

Master Use Case List (12/15/24)

CSE416, S03 – Fall 2024

Terminology

- Districting (District plan) – a collection of congressional districts in a state generated by the ReCom algorithm. Each such districting will be a random and will be a subset of all of the possible graph partitions that are constrained by the user-specified constraints on population equality and compactness.
- Population – the population of a region (e.g., state, precinct, etc.) refers to the total population as defined by the US Census Bureau. Some calculations might refer to the voting age population (VAP) or citizen voting age population (CVAP). You can use any measure of population, but it should be done consistently throughout the application.
- Ensembles – the collection of district plans generated on the SeaWulf. There will be at least one ensemble for SMD (single member districting) and MMD (multi-member districting).

Notation

Use cases listed below include a categorization following the use case title. Three categories of use cases are provided as “required,” “preferred,” and “optional.” Use cases with an “SD,” “AD,” and “part of GUI SD” indicate that the use case might be requested in the design review with “SD” referring to a sequence diagram and “AD” referring to an activity diagram. For some use cases, the GUI and the server part can be combined into one use case diagram. Many of the use cases that are related and that require activity diagrams can and should be combined into one activity diagram.

General GUI (11 required)

GUI-1. Display a pan-able and zoom-able map of the US (required)

The initial page will have a pan-able and zoom-able map, covering the entire United States. It will have a max zoom-out level and a bound on how much the user can pan such that the User is guided to stay near the selectable states. Note: this map does not need to be pan-able and zoom-able if the state map is pan-able and zoom-able.

GUI-2. Select state to display (required) (SD)

User can pick a state through a dropdown menu and also through clicking on the state in the map of the US. The state selection will cause the map display to show the state centered in the GUI map area and at a zoom level so that it nearly fills the map area. The selected state should display the current district plan.

GUI-3. Map view filter (preferred)

The GUI should employ a filter that allows selected boundaries to be displayed (or not displayed). The filter should include the current district plan as well as any random district plans available on the server. Multiple filters can be set, so the corresponding boundaries will be displayed simultaneously. Line color and thickness should be set so that the display of multiple boundaries can be easily understood by the user. The demographic heatmap (GUI-XX) will also be available in the filter (if calculated).

GUI-4. Display summary of SMD/MMD ensembles (required) (SD)

A summary of the SMD and MMD ensembles will be displayed, either as a default part of the GUI or in response to a user selection. For each ensemble, the summary will include the number of district plans, average number of minority representatives per plan, average equal population measure, average Republican/Democratic split, and other implemented measure values. The MMD ensemble summary will include the layout for the state (e.g., 3, 3, 4).

GUI-5. Display whether an MMD district is 3/4/5. (preferred)

When a MMD district is displayed on the map, the user should be able to determine the number of representatives for each district. This can be done with different colors for districts, a tool tip, or some other interface approach.

GUI-6. Display detailed election data for ~~an interesting random plan~~ **Simulated Elections (preferred)**

The user should have the option to view the detailed election results of an “interesting” simulated election in an ensemble. Details include the district number, number of representatives for that district, party affiliation, etc. **The plan will be selected from among those shown in GUI-10.** The user should be able to view the results for all candidates, winners, and losers, along with vote totals.

GUI-7. Display results of MMD simulation with unequal number of candidates (preferred)

Within the MMD ensemble, compare the results of the MMD election simulation for an equal number of candidates with the results for an unequal number of candidates. For example, one of your election simulations might have 3 Republican candidates and 5 Democratic candidates for a 3-representative district. The goal of this use case is to determine the impact of additional candidates for a political party.

GUI-8. Display the names of actual candidates (preferred)

Display the names of election winners and losers when the names of actual candidates are used in the election simulation.

GUI-9. Compare enacted plan results with the “average” MMD random district plan (required)

Display a table comparing the data for the enacted plan with the data for the “average” MMD random district plan. Items for comparison include Republican/Democratic splits, number of opportunity representatives, vote share, and seat share.

GUI-10. Display available district plans (required)

Some of the random district plans generated on SeaWulf will be available to a user in the GUI. The GUI will show a list of those plans in a menu so that the user can select an individual district plan to examine. The menu should identify each plan by its distinguishing feature (e.g., extreme Republican seat split).

GUI-11. Display summary of a random district plan (required) (SD)

When the user clicks on a district plan shown in the menu of available plans, details about the selected district plan will be displayed. Details include number of districts and summary of each district in the plan (including population and population by demographic group), Republican/Democratic split, election from which voting preference is used, number of opportunity districts, threshold used for opportunity district calculation, and number of safe districts. **The safe district entry is only applicable to SMD district plans.**

GUI-12. Map display of selected district plan (required)

In conjunction with the Display Plan Summary use case above, any of the individual districts in the district plan will be shown on the map. A mouseover or area click feature will cause the details of a given district to be displayed (e.g., district number, estimated vote split, etc.). For an MMD plan, the details will include the political party of each elected representative for that district.

GUI-13. Display SMD seat share / vote share plot (preferred) (SD)

Display the seat share / vote share plot for a selected SMD district plan. The displayed curve will have been previously generated from a simulation of voting results. Annotation for the plot at a minimum will include values for symmetry, bias at 50%, and responsiveness. The plot should also include highly visible dots for the vote/seat share of both ensembles (SMD and MMD). For the two dots, the vote share is the vote share of the party (e.g., Republican) for the entire state and the seat share is the average percentage of seats for that party among all the random district plans. For example, for a 15-seat state, if 2,500 sample plans had 8 Republican representatives and another 2,500 plans had 7 Republican representatives, the average number of Republican representatives would be 7.5 and the average Republican seat percentage would be 50%.

GUI-14. Opportunity district threshold sensitivity (preferred)

Display a graph of the relationship between the opportunity district threshold and the percentage of opportunity districts. **Recall that the threshold is the percent of some racial group in a district sufficient to elect a representative of that group.** This should be done for both SMD and MMD. **For SMD, calculate the number of opportunity districts in each plan with threshold values from 37% to 50%. The MMD chart would be similar, but separate. Remember, the threshold for an MMD is smaller and based on the number of representatives in the district. For example, the threshold for a 5-representative district might be 20% , and two minority representatives might be 40%. You can show the y-axis either as a percentage or an absolute number of opportunity districts.**

GUI-15. Display summary data from SMD and MMD ensembles (required) (SD)

Display the data previously calculated on the SeaWulf for both the SMD and MMD ensembles. Displays should include bar charts for range of opportunity representatives, and range of Republican/Democratic splits. The display should be a bar chart. Display should be supplemented with text that shows the vote share (constant for all plans in the ensemble) and the average seat share.

GUI-16. Display SMD box & whisker data (required) (SD)

The user will be able to request the display of box & whisker data for an SMD SeaWulf ensemble of district plans (e.g., 5,000). The user will be able to select the basis for comparison (e.g., African American population percent, Democratic population percent, Republican population percent, etc.). The display should be sufficient in size to show your largest state and should include a legend and color selection to make the chart easily readable. The enacted district plan should be included as dots for each of the x-axis districts.

GUI-17. Display MMD box & whisker data (required) (SD)

The user will be able to request the display of box & whisker data for an MMD SeaWulf ensemble of random district plans (e.g., 5,000). The user will be able to select the basis for comparison (e.g., African American population percent, Democratic population percent, Republican population percent, etc.). The display should be sufficient in size to show your largest state and should include a legend and color selection to make the chart easily readable. There should be some indication in the graph that shows the number of representatives in each district.

GUI-18. Display box & whisker data for both SMD and MMD (preferred)

One chart will show data for both the SMD and MMD ensembles in the state. The chart will be organized in a way that allows a direct comparison between both ensembles.

GUI-19. Display actual representatives (preferred)

When any district plan is displayed (e.g., enacted or SeaWulf generated random plan), the names of the actual representatives (for enacted) or simulated winners (random plans) will be displayed. The display can either be visible in a mouseover or in a separate table.

GUI-20. Display variations in proportionality by seat layout (preferred)

A research paper suggested that the MMD scheme used in the FRA (mix of 3, 4, and 5 member districts) permits a greater degree of gerrymandering as compared with a MMD plan with a narrower range of representatives per district (e.g., 2). Generate a display that shows this effect.

GUI-21. Reset page (preferred)

When the user clicks a reset button, the GUI will reset to the condition before the user selected a state.

GUI-22. Compare a sample plan with the current plan on the map (preferred)

Show one sample plan and the enacted plan, either simultaneously or with a quick toggle between the two maps. Visual design of the interface should allow the user to easily compare the geographic layout of the two plans.

GUI-23. Demography heatmap (preferred)

The color of each district in a map display (e.g., interesting random district plan from the SeaWulf) will vary based on the concentration of the user's selected demographic group or political party in that area. A legend will show the percentages associated with each color in the heatmap. Color variations in the legend (and heatmap) will be based on the thresholds used in the calculation (e.g., one color for below 50% in an SMD and a different color used for above 50%). The user can set the demographic (e.g., African American) enacted and the selected demographic will be displayed in the chart.

GUI-24. Data Sources and Additional Information (optional)

There should be a clear way for the user to learn of the data sources used to create the application. Allow the user to independently evaluate/validate the data that is shown on the application. Likely implementation will be through a tab at the top or bottom of the application labeled clearly (e.g., “Sources”, “Data Sources”, “References”).

GUI-25. Precinct based heatmap (optional)

As a variation on GUI-23, the GUI can include a precinct-based heatmap with increments of population percentage for differing colors being in the range of 5%-10%. If both this use case and use case GUI-23 are implemented, there should be a toggle to allow the user to see either of these options (but not both simultaneously).

GUI-26. Display of MMD ensemble summary per layout (optional)

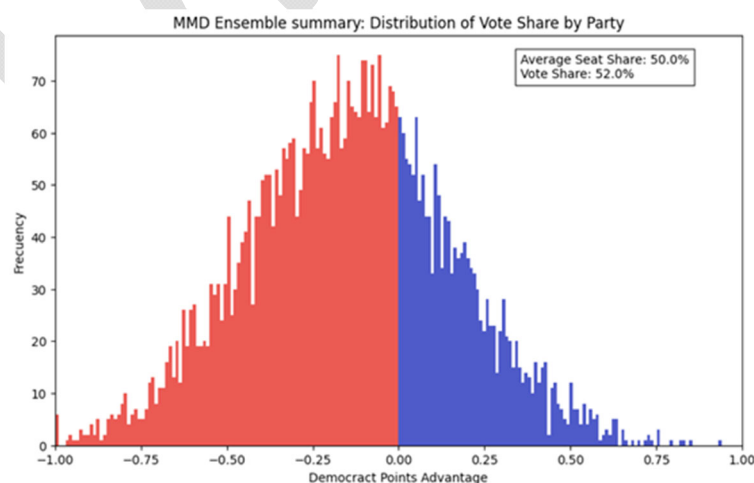
This use case is an extension of GUI-15. Display the data previously calculated on the SeaWulf for the MMD ensembles should be broken out by layout. Displays should include bar charts for range of opportunity representatives, and range of Republican/Democratic splits. The display should be a bar chart. Display should be supplemented with text that shows the vote share (constant for all plans in the ensemble) and the average seat share.

GUI-27. Display theme customization (preferred)

This use case will provide theme customization options that allow users to switch between light mode, dark mode, and system theme preferences. The default setting should match the user's system theme preference, and should be toggleable based on user selection. When the theme changes, all GUI elements including maps, charts, district visualizations, and data displays should automatically adjust their colors and contrast levels to maintain readability.

GUI-28. Ensemble distribution vote share win by party (optional)

This visualization chart (a sample of which is shown below) simplifies the comparison of party performance across 5,000 district plans by focusing on the point difference in vote share between Democrats and Republicans. Negative values indicate where Democrats perform worse than Republicans, while positive values show where Democrats outperform them. By emphasizing relative performance rather than absolute percentages, the chart makes it easier to identify trends, such as consistent margins of victory or loss. This approach helps users quickly assess the competitiveness and fairness of district outcomes across all simulations.



Preprocessing (5 required)

Prepro-1. Integrate multiple data sources (required) (AD)

Integrate and merge US Census data (population, both for total and for the required minority groups), precinct data (boundary, name, demographics, etc.), and existing district data (boundary, name, district#, etc.). Geographic boundary data should be converted (if necessary) to a consistent format (e.g., GeoJSON).

Prepro-2. Identify precinct neighbors (required) (AD)

Identify two precincts as neighbors if they share a common boundary of at least 200 feet and the edges of each precinct are within 200 feet of its neighbors' edges. If possible, try to locate a data source for which this computation is already done or use the maup library.

Prepro-3. Integrate approved plan with dataset (required)

Integrate the approved plan for the state with the data set.

Prepro-4. Store preprocessed data (required)

The preprocessed data should be stored in the NoSQL or relational database. If a relational database is used, the data should be stored in third normal form. Data might also be stored in a file system accessible to the server.

Prepro-5. Store SeaWulf data (required) (AD)

Retrieve generated data from the SeaWulf for each of your states, convert to an appropriate format, and store either in your database or in a file system. Data stored in a file system should be accessible through a path obtained from your database.

Prepro-6. Neighbor calculation using Delaunay triangulation (optional)

Use Delaunay triangulation to capture the relationship between closest points; this is justified by the 'empty circle' property, which ensures that no other point can lie on the circle defined by two neighboring points. This algorithm creates 'simplices' which is the set of points closest points (neighbors). I then mapped these related points and stored them back into the DataFrame, so that each district now stores its closest neighbors.

SeaWulf (10 required)

SeaWulf-1. Server dispatcher (required)

Establish a protected directory on the SeaWulf to store your team's data. Pre-stage any data that might be used repeatedly for SeaWulf runs. Prior to submitting a batch districting run request to the SeaWulf, the data required for the run should be marshalled (from memory and/or DB) and passed to the SeaWulf as a file (or multiple files) to be stored in the team's SeaWulf file system.

SeaWulf-2. Run MGGG Recom algorithm on the SeaWulf for SMD (required) (AD)

Set the constants in the MGGG code to define the properties (e.g., constraints) of the run. Any run-control information should be packaged in a SeaWulf acceptable format (e.g., script commands) and executed on the SeaWulf. Your activity diagram should demonstrate that you understand how the MGGG algorithm operates.

SeaWulf-3. Coordinate/aggregate SeaWulf core generated data (preferred) (AD)

You will run the algorithm on a single SeaWulf node, one that has multiple cores. Each of the cores will generate one or more random graph partitions (i.e., district plans) and store a concise version of those results in a shared file directory for your team. Following the completion of each random graph partition, the core begins the generation of the next random district plan. You will need to coordinate the work of the multiple cores so that when the target number of district plans is completed, each of the cores ends its processing.

SeaWulf-4. Determine MMD districts in each random district plan (required) (AD)

For each random SMD district plan calculated by the ReCom algorithm, determine a suitable random MMD plan. You can do this by “gluing” together individual SMD plans to form a MMD district. Each of the MMD districts should have a reasonable compactness value (e.g., Polsby-Potter algorithm) and be contiguous. Apply random selection as much as possible.

SeaWulf-5. Calculate SMD random plan election winners (required)

Using a winner-take-all approach and 2020 or 2022 election results, estimate the election results in each district of each SMD ensemble district plan. You will calculate this by summing up the election results in each node (i.e., precinct) of a partition sub-graph. You do not need to consider third party candidates.

SeaWulf-6. Simulate FRA election rules in the simulated election for MMD (required)

To predict the election results in a precinct under the FRA procedure, use a simple prediction model. The model will assume that voters will select candidates strictly according to party. For example, if there were 1,000 votes for a Republican candidate in the reference election (e.g., 2020 Presidential) and the MMD district had three representatives, three of the Republican candidates would be predicted to receive 1,000 selections in some arbitrary order of candidates. A similar process would be followed for MMD districts with 4 or 5 representatives. A similar rule would apply for Democratic vote totals. The arbitrary order of candidates would be fixed for all precincts in that MMD district. In this simple election model, you can assume that there are exactly n Republican candidates and n Democratic candidates in each n -member district.

SeaWulf-7. Calculate Republican/Democratic splits for each SMD random district plan (required) (AD)

For each generated plan in an ensemble, estimate the Republican/Democratic votes in each district. Since each district is a collection of precincts, use the historic precinct vote totals (e.g., 2020 Presidential) to estimate the winner of a Congressional election in each district.

SeaWulf-8. Simulate FRA election rules – unequal number of candidates (preferred)

Generate an alternate set of election results based on an unequal number of candidates. You can set the number of candidates in one party to n , but set the number of candidates in the other party to a quantity greater than N (e.g., $n+2$). Use an unequal probability distribution for the likelihood of a given candidate being selected. For example, in a 3-representative district, there might be 3 Republican candidates and 5 Democratic candidates. The candidate selection probability for the Democratic candidates might be (.4, .2, .2, .1, .1).

SeaWulf-9. Include candidate minority status in simulated election (preferred)

For MMD simulated elections, include racial/ethnic data for candidates when the minority population of a district exceeds the opportunity threshold (defined as a number sufficient to elect at least one minority candidate assuming racially polarized voting). Modify the voter/candidate selection probabilities so that a high percentage of the minority voters in the district vote for the minority candidate. For example, you might assume that 90% of the African American voters will list a Democratic African American candidate as their first choice.

SeaWulf-10. Use actual candidate names in election simulation (preferred)

Use the names of actual 2022 Republican and Democratic candidates in the MMD election simulation. For a given precinct, modify the election prediction model so that a high percentage of first choice votes are given to any incumbent candidate of the same party in that precinct.

SeaWulf-11. Calculate the number of opportunity (minority) representatives in a random districting (preferred) (AD)

For each of the minority groups of interest for your states, calculate the number of districts in a district plan with a minority population percentage in excess of a predetermined threshold. Note: you set the threshold, but in most cases, this should be 50% for an SMD plan. For an MMD plan, the threshold should be proportional to the number of representatives in the district. For example, a 5-representative district would have a 10% threshold. This calculation should be done for any population categories you are implementing in your system (total population, voting age population (VAP), and citizen voting age population (CVAP)). For a MMD district, you can use either the result of a simulated election that includes a minority voting factor or estimate the number of minority representatives based on the minority population of the district as a percentage of the number of representatives.

SeaWulf-12. Calculate safe representative seats for a random district plan (preferred)

For each random district plan (SMD and MMD), calculate the number of safe districts. A safe district in an SMD district plan is defined as one in which the expected margin of victory for a candidate is greater than some threshold (e.g., 10%). For MMD districts, there is no standard definition. You can develop your own approach or use a simple approach in which you determine for each district when all candidates of a given party will easily defeat any candidate from the opposing party.

SeaWulf-13. Identify the average random MMD plan (required)

Identify the “average” MMD district plan as a plan that has the average Republican/Democratic splits and optionally the average opportunity representatives as compared with ensemble-wide results. There are various approaches to satisfy this requirement. Perhaps the simplest is to select a plan that matches the measures of the ensemble (e.g., Republican/Democratic splits, opportunity districts, vote share / seat share).

SeaWulf-14. Identify and store additional random district plans of note (required)

Identify additional random district plans, both SMD and MMD, that are worthy of attention. For example, such plans might include extreme Republican, extreme Democratic, large number of opportunity (minority) representatives, small number of opportunity representatives, and plans with a seat share approximately equal to the vote share. Summary information and detailed information for each of these plans should be stored in your server database for eventual display by the user.

SeaWulf-15. Generate minority representative summary data (required) (AD)

For each generated district plan, determine the number of racial/ethnic minority representatives. For an SMD district, you can assume a minority candidate will be selected if the minority population of the district exceeds a predetermined threshold. For a MMD district, you can use either the result of a simulated election that includes a minority voting factor or estimate the number of minority representatives based on the minority population of the district as a percentage of the number of representatives.

SeaWulf-16. Generate an ensemble using a 2-rep MMD approach (preferred)

Modify the MMD district generation rule to limit a MMD district to 2 representatives. If the number of representatives is odd, there will be one 1-member district, otherwise all districts will be 2-representative districts.

SeaWulf-17. Run on multiple Seawulf nodes (optional)

Run the MGGG algorithm and supplemental code on multiple nodes on the SeaWulf. Node coordination should be done using a suitable coordination mechanism (e.g., MPI). Be able to estimate speed-up during your final project presentation. Note that this is optional since your algorithms should run effectively on a large-core single node processor.

SeaWulf-18. Python profiler (preferred)

Profile your algorithm performance on the SeaWulf using a Python profiler tool. Identify the procedures that consume the most CPU time. Results can be displayed using some Python-appropriate tool, and displayed as an image in your final presentation.

SeaWulf-19. Calculate box & whisker data (required) (AD)

Calculate the box & whisker summary data for all the random district plans generated by the SeaWulf. These calculations will be for Republican population share, Democratic population share, and minority population percentage for any minority population that might require at least one majority-minority district in any of the team's states. This should be calculated for both SMD and MMD district plans.

Supplemental Processing (1 required)

Server-1. Generate summary data for a district plan (required)

Calculate the information that summarizes the data contained in the SMD and MMD ensembles. At a minimum, the data should contain all the information needed to display a district plan and an ensemble summary in the GUI.

Server-2. Calculate range of values for the seat/share plot for a sample district plan (preferred)

Calculate the range of values used in the seat/share plot by modifying the total votes for party in a district by applying a random variable to the vote total for each party. Use an approximation of the Gelman/King algorithm. **Alternatively, this can be performed in preprocessing so the server just returns the previously calculated values.**

Additional Use Cases

DRAFT