Routing a Maximum Number of Disks through a Scene of Moving Obstacles

Joondong Kim *
jdkim@ams.sunysb.edu
Dept. of Applied Mathematics and Statistics
Stony Brook University
Stony Brook, NY 11794-3600, USA

Joseph S. B. Mitchell †
jsbm@ams.sunysb.edu
Dept. of Applied Mathematics and Statistics
Stony Brook University
Stony Brook, NY 11794-3600, USA

Valentin Polishchuk
valentin.polishchuk@helsinki.fi
Helsinki Institute for Information Technology
University of Helsinki and Helsinki University of Technology
P. O. Box 68, FI-00014, University of Helsinki, Finland

Arto Vihavainen
avihavai@cs.helsinki.fi
Computer Science Dept.
University of Helsinki
P. O. Box 68, FI-00014, University of Helsinki, Finland

ABSTRACT

This video illustrates an algorithm for computing a maximum number of disjoint paths for unit disks moving among a set of dynamic obstacles in the plane. The problem is motivated by applications in air traffic management: aircraft must be routed while avoiding no-fly zones and weather constraints and while maintaining at least a specified horizontal separation distance between themselves. Given a polygonal domain with moving obstacles, our goal is to determine the maximum number of unit disks (aircraft with safety zones) that can be routed safely through the domain, entering/exiting through specified edges of the domain.

The video is meant to accompany the paper [1], which gives details of the algorithm and its analysis.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*Geometrical problems and computations*; I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*Geometric algorithms, languages, and systems*

General Terms

Algorithms

1. INTRODUCTION

Motivated by applications in air traffic management (ATM), we consider the problem of computing multiple disjoint paths for a set of disks moving at (approximately) constant speed among a set of static or moving obstacles. In the ATM application, the obstacles correspond to *special use airspace* or *hazardous weather constraints*, and the disks correspond to aircraft, each of which is surrounded by a *protected airspace zone*, which no other aircraft should enter. Typically, two aircraft at approximately the same altitude should never be within 5 nautical miles of each other; thus, within a single flight level, we model the problem as the motion of disks of radius 2.5 miles. The hazardous weather constraints are modeled as polygonal obstacles in motion, given by weather forecast data.

Our goal is to determine the maximum *throughput*, or *capacity*, in terms of number of disks that can be routed through a given polygonal region P, entering the region through a *source edge* of P and exiting through a *sink edge* of P. We assume that the obstacles move along known trajectories, given by weather forecasts.

This video animates the pseudopolynomial-time dual-approximation algorithm that is described and analyzed in Arkin et al. [1]. If K is the maximum number of unit disks moving with speed at most 1 that can be routed, then the algorithm computes routes for at least K disks of radius $\Omega(1)$ moving with speed $O(1)$. More precisely, the algorithm finds at least K routes for disks of radius $(1/3-1/2\Delta t)^2$ moving with speed at most $10/\Delta t$, for any choice of the time discretization step $\Delta t < 1/2$. This means that the algorithm produces just as many routes as can be routed optimally for unit disks with maximum speed 1, but it does so by compromising on the radius of the disks being routed (making them smaller, to allow for more flexibility in routing) as well as the maximum speed of motion of the disks (allowing them to move faster, by a constant factor, in order to maneuver around each other and the obstacles).

The method is to compute a maximum flow with *forbidden pairs* in a certain discrete "adaptive" grid graph in space-time. Since the maximum flow with forbidden pairs problem is hard (even to approximate), the algorithm of [1] instead avoids forbidden pairs by reducing the radii of the moving disks by an amount sufficient to obtain disjoint routes that deflect slightly to avoid each other, while not causing any other interferences between pairs of routes. The running time is polynomial in the largest coordinate of the domain, and in $1/\Delta t$.

The main intent of the video is to show an animation of the al-

*J. Kim is partially supported by NASA Ames and Metron Aviation.

†J. Mitchell is partially supported by the National Science Foundation (CCF-0431030, CCF-0528209, CCF-0729019), NASA Ames, and Metron Aviation.

SCG'08, June 9–11, 2008, College Park, Maryland, USA.
ACM 978-1-60558-071-5/08/04.

gorithm of [1] in three-dimensional space-time, (x,y,t); ideas of the algorithm are best viewed dynamically or with 3D graphics in space-time. (See e-version of the paper for color images.)

2. DESCRIPTION OF VIDEO

The video was created with the software Pov-Ray.

The video opens by showing images from the motivating ATM application. It then shows the polygonal domain P, which is a rectangular region with polygonal obstacles in red. Flights of significantly different headings are usually kept altitude-separated (in z): The video shows aircraft surrounded by blue disks flying left-to-right, while aircraft surrounded by green disks are flying top-to-bottom; while they appear to be in conflict in the 2-dimensional projection, the 3-dimensional view in (x,y,z) shows that they are safely separated.

We next highlight in blue the *source* and *sink* edges of P. The goal is to route disks through P, entering P through the source edge and exiting P through the sink edge. The red obstacles are shown moving (slowly).

The algorithm works within 3-dimensional space-time, (x,y,t). The video illustrates this with 3-dimensional views. (Note that these are different from the earlier 3-dimensional views in (x,y,z) space.) In space-time, the moving obstacles define "tilted" polyhedral obstacles (shown in red), whose slope with respect to the (x,y)-plane depends on the speed of motion. Similarly, the trajectories we compute for disks define slanted and tilted *tubes*, shown in blue, which interconnect intermediate *waypoints*, where the disk changes heading (or speed). Refer to Figure 1(a). The blue tubes should be pairwise-disjoint in addition to being disjoint from the obstacles.

Consider one trajectory, which defines a blue tube in space-time. Using the fact that obstacles move slowly, we know that no obstacle penetrates a right circular cylinder (green) within a portion of the tube. The stack of such cylinders through a tube (Figure 1(b)) is known to be clear of obstacles in space-time; thus, we know that there exists a feasible chain of oblique cylinders (shown in yellow); see Figure 2. Our algorithm computes a set of such chains by discretizing space-time, slicing time into equal-length intervals, and packing within each slice a maximal number of disks of radius 1/2. The algorithm searches the *motion graph*, whose nodes are the disks in the packings and whose edges interconnect pairs of disks from consecutive time slices that correspond to feasible motions (feasible with respect to the obstacles and the speed bound of the disks); see Figure 3(a). The graph is augmented by a super-source and a super-sink, connected to disks on the boundary by the source and sink edges, respectively (Figure 3(b)).

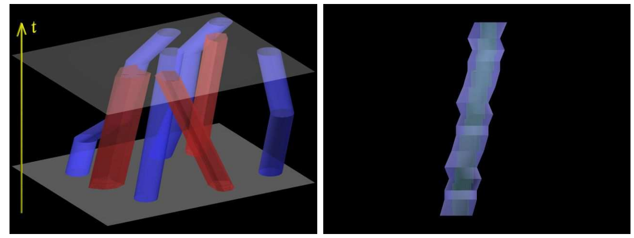
The algorithm computes a maximum number of disjoint paths from super-source to super-sink in the motion graph (Figure 4(a)). These disjoint paths correspond to obstacle-free yellow tubes (Figure 4(b)). While these tubes are individually feasible for disks, the tubes can intersect each other. In order to get disjoint tubes within these original tubes, we decrease the radius of the tubes and slightly bend them, as shown in animation in the video.

Acknowledgments

We thank Alexander Kröller, Jimmy Krozel, and Girish Sabhnani for helpful comments and input.

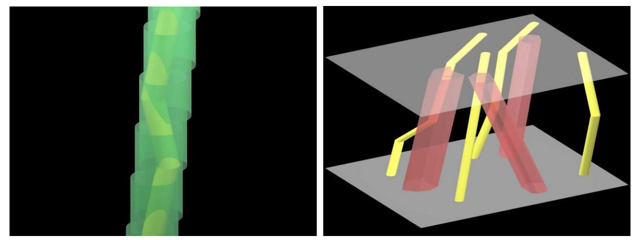
3. REFERENCES

[1] E. M. Arkin, J. S. B. Mitchell, and V. Polishchuk. Maximum thick paths in static and dynamic environments. *These proceedings*.



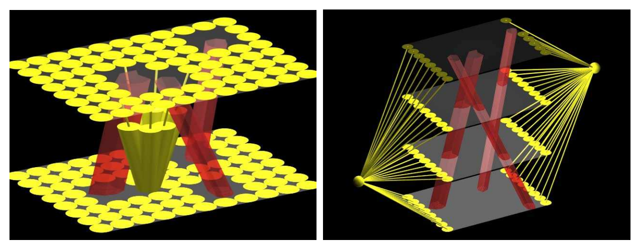
(a) (b)

Figure 1: (a) Red tilted obstacles and blue tubes within space-time, and (b) stack of green cylinders within a blue tube.



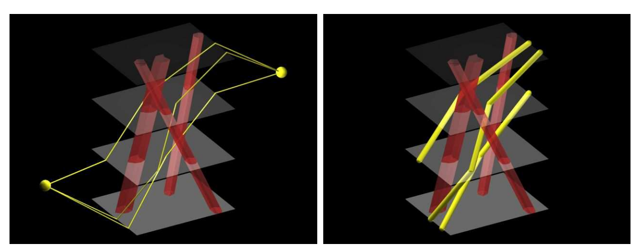
(a) (b)

Figure 2: (a) Smaller yellow tubes within the stack of green cylinders and (b) disjoint yellow tubes among red obstacles in space-time.



(a) (b)

Figure 3: (a) The motion graph edges interconnecting packed disks in consecutive time slices and (b) the super-source and super-sink connected to disks along the source and sink edges.



(a) (b)

Figure 4: (a) The disjoint routes computed (in yellow) and (b) the corresponding thick yellow tubes.