The ultimate goal when constructing a readable graph drawing is to avoid clutter that prevents viewers from grasping the structure of the graph. In practice, however, graphs are rarely planar, but can still have a planar substructure and thus be nearly planar. Can we draw graphs in a near-planar manner? Formal definitions of near-planarity lead to hard computational problems [5, 2, 1, 10].

Avoid clutter

We know humans perform better on user tasks in drawings with fewer crossings [13] and tend to prefer such drawings [12, 6].

Avoid crossings


Planarity is easy

Can we do better if we know the "cluttering edges"? YES – reduce the weights of the cluttering edges to 0.01 and rerun FA2.

What if we knew bad edges?

Which edges of a given nearly planar graph create clutter in the drawings generated by a spring-embedding algorithm?

Heuristic

Find the lengths of the vertex-disjoint paths between end-vertices of an edge $e = (u, v)$ using the max-flow Edmonds-Karp algorithm [4] as described in [9].

Construct a footprint $f(e) = \{t_0, t_2, \ldots \}$ for each edge $e$. Here $t_i$ – the number of vertex-disjoint paths of length $d$ between the end-point of $e$.

Standardize the footprints $f(e)$ to a user-specified number of dimensions $k$ and some function $M\cdot$, where $\text{min}/\text{mean}/\text{min}/\text{max}$ can be used.

Set the weight of each edge $e$ classified as cluttering to $\sharp f(e)$.

Original layouts

Results

Layouts produced by the heuristic

References