CS 292F: Spectra of Geometric Graphs

Nathan Wachholz

June 9, 2022

We will investigate geometric graphs in the context of Spectral Graph Theory. We look for interesting properties of this class of graph, and investigate the viability of spectral graph partitioning as a strategy for clustering points in geometric spaces.

A geometric graph is a weighted graph whose vertices can be embedded in a low-dimensional metric space, such that edge weights between vertices correspond to their distance in the metric space. Any weighted graph can be embedded in a sufficiently high-dimensional metric space, so we will focus on graphs embeddable in $\mathbb{R}^2$, under Euclidean distance.

We begin by experimentally looking for interesting properties in the spectra of geometric graphs. Several classes (random, delaunay triangulation, spanning tree, minimum spanning tree) of geometric graphs are considered and compared to non-geometric graphs. Their spectra are examined as parameters (dimension, node count, edge count, etc.) are varied. A few interesting properties are noted, but unfortunately, few are related to the graphs being geometric, and none are novel. Relevant papers are cited as we go.

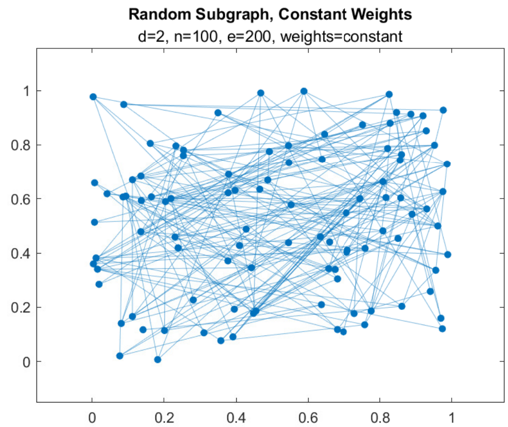
Then we briefly consider if the spectral layout could be used to reconstruct the positions of the vertices, given only the edge distances. The answer seems to be no, spectra are unrelated to the original geometry.

Lastly, we experiment with spectral graph partitioning on these geometric graphs. Unfortunately, this relies on spectra, which is seemingly unrelated to the original geometry. We conclude that spectral analysis is not useful for geometric clustering, nor any geometric problem.

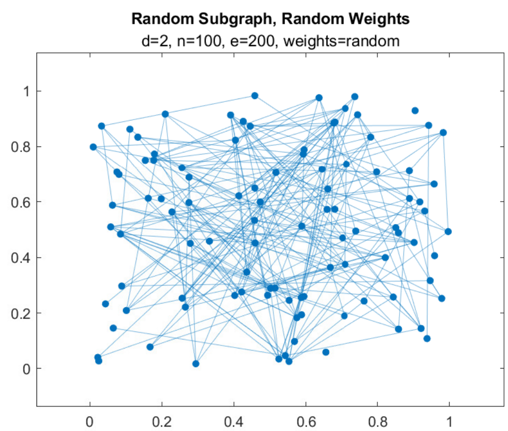
1 Properties of Geometric Graphs

1.1 Graphs Considered

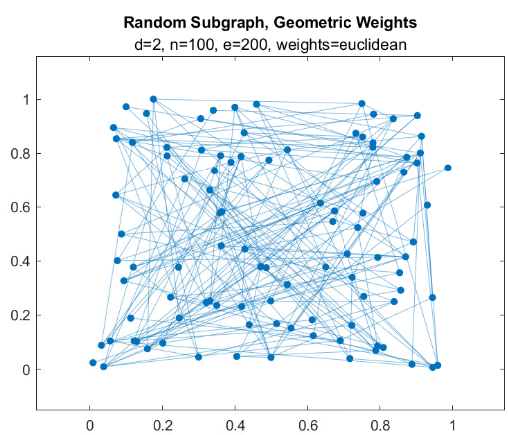
Graphs will be generally defined on a set of n points in $\mathbb{R}^2$, typically bounded to $[0, 1]^2$. We are sure to include some non-geometric graphs, to make sure any patterns we discover are due to them being geometric. The following graphs will be considered:



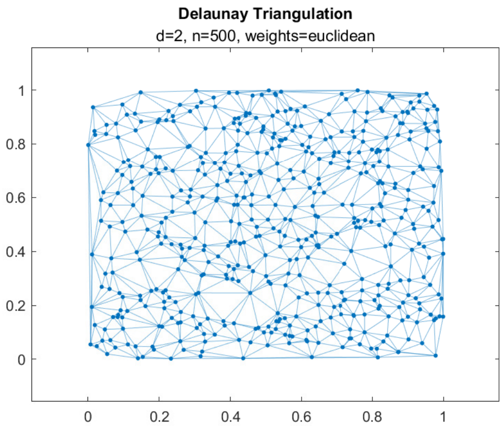
A random subgraph of the complete graph. Edge weights are *constant*. This serves as a control to see if any patterns are unique to geometric graphs.



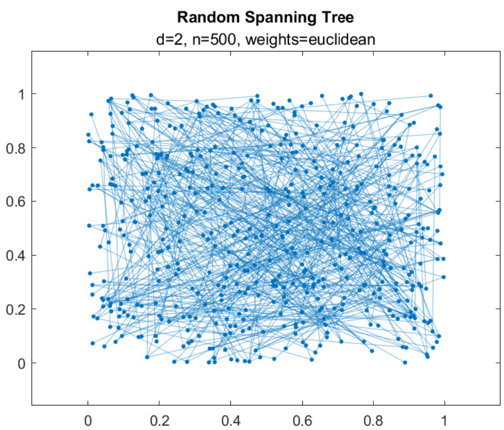
A random subgraph of the complete graph. Edge weights are *random*. This serves as a control to see if any patterns are unique to geometric graphs.



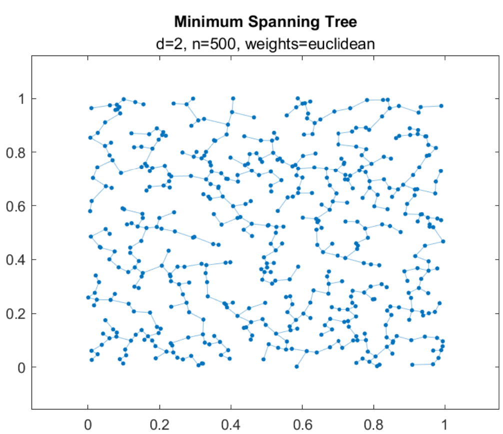
A random subgraph of the complete graph. Edge weights are given by the metric distance between connected vertices.



The edges forming the Delaunay triangulation of the points. Edge weights are given by the metric distance between connected vertices.



A random spanning tree of the complete graph. Edge weights are given by the metric distance between connected vertices.



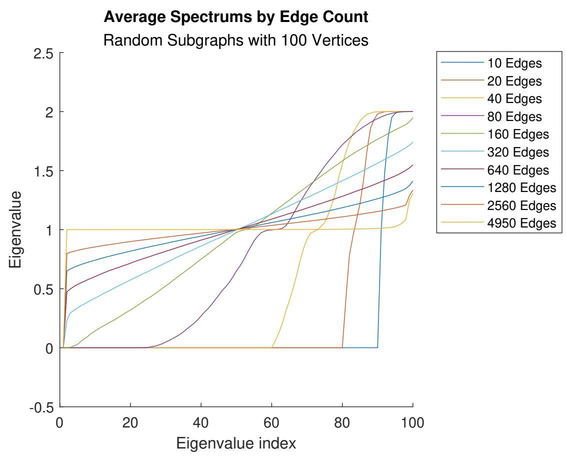
The minimum spanning tree of the complete graph. Edge weights are given by the metric distance between connected vertices.

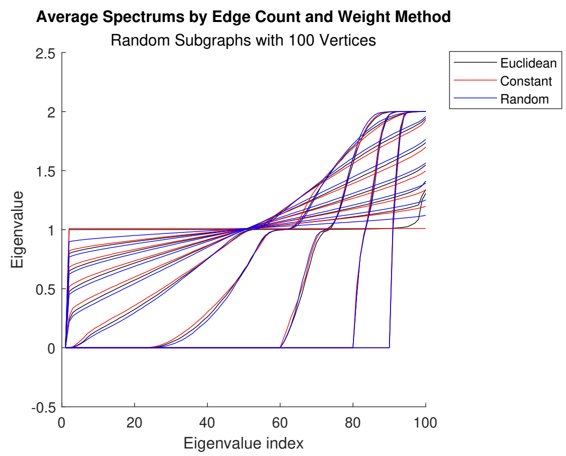
1.2 Graph Spectra

Now that we have some graph classes defined, we will generate a large number of each of them, and analyze their Normalized Weighted Laplacians.

1.2.1 Random Subgraphs

All random subgraphs had similar spectra, regardless of their weighting method (random, constant, or metric). The main factor was the relative number of edges in the subgraph, although the weighing method had a minor (consistent) effect:



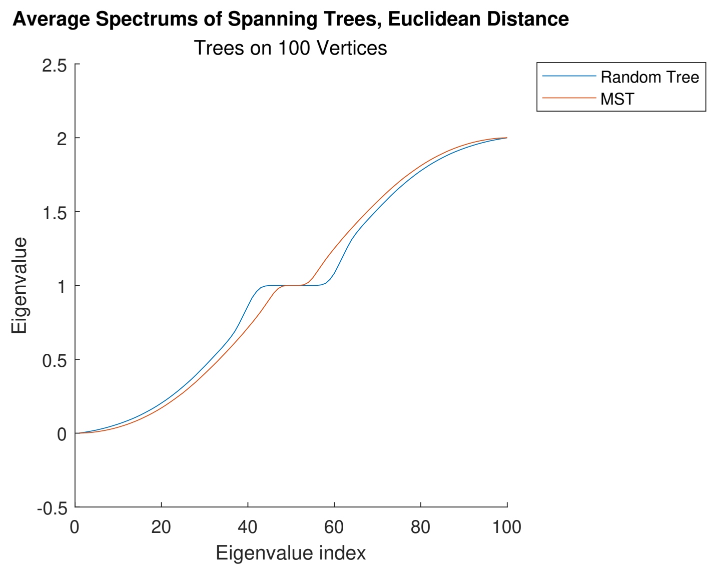


First consider the greatest eigenvalues in each spectrum. Some rounding errors are evident, particularly in the complete graph. But notice also that the eigenvalues are bounded by 2. This is a known property of the normalized weighted (or unweighted) Laplacian [1]. In fact, $\lambda_{n-1} = 2$ if and only if the graph is bipartite[1], which explains why this bound is only reached for the lower edge counts.

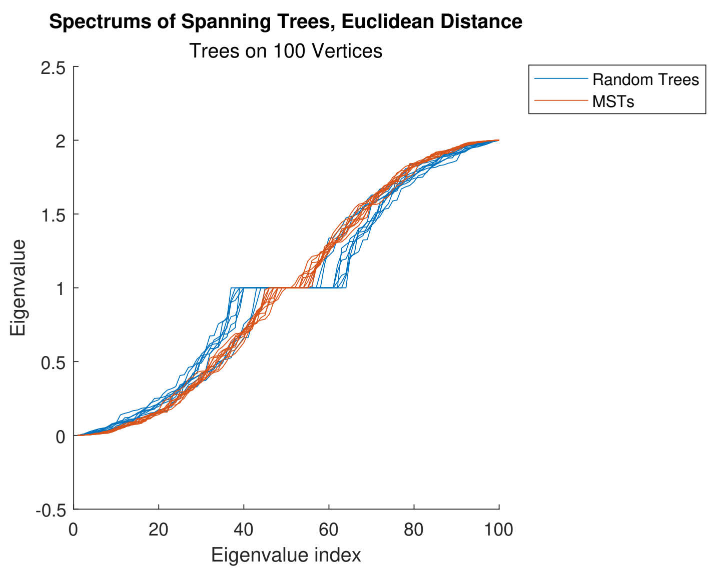
The change due to weighting method is interesting. Because we are looking at a Normalized Laplacian, the magnitude of the weights doesn't affect the spectrum. The change must stem from the change in weight *variation*. Sure enough, changing the dimension of the Euclidean method affects the spectrum. If $d = 1$ (so our points are in $\mathbb{R}^1$, and weights follow a uniform distribution), the curves line up with the random (uniform) method. If $d > 5$ or so, we see the curves line up with the constant method. This makes some intuitive sense: the Curse of Dimensionality[2] tells us that the distance between random points increases in higher dimensions, and (importantly) the variation decreases (more constant-like).

1.2.2 Spanning Trees

Now we consider random spanning trees and minimum spanning trees. These have spectra very similar to random subgraphs with the same number of edges (see 80 Edges above):



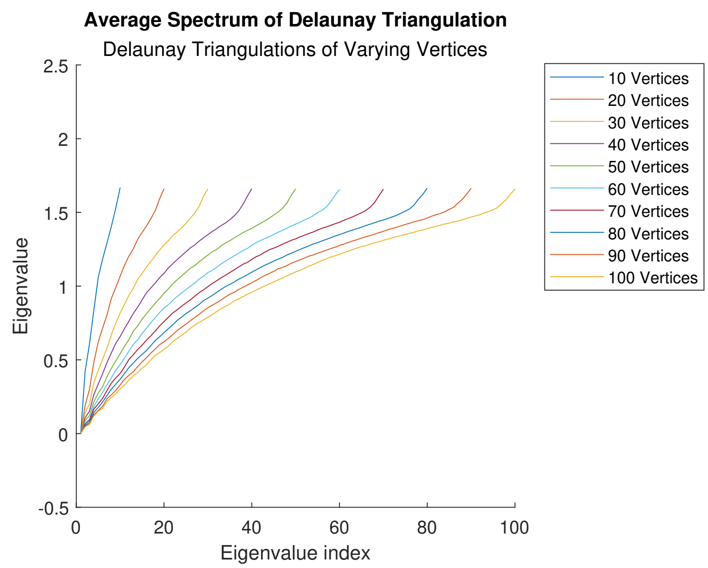
But notice an interesting difference in the multiplicity of $\lambda = 1$. Let's examine the spectra of individual graphs (not averages):



It turns out that the multiplicity of $\lambda = 1$ relates to the minimum vertex cover of the graph[3]. Specifically, "its multiplicity is related to the size of a minimum vertex cover, and zero entries of its eigenvectors correspond to vertices in minimum vertex covers"[3]. Looking at these graphs, we can guess that the minimum vertex cover of an MST is usually smaller than that of a random spanning tree. We can also notice that the spectra are symmetric about $\lambda = 1$. This is also a known property[1].

1.2.3 Delaunay Triangulation

Lastly, we consider Delaunay Triangulations. These are the most likely candidates for interesting properties. A lot of geometry seems to be preserved. Nodes have similar degrees, and edges have similar lengths. Here is their average spectrum, as the number of vertices are varied:



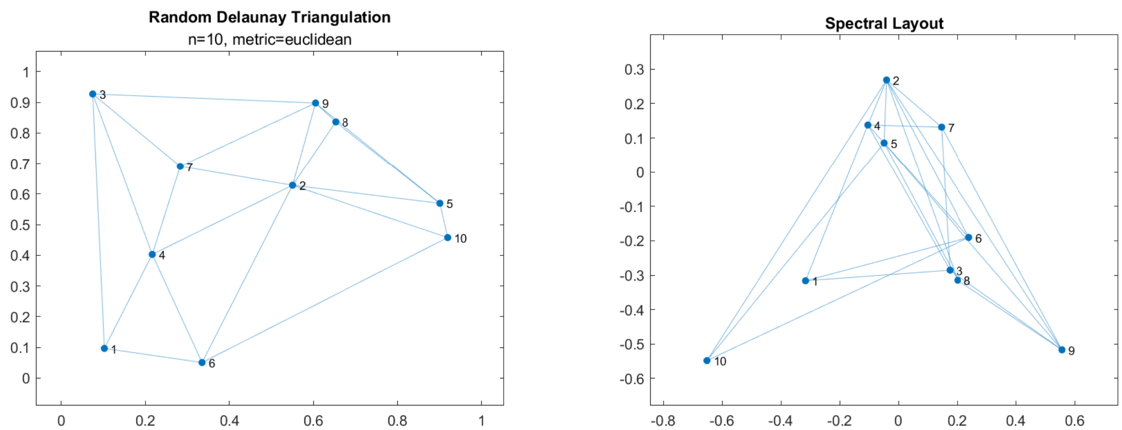
It has the same basic curve, regardless of the number of vertices. And, this curve is unlike those of any of the random subgraphs we saw earlier. We can also notice that the maximum eigenvalue is consistent, at around 1.65. This bound can be explained to some degree by the fact that this graph is planar, and a triangulation. The papers [4] and [5] give bounds for the largest eigenvalue of triangulations. These bounds involve the minimum and maximum degrees of vertices, and the number of edges. For random Delaunay Triangulations, these values are fairly consistent[6].

1.3 Summary

Although we did not find any properties unique to geometric graphs, our experimentation led us to corroborate several known properties of Normalized Laplacians in general. And we discovered that the Delaunay Triangulation seems to be our best bet for preserving geometric data, as its spectrum is unlike any of the non-geometric control graphs.

2 Spectral Layout

We briefly consider if the spectral layout of a geometric graph relates to the original geometry. Given only a set of edge weights, we hope to recover the original positions of vertices. Let's generate a random Delaunay Triangulation, and then plot its vertices according to the first and second smallest non-trivial eigenvectors of its normalized weighted Laplacian:



There is no correlation. Nor is there a correlation when using the largest eigenvalues, or when using the (non-normalized) weighted Laplacian doesn't change this either. Even when G is a complete geometric graph (all distances preserved), the vertex positions in the spectral layout are seemingly random.

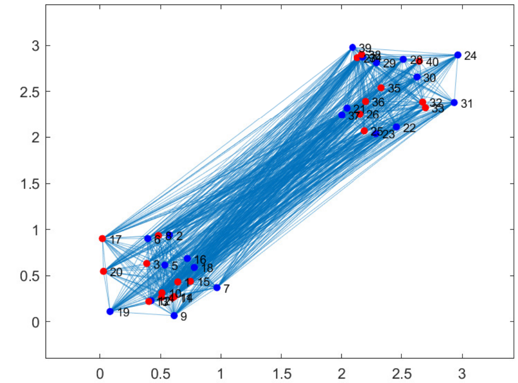
We are unable to find a (simple) way to restore the vertex positions using spectral layouts.

3 Clustering

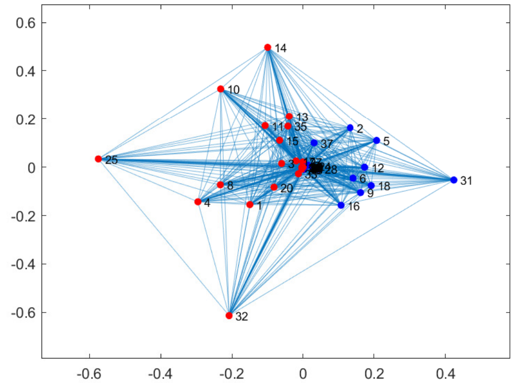
Given a set of n points in euclidean space, the k-clustering problem asks us to cluster these points into k groups, minimizing some property (such as max distance between any two points in a group). This optimization problem is NP-Hard in general[7], and we instead rely on approximation algorithms.

We finish by considering spectral graph partitioning as a method of approximating the k-clustering problem. Based on the findings in Section 2, it seems like the eigenvectors do not contain any geometric information. This makes a spectral approach to clustering seem unlikely to work. Nevertheless, we generate a few graphs, and attempt to cluster them using spectral partitioning. We ensure clear clusters are present in the original geometry, and only partition into two sets. The spectral layout is shown, to explain the clustering.

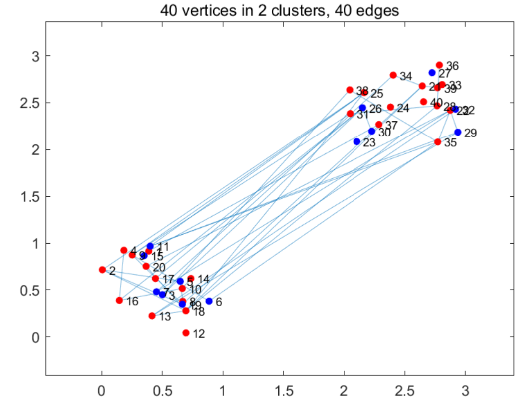
Random Complete Graph, Geometric Layout
40 vertices in 2 clusters



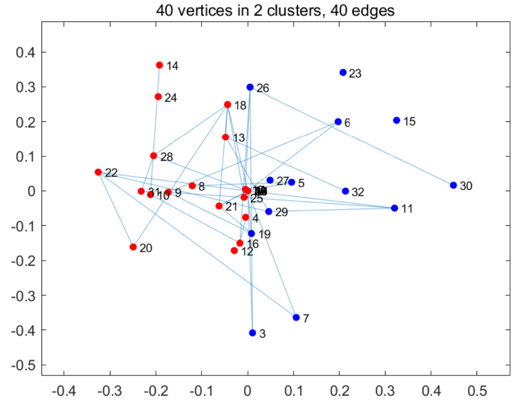
Random Complete Graph, Spectral layout
40 vertices in 2 clusters



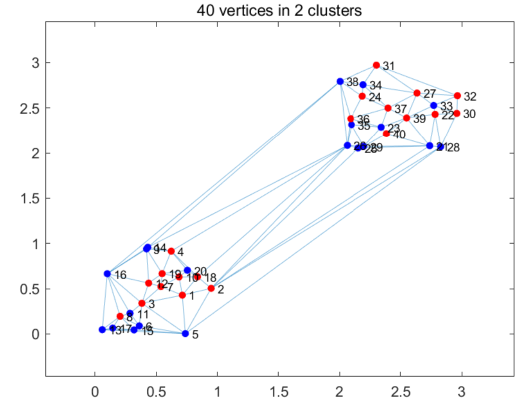
Random Subgraph, Geometric Layout
40 vertices in 2 clusters, 40 edges



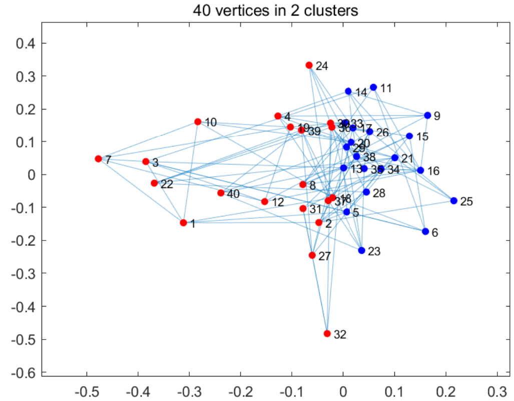
Random Subgraph, Spectral layout
40 vertices in 2 clusters, 40 edges

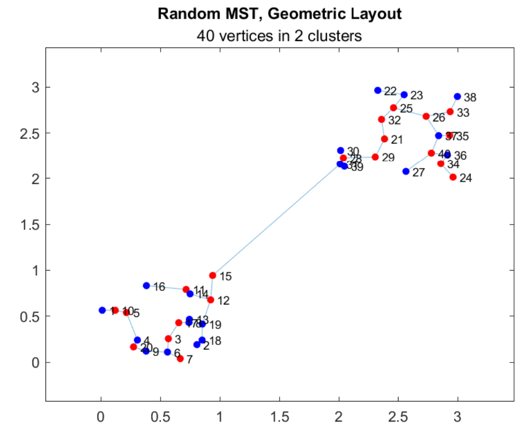


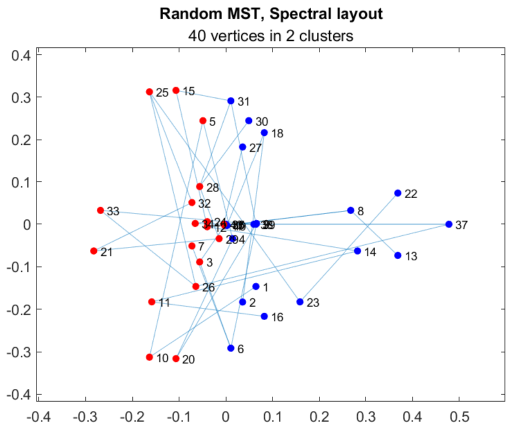
Random Delaunay Triangulation, Geometric Layout
40 vertices in 2 clusters



Random Delaunay Triangulation, Spectral layout
40 vertices in 2 clusters







As expected, we see that there is no discernible pattern. It seems again that spectra are not useful in answering geometric questions about geometric graphs.

References

[1] Anirban Banerjee and Jürgen Jost. On the spectrum of the normalized graph laplacian. *Linear Algebra and its Applications*, 428(11):3015–3022, 2008.

[2] Wikipedia. Curse of dimensionality. https://en.wikipedia.org/wiki/Curse_of_dimensionality, 2022.

[3] Hao Chen and Jürgen Jost. Minimum vertex covers and the spectrum of the normalized laplacian on trees. *Linear Algebra and its Applications*, 437(4):1089–1101, 2012.

[4] Jianxi Li, Ji-Ming Guo, and Wai Chee Shiu. Bounds on normalized laplacian eigenvalues of graphs. *Journal of Inequalities and Applications*, 2014(1):316, Aug 2014.

[5] Mei Lu, Huiqing Liu, and Feng Tian. An improved upper bound for the laplacian spectral radius of graphs. *Discrete Mathematics*, 309(21):6318–6321, 2009.

[6] Marshall Bern, David Eppstein, and Frances Yao. The expected extremes in a delaunay triangulation. *International Journal of Computational Geometry & Applications*, 01(01):79–91, 1991.

[7] Nimrod Megiddo and Kenneth J. Supowit. On the complexity of some common geometric location problems. *SIAM Journal on Computing*, 13(1):182–196, 1984.