Some progress toward tethered pairs

Reza H. Teshnizi and Dylan A. Shell

1 Introduction and Problem Statement

Recently there has been surge of research in motion planning for tethered robots, e.g., Igarashi and Stilman (2011). When a mobile robot is tethered, the base point to which it is tethered is most often itself mobile, even if only at substantially greater cost. As illustrated in Figure 1, connecting a pair of agents together can benefit both. Indeed, Kim et al (2013) explored how a cable connecting robots can be used to manipulate objects. Thus, we are interested in developing a method for planning motions of two robots, r^+ and r^-, that are connected to each other via a cable. We believe this problem can be addressed by extending the methods developed in our prior work (Teshnizi and Shell, 2014; Teshnizi, 2015) for planning motions of a single tethered robot.

We refer to this problem as the two tethered robot motion planning (2TRMP). In order to develop the foundation of a method, we assume the obstacles and robots are points in $\mathbf{R}^2$ for simplicity. We also assume, similar to our preliminary work, that the tethering cable is of the retracting type (i.e., it can be assumed to be always taut). Thus, we are permitted to model its configuration with a set of straight line segments. We assume we are given: (i) a pair of points $(P^s_{r_+}, P^s_{r_-})$ describing the (non-oriented) robots' initial locations; (ii) a pair of points $(P^d_{r_+}, P^d_{r_-})$ describing the target locations; (iii) a set of obstacles; (iv) a path describing the tether's current configuration and its maximal length L. A solution to the 2TRMP is a pair of paths which, when executed simultaneously by the pair of robots brings them to their target locations and that the length constraint of the cable is never violated *en route*. (In general the execution phase of the robots may involve one robot waiting for the other if the cable is taut, but the robots can compute when to wait locally). Otherwise, if no such pair of paths exists, solving the 2TRMP requires an indication of this fact. The optimality of the solution is dependent on some specific cost function; we consider sum of robot traversal distances.

Fig. 1: Climbers moving in a rope team illustrates how connections between agents helps provide balance between freedom and security. Image source: http://mtbakerguides.com/

2 Progress to date

We begin by partitioning the obstacle points into three sets. The first set are obstacles around which the tether, in its initial pose, is currently contacting. The second set are those within the ellipse whose foci are $P^d_{r_+}$ and $P^d_{r_+}$ with semimajor axis $\frac{L}{2}$. The remaining points form the third set. Figure 2 gives an example of this process. If, during the execution of a given solution, the cable contacts one (or possibly more) of the green obstacles, the robots are still able to reach their destinations. After execution of any feasible solution the final configuration of the cable never has a contact with any red obstacle.

We consider the problem as belonging to one of two classes: a (C1) *Category 1 Solution* to the 2TRMP problem keeps at least one black obstacle in contact with the cable after its execution. Otherwise, if not a

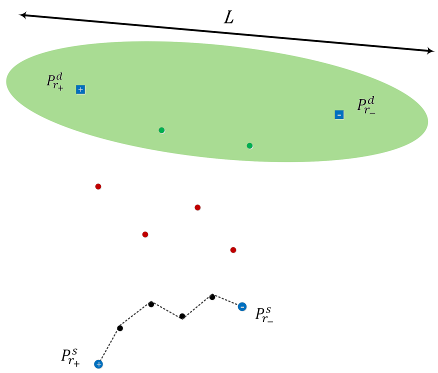


Fig. 2: A simple 2TRMP instance showing the ten obstacles that have been partitioned into the three sets we describe, shown by color (black, green, red, respectively).

Reza H. Teshnizi E-mail: reza.teshnizi@gmail.com

Dylan A. Shell
E-mail: dshell@tamu.edu

Computer Science and Engineering, Texas A&M University, College Station, TX, USA

C1 solution, we it is a (C2) *Category 2 Solution.* This provides some structure to the solution:

- **Proposition 1:** An optimal solution always starts with detaching the cable from zero or more obstacles and continues with attaching the cable to zero or more obstacles.
- **Proposition 2:** The shortest C1 solution is always shorter than all the C2 solutions.

The above propositions are a starting point for a method. We will begin our search by looking for C1 solutions. If no C1 solutions are found, we will look into C2 solutions and pick the optimal, if such a solution exists. Due to the combinatorial nature of C2 solutions, the process of finding the optimal C2 solution is much more computationally expensive than finding a C1 solution.

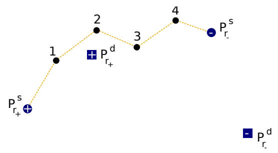
As of now, we are able to make use of our prior work to find C1 solutions efficiently. We are aware of some heuristics to simplify search for C2 solutions but these are *ad hoc* and, consequently, are not discussed further.

Finding C1 Solutions

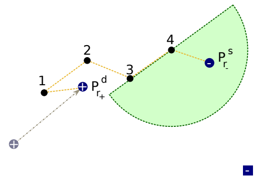
A C1 solution requires the cable connecting the two robots to touch at least one point obstacle in the environment. In our prior work (Teshnizi and Shell, 2014), we describe a decomposition of the c-space based on visibility cells. In the initial condition for a C1 solution, the two robots reside in two different visibility cells. The key to finding a solution is that the two robots share a fixed amount of cable, so that L is the most they can possibly consume individually for reaching their respective destinations.

We give only a sketch of the algorithm and show its operation. Basically, we employ a Dynamic Programming approach. We consider all the solutions that fit into the definition of a C1 solution, but reusing partial solutions to save work. To find the optimal solution to the problem we construct multiple n-element arrays. The first array, d_{r_+}, in which the ith element holds the shortest path for r^+ from $P^s_{r_+}$ to $P^d_{r_+}$ such that after the execution of the motion, the cable is still contacting obstacle i. A similar array, d_{r_-}, is constructed for r. We also store the consumed cable associated to the paths in d_{r_+} and d_{r_-} in two n-element arrays, c_{r_+} and c_{r_-}. The ith element in c_{r_+} holds the length of the cable consumed from obstacle i to the destination. Once the values for d_{r_+}, d_{r_-}, c_{r_+} and c_{r_-} have been determined, finding the optimal solution is straightforward. It is worth noting that this procedure does not rely on our specific planner.

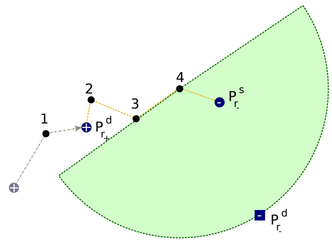
The table in Figure 4 shows the values stored in the respective arrays for the example shown opposite.



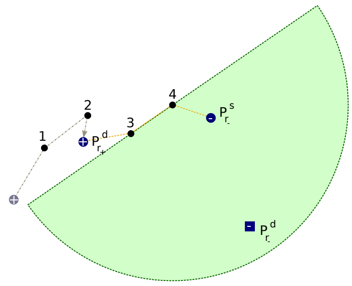
(a) The problem input.



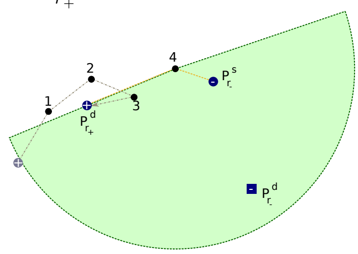
(b) Robot r^+ goes directly from $P^s_{r_+}$ to $P^d_{r_+}$. The green circle sector shows the radius that is reachable to r with the remaining cable length.



(c) The shortest path for r^+ that untangles obstacle 1 while going from $P^s_{r_+}$ to $P^d_{r_+}$. This frees more cable for r to use.



(d) Here r^+ untangles obstacles 1 and 2 while going from $P^s_{r_+}$ to $P^d_{r_+}$.



(e) Obstacles 1, 2, and 3 are untangled *en route.*

Fig. 3: A simple example.

Figure No.	Motion Steps by r^+	Travelled Distance	Cable Config. after r^+'s Motion	Cbl. Consumed by r^+	Cbl. Left for r^-
3(b)	$P^s_{r^+}$, $P^d_{r^+}$	28.3	$P^d_{r^+}$, 1, 2, 3, 4, $P^s_{r^-}$	75.9	24.1
3(c)	$P^s_{r^+}$, 1, $P^d_{r^+}$	31.9	$P^d_{r^+}$, 2, 3, 4, $P^s_{r^-}$	55.4	44.6
3(d)	$P^s_{r^+}$, 1, 2, $P^d_{r^+}$	46.1	$P^d_{r^+}$, 3, 4, $P^s_{r^-}$	45.3	54.7
3(e)	$P^s_{r^+}$, 1, 2, 3, $P^d_{r^+}$	67.8	$P^d_{r^+}$, 4, $P^s_{r^-}$	43.6	56.4

Fig. 4: A table showing the dynamic programming form of the solution for the problem in Figure 3.

References

Igarashi T., Stilman M. (2011) Homotopic path planning on manifolds for cabled mobile robots. In: Algorithmic Foundations of Robotics IX, Springer, pp 1–18

Kim S., Bhattacharya S., Heidarsson H. K., Sukhatme G., Kumar V. (2013) A topological approach to using cables to separate and manipulate sets of objects. In: Robotics: Science and Systems

Teshnizi R. H. (2015) Motion planning for a tethered mobile robot. Master's thesis, Texas A&M University, Department of Computer Science & Engineering

Teshnizi R. H., Shell D. A. (2014) Computing cell-based decompositions dynamically for planning motions of tethered robots. In: Robotics and automation (ICRA), 2014 IEEE International Conference On, IEEE