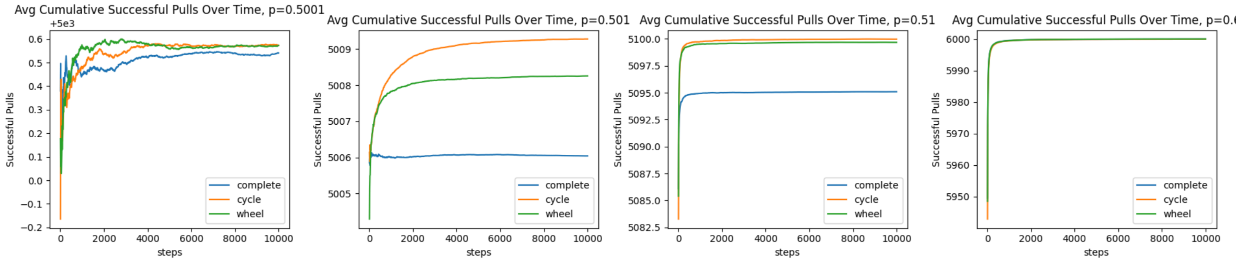
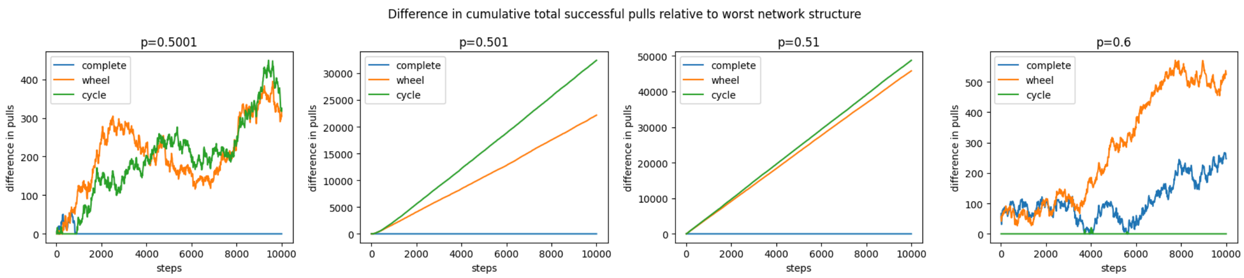
Network Simulations Updates Log - Fall 2024

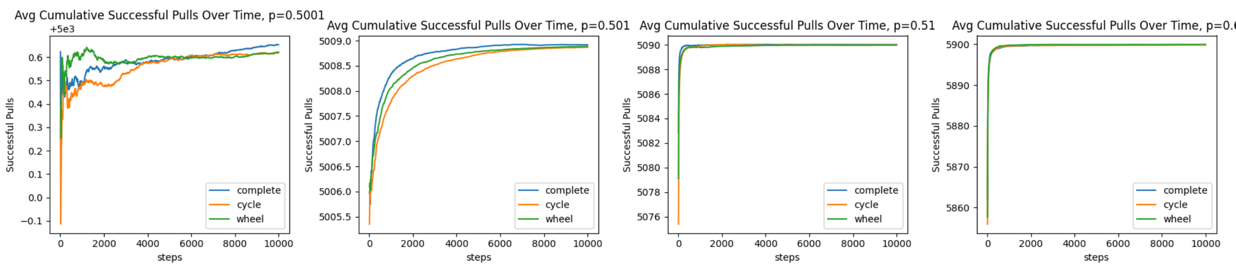
11/12/2024

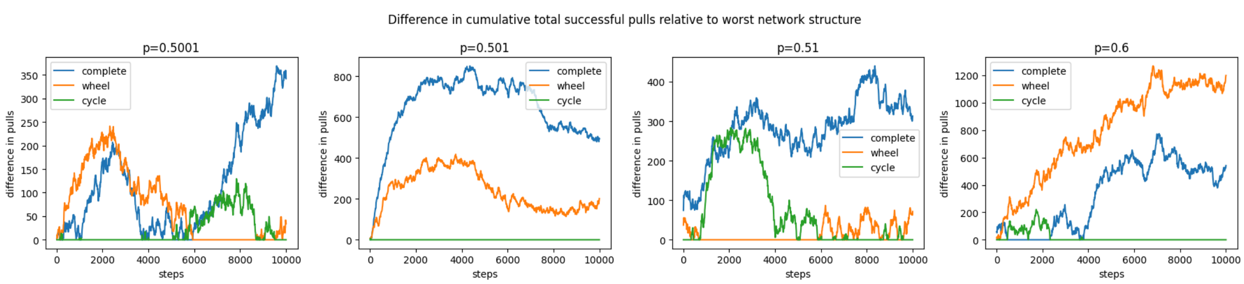
Total successful pulls for baseline zollman





Total successful pulls for epsilon greedy zollman w/ eps = 0.1



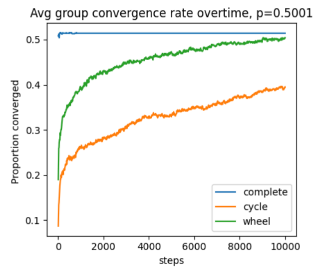


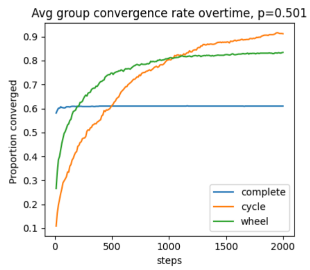
11/6/2024

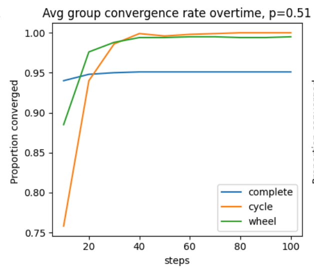
Total successful pulls and time to converge for baseline zollman setup, no epsilon.

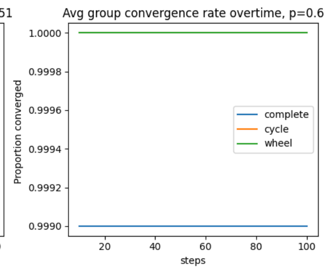
Time to Converge: Group and Individuals

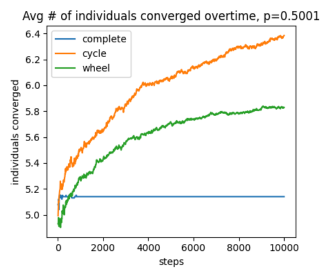
Group convergence only includes convergence to correct arm

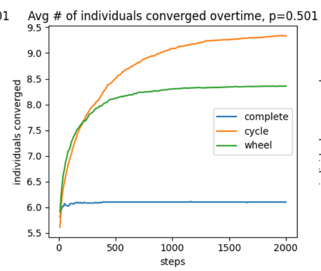


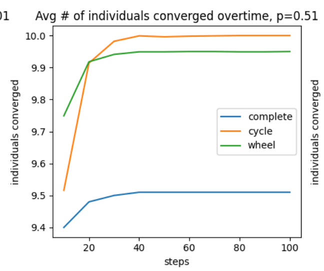


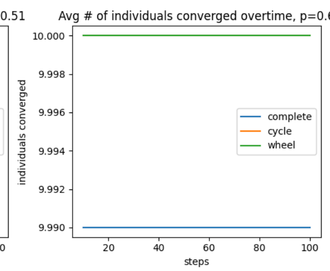






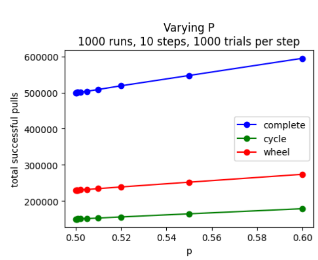


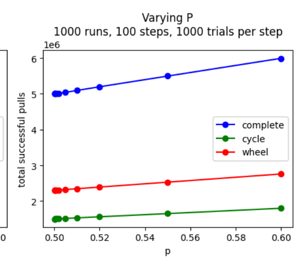


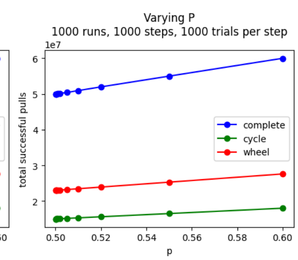


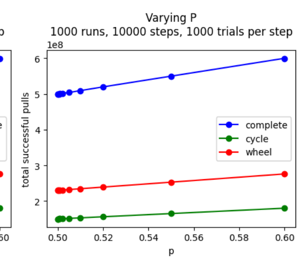
Total Successful Pulls

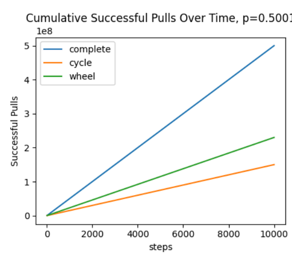
Looking at successful trials from both arms

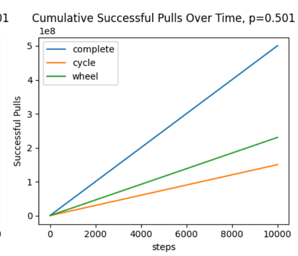


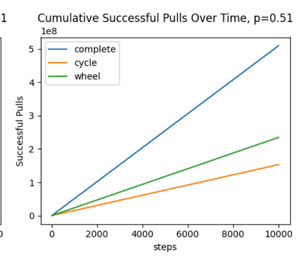


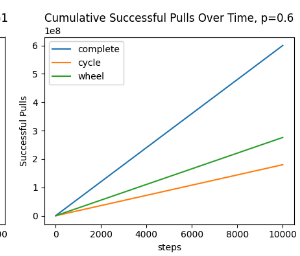


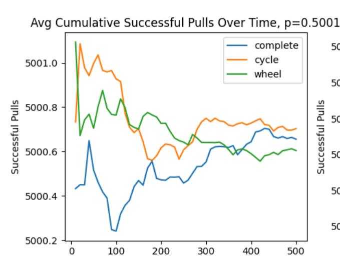


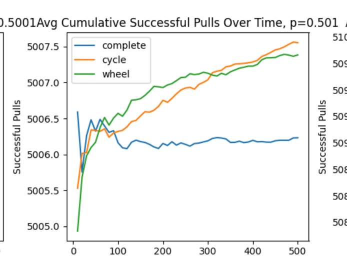


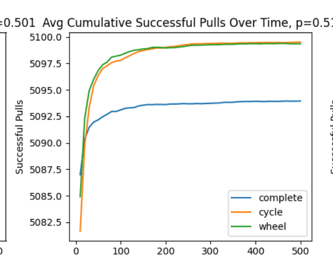


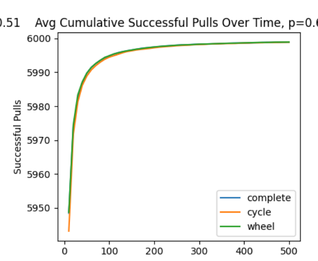








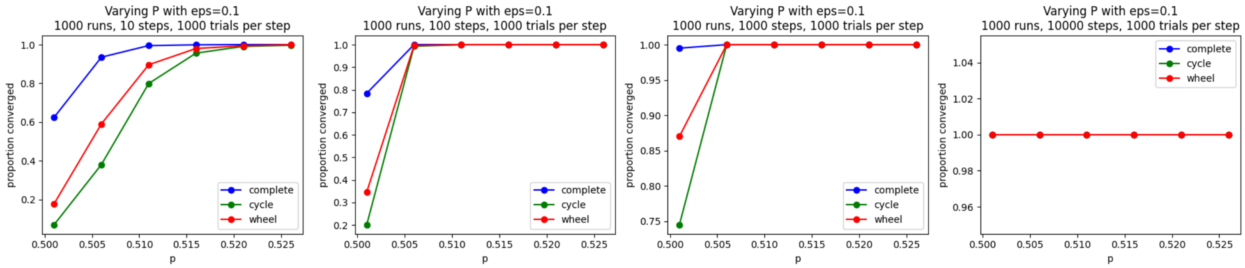




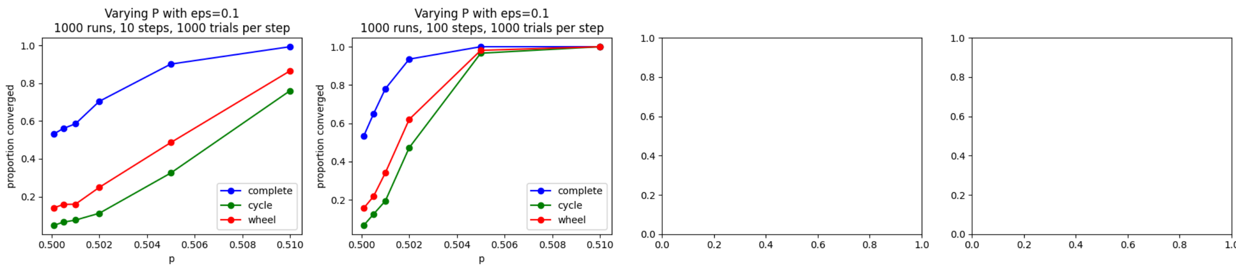
10/30/2024

Proportion converged

First, we recreate the previous graph (Varying P) with the addition of eps=0.1. It turns out that 10,000 steps with 1/10th exploration is plenty to guarantee convergence, so we also look at smaller orders of magnitudes of steps.

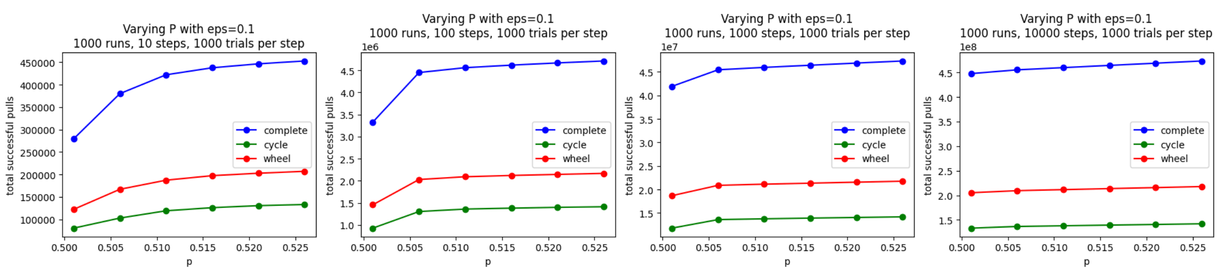


Zoomed in on smaller values of p

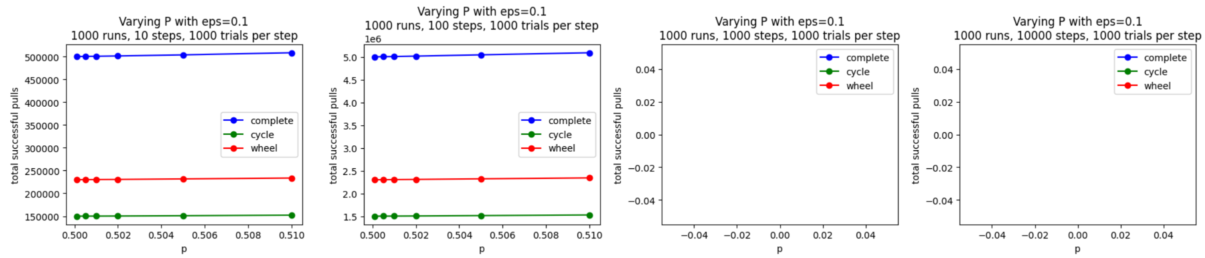


Total successful pulls

Average total successful trials at the end of the simulation. Note: this is looking at the individual trials. i.e. one step/pull yields 513/1000 successful vaccine doses, so add 513 to the total. This seems to make more sense as the total reward in this scenario instead of +1 for pulling the correct arm, which makes more sense in the scientific progress scenario.

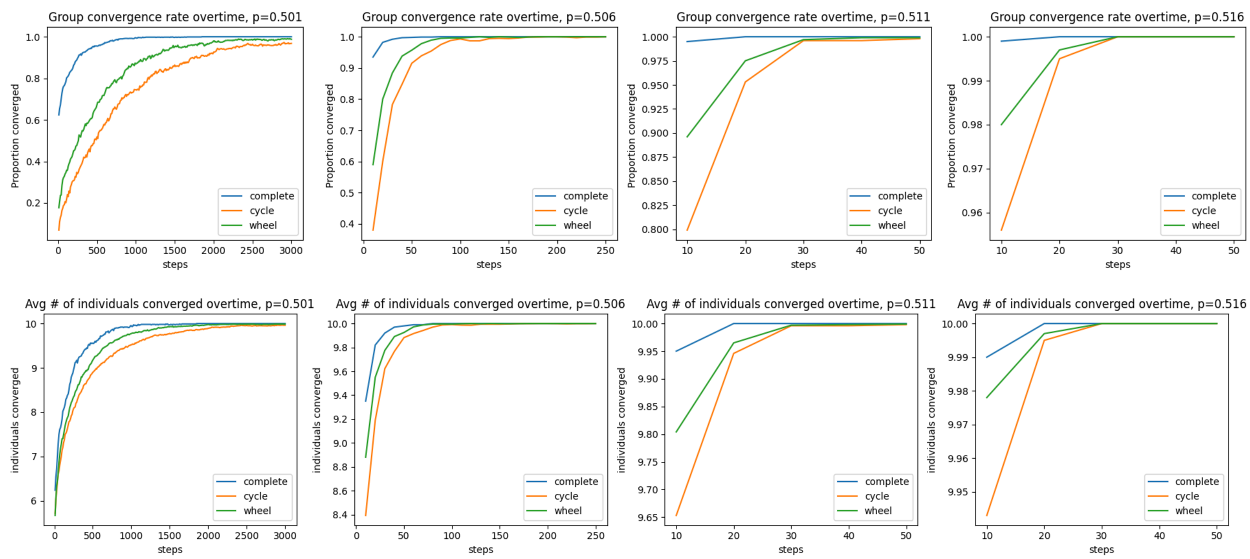


One caveat to this is that I think I messed it up and only recorded successful trials from the correct arm. This would be highly correlated with the number of correct arm pulls, which is still informative, but doesn't accurately represent the total "reward" of the simulation. This was corrected for in a mini rerun of the simulation where now the value is the total reward from both arms, and the results seem to reach the same conclusions:

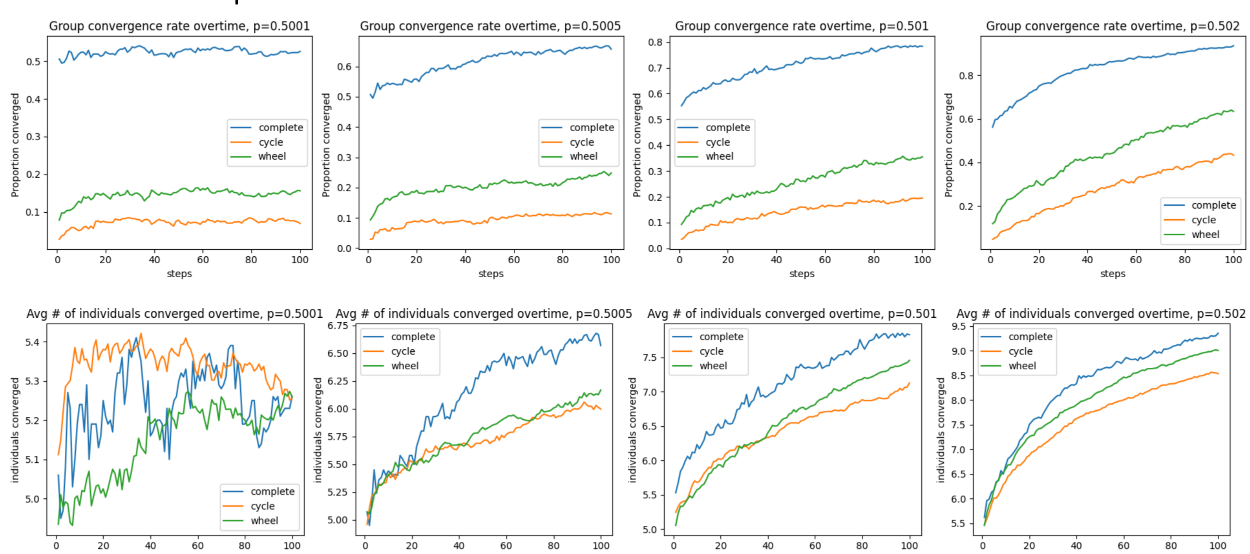


Time to converge

As p increases, the simulations become trivial as all agents converge almost immediately. So we only look at the smallest values of p for both group convergence and individual convergence (note the difference in x-axis step ranges across p values):



Smaller values of p from mini rerun



Additional Notes and Conclusions:

1. Across the board, denser graphs are better with epsilon.

2. I still need to rerun the regular simulations (without epsilon) with the total number of successful pulls and time to converge.

3. The mistake I made here was that I wanted the p values to go up to 0.6 to match the wagner and herington setup, but forgot that values above .52 with epsilon=0.1 were trivial in the sense that they all converge almost instantly.

4. For the most recent simulation, I tested p values from .501 to .6 incrementing by .505 each time. 10000 steps and 1000 runs for each parameter, and recording the state of the group every 10 steps. (and a mini run with 100 steps but using the p values from last week's meeting.)

5. Another thing I noticed that is different between the W&H and Zollman setup that I don't think we talked about before is that in W&H only one arm uses the restricted graph and one uses the complete graph. But in Zollman, both arms use the same graph.

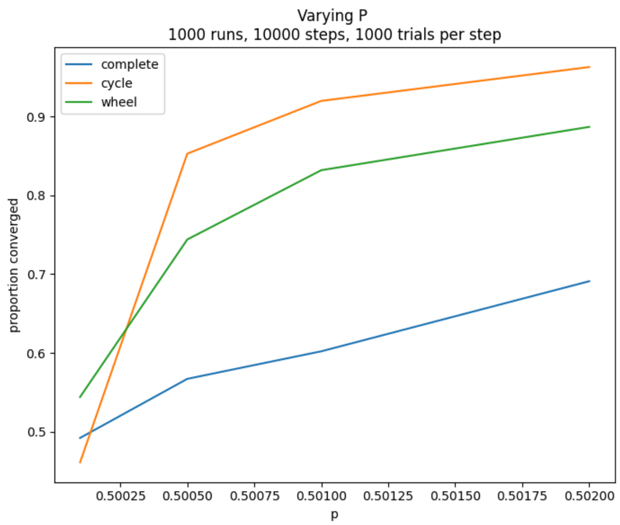
6. An interesting but irrelevant observation: In the complete graph, the group convergence perfectly corresponds with the average number of individuals converged ex:

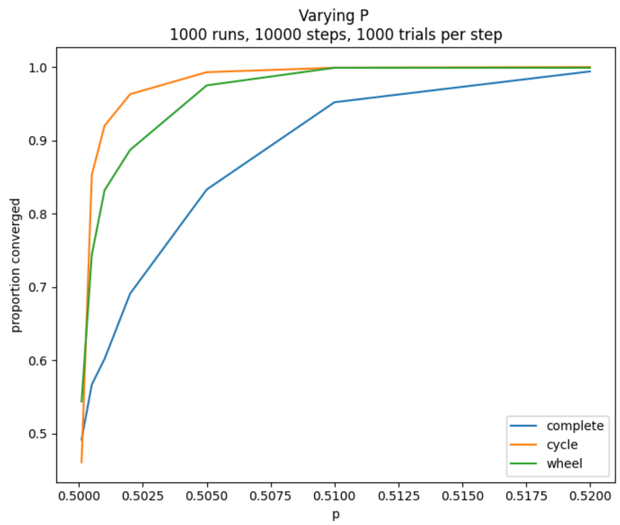
success_convergence_rate	individuals_conver...
0.515	5.15
0.507	5.07
0.51	5.1
0.52	5.2
0.53	5.3
0.514	5.14
0.516	5.16
0.514	5.14
0.522	5.22

   At first I was worried this was the result of a coding error, but it only happens with the complete graph, and not with the others, so there must be some cool underlying mathematical relationship or something.

Meeting: 10/23/2024

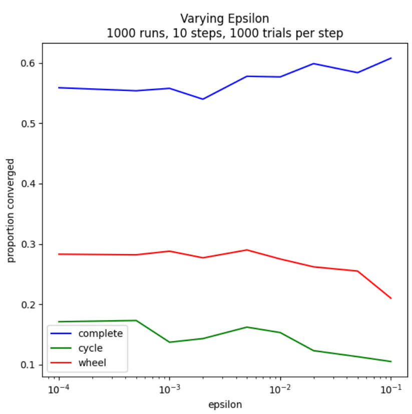
Effect of P in Zollman setup. (9 agents)
-Note: graphs are the same, but one is more zoomed in on the smaller values

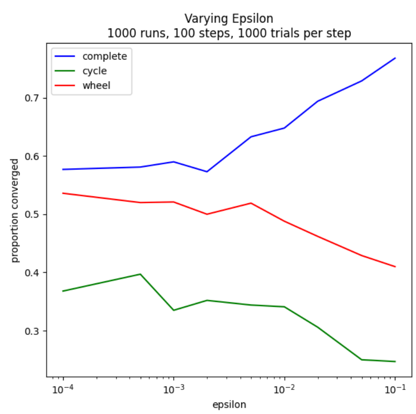


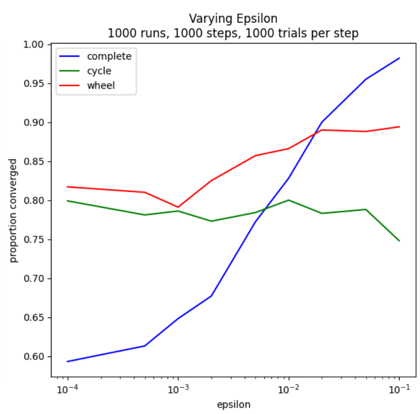



```
P Values: [0.5001, 0.5005, 0.501, 0.502, 0.505, 0.51, 0.52, 0.55, 0.6]
```

Adding Epsilon Greedy into the Zollman setup. (9 agents, p=.501)



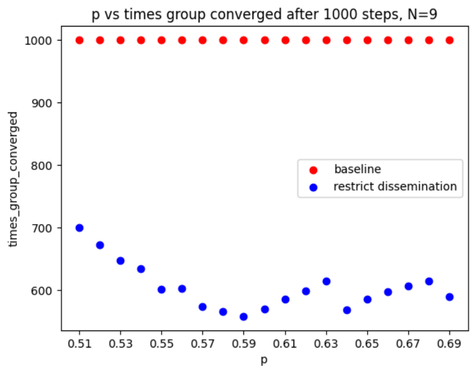


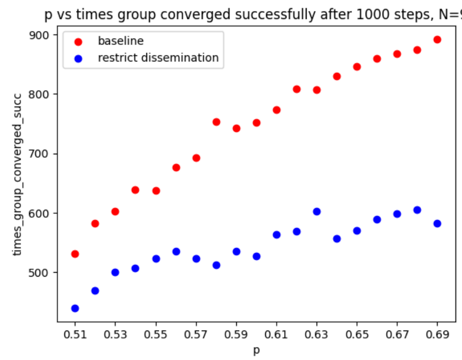


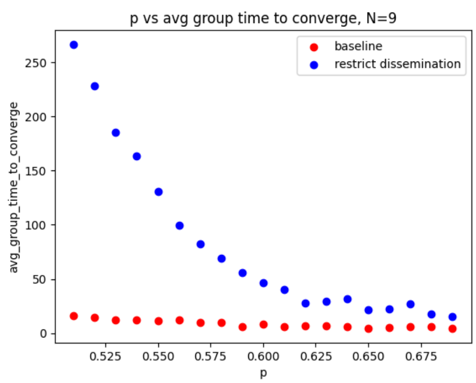

```
Epsilon Values: [.0001, .0005, .001, .002, .005, .01, .02, .05, .1]
```

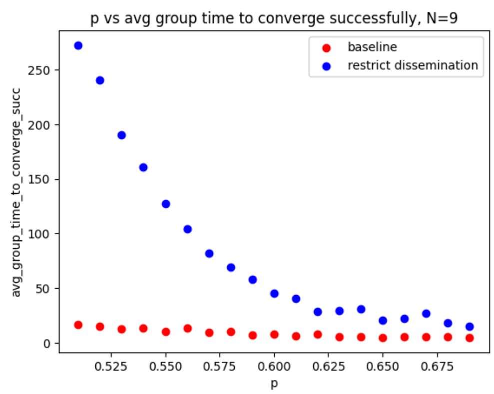
Meeting 2:

Investigating times group converged vs times group converged successfully:

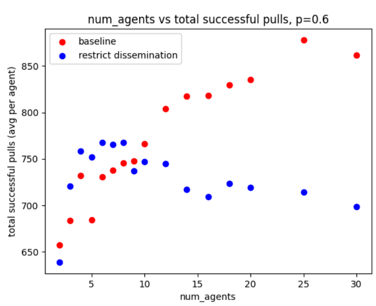


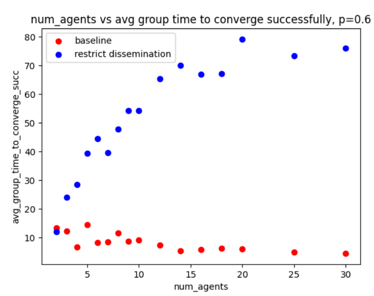


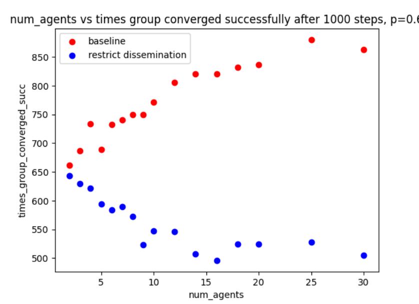




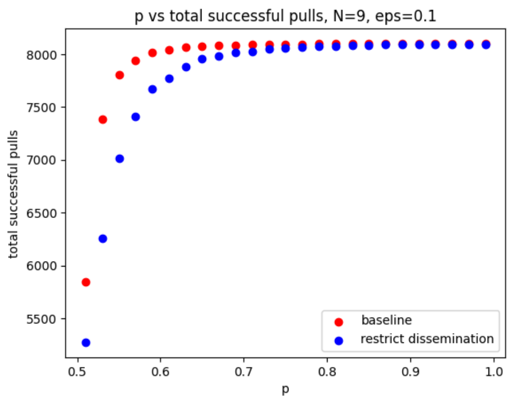
Effect of N, default scenario

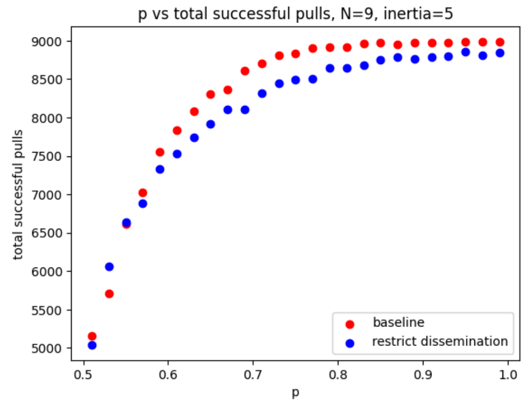




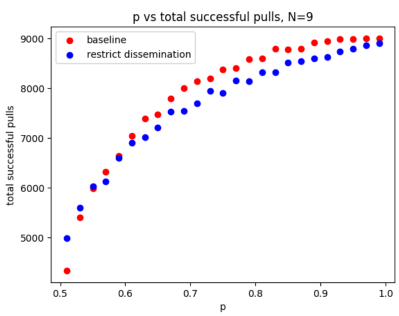


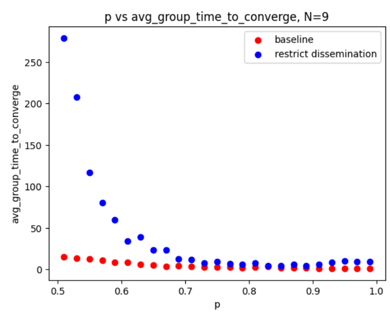
Effect of p on total successful pulls for epsilon and inertia scenarios
(messed up convergence conditions so those plots are not included)

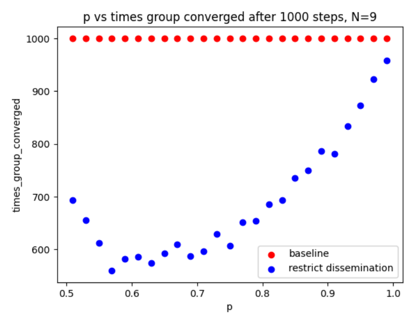




Meeting 1: Default scenario, effect of P on total successful pulls and group time to converge



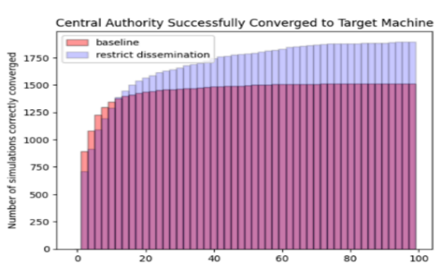


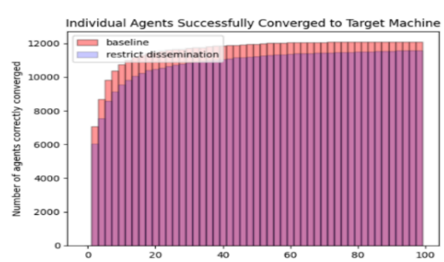


Summary of Results on Network Simulations - Spring 2024

<u>Central Authority vs Individual Agents</u>

These simulations are the same as in Wagner and Herington's setup, but with keeping track of individual beliefs as well. In virtually all cases, the individual agents not only do not learn the correct arm as well as the central authority does when dissemination is restricted, but they even perform worse than the baseline.

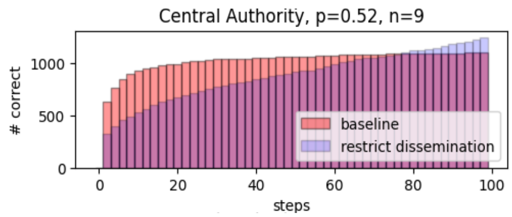


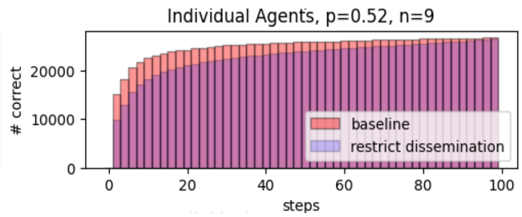


This graph shows p=0.6 and n=9, but this trend holds for most other parameters. (including when inertia and epsilon features are added)

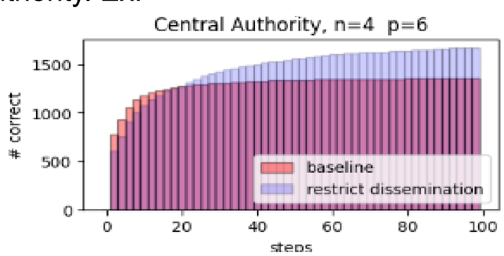
The two slight exceptions for this are:

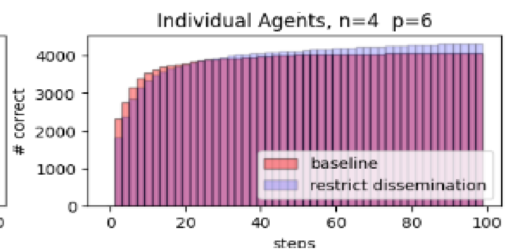
1. For low target p values (.51-.54 ish), the individual agents initially seem to learn better than the central authority relative to the baseline case, but are surpassed by the central authority by the end of 100 steps. Ex:





2. For networks with few agents (3-5 ish) the individual agents do often benefit from restricting dissemination, however they still perform relatively worse than the central authority. Ex:





Notes:

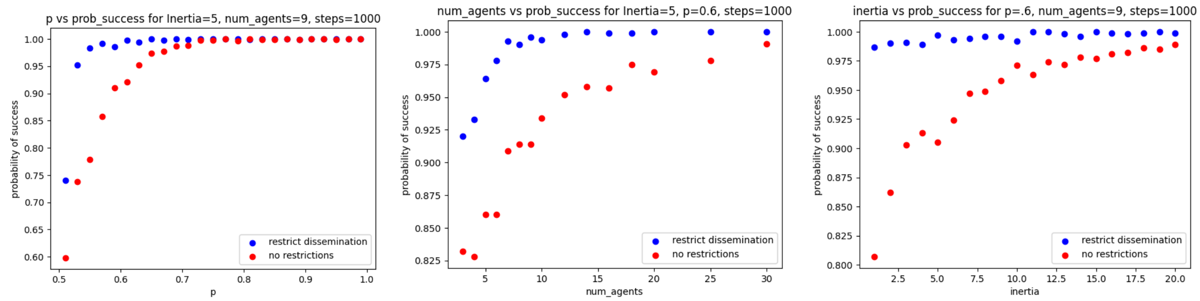
- Y-axis' are not standardized between graphs, so findings are only based on relative performance without direct comparison between parameter values.
- Many of the graphs seem like they continue to evolve past 100 steps, but in most practical cases 100 steps is probably enough to evaluate a network's success.

In the next 3 sections, we add a new feature in an attempt to make the simulations more realistic. Each feature is isolated from the rest to focus only on its own effect.

Inertia

In these simulations, agents don't switch their action until their belief has been changed for x consecutive number of steps. This feature helps the network tremendously in all cases (both baseline and restricting dissemination)

So far, results suggest that restricting dissemination is always better than the baseline for any parameters. The only exception is when the target p value is very high, where both the restricted dissemination and baseline case are practically guaranteed to succeed anyway.

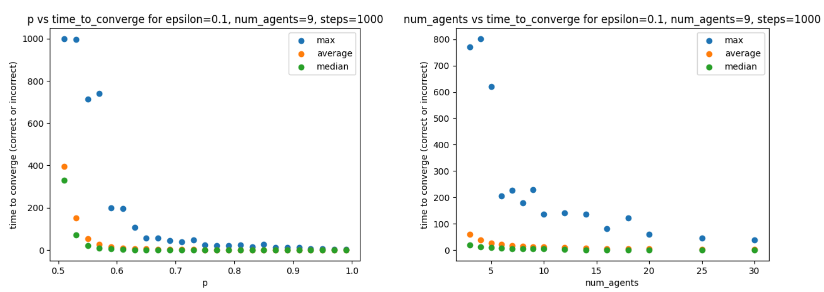


Effect of various parameters on central authorities' probability of successful learning after 1000 steps.

Epsilon-greedy

In these simulations, agents explore what they believe to be the worse arm with probability epsilon. Results suggest that for any values of p, number of agents, or epsilon (minimum is 0.01), the network is virtually guaranteed to successfully converge. This would seem to suggest that restricting dissemination does not have a substantial effect on the network, although all these results are after 1000 steps, so the effect on time to converge remains to be completely seen.

Some analysis on this shows that for small values of p and num_agents, it is possible for the network to take a *very* long time to converge, but otherwise it is usually quite fast. Though further analysis may be necessary.



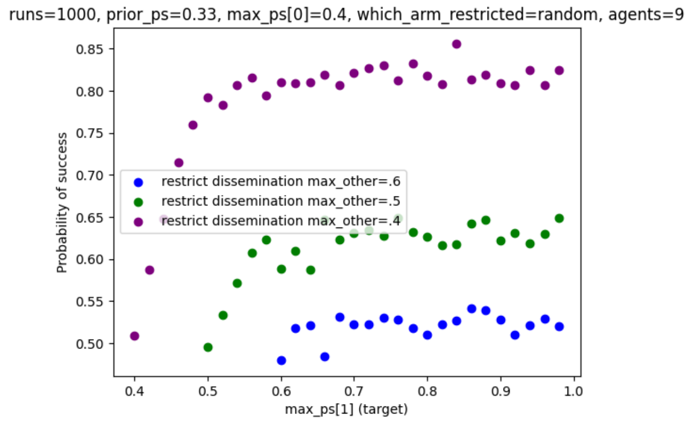
Time to converge statistics for restricted network

Dynamic p Values

In these simulations, an arm's p-value gets a bonus for each successful pull that has been seen by the agent pulling it, capped at the machine's max p value. These diverge the most from the original setup, since now we are concerned not only with both arms' initial/current p-value, but also their max p-value, which serves as the basis for determining the objectively better target arm. For this reason, all the simulations run so far have the initial p value for both the target arm and the other arm start at the same value. Here are some of the trends:

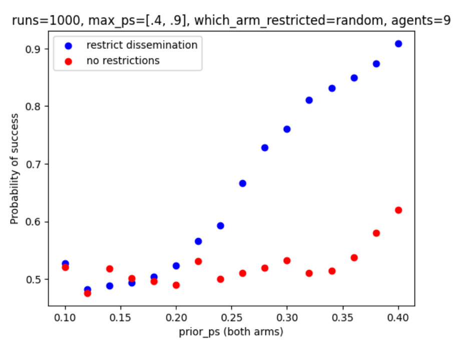
1. In general, restricting dissemination always causes the probability of success to improve or stay the same, but never hurts it.
2. Increasing the max p value of the target arm (starting at approximately equal to the max p value of the other arm) initially provides a huge benefit to the probability of success,

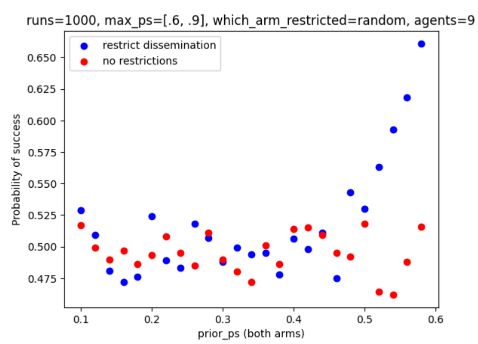
but quickly converges to some capped value, and this value is inversely proportional to the max p value of the other arm.



(here the no restrictions case is always around .50 for all max_ps[1] (target) values, with maybe a barely perceptible positive slope)

3. Increasing the initial p values of both arms initially does nothing, with both restricted and non restricted cases hovering at about .50 probability of success until a certain threshold is hit, at which point the probability of success rapidly increases with the initial p values. The threshold that this occurs seems to be proportional to the max p value of the other arm.





4. The baseline case only sees a boost in success probability whenever the initial p value and the max p value of the other arm are the same or close together.