Jeff McCrea
CSS 600 – Winter 2023
Professor Fukuda

CSS 600 Term Report

Table of Contents

Overview

For my CSS 600 independent study this quarter, the goal was to survey various topics in agent-based modeling in preparation for identifying a capstone project in Professor Fukuda's and the Distributed Systems Laboratory's (DSL) Multi-Agent Spatial Simulation (MASS) library. Over the course of the quarter, I looked at several different topics, but I focused on three main topic areas: Political Science, machine learning, and agent intelligence. To assess the academic importance and the applicability of these topics towards a capstone project, I conducted a systematic survey of recent papers published on the previously mentioned topics to understand the topic areas better and identify where the state-of-the-art is as it pertains to those research topics.

In the following sections, I will review the goals and work completed this quarter, provide an overview of one general proposal idea related to redistricting simulation and gerrymandering identification, preview my goals for CSS 600 in the coming quarter, and conclude with a reflection on the work done this quarter.

Goals & Work Completed

Per the CSS 600 proposal form created at the start of the quarter, I entered the quarter with three main goals:

1.) Identify topic areas of interest in ABM
2.) Conduct literature of reviews of current academic research as it relates to the topic areas
3.) Develop a proposal outline on topic area

I started the quarter by surveying topic areas within ABM related to my political science background. Initially, I started looking at election and voter simulation. While I found the work interesting, many simulation models seemed ill-suited for implementation in MASS. From there, I looked at agent intelligence through machine learning integrations in ABM. I conducted an in-depth literature review on agent intelligence through machine learning. As a result, I gained a well-versed understanding of how machine learning can be integrated into ABM, where it can be integrated (micro/macro-level), and how machine learning can impact the overall model. I found this topic to be interesting and certainly a worthwhile area of study, but I struggled to find a topic area that I felt was attainable with my skillset and academically worthwhile. I am still researching this topic and hope to determine a potential avenue for use in MASS.

Following my literature review on machine learning integration in ABM, I returned my attention to political science simulations. During this survey of topics in political science, I focused specifically on topic areas that can benefit from MASS's ability to store and explore a large amount of data in parallel. As I will go into greater detail in the proposal idea section, one area of political science that can benefit from how MASS is set up is redistricting simulations. Modern redistricting approaches require large amounts of data on demographics, geography, and state/federal constraints to generate potential redistricting plans. Enabling the generation of district plans through parallel simulation in the MASS environment could significantly increase performance for district plan generation and improve the scale at which the simulations can be conducted.

After reviewing literature on redistricting simulations, I turned my attention back to machine learning. I reviewed the work completed by Collin Gordon and Chang Liu on implementing machine learning and data science algorithms within MASS. In their works, they implemented ML and data science algorithms within MASS to take advantage of the parallel execution and rated the algorithm's performance against distributed data processing frameworks like Hadoop MapReduce and Spark. These works are similar to the final project in Professor Fukuda's CSS 534 class, albeit more rigorous and systematic. I feel these works could be extended with additional algorithms that have been implemented successfully in MapReduce and Spark to further reinforce MASS as a viable, if not a superior, platform for this type of work.

Proposal Idea Overview: Redistricting Simulation & Gerrymandering Detection

Overview:

When surveying the application of agent-based modeling in the field of political science, there are many different avenues where agent-based simulations are useful, such as election simulation, voter behaviors models, and political polarization studies, among many others. While many of these use cases lend themselves well to agent-based modeling approaches, legislative redistricting is one use case that seems well suited to the MASS library.

Legislative redistricting is the process of redrawing electoral district boundaries following the constitutionally mandated decennial census. While many people have little interest in the process compared to political elections or cross-cutting issue debates, redistricting significantly impacts political representation. The actual process of redistricting varies from state to state, but the inherently political nature of the process has historically left it open to manipulation by political actors to advantage certain political entities unfairly. Studies have found that when districts are created in a way that yields a partisan slant, voter turnout and, thus, representation is negatively impacted [8].

As mentioned, redistricting can be used as a tool to give an unfair advantage to a particular political group. Gaining a political advantage through redistricting is known as gerrymandering and has a long history of use in the United States to cement political power. The goal of gerrymandering is to amplify a desired political party's power beyond what they would receive based on the makeup of the voting population. The amplification of power typically comes in the form of two complementary methods known as packing and cracking, shown in figure 1 [5]. Packing is the process of diluting the power of one voting block by consolidating a majority of the target voting population into a small number of districts. While this would award the packed party the illusion of proportional representation through overwhelming victories in the consolidated districts, the remaining members would be spread across a large number of districts, known as cracking. In combination, packing and cracking create what are known as gerrymandered districts which insulate the current ruling party from competition and deprive marginalized groups of meaningful representation.

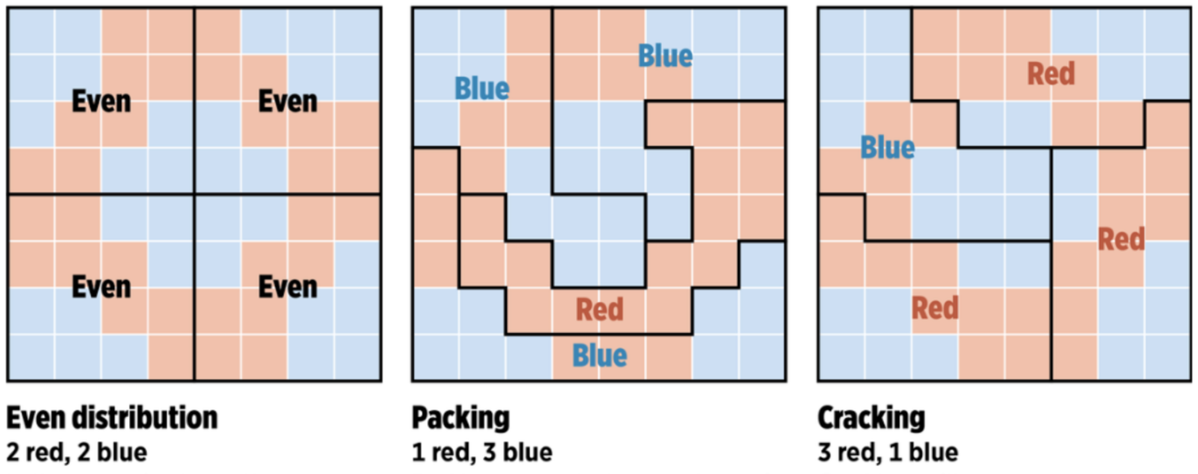

Figure 1 - Classic Gerrymandering Setup [5]

While some states have sought to remove political influence from the redistricting process with independent redistricting commissions, those efforts still leave the process open to human biases and partisan influence. To address the issue of gerrymandering, academics have proposed empirical methods to remove partisan influence and generate redistricting plans based on geographical and voting population requirements [3]. However, evaluating redistricting plans is not a simple task. Each state and the federal government have specific redistricting criteria and political geography that must be considered. Washington, for example, lays out requirements in Article 2, Section 43 of the state constitution as well as in the Revised Code of Washington (RCW) [6] that districts must be:

1.) Be contiguous
2.) Have equal population
3.) Be geographically compact
4.) Preserve county and municipality boundaries as much as possible
5.) Not be connected across geographic barriers, although ferries across water may establish contiguity
6.) “Provide fair and effective representation and ... encourage electoral competition.”

Given these unique and varying state and geographical requirements, a one size fits all approach to generating redistricting plans is not feasible. Instead, generating redistricting plans requires careful consideration of a multitude of different data points, including demographics, political affiliation, geography, candidates, and other state and federal-specific factors. A naïve approach might attempt to compare a state’s current redistricting plan to past district plans to detect gerrymandering. However, that approach is prone to error as demographics, politics, and state-specific requirements can change dramatically during the period between redistricting. Recent approaches have sought to identify gerrymandered districts by comparing a proposed district plan with a slate of possible plans generated algorithmically [11]. Under this approach, a redistricting plan can be considered gerrymandered if it constitutes an outlier relative to a sample of alternative plans that satisfy the same set of statutory guidelines and requirements with respect to certain partisan bias metrics.

Simulation algorithms to generate redistricting plans have risen to greater prominence in recent years, mainly due to the increasing availability of granular data about voters and elections and increased computing power [3]. With ample data and computing power, simulations can incorporate the multitude of different federal and state requirements with respect to voter and election data to generate a large sample of district plans that meet the necessary requirements. The simulation methods have proved successful with numerous federal and state court cases featuring their results in challenges against redistricting plans [3].

Related Works

Before the turn of the century, many of the conceptual redistricting simulation algorithms could not capture all the necessary data points and constraints due to a lack of data and computing power. However, starting in the early 2010s, the field of algorithmic redistricting simulation saw a renewed focus thanks largely to the better availability of data and rapid growth in local and distributed computing systems. In the field of algorithmic redistricting, two main frameworks proved successful: Better Automated Redistricting (BARD) and Harvard's Algorithm-Assisted Redistricting Methodology (ALARM) project's Redist library.

BARD

BARD is an open-source software package introduced in 2011 for general redistricting and redistricting analysis [1]. The project aimed to provide a framework for analysts to create, display, compare, edit, and automatically refine and evaluate districting plans. The hope was that by providing this framework, the difficulty associated with creating and analyzing district plans could be abstracted away and allow wider public participation in the creation of new plans.

BARD provided several features that are notable in the field of redistricting analysis. First, BARD provided functionality to read and process redistricting data in the form of shapefiles. Shapefiles are useful as most data sources available for redistricting analysis utilize the format and enable interoperability with existing GIS and map-drawing software common in the redistricting process. Second, BARD enables the evaluation of redistricting plans through metrics like compactness score, population deviation, continuity, political competitiveness, and majority-minority political party breakdown. Third, BARD can automatically generate district plans for the analyst to use as a starting point for manual optimization [1]. This feature is one of the more critical features as it enables the user to focus on optimizing a generated plan that meets the baseline district requirements. BARD provides several methods to generate the plans, including random-walk and simple/weighted k-means. Finally, BARD enables the analyst to compare multiple plans with respect to the specified scoring criteria.

REDIST

Redist is a software library developed by the ALARM project for the automated generation of redistricting plans coupled with ample analysis functionality [2]. Much like BARD, the overall goal of Redist is to make modern redistricting simulation algorithms and analysis methods available to the public. Redist allows users to simulate and analyze alternative plans under user-specified criteria. The project has been consistently updated since its introduction, with new simulation algorithms and analysis tools added regularly.

The ALARM project runs its own simulations and generates 5,000 sample plans for each state with respect to the state's redistricting rules [2]. In addition to releasing the library as an open-source project, the data used to generate the simulations, including precinct shape files, demographic data, voting history, and other relevant data points, are also publicly available.

Redist utilizes a collection of different simulation rules that allow the user to tailor the simulation to a particular state [10]. The collection of rules includes:

- County splitting: the simulated plan will split fewer counties than the number of districts
- Population constraint: Ensures congressional districts must be roughly equal in population, so all state simulations contain a user-specified cap on the maximum deviation of any one district from the target population.
- Compactness: algorithm nudges towards compact districts by an adjacency-graph-based measure of compactness
- Municipality constraint: simulation avoids splitting municipalities within large counties to preserve political subdivision
- Voting Rights Act (VRA) constraint: simulations are encouraged to accept plans that have a concentrated minority share of the voting age population or citizen voting age population, in accordance with the Voting Rights Act of 1965

These rules allow for the Redist to maintain compliance with state and federal requirements and enable flexibility to allow the user to customize the simulation to their specified environment.

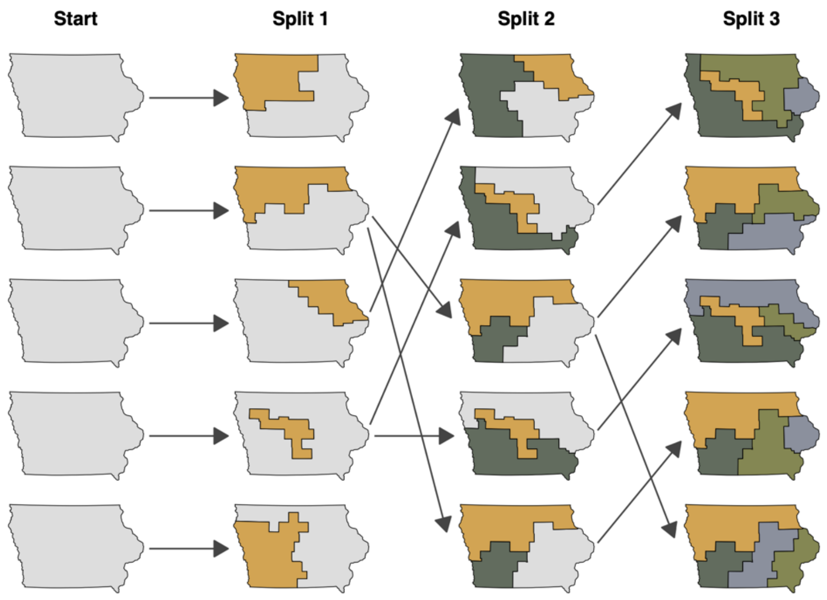

Figure 2 - Sequential Monte Carlo District Simulation Process [5]

In addition to redistricting-specific rules, Redist includes a suite of different algorithms for generating district simulations [13]. The algorithms include:

- Sequential Monte Carlo: starts with a blank state and uses a spanning approach to sample one district at a time, resampling at each step with weights. This represents the most modern district simulation and is shown in figure 2.
- Merge-split: pairs of adjacent districts are chosen randomly, merged, and split into two new districts using uniform spanning trees.
- Flip: beings with a valid district plan and then reassigns units on district boundaries with respect to a target distribution. This algorithm is useful for incremental changes and applies mainly to local exploration. Unfortunately, it suffers from scalability issues and is rarely used for state-wide simulations.

Redist has been demonstrated to produce district plans that are accurate to state and federal constraints, encourage politically competitive districts, and ensure proportional presentation with respect to district populations. As mentioned previously, the library's success and the expertise of its developers have led to its use in federal and state courts challenging unfairly drawn districts [6].

Proposal Goals

In order to emulate the type of redistricting simulation present in BARD and, more notably, Redist, the project would seek two accomplish one primary object and one secondary objective.

Primary Objective (Redistricting Simulation):

This project's primary goal would be to implement a parallel district simulation algorithm similar to the Sequential Monte Carlo and Merge Split utilized in the Redist library.

Secondary Objective (Gerrymander Detection):

If the parallel district generation element is successfully implemented in MASS, the natural segue would be to implement a module to analyze and compare the slate of generated districts against enacted or proposed redistricting plans. With the 2020 census and redistricting already complete, this would most likely focus on analyzing previously proposed and enacted district plans to determine their potential for gerrymandering. Analyzing previous plans has the added benefit of utilizing data produced by the ALARM project and other sources to verify the accuracy of the implementation.

Work to be Completed Next Quarter:

After completing this quarter of CSS 600, there are some clear areas where I need to focus on to develop and implement my potential proposal ideas successfully.

First, I need to focus on gaining a much deeper understanding of MASS and the available features as it relates to GIS systems and graph implementations. Notably, as it pertains to redistricting simulations, this requires the ability to work with geographic information to determine where current district and precinct boundaries lie, how modifying current boundaries relates to demographic shifts, and compliance with state and federal constraints. While the DSL group has done some impressive work on integrating GIS systems within MASS, I will need to work to assess the applicability to a project like redistricting simulation.

Second, I need to spend more time assessing the feasibility of implementing these proposals in MASS. While proposal ideas seem conceptually possible, a deeper look and testing are needed to demonstrate feasibility.

Third, I will create a concrete proposal on the selected topic. This was one of the goals I had set for this quarter, which was not completed. After assessing the feasibility and receiving feedback from Professor Fukuda, in the next quarter, I will seek to design my methodologies for the project implementation in MASS, identify proper data sources, and start work on developing a proof-of-concept implementation in MASS.

In the coming quarter, my overall goal is to lock down a topic idea, generate a proposal, and begin implementing the necessary features in MASS to facilitate the production of building blocks for the project. By completing these goals in the coming quarter, I will set myself up for independent work over the summer break to enable a successful beginning of CSS 595 in the fall.

Conclusion:

Overall, this quarter exposed me to many different topics in agent-based modeling and allowed me to increase my understanding of topics in agent-based modeling immensely. When reflecting on this quarter, I, unfortunately, did not accomplish all I set out to do. While I was able to complete some of the goals I set out to achieve, the main goal of identifying and developing a proposal outline was not completed. In my quest to identify what I felt was an academically worthwhile and valuable topic to the work of the DSL group, I spent far too much time researching topics that ended up being unfeasible, beyond my capabilities, or lacking academic merit. On the bright side, I feel I was able to narrow down my topic areas to some ideas that are worth exploring further. In the coming quarter, I look forward to further solidifying my topic and working with Professor Fukuda and the DSL group.

References

[1]
M. Altman and M. P. McDonald, “BARD: Better Automated Redistricting,” *Journal of Statistical Software*, vol. 42, pp. 1–28, Jun. 2011, doi: 10.18637/jss.v042.i04.

[2]
A. A. P. N. published yet DOI, “ALARM Project: The Algorithm-Assisted Redistricting Methodology (ALARM) Project,” *ALARM Project*. https://alarm-redist.org/about.html (accessed Mar. 13, 2023).

[3]
B. Fifield, Higgins Michael, K. Imai, and A. Tarr, “Automated Redistricting Simulation Using Markov Chain Monte Carlo,” *Journal of Computational and Graphical Statistics*, vol. 29, no. 4, pp. 715–728, Oct. 2020, doi: 10.1080/10618600.2020.1739532.

[4]
B. Fifield, K. Imai, J. Kawahara, and C. T. Kenny, “The Essential Role of Empirical Validation in Legislative Redistricting Simulation,” *Statistics and Public Policy*, vol. 7, no. 1, pp. 52–68, Jan. 2020, doi: 10.1080/2330443X.2020.1791773.

[5]
K. Imai, “Using Simulation Algorithms to Detect Gerrymandering: Evaluation of the 2022 Congressional Maps”.

[6]
A. 50-S. S. P. T. P. July 25 and 2022 Doi 10.7910/Dvn/Slcd3e, “ALARM Project: Washington Congressional Districts,” *ALARM Project*. https://alarm-redist.github.io/fifty-states/WA_cd_2020/ (accessed Mar. 13, 2023).

[7]
C. McCartan and K. Imai, “Sequential Monte Carlo for Sampling Balanced and Compact Redistricting Plans”.

[8]
C. McCartan *et al.*, “Simulated redistricting plans for the analysis and evaluation of redistricting in the United States,” *Sci Data*, vol. 9, no. 1, Art. no. 1, Nov. 2022, doi: 10.1038/s41597-022-01808-2.

[9]
“[2008.06131] Sequential Monte Carlo for Sampling Balanced and Compact Redistricting Plans.” https://arxiv.org/abs/2008.06131 (accessed Mar. 13, 2023).

[10]
“An R package for legislative redistricting.” https://alarm-redist.org/redist/ (accessed Mar. 13, 2023).

[11]
“Automated Redistricting Simulation Using Markov Chain Monte Carlo: Journal of Computational and Graphical Statistics: Vol 29, No 4.” https://www.tandfonline.com/doi/full/10.1080/10618600.2020.1739532 (accessed Mar. 13, 2023).

[12]
“cran/BARD: Better Automated ReDistricting.” https://github.com/cran/BARD (accessed Mar. 13, 2023).

[13]

"Function reference." https://alarm-redist.org/redist/reference/index.html (accessed Mar. 16, 2023).

[14]
"Introduction to redist • redist." https://alarm-redist.org/redist/articles/redist.html#package (accessed Mar. 16, 2023).

[15]
"redist: Simulation Methods for Legislative Redistricting." ALARM Project, Jan. 02, 2023. Accessed: Mar. 16, 2023. [Online]. Available: https://github.com/alarm-redist/redist