The Rectilinear Minimum Bends Path Problem in Three Dimensions

David P. Wagner * Robert Scot Drysdale*

Clifford Stein†

http://www.cs.dartmouth.edu/~dwagn
http://www.cs.dartmouth.edu/~scot
http://www.ieor.columbia.edu/~cliff

October 11, 2002

Abstract

This is a presentation of a "work-in-progress", on which which we are still working out many details, but which we feel may still be of interest to the community.

We consider the Recitilinear Minimum Bends Path Problem among Rectilinear Obstacles in Three Dimensions. The problem is well studied in two dimensions, but is relatively unexplored in higher dimensions. We give an algorithm which solves the problem in worst-case $O(n^{5/2}\log^2(n))$ time. Previously known algorithms have a worst-case running time of $O(n^3)$ [1].

The Rectilinear Minimum Bends Path problem has been considered by several authors for over a decade. The problem attempts to find the path among obstacles between two points which minimizes the number of times a turn is made. The algorithms of Mitchell, et al., and independently of Lee, et al. solve the two-dimensional problem in $O(n\log(n))$ time, where n is the number of corners among all obstacles [5, 4, 3, 8, 9, 2, 6, 7] .

Here we investigate the problem in higher dimensions. In particular we consider the three-dimensional problem. This version of the problem has direct applications in robotics.

Our general approach to this problem will be to think of it in the graph theoretic sense as much as possible. Thus our method is to first translate the input into a graph, and then to use techniques from known graph algorithms. Translating the input into a graph is accomplished by mapping obstacle faces to graph nodes. This allows us to consider paths which move between obstacle faces, and represent them as paths through the graph.

An important aspect of this approach is our ability to prove that it is sufficient to consider only these types of paths. That is, that we only need to consider those paths which move between obstacle faces.

We show how to find an optimal path which touches an obstacle face often enough, so that we can bound the number of bends between segments which touch an obstacle face. Thus adjacent nodes in the corresponding graph are those which can be reached by a path having a bounded number of bends. Edges between such nodes can then be weighted by the number of bends between the corresponding obstacle faces.

Here is an overview of our method:

- Translate the set of obstacle faces into nodes of a graph. Include the starting and finishing points in this set.
- Create a set of explored nodes, which initially contains only the starting point. Mark this node as unexpanded, having bend distance zero.
- Repeat the following until the finishing point is

*Dartmouth College, Department of Computer Science {scot,dwagn}@cs.dartmouth.edu. Research partially supported by a Dartmouth Fellowship, and The NSF/KOSEF Summer Institute in Korea.

†Columbia University, Department of Industrial Engineering and Operations Research cliff@ieor.columbia.edu.

explored or there are no new nodes reachable.

- Choose a node, x, from the set of explored, unexpanded nodes, having the minimum bend distance in this set.
- Expand x, finding all nodes which it is adjacent to.
- For each node y found
  - If y is unexplored and unexpanded then mark it as explored and unexpanded. Set $bend\_distance(y) = bend\_distance(x) + 1$.
  - If y is unexplored and expanded $bend\_distance(y) = \min(bend\_distance(y), bend\_distance(x) + 1)$.
- Mark x as expanded.

The similarities between the Minimum Bends Path Problem and the Shortest Path Problem allow us to use many methods from Dijkstra's well known solution to the Shortest Path Problem. An essential feature of Dijkstra's algorithm is the existence of a priority queue containing nodes from the graph. This is indexed by a key value, and our choice for the key is the number of bends it takes to reach the location. Thus, a single "location" should represent a homogeneous region of space, all of which can be reached with the same number of bends.

Due to the possibility of adjacent obstacles, different regions of the same face may be reachable with different numbers of bends. Therefore, we cannot make a direct, one-to-one mapping between obstacle faces and "locations".

Each of the $O(n)$ faces has the potential to be partitioned by any or all of the $O(n)$ obstructing obstacles approaching the plane containing the face from the opposite side. It follows that each face can be partitioned into as many as $O(n)$ separate regions, for a total of $O(n^2)$ regions.

We are able to reduce this somewhat by noticing that a pair of coplanar rectangular regions can be joined into a single, larger region, if there are no obstructing obstacles within the region. We can show that this reduces the total number of regions to $O(n^{3/2})$. It is these regions which we use as our "locations".

Each invocation of our implementation of the expand step, runs in $O(n \log^2(n))$ time. Since it will be called $O(n^{3/2})$ times, the expand step has an total running time of $O(n^{5/2} \log^2(n))$. It is this step which dominates the overall running time of the algorithm.

References

[1] Robert Fitch, Zack Butler, and Daniela Rus. 3D recitilinear motion planning with minimum bend paths. In *International Conference on Intelligent Robots and Systems*, Wailea, Maui, HI, 2001.

[2] D.T. Lee, C.D. Yang, and C.K. Wong. Problem transformation for finding rectilinear paths among obstacles in two-layer interconnection model. Technical Report 92-AC-104, Dept. of EECS, Northwestern University, 1992.

[3] D.T. Lee, C.D. Yang, and C.K. Wong. Rectilinear paths among rectilinear obstacles. *Discrete Applied Mathematics*, 70, 1996.

[4] J.S.B. Mitchell. L_1 shortest paths among polygonal obstacles in the plane. In *Algorithmica*, pages 55–88, 1992.

[5] J.S.B. Mitchell, C. Piatko, and E.M. Arkin. Computing a shortest k-link path in a polygon. In *Proceedings of the 33rd Annual Symposium on Foundations of Computer Science*, pages 573–582, 1992.

[6] C.D. Yang, D.T. Lee, and C.K. Wong. On bends and lengths of rectilinear paths: a graph-theoretic approach. *Internat. J. Comp. Geom. Appl.*, 2:61–74, 1992.

[7] C.D. Yang, D.T. Lee, and C.K. Wong. On minimum-bend shortest recilinear path among weighted rectangles. Technical Report 92-AC-122, Dept. of EECS, Northwestern University, 1992.

[8] C.D. Yang, D.T. Lee, and C.K. Wong. Rectilinear path problems among rectilinear obstacles revisited. *SIAM Journal on Computing*, 24:457–472, 1992.

[9] C.D. Yang, D.T. Lee, and C.K. Wong. On bends and distance paths among obstacles in two-layer interconnection model. *IEEE Transactions on Computers*, 43:711–724, 1994.