A Computational Investigation of Redistricting Using Simulated Annealing

Vjatšeslav Antoškin[1] and Benson K. Muite[2]

Abstract

Political redistricting is done to ensure fair selection of electoral representatives. It can be formulated as a combinatorial optimization problem. 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.

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.

One way to do redistricting is by assigning census blocks or counties to particular districts. In doing so, the main principles to be followed for American congressioinal districts are[1, 8]:

i) A congressional district cannot entirely enclose another congressional district (hole-free)

ii) It is possible to traverse from any point in the congressional district to any other point in the congressional district without leaving the congressional district (contiguity)

iii) Congressional districts have the minimum distance between all its parts (compactness)

iv) Approximately equal number of voters per congressional district

v) Where possible competitive congressional districts with as close to even partisan support

The first two principles, hole-free and contiguity, are constraints which are usually strictly enforced[8, 6]. To capture the last three desired 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.

Previous Work

- PEAR (Parallel evolutionary algorithm for redistricting) is a recent work that combines parallel computing with genetic evolution to find good redistricting plans[8, 9].
- Earlier influential work includes BARD (Better automated redistricting) - open source R package for computational redistricting[1].
- Human input will still be needed for redistricting[2].

Research Question

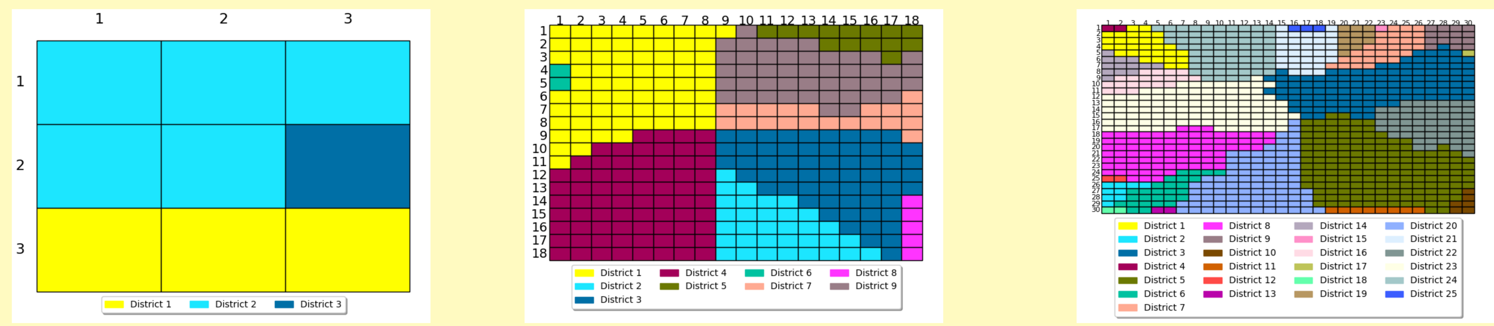
If the objective function has been correctly chosen, what is the amount of computational work that is required to obtain the best redistricting plan(s)?

Method

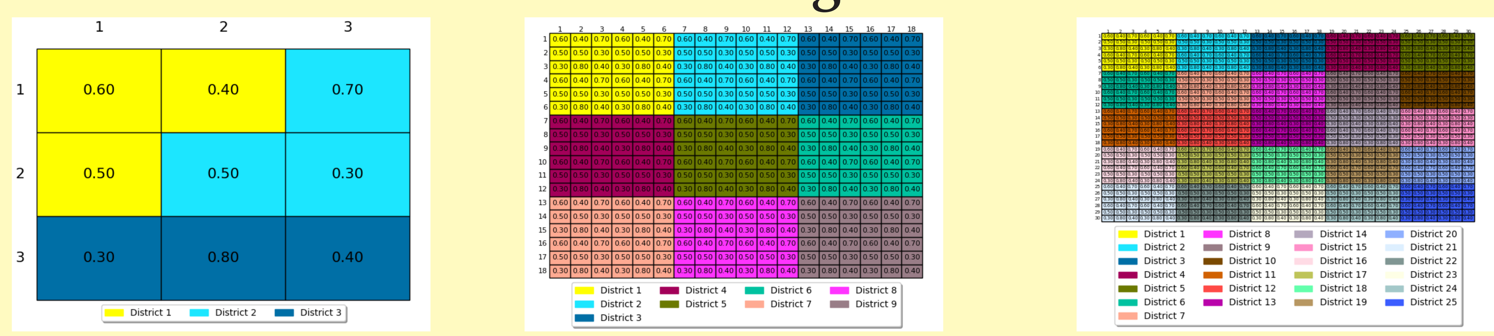
In this study simulated annealing was used to find good solutions. An initial configuration was chosen that satisfied the contiguity and hole free constraints. This was then evolved for between 1000 to 5000 iterations. The configuration with the best score was recorded. Parallelization involved running multiple independent initial conditions in an ensemble (without interaction). More details and source code are in [3, 4].

Results

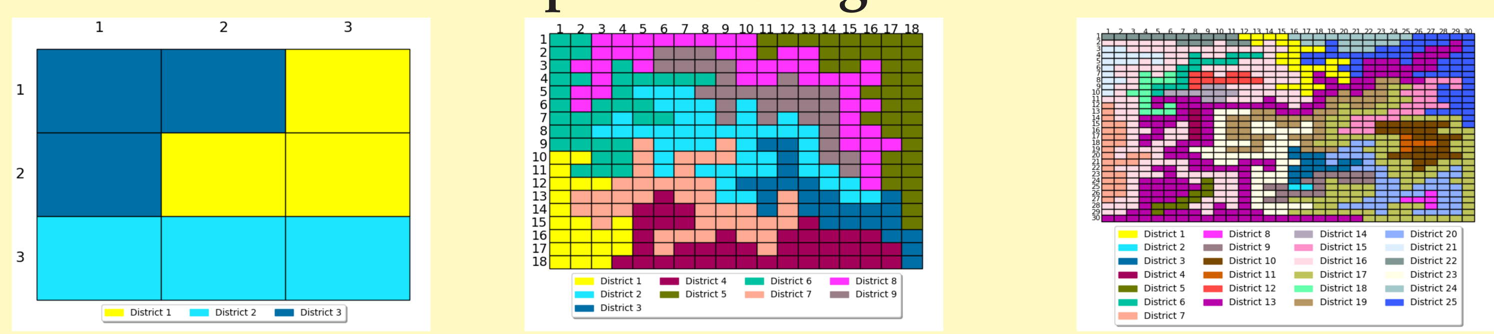
A serial run	Parallel run 60 configurations explored	Parallel run 60 configurations explored
3×3	18×18	30×30



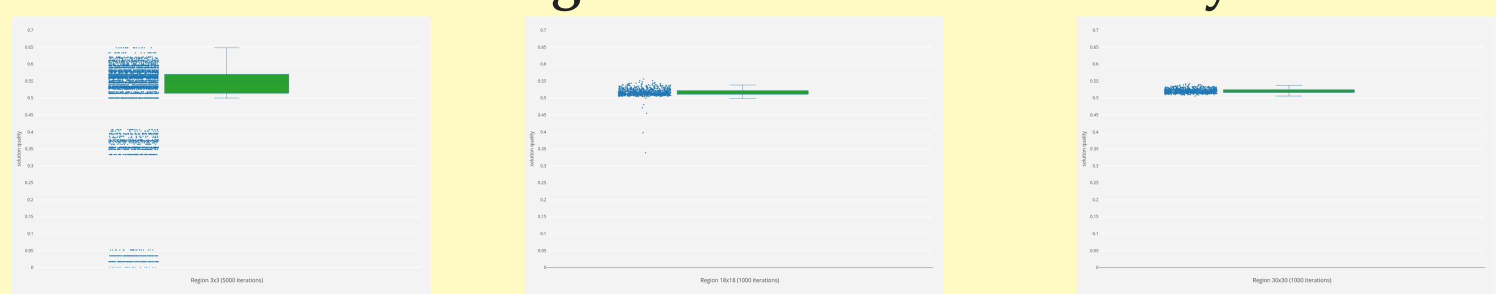
Initial configuration



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

Summary

- Initial results demonstrate parallelization can help in searching a wider space to obtain good redistricting plans.
- The example programs are written in Python and do not have optimal computational complexity but should be easy to update and experiment with[4].
- As found in [5] for states with a small number of congressional districts (such as Idaho and Oregon), computers can obtain good redistricting plans.
- 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.
- It would be interesting to use genetic evolution algorithms in cases where the optimal redistricting plan is known to determine their effectiveness in real world use.

References

[1] M. Altman and M.P. McDonald, "BARD: Better Automated Redistricting", J. of Statistical Software. 42(4), 1–28 (2011).

[2] M. Altman and M.P. McDonald "The Promise and Perils of Computers in Redistricting" Duke J. Constitutional Law and Public Policy. 5, 69–111 (2010).

[3] V. Antoškin, BSC Thesis, "Analysis of a Metaheuristic for Redistricting", University of Tartu (2018).

[4] V. Antoškin, `https://github.com/almaveln/metaheuristic_for_redistricting` (2018).

[5] M.J. Kim, "Optimization approaches to political redistricting problems", PhD Thesis, Ohio State University (2011).

[6] D.M. King, S.H. Jacobson, E.C. Sewell and W.K.T. Cho "Geo-Graphs: An Efficient Model for Enforcing Contiguity and Hole Constraints in Planar Graph Partitioning" Operations Research. 60(5). 1213–1228 (2012).

[7] S. Kirkpatrick, D.C. Gelatt and M.P. Vecchi,"Optimization by Simulated Annealing", Science 220(4598), 671–680 (1983).

[8] Y. Liu, W.K.T. Cho and S. Wang, "PEAR: a massively parallel evolutionary computation approach for political redistricting optimization and analysis", Swarm and Evolutionary Computation 30, 78–92 (2010).

[9] Y. Liu, "High-performance evolutionary computation for scalable spatial optimization", PhD Thesis, University of Illinois at Urbana-Champaign (2017).

Acknowledgments

We thank Yan Liu for helpful discussions. BKM acknowledges partial funding support from the Estonian Research Council [IUT34-4]. This work was carried out in the High Performance Computing Center of the University of Tartu.

Contact

Institute of Computer Science, University of Tartu, Tartu, Estonia

[1] vjatseslav.antoskin@ut.ee [2] benson.muite@ut.ee