Soft Comput (2005) 9: 341–348
DOI 10.1007/s00500-004-0413-4

FOCUS

Fernando Bação · Victor Lobo · Marco Painho

Applying genetic algorithms to zone design

Published online: 16 September 2004

Abstract Genetic algorithms (GA) have been found to provide global near optimal solutions in a wide range of complex problems. In this paper genetic algorithms have been used to deal with the complex problem of zone design. The zone design problem comprises a large number of geographical tasks, from which electoral districting is probably the most well known. The electoral districting problem is described and formalized mathematically. Different problem encodings, suited to GA optimization, are presented, together with different objective functions. A practical real world example is given and tests performed in order to evaluate the effectiveness of the GA approach.

Keywords Zone design · Genetic algorithms · Electoral districting

1 Introduction

The zone design problem (also known as districting) occurs when n areal units are aggregated into k zones such that some value function is optimized, subject to constraints on the topology of the zones (e.g. internal connectivity) [1]. Probably the most well known instance of the zone design problem is the electoral districting problem (for some recent proposals see [2–5]). Electoral districting consists of the partitioning of areal units, generally administrative units, into a predetermined number of zones (districts) such that the units in each zone are contiguous, each zone is geographically compact and the sum of the populations of the areal units in any district are as similar as possible or lies within a predetermined range [5]. It is clear that this problem can be seen as a special case of the more general knapsack or clustering problems.

Zone design is an important geographical problem that is present in a number of geographical tasks besides electoral districting (references to other areas of application can be found in [4, 6]). Zone design algorithms have been used in school districting [7], in the design of zones with appropriate characteristics for posterior socio-economic and epidemiological analysis [8–10], in the design of sales territories [11] and the design of census output geography [12, 13].

The constraints of the zone design problem are similar to the ones that characterize the clustering problem. Let the set of initial areal units be $\boldsymbol{X} = \{x_1, x_2, \ldots, x_n\}$, where x_i is the ith areal unit. Let the number of zones be k. Let $\boldsymbol{Z}_i$ be the set of all the areal units that belong to zone $\boldsymbol{Z}_i$. Then:

$$\mathbf{Z}_i \neq \emptyset, \text{ for } i = 1, \ldots, k, \tag{1}$$

$$\mathbf{Z}_i \cap \mathbf{Z}_j = \emptyset, \text{ for } i \neq j, \tag{2}$$

$$\cup_{i=1}^{k} \mathbf{Z}_i = \mathbf{X} \tag{3}$$

These constitute the set of constraints that can be applied equally in clustering and in zone design. Nevertheless, in zone design an additional constraint has to be included, which accounts for contiguity and creates a more complex problem. This constraint limits the set of acceptable solutions to the problem and consists in assuring contiguity between all the areal units that build up a zone. Contiguity means that each areal unit in a zone is connected to every other areal unit via areal units that are also in the zone.

Algorithms designed to deal with the zone design problem (for a thorough review see [14]) can broadly be divided into three categories based on how they approach the problem. The first category starts by building all individual zones separately and then aggregating them into a global solution [15–17]. The second category consists in modifying an existing plan (a plan is an acceptable solution to the problem) by

F. Bação (✉) · V. Lobo · M. Painho
Instituto Superior de Estatística e Gestão de Informação,
Universidade Nova de Lisboa, Campus de Campolide–1070-312 Lisboa
E-mail: bacao@isegi.unl.pt
Tel.: +21 387 15 73
Fax: +21 387 21 40

swapping areal units between zones. Typically a random solution to the problem is generated or a previous existing one is used as seed. The idea is to incrementally modify the solution in order to improve it [4, 5, 8, 18, 19]. The last category consists in generating an entire solution all at once by simultaneously assigning every areal unit to a zone [20].

Different optimization strategies have been used ranging from hill climbing procedures [4] to simulated annealing [2, 8], tabu search [8] and linear programming associated with branch-and-bound [5]. Nevertheless, genetic algorithms remain largely unexplored in this field. The only reference we found to their use appears in [21], that provides no details on how GA were applied to this particular problem. However, GA have been used extensively as search procedures in related fields such as the P-Median Problem [22] and Cluster Analysis [23]. Other fields where complex optimization problems need to be dealt with, such as Pattern Recognition, Image Processing and Machine Learning [24], [25] and [26] have also benefited from the use of genetic algorithms.

In this paper we will present a solution to the zone design problem based on genetic algorithms. Two different encoding schemes are proposed, together with two different optimization functions. These techniques were applied to a real world problem, and the results are compared with a well established algorithm and software program. In sect. 2 a detailed statement of the problem is presented, along with a description of the search space. In sect. 3, a brief overview of the inner workings of a GA is presented, and our encoding and parameterization is given. Section 4 presents the real world problem and the results of the application of our GA to it. Finally, Sect. 5 contains the final conclusions.

2 Problem statement

A number of optimization criteria can be used in zone design, depending on the specific task at hand. Even limiting the domain to electoral districting there is a wide range of criteria that can be assumed to be relevant [14, 27]. An important criteria in political districting (as is the case of the real world problem that we shall present), is to dismiss the suspicion of deliberate manipulation of the zones in order to achieve a particular political result, also known as gerrymandering.[1]

As such, clear optimization criteria must be specified. It is generally accepted that there are three essential characteristics that districts should have [5]:

Population equality;
contiguity;
geographical compactness.
Therefore, our goal is to provide a method able of producing equally populated, contiguous and compact zones.

Achieving equal population size zones is central in any political districting problem. A measure of this goal can be obtained by calculating the sum of the differences between the population of each zone and the average population of all the zones. Thus, the simplest objective function that can be used in the electoral districting problem is:

$$\min \sum_{j} |P_j - \mu| \tag{4}$$

where P_j represents the population of the j*th* zone and μ the average population per zone. However, sometimes the ''obsession'' with population equality will lead districting plans to extremes [28] where contiguity and compactness are completely overlooked, producing very thin and long zones, that are intrinsically unnatural. Therefore, compactness is usually an important factor in any political districting solution.

There are many ways of defining the ‘‘compactness’’ of a zone [4, 21], but the general idea is that each zone should be as ‘‘circular’’ as possible. One measure of compactness, defined here as ‘‘radial compactness’’ is the sum for all zones, of the sum of Euclidean distances between the centroids of it's areal units and the center of that zone:

$$\textit{Radial Compactness} = \sum_{j} \sum_{i=\in Z_j} d_{ij} \tag{5}$$

where d_{ij} represents the Euclidean distance between the i*th* areal unit and the j*th* zone center. One of the ways of including this compactness measure in the objective function, is to include it as a second additive term:

$$\min \sum_{j} \left(|P_j - \mu| + \sum_{i=\in Z_j} d_{ij} \right) \tag{6}$$

There is also the possibility of using the product of all Euclidean distances and population differences within each zone, giving raise to the following objective function:

$$\min \sum_{j} \left(|P_j - \mu|^{*} \sum_{i=\in Z_j} d_{ij} \right) \tag{7}$$

Another compactness measure, defined here as ‘‘circumferential compactness’’ is the sum of the ratios between the square of the perimeter of each zone and it's area:

$$\sum_{i} \frac{pr_i^2}{a_i} \tag{8}$$

[1] The term is named for Elbridge Gerry who was Massachusetts governor and in 1812 and with the help of his political party crafted a district for his own election. At the time someone produced an illustration of the districting emphasizing its similarities with a salamander. The term, coined at the time, comes from putting together Gerry and mander.

where pr_i is the perimeter of zone i, and a_i is the area of that zone.[2] This measure should be minimized to achieve the desired result.

The approach used to verify that the contiguity restriction is respected involves the use of a binary contiguity which depicts the topological structure of the areal units. A method to verify contiguity using these matrices is described in [2]. This verification constitutes an intermediate step in the algorithm.

When solving the zone design problem, there is always the possibility of treating different objectives as hard constraints or make them part of the optimization function. Here we treat compactness and population equality as optimization objectives and contiguity as a quasi-hard constraint. This is done by strongly penalizing non-contiguous solutions, which in practice excludes them from final solutions.

The basic idea of this work is to provide the decision makers, in this case politicians, with good viable plans (solutions to the problem). These plans should not be affected by political criteria and should be seen as options which can be considered before a final decision is made. Nevertheless, additional criteria can easily be incorporated in our approach.

3 Problem complexity

One of the reasons why the zone design problem is especially difficult is due to the size of the solution space. The dimension of a ''*usual*'' real world problem makes unfeasible any attempt to explicitly enumerate all the possible solutions [29–31]. The calculation of the total number of possible solutions for a zone design problem is similar to the clustering problem and is given by the Stirling number of the second kind [32], [33], [34]:

$$S(n,k) = \frac{1}{k!}\sum_{i=1}^{k}(-1)^i\left(\frac{k}{(k-i)!i!}\right)(k-i)^n \qquad (9)$$

This means that a medium size problem like the one addressed in the results section of this paper, S (53, 7) yields $1.22*10^{41}$ possible solutions. [34] provides a formula that accounts for a basic connectivity structure, where the building blocks are connected in a chain. It is clear that this is a very special case of connectivity, very rarely found in real world problems, in which each building block is connected to only two neighbors. The dimension of the combinatorial problem of districting under chain connectivity conditions is given by:

$$S = \frac{(n-1)!}{(k-1)!(n-k)!} \qquad (10)$$

Additionally, in terms of computational complexity the zone design problem has been shown to be NP-Complete [29]. Thus, heuristic techniques seem to be the best road available to produce solutions to the problem in reasonable computational time. This is certainly a compromise but guaranteed optimality, independently of the problem's dimension, seems at this stage simply too difficult.

4 Genetic algorithms

Genetic algorithms (GA) are a subset of a broader and rapidly expanding area known as Evolutionary Computing [35]. As the name indicates, these algorithms drew inspiration on Darwin's theory of evolution, and have been used to solve hard optimization and machine learning problems [36]. The basic idea is that each solution to the problem is coded as a bit string, taken to be a chromosome, possibly with a number of sub-strings that act as genes. At any given point in time (or generation), a number of such chromosomes are kept, each representing a different ''individual'' or solution to the problem. Natural selection is simulated by evaluating the fitness (or goodness) of each solution, measured by how well it solves the problem at hand, and giving the best individuals a higher probability of remaining in the solution pool during the next generation. To obtain new solutions, two operators are used: crossover, and mutation. Crossover is implemented by combining bits of two different chromosomes (possibly divided along genes), to form a new solution. Mutation is implemented by randomly changing some bits or chromosomes. The details of how this basic idea is implemented may vary considerably.

Given enough time, a conveniently coded GA will always find an optimal solution. However, to obtain reasonable solutions in reasonable time, care must be taken in the encoding of the problem into chromosomes, and in the choice of the fitness function that will be optimized.

4.1 Encoding and parameters used

In order to be able to solve the zone design problem using a genetic algorithm it is fundamental to encode the partition in such a way that genetic operators may be used. There are a number of different ways that can be used to encode a solution to the zone design problem. Two different encodings are used in this paper.

The first encoding forces each zone to be centered at a point which represents a centroid of an areal unit. In

[2] The reason for choosing this measure in particular is related with our real world problem. In fact this was the measure proposed by consultants of the Portuguese government to evaluate the compactness of the different solutions produced. We do not explicitly use it as an optimization criterion, because it was provided at a late stage of the development of the project and because its inclusion would require extensive modification to the software. It is however used as a final performance measure

other words the process of finding the appropriate configuration consists of searching which set of areal unit centroids should represent the centers of the zones. A similar encoding was used by [22] to solve the P-Median problem. All areal units centroids are numbered and a solution is encoded as a string of length k (where k is the number of zones) where the ith element of the string denotes the position of the center of the ith zone. The second encoding enables the center of each zone to be placed anywhere in the study region. Centers do not have to match a centroid of an areal unit. In this case the encoding represents the pairs of x,y coordinates that define the positioning of each zone center. In both encodings each string represents a possible plan configuration and functions 4, 6 and 7 can be used to compute the fitness function of each string.

The genetic algorithm is randomly initialized with a population of size p. For the first encoding described above, p strings of length k are generated and the value that each position can assume is between 1 and n. In the second encoding p strings of length $2*k$ are initialized. For this encoding, initialization is done forcing all elements of the strings to be located within the study region. There are no guidelines for choosing the size of the initial population. In both cases 10 parallel populations were used, each one with 25 strings. Migration of strings between populations can occur with a probability of 0.001. Identical strings are not allowed, so there are no twins in the population.

At this point, an optimization function is needed in order to evaluate the p solutions. Functions 6 and 7 were used, because they combine both compactness and population equality, as optimization objectives. The type of selection used is tournament [36], and crossover probability is 0.95, uniform crossover was used [36] after some tests with other types of crossover. Mutation rate was 0.001, and an elitist strategy was used, assuring that the best individual of the population would always be carried to the next generation. Finally, the stopping criterion consisted of ending the algorithm after 5000 generations without improvements.

Thus, the algorithm procedure is as follows:

1. Generate p sets of k points, according to the selected encoding.
2. For each of the n areal unit centroids find the closest zone center, and assign the unit to the zone string.
3. Evaluate the fitness of each string, based on the chosen function and contiguity check.
4. Apply selection, crossover and mutation operators, creating a new population;
5. Return to step 2 until the stopping criterion is met.

5 Results

This algorithm was applied to a problem that was posed by the Portuguese Government. The discussion of the merits and shortcomings of the proportional system, used in Portugal since the 1974 revolution, has grown in recent years. The necessity of closing the gap between electors and elected has been present in the political agenda of the two main parties in Portugal.

Back in 1998 the Portuguese Government commissioned ISEGI-UNL and 3 other teams to divide the country into 93 electoral districts where the electors would choose only one candidate. Portugal is divided into 18 major regions and each of this should be divided into electoral districts. The number of districts in each of these regions depends on it's population size.

The proposed law itself imposed the first set of rules:

The districts zones must be contiguous.
If possible the unity of the Concelho (NUTS IV[3]) must be preserved.
Nevertheless in certain cases it was necessary to use *Freguesias* (NUTS V) in order to preserve the next rule.
The number of voters in each electoral district must be contained within an upper and lower limit of 25% of the mean ($0.75 > \times > 1.25$) of the total electors.

Although the problem was solved for the whole country in this paper, for the sake of simplicity, we will present only the results regarding the region of Lisbon. In Figs. 1 and 2, the areal units are shown together with their population. As can be seen there are great asymmetries amongst the population and size of the different units, increasing the difficulty of finding good solutions.

We compare the results produced using ZDES software, from the School of Geography from University of Leeds [37], and results produced by the GA proposed here. ZDES uses a simulated annealing algorithm based on the second approach described in Sect. 1 (it starts with a random solution to the problem as seed, and then incrementally modifies it in order to improve it). ZDES uses Eq. 4 as optimization criterion subject to the contiguity restriction. We choose ZDES for our comparisons because it is a well established and well known tool for this problem and also because it is available as freeware. Moreover this was the software used to obtain the solutions that were presented to the politicians in 1998.

Our GA was implemented in C programming language. Both encodings of the proposed GA were tested using Eq. 7 and additionally the second encoding (encoding 2) was tested with Eq. 6. A summary of the numerical results is presented in Table 1. The maps produced by each method can be evaluated in Figs. 3, 4, 5 and 6.

The result obtained with ZDES for the Lisbon area (Fig. 3) shows a total sum of differences between the population of each zone and the average population of

[3] NUTS is the nomenclature of statistical territorial units, and constitutes a five-level hierarchical classification, defined by Eurostat and used in all EC countries as statistical reporting units.

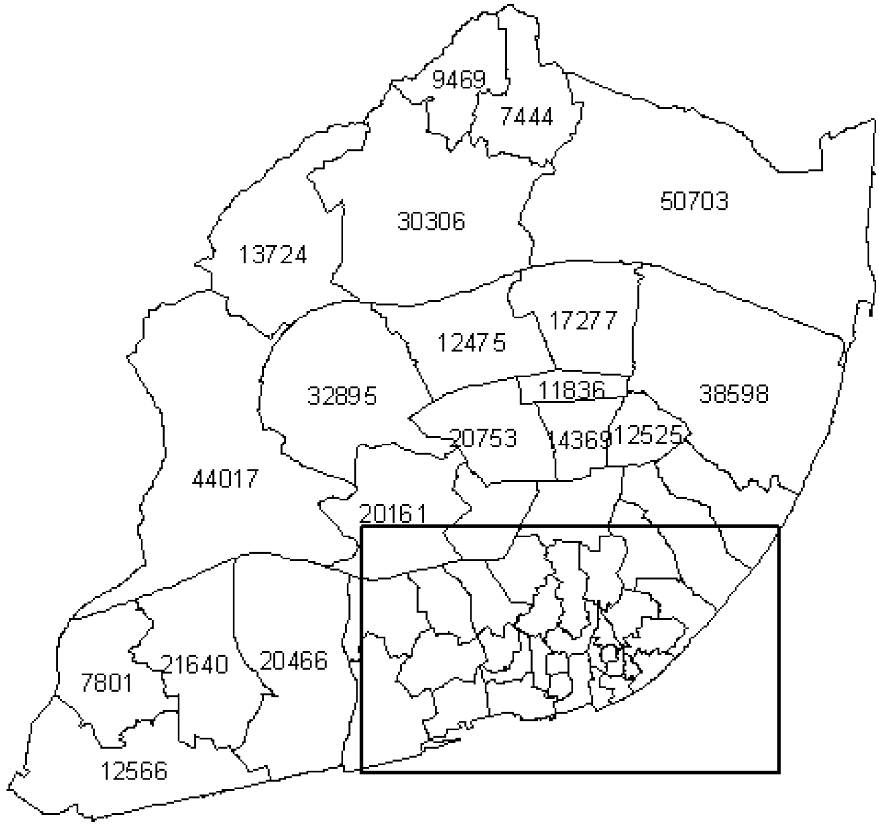

Fig. 1 Areal units of Lisbon and their population

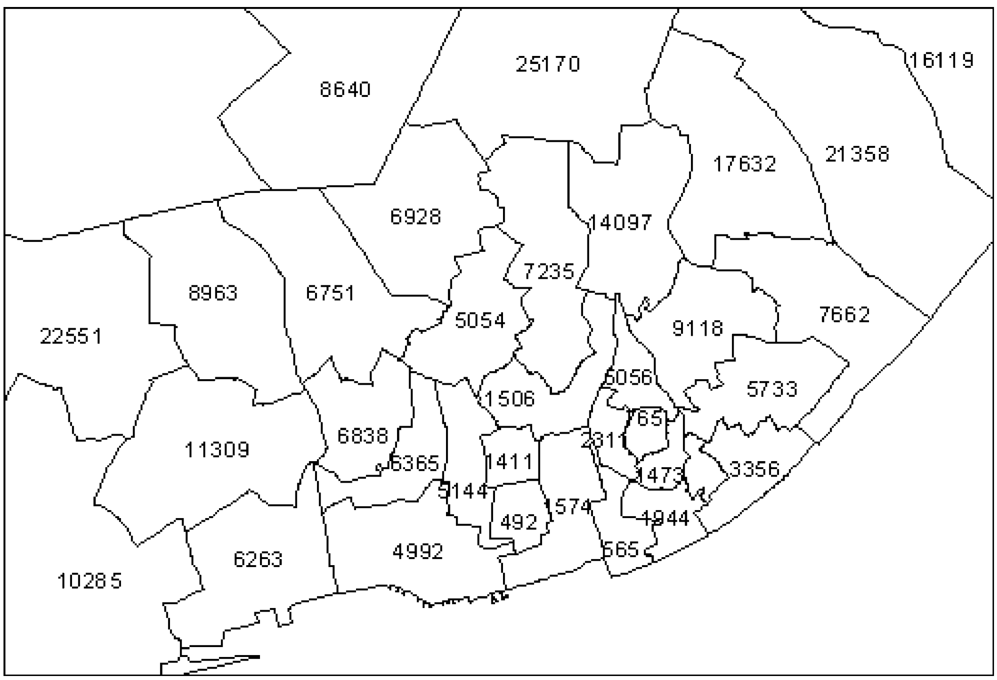

Fig. 2 Detail of the smaller areas of Lisbon and their populations

9327.43. A critical review of the map shows that the zones are quite awkward since they lack the compactness that was described in Sect. 2. If compactness is computed according to Eq. 8 one obtains the value of 315.17, as shown in Table 1. It thus becomes apparent that some measure of compactness must explicitly be considered during the optimization process.

The results produced using ZDES which were discussed with political representatives attracted major criticism due to the "exotic" design of some districts.

Table 1 Comparison of the results obtained

Zone design method	Compactness (equation 8)	Sum of population differences
ZDES	315,17	9327,43
Encoding 1 using equation 7	225,28	13665,43
Encoding 2 using equation 7	270,37	6069,43
Encoding 2 using equation 6	261,47	6220,57

From our experience, compactness tends to be especially important to politicians and people related to non-quantitative scientific areas like geography and law. Bearing in mind that the objective of political districting is to define areas which will have the same political representative, it is arguable that compactness is a desirable attribute.

To improve the solution obtained by ZDES and incorporate this new optimization criterion we applied the different GA's described in Sect. 3.1.. Using the first encoding (zone centers coincide with areal unit centroids) and Eq. 7 as objective function we obtained for Lisbon the zones presented in Fig. 5. Compactness measured by Eq. 8 is greatly improved as can be seen in Table 1. Unfortunately the sum of differences in population increases significantly. This increase is due to the strong restriction in the positioning of the zone centers. Using the second encoding this problem is overcome, as can be seen in Table 1.

The results obtained with encoding 2 and Eqs. 7 and 6 respectively are presented in Figs. 4 and 6. As can be observed in Table 1, encoding 2 produces important improvements in terms of population equality, when compared with encoding 1, dropping from 13665.43

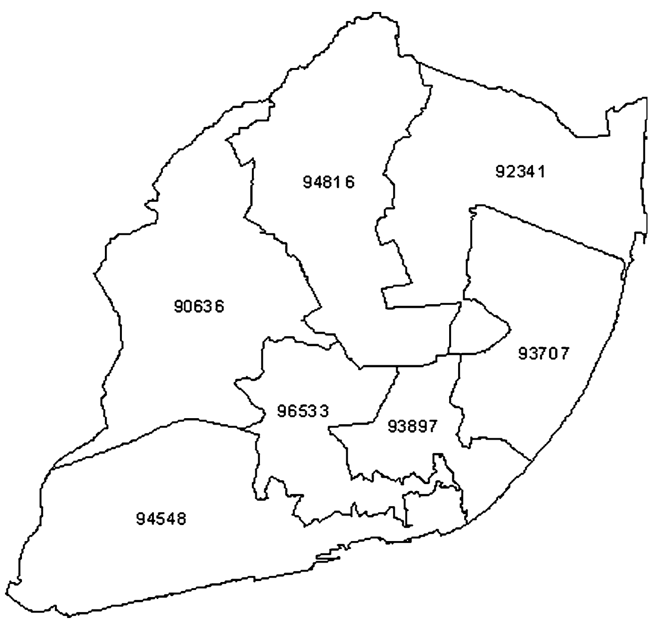

Fig. 3 Zones obtained with ZDES

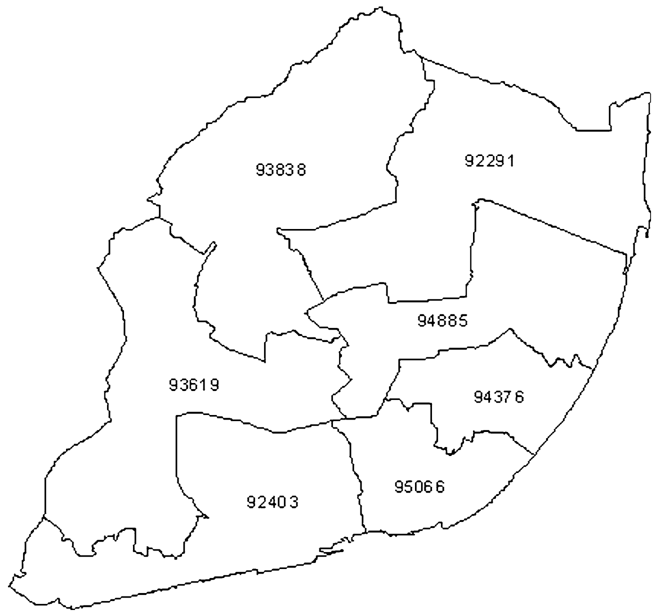

Fig. 4 Zones created by the Genetic Algorithm using encoding 2 and equation 7 as fitness function

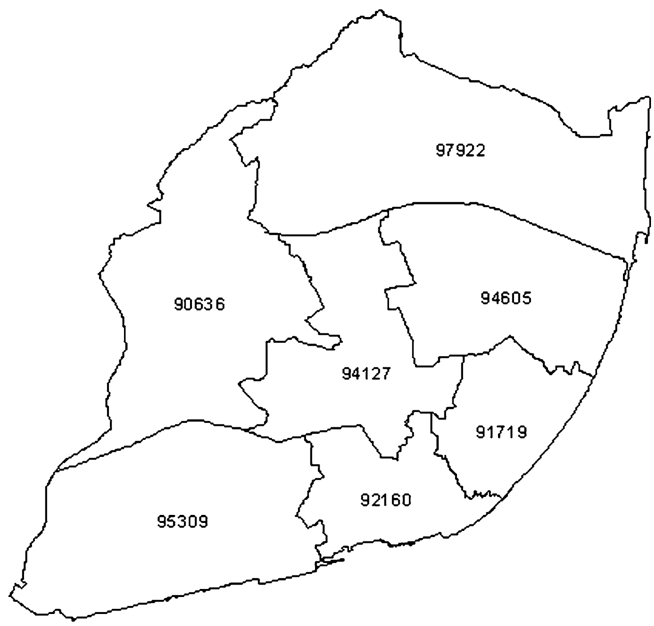

Fig. 5 Zones created by the Genetic Algorithm using encoding 1 and equation 7 as fitness function

(encoding 1) to 6069.43 and 6220.57 (encoding 2 Eqs. 7 and 6 respectively). This result should be expected because the second encoding allows the evaluation of a larger number of solutions. It would also be expected that the values of compactness would degrade somewhat, as happens. But the most interesting fact raised by these results is that with the second encoding both objective functions used perform better than ZDES. The compactness criterion is improved, as would be

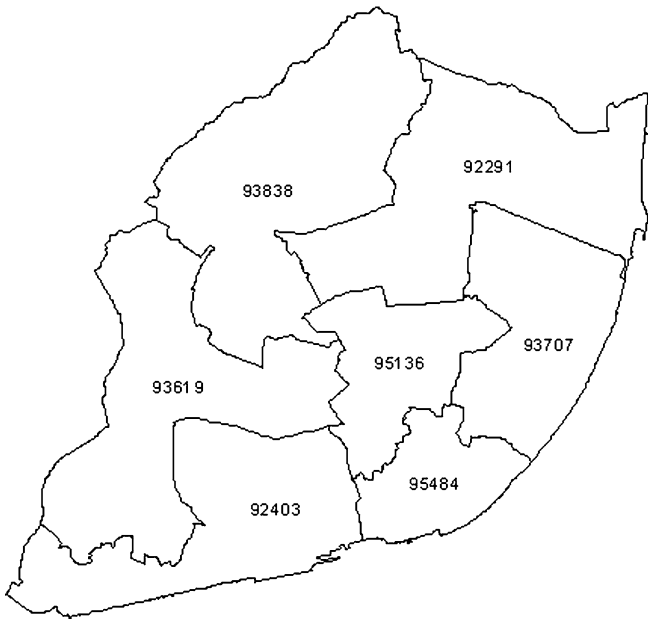

Fig. 6 Zones created by the Genetic Algorithm using encoding 2 and equation 6 as fitness function

expected, but the sum of differences in population (the objective function of ZDES) is also lowered. This latter result is very encouraging, giving a clear indication that GA can constitute a research direction with potential to improve solutions of the zone design problem. The differences that can be observed between Eqs. 7 and 6 are relatively small and seem to depict the trade-off between compactness and population equality. The results obtained when applying Eq 7 are slightly better in terms of population but less compact. Nevertheless differences are very small, and it is difficult to draw conclusions.

The stopping criterion used has an important effect in the number of strings evaluated for each of the two encoding schemes. In the case of the first encoding, generally, after 2000 generations no improvements are made, which means that approximately 7000 generations are evaluated (1750000 strings). For the first encoding the number of possible solutions is the same as in a location-allocation[4] problem. In our real world problem the total search space is 154143080, meaning that the genetic algorithm explores approximately 1.1% of the search space. In the second encoding the number of possible solutions is not known. Nevertheless using Eq. 9, which constitutes the upper bound, the search space has $1.22*10^{41}$ solutions. Typically, after 20000 generations improvements stop, meaning that only a very small fraction of the search space (6250000 strings) is explored, corresponding to $5*10^{-33}$% of that space.

[4] The number of solutions of the location-allocation problem is given by $\frac{n}{k!(n-k)!}$

6 Conclusions

A formalization of the electoral districting problem, suitable for processing by Genetic Algorithms was given. It was shown that GA provides solutions that can be considered better than those of other heuristic approaches, for this specific problem. It is also shown that choices of encoding and optimization functions have a major impact on outcomes, as would be expected. Allowing zone centers to be located anywhere within the study region (what was here called encoding 2), while not common in the literature, has enabled the GA to find better solutions. The application of GA to electoral districting needs further experimental work, especially concerning the use of different objective functions. The possibility of easily parallelizing these algorithms opens a relevant alternative that can be further explored. Parallelization is particularly adequate to complex combinatory problems, which is the case of zone design. Finally, the possibility of introducing heuristic procedures within the GA, in order to increase efficiency, has shown valuable results in other fields and can bring improvements here too.

References

1. Openshaw S (1996) Developing GIS-relevant zone-based spatial analysis methods. In: Longley P, Batty M, (eds) Spatial Analysis: Modelling in a GIS environment, GeoInformation International, Cambridge, pp. 55–73
2. Macmillan B, Pierce T (1994) Optimization Modelling in a GIS Framework: The Problem of Political Redistricting. In: Fotheringham S, Rogerson P (eds) Spatial analysis and GIS. Taylor & Francis Inc, Bristol, pp. 221–246
3. George JA, Lamar BW, Wallace CA (1997) Political District Determination Using Large-Scale Network Optimization. Socio-Economic Plan Sci 31 (1):11–28
4. Horn MET, (1995) Solution techniques for large regional partitioning problems. Geographical Anal, 27, pp. 230–248
5. Mehrotra A, Johnson EL, Nemhauser GL (1998) An optimization based heuristic for political districting. Manage Sci, 44, pp. 1100–1114
6. Martin D, (2000) Automated zone design in GIS. In: Atkinson P, Martin D (eds) GIS and Geocomputation, Innovations in GIS 7: Taylor and Francis, London, pp. 103–113
7. Ferland JA, Guénette G (1990) Decision support system for a school districting problem. Operations Res. 38 15–21
8. Openshaw S, Rao L (1995) Algorithms for reengineering 1991 Census Geography. Environ Plan A 27 425–446
9. Openshaw S Alvanides S, (1999) Applying geocomputation to the analysis of spatial distributions, In: Longley PA, Goodchild MF, Maguire DJ, Rhind DW (eds) Geographical Information Systems Principles, Techniques, Management and Applications, Wiley, Chichester, pp. 267–282
10. Haining R, Wise S, Blake M (1994) Constructing regions for small-area analysis-material deprivation and colorectal cancer. J. Public Health Med, 16, pp. 457–469
11. Fleischmann BJ, Paraschis N (1988) Solving a large scale districting problem: a case report. Comput. Operations Res 15:521–533
12. Martin D (1997) From enumeration districts to output areas: experiments in the automated creation of a census output geography. Popul Trends 88 pp. 36–42
13. Martin D (1998) 2001 Census output areas: from concept to prototype. Population Trends, 94 pp. 19–24

14. Williams JC (1995) Political Redistricting - A Review. Pap Reg Sci 74(1):13–39
15. Vickrey W (1961) On The Prevention of Gerrymandering. Political Sci Q 76 pp. 105–110
16. Harris CC (1964) A scientific method of districting. Behav Sci 9 pp. 219–225
17. Thoreson JD, Liittschwager JM (1967) Legislative districting by computer simulation. Behav Sci 12 pp. 237–247
18. Garfinkel RS, Nemhauser GL (1970) Optimal political districting by implicit enumeration techniques. Manage Sci 16 pp. 495–508
19. Nagel SS, (1972) Computers and the law and politics of redistricting. Polity 5: pp. 77–93
20. Weaver JB, Hess SW, (1963) A procedure for nonpartisan districting: development of computer techniques. Yale Law J 72 pp. 288–308
21. Altman M (1998) Modeling the Effect of Mandatory District Compactness on Partisan Gerrymanders. Political Geography 17 (8): 989–1012
22. Correa ES, Steiner MTA, Freitas AA, Carnieri C (2001) A genetic algorithm for the P-median problem. Proc. 2001 Genetic and Evolutionary Computation Conf. (GECCO-2001), pp. 1268–1275. Morgan Kaufmann
23. Murthy CA, Chowdhury N (1996) In search of optimal clusters using genetic algorithms. Pattern Recognit Lett, vol. 17, pp. 825–832
24. Ankenbrandt CA, Buckles BP, Petry FE, (1990) Scene recognition using genetic algorithms with semantic nets. Pattern Recognit Lett 11(4), pp. 285–293
25. Belew RK, Booker LB (eds) (1991) Proceedings of the Fourth International Conference on Genetic Algorithms. La Jolla, CA: Morgan Kaufmann
26. Back T, Fogel DB, Michalewicz Z (eds) (1997) Handbook of Evolutionary Computation. Oxford University Press, NewYork
27. Morril RL (1981) Political redistricting and geographic theory. Association of American Geographers, Seattle
28. Hayes B (1996) Computing Science: Machine Politics. American Scientist, vol. 84, No. 6, November–December, pp. 522–526
29. Altman M, (1997) The Computational Complexity of Automated Redistricting: Is Automation the Answer?. Rutgers Comput and Technol Law J 23 (1):81–142
30. Rossiter DJ, Johnston RJ (1981) Program GROUP: the identification of all possible solutions to a constituency-delimitation problem. Environ Plan A 13:231–8
31. Cliff AD, Hagget P (1970) On the efficiency of alternative aggregations in region-building problems. Environ Plan 2, pp. 285–294
32. Anderberg MR (1973) Cluster Analysis for Applications. Academic Press
33. Vilares M, Nazareth M, Painho M, Bacao F, Coelho P (1998) Gis And The Design Of Electoral Districts. in Gisplanet 98, International Conference And Exhibition On Geographic Information, Lisboa 7–11 September 1998
34. Keane M (1975) The size of the region-building problem. Environ Plan A 7, pp. 575–577
35. Fogel DB (2000) Evolutionary Computation: Toward a New Philosophy of Machine Intelligence. 2nd edn, IEEE Press, Piscataway, NJ
36. Goldberg DE, (1989) Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, Mass
37. University of Leeds, (2003) ZDES, http://www.geog.leeds.ac.uk/pgrads/s.alvanides/zdes3.html, (June 2003)