Beyond Planarity: A Spring-Based Approach

Simon van Wageningen · Tamara Mchedlidze · Alexandru Telea

avoid clutter

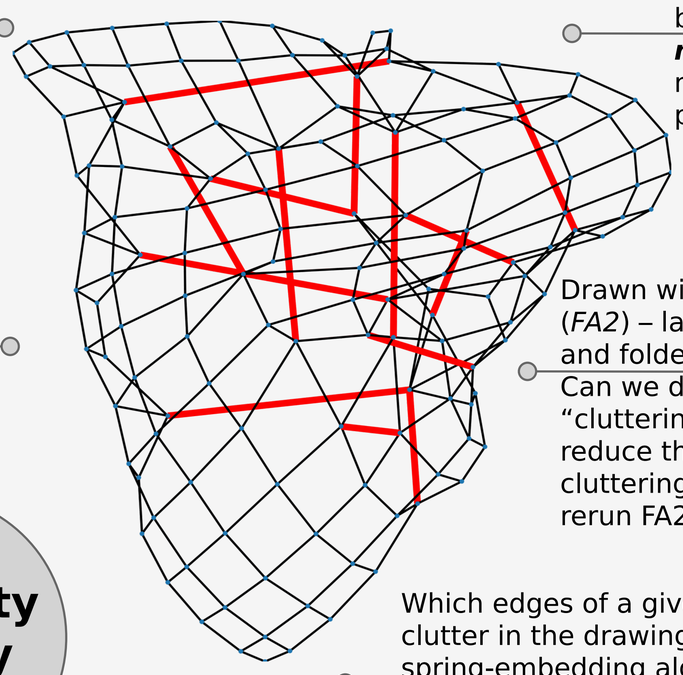
The ultimate goal when constructing a readable graph drawing is to avoid *clutter* that prevents viewers from grasping the structure of the graph

avoid crossings

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

planarity is easy

Testing whether a graph can be drawn without edge crossings [7], i.e. in a *planar* way, as well as constructing its planar drawing [14], can be done in linear time

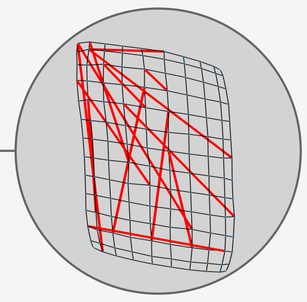


near-planarity is hard

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]

what if we knew bad edges?

Drawn with ForceAtlas2 [8, 3] (*FA2*) – layout appears cluttered and folded inwards.
Can we do better if we know the "cluttering edges"? YES - reduce the weights of the cluttering edges to 0.01 and rerun FA2.



research question

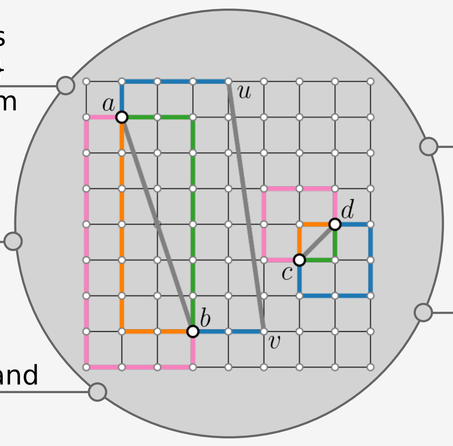
Which edges of a given nearly planar graph create clutter in the drawings generated by a spring-embedding algorithm?
How do we weight these edges to create a drawing where a planar substructure is clearly visible?

Heuristic

1a 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].

1b Construct a *footprint* $f(e) = [\ell_1, \ell_2, \dots]$ for each edge e. Here ℓ_i – the number of vertex-disjoint paths of length d between the end-point of e

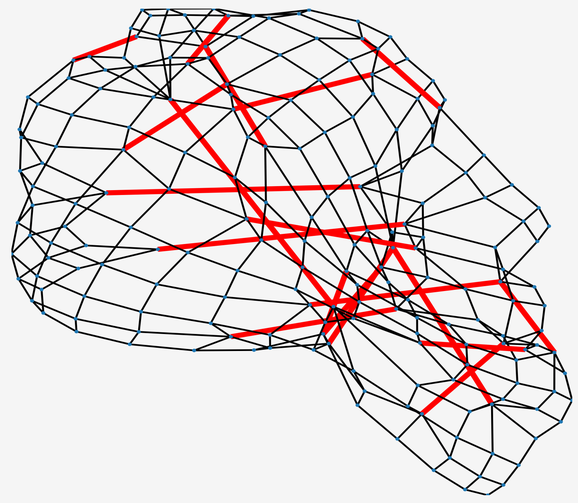
1c Standardize the footprints $f(e)$ to a user-specified number of dimensions k and some function $\mathcal{M}$, where $\mathcal{M}$ can be mean/minimum/maximum.

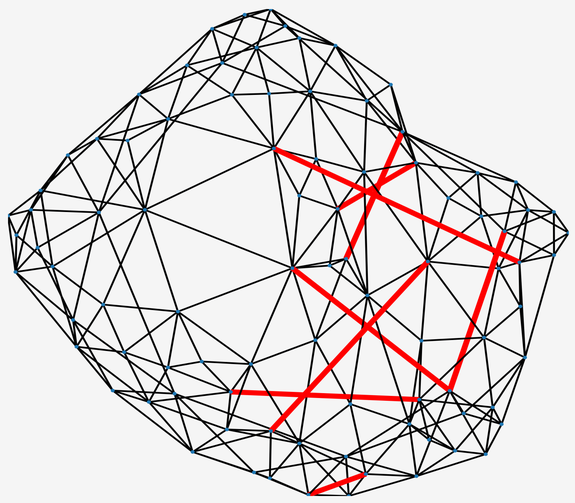


2 Using footprints, find cluttering edges using the *Isolation Forest* technique [11], which is designed to find anomalous data points in a high-dimensional space.

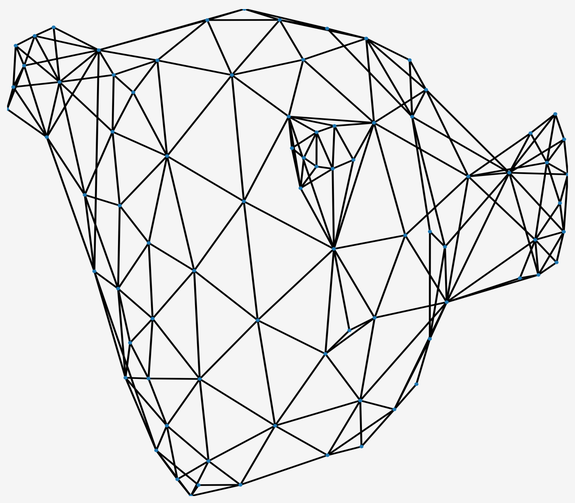
3 Set the weight of each edge e classified as cluttering to $\mathcal{M}(f(e))$

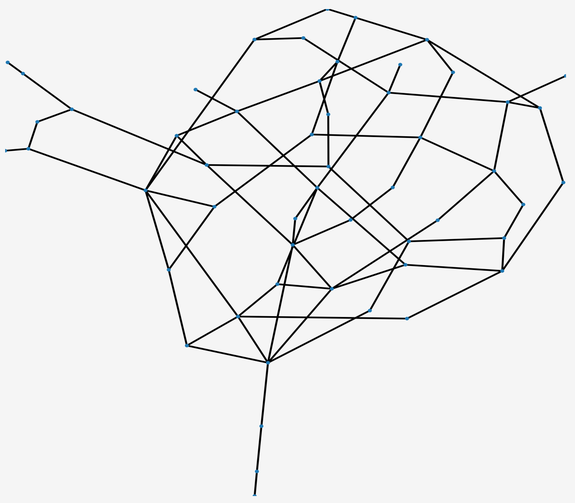
Original layouts



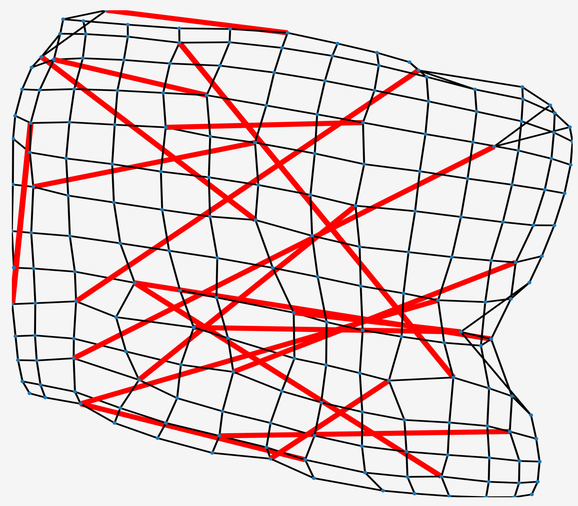


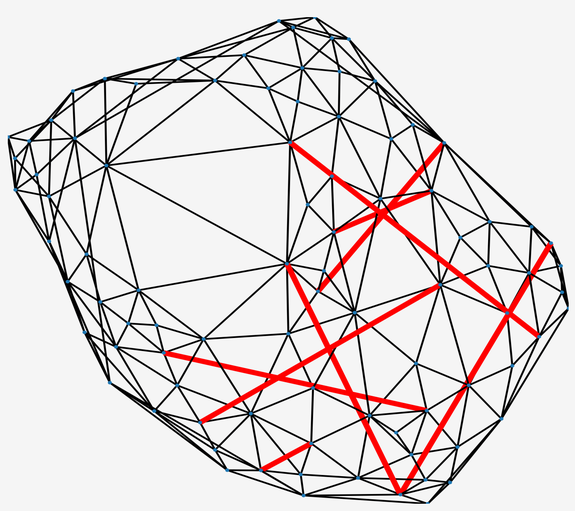
Results

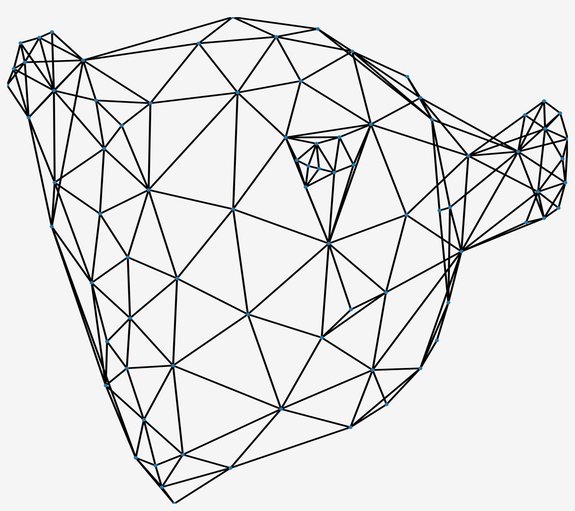


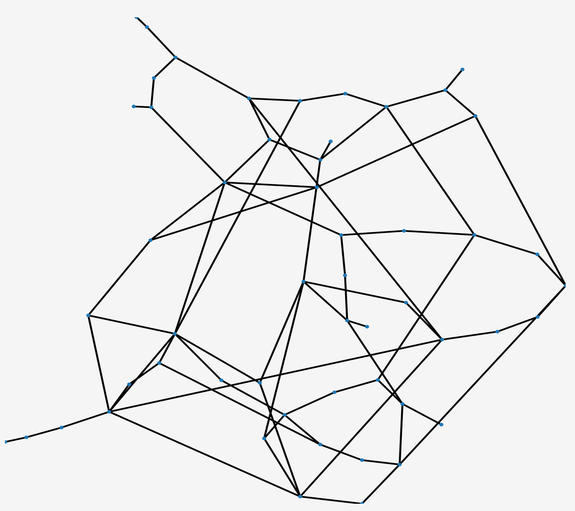


Layouts produced by the *heuristic*









References

[1] Argyriou, E., Bekos, M. & Symvonis, A. The straight line RAC drawing problem is NP-hard. *J. Graph Algorithms Appl.16(2): 569-597 (2012) [2] Cabello, S. & Mohar, B. Adding One Edge to Planar Graphs Makes Crossing Number and 1-Planarity Hard.* SIAM Journal On Computing. ***42**, 1803-1829 (2013) [3] Chippada, B. ForceAtlas2 Python. (2019), https://github.com/bhargavchippada/forceatlas2 [4] Edmonds, J. & Karp, R. Theoretical improvements in algorithmic efficiency for network flow problems.* Journal Of The ACM. ***19**, 248-264 (1972) [5] Garey, M. & Johnson, D. Crossing number is NP-complete.* SIAM Journal On Algebraic And Discrete Methods. ***4**, 312-316 (1983) [6] Ham, F. & Rogowitz, B. Perceptual Organization in User-Generated Graph Layouts.* IEEE Transactions On Visualization And Computer Graphics. ***14**, 1333-1339 (2008) [7] Hopcroft, J. & Tarjan, R. Efficient planarity testing.* Journal Of The Association For Computing Machinery. ***21**, 549-568 (1974) [8] Jacomy, M., Venturini, T., Heymann, S. & Bastian, M. ForceAtlas2, a Continuous Graph Layout Algorithm for Handy Network Visualization Designed for the Gephi Software.* PLOS ONE. ***9** (2014) [9] Kleinberg, J. & Tardos, E. Algorithm design. (Addison-Wesley,2006) [10] Korzhik, V. & Mohar, B. Minimal obstructions for 1-immersions and hardness of 1-planarity.* Journal Of Graph Theory. ***72**, 30-71 (2013) [11] Liu, F., Ting, K. & Zhou, Z. Isolation-Based Anomaly Detection.* ACM Trans. Knowl. Discov. Data. ***6** (2012) [12] McGrath, C., Blythe, J. & Krackhardt, D. Seeing Groups in Graph Layout.* Connections. ***19**, 22-29 (1996) [13] Purchase, H. Which aesthetic has the greatest effect on human understanding?* Graph Drawing. ***1353** pp. 248-261 (1997) [14] Tamassia, R. Planar straight line drawing algorithms.* Handbook On Graph Drawing And Visualization. *pp. 193-222 (2013)*

Utrecht University