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:

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

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored.

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.
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.

REFERENCES


