Proper Utilization of the U and L Constraint in Labeling Model

Ryne Garrison

May 25, 2021

1 Introduction

For my research paper, Dual Objective Optimization in Political Redistricting, I have been investigating the trade-offs necessary to optimize either population deviation or compactness further. As with many dual objective problems, we are interested in finding the Pareto optimal solution– that is, the set of solutions that are not dominated by another feasible solution.

If we are concerned with minimizing two functions f_1 and f_2, a "non-dominated solution" is one in which further minimization cannot be made in either function without the other function becoming less further from optimal. In our case, we can represent f_1 and f_2 as the minimization of population deviation, and the minimization of cut edges, respectfully.

There are many differing methods to investigate multi-objective optimization, and many ways in Gurobi to implement these methods. The weighted sum, method, for example, is used by translating a set of objectives to a singular, scalar objective, and by multiplying each objective by a user-determined weight. Weights can be chosen in proportion to a relative importance of the object.

However, this method makes it difficult to set weight vectors to obtain Pareto-Optimal solutions in a desired region of objective space. Because of this, we will investigate a better potential method for our dual-objective optimization problem.

1.1 ϵ-Constraint Method

An alternative method to the Weighted Sum Method is the ϵ-Constraint method. This method utilizes just one f_1 or f_2 as an objective, and turns the other function into a constraint with user defined values, which can be reduced or increased to measure changes.

The ϵ constraint makes inherent sense for what we are trying to accomplish with our dual-objective optimization. Since redistricting literature constantly affirms that 1.00% is a hard cut-off in the eyes of the courts for compliance with the "one-person, one-vote" principle, we can define how we set our first constraint for our model.

How we define population deviation is of the utmost importance in setting our first ϵ constraint. Since we will base all subsequent ϵ values off of the results from the experiment before it, we need a strong definition that does not discriminate in erroneous ways.

The rest of this document defines how population deviation will be defined. Section 2 explains the Labeling Model, which is used for our experiments. Section 3 explains an inherent disadvantage in how one of the constraints within this model is used, and Section 4 proposes an alternative to this disadvantage that will allow for better results.

2 Overview of Labeling Model

When reviewing former literature, it was determined that the Labeling model for Redistricting would make the most sense for my experiments. The Labeling model utilizes variables $x_{ij} = 1$ if vertex $i \in V$ is assigned to district $j \in [k]$. Here, k is the number of districts assigned to a state, and V is the set of vertices 1, 2, ..., n representing a different county or tract. Note that n is the number of counties or tracts in the state. For our experiments, most observations will be made on the county level.

$$\min \sum_{e \in E} y_e \tag{1a}$$

$$x_{uj} - x_{vj} \leq y_e \qquad \forall e = \{u, v\} \in E,\ \forall j \in [k] \tag{1b}$$

$$\sum_{j \in [k]} x_{ij} = 1 \qquad \forall i \in V \tag{1c}$$

$$L \leq \sum_{i \in V} p_i x_{ij} \leq U \qquad \forall j \in [k] \tag{1d}$$

$$x \in \{0,1\}^{n \times k},\ y \in \{0,1\}^m. \tag{1e}$$

In this model, the constraint (1d) is tasked with ensuring that the population of each is between two constraints, U, representing the maximum population that a district will be accepted with in this model, and L, representing the minimum population that a district will be accepted with in this model.

2.1 The U and L Constraint

The constraint (1d), mentioned to be a population deviation constraint, ensures that each district is in a certain range. In most models and literature that I have seen, this is usually done by picking a specific value to set U to, and a specific value dependent on that to set L to, to ensure that the population deviation remains under a certain maximum.

Often, to ensure that the deviation fits the original 1.00% model the following calculations are used for U and L:

$$U = (1 + \frac{deviation}{2}) * \frac{\sum(population)}{k} \tag{2a}$$

$$L = (1 - \frac{deviation}{2}) * \frac{\sum(population)}{k} \tag{2b}$$

For a deviation of 1.00%, this creates a range where the maximum possible district size is 1.005 of the ideal district size, and the minimum possible district size is 0.995 of the ideal district size. This works in many optimization problems, but is not the best definition of population deviation.

3 The Problem With the U and L Constraint

Take, for example, a graph with 3000 points that must be partitioned into three regions. Ideally, then, the best proposed solution would be three regions that each have 1000 points. By our L and U constraints as defined in the previous section, we would tolerate a plan in which the minimum district has up to 1005 people, the smallest district has 995 people, and the third district has 1000. The population deviation between the largest and smallest districts would be 10 people, which is 1% of 1000. This then, would be a feasible model.

But, take a solution in which the largest district has 1006 people, the smallest 996, and the third, 998. This solution also has a deviation of 10 people between the largest and smallest district, but has a district that violates the constraint we set on the maximum size, U. But what if this alternative solution returns a result that is more compact, or contiguous, or achieves some other goal on a better scale than our previous example?

Such is the case for an experiment ran on Iowa:

Figure 1: Results for Iowa with a U set to 1.005 and L set to 0.995, compared with results for IA with different restrictions.

This experiment was run in both cases with the maximum allowed deviation to be set to 0.01. That is, a difference of a maximum of 7615 people between the largest and smallest district. The plan on the left, found by the old constraints, has four districts of the following populations: 758443, 760096, 763849, and 763967. These all fit in the defined L and U range, [757781, 765396]. This solution, optimized for minimal cut edges returns an objective value of 33, with a population deviation of 5524.

The example on the right has four districts of the following populations: 759188, 759573, 761697, and 765897. Note, the largest district does *not* fall into the range for L and U. In fact, it exceeds the U value by 501 people. However, the population deviation is only 6709 people between the largest and smallest districts, which is inside the maximum deviation constraint we set. This solution, interestingly, further optimizes the cut-edges objective, yielding a result of only 32 cuts edges.

But how do we set constraints to allow a plan to take *any* possible configuration that falls under the population deviation constraint without cutting off values above our initial constraints that could yield better results?

4 Proposed Alternative

For the proposed alternative, we create variables, P_{ij}, that measure the difference in population between two districts i and j. Currently, a constraint is placed that does not allow these variables to be above the number of people in our maximum deviation plan, which in the previous example would be 7615. For the next iteration of our experiment, we can find the true deviation from the results of the last experiment, and set a new constraint that the results this time must be less than or equal to that number of people *minus* one.

There is the potential for fixing here as well. When i = j, the varible P_{ij} will always be 0, since the difference in the population of a district and itself. Also, similarly to D-fixing for x_{ij} variables, there may be a way to cut the number of remaining variables in half. Since the difference in populations of districts i and j is the negative of the differences in the populations j and i, we can fix half of the variables to 0, so long as we ensure that we are taking the absolute values of the remaining variables.

The time complexity additions of this method are unknown, as are the potential downfalls. There is currently a bug in my code that allows Iowa to run three iterations, each with a smaller population deviation and slowly increasing objective value, before the third iteration of the model (population deviation = 3500, cut edges = 33). This is under further investigation, but has not been an easy fix.