Èdouard Lucas:

The theory of recurrent sequences is an inexhaustible mine which contains all the properties of numbers; by calculating the successive terms of such sequences, decomposing them into their prime factors and seeking out by experimentation the laws of appearance and reproduction of the prime numbers, one can advance in a systematic manner the study of the properties of numbers and their application to all branches of mathematics.

Computational Approaches for Political Redistricting Part III: Graph Partitioning

Daryl DeFord

CSAIL – GDP Group

IAP 2019
Massachusetts Institute of Technology
January 22, 2019

SUBTITLE:

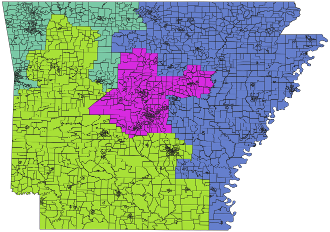
Why *insert–your–field–here*'s favorite districting method isn't the answer.

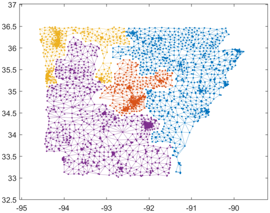
Advertisements

1. VRDI – 6 week summer program for graduate and undergraduate students (Deadline 2/1)
    - Application: tinyurl.com/apply-vrdi-2
    - Information: gerrydata.org
2. Contact:
    - Email: ddeford at mit.edu
    - Website: mggg.org
    - Slack channel: GerryChat.slack.com
3. Research Projects
    - Math Problems: tinyurl.com/gerryprojects
    - Data Problems: tinyurl.com/GerryChainProjects
4. IAP Info:
    - Resources: people.csail.mit.edu/ddeford/CAPR
    - Today 12-1 Graph Partitions
    - 1/29 12-1 In–depth state examples

Outline

Problem Setting





Problem

Given a fixed set of geography we wish to construct representative examples of permissible districting plans.

Desirable Characteristics

Example (What properties do we want?)

Desirable Characteristics

Example (What properties do we want?)

- Efficiency
- Parameter Variability
- Robustness
- Interpretability
- Mathematical Elegance
- All permissible plans are possible

Legal “Requirements”

- Population Balance
- Contiguity
- Compactness
- Municipal Boundaries
- VRA Compliance
- Communities of Interest

Legal "Requirements"

- **Population Balance**
- **Contiguity**
- Compactness
- Municipal Boundaries
- VRA Compliance
- Communities of Interest

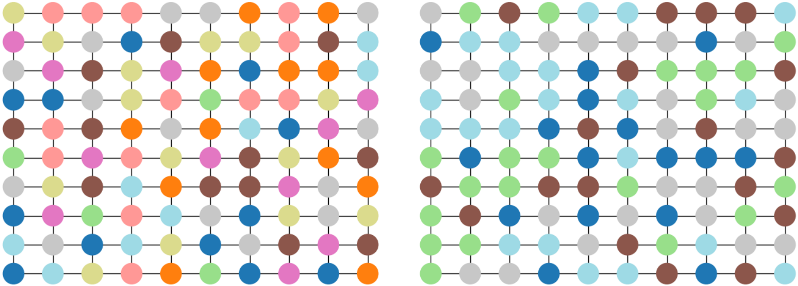
Legal “Requirements”

- Population Balance
- Contiguity
- **Compactness**
- Municipal Boundaries
- VRA Compliance
- Communities of Interest

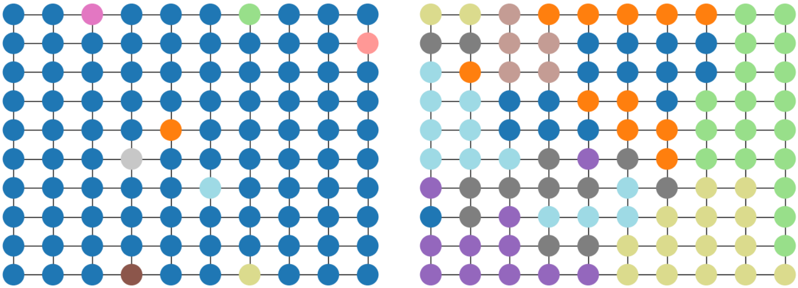
Legal “Requirements”

- Population Balance
- Contiguity
- Compactness
- **Municipal Boundaries**
- **VRA Compliance**
- Communities of Interest

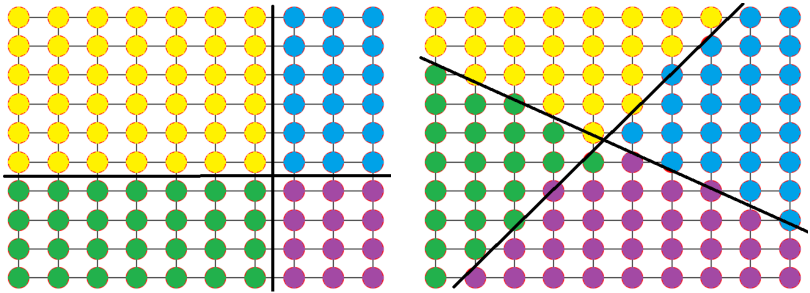
Toy Example: Random Assignment



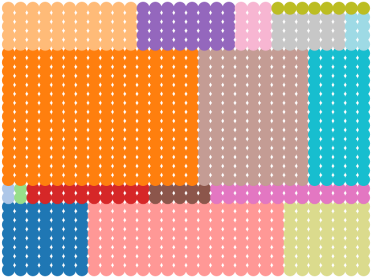
Toy Example: Random Walkers

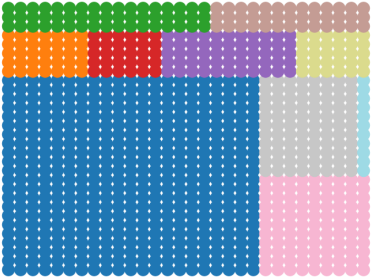


Toy Example: Random Lines



Toy Example: Random Rectangles





Activity

Take a few minutes and partition the four graphs into the indicated number of districts. Think about how you might write an algorithm expressing your approach.

MORAL:

Computational Redistricting is NOT a solved problem!

Power Diagrams

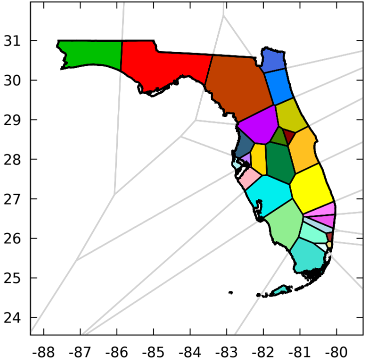


Figure 1: Florida (27 districts)

Figure: Power diagram for Florida: Balanced power diagrams for redistricting: V. Cohen–Addad, P. Klein, and N. Young.

Other Straight Line Methods

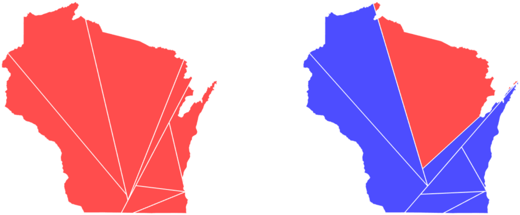


Figure: Split line partitioning of Wisconsin: Partisan gerrymandering with geographically compact districts: B. Alexeev and D. Mixon

- Split Line Methods
- Pretend that everything is a grid
- (Optimization) Draw lines even within households
- Alternatively, embed all voters on a circle

Problems?

- No clean mapping on to discrete units
- Difficult to preserve municipalities, COI, VRA, etc.
- Assumes better control over data than actually exists
- Very hard to tune to arbitrary legal constraints

Growing Districts

- Another popular class of methods are colloquially known as flood fills
- This procedure iteratively creates districts by growing them one node at a time
- Usually, contiguity is enforced at each step
- The process continues until the population is nearly balanced

Flood Fill

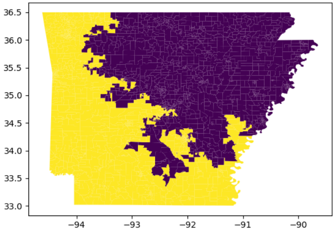
Method

- *Select a node at random*
- *Select a random neighbor of the current cluster*
- *Alternatively, generate a list of neighbors and append sequentially*
- *Add if population allows and doesn't disconnect the complement*
- *Repeat until population balanced*

Path Fill

Method

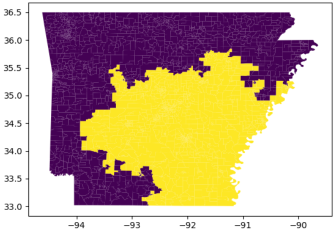
- *Start with an arbitrary node*
- *Select a node not in the district*
- *Add all the nodes on a shortest path from the new node to the district if it doesn't disconnect the complement or add too much to the population*
- *Repeat until population balanced*



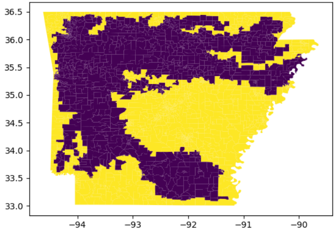
Agglomerative

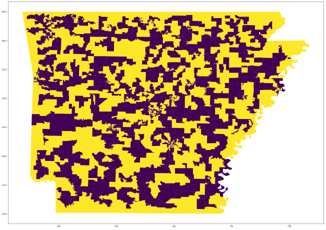
Method

- *Start with each node in own component*
- *Select an arbitrary edge between two components*
    - *Merge clusters if population allows and doesn't disconnect the complement*
    - *If population doesn't allow, delete edge*
    - *If merging would disconnect the graph, merge the smallest population component*
- *Repeat until only 2 clusters*



What can go wrong?

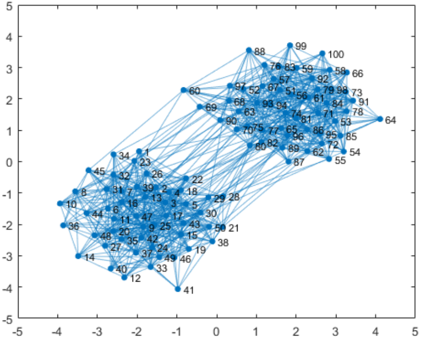




Problems?

- High failure rate
- No control over distribution
- Medium hard to tune to arbitrary legal constraints
- Requires separate cleaning steps

Network Clustering



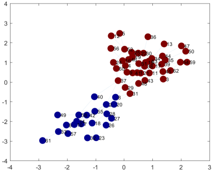
What is a community?

What is a community?

- Many intra–community links
- Few inter–community links
- Any measure that allows for dimension reduction
- Depth or closeness measures
- Different type of eyeball test

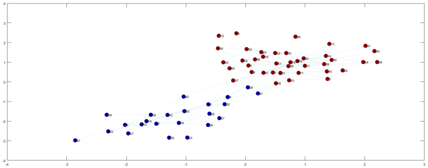
Spectral Clustering

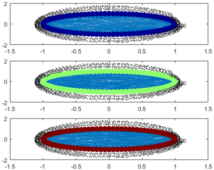
The idea behind spectral clustering is that communities should be sparsely connected to each other. This is usually defined in terms of an isoperimetric ratio, expressing the difference between the size of the boundary and the number of nodes in the community. The solution is given in terms of the eigenvectors of the Laplacian matrix.

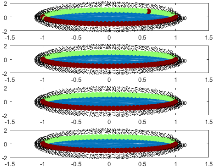


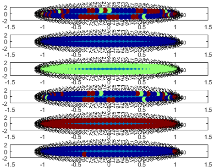
Modularity

For modularity, we take the opposite definition. Now we define a community as a group of nodes that have more connections to each other than would be expected if we rewired the whole network. The solution is given in terms of the eigenvectors of the Modularity matrix.





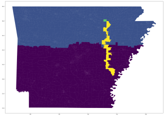


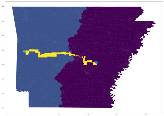


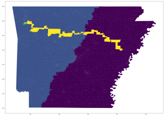
Min Cut

Method

- *Select random source and sink nodes*
- *Weight the edges in the graph by* $10^{min\ distance-3}$
- *Compute the min cut*
- *Repeat until population balanced*



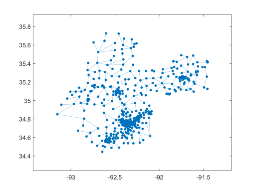


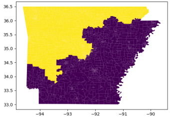


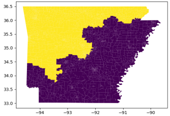
Tree Partitions

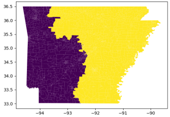
Method

- *Generate a uniform spanning tree*
- *Cut an edge that leaves population balanced components*









Problems?

- The underlying assumption for all of these methods is that the graph structure contains all of the relevant information for defining communities.
- However, for our setting, the useful information is usually annotations, not the nodes/edges themselves.
- For example, spectral clustering and modularity perform quite poorly on dual graphs that are very grid like
- Hard to optimize for many different functions at once

Potential Solution

- Although the naive version of the network approaches seems poorly tuned for our setting there is some hope:
- These methods permit weighted generalizations that allow us to encode some measures of similarity between nodes
  - Demographics
  - Shared Geography
  - COI
  - Municipal Boundaries
- These weighted versions can then be interpreted as maximizing similarity within/minimizing similarity without and used to find larger partitions.
- Some success already, still a long way to go!

Recursive Constructions

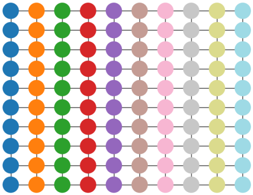
- Choose a methods for constructing a single (contiguous, population balanced, etc.) a district
- Create one and repeat
- In general, bipartitioning, even in the unbalanced setting, is easier than k-partitioning
- Particularly true for many of the network methods, which tend to be significantly more stable for 2–partitions.

Initial Seeds

- One use for these randomly drawn plans is as initial seeds for MCMC
- This provides a good heuristic check for convergence
- This can also solve data issues!

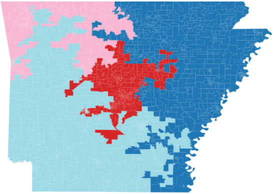
Single Edge Flip Proposals

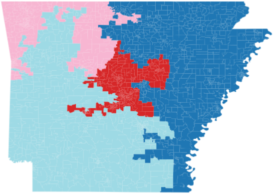
1. Uniformly choose an edge between districts
2. Change one of the incident node assignments to match the other



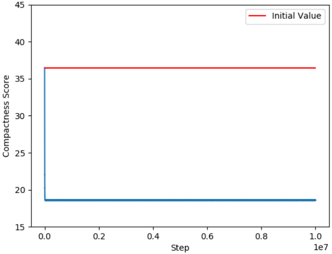
- Mattingly et al. (2017, 2018) Court cases in NC and WI.
- Pegden et al. Assessing significance in a Markov chain without mixing, PNAS, (2017). Court case in PA.

Single Edge Ensembles

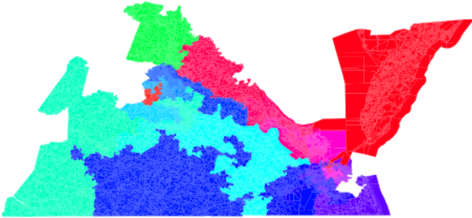




Boundary Flip Distribution

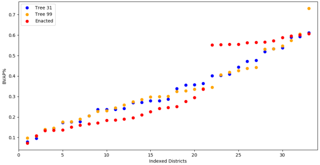


(a) Compactness

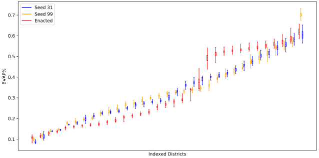


(b) 11996 cut edges

Boundary Flip Mixing

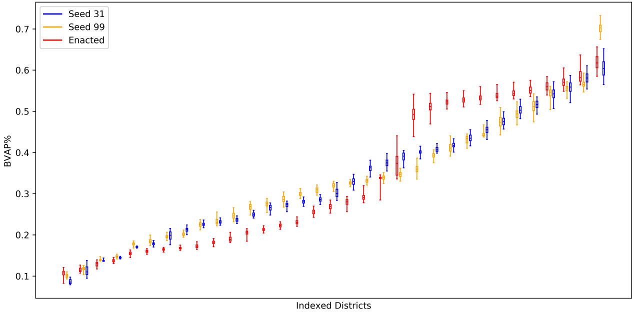


(a) Initial



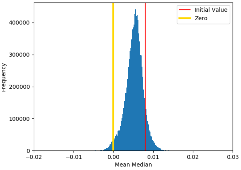
(b) 10,000,000 Flip Steps

Booundary Flip Mixing

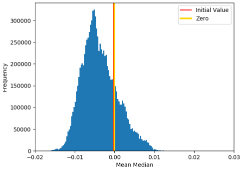


(b) 10,000,000 Flip Steps

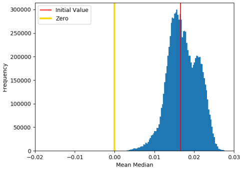
Boundary Flip Mean–Median



(a) Flip Seed31



(b) Flip Seed99



(c) Flip Enacted

Local Variants

- Chunk Flips
- Flip every district
- Snake flips
- Flip whole boundary
- Random walk flips
- ...

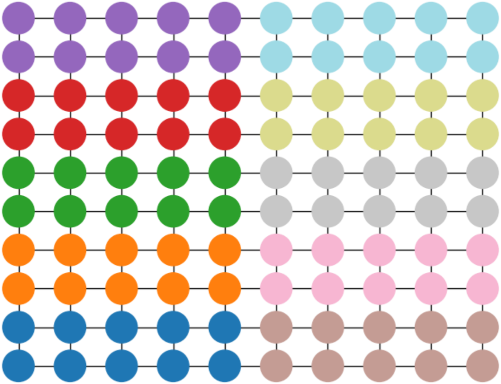
Recombination Steps

1. At each step, select two adjacent **districts**
2. Merge the subunits of those two districts
3. Draw a new district using **your favorite** bipartitioning method.
4. Repeat

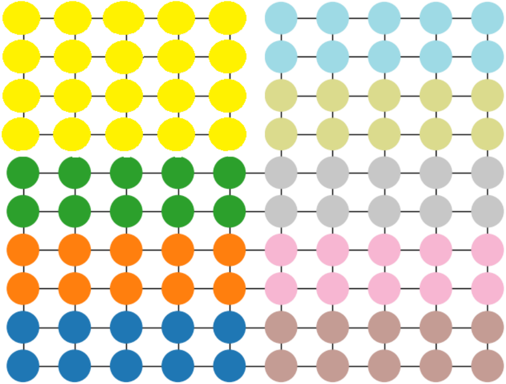
Recombination Steps

1. At each step, select two adjacent **districts**
2. Merge the subunits of those two districts
3. Draw a spanning tree for the new super–district
4. Delete an edge leaving two population balanced districts
5. Repeat
6. (Optional) Mix with single edge flips

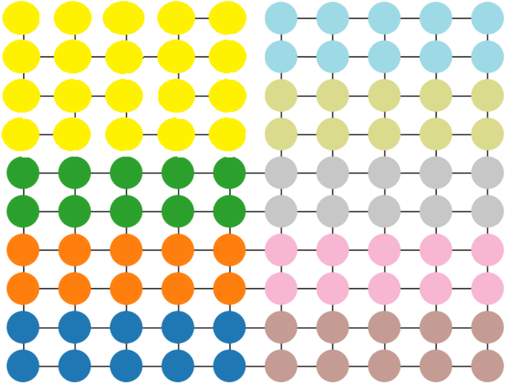
Recombination Step Example



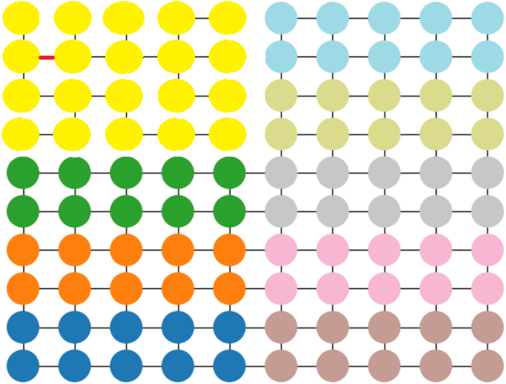
Recombination Step Example



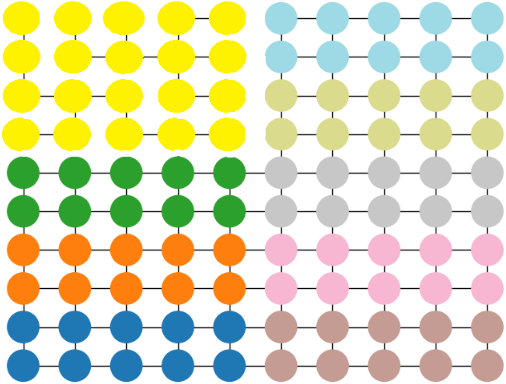
Recombination Step Example



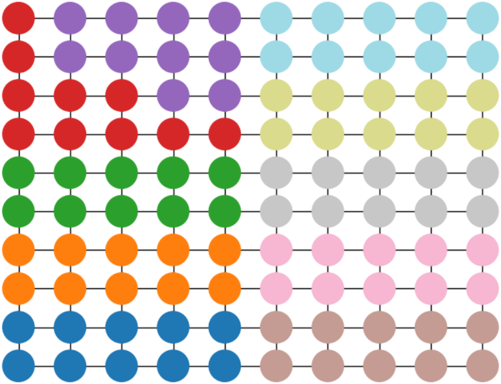
Recombination Step Example



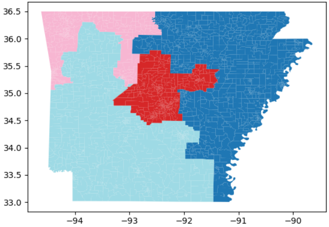
Recombination Step Example

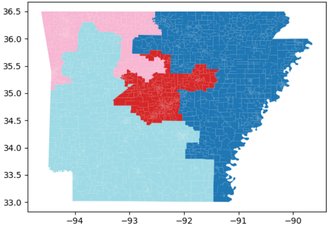


Recombination Step Example



AR Ensembles





PA Recombination Steps

General Tree Proposals

1. Form the induced subgraph on the complement of the cut edges
2. Add some subset of the cut edges
3. Uniformly select a maximal spanning forest
4. Apply a Markov chain on trees
5. Partition the spanning forest into k population balanced pieces

Special Cases

- Uniform Trees: Add all cut edges
- k–edges: Uniformly add k cut edges
- Recombination: Add all cut edges between one pair of districts.
- Super-Recombination: Take a maximal matching on the dual graph to the districts and add all cut edges between matched districts.
- Bounce Walk: Add a single cut edge between enough pairs of districts to make a tree in the dual graph of districts.

Tree Partitioning Questions

- Characterizing the distribution on partitions defined by cutting trees!
- How bad is the best cut?
- Criteria for determining when a tree is ε cuttable?
- Criteria for determining when all spanning trees of a graph are ε cuttable?
- How hard is it to find the mininum ε for which a cut exists?
- As a function of ε what proportion of spanning trees are cuttable?
- As a function of ε what proportion of edges in a given tree are cuttable?
- What is the fastest way to sample uniformly from $k-1$ balanced cut edges?

The End

Thanks!