Our goal is to move the robot to a target position t using the smallest number of steps possible. Let us call this graph motion planning with one robot, or GMP1R for short.

The problem is a simple abstraction of a robot motion planning problem with the geometry replaced by the adjacencies in the graph. It was first formulated on Proceedings of the 35th Annual Symposium on Foundations of Computer Science (1994) in an article called Motion Planning on a Graph [5].

To get an appreciation of how hard the problem is, let us consider a relatively simple instance presented in figure 1.

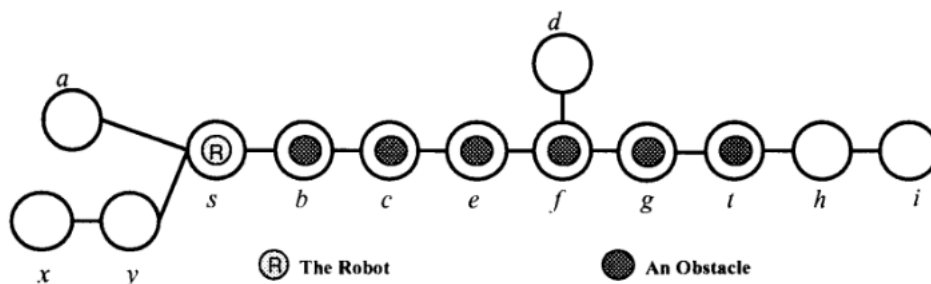


Figure 1: An instance of GMP1R problem [5]

The only feasible plan for moving the robot from s to t is the following:

1. Move the robot to a
2. Move the obstacles from b and c to x and y
3. Move all obstacles on a path from e to t two steps to the right
4. Move the robot to d
5. Move obstacles on g and t to the right past f , clearing the way for the robot from d to t

As you can see, the path of the robot (and obstacles) may be non-monotone which means sometimes it needs to move away from the target t .

State of the knowledge

By reduction from 3-SAT authors of the problem [5] have proven that given an instance of GMP1R and a positive integer k , it's NP-complete to decide whether a solution of length k exists. The problem remains NP-complete when restricted to a planar graph.

Trees

If G is a tree, there's a polynomial time algorithm that computes an optimal plan. The algorithm itself is quite complex and depends on a lot of definitions, but in the end it boils down to solving $O(n^6)$ mincost flow problems on networks with $O(n)$ nodes each, and a shortest path problem on a graph

with $O(n^3)$ nodes (you can read an exact algorithm in [5] section 2.2). The running time of the algorithm is quite large so the authors of the problem also presented a faster approximation algorithm that achieves a plan of length at most 7 times the optimal length. The algorithm is a simplified version of the previous one that requires solving mincost flow problems on at most n networks with $O(n)$ nodes each.

General graphs

Definition 0.1. We call a path of G a *degree – 2 chain* or simply a *chain* if all of its internal vertices are of degree 2 and neither of the endpoints is of degree 2.

No exact polynomial algorithms have been found for GMP1R on general graphs so we need to resort to approximation ones. A natural heuristic for an approximate solution is to let the robot flow the shortest path from s to t , with possible side-stepping. The plan obtained by this heuristic can be as bad as $\Omega(l_{max})$ times optimal, where l_{max} is the length of the longest chain in G . It's because a chain of length l packed with obstacles requires $\Omega(l^2)$ steps to clear, while the optimal plan may choose a slightly longer path with few obstacles. Another case in which the shortest path performs poorly is when there is a rich pool of holes “closer” to the longer path than the shorter path. After some in-depth analysis, authors of the problem [5] present a polynomial time $O(\sqrt{n})$ -approximation algorithm for GMP1R on general graphs ([5] section 3).

Cartesian product graphs

Definition 0.2. The Cartesian product $G_1 \square G_2$ of two graph G_1 and G_2 (figure 2) is a graph with vertex set $V(G_1) \times V(G_2)$ in which (u_i, v_j) and (u_p, v_q) are adjacent if one of the following conditions holds:

1. $v_j = v_q$ and $u_i, u_p \in E(G_1)$
2. $u_i = u_p$ and $v_j, v_q \in E(G_2)$

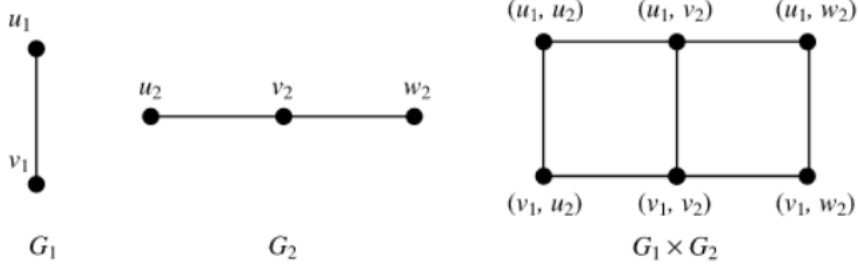


Figure 2: Cartesian product of two graphs

Consider a version of GMP1R with $n - 2$ obstacles, i.e. there's a single hole. The authors of [2] showed that the path traced by the robot coincides with a shortest path in case of Cartesian product graphs. They also gave the minimum number of moves required for the motion planning problem in the Cartesian product of two graphs having girth (length of the shortest cycle in a graph) 6 or more, but it requires too much definitions to fully express it in this overview.

Lexicographic product graphs

Definition 0.3. The Cartesian product $G_1 \circ G_2$ of two graph G_1 and G_2 (figure 3) is a graph with vertex set $V(G_1) \times V(G_2)$ in which (u_i, v_j) and (u_p, v_q) are adjacent if one of the following conditions holds:

1. $u_i, u_p \in E(G_1)$
2. $u_i = u_p$ and $v_j, v_q \in E(G_2)$

Unlike Cartesian product, the lexicographic product is not commutative, i.e. $G_1 \circ G_2$ is not the same as $G_2 \circ G_1$.

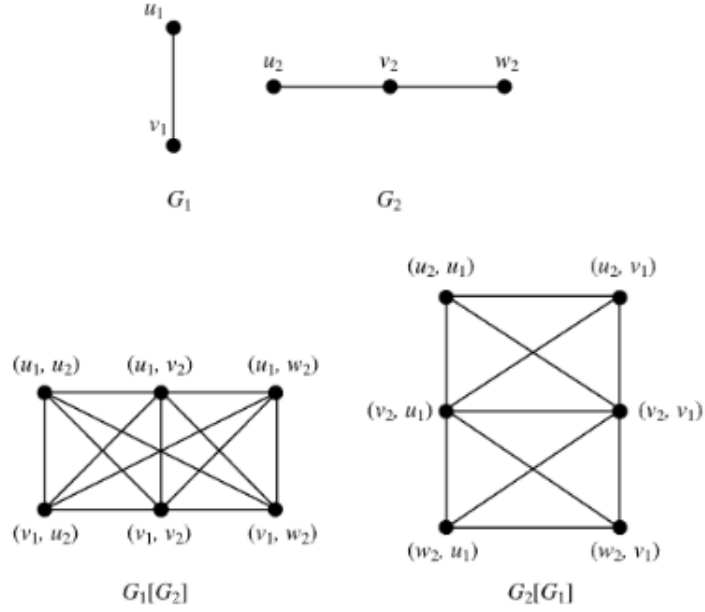


Figure 3: Lexicographic product of two graphs

Akwu and Oyewumi in [1] showed the minimum number of moves required for the motion planning problem in lexicographic products of some graphs. In addition, they proved the necessary and sufficient condition for the connectivity of the lexicographic product of two graphs.

Similar problems

A generalisation of GMP1R is $\text{GMP}k\text{R}$, where there are k robots with respective destinations. A special case of $\text{GMP}k\text{R}$, in which there are no obstacles, has been studied previously. Wilson [6] studies the case $k = n - 1$ and gives an efficiently checkable characterization of the solvable instances of the problem. Kornhauser, Miller, and Spirakis [4] extend his result to any $k \leq n - 1$ and also give an upper bound of $O(n^3)$ on the number of steps to solve any solvable instance. They give an example showing that this bound is the best possible. Goldreich [3] has studied the complexity of determining the shortest move sequence for the $\text{GMP}k\text{R}$ problem and shown that this is NP-hard in the case $k = n - 1$.

References

- [1] AD Akwu and O Oyewumi. Motion planning in certain lexicographic product graphs. *International J. Math. Combin*, pages 146–153, 2018.
- [2] Biswajit Deb and Kalpesh Kapoor. Motion planning in cartesian product graphs. *Discussiones Mathematicae Graph Theory*, 34(2):207–221, 2014.
- [3] Oded Goldreich. Shortest move-sequence in the generalized 15-puzzle is np-hard. *Manuscript, Laboratory for Computer Science, MIT*, 1, 1984.
- [4] Daniel Martin Kornhauser, Gary Miller, and Paul Spirakis. Coordinating pebble motion on graphs, the diameter of permutation groups, and applications. Master’s thesis, M. I. T., Dept. of Electrical Engineering and Computer Science, 1984.
- [5] C.H. Papadimitriou, P. Raghavan, M. Sudan, and H. Tamaki. Motion planning on a graph. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 511–520, 1994.
- [6] Richard M Wilson. Graph puzzles, homotopy, and the alternating group. *Journal of Combinatorial Theory, Series B*, 16(1):86–96, 1974.