CSE 516: Final Project

Due: Dec 16, 2019, by 1:00 PM on Gradescope

- You may work in teams of 2 or 3 for this project, picking one of the suggested projects below or work on your own idea with prior approval from Sanmay. You **must post a private note on Piazza by Dec 6 with the names of all members of your team (just yourself if you are working individually) and the topic of the project you are working on.**
- Post a note on Piazza if you are looking for teammates.
- The final deliverable is a (maximum) 4-page PDF file prepared using the ACM template available at `http://www.acm.org/publications/authors/submissions`. 1 PM on Dec 16th is a HARD deadline. Note that your work will be judged on the quality of the writeup you submit!
- This should not be a huge amount of work, especially in teams; the idea is more to provide you with an opportunity to explore topics and demonstrate your creative range in thinking about problems in multi-agent systems. You can think of it more as a team take-home final than a major project in terms of effort.

1 Matching papers to reviewers

Take a look at the AAMAS Bidding Data available at `http://www.preflib.org/data/matching/aamas/`. In particular, you'll want to eventually work with the data for AAMAS 2015 on this problem. Learn about the data and what the bids represent. Then, look at the paper (and code, if you wish) available here: `http://www.cis.upenn.edu/~cjtaylor/RESEARCH/projects/OptimalAssignment/OptimalAssignment.html`.

The key goal here is to take this idea for how to do paper assignment from the paper above, and apply it to the conference bidding data. There isn't necessarily an exact mapping between the exact forms of the two problems, but you can make reasonable assumptions to transform them. Following that, the main goal is to understand the qualities of allocations as a function of different parameters of the problem. Here are a few examples of things you can change about the input data in order to study questions about how preferences and constraints affect the qualities of allocations:

1. The number of papers each reviewer can be assigned at most, and the number of reviews that each paper needs at least.
2. The relative strengths of preferences in the different categories (yes/maybe/no-response) for each reviewer.
3. The range of preferences – for each reviewer, consider different ways of breaking up their preferences within each category and assigning different "points" to those. For example, you could take "yes" to be 10 points, "no" to be 0, and vary the values of "maybe" and "no response" to see what kinds of outcomes you get.

One important question to think about is how you should measure allocation quality. Think about designing at least a couple of measures for the quality of the eventual allocation. While one of these (e.g. a total score) could be the input to the solver, you can also look at performance on the other metrics (e.g. what proportion of papers are assigned to those who did not express any interest in them, and what proportion of top choices do bidders get)?

2 Repeated congestion games

We saw congestion games, and, in particular, Braess' paradox, in class. The goal here is to try and understand whether agents playing the game *repeatedly* converge to equilibrium, and how different parameters may affect long-term outcomes. This can be a useful model of dynamics in road traffic networks, for example, where agents could be playing the same game repeatedly and have to decide on a path each time.

Develop an infrastructure where a fixed set of agents can play the versions of Braess' paradox with and without the "superhighway" on a repeated basis, and keep track of the choices and rewards of each agents at each point in time. The key implementation detail here will be an algorithm for learning about costs on paths and making decisions about which path to take based on estimates of these. There are only two or three possible paths, so you can restrict your focus to thinking of these as 2- or 3-action games. For the learning model, algorithms you can consider include *fictitious play*, multi-armed bandit algorithms like ϵ-greedy, UCB1, or Thompson sampling, or algorithms that maintain and update a prior over what the population of other agents will do and choose an action based on that. (You should look up these algorithms; they should all be simple to implement).

Again, feel free to tackle this in any way that you find interesting, but here are some suggestions for questions and issues to explore in this domain:

1. How are outcomes different in the two versions of the road network (with and without the "superhighway")?
2. How does the number of agents playing affect outcomes?
3. How are things different if you can see your own cost, but not the cost along every path at a given time, versus situations where you can see the cost along every path even ones you don't traverse?
4. What are the effects of agents using different learning strategies? (I suggest sticking to one strategy across the whole set of agents initially, but you're welcome to try mixing the learning algorithms and having different proportions of agents use different learning algorithms as an extension; one interesting possibility might be to fix the learning algorithms for all bue one agent and then find which algorithm performs best in)
5. What are the effects of different initial choices or prior beliefs for the agents?

3 School Choice

Read the paper "Leveling the Playing Field: Sincere and Sophisticated Players in the Boston Mechanism", available here `https://economics.mit.edu/files/3025`. Now, design a simulation that creates schools and students, with meaningful priorities that the schools have over students, and student preferences over schools, and a large number of slots per school. The priorities should be restricted to a small number of categories (e.g. each student is placed into one of four priority categories 1-4 by each school), while student preferences should be over the entire set of schools. For example, you may want to generate schools with different qualities on the 2-D plane, and have school priorities be based on distance from the school, while student priorities could be based on a combination of quality, distance from school, and an idiosyncratic random factor. However, you can come up with your own simulation mechanism.

Assume two types of students: *sincere*, who always report their true preferences, and *sophisticated*, who try and figure out the best preferences to report given what they know. Design and explain a decision rule or algorithm for sophisticated players to figure out the preference list they will report. Be explicit about your assumptions and why you think this is a good rule or algorithm. Now compare the outcomes of the Boston mechanism versus student-proposing Gale-Shapley (or deferred acceptance) under different sizes of the sophisticated/sincere split, and perhaps different ratios of schools and students. Explain the measures you use to analyze how good the outcomes are, and justify your use of these measures.

A couple of notes:

- Repeat the simulations many times in order to report meaningful results.
- Examine the robustness of your results to small changes in parameters.
- Another paper you might want to look at is available at `https://www.cemfi.es/ftp/pdf/papers/wshop/School%20choice.pdf`.
- Be sure to justify your choices (there are many reasonable choices you can make!) and summarize your conclusions.

4 Your own idea

Feel free to come up with an idea for your own project and run it by Sanmay if you're more compelled by that than any of the above options. In that case, please initiate a discussion via private message on Piazza well before December 6!