INDUSTRIAL AND SYSTEMS ENGINEERING



A New Multiobjective Heuristic For Creating Political Redistricting Plans While Maximizing Similarity To A Previously-Used Plan

BRENDAN RUSKEY¹ AND LAWRENCE V. SNYDER¹

¹Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA

ISE Technical Report 23T-021



A new multiobjective heuristic for creating political redistricting plans while maximizing similarity to a previously-used plan

Brendan Ruskey and Lawrence V. Snyder Lehigh University bjr221@lehigh.edu lvs2@lehigh.edu

September 4, 2023

Abstract

In this paper, we introduce a multiobjective genetic algorithm (MOGA) for generating political redistricting plans. Unlike all existing MOGAs for redistricting, and most other heuristic algorithms for redistricting, our MOGA produces plans that maximize similarity to an existing plan. Our focus on similarity with a given plan addresses the real-life phenomenon that new redistricting plans are typically created with the previous plan in mind. As part of our algorithm, in order to promote plans that resemble some given original plan, we utilize a new initialization method for setting the initial population of the MOGA. In addition to similarity with the given plan, our algorithm considers the classic objectives of population deviation and compactness, and returns a set of redistricting plans that are nondominated in these three objectives. Our numerical experiments demonstrate the effectiveness of our MOGA in generating redistricting plans that balance the objectives of similarity, compactness, and population deviation.

keywords: redistricting, elections, democracy, similarity, voting

1 Introduction

In this paper, we describe a new multiobjective heuristic algorithm for the political districting problem. The political districting problem is the process of grouping individual voters into electoral districts. In the United States, each congressional district elects a single representative to advocate for its interests in Congress. Specifically, we focus on the *redistricting* problem, in which voters are regrouped into electoral districts from some pre–existing grouping. In the US, this process occurs every ten years, following the US Census, in between which the population of districts can change. While the political redistricting problem is relevant to elections in many nations, and can occur at either the national or local level, we will primarily apply our method to redistricting for the United States House of Representatives. The terms "districting" and "redistricting" are often used interchangeably, in our paper and in the literature as a whole.

Redistricting plans are evaluated by several criteria. First, a plan is expected to have low population deviation among its districts, meaning each district in the plan has roughly the same number of inhabitants as other districts. Also, the districts of a plan must be geographically contiguous—that is, given two points in a district, there must exist a path between these two points that is contained within the district. Plans can also have other desirable properties, including "compactness," which is a measure of the shape of a district, and how close its voters are to one another. To these three criteria, we introduce a fourth, namely, similarity.

Thus, the heuristic method that we present in this paper returns sets of feasible redistricting plans that are contiguous, and which balance three objectives: minimizing population deviation, maximizing compactness, and maximizing similarity with some base plan P_0 . While similarity is not a legal constraint in the same manner as population deviation or contiguity, and is not as emphasized in the relevant literature as criteria such as compactness, it is apparent that in practice, new redistricting plans are typically based on the previous plan. As a result, just as redistricting plans with high population deviations or low compactness scores are unlikely to be adopted by state legislatures, proposed plans that are highly dissimilar from the previously used plan may be unlikely to receive consideration. Both voters and politicians prefer districts that do not change much from one plan to the next, as we discuss in Section 2.3, and demonstrate empirically in Section 2.4.

We use the Precinct Pairs Change similarity measure from Becker and Gold [7] to evaluate the similarity between two redistricting plans, and we maximize the value of this measure in our heuristic. As a result, our heuristic produces plans with districts that are contiguous, compact, roughly equally populous, and resemble a given base plan. Our new multiobjective genetic algorithm is the first multi-objective genetic algorithm (MOGA) to produce plans which address similarity with some base plan P_0 . Our MOGA utilizes a new method for initializing the population of the genetic algorithm. Our initialization method creates a set of plans based on the base plan P_0 , using it as a starting point, then applies sets of randomized perturbations that maintain contiguity and decrease the population of the plan.

This paper is organized as follows. Section 2 explains the political process behind redistricting and the requirements for feasible plans. This section will also motivate and explain our similarity measure for comparing two given redistricting plans. For each state, plans from the previous redistricting cycle will be compared with plans from the current redistricting cycle, in order to demonstrate the typically high level of similarity between the two. The results strongly suggest that similarity with the previous plan is an expected feature of redistricting plans in the United States. Section 3 describes previous work in algorithms for generating redistricting plans. In Section 4 we describe our new multiobjective heuristic method for generating redistricting plans. In Section 5, we present the results of several experiments which demonstrate the effectiveness of our method. Finally, Section 6 offers concluding remarks.

2 The political redistricting problem

2.1 Introduction

Redistricting is a crucial part of national and state politics in the US, as it can impact which political party has a majority in government. First, at the national level, the US House of Representatives and the US Senate control many of the functions of the US government. While US Senators win their seats in statewide elections, and therefore do not require districting plans, US House representatives earn their jobs by winning elections in their respective districts within each state. Following each Congressional election, which occur in even-numbered years, the majority party in the House of Representatives similarly controls which bills are passed to the US Senate. Thus, the majority party in the House of Representatives is positioned to enact its agenda, from crucial budgetary proposals to impeachment votes. The majority party in both chambers (House and Senate) also controls committee assignments and which legislators will be the leaders of those committees. Since the outcome of US House of Representative elections is impacted by redistricting, redistricting itself plays a role in the actions taken by US Congress. US states also have their own legislatures, most of which feature an upper house analogous to the US Senate, and a lower house analogous to the US House of Representatives.

Redistricting is done every ten years in the United States following the US Census. Most state legislatures create their state's redistricting plans—one plan for its lower house, one for its upper house, and one for its US Congressional districts.¹ In theory, the goal of redistricting is to create a set of contiguous, similarly populated districts that preserve communities and satisfy conditions such as the Voting Rights Act of 1965, i.e., they must be drawn in a way that does not disadvantage people of any particular race [6]. Each state has its own procedure for determining its redistricting plans. For example, as of 2020, Michigan used an independent commission consisting of Democrats, Republicans, and non-partisan commissioners who combined data from previous elections and community input to draw a new plan [2]. In most other states, however, there is no such independent commission, and the majority party in the state legislature controls the redistricting process—they can propose the state's redistricting plans with little outside input or influence.

During the redistricting process, plans approved by state legislatures are subject to review. For example, in many states, they are subject to veto by the state's governor, although a state legislature with a large

¹States with only three electoral votes, and therefore only a single congressional district, do not need to create a redistricting plan for US Congress. Nebraska, which has three US Congressional districts, is the lone state with a unicameral (one house) state legislature, and thus requires only two redistricting plans.

majority can override the governor's veto, as occurred in Kentucky in 2021 [29]. The approved redistricting plan can also be brought to the Supreme Court of that particular state for review. In the 2018 court case *League of Women Voters of Pennsylvania v. Commonwealth of Pennsylvania*, the Pennsylvania Supreme Court ruled that the 2012–2021 redistricting plan approved by the Pennsylvania General Assembly (lower house) violated the Free and Equal Elections Clause of the Pennsylvania Constitution due to the plan unfairly benefiting a political party. The Pennsylvania Supreme Court removed the plan that had been in place for several years, and replaced it with a plan featuring much more compact districts and fewer counties split between districts [1].

The geographic units grouped into districts by states during redistricting are called "voting precincts," and can be formed using any geographical criteria. These voting precincts do not need to be based on US Census data, such as census tracts or census blocks (which are smaller than census tracts). While most states do not define rules for the size and shape of units used for redistricting, Iowa and West Virginia require that counties are not split in their US House redistricting plan [10]. Therefore, for these two states, redistricting is done at the county level, with each county being assigned to a Congressional district.

Since voting precinct data is not readily available to researchers, algorithms are typically tested using census tracts (e.g., Validi et al. [42], Vanneschi et al. [43], Lara-Caballero et al. [21]) or using census blocks. When solving the redistricting problem using an optimization model, especially as a mixed integer problem, the type of geographic unit considered affects the performance of the algorithm. When smaller units are used (for example, census blocks instead of census tracts), then a lower minimum population deviation is typically possible, due to an increase in the number of feasible plans. However, using smaller units typically increases solver time, especially for MIPs, where more units implies more decision variables and constraints in the resulting model.

The politicians in charge of the redistricting process in each state are faced with a multiobjective optimization problem, balancing population deviation, district compactness, similarity with the previous decade's plan, number of splits of counties or municipalities, number of majority-minority districts, as well as several measures of partisan fairness. Therefore, the politicians and their political parties have used various forms of (exact or approximate) optimization when creating plans. For instance, many state legislatures have used mapping software such as Maptitude [24] to create redistricting plans and evaluate their population deviation, compactness, and partisan fairness metrics. This software has a feature that automatically creates compact and contiguous districts that satisfy a given population deviation constraint. Similarly, in 2021, the Montana State Legislature encouraged citizens to submit plans via *Dave's Redistricting App*, which also reports metrics associated with the submitted plans [4]. Other states have used basic redistricting software to automatically generate new plans that optimize statistical criteria [25]. However, McDonald and Altman [25] note that state legislatures are often reluctant to fully hand over the redistricting process to any software or algorithm, since software may not be able to identify and preserve communities of interest and other political expectations that may exist. Therefore, legislators may prefer an automated redistricting algorithm that considers a previous plan, with all of its communities of interest and unique features, and aims to create a desirable plan from this existing plan while changing the plan as little as possible. This is the distinctive feature of the multiobjective genetic algorithm we present in Section 4, which maximizes similarity with some previous plan while minimizing population deviation and maximizing compactness.

2.2 Redistricting plan criteria

Several numerical criteria are considered when state legislatures evaluate a redistricting plan. In the US Supreme Court case *Wesberry v. Sanders* (1964), the Court ruled that "as nearly as practicable, one man's vote in a congressional election is to be worth as much as another's" [44]. Since then, state legislatures have paid close attention to the population deviation of their plans. Specifically, they are interested in plans with low population deviation— that is, plans with districts having roughly equal population. We will use the following numerical definition of population deviation (PD):

$$PD = \sum_{k=1}^{K} |pop_k - \mu|, \tag{1}$$

where K is the number of districts, pop_k is the population of district k in the plan, and μ is the target population for each district (the population of the state divided by K). Several US states define the maximum population deviation level allowed for their redistricting plans [42]. For example, in Iowa, the population deviation of its US congressional redistricting plan must be less than 1% of the target populations [10]. In practice, this implies that the population deviation of a feasible Iowa plan must be less than or equal to about 6000. Many other states have constitutional requirements for maximum population deviation; for example, Colorado's state constitution imposes a constraint on the difference between the most populous district and least populous district in its state senatorial and state representative districts [37]. Also, in practice, many redistricting plans, including Pennsylvania's US congressional redistricting plan, feature districts whose populations differ from the target population by at most one voter.² In [21], mean deviation (MD) is used:

$$MD = \sum_{k=1}^{K} \frac{|pop_k - \mu|}{K\mu}.$$
 (2)

This metric gives the average relative deviation from the target population μ across all K districts for a given plan. The multiobjective heuristic for redistricting that we present in Section 4 considers population

²If μ is the target population, then each district has population $\mu - 1$, μ , or $\mu + 1$. This level of equality is difficult to achieve when redistricting is done at the census tract level, which is the level considered by our algorithm, but is easier to achieve when using smaller voting precincts, as is commonly done is practice.

deviation, as defined by (1). However, for three problem instances with a large number of districts, we evaluate plans using the mean deviation measure defined by (2).

Another recognized measure for redistricting plans is the geometric compactness of its districts. One popular method of measuring compactness in an optimization setting is unit dispersion, which calculates the sum of the distances of each unit in a district from its geographical center [33]. A number of geometric compactness measures have also been proposed. One measure is the Reock compactness score, first presented by Reock [32]. This simple measure takes the ratio of the area of the district to the area of the smallest circle that contains the district. A rating close to 1 indicates a higher level of compactness. A different measure that has been used by redistricting commissions such as the Arizona Independent Redistricting Commision is the Polsby-Popper score [31, 28]. This measure of compactness for a district D, which also takes a value between 0 and 1, is calculated as follows:

$$PP(D) = \frac{4\pi A(D)}{P(D)^2},$$
 (3)

where A(D) is the area of district D and P(D) is its perimeter. As with the Reock compactness score, a lower value is indicative of a less compact district. A redistricting plan with consistently low Reock compactness scores and Polsby-Popper scores may be indicative of partian manipulation. The multiobjective heuristic for redistricting that we present in Section 4 considers the Polsby-Popper score for compactness, as defined by (3).

Other redistricting criteria exist for determining a given plan's feasibility. For example, recently, emphasis has been placed on minimizing the number of counties that are split between multiple districts. Iowa and West Virginia are two states which forbid the splitting of counties in their US congressional redistricting plans. In other states, the unnecessary splitting of counties has been used by courts as a basis for ruling that a plan is illegal. An example of this in the 2018 cycle occurred in Ohio [5]. In addition, Section 2 of the US Voting Rights Act ensures that redistricting plans must be drawn in a way that does not unfairly impact voters of any particular race. The landmark US Supreme Court case *Shaw v. Reno* (1993) also ruled that gerrymandering on the basis on race is prohibited by the Voting Rights Act [40]. There exist other measures, including measures for evaluating partisan fairness like the efficiency gap [34], partisan symmetry [35], and competitiveness of districts in the plan, such as those presented in [35].

2.3 Similarity between redistricting plans

In practice, another factor considered in selecting a new redistricting plan is whether the new plan resembles the previous decade's enacted plan. In this paper, we say that two plans are "similar" if few voters change districts from the first plan to the next (we will introduce a more formal definition of similarity below).





(a) Mississippi base plan P_0 with population deviation 357145

(b) Mississippi solution plan with population deviation 3988

Figure 1: Mississippi map comparison: Similarity Score = 0.858

The two plans shown in Figure 1 demonstrate the concept of similarity. While our definition of similarity is based on voters and not geography, in this case, the similar district boundaries between the two plans demonstrate the similarity in each district's electorate before and after redistricting. On the other hand, the redistricting plan shown in Figure 13 of Section 8 shows two plans that are less similar to each another. Note that our definition of similarity accounts only for similarity of population, and not necessarily similarity in geographical regions covered by the districts of each plan.

In decennial redistricting, there is a strong political motivation for maximizing similarity between the previous decade's plan and the next decade's plan. It may be confusing for voters if many of them change districts between plans [45]. At the time of redistricting, the voters of each state have elected their current representatives in the previous election; therefore, many voters may prefer to remain in their current district so that they are represented by the same congressperson. McKee et al. [26] provide statistical evidence of how the representative–constituent relationship is adversely impacted by redistricting plans that are dissimilar to the plan of the previous decade. Their analysis suggests that redrawn voters (voters whose district changed from one decade to the next) are less likely to know the name, gender, and race of their representative. They are also less likely to approve of their representative, and more likely to believe that redistricting in their state was conducted in unfair manner. The authors conclude by claiming that the much of the literature of political redistricting has been focused on the difficult-to-quantify ideas of "partisan fairness," at the cost of considering the politically significant and more straightforward notion of preserving representative–constituent relationships during redistricting.

It is also worth noting that similarity has been explicitly mentioned in court rulings as a desirable attribute of redistricting plans, especially when Democratic and Republican legislators are unable to compromise on a plan [7]. For example, in the ruling of Colleton County Council v. McConnell (2001), the US District Court in charge of drawing the South Carolina State House plan said that they had been "seeking to preserve the core [of the district of the previous plan] of each existing district where possible ... adding or subtracting population surrounding the core in a compact and contiguous manner" [36].

Incumbent legislators at the time of redistricting also have a motivation to preserve the districts of the original plan. If a given congressperson's district is very different under the new plan, then they will have to represent a very different group of voters than they are accustomed to representing. This could force them to change their campaign strategy for re-election or to reevaluate their policy priorities [45]. One of the foundational papers in redistricting heuristics, Nagel [30], stated that practical experience suggests that when incumbent legislators evaluate new plans, they are concerned with how changes will impact their own districts. In a review of contemporary literature on political redistricting, Williams [45] commented that "a new political redistricting plan that resembles the old plan may be more socially and politically acceptable than a totally reworked scheme."

Several measures have been proposed for measuring similarity between two redistricting plans. Abrishami et al. [3] present a similarity measure for comparing two graph partitions, and uses political redistricting as the primary setting for demonstrating the measure. Bozkaya et al. [8] present a multiobjective tabu-search heurstic for generating redistricting plans, and include similarity with some original plan as an objective. It uses a similarity measure based measuring the overlap of districts between the two plans. Recently, Becker and Gold [7] presented over a dozen new similarity measures for comparing the populations of two redistricting plans. One measure of similarity presented in the paper is called the *Precinct Pairs Change* measure. This measure tracks the share of all people who previously shared a district, but under the new plan are in different districts. For instance, suppose that prior to redistricting, District 1 is composed of 10000 inhabitants. Then, there are $\binom{10000}{2}$ pairs of voters in the original District 1 who will either remain in the same district or will be broken into separate districts in redistricting. For our analysis, will use the Precinct Pairs Change measure the similarity for an individual district in Plan X under a new Plan Y, as well as the similarity between two full plans X and Y, can described mathematically as follows:

$$\Delta(d,Y) = \frac{\sum_{k \in \mathcal{D}_Y} \binom{y(d,k)}{2}}{\binom{x(d)}{2}} \tag{4}$$

$$\Delta(X,Y) = \frac{1}{D_X} \sum_{d \in \mathcal{D}_X} \frac{\sum_{k \in \mathcal{D}_Y} \binom{y(d,k)}{2}}{\binom{x(d)}{2}},\tag{5}$$

where the notation is as follows:

- $\Delta(d, Y)$: Precinct Pairs Change score for district d in Plan X under Plan Y
- $\Delta(X, Y)$: Precinct Pairs Change score for Plans X and Y (the average proportion of neighbor pairs maintained under new plan Y across all districts in the original plan X)
- $\mathcal{D}_{\mathcal{X}}$ is the set of districts in plan X
- D_X is the number of districts in plan X
- x(d) is the population of district d
- y(d,k) is the number of voters from district $d \in \mathcal{D}_X$ that are in district $k \in \mathcal{D}_Y$ in plan Y.

Suppose that after redistricting, 6000 of the 10000 voters in our example district remain in District 1, but 3000 are moved to District 2, and 1000 are moved to District 3. Then, $\binom{6000}{2} + \binom{3000}{2} + \binom{1000}{2}$ pairs of the original $\binom{10000}{2}$ pairs are preserved in the new plan. This district would have a Precinct Pairs Change score of

$$\frac{\binom{6000}{2} + \binom{3000}{2} + \binom{1000}{2}}{\binom{10000}{2}} = 0.45\bar{9}.$$

If a district's population is P in Plan X and there are K districts in both Plan X and Plan Y, then the lowest Precinct Pairs Change score possible for an individual district is 1/K, which is attained when the Pmembers of the district in Plan X are placed evenly among the K districts of Plan Y.

Note that $\Delta(X, Y) \approx 1$ implies a high degree of similarity between the two plans, while $\Delta(X, Y) \approx 0$ implies low similarity. This paper will occasionally discuss the individual district Precinct Pairs Change value (4), but we are primarily interested in the full-plan Precinct Pairs Change value (5). Note also that the Precinct Pairs Change measure can be calculated for a single district $d \in D_X$ by fixing d and removing the summation, as well the $1/D_X$ constant. Lastly, note that if a district's population decreases from plan X to plan Y, than it cannot achieve a "perfect" score of 1.0, since it is forced to move some of its population to other districts. On the other hand, if a district gains in population from plan X to plan Y, it can achieve a score of 1.0 despite adding people it did not contain previously, so long as it contains all of the same people as in the previous plan.

The multiobjective heuristic for redistricting that we present in Section 4 considers similarity as a measure, using our modification of the Precinct Pairs Change measure (5). For the remainder of the paper, we will refer to the Precinct Pairs Change measure as "the similarity score."

2.4 Analyzing similarity in 2022–2031 United States redistricting

Though the philosophical importance of considering similarity across plans is clear, we also motivate its importance by analyzing the degree to which actual districting plans resemble the previous plan. Although similarity is not often discussed as an explicit requirement when evaluating plans, it is clear from examining states' currently enacted plans that similarity to the previously-used plan is a desirable attribute. In this section, for each state with available data, we compare the redistricting plan of the previous decade (the plan used from 2012–2021) to the most recently adopted plan (in effect from 2022–2031). This data is reported in Table 4. In the second column, the number of districts in each state are given. In the third column, the population deviation of each previous plan using the 2020 Census population is recorded, rounded to the nearest integer value. In many cases, districts of the previous plan will be overpopulated or underpopulated, creating the need for redistricting. In addition, the similarity scores (5) are calculated between the previous and new plans. That is, in the fourth column, the average individual district similarity score is reported. The most-preserved district (maximum individual district similarity score) and least-preserved district (minimum individual district similarity score) are reported in the fifth and sixth columns. The results in Table 1 compare US House maps between decades. Similar results for State Senates and State Houses/Assemblies are given in the Appendix (Section 10). The numerical results here are calculated from DRA [14], which stores data from previous and currents plans, including how districts from the previous plan overlap with districts from the current plan. Certain states do not have data for this comparison. Examples include states with only one US congressional district during the 2012–2021 redistricting cycle, and states such as New Jersey and Arizona, which use the same districts for their State House and State Senate. Results from Vermont are not considered at any of the three levels, since the relevant data are not included in DRA [14].

These results show that for most US states, there is a high degree of similarity between the previous plan and the newly enacted plan, at all three levels of redistricting. In many instances, some districts of the previous plan receive individual similarity scores of 1.000. Table 2 reports average, maximum, and minimum values of similarity statistics across the 43 US states requiring redistricting in the previous decade. In 34 of these 43 states, there is a district in the original plan with similarity score over 0.90 under the newer plan. Similarly, 28 original plans feature a district with similarity score over 0.95 in the new plan, and 20 original plans feature a district with similarity score over 0.95 in the new plan, and 20 original plans feature a district, sprevious and current US House districting plans. For instance, Connecticut's 2nd congressional district, encompassing the western portion of the state, is almost unchanged. The previous territory is preserved, and the new boundary only expands the previous boundary. Similarly, consider the plans in Figure 3. The district boundaries are highly similar between the previous and current US House districting plans. For instance, Kentucky's 3rd Congressional district has an individual similarity score in the previous plan with under the new plan of 0.977.

State	K (2012–2021)	Deviation	Sim Score	High Ind. Sim.	Low Ind. Sim.
Alabama	7	184236	0.891	0.977	0.780
Arkansas	4	205534	0.862	0.927	0.734
Arizona	9	277638	0.578	0.751	0.356
California	53	1407899	0.600	1.000	0.278
Colorado	7	168593	0.615	0.830	0.391
Connecticut	5	61244	0.962	1.000	0.934
Florida	27	993372	0.598	0.964	0.272
Georgia	14	418169	0.682	0.968	0.360
Hawai'i	2	4370	0.996	1.000	0.991
Idaho	2	70621	0.964	1.000	0.929
Illinois	18	418978	0.480	0.755	0.183
Indiana	9	225448	0.736	1.000	0.381
Iowa	4	122731	0.739	0.785	0.671
Kansas	4	115642	0.774	1.000	0.584
Kentucky	6	178780	0.859	0.992	0.712
Louisiana	6	171849	0.932	1.000	0.879
Maine	2	46046	0.926	0.954	0.897
Maryland	8	189353	0.504	0.778	0.206
Michigan	14	328890	0.521	0.912	0.325
Massachusetts	9	189353	0.504	0.778	0.256
Minnesota	8	199638	0.863	0.964	0.781
Mississippi	4	131741	0.924	0.998	0.836
Missouri	8	209729	0.698	0.939	0.387
Nebraska	3	106312	0.871	1.000	0.792
Nevada	4	178000	0.657	1.000	0.419
New Hampshire	2	17906	0.987	1.000	0.975
New Jersey	14	296306	0.692	0.960	0.448
New Mexico	3	22550	0.624	0.667	0.571
New York	27	659139	0.598	1.000	0.269
North Carolina	13	645614	0.509	0.915	0.279
Ohio	16	279558	0.702	0.940	0.371
Oklahoma	5	152829	0.793	0.998	0.603
Oregon	6	47742	0.577	0.653	0.405
Pennsylvania	18	446919	0.826	1.000	0.377
Rhode Island	2	11200	0.982	0.993	0.971
South Carolina	6	240200	0.866	0.961	0.716
Tennessee	9	398133	0.710	1.000	0.351
Texas	36	1981204	0.531	0.946	0.149
Utah	4	130537	0.752	0.835	0.624
Virginia	11	426306	0.573	0.829	0.294
Washington	10	164174	0.795	1.000	0.456
West Virginia	3	54639	0.825	1.000	0.500
Wisconsin	8	135849	0.901	1.000	0.318

Table 1: Map comparison statistics: US House, 2012–2021 to 2022–2031

	Mean	Max	Min
Full-Plan Similarity Score	0.752	0.995	0.480
Max. Individual District Similarity Score	0.933	1.000	0.653

Table 2: Summary of similarity scores between 2012-2021 plans and 2022-2031 plans (for the 43 US States requiring redistricting in the 2012-2021 redistricting cycle)



(b) Connecticut 2022–2031 [38]

Figure 2: Connecticut US House Plan Comparison



(a) Kentucky 2012–2021 [39]





Figure 3: Kentucky US House Plan Comparison

Since, in many instances, new redistricting plans are created from the previous plan, it is crucial that algorithms for producing redistricting plans consider the notion of similarity with some base plan P_0 . As we will demonstrate in Section 5, if similarity is not explicitly considered, it is unlikely that the plans produced will be similar to some arbitrary base plan. Therefore, it is evident that algorithms for generating redistricting plans should aim to produce plans that, in addition to meeting other criteria, are based on the previously used districting plan. Motivated by these results, our multiobjective genetic algorithm, described in detail in Section 4, generates plans that, while maximizing the typically-considered measures, also maximize similarity to a previously created plan.

3 Related work

3.1 MIP approaches

Given the societal importance of redistricting, many researchers have proposed methods for generating redistricting plans which perform well in terms of population deviation and compactness. Specifically, some have formulated the redistricting problem as a mixed integer program (MIP). In 1965, Hess et al. [19] proposed an integer programming formulation that separated geographic units into districts of roughly equal population, constrained to some level of tolerance. In the objective function, the sum of the distances from each unit to the "center" of its assigned district is minimized. This objective promotes compact solutions, but there are no constraints enforcing contiguity. As a result, solutions to the Hess model would have to be manually edited to produce contiguous redistricting plans. Later, researchers would produce MIPs which enforced contiguity through their constraints. For example, Shirabe [33] presented several new models, each with network flow-based contiguity constraints. Given the adjacency graph of the units, treating the units as nodes and their adjacencies as arcs, their definition of contiguity ensured that the subgraph on the units/nodes in each district is connected. While this model accomplished its goal of enforcing contiguity, it failed to solve large instances of the redistricting problem (instances with adjacency graphs having thousands of nodes/arcs), since the set of constraints grows exponentially as the number of nodes and arcs increase. Other MIP models with and without contiguity constraints have been proposed, including those of Caro et al. [9], who studied redistricting of school districts, and Gentry and Chow [15], who proposed a MIP formulation for the problem of determining optimal organ donation allocation.

Enforcing contiguity was long thought to be the "bottleneck" preventing large instances of the political redistricting problem from being solved to optimality by a MIP. However, recently, Validi et al. [42] showed that contiguity can more easily be enforced by using cut generation, and that solving the LP root relaxations is what prevents many large instances of the problem from being solved. Their paper uses the concept of a-b separators to generate cuts which remove discontiguous solutions from the set of feasible regions defined

by the Hess model. Their results are the best to date among MIP approaches, in that they solve census tract-level instances of the redistricting problem for states that had not been previously been solved. For example, using their cut generation method, they solved the census-level Indiana instance (1,511 census tracts, over 2 million binary decision variables) of the problem to optimality [42].

3.2 Heuristics

Despite this recent progress, for many practical instances, the redistricting problem cannot be solved to optimality using any current MIP formulation. For example, California has over 8,000 census tracts [41], meaning that the formulation presented by Validi et al. [42] would require over 64 million binary decision variables. As a result of the challenges associated with using MIPs for the political redistricting problem, many researchers have developed heuristic methods for generating valid plans. Heuristics have been developed for the problem for as long as MIP approaches have been proposed. Nagel [30] proposed a heuristic based on swapping units on the borders between districts to minimize population deviation. When discussing practical implementation, the author proposes using the previous decade's redistricting plan as the starting point, which is altered until the resulting plan's population deviation falls within a predefined tolerance. While the paper only presents results for small instances, many heuristics have drawn inspiration from this approache. Other, more recent papers, including those by Kaiser [20] and Hayes [18], also used "swapping" approaches, but only report solutions with impractically high population deviations. Our method, presented in Section 4, uses a swapping procedure to reduce population deviation in plans.

Bozkaya et al. [8] presents a tabu search heuristic for generating redistricting plans from some original plan. Several objectives, such as compactness, population deviation, and similarity, were considered. This is one of the few existing approaches to consider similarity as an objective. To measure the similarity of a new plan with the original plan, the largest overlap between each district in the new plan with some district in the original plan is recorded. Results for a small example are reported, but this approach is unlikely to scale to larger problems. Previously, Mehrotra et al. [27] introduced a column generation algorithm for producing near-optimal plans. Preprocessing and a customized branching rule for solving the problem are presented. Their proposed algorithm is tested on the county-level South Carolina instance, a relatively small problem. In 2011, Gurnee and Shmoys [17] proposed a column generation heuristic that is based on finding feasible districts within subdivisions of a given state. Unlike an exact column generation approach, which is subject to difficulties like degeneracy in the master problem, their approach focuses on generating quality columns/districts. Their results produced large ensembles of districting plans for large instances of the redistricting problems. The authors use this large set of produced plans to show how statistics like expected partisan control of a state's districts vary between states. Ensemble approaches such as this and in Deford et al. [13] generate large sets of plans for comparison against existing plans in order to provide evidence of fairness or of partian manipulation. In Validi et al. [42], in addition to the proposed MIP model and associated cut generation technique, the authors also proposed a local search heuristic for generating near-optimal redistricting plans at the census tract-level from an initial plan.

3.3 Multiobjective genetic algorithms

A genetic algorithm (GA) is a type of evolutionary algorithm that mimics the process of natural evolution on a set of solutions, maximizing or minimizing some objective function. This metaheuristic features several biologically-inspired operations that are performed on the set of candidate solutions, known as the population. These operations include *selection*, in which members of the population are chosen based on their "fitness" to participate in the next operation, *crossover*. In crossover, two candidate solutions are combined to form two new candidate solutions. Then, in *mutation*, candidate solutions are altered in a process analogous to biological mutation.

In a multiobjective genetic algorithm, the selection step requires defining a rule for comparing candidate solutions with more than one objective. Deb et al. [11] propose a nondominated sorting procedure for ranking the fitness of candidates with two or more objectives, based on building nondominated fronts and evaluating of the position of candidates within these fronts. The candidate solutions to a given problem must be encoded in way that allows the problem to be solved as a GA; in Section 4, we describe how Vanneschi et al. [43] encodes the political redistricting problem as a GA.

Xiao [46] proposed a genetic algorithm for "geographic optimization" problems, including political redistricting. The paper discussed how to incorporate problem specific knowledge into the GA, such as a pre-existing contiguous plan. Subsequently, Liu et al. [22] introduced a GA with a new mutation operation, shifting multiple units at a time, and a crossover operation similar to that of Xiao [46]. The authors used this GA within a parallel computing setting, and consider three objectives: population deviation, compactness, and partisan competitiveness (using voting data from previous elections). They provided results for a single instance of the political redistricting problem—the North Carolina census tract-level instance (2590 nodes/tracts). The authors compared the performance in this example to several existing heuristics, including some mentioned in Section 3.2. Later, Lara-Caballero et al. [21] introduced a new multiobjective genetic algorithm (MOGA), which, like Liu et al. [22], considers three objectives. However, the three objectives considered are mean deviation (as in (2)), population range (difference between the population of the most populous district and the population of the least populous district), and compactness. The GA is encoded similarly to the previously mentioned methods, and the initial population is determined by choosing Kunits, where K is the desired number of districts, to form K initial districts, with the other units unassigned. Then, unassigned units on the borders of these districts are iteratively assigned to the districts which they border. Two versions of this MOGA are presented, one using the NSGA-II [11] fitness criteria for selection,

and one using the SPEA-II fitness criteria. In the new crossover operation this MOGA proposes, a "repair" method for fixing discontiguous districts was proposed. A modified version of this repair process will be used in our proposed method. The mutation operation used is similar to that of Vanneschi et al. [43] which will be described in detail in the next subsection. This MOGA produced compact, contiguous plans with low population deviation for the three largest census tract-level problem instances (California, Texas, New York). For the California example, nondominated plans with mean deviation of roughly 0.002 were reported, equivalent to a population deviation of 67743.2.

Another MOGA approach for generating nondominated sets of politcal redistricting plans is described by Vanneschi et al. [43]. Our method, described in the next section, uses a similar GA encoding, crossover operation, and mutation operation as those presented in Vanneschi et al. [43], although we add certain enhancements, such an improved method for generating the initial GA population. Their MOGA considers two objectives—population deviation, as defined by (1), and compactness—as well as some features which do not appear in our proposed method. For instance, in the final stage of each generation of the Vanneschi GA, a variable neighborhood search (VNS) is performed the plans in the GA population. Incremental improvements to the plans in the GA population were made by solving local search problems around selected units. In addition, the Vanneschi MOGA also copies the best plan from the previous generation to the GA population of the next generation, without crossover or mutation being applied. The authors demonstrated their results on 5 US states of various sizes, showing that for each considered problem instance, their method produces compact plans with low population deviation. The largest problem instance the authors considered was the census tract-level Pennsylvania instance, whose adjacency graph features 3218 nodes (tracts) and 8984 arcs. Although Lara-Caballero et al. [21] show similar performance on larger instances, the Vanneschi MOGA is described in detail in this section since our MOGA uses some of the same GA operations, including crossover and mutation.

4 Proposed method

In this section, we describe our multiobjective genetic algorithm (MOGA) for generating sets of redistricting plans. Our method uses a new initialization procedure for generating a set of feasible plans for the initial population of the genetic algorithm, which then uses the crossover and mutation operators proposed by Vanneschi et al. [43]. The MOGA considers three objectives:

- minimizing population deviation, as measured by (1)
- maximizing compactness, as measured by the minimum Polsby-Popper score (3) across all districts
- maximizing full-plan similarity, as measured by (5)

At the end of each generation of the GA, our method performs an iterative improvement to population deviation on each plan, similarly to Nagel [30]. Following the last generation of the genetic algorithm, our method returns entire set of nondominated plans for consideration.

Our method uses the GA encoding used proposed by Vanneschi et al. [43], which is described as follows. For an instance of the problem, the number of geographical units (census tracts) in the state in question is known. Calling this value S, a redistricting plan is encoded as an array s of length S, where the *i*th entry takes value k_i , where k_i is the district to which unit *i* is assigned. For example, if the third unit within a state is assigned to district 2 within a candidate solution s, then s(3) = 2.

In the redistricting problem, an adjacency graph G = (N, A) is used to represent the units in the plan and their adjacencies. The nodes N of the plan are the geographical units that will be separated into districts. For our analysis and experiments, N is the set of tracts defined by the 2010 US Census that compose the given state, and the set of arcs A represents adjacency relationships between the tracts. Using a definition from graph theory, a contiguous district k is a district whose nodes N_k are such that the *induced* subgraph G_k of G on the set of nodes N_k is a single connected component.

4.1 Initialization

The initialization procedure, described by Algorithm 1 in Section 7, produces an initial population to be used in the first iteration of the genetic algorithm. Given a desired population size N, the procedure begins by creating N copies of the base plan P_0 . For each of these copies, several iterations of the main swapping step are carried out. We define S(k) to be the set of districts that border district $k \in K$, where K is the set of districts. We also define N(a, b) to be the set of units in district $a \in K$ that border district $b \in K$. For each plan we do the following until an iteration count limit is reached: two districts of the plan are selected to receive and provide swaps of units on their border. First, we select a district k_{gain} from K according to a user-defined discrete probability distribution \mathbb{P}_{gain} . Our method will add units to this district. Since it is desirable to decrease population deviation via the swaps in this step, the discrete probability distribution \mathbb{P}_{gain} should be chosen so that less populous districts have a higher probability of being selected. Given the selection of k_{gain} , we construct the set $S(k_{gain})$ of districts which border k_{gain} . Then, our method randomly selects a second district k_{lose} from $S(k_{\text{gain}})$ according to another user-defined discrete probability distribution \mathbb{P}_{lose} . In order to encourage a decrease in population deviation, \mathbb{P}_{lose} should be chosen so that more populous districts in $S(k_{gain})$ have higher probability of selection. Once k_{gain} and k_{lose} have been chosen, we identify set of units $N(k_{\text{lose}}, k_{\text{gain}})$ in k_{lose} which border k_{gain} . We transfer the set $N(k_{\text{lose}}, k_{\text{gain}})$ from k_{lose} to k_{gain} , regardless of whether this transfer produces a discontiguous district.

In the final step of each iteration of the initialization procedure, we use a simple procedure to identify and repair any discontiguities within the current plan caused by the transfer of units in $N(k_{\text{lose}}, k_{\text{gain}})$. The repairing is done as in [21]. First, we examine each district $k \in K$, where K is the set of districts, to determine whether it is contiguous (by counting the components of its induced subgraph G_k). Once a district k is identified as discontiguous, the smallest connected component of G_k is identified. The set S_k of units within this smallest component that border some district other than k in the full graph G is also defined. Then, we transfer each unit in S_k to the district (other than k) which it borders. If the unit does not border some district other than k, we randomly choose another unit and transfer it to a neighboring district. Our method identifies these sets S_k and transfers them to other districts until every unit in the component has been reassigned. We repeat this transfer process for the smallest remaining connected component of G_k until only one connected component remains. Once only one connected component remains, the district kis contiguous. The swapping and repair process is illustrated in Figure 4.

Because this procedure uses the initial plan P_0 as its starting point, the procedure produces an initial population of plans that are more similar to P_0 than would be obtained using the randomized method presented in Vanneschi et al. [43]. The selection and crossover steps of the genetic algorithm, described in the next subsection, also promote similarity within in the population of plans. A comparison between similarity scores of nondominated plans generating using our MOGA and the Vanneschi et al. [43] MOGA is presented in Section 5.

Our experimentation shows that temporarily allowing for discontiguities improved the performance of the initialization method. Our first implementation, which we later rejected, offered units from k_{lose} to k_{gain} one at a time, and only completed the transfer if it maintained contiguity of each district in the plan. Using this approach sometimes restricts the set of possible swaps. For example, consider the units labeled A and B in Figure 4. Figure 4b shows that if unit B is to be transferred to the Blue district, then unit A is disconnected from the Purple district. If temporary discontiguities are not allowed, then unit B could not be transferred. Enclave units like this are common in certain US states, often arising as municipal entities called boroughs within townships. Other enclave units are present near the border of the Blue and Purple districts—when this is the case, the set of non-transferable units form a barrier between the two districts. Therefore, for units to be transferred between these two districts, the swaps are required to squeeze through gaps between non-transferable units, sometimes resulting in long, snake-like chains of units in the resulting districts. These chains negatively impact a district's compactness score under most compactness measures (including (3)), as they increase the ratio of the district's perimeter to its area, and increase the distance of its units from each other. Figure 5 demonstrates the difference in performance when allowing and disallowing for temporary discontiguities—without them, compactness seems to suffer, with long tendrils of units forming in some districts. The performance gap is larger in states with more enclave units. For instance, in the Pennsylvania example shown, allowing for temporary contiguities is more beneficial, as doing so avoids the pitfall of nontransferable districts and the resulting tendrils. However, in the Georgia example (Section 5, Figure 12),



(a) Original unit assignment, with Blue district chosen as $k_{\rm gain},$ Purple district chosen as $k_{\rm lose}$



(b) Set of Blue/Purple Districts border units in the Purple district identified (in green).



(c) Units in $N(k_{\text{lose}}, k_{\text{gain}})$ transferred from k_{lose} (Purple district) to k_{gain} (Blue district)



(d) Discontiguity repaired - assigned to Blue district

Figure 4: Transfer/Repair during Initialization



(a) Base Indiana Plan P_0



(b) Generated initial GA population plan - Without temporary discontiguities/repair



(c) Generated initial GA population plan - With temporary discontiguities/repair

Figure 5: Initial Plan Generation

there are far fewer enclave units, the performance of the two initialization methods was observed to be more similar.

When choosing the probability distributions \mathbb{P}_{gain} and \mathbb{P}_{lose} , there is a tradeoff between initializing lowdeviation plans and creating a diverse initial population. If \mathbb{P}_{gain} places weight evenly on each district for receiving unit transfers, then the resulting plans are less likely to resemble one other, at the cost of less reduction to population deviation. However, if higher weights are given to the less populous districts than to highly populous districts, then the resulting plans will have lower population deviations, though the plans will make more unit transfers, and therefore each plan is likely to resemble other plans within the initial GA population. We set \mathbb{P}_{gain} to have more weight on underpopulated districts, but give positive weight to every district. We discuss these details in Section 5.

4.2 Parent selection, crossover, mutation, and selection of next generation

Following the initialization procedure, the first generation of the GA begins. The crossover operation which we will use for our method, which is presented in Vanneschi et al. [43], is designed to combine candidate solutions in a way that preserves the contiguity of the districts in each plan. The operation works in two steps:

- 1. The tournament selection procedure (described later in this section) selects two plans. These plans are labelled p_1 and p_2 .
- 2. For every unit i with $p_1(i) = k_{gain}$ and $p_2(i) = k_{lose}$, our method does the following:
 - In p_1 , if assigning unit *i* to district k_{lose} results in a contiguous plan with no empty districts, then set $p_1(i) = k_{\text{lose}}$. Likewise, in p_2 , if assigning unit *i* to district k_{gain} results in a contiguous plan with no empty districts, then set $p_2(i) = k_{\text{gain}}$.

Defining p_m as the probability of a unit mutating, the following mutation operation for a given plan is also proposed by Vanneschi et al. [43] and is used in our approach:

• For each unit i in the plan, with probability p_m , we move unit i from its current district to a bordering district if doing so does not violate the contiguity of the plan, and does not produce an empty district.

Both the crossover and mutation operations ensure that the contiguity of each of the plans in the GA population is preserved throughout the generations of the GA.

We use the following method to sort and rank the plans within the GA population. Given a set P of N plans, these plans are sorted into fronts $\{F_1, F_2, \ldots, F_J\}$, with F_1 being the nondominated front, F_2 being the nondominated front if the plans in F_1 were removed from P, and so on. The $|F_1|$ plans within the set F_1 are ranked by the proximity of their population deviation to some "target" population deviation value, with the nearest plan being ranked first within P, and the furthest plan being ranked $|F_1|$ st. Vanneschi et al. [43] compute the median population deviation among plans in the front and uses this as the "target" by which plans in the front are compared. In contrast, in our algorithm, we generate the target M according to a uniform distribution U[a, b], where a is the lowest population deviation among plans in F_1 , and b is the highest population deviation among plans in F_1 . We break ties arbitrarily. We continue this process in the next front F_2 . The median of the second nondominated front M_2 is determined similarly, and the plans within F_2 are assigned ranks $|F_1| + 1$ through $|F_2|$. We repeat this ranking method for each of the J fronts until each of the |N| plans have received a rank.

Our ranking method is illustrated in Figure 6b, while an example of the ranking approach used in [43] in is shown in Figure 6a. Note that the sorting method does not directly consider the compactness objective. For this reason, the "2-objective" Pareto Front is shown in Figures 6a, 6b, and other figures relating to the ranking of plans. We define the 2-objective Pareto front as the subset of the overall (3-objective) Pareto front of plans that are nondominated when considering only the population deviation and similarity objectives. This approach differs from the nondominated sorting algorithm defined in Deb et al. [11], but we found that using our sorting method provides good experimental results. The motivation for randomizing the target point is to encourage diversity among the plans that are selected for crossover. Since the ranking method used by Vanneschi et al. [43] always sets the target point as the median, it may be biased towards plans with middling population deviation. Results on the diversity of plans produced by our method are given in Section 5. Our method also uses this modified ranking method to rank the plans at the end of each generation. If the size of the GA population is |N|, then |N|/2 crossover operations are executed to produce a new population to which the mutation operation is then applied. Our modified nondominated algorithm is illustrated in Algorithm 2.

We use this nondominated sorting algorithm at two stages in our MOGA. First, it is used in the tournament selection step for determining the plans that are used in the crossover, then again at the end of each generation when determining the population of the next generation. The tournament selection procedure selects two plans at a time, p_1 and p_2 from the population P for crossover. First, our method chooses p_1 by randomly drawing three plans from the set P, then applying the previously mentioned sorting method. The 1st ranked plan among these is the winner of the tournament, and is chosen as p_1 . The plan p_2 is selected similarly. Note that N/2 selection/crossover steps are carried out in each generation of the GA. The nondominated sorting algorithm is also used after mutation, at the end of each generation to determine which plans will be carried over into the next generation. Since we generally seek to improve the quality of plans from one generation to the next, we do the following (as in Vanneschi et al. [43]): We take the union of the GA population at the start of the generation (P_1) with the GA population after selection, crossover, and mutation have been performed (P_2) , to form a set P' of cardinality 2N. The nondominated sorting algorithm described previously is then applied to P', assigning each plan in the set a rank from 1 to 2N, where N is the size of the GA population. The N highest-ranked plans are chosen as P'', the set of plans to be carried over into the next generation of the GA. After applying the iterative improvement heuristic to these plans, described in the next subsection, we carry over this set of plans into the next generation of the GA. Algorithm 2 in Section 7 outlines the steps of our MOGA.



(b) Our nondominated sorting approach, randomly selecting target points for each front (in red)

Figure 6: Ranking Methods

4.3 Iterative improvement to population deviation

After forming the GA population P'' through crossover and mutation, we apply a simple heuristic to each plan $p \in P''$ to decrease the population deviation of plan p. For each plan $p \in P''$, several iterations of a swapping procedure similar to that of Nagel [30] are performed. For the selected plan p, in each iteration of the swapping step, two districts of the plan are selected. Two neighboring districts k_{gain} and k_{lose} are selected in the manner described in the initialization method from Section 4.1, using user-defined probability distributions \mathbb{P}_{gain} and \mathbb{P}_{lose} . Once k_{gain} and k_{lose} are chosen, the set of units $\mathbf{N}(k_{lose}, k_{gain})$ in k_{lose} which border k_{gain} is found. In a randomly generated order, each $i \in \mathbf{N}(k_{lose}, k_{gain})$ is checked, one at time, for whether its transfer to district k_{lose} would result in a discontiguous plan. If the resulting plan would be contiguous, then we transfer unit b from k_{gain} to k_{lose} . Also, for each unit $i \in \mathbf{N}(k_{lose}, k_{gain})$, the effect of the potential transfer on population deviation is checked—if population deviation would increase, the transfer is cancelled and two new districts k_{gain} and k_{lose} are chosen. At the end of this process, the iteration is complete. For each plan $p \in P''$, this iterative population deviation improvement heuristic runs for a predetermined number of iterations, or until a predefined tolerance (sufficiently low population deviation) is reached.

Our full method is described in Algorithm 2 of Section 7. The initialization algorithm is given by Algorithm 1. After the last generation of the GA, our method returns the full set of nondominated plans created throughout the run of the GA.

4.4 Multiobjective nondominated set measures

We are interested in comparing the "quality" of the Pareto fronts found by our heuristic compared to those from other methods. Roughly speaking, a Pareto front is of higher quality if it is "steep," meaning that as we move from one point (solution) to another, we obtain a large improvement in one objective with only a small degradation of the other. Steepness in this sense is a desirable property and is one way to evaluate how well a heuristically generated Pareto front approximates the true one.

Therefore, we introduce four measures of Pareto front quality. The definitions below assume that the Pareto front is plotted with population deviation (which we want to minimize) on the x-axis and similarity (which we want to maximize) on the y-axis, but the definitions can be adapted for other objective functions or their senses.

- 1-2 Slope: The slope between the two points with lowest population deviation in the set of nondominated plans.
- 1-M Slope: The slope between the point with lowest population deviation and the point with median population deviation among plans with $\leq 1\%$ population deviation.

- 1-F Slope: The slope between the point with lowest population deviation and the point with highest population deviation among plans with $\leq 1\%$ population deviation.
- Area Under Curve: The area underneath the part of the Pareto front containing plans with population deviation less than the predefined "low deviation" cutoff.

High values of these slope measures indicate that, on the 2-objective Pareto front, there is steep tradeoff between population deviation and similarity scores among plans in the nondominated set. That is, we can locate two plans in the nondominated set with high difference in similarity scores, but low difference in population deviation.

Figure 7 demonstrates the three slope measures. Figures 8 and 9 show that a nondominated set with a higher "area under curve" value is more likely to contain a diverse set of plans. The slopes reported will represent change in similarity over the change in tens of thousands of voters for population deviation.



Figure 7: Slope measure description

5 Results

In this section, we present computational results that show the effectiveness of our proposed heuristic algorithm, comparing the performance of our method with that of the method presented in Vanneschi et al. [43] where possible. In particular:

1. We report average-case results for the census-tract level redistricting problem for each of the 44 US states that currently require decennial redistricting. We also compare our results with the Vanneschi



Figure 8: Area under curve measure



Figure 9: Comparison of areas under curve

MOGA on the 5 states for which they report results. The average-case statistics of the returned nondominated plans are reported.

2. We demonstrate how the population deviation of the base plan affects the quality of the results returned by our method.

For each instance of the problem, we ran our MOGA 20 times. The tables in subsequent sections report the average performance of our method across these 20 runs. In particular, we report:

- Run time: average solution time, in seconds.
- Population deviation of initial plan P_0 : as demonstrated in Section 5.2, this value can impact the performance of the algorithm.

Parameter	Setting
GA population size N	30
GA generation count	15
Target population	(total state population) / $ \mathbf{K} $
Low-deviation cutoff	0.01μ
Crossover rate	0.6
Mutation rate	0.15
Max. iteration count for initialization method	$0.1 \times (\text{number of tracts})$
Max. iteration count for iterative improvement	$0.04 \times (\text{number of tracts})$

Table 3: Table of Parameter Settings

- ND Plans: number of nondominated plans produced.
- Low Dev. ND plans: number of plans whose population deviation is less than 1 percent of the target population for each district. For instance, the target population of a district in Georgia is 681,820. Therefore, for Georgia, low-deviation plans are plans with population deviation below 6818.
- Min. Dev. Min. Sim. Min. Comp.: minimum population deviation, similarity score, and compactness score (lowest Polsby-Popper score across districts) among plans in the nondominated set having low population deviation.
- Med. Dev. Min. Sim. Min. Comp.: median population deviation, similarity score, and compactness score (lowest Polsby-Popper score across districts) among plans in the nondominated set having low population deviation.

Table 3 describes the parameters associated with our MOGA and the chosen values used in our experiments.

In addition to the parameter settings listed in Table 3, we must also define the probability distributions \mathbb{P}_{gain} and \mathbb{P}_{lose} for initialization and iterative improvement steps (see Section 4.1). Suppose the districts are sorted in increasing order of population, so that k_n is the *n*th least populous district. We set $\mathbb{P}_{\text{gain}}(k_n) = 1.5\mathbb{P}_{\text{gain}}(k_{n+1})$ for $n \in \{1, ..., |\mathbf{K}| - 1\}$. Under this definition, the probability that k_n is selected is 50% greater than the probability that the next-most-populous district, k_{n-1} , is chosen. For example, if there are K = 3 districts, then $\mathbb{P}_{\text{gain}}(1) = 9/19$, $\mathbb{P}_{\text{gain}}(2) = 6/19$, and $\mathbb{P}_{\text{gain}}(3) = 4/19$. Similarly, the discrete probability distribution \mathbb{P}_{lose} is defined over the districts in $\mathbf{S}(k_{\text{gain}})$ as $1.5\mathbb{P}_{\text{lose}}(k_n) = \mathbb{P}_{\text{lose}}(k_{n+1})$ for $n \in \{1, ..., |\mathbf{S}(k_{\text{gain}})| - 1\}$, where k_n is the *n*th most populous district in $\mathbf{S}(k_{\text{gain}})$. These distributions are chosen so that less populous districts have a greater chance of being selected as the district k_{gain} to receive unit transfers, and more populous districts which border k_{gain} are more likely to be selected as the district k_{lose} transferring units into district k_{gain} .

These parameters were chosen after several rounds of testing. Increasing the number of plans in the population, and/or the number of generations, tends to increase the size of the returned nondominated set. Increasing mutation rate tends to increase diversity among plans in the nondominated set, but tends to increase population deviation in the nondominated set.

For the California, Texas, and New York problem instances, we use *mean* deviation (2) rather than the (absolute) population deviation to measure spread of district populations. The high number of districts in these problems made achieving a sufficiently low population deviation unrealistic. For example, in the California problem instance, the target population is 651378. In order to achieve a population deviation of less than 6513.78, the population of each district would have to vary from the target population by 125.25, on average. 7997 of the 8057 of California's census tracts have population greater than 125, and many have population greater than 10000. Balancing the population deviation to within 1 percent of the target population is therefore highly challenging, and possibly infeasible. Therefore, while for most problems we define a *low-deviation* plan as one having population having less than 1 percent of the target population, for the California, Texas, and New York problem instances, we require the average deviation across districts $k \in \mathbf{K}$ from the target population μ to be less than 1% of μ .

5.1 Performance: tract-level instances

In this section, we apply our algorithm to the US Congressional tract-level redistricting problem for each of the 44 states with more than one district. These experiments are conducted using available 2010 tract-level and US Census data from DeFord [12]. The maps produced in Section 8 were rendered using the GeoPandas Python package GeoPandas [16] and the shape file data compiled at Lykhovyd [23]. However, for each problem instance, we use the number of districts assigned by the 2020 reapportionment, since 2020 US Census data was not available in DeFord [12]. This means that the problem instances of Vermont, South Dakota, North Dakota, Alaska, and Wyoming, and Delaware are omitted, as they only comprise one congressional district.

For our analysis, we perform 20 runs of each problem instance. For each problem instance, we create a base plan P_0 using a modified version of the plan initialization method from Vanneschi et al. [43]. We do not consider the actual plan from the previous decade, since most of these plans are constructed using voting precincts, which are not made of census tracts, census blocks, or any geographical units for which we have US Census or adjacency data.

Table 4 reports general statistics for each problem instance. The results show that all instances of the problem can be solved in a matter of minutes by our MOGA. As expected, run time is shorter for smaller instances, such as New Hampshire and Montana, and is longer for the largest of problem instances, such as Pennsylvania, Florida, or California. In larger problem instances, we see that more of the returned

State	Nodes	Arcs	Districts	Run Time (seconds)	ND Plans	Low Dev. ND Plans
Alabama	1181	3295	7	234.3	48.9	35.5
Arkansas	696	1858	4	111.4	36.0	34.8
Arizona	1526	3930	9	357.3	47.8	41.4
California	8057	22069	52	9969.2	114.0	55.5
Colorado	1249	3360	8	156.7	40.9	34.4
Connecticut	833	2319	5	168.7	37.9	30.8
Florida	4245	11585	28	822.3	62.2	15.4
Georgia	1969	5491	14	384.4	117.2	34.8
Hawai'i	243	694	2	331.3	54.0	53.0
Idaho	298	792	2	74.4	48.0	47.0
Illinois	3123	8323	17	1159.3	59.6	20.1
Indiana	1511	4116	9	257.0	63.6	36.8
Iowa	825	2151	4	74.1	45.4	39.4
Kansas	770	2005	4	57.9	54.1	48.4
Kentucky	1115	3040	6	181.0	66.0	44.7
Louisiana	1148	3190	6	368.8	59.6	43.9
Maine	358	999	2	29.1	32.6	31.6
Maryland	1406	3788	8	243.3	58.8	41.7
Massachusetts	1478	4189	9	281.4	40.3	38.5
Michigan	2813	7438	13	534.4	52.5	41.3
Minnesota	1338	3640	8	180.0	60.1	47.3
Mississippi	664	1784	4	62.1	51.6	36.9
Missouri	1393	3743	8	120.9	56.9	37.4
Montana	271	706	2	10.3	41.4	41.4
Nebraska	532	1369	3	102.3	43.5	39.4
Nevada	687	1781	4	123.4	52.9	37.5
New Hampshire	295	786	2	38.6	37.1	36.1
New Jersey	2010	5575	12	297.3	69.5	28.3
New Mexico	499	1335	3	36.7	35.0	34.0
New York	4919	13474	26	2878.3	107.2	21.8
North Carolina	2195	6103	14	779.7	107.8	14.6
Ohio	2952	8159	15	564.3	53.2	24.4
Oklahoma	1046	2765	5	171.2	53.2	46.7
Oregon	834	2316	6	110.2	50.1	38.4
Pennsylvania	3218	8984	17	1716.3	50.8	16.4
Rhode Island	244	654	2	33.1	37.7	36.7
South Carolina	1103	3063	6	149.9	51.8	40.5
Tennessee	1497	4127	9	554.7	74.0	17.1
Texas	5265	14092	38	3185.6	81.4	55.2
Utah	588	1605	4	58.0	54.5	45.0
Virginia	1907	5263	11	322.8	76.4	29.3
Washington	1458	4061	10	278.2	76.7	31.7
West Virginia	484	1266	2	21.9	37.2	36.2
Wisconsin	1409	3857	8	236.6	37.5	30.8

Table 4: Average statistics across 20 runs: Summary Data

State	Min. Dev.	Med. Dev.	Min. Comp	Med. Comp.	Min. Sim.	Med. Sim.	P_0 Dev.
Alabama	1063	2155	0.074	0.140	0.649	0.735	787347
Arkansas	335	1464	0.054	0.070	0.808	0.904	207164
Arizona	1538	5837	0.048	0.088	0.489	0.616	2074789
California	22606	35679	0.018	0.022	0.506	0.521	18368083
Colorado	1175	2265	0.056	0.094	0.526	0.601	1253058
Connecticut	1188	2564	0.072	0.149	0.754	0.846	618853
Florida	1611	4922	0.039	0.062	0.465	0.499	6395205
Georgia	4639	6075	0.023	0.036	0.514	0.555	4564367
Hawai'i	2	258	0.032	0.052	0.737	0.756	132302
Idaho	5	435	0.048	0.078	0.509	0.561	140416
Illinois	4614	6451	0.031	0.049	0.476	0.532	5475499
Indiana	2427	4288	0.035	0.062	0.480	0.565	1932278
Iowa	563	2029	0.063	0.137	0.693	0.759	1046275
Kansas	451	1498	0.080	0.176	0.684	0.728	990488
Kentucky	1094	2244	0.025	0.045	0.594	0.664	2920792
Louisiana	963	2117	0.020	0.041	0.518	0.567	2770968
Maine	10	239	0.080	0.120	0.839	0.969	9311
Maryland	1363	2895	0.036	0.062	0.463	0.595	2267736
Massachusetts	1769	3772	0.043	0.089	0.538	0.607	1186597
Michigan	2861	4925	0.058	0.103	0.587	0.659	2995620
Minnesota	1769	3722	0.043	0.089	0.539	0.607	2229427
Mississippi	476	1605	0.067	0.106	0.801	0.856	357351
Missouri	2316	4316	0.056	0.084	0.633	0.799	688041
Montana	29	257	0.141	0.257	0.909	0.939	62151
Nebraska	117	880	0.078	0.137	0.687	0.851	417665
Nevada	289	1215	0.054	0.105	0.666	0.707	1300569
New Hampshire	29	874	0.138	0.236	0.891	0.971	15582
New Jersey	3138	5670	0.035	0.062	0.558	0.675	2265207
New Mexico	86	541	0.129	0.210	0.863	0.964	40414
New York	7989	11925	0.022	0.031	0.469	0.512	10766558
North Carolina	3857	5807	0.028	0.039	0.501	0.550	3680340
Ohio	4157	5980	0.043	0.074	0.574	0.642	3346747
Oklahoma	492	1747	0.048	0.085	0.640	0.827	354235
Oregon	1304	2555	0.041	0.076	0.674	0.776	669806
Pennsylvania	4893	6273	0.026	0.040	0.542	0.602	3722652
Rhode Island	25	1585	0.179	0.272	0.852	0.961	16453
South Carolina	790	1996	0.037	0.070	0.616	0.729	1168030
Tennessee	3133	4551	0.023	0.035	0.519	0.552	2644482
Texas	13557	20174	0.021	0.029	0.547	0.565	8561460
Utah	438	1471	0.060	0.122	0.666	0.760	951181
Virginia	2834	4693	0.041	0.049	0.561	0.603	2613543
Washington	2441	4531	0.025	0.052	0.523	0.558	2157684
West Virginia	45	972	0.078	0.122	0.917	0.967	44908
Wisconsin	1816	3389	0.054	0.102	0.593	0.666	1592858

Table 5: Average statistics across 20 runs: Population deviation, compactness, similarity

State	1-2 Slope	1-M Slope	1-F Slope	Area Under Curve
Alabama	1.57	0.74	0.39	0.78
Arkansas	0.32	0.05	0.04	0.95
Arizona	2.51	0.97	0.58	0.66
California	0.09	0.01	0.01	0.70
Colorado	1.38	0.33	0.29	0.64
Connecticut	6.93	1.24	0.97	0.87
Florida	0.08	0.05	0.02	0.62
Georgia	4.04	2.01	0.58	0.63
Hawai'i	18.13	0.92	0.10	0.76
Idaho	7.74	7.74	7.74	0.86
Illinois	0.71	0.83	0.32	0.59
Indiana	0.75	0.32	0.21	0.63
Iowa	0.98	0.37	0.19	0.75
Kansas	1.75	0.46	0.32	0.74
Kentucky	2.00	0.34	0.19	0.70
Louisiana	4.35	0.53	0.17	0.61
Maine	41.89	5.76	3.19	1.00
Maryland	5.84	1.08	0.28	0.65
Massachusetts	1.95	1.14	0.38	0.71
Michigan	1.57	0.21	0.23	0.69
Minnesota	0.47	0.36	0.24	0.67
Mississippi	0.87	0.35	0.12	0.88
Missouri	1.04	0.82	0.50	0.85
Montana	11.25	0.29	0.08	0.95
Nebraska	7.74	1.83	0.37	0.89
Nevada	2.06	1.69	0.57	0.74
New Hampshire	8.40	2.54	1.07	1.00
New Jersey	0.35	0.48	0.25	0.68
New Mexico	10.52	8.15	1.33	0.99
New York	1.12	0.62	0.52	0.66
North Carolina	0.06	0.12	0.44	0.69
Ohio	0.28	0.29	0.22	0.70
Oklahoma	7.91	2.57	1.14	0.88
Oregon	5.41	4.31	4.12	0.83
Pennsylvania	0.34	0.42	0.27	0.66
Rhode Island	2.05	1.51	0.38	1.00
South Carolina	2.94	0.85	0.35	0.80
Tennessee	0.45	0.43	0.24	0.67
Texas	0.01	0.02	0.02	0.64
Utah	1.82	0.74	1.11	0.12
Virginia	0.90	0.29	0.11	0.68
Washington	0.38	0.30	0.13	0.61
West Virginia	4.45	0.83	0.27	1.00
Wisconsin	0.45	0.35	0.23	0.69

Table 6: Average statistics across 20 runs: Slope data. Results are multiplied by 10^4 .



Figure 10: Example 2-objective Pareto Front using our method (blue) vs. the Vanneschi method (red)

nondominated plans, on average, tend to fall outside of the low-deviation threshold. However, for all of the states, our MOGA returns several low population deviation plans.

Table 5 shows the average statistics among low-devation nondominated plans. The table also lists the population deviation of the base plan P_0 used for each state. We can see that solution quality is better for the smaller problem instances as well—on average, population deviations in the returned set of nondominated plans for the smaller problem instances tend to be lower than for larger problem instances. Similarly, the average minimum and median compactness values in the nondominated set across in 20 runs are higher for larger problem instances. For problem instances with larger population deviations in their base plan P_0 , the results showed lower similarity in the returned plans. The relationship between the population deviation of P_0 and similarity scores is examined further in Section 5.2.

Figure 11 describes an example of a low-deviation, nondominated plan and its accompanying base plan, along with some relevant statistics. In this Missouri example (Figure 11) in which the pair of plans has similarity score greater than 0.8, the resemblance between the base and solution plan is easily visible. In the Pennsylvania example (Figure 14), the similarity score between the two plans is lower (0.628), likely in part due to the high population deviation of the base plan. However, the districts of the nondominated plan produced by the genetic algorithm are still clearly based on the districts of the base plan, with several



Figure 11: Missouri map comparison: Similarity Score = 0.827

districts in the southeastern region of the state largely intact. The results also show that several of our nondominated plans for California, Texas, and New York have mean deviation of less than 0.2 percent of the target population, which is lower than that of any plan reported by Lara-Caballero et al. [21].

Some of the 44 tract-level problem instances that we consider have unique characteristics which needed to be handled outside of the previously described framework. For example, in the California problem instance, certain tracts representing islands are not adjacent to any tracts on the mainland. As a result, any proposed plan is labeled as discontiguous by the algorithm, which prevents several operations (such as swapping during initialization, crossover, and mutation) from working as intended. Similarly, in the Florida problem instance, an island tract in the Florida Keys archipelago is not adjacent to any other tract. To resolve this issue, we removed these tracts prior to running the genetic algorithm, then added them to the nearest districts in the finished plans produced by the genetic algorithm. We address neighborless units in New York, Rhode Island, and California similarly. In these examples, the island tracts assigned to districts after the final generation of the genetic algorithm have an insignificant effect on performance, since these tracts are unpopulated or sparsely populated.

A different customization was needed for solving the Hawai'i problem instance, whose tracts are divided into two equally populous units. Since Hawai'i is an archipelago consisting of 8 main islands, the corresponding tract adjacency graph is disconnected and separated into 8 connected components. In practice, the most populous island, O'ahu, is the only district which is split between the two districts, while the other 7 islands share a common district. Therefore, to produce solutions that satisfy this practical constraint and to avoid the problem of disconnected components, the tracts from the other 7 islands are removed from the tract set and adjacency graph, and are assigned to the 1st district. The remaining O'ahu tracts are then divided between the two districts by the MOGA. The results in Table 5 use this approach; the tract and arc counts

	Our Method				Vanneschi et al. [43]					
State	Median Similarity	1-2	1-M	1-F	AUC	Median Similarity	1-2	1-M	1-F	AUC
NH	0.97	8.40	2.54	1.07	1.00	0.62	6.33	0.18	0.11	0.63
NE	0.85	7.74	1.83	0.37	0.89	0.71	0.08	0.10	0.04	0.71
NM	0.76	0.98	0.37	0.19	0.75	0.65	0.04	0.03	0.04	0.63
ID	0.78	5.41	4.31	4.12	0.90	0.77	1.26	0.15	0.05	0.72

Table 7: Similarity/slope comparison - average results across 20 runs

given are only those contained within O'ahu.

For each problem instance, the average values of the slope and area under curve statistics defined in Section 4 for each state are given in Table 6. Without context, these values are difficult to interpret. Accordingly, we can compare these slope values returned by our method with the average slope values returned by our implementation of the the Vanneschi method across 20 problem instances. Consider, for example, a medium-sized problem instance such as Nebraska. Figure 10 shows the 2-objective (population deviation and similarity) Pareto front on a set of nondominated plans generated by our method, as well as the 2-objective Pareto front for a nondominated set resulting from of 1 of the 20 problem instances. Table 7 reports average slopes across 20 runs of both our method and the Vanneschi method, for Nebraska and a few other representative states. The "Median Similarity" column reports the average value of the median similarity among the nondominated plans returned by each algorithm. These results demonstrate that, unlike the Vanneschi method, our method can identify high-similarity plans. In addition, these results show that for our method there is a steep tradeoff on the 2-objective Pareto curve between low deviation plans and high-similarity plans.

5.2 Varying population deviation of base plan P_0

Our initial experiments suggested that the population deviation in the base plan P_0 affected the performance of our MOGA, especially in producing plans similar to P_0 . Intuitively, a plan with higher population deviation must be more significantly altered to meet legal requirements than a plan with low population deviation. Therefore, the plans in the nondominated set returned by our algorithm are likely to have higher similarity scores (with the base plan P_0) when P_0 has lower population deviation. To illustrate the effect of population deviation of the base plan P_0 on similarity of the output plans, we examine five Georgia redistricting plans. Figures 12a and 12b show the districts of Plan 1 (population deviation 132051) and Plan 4 (population deviation 698901). All five of these plans were generated using the initialization method presented in Vanneschi et al. [43].

For each of these 5 base plans, we performed 20 runs of our method and observed the average similarity scores of the low-devation non-dominated plans under each base plan P_0 . The results in Table 8 demonstrate

Plan	P_0 Deviation	Mean Similarity
1	132051	0.931
2	277855	0.909
3	521747	0.855
4	698901	0.833
5	1624123	0.781

Table 8: Statistics across 20 runs: Average similarity scores of nondominated low-deviation plans with P_0 . "Low" population deviation: less than 6818

how the quality of solutions found by our method depends on the quality of the base plan P_0 . There is an inverse relationship between similarity of the returned plans (with P_0) and the population deviation of the base plan.

Since quality of plans seems to depend on quality of the base plan P_0 , the results motivate the need for our customized initialization method, which produces an initial GA population with low population deviation plans. In this and in previous experiments, several of the base plans considered have population deviations that are unrealistically high. The base plans used for our tract-level instance results were generated randomly using the plan initialization method from Vanneschi et al. [43], which often struggles to create plans which have a reasonably low population deviation. For instance, in the fifth experiment/row of Table 8, a plan with population deviation of over 1.6 million is considered. In practical decennial redistricting, a base plan is unlikely to have such a high population deviation, since ten years prior, the plan was likely required to have roughly equal populations among its districts, and such large shifts in population within a state are unlikely to occur. As demonstrated in Table 1, the population deviation of Georgia's 2012–2021 plan using 2020 Census data is 418169 DRA [14]. In this experiment and the set of results in Section 5, the population deviation of each P_0 is often significantly higher than the real-world population deviations shown in Table 1. That our method can produce sets of low-deviation, nondominated plans despite these high-deviation base plans is evidence of its strength in handling difficult problem instances.

6 Discussion and conclusion

In this paper, we motivated the need for considering similarity in algorithms for redistricting, both because of the preferences of politicians and voters, and because, in practice, redistricting plans are typically based on the previously used plan. We introduced a multiobjective genetic algorithm (MOGA) to find redistricting plans that balance three objectives: compactness, population deviation, and similarity. Our experimental results show that our MOGA generates sets of high-quality, nondominated redistricting plans. Our new method for initializing the initial population of the genetic algorithms aids performance of the MOGA by providing lower-deviation initial plans than would have been created using the existing methods, as well as



(b) Georgia plan with population deviation 698901

(a) Georgia plan with population deviation 132051

Figure 12: Georgia deviation map comparison

ensuring that the plans in the initial population resemble a given base plan. Since this initialization method is separate from the crossover, mutation, and the other operations of this or any MOGA, it could be used to initialize the starting population in other MOGAs for redistricting. Our experimental results suggest that our method can quickly return high-quality redistricting plans for any realistically-sized problem instance. Furthermore, our method produces plans that are similar to some base plan P_0 , which is a key criterion in practical redistricting.

In future work, several modifications of our algorithm and the problem we studied could be considered. For instance, the crossover and mutation operations from Vanneschi et al. [43] could be replaced with other crossover and mutation operations, such as those found in Xiao [46]. The recombination approach for crossover could potentially lead to better performance, especially in finding a more diverse set of nondominated plans. Lastly, changing the similarity measure used, perhaps to one of the many other measures introduced in Becker and Gold [7], could affect the performance of our algorithm. In addition, while we did not consider objectives involving political party data, such as partisan composition of districts, proportionality, partisan symmetry, and competitiveness, these could be considered by our method, which is designed to accommodate an arbitrary number of objectives. Another important aspect of redistricting that our work does not consider is fairness with respect to race, income, or other demographics. Our MOGA could be adapted, for example, to encourage the formation of so-called majority-minority districts that help protect that the voting power of minority groups, or preserve districts of this type from a previous plan.

7 Algorithms

Algorithm 1 Genetic Algorithm Population Initialization

1: N - desired population size 2: P_0 - initial plan 3: M - max. number of iterations allowed 4: P[k] - set of units in district k of plan P 5: Loc[i] - district location of unit i6: B[i][j] - Boundary set: set of units in district j that border district i 7: Initialize B, Loc, Pfor $i \in \{1, ..., N\}$ do 8: $P_i = P_0$ 9: iter = 010: while iter < M do 11:12: iter = iter + 1Sort districts from most to least populous. 13:Select district k_{gain} according to probability distribution \mathbb{P}_{gain} . 14:Create probability distribution \mathbb{P}_{lose} for selecting the neighboring district k_{lose} which will transfer 15:units to k_{gain} Select district k_{lose} to transfer border units to k_{gain} , using \mathbb{P}_{lose} . 16: $Border = B[k_{gain}][k_{lose}]$ 17:18: $P_i[k_{\text{lose}}] = P_i[k_{\text{lose}}] \setminus Border$ $P_i[k_{\text{gain}}] = P_i[k_{\text{gain}}] \bigcup Border$ 19:for each unit $u \in Border$ do 20: 21: $Loc[u] = k_{gain}$ end for 22:A[u]: set of nodes adjacent to u23:for $u \in Border$ do 24:for $a \in A[u]$ do 25: $B[Loc[a]][k_{lose}] = B[Loc[a]][k_{lose}] \setminus \{u\}$ 26: $B[k_{\text{lose}}][Loc[a]] = B[k_{\text{lose}}][Loc[a]] \setminus \{a\}$ 27: $B[k_{gain}][Loc[a]] = B[k_{gain}][Loc[a]] \cup \{a\}$ 28: $B[Loc[a]][k_{gain}] = B[Loc[a]][k_{gain}] \cup \{u\}$ 29:end for 30: 31: for $k \in \{1, ..., K\}$ do 32: if $u \in B[k][k_{\text{lose}}]$ then $B[k][k_{\text{lose}}] = B[k][k_{\text{lose}}] \setminus \{u\}$ 33: end if 34: end for 35: end for 36: 37: end while Repair discontiguities using method described in Section 4 [21] 38:39: end for 40: Return P

Algorithm 2 New Multiobjective Heuristic

- 1: N desired population size
- 2: G desired number of generations
- 3: P_0 initial plan
- 4: P Genetic algorithm population (set of redistricting plans)
- 5: $P = Initialization(P_0, N, M)$ (Section 4.1, Algorithm 1)
- 6: Nondom = \emptyset : set of plans that are nondominated at the end of some generation $g \in \{1, \ldots, G\}$
- 7: for $g \in \{1, \ldots, G\}$ do
- 8: Make copy $P_1 = P$
- 9: **for** $i \in \{1, ..., N/2\}$ **do**
- 10: Select two plans from P_1 according to selection procedure described in Section 4.2
- 11: Perform crossover to each pair of plans as described in Section 4.2
- 12: end for
- 13: Perform mutation on each plan in P_1 as described in 4.2
- 14: $P_2 = P \cup P_1$
- 15: Add nondominated plans in P_1 to Nondom
- 16: Rank each of the 2|N| plans in P_2 according to 4.2
- 17: Select N best plans as new population P
- 18: Perform iterative improvement to each plan in P as described in 4.3
- 19: **end for**
- 20: Find and return nondominated plans in Nondom

8 Maps

9 Appendix



(a) Arizona base plan ${\cal P}_0$ with population deviation 1524923

(b) Solution plan with population deviation 4112

Figure 13: Arizona map comparison: Similarity Score = 0.538





(a) Pennsylvania base plan ${\cal P}_0$ with population deviation 3722052

(b) Solution plan with population deviation 6290

Figure 14: Pennsylvania map comparison: Similarity Score = 0.628

State	Districts (2012–2021)	Deviation under 2020 Census	Avg. Sim.	Max. Sim.	Min. Sim.
AL	35	302523	0.774	1.000	0.512
AK	20	36906	0.686	1.000	0.406
AZ	30	459519	0.540	1.000	0.278
CA	40	1333660	0.604	1.000	0.252
CO	35	420133	0.576	1.000	0.145
CT	36	126420	0.942	1.000	0.527
DE	21	90641	0.695	0.945	0.466
FL	40	1183020	0.646	0.973	0.358
\mathbf{GA}	56	625883	0.691	1.000	0.289
HI	25	123244	0.808	1.000	0.503
ID	35	171072	0.679	1.000	0.240
IL	59	442349	0.573	0.896	0.239
IN	50	359637	0.601	1.000	0.337
IA	50	230395	0.540	1.000	0.259
KS	40	186637	0.688	1.000	0.249
KY	38	253815	0.616	1.000	0.286
LA	39	313298	0.719	1.000	0.253
ME	35	68844	0.756	1.000	0.359
MD	47	369932	0.800	1.000	0.329
MA	40	345933	0.735	1.000	0.309
MI	38	592741	0.488	1.000	0.216
MN	67	260811	0.694	1.000	0.300
MS	52	233931	0.702	1.000	0.323
MO	34	335795	0.673	1.000	0.356
MT	50	60461	0.616	1.000	0.257
NE	49	151127	0.729	1.000	0.348
NV	21	230510	0.631	0.942	0.306
NH	24	32713	0.756	1.000	0.285
NJ	40	358824	0.736	1.000	0.289
NM	42	124385	0.648	0.994	0.241
NY	63	1292120	0.509	1.000	0.262
NC	50	880785	0.575	1.000	0.297
ND	47	133794	0.725	1.000	0.264
OH	33	574091	0.679	1.000	0.310
OK	48	309863	0.646	1.000	0.322
OR	30	157861	0.632	0.999	0.235
PA	50	717170	0.654	1.000	0.347
RI	38	31167	0.793	1.000	0.427
\mathbf{SC}	46	570101	0.688	1.000	0.276
SD	35	71611	0.687	1.000	0.274
TN	33	493766	0.662	1.000	0.313
TX	31	1939900	0.646	1.000	0.323
UT	29	312059	0.774	1.000	0.369
VA	40	542862	0.425	0.765	0.239
WA	49	316015	0.670	0.955	0.367
WV	17	114309	0.702	1.000	0.473
WI	33	173844	0.863	1.000	0.640
WY	30	33389	0.730	1.000	0.318

Table 9: Map comparison statistics: State Senate, 2012–2021 vs. 2022–2031





(a) New York base plan P_0 with population deviation 10766558

(b) Solution plan with population deviation 7438



10 Appendix

Data Availability Statement: The data and resources uses in our experimentation (DRA [14], Lykhovyd [23], DeFord [12]) are available and with links provided in the References section.

State	Districts (2012–2021)	Deviation under 2020 Census	Avg. Sim.	Max. Sim.	Min. Sim.
AL	105	407488	0.670	1.000	0.285
AK	40	44322	0.706	1.000	0.277
AZ	30	459519	0.540	1.000	0.278
CA	80	1527170	0.582	0.960	0.246
CO	65	511124	0.572	1.000	0.233
CT	151	139332	0.803	1.000	0.320
DE	41	76904	0.818	1.000	0.239
FL	120	1386340	0.588	1.000	0.203
GA	180	768307	0.573	1.000	0.203
HI	51	146858	0.731	1.000	0.375
ID	35	171072	0.679	1.000	0.240
IL	118	515166	0.537	0.982	0.243
IN	100	378866	0.670	1.000	0.218
IA	100	253790	0.588	1.000	0.229
KS	125	214949	0.740	1.000	0.228
KY	100	290155	0.569	1.000	0.195
LA	105	376002	0.739	1.000	0.241
ME	151	68922	0.742	1.000	0.230
MI	110	598263	0.496	1.000	0.206
MN	134	294195	0.626	1.000	0.266
MS	122	278716	0.758	1.000	0.213
MO	163	419428	0.529	1.000	0.233
NE	49	151127	0.729	1.000	0.348
NV	42	295288	0.553	0.965	0.329
NJ	40	358824	0.736	1.000	0.289
NM	70	140200	0.642	1.000	0.198
NY	150	705556	0.793	1.000	0.270
NC	120	969887	0.621	1.000	0.241
OH	99	669981	0.595	1.000	0.232
OK	101	359949	0.714	1.000	0.254
OR	60	172884	0.619	1.000	0.259
PA	203	658332	0.588	1.000	0.258
RI	75	40439	0.868	1.000	0.378
\mathbf{SC}	124	591401	0.717	1.000	0.288
SD	37	110042	0.693	1.000	0.274
TN	99	561846	0.707	1.000	0.229
TX	150	2986150	0.644	1.000	0.277
UT	75	375546	0.706	1.000	0.329
VA	100	572734	0.478	1.000	0.266
WA	49	316015	0.670	0.955	0.367
WV	67	799200	0.557	1.000	0.167
WI	99	219666	0.758	1.000	0.324
WY	60	43161	0.724	1.000	0.345

Table 10: Map comparison statistics: State House, 2012–2021 vs. 2022–2031

References

- [1] (2018). League of Women Voters, et al. v. the Commonwealth of Pennsylvania, et al. (159, 2017). https://www.pacourts.us/news-and-statistics/cases-of-public-interest/ league-of-women-voters-et-al-v-the-commonwealth-of-pennsylvania-et-al-159-mm-2017.
- [2] (2023). Michigan Independent Citizens Redistricting Commission. https://www.michigan.gov/micrc.
- [3] Abrishami, T., Guillen, N., Rule, P., Schutzman, Z., Solomon, J., Weighill, T., and Wu, S. (2020). Geometry of graph partitions via optimal transport. *Methods and Algorithms for Scientific Computing*, 42(5):A3341–A3366.
- [4] Ambarian, J. (2021). Montana redistricting maps. https://missoulacurrent.com/ montana-redistricting-maps/.
- [5] Balmert, J. (2021). Ohio republicans pass a new congressional map. will it pass ohio supreme court scrutiny? *The Columbus Dispatch*.
- [6] Becker, A., Duchin, M., Gold, D., and Hirsch, S. (2021). Computational redistricting and the voting rights act. *Election Law Journal: Rules, Politics, and Policy*, 20(4):105–155.
- [7] Becker, A. and Gold, D. (2022). The gameability of redistricting criteria. Journal of Computational Social Science, 5:1735—1777.
- [8] Bozkaya, B., Erkut, E., and Laporte, G. (2003). A tabu search heuristic and adaptive memory procedure for political districting. *European Journal of Operational Research*, 144(1):12–26.
- [9] Caro, F., Shirabe, T., Guignard, M., and Weintraub, A. (2004). School redistricting: embedding GIS tools with integer programming. *Journal of the Operational Research Society*, 55(8):836–839.
- [10] Cook, E. (2007). Legislative guide to redistricting in iowa. https://www.legis.iowa.gov/docs/ publications/LG/9461.pdf.
- [11] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.
- [12] DeFord, D. (2021). Dual graphs for 2010 census units. https://www.math.wsu.edu/faculty/ ddeford/dual_graphs.html.
- [13] Deford, D., Duchin, M., and Solomon, J. (2021). Recombination: A family of markov chains for redistricting. *Harvard Data Science Review*, 3(1).

- [14] DRA (2020). Dave's redistricting app 2020. https://davesredistricting.org/maps#home. Accessed: 2022-04-01.
- [15] Gentry, S. and Chow, E. (1996). Gerrymandering for justice: Redistricting u.s. liver allocation. Operations Research, 13(6):522–526.
- [16] GeoPandas (2021). Geopandas. https://geopandas.org/en/stable/.
- [17] Gurnee, W. and Shmoys, D. (2021). Fairmandering: A column generation heuristic for fairness-optimized political districting. In *Proceedings of the 2021 SIAM Conference on Applied and Computational Discrete Algorithms (ACDA21)*, SIAM, pages 88–99, New York, NY. SIAM.
- [18] Hayes, B. (1996). Machine politics. American Science, 84(6):522–526.
- [19] Hess, S., Weaver, J., Seigfeldt, H., Whelan, J., and Zitlau, P. (1996). Nonpartisan political redistricting by computer. Operations Research, 13(6):522–526.
- [20] Kaiser, H. (1966). An objective method for establishing legislative districts. Midwest Journal of Political Science, 10(2):200–213.
- [21] Lara-Caballero, A., de-los-cobos Silva, S., Mora-Gutiérrez, R., Rincón-García, E., Gutiérrez-Andrade, M., and Lara-Velásquez, P. (2019). Multiobjective genetic algorithms for reinforcing equal population in congressional districts. Annals of the Association of American Geographers, 98(4):795–817.
- [22] Liu, Y., Tam Cho, W., and Wang, S. (2016). PEAR: a massively parallel evolutionary computation approach for political redistricting optimization and analysis. *Swarm and Evolutionary Computation*, 30:78–92.
- [23] Lykhovyd, E. (2020). Index of /files/public/districting/. https://lykhovyd.com/files/public/districting/.
- [24] Maptitude (2022). Maptitude. https://www.caliper.com/maptitude/solutions/ alternative-to-bard-better-automated-redistricting.htm.
- [25] McDonald, M. and Altman, M. (2009). BARD: Better automated redistricting. Environment and Planning B: Urban Analytics and City Science, 36(6):1053–1066.
- [26] McKee, S., Shino, E., and Smith, D. (2022). Redrawn, withdrawn: Effects of redistricting on the representative constituent relationship. *Election Law Journal*, 21(4):1–26.
- [27] Mehrotra, A., Johnson, E., and Nemhauser, G. (1998). An optimization based heuristic for political districting. *Management Science*, 44(8):1100–1114.

- [28] Moncrief, G. (2012). Reapportionment and redistricting in the West. Lexington Books.
- [29] Mutnick, A. (2022). GOP considers more ruthless redistricting. *Politico*.
- [30] Nagel, S. (1965). Simplified bipartisan computer redistricting. *Stanford Law Review*, 17:863–899.
- [31] Polsby, D. and Popper, R. (1991). The third criterion: Compactness as a procedural safeguard against partisan gerrymandering. Yale Law & Policy Review, 9:301–353.
- [32] Reock, E. (1961). A note: Measuring compactness as a requirement of legislative apportionment. Midwest Journal of Political Science, 5:70–74.
- [33] Shirabe, T. (2009). Districting modeling with exact contiguity constraints. Environment and Planning B: Urban Analytics and City Science, 36(6):1053–1066.
- [34] Stephanopoulous, N. and McGhee, E. (2015). Partian gerrymandering and the efficiency gap. University of Chicago Law Review, 82:831–900.
- [35] Swamy, R., King, D., and Jacobson, S. (2022). Multiobjective optimization for politically fair districting: A scalable multilevel approach. *Operations Research*, Articles in Advance:1–17.
- [36] Colleton County Council v. McConnell (2002). Colleton County Council v. McConnell, 201 F. Supp.2d 618 (District Co
- [37] Colorado State Legislature (1966). Colorado state constitution. https://law.justia.com/ constitution/colorado/cnart5.html.
- [38] Connecticut General Assembly (2023). 2021 congressional redistricting plan maps. https://www.cga. ct.gov/rr/tfs/20210401_2021%20Redistricting%20Project/cmaps.asp.
- [39] Kentucky General Assembly (2023). 2022 redistricting maps. https://legislature.ky.gov/Public% 20Services/GIS/Maps/Pages/2022-Redistricting-Maps.aspx.
- [40] Shaw v. Reno (1993). Shaw v. Reno, (509 U.S. 630). https://www.oyez.org/cases/1992/92-357.
- [41] US Census Bureau (2020). US Census. https://www.census.gov/newsroom/press-kits/2021/
 2020-census-redistricting.html. Accessed: 2022-10-01.
- [42] Validi, H., Buchanan, A., and Lykhovyd, E. (2022). Imposing contiguity constraints in political districting models. Operations Research, 70(2):867–892.
- [43] Vanneschi, L., Henriques, R., and Castelli, M. (2017). Multi-objective genetic algorithm with variable neighbourhood search for the electoral redistricting problem. *Swarm and Evolutionary Computation*, 36:37–51.

- [44] Wesberry (1964). Wesberry v. Sanders, 376 U.S. 1.
- [45] Williams, J. (1995). Political redistricting: A review. Papers in Regional Science, 74(1):13-40.
- [46] Xiao, N. (2008). A unified conceptual framework for geographical optimazation using evoluationary algorithm. Annals of the Association of American Geographer, 98(4):795–817.