European Journal of Operational Research 144 (2003) 12–26

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/dsw

Discrete Optimization

A tabu search heuristic and adaptive memory procedure for political districting

Burcin Bozkaya [a], Erhan Erkut [b], Gilbert Laporte [c,*]

[a] *Transportation/Logistics Services, Environmental Systems Research Institute, 380 New York Street, Redlands, CA 92373, USA*
[b] *School of Business, University of Alberta, Edmonton, Alta., Canada T6G 2R6*
[c] *Canada Research Chair in Distribution Management, HEC-Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montreal, Canada H3T 2A7*

Received 6 May 2000; accepted 17 October 2001

Abstract

In political districting problems, the aim is to partition a territory into electoral constituencies, subject to some side constraints. The most common side constraints include contiguity, population equality, compactness, and socio-economic homogeneity. We propose a formulation in which the various constraints are integrated into a single multicriteria function. We solve the problem by means of a tabu search and adaptive memory heuristic. The procedure is illustrated on real data from the city of Edmonton.

Keywords: Political districting; Multicriteria optimization; Tabu search heuristic

1. Introduction

In districting problems, the aim is to partition a territory into *districts*, subject to some side constraints. Typical districting problems include the drawing of political constituencies, school board boundaries, sales or delivery regions. Here we focus our attention on political districting. This problem is particularly important in democracies where each district elects a single member to a parliamentary assembly. One important issue at stake is *equity* (or *population equality*), i.e., all districts should have approximately the same number of voters in order to respect the "one-man, one-vote" principle. Furthermore, political districts must not be seen as favoring a particular political party. A famous case arose in Massachusetts in the early 19th century when the state legislature proposed a salamander-shaped district in order to gain electoral advantage. The governor of the state at that time was Elbridge Gerry and this practice became known as *gerrymandering*. Interesting accounts of gerrymandering cases are provided by Cain (1984) and Lewyn (1993). For a recent book on political districting the reader is referred to Grilli di Cortona et al. (1999).

To prevent political interference in the districting process, several states have set up a neutral

* Corresponding author. Present address. Centre de recherche sur les transports, Université de Montréal, CP 6128, Succursale Centre-ville, Montréal, Que., Canada H3C 3J7. Tel.: +1-514-343-7575; fax: +1-514-343-7121.

E-mail address: gilbert@crt.umontreal.ca (G. Laporte).

0377-2217/03/$ - see front matter
PII: S0377-2217(01)00380-0

commission whose functions include the drawing up of political boundaries satisfying a number of legislative and common sense criteria. Some criteria define the feasibility of a solution. For example, several legislatures impose that districts should be contiguous and that their (voting) population should fall within some interval. Other criteria help to assess the quality of a solution. An important criterion is that districts should be compact, the idea being to prevent the formation of odd-shaped districts that raise suspicions of gerrymandering. Other commonly used criteria are:

- the respect of natural boundaries, such as major bodies of water;
- the respect of some existing administrative or political subdivisions, like census tracts, townships;
- socio-economic homogeneity to ensure a better representation of residents who share common concerns or views;
- similarity to the existing plan so that an incumbent runs again in a similar district;
- the respect of integrity of communities, i.e., avoiding splitting some communities between several districts;
- equal probability of representation to ensure that some important minority groups have their fair share of representatives.

Some of these criteria can be disputed. For example, it may be argued that it is just as desirable to achieve socio-economic heterogeneity as it is important to reach homogeneity. Also the last criterion, and the reverse consisting of diluting the strength of any particular group, is itself a form of gerrymandering and should be handled with care. Similarly, any criterion aimed at providing a fair representation of political parties or at protecting safe seats may also be perceived as suspicious. For this reason, it may be argued that political data should not be used when designing districts. In the same spirit, more confidence is likely to be put in the process if it is computer-based, but even then some human intervention is required for the selection of criteria and for the determination of their relative weights.

The scientific literature on districting shows that there is no consensus on which criteria are legitimate and on how these should be measured (Williams, 1995). In this study, we will consider some of the most commonly accepted criteria: respect of major natural boundaries, contiguity, population equality, compactness, socio-economic homogeneity, similarity with the existing plan, and integrity of communities. This list should not, however, be viewed as limitative since our method can in principle work with any number of criteria as long as these can be measured. We propose a model that assigns weights to the various criteria, and a flexible solution methodology capable of producing high quality districting plans with respect to a set of weights. By altering weights or making several runs of the algorithm, decision makers should be able to generate a variety of solutions that will appeal to diverse interests.

Since the early 1960s, several heuristics have been proposed for the districting problem. All attempt to combine indivisible basic units such as census tracts or enumeration areas into feasible districts. These methods are based on one of the two integer linear programming formulations of the problem.

The first mathematical programming approach was proposed by Hess et al. (1965). It formulates the problem as an assignment problem with side constraints. Let I be the set of all basic units, and let J be the set of basic units used as potential district "seeds". The cost c_{ij} of assigning unit i to seed j is a function of the Euclidean distance between the center of j and the center of i (typically c_{ij} is the square of that distance). The number of districts to be created is given and equal to m. The population of unit i is equal to p_i and the population of any district must lie within an interval $[a, b]$. Let x_{ij} be a binary variable equal to 1 if and only if unit i is assigned to seed j. The problem is then modeled as a capacitated m-median problem as follows:

$$\text{(F1)} \quad \text{minimize} \quad \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}$$

$$\text{subject to} \quad \sum_{j \in J} x_{ij} = 1 \quad (i \in I), \tag{1}$$

$$\sum_{j \in J} x_{jj} = m, \tag{2}$$

$$x_{ij} \leqslant x_{jj} \quad (i \in I, j \in J), \tag{3}$$

$$a \leqslant \sum_{i \in I} p_i x_{ij} \leqslant b \quad (j \in J), \tag{4}$$

$$x_{ij} = 0 \text{ or } 1 \quad (i \in I, j \in J). \tag{5}$$

In this formulation, the objective function measures compactness, while population equity is taken into account by constraints (4). Constraints (1) ensure that each basic unit is assigned to one district, and the number of districts is equal to m by constraint (2). By constraints (3), no basic unit can be assigned to an unselected seed. Note that these constraints are left out of the Hess et al. model. There is no guarantee that this formulation will produce contiguous districts, although this will be favored by the objective function. Thus solving (F1) to optimality does not in general yield a suitable solution to the districting problem, but it can produce an embryonic infeasible solution that can then be patched through local search.

This formulation can also serve as a guide for a heuristic. Given a reasonable set of m seed units, the remaining units can be iteratively assigned to these in a greedy fashion while ensuring that constraints (4) are satisfied. Again, it may be impossible to achieve contiguity. Since this type of approach does not require a linear objective, it is relatively easy to incorporate several terms into the objective, including constraints (4), each corresponding to a criterion to be satisfied. This is essentially the approach taken by most authors in the field, with various degrees of sophistication and a large number of variants (see, e.g., Vickrey, 1961; Weaver and Hess, 1963; Hess et al., 1965; Kaiser, 1966; Nagel, 1965, 1972; Thoreson and Liittschwager, 1967; Morrill, 1973, 1976; Bourjolly et al., 1981; Plane, 1982; Fleischmann and Paraschis, 1988; Browdy, 1990; Macmillan and Pierce, 1992). Several of these methods include a local search post-optimization phase consisting of moving basic units to adjacent districts, or swapping units between adjacent districts. The most sophisticated of these schemes, by Browdy (1990) and Macmillan and Pierce (1992), is based on simulated annealing. Local search is not only useful for improving the objective function, but it may also help the process move from an infeasible to a feasible solution.

In the second mathematical programming formulation, I is again the set of basic units and J is the set of all feasible districts. A binary coefficient a_{ij} is equal to 1 if and only if unit i belongs to district j. A cost c_j is assigned to district j and the number of districts is again equal to m. Binary variable x_j takes the value 1 if and only if district j is selected. The formulation, first proposed by Garfinkel and Nemhauser (1970) is

$$\text{(F2)} \quad \text{minimize} \quad \sum_{j \in J} c_j x_j$$

$$\text{subject to} \quad \sum_{j \in J} a_{ij} x_j = 1 \quad (i \in I), \tag{6}$$

$$\sum_{j \in J} x_j = m, \tag{7}$$

$$x_j = 0 \text{ or } 1 \quad (j \in J). \tag{8}$$

Since the number of potential districts is astronomical in most situations, Garfinkel and Nemhauser (1970) propose obtaining a good solution to (F2) by means of a truncated branch-and-bound tree. Recently, Mehrotra et al. (1998) have developed a column generation algorithm for this model. Their method remains a heuristic since the subproblem in which new columns are generated is NP-hard and is not solved optimally.

The purpose of this paper is to develop a tabu search (TS) algorithm for the political districting problem, an approach that has been highly successful for the solution of a host of combinatorial optimization problems (see, e.g., Glover and Laguna, 1997). Tabu search is advantageous in that it does not require a sophisticated integer linear programming apparatus, which makes its adoption by end-users easier. In addition to the standard TS methodology, we embed our algorithm within an adaptive memory procedure by Rochat and Taillard (1995).

The rest of this paper is organized as follows. In Section 2 the various criteria constituting the objective function of the model are formalized and cast in mathematical terms. The TS algorithm itself is described in Section 3. This is followed by computational results in Section 4, and by the conclusions in Section 5.

2. Political districting criteria

The political districting problem falls in the class of multicriteria optimization for which several approaches are possible (see, e.g., Buchanan and Daellenbach, 1987; Roy, 1985; Vincke, 1992; Yu, 1989). A common approach is to treat some criteria as hard constraints and others as soft requirements or as terms in an objective function (Eiselt and Laporte, 1987). In this study, we treat contiguity as a hard constraint and all other criteria through the minimization of a weighted additive multicriteria function $F(x) = \sum_r \alpha_r f_r(x)$, where α_r is a weight and $f_r(x)$ is the value of a function assigning a value for criterion r to any given solution x. Several methods have been proposed by various authors to define the measurements f_r. We now describe how this was done in our study.

2.1. Population equality

There are several way to define population equality. In this study, we proceed as follows. Denoting by J the set of all districts in solution x (feasible or not), and by $P_j(x)$ the population of district j, the average district population is equal to $\bar{P} = \sum_{j\in J} P_j(x)/m$. We require the population of each district to lie within some interval $[(1-\beta)\bar{P}, (1+\beta)\bar{P}]$, where $0 \leqslant \beta < 1$. We then define the population equality function as

$$f_{\text{pop}}(x) = \left(\sum_{j\in J} \max\left\{ P_j(x) - (1+\beta)\bar{P}, (1-\beta)\bar{P} - P_j(x), 0 \right\} \right) \Big/ \bar{P}. \tag{9}$$

This function will take a value equal to zero if each district population lies between $(1-\beta)\bar{P}$ and $(1+\beta)\bar{P}$. Otherwise it will take a positive value equal to the sum of infeasibilities.

2.2. Compactness

As observed by Young (1988) and Niemi et al. (1990), several measures can be used to assess the compactness of a district, none of which is perfect. Within a local search algorithm such as ours, it makes sense to use a measure that can be easily computed, but we do not require a linear measure as in a number of mathematical models. After some experimentation, we have opted for two possible compactness measures associated with a solution x. The first, $f_{\text{comp1}}(x)$, is based on the total length of the all boundary lengths between districts, excluding the outside boundary of the territory. The divisor R is used for scaling purposes. The second, $f_{\text{comp2}}(x)$, compares the perimeter of each district to that of a circle having the same area. Formally, we define in the first case

$$f_{\text{comp1}}(x) = \left(\sum_{j\in J} R_j(x) - R \right) \Big/ 2R, \tag{10}$$

where $R_j(x)$ is the perimeter of district j in the solution x, and R is the perimeter of the entire territory. In the second case, the compactness measure is defined as

$$f_{\text{comp2}}(x) = \sum_{j\in J} \left(1 - \frac{2\pi\sqrt{A_j(x)/\pi}}{R_j(x)} \right) \Big/ m. \tag{11}$$

2.3. Socio-economic homogeneity

Socio-economic homogeneity can be measured indirectly by personal income, as suggested by Bourjolly et al. (1981). A reasonable objective is to minimize the sum, over all districts j, of the standard deviation $S_j(x)$ of income. To obtain a dimensionless measure, this sum is divided by the average income $\bar{S}$ to yield

$$f_{\text{soc}}(x) = \sum_{j\in J} S_j(x)/\bar{S}. \tag{12}$$

The standard deviation is approximated by working with the average income of each basic unit in the district.

2.4. Similarity to the existing plan

We have developed a new measure in order to compare the similarity of a proposed solution x with the existing plan. It computes for each district j of the existing plan, the largest overlay $O_j(x)$ with a district contained in a new solution x. Formally, the similarity index is defined as

$$f_{\text{sim}}(x) = 1 - \sum_{j\in J} O_j(x)/A, \tag{13}$$

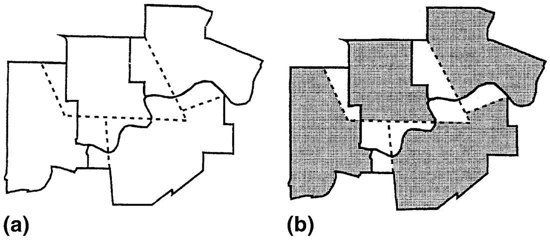

Fig. 1. Illustration of the similarity index: (a) old and new boundaries; (b) overlaying regions.

where A is the entire area. This index always takes a value between 0 and 1 and can be used even if the old and new plans do not contain the same number of districts.

To illustrate, consider the example depicted in Fig. 1. In Fig. 1(a), the existing district boundaries are shown in bold lines while the new district boundaries are represented by dashed lines. The largest overlaying areas correspond to the shaded regions of Fig. 1(b).

2.5. Integrity of communities

To measure the integrity of communities we again use an overlay concept, as for the similarity index. This can be done as long as databases allow a representation of communities as geographical areas. We minimize the function

$$f_{\text{int}}(x) = 1 - \sum_{j \in J} G_j(x) \Big/ \sum_{j \in J} P_j(x), \tag{14}$$

where $G_j(x)$ is the largest population of a given community in district j of solution x.

3. Tabu search and adaptive memory algorithm

We describe in this section the tabu search (TS) algorithm we have developed for the districting problem, and how this method was embedded within a broader search engine called the adaptive memory procedure (AMP). We start with the description of the main components of the TS heuristics, followed by that of the AMP.

Tabu search is an iterative optimization method now commonly used in combinatorial optimization. Since it was first proposed by Glover (1977), it has been applied with a striking degree of success to a host of problems (see, e.g., Osman and Laporte, 1996; Aarts and Lenstra, 1997; Glover and Laguna, 1997). Starting from an initial solution x_0, the method moves at each iteration t from a solution x_{t-1} to the best solution in its neighbourhood $N(x_{t-1})$, even if this causes a deterioration in the value of the objective function. To avoid cycling, some solutions possessing particular attributes are declared forbidden, or *tabu* for a given number of iterations. The search stops whenever a given stopping rule is satisfied. The method can be enhanced through the incorporation of several features, some of which exploit the characteristics of the application at hand. Here is how we have applied TS to the districting problem.

3.1. Objective function

The objective function minimized throughout the search is

$$\begin{aligned} F(x) = {} & \alpha_{\text{pop}} f_{\text{pop}}(x) + \alpha_{\text{comp}} f_{\text{comp}}(x) + \alpha_{\text{soc}} f_{\text{soc}}(x) \\ & + \alpha_{\text{sim}} f_{\text{sim}}(x) + \alpha_{\text{int}} f_{\text{int}}(x), \end{aligned} \tag{15}$$

where f_{comp} is either f_{comp1} or f_{comp2}, and α_{pop}; α_{comp}; α_{soc}; α_{sim}; α_{int} are multipliers. All these, except α_{pop}, are user-defined input values. The α_{pop} multiplier is initially set equal to 1 and allowed to vary during the search to account for the fact that any given solution x may be infeasible with respect to population equality. This will be explained in Section 3.7.

3.2. Preprocessing

Given a set I of basic units and attributes, it is straightforward to draw an adjacency list and to declare as non-adjacent any two units separated by a natural frontier. Also, two neighboring units that only have a finite number of points in common, as opposed to an edge, are considered to be non-adjacent. At this stage, any two units that must be part of the same district in the final solution are merged, and any basic unit enclaved within another unit is merged with its neighbor unless it has a large enough population to constitute a district by itself.

3.3. Initial solution

To create an initial solution, we proceed as in Vickrey (1961). We first select a seed unit to initialize a district. We then gradually extend this district by adjoining to it one of its adjacent units. The district is complete whenever its population attains $\bar{P}$ for the first time or when no adjacent units are available. If the number of districts created in this fashion is larger than m, we reduce it by iteratively merging the least populated unit with its least populated neighbor. If the number of districts is less than m, we gradually increase it by iteratively splitting the most populated district into two, while preserving contiguity. At the end of this process, the initial solution x_0 is made up of m contiguous districts, some of which may be infeasible with respect to population equality.

3.4. Neighborhoods

We use two neighborhoods $N_1(x)$ and $N_2(x)$. The first, $N_1(x)$, is made up of all solutions reachable from x by moving a basic unit i from its current district j to a neighbor district l without creating a non-contiguous solution. Such a move is said to be of Type I and denoted by (i, j, l) and is illustrated in Fig. 2. Here, moving unit i_1 from district j to district l would disconnect unit i_2 from the remainder of district j and such a move would not be allowed. The second neighborhood, $N_2(x)$, is made up of all solutions that can be reached from x by swapping two border units i and k between their respective districts j and l, again without creating discontiguities. Such a move is said to be of Type II and denoted by (i, k, j, l). The reason for using two neighborhoods is that Type I moves are not, as a rule, powerful enough to identify good quality solutions. Type II moves work much better but are computationally more expensive. Thus, in the algorithm, two passes are made. In the first pass, only Type I moves are used. In the second pass, both types of moves are applied.

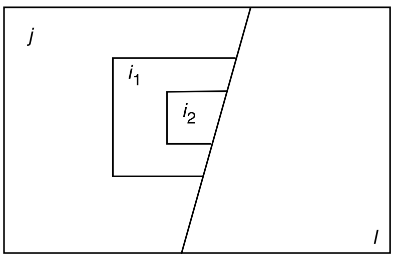

Fig. 2. Move creating a non-contiguous solution.

3.5. Recency-based memory and tabu tenure

To help prevent cycling, whenever a move (i, j, l) or (i, k, j, l) is performed, any move that puts unit i back into district j or unit k back into district l (e.g., $(i, l, j), (i, l', j), (i, k, l, j)$, etc.) is declared tabu for θ iterations, where θ is randomly selected in some interval $[\theta_{\min}, \theta_{\max}]$. The idea of using random tabu tenures was first proposed by Taillard (1991) in the context of the quadratic assignment problem and has since been used by several authors. As opposed to a fixed tabu tenure, it virtually removes the probability of cycling provided $\theta_{\min}$ and $\theta_{\max}$ are sufficiently large. However, using too large values may impair the search as most potential moves will soon became tabu. Some problem related calibration is therefore recommended. The only circumstance where the algorithm will perform a tabu move is when this would yield a better incumbent.

3.6. Frequency-based memory

A good way to diversify the search and to help it explore a wider part of the solution space is to add a penalty term to the objective function value corresponding to frequently performed moves. In practice, this is applied to non-improving moves since it makes little sense to penalize improving solutions. This term is the product of four factors: (1) the past frequency of the move, (2) the problem size measured by $\sqrt{m}$, (3) the largest change δ observed so far in the objective function from one iteration to the next, and (4) a scaling constant ρ.

The first factor is the driving force among the four. It has two components that keep track of two different attributes of a move: η_{it}, the average number of times unit i was moved over the previous $t - 1$ iterations, and ϑ_{jt}, the average number of times district j has been involved in a move in the previous $t-1$ iterations. In (i,k,j,l) moves, both districts j and l are involved twice. The frequency factor is defined as

$$\phi(i,j,l) = (1+\eta_{it})\left(1+\frac{\vartheta_{jt}+\vartheta_{lt}}{2}\right) - 1 \tag{16}$$

for Type I moves and as

$$\phi(i,k,j,l) = \left(1+\frac{\eta_{it}+\eta_{kt}}{2}\right)\left(1+\frac{\vartheta_{jt}+\vartheta_{lt}}{2}\right) - 1 \tag{17}$$

for Type II moves.

The second factor $\sqrt{m}$ in the penalty term is related to problem size. Choosing $\sqrt{m}$, rather than m for example, is common to several TS implementations (see, e.g., Taillard, 1991; Gendreau et al., 1994; Cordeau et al., 1997). The idea is to use a multiplier that reflects problem size. It was observed empirically that m overemphasizes problem size whereas $\sqrt{m}$ produces a smoother search. The third factor δ is used to scale the magnitude of the penalty term relative to the objective function. The last factor ρ allows an overall scaling of the penalty term and is user-controlled.

The four factors multiplied together are added to $F(x)$ to produce a penalized objective:

$$F'(x) = F(x) + \phi\delta\rho\sqrt{m} \tag{18}$$

with potential non-improving Type I or Type II moves.

3.7. Self-adjusting parameter α_{pop}

Parameter α_{pop} in the objective function is initially set equal to 1, and adjusted every μ iterations, where μ is a user-controlled multiplier. If all previous $\bar{\mu}$ solutions were infeasible, then $\alpha_{\text{pop}} := 2\alpha_{\text{pop}}$; if they were all feasible, then $\alpha_{\text{pop}} := \alpha_{\text{pop}}/2$; otherwise α_{pop} remains unchanged. The parameter $\bar{\mu}$ is also user-controlled and takes a value in the interval $[\mu/2, \mu]$. This way of operating, introduced by Gendreau et al. (1994) in the context of the vehicle routing problem, helps to produce a good mix of feasible and infeasible solutions with respect to the population equality criterion.

3.8. Stopping rules

The algorithm records F_1^*, the value of the best-known feasible solution, and F_2^* the value of the best-known (feasible or infeasible) solution. The search stops whenever no improvement in F_1^* or F_2^* has been observed for τ_1 consecutive iterations, or if a total of τ_2 iterations have been performed, where τ_2 is user-controlled and equal to 30,000 in our implementation.

3.9. Summary of the tabu search algorithm

The step by step description of the TS algorithm is now provided.

Step 1 (Initialization). Construct a starting solution x_0. Set $x^* := x_0$. Set $\alpha_{\text{pop}} := 1$; $F_2^* := F(x_0)$; $F_1^* := F(x_0)$ if x_0 is feasible, and $F_1^* := \infty$ otherwise. Set the iteration count $t := 0$. Set $\delta := 0$, and all η_{it} and ϑ_{jt} values equal to 0.

Step 2 (Neighborhood search, pass 1). Repeat this step as long as the stopping criterion has not been met:

- Set $t := t+1$.
- If $t = 0 \pmod{\mu}$, update α_{pop}.
- Identify all neighbors x_{ijl} of x_{t-1} with respect to Type I moves and compute $F(x_{ijl})$ in each case. Whenever $F(x_{ijl}) \geqslant F(x_{t-1})$, set $F'(x_{ijl}) := F(x_{ijl}) + \phi(i,j,l)\delta\rho\sqrt{m}$. Otherwise, set $F'(x_{ijl}) := F(x_{ijl})$. Sort all $F'(x_{ijl})$ values in non-decreasing order and implement the first non-tabu move, or the first tabu move that improves upon F_1^* or F_2^*.
- Set the appropriate moves tabu for θ iterations, where $\theta \in [\theta_{\min}, \theta_{\max}]$.
- Update $x^*, F_1^*, F_2^*, \delta$, all η_{it} and ϑ_{jt} values.

Step 3 (Neighborhood search, pass 2). This step is identical to Step 2, except that the neighbors of x_{t-1} now include all solutions x_{ijl} that can be reached through Type I moves, as well as all solutions x_{ikjl} that can be reached through Type II moves.

3.10. Adaptive memory procedure

As mentioned, the TS algorithm is embedded within an adaptive memory procedure (AMP), also referred to as "probabilistic diversification and intensification", which was first proposed by Rochat and Taillard (1995) in the context of the vehicle routing problem. The AMP is based on the idea that the components of high quality solutions can be used to construct other high quality solutions, similar to what is done in genetic algorithms. In the vehicle routing context, components of a solution are vehicle routes. In our problem, they are political districts. The method therefore stores in a constantly updated pool a set of districts belonging to some of the best-known solutions. Then, disjoint districts can be extracted from the pool to serve as a basis for a new solution. Each district of the pool, or adaptive memory, is given a larger probability of being selected if it belongs to a better solution. More specifically, if there are γ districts in the pool, ranked in non-decreasing order of the objective value F of the solution to which they belong, then the hth district from the top of the list is extracted with probability $2(\gamma - h + 1)/(\gamma^2 + \gamma)$. Districts are therefore extracted from the pool until it becomes impossible to extract more without creating overlaps. The selected districts do not as a rule constitute a full districting solution. The TS process just described is then applied to construct a full districting plan from a partial solution. Whenever a new solution is thus created, its districts are candidates to become members of the pool. The size of the pool is kept constant at γ, meaning that its worst elements are regularly replaced by better ones. The number of times TS is called within the AMP is equal to a user-controlled value τ_3.

4. Computational results

The algorithm just described was coded in C and run on a Pentium 233MMX PC with 64 MB RAM. Tests were conducted on the City of Edmonton, Canada. For the sake of brevity, we only report some of our experiments. Detailed test results are provided in Bozkaya (1999). We first discuss the data requirements. We then explain how the values of the various parameters of the algorithm were set. Finally, we illustrate some scenarios.

4.1. Data requirements

All necessary data were obtained through Statistics Canada's databases. Enumeration areas (EAs) were selected as basic units. For each unit, all population data and geographical information are available. All input and output information were handled and visualized using the ArcView Geographical Information System. An ArcView sample screen showing the city of Edmonton partitioned into basic units and political districts is displayed in Fig. 3.

The Edmonton data were extracted from the 1996 census results and the current districting plan was obtained from Government of Alberta (1996). The database is made up of 828 EAs from which 19 districts have to be created. The relevant population data used in our study includes population size, ethnic composition, and income level. Also, we used ArcView's scripting language Avenue to extract additional geographical information such as the adjacency of basic units, and the length of the border between any pair of basic units.

4.2. Parameter tuning

In any TS algorithm, it is necessary to properly calibrate the various parameters of the algorithm. Barr et al. (1995) suggest using statistical design. Accordingly, we conducted a study of the following parameters:

$[\theta_{\min}, \theta_{\max}]$	tabu tenure range
α	starting value for α_{pop}
μ	frequency of updating α
$\bar{\mu}$	maximum number of feasible/infeasible solutions needed to update α
ρ	scaling factor for the frequency based memory
τ_1	number of successive TS iterations without improvement
γ	size of adaptive memory
τ_3	number of AMP iterations

The setting of $\theta_{\min}$ and $\theta_{\max}$ was done indirectly by using two auxiliary parameters $\bar{\theta} = (\theta_{\min} + \theta_{\max})/2$ and $\theta' = \theta_{\max} - \theta_{\min}$. All parameters

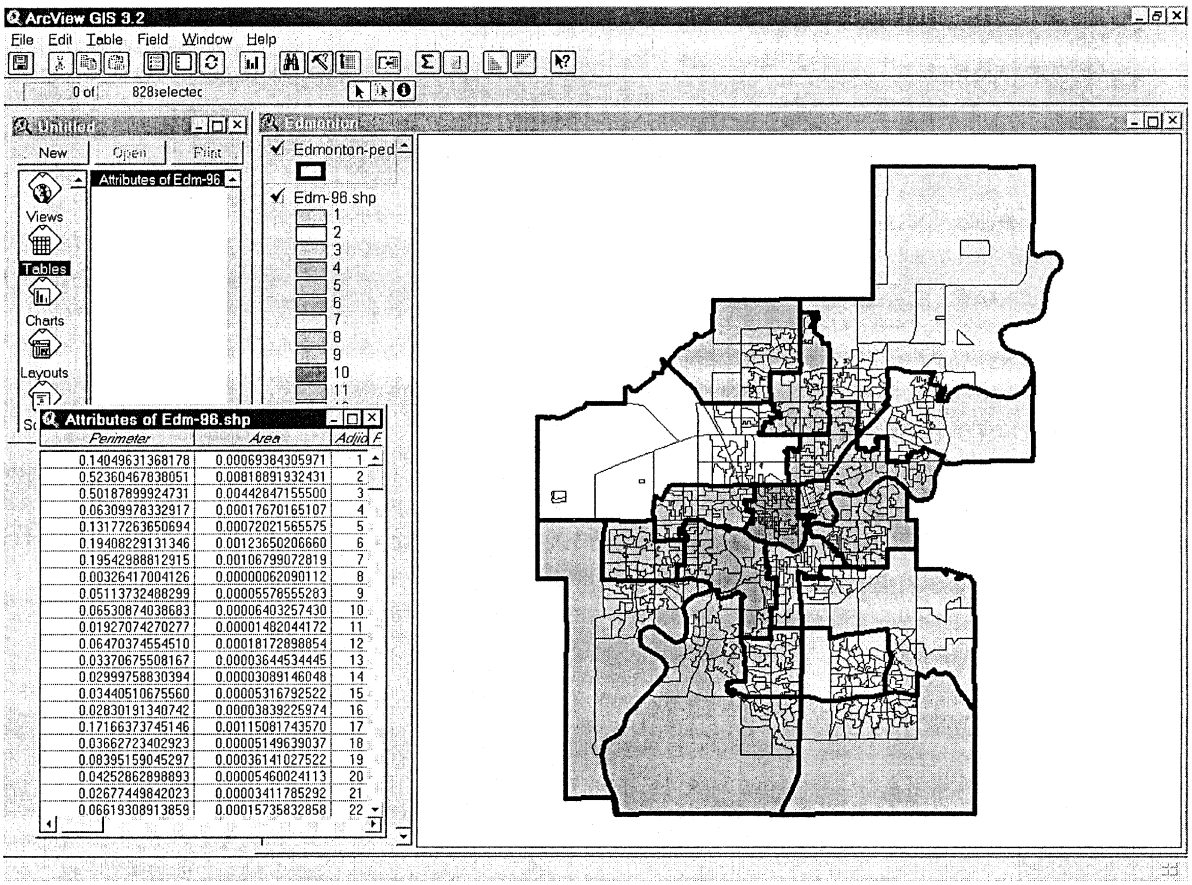

Fig. 3. ArcView sample screen showing the city of Edmonton.

of this list, apart from the last two, are related to the TS algorithm per se. The last two are used within the AMP procedure. Our analysis follows this classification.

Since it is unrealistic to test all combinations of all candidate values for all seven TS parameters, we employed a fractional factorial design (FFD) technique of the type described by Moen et al. (1991). Each parameter was assigned a low and a high value, identified by "–" and "+", respectively, but instead of considering all 2^7 combinations, only eight were selected. Each of them was tested using five different starting solutions, using only the population criterion with $\beta = 0.25$, and the first compactness criterion with a weight $\alpha_{\text{comp1}} = 1$. This means that a total of 40 runs were made. The eight combinations of parameter values are displayed in Table 1.

In FFD, the aim is to determine which parameters have the most impact on the solution value. To this end, the objective function value is recorded for each run. For each parameter, the difference between the sum of the objective values for the tests with minus sign and the sum for the tests with plus sign is computed. This difference is further divided by 4, the number of pluses (or minuses) in each column. The resulting value indicates the relative influence of the parameter on the solution quality with respect to the other parameters. This analysis ignores the inter-dependence of the parameters (e.g., one parameter setting consistently performing well with a partic-

Table 1
Eight combinations of the TS parameter values in the factorial design technique

Parameter	$\bar{\theta}$	θ'	α	μ	$\bar{\mu}$	ρ	τ_1
Low (−)	15	0	0.5	15	0.6μ	0.05	$[115\sqrt{m}]$
High (+)	95	20	2	150	μ	0.2	$[230\sqrt{m}]$
Run 1	−	−	−	+	+	+	−
Run 2	+	−	−	−	−	+	+
Run 3	−	+	−	−	+	−	+
Run 4	+	+	−	+	−	−	−
Run 5	−	−	+	+	−	−	+
Run 6	+	−	+	−	+	−	−
Run 7	−	+	+	−	−	+	−
Run 8	+	+	+	+	+	+	+

ular setting of another parameter), but it provides a quick way of identifying the most influential parameters. Once the fractional factorial experiments are completed, a subset of the parameters are identified as the most critical ones and are subject to further detailed testing. In our case, two parameters were chosen. These two parameters were then assigned several settings and the best objective values reported by the TS algorithm for these settings were recorded in a two-dimensional table, with rows corresponding to the settings of one parameter and the columns corresponding to those of the other parameters. This step was repeated for 20 random starting solutions. The best parameter combination was then used in tests with the AMP. The average of the 20 objective function values was computed for each combination. From the resulting two-dimensional table of averages, the following four groups of combination were included in the set of combinations tested with the AMP:

1. The combination yielding the *minimum* value in the table of averages.
2. The combination yielding the minimum objective value across *all* 20 runs.
3. The combination yielding the best of the row minima and the best of column minima.
4. Compute, for each row (column), the number of times the parameter setting for that row (column) has yielded the column (row) minimum. Then, include the combination associated with the row and column having the highest such score.

In case of ties, all tied combinations were selected for the AMP tests. For the parameter tuning experiments, we fixed α_{pop} at 10 and we successively used $\alpha_{\text{comp1}} = 1$ and $\alpha_{\text{comp2}} = 1$. In the population equity function, we successively used $\beta = 0.25$, 0.10 and 0.05. Each of these combinations was then run with $\gamma = 10$ and 20, and with $\tau_3 = 5, 10$, and 20, using each of the best combinations of parameters of the TS algorithm.

Our analysis showed that the two most critical parameters were $\bar{\theta}$ and μ, and that the others could safely be set at the following values: $\theta' = 10$, $\alpha = 1$, $\bar{\mu} = \mu$, $\rho = 0.1$ and $\tau_1 = \lceil 230\sqrt{m} \rceil$. Further tests were conducted on $\bar{\theta}$ and μ for $\beta = 0.05, 0.10$ and 0.25, for each of the two compactness measures, on the Edmonton data. The best tabu tenure ranges and μ values resulting from these experiments are given in Table 2 and all were used in our scenario analysis.

We observed that the use of an adaptive memory procedure produces an average 4% improvement in the objective function. However, the larger the value of β, the wider the range of improvements. For example, the largest improvement was 10.29% for $\beta = 0.25$, 7.40% for $\beta = 0.10$, and 4.77% for $\beta = 0.05$. Improvements also tended to be smaller in instances where the population equality constraint was smaller.

4.3. Scenario analysis

Using the parameter values determined in Section 4.2, we then analyzed several scenarios (for

Table 2
Best tabu ranges and μ values

	Compactness measure 1		Compactness measure 2	
β	$[\theta_{\min}, \theta_{\max}]$	μ	$[\theta_{\min}, \theta_{\max}]$	μ
0.05	[90, 100]	15	[70, 80]	75
0.10	[80, 90]	15	[90, 100]	90
0.25	[80, 90]	15	[90, 100]	15

the Edmonton data), using four criteria with the weights shown in Table 3. These scenarios were repeated with $\beta = 0.05$, 0.10 and 0.25.

The districting map corresponding to each experiment was drawn, enabling a visual appreciation of each solution. It is impossible to report here all tests and analyses that were performed. The following example illustrates the type of conclusion that can be drawn by comparing solutions and maps obtained under different conditions. Using the weights of Scenario 5, one can visualize the impact of β if the first compactness measure is used and all remaining parameters and weights remain constant. Figs. 4 and 5 depict the solutions obtained with $\beta = 0.25$ and $\beta = 0.05$, respectively. It can be seen that reducing the allowable population deviation from 25% to 5% has some adverse impact on compactness, but does not result in an unacceptable solution.

The test results for the scenario runs also indicate that the TS algorithm can produce districting maps that are as much as 27% more compact than the existing plan in use (with $\beta =$ 25%). Even with $\beta = 5\%$, one can construct a map that is 16% more compact than the current plan using the algorithm. Very tight constraints on population deviation (as low as 1%) are also within the algorithm's reach.

The tabu search algorithm can produce solutions that maintain the integrity of communities better than the existing plan. The districting map in Fig. 4 is superior to the existing plan: its compactness score (using measure 1) is 8% better and the integrity of communities measure is 13% better. As for the similarity to the existing plan (Scenarios 2–4), the algorithm produces maps that are up to 12% more compact than the existing plan, and at the same time are fairly similar to it.

We should note here that the time it takes the algorithm to complete a single run (without AMP) is about 3.5 minutes, based on the experiment parameters described earlier. When integrated with AMP, the algorithm executes a fixed number of such runs, therefore the total run length is roughly an integer multiple of 3.5 minutes. It is also possible to use a smaller (and less detailed) set of basic units (e.g. counties) to reduce the execution time. In this case, it could also be possible to implement the algorithm on-line, facilitating a client-server communication system. This would allow users to run multiple scenarios simultaneously and generate districting maps "on the fly".

Countless other analyses are of course possible. We believe one of the main benefits of the interactive tool we have developed is to enable decision makers to quickly produce alternative scenarios, visualize them, determine how to weigh various criteria and eliminate from consideration criteria that do not seem to significantly affect the shape of the solution.

Table 3
Weights used for the various scenarios

Scenario	Population equity	Compactness	Similarity to the existing plan	Integrity of communities
1	10	1	0	0
2	10	1	1	0
3	10	1	5	0
4	10	1	10	0
5	10	1	0	1
6	10	1	0	5
7	10	1	0	10

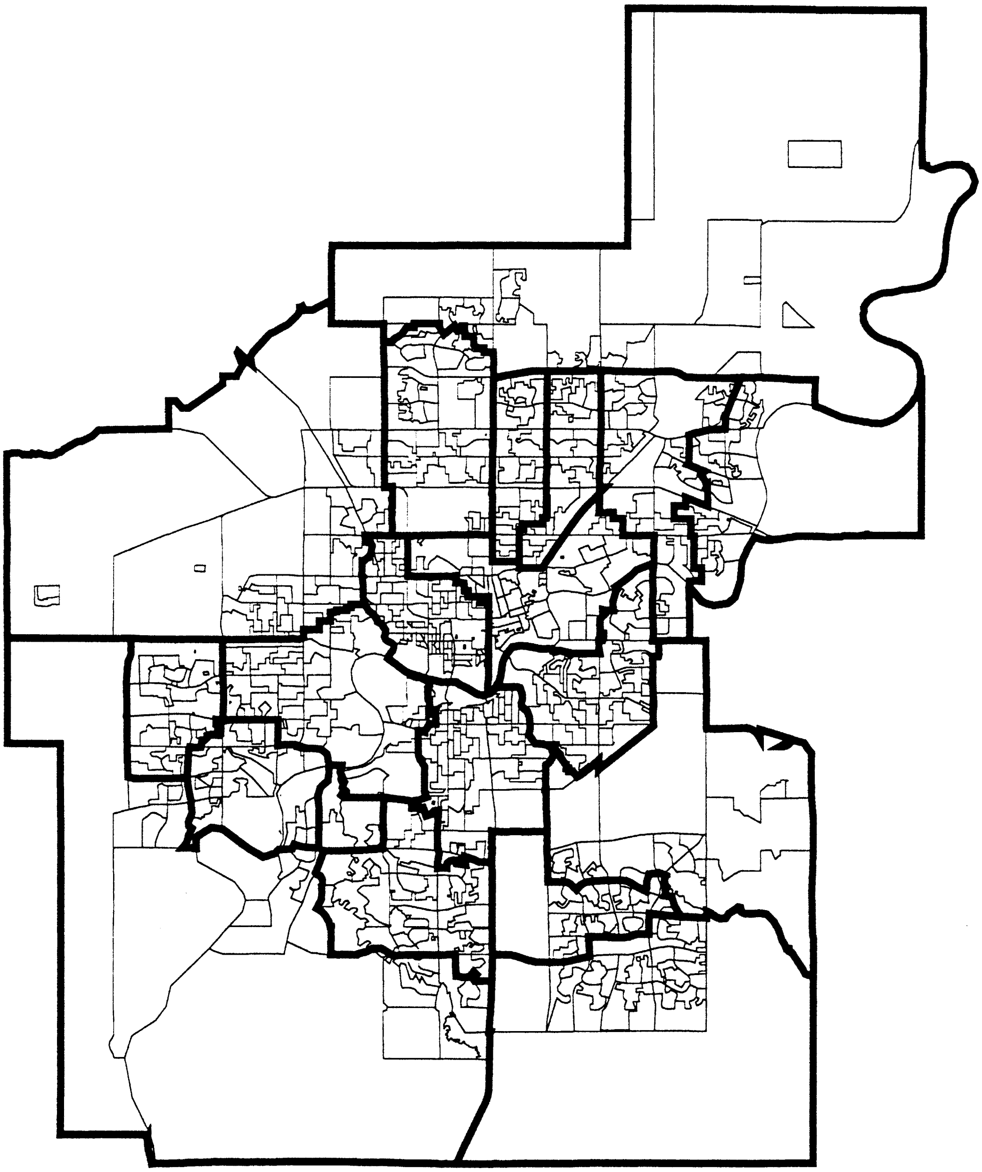

Fig. 4. Edmonton districting plan for scenario 5, compactness measure 1, and $\beta = 0.025$.

5. Conclusions

We have formulated the districting problem as a multicriteria problem and we have developed a tabu search heuristic for its solution. An interactive system was developed to visualize the solutions produced by the algorithm, and hence enable users to produce and compare various scenarios. With respect to first generation heuristics, the proposed method is robust and powerful: it can easily encompass a large number of criteria and it produces feasible and high quality solutions. It requires no sophisticated integer linear programming capabilities and can be easily implemented

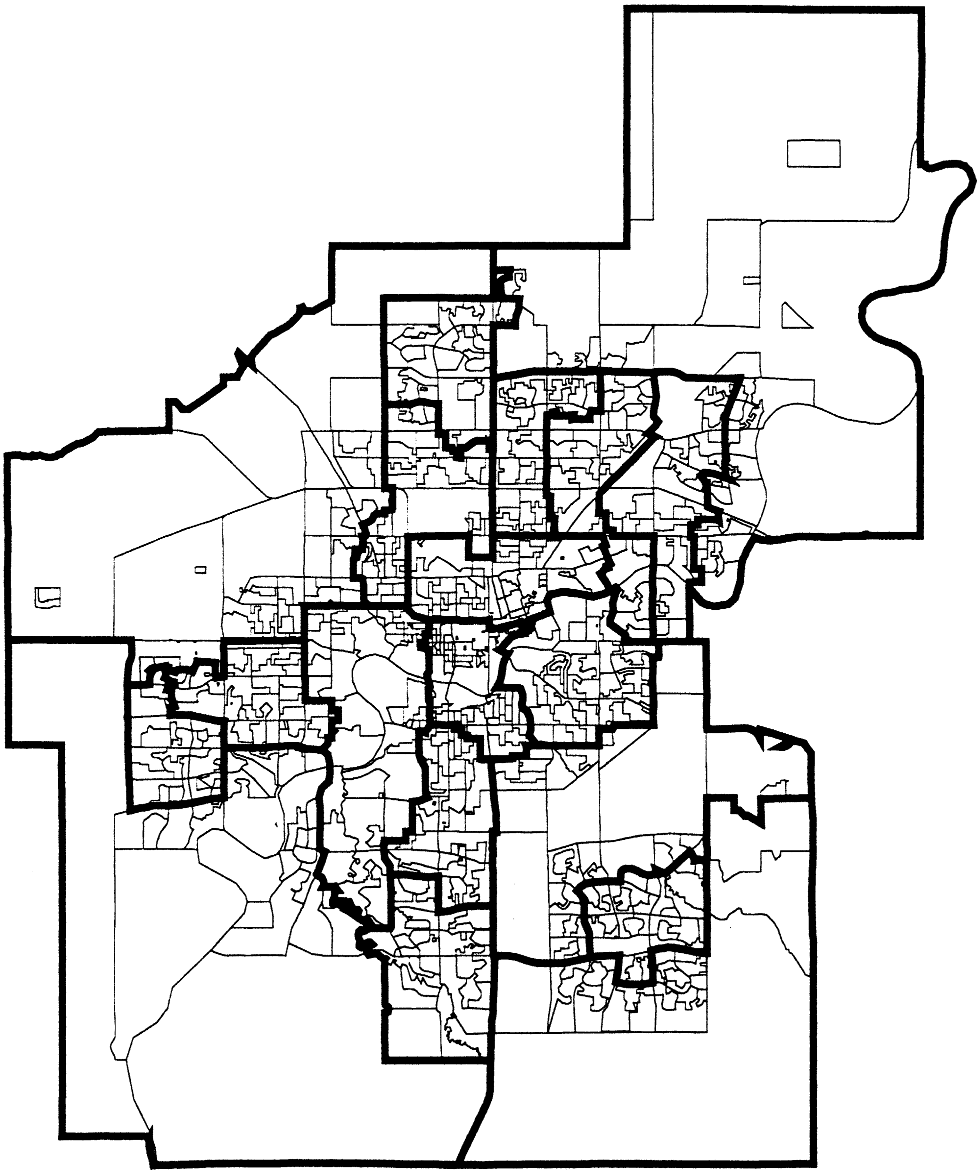

Fig. 5. Edmonton districting plan for scenario 5, compactness measure 1, and $\beta = 0.05$.

on almost any system. Our test results indicate that the algorithm can produce maps that dominate the existing districting map of Edmonton with respect to compactness and integrity of communities. It can also reduce the amount of deviation around the average district population from the current 25% to much lower levels (such as 1%), improving on the equality of representation. Finally, by changing the weights of the objective function terms, one can quickly generate multiple districting maps that appeal to different interests. It is this feature of our multicriteria approach, powered by intelligent problem solving, that we believe is going to be the key element of the new generation of algorithms for computer-aided political districting.

Acknowledgements

This work was partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grants OGP5547956 and OGP0039682. This support is gratefully acknowledged. Thanks are also due to Terry Daniel, M. John Hodgson, Armann Ingolfsson, Paul Johnston, Susan Neuman and three anonymous referees for their valuable comments.

References

Aarts, E.H.L., Lenstra, J.K., 1997. Local Search in Combinatorial Optimization. Wiley, Chichester.

Barr, R.S., Golden, B.L., Kelly, J.P., Resende, M.G.C., Stewart Jr., W.R., 1995. Designing and reporting on computational experiments with heuristic methods. Journal of Heuristics 1, 9–32.

Bourjolly, J.-M., Laporte, G., Rousseau, J.-M., 1981. Découpage électoral automatisé: application à l'Île de Montréal. INFOR 19, 113–124.

Bozkaya, B., 1999. Political Districting: A Tabu Search Algorithm and Geographical Interfaces. Ph.D. Thesis, University of Alberta.

Browdy, M.H., 1990. Simulated annealing: An improved computer model for political redistricting. Yale Law and Policy Review 8, 163–179.

Buchanan, J.T., Daellenbach, H.G., 1987. A comparative evaluation of interactive solution methods for multiple objective decision models. European Journal of Operational Research 29, 353–359.

Cain, B.E., 1984. The Reapportionment Puzzle. University of California Press, Los Angeles.

Cordeau, J.-F., Gendreau, M., Laporte, G., 1997. A tabu search heuristic for periodic and multi-depot vehicle routing problems. Networks 30, 105–119.

Eiselt, H.A., Laporte, G., 1987. Combinatorial optimization problems with soft and hard requirements. Journal of the Operational Research Society 38, 785–795.

Fleischmann, B., Paraschis, J.N., 1988. Solving a large scale districting problem: A case report. Computers and Operations Research. 15, 521–533.

Garfinkel, R.S., Nemhauser, G.L., 1970. Optimal political districting by implicit enumeration techniques. Management Science 16, 495–508.

Gendreau, M., Hertz, A., Laporte, G., 1994. A tabu search heuristic for the vehicle routing problem. Management Science 40, 1276–1290.

Glover, F., 1977. Heuristic for integer programming using surrogate constraints. Decision Sciences 8, 156–166.

Glover, F., Laguna, M., 1997. Tabu Search. Kluwer, Boston.

Government of Alberta. 1997. Report of the Chief Electoral Officer: November, 1996 General Enumeration and Tuesday, March 11, 1997 General Election Twenty-Fourth Legislative Assembly, p. 76.

Grilli di Cortona, P., Manzi, C., Pennisi, A., Ricca, F., Simeone, B., 1999. Evaluation and optimization of electoral systems. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia.

Hess, S.W., Weaver, J.B., Siegfeldt, J.N., Whelan, J.N., Zhitlau, P.A., 1965. Nonpartisan political districting by computer. Operations Research 13, 998–1006.

Kaiser, H., 1966. An objective method for establishing legislative districts. Midwest Journal of Political Science 10, 200–213.

Lewyn, M.E., 1993. How to limit gerrymandering. Florida Law Review 45, 403–486.

Macmillan, W., Pierce, T., 1992. Optimization modelling in a GIS framework: the problem of political districting. Specialist meeting, April 16–18, 1992. National Center for Geographic Information and Analysis.

Mehrotra, A., Johnson, E.L., Nemhauser, G.L., 1998. An optimization based heuristic for political districting. Management Science. 44, 1100–1114.

Moen, R.D., Nolan, T.W., Provost, L.P., 1991. Improving Quality through Planned Experimentation. McGraw-Hill, New York.

Morrill, R.L., 1973. Ideal and reality in reapportionment. Annals of the Association of American Geographers 63, 463–477.

Morrill, R.L., 1976. Redistricting revisited. Annals of the Association of American Geographers 66, 548–556.

Nagel, S.S., 1965. Simplified bipartisan computer redistricting. Stanford Law Review 17, 863–899.

Nagel, S.S., 1972. Computers and the law and politics of redistricting. Polity 5, 77–93.

Niemi, R.G., Grofman, B., Carlucci, C., Hofeller, T., 1990. Measuring compactness and the role of a compactness standard in a test for partisan and racial gerrymandering. Journal of Politics 52, 1155–1181.

Osman, I.H., Laporte, G., 1996. Metaheuristics: A bibliography. In: Laporte, G., Osman, I.H. (Eds.), Metaheuristics in combinatorial optimization. Annals of Operations Research 63, 513–623.

Plane, D.A., 1982. Redistricting reformulated: A maximum interaction/minimum separation objective. Socio-Economic Planning Sciences. 16, 241–244.

Rochat, Y., Taillard, É.D., 1995. Probabilistic diversification and intensification in local search for vehicle routing. Journal of Heuristics 1, 147–167.

Roy, B., 1985. Méthodologie Multicritère d'Aid à la Décision. Economica, Paris.

Taillard, É.D., 1991. Robust taboo search for the quadratic assignment problem. Parallel Computing 17, 433–445.

Thoreson, J.D., Liittschwager, J.M., 1967. Legislative districting by computer simulation. Behavioral Science 12, 237–247.

Vickrey, W., 1961. On the prevention of gerrymandering. Political Science Quarterly 76, 105–110.

Vincke, P., 1992. The Multicriteria Decision Aid. Wiley, Chichester.

Weaver, J.B., Hess, S.W., 1963. A procedure for nonpartisan districting: development of computer techniques. The Yale Law Journal 72, 288–308.

Williams Jr., J.C., 1995. Political redistricting: A review. Papers in Regional Science 74, 13–40.

Young, H.P., 1988. Measuring the compactness of legislative districts. Legislative Studies Quarterly 13, 105–115.

Yu, P.L., 1989. Multiple criteria decision making: Five basic concepts. In: Nemhauser, G.L., Rinnooy Kan, A.H.G., Todd, M.J. (Eds.), Optimization Handbooks in Operations Research and Management Science, vol. 1. North-Holland, Amsterdam, pp. 663–699.