INF580 report
Tutte planar graph embedding and distance geometry problem

Nikita Dudorov, Ecole Polytechnique

March 2023

1 Problem

In this project we would like to implement the Tutte planar graph embedding algorithm [1], draw with it some planar graphs in 2D, and than use these drawings as instances of the distance geometry problem: try to reconstruct the drawings by pair-wise distances between vertices using mathematical optimization.

2 Overview

Tutte proved the following result: any 3-connected planar graph admits a planar (edge intersection free) drawing, in which each face of the graph is convex, and each vertex is the barycenter of its neighbors (except for the vertices on the outer face). The problem of finding such a drawing can be formulated as a energy minimization problem if we imagine that all the vertices of the graph are attached with springs:

$$\sum_{(u,v)\in E} (x_u - x_v)^2 \to \min_x$$

This is equivalent to

$$x^T L x \to \min_x$$

where L is the Laplacian matrix of the graph, defined as follows (A is the adjacency matrix of the graph):

$$Diag(\deg(v)) - A$$

Finally to find the drawing we could solve the sparse linear system:

$$L'x = b$$

Tutte showed that for the solution to be unique and different from 0 we should fix the coordinates x_v^* of any face f of the graph on a convex polygon and then modify the system according to these constraints. So:

$$L'[v] = L[v] \ \forall v \notin f$$

$$L'[v] = e_v \ \forall v \in f$$

$$b[v] = 0 \ \forall v \notin f$$

$$b[v] = x_v^* \ \forall v \in f$$

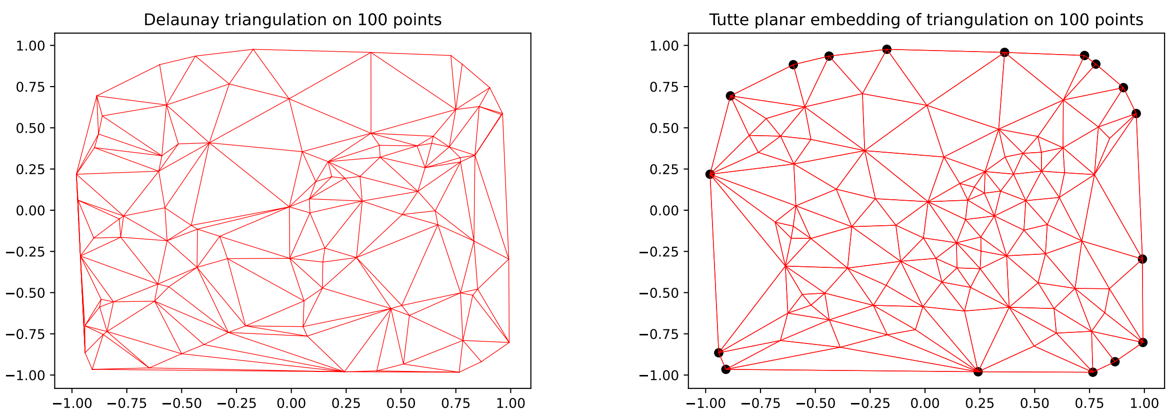
Unfortunately after these modifications the matrix L is not symmetric anymore.

To reconstruct the drawings produced by Tutte algorithm using just pair-wise distances d_{uv} between vertices we solve the distance geometry problem:

$$\sum_{(u,v)\in E} |d_{uv} - \|x_u - x_v\|| \to \min_x$$

3 Experiments

A planar triangulation is a planar graph in which each face (including the outer) is a triangle. It is a known fact that any planar graph can be represented as as sub-graph of planar triangulation. For that reason we will draw only triangulated planar graphs. To generate such a graph we will sample n random points in $[-1, 1] \times [-1, 1]$ square, then find Delaunay triangulation of these points and use this triangulation as our planar graph instance. Than we re-embed this planar graph with Tutte algorithm and use this embedding as distance geometry problem instance. Look at examples in figure 2.



(a)

(b)

Figure 1: Delaunay triangulation and its Tutte embedding

To solve the distance geometry problem we will try to approximate the gram matrix of vertex coordinates:

$$G = XX^T$$

Knowing the gram matrix we then will be able to restore coordinates using principal component analysis:

$$G = P^T \Lambda P = (\sqrt{\Lambda} P)^T (\sqrt{\Lambda} P)$$
$$\Rightarrow X = P^T \sqrt{\Lambda}$$

by keeping in eigenvalue matrix Λ just 2 largest positive components (as we already know that the solution of our problem is 2-dimensional)

To approximate the gram matrix we will use two approaches:

1. Complete pair-wise distance matrix D of the graph with shortest paths using Floyd-Warshall algorithm, then find gram matrix by the following formula:

$$G = -\frac{1}{2} JDJ$$

where:

$$J = I - \frac{1}{n} \mathbf{1}\mathbf{1}^T$$

(theoretical expression for gram matrix from pair-wise distances).

2. Approximate gram matrix G by solving SDP relaxation of distance geometry problem. We used the following variant of it:

$$\sum_{(u,v)\in E} G_{uu} + G_{vv} - 2G_{uv} + 0.01 \cdot tr(G)$$

$$G_{uu} + G_{vv} - 2G_{uv} \geqslant D_{uv} \ \forall (u,v) \in E$$

$$G \succeq 0$$

At the end we will refine each solution by solving the following NLP problem with starting point:

$$\sum_{(u,v)\in E} \left(d_{uv}^2 - \|x_u - x_v\|^2\right)^2 \to \min_x$$

To do that we used AMPL solver "conopt".

In the following part we use the notation: PCA - Principal Component Analysis (extraction from gram matrix of two most "informative" components); FW - Floyd-Warshall algorithm (completing pair-wise distance matrix of the graph with shortest paths); SDP - described above relaxation of distance geometry problem, local - refining solution of distance geometry problem with a local NLP solver.

LDE loss:

$$LDE(X, D) = \max_{(u,v)\in E} |D_{uv} - \|x_u - x_v\||$$

MDE loss:

$$MDE(X, D) = \frac{1}{|E|} \sum_{(u,v)\in E} |D_{uv} - \|x_u - x_v\||$$

Here are our results (figures 2, 5, 4):

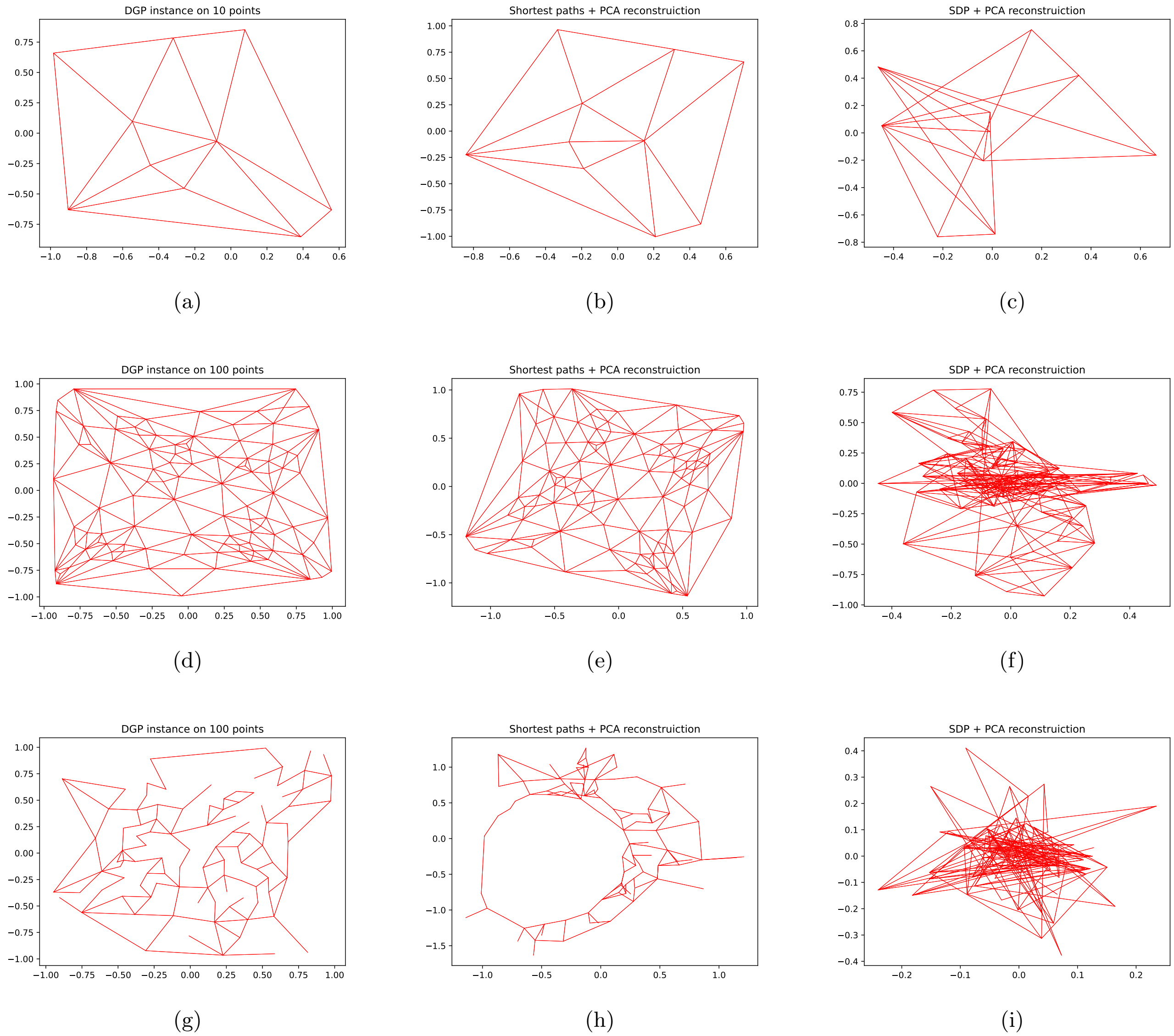
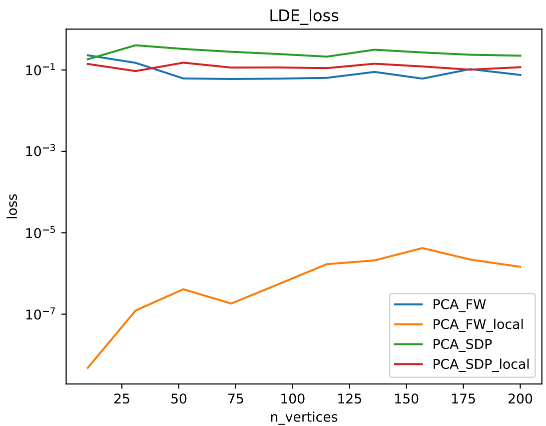
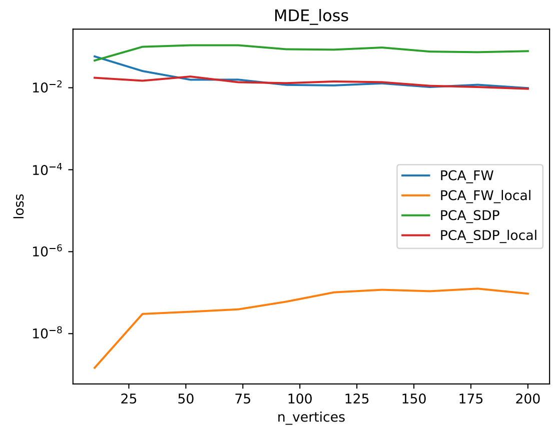


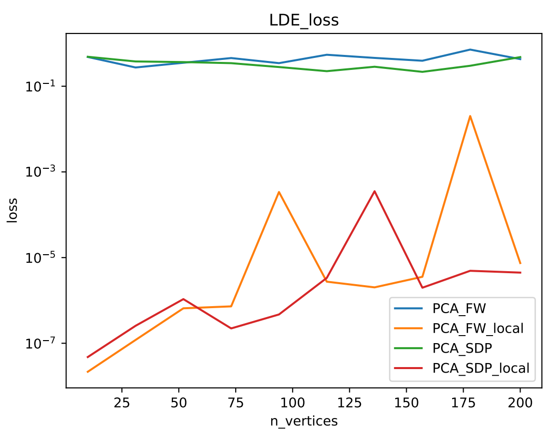
Figure 2: Examples of drawing reconstruction



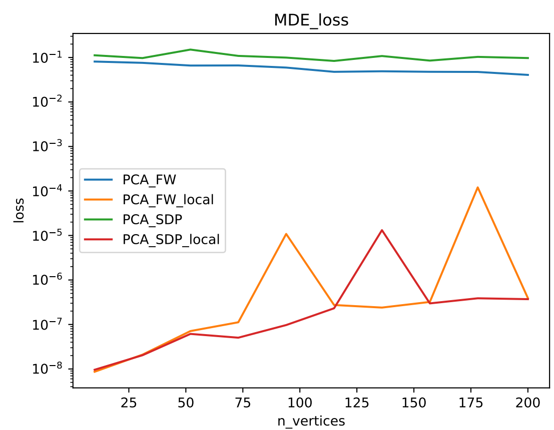
(a) edge density = 3



(b) edge density = 3

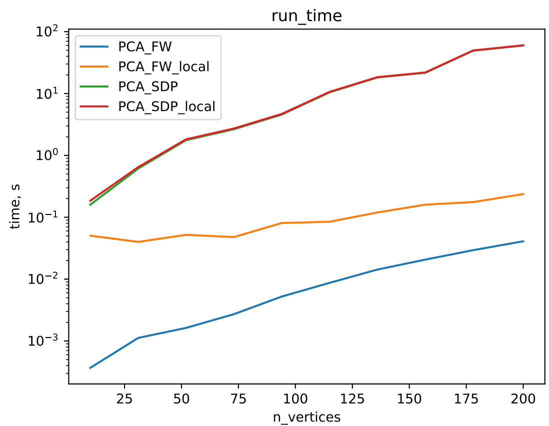


(c) edge density = 1.5

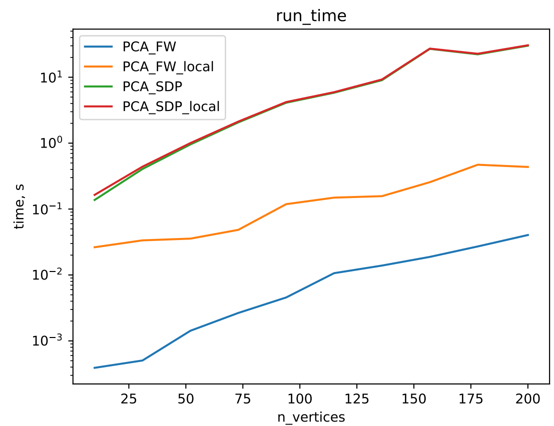


(d) edge density = 1.5

Figure 3: Mean and largest distance loss for drawing reconstruction



(a) edge density = 3



(b) edge density = 1.5

Figure 4: Run time for drawing reconstruction

4 Conclusions

- By observing the running time we see that the approach using minimal distances calculation with Floyd-Warshall algorithm is much faster then solving SDP formulation (it is expectable as the latter is a complex optimization problem with n^2 variables).

  For the PCA+FW approach the time of refining local minimum is very costly with respect to the time needed to find the initial approximation, whereas in the PCA+SDP approach the time needed to refine the solution could be neglected.

  Also we note that in our experiment we didn't managed to go further than graphs on $n = 200$ vertices, as for the SDP formulation taking $n = 500$ will already mean solving a problem with $\sim 10^5$ variables.

- By observing the loss we can say that refining the solution with local NLP solver considerably improves the loss for both PCA+FW and PCA+SDP approaches (but it doesn't seem to improve visual quality of the graph drawing)

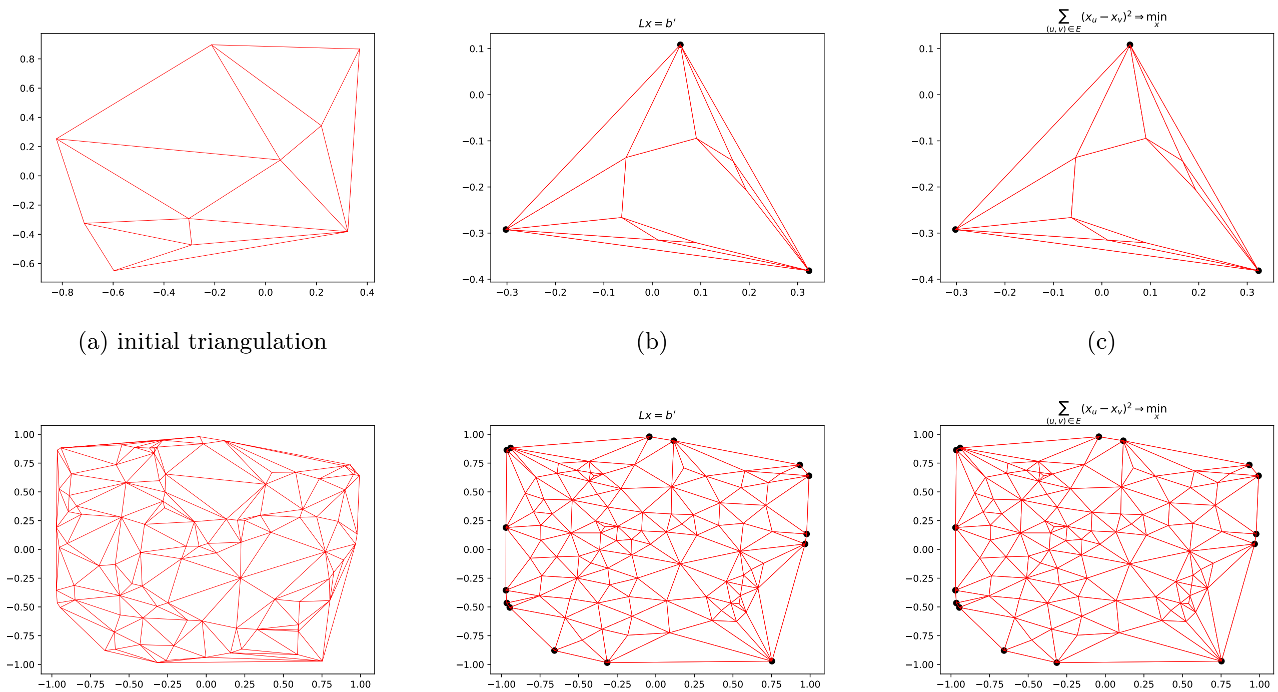
- By looking at reconstructed drawings and loss for graphs with different edge density we observe that when graph is getting further from a triangulation the shortest paths calculation is not that precise anymore to approximate pair-wise distances, and so the drawings produced by PCA+FW approach gets corrupted and PCA+SDP approach catch up PCA+FW approach in terms of loss.

- Generally drawings reconstructed with PCA+SDP approach are worse than that produced by PCA+FW approach. By the way, when edge density is low enough, PCA+FW approach's drawing performance is as bad as that of the PCA+SDP approach, and PCA+SDP is even better in the sense that when the graph gets disconnected we can't anymore use PCA+FW but PCA+SDP will still be a valid approach.

References

[1] W. T. Tutte. "How to Draw a Graph". en. In: *Proceedings of the London Mathematical Society* s3-13.1 (1963), pp. 743–767. DOI: `10.1112/plms/s3-13.1.743`. URL: `http://dx.doi.org/10.1112/plms/s3-13.1.743`.

5 Annexe



(a) initial triangulation

(b)

(c)

(d) initial triangulation

(e)

(f)

Figure 5: Comparison of different Tutte embedding formulations. In black are fixed constraint vertices.