

# Optimal de-gerrymandering of voting districts with differential evolution

Rory Lipkis

*Department of Aeronautics and Astronautics, Stanford University*

## Abstract

A mathematical model is developed to explore the problem of drawing voting districts to replicate democratic preferences in a body of elected representatives. The model is optimized using differential evolution, and the quality, stability, and interpretation of the solutions is discussed.

## Introduction

Gerrymandering is the practice of modifying the boundaries of voting districts in order to enforce an electoral outcome that does not reflect the distribution of preference in the general population. Specifically, in representative democracies in which each voting district elects a delegate to a congressional or parliamentary body in winner-takes-all election, gerrymandering produces a gap between the preference distribution of the population and that of the congress.

This can be accomplished in a number of ways. A particular faction of voters can be divided in order to ensure that they remain a minority in any given district and thus elect no delegates. Alternatively, a faction can be consolidated in order to ensure that they hold a majority in only a few districts, and thus elect as few delegates as possible. Gerrymandering does not necessarily entail drawing convoluted district boundaries. Different choices of seemingly unbiased boundaries, such as a simple grid, can produce wildly different outcomes, depending on the specific parameters and placement. Figure 1 shows an example of how district boundaries can affect the composition of a congress. This can lead to gridlock in discussions about gerrymandering, as there is rarely a configuration that is both simple and completely neutral.<sup>1</sup> The purpose of this paper is to explore configurations that are to the greatest extent simple, neutral, and stable, by various metrics to be developed in the following sections.

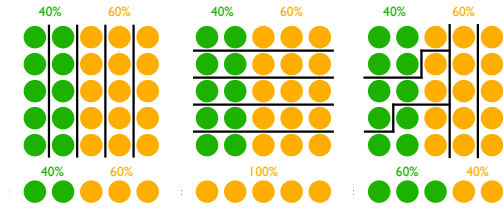


Figure 1: Three drawings of even districts, and the resulting congressional outcomes.

## Problem statement

For a given population of  $n$  people, each with a location  $\mathbf{x} \in \mathbf{R}^2$  and affiliation  $a \in \{1 \dots m\}$ , draw  $k$  voting districts such that:

- Districts are convex.
- Districts have nearly equal populations.
- The distribution of plurality affiliations across districts is maximally similar to the distribution of affiliations among the general population.

## Approach

### Design space

The districts can be efficiently specified by Voronoi regions. In this formulation, a district center  $\mathbf{c}_i \in \mathbf{R}^2$  defines the region for which it is the closest center for all interior points:

$$D_i = \{\mathbf{x} \in \mathbf{R}^2 \mid d(\mathbf{x}, \mathbf{c}_i) \leq d(\mathbf{x}, \mathbf{c}_j) \text{ for all } j \neq i\}$$

<sup>1</sup>Complicating such discussions is the fact that one party typically has a vested interest in maintaining the disproportionate outcomes of gerrymandering, a phenomenon that this paper does not explore.

where the distance metric  $d(\cdot, \cdot)$  is typically taken to be the Euclidian distance. In that case, it is easily shown that Voronoi regions are equivalent to intersections of halfspaces, and thus the convexity requirement is automatically satisfied. The problem then reduces to a more tractable optimization over  $\mathbf{R}^{2k}$ , the design space formed from the concatenation of the district centers.

## Objective function

The objective function takes a design point, unpacks it into a set of districts, then assigns the members of the population to their appropriate districts according to the Voronoi criterion. It then simulates voting for congressional delegates. This process is efficiently implemented through the use of specialized classes, though the evaluation can still be expensive for large populations.

The *preference inequality* is computed as the  $L_2$  distance between the normalized distribution of affiliations among the general population and among congressional delegates, and is scaled to lie between 0 and 1.

$$J_p = \left[ \frac{1}{2} \sum_a \left( \frac{n_{a,\text{gen}}}{n} - \frac{n_{a,\text{del}}}{k} \right)^2 \right]^{1/2}$$

where  $n_a$  is the number of people or delegates with affiliation  $a$ . The *size inequality* is computed as the discrepancy of district sizes from an equal partition, and is scaled to lie between 0 and  $\sqrt{k(k-1)}/k \approx 1$ .

$$J_s = \left[ \sum_d \left( \frac{n_d}{n} - \frac{1}{k} \right)^2 \right]^{1/2}$$

where  $n_d$  is the number of people assigned to district  $d$ . This term represents a penalty function for the equal population requirement. It is necessary to discourage trivial solutions in which preference inequality is satisfied through a highly unequal partitions (small districts formed around especially partisan areas). The total cost is given by  $J = J_p + J_s$ .

The objective function is highly nonlinear, exhibiting three features that make the task of optimization especially difficult:

- (a) Plateaus occur when a change in the design variable fails to change the allocation of people to districts.
- (b) Discontinuities occur when a small change in the design variable tips a district vote.
- (c) Symmetries occur due to the  $n!$ -fold degeneracy of the district center placement, which is independent of order.

The objective function is discretely valued, with two separate scales of discretization associated with  $J_p$  and  $J_s$ . Therefore, the optimization “landscape” is entirely composed of plateaus and discontinuities.

## Optimization

Due to the objective function’s nonlinearity and lack of a gradient approximant, a population method is an appropriate approach. A genetic algorithm was selected for its ability to offer reasonably robust optimization over a moderately high-dimensional space. In particular, the method of differential evolution<sup>2</sup> provides a good heuristic for generating successful mutations by searching for linear combinations of members of the candidate population.

The plateaus and discontinuities are mitigated to an extent by the addition of  $J_s$ . The combinatorial symmetry, however, greatly obstructs the genetic algorithm. Since the design vector is symmetric under re-ordering (provided that the components of any individual center remain adjacent), the crossover procedure often fails to general meaningful variation in the population. This leads to a general failure of genetic algorithms to converge, since two candidate points can be very far apart in the design space but have identical objective values. The production of such offspring tends to pull the population away from convergence.

However, this symmetry is fully reducible with the application of an ordering constraint. Since all  $k!$  permutations are identical under some imposed ordering, it is sufficient to choose an arbitrary reference  $\mathbf{c}_0$  and metric  $L_p$ , then impose the system of constraints

$$|\mathbf{c}_1 - \mathbf{c}_0|_p \leq |\mathbf{c}_2 - \mathbf{c}_0|_p \leq \dots \leq |\mathbf{c}_k - \mathbf{c}_0|_p$$

The feasible set associated with the constrained problem contains every unique solution.

*Proof.* For any  $\mathbf{x} = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k]$  in the original design space, there exists a unique reordering  $\mathbf{x}'$  in which the  $\mathbf{c}_i$  are sorted in ascending order by distance to a reference  $\mathbf{c}_0$  (assuming without loss of generality that there are no exact equalities). Since  $\mathbf{x}'$  satisfies the imposed constraints, it lies in the feasible set of the constrained problem. Since it maps to the same underlying Voronoi representation, it has the same objective value.  $\square$

With the application of these constraints, the genetic algorithm performs significantly better and exhibits typical convergence characteristics. An additional benefit of this formulation is that the constraints do not determine invalidity in any strict

<sup>2</sup>Implemented by the SciPy package.

sense. Therefore, if the algorithm explores infeasible points in the design space as part of its evolutionary strategy, they still represent perfectly valid (though potentially redundant) solutions to the problem.

## Initial conditions

An overly homogeneous population results in degenerate (or simply uninteresting) solutions, as many district placements produce the similar values of  $J_p$  and  $J_s$ . However, as a political tool, gerrymandering usually takes advantage of the real-world non-uniformity of preference. More realistic population and affiliation distributions are generated by Gaussian random fields, which are a sort of two-dimensional analog to Gaussian processes.<sup>3</sup> These fields can be shifted and normalized to represent “clumpy” probability distributions. As an optional intermediate step, they can be exponentiated to increase clumping. The population is generated by specifying weights  $[w_1 \dots w_m]$  for the occurrence of affiliations in the general population. Then, for each affiliation, a separate Gaussian random field distribution is generated and sampled the appropriate number of times. This creates an overall population in which different affiliations have different geographic distributions which exhibit large-scale structure. Figure 2 shows three Gaussian random fields generated with different power law exponents, interpreted as a scalar function and as a distribution sampled with  $n = 10^4$ .

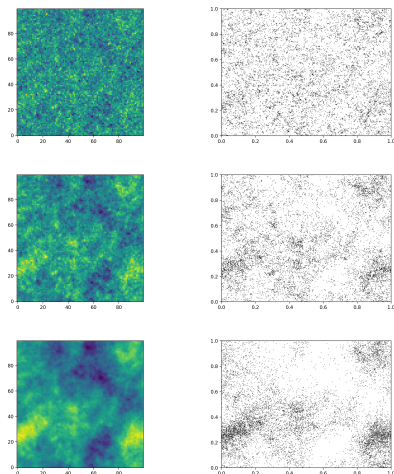


Figure 2:  $P_k = 1/k^\alpha$  for  $\alpha = 1, 2$ , and  $3$ , *descending*. Note the emergence of large-scale random structure.

<sup>3</sup>Gaussian random fields have use in physics to describe or generate stochastic systems with an autocorrelation function associated with a power spectrum. By parameterizing the spectrum with a power law, it is possible to generate random fields with a specified degree of large-scale structure.

## Robustness

When a solution is returned at the completion of the genetic algorithm, it is necessary to examine how well it withstands small population shifts. The value of  $J_s$  is more sensitive to changes in the design space but tends to vary by amounts on the order of  $1/n$ , while  $J_p$  changes much more infrequently but tends to vary by amounts on the order of  $1/k$ . As a result, optimization often effectively occurs over  $J_s$  while  $J_p$  remains constant. This leads to a tendency for an optimal point to lie near the boundary of a region of constant  $J_p$ . This sort of fine-tuning results in solutions that are extremely sensitive to small changes.

For a given solution, a robustness analysis is performed by evaluating the objective function over an ensemble of populations varied by a small amount of random noise. The distribution of costs characterizes the sensitivity of the solution. A small amount of variance is expected due to  $J_s$ , but a change in  $J_p$  would be less acceptable.

To mitigate the sensitivity, an optimization can be carried out over the expected value of the objective function under population uncertainty. At each design point, the objective function is averaged as described above. The estimation is noisy but significantly smoother than the original objective function, as many of the discontinuities are effectively softened. The solutions produced by this procedure can be considered more trustworthy, although they are worse performing. Unfortunately, this analysis is computationally expensive.

## Results

The optimization was run for a system with  $n = 10^3$ ,  $k = 10$ , and  $m = 2$ , which is sufficient to model many regions in which there are two dominant but unequal affiliations. The affiliation weight vector is  $\mathbf{w} = [0.7, 0.3]$  and the Gaussian random field distribution is close to homogeneous. The optimal solution is plotted in Figure 3.

The algorithm succeeds in removing all preference inequality. Of the ten districts, seven are majority  $a_0$  and three are majority  $a_1$ . Size discrepancy is also small, indicating that the solution satisfies all three criteria of the problem statement. However, the convergence plot (Figure 4) shows that a large portion of the optimization is spend in the tail-end. Since  $J_p$  is already optimized by this point (indicated by a cost  $\lesssim 1/k$ ), this suggests the destabilizing fine-tuning described above.

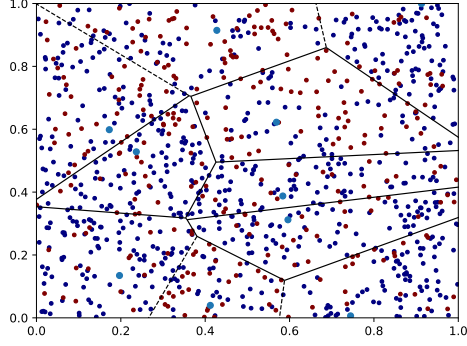


Figure 3: Optimal solution.  $J_p = 0$  and  $J_s = 0.046$ .

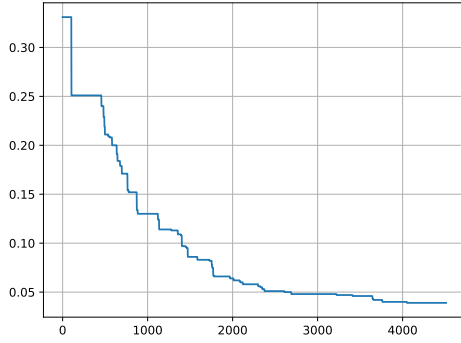


Figure 4: Cost versus iteration of genetic algorithm.

The robustness analysis (Figure 5) confirms that this occurred. The large gaps in the histogram indicate that jumps in  $J_p$  occur with small changes in the population positions – a highly sensitive solution.

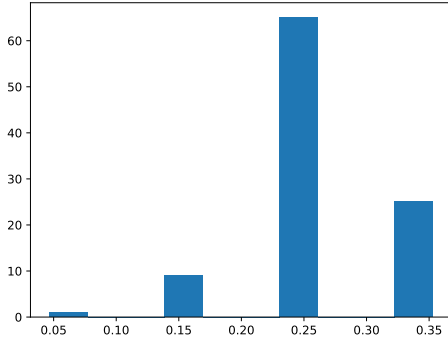


Figure 5: Distribution of cost under uncertainty for optimal solution.

By comparison, the optimization is also performed over the expected value of the objective function under population change, calculated with a 10-fold average. The resulting solution is shown in Figure 6.

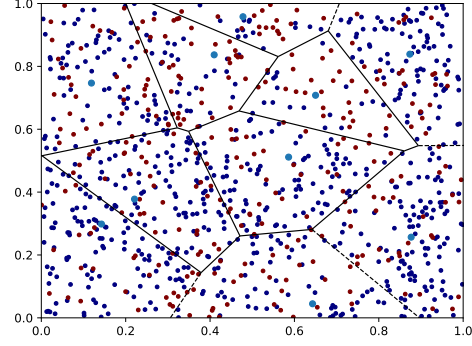


Figure 6: Optimal solution for expected value.  $J_p = 0$  and  $J_s = 0.119$ .

As before, preference inequality is zero, but size inequality is significantly higher than for the non-averaged solution, since extreme fine-tuning cannot take place. The benefit is made apparent by the robustness analysis (Figure 7), which shows that the solution is far more stable.

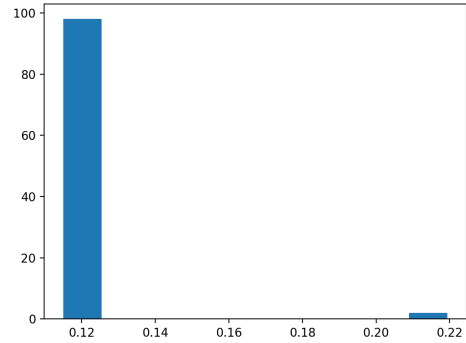


Figure 7: Distribution of cost under uncertainty for expected value optimal solution.

## Discussion

This paper develops an extremely simplistic model of district-based voting, which lends itself to a reasonably tractable optimization problem. Differential evolution proves to be a viable strategy for this optimization. Regardless, the problem's nonlinearities lead to slow and somewhat unreliable convergence characteristics, although this may be unavoidable.

The inherent symmetry of the problem can be removed by an appropriate set of constraints, which are necessary for the algorithm to converge. Experimentation suggests that the  $L_1$  norm allows for significantly better convergence than higher norms – this is likely because the constraints become affine. However, it is not clear what effect the constraints

have on the algorithm’s ability to locate a global or near-global optimum.

The algorithm’s tendency to return high-sensitivity optima can be mitigated by optimizing over an averaged objective function, although the computational cost is significant. A potentially better approach, which is not explored in this paper, would be to modify  $J_s$  to clip to zero below a certain threshold. This would avoid the lengthy terminal phase of optimization during which solutions can gravitate towards sensitive regions.

Future work would involve restructuring the project back-end to take full advantage of multiprocessing. Although the differential evolution algorithm employs multiprocessing in order to speed up evaluations of populations, the individual evaluations contain a significant amount of unexploited parallel structure. An additional extension would also examine meta-optimization over  $k$ , the number of districts. It is likely that certain values of  $k$  are better suited to certain populations. Over-fitting would be discouraged by a suitable penalty.

## Conclusion

Gerrymandering has existed in the United States for hundreds of years – its non-illegality makes it a powerful instrument of deliberate voter suppression. Although it is possible to find “intentionally fair” district boundaries that are optimal with respect to certain parameters, as this paper attempts to do, it is difficult to guarantee stability under long-term population change. The question of if and how a representative democracy can genuinely reflect the interests of the general population remains unsettled.

## Acknowledgments

The author would like to thank Mykel Kochenderfer and the AA222 course staff.

## Code

The code for this project can be accessed on GitHub.<sup>4</sup>

---

<sup>4</sup><https://github.com/rlipkis/gerrymandering>