Investigating Fiedler Values in Randomized Graphs

Leron Reznikov, Yuval Steinhart

Introduction

The Fiedler value, or second-smallest eigenvalue, of the Laplacian is an important measurement, describing the connectivity of an unweighted, undirected graph. Also known as the algebraic connectivity of a graph, it has many real-world applications and thus it is advantageous to study its behavior on various graphs. Some aspects of it are very simple: for example, we know that a complete graph's Fiedler value is equal to the number of vertices, and a disconnected graph has a Fiedler value of 0. Knowing these facts, it makes sense that as a graph becomes less sparse (by adding edges), its Fiedler value follows some increasing distribution until it reaches n.

While this basic idea is simple, we seek to investigate more in-depth how varying different parameters in graph generation affects the Fiedler value. Rather than focusing solely on modifying the size of the graphs, we adjust underlying parameters in several graph classes to find a relationship between them and the Fiedler value. Using the powerful tools provided by Python's networkx library, we are able to quickly generate relatively large graphs with finely-tuned parameters.

Fiedler Values of Randomized Graphs

For many classes of graphs, the Fiedler value is well studied as some function of the size of the graph. For example, as we saw in class, complete graphs, path graphs, star graphs, etc. all have easily computed Fiedler values as a function of the number of vertices. However, we were interested in graphs that are parameterized by more than just the size. More specifically, we sought to explore how changing the parameters of randomized graphs changes the Fiedler value. This is interesting for several reasons. First, with graphs that are parameterized by more than just the number of vertices, there are more factors at play in determining the Fiedler value. While the Fiedler value of complete graphs can be easily computed in terms of the number of vertices, more complex graphs require a deeper understanding of the structure to determine the algebraic connectivity. Second, unlike deterministic graphs, randomized graphs can vary wildly between executions, even with the same parameters. This means that not only does the Fiedler value no longer follow a simple formula, but there is no general formula at all since the exact structure of the graph cannot be determined just from the parameters. We explored several graph classes, and for each one, we varied the parameters and attempted to understand the asymptotic growth of the Fiedler value.

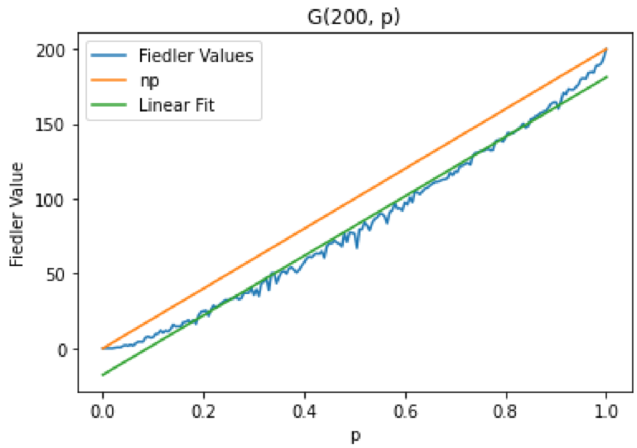
G(n,p)

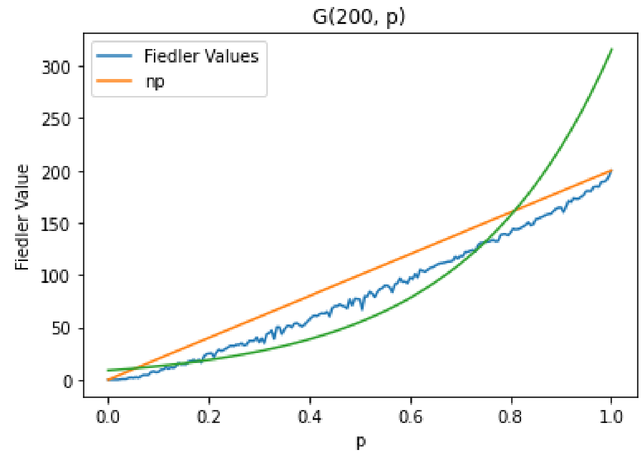
The first graph class we explored was the G(n, p) graph model (also known as the Erdos-Renyi model), parameterized by the values of $n,\ p$. n represents the number of vertices and p is used for edge generation. The description is simple: for each of the possible $\binom{n}{2}$ edges, it is randomly added with probability p. The extremes of this are obvious. If p=0, the graph is disconnected and, as we know, the Fiedler value of a disconnected graph is 0. On the other extreme is p=1, in which case every edge is added and the graph is connected, where we know the Fiedler value is n. We sought to explore the values between these two extremes.

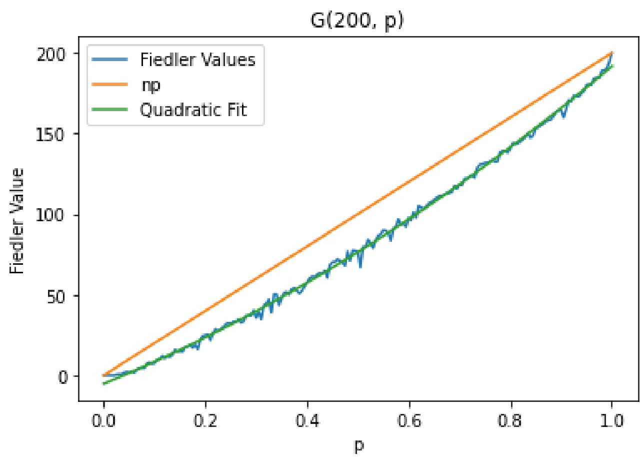
We first examined some theoretical work. G(n, p) is a well studied graph class. In particular, the connectivity of a G(n, p) graph is a well studied problem. It turns out that when $p < \frac{\ln(n)}{n}$, the graph is disconnected with high probability and when $p\ >\ \frac{\ln(n)}{n}$, the graph is likely connected. In terms of the Fiedler value, this means that for $p < \frac{\ln(n)}{n}$, the Fiedler value is equal to 0 with high probability. As we will discuss later, this result was also observed empirically on graphs that we generated. Thus further restricts the set of "interesting" values of p to where $\ln(n) < np < n$.

We first generated a series of G(200, p) graphs, varying p between 0 and 1 in increments of .005. For each graph, we computed its Fiedler value and the value of np. The plot is shown below:

As expected, we observed that for p < $\frac{\ln(200)}{200}$, the Fiedler value was equal to 0, verifying the theoretical result that it would be disconnected. As expected, we see the larger values steadily increase as np increases. We attempted to fit a curve to the graph in order to approximate the order of growth.







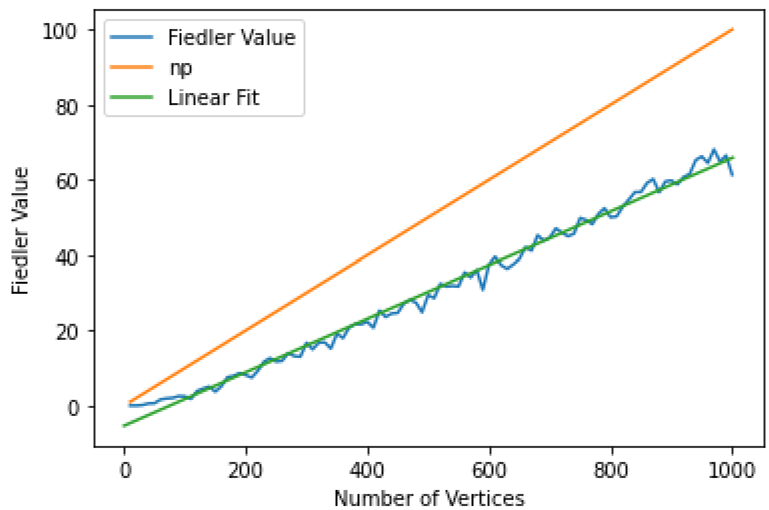
As you can see, both the linear and quadratic models seem to fit the data well. This seems to approximately agree with Kolokolnikov, Osting, Von Brecht[1], who showed that

$\frac{\lambda_2}{np} \sim a(p_0) + O(\frac{1}{\sqrt{np}})$, where $p_0 = \frac{np}{\ln(n)}$ and $a(p_0)$ is the solution to

$p_0 - 1 = ap_0(1 - \ln(a))$. $a(p_0)$ is non-trivial to compute explicitly, but grows very slowly for

[1] Von, T. "ALGEBRAIC CONNECTIVITY OF ERDŐS-RÉNYI GRAPHS NEAR THE CONNECTIVITY THRESHOLD." (2014).

larger values of p_0, thus agreeing with a bound approximately linear in np.

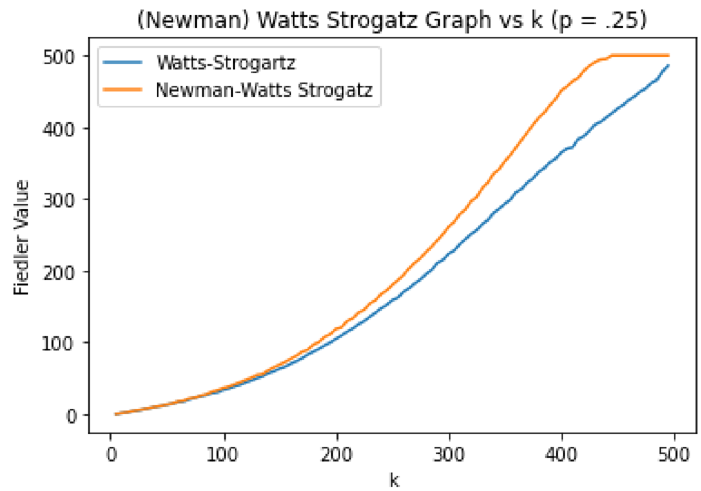


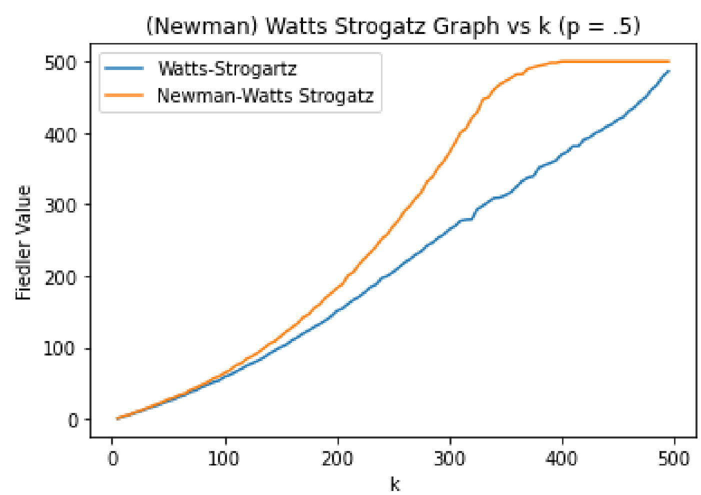
This is even clearer when we look at the Fiedler value as the number of vertices increases for a fixed value of p, in this case $p = .1$. For small n, we still observe that the Fiedler value is equal to 0, since np is then very small. However, as n increases, we see that the Fiedler value grows linearly with the number of vertices. This backs up our hypothesis that the Fiedler value grows approximately linear in np.

Watts-Strogatz and Newman-Watts-Strogatz Graphs

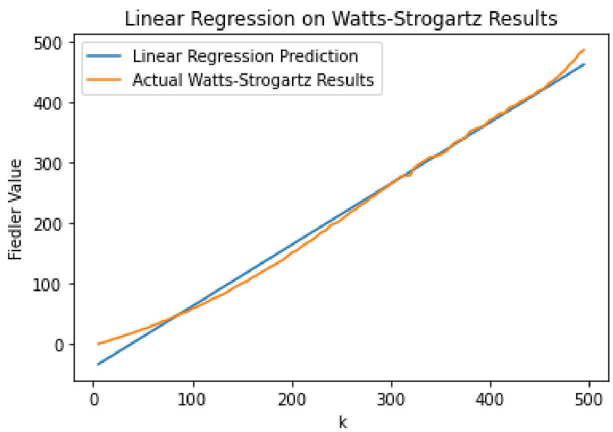
We also explored two related graph categories. A Newman-Watts-Strogatz graph is generated with three main steps: First generate a ring of vertices, then connect each vertex with its closest neighbors, and finally for each existing edge add a random edge with some probability. A Watts-Strogatz graph is quite similar, except in the final step if a new edge is added the original edge is deleted. Both graphs are parameterized with n, the number of vertices; k, the number of nearest-neighbors to connect with in the second step; and p, the probability of adding (or replacing in the case of Watts-Strogatz) an edge in the final step. Note that a

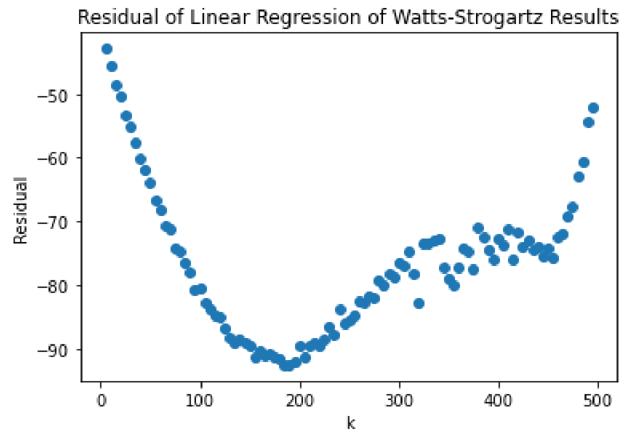
Newman-Watts-Strogatz graph is guaranteed to be connected, as the nearest neighbor connections are never removed. A Watts-Strogatz graph can technically be disconnected, but even with a moderately large *k* value it is extremely likely to be connected. We choose to study these graphs both because of the multiple parameters, and also because we were interested in how the general localization of connections resulting from connecting the nearest neighbor affects the Fiedler value.



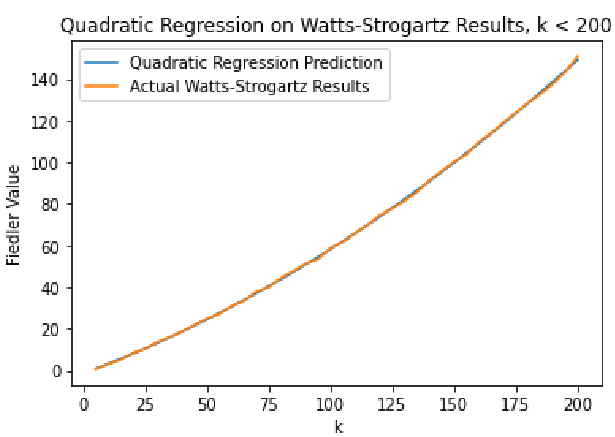


We first investigated the result of increasing *k*, fixing *n* at 500. As expected, the Fiedler value monotonically increased as *k* did. The Watts-Strogatz graphs did not appear to be affected too much whether *p* was 0.5 or 0.25. The Newman-Watts-Strogatz, however, appears to increase more rapidly with higher *p* and tops off at the maximum Fiedler value of *n* earlier. This makes sense – since edges aren't removed, increasing the probability of adding a new edge effectively turns it closer to a fully-connected graph. Since this maximum was reached earlier with Newman-Watts graphs, we chose to focus our regression on Watts-Strogatz. At first glance, a linear regression appears to be a relatively accurate representation of the data. However, looking at the residual, it is clear that there are other



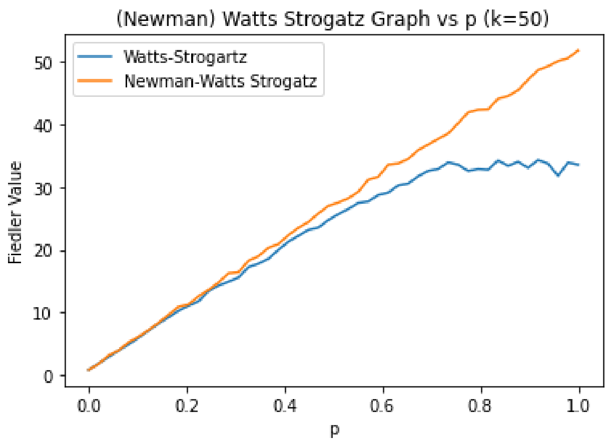


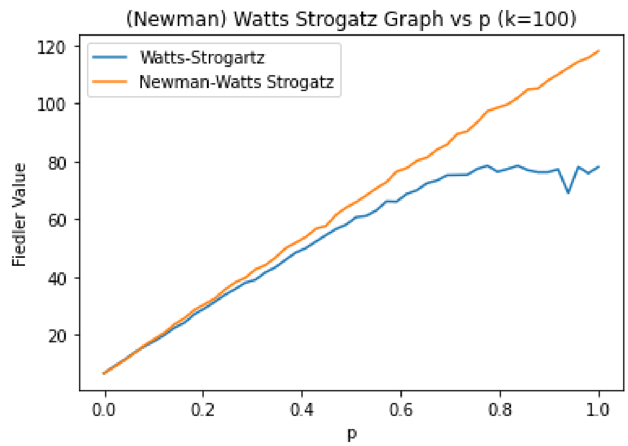
underlying patterns at play. It appears from the residual that the pattern tends to change at around *k=200*; we hypothesize that this is where there are enough random edges added that it appears to behave more like a G(n, p) random graph. In fact, if we analyze only up to *k=200*, we get an almost perfect quadratic fit. Besides simply giving an important lesson in paying attention to residuals when attempting to fit to data, we thought that this quadratic trend was very interesting. It means that when $k << n$, adding another neighbor to every node increases the connectivity of a



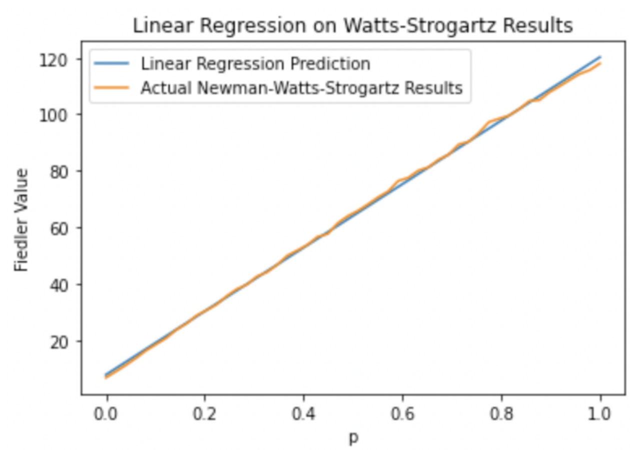
graph quadratically. While this is likely a known result, we believe that this holds interesting applications to many real-world areas, such as internet network connectivity. A Watts-Strogatz graph with $k << n$ can be viewed as a relatively accurate representation of many real-world networks, where a node is connected to many around it, as well as a few random distant connections. The quadratic increase (seen even with small values of p) shows that even adding a few additional neighbor connections goes a long way towards increasing the general connectivity or “robustness” of the graph.

We also investigated how varying *p*, while keeping *n* and *k* constant, affected the Fiedler value. In these graphs, we kept *n=500* as before. Once again, the Newman-Watts Strogatz and





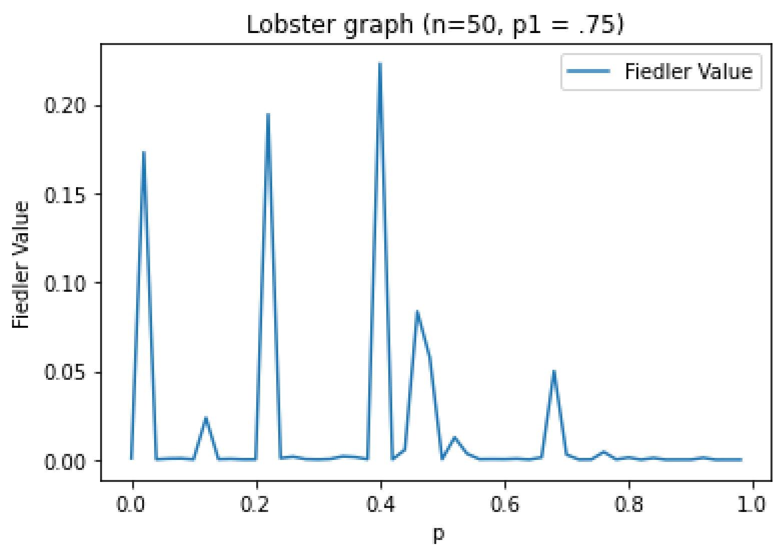
Watts-Strogatz graphs appear to behave differently at the extremes. From a purely visual analysis, it appears that both begin on a steady linear increase, but at a certain point the latter begins to taper off while the former continues on until the Fiedler value reaches around (or just over) *k*. When *p=0*, the two graphs are the same, so it makes sense that they begin at around the same point. Moreover, it is also intuitive that a Newman-Watts-Strogatz graph continues increasing as *p* increases; since no edges get deleted, the higher *p* is the more edges exist in the graph, thus the connectivity increases as well. This fact is confirmed by conducting a linear regression on the Newman Watts Strogatz Graph, clearly showing a linear relationship.



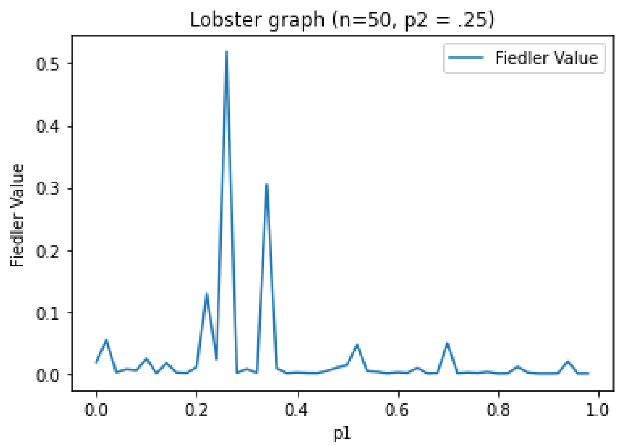
The Watts-Strogatz graph, though, is more interesting. It appears that at a certain point, increasing the probability of replacing a connection to a neighbor with a completely random connection stops increasing connectivity of the graph. Moreover, from the visualizations and additional experimental results, it appears that as k increases, the maximum bound for the Fiedler value approaches closer and closer to k, but it always appears to be hit around a p value of 0.7 or so. In future work, we hope to explore in more depth various details about this apparent maximum value, especially why it gets hit around the same p value.

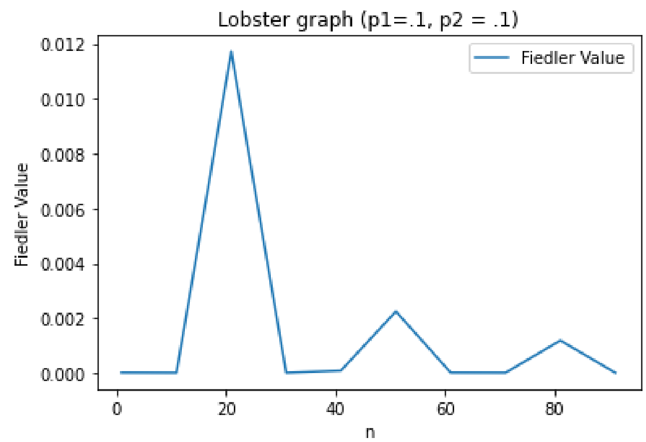
Random Lobster Graphs

The final graph class we looked at was Random Lobster Graphs. A lobster graph is a tree that becomes a caterpillar graph when all leaves are removed. A caterpillar graph is a tree that becomes a path graph when all leaves are removed. In other words, a lobster graph is a tree in which every node is either on the main backbone, one away from the backbone, or two away. Lobster graphs are parameterized by three values: n, p_1, and p_2. Like before, n is the expected number of vertices in the graph. Here, p_1 is the probability of an edge being added at level 1 and p_2 is the probability of an edge being added at level 2. When p_2=0, it is simply a caterpillar graph. We first looked at how changing p_2 affects the Fiedler value.



We can see that increasing p_2 does not seem to have a significant effect on the Fiedler value. This makes sense, since increasing p_2 will add some number of additional nodes and a single edge for each node that is added. Since this does not increase the overall connectivity of the graph, we do not expect to see a significant increase in the Fiedler value.





We see similar results as we vary p_1 and n. Regardless of the value of p_1 or n, the graph is a tree, which is minimally connected. This ensures that the algebraic connectivity will be low, but non-zero. In addition, varying n does not seem to affect the size of the graph significantly. n is the expected number of nodes, but in practice, the number of nodes can range much higher than n depending on the values of p_1 and p_2. Either way, we see that the Fiedler value does not appear to scale very much with n, p_1, or p_2. We would expect similar results for any tree, but it was interesting to verify it on a random tree class.

Conclusion

We investigated three random graph classes to see how varying the parameters affects the Fiedler value. We first looked at the well-studied problem of G(n, p) graphs, where an edge between any two nodes is added with probability *p*. We explored the probability that such a graph is connected, and presented several regressions to model the Fiedler's value increase as *p* increases, comparing our results with theory. We also looked at Watts-Strogatz and Newman-Watts-Strogatz small-world graphs, which are extremely helpful in modeling real-world scenarios. We saw how varying *p* and *k* affected the connectivity of these graphs. Finally, we investigated the lobster graph, which was the sole graph class that did not appear to have significant results from the parameter variation.

Algebraic connectivity is extremely useful to study. Obvious applications include using it as a metric to check the robustness of a computer network; higher connectivity could result in a network that is more resistant to attacks or other technological failure. It can be even used in some unexpected areas. We found studies discussing how algebraic connectedness of the brain network is connected to risk of developing Alzheimer's Disease.[2] Other papers discuss algebraic connectivity related to gene expression[3] and other biological applications.[4] We hope that our work serves as a good foundation for future research into random graphs' parameters and how they affect the Fiedler values.

---

[2] Daianu, M., Jahanshad, N., Nir, T. M., Leonardo, C. D., Jack, C. R., Jr, Weiner, M. W., Bernstein, M. A., & Thompson, P. M. (2014). Algebraic connectivity of brain networks shows patterns of segregation leading to reduced network robustness in Alzheimer's disease. Computational diffusion MRI : MICCAI Workshop, Boston, MA, USA, September 2014. CDMRI (Workshop) (6th : 2014 : Boston, Mass.), 2014, 55–64. https://doi.org/10.1007/978-3-319-11182-7_6

[3] Zoran Nikoloski, Patrick May, Joachim Selbig,Algebraic connectivity may explain the evolution of gene regulatory networks,Journal of Theoretical Biology,Volume 267, Issue 1,2010,Pages 7-14, ISSN 0022-5193,

[4] Kim, Y. (2016). Algebraic Connectivity of Graphs , with Applications.