Solutions to Assignment 1
Advanced Tools for Complex Network Analysis
CNET 5052, Spring 2026

Caleb Chandler*

1. *Question 1.*

(a) To reproduce the figure, I first obtain the "normalizing" clustering/path length values by calculating them on a non-rewired ring lattice. I then run a loop to run and save the results of 20 trials as a list of values in a dictionary mapping p-values to their results. I then create a list of averaged values and plot the results in the manner – to the best of my ability – of the original paper. Path lengths drop rapidly following increase in p, whereas clustering only begins dropping at about 0.15, at which point it does so even more rapidly. The most important insight from this figure is the existence of a wide range of p where the network has both high clustering and a short average path length, the defining feature of small-world networks. The immediacy with which path length drops demonstrates the outsized efficacy of rewiring as far as improving connectivity.

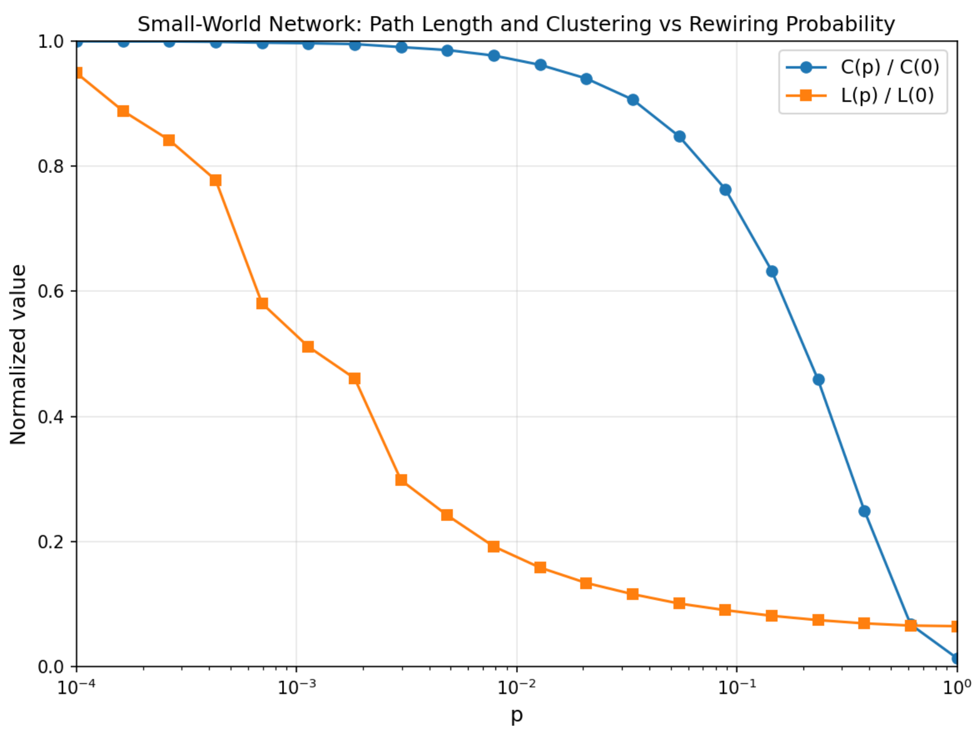
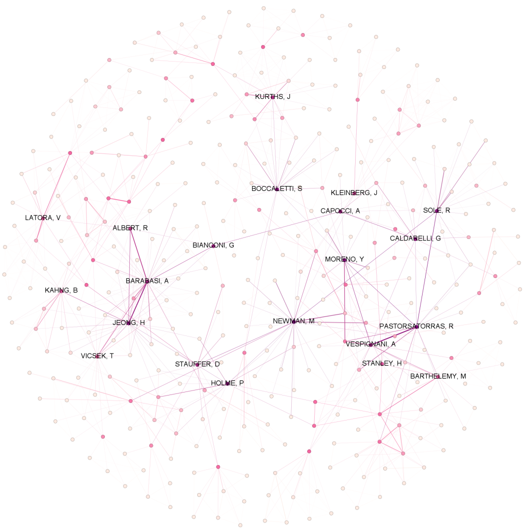


FIG. 1: Reproduction of Watts-Strogatz figure.

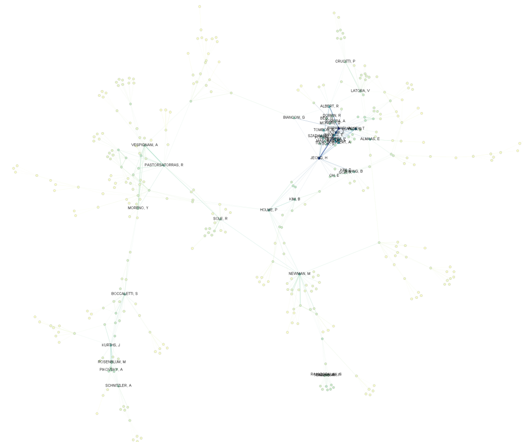
* **Repo Fork:** https://github.com/caleb-chandler/cnet5052_sp26

2. *Question 2.*

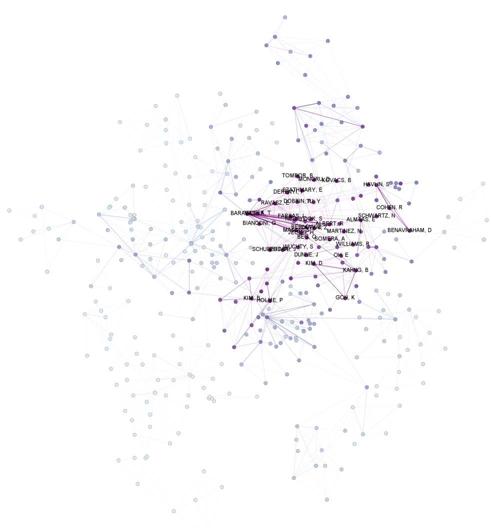
(a) I created 4 visualizations in Gephi using 3 layout algorithms. I used Fruchterman-Reingold to visualize betweenness centrality and communities, Yifan Hu to visualize eigenvector centrality, and OpenOrd to visualize the HITS authority scores. For centralities, labels are restricted to only the highest nodes; for the communities, they (along with edge thickness) are determined by weighted degree. I wanted to get a wide range of viewpoints on the network which is why I chose to include several layouts. An interesting observation I found is that the HITS scores are all extremely localized around Barabasi; this is likely due to the recursive/self-referential nature of the HITS algorithm, where authority is concentrated on nodes linked to by the most prominent hubs. In the context of the netscience dataset, since Barabasi sits at the center of the most densely interconnected core of the network, the algorithm "drains" authority from the periphery and localizes it almost entirely on him and his immediate circle.



(a) Betweenness Centrality (FR)



(b) Eigenvector Centrality (Yifan Hu)



(c) HITS Authority (OpenOrd)

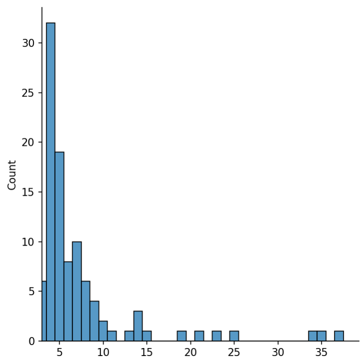
(d) Communities (FR)

FIG. 2: Visualizations of netscience.gml

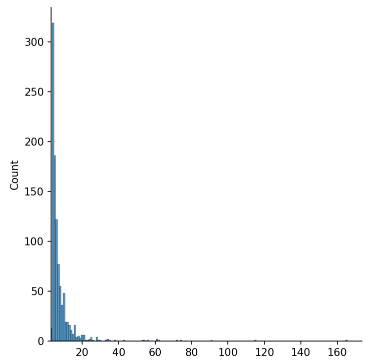
3. *Question 3.*

(a) I found that the way I was doing it at first (calculating the connection probability over and over within the loop for each node) was too slow, so my implementation uses a list of nodes where the node ID repeats the same number of times as its degree. This way I can just randomly sample from that list and it accomplishes the same thing.

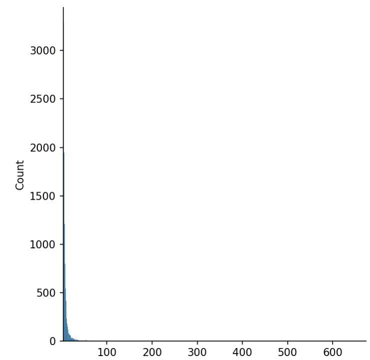
(i) The distributions remain heavy-tailed regardless of size, but become continually more so as the network grows. This is because preferential attachment is a cumulative process that "snowballs" as N increases.



(a) N=100



(b) N=1000



(c) N=10000

FIG. 3: Degree distributions at various sizes.

(ii) skipped per announcement

(iii) Apart from the beginning where N=0, clustering remains at a very low level, resulting from the predominance of hubs in the absence of an attachment-based-on-neighbors mechanism.

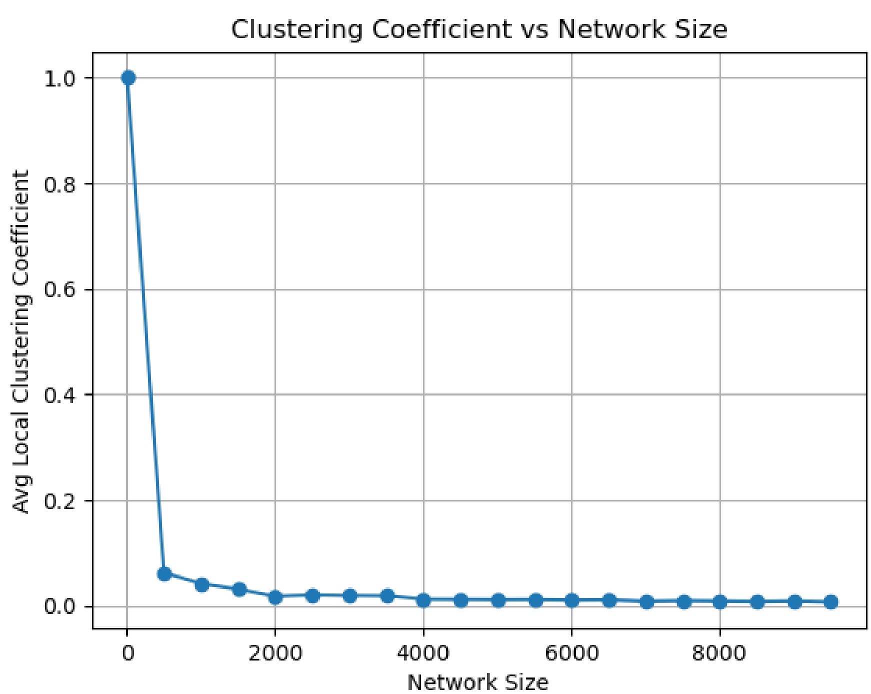


FIG. 4: Avg Local Clustering Coefficient vs. Size

(iv) The nodes added later on take a bit to get going, but once they do, they seem to grow just as rapidly relative to their size as the ones before them. This is because all nodes are subject to the same preferential attachment mechanism.

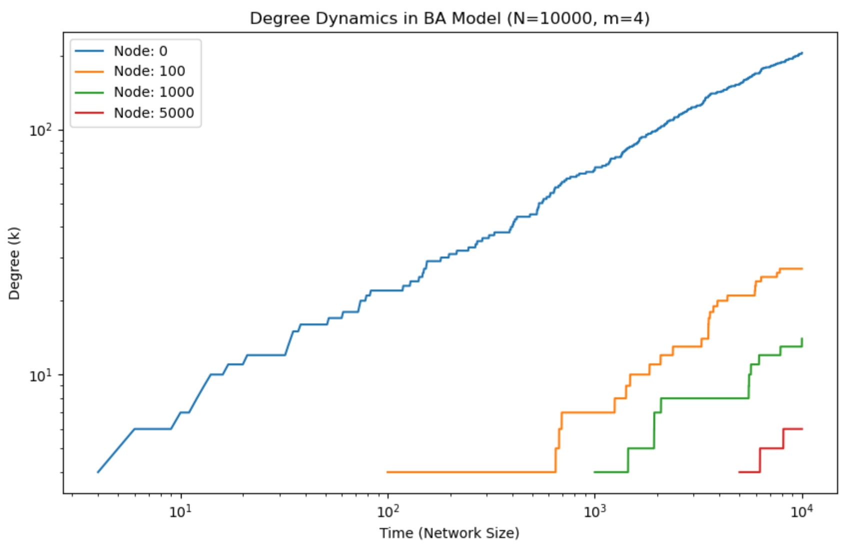
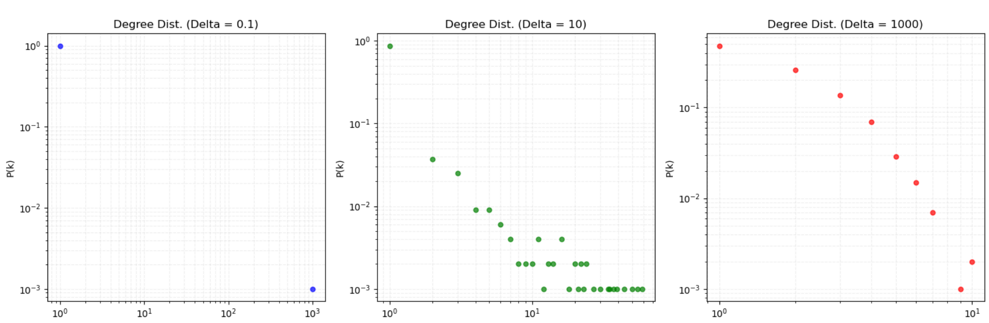
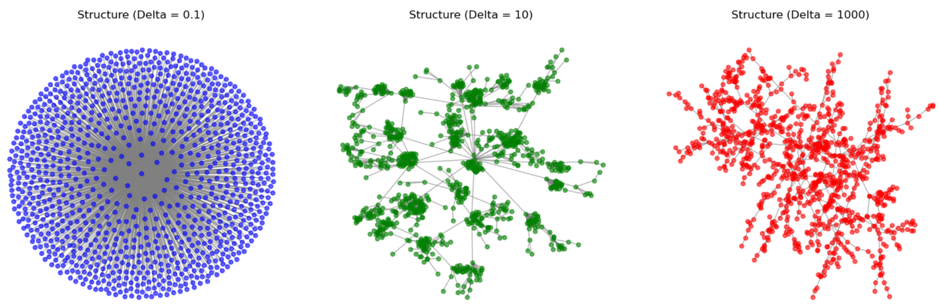


FIG. 5: Degree Dynamics Plot

(b) My implementation initializes a root node at (0,0) and iteratively adds N nodes with random coordinates. For each new node, it calculates a cost for every existing node using the formula $cost = \delta \cdot h + d$. It then attaches the new node to the single existing node that minimizes this cost.

(c) Delta = 0.01 generates a star network, 10 generates a scale-free network, and 1000 creates a purely geometric network where nodes connect only to their nearest physical neighbor. This is reflected both in the degree distributions and the visualizations. Delta is essentially a slider for how much you want centrality to matter vs geography.





(d) For this I wrote a function to measure pi(k) for all ks in all the generation models, then used it as input to a plotting function.

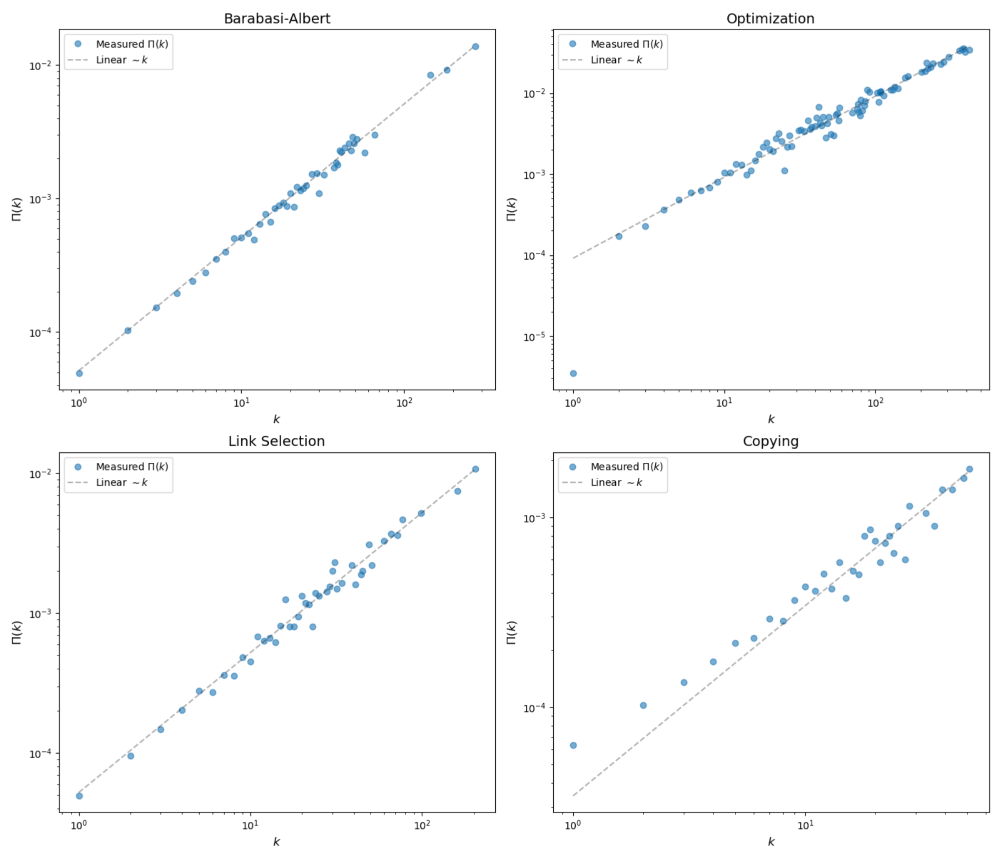


FIG. 6: K vs. Pi(K)

(e) We observe that all but Optimization seem to show a direct proportional relationship between k and pi(k). This is because Optimization is the only model whose rules don't involve degree and therefore don't mathematically force proportionality.

4. *Question 4.*

(a) (for both a and b) I came up with a distance measure designed for a specific scenario where you have two graphs and you want to see the extent to which the "importance" distributions are the same. The descriptor takes a graph G and returns an ensemble of probability distributions from four centrality measures: harmonic, betweenness, eigenvector, and degree. Distance is then calculated as the average of the Wasserstein distance between each pair of distributions. Harmonic centrality is used in place of closeness to make the descriptor robust to disconnected graphs. Wasserstein distance is chosen because it compares probability distributions directly. I think this is a good descriptor for the stated purpose because it accounts for each of the main four centrality metrics for a more holistic and "complete" output than if you were to just use one or two.

(b) answered above

(c) The descriptor function extracts the four centrality vectors for a given graph (betweenness is approximated with $k = 50$ nodes for efficiency). The final distance metric between the two graphs is then calculated as the arithmetic mean of the Wasserstein distances as discussed.

(d) (for last three) The plots demonstrate that the distance metric is indeed capturing something to the extent that it correctly places graphs of the same topology close together and can easily tell the difference between graphs of a different topology. The main thing I notice, apart from this, is that the distribution for BA vs ER is also much wider, meaning there's more variability in how much distance is between them.

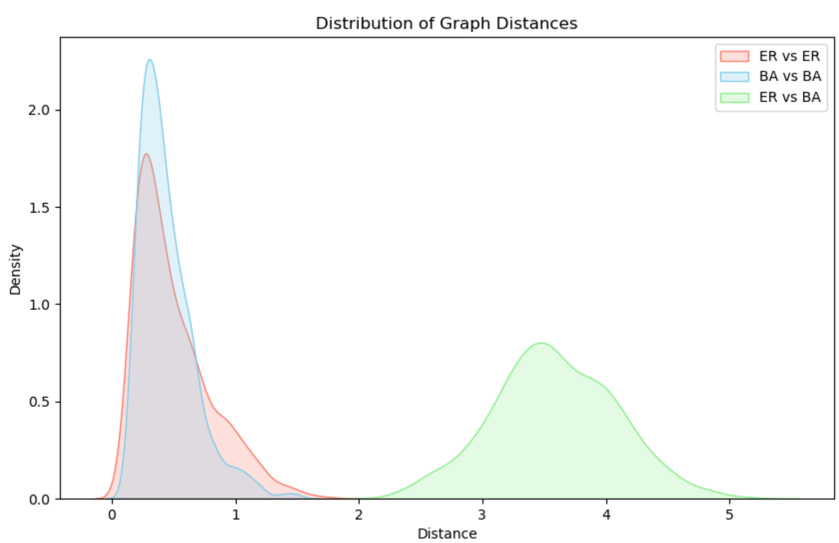


FIG. 7: Graph Distances Plot

5. *Question 5.*

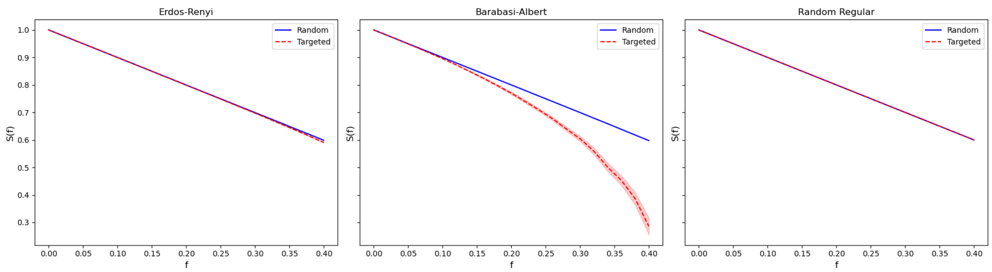


FIG. 8: Random vs Targeted Attacks

The BA network is the only one which is more susceptible to targeted attack. This makes perfect sense, given that it's the only one where the degree order matters. What is interesting, though, is the uneven *rate* at which the BA network collapses with targeted removal, where it doesn't seem to matter at all until it reaches about 10%. This "delayed" collapse is a consequence of the high average degree ($\langle k \rangle \approx 10$, so $m = 5$). The m parameter essentially encodes the level of redundancy, and the network only disintegrates once this redundancy is sufficiently eroded, which does not happen at a linear rate. There is also a vanishingly small uncertainty range, essentially a byproduct of the law of large numbers; a random sample of 100 nodes ($5000 \times 0.02 = 100$) will always have an average degree that is representative of the population.

If you reference anything, add it to the bibliography below and cite it in the text using `\cite{Hartle2020_wegd}` [1].

[1] Hartle, H., Klein, B., McCabe, S.D., Daniels, A., St-Onge, G., Murphy, C., & Hébert-Dufresne, L. (2020). Network comparison and the within-ensemble graph distance. *Proceedings of the Royal Society A*, 476: 20190744. doi: 10.1098/rspa.2019.0744.