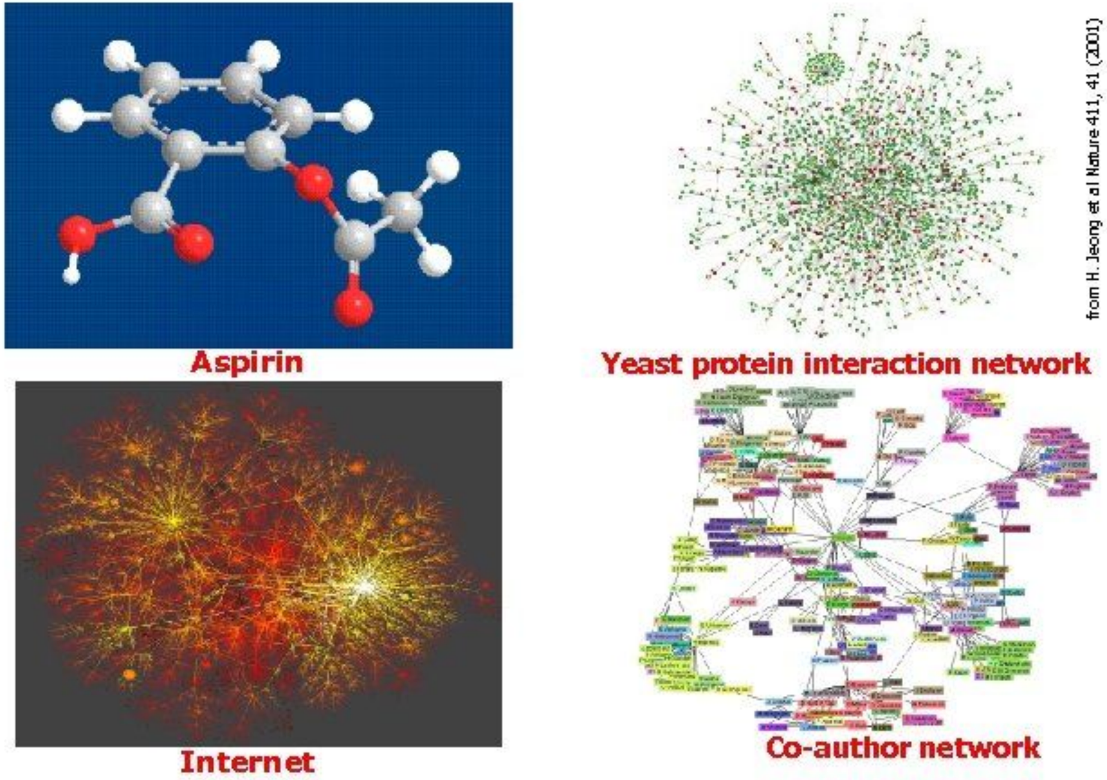
Spectral Graph Layout

by Peanut Butter

Elizaveta Lysova
Ekaterina Orlova
Mikhail Zybin

Skoltech
Skolkovo Institute of Science and Technology

Graph, Graph, Everywhere



There are two potential advantages of the spectral approach that we ruminate on.

- First, it provides us with a mathematically-sound formulation leading into an exact solution to the layout problem, whereas, almost all other formulations result in an NP-hard problem, which can only be approximated.
- The second advantage is computation speed.

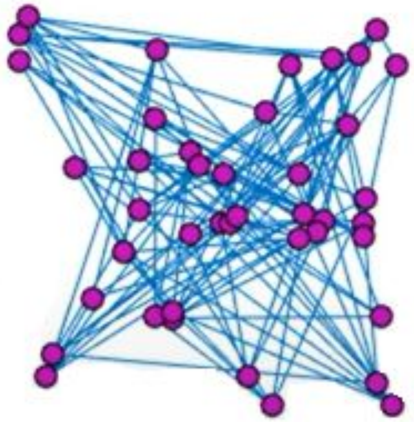
Laplacian

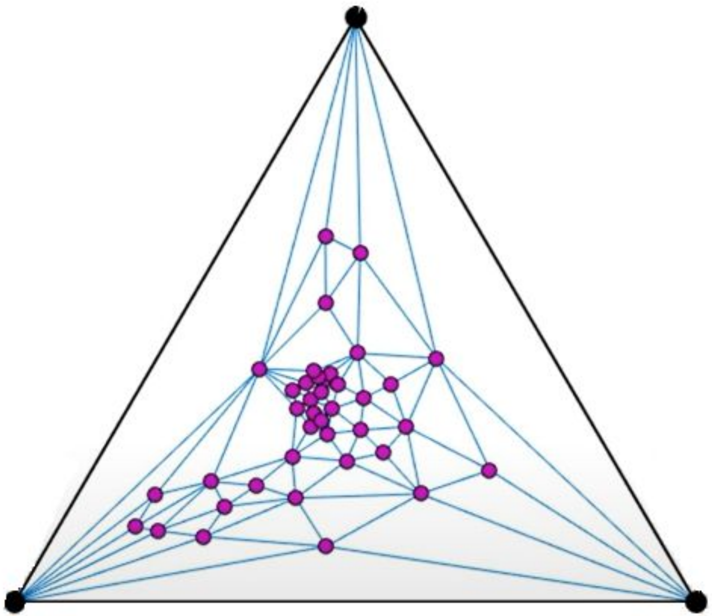
D - diagonal matrix with degrees of vertices on the diagonal

A - adjacency matrix

$L = D - A$, Laplacian, symmetric positive definite matrix

Example of spectral drawing





Skoltech
Skolkovo Institute of Science and Technology

Algorithms

Hall's algorithm

To use the two smallest eigenvectors of the Laplacian for layout

Koren's algorithm

To use the p lowest degree-normalized eigenvectors

or the p leading nondegenerate eigenvectors of $D^{-1} A$

Tutte's algorithm

Find a cycle in the graph. Then, for each vertex outside the cycle, look for neighbours (adjacent vertices) and move this vertex to their centroid.

Dataset of graphs

- SuiteSparse Matrix Collection
- DIMACS10 group
- 34 graphs that have been very popular as benchmarks for graph algorithms

Skoltech
Skolkovo Institute of Science and Technology

Shape-based quality metric

In order to estimate the quality of the layout of graph G1, we remove the edges from the graph and perform the Delaunay triangulation and get the graph G2. The more similar the original graph to G2, the better the layout. We have tried to build G2 as Euclidean Minimal Spanning Tree, but this approach was too time-costly.

Shape-based quality metric

the similarity between two graphs that have the same sets of vertices is calculated as follows

Suppose that $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ are two graphs with the same vertex set. A simple measure for the similarity of G_1 and G_2 is the *mean Jaccard similarity*:

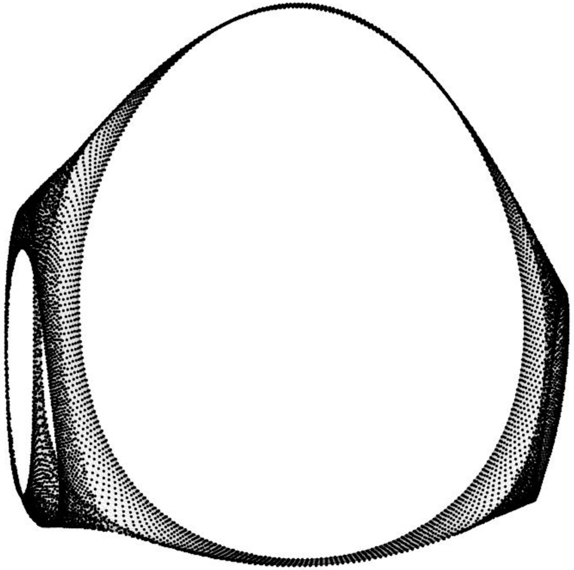
$$MJS(G_1, G_2) = \frac{1}{|V|} \sum_{u \in V} \frac{|N_1(u) \cap N_2(u)|}{|N_1(u) \cup N_2(u)|}, \tag{2}$$

where $N_i(u)$ is the set of neighbours of u in G_i for $i = 1, 2$. It is straight-forward to compute the mean Jaccard similarity in linear time.

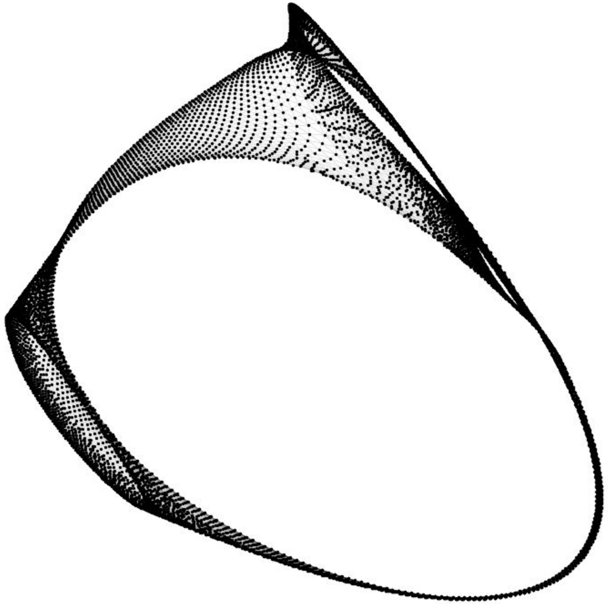
Note that $0 \leq MJS(G_1, G_2) \leq 1$. Also, if G_1 and G_2 share many edges, then $MJS(G_1, G_2)$ is close to 1; if they share very few edges then $MJS(G_1, G_2)$ is close to 0.

Graph	Method	Metric
Graph_4elt	Hall’s algorithm	0,196
	Koren’s algorithm	0.304
	Tutte method	0,024
Graph_144	Hall’s algorithm	0,048
	Koren’s algorithm	0.049
	Tutte method	0,003
Graph_data	Hall’s algorithm	0.147
	Koren’s algorithm	0.159
	Tutte method	0.108

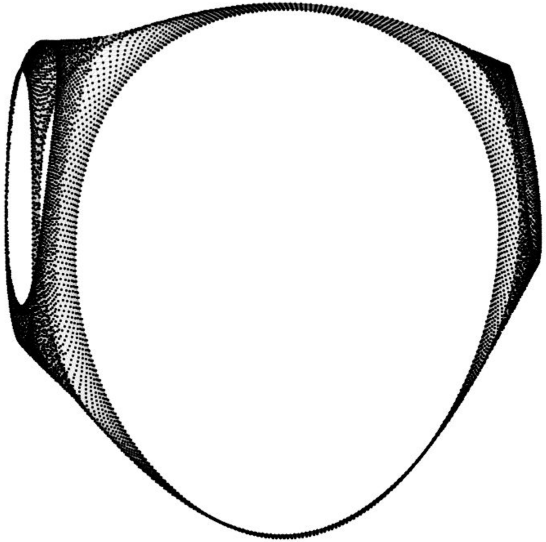
Graph_4elt



Hall's algorithm



Koren's algorithm

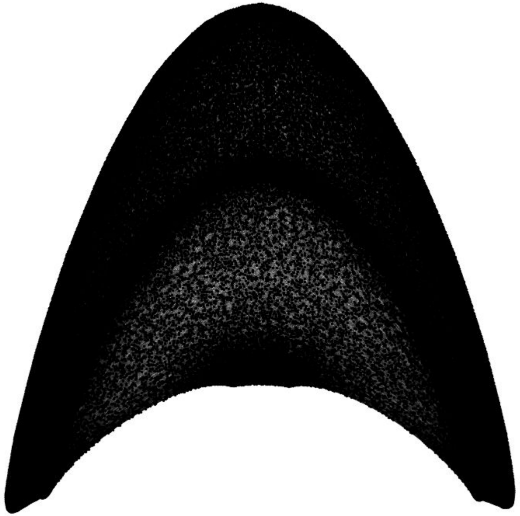


Tutte's method

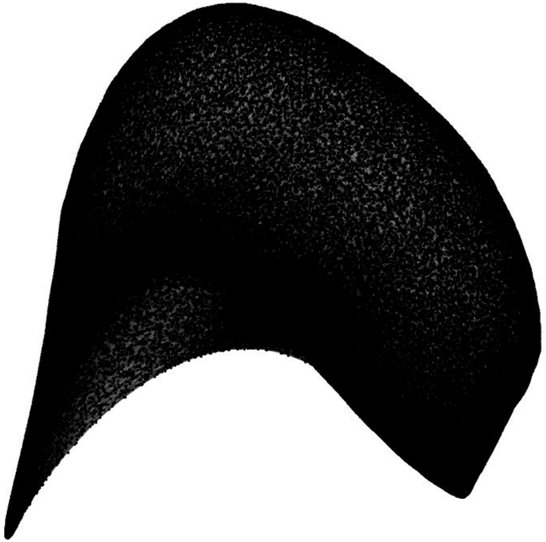
Skoltech

Skolkovo Institute of Science and Technology

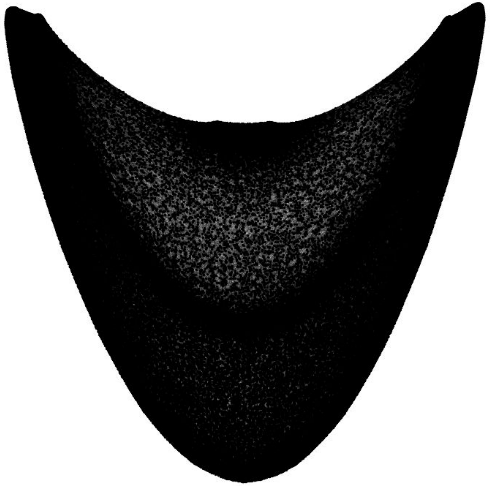
Graph_144



Hall's algorithm

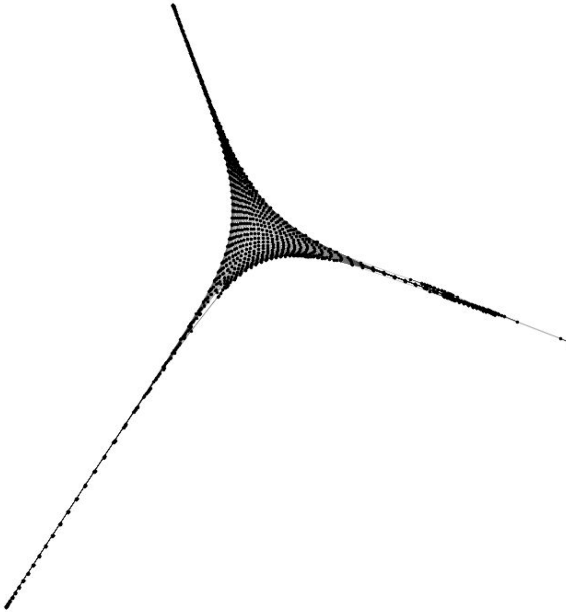


Koren's algorithm

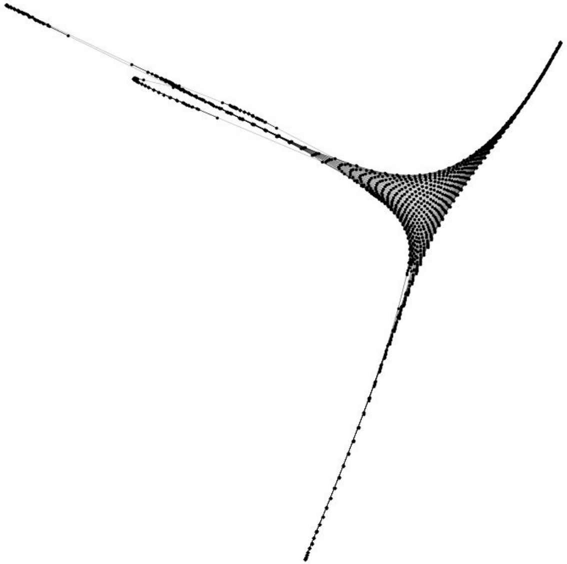


Tutte's method

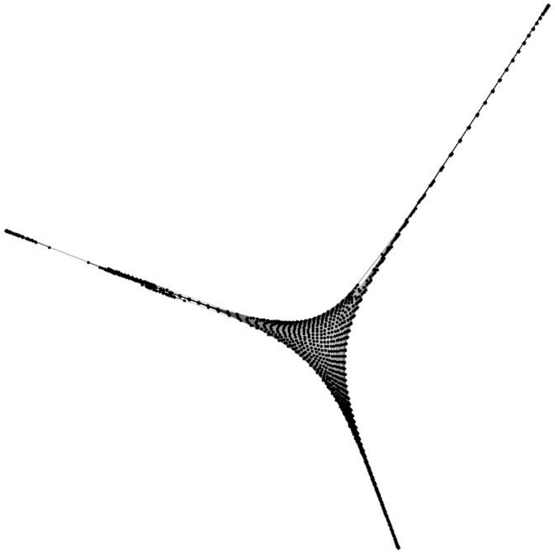
Graph_data



Hall’s algorithm



Koren’s algorithm

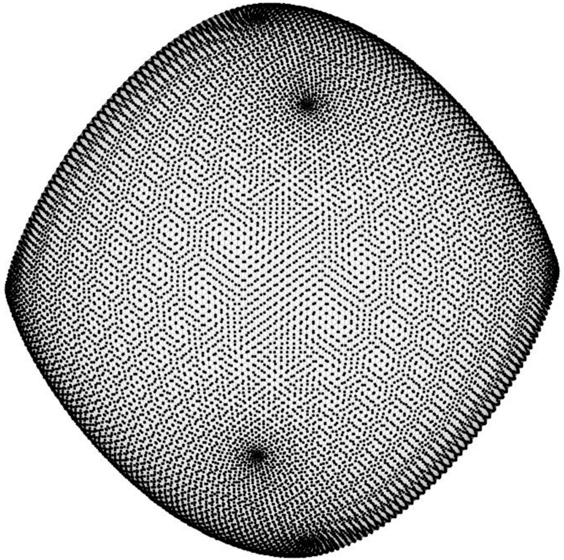


Tutte’s method

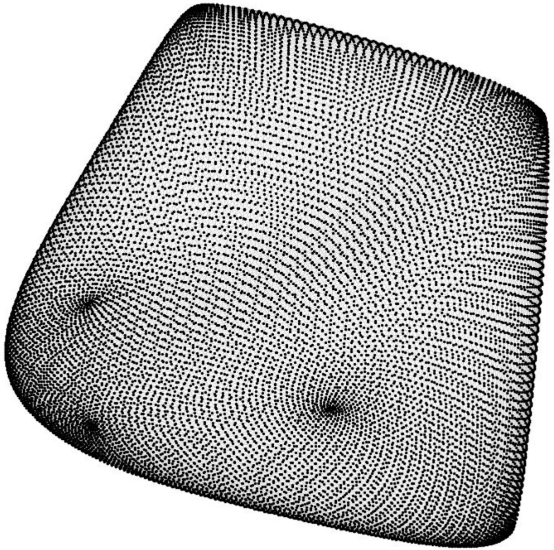
Skoltech

Skolkovo Institute of Science and Technology

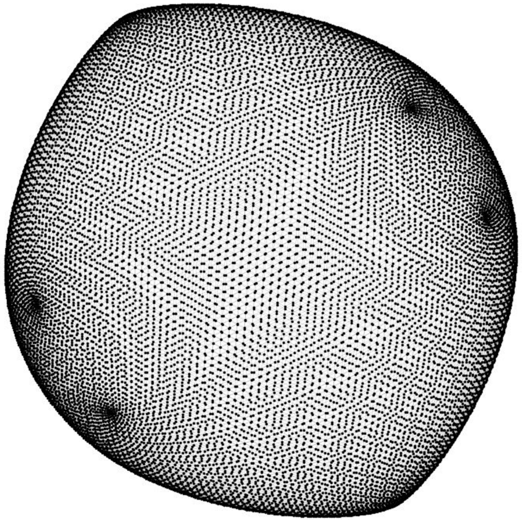
Graph_fe_sphere



Hall's algorithm



Koren's algorithm



Tutte's method

Skoltech

Skolkovo Institute of Science and Technology

Our contributions:

- New implementations of the Koren's, Hall's and Tutte algorithms
- Implementation of Jaccard similarity metric
- Evaluating these algorithms on a collection of graphs

Skoltech
Skolkovo Institute of Science and Technology