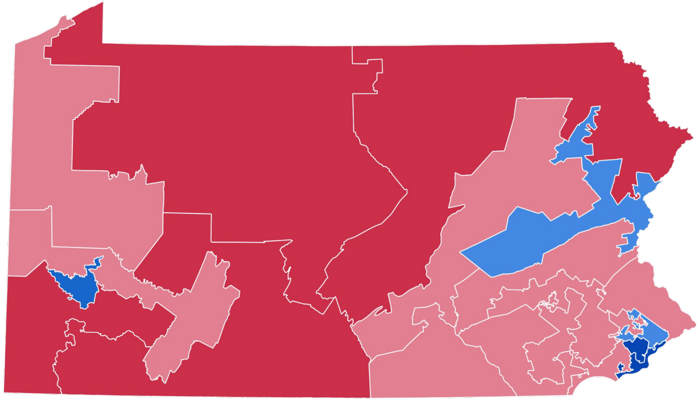
Automating Redistricting using an Algorithmic approach

Final Project Presentation
Deven Hagen
5/21/25
Systems Lab - Yilmaz 1

Problem

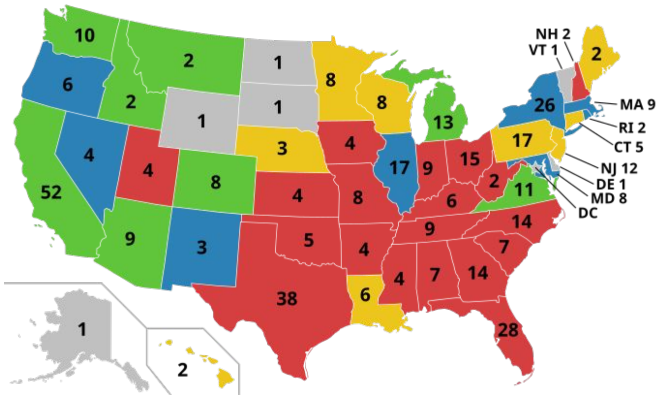
- States get delegates to the House proportional to their population
- State divided up into single-member districts
- Subject to partisan and racial gerrymandering [11]
  - Partisan: maps that give certain parties more power
  - Racial: maps that dilute the influence of certain demographics



Party	Republican	Democratic
Last election	**12**	7
Seats won	**13**	5
Seat change	▲ 1	▼ 2
Popular vote	2,710,070	**2,793,538**
Percentage	48.77%	**50.28%**

Background

- Reason for bias is partisanship of state legislatures
- Independent Redistricting Commissions (IRCs) established to combat this
- Humans still have bias - can we come up with a truly unbiased solution using computers?



Partisan control of congressional redistricting after the 2020 elections, with the number of U.S. House seats each state will receive.

Other Solutions

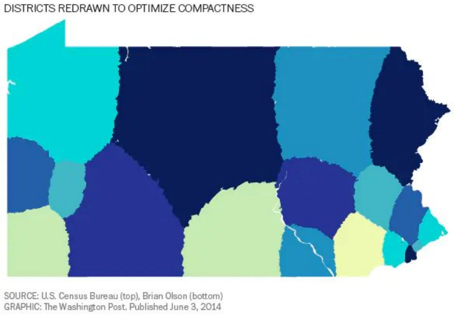
Two main types - **heuristic** and **ensemble**

Heuristic	**Ensemble**
● "Heuristic" that ranks maps based on various criteria ● Can be thought of as optimization ● Result is a single map with a high value for your heuristic	● Algorithm that creates random maps ● Can be used to compare a map to a sample of the population of all maps ● Can select maps that best fit your criteria

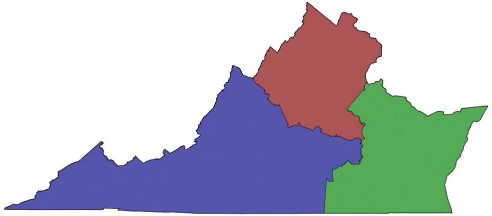
Other Solutions

- Heuristic:
  - Brian Olson's compactness-based approach [9]
  - Auto-Redistrict: genetic algorithm [7]

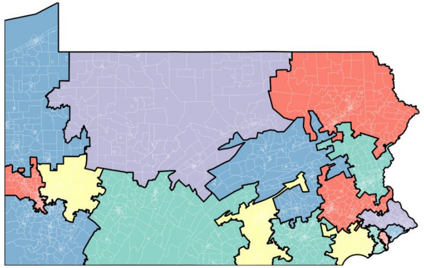
- Ensemble:
  - Haas et al. - Seed-Fill-Shift-Repair (SFSR)[6]



Brian Olson's



Auto-Redistrict

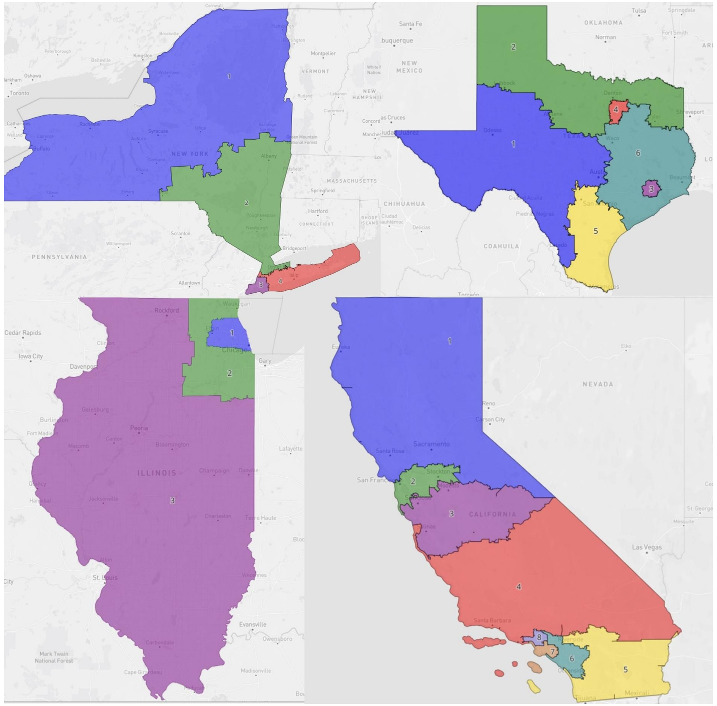


SFSR

Why Is Ours Better?

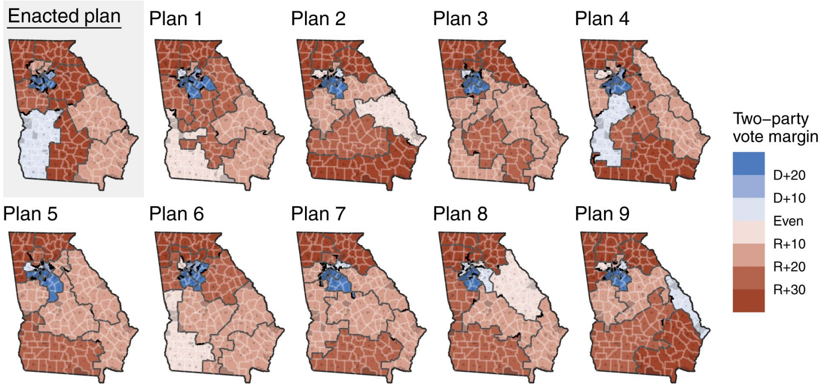
- Novel heuristic that incorporates a wide variety of metrics

- Can compare multi-member options as well
  - **Single-member:** each district elects one delegate to Congress
    - Default
  - **Multi-member:** each district elects multiple delegates to Congress in the same election



Novelty

- First to use heuristics comparing single and multi-member options
- Heuristic is novel and incorporates partisanship, compactness, county splitting, and racial bias



Impact

- Independent redistricting commissions can use algorithm
- Allows for fairer maps with less racial and partisan bias
- Provides impartial but not binding guidance

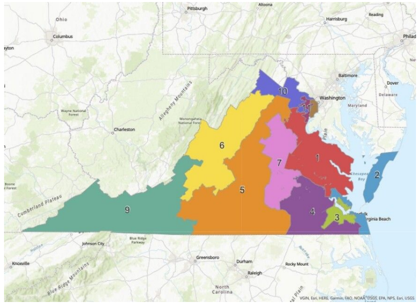
MICHIGAN
INDEPENDENT
CITIZENS
REDISTRICTING
COMMISSION

Method

- Input: data from Census and Department of Elections [10]
- Transform/preprocess data
- Run algorithm
- Output: a congressional district map, with all precincts assigned

GEOID20	DISTRICT	NAME	ADJPOP	CVAP	WHITE	HISPANIC	BLACK	ASIAN	NATIVE	PACIFIC	D_VOTE_S
5.1E+10	100	Chincoteag	3357	2800	2660	75	10	0	24	0	0.34589
5.1E+10	100	Atlantic	1519	1271	1147	47	181	3	3	0	0.276382
5.1E+10	100	Greenbackv	2890	1746	1669	53	46	13	0	0	0.309561
5.1E+10	100	New Churcl	3590	2738	1167	76	1347	92	0	0	0.553127
5.1E+10	100	Bloxom	1384	949	831	4	117	1	21	0	0.330986
5.1E+10	100	Parksley	1742	1041	836	39	184	4	20	0	0.378378
5.1E+10	100	Saxis	411	302	268	4	18	0	0	0	0.313725
5.1E+10	100	Mappsville	1806	1375	627	13	578	3	70	0	0.62521
5.1E+10	100	Rue	1922	1142	327	157	611	0	10	0	0.733624
5.1E+10	100	Accomac	2898	2097	1345	106	649	14	3	0	0.432487
5.1E+10	100	Tangier	440	315	315	0	0	0	0	0	0.192568
5.1E+10	100	Nandua	3720	2987	2073	212	667	4	45	0	0.440341
5.1E+10	100	Bobtown	1156	963	824	18	98	28	2	0	0.409396
5.1E+10	100	Melfa	1921	1297	732	28	429	22	8	0	0.520384
5.1E+10	100	Wachaprea	1009	878	663	30	140	0	4	0	0.336299
5.1E+10	100	Painter	3773	2994	1296	29	1743	4	8	0	0.627647
5.1E+10	100	Woodbrool	4043	3235	2440	35	569	174	14	0	0.693621
5.1E+10	100	Branchland	2395	1965	1580	35	325	0	14	0	0.757196
5.1E+10	100	Agnor-Hurt	4153	2616	1607	156	805	53	11	0	0.767365
5.1E+10	100	Dunlora	4419	3555	3155	70	195	130	0	0	0.750551
5.1E+10	100	Northside	5516	4593	3642	242	536	164	0	0	0.617262
5.1E+10	100	Jack Jouett	4982	3523	2974	127	309	101	13	0	0.643891
5.1E+10	100	University	7912	7910	5375	385	867	1259	22	0	0.852132

Algorithm



Method

- Use Python
- Parse .csv and .shx using Pandas, GeoPandas
- Project geospatial data so area calculations are correct
  - Conus Albers projection
- Use GerryChain to turn map into a graph and run algorithm
- Use matplotlib to visualize map created

Method: GerryChain

- Python library for using Markov Chain Monte Carlo methods (MCMC) with redistricting
- Stores map as a graph, decreasing runtime
- Allows for quick initial partitions
- Built-in *metrics* library

Method: GerryChain

- Can read in a GeoDataFrame or a .json file
- Input election data and population column
- Create the random initial partition

```
graph = Graph.from_json("va_crs_corrected_data.json")
graph.to_json("va_crs_corrected_data.json")

election_names = ["COMP16-20"]
election_columns = [["E_16-20_COMP_Dem", "E_16-20_COMP_Rep"]]
pop_col = "ADJPOP"

myupdaters = {
    "population": updaters.Tally(pop_col, alias="population"),
    "cut_edges": cut_edges,
    "polsby-popper": polsby_popper
}

elections = [
    Election(name,{"Democratic": dem, "Republican": rep})
    for name, (dem, rep) in zip(election_names, election_columns)
]
election_updaters = {election.name: election for election in elections}
myupdaters.update(election_updaters)

total_population = sum([graph.nodes[n][pop_col] for n in graph.nodes])
assignment = recursive_tree_part(
    graph,
    range(NUMDISTRICTS),
    total_population/NUMDISTRICTS,
    pop_col,
    0.05
)

initial_partition = GeographicPartition(graph, assignment, myupdaters)
```

Method: GerryChain

- Create proposal
- Use the *SingleMetricOptimizer* library to optimize the value of your heuristic
- Choose method:
  - Short bursts
  - Simulated annealing
  - Tilted runs
- Export new map

```
myproposal = partial(
    recom,
    pop_col=pop_col,
    pop_target=total_population/NUMDISTRICTS,
    epsilon=0.05,
    node_repeats=2,
)

heur = lambda x: 20*(abs(x["COMP16-20"].efficie

optimizer = SingleMetricOptimizer(
    proposal=myproposal,
    constraints=myconstraints,
    initial_state=initial_partition,
    optimization_metric=heur,
    maximize=False
)
total_steps = 1000
```

```
data = [["GEOID20", "District"]]
for n in list(optimizer.best_part.graph.nodes):
    data.append([optimizer.best_part.graph.nodes[n]["GEOID20"], optimizer.best_part.assignment[n]])
with open("final.csv",'w', newline='') as f:
    writer = csv.writer(f)
    writer.writerows(data)
```

Method: Heuristic

- Common considerations in redistricting process:
  - Population balance
  - Contiguity
  - Compactness
  - Partisan bias
  - Racial bias
- Compound statistic:
  - Compactness
    - Polsby-Popper
  - Partisan bias
    - Efficiency gap, Mean-Median difference, “partisan bias” metric
- Correlation between racial and partisan bias

Heuristic:

20 * the average of the three partisanship metrics + 1 - the average district Polsby-Popper score

Goal is to minimize the above

Results

- Can measure maps created using a variety of metrics
  - Efficiency Gap
    - Calculates number of “wasted votes” for each party
  - Polsby-Popper Score
    - Measure of compactness - circle has highest score
- Use Dave’s Redistricting or PlanScore to easily calculate metrics

District	A votes	B votes	Winner	A Wasted Votes	B Wasted Votes
1	53	47	A	2	47
2	53	47	A	2	47
3	53	47	A	2	47
4	53	47	A	2	47
5	15	85	B	15	34
total	227	273	4-A, 1-B	23	222

The efficiency gap is the difference in the two party's wasted votes, divided by the total number of votes.

- All votes for a losing candidate are wasted .
- To win a district, 51 votes are needed, so the excess votes for the winner are wasted votes.

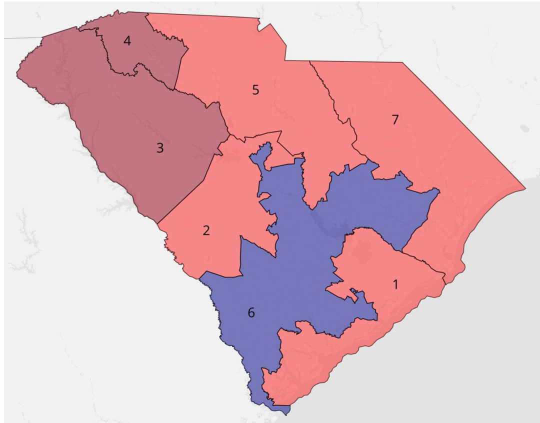
Efficiency gap = $\frac{222-23}{500} = 39.8\%$ in favor of Party A.

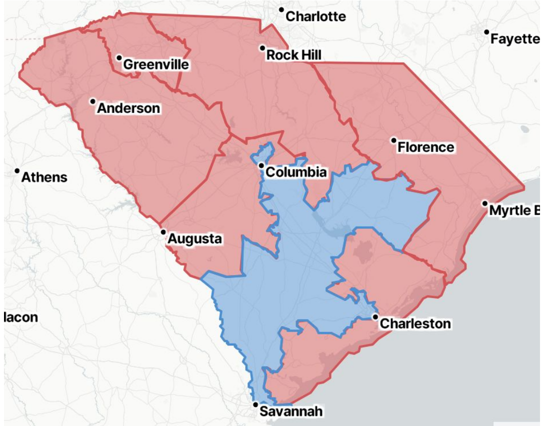
Efficiency Gap

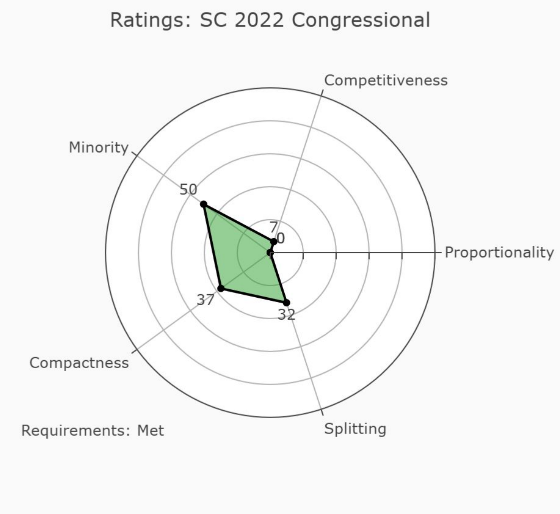
$$PP(D) = \frac{4\pi A(D)}{P(D)^2}$$

Polsby-Popper Score

Results: Current Map (SC)





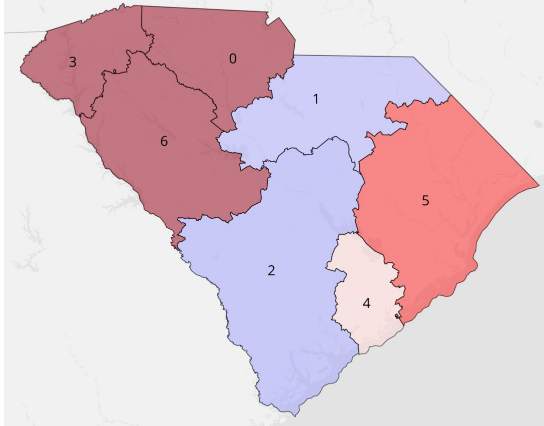


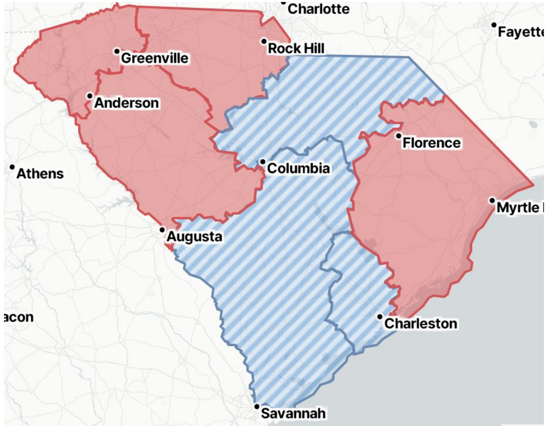
Bigger is better, for the ratings above.

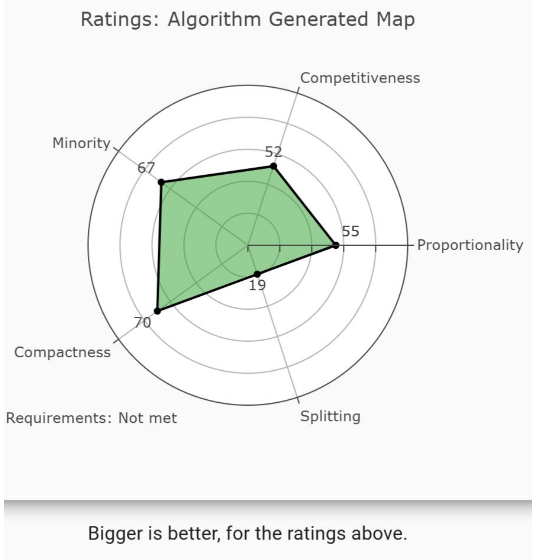
Metric	
• Reock	0.3611
• Polsby-Popper	0.2147

Metric	Value	Favors Democrats in this % of Scenarios*	More Skewed than this % of Historical Plans‡	More Pro-Democratic than this % of Historical Plans‡
Efficiency Gap	19.3% Pro-Republican	<1%	99%	1%
Declination	0.51 Pro-Republican	<1%	91%	5%

Results: My Map (SC)



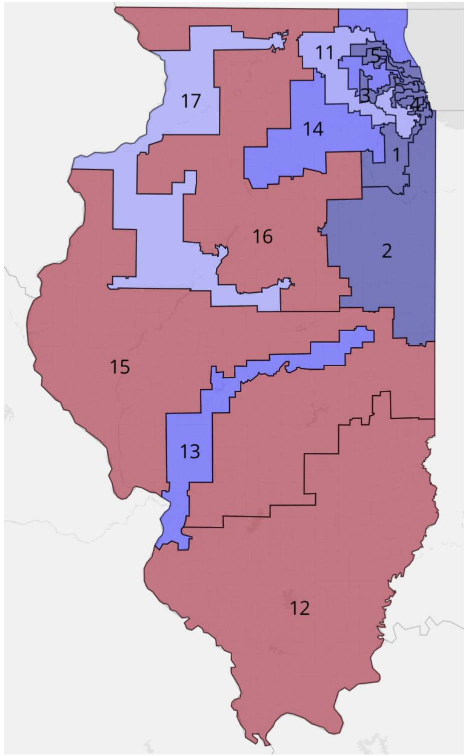


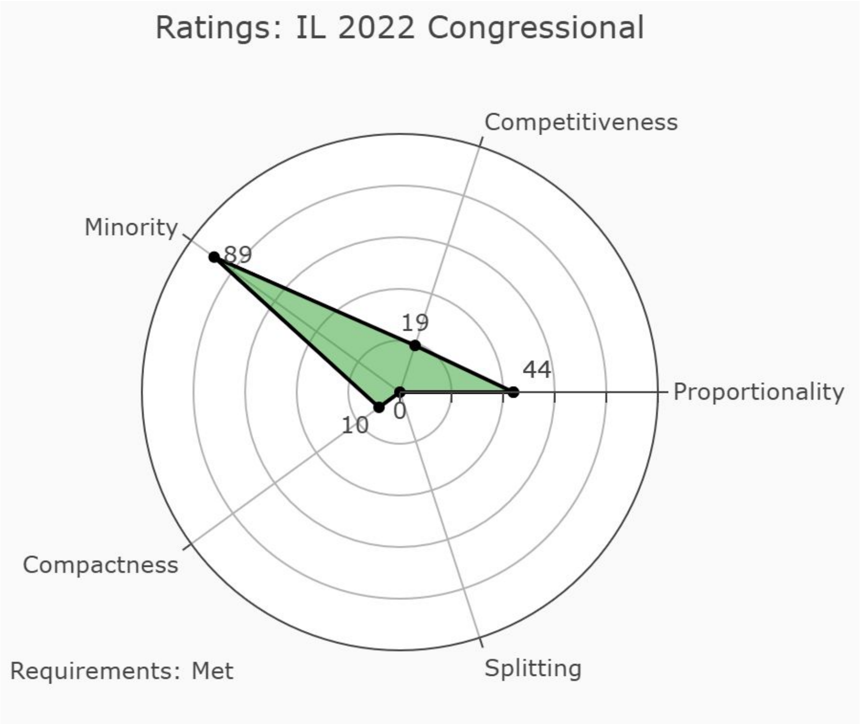


Metric	
• Reock	0.4816
• Polsby-Popper	0.2891

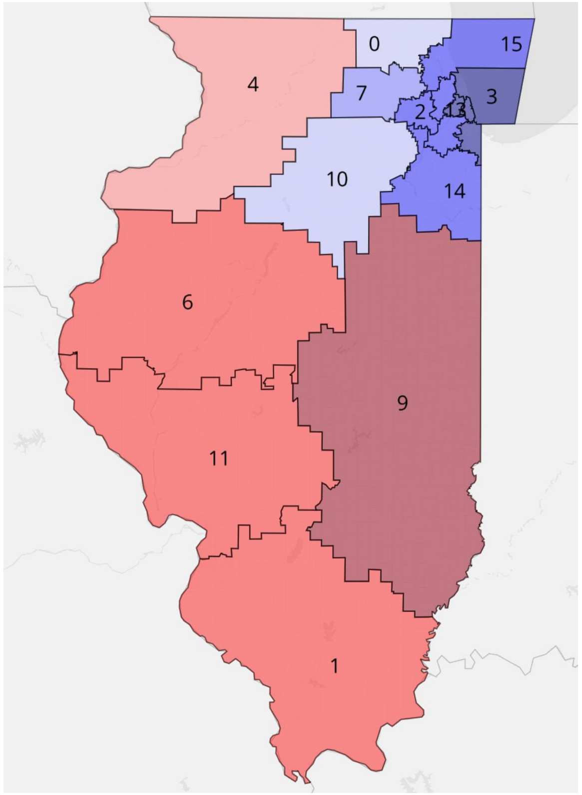
Metric	Value	Favors Democrats in this % of Scenarios*	More Skewed than this % of Historical Plans‡	More Pro-Democratic than this % of Historical Plans‡
Efficiency Gap	5.0% Pro-Republican	34%	51%	22%
Declination	0.02 Pro-Democratic	68%	6%	54%

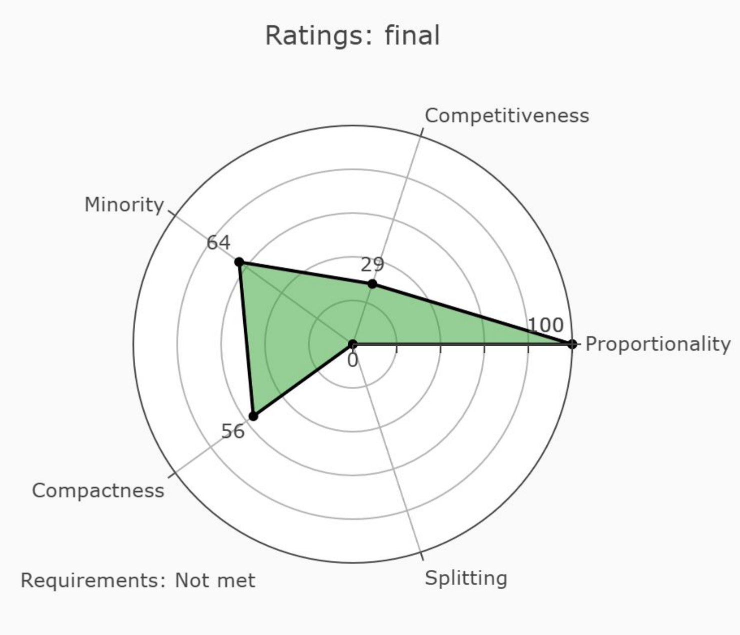
Results: Current Map (IL)





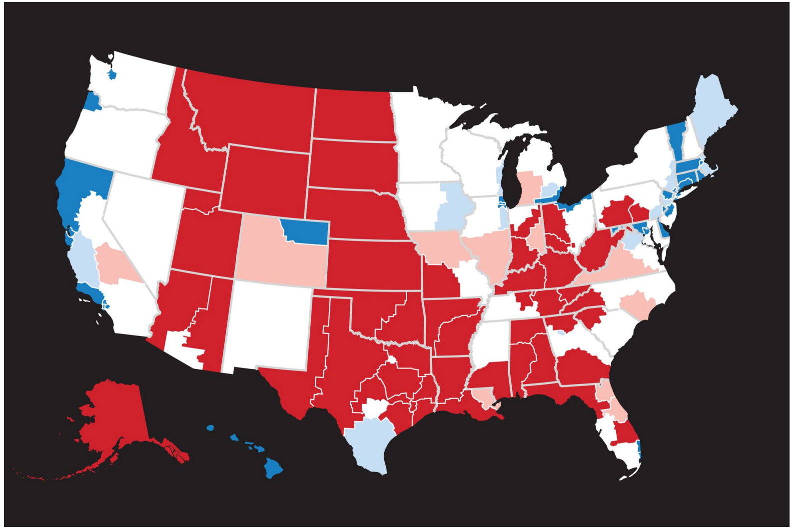
Results: My Map (IL)





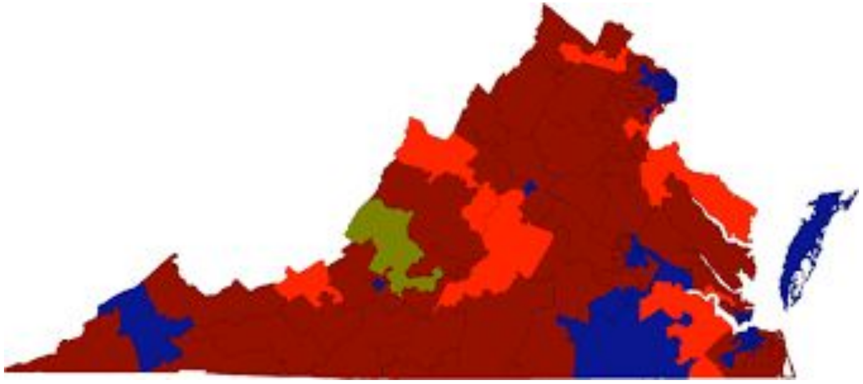
Limitations

- Using precincts as building blocks for districts
  - Cannot make districts of ideal size (784,672)
- Not examining every combination of district sizes
- May be impossible to find “perfect heuristic”



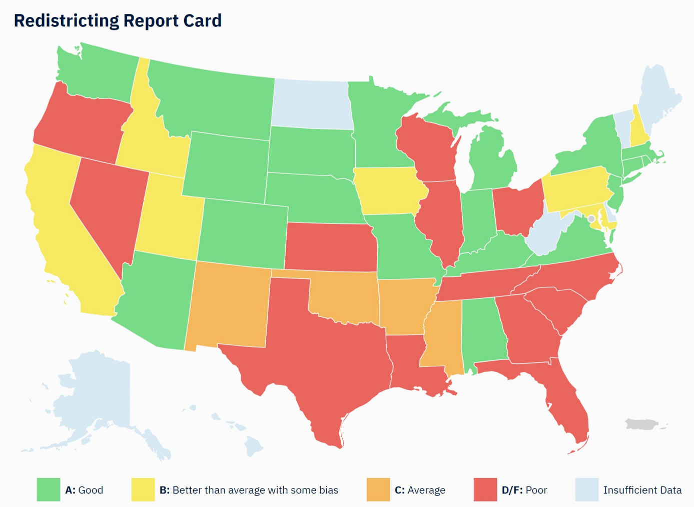
Next Steps

- Continue to make heuristic more complex
- Explore use of Simulated Annealing/Tilted Runs
- Create visualization of algorithm’s performance for all states



Conclusion And Future Work

- I was able to create a novel approach to automating redistricting to combat gerrymandering
- Future Work
  - Attempt to grade current congressional district maps
  - Put up a website for users to test out various approaches



References

[1] Altman, M., & McDonald, M. P. (2013). Paradoxes of political reform: Congressional redistricting in florida. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2421908

[2] Chen, J., & Rodden, J. (2015). Cutting through the thicket: Redistricting simulations and the detection of partisan gerrymanders. *Election Law Journal*, *14*(4).

[3] Cirincione, C., Darling, T. A., & O'Rourke, T. G. (2000). Assessing south carolina's 1990s congressional districting. *Political Geography*, *19*(2), 189-211. https://doi.org/10.1016/s0962-6298(99)00047-5

[4] Duignan, B. (2024, September 21). Gerrymandering. Encyclopedia Britannica. https://www.britannica.com/topic/gerrymandering

[5] Fifield, B., Higgins, Michael, Imai, K., & Tarr, A. (2020). Automated redistricting simulation using markov chain monte carlo. *Journal of Computational and Graphical Statistics*, *29*(4), 715-728. https://doi.org/10.1080/10618600.2020.1739532

[6] Haas, C., Hachadoorian, L., Kimbrough, S. O., Miller, P., & Murphy, F. (2020). Seed-Fill-Shift-Repair: A redistricting heuristic for civic deliberation. *PloS one*, *15*(9), e0237935. https://doi.org/10.1371/journal.pone.0237935

[7] Kevin Baas. (2018, May 11). *About*. AutoRedistrict. http://autoredistrict.org/

[8] Liu, Y. Y., Cho, W. K. T., & Wang, S. (2016). PEAR: A massively parallel evolutionary computation approach for political redistricting optimization and analysis. *Swarm and Evolutionary Computation*, *30*, 78-92. https://doi.org/10.1016/j.swevo.2016.04.004

[9] Olson, B. (2020, October 17). *2010 redistricting results*. BDistricting. https://bdistricting.com/2010/

[10] *Virginia data*. (2024). Redistricting Data Hub. https://redistrictingdatahub.org/state/virginia

[11] Volić, I. (2024). *Making democracy count: How mathematics improves voting, electoral maps, and representation*. Princeton University Press.

Q&A

Any questions?

Demo

THANKS!