How to manipulate elections?: Introducing gerrymandering heuristics

Jozef Janovský
graduate student at Department of Political Science, Masaryk University, Brno, Czech Republic

Introduction

The term *gerrymandering* dates its origin back to 1812, when Governor of Massachusetts Elbridge Gerry signed a bill which redrew boundaries of state senatorial electoral districts. His intention to manipulate electoral results was clear right from the sight of oddly shaped new districts. Since one of them reminded a local cartoonist of a legendary creature salamander, a junction of this word with a surname of the Governor gave rise to the term "**gerrymandering**".

Figure 1. The gerrymandered Essex County salamander. [1]

Even today, the **redistricting process** in the U.S. **is not transparent**. Even though there have been efforts to transfer redistricting rights to independent bodies, it remains in hands of incumbent politicians. Thus, it is still common to encounter apparently gerrymandered districts.

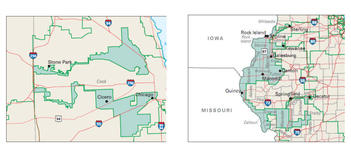

Figure 2. Illinois 4th and 17th congressional district as examples of contemporary gerrymandering. [2]

Graph-theoretical formalization

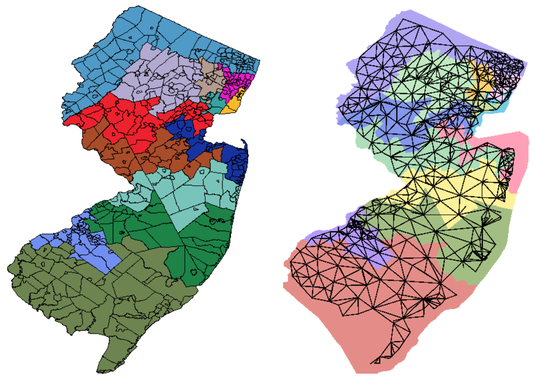

Figure 3. The problem of gerrymandering is, in fact, a graph-theoretical partitioning problem. We transform an electoral map to a graph as follows.

electoral map		graph partition
ward	→	vertex
vote gains in wards	→	vertex weights
ward adjacency	→	edge
map of wards	→	graph
district	→	subgraph

Despite a possibility to modify our assumptions, for the sake of simplicity, we work in the following setting:

- Elections are decided by **plurality** electoral system.
- The same **two political parties** compete in all districts.
- **Electoral results** are **known** in advance.

In this setting, we define **the problem of balanced gerrymandering** on a given graph G as a problem of finding a partition of G into *k* districts such that the number of seats obtained by a given political party is greater or equal to a given constant while all districts of the resulting partition satisfy conditions of **connectivity** and **balance** in terms of their total number of voters.

In our research we have shown that:

The problem of balanced gerrymandering is NP-complete.

In other words, in larger graphs it is generally impossible to find an electoral map with the maximal seat gain for a given political party simply because there are too many attainable partitions that need to be considered.

The gerrymandering heuristics

Since the exact solution to the gerrymandering problem is too hard to find, we have developed a heuristic algorithm aimed at finding a somehow good (although generally not the best) redistricting. It draws on the **actual districting** and attempts to increase the number of seats won by alteration of that electoral map. We provide a description of its functioning (whenever relevant, we control district connectivity):

The algorithm consists of **two main phases**:

1. **The gerrymandering subalgorithm**:
   a) find a losing district with the least number of votes needed to become winning
   b) try to turn the district winning by adding / removing bordering wards (if it is not possible, we rule out the district and return to 1a)

2. **The rebalancing subalgorithm**:
   a) in the acquired partition find the most populationally unbalanced district and by adding / removing bordering wards change it so that it is not the most populationally unbalanced while the number of gained seats does not decrease
   b) If there is no unbalanced district return to 1a (otherwise rule out the district chosen in 1a , return to partition from 1a a go to 1a)

We have also proven that **the algorithm works properly;** it either returns a better gerrymandered partition or the initial partition. However, since the heuristic algorithm is computationally intensive, we have developed its **implementation in Java**. On its input, it has:

- a .shp map of wards,
- a .shp map of districts,
- a .tab file with electoral results on the level of wards.

It returns a .txt file with detailed information on the resulting partition.

Acknowledgments

First of all, we thank Petr Hliněný for his valuable comments and guidance. Furthermore, the contribution of Ján Koščák and Ondrej Moriš was essential for successful technical implementation of our work.

For further information

Please contact me at *jozef.janovsky@gmail.com*. The thesis containing comprehensive material from which this poster is drawn is accessible at http://is.muni.cz/th/273898/prif_b/bakalarka.pdf.

Application to electoral data

We have applied the algorithm to electoral data from 1986 U.S. Congressional election in New Jersey, containing 557 wards grouped into 13 districts. [3, 4] The initial partition is shown in Figure 3 and the gerrymandered partitions in Figure 4.

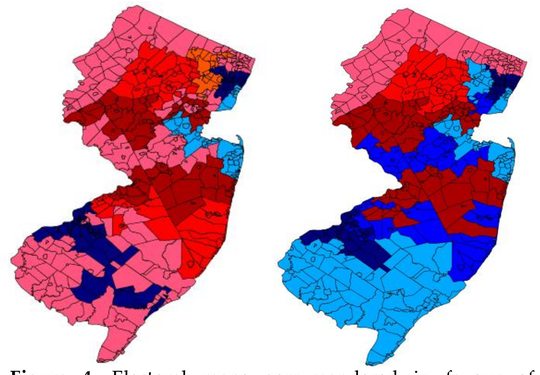

Figure 4. Electoral maps gerrymandered in favour of Republicans (on the left) and Democrats (on the right). The districts shaded in red are won by Republicans and those shaded in blue by Democrats.

From the results of our simulation, it follows that the 1986 election was **highly manipulable**, even though the resulting districts were not allowed to be more unbalanced than the most unbalanced districts from the initial partition (this holds for elections from 1984 and 1988 which we examined as well). If Republicans were to gerrymander, they could have won 9 seats, if it was up to Democrats, they could have also won 9 seats (out of 13). The salamander thus still remains creeping around our elections.

Table 1. Manipulability of the 1986 election

	Rep. seats	Dem. seats
Initial districting	**6**	**7**
Rep.-gerrymandered	**9**	4
Dem.-gerrymandered	4	**9**

Sources

[1] The picture was downloaded from wikipedia.org.
[2] The maps were downloaded from nationalatlas.gov.
[3] U.S. Census Bureau. 2009 TIGER/LineR Shapefiles, 2009 [cit. June 2, 2011]. Available at World Wide Web: <http://www2.census.gov/cgi-bin/shapefiles2009/national-files>.
[4] KING G., et al. The Record of American Democracy, 1984-1990. Cambridge: Harvard University, 1997.