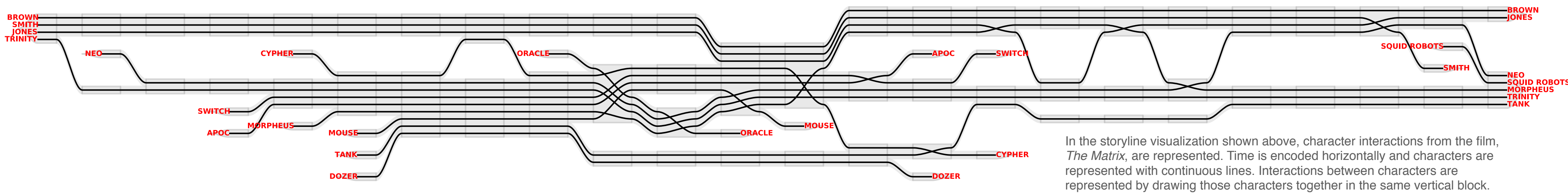
SVEN

(Storyline Visualization of Events on a Network)

An Alternative Storyline Framework for Dynamic Graph Visualization

Dustin Arendt
Pacific Northwest National Laboratory
dustin.arendt@pnnl.gov



In the storyline visualization shown above, character interactions from the film, *The Matrix*, are represented. Time is encoded horizontally and characters are represented with continuous lines. Interactions between characters are represented by drawing those characters together in the same vertical block.

1. Flow Graph

(minimize line crossings)

A directed acyclic graph is constructed whose nodes represent groups of storylines within each time window. Off the shelf digraph layering algorithms are effective at minimizing crossings in the flow graph.

2. Alignment Graph

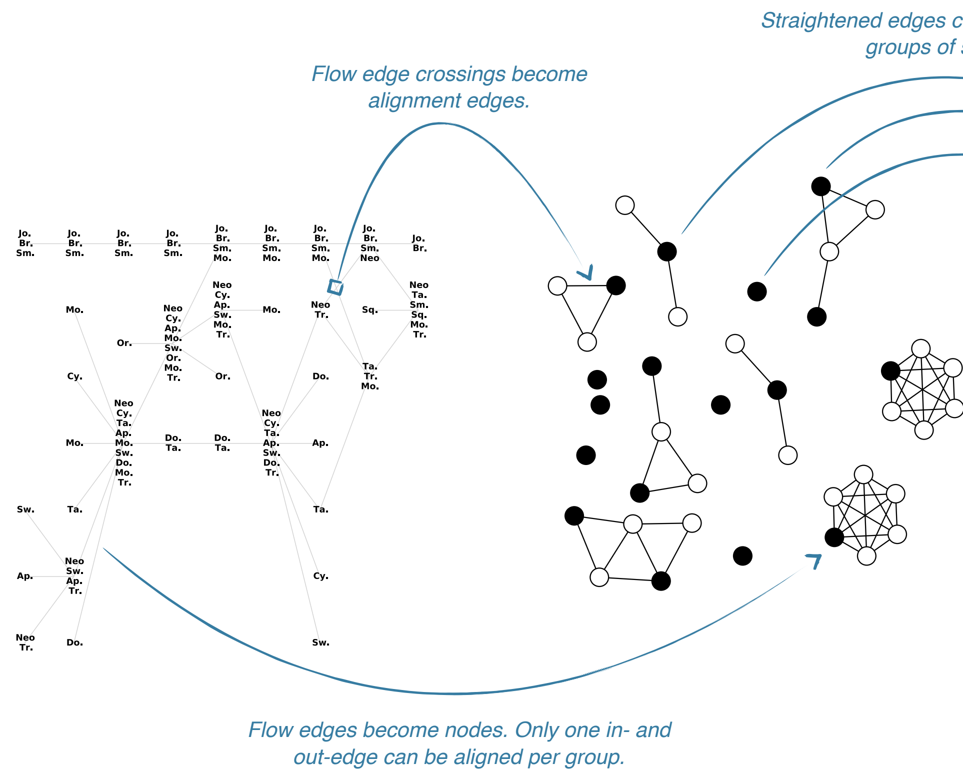
(maximize line straightness)

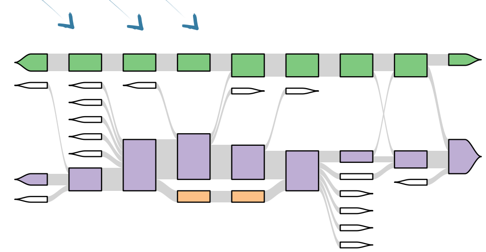
A graph is constructed to represent alignment constraints. Computing a maximum weight independent set of this graph finds a large set of flow edges to straighten.

3. Linear Program

(solve for whitespace)

The previous steps define a set of linear inequality constraints (ordering from the flow graph) and equality constraints (straightened lines from the alignment graph). Using a linear cost function that minimizes the separation between flow groups, a linear program will solve for the appropriate whitespace in the drawing.





Problem

Graphs that model real data will change over time. It is difficult to effectively communicate this change using traditional methods based on small multiples or animation.

Solution

Storylines can be used to communicate changing graph structure by drawing nodes as continuous lines (over time). Nodes/lines move together when interaction is occurring and move apart when the interaction ends. Rather than drawing individual links, the focus is on conveying changing communities, which are partitions of the network into relatively dense groups.

What's the input?

The algorithm assumes we have a dynamic partitioning of the entities we wish to draw—one partition for each time window. In this case study, we find this partition by applying a community detection algorithm independently to each graph for each time window. This changing partition could come from other sources (e.g., unsupervised clustering of n-dimensional samples over time).

What's the challenge?

Arbitrary placement of storylines produces drawings with degraded usability. Special care must be taken to account for important aesthetic criteria, which includes reducing line crossings and wiggles, and making effective use of whitespace. This is a combinatorial optimization problem.

What's next?

Being able to draw storylines effectively is an important first step. However, storyline visualizations, in general, suffer from scalability issues. Interactive visualization techniques including zooming, filters, and degree of interest are needed to effectively apply this technique to larger datasets.

Related Work

1. Beck, F., Burch, M., Diehl, S., & Weiskopf, D. (2014). The State of the Art in Visualizing Dynamic Graphs. EuroVis STAR. doi:10.2312/eurovisstar.20141174
2. Liu, S., Wu, Y., Wei, E., Liu, M., & Liu, Y. (2013). StoryFlow: Tracking the evolution of stories. IEEE Transactions on Visualization and Computer Graphics, 19(12), 2436–2445. doi:10.1109/TVCG.2013.196
3. Tanahashi, Y., Hsueh, C.-H., & Ma, K.-L. (2015). An Efficient Framework for Generating Storyline Visualizations from Streaming Data. IEEE Transactions on Visualization and Computer Graphics, 6(1), 1–1. doi:10.1109/TVCG.2015.2392771
4. Vehlow, C., Beck, F., Auwärter, P., & Weiskopf, D. (2015). Visualizing the evolution of communities in dynamic graphs. In Computer Graphics Forum (Vol. 34, pp. 277–288).