Montana Redistricting Final Report

Dream Team

Ashton Parkey, Elizabeth Berka, Zoe Ward

IEM 4013 - Operations Research

Executive Summary Letter

This study addresses the intricate task of redistricting the state of Montana by implementing a linear program. Every decade, states must undergo congressional redistricting to ensure proportional representation and population equality. The Voting Rights Act, enforced by the federal government, emphasizes the importance of minimizing the dilution of minority voting strength within congressional districts.

The state of Montana has 2 congressional districts, the criteria emphasize contiguity, compactness, and fair voting representation. In order to maintain an optimal redistricting plan, linear programming is used, with a focus on minimizing cut edges while imposing contiguity constraints. The resulting plan, consisting of 2 districts, satisfies all federal and state regulations, while achieving a balanced population and contiguity amongst counties.

The results of the model are shown in a map, where the population of District 1 is 539,404 people and the population of District 2 is 544,821. The use of linear programming ensures the compliance with all constraints and provides a straightforward and efficient approach to modeling the revised districts.

Introduction

Every decade, a national census is conducted to determine the need for redistricting in each state, ensuring that the districting systems stay relevant in relation to population distribution. States leverage the redistricting process to draw new electoral district boundaries around counties, aligning appropriate representation with the changing demographics. Given the varying population density across counties, the significance of redistricting boundaries lies in its role of protecting resident's individual vote and maintaining the integrity of the democratic process. (Krause & Jorgensen, n.d.)

Criteria

Federal Criteria:

Every decade, states are mandated to undertake congressional redistricting, aligning with both state and federal standards, The federal guidelines for congressional redistricting focuses heavily on population equality, ensuring that each district within a state is proportionate in size. According to the Congressional Research Service, population equality is enforced under the "equality standard" or "one person, one vote" principle. Additionally, a crucial federal standard, governed by Section 2 of the Voting Rights Act (VRA), "prohibits states or political subdivisions from imposing any voting qualification, practice, or procedure that results in denial or abridgement of the right to vote based on race, color, or membership in a language minority." This provision specifically prevents the drawing of maps that "have the effect of reducing or diluting minority voting strength." (Ragar, 2021)

State Criteria:

According to the Montana code governing redistricting criteria, legislative and congressional districts must be established on the basis of population, aligning with federal law and the The Voting Rights Act. The emphasis is on achieving population equity among districts, that the districts must be as "equal as practicable in population." The district boundaries should coincide with political subdivisions, aiming to minimize the divisions of counties and cities. Priority is given to populus subdivisions, except in instances where a county line passes through a city. All districts must be contiguous and compact, with dimensions deviation approaching zero, and the average length not exceeding the average width by more than three times (unless mandated by the Voting Rights Act). The congressional districts cannot be drawn to favor a political party, incumbent, or member of congress. Outside personal knowledge related to incumbent legislator's addresses, political affiliations, partisan voter lists, or previous election results cannot be considered when redistricting unless required by court. (Ragar, 2021)

Problem Statement

Addressing the complex task of redistricting in Montana, this study aims to leverage linear programming to devise an optimal plan within the constraints of federal and state law. The objective is to achieve an optimal redistricting plan that is contiguous and compact, while maintaining fairness in voter representation and demographic balance.

OR Model (in words)

When considering what to run for our model, we started with looking at deviation and deviation with contiguity, however we found that running minimum cut edges with contiguity worked better for our model and achieved what we were wanting for our redistricting. By imposing a model that minimizes cut edges, the new proposed redistricting map will display modularity and coherence between counties, in turn making the districts more compact. The model below displays an objective of minimizing cut edges, while implementing a contiguity constraint. The population of the district is set to lie within lower and upper bounds, to ensure that deviation is minimized. All counties must remain whole and each county can only be assigned to one district. (Buchanan, 2023)

OR Model (in math)

Minimizing Cut Edges & Implementing Contiguous Districts:

Sets:

- V is the set of nodes (counties)
- E is the set of edges
- N(i) is the set of neighbors of node/county i

Indices:

- i is a county in the map, each node has an index 1 to n
- j is a district within the map, each district gets an index, 1 to k
- e is an edge connecting two counties on the map (each connecting county has one edge)

Variables:

$$x_{ij} = \begin{cases} 1 \; if \; vertex \; i \in V \; is \; assigned \; to \; district \; j \in \{1,2,...,k\} \\ 0 \; otherwise \end{cases}$$

$$y_e = \begin{cases} 1 \; if \; edge \; e \in E \; is \; cut \\ 0 \; otherwise \end{cases}$$

f^v_{ij}= the amount of flow, originating at district center v, that is sent across edge {i,j} (from i to j).

$$r_{ij} = \begin{cases} 1 \; node \; i \; is \; in \; the \; root \; of \; district \; j \\ 0 \; otherwise \end{cases}$$

Parameters:

- U is the upper population bound
- L is the lower population bound
- P_i is the population of county i
- k is the number of districts
- n is the number of nodes / counties
- M = n - k + 1

Objective: minimize number of cut edges	Min $\sum_{e \in E} y_e$
Constraints	s.t
indicate that edge e = {u, v} is cut if vertex u but not v ∈ V is assigned to district j.	$x_{uj} - x_{vj} \leq y_e \quad \forall_e = \{u, v\} \in E, \forall j \in \{1, 2, . . . , k\}$
ensure that each vertex i ∈ V is assigned to one district.	$\sum_{j=1}^{k} x_{ij} = 1 \qquad \forall_i \in V$

ensure that each district's population is between L and U	$L \leq \sum_{i \epsilon V} p_i x_{ij} \leq U \quad \forall_j \in \{1, 2, . . . , k\}$
(Implement Contiguity Constraints) Each node i is assigned to only one district j	$\sum_{i \epsilon V} r_{ij} = 1 \quad \forall_j \in \{1, 2, . . . , k\}$
The population of each district lies between lower and upper population bounds	$r_{ij} \leq x_{ij} \quad \forall_i \in V, \forall_i \in \{1, 2, . . . , k\}$
Force vertex i to consume flow if it is not a root	$\sum_{u \epsilon N(i)} (f_{ui} - f_{iu}) \geq 1 - M \sum_{j=1}^{k} r_{ij} \quad \forall_i \in V$
Disallow flow across cut edges	$f_{ij} + f_{ji} \leq M(1 - y_e) \quad \forall_e = \{i,j\} \in E$
Binary constraints	$x_{ij} \in \{0, 1\} \quad \forall_i \in V, \forall_j \in \{1, 2, . . . , k\}$ $y_e \in \{0, 1\} \quad \forall_e \in E$ $f_{ij}, f_{ji} \geq 0 \quad \forall \{i, j\} \in E, \forall v \in V.$ $r_{ij} \in \{0, 1\} \quad \forall_i \in V, \forall_j \in \{1, 2, . . . , k\}$

(Buchanan, 2022) (Buchanan, 2023)

Python/Gurobi Code

```
import json
from networkx.readwrite import json_graph

def read_graph_from_json(json_file):
    with open(json_file) as f:
        data = json.load(f)
    return json_graph.adjacency_graph(data)
```

```
filepath = '\\Users\\Valued Customer\\Downloads\\redistricting\\'
filename = 'MT_county.json'

G = read_graph_from_json( filepath + filename )
```

```
for node in G.nodes:
    G.nodes[node]['TOTPOP'] = G.nodes[node]['P0010001']
```

```
#Impose a 1% population deviation (+/-1%)
deviation = 0.01

import math
k = 2          # number of districts
total_population = sum( G.nodes[node]['TOTPOP'] for node in G.nodes )

L = math.ceil( ( 1 - deviation / 2 ) * total_population / k )
U = math.floor( ( 1 + deviation / 2 ) * total_population / k )
print("Using L =",L,"and U =",U,"and k =",k)
```

```
Using L = 539402 and U = 544823 and k = 2
```

```python
import gurobipy as gp
from gurobipy import GRB

# create model
m = gp.Model()

# create variables
x = m.addVars(G.nodes, k, vtype=GRB.BINARY) # x[i,j] equals one when county i is assigned to district j
y = m.addVars(G.edges, vtype=GRB.BINARY)    # y[u,v] equals one when edge {u,v} is
```

```python
m.setObjective( gp.quicksum( y[u,v] for u,v in G.edges ), GRB.MINIMIZE )
```

```python
# add constraints saying that each county i is assigned to one district
m.addConstrs( gp.quicksum(x[i,j] for j in range(k)) == 1 for i in G.nodes)

# add constraints saying that each district has population at least L and at most U
m.addConstrs( gp.quicksum( G.nodes[i]['TOTPOP'] * x[i,j] for i in G.nodes) >= L for j in range(k) )
m.addConstrs( gp.quicksum( G.nodes[i]['TOTPOP'] * x[i,j] for i in G.nodes) <= U for j in range(k) )

# add constraints saying that edge {u,v} is cut if u is assigned to district j but v is not.
m.addConstrs( x[u,j] - x[v,j] <= y[u,v] for u,v in G.edges for j in range(k))

m.update()
```

```python
#Add contiguity constraints and re-solve the model.


# Add root variables: r[i,j] equals 1 if node i is the "root" of district j
r = m.addVars( G.nodes, k, vtype=GRB.BINARY )

# Add flow variables: f[u,v] = amount of flow sent across arc uv
#  Flows are sent across arcs of the directed version of G which we call DG

import networkx as nx
DG = nx.DiGraph(G)      # directed version of G

f = m.addVars( DG.edges )
```

```python
M = G.number_of_nodes() - k + 1

# Each district j should have one root
m.addConstrs( gp.quicksum( r[i,j] for i in G.nodes ) == 1 for j in range(k) )

# If node i is not assigned to district j, then it cannot be its root
m.addConstrs( r[i,j] <= x[i,j] for i in G.nodes for j in range(k) )

# if not a root, consume some flow.
# if a root, only send out (so much) flow.
m.addConstrs( gp.quicksum( f[j,i] - f[i,j] for j in G.neighbors(i) )
             >= 1 - M * gp.quicksum( r[i,j] for j in range(k) ) for i in G.nodes )

# do not send flow across cut edges
m.addConstrs( f[i,j] + f[j,i] <= M * ( 1 - y[i,j] ) for i,j in G.edges )

m.update()
```

```python
# solve IP model
m.optimize()
```

```
Gurobi Optimizer version 10.0.3 build v10.0.3rc0 (win64)

CPU model: Intel(R) Core(TM) i7-10875H CPU @ 2.30GHz, instruction set [SSE2|AVX|AVX2]
Thread count: 8 physical cores, 16 logical processors, using up to 16 threads

Optimize a model with 644 rows, 638 columns and 2578 nonzeros
Model fingerprint: 0x1f23a8d2
Variable types: 276 continuous, 362 integer (362 binary)
Coefficient statistics:
  Matrix range     [1e+00, 2e+05]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 5e+05]
Presolve removed 58 rows and 56 columns
Presolve time: 0.00s
Presolved: 586 rows, 582 columns, 2354 nonzeros
Variable types: 276 continuous, 306 integer (306 binary)

Root relaxation: objective 0.000000e+00, 151 iterations, 0.00 seconds (0.00 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.00000    0   60          -    0.00000      -     -    0s
     0     0    0.00000    0   66          -    0.00000      -     -    0s
     0     0    0.00000    0   66          -    0.00000      -     -    0s
H    0     0                      13.0000000    0.00000   100%     -    0s
     0     0    5.43869    0   63   13.00000    5.43869  58.2%     -    0s
     0     0    8.00000    0   64   13.00000    8.00000  38.5%     -    0s
     0     0    8.72548    0   64   13.00000    8.72548  32.9%     -    0s
H    0     0                      10.0000000    8.72548  12.7%     -    0s

Cutting planes:
  Gomory: 2
  MIR: 18
  Zero half: 4


Explored 1 nodes (582 simplex iterations) in 0.11 seconds (0.05 work units)
Thread count was 16 (of 16 available processors)

Solution count 2: 10 13

Optimal solution found (tolerance 1.00e-04)
Best objective 1.000000000000e+01, best bound 1.000000000000e+01, gap 0.0000%
```

```python
print("The number of cut edges is",m.objval)

# retrieve the districts and their populations
districts = [ [i for i in G.nodes if x[i,j].x > 0.5] for j in range(k)]
district_counties = [ [ G.nodes[i]["NAME20"] for i in districts[j] ] for j in range(k)]
district_populations = [ sum(G.nodes[i]["TOTPOP"] for i in districts[j]) for j in range(k) ]

# print district info
for j in range(k):
    print("District",j,"has population",district_populations[j],"and contains counties",district_counties[j])
    print("")
```

```
The number of cut edges is 10.0
District 0 has population 539404 and contains counties ['Big Horn', 'Custer', 'Richland', 'Wibaux', 'Golden Valley', 'Carbon',
'Toole', 'Blaine', 'Judith Basin', 'Park', 'Roosevelt', 'Fallon', 'McCone', 'Stillwater', 'Teton', 'Dawson', 'Powder River', 'G
lacier', 'Hill', 'Treasure', 'Yellowstone', 'Wheatland', 'Valley', 'Petroleum', 'Phillips', 'Sheridan', 'Daniels', 'Prairie',
'Musselshell', 'Chouteau', 'Meagher', 'Rosebud', 'Sweet Grass', 'Lewis and Clark', 'Pondera', 'Garfield', 'Carter', 'Liberty',
'Cascade', 'Fergus']

District 1 has population 544821 and contains counties ['Broadwater', 'Sanders', 'Lincoln', 'Granite', 'Powell', 'Lake', 'Miner
al', 'Jefferson', 'Deer Lodge', 'Ravalli', 'Missoula', 'Madison', 'Beaverhead', 'Gallatin', 'Silver Bow', 'Flathead']
```

(Buchanan, 2023)

Experiments

The optimization model for redistricting Montana was completed using Gurobi Optimizer version 10.0.3. This task was performed on the Intel ® Core(™) i7-10875H CPU with 2.30GHz. The thread count is 8 physical cores, 16 logical processors, using up to 16 threads. Gurobi optimized a model with 644 rows, 638 columns, and 2578 nonzeros. The variable type breakdown consists of 276 continuous variables and 362 binary variables. There was 1 node explored using 582 simplex iterations, in a speed of 0.11 seconds. The optimal solution was found with a tolerance of 1.00e-0.4.

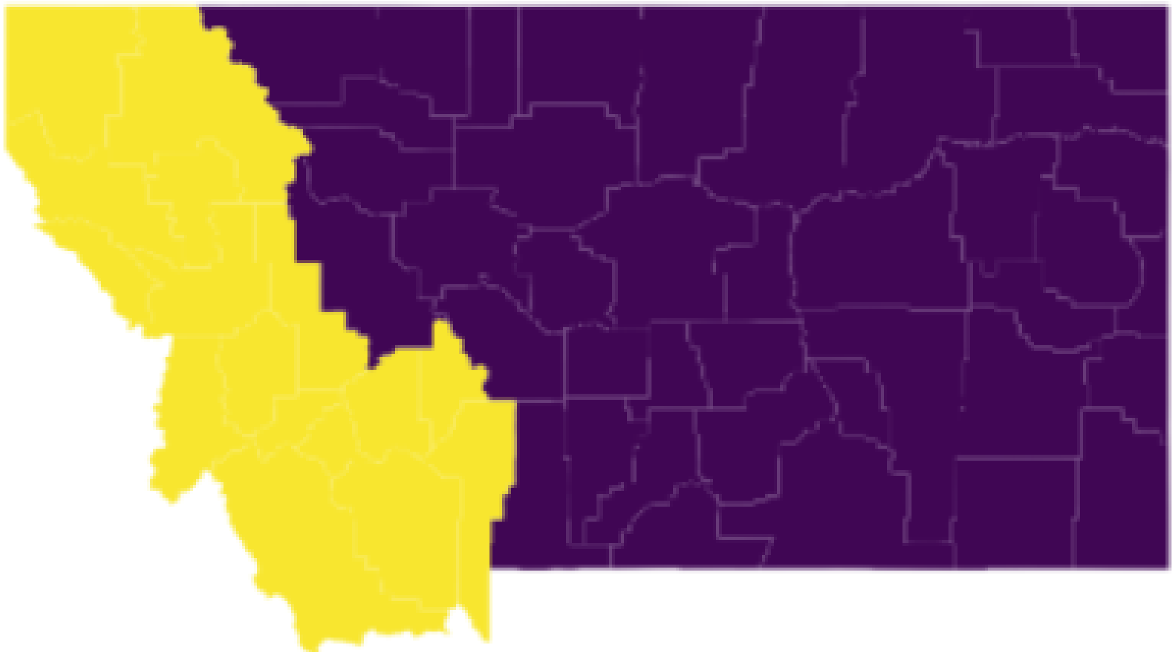
Plan(s) and Map(s)

Plan 1: Minimize Population Deviation

Plan 1 was discontinued upon running the optimization model because it was unable to impose the necessary constraints mandated by Montana law.

Plan 2: Minimize Cut Edges

This model has a total number of 10 cut edges. All counties within the districts are contiguous. A deviation of 0.1% was used. The map below displays the proposed new plan.



(purple represents district 1, yellow represents district 2)

Evaluation of Plan(s)

When tasked with redistricting the state of Montana, the original approach was to minimize the population deviation between districts. The original plan was discontinued upon running the optimization model because it was unable to impose the necessary constraints mandated by Montana law. Although the model was able to minimize the population deviation, it was unable to ensure contiguity between counties. A second plan was imposed with the goal of compactness and contiguity. This plan satisfies all federal and state regulations; counties are contiguous and compact while maintaining fairness for voting demographics.

Conclusions

In conclusion, the new district map for Montana has 2 districts, each with populations around 540,000, both within the required deviation. The results for each district is as follows:

- District 1 has a population of 539,404 people and contains counties ['Big Horn', 'Custer', 'Richland', 'Wibaux', 'Golden Valley', 'Carbon', 'Toole', 'Blaine', 'Judith Basin', 'Park', 'Roosevelt', 'Fallon', 'McCone', 'Stillwater', 'Teton', 'Dawson', 'Powder River', 'Glacier', 'Hill', 'Treasure', 'Yellowstone', 'Wheatland', 'Valley', 'Petroleum', 'Phillips', 'Sheridan', 'Daniels', 'Prairie', 'Musselshell', 'Chouteau', 'Meagher', 'Rosebud', 'Sweet Grass', 'Lewis and Clark', 'Pondera', 'Garfield', 'Carter', 'Liberty', 'Cascade', 'Fergus'].
- District 2 has a population of 544,821 people and contains counties ['Broadwater', 'Sanders', 'Lincoln', 'Granite', 'Powell', 'Lake', 'Mineral', 'Jefferson', 'Deer Lodge', 'Ravalli', 'Missoula', 'Madison', 'Beaverhead', 'Gallatin', 'Silver Bow', 'Flathead'].

The districts found follow the state requirements including contiguity and are close to equivalent. The map found on the district website was similar to the map we got from our code. Being an optimal solution, this could be used as a new districting plan for the state of Montana.

References

Buchanan, A. (2022, April 8). *IEM 4013: Two MIPs for redistricting*. Canvas. Retrieved December 7, 2023, from https://canvas.okstate.edu/courses/159095/files/18529676?module_item_id=5259640

Buchanan, A. (2023, June 16). *Districting Examples*. Github. Retrieved December 7, 2023, from https://github.com/AustinLBuchanan/Districting-Examples-2020

Jackson, S. (2023, September 13). *Montana Districting and Apportionment Commison*. MDAC – Developing Legislative Districts for Montana. Retrieved December 7, 2023, from https://mtredistricting.gov/

Krause, J., & Jorgensen, J. (n.d.). *Montana*. Data USA. Retrieved December 6, 2023, from https://datausa.io/profile/geo/montana

Ragar, S. (2021, November 12). *After an amendment, Montana adopts final congressional map*. Montana Public Radio. Retrieved December 6, 2023, from https://www.mtpr.org/montana-news/2021-11-12/after-an-amendment-montana-adopts-final-congressional-map