Algorithm Development for Maximally - Stable Time - Dependent Graph Layout

Alexander V.Outkin

1. Introduction

When a time - dependent graph is displayed over time, it is advantageous to display it in such a fashion as to preserve the ability to understand both the graph and the changes to it across the entire timeline.This is a non trivial problem, because the graph may change discontinuously and unexpectedly.The graph layout algorithms often are arbitrary for the final layout orientation and that is one of the parameters we would like to preserve for a graph.We start with an outline of the essential aspects of the problem.

2. Define the "Maximally-Stable Time-Dependent Layout"

This could mean that parts of the graph that did not change remain in roughly the same place in the new layout.This could also mean that the new graph is easy to understand and relate to its previous instance, or that the entire sequence of graphs it is easy to understand.Graph layout can be formulated as optimization problem to attempt achieving one or all of these objectives.

3. Define the set of graphs, the changes possible in incremental graphs, and graph arrival rules

These can be informed by the domain the graphs represent.For example, the layout of the graph representing science papers where papers and citation links persist may be different from layout of the graphs representing changes in a social network of an individual where both the other people and the nature of connections can change suddenly.

4. Define the specific optimization problem

Possibilities include :

- On - line consecutive display of a set of graphs that arrive in a particular order, when the information at each time step is available about the current and all of the past graphs.
- Consecutive display of a set of graphs with the entire set available at the beginning of the layout.

5. Decide on what to display over time

There are two possibilities:

- New graph, and perhaps some of the old graphs.
- Changes to the graph in addition to 1. above.

6. Initial Approach

We start with two consecutive graphs.We will use the FructermanReingold algorithms to display them one after another, and we will display the difference between the graphs separately.We then will test the algorithm on a set of examples to evaluate if it works.

7. Conclusions

We will start with a simple example first of two graphs displayed consecutively, address the issues identified above for this simple example, and then extend the solution to a more general problem if time permits.