Congressional Districting Using a TSP-Based Genetic Algorithm

Sean L. Forman* and Yading Yue

Mathematics and Computer Science Department
Saint Joseph's University
Philadelphia, PA 19131, USA**

Abstract. The drawing of congressional districts by legislative bodies in the United States creates a great deal of controversy each decade as political parties and special interest groups attempt to divide states into districts beneficial to their candidates. The genetic algorithm presented in this paper attempts to find a set of compact and contiguous congressional districts of approximately equal population. This genetic algorithm utilizes a technique based on an encoding and genetic operators used to solve Traveling Salesman Problems (TSP). This encoding forces near equality of district population and uses the fitness function to promote district contiguity and compactness. A post-processing step further refines district population equality. Results are provided for three states (North Carolina, South Carolina, and Iowa) using 2000 census data.

1 Problem History

The United States Congress consists of two houses, the Senate (containing two members from each of the fifty states) and the House of Representatives. The House of Representatives has 435 members, and each state is apportioned a congressional delegation in proportion to its population as determined by a national, decennial census.

Each state (usually the state's legislative body) is responsible for partitioning its state into a number of districts (a *districting plan*) equal to its apportionment. Through years of case law, the courts have outlined several requirements for the drawing of districts [1].

- The districts must be contiguous.
- The districts must be of equal population following the "one-man one-vote" principle.[1]
- The districts should be of a pleasing shape.[2]

* Corresponding author: sean.forman@sju.edu, http://www.sju.edu/~sforman/

** Special thanks to Alberto Segre for providing support and a sounding board for this work, Saint Joseph's University for support of this work, the U.S. Census Bureau Geography Division for their guidance, and the reviewers for their comments.

[1] Despite a census error of 2-3%, a federal court recently threw out a Pennsylvania plan in which the smallest and largest districts varied by 19 people in a state with 12.3 million people [2].

[2] The 1990 North Carolina plan was thrown out due to a district where one candidate quipped he could hit everyone in the district by driving down Interstate-75 with both doors open [3].

E. Cantú-Paz et al. (Eds.): GECCO 2003, LNCS 2724, pp. 2072–2083, 2003.

- The districts should strive not to divide regions with a common interest.
- The districts should not be a dramatic departure from previous plans.

For the purposes of this paper, a districting plan will be considered *valid* if each district created is (1) contiguous and (2) its population is within 1% of the state's total population divided by the number of districts (the population for perfect district population equality). Within these constraints, the algorithm described will attempt to find districts with a compact (pleasing) shape.

2 Current Practices

The redistricting process is largely a partisan process with the political party in power drawing districts likely to keep them in power. This can lead to elongated, poorly shaped districts, or gerrymanders–a contraction of long ago Massachusetts Governor Eldridge Gerry and a salamander. Due to this politicization of the redistricting process, automated redistricting is an appealing alternative.

Automated redistricting typically attempts to find a partition of the state into contiguous districts that minimizes some objective function promoting compactness and an equal distribution of the population. The number of entities to partition can vary from counties (10-150 per state) to census tracts (100-10,000). However, one can quickly see that the number of potential solutions grows very quickly as smaller and smaller entities are used.

In the 1960's, there was a flurry of efforts to apply new computational technology to the redistricting problem beginning with Vickrey's initial work in 1961 [4]. In 1963, Weaver and Hess [5] used an operations research approach similar to the methods used to locate warehouses. In 1965, Kaiser and Nagel developed methods that take current districting plans and improve them by swapping units between adjoining districts [6,7]. In 1973, Liittschwager applied Vickrey's technique to redistricting in Iowa [8].

More recently, several groups have attempted to use local search techniques to find optimal or near-optimal solutions to the redistricting problem [9]. Altman sketches a solution based on node partitioning [10] where the fitness function promotes equality of population and compactness. Mehrotra, et al [11], propose a constrained graph partitioning solution with pre and post-processing steps. Their solution of a transshipment problem to completely balance district populations is mimicked in this paper.

di Cortona [1] and Altman [12] provide excellent backgrounds on redistricting techniques. Altman also provides a proof that the redistricting problem is, in fact, NP-complete by relating it to the class of set partitioning problems.

3 A Redistricting Genetic Algorithm

The genetic algorithm described here (*ConRed*, for *Con*gressional *Red*istricting) takes as an input the number of districts to be drawn, a set of *tracts*, which could mean any partition of a state (counties, townships, census tracts, census blocks, or some combination), each with a population and an area, a list of all inter-tract borders and their lengths, and a list of all borders between a tract and a neighboring state or shoreline and their respective lengths. It then outputs the best districting plan it finds as a list of tracts assigned to districts.

3.1 Fitness Function

The encoding used (Section 3.2) forces every solution to have approximately equal (at worst, 5%-8% error) populations among the created districts.[3] Therefore, the fitness function focuses primarily on contiguity and compactness. Some consideration is given to population equality to encourage solutions within 1% of perfectly balanced.

Methods used to determine the compactness of individual districts and districting plans fall into three primary categories [13,14]:

- Dispersion measures – district length versus width; ratio of district area to that of the smallest circumscribed circle; moment of inertia.
- Perimeter-based measures – sum of all perimeters; ratio of perimeter to area.
- Population measures – population's moment of inertia; ratio of population in district to the population within the smallest circumscribed circle.

Each of these techniques has its own biases and pathological examples. The Schwartzberg Index [15], a perimeter-based measure, was chosen for ConRed because it can be computed quickly and incrementally as tracts are added and subtracted from districts. The Schwartzberg Index measures the compactness of a district as the perimeter of the district squared divided by its area. For this measurement, lower is better, and it is minimized by a circular shape.

The fitness function for ConRed consists of a measurement of the plan's shape and discontiguity (Eq. 1) and a measurement of the variance from equal district populations (Eq. 2) with the final fitness a linear combination of these two parts (Eq. 3).

The shape fitness function (Eq. 3) modifies the Schwarzberg Index to reward contiguity by multiplying this value by one plus the number of excess discontiguous pieces ($pieces_i$) found in the district (preferably, zero) weighted by a parameter, ϕ. The more discontiguous pieces a proposed district has the more severe the penalty will be. The products for individual districts are then averaged across all n districts to give the overall districting plan's shape fitness.[4] Unlike some possible techniques, the encoding does not force a possible plan to be contiguous, but a lack of contiguity is penalized severely in the fitness function by the multiplication of the district's compactness with the number of pieces.

$$Fitness_{shape} = \frac{\sum_{i=1}^{n}(1 + \phi\ (pieces_i - 1))\frac{(perimeter_i)^2}{area_i}}{n} \tag{1}$$

The plan's population variance function depends on the *ideal district population* ($idealDistPop =$ state population divided by the number of districts) and the maximum error allowed in a valid solution ($Var_k = idealDistPop * k$, in our case $k = 1\%$). The first term of Eq. 2 serves as a penalty function (γ determines just how severe) for each district within a districting plan that violates the population constraint. The second term of Eq. 2 drives the population differences towards zero once a valid plan has been found.

[3] Recall that according to the courts, the primary determinants of a hypothetical districting plan's fitness are the equality of its population distribution and the compactness and contiguity of its districts.

[4] One could also choose to find the minimax compactness or any similar measurement.

$$Fitness_{pop} = \frac{\gamma \sum_{i=1}^{n} MAX(|pop_i - idealDistPop| - Var_{1\%}, 0)}{n \times idealDistPop} + \frac{\sum_{i=1}^{n} |pop_i - idealDistPop|}{n \times idealDistPop} \quad (2)$$

Then adding the two together gives the overall fitness (Eq. 3). The parameter θ can be tuned to balance population and shape considerations.

$$Fitness = Fitness_{shape} + \theta \, Fitness_{pop} \quad (3)$$

3.2 Encoding

The encoding chosen for this genetic algorithm is the path representation used for Traveling Salesman Problems (TSP) [16,17,18]. As in the TSP, a single chromosome travels through each tract, and as the tracts are traversed, districts are formed by the sequence of tracts.

This differs significantly from GA techniques used to solve node partitioning problems (NPP) or clustering problems [19]. ConRed does not store the partition of the state in its chromosome per se, but extracts it from the order of the entries. Though details may vary, NPP solutions typically encode the actual partitions of the graph vertices as chromosomes [20]. Clustering problems may be solved similarly, or they may encode the chromosomes as coordinates for the location of the cluster center with data points assigned to the nearest cluster center [19]. The clustering approach would guarantee compactness and contiguity, but the population equality constraint will be difficult to satisfy since it may not always be possible to draw compact districts for a given problem. An NPP encoding that would maintain contiguity and population equality in every chromosome and optimize on compactness would require a significant amount of post-mutation and post-crossover processing to maintain chromosome validity.

Encoding the redistricting problem in a manner similar to the TSP causes each chromosome to have districts with approximately equal populations and relies upon the fitness function to create compactness and contiguity. The trade-off is that the search space is dramatically enlarged with redundant solutions as will be shown below.

To demonstrate how the encoding is translated from a permutation of tracts to a set of districts, take a fictitious state named Botna, with three districts and nine tracts arranged in a 3×3 array (tract population is given in the subscripts), and a chromosome, which is just a permutations of length nine.

1_{30}	2_{20}	3_{10}
4_{10}	5_{20}	6_{30}
7_{10}	8_{20}	9_{30}

$$\mathbf{A} = 1 - 4 - 5 - 2 - 3 - 6 - 9 - 8 - 7$$

Now to convert this permutation into a districting plan, one travels along the chromosome summing the populations until some threshold population is met. The threshold chosen is typically the $idealDistPop \pm \delta$, where δ is typically 5% of the *idealDistPop*. In cases where the number of tracts is small, there is no guarantee that the population

will be split so neatly as they are in this example, but for a large numbers of tracts the approximate equality of district populations is virtually guaranteed.

Table 1. Converting chromosome $\mathbf{A}$ to a districting plan. State population is 180, so the $idealDistPop = 60$.

Tract	Pop.	$\sum$Pop.	District
1	30	30	1
4	10	40	1
5	20	60	1
2	20	20	2
3	10	30	2
6	30	60	2
9	30	30	3
8	20	50	3
7	10	60	3

1	2	3
4	5	6
7	8	9

In the districting plan, $\mathbf{A}$, shown in Table 1, every tract has an area of one and each inter-tract boundary is of length one. The $fitness_{shape}$ of this chromosome, $\mathbf{A}$, can then be evaluated using Equation (1) with $\phi = 2$.

District 1 (1-4-5) Perim. = 8, Area = 3, Fitness = 21.33

District 2 (2-3-6) Perim. = 8, Area = 3, Fitness = 21.33

District 3 (9-8-7) Perim. = 8, Area = 3, Fitness = 21.33

Taking the average of the three districts gives $fitness_{shape} = 21.33$. Here is another example, which illustrates a problem with this encoding (again $\phi = 2$).

$$\mathbf{B} = 1 - 6 - 3 - 2 - 5 - 7 - 9 - 8 - 4$$

Table 2. Example displaying encoding shortcomings.

Tract	Pop.	$\sum$Pop.	District
1	30	30	1
6	30	60	1
3	10	10	2
2	20	30	2
5	20	50	2
7	10	60	2
9	30	30	3
8	20	50	3
4	10	60	3

1	2	3
4	5	6
7	8	9

District 1 (1-6) Perim. = 8, Area = 2, and Pieces = 2, Fitness = 96.
District 2 (3-2-5-7) Perim. = 12, Area = 4, and Pieces = 2, Fitness = 108.
District 3 (9-8-4) Perim. = 10, Area = 3, and Pieces = 2, Fitness = 100.
Taking the average of the three districts gives $fitness_{shape} = 101.33$.

So while this encoding does not force the districts to be contiguous, the fitness function does penalize those districting plans which are not contiguous.

An additional problem with this encoding is that it allows a large number of redundant solutions to be considered. For instance, $(1-6)-(3-2-5-7)-(9-8-4)$ and $(1-6)-(2-3-5-7)-(9-8-4)$ would produce the exact same districting plans in the example above.

3.3 Initial Population & Selection

Several heuristics are used to find an initial population. The algorithm begins each new chromosome at a randomly selected border tract.[5] Adjoining, unvisited tracts are added randomly to the chromosome. The direction of the next selected tract is random, but biased toward the selection of tracts adjoining previously visited elements. This gives some preliminary structure to the permutations. If the chromosome finds no adjoining, unvisited tracts to add, the chromosome jumps randomly to a non-adjoining tract elsewhere in the state and continues adding tracts to the chromosome.[6]

After the fitnesses are calculated for each member of the population, they are ranked and are selected proportionally by their ranks [21]. For example, in a population of ten chromosomes the best chromosome is ten times more likely to be selected than the worst chromosome. Additionally, a copy of the best chromosome is passed to the next generation.

3.4 Genetic Operators

Of the many different operators used on the traveling salesman problem, a crossover, mutation and heuristic operator have been chosen.

Maximal Preservative Crossover. This operator uses a donor and a receiver chromosome [22]. From the donor, a random substring is chosen, call it Ω. All of the elements in Ω are then deleted from the receiver chromosome, and Ω is inserted where the first element of Ω occurs in the receiver. This implementation is slightly altered from Mühlenbein, et al.'s original implementation as they placed Ω at the beginning of the receiver chromosome.

Suppose the following chromosomes are chosen,
$\mathbf{A} = 1-4-5-8-7-2-3-6-9$
$\mathbf{B} = 1-6-3-2-5-4-7-8-9$

[5] While testing was inconclusive, this appeared to be a somewhat better strategy than using the largest tract (by area or population) or a random tract.

[6] This method does provide some structure, but none of the millions of initial chromosomes created in the tests performed (Section 4) represented valid or even contiguous districting plans.

where $\mathbf{A}$ is the donor and $\mathbf{B}$ is the receiver. Suppose $\Omega = 5-8-7-2$. In $\mathbf{B}$, Ω will be inserted after the 3 since 2 is in Ω.

$\mathbf{B} = 1-6-3-\ -\ -4-\ -\ -9 \Rightarrow \mathbf{B} = 1-6-3-\mathbf{5}-\mathbf{8}-\mathbf{7}-\mathbf{2}-4-9$

Exchange Mutation. Exchange mutation [23] cuts a piece out of chromosome and then switches it with a piece of the same size somewhere else in the chromosome. The size of this cut is a parameter `swapSize` that may be entered by the user. Suppose `swapSize` $= 2$, `cutPoint` $= 2$ and `pastePoint` $= 6$.

$\mathbf{B} = 1-(6-3)-2-5-(4-7)-8-9 \Rightarrow \mathbf{B} = 1-(4-7)-2-5-(6-3)-8-9$

Discontiguity Patch. In the course of building chromosomes, one often finds islands of tracts from a district trapped within another district (see Table 2). A heuristic clean-up of these islands can significantly improve a chromosome's fitness while not altering its underlying character. The clean-up process involves searching for discontiguous districts in a districting plan, removing one of the smaller pieces from the string of tracts and then re-inserting it after a tract that borders the first tract in the island piece. For example with Table 2, tracts 4, 8, and 9 are part of the same district, but tract 4 is discontiguous from tracts 8 and 9. The heuristic snips tract 4 from the string, and then randomly inserts it after one of the tracts that it borders (1, 5, or 7). The resulting plan removes several of the discontiguities and has an improved fitness of 75.33.

$1-6-3-2-5-7-9-8-(4) \Rightarrow 1-(4)-6-3-2-5-7-9-8$

3.5 Pre-processing

Population equality tends to improve and compactness and contiguity tends to worsen as the number of tracts increases. The coarsest tract structure available is the county level. But since most states have at least some counties with very large populations, it is often necessary to partition a large county into a set of subtracts, ideally each with an equal population. For instance, a large county can be divided into a number of smaller tracts[7] and then used with the other unbroken counties to give a data set with the population more evenly divided among the input tracts. To create these subtracts (itself essentially a districting problem), ConRed was applied on individual counties with populations above a prescribed threshold. In Table 3, the number of tracts increases as this threshold is lowered. A low population threshold for subdividing a county will lead to more tracts in the input set and possibly less contiguity, but it will increase the probability that a solution will have a low population variance as Table 3 shows.

3.6 Post-processing

The encoding used provides approximate equality of district populations, but often only within 5% of the ideal district population rather than the 1% that is considered valid in this context. To bring the district populations into this desired range, a post processing step has been added. This step is applied following the fitness evaluation and only to

[7] For these experiments, this number was capped at nine.

chromosomes that already generate contiguous plans. A desired population reapportionment is calculated using a transshipment algorithm. Guided by this reapportionment, ConRed then proceeds to swap boundary tracts between districts until no further population balancing is possible. The transshipment algorithm [24] is a common operations research technique used to determine the optimal shipments between a set of known supply and demand points (commonly factories, warehouses and stores). Each possible branch has a cost associated with it, so the objective is to minimize the overall cost while satisfying the given demand using the given supply.

As applied to this problem, the districts with larger-than-ideal populations are suppliers for demanders which are districts with smaller-than-ideal populations. The costs are structured to guarantee that swapping occurs only between districts bordering each other, therefore some districts may act as go-betweens (both suppliers and demanders) for districts needing to adjust their populations. This problem is solved using the transportation simplex method. This technique for balancing district population was suggested by Mehrotra, et al [11].

3.7 Parameters

The primary parameters for tuning are the probabilities for the three operators, the maximum and minimum lengths of the segments removed and inserted by the operators, and the fitness function's parameters: γ, θ, and ϕ. Some work has been done on tuning parameters for this GA, but it is an area where considerable improvement may still be possible. For the experimental results that follow, $\phi = 2 * districts$, $\gamma = 10$, $\theta = 500$, $p_{mpx} = .40$, $p_{exc} = .10$, and $p_{disc} = .40$. The length of any exchange or patch is at most ten tracts and the crossover can be of any length.

4 Experimental Results

ConRed[8] performed a total of 80 runs for each of three states: North Carolina (13 districts), South Carolina (6 districts), and Iowa (5 districts).[9] For each state, 20 runs were made on 4 different tract layouts created by decrementing the pre-processing population threshold from 100,000 to 25,000 (Section 3.5). For these 20 runs, GA population sizes were incremented from 500 to 2000 chromosomes (by 500) for 5 runs each. The number of generations was chosen such that there were 2,000,000 plan evaluations for each run.

Results are presented in Table 3. *Contig* gives the number of runs whose best, final solutions are contiguous and *N valid* is the number of those plans that are both contiguous and have a population variation within 1.0% of ideal. *PopVar* is a plan's largest district population variance from the ideal. Time is in seconds. The results in the **Best** column

[8] ConRed is written in ANSI C and was tested on a 867MHz Pentium III using RedHat Linux.

[9] The U.S. Census Bureau provides a pair of files that are needed to run tests on actual state data. *Summary File 1* (SF1) [25] provides information on population, area, and latitude and longitude for a variety of entity sizes (county, county subdistrict, census tract, census block) for each state. *TIGER/Line files* [26] contain all boundary information between various entities found in the SF1 files. A series of Perl scripts were written to build the necessary input files and generate the postscript files used to visualize final solutions.

are the best fitness across all runs and the minimax population variance across all of the runs, so the two numbers may be from different runs. The best, valid solutions are shown in Figure 1.

The results show that for small states with six or fewer representatives (26 of 50 states), ConRed will draw compact districts of essentially equal populations. For a larger state like North Carolina (42 of the 50 states have 13 representatives or less), ConRed can produce contiguous districts, but is not able to consistently bring the populations close enough together to be a valid solution. This may be a matter of parameter tuning. A higher setting for θ may produce better results. Still these results could be used as rough plans and then manipulated along the borders to produce an equal population distribution.

Unfortunately, it has been difficult to locate other existing solvers with which to compare ConRed. Comparisons to the existing districting plans are interesting, but legislatures are not attempting to form compact districting plans, so they are of little informational value. Almost any computational technique will produce a solution more compact than those drawn by a legislature. In the 1960's, Kaiser-Nagel [6] and Weaver [5] produced plans with compactness, but population variances greater than 1%. In 1996, Hayes [3] produced a twelve-district plan for North Carolina with population variation of less that 1%, but no concern for compactness. In 1998, Mehrotra [11] produced a plan for South Carolina beginning with 48 tracts. They initially found a solution with a population variation of 4.4%. A transshipment step determined the amount of population that needed shifting to achieve district population equality, and then by hand (it appears) tracts were sliced away on district boundaries until an equal population distribution was created. A visual inspection suggests that the valid solutions produced by ConRed are as compact as their plan.

Table 3. ConRed experiments. See text (Section 4) for an explanation.

				N	Average		Time	Best	
	Tracts	Runs	Contig	valid	Fitness	PopVar%	(sec)	Fitness	PopVar%
IA	108	20	20	20	27.8	0.66	568	25.1	0.19
5 dist.	115	20	20	20	28.1	0.60	612	26.1	0.20
	124	20	20	18	29.2	0.65	624	25.4	0.25
	164	20	20	20	28.5	0.49	722	25.2	0.12
SC	69	20	20	15	37.1	0.89	571	32.4	0.48
6 dist.	83	20	20	13	35.0	0.90	607	30.6[8]	0.58
	115	20	20	18	39.9	0.84	725	33.3	0.55
	183	20	20	19	38.7	0.69	883	34.1	0.43
NC	147	20	17	0[9]	71.5	2.33	1669	43.3	1.36
13 dist.	166	20	16	0	79.6	2.58	1797	38.7	1.06
	212	20	17	0	73.9	1.65	1792	43.0	1.08
	352	20	8	2	173.3	1.95	1715[10]	41.9	0.83

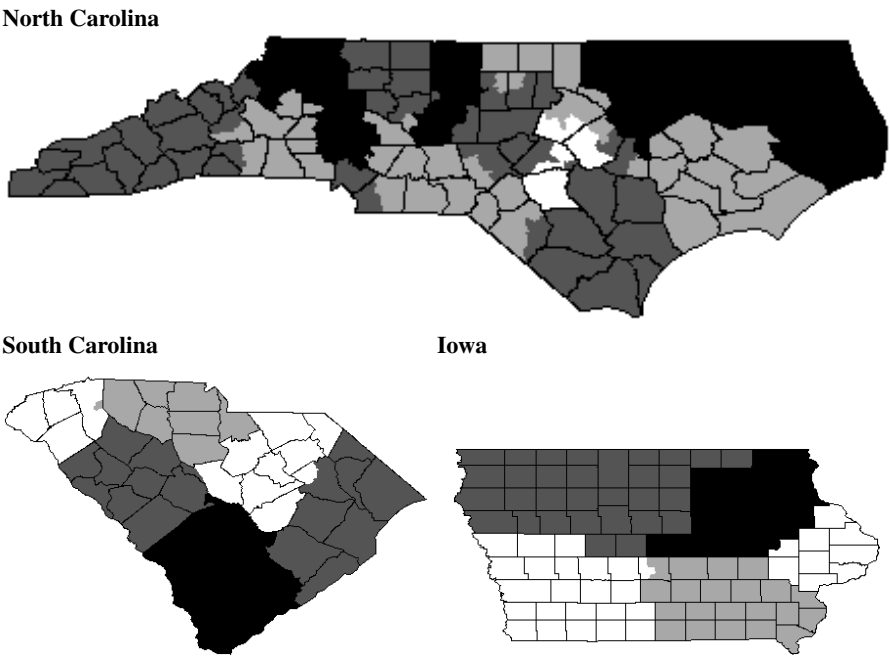

Fig. 1. ConRed's best, valid, final solutions for North Carolina, South Carolina and Iowa. North Carolina's solution divides 352 tracts into 13 districts and has a fitness of 45.1 and a maximum population variation of 0.86%. There are better shaped solutions than this, but their population variation fell just outside the 1% cutoff. South Carolina's solution divides 83 tracts into 6 districts and has a fitness of 30.6 and a maximum population variation of 0.81%. Iowa's solution divides 108 tracts into 5 districts and has a fitness of 25.1 and a maximum population variation of 0.44%. All interior borders shown are county borders.

5 Future Work

Additional heuristics, improved genetic operators, and additional parameter tuning may provide further improvement in the quality of the solutions produced. Thus far, most of the work has focused on the heuristics and the visualization aspects of the project, so further attention to parameter tuning and the choice of operators will be the next area of study. In addition, politicians may also be interested in the distribution of political affiliations within districts, and applications to other location or partitioning problems have yet to be considered.

[8] This particular solution was found by ConRed on four separate runs.

[9] Of the 58 contiguous solutions for NC, maximum population variation ranged from 0%-1% for two solutions, 1.0%-1.5% for 23, 1.5%-2.0% for 18, and above two for 15.

[10] Due to the lower number of contiguous solutions, the post-processing step was performed less often than in other problems, hence the lower time required despite a larger number of tracts.

6 Conclusions

The genetic algorithm presented in this paper, ConRed, uses an encoding adapted from traveling salesman problems to produce possible congressional districting plans. ConRed's fitness function promotes compactness and contiguity and its encoding provides approximate equality of district population for free. Two genetic operators have been implemented in ConRed along with a single heuristic operator. Experimental results on three small to medium states show significant success at producing compact, contiguous districts of nearly equal population.

References

1. di Cortona, P.G., Manzi, C., Pennisi, A., Ricca, F., Simeone, B.: Evaluation and Optimization of Electoral Systems. Monographs on Discrete Mathematics and Applications. SIAM (1999)
2. Milewski, M.: Court overturns redistricting. The Intelligencer (2002)
3. Hayes, B.: Machine politics. American Scientist **84** (1996) 522–526
4. Vickrey, W.: On the prevention of gerrymandering. Political Science Quarterly **76** (1961) 105
5. Hess, S., Weaver, J., Siegfeldt, H., Whelan, J., Zitlau, P.: Nonpartisan political redistricting by computer. Operations Research **13** (1965) 998–1006
6. Kaiser, H.: An objective method for establishing legislative districts. Midwest Journal of Political Science (1966)
7. Nagel, S.: Simplified bipartisan computer redistricting. Stanford Law Review **17** (1965) 863–899
8. Liittschwager, J.: The Iowa redistricting system. Annals of the New York Academy of Science **219** (1973) 221–235
9. Ricca, F., Simeone, B.: Political redistricting: Traps, criteria, algorithms, and trade-offs. Ricera Operativa **27** (1997) 81–119
10. Altman, M.: Is automation the answer? the computational complexity of automated redistricting. Rutgers Computer and Technology Law Journal **23** (2001) 81–142
11. Mehrotra, A., Johnson, E.L., Nemhauser, G.L.: An optimization based heuristic for political districting. Management Science **44** (1998) 1100–1114
12. Altman, M.: Modeling the effect of manadatory district compactness on partisan gerrymanders. Political Geography **17** (1998) 989–1012
13. Young, H.: Measuring compactness of legislative districts. Legislative Studies Quarterly **13** (1988) 105–111
14. Niemi, R., Grofman, B., Carlucci, C., Hofeller, T.: Measuring compactness and the role of a compactness standard in a test for partisan and racial gerrymandering. Journal of Politics **52** (1990) 1152–1182
15. Schwartzberg, J.: Reapportionment, gerrymanders, and the notion of compactness. Minnesota Law Review **50** (1966) 443–452
16. Jog, P., Suh, J.Y., van Gucht, D.: Parallel genetic algorithms applied to the traveling salesman problem. SIAM Journal of Optimization **1** (1991) 515–529
17. Grefenstette, J., Gopal, R., Rosimaita, B., van Gucht, D.: Genetic algorithms for the traveling salesman problem. In: International Conference on Genetic Algorithms and their Applications. (1985) 160–168
18. Larra naga, P., Kuijpers, C., Murga, R., Inza, I., Dizdarevic, S.: Genetic algorithms for the traveling salesman problem: A review of representations and operators. Artificial Intelligence Review **13** (1999) 129–170

19. Sarkar, M., Yegnanarayana, B., Khemani, D.: A clustering algorithm using an evolutionary programming-based approach. Pattern Recognition Letters **18** (1997) 975–986
20. Chandrasekharam, R., Subramanian, S., Chaudhury, S.: Genetic algorithm for node partitioning problem and applications in VLSI design. IEEE Proceedings: Comput. Digit. Tech. **140** (1993) 255–260
21. Baker, J.: Adaptive selection methods for genetic algorithms. In Grefenstette, J., ed.: Proc. of the 1st International Conference on Genetic Algorithms and their Applications, Hillsdale, NJ, Lawrence Erlbaum Associates (1985) 101–111
22. Mühlenbein, H., Gorges-Schleuter, M., Krämer, O.: Evolution algorithms in combinatorial optimization. Parallel Computing **7** (1988) 65–85
23. Banzhaf, W.: The "molecular" traveling salesman. Biological Cybernetics **64** (1990) 7–14
24. Winston, W.L.: Operations Research: Applications and Algorithms. Volume 3. Duxbury Press (1994)
25. U.S. Census Bureau Washington, DC: Census 2000 Summary File 1. (2001)
26. U.S. Census Bureau Washington, DC: Census 2000 TIGER/Line Files. (2000)
27. Goldberg, D.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley (1989)