A Computational Investigation of Redistricting Using Simulated Annealing

Vjatšeslav Antoškin
Institute of Computer Science, University of Tartu
Tartu, Estonia
vjatseslav.antoskin@ut.ee

Benson K. Muite
Institute of Computer Science, University of Tartu
Tartu, Estonia
benson.muite@ut.ee

ABSTRACT

Political redistricting is an important operation that is done to ensure a fair selection of electoral representatives. It can be formulated as a combinatorial optimization problem. In realistic cases, this problem can be challenging to solve due to the large number of solutions. The effectiveness of parallel computing to more effectively search the solution space is examined in specially designed test cases where the optimal solution is known.

CCS CONCEPTS

• **Theory of computation → Simulated annealing**; • **Computing methodologies → Massively parallel algorithms**; • **Social and professional topics →** *Governmental regulations*;

KEYWORDS

ACM proceedings, LaTeX, text tagging

ACM Reference Format:
Vjatšeslav Antoškin and Benson K. Muite. 2018. A Computational Investigation of Redistricting Using Simulated Annealing. In *Proceedings of International Conference on Parallel Processing (ICPP'18),* Jennifer B. Sartor, Theo D'Hondt, and Wolfgang De Meuter (Eds.). ACM, New York, NY, USA, Article 4, 2 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

Gerrymandering is the process of creating electoral districts that favor election of a particular candidate or party. In some countries, the redistricting process is done by elected members, who can perform the redistricting process to favor re-election of the incumbent and reduce the competitiveness of the electoral process. It has been suggested that the use of computers to perform redistricting can help obtain a fairer outcome[1, 2, 4].

One way to do redistricting for American congressional districts is by assigning census blocks or counties to particular congressional districts. In doing so, the main principles to be followed are:

- Approximately equal number of voters per congressional district
- Compact congressional districts
- Hole free congressional districts
- Where possible competitive congressional districts with as close to even partisan support

To capture these principles, one defines a global objective function to capture the effectiveness of a particular redistricting plan. An optimization routine is then used to find a good redistricting plan.

2 PREVIOUS WORK

Recent work has introduced PEAR (Parallel evolutionary algorithm for redistricting), that combines parallel computing with genetic evolution to find good redistricting plans[4]. Earlier influential work includes BARD (Better automated redistricting), an open source R package for computational redistricting which has been used in political science courses[1].

Previous work concludes that computers can aid in redistricting, but not completely replace humans in the redistricting process[2]. Important geographical features and historical considerations may be difficult to encode in an objective function that will give rise to a reasonable set of congressional districts. The choice of objective function for computational redistricting will also have an important effect and further study is needed on how to best choose the objective function for a fair redistricting plan.

From a theoretical point of view, if one assumes that the objective function has been correctly chosen, a question of interest is what is the amount of computational work that is required to obtain the best redistricting plan(s). It is expected that parallel computing can help, but unclear what resources are necessary. The novel contribution of this study is to examine cases where the optimal redistricting plan is known.

3 METHODS

In this study, simulated annealing was used to find good redistricing plans. An initial configuration was chosen that satisfied the contiguity and hole free constraints. This was then evolved for between 1000 to 5000 iterations. The configuration with the best score was recorded. Parallelization involved running multiple independent initial conditions in an ensemble (without interaction). More details and source code are in [3].

4 RESULTS

Sample results of finding redistricting plans are shown in table 1. They demonstrate that parallelization can help in searching a wider space to obtain better redistricting plans. However, the amount of computational resources as a function of the number of districts and counties required to have a high chance of obtaining a good solution remains unclear.

ICPP'18, August 2018, Eugene, Oregon USA

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

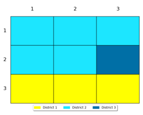
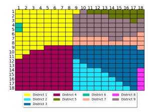
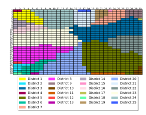
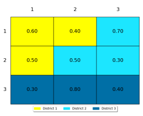
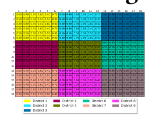
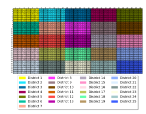
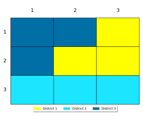
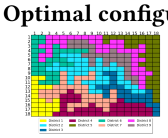
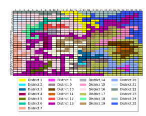
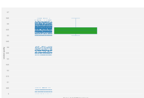
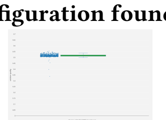
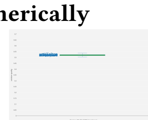
A serial run	Parallel run 60 configurations explored	Parallel run 60 configurations explored
3×3	18×18	30×30
		
	Initial configuration	
		
	Optimal configuration	
		
	Best configuration found numerically	
		
	Explored states	
Optimal configuration found	Parallel computing helps find low score but not optimal configuration	More resources required to obtain a good configuration

Table 1: Results of stochastic annealing in finding known minimal states from randomly generated initial conditions.

5 CONCLUSIONS AND FURTHER WORK

The study presents initial results demonstrating that parallelization can help in searching a wider space to obtain good redistricting plans. The example programs have been written in Python and do not have optimal computational complexity but should be easy to update and experiment with. The results of running the programs indicate that for regions with a small number of districts (such as congressional districts for Idaho and Oregon), parallel computing can obtain the optimal redistricting plan if counties are used for redistricting. For redistricting using census blocks and for regions with many districts, further work is required. For the model problems introduced in this study, an enumeration of the number of possible redistricting plans would be very helpful in determining appropriate computational resources to use to give a high probability of finding the optimal redistricting plan. This study has used simulated annealing to find the optimal redistricting plan. Recent computational studies indicate that genetic evolution algorithms are more effective than simulated annealing. It would be interesting to use these in cases where the optimal redistricting plan is known to determine their effectiveness in real world use.

REFERENCES

[1] Micah Altman and Michael P. McDonald. 2010. The Promise and Perils of Computers in Redistricting. *Duke J Const Law Pub Poly* 5 (2010), 69–159. http://www.law.duke.edu/journals/djclpp/?action=downloadarticle&id=167

[2] Micah Altman and Michael P. McDonald. 2011. BARD: Better Automated Redistricting. *Journal of Statistical Software* 42, 4 (2011), 1–28. http://www.jstatsoft.org/v42/i04

[3] Vjatšeslav Antoškin. 2018. High-performance evolutionary computation for scalable spatial optimization. (2018). https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=61969&year=2018.

[4] Yan Y. Liu, Wendy K.Tam Cho, and Shaowen Wang. 2016. PEAR: a massively parallel evolutionary computation approach for political redistricting optimization and analysis. *Swarm and Evolutionary Computation* 30 (1 10 2016), 78–92. DOI: http://dx.doi.org/10.1016/j.swevo.2016.04.004