Influence Maximization With the Independent Cascade Model Using Message Passing

Danny Satterthwaite[1,*]

[1]Vermont Complex Systems Institute, University of Vermont, Burlington, VT, USA
[*]dsattert@uvm.edu

ABSTRACT

This project explores potential performance gains that can be achieved when implementing message passing-based strategies for influence maximization using the Independent Cascade Model (ICM). I develop four algorithms rooted in ranking nodes based on their probability of becoming influenced at the end of an influence percolation process. Two of the approaches use clustering — either via connected components or spectral methods — to refine and improve seed selection by distributing influence across distinct regions of the network. These four approaches are then compared to a traditional greedy approach and heuristic approaches in three types of networks of various sizes — a Barabási-Albert network; a minimally connected clustered Barabási-Albert network; and a disjoint clustered Barabási-Albert network. The results show that my message passing (MP) approaches scale (in terms of run time) significantly better than the greedy methods while having a similar or better ability to detect the individuals who, when initially selected, lead to the greatest influence spread, particularly in clustered or disconnected networks.

1 Introduction

Understanding how individuals impact each other's actions is top of mind for nearly all businesses. Yet in our increasingly interconnected world, some people's actions seem to be more influential than others. Determining who said individuals are is not nearly as simple as it appears at first glance. In fact it is NP-hard[1]. The area of influence maximization attempts to do just this. Naturally, the marketing field has taken great interest in studying influence maximization. For example, if a company is launching a product, who should the brand send sample products to so the most people end up hearing about the product?

In this project, I consider the Independent Cascade Model (ICM), which models the diffusion of influence by allowing each individual a single chance to influence any of its uninfluenced neighbors. That is, given a system of individuals and their relationships represented by a network graph, our goal is to determine the most influential set of individuals to target so that they trigger a large cascade of further adoptions (influence).

Unsurprisingly, there are dozens of algorithms that attempt to solve this influence maximization problem. Simulation-based methods utilize repeated random simulations to estimate the expected spread of influence from a given set of initial nodes[2]. The most famous such approach uses a greedy algorithm to select seed nodes that produce the largest marginal increase in the expected influence spread and has been proven to capture 63.2% of the total influence spread if the true optimal seed is chosen[1]. However, as I will show, this approach scales poorly. Alternatively, heuristic-based techniques employ ranking metrics or simplified diffusion models to quickly identify influential nodes without the computational cost of extensive simulations[2].

I use message passing (MP), a state-of-the-art technique for quickly calculating properties of individual nodes in a network by propagating information between them, to develop and explore various new approaches and algorithms for the influence maximization problem[3]. These approaches leverage the idea of Giant Components (GCs) within a network and conceptually fall somewhere in between simulation-based algorithms and heuristic seed selection strategies. I then compare both the run time and final influence from my MP approaches to both the traditional greedy approach and the heuristic approaches for choosing starting seeds.

2 Methods

MP - Why and How

Many social networks have components of nodes that are clustered (closely connected) together, and often there exists a dominant component, a so-called Giant Component (GC), where a majority of nodes are connected together. MP has been shown to be a very efficient technique for calculating the probability that each node is in the GC. In the context of the ICM, given one has an existing network of users (our starting network), the ICM can be seen as a stochastic percolation process where each edge is retained with its influence probability (p_{ij}) and influence spreads through the network. Thus, the GC for an ICM percolation model is the largest group of nodes that are connected through paths of successfully activated edges. It represents the maximum possible set of nodes that could be influenced in a single run, based on where the initial seeds are placed. That is to say, the GC in the ICM percolation model is the limiting (in the limit) influence spread given an initial diffusion of influence.

Mathematically, following the work by Newman in his paper *MP Methods on Complex Networks* we have:

Each node indexed from $i = 1....n$ and μ_i = the probability that node i is not in the GC. In English, "node i is not in the GC if and only if none of its network neighbors are in the GC"[3]. Let N_i "be the set of nodes that are neighbors of i." Also, define p_{ab} as the probability that an edge is occupied (node a influences node b). Let $\mu_{i \leftarrow j}$ "be the probability that node j is not in the GC when node i is removed"[3]. Then, accounting for the edge probabilities, we have

$$\mu_i = \prod_{j \in N_i} (1 - p_{ij} + p_{ij}\mu_{i \leftarrow j})$$

which is the probability that node is not in the GC. But, it depends on $\mu_{i \leftarrow j}$ so calculating it, we have

$$\mu_{i \leftarrow j} = \prod_{k \in N_j \ , \ k \neq i} (1 - p_{jk} + p_{jk}\mu_{j \leftarrow k})$$

.

These are the core equations and form the basis of my algorithms and code. While the core assumption here is that nodes are independent (no loops), which is clearly not the case, this approach is still a plausible starting point as "MPA [MP approach] can be strikingly accurate even for networks whose local structure contains a high density of loops"[4]. In the future, I would like to explore implementing more sophisticated MP algorithms that use loops such as the formulas outlined in Erik Weis's paper *MP for Epidemiological Interventions on Networks with Loops*[5].

Parameters and Network Set Up

Before diving into the models, I first define several key parameters. In future works, a more exhaustive combination of parameters could be explored. The following is a list of parameters and rationale for their values. As my goal is to optimally and efficiently choose the optimal seeds (who to start the cascade of influence so that the most people end up influenced), core to my approaches (outlined below) is that at least one of the seeds must be in the GC, the limiting state of influenced people.

Network Size

As my core goals are to measure the quality of seed selection (measured via the percent of network that gets influenced when the seeds are chosen) and scalability of the MP approaches, I increase the number of nodes in the network from 50, to 75 to 150, to 200 and evaluate performance. Obviously, exploring larger networks would be ideal if compute time weren't an issue, but this selection of node sizes gives us a general sense of the performance trends.

Network Type and Structure

While there are various structures to consider, in all experiments going forward, I will use variations of a Barabási-Albert (BA) network as it captures preferential attachment — "rich get richer" — a phenomenon that is common in social networks. For parameter m, I use a value of ten, which means each new node forms ten connections, creating a realistic social network. However, motivated by the structure of more isolated communities like mountain towns, and to test the ability of my models to identify clusters, I also ran the same experiments on a modified graph that creates connected clusters of BA networks. More specifically, each cluster is generated as an independent BA network with a fixed number of nodes and preferential attachment parameter m=10. These clusters are sparsely interconnected with a fixed probability, with a one percent chance of connecting two nodes in different clusters – which allows the overall graph to grow in size by increasing the number of communities while maintaining local structure (25 nodes per cluster) and sparse inter-cluster connectivity. Lastly, I explore completely isolated communities, where the inter-cluster connectivity is zero. See Figures 1-3 below for visuals of the networks the experiments are based on.

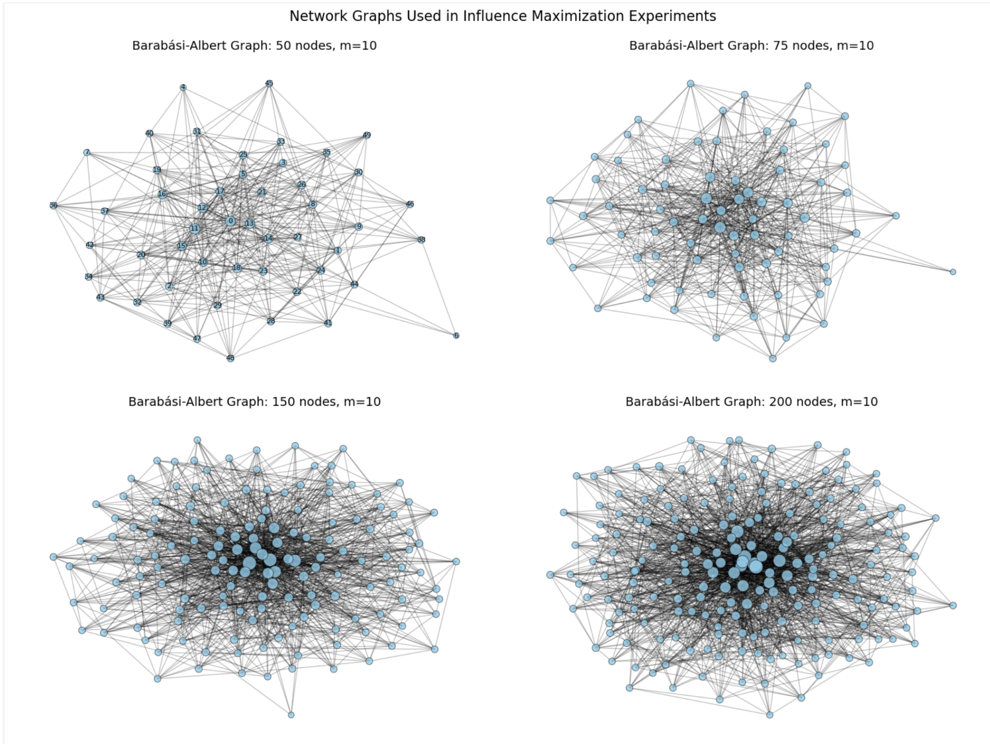


Figure 1. Stock BA Networks

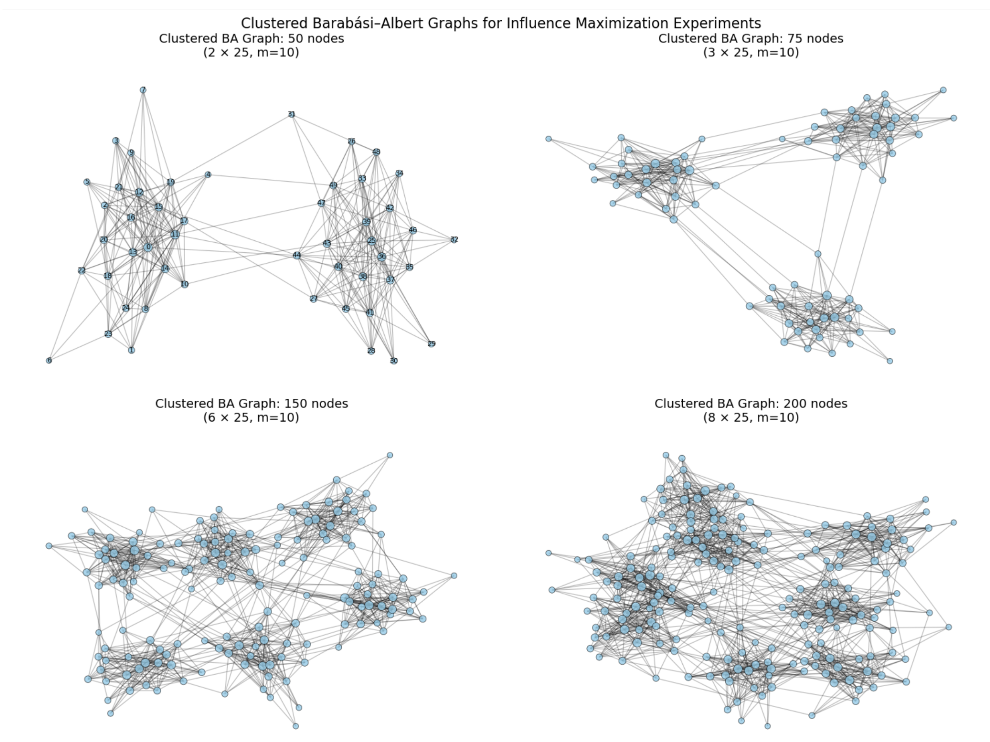


Figure 2. Minimally Connected Clustered BA Networks

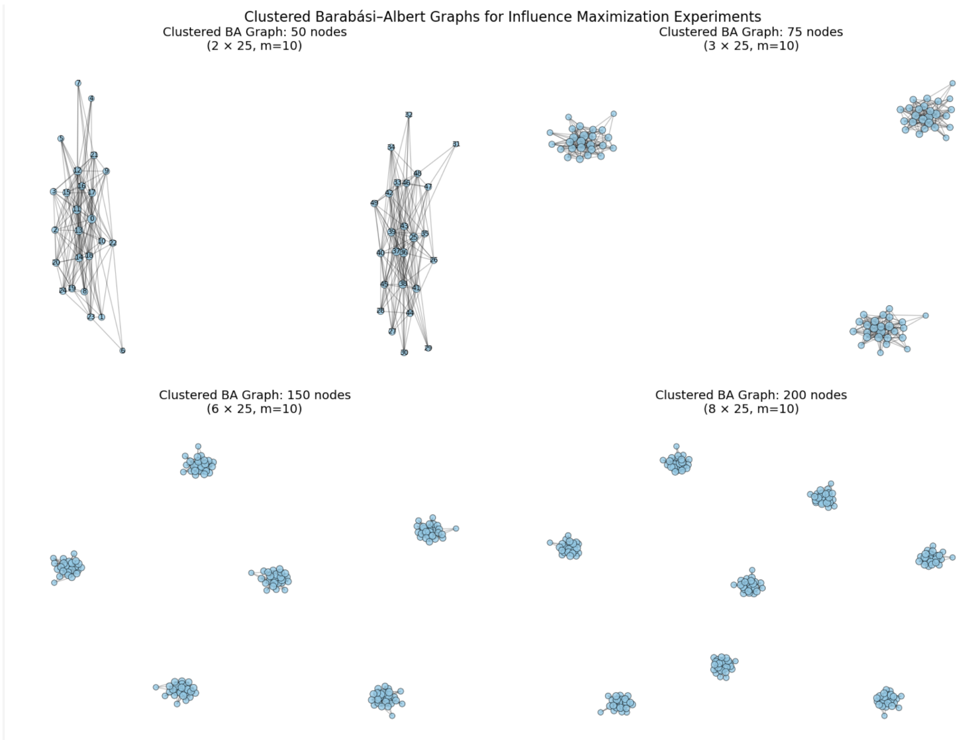


Figure 3. Isolated Clustered BA Networks

Simulation Approaches

In this project, both discrete and continuous simulation approaches were explored, but the continuous approach was chosen to be more accurate on larger networks and also to demonstrate the approach we have learned in class (MOCS II). Per the ICM, the simulation starts from an initial set of seed nodes, and each newly activated node attempts exactly once to activate each of its inactive neighbors with a given probability (p). The process repeats in rounds until no more activations happen. Looking more closely at the continuous approach, I use a General Event-Driven Queue to process influence events in chronological order, with exponentially distributed time delays determining when activation attempts occur, just like we have done in class. Starting from seed nodes, the process continues until either no more activations are possible or the maximum time is reached. The equivalence of both discrete and continuous approaches can be seen below in Figures 4 and 5, where all nodes are the same color between the graphs (percent that get influenced over ten trials of each approach for a set of seed nodes).

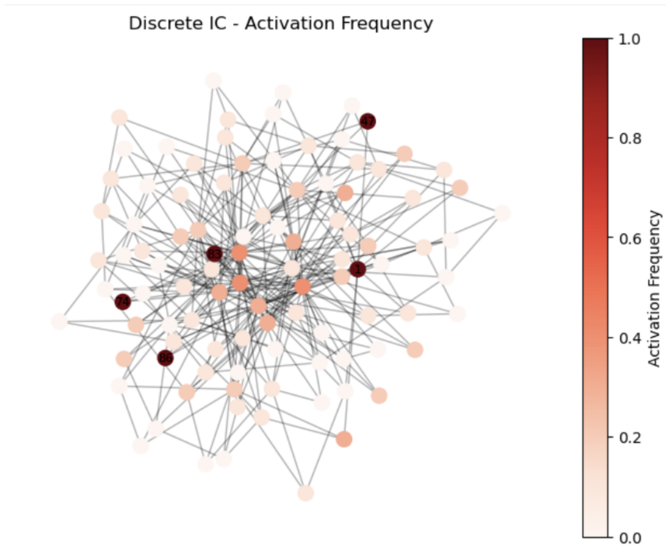


Figure 4. Sample Discrete Simulation

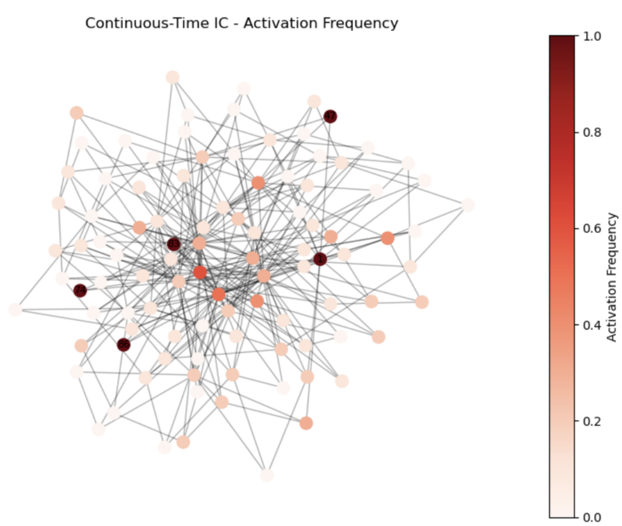


Figure 5. Sample Continuous Simulation

Number of Optimal Seeds to Find (k)

This depends on various factors such as marketing budgets, etc., but to start, I compare two values, three and ten, as these seem to fit the network sizes explored. Three provides us with a small test case and could be relevant for a small company, while ten allows us to test penetration into more diverse segmented communities.

Trials
All results are averaged over ten trials.

Influence probability (p)
The probability a node influences another node. All experiments use $p = 0.1$. Anecdotally, a ten percent chance that someone influences another person seems reasonable. A further exploration of p values will follow in the results.

Other:
For all other parameters, see code. If not explicitly stated, "thresholds" are set to 0.5

Using these, I develop and test four algorithms that build off the aforementioned MP ideas. These are outlined and discussed below.

Model 1: IM-CP-G
This is the greedy approach outlined by Kempe in one of the original influence maximization papers *Maximizing the Spread of Influence through a Social Network*[1]. It is a frame of reference and a starting point, and I will show that it scales poorly and still does not always outperform than other approaches. The algorithm is greedy and iteratively selects k seed nodes that maximize influence spread. For each selection, it evaluates every remaining candidate node by running Monte Carlo simulations of the ICM to estimate the expected influence spread. After identifying the best node in each iteration (the one providing maximum marginal gain in influence), it adds that node to the optimal seed set and continues until k seeds are selected.

Model 2: MP-GC-1
My first model implements a MP algorithm that computes the probability of each node not being in the GC of a network. More specifically, messages are initialized and iteratively updated between connected nodes until convergence, with each message representing potential influence between nodes. It calculates the probability that a node is in the GC of the graph and ranks by said probability accordingly (largest to smallest). It finally picks the top (based on the aforementioned probability of being in the GC) k (parameter for number of seeds) nodes and uses these as the seeds. This approach is clearly quite similar to the heuristic approaches of choosing the nodes that are most central or have the highest degree, but its implementation of MP is unique and has strong potential. An example visualization is shown below in Figure 6 on a BA network of size 50, $m = 10$, $k = 3$, and influence probabilities $p = 0.1$.

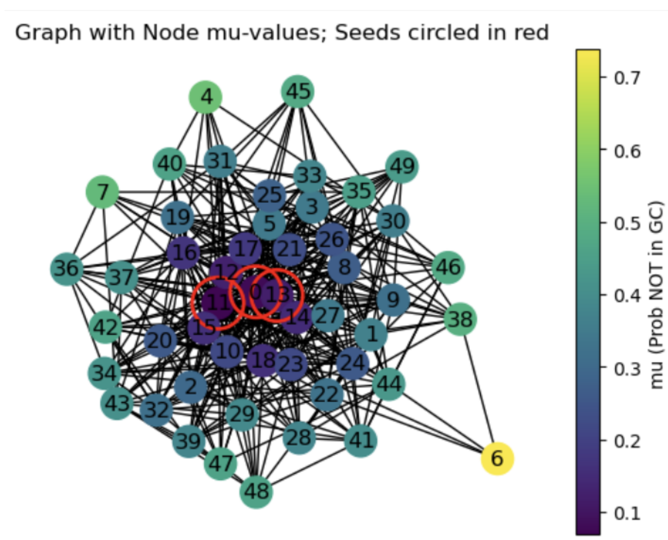


Figure 6. MP-GC-1 Example

Model 3: MP-GC-2
Next, to build off MP-GC-1, I wanted to account for the fact that my first approach could theoretically choose high-ranking nodes that are still close to each other, which could limit our potential spread of influence within the GC. In other words, the goal of this approach (and the next two) is to identify influential nodes that are spatially distant in the network. I begin by identifying nodes that are likely to be in the GC, exactly as I did in MP-GC-1. From there, these nodes are used to create a subgraph that

only includes the nodes most confidently in the GC, where $1-\mu_i$ exceeds a threshold of 0.5. While the GC in the full graph is, by definition, a single connected component, this filtered subgraph may break into multiple disconnected regions as it filters out the weakly connected nodes. Within this filtered subgraph, I identify connected components using NetworkX's connected components function and discard any that are smaller than 5% of the network's size. From each remaining component, I pick a single seed node, the one with the highest $1-\mu_i$. If fewer than k such regions exist, I fill in the remaining seeds by the global GC probability rankings. So, this strategy should help ensure that I not only target influential nodes but also distribute them across different well-connected regions, avoiding redundancy and improving robustness.

Model 4: MP-GC-3

Now, using the same ideas, I wanted to take this a step further. For example, say I had clear minimally connected regions of nodes (like my clustered BA graphs in Figures 2 and 3 or a planted partition graph), how can I make sure that I don't choose seeds all in the same (and isolated) region? That is the goal of this model. It identifies all connected components in a graph by returning a list of node sets using NetworkX's built-in component functions. From there, similar to earlier approaches, it first identifies the most influential node in each connected component based on their probabilities of being in the influence percolation GC. The function prioritizes components with nodes most likely to be influential, selecting these nodes first before checking when there are more or fewer components than required seeds. More specifically, it selects the top k components whose best node has the highest probability of being in the GC. When there are too few components, additional influential nodes are selected from the entire graph. Comparing this to the previous approach (MP-GC-2), while MP-GC-2 first filters the graph to include only nodes with high influence potential (above a threshold) and focuses only on sufficiently big connected regions within this filtered subgraph, MP-GC-3 identifies natural connected components in the original graph and selects the best node from each. Furthermore, given the component-based approach, MP-GC-3 has the potential to struggle when there are no disconnected components in the graph, as is the case with my first two types of network, where there is only one component. I attempt to remedy this next in MP-GC-4.

Model 5: MP-GC-4

The final approach I will try is a variation on the previous approach that uses Spectral Clustering to identify clusters. Again, the goal is to try to minimize the likelihood that the seeds chosen are close to each other. This approach differs from the previous approach in the sense that it doesn't consider a global percolation of influence. Instead, it runs the spectral-based clustering to find as many clusters as there are seeds (k), and then within each cluster runs the MP percolation — a sort of divide and conquer approach. More specifically, it first partitions the graph into exactly k clusters using Spectral Clustering. Then, within each cluster, it runs my MP percolation algorithm to estimate the probability each node is in the GC (influenced in the limit) and then chooses the most influential node from each cluster. If the number of selected seeds doesn't match k, it adjusts by selecting or removing nodes based on global influence. An example of MP-GC-4 is shown below in Figure 7 on a minimally connected clustered graph. As desired, the seeds chosen have the highest probability of being in the GC (darkest) and are in different clusters.

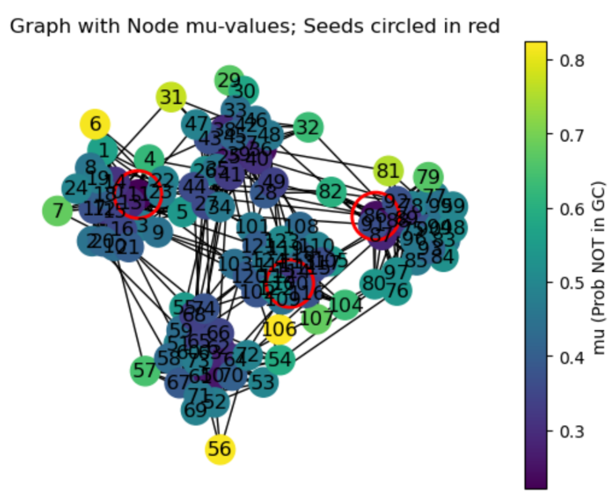


Figure 7. MP-GC-4 Example

3 Results

Overview

I first discuss the viability of MP on my networks by comparing a brute force (empirical) GC membership simulation with a MP simulation. From there, in order to evaluate both the scaling ability and quality of seed selection, I tracked both run time and the percentage of the network that gets influenced given a choice of seeds. To convey run time performance and scalability, I created a plot comparing run time vs the size of the network. Next, to compare the quality of seed selection, I created violin plots for each approach in the context of the percentage of the network that gets influenced by the selected seeds. Lastly, I conduct sensitivity analyses on the two core parameters k and p.

A Further Look at MP

As mentioned above, MP can be strikingly accurate even when a network has loops[4]. But to what extent is MP accurate on a BA network with varying amounts of attachment (m parameter)? To answer this, I used a brute force (empirical) approach to estimate the probability that each node is part of the GC in the influence percolation process by running 10,000 simulations where edges are retained with probability $p = 0.1$. For each trial, it constructs the largest connected component and tracks how often each node appears in it. I conducted experiments with varying network sizes and m parameters, with the results being similar across network sizes. For the sake of this example, I use a network of size 200, fixed $p = 0.1$, and with m increased from three to ten to 75. I create three plots. First, for each node, it plots the probability of being in the GC from the empirical approach vs. the probability of being in the GC found using MP. I next plot the graph and color the nodes in the top 20 percent of being in the GC from either approach with the color corresponding to the difference in the MP probability and the empirical probability. This was done as my approaches revolve around nodes with a high probability of being in the GC, the closer the difference is to 0, the closer the MP approach is to reality. Lastly, I rank each node by the probability it is in the GC for each approach and plot said ranks against each other. See Figure 8 below (analysis comes after Figure 8): the first row shows results for $m = 3$, the second for $m = 10$, and the third for $m = 75$.

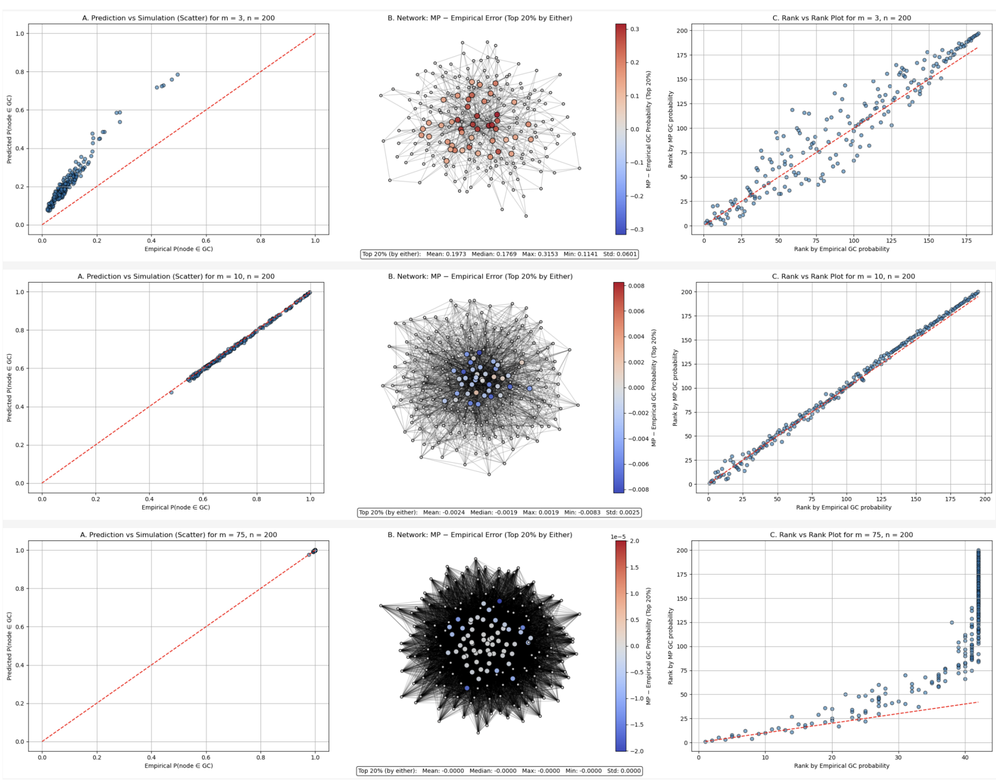


Figure 8. Comparing MP with a Brute Force Empirical Simulation Approach for Computing GC Membership Probability

Looking at the first row ($m = 3$), we see that across all nodes, the probability of being in the GC according to the MP approach exceeds the probability that said node is in the GC when using the empirical simulation approach. This can be further seen in the second plot in the first row, looking at the nodes in the top 20 percent of likelihood of being in the GC. We see the

MP-based probability for said nodes is on average (median) 17.7 percent larger than the empirical approach. Yet, looking at said nodes with the largest difference, they are clearly in the GC, and thus, this is not a concern. Looking at the rank (rightmost plot), we see a reasonable correlation between ranks, with many points falling around, but above and below the 45-degree line. The highest ranking nodes being above the 45-degree line again demonstrates the moderate (but certainly not overly severe) overestimation of the MP approach. Thus, for $m = 3$, the network's sparsity highlights the sensitivity of MP to network structure, particularly around the well-connected nodes. While MP tends to overestimate GC probabilities in this example, it still captures broad trends and provides useful insights into which nodes are structurally influential. Thus, my choice of MP and algorithms seems well-founded.

Next, looking when $m = 10$, which is what all the following experiments use, we turn to the second row of the above plot. Interestingly, despite a higher connectivity (higher m) and naturally more loops, all three plots demonstrate that the MP approach performs very similarly to the empirical approach. This can be seen in the first plot comparing node-level GC probabilities, where we see all nodes fall on the 45-degree line, indicating that they have the same probabilities. The second plot shows us that the top 20 percent of nodes that are likely to be in the GC differ in MP and empirical probabilities (MP probability minus empirical probability) by -0.0019 (median), suggesting, in terms of probabilities, the MP is actually ever so slightly underestimating GC membership for these top 20 percent nodes. Note, this is very close to zero, and with more trials, I expect it to converge to zero. The last plot, the rank-rank plot, confirms this strong performance with all nodes falling nearly exactly (most ever so slightly higher) on the 45-degree line, which demonstrates a great match in MP and empirical approaches. This suggests that at this network density, MP not only estimates the correct probabilities but also correctly identifies the relative importance of each node, as is crucial for my algorithms. This could make sense as this network has sufficient alternative pathways to reduce correlation between connections, making the network behave more like a tree locally. Nevertheless, strong results here bode well for my next experiments.

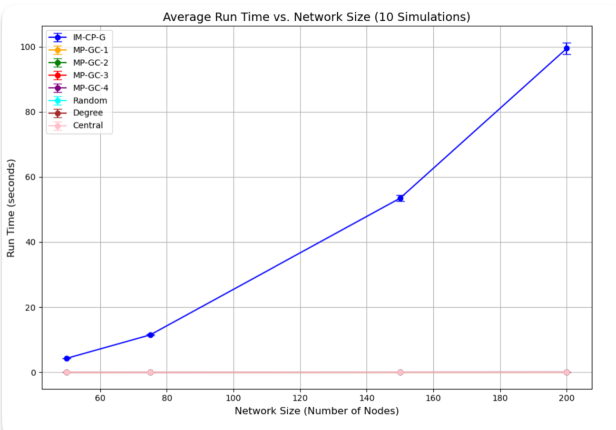
Lastly, for a super large m ($m = 75$), even with our relatively low $p = 0.1$, there are so many connections between nodes that all nodes are in the GC. This is exactly what we see. The first plot shows that all nodes have between a 95 and 100 percent chance of being in the GC for both approaches. The second plot confirms this, where we see the top 20 percent nodes differ by an average of 0 percent between approaches. The results of the rank-rank plot appear slightly more blurry, but they are not. More specifically, for all nodes, the rank for MP appears higher than that of the empirical approach, but this is likely caused by a difference in GC probability of something like 98 percent to 99 percent, causing a drop in rank by many places. That is, the plot shows vertical lines of nodes because all nodes have nearly identical ≈ 1.0 probabilities of being in the GC, making their exact ranking order arbitrary and meaningless despite perfect agreement in the actual probabilities. Thus, once again, we see strong performance from the MP approach.

Thus, overall, these experiments demonstrate the ability of MP approaches to reasonably accurately estimate the probability that a given node is in the GC, which paves the way for my algorithms and results.

Stock BA Network Results

Timing

Looking at the timing of the approaches on the Stock BA network, we see that for both seed counts (3 and 10), the greedy approach is significantly slower than other approaches, with run time increasing quadratically. This is to be expected given the brute force nature reliant upon Monte Carlo simulations. The heuristic approaches run instantly, and slightly surprisingly, it appears my MP approaches do as well. In reality (as I explore in the final results section), they are certainly slower than the heuristic approaches, but by just under a second. This suggests quick convergence within the MP algorithm, which is to be expected given the loops and, as we just saw, high probability of GC existing when $m = 10$ (even for the smallest network of size 50). The main difference between the three and ten seed trials is that for more seeds, the greedy approach takes even longer (scale of y axis), while there is no noticeable increase in time for my MP approaches. See Figure 9 below.



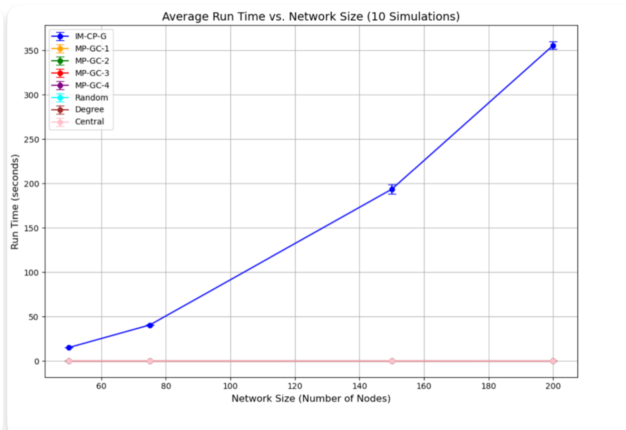


Figure 9. Stock BA Timing Three (Left) and Ten (Right) Seeds

k=3 Optimal Seeds

With three optimal seeds, we see similar and strong performance overall, with influence spreading being about 70 percent across all approaches. The MP approaches have some of the highest mean and median percent of influence spread, with some of the least amount of variability. The greedy approach performs well, and the heuristic approaches also perform surprisingly well, yet have significantly more variability. Looking more closely at the influence spread across each network size, MP-GC-3 and MP-GC-4 (two of my MP approaches) appear particularly strong with minimal variability and strong performance on larger networks. The greedy approach also performs well on larger networks and the random heuristic approach has a significant amount of variability on larger networks. Overall, message-passing approaches achieve similar or better performance, remaining stable and scalable across different network sizes. See Figure 10 below.

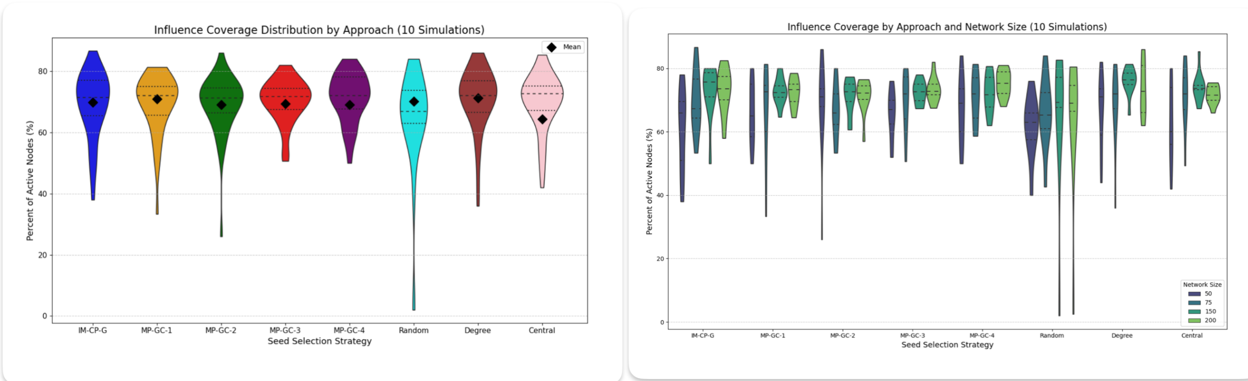


Figure 10. Stock BA with Three Seeds Average Influence Spread (Left) and Network Size Specific Influence Spread (Right)

k=10 Optimal Seeds

With more seeds (10), we see a similar overall influence spread, with it remaining around 70 percent (like we saw with three seeds). This is interesting and is explored more in later sections. The MP approaches perform similarly, with medians slightly lower than the greedy and random approaches. However, MP-GC-2 has one of the highest median performances, along with the heuristic central approach. MP-GC-1 and MP-GC-3 also show particularly strong and consistent performance. Looking at performance by size, we see that the greedy approach and MP-GC-3 have the least amount of variability across network sizes. The heuristic approaches have a large amount of variability, particularly on smaller networks, while the greedy approach performs well. MP-GC-1, MP-GC-2, MP-GC-3 also show a minimal drop-off in performance as the network size increases, which is good and unlike the heuristic approaches. See Figure 11 below.

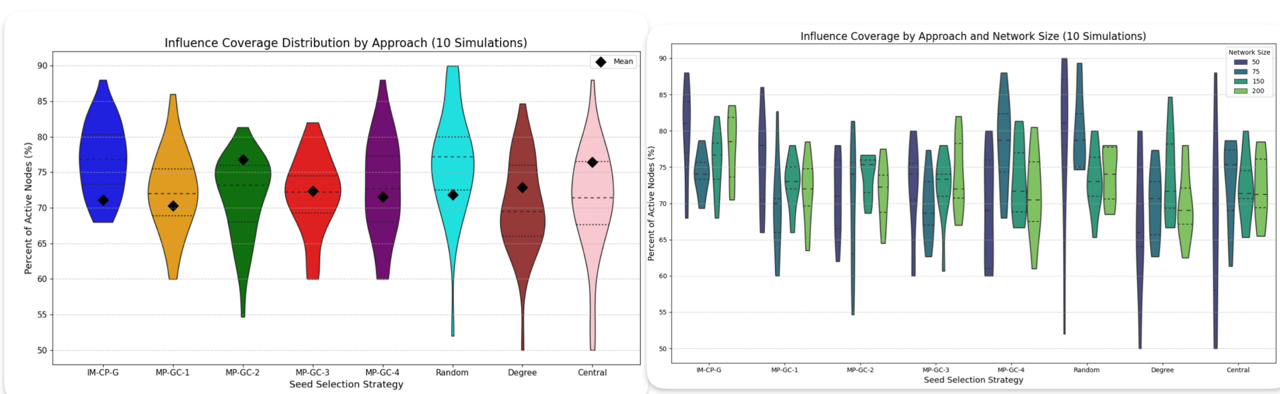


Figure 11. Stock BA with Ten Seeds Average Influence Spread (Left) and Network Size Specific Influence Spread (Right)

Intro to the Custom Clustered BA Networks

Before diving into my custom clustered BA network experiments, I first want to note that the results, particularly the percentage of nodes that get influenced for given seeds, are not directly comparable. In particular, for the first two experiments using a stock BA network with $m = 10$ and varying the number of nodes (n) the rich-get-richer effect takes place on all n nodes. However, my custom clustered BA networks are comprised of BA network clusters of size 25 (and n/25 clusters) where there is a one (or zero for the third network type) percent chance of adding an edge between clusters. Thus, the rich-get-richer effect is isolated to the clusters, which are of constant size. This means there are fewer total edges in the custom clustered BA Networks for a given total number of nodes, and therefore there are fewer pathways (connections). Thus, I would expect both the total spread and the time to compute to be smaller, and this is exactly what I see.

Minimally Connected Clustered BA Network

Timing

The same overall behavior with respect to timing that we saw with the stock BA network also occurs with the minimally connected clustered network. That is, on the Stock BA network, the greedy method's runtime grows sharply with network size, while all MP and heuristic approaches—including my MP-GC variants—remain effectively constant, likely due to fast convergence driven by network density and high GC probability at m = 10. However, as alluded to earlier, the big difference is that there are fewer edges overall, so the greedy approach is quicker (for example, three seeds on the stock graph has a max time of 100 seconds, while here it is 30 seconds). See Figure 12 below.

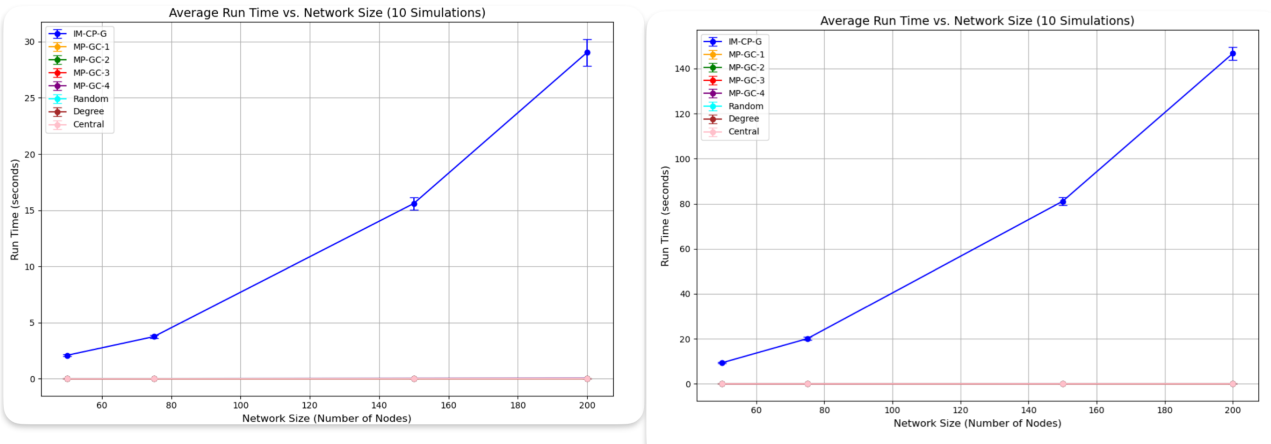


Figure 12. Minimally Connected Clustered BA Timing Across Three (Left) and Ten (Right) Seeds

k=3 Optimal Seeds

For three seeds on the minimally connected clustered BA network, we see a significant amount of variability across all networks. Most importantly, however, we see very strong performance across the MP approaches, with each having among the highest mean and median percent of influence spread. More specifically, MP-GC-2 has both the highest mean and median, which suggests that the sub-graph-based filtering and selection applied work very well on more clustered networks. Similarly, MP-GC-4, which uses the most advanced clustering of the approaches, also performs very well (appearing to be the second best). While the greedy approach has a comparatively high median, it has one of the lowest means. The heuristic approaches have competitive mean performance with lower median performance. Looking more closely at network size-specific behavior, we again see a large amount of variability across models. Also, generally speaking, we see performance fall as network size increases, which makes sense as with 200 nodes of size 25, there are eight minimally connected clusters and only three seeds, so the influence is harder to escape a given cluster. MP-GC-2 also stands out as having some of the best performance on larger networks and having the least pronounced aforementioned drop in performance. See Figure 13 below.

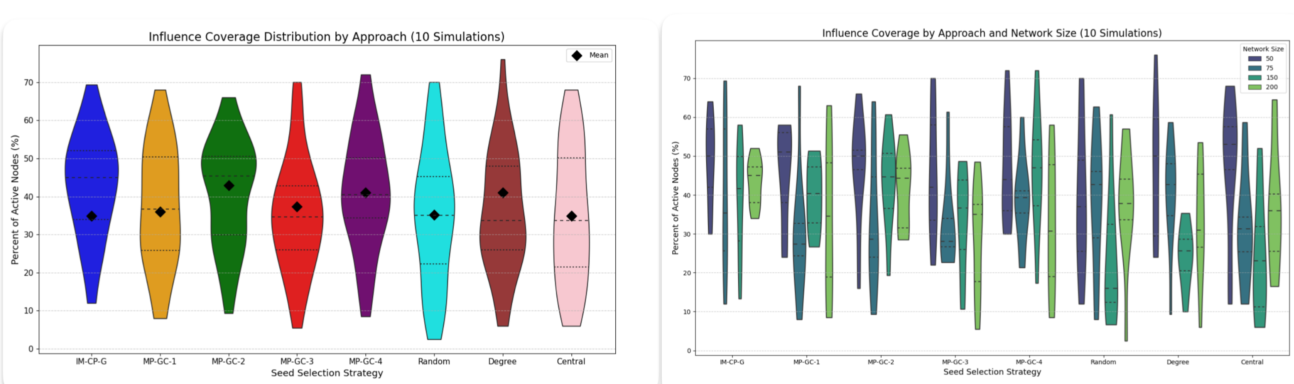


Figure 13. Minimally Connected Clustered BA with Three Seeds Average Influence Spread (Left) and Network Size Specific Influence Spread (Right)

k=10 Optimal Seeds

Now, when we increase the number of seeds to 10, we do in fact see an increase in influence spread (unlike the stock BA networks, which didn't show this). More specifically, with three seeds, a minimally connected clustered BA network had approximately 35 to 45 percent influence spread, while here, with ten seeds, we see between 50 and 60 percent of the nodes get influenced. This makes sense as we have more seeds to distribute among the clusters, better capturing the more isolated communities. Once again, the MP approaches have the highest median influence spread and among the highest mean influence spread (with the heuristic approaches performing quite well in that regard). MP-GC-3 and MP-GC-4 also perform the best (with respect to mean and median), again demonstrating their ability to choose seeds in different clusters and maximize spread. Looking at the network size, we see that MP-GC-3 has the least amount of variability. Furthermore, we again see the trend that, generally speaking, across all approaches, average influence drops as network size increases. So, while we have more seeds than clusters even in the largest case, in smaller networks we can choose more seeds per cluster to maximize spread within the clusters. Once again, this graph also demonstrates the strong and robust performance of the MP approaches on more clustered graphs. See Figure 14 below.

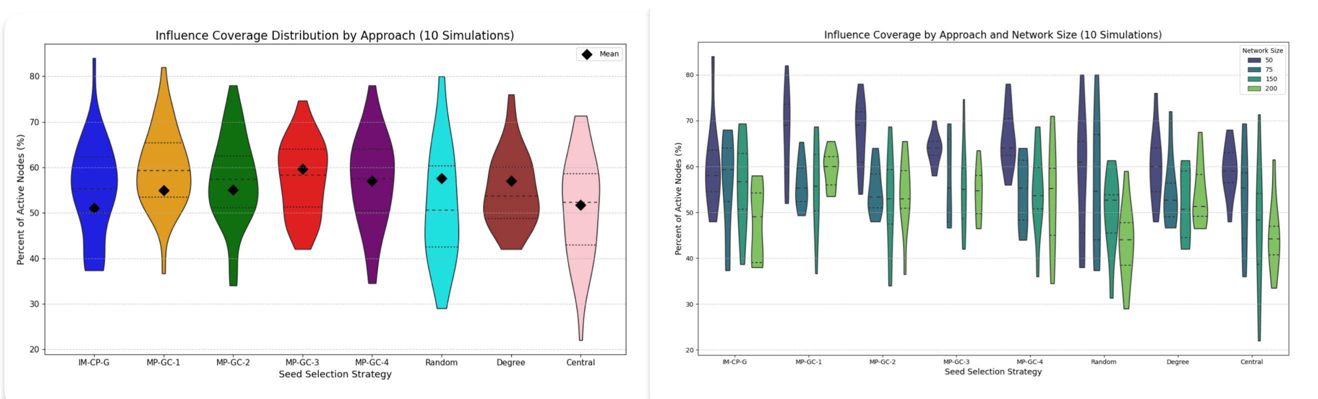


Figure 14. Minimally Connected Clustered BA with Ten Seeds Average Influence Spread (Left) and Network Size Specific Influence Spread (Right)

Disjoint Clustered BA Network

Timing

Lastly, looking at a disjoint clustered BA network, we once again have fewer overall edges, so we expect the greedy approach to be even quicker than it was on the previous two networks — and this is exactly what we see (for three seeds, the max runtime is now 8.5 seconds). This is accompanied by more linear (not quadratic) scaling in run time for the greedy approach. This is due to the disjoint nature of the clusters, which makes influence spread confined to small, isolated components, thereby reducing the cost of each simulation. Like we saw before, all MP approaches and heuristic approaches are nearly instantaneous, again likely due to the relatively large m of 10. See Figure 15 below.

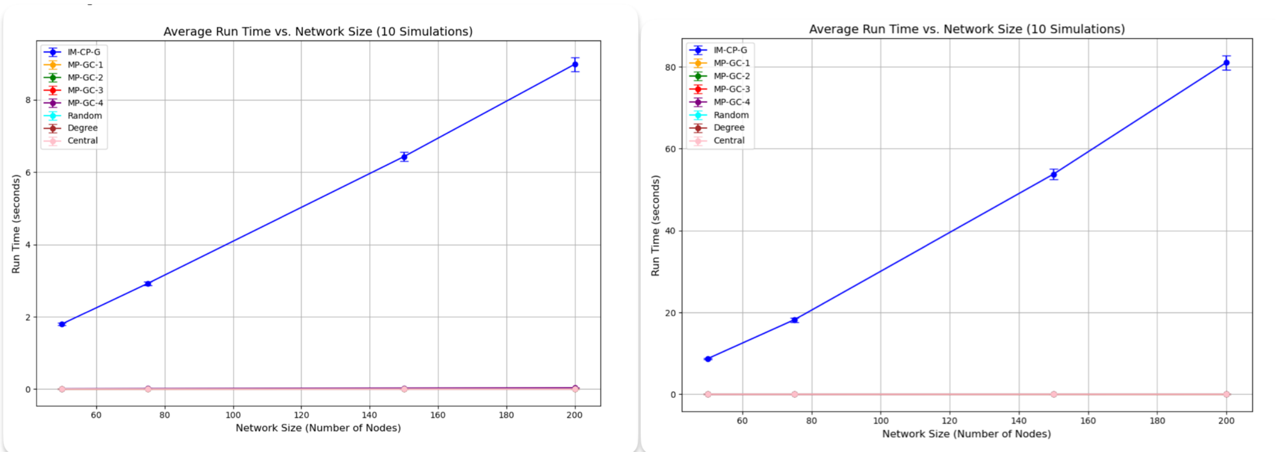


Figure 15. Disjoint Clustered BA Timing Across Three (Left) and Ten (Right) Seeds

k=3 Optimal Seeds

Taking the clustering to the next level, we are now looking for three seeds in networks with completely disconnected clusters. Thus, these clusters are even clearer and well defined, which should suggest strong performance out of MP-GC-3 and MP-GC-4, and this is exactly what we see. More specifically, MP-GC-3 and MP-GC-4 have the highest median influence spread by a significant margin, with MP-GC-3 also having the highest mean influence spread along with MP-GC-2 (again showing its strong performance). The greedy approach performs well but has a lower mean compared to all the MP approaches. The heuristic approaches have among the highest mean influence spread while also having the lowest median influence spread, again showing their lack of robustness. Looking at the size-specific plots, we again see the dominant pattern: that as network size increases, overall influence spread falls. This is once again because as we have larger networks, we have more clusters (more than the three seed nodes). More specifically, for the network of size 200, which has eight disjoint clusters, since we only have three seeds, and the algorithms try to force one seed into each cluster, the performance is bounded as there are no connections between clusters. With that being said, even in the larger networks, the MP approaches have the largest amount of influence spread, again showing their power. See Figure 16 below.

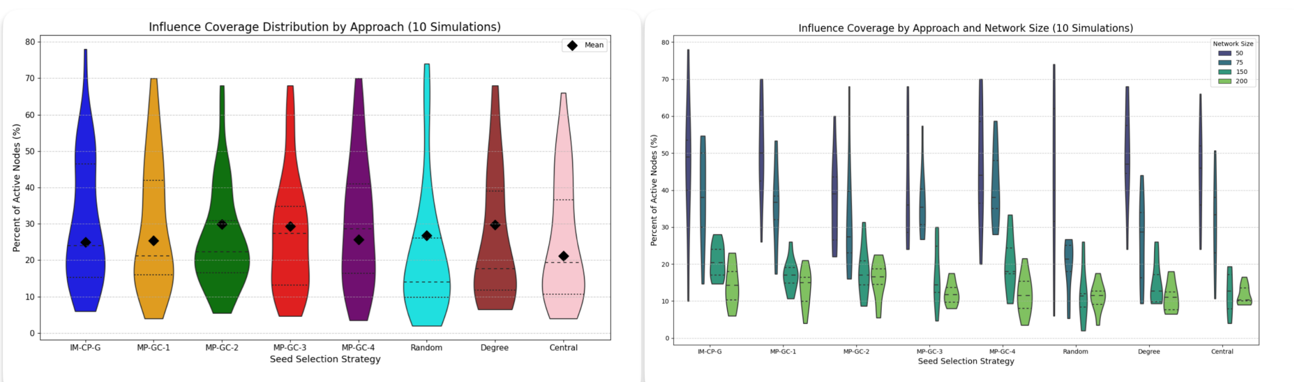


Figure 16. Disjoint Clustered BA with Three Seeds Average Influence Spread (Left) and Network Size Specific Influence Spread (Right)

k=10 Optimal Seeds

When we have ten seeds, however, the total influence spread in the network increases significantly. As previously discussed, with three seeds, as previously discussed total influence spread is capped because in larger networks, there are more clusters than seeds. This is reflected in the results, where for three seeds we saw average influence spread was between 15 and 30 percent, depending on the approach. However, with ten seeds even on larger networks, the number of seeds exceeds the number of clusters. Thus, there is no bound on influence spread, and the average influence spread nearly doubles to between 40 and

55 percent. Once again, the MP approaches are the strongest performers (in terms of mean and median influence spread), with MP-GC-3 and MP-GC-4 having the highest median influence spread. Again, this is to be expected as we have a more clustered network, and the approaches that leverage clustering perform better. Interestingly, the heuristic approaches also have relatively high mean influence spread but also have far more variability. Looking more closely at size-specific plots, we still see average influence fall, but this effect is less pronounced than when we had three seeds. This is likely due to the fact that smaller networks have fewer clusters, so with ten seeds, the approaches can almost guarantee that the individuals in each cluster are influenced. But with ten seeds and eight clusters, ideally six of the eight clusters get one seed (with the other two clusters getting two seeds), and then with a probability of influence being 0.1, clearly not all of the cluster will be influenced. This effect is even less pronounced among the MP approaches, again demonstrating their strong performance. See Figure 17 below.

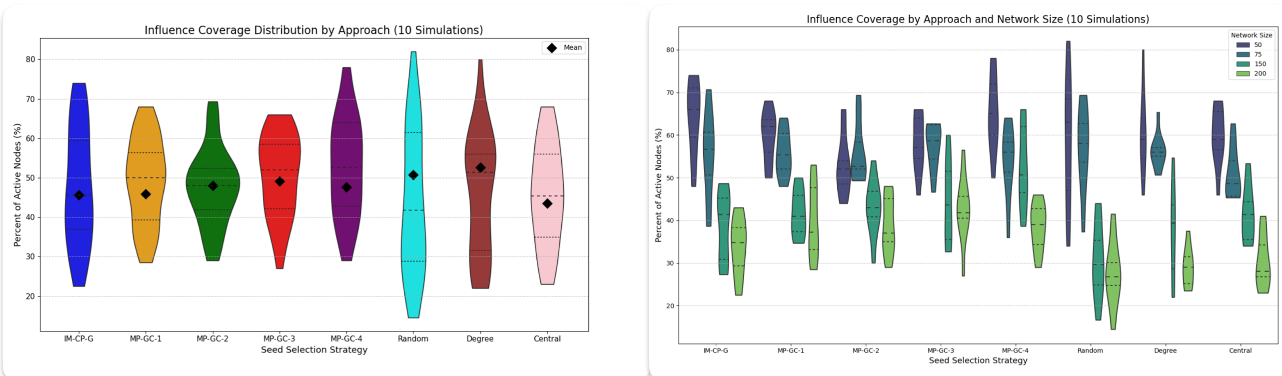


Figure 17. Disjoint Clustered BA with Ten Seeds Average Influence Spread (Left) and Network Size Specific Influence Spread (Right)

Sensitivity Analysis of k (Number of Seeds)

For all three network types with a fixed size of 75 nodes and fixed $p = 0.1$, I now consider the impact on time and influence spread as k increases $1 \to 3 \to 10 \to 25 \to 50$. Across all three network types, the timing performance and trends remain the same (naturally with longer run time on the network types with more edges). More specifically, looking at run time, we see that the greedy approach is the slowest by a significant margin. That being said, unlike previous experiments increasing n, as k increases, the rate of growth of time appears to become more gradual. Once again, both my MP approaches and the heuristic approaches take almost zero time to run with no noticeable increase in time as k increases. If anything, zooming in, my MP approaches are ever so slightly slower than the heuristic approaches (no more than a second slower). This once again shows that, regardless of the base network, the MP approaches are quicker and scale better than the greedy algorithm. They are also incredibly quick, negligibly slower than heuristic approaches. See Figures 18, 19, and 20 below. I now examine the influence spread. Most importantly, as desired and expected, we see that as we increase the number of seeds, across all network types and approaches, the total spread of influence increases. More details are discussed below.

Stock BA

We first see that as k increases, the overall variability of the results drops for all approaches. The greedy approach's performance tends to increase the most as the number of seeds increases, having the most amount of influence spread when k is large. That being said, when we have fewer seeds, as is likely most common in a company setting, my MP approaches perform well, particularly MP-GC-2 and MP-GC-4. Once again, we see surprisingly strong median performance from the heuristic approaches, but their results are more variable. The most important trend, however, is that the rate of increase in spread as we increase k is far more gradual than that of the more clustered networks (as was observed in the earlier results and shown in more detail below). More specifically, with one seed we see about 70 percent influence spread, while with 50 seeds it is over 90 percent — a 20 percent increase for 50x increase in the number of seeds. That is, there is a noticeable but not large increase in influence spread as k increases. This is likely due to the lack of clustering and strong connectivity between all nodes. See Figure 18 below.

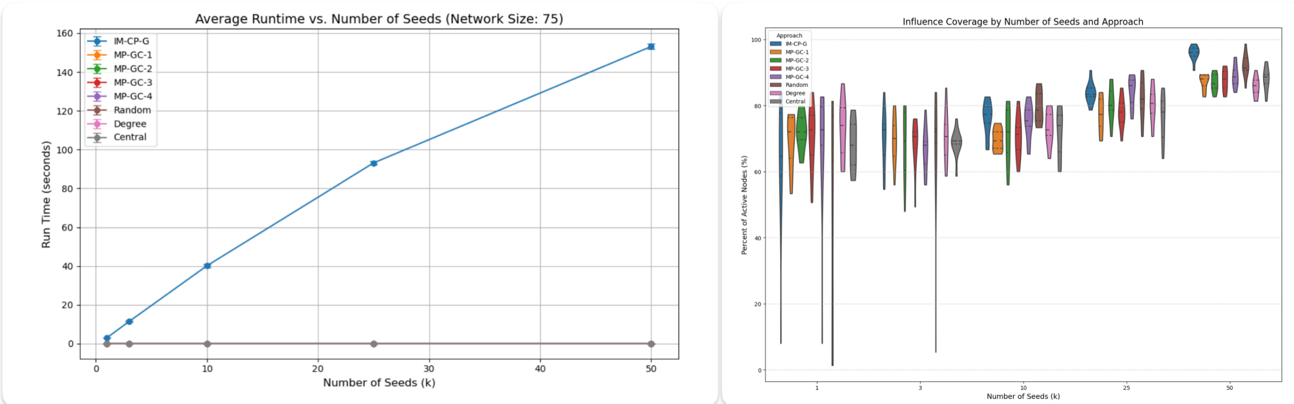


Figure 18. Stock BA Network - Timing (Left) and Influence Spread (Right) Performance as *k* Increases

Minimally Connected Clustered BA

Once again, we see that as k increases, the overall variability of the results drops for all approaches. Furthermore, it is clear that my most sophisticated clustering approach, MP-GC-4, has the strongest median performance (amount of influence spread) over the number of seeds. The other MP approaches perform slightly worse than MP-GC-4, but better than the greedy and heuristic approaches. We also see that the greedy approach performs the best when we have a large number of seeds, but this is, of course, at the cost of run time and scalability. Most importantly, however, is that as we increase k, across all approaches, we see a much steeper increase in influence spread compared to the stock BA networks. More specifically, with one seed we see about 15 to 20 percent influence spread, while with 50 seeds it is over 80 percent. This is due to a lack of frequent connection between clusters, and with more seeds, we gain more access and penetration into clusters. See Figure 19 below.

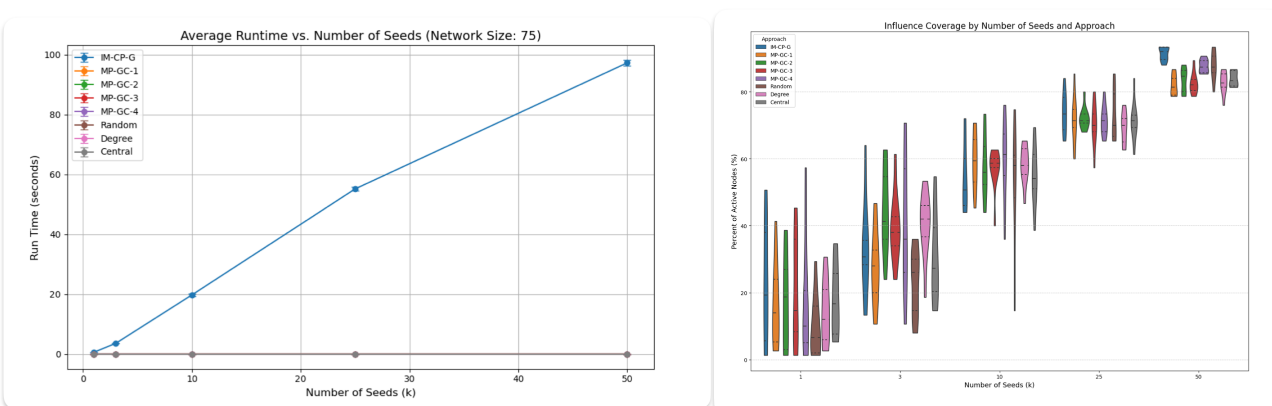


Figure 19. Minimally Connected Clustered BA - Timing (Left) and Influence Spread (Right) Performance as *k* Increases

Disjoint Clustered BA

Once again, we see variability in influence spread decrease as the number of seeds increases. As the clusters are disjoint (3 clusters of size 25), if the algorithms have too few seeds (like when there is one seed), influence spread is minimal, as it can only happen in one cluster. The MP approaches have strong performance overall, often better, but sometimes worse than the greedy approach, which once again performs very well with this combination of network size and probability of influence (p). Most importantly, as we increase k, across all approaches, we see the steepest increase in influence spread. More specifically, with one seed we see about five to ten percent influence spread, while with 50 seeds it is over 80 percent. This is due to a lack of any connection between clusters, and with more seeds the algorithms can guarantee all clusters have at least one seed so that we can influence that specific cluster of nodes. See Figure 20 below.

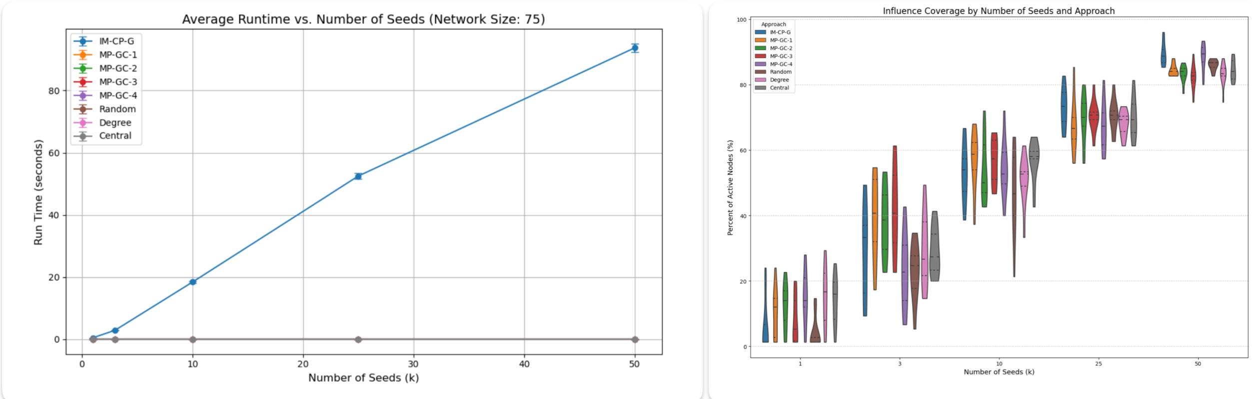


Figure 20. Disjoint Clustered BA - Timing (Left) and Influence Spread (Right) Performance as k Increases

Sensitivity Analysis of p (Probability of Influencing a Neighbor)

For all three network types with a fixed size of 75 nodes and fixed $k = 3$, I now consider the impact on time and influence spread as p increases $0.05 \rightarrow 0.1 \rightarrow 0.3 \rightarrow 0.6 \rightarrow 0.8$. Across all three network types, the timing performance and trends remain the same (naturally with longer run time on the network types with more edges), so I will discuss it all together here. Like we saw when we increased k, my MP approaches and the heuristic approaches run nearly instantaneously. Zooming in, the MP approaches appear negligibly slower, but this is almost unnoticeable without looking more closely at the code. The greedy approach has the slowest run time across all network types and p values. Yet, unlike before, after the sharp increase, runtime plateaus as p increases with the greedy algorithm. As we will see next, this is because for large p values, 100 percent influence spread is achieved. Furthermore, across all network types and approaches, influence spread behaves the same as p increases. More specifically, as activation probability increases, all approaches converge to near 100 percent total influence spread. At lower probabilities (0.03–0.1), the greedy algorithm and my MP approaches have more consistent performance, both in terms of median and variability in spread (variability is particularly high for the random heuristic approach on the disjoint cluster graphs). See Figures 21, 22, and 23 below for network-specific timing and influence spread performance as p increases.

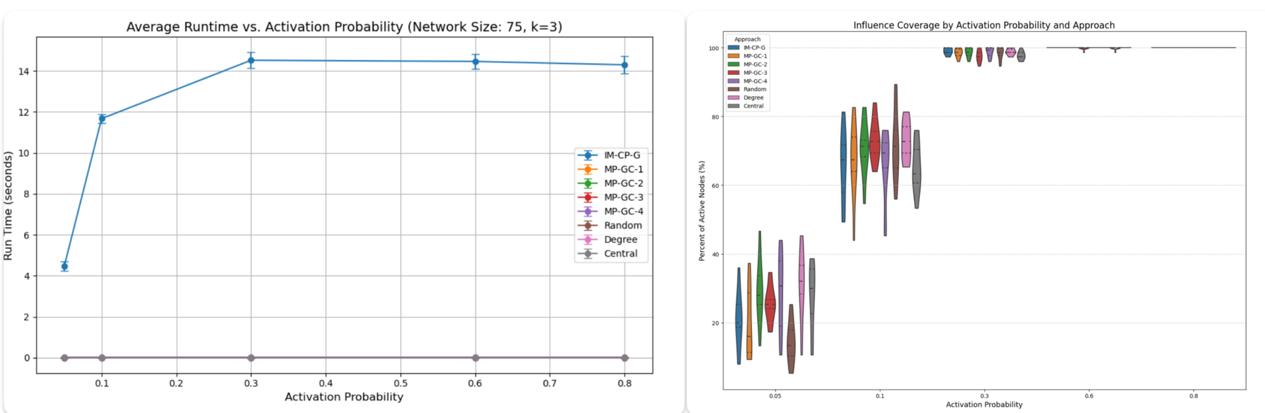


Figure 21. Stock BA Network - Timing (Left) and Influence Spread (Right) Performance as p Increases

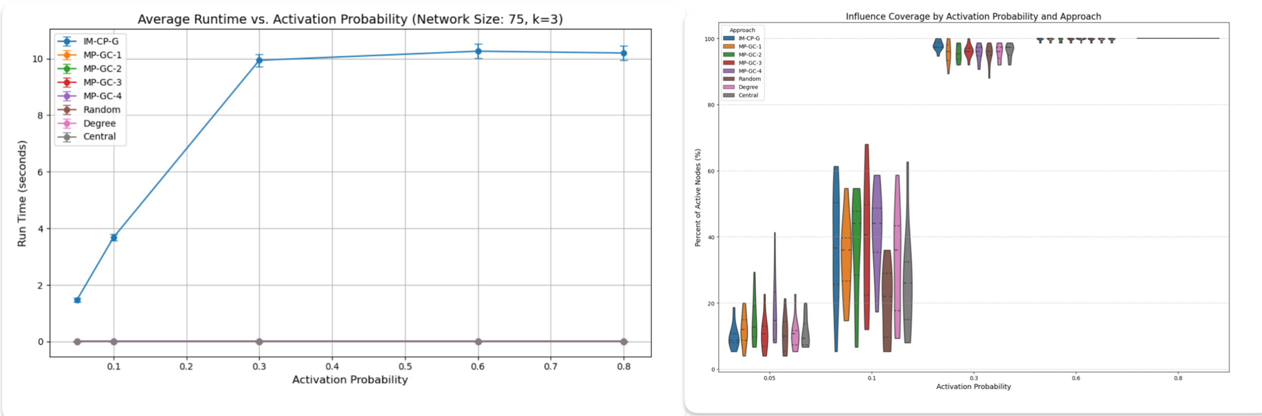


Figure 22. Minimally Connected Clustered BA - Timing (Left) and Influence Spread (Right) Performance as p Increases

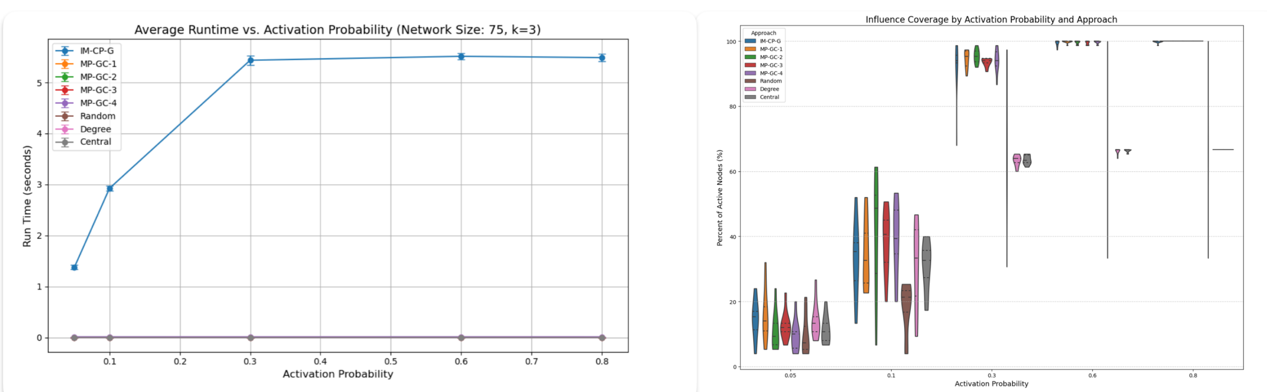


Figure 23. Disjoint Clustered BA - Timing (Left) and Influence Spread (Right) Performance as p Increases

A Further Note on Timing

All the experiments have shown that my MP approaches are near instantaneous and often appear as quick as the heuristic approaches. So, I conducted a simple experiment to demonstrate that, in fact, the MP approaches are slightly slower. More specifically, over 100 trials with $k = 10$ seeds, a stock BA network with $m = 10$ of size 100, and $p = 0.1$, I compared the run time of my MP approaches to the heuristic approaches. See Figure 24 below. We see, as expected, that the MP approaches are slower than heuristic methods (Random, Degree, and Central) by about 0.01 seconds (a very small but noticeable amount when viewed this way). MP-GC-4 is the slowest, which makes sense as it uses the most advanced clustering.

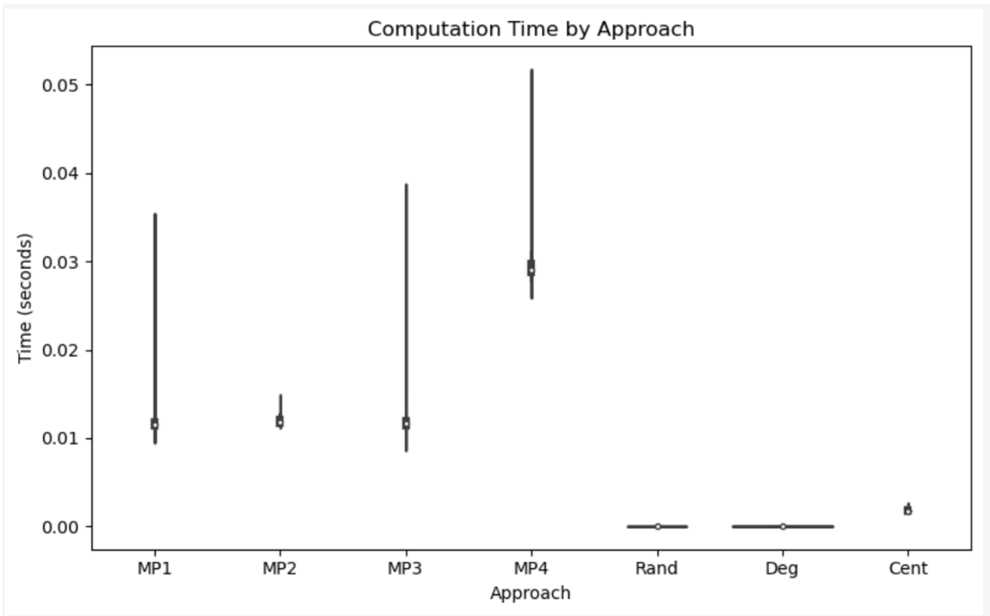


Figure 24. Timing Comparison Between MP and Heuristic Approaches

4 Discussion and Conclusion

In this work, I develop, implement, and test four message passing-based approaches (MP-GC-1, MP-GC-2, MP-GC-3, and MP-GC-4) for influence maximization using the Independent Cascade framework. While message passing is known to perform perfectly on networks with no loops, I show that when run on networks with loops (like all BA networks used in my experiments), it is still surprisingly accurate when determining Giant Component membership[4]. Thus, it provides a reasonable starting point for my algorithms. Most importantly, I show that my MP approaches exhibited significantly better scalability and runtime performance compared to the traditional greedy method and heuristic methods, while having comparable or better ability to choose optimal starting seeds that maximize influence spread. More specifically, while the greedy algorithm's runtime increased steeply with network size and number of seeds, the MP approaches run marginally slower than the heuristic approaches, but still incredibly quickly. The MP approaches also showed similar or better ability to choose optimal seeds when compared to both the greedy and heuristic approaches across various network types and parameter values. My three MP approaches that integrate clustering algorithms (MP-GC-3 and MP-GC-4) show robust and superior performance on clustered BA networks. This highlights their capability to effectively distribute seed nodes across different, potentially isolated, network regions as desired. Through sensitivity analysis of core parameters like the number of seeds (k) and influence probability (p), the message passing approaches also maintained consistent and strong performance across varying parameter values. Thus, these results clearly highlight that message passing is an efficient and effective tool used to tackle the notoriously difficult influence maximization problem.

There are a plethora of future avenues that could further leverage message passing. Future works could integrate loop-aware message passing, such as the frameworks proposed by Weis, which likely trades some run time performance for more optimal seed selection[5]. Additionally, the strong results of subgraph and clustering-based approaches (MP-GC-2, MP-GC-3, and MP-GC-4) suggest that future work should continue to leverage knowledge about the target network's structure to avoid redundant seeding and maximize spread across communities. Overall, I show that MP approaches can bridge the gap between simulation-heavy greedy algorithms and heuristic-based approaches in terms of run time and scalability, all while being very effective at choosing optimal seeds.

References

1. Kempe, D., Kleinberg, J. & Tardos, É. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 137–146 (2003).

2. Ye, Y., Chen, Y. & Han, W. Influence maximization in social networks: Theories, methods and challenges. *Array* **16**, 100264 (2022).

3. Newman, M. Message passing methods on complex networks. *Proc. Royal Soc. A* **479**, 20220774 (2023).

4. Allard, A. & Hébert-Dufresne, L. On the accuracy of message-passing approaches to percolation in complex networks. *arXiv preprint arXiv:1906.10377* (2019).

5. Weis, E. *Robust Interventions in Network Epidemiology*. Ph.d. dissertation, University of Vermont (2024). Graduate College Dissertations and Theses, No. 1813.

Acknowledgments

Thank you for a great semester, Professor Hébert-Dufresne!

Additional information

Code availability statement: Uploaded to Brightspace. Can certainly post on GitHub if that is what you prefer.

Files: Write-up (this file, named “Danny-Satterthwaite-MOCSII-Final-Write-Up”), “MOCS2-Final-Proj-Main-Code”, which is where all the experiments, set-up, visuals, etc. are created. The “Additional-MP-Testing” file is an extra file that uses the same message passing approaches on various other graphs to gut-check the code implementation of the message passing algorithm.