Research Problems in Algorithmic Motion Planning

Ethan D.H. Kim
School of Computer Science, McGill University
Montreal, Canada H3A 2A7
ethan@cs.mcgill.ca

December 13, 2005

Abstract

This report reviews the literature discussed in class, with an emphasis on the graph theoretic approaches used to model and solve the motion planning problems. Furthermore, we highlight several open problems that were raised in the domain of algorithmic motion planning.

1 Graph Models

Many problems in the real world are continuous in nature, and dealing with continuous motion can be computationally difficult. In the domain of algorithmic motion planning, techniques from discrete mathematics are used to discretize the problems. Graph models, in particular, help us represent the continuous real world environment as a discrete space, enabling us to design simpler algorithms.

1.1 Locating and Capturing an Evader in a Polygonal Environment

In the Capturing an Evader paper [IKK04], the entire polygonal environment is represented as a graph. The authors first assume that the environment is a connected simple polygon P without holes. Using the technique from the Computational Geometry literature, the polygon P is first triangulated, and the dual graph of the triangulation is used to model their environment. It turns out that this dual graph is a *tree*, which is a very pleasing result for the capturing an evader scenario. Because there is always a unique path between two vertices in a tree, the search space becomes simple and small as the pursuer tries to locate the evader. By using a randomized strategy to locate the evader, the robot chooses a subtree adjacent to its current position. This gives us a bound on the probability of locating the evader.The graph model used in this paper, therefore, plays a crucial role to discretize and solve the problem in an extremely efficient manner.

1.2 Multi-Step Motion Planning for Free-Climbing Robots

The graph model used in [BLLR04] is a less obvious abstraction than that of [IKK04]. Here, the authors first look at the set of constraints from the robot model such as equilibrium, collision, joint-torque limits, etc. The given *map* or *wall* contains a set of holds that the robot can hold onto. The configuration of the robot's arms can be computed satisfying the constraints. Thus, each node in the graph represents a component of such feasible configuration, and two vertices are adjacent by an edge if the robot can move from one configuration to another by a single step move.

In this model, however, there are practical problems to consider. First, when the robot is holding onto a set of holds in the environment, it can have several different *stances* satisfying the constraints, and based on these stances can the robot move to an adjacent feasible space. However, if we represent every configuration as a single node, the graph can be extremely large and contain cycles, which in turn adds to the complexity of the search space. In particular, given the DOF(degree of freedom) of the robot and the set of handles in the environment, it is impractical to preprocess all possible configurations to construct the entire graph. The authors instead chose an incremental sample-based approach, which samples a set of transition points adjacent to the current stance. The use of graph search heuristics and *lazy* approach to checking the feasibility of edges can have disadvantages, however. Since the graph structure is not known ahead of time, we may introduce redundant nodes and edges to our graph, increasing the time to plan our motion. Furthermore, when sampling for the adjacent transition points with some probability, we may be missing critical transition points that are on the path to the goal position, reducing the probability of success. Thus, there is a trade-off between efficiency of the algorithm and probability of success.

1.3 Robotic Origami Folding

The thesis Robotic Origami Folding [Bal04] studies motion planning for folding a paper or metal sheets into a desired state. For the theoretical part of the thesis, the paper is assumed to be an articulated rigid body, in which the paper can be folded only at the creases in the given crease pattern. One graph model used in this work is called a *facet graph*: take each facet of the paper to be a node in the graph, and connect adjacent facets on the pattern with an edge. Any spanning tree of this graph is called a *facet tree.* These two graphs from a crease pattern can help us study the properties of origami folding. For example, from the result of graph theory, we know that a tree can be 2-colored. Thus, if we color the vertices of the graph by walking along the facet tree, any flat origami is two-colorable. Moreover, the following theorem can be derived using the facet tree: *The facet graph and the goal stacking order completely determine which creases are mountain creases and which are valley creases.*

Not only can we study the properties of a given fold, we could also design a motion planner for the folding. Let each vertex of a graph represent a flat origami state, and connect two vertices by an edge if one state can be achieved from another by a book-fold. In chapter 4, the author describes a motion planner for a book-foldable origami, which uses the state graph to search for the steps needed to take from the initial state to the goal folding state.

1.4 Conclusion

Graph structure and configuration space are useful tools for modeling the problems in algorithmic motion planning. As in the Capturing an Evader paper [IKK04], sometimes the underlying notion of a graph is natural, and we are to devise an algorithm to exploit the structure of the given graph. In these cases, constructing the graph is easy(for example, a linear time algorithm exists for triangulation of a polygon) and the difficulty lies in the algorithm to solve our problem on the graph. In other cases, as in Origami Folding thesis [Bal04] and Stealth Tracking paper [BLLR04], the abstraction of the problem involves a lot of work to construct the desired graph structure. Once such graph is constructed, the problem can be solved trivially using well known algorithms such as shortest path algorithm on graphs. In both cases, however, these graphs pose an ideal abstraction of the real world, and practical decisions must be made in order to obtain an efficient and exact algorithm to solve our problems.

2 Open Problems

The papers discussed in class have uncovered a wide range of open problems for future research. From the Capturing an Evader paper [IKK04], the pursuer had a map of the polygonal environment, with which the dual tree of the triangulation is computed. An interesting work in [TLM03] shows that the moving robot does not necessarily need a global map to navigate the environment. An open problem has been raised in this account: given only a minimal representation of the environment, can we devise a motion planner for the pursuer to capture the evader?

Also, the authors used two pursuers to locate and capture the evader in steps polynomial in the number of vertices of the polygon. When the same algorithm is applied with only one pursuer, the algorithm still terminates with a finite number of steps, but it is not clear whether the number of steps required is bounded by a polynomial in the number of vertices. A further investigation is required to answer this question.

In the Free-Climbing Robot paper [BLLR04], a pre-surveyed model of the terrain is assumed. However, in real world applications, the robots often do not have a global vision on its environment, and only a limited sensing ability is allowed. With local sensing ability, the robot will have to explore the environment incrementally, and a future work may be to plan the motion to reach its goal position using this incremental search for transition configurations. Moreover, the algorithm discussed in this paper are approximate as well as probabilistic – it would be a challenging task to devise an algorithm to compute an exact solution in an efficient manner.

In the Stealth-Tracking paper [BLJH04], on the other hand, only a local visibility is assumed. Some of the limitations of the authors' approach is due to the limitations in sensing ability. For instance, the lookout region may suddenly disappear, because the tracker uses only local information from the sensor. Should the robot have global information, it may carry out a more sophisticated strategy to track the target with improved performance.

Finally, the Robotic Origami Folding thesis [Bal04] raised a whole family of open problems. The families of folding types discussed in the thesis do not cover all possible folds that are known to classical origami folders. For example, in the case of multi-vertex origami pattern, very little is known about the configuration space of the rigid body model. Only single-vertex folding origami is studied in Chapter 5, and we still do not have a complete planner for vertex folding origami models.

Section 5.6 discusses an example of 3D origami folding, using the shopping-bag example. It is shown that the shopping bag example cannot be collapsed into a flat state, but with a finite number of additional creases, it can be collapsed [BDD04]. However, the collapsed pattern of this shopping bag is different from the original collapsed pattern that we can see with real paper shopping bags. Thus, the obvious question then is, given the collapsed shopping bag, can it be unfolded to its open state, using only a finite number of additional creases. The authors give a conjectured solution in [BDD04] and [Bal04].

3 Research Problems

3.1 Evader Problem

In the work of Isler, Kannan, and Khanna [IKK04], the authors studied the motion planning of the pursuer capturing the evader. A similar question can be asked from the evader's perspective: given a polygonal environment, we wish to plan a motion for the evader such that the evader can safely exit the room without being captured (or seen) by the pursuers.

Many different versions of the problem can be considered depending on the environment model. If the pursuers are simple watch-posts fixed at some stationary positions with limited visibility and the evader has a global vision, the problem can be reduced to the classical piano mover's problem, where each lookout post is considered to be an obstacle that the robot must avoid. Due to Schwartz and Sharir [SS83], a polynomial time solution for this problem is known to exist. Little is known, however, for evaders with limited visibility. For example, the following question can be asked:

Problem: Escaping from watch-posts. Given a polygonal map M with possible obstacles(or holes) and a set P of stationary watch-posts with visibility r_P, plan a motion for an evader E with visibility r_E. Assume that the evader has the map of the environment, but the positions of the watch-posts are unknown.

Clearly, if $r_e \leq r_p$, the problem cannot be solved. It is not clear, however, that the problem can be solved even when the evader always has a greater visibility than the watch-posts.

For another version of the problem, suppose that there are multiple pursuers randomly moving around to guard the room. As before, the pursuers have a limited visibility r_p, but the evader now has a global vision. Now the problem becomes similar to that of the famous computer game *Pac-man*, where the player tries to get to goal positions while avoiding the pursuers.

One can ask many more questions of similar nature by changing the visibility constraints on the mobile agents. How to model and control objects in a dynamic environment remains a challenging problem, in spite of the fact that this problem arises in major application areas such as robotics and computer simulations. Tovar, LaValle and R. Murrieta [TLM03] studies navigation techniques for a mobile agent without a geometric map or localization. The dynamic data structure for robot navigation described in this paper will be a great starting point for planning a motion with limited visibility.

3.2 Computational Origami

Computational Origami is a recent field of computer science studying algorithms for paper-folding problems. For a survey on recent results in Computational Origami, see [DD01]. The thesis of Balkcom [Bal04] uncovered a plethora of open problems to be investigated. In particular, the shopping bag example is studied extensively and left us an open problem discussed in previous section. Here we give a generalized version of the unfolding problem.

Problem: Collapsing Open Polyhedra. Given a three-dimensional open polyhedra, can it be collapsed into a flat state, using a finite number of creases?

Due to bellows theorem([Con78], [CSW97]), no polyhedron with a fixed, finite number of creases can be deflated into a flat state. Thus, it is necessary for a polyhedron to be *open* if we wish to collapse it into a flat state. This question was answered positively by [Bal04] for the shopping bag model, which is a hexahedron with one face missing. It would be interesting to see the same result for all regular polyhedra, and also for other types of polyhedra.

Since we are allowing additional crease patterns, we might also think about designing our own crease patterns from scratch, in order to achieve the final goal state. Thus the following problem:

Problem: 3D Origami from scratch. Given a rectangular piece of paper and a final three dimensional polyhedra, can we design the crease pattern and set of instructions to fold into the goal polyhedra?

In this problem, we are allowed to glue the edges of facets if needed. The corresponding question for flat-foldable origami folding has been answered positively by [Lan96].

4 Final Remark

Problems in algorithmic motion planning are known to be computationally hard in general. Many algorithmic techniques as well as advanced data structures are introduced to deal with a highly dynamic environment, and as we saw in Section 1, graph theoretic approach is one way of discretizing and simplifying the complex environment. However, there often remains a trade-off between exactness of the solution and efficiency of the algorithms. Moreover, many open problems can be raised from a well-known problem with slight modifications to the problem constraints. With such variety of problems that are difficult to solve, it may be useful to obtain more lower bound results, hence letting us focus on approximate or probabilistic approaches.

References

[Bal04] D. Balkcom. *Robotic Origami Folding.* PhD thesis, The Robotics Institute, Carnegie Mellon University, 2004.

[BDD04] D. Balkcom, E. Demaine, and M. Demaine. Folding paper shopping bags. *Proceedings of the 14th Annual Fall Workshop on Computational Geometry*, 2004.

[BLJH04] T. Bandyopadhyay, Y. Li, M.H. Ang Jr., and D. Hsu. Stealth tracking of an unpredictable target among obstacles. *Workshop on the Algorithmic Foundations of Robotics*, 2004.

[BLLR04] T. Bretl, S. Lall, J-C. Latombe, and S. Rock. Multi-step motion planning for free-climbing robots. *Workshop on the Algorithmic Foundations of Robotics*, 2004.

[Con78] R. Connelly. Conjectures and open questions in rigidity. *Proceedings of the international congress of mathematicians*, 1978.

[CSW97] R. Connelly, I. Sabitov, and A. Walz. The bellows conjecture. *Contributions to Algebra and Geometry*, 1997.

[DD01] E. Demaine and M. Demaine. Recent results in computational origami. *Proceedings of the 3rd International Meeting of Origami Science, Math, and Education*, 2001.

[IKK04] V. Isler, S. Kannan, and S. Khanna. Locating and capturing an evader in a polygonal environment. *Workshop on the Algorithmic Foundations of Robotics*, 2004.

[Lan96] Robert J. Lang. A computational algorithm for origami design. *12th Symposium on Computational Geometry*, 1996.

[SS83] J. Schwartz and M. Sharir. On the piano mover's problem ii. *Advances in Applied Mathematics*, 1983.

[TLM03] B. Tovar, S. M. LaValle, and R. Murrieta. Optimal navigation and object finding without geometric maps or localization. *IEEE International Conference on Robotics and Automation*, 2003.