Optimizing the Compactness of Political Districts in Oklahoma

Elizabeth Bunting

Oklahoma State University

1. Abstract

This work explores a mathematical optimization model to determine the optimal political districts for a state. Originally created in 1965 by Hess et al, the model assigns each population unit (such as a county) in the state to a population unit that is designated by the model to be the district center. In this way, it determines optimally compact political districts. Compactness is based on two constraints: the distance between a population unit and the population unit that is its potential district center, and the population of each county.

This work applies the Hess model to Oklahoma, using Python and Gurobi to implement the model. ArcMap was used to visualize the results of this implementation. In addition to applying the Hess model to Oklahoma, this work explores the effect of population bound variations in each district. Political districts should be as equal as possible, especially congressional districts (the focus of this research). However, this model does not allow exact equality. Therefore, experimentation was required to determine the level of population equality that also allowed the districts to be connected (another requirement of political districts). In addition, visual compactness is compared with population-weighted compactness. Because the population is a variable included in the objective function, it has a significant effect on the results of the model. If population is removed from the objective function, the district compactness is defined entirely by distance.

Through my research, I created a Python program that calculated optimal political district centers for Oklahoma and assigned each county in Oklahoma to its ideal center based on distance between and population of the counties. I also created a map in ArcGIS to visualize the districts created from this program. In addition, I experimented with the parameters and constraints included in the program to determine their effect on the districting solution.

1. Introduction

Political districting is the process of dividing an area (such as a state) into a certain number of districts, which are used for election purposes. The creation of political districts' boundaries is an important process with significant and numerous ramifications. Historically, political district lines have often been drawn with a certain goal in mind. They have often been used to shift power to a chosen demographic – for example, boundary lines can be manipulated to split certain minority groups across several districts, diluting the minorities' votes and preventing them from having a majority in any one district. This practice is known as gerrymandering. Last year, the Supreme Court ruled that the state of Pennsylvania would have to redraw its political district map to rectify the results of partisan gerrymandering. There have been similar cases in North Carolina, Wisconsin, and Maryland. (Leonhardt, 2018) Currently, the process of redistricting – redrawing political district lines after a census – is conducted every ten years. The process is completed in various ways; they are often chosen by the state legislature and are traditionally controlled by humans. (Levitt, 2019) In Oklahoma, those tasked with drawing district lines are officials who are elected by the districts they draw. (Denwalt, 2018) There is significant room for improvement in this process to allow for a fairer and more impartial districting system.

The focus of this research is a mathematical optimization model, created by Hess et al. in 1965, which aims to optimize the compactness of political districts in a state. For the purposes of this research, the ideal district is defined as one that is optimally compact. compactness is measured by the distance from territorial units included in the district to the district center, squared, multiplied by the population of a territorial unit. In addition, the population of a district must be within the upper and lower bounds defined in the model. The objective function, a mathematical representation of compactness, is composed of the moment of inertia function (the measure of compactness described above) and a binary variable for district assignment. In order to find these district centers and the counties that were assigned to them, I implemented the Hess districting model using Python and Gurobi. This model imposes certain constraints on the districts, such as the upper and lower population bounds allowed in a district and the number of districts in the state. After formulating this mathematical model in Python, I used ArcGIS to visualize the results of the districting program.

2. Literature Review

Operations research was first developed during World War II. Since its inception, the science of optimization has evolved dramatically. Initially, it was simply the application of scientific thought and method to military operational issues. Today, it has evolved to include the application of mathematical models to describe a wide variety of situations and problems across industries. It can be used to optimize problems such as the placement of ambulances, creating better schedules for workers, and the optimal location of warehouses. There are several optimization problems that focus on solving the districting problem.

3.1. Exact Optimization Models for Districting

There are several models to obtain exact solutions to the districting problem. Garfinkel et al. (1970) is viewed as foundational in achieving an exact solution. This paper proposes an algorithm that consists of two phases. These phases first apply the constraints of contiguity, population equality, and compactness. Next, they partition the model into feasible districts.

Mehrotra et al. (1998) and Nygreen (1988) built on the work of Garfinkel et al. Nygreen used a similar approach but used a graph-theoretic model. This model guaranteed connectivity, and imposed compactness and conformed to administrative requirements. Mehrotra et al. was very similar to Garfinkel et al., but used a different objective function. Both the Nygreen and Mehrotra et al. methods have been applied to small problems and can provide exact solutions for these small instances.

3.2. Heuristic Methods for Districting

The multi-kernel growth approach to districting was one of the first to be suggested. it was first presented in Vickrey (1961), and was also explored by Bodin (1973) and Arcese et al. (1992). This is an incremental approach which first selects district centers, and then assigns units to them one at a time, starting with those in closest geographical proximity to the center. This process continues until a population limit is attained. Vickrey considered contiguity, population equality, compactness, and conformity to administrative boundaries. Bodin and Arcese et al. both use various forms of a contiguity graph, which represents the geographical area being studied as a graph with nodes as the territory units and an edge between neighboring nodes. (Ricca et al) Bodin uses a directed graph, and Arcese et al. considers population weights associated with the nodes.

Another approach to the districting problem is an approach through mathematical formulation focused on location. This approach adapts the warehouse location technique to assign territorial units to districts. The Hess model (discussed in section 3.2) is an example of this kind of model. Common constraints are distance and population. Hojati (1996) and George et al. (1997) use approaches similar to the Hess model, but Hojati uses a mixed integer model; George et al. uses a different process to assign territorial units to districts.

3.3. Hess et al. (1965)

In 1965, S. W. Hess and J. B. Weaver published a paper entitled "Nonpartisan Political Redistricting by Computer". In their paper, they described how operations research could be used to optimally draw political district lines. The Hess model is based on the idea that the requirements for political districts (approximately equal population and compact shape, and contiguity) can be enforced through the use of an objective function and constraints. In the Hess model, there are n territorial units and k districts. The problem is similar to a warehouse location problem. Each unit i is assigned to a district k, using a binary variable. This binary variable, x_{ij}, relates a territorial unit i to another territorial unit j. If j is defined as a district center and i is assigned to j, then xij is equal to 1. Otherwise, x_{ij} is equal to 0. Another binary variable, x_{jj}, is used to define district centers. If unit j is a district center, then x_{jj} is equal to 1; otherwise, it is equal to 0. a and b are percentages of population difference for imposing population bounds.

$$min \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij}^2 \, p_i x_{ij}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad i = 1, \dots, n$$

$$\sum_{j=1}^{n} x_{jj} = k$$

$$a\bar{P}x_{jj} \leq \sum_{i=1}^{n} p_i \, x_{ij} \leq b\bar{P}x_{jj} \qquad j = 1, \dots, n$$

$$x_{ij} \in \{0, 1\} \qquad i, j = 1, \dots, n$$

Hess et al Model, 1965; with additions from Ricca, Scozzari, and Simeone, 2013.

The first line of the model is the objective function, which relies on the summation of the product of the squared distance between unit i and unit j (d_{ij}^2), and the population of a unit, p_i, for all territorial units i and j. The second line of the model requires each unit to be assigned to one district center. The third line of the model requires each district center (a unit assigned to itself) to be equal to the fixed number of districts. The fourth line of the model imposes the population bounds on the district. Finally, the fifth line, an addition by Ricca, Scozzari, and Simeone (2013) requires that x_{ij} is either equal to 0 or 1.

3. Data Preparation

The data needed for this project was the distances between the centers of counties in Oklahoma, and the population of Oklahoma's counties. I also needed an Excel file that contained the numbers of each Oklahoma county (0 through 76) that were used in my code, and showed the corresponding county code (a five-digit number unique to each county) for each number. This information was necessary to identify the results of my code on a map and identify which district the counties belonged to. All of this data was provided by Eugene Lykhovvd, which was obtained from 2010 US Census data. The distance data was organized in Excel as a matrix, showing the distance between each county. The population data was a list in Excel which contained the number for each district and its corresponding population. My code reads these Excel files in order to apply the Hess districting model to Oklahoma. After the code has found optimal district assignments, it writes the data into a new excel file, entitled "Hess Solution.xls". This file can be used to view an optimal solution for district assignments.

4. Experimentation

5.1. Computational Model

Using Python and Gurobi, I defined the objective function, variables, and constraints of the Hess districting model. I used population and distance data for Oklahoma's counties, which I defined as the population units (further discussed in the Section 4, Data Preparation).

I programmed this model using a function called build_and_solve_hess_model (the full code is included in Appendix A). This function defines the parameters used to solve the districting model as distance_filename, population_filename, upper_bound, lower_bound, and number_of_districts. The distance_filename and population_filename parameters import distance and population data, respectively, from excel files (further discussed in Section 4, Data Preparation). These files can be changed in order to add versatility to the model. While I use the data for Oklahoma, if data from another state was used, the model could be adapted to work for another state.

These parameters were used to define the variables in the model:

- *distance*: the Euclidean distance between the centers of two counties i and j. Defined as a list filled with values from the distance excel file.
- *population:* the population of a county i. Defined as a list filled with values from the population excel file.
- *L:* the lower bound of the district population, defined by the user as the parameter lower_bound.
- *U:* the upper bound of the district population, defined by the user as the parameter upper_bound.
- k: The number of districts, defined by the user as the parameter number_of_districts. k value is determined based on the total population of a state, and is a fixed value. If this model was used for another state, the k value should be changed to reflect that state's number of political districts.
- *vertices*: the individual counties in Oklahoma. This value is taken from the population file, and is defined as the range of the length of the population list.

5.2. Population Bound Experimentation

Population equality between districts is an important factor in their creation, in order to ensure the one-person, one-vote rule is applied. It is a constitutional requirement that congressional districts should have upper and lower population bounds within 1% of each other. However, since the population units for this formulation are counties, there is a limit to the level of equality that is feasible due to the fixed populations of the counties. For this project, I used the ideal population suggested in Hess et al, defined as P = (Total population)/(Number of Districts). For this case, P = 3751351/5 = 750270.2. The P value is multiplied by a factor of a number slightly greater than 1 (such as 1.01) for the upper bound (U) and slightly less than 1 (such as 0.99) for the lower bound (L).

After running my program with the upper and lower population bounds for each district 1% apart, with a lower bound factor of 0.995 and an upper bound factor of 1.005, I manually created a map in ArcGIS, using the resulting district assignments from the program. In order to quickly observe the effect of variations in the upper and lower population bounds, I needed a program that would automatically create a map that was color coded based on district assignments (Yeager, 2019). Using this program, I observed the resulting changes to the district maps when the upper and lower bound factors change. I started with broader population bounds, allowing a difference of up to 10% between the upper bound and lower bound. I changed the interval between a and b values first within an interval that would result in population differences betwenn this larger interval was due to the assumption of indivisible counties. After running ten iterations, changing the bound factor by .01, the code had created nine district maps (the case in which both upper bound and lower bound factors are equal to 1 was not included because this case causes the model to be infeasible). As the population bounds change, the district assignments also change. I was looking for maps that had attributes of connectivity, another important requirement of political districts. Connectivity is not a constraint specifically imposed by the Hess model, it is just included as an incidental property of compactness; I wanted to choose upper and lower population bounds that would result in a model in which the districts were connected in a practical way. Figure 5.1 shows an example of one of these tests. There has been research conducted more recently than the Hess model that explores rigid constraints to impose contiguity. (Shirabe, 2004). These methods include heuristics (Brookes, 1997) and MIP (Mixed Integer Programming) approaches (Williams, 2002). However, implementation of these models is beyond the scope of this research.

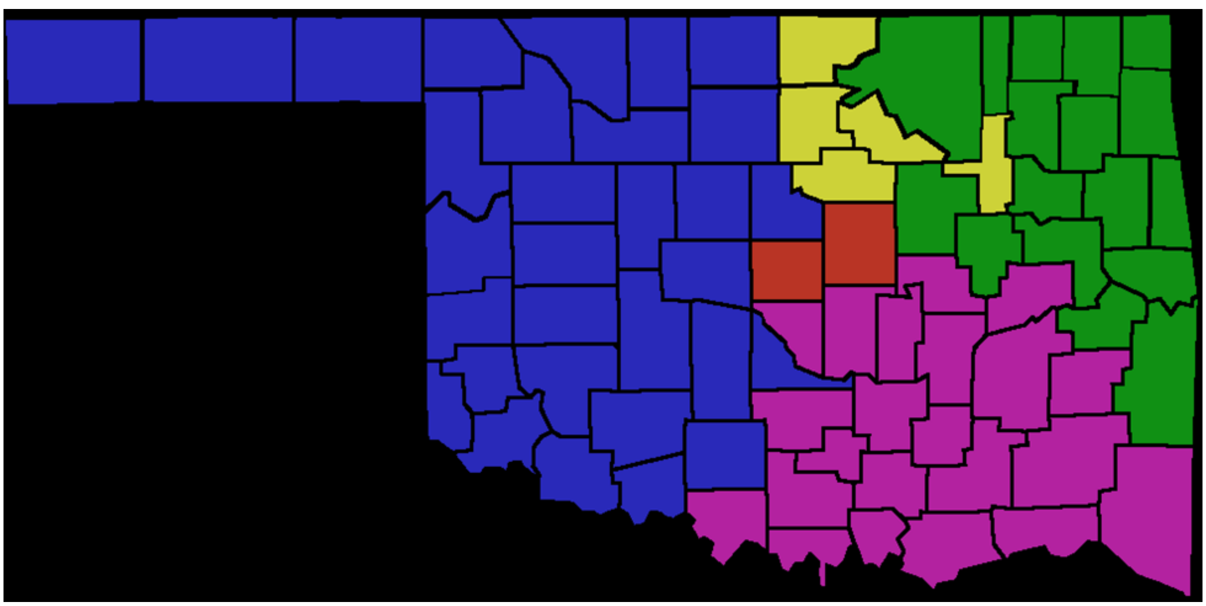

Figure 5.1: District assignments when L = 0.99(P) and U = 1.01(P).

I also ran experiments which fit the restrictions of a population difference of 1% or less between congressional districts. I ran 10 iterations of code such that the lower population bound (L) was *P* multiplied by a minimum lower bound factor of 0.995, and the upper population bound (U) was *P* multiplied by a maximum upper bound factor (U) of 1.005. These tests resulted in maps with district population differences ranging from 0.1% to 1%. These maps were consistently disconnected. I was unable to find population bounds within 1% of each other for which the congressional districts were connected. I identified two reasons for this: the lack of connectivity constraints in the Hess model, and the large population of counties. Because individual counties cannot be split between districts, but must be kept as one unit, there is a limit to the change that can occur in the population of a district from changing the assignment of one county. As the population bounds are enforced and tightened, it becomes more difficult to impose contiguity as an incidental property of compactness, which is the approach used in the Hess districting model (Hess et al, 1965).

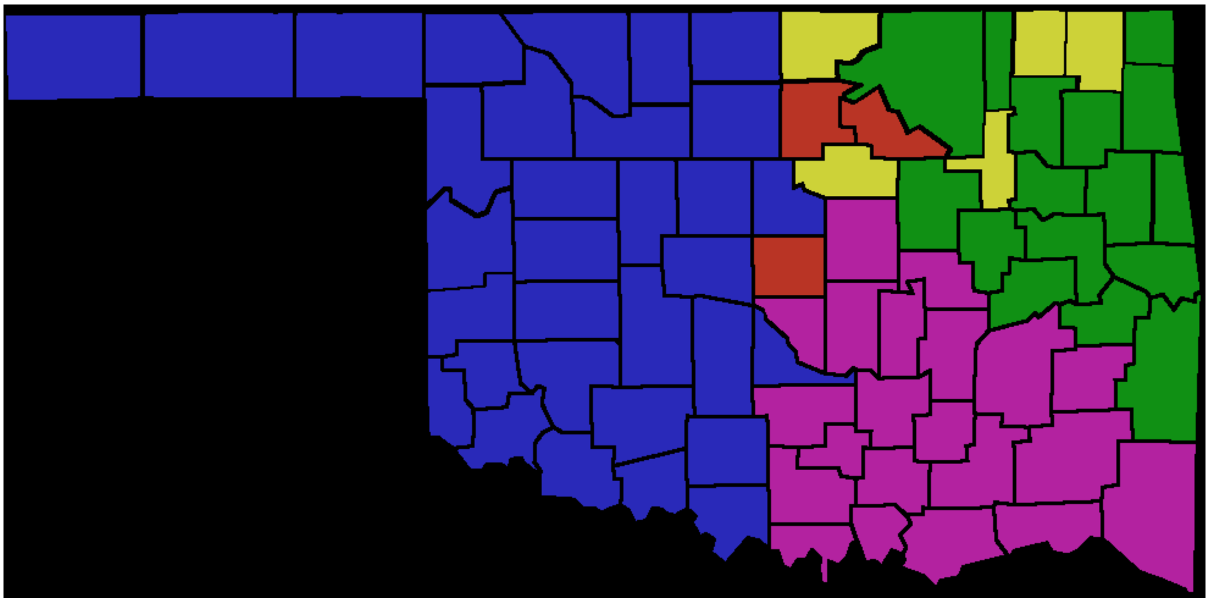

Figure 5.2: District assignments when L = 0.995(P) and U = 1.005(P).

5.3. Objective Function: Population-Weighted vs. Visual Compactness

The Hess districting model minimizes population-weighted compactness, meaning the squared distance in the objective function is multiplied by the population of the population units. As a result, the district maps created by this model are compact in respect to the moment of inertia defined by the objective function, but are not geographically or visually compact. I wanted to see the effect of removing population from the objective function, which would cause the model to simply minimize the distance between counties in a district without accounting for their population. I kept the other constraints the same, which meant that there were still k = 5 districts, and each district still had to follow the upper and lower population bounds defined in the model. I hypothesized that this method would result in districts that were geographically and visually more compact and were generally more connected.

I used the map-creating code (Yeager, 2019) to test my hypothesis. I used an upper bound factor of 1.005 and a lower bound factor of 0.995. A sample of the resulting maps are shown in figure 5.3. Because of the difference in the objective function, the district centers changed, as did the geographical size of the districts. However, I was incorrect in my assumption that the districts would be more connected geographically. There were consistently unconnected portions for each district. I ran the code for an upper bound factor range of 1.001 to 1.005, and a lower bound factor of 0.995 to 0.999, resulting in a range of tests where the difference in population ranged from 1% (lower bound factor = 0.995 and upper bound factor = 1.005) to 0.2% (lower bound factor = 0.999 and upper bound factor = 1.001). The maps stayed relatively constant over this range of population bound differences. The images generated by these tests are included in Appendix B. Most likely, the disconnected counties such as the yellow district county in the center of the state is a result of the population bound requirements. The disconnected counties are Oklahoma County and Tulsa County, both of which have significantly large populations. Previously, these counties were district centers because of their large populations. When population was removed from the objective function, it is not as critical that they be centers. These counties have been assigned to districts that contain counties with relatively small populations, which was most likely necessary for population balance. This is another case where smaller population units might be beneficial. If the counties could be split between districts instead of the entire county included in one district, it might be possible to create more connected district maps.

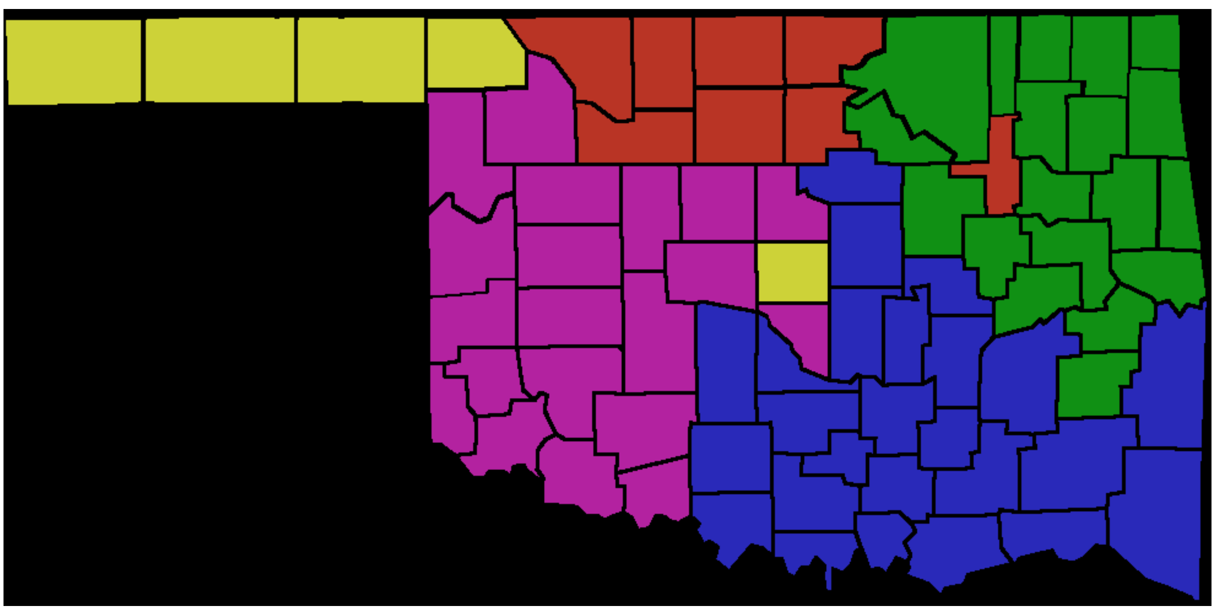

Figure 5.3: District assignments when population is removed from the objective function, L = 0.995(P), and U = 1.005(P).

5. District Visualization

Visualization was an important step in this research process, in order to see the districts that were created and ensure they were practical and connected. The current map of Oklahoma's congressional districts is shown in Figure 6.1. In this figure, there are some unusually shaped districts, such as the long shape of district 1, and the protrusions from district 4 into districts 5 and 3.

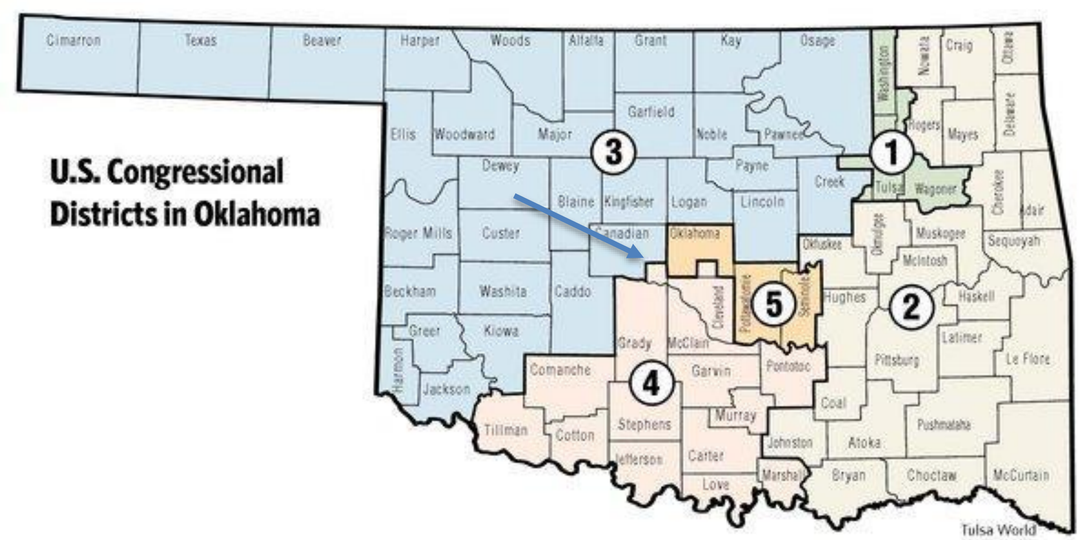

Figure 6.1: Oklahoma's Current Congressional Districts. The blue arrow identifies the protrusions between districts (tulsaworld.com)

In order to visualize the results from the mathematical model I coded from Hess et al (1965), I created a map of my initial results using ArcMap 10.6 by manually coloring Oklahoma's individual counties based on district assignments. I used this map to verify the effectiveness of the automatic map-creating code, and to begin to observe the relationship between compactness, contiguity, and the tightness of population bounds. This process is discussed in section 5.1.

6. Conclusion

Mathematical models can provide a valuable tool in redistricting. Using the Hess model implemented for Oklahoma, I was able to find optimally compact congressional districts, using counties as the population units. I found that, in order to have realistically approvable districts using this model, a smaller population unit such as cities or towns might be beneficial. Counties seem to have too large a population to incidentally impose connectivity as a result of compactness when the upper and lower population bounds are within 1%, a requirement for congressional districts.

I found that the inclusion of population in the objective function results in districts that vary significantly in respect to geography and visual inspection but allow for more connected districts. Removing population from the objective function results in more visually and geographically equal districts, but has counties that are disconnected from the district (such as Oklahoma County and Tulsa County) in order to satisfy population bound requirements.

In my project, I only included connectivity as an incidental property of compactness (Shirabe, 2004) and by changing the population bounds. Other models have explored and defined constraints that impose contiguity (Shirabe, 2004), but the implementation of those models is beyond the scope of this project. In addition, in order to obtain a more realistic district map, data could be obtained for population units that are smaller than counties, such as cities or towns; this could result in more variability allowed in the districts, which might eliminate the disconnected counties illustrated in Figure 5.3.

References

Arcese, F., Battista, M.G., Biasi, O., Lucertini, M., Simeone, B. (1992). Un Modello Multicriterio per la Distrettizzazione Elettorale: la Metodologia C.A.P.I.R.E. A.D.E.N. (unpublished manuscript)

Bodin, L.D. (1973). A District Experiment With a Clustering Algorithm. Ann N Y Accad Sci 219:209–214

Brookes, C. J. (1997). A Parameterized Region-Growing Programme for Site Allocation on Raster Suitability Maps. International Journal of Geographical Information Science 11, 375– 96.

Denwalt, Dale. (2018, January 4). Proposal Could End Oklahoma Gerrymandering. Retrieved February 5, 2018, from http://newsok.com/proposal-could-end-oklahoma-gerrymandering/article/5578176

Garfinkel, R.S., Nemhauser, G.L. (1970). Optimal Political Districting By Implicit Enumeration Techniques. Manag Sci 16:495–508

George, J.A., Lamar, B.W., Wallace, C.A. (1997). Political District Determination Using Large-Scale Network Optimization. Socioecon Plann Sci 31:11–28

Hess, S. W., Weaver, J.B., H.J., Whelan, J.N., & Zitlau, P.A. (1965). Nonpartisan political redistricting by computer. *Operations Research, 13*, 998-1006.

Hojati, M. (1996). Optimal Political Districting. Comput Oper Res 23:1147–1161

Ingraham, Christopher. (2015, March 1). This is the Best Explanation of Gerrymandering You Will Ever See. Retrieved February 3, 2018, from https://www.washingtonpost.com/news/wonk/wp/2015/03/01/this-is-the-bestexplanation-of-gerrymandering-you-will-ever-see/?utm_term=.65cb68fcd49c

Krehbiel. R. (2011, Feb 18). Oklahoma congressional districts must all change, some. Retrieved from https://www.tulsaworld.com/news/local/government-and-politics/oklahoma-congressional-districts-must-all-change-some/article_4a958991-245a-5b8c-8c13-53d843317f30.html

Leonhardt, David. (2018, February 6). A Blow Against Gerrymandering. Retrieved February 10, 2018, from https://www.nytimes.com/2018/02/06/opinion/gerrymandering-pennsylvania.html

Lykhovyd, E. (2019). Index of files. Retrieved from https://lykhovyd.com/files/public/districting/OK/counties/graph/

Mehrotra, A., Johnson, E.L., Nemhauser, G.L. (1998). An Optimization Based Heuristic for Political Districting. Manag Sci 44:1100–1114

Nygreen, B. (1988). European Assembly Constituencies for Wales. Comparing of Methods for Solving a Political Districting Problem. Math Program 42:159–169

Ricca, F., Scozzari, A., Simeone, B. (2011). Political Districting: from classical models to recent approaches. *4OR, 9*, 223-254.

Shirabe, T. (2005), A Model of Contiguity for Spatial Unit Allocation. Geographical Analysis, 37: 2-16. doi:10.1111/j.1538-4632.2005.00605.x

Vickrey, W. (1961). On the Prevention of Gerrymandering. Polit Sci Quart 76:105–110

Williams, J. C. (2003). Convex Land Acquisition with Zero-One Programming. Environment and Planning B 30, 255– 70.DOI: 10.1068/b12925

Yeager, J (2019). Python source code (Version 1.0) [source code].

Appendix A

```
"""
Created on Thu Jun 14 14:16:49 2018
Last edit March 13, 2019

@author: eliza

The purpose of this code is to determine the optimal counties to be political district
centers in Oklahoma,
```

```
And assigns each county to its optimal district center based on population and distance, resulting
in optimally compact political districts for Oklahoma
Using an objective function with population of counties and distance between counties as inputs.
Require inputs: excel file containing distances between Oklahoma counties, excel file containing the populations of
Oklahoma counties, upper and lower population bounds for political districts, and the number of political districts
in Oklahoma (K), which is 5.
"""

from gurobipy import Model, LinExpr, GRB

import xlrd, xlwt

def build_and_solve_hess_model(distance_filename, population_filename, L, U, K):
    """

    :param distance_filename: (list) the distance between each county
    :param population_filename: (list) the population of each county
    :param L: (float) lower bound of counties allowed in a district
    :param U: (float) upper bound of counties allowed in a district
    :param K: (int) number of districts
    :return: (tuple) returns model and decision variable
    """
    #Import distance between districts from excel file (originally from Eugene Lykhovyd, www.lykhovyd.com)
    distance = read_distance_data(distance_filename) #a list for distance matrix

    #Import the population of Oklahoma counties from excel file (originally from Eugene Lykhovyd, www.lykhovyd.com)
    population = read_population_data(population_filename)

    m, Z = build_model(L, U, K, distance, population, model_name='District Model')

    #Optimize
    m.optimize()

    #Results: which districts are assigned to each district center?
    HessSolution = []
```

```
    if m.status == GRB.OPTIMAL:
        for i in range(len(population)):
            for j in range(len(population)):
                if Z[i,j].x > 0.5:
                    HessSolution.append((i, j))
        return HessSolution
    else:
        return "Model is infeasible"

def HessSolution_Excel(district_assignments, file_name='Hess Solution'):

    book = xlwt.Workbook()
    sheet1 = book.add_sheet('Hess Solutions')
    sheet1.write(0, 0, 'County')
    sheet1.write(0, 1, 'District')
    for row, tup in enumerate(district_assignments):
        for col, val in enumerate(tup):
            sheet1.write(row + 1, col, val)
    book.save(file_name + '.xls')

def read_distance_data(filename):
    """

    :param filename: (string) name of Excel file to open
    :return: (list) Excel sheet 1 as a 2D matrix (or 1D if applicable)
    """
    wb = xlrd.open_workbook(filename) #open workbook
    sheet = wb.sheet_by_index(0) #indexing of sheet
    rows = sheet.nrows #n rows in sheet
    columns = sheet.ncols #n cols in sheet
    result = [] #a list for distance matrix
    for i in range(1, rows): #i is the row number
        tmpList = [] #temporary list
        for j in range(1,columns): #j is column number
            val = sheet.cell_value(i, j)
            if val != '':
                tmpList.append(val) #add the value of cell (i,j) to tmpList
        if len(tmpList) > 1:
            result.append(tmpList) #add tmpList to distance list
        else:
            result.append(tmpList[0])
```

```
        tmpList.clear #clear tmpList
    return result


def read_population_data(filename):
    """

    :param filename: (string) name of Excel file to open
    :return: (list) Excel sheet 1 as a 2D matrix (or 1D if applicable)
    """
    wb = xlrd.open_workbook(filename) #open workbook
    sheet = wb.sheet_by_index(0) #indexing of sheet
    rows = sheet.nrows #n rows in sheet
    result = [] #a list for distance matrix
    for i in range(1, rows): #i is the row number
        result.append(sheet.cell_value(i, 1))
    return result


def build_model(L, U, K, distance, population, model_name='district assignments'):
    """

    :param L: (float) lower bound of counties allowed in a district
    :param U: (float) upper bound of counties allowed in a district
    :param K: (int) number of districts
    :param distance: (list) the distance between each county
    :param population: (list) the population of each county
    :param model_name: (string) optional name for model
    :return: (tuple) returns model and decision variable
    """

    #The vertices are individual counties (population units)
    vertices = range(len(population))

    # Model
    m = Model(model_name)

    #Decision variable to decide if population unit i is assigned to region j
    Z = m.addVars(vertices,
              vertices,
              vtype=GRB.BINARY,
```

```
            obj = distance,
            name = "assign")

#The objective is to minimize the distance between all
#population units and their district centers
#adding objective function

expr = LinExpr()

for i in vertices:
   for j in vertices:
      expr += distance[i][j]**2*population[i]*Z[i,j]

m.setObjective(expr, GRB.MINIMIZE)

#Constraints

#Each population unit is assigned to 1 district
#(2) for all i, the sum of Zij over j = 1.

#Zij = 1 when unit i is assigned to unit j

for i in vertices:
   expr = LinExpr()
   for j in vertices:
      expr += Z[i,j]
   m.addConstr(expr == 1)


#(3) sum of Zjj = k
#The required number of political districts is k
#equal to the sum of each district assigned to itself

expr = LinExpr()
for j in vertices:
   expr += Z[j,j]

m.addConstr((expr == K), "3")

#(4) Population bounds: minimum and maximum population allowed in each district
```

```
    for j in vertices:
        expr1 = LinExpr()
        for i in vertices:
            expr1 += population[i] * Z[i,j]
        m.addConstr(expr1 <= U * Z[j,j] )

    for j in vertices:
        expr1 = LinExpr()
        for i in vertices:
            expr1 += population[i] * Z[i,j]
        m.addConstr(expr1 >= L * Z[j,j])

    #(5) The number of assigned units is less than the
    #number of districts

    for i in vertices:
        for j in vertices:
            if (i != j):
                m.addConstr(Z[i,j] <= Z[j,j])

    return m, Z

def print_to_string(district_assignment):
    for tup in district_assignment:
        print('assign %g to %s' % tup)


if __name__ == '__main__':
    HessSolution = build_and_solve_hess_model("C:\\Users\\eliza\\Desktop\\Wentz 
2018-2019\\OK_distances_01172019.xlsx",
                    "C:\\Users\\eliza\\Desktop\\Wentz 2018-2019\\OK-Population-
Counties.xls",
                    746519, 754022, 5)

    HessSolution_Excel(HessSolution)

    print_to_string(HessSolution)
```

Appendix B: District Maps with Population in Objective Function

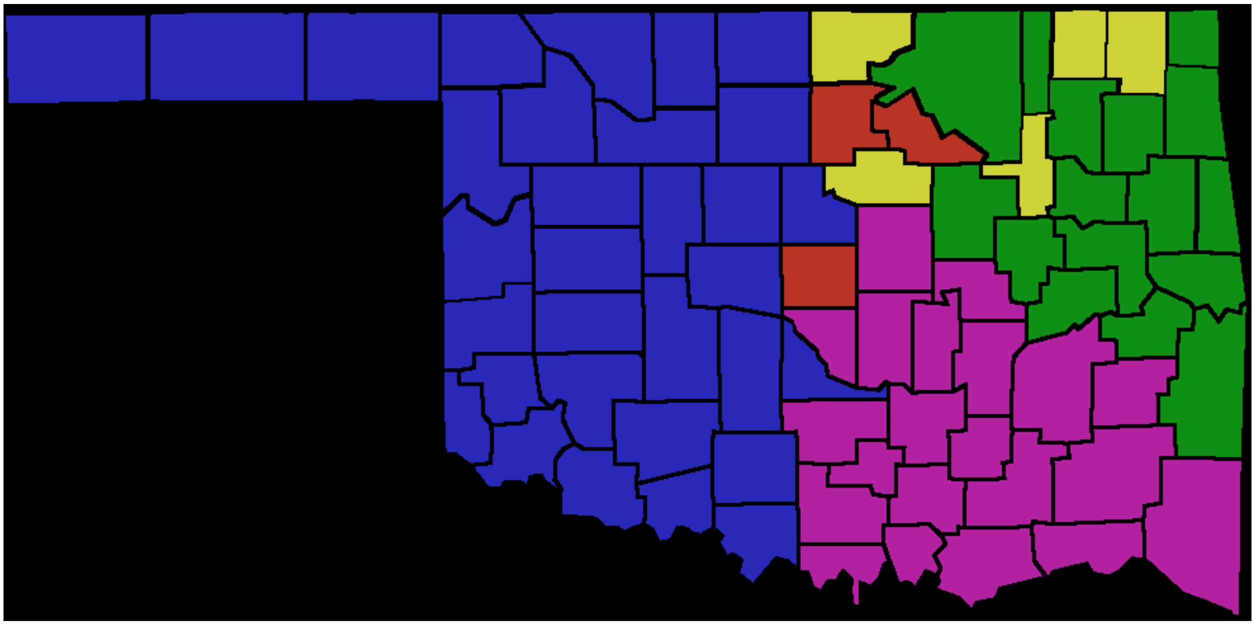

Figure B.1: District assignments with L = 0.995(P) and U = 1.005(P).

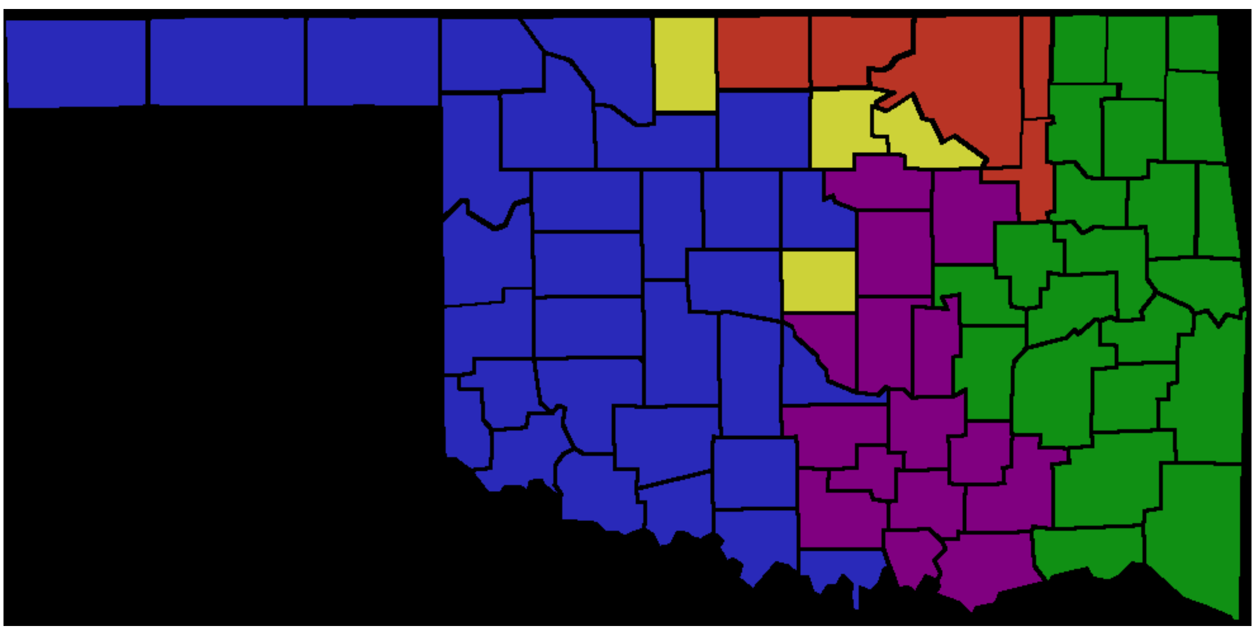

Figure B.2: District assignments when L = 0.996(P) and U = 1.005(P).

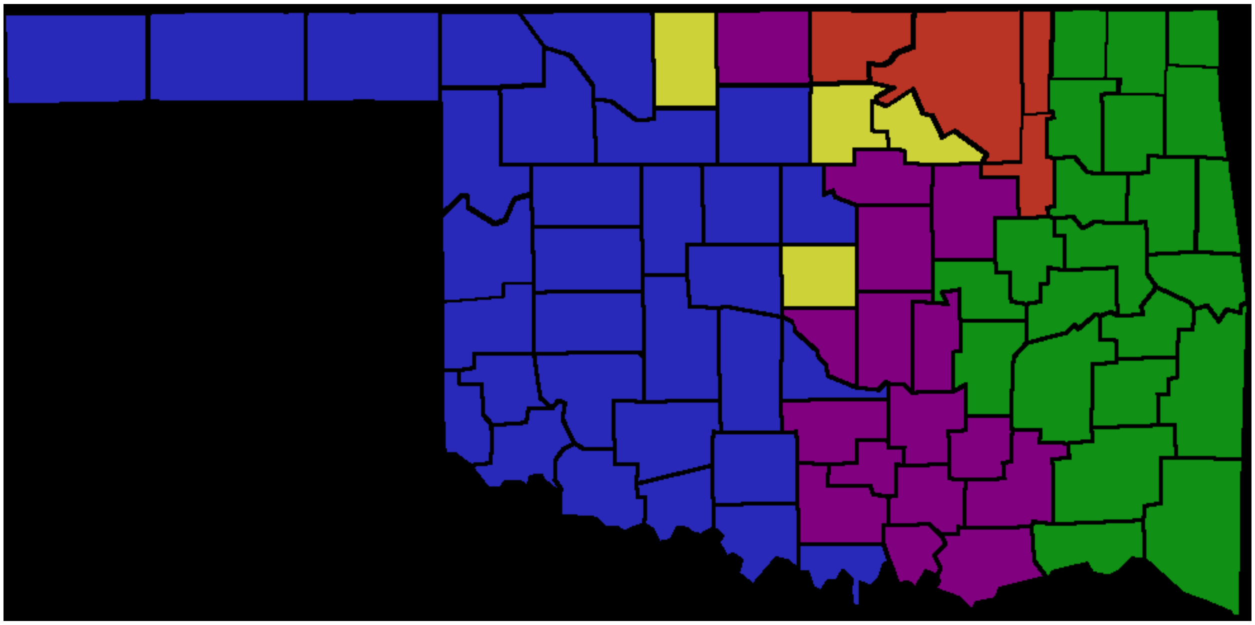

Figure B.3: District assignments when L = 0.997(P) and U = 1.005(P).

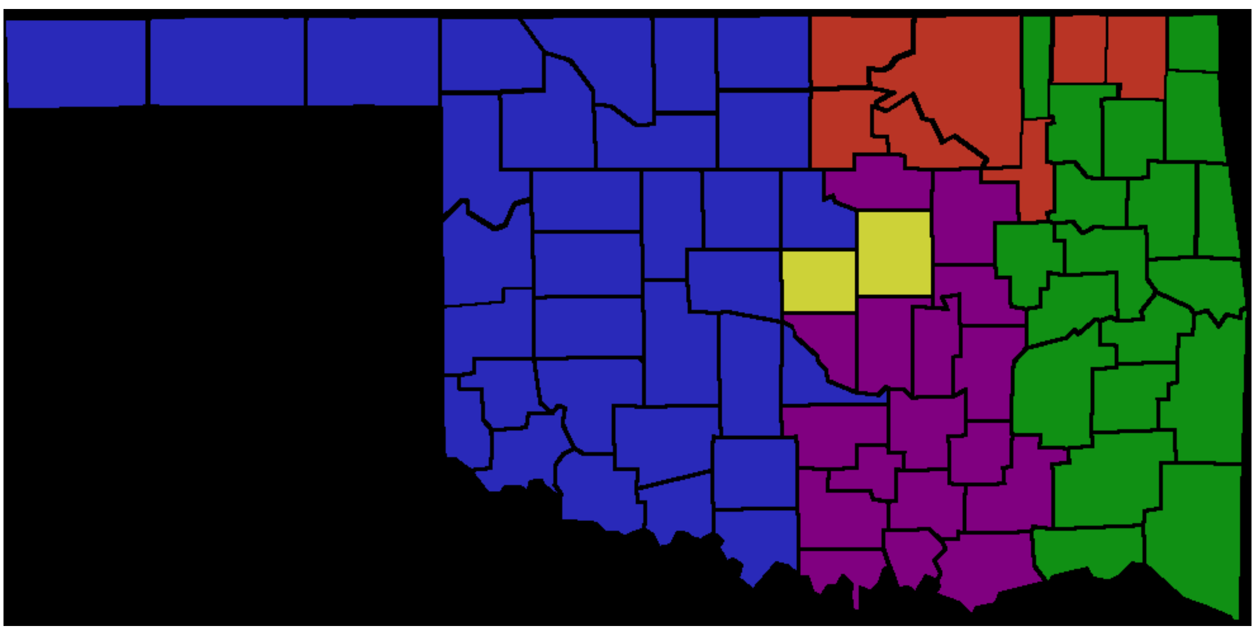

Figure B.4: District assignments when L = 0.998(P) and U = 1.005(P).

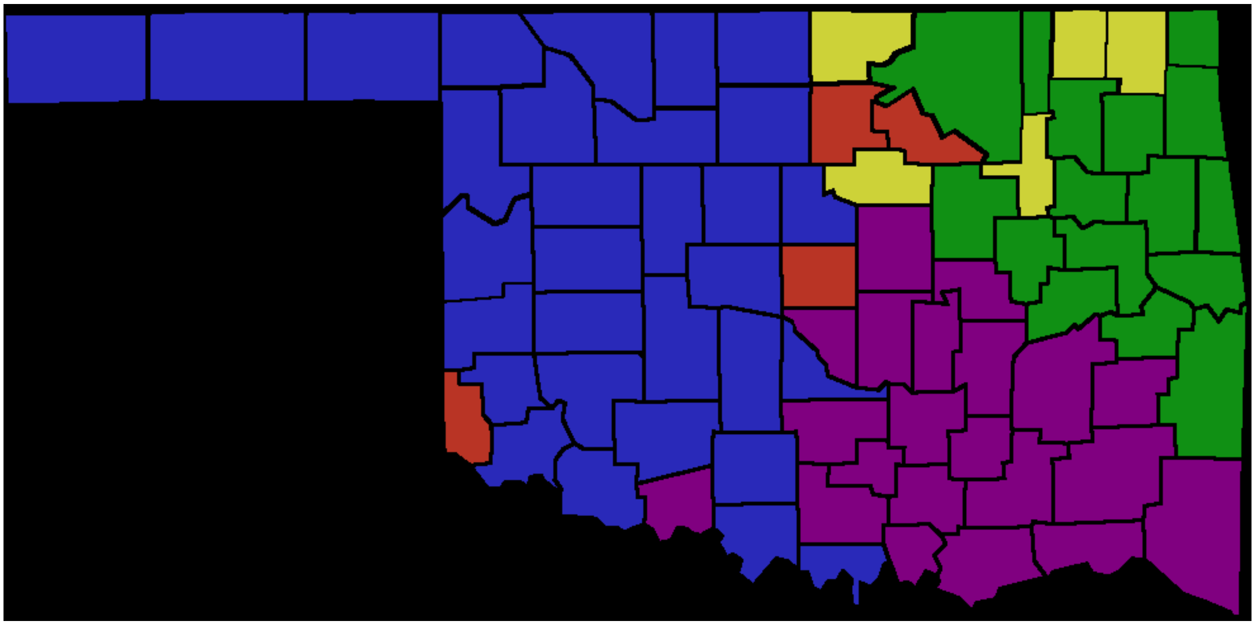

Figure B.5: District assignments when U = 0.999(P) and U = 1.005(P).

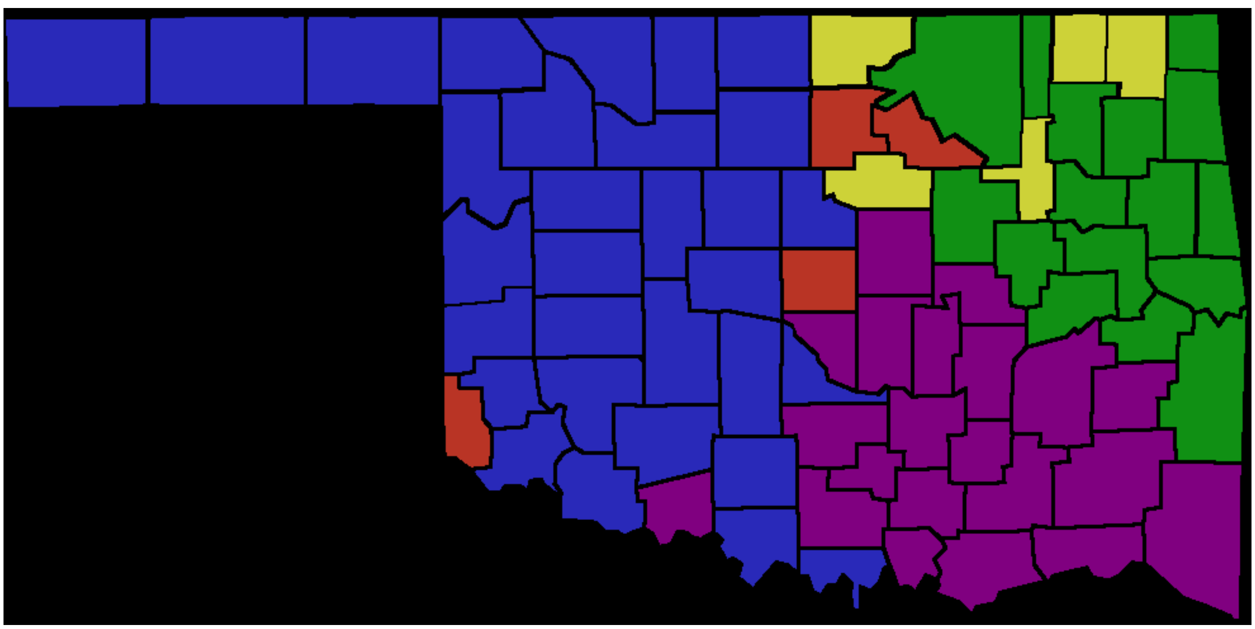

Figure B.6: District assignments when L = 0.999(P) and U = 1.004(P).

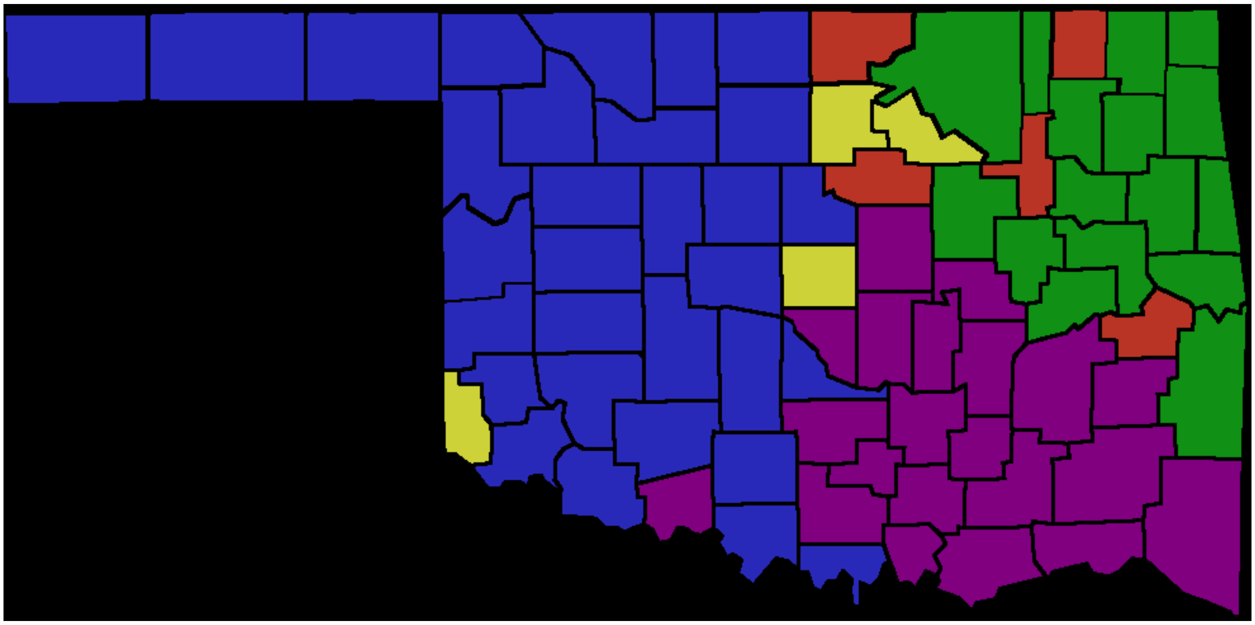

Figure B.7: District assignments when L = 0.999(P) and U = 1.003(P).

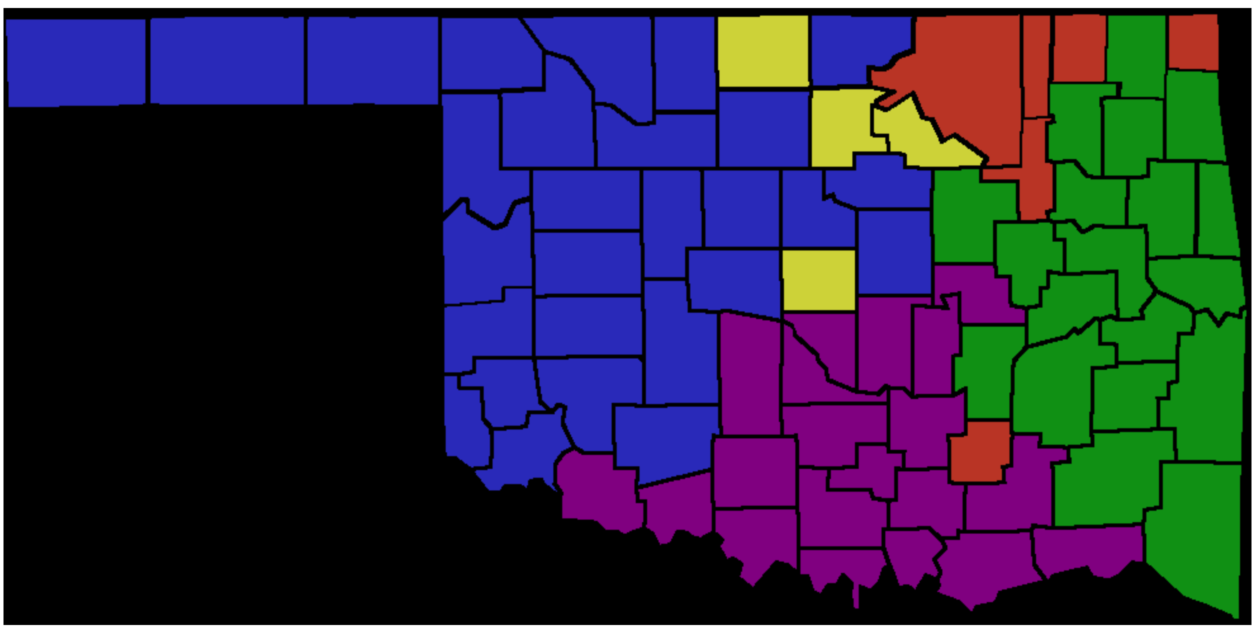

Figure B.8: District assignments when L = 0.999(P) and the U = 1.002(P).

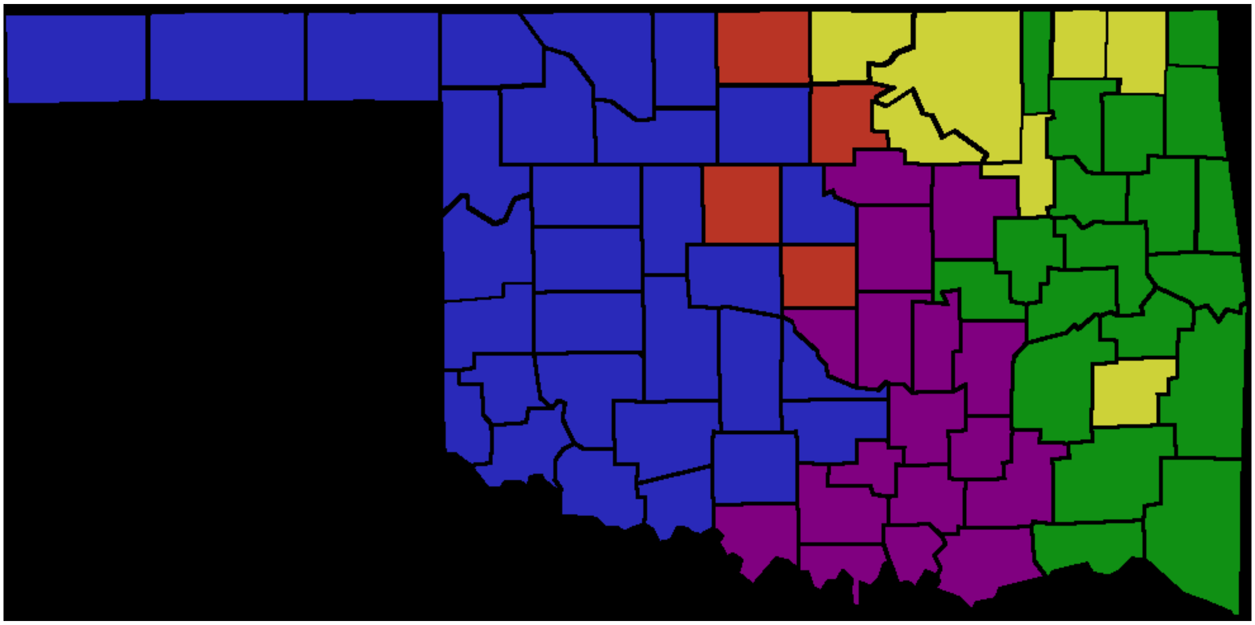

Figure B.9: District assignments when L = 0.999(P) and U = 1.001(P).