Graph partitioning for gerrymandering

Irene Simó Muñoz

Algorithms for VLSI
Prof. Jordi Cortadella Fortuny

January, 2024

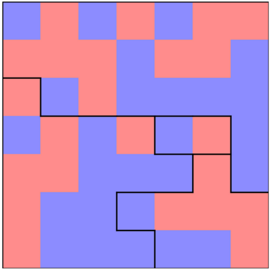
UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

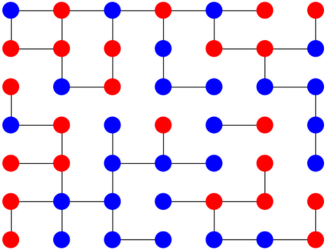
The problem

Goal

Find a graph partition $\mathcal{P}$ s.t.

$$\#\text{ of CCs with blue node majority } > \#\text{ of CCs with red node majority}$$



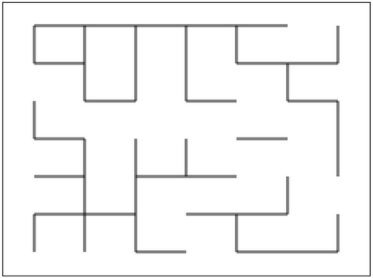


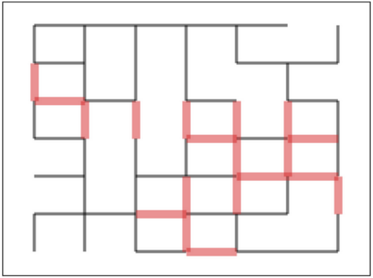
Basics

Two algorithms designed and tested:

- Greedy
- "Smart"

Both based in the addition and removal of *bridge edges*





Main idea of the algorithms

```
Initialize $k := 0$
Start with a random partition $\mathcal{P}^k$
Compute winner $w^k \in \{A, B, \emptyset\}$
while criteria$^k$ and $w^k \neq A$ do
    Compute and perform best migration
    Update $\mathcal{P}^{k+1}$, $w^{k+1}$ and criteria$^{k+1}$
    $k := k + 1$
end while
Return: $\mathcal{P}^* = \mathcal{P}^k$
```

`criteria`: Found local minima, assessment of potential improvements

Key points

Will stop as soon as $w^k = A$

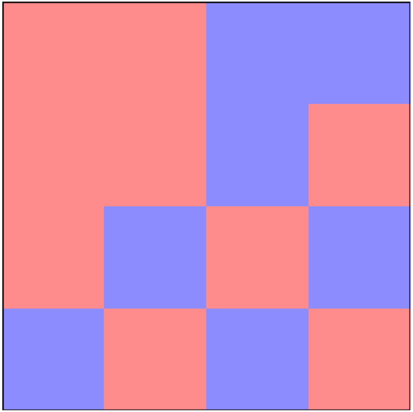
Greedy algorithm

At each iteration:

1. Compute edges not in the graph that would be bridges added to the graph
2. Select u.a.r. one of them that would improve (*w* from opponent to tie or *w* from tie to winner) or keep the overall status
3. Do again until local minima reached

Example 1: Greedy algortihm

Small grid: 4x4, 4 districts



Iteration: 0

Winner: B

"Smart" algorithm

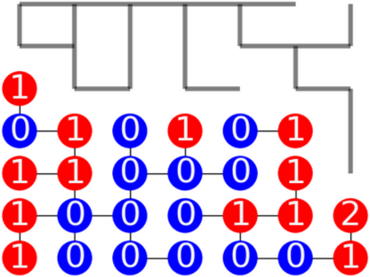
Scoring and potential and Choosing how to redraw borders

- Score of CC. For each connected component $c \in CC$:

$$B_c = \#\text{ Blue nodes } \in c \qquad R = \#\text{ Red nodes } \in c$$

$$s_c = B_c - R_c$$

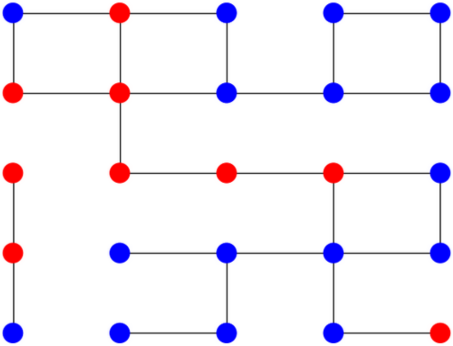
We need to check if we can "traspass" enough blue nodes from winning to loosing districts s.t. we only increase the number of winning districts.



Example of iteration

Iteration: 0

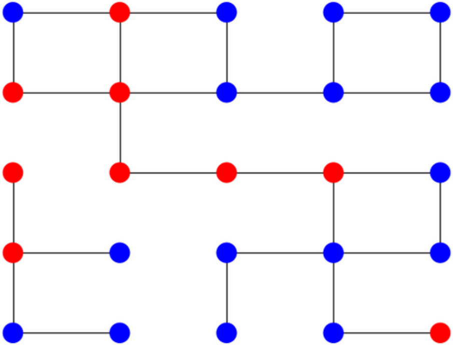
- 5x5 grid with 2 districts
- Configuraiton: Tie
- $s_0 = -1$, $s_1 = 8$. Needed: 2, available: 7



Example of iteration

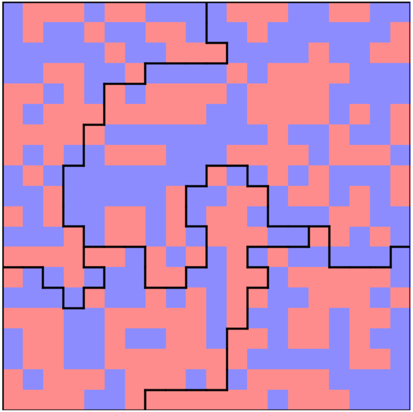
Iteration: 1

- 5x5 grid with 2 districts
- Configuraiton: Blue wins
- $s_0 = 1$, $s_1 = 6$.



Example 2

Greater grid: 10x10, 4 districts



Iteration: 0

Winner: B

Thank you!