Constructing Playable Maps for use in Simulating Redistricting Protocols

Evan DeSantola

***Abstract*— Wesley Pegden and Ariel Procaccia recently proposed a non-partisan redistricting protocol, whereby parties can split a state into districts by taking turns proposing a districting and freezing a single district. From this semester through Spring 2019, I will be constructing an interactive online simulation to allow users to play the protocols against artificial agents within a simplified game UX that corresponds to real world electoral data. The first part of building such a simulation requires constructing an interface that allows the user to easily select regions corresponding to this electoral data while maintaining a good user experience. Once this interface is built, a first scratch of the simulation can be built for human-human play. In this work, we examine questions related to developing such a UI/UX, with a specific focus on potentially useful computational geometry algorithms, cartogram research and MILP formulations.**

I. INTRODUCTION

A. Background & Related Work

Gerrymandering is a process where a state's legislature divides that state's congressional districts in a manner so as to maximize the advantage of one political party. Although extensive investigation has been conducted into quantifying gerrymandering and examining its effects, most existing solutions rely on non-partisan agents, such as judges or independent commissions, to moderate the redistricting process. Unfortunately, the use of independent agents presents additional problems: its difficult to find agents both experienced and capable at performing the task while remaining truly impartial. In their working paper [6], Pegden and Procaccia propose an I-Cut-You-Freeze protocol, whereby two parties acting selfishly create a non-partisan electoral division. In the protocol, parties take turns dividing the state map into districts, and then choosing from the districts proposed by the other party on which district to freeze. In addition to guaranteeing districts are divided roughly in proportion to the party membership of the state, this protocol promises that one party cannot pack a minority group into less than $\frac{\sqrt{n}}{2}$ districts if the other party opposes it.

While the protocol has some nice theoretical properties, there are many unanswered questions regarding its effects in practice. A significant amount of work need be undertaken in order to understand how the protocol affects *cracking* or *packing* under non-idealized conditions, and to understand whether the protocol produces socially beneficial outcomes under more complex scenarios, such as when in the presence of multiple or non-disjoint minority groups. Investigations into how to efficiently play the protocol optimally have not yet been conducted, making it difficult to understand what second-order effects are produced from merely near-optimal play. Producing a platform that either a human or AI may play near-optimally or oprtimally would help thoroughly explore such effects, and would also help convince users that such a protocol would be viable in practice. Other open questions related to the work include examining the effects the protocol produces when parties want to ensure that they have a stronger hold on the districts than a simple majority, or exploring what happens when users interact with the protocol under different geographical constraints. By gathering data about how the protocol works in practice, we hope to answer some of these unknowns, identify new questions and better educate users on the usefulness of such an approach.

B. The Problem

Before we may begin to examine the effects of the redistricting protocol under human-play on real-world data, we must first construct a way for humans to interact with the protocol on real-world data. To do this, we need to first gather and preprocess electoral data from heavily gerrymandered states, and then construct a user interface that allows users to interact with this data.

Unfortunately, constructing a human-friendly interface that interacts with real world population data is a non-trivial task; in many states, population distributions vary wildly from densely packed downtowns to rural

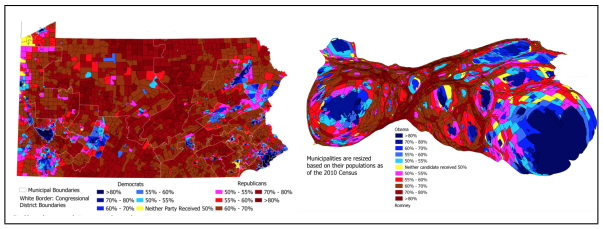

Fig. 1. Population density in cities makes constructing an equal area playable game map difficult. In many states such as Pennsylvania, a very small region contains the majority of voters [5].

farmland, making it difficult to break the state into equal-area selectable regions whose topology closely resembles that of the original state.

C. Overview of Approach

To tackle this challenge, data is first collected and preprocessed from multiple public sources. Next, two separate approaches are pursued to address the issue of variations in population density. In the first approach, three cartogram algorithms are explored, with a goal of constructing a game map that resembles a distorted geometry of the state. Unfortunately, this first approach creates user-unfriendly game maps that do not intuitively correspond to the states' original geometries. Due to the shortcomings of the first approach, a second approach is attempted, wherein the problem of constructing a human playable game map is formulated as an integer linear program and then solved using the GNU Linear Programming Kit. This approach yields more intuitive geometries and a much more user-friendly game map.

D. Contributions

During the work of this semester, we gathered data, constructed the mapping to the data, and constructed the interface. Data from three heavily gerrymandered states is preprocessed and human-playable game maps are produced. The maps produced represent a bijective mapping between atomic voting precincts and the human-friendly game grid. Resultant game maps are imported into Javascript and a first scratch of the non-geometric protocol is implemented, allowing humans to play a game against another human, alternating choosing district allocations and freezing districts on the game map.

II. DATA COLLECTION

Data are collected from the three most gerrymandered states: Wisconsin, Pennsylvania and Michigan. Pennsylvania data are obtained from the Harvard Election Data Archive [1]. For Wisconsin, data are sourced directly from the Wisconsin Legislative Technology Service [3]. For Michigan, data are obtained from Dave's Redistricting App, a free web-based redistricting tool maintained by David Bradlee [2].

The data are preprocessed, with voter regions of roughly equal populations divided into atomic units, or "precincts." Next, the geometries of the shapefiles simplified as much as is possible without changing the topological relationships between the precincts. This preprocessing allows the files to be more efficiently used in computational geometry and cartogram algorithms, which tend to become computationally intractable if the input shapes are too complex.

III. APPROACH 1: A CARTOGRAM APPROACH

A. Background and Motivation for a Cartogram Based Approach

In recent decades, there has been an increased interest in representing real world electoral data from journalists, educators, and other groups. While much of the work of this area has been within the realm of designers, a small research community consisting mainly of computational physicists, geographers, cartographers and computer scientists publish research on how to algorithmically generate such representations. At first glance, Cartogram based approaches seem especially attractive for this application because they open the possibility directly applying the work of area experts, allowing us to avoid reinventing the wheel for a task non-central to the main goal of the project.

The inspiration for our cartogram based method is simple: if we can successfully use an existing approach to construct an equal area cartogram built off the population of voter precincts, we can treat the resulting geometry as having a roughly uniform population distribution, allowing the cartogram geometry to be directly used as a game map.

By using a cartogram algorithm to resize our atomic units, or "precincts", to be of equal area on a human playable game map, we can easily create distorted maps for the user to play on. At the end of the game, we transform the map back into the original state to show how the result's of the user's game-play corresponds to an actual districting.

B. Methods

Three cartogram algorithms were selected for exploration:

- Rubber Sheet Distortion Method [4]
- Carto3F [8]
- Mathematical Morphology-Based Cartograms [7]

Each approach is selected for a specific property that suggests viability for our application. Dougenik's Rubber Sheet Distortion Method is chosen due to its well-studied and intuitive nature, which initially suggests that it will be amenable to parameter-tuning to make more aesthetically pleasing cartograms. The second approach, which uses Sun's Carto3F, is explored for its strong topological preservation guarantees, which could allow for stronger theoretical guarantees if the approach produces a viable cartogram. The third approach, which uses Sagar's Mathematical Morphology-Based Cartograms, is investigated as a result of its unique properties, which allow for preservation of the global geometry of the state at a tradeoff for much weaker local topological guarantees.

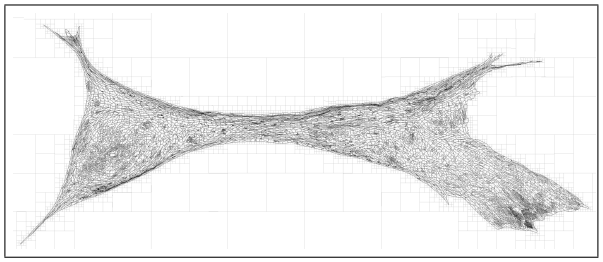

Fig. 2. Carto3F results on 2010 PA gubernatorial data

In *Fig. 2,* we see one of the major difficulties presented when using Cartogram algorithms on population data. The resultant geometries only vaguely resemble the original geometries. Additionally, the actual shape of the individual atomic units become extraordinarily oblong, making it difficult to determine which bucket of precincts they belong to. Both of these cause challenges when constructing a game map from the geometry.

C. Results

For each Cartogram, a game map was produced by directly layering a game grid on top of the produced Cartogram.

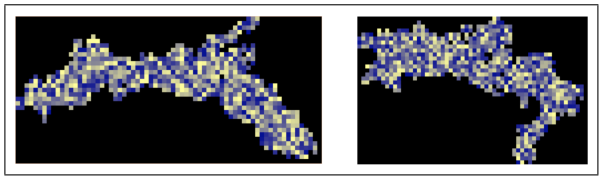

Fig. 3. Best looking game maps produced using (left) Carto3F and (right) Rubber Sheet Distortions, constructed from 2010 Pennsylvania gubernatorial data

As we can see from *Fig. 3*, the game maps produced are not very human playable, and they do not intuitively correspond to their original state's geometries. This suggests that a different approach is needed if we are to create an aesthetic interface that accurately represents the real-world data.

One of the major challenges in producing a gamemap with a cartogram is the lack of control over the behavior of the cartogram algorithm. Another challenge was the large amounts of time necessary necessary tuning numerically "stiff" contraints, each of which could cause the resultant geoemtery to change wildly at the smallest perturbation.

IV. Approach 2: An MILP Approach

A. Motivation for Formulating the Problem as a MILP

In the cartogram method, we attempted to force an algorithm to produce an aesthetically pleasing result from the data, rather than starting with an aesthetically pleasing result and working backwards. The difficulties faced with the cartogram approach suggest that we may have better experiences with an approach that gives us more control over the final game map.

This leads us to suspect that fixing the final game map *a priori*, and then formulating mapping the data to this game map as an optimization problem will allow us to gain a greater degree of control over the outcome. Such an approach also would have the benefit of allowing us to continuously refine our constraints until we come across a set of constraints that best fit our purposes, and to remove constraints which cause unpredictable behavior.

B. Methods

First, $r * c$ total precincts are chosen as centers according to the "eyeball heuristic," where the final game grid consists of r rows and c columns. The eyeball heuristic is a very informal heuristic that treats the state as roughly a grid and distributes the centers in accordance with the density of precincts, with slight variations when encountering the weird geometries that can occur at the edges of states.

Let $n = r * c$ be the number of total buckets, where each "bucket" has exactly one "center," and m be the number of precincts.

We construct a mixed integer linear program with the following constraints:

$$i \in [0,n], \sum_{0}^{m} p_{i,j} < 1.05(\frac{m}{n}) \tag{1}$$

Where $p_{i,j} = 1$ if j in bucket i, and 0 otherwise.

We then minimize the maximum sum of the distance in the precinct graph (integral) to the bucket centers via the addition of min-max auxiliary constraints.

It is interesting to note that we had initially formulated the MILP without the eyeball heuristic, doing so through the addition of constraints to force the program to solve for m*n centers with topological relationships to their neighbors resembling that of a grid. However, this approach dramatically increased the number of constraints required and we were unable to get the program to solve with such a large number of constraints. Fortunately, empirical results formed from several different selections of centers using the "eyeball heuristic" suggest that the actual result of the MILP is fairly agnostic to the choice of centers. Indeed, it seems "centers" may be chosen rather arbitrarily without a serious impact on the produced map.

The constraints were created in Python using PuLP and solved using GLPK. After some tweaking of the parameters, and post-processing of the resulting game map to make the selected regions more contiguous (but still equivalent solutions), we cross reference the assigned precincts with their voter data and obtain a nice game map that corresponds well to real-world electoral data.

C. Results

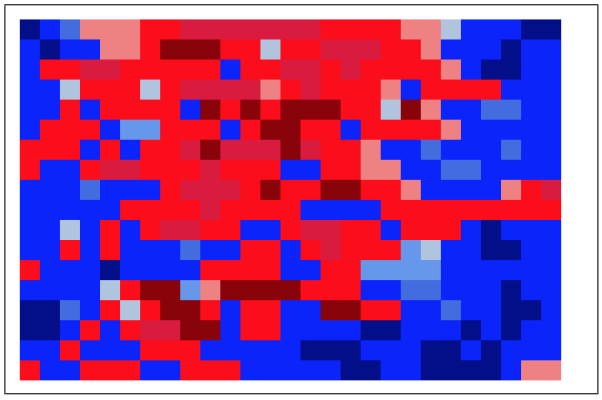

Fig. 4. Image of the final game map for Pennsylvania, a 26x18 grid. On the ith turn of the protocol, a user's turn consists of selecting $18-i$ districts, with each district consisting of 26 squares from the rectangular grid. Because of the rectangular nature of the grid, intuitive drag selection tools can be provided to the user, allowing for a quickpaced and fun game play.

From *Fig. 4*, its clear that the game map resulting from this method is superior to the cartogram method for a number of reasons, including that it:

- Intuitively corresponds to its state's electoral data. In *Fig 4.*, we see the deep-Blue areas consist of Philadelphia, Pittsburgh and other metropolitan areas occurring relatively close to where you would expect, with the more Republican regions occupying central P.A.
- Allows for human-friendly rectangular drag selection tools when selecting districtings. This is critical, as a typical game will consist of as many rounds of there are congressional districts. For a state like Pennsylvania containing 18 districts, during the course of the game a user would have to select $\sum_0^{18}(18-i) = 171$ individual districts. Each district consists of 26 squares. Thus, selecting these districts individually would be an exceedingly frustrating user experience, consisting of individually clicking selecting 4446 squares.
- Contains stronger guarantees about the topological properties of the game grid. Because we can add, remove, tighten or loosen our constraints, and because we know the value of the resulting objective function, we can make provably correct assertions about the properties of this mapping.

V. Surprises and Lessons Learned

A. Surprises

It was very surprising that the issue of creating human playable game maps off electoral data was not solved robustly. Although there are a few greedy methods, there were no existing approaches that both preserved geometric locality while allowing for the creation of a human-friendly interface. Additionally, it was very surprising that the cartogram methods worked so poorly for this application. Both of these surprises combined to significantly increase the amount of time it took to correctly identify a way to construct an appropriate game map.

During the course of the research project, there were a surprising number of unanticipated questions that popped up. For instance, it was not realized until rather late in the semester that such a game-map based approach would only work for the geography-agnostic simulation of the protocol. If users were given the constraint that they had to form geographically contiguous districtings, it is likely that users would either create invalid districtings immediately, or accidentally create districtings that box themselves in to an invalid districting during a later round in the game. In order for such methods to be applied to districtings without requiring contiguous districts, we would need the naive assumption that a user will properly and presciently select valid districtings. Realistically, this means that

we will have to build some way of snapping users' selection to the closest valid districting upon the end of every turn or the completion of the game. I have been exploring methods on how to do this for the past 2-3 weeks, but currently do not have any results to present for this challenge.

B. Lessons Learned

This project has reinforced the idea that methodical planning is necessary for research. Specifically, it has taught me that I should shallowly explore multiple options before planning out a full semesters work, in order to avoid unforeseen surprises such as the lack of fitness of a specific approach for a problem. Additionally, in future tasks that require UX design, I will make sure to consider the entire scope and purpose of a task. This would help avoid situations where the research goal is technically completed, but the project goal is inadequately fulfilled, as was the case when producing a game map using the Cartogram methods.

Another lesson learned is that, when working on a large project, one should select the language or framework most appropriate for the task, rather then select the language or framework with which one is most familiar. I had to switch over from React because I realized that it was not appropriate for the building of a simulation game. This cost me a big chunk of time and taught me a painful lesson.

During the course of this semester, I learned a lot about computational geometry, constraint satisfaction and the future needs of the project. In addition to building my knowledge on cartogram and computational geometry algorithms, I learned how to use Python's PuLP and became familiarized with formulating problems as mixed integer linear programs. In terms of future needs of the project, we discovered a need for a robust solution to the issue of players constructing invalid game maps.

VI. CONCLUSIONS

During the first part of this three-semester senior thesis project, we explored how to best construct a human-interactive game map off real-world electoral data. At the outset of this semester's portion, there were three major goals: (1) collect and preprocess real world data, (2) establish a bijective mapping between said real world data in order to construct a human-friendly map and (3) construct an interactive interface using this human-friendly map.

In order to accomplish (1), we collected data from several publicly available sources and preprocessed them for use in both our cartogram and optimization algorithms. In order to accomplish (2), we explored two classes of methods. We first attempted to solve the problem with cartograms, but realized that doing so produced human-unfriendly game maps. In the face of this major setback, we next attempted to solve the problem by formulating it as a MILP, which resulted in significantly nicer results. In order to accomplish (3), we exported our map produced in (2) into a JavaScript app where we implemented the first scratch of the protocol for human-human play.

VII. FUTURE WORK

In the shorter term, we will continue to investigate this new question related to the issue of "snapping" to the closest valid districting for the geometric case. This is a non-trivial problem, and initial ideas include formulating it as an ILP or somehow sampling the space of valid districtings and finding the closest one as per some distance metric.

Over the next two semesters, we will begin the next phase of the project, which seeks to answer questions related to game theory and computational social choice, as opposed to computational geometry and cartogram generation. As part of the research task for the next two semesters, I plan to:

- Construct an artificial agent capable of efficiently playing the game with some reasonable ability. Potential early approaches include NEAT or CFR. Later approaches could incorporate more advanced methods, such as bootstrapping a machine learning method using human gameplay data
- Launch a website using this agent, for which users can play against on the interface constructed during this semester.
- Analyze the data produced from users gameplay against the artificial agent (and possibly even against other)
- Investigate the real world effects of redistricting protocols to determine if such protocols have any unforeseen side effects or benefits (e.g. how the protocols affect minority representation)
- Examine the practical effectiveness of the redistricting protocols in preventing gerrymandering

Clearly, there are many open questions related to this work. Many of these questions relate to the properties of redistricting protocols when run on real world data, as well as an exploration on how decent of an AI can be initially constructed to play these protocols. Some of

these questions relate to how users will react to merely learning about the protocol.

Equally important as my hope for interesting research results is my hope that creating a human-friendly interface will attract public attention to the role partisan districting protocols can play in producing fairer electoral outcomes. Idealistically, I believe awareness of these protocols will spur their adoption, or at least help force political action on some of the more egregious cases of gerrymandering.

ACKNOWLEDGMENTS

I'm thankful for my research advisor, Ariel Procaccia, for his guidance, insight and ideas. I'd also like to thank Wesley Pegden for his guidance navigating the difficult process of collecting and cleaning electoral data. I'd like to thank Shipeng Sun, for chatting with me on the phone to help me better understand Carto3F and the intuition lurking behind it. Finally, I'd like to thank Johnathen Aldrich and Todd Mowry for providing a supportive and structured environment for student research via 15-400.

REFERENCES

[1] Stephen Ansolabehere and Jonathan Rodden. Pennsylvania data files, 2011.

[2] Dave Bradlee. Mi - gov., ag, sos 2006.

[3] Legislative Technology Service Data. Wisconsin 2010 election dataset.

[4] James A. Dougenik, Nicholas R. Chrisman, and Duane R. Niemeyer. An algorithm to construct continuous area cartograms. *The Professional Geographer*, 37(1):75–81, 1985.

[5] Gregory Naigles. *U.S. Presidential Election Results*.

[6] W. Pegden and A. Procaccia. A Partisan Districting Protocol with Provably Nonpartisan Outcomes. *Working Paper*.

[7] B S Daya Sagar. Cartograms via mathematical morphology. *Information Visualization*, 13(1):42–58, 2014.

[8] Shipeng Sun. An optimized rubber-sheet algorithm for continuous area cartograms. *The Professional Geographer*, 65(1):16–30, 2013.