Visualization of Trajectories*

Stephen Borgatti[1], Ulrik Brandes[2], Michael Kaufmann[3], Stephen Kobourov[4], Anna Lubiw[5], and Dorothea Wagner[6]

[1] Gatton College of Business, University of Kentucky, Dept of Management
Lexington, KY 40506, USA
sborgatti@uky.edu
[2] University of Konstanz, Department of Computer & Information Science
D-78457 Konstanz, Germany
ulrik.brandes@uni-konstanz.de
[3] Wilhelm-Schickard Institut für Informatik, Arbeitsbereich für Paralleles Rechnen
Sand 14, D-72076 Tübingen, Germany
mk@informatik.uni-tuebingen.de
[4] University of Arizona, Department of Computer Science
1040 E. Fourth St, Tucson, AZ 85721-0077, USA
kobourov@cs.arizona.edu
[5] University of Waterloo, David R. Cheriton School of Computer Science
Waterloo, Ontario, Canada N2L 3G1
alubiw@uwaterloo.ca
[6] University Karlsruhe, Faculty of Informatics,
76128 Karlsruhe, Germany
wagner@ira.uka.de

Problem description

We considered the following problem: Given a set of vertices V and a set of paths P, where each path is a sequence of vertices, represent these paths somehow.

We explored representations in different dimensions and with different conditions on the paths.

1-D

In this case we want a linear order of V. Possible conditions on paths:

(a) each path is increasing
(b) each path is "monotone" - i.e. either increasing or decreasing
(c) each path is "unimodal" - i.e. increasing for some prefix and decreasing for the rest. Note that the prefix or suffix may be empty, so a monotone path is unimodal.

* Work on this problem began at Dagstuhl Seminar No. 08191 "Graph Drawing with Applications to Bioinformatics and Social Sciences", May 2008.

Dagstuhl Seminar Proceedings 08191
Graph Drawing with Applications to Bioinformatics and Social Sciences
http://drops.dagstuhl.de/opus/volltexte/2008/1555

2-D

In this case each vertex of V is represented as a point in the plane. Possible conditions on paths:

(a) each path is increasing in x-order or increasing in y-order
(b) each path is monotone in x-order or monotone in y-order
(c) for each path there is some direction in which it is montone

Note that unimodal is also a possibility, but we didn't yet explore that far.

Results

1-D (a)

We include this case only to point out that it is just a question of whether the resulting directed edges form an acyclic graph.

1-D (b)

The decision version of this problem is NP-complete even for paths of lenghts two [J. Opatrny: Total ordering problems. SIAM J. Computing 8(1):111-114, 1979].

1-D (c)

This can be tested in linear time. We eliminate the points one by one, placing them from left to right. Suppose vertex set U remains; consider the subpaths induced by U. As the next vertex to eliminate, choose a vertex v of U such that no path uses v as an intermediate node. Paths may originate at v and may terminate at v.

Claim 1: We obtain a valid solution for U by adding v at the left of a valid solution for U-v. Proof. Adding a new origin/terminus at the left of a unimodal path yields a unimodal path.

Claim 2: If no such v exists, then there is no unimodal ordering. Proof. We prove the contrapositive: if there is a unimodal ordering then such a v alwyas exists. Take a unimodal ordering and let v be the first vertex of U in this ordering.

2-D (a) (b)

We didn't have anything on these.

2-D (c)

Observation 1. If every path has at most 2 edges, then any non-degenerate placement of the points will do. Note that this extends to paths of 3 edges in 3-D.

Observation 2. A monotone path is non-self-intersecting. Thus, if there is a representation as monotone paths, then there is a simultaneous planar embedding. The converse is not true, but if can find a simultaneous planar embedding such that paths remain non-self intersecting even when edges are made longer, then this is a monotone embedding.

Observation 3. (c) is more general than (b): consider 3 points a, b, c and paths abc, bca, cab. By observation 1 any placement of points makes all paths monotone. However, if we are restricted to x and y directions of montonicity, then some 2 paths must share the x direction (say). But no two of those paths have a 1-D monotone ordering.

Other Variant

In 1-D an interesting variant is to allow clustering: group the points into clusters and find a linear order of clusters that makes the paths behave nicely if you ignore what a path does inside a cluster. I believe that the unimodal algorithm extends to the case where every cluster has some bounded size k: basically, instead of eliminating a single point, we consider eliminating a set of k points. Correctness should carry through, and the algorithm is poly time if k is fixed.