Group Activities - An Economics and Computation final project

Abstract— **In this paper we examine matching agents to group activities when the agents have utilities over activities and mutual friendships with other agents. We present algorithms based on real world matching schemes and examine the utility, stability, and fairness of the results.**

I. INTRODUCTION

Group Activity Selection is abundant in professional life and education. In schooling when there is a list of possible projects or reading groups, each with a set capacity, the students must consider which group to join based off their own preferences and what their friends might pick. In Industry, companies assign their employees to different projects. In Academia research groups are formed. In all of these cases and more the agents need to be capable to work on their activity, and they should be willing to cooperate with other members of their group. The ability to work and cooperate has been passed over in many matching algorithms, however "motivational capital", a byproduct of internalized roles, norms, and values from a strong social network, is very important for providing incentives otherwise missing.

In this paper we present algorithms similar to those used implicitly or explicitly by people, namely drafts, cascades, induction, and community selection. These algorithms are further refined by considering different ways people might value the resulting subnetwork of group mates. We review their benefits through average utility, the envy-freeness, and the α-stability of the matching.

II. MOTIVATION

In the real world people are situated in a social network, normally with a community structure [1]. By selecting a group, the people are then in a subset of the full network where they would still like to have some sort of community structure for better cooperation [2]. In Network Science different centrality measures have been created, some of which are relevant to social situations. By using these criteria our matching can achieve the benefits of a community without explicitly discovering said communities. The logic for these centrality measures is explained in depth, but they mostly boil down to people wanting to be important in a group and have a tight knit group. Based on this we implement group selection algorithms and determine which ones can best achieve a good outcome for the agents based on their friendships and preferences.

III. OUR MODEL

Inputs:

- Agents A.
- Number of groups G.
- Group Capacities $\forall g \in G : C_g \in [0, |A|]$.
- Preferences $\forall a \in A, \forall g \in G : P_{ag} \in (0, 1)$.
- Connections $N = (A, E) : E = \{(a_1, a_2) : a_1 \in A$ and $a_2 \in A$ and a_1 and a_2 are mutual friends $\}$

Outputs: O, A list of length $|G|$ s.t $\forall i \in G, O_i \leq C_i$ and $\bigcup_{g \in G} O_g = \emptyset$

IV. FRIENDSHIP MEASURES

Original reports on centrality from the 50s all concluded that centrality was related to group efficiency in problem-solving, perception of leadership and the personal satisfaction of participants. [4]. Harold Leavitt (1949), Sidney Smith (1950), Bavelas (1950), Bavelas and Barrett (195l), and Leavitt (1951) all concluded that centrality was related to group efficiency in problem-solving, perception of leadership and the personal satisfaction of participants. we present some measures currently defined that we use for our model. They are

- Betweeness Centrality
- Clustering coefficient
- Closeness Centrality
- Degree Centrality

Each measure is associated with some sort of intuitive basis or rationale for its own particular structural property. These are based, variously, on the psychology, politics or economy of human communication. Betweenness Centrality indicates how important each agent is for the overall communication of the group and is defined as

$$\Sigma_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

where σ_{st} is the total number of shortest paths from node s to node t and $\sigma_{st}(v)$ is the number of those paths that pass through v. Clustering Coefficient indicates how tight-knit of a friend group the agent is in within the assigned group. It is defined as

$$C_i = \frac{|\{e_i j : v_j, v_k \in N_i, ejk \in E\}|}{k_i(k_i - 1}$$

where N_i is the neighbors of i. Closeness Centrality the reciprocal of the sum of the length of the shortest paths between the node and all other nodes in the graph. Thus, the more central a node is, the closer it is to all other nodes

$$C_i = \frac{1}{\Sigma_j d(i,j)}$$

,,And finally Degree Centrality is simply the degree of each node. It is the most basic measure. As the process of communication goes on in a social network, a person who is in a position that permits direct contact with many others should begin to see himself and be seen by those others as a major channel of information. In some sense he is a focal point of communication, at least with respect to the others with whom he is in contact, and he is likely to develop a sense of being in the mainstream of information flow in the network.

$$C_i = Degree(i)$$

V. Algorithms

Group selection is sometimes left entirely in the hands of the agents. In school projects this happens quite often and even at some companies such as Valve "Employees collectively decide what projects they would like to work on. People vote with their feet by joining the projects they choose. Projects that don't get enough people don't get done." [10] Therefore we base the design of our algorithms on schemes people regularly do in real life. To this end we hope to present analysis that is useful in the real decisions people make when performing this operation. Designing algorithms that create the perfect group assignment but are NP-hard to compute, difficult to explain, or untrustworthy are unlikely to gain traction outside of social choice theorists.

Based on these criteria the algorithms we present in this paper are as follows

- Captains draft ($O(n^2)$)
- Plurality draft ($O(n^3)$)
- Independent Cascade ($O(i * n * |neighbors(n)|)$
- Modularity Assignment ($O(nlgn)$)
- Backwards Induction ($O(|G|^n)$

Backwards Induction is when a order is declared and each agent in that order states their choice to all other agents. Because of the nature of this game, each agent will pick their dominant strategy and therefore the early agents can use induction to determine their optimal strategy. This method is more likely to benefit agents with high degree (or whichever measure is used in creating the order). This method could only be used in the real world, however, when size permits and when all information (preferences, connections, capacities) are known.

Captains Draft represents a draft pick which is commonly used for sports. In the Captains Draft implementation the only criterion used for deciding each candidate is the captains increased utility from that candidate (i.e a selfish captain). To mitigate this we seed the captains based off centrality measures. Similarly Plurality Draft follows the same procedure but the entire group votes on the next member. The voting mechanism used is plurality.

Result: O
captains = TopDegree(Agents)
assign each captain to a group
while *not all agents are in a group* **do**
  | Each captain picks their most preferred agent
end

Algorithm 1: Captains Draft

Independent Cascade corresponds to the situation in which people know their preferences over activities but not their friends preferences. In this case people keep switching and observing choices for a set number of discrete time steps until groups are finalized.

Result: O
for *all agents* **do**
  | Assign agent to most preferred group
end
for $t \in 1, \ldots, T$ **do**
  | Each agent switches to a (possibly) different group based on
  | • Number of Direct Neighbors in each Group
  | • Preferences over the groups
  | • How much capacity each group has
end
for *all agents* **do**
  | Finalize agent assignment to most preferred group if the group is not full. Otherwise save agent for later.
end
for *all remaining agents* **do**
  | Assign to most preferred non-full group
end

Algorithm 2: Independent Cascade

Modularity Assignment is different from all other algorithms in that it first tries to detect community structure through maximizing modularity with a greedy algorithm, then having these communities vote on which group to join.

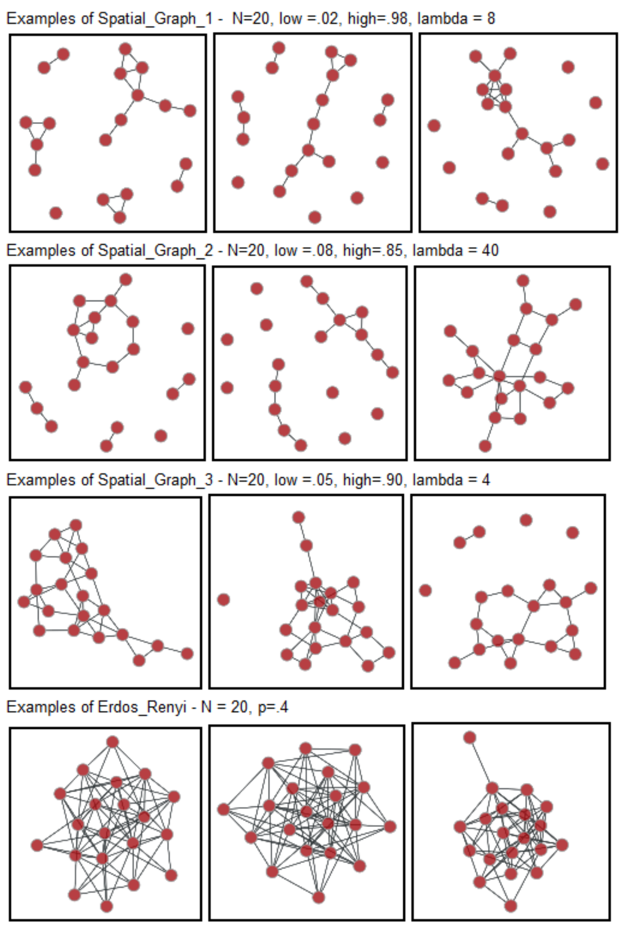
VI. Past Work

Many past papers have dealt with the group activity selection problem (GASP), introduced by Darmann et al. (2012). In GASP players have preferences over pairs of the form (activity, group size). This is because certain tasks are best performed in small or large groups, and agents may differ in their preferences over group sizes; however, they are indifferent about other group members identities. According to our setup however we abstract away the decision of what size group is necessary for each activity by creating the group capacity as an input, and allow agents to have friends who they would like to be matched with. The preference over large vs small groups is included in the preferences of agents over groups. the GASP problem has been extended to agents in a network who have explicit preferences over other agents in [3]. The authors study the computational complexity of finding stable outcomes in a small set of networks: paths, stars, and graphs with small connected components. Their model continues to follow preferences of the form (activity, group size). The biggest difference between our models is that we consider the value of a community to be intrinsic and based on the topology of the network and also for only friendships to be defined and not the value of the friendship. Additionally we view groups as having fixed capacities and agents having numerical, not ordinal, preferences over groups.

In [5], Anshelevich studies algorithms to find an approximately stable solution that leads to self-interested agents benefiting from mutual coordination. Most of the paper deals with additive utilities from each agent, however not all of our Friendship measures behave this way. When they do, like degree centrality, Anshelvich gives a $\sqrt{2}$ α-approximation algorithm for $m = 3$ and a $[\phi, 2]$ α-approximation algorithm for $m > 3$. The algorithm that achieves this, known as the One-shot-α-BR initializes all agents to group k_0. While there exists a player whose current strategy is k_0, who can improve her utility by at least a factor α by deviating to another strategy, allow her to perform best-response. When Anshelevich considers utilities that are sub-modular he reveals that no bound on α is possible.

In [6] Hogan wrestles with the same problem but considers the communities of agents to be mutually exclusive. This goes against [1] and other community detection literature. However the algorithms Hogan uses are very helpful in the creation of our own schemes. The paper also describes how these algorithms can greatly improve sorority rush, a very practical application.

VII. Graphs



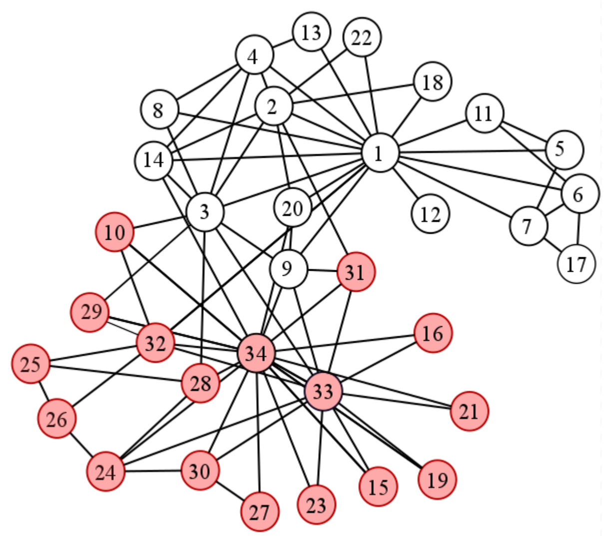
We created two methods for generating random graphs that we use to test our algorithms. The first method, A spatial model for social networks [7], is able to capture many generic properties of social networks, including the small-world properties, skewed degree distribution, and most distinctively the existence of community structures.

Result: Graph
for *all agents* **do**
  Randomly assign agent to a location on a 2D graph using Poisson Distribution using a given lambda
end
for *all pairs of agents (i,j)* **do**
  if the distance D(i,j) $\leq$ H, (i,j) is considered a "high pair" else (i,j) is considered a "low pair"
end
for *all "low pairs"* **do**
  Randomly assign edges to these pairs at a low rate of PL
end
for *all "high pairs"* **do**
  Randomly assign edges to these pairs at a high rate of PH
end

Algorithm 3: Spatial Social Network

Our alternate random graph generation model does not create community structure. It is the Erdos-Reyni model [8] in which agents have a constant probability (in our case $.4$) to connect with each of the $\binom{N}{2}$ other agents.

We also consider a classic community detection benchmark graph known as Zachary's Karate Club [9]. In this graph there are two communities which are known due to a split and fallout of the karate club over drama between the two most central members.

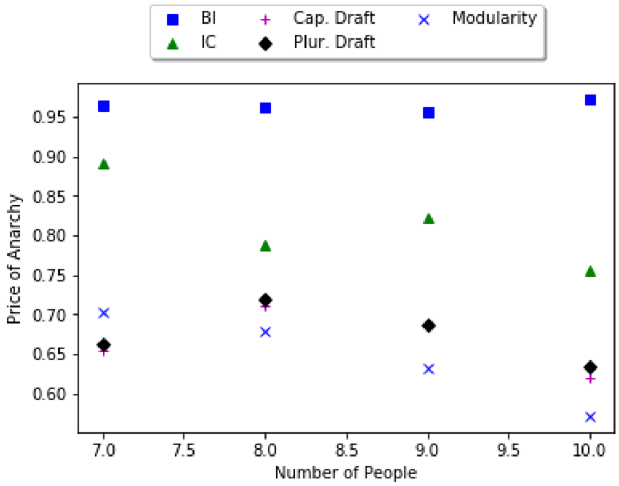


VIII. Results and Analysis

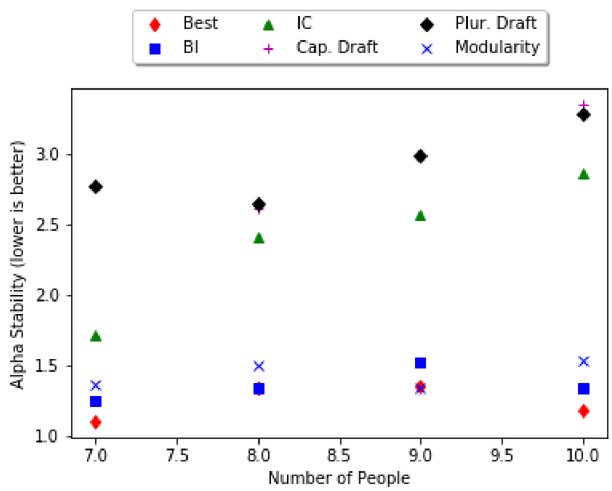
The algorithms tested differ in run time, thus we ran tests on two sizes of graphs. For small graphs, we tested all of our algorithms. For large graphs, we tested our polynomial time algorithms; Independent Cascade, Captain's Draft and Plurality Draft.

A. small graphs

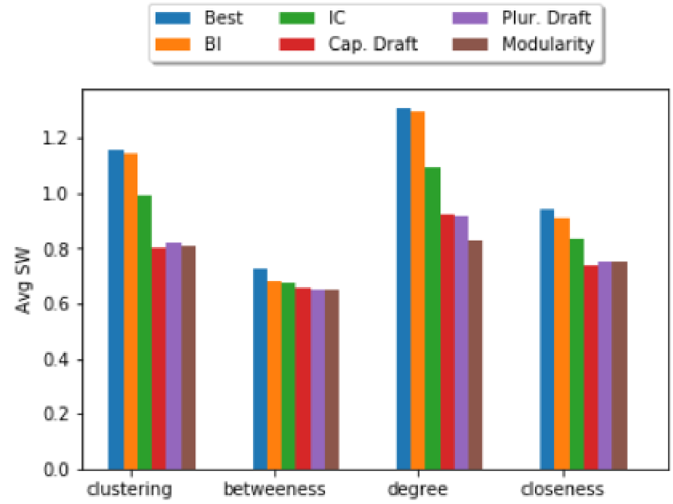
On small graphs we were able to test our algorithms against the best possible social welfare achievable. All tests were run 50 times and averaged. To show how good from a utilitarian perspective the algorithms are, the first graphic is of the proportion of it to the best average social welfare solution that was discovered through brute force. The friendship measure is the clustering coefficient.

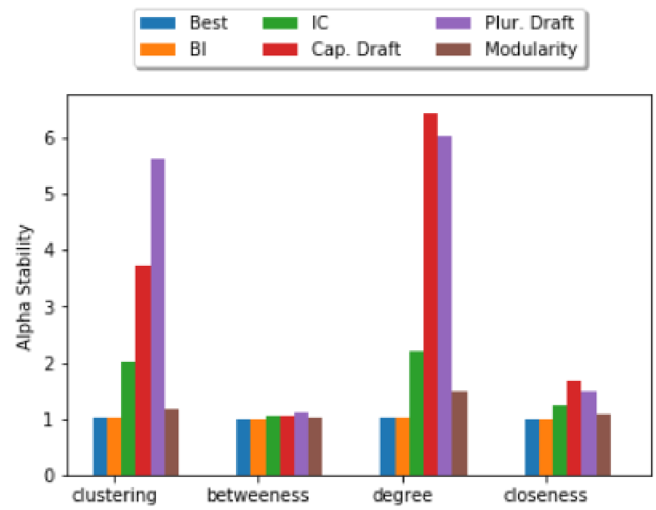


However contrasting these results with the α-stability we see that some algorithms thought to perform poorly have good properties. Note: ”Best” is for the max average social welfare outcome

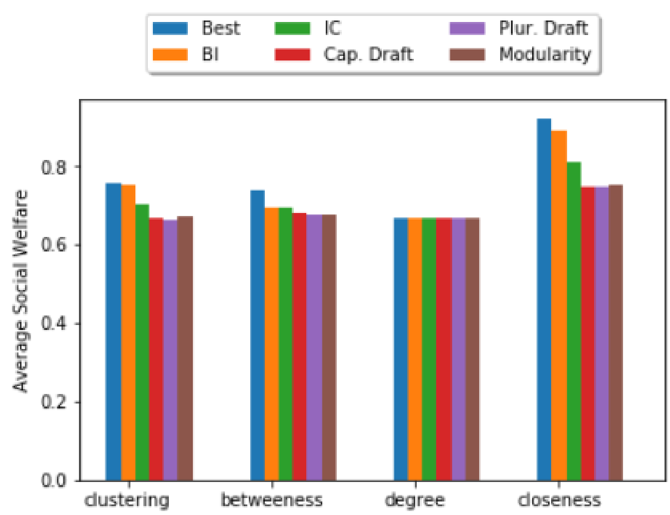


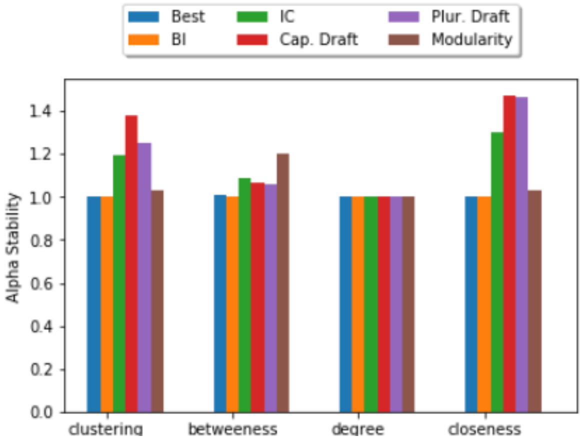
These past graphics showed how the number of people effected the outcomes. In the next images one can see what the relative differences between different friendship measures is. Below is a chart comparing just that when $N = 7, G = 2$





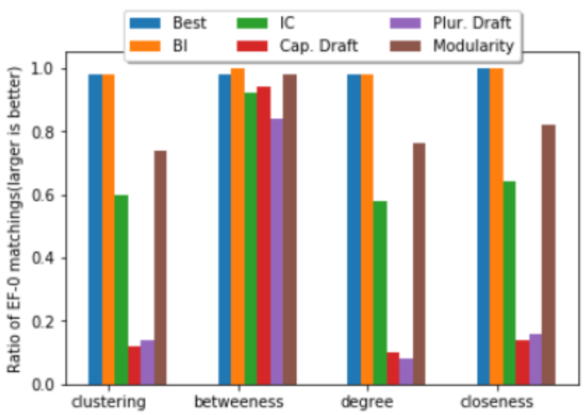
We found that the topology of the graph effected the results quite a bit. The next graphics we present are created from the erdos-reyni generation graphs. These graphs, due to being completely random, do not exhibit community structure. Again the following graphics are created from $N = 7, G = 2$

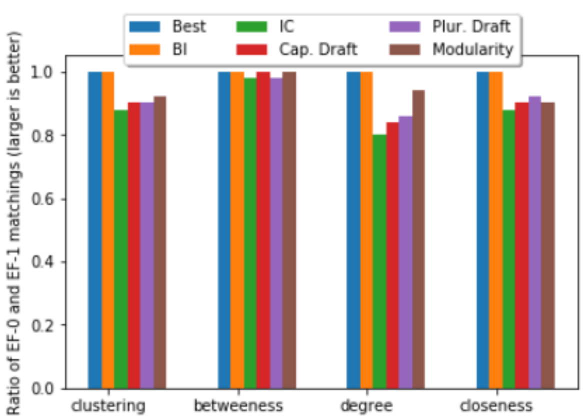




What all these graphics show is that maximizing social welfare is often the best approach (But NP-hard). Independent cascade has good social welfare properties, and Modularity Groups results in good α-stability Scores.

To further examine the trade off between stability and social welfare we look at the envy-freeness of each algorithm. That is the proportion of times the allocation was EF-0 and in the second chart the proportion of times it was EF-0 or EF-1.

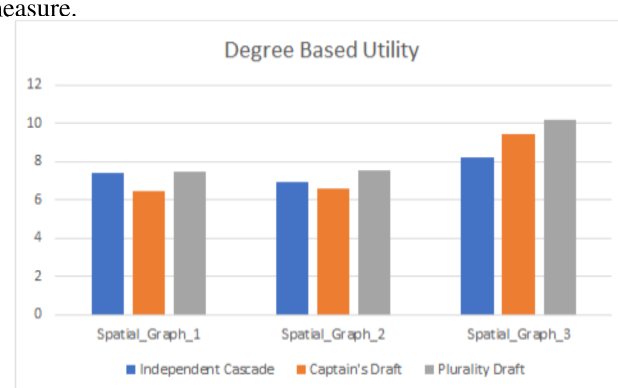


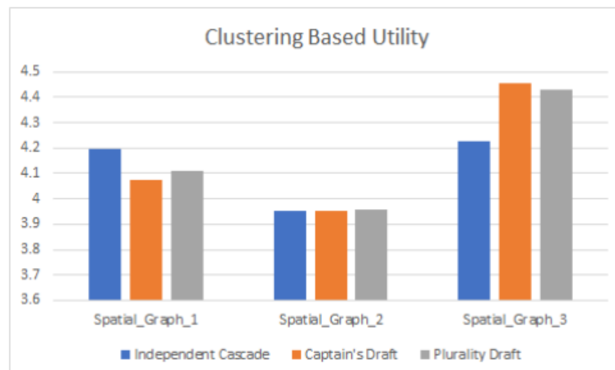


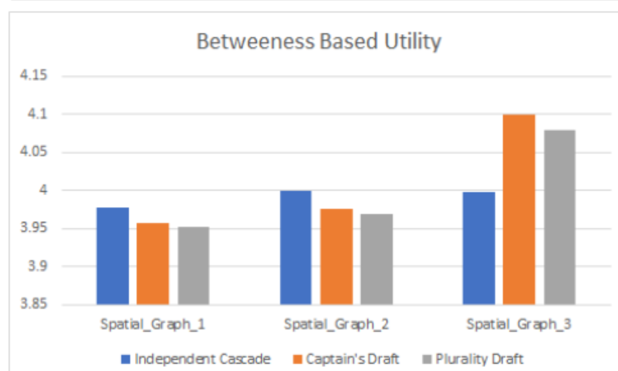
B. *large graphs*

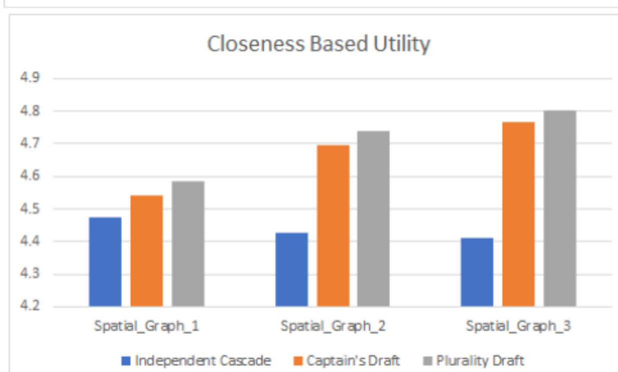
For large graphs, we ran tests on the three types of Spatial Graphs (shown in section VII) with 20 agents each. We ran tests for each combination of these types of graphs, group sizes ranging from 2 to 6, and the four types of friendship parameters. In total 60 tests were run. For each test, 50 random cases were generated to the parameter's specifications, and the Independent Cascade, Captain's Draft and Plurality Draft were run on them. The average utility, the average alpha stability, and the total number of cases where envy-freeness was 0, 1 or greater were recorded for each algorithm.

The following four graphics visualize the average social welfare for each type of graph, algorithm, and friendship measure.



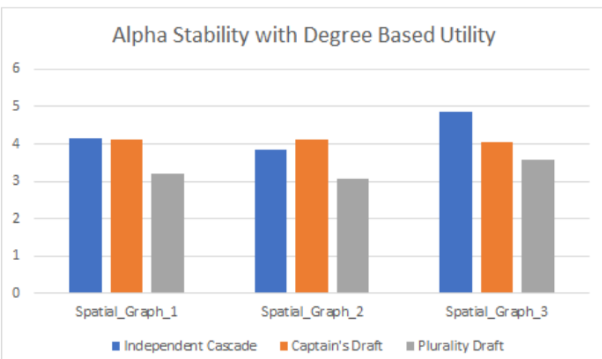


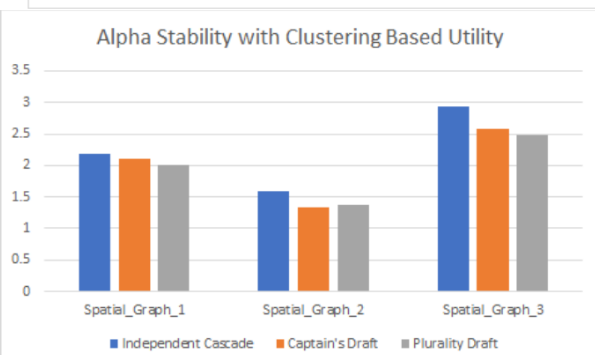


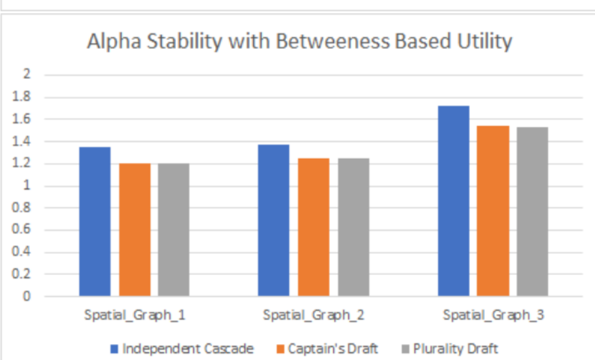


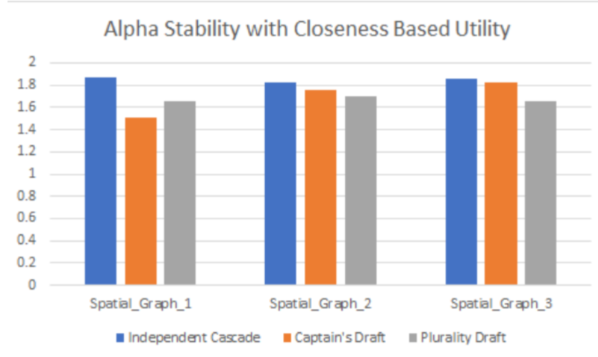
Consider that Spacial Graph 1 represents social networks that are the least connected and Spacial Graph3 represents social networks that are the most connected. The graphics show conclusive evidence that Captain's Draft and Plurality Draft tend to yield similar results. The graphics also show that the composition of a social network will effect which algorithms produce better results. In social networks with fewer connections, Independent Cascade will produce a comparatively higher average social welfare. In social networks with more numerous connections, a Draft Pick will produce comparatively higher average social welfare.

Moving on from average social welfare, we found that when examining other qualities of the algorithms, the results were more conclusive. The next four graphics show the average alpha stability for each type of graph, algorithm, and friendship measure.





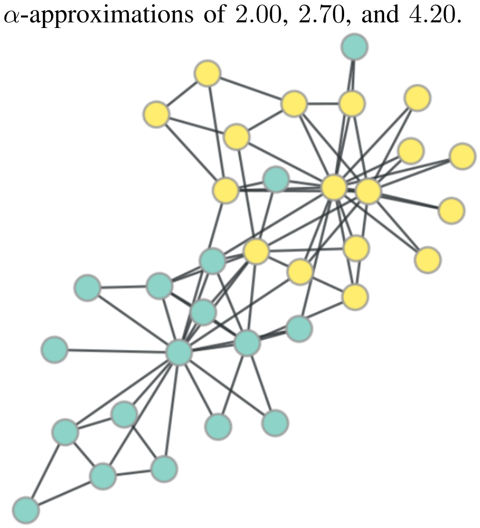


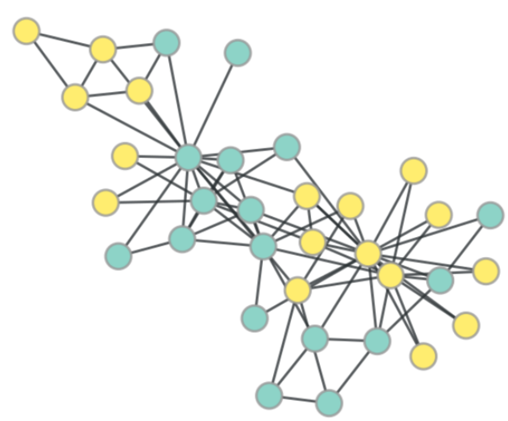


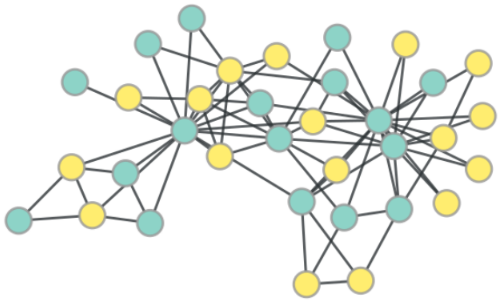
We can conclude that Plurality Draft consistently produces the best results in regards to alpha stability, even in cases where Independent Cascade produced higher Average Social Welfare.

C. Real world social networks

Here we test on real community structure graphs Zachary's Karate Club [9] is put through our algorithms with $G = 2$. Our results are below for Captains Draft, Plurality Draft, and Independent Cascade. They have α-approximations of 2.00, 2.70, and 4.20.







IX. CONCLUSIONS AND FUTURE WORK

In this study we blended together concepts from social choice, economics, and network science to provide validity to the effectiveness of group selection schemes. In future work we would like to expand the types of voting mechanisms agents could use within these schemes. Additionally we would like to find bounds for possible α-stability in social network group selection.

REFERENCES

[1] Community structure in social and biological networks M. Girvan, M. E. J. Newman Proceedings of the National Academy of Sciences Jun 2002, 99 (12) 7821-7826; DOI: 10.1073/pnas.122653799 .

[2] Alfano G, Marwell G. 1980. Experiments on the provisions of public goods by groups. III. Nondivisibility and free riding in real groups. Soc. Psychol. Q. 43(3):3009

[3] Igarashi, Ayumi Peters, Dominik Elkind, Edith. (2016). Group Activity Selection on Social Networks.

[4] Freeman, Linton C. "Centrality in social networks conceptual clarification." Social networks 1.3 (1978): 215-239.

[5] Anshelevich, Elliot, and Shreyas Sekar. "Approximate equilibrium and incentivizing social coordination." Twenty-Eighth AAAI Conference on Artificial Intelligence. 2014.

[6] Hogan, Virginia. "Matching In Groups: A Theoretical and Empirical Study." (2013).

[7] Wong, Ling Heng, Philippa Pattison, and Garry Robins. "A spatial model for social networks." Physica A: Statistical Mechanics and its Applications 360.1 (2006): 99-120.

[8] Erds, Paul, and Alfrd Rnyi. "On random graphs, I." Publicationes Mathematicae (Debrecen) 6 (1959): 290-297.

[9] Wayne Zachary. An information flow model for conflict and fission in small groups. J. of Anthropological Research, 33:452–473, 1977.

[10] Pronschinske, Mitch. Lessons from 7 Highly Successful Software Engineering Cultures. TechBeacon, TechBeacon, 5 Feb. 2019, techbeacon.com/app-dev-testing/lessons-7-highly-successful-software-engineering-cultures.