Introduction

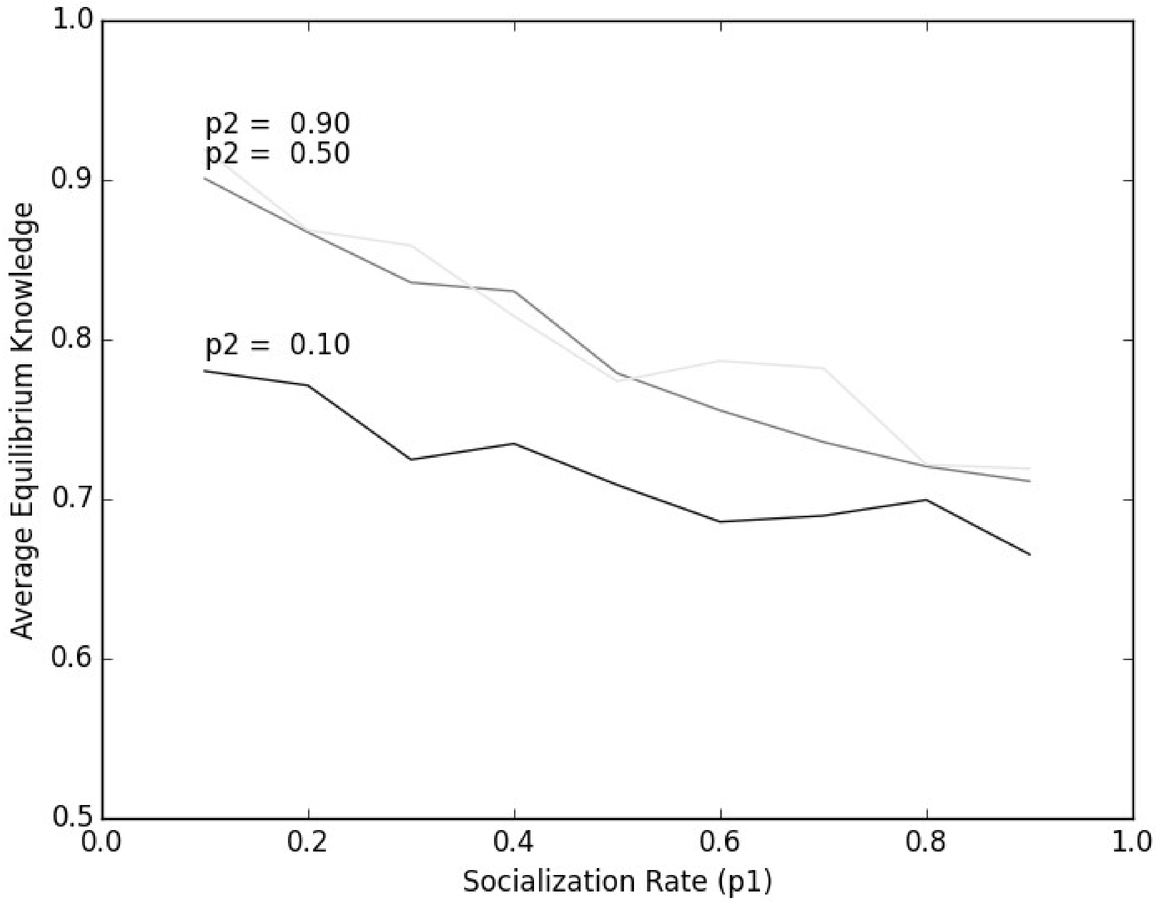
We start off this paper by looking at James March's paper[1] exploring the topic of exploration vs. exploitation. This paper compares and contrasts how a system or organization can achieve optimal knowledge through exploration (experimentation) or through exploitation (learning from experience). The simulation that March used is reproduced here and the results of interest are shown to be similar to March's results.

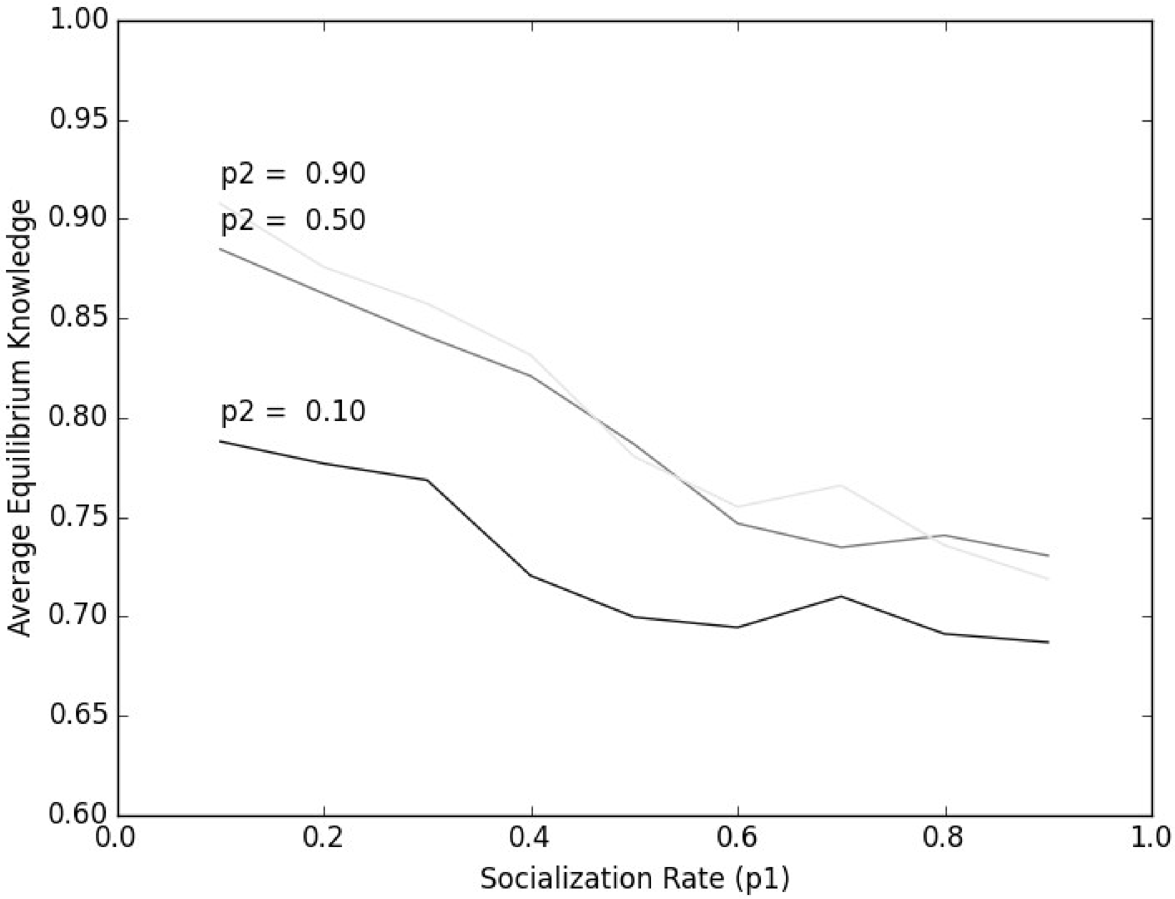
Next we look at a much newer paper that builds on the March idea. This paper is the Fang, Lee, and Schilling paper that explores the role of structural design of an organization and how it impacts learning. Once again the simulations that are used are reproduced here and the data is graphed to show that the results are very similar.

Exploration vs Exploitation

The purpose of this simulation is to find a balance between between exploration and exploitation in an organization. For starters a reality of m dimensions is generated with each dimension having equal probability of being -1 or 1. This simulation does not take into account the structure of the organization instead relying on the agents with a lower payoff learning from a code of beliefs.
The code itself is directed by the agents with a higher payoff then the code. The payoff function is simply defined by how close an agent is to reality. Each agent is assigned set of beliefs which is an array of m dimensions. Each dimension can have a value 1, 0, -1. The beliefs are assigned randomly with equal probability. The learning from the code (socialization) takes place with probability p1, while the code itself learns with probability p2.

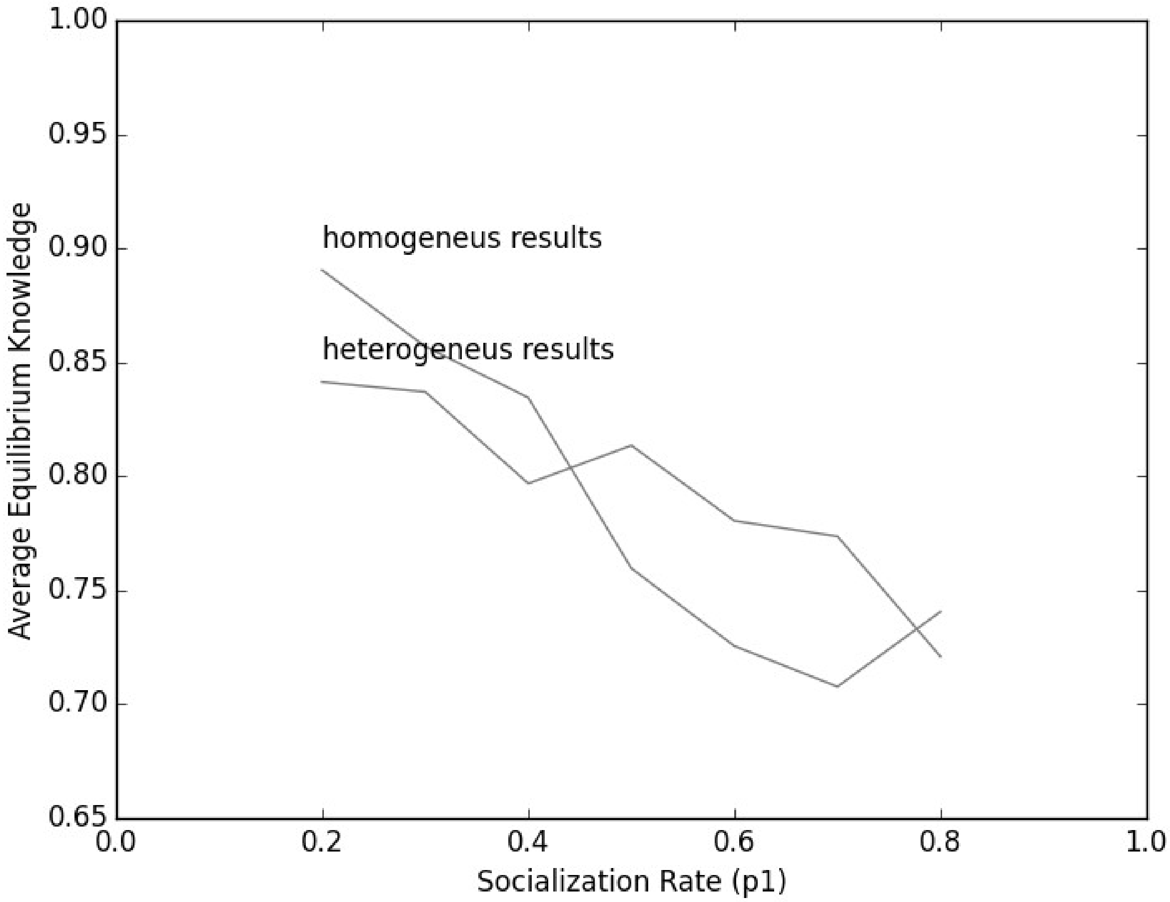
The following two graphs show runs with 25 and 50.


p2 = 0.90
p2 = 0.50
p2 = 0.10
Average Equilibrium Knowledge
Socialization Rate (p1)
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.2
0.4
0.6
0.8
1.0

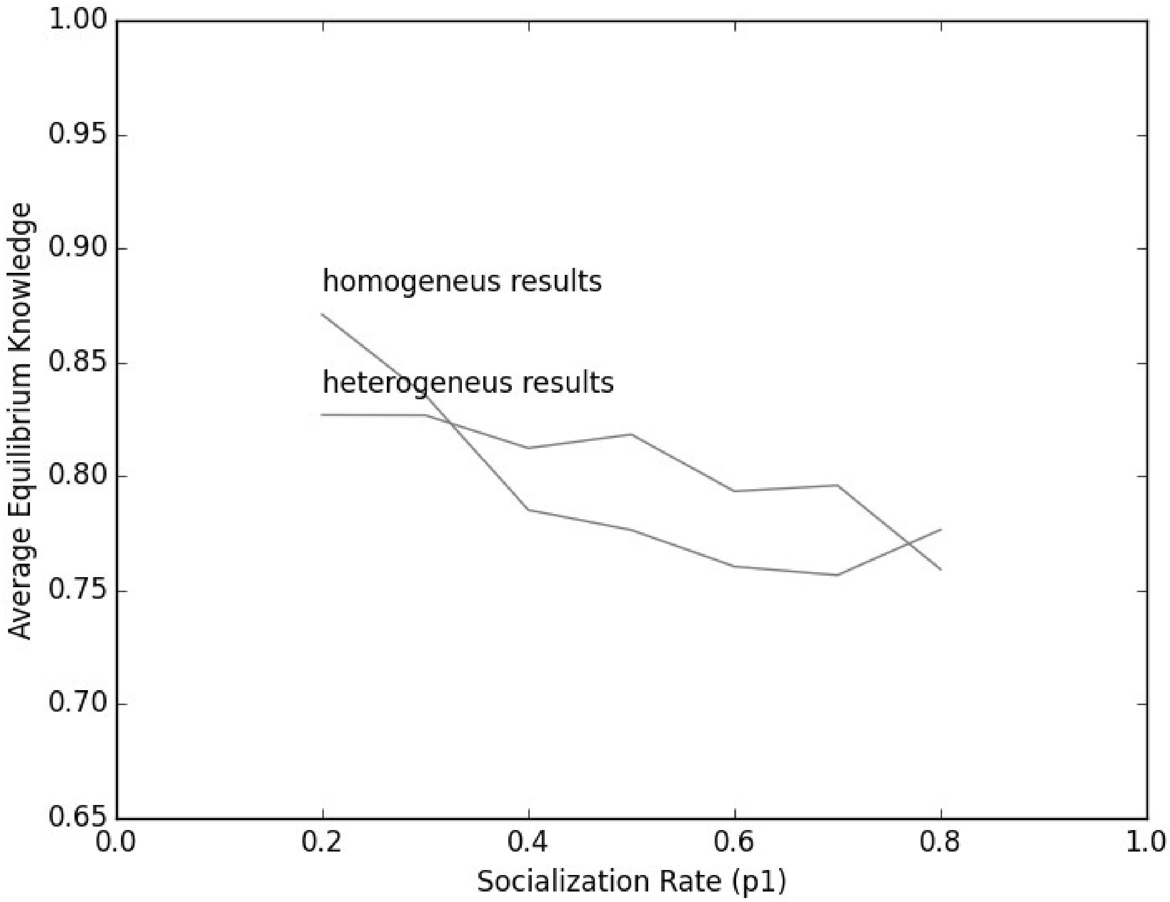


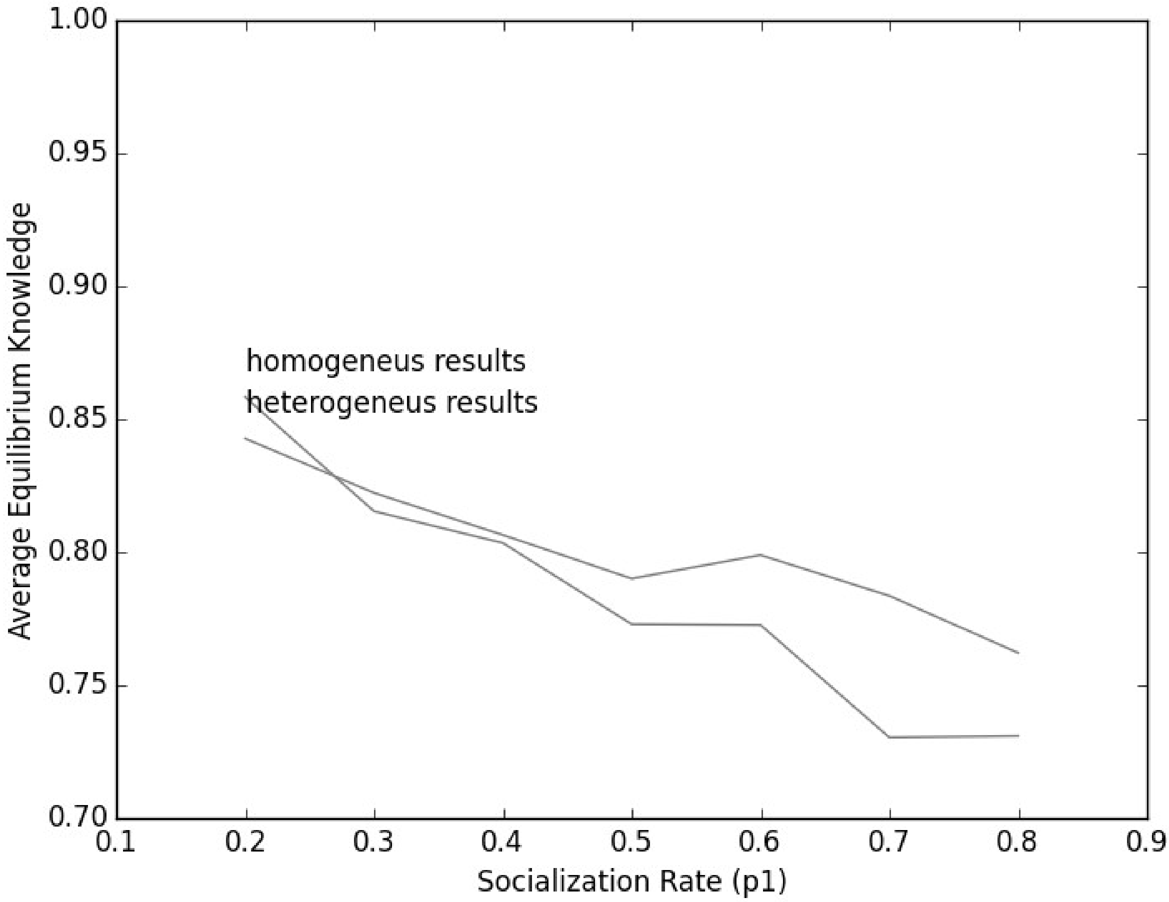
By running with more iterations the data starts to show the expected value after any given run. This also shows a close match to March's results and allows me to draw the same conclusion. Slower adaptation of the code allows for more diversity which in turn allows the code to grow closer to reality.

The second part of the March experiment that I reproduced was homogeneity vs heterogeneity. This extends the model to allow each agent to have their own learning rate. In this case we assign p1 (probability of learning from the code) to be either 0.1 or 0.9. we assign 0.9 with probability pHL and 0.1 with probability 1 - pHL (heterogeneous learning). This is then compared to a population in which every one learns with the same probability. (homogeneous learning). The following graphs are the results of my simulation. As like before the first graph shows only 10 runs, which is not enough to have high confidence in the expected values. The second graph shows 25 runs and the third graph shows 50 runs.



1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.0
0.2
0.4
0.6
0.8
1.0
Average Equilibrium Knowledge
Socialization Rate (p1)
homogeneus results
heterogeneus results

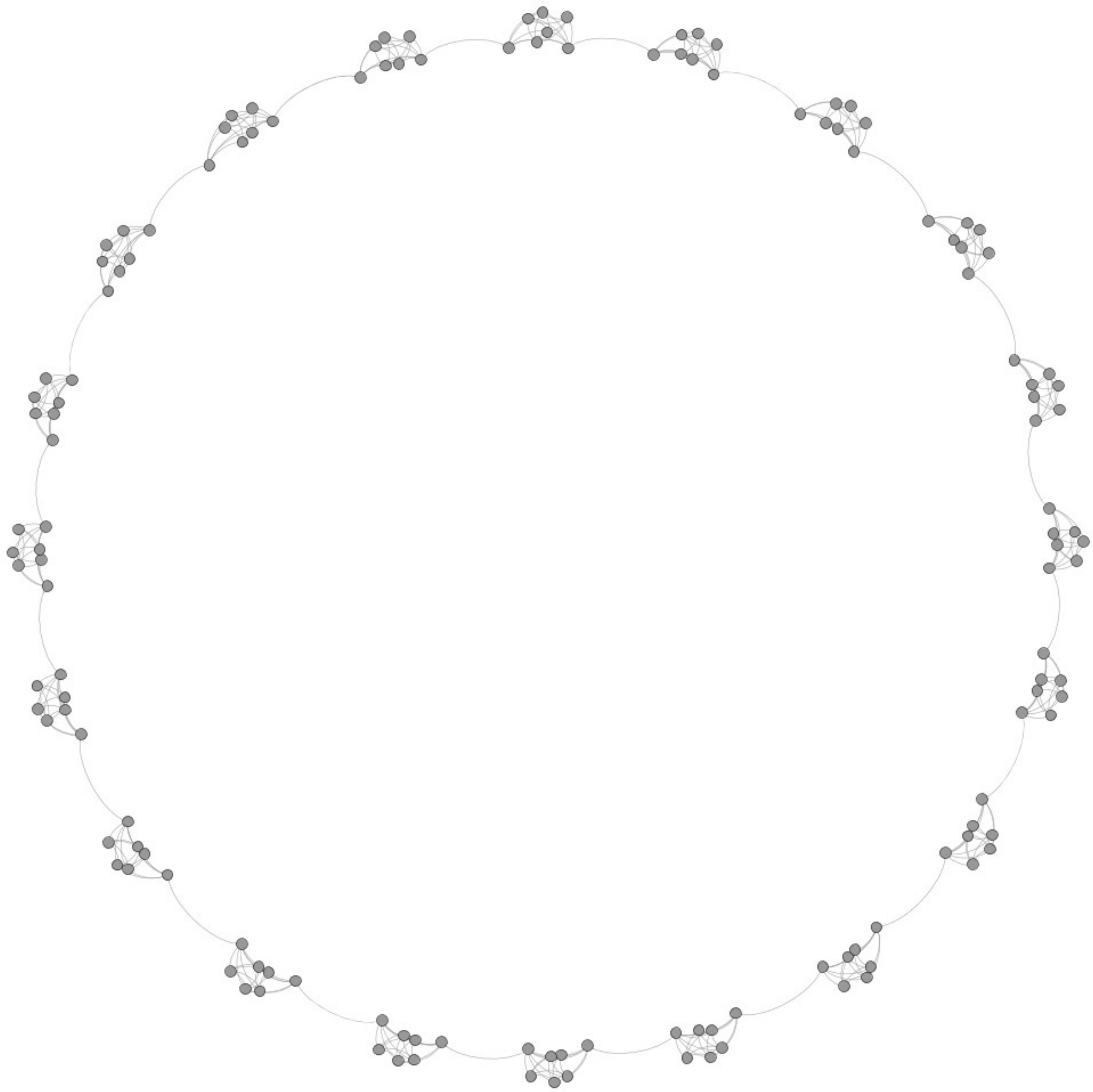

homogeneus results
heterogeneus results
Average Equilibrium Knowledge
Socialization Rate (p1)
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.0
0.2
0.4
0.6
0.8
1.0



Once again the results get closer to March's the more simulations I run. The graphs are showing an average of heterogeneity learning vs homogeneity learning at that same rate. In the case here we are showing that heterogeneity produces higher results. However unlike March my simulations showed that a lower homogeneous rate as being higher then a lower average heterogeneous rate.

Effect of organizational structure on learning

The paper by Fang [2] extended the March idea by looking at how network structure affects learning in a network. This is an actual graph generated for the simulation as displayed by gephi.



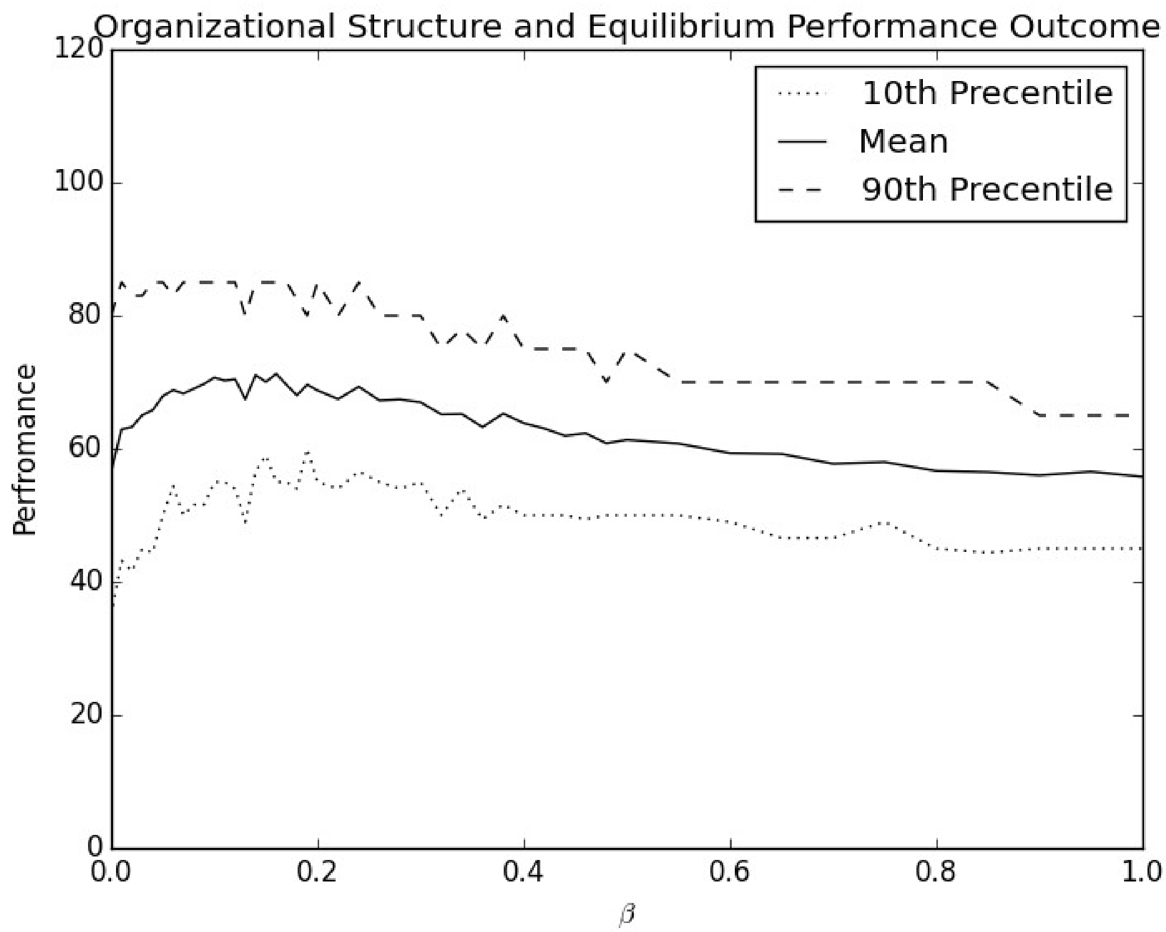
I did not expand all the subgroups out, but the one I did shows what all the subgroups look like and how they are all connected. Just like the March paper each agent is given a set of beliefs -1,0,1 generated with equal probability. One major difference here though is that there is no code to learn from. Everyone learns from agents connected to them that have a higher payoff.

Another big difference is the payoff function itself. Rather then just compare each dimension. The comparison is made in groups by a complexity factor between 1-10. A complexity factor of 1 is simply the March payoff function. But a payoff of 5 means that an agent has to match reality for 5 dimensions or they get 0 for that group of 5. What is not captured is which dimensions an agent is wrong. This models not knowing how close you are to solving a problem.

A new addition to the simulations is the disimilarity index. This is simply a pairwise comparison between every vertice. This gives us an idea of the diversity in a graph at any given time. This can be defined:

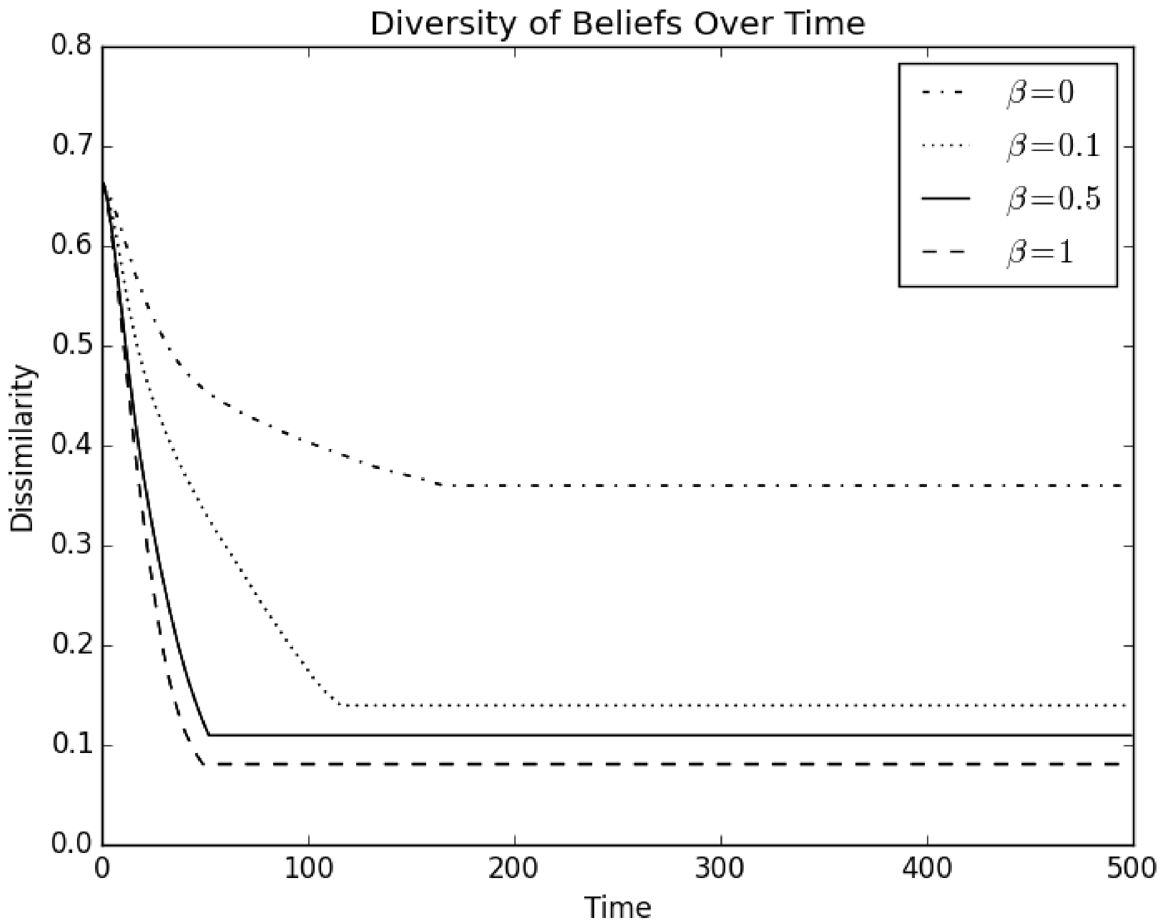
$$Dissimilarity = \frac{2}{mn(n-1)} \sum_{i=1}^{(1/2)n(n-1)} \sum_{j=1}^{n} \omega_{ij}$$ (w being the vertices's between I and j)

The first simulation ran the caveman graph with a rewiring probability between 0 (original graph with intact subgroups) and 1 (complete integration of the subgroups). This simulates connections between the isolated groups. This is running the simulations 200 times as suggested in the Fang paper. Complexity was kept at 5 and probability of learning was kept at 0.3.



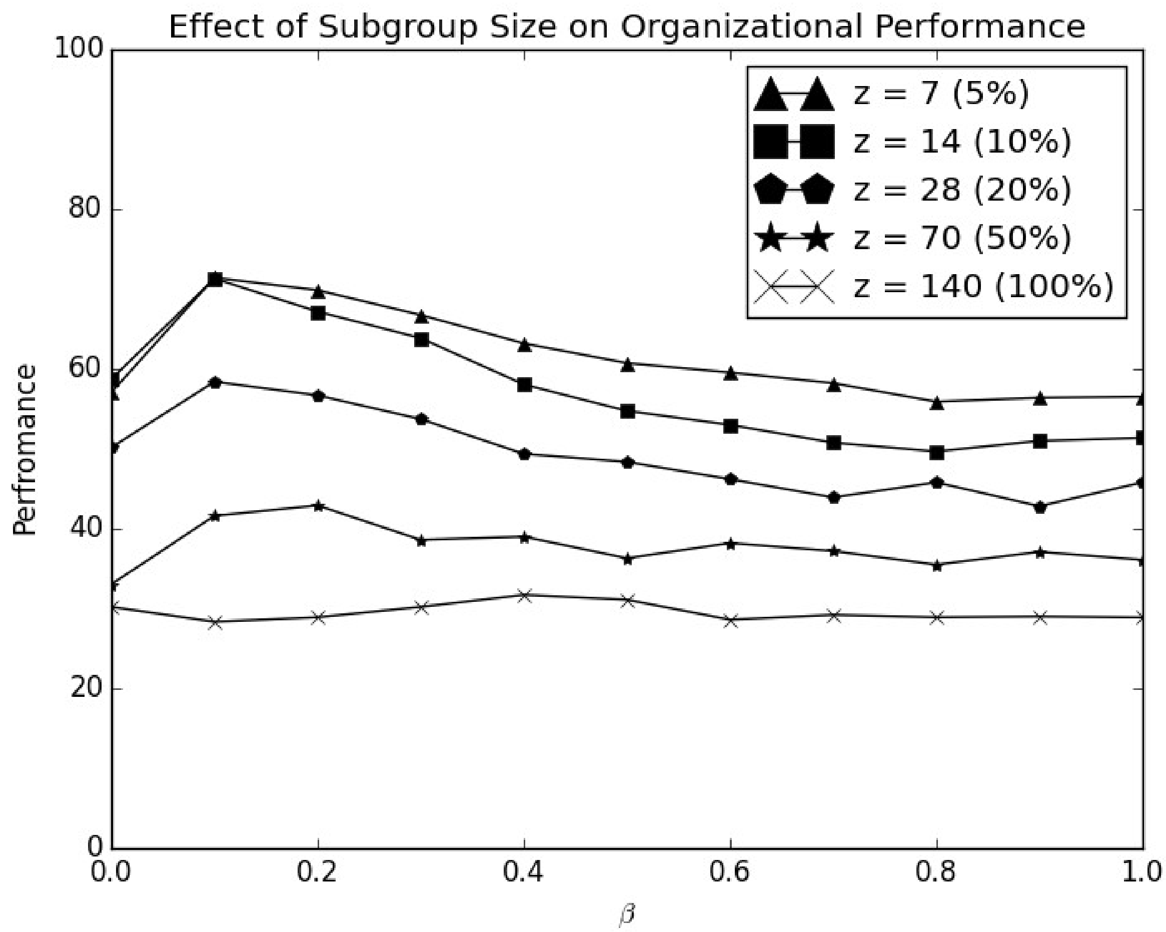
The results are quite similar with my results being slightly lower. But the graphs are showing the same optimal structure of ~ 0.12 rewiring probability.

The next experiment showed the dissimilarity of beliefs as defined above. I ran this over the probability of rewiring (Beta) 0, 0.1, 0.5, 1.



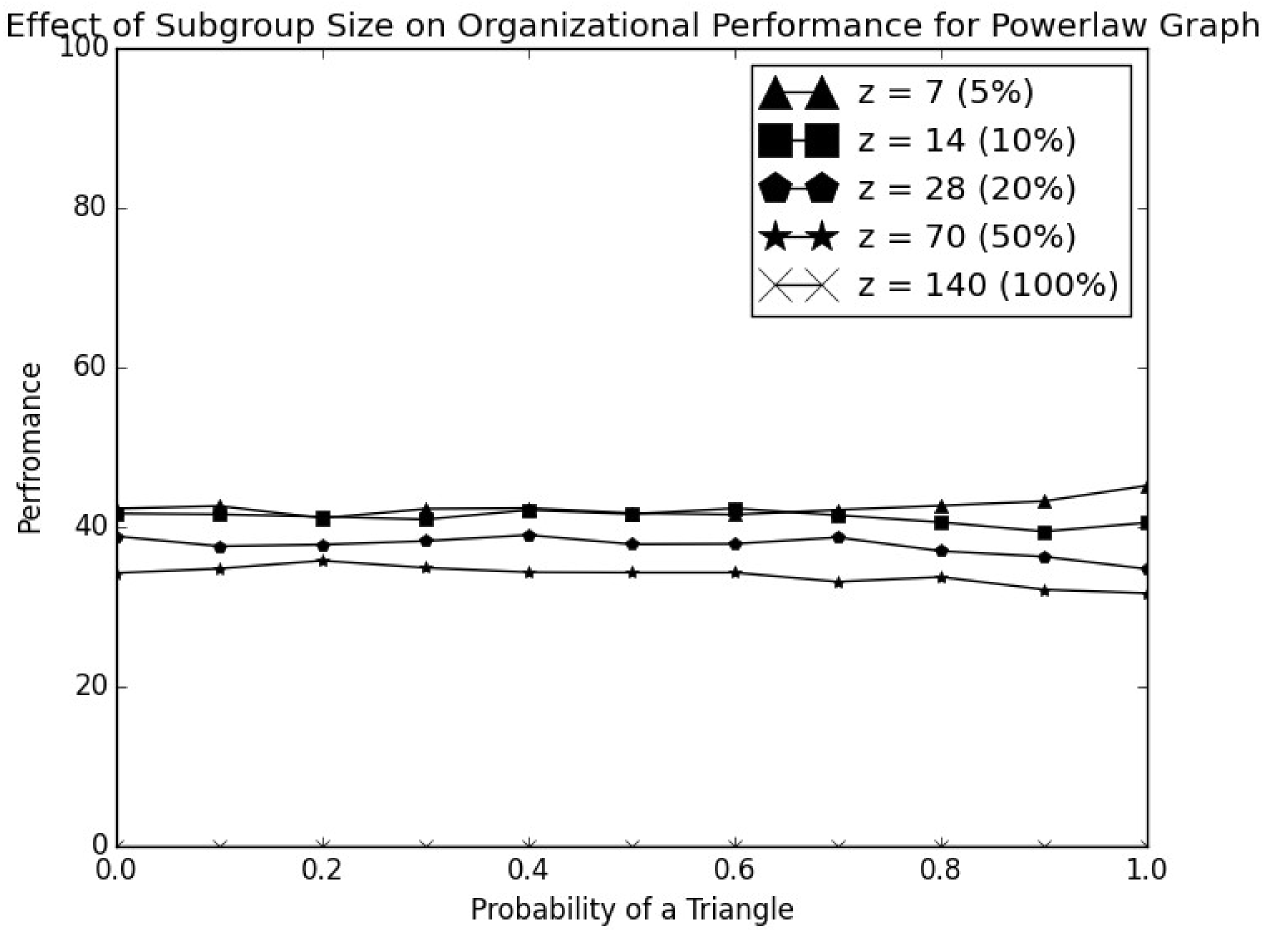
Again I achieve very similar results to the Fang paper, showing how an increase in Beta makes convergence quicker and thus diversity is lost faster when Beta is higher.

The next experiment looks at subgroup size, 7, 14, 28, 70, 140 over Beta values 0 – 1.

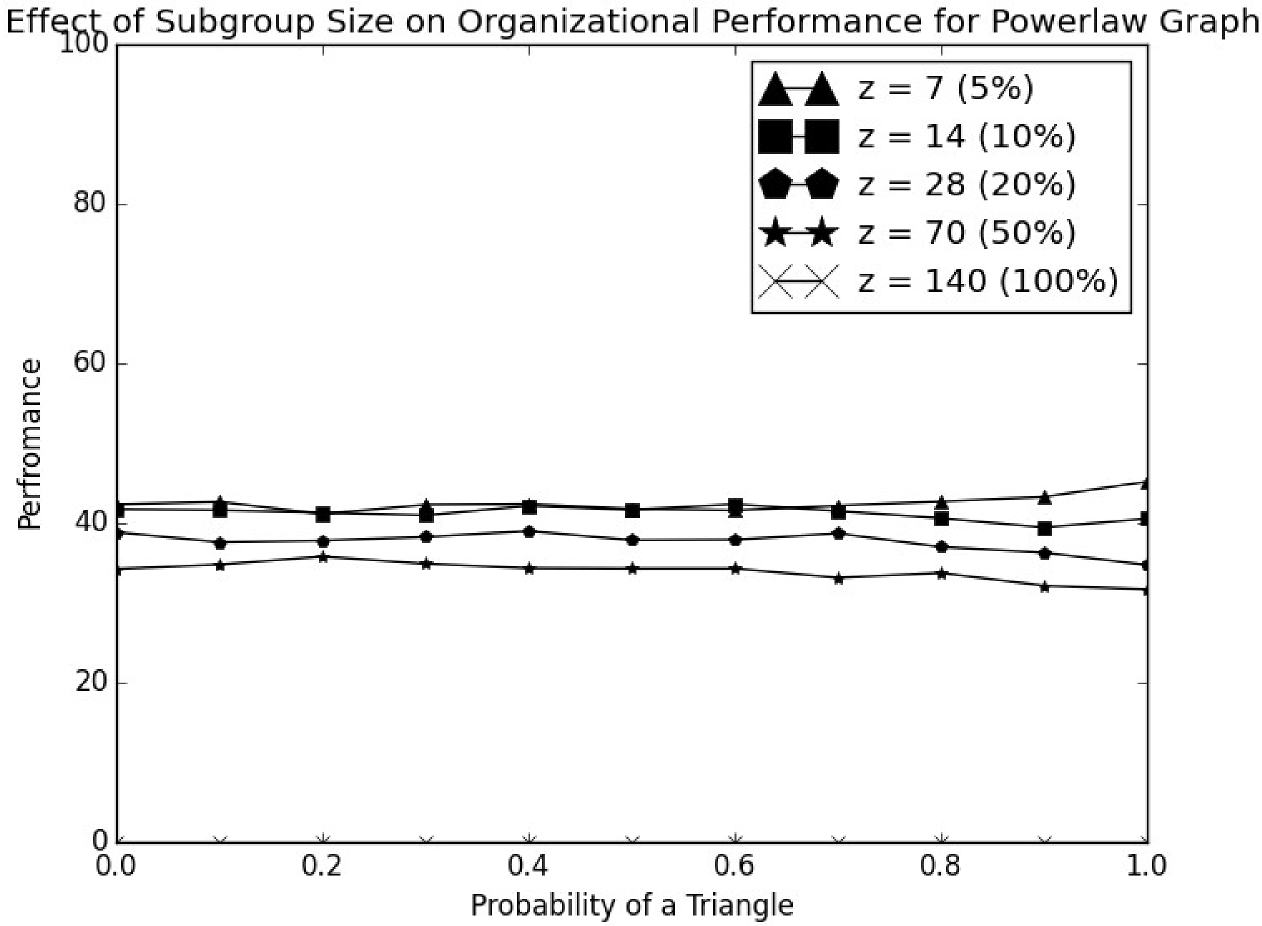


Effect of Subgroup Size on Organizational Performance
Perfromance
β
z = 7 (5%)
z = 14 (10%)
z = 28 (20%)
z = 70 (50%)
z = 140 (100%)
0
20
40
60
80
100
0.0
0.2
0.4
0.6
0.8
1.0

Again my results are very similar to the Feng paper. Once again they show that diversity is the key. A low subgroup size and low beta give us increased isolation and diversity and higher performance. This next graph shows what happened when I ran a powerlaw graph over the same simulation. This software has been designed to take any kind of graph structure. This is where I intend to take the research exploring a variety of graph structures.

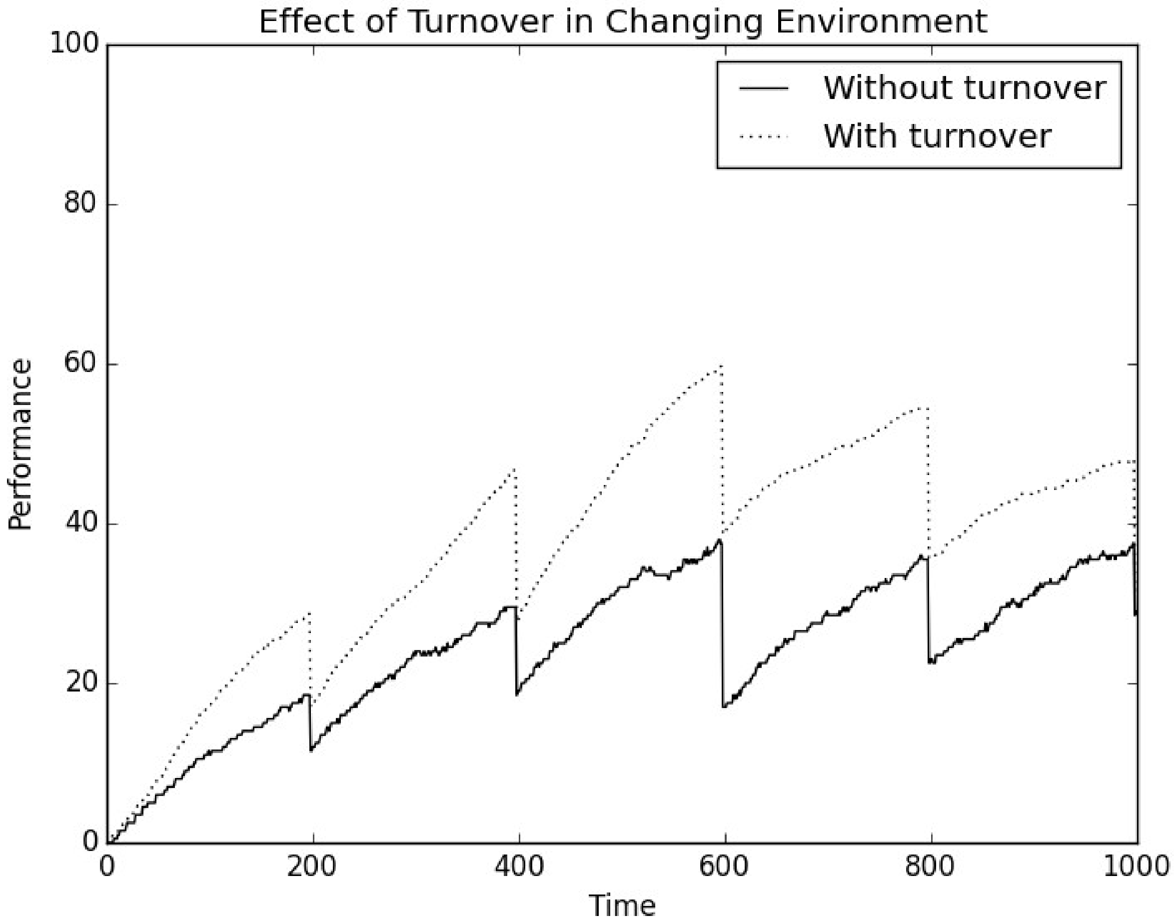


Powerlaw graphs are randomly generated preferential attachment graphs. The more connections you have the higher your probability is of having more. These mimic many kind of social networks so I thought they would be a good place to start my expansion. Powerlaw graphs also allow you to set the probability of triangle being created, thus increasing the probability of cliques or isolated subgroups.



However as this graph shows I appear to have been wrong. The performance is much worse then the caveman graph (strategically formed) and the probability of triangles seems to have no affect.

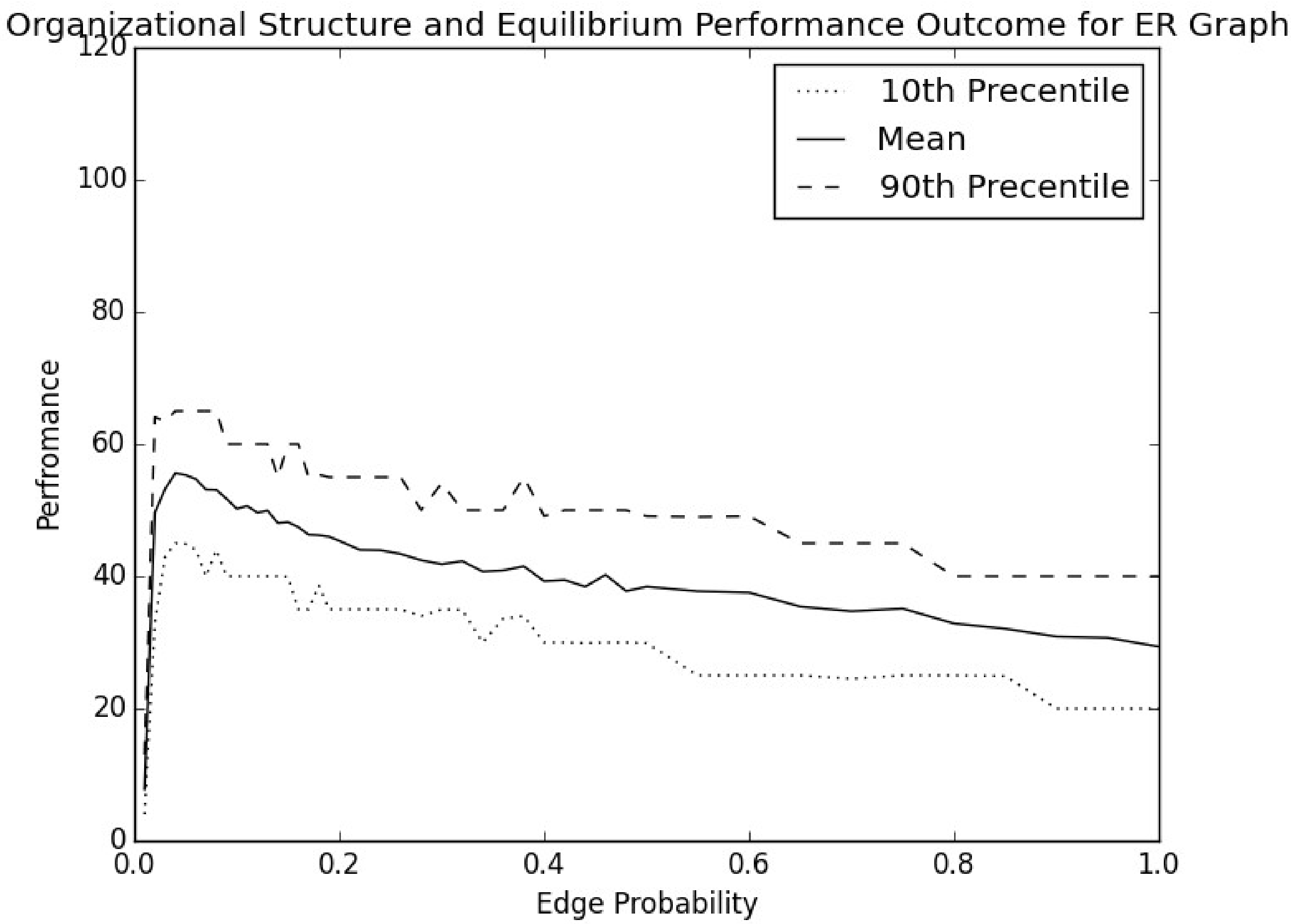
Finally I looked at the turnover and reality changes. In these simulations at every tick there was a 1% chance any given node would 'leave' and be replaced by a newly generated set of beliefs. Also every 200 ticks there was a 10% chance a given dimension in reality would flip its value. Thus mimicking the moving targets we all seem to be trying to hit these days.



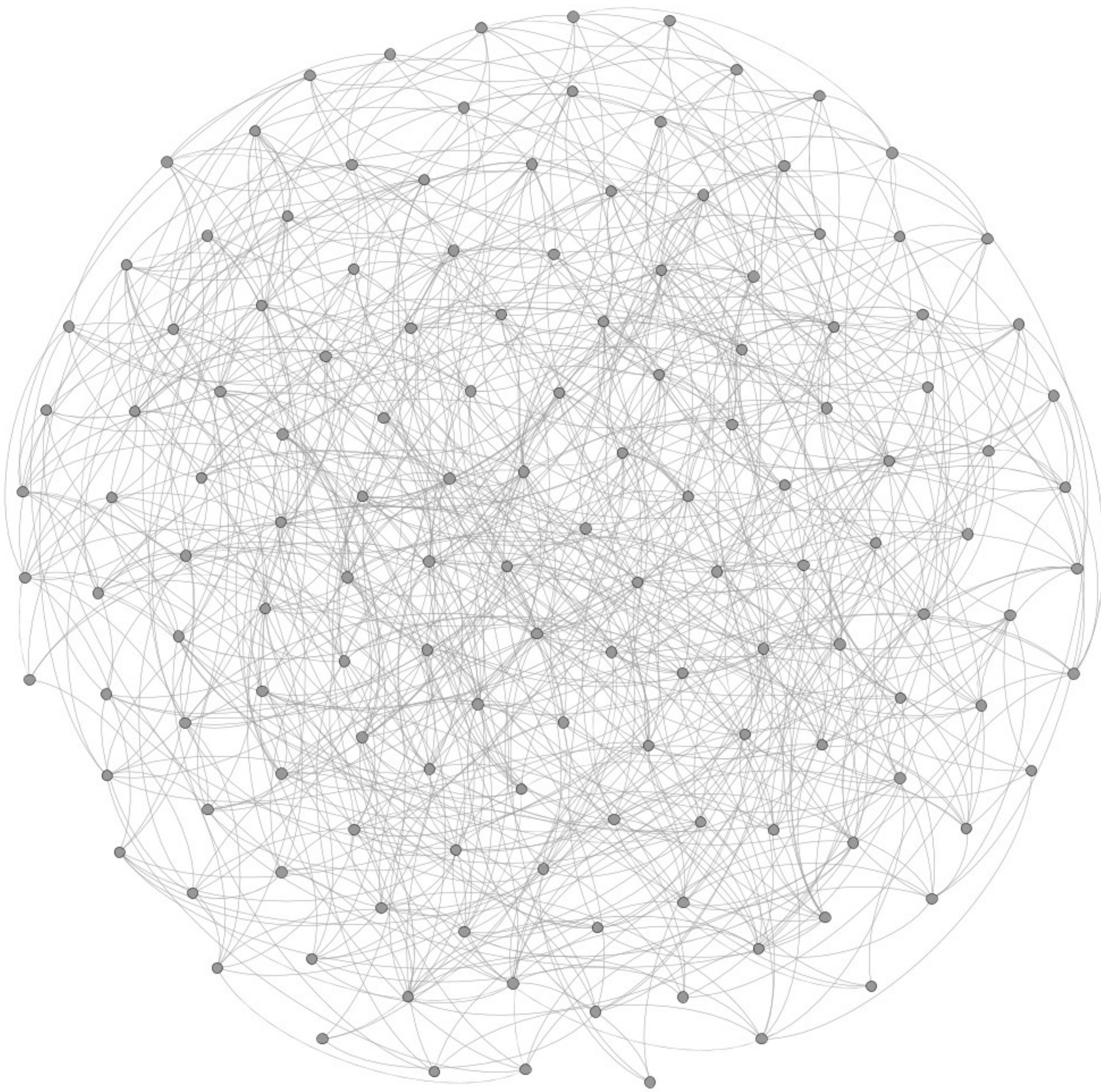
This graph showed the biggest difference between my results and Fang's. Also of note I had to change the probability of learning to 0.1 as 0.3 (the number Fang used) converged way to quickly. I am a bit dubious of Fang's results though specifically where the performance goes flat without turnover. Once reality flips I would still expect to see learning. What this simulation is telling us is that with regular turnover in an organization to bring in new ideas a higher level or performance can be reached in the face of moving targets.

ER Graphs (Erdos-Renyi)

I ran ER graphs through the same simulator over the probability of edge creations. Each point is generated over 100 runs over 2 separately generated graph. Complexity was set to 5 and Plearning was set to 0.3. I found a similar results to the caveman graph, with low performance in the low (pEdge < 0.1) probability. As expected the lack of connectedness in the graph early can be seen in the steep jump in performance at the left side of the chart. You can see performance peak right around 0.08, very similar to the cavemen graph peaking when Beta was a similar value.
This is also expected as this is the optimal point of low number of verticies, thus once again showing that diversity is key to optimal performance.



Here is an Er Graph with 0.08 edge probability as shown in Gephi



FUTURE

run ER graph with turnover. (running while you are reading this)

WS Graphs (Watts-Strogatz)

BA Graphs (Barbasi-Albert)

Lattice, McGee, Goldberg Graphs

Tweak the sim software to use multiple threads to speed up simulations.

NetworkX has a function that will generate all graphs in “An atlas of Graphs”, Reed-Wilson
Do you happen to know where I can get it in pdf form. I can order the hardcover but would like it in pdf.

start generating centrality measures and showing those in the graphs.
Build matrix of data describing the graphs.
Start running data analysis on this matrix pca/mds/lda (dimensionality reduction)

add to the learning function eigenvalue of neighbor as a parameter, centrality as a proxy, temperature

internal transfers. Like tunrover what about having nodes swap beliefs every say 50 ticks with probability p_transfer

changing structure during a run. Forming new connections, dropping existing every say 100 ticks with probability p_change chart changing graph values vs performance dissimilarity.

Keeping these notes up-to-date and creating power point slides as well

Conclusion

References

1. March, J. G. 1991. Exploration and exploitation in organizational learning.
   Organ. Science. 2(1) 71–87.

2. Fang, Lee, and Schilling 2010 : Balancing Exploration and Exploitation Through Structural Design Organization Science 21(3), pp. 625–642