MANAGEMENT SCIENCE
Vol. 16, No. 8, April, 1970
Printed in U.S.A.

OPTIMAL POLITICAL DISTRICTING BY IMPLICIT ENUMERATION TECHNIQUES*†

R. S. GARFINKEL‡ AND G. L. NEMHAUSER§

An algorithm is given which finds all optimal solutions, for a given set of criteria, to political redistricting problems. Using "population units" as indivisible elements, the first phase generates all feasible districts, where feasibility indicates contiguity, compactness and limited population deviation. The second phase finds that set of M feasible districts which "covers" each population unit exactly once, and minimizes the maximum deviation of any district population from the mean district population. Computational results indicate that states with 40 counties or fewer can be solved in less than 10 minutes on an IBM 7094. However, our attempt to solve a 55 county state was unsuccessful.

1. Introduction

Political districting[1] is the process by which an area (e.g., a state) is partitioned into smaller areas (districts) each of which is assigned an (integral) number of representatives. Historically, the state has been partitioned into *population units* (e.g., counties or census tracts) each of which must be assigned to one and only one *district*. The resulting districting scheme is called a *plan*.

We shall assume that each district is assigned one representative and that there are to be M districts. Our objective will be to determine all optimal plans. This is done in two phases. In Phase I, we generate districts from the population units, that satisfy specified criteria on population, compactness and contiguity. In Phase II, we combine these districts to determine those plans that minimize maximum population deviation. After presenting the algorithms, we will give results for some real problems.

2. The Basic Model

Population

The vast majority of districting plans which have been ruled unconstitutional failed on the grounds that there existed some district which had a disproportionately large or small population. As a result of these rulings, it is generally possible to get an indication of the maximum deviation allowed by the courts from the mean district population. Consequently a constraint may be determined which says that no district population may deviate from the mean by more than a given number.

Specifically, let p_i, $i = 1, \cdots, N$ be the population of unit i. The mean population is $\bar{p} = \sum_{i=1}^{N} p_i/M$. Let $P(j) = \sum_{i=1}^{N} a_{ij}p_i$ be the population of district j, where $a_{ij} = 1$ if population unit $i \in$ district j and 0 otherwise. District j is feasible only if

$$(1) \qquad | P(j) - \bar{p} | \leq \alpha\bar{p},$$

where $100\,\alpha$, $(0 \leq \alpha \leq 1)$, is the maximum allowable percentage deviation of the population of a district from the mean district population.

* Received April 1969; revised September 1969.
† Research supported in part by CROND, Inc., Wilmington, Delaware.
‡ The University of Rochester.
§ Presently Visiting Professor at the Center for Operations Research and Econometrics, University of Louvain, Belgium, on leave from Cornell University.
[1] An excellent legal study of redistricting is McKay [12].

The Apportionment Act of 1911, besides requiring nearly equal population in congressional districts, also stated that districts must be of contiguous and compact territory. Since then, contiguity and compactness have often been included as desirable features of districts.[2]

Contiguity

Loosely speaking, a district is contiguous if it is possible to walk from every point in the district to every other point without crossing another district.

Contiguity can rigorously be defined as follows: Let $B = \{b_{ik}\}$ be a symmetric $N \times N$ matrix with

$$b_{ik} = 1 \quad \text{if units } i \text{ and } k \text{ have a common boundary greater than a point (are contiguous),}$$
$$= 0 \quad \text{otherwise.}$$

Consider a district to be an undirected graph, whose vertices are the units of the district, and an arc exists between vertices i and k if and only if $b_{ik} = 1$. The district is contiguous if and only if the graph is connected (a path exists between every pair of vertices).

(2) A district is feasible only if it is contiguous.

Compactness

There are many possible definitions of compactness, but they can essentially be divided into geographical compactness and population compactness. In essence, a district is said to be geographically compact if it is somewhat circular or square in shape rather than long and thin. The main desirability of compactness is that the possibility of gerrymandering is reduced. In fact, compactness has recently [13] been defined as simply the absence of gerrymandering. The importance of compactness, however, is generally agreed to be less than that of other criteria [8]. This is especially true because gerrymandering is not a problem when an algorithm does not use political data. Hess [9] argues that a more reasonable measure of compactness is one which takes into account the distance of the population from the center of the district. He has developed a solution technique [11] which uses such a measure.

Here we only consider geographical compactness and divide it into "distance" compactness and "shape" compactness. As we shall see judicious use of distance compactness is critical in any large ($N \geq 30$) problem.

Let the distance between the centers of units i and k be $d(i, k)$. For each pair of units an "exclusion distance" $e(i, k)$ is determined. District j is feasible only if

$$d(i, k) > e(i, k) \rightarrow a_{ij} \cdot a_{kj} = 0. \tag{3}$$

For district j, $d_j = \max_{i,k} \{d(i, k)\, a_{ij} a_{kj}\}$, $i, k = 1, \cdots, N$, is defined to be the distance between the units of j which are farthest apart.[3] ($d_j = 0$ if district j contains only one unit).

Let $A(j)$ be the area of district j. Then $c'_j = d_j^2/A(j)$ is a dimensionless measure of the shape compactness of district j. As c'_j decreases district j is said to be more compact.

[2] The act was repealed in 1929.

[3] Note that $d(i, k)$ is not measured within district j. We could get the maximum distance within j by letting $d'(i, k)$ be the length of a shortest path from i to k within j, and then define $d'_j = \max_{i,k} d'(i, k)\, a_{ij}\, a_{kj}$. Our measure of shape compactness would, most likely, be improved by using d', and this extension is being included in the computer program.

District j is feasible only if

$$c'_j \leq \beta, \qquad 0 \leq \beta \leq \infty. \tag{4}$$

A district is feasible if and only if it satisfies (1)–(4).

It seems desirable for reasons of voter equity to strive to minimize the maximum deviation of any district population from $\bar{p}$. Let $c_j = | P(j) - \bar{p} |/\alpha\bar{p}$ and assume that the ($N \times S$) matrix A of feasible districts has been determined. The objective is to find those plans (binary S-vectors X) which:

$$\text{minimize} \max_{j=1}^{S} c_j x_j \tag{5}$$

subject to

$$\textstyle\sum_{j=1}^{S} a_{ij} x_j = 1 \qquad i = 1, \cdots, N \tag{6}$$

$$\textstyle\sum_{j=1}^{S} x_j = M \tag{7}$$

$$x_j = 0, 1 \qquad j = 1, \cdots, S \tag{8}$$

where $x_j = 1$ if district j is in a plan and $x_j = 0$ otherwise.

Deleting (7) and replacing (5) with the more conventional min sum objective function yields a version of the set-covering problem (the partitioning problem) which has received a good deal of attention in the operations research literature; see [1], [2], [6] and [14]. The min-max objective function (5) is sometimes referred to as a "bottleneck" problem; see [4] for a more general discussion.

Other Criteria

A criterion which is thought [8] to be more important than compactness is that districts shall cross the boundaries of counties (or some other political subdivision) as infrequently as possible. A constraint of this kind is developed as an extension.

In addition to the above there are a number of other criteria which could be considered. In general these either lead to highly partisan considerations or else their desirability has been the subject of heated debate. For example, one could strive to: maximize the number of safe (or "swing") districts, attain homogeneity (or heterogeneity) of interests within each district, minimize the number of incumbents who will lose their present seats, guarantee one political party a victory in the election, or guarantee a close election. Some of these criteria are considered later.

Other Approaches

Brief reviews of some of the previous approaches to the problem are found in [11] and [15]. These techniques share the advantage of being fast and of being able to handle large problems. However, they also share the disadvantage of being inexact, they provide no guarantee of satisfying constraints nor do they optimize an objective function. In addition to these, Thoreson and Liitschwager [15] and Castellan [3] have developed heuristic techniques and Wagner [16] uses integer programming to minimize total population deviation.

3. Phase I—District Generation

In Phase I we find all binary column vectors a_j which satisfy (1)–(4). This is accomplished using a "tree search" algorithm. A general description of this type of algorithm is given by Geoffrion [7]. Briefly, the procedure is to start at an arbitrary unit and adjoin contiguous units until the combined population becomes feasible. If the district is compact we record it. When combined population exceeds the upper limit we "backtrack" on the enumeration tree. Checks are made periodically for enclaves (small groups of units isolated from the rest of the state by the proposed district). By applying some tests to the matrix $A = \{a_{ij}\}$ as it is being generated, it is frequently possible to eliminate a substantial number of rows and columns. These tests are called reductions and are generally quite important for efficient execution of the algorithm. However, because the reductions are not a central part of the algorithm they are given in the Appendix. Reductions are attempted every time we backtrack to the point where there are no units left in the proposed district.

The algorithm is described in detail below.

Terminology

After the numbers $e(i, k)$ have been determined we create the $(N \times N)$ symmetric "exclusion" matrix Z where

$$\begin{aligned} z_{ik} &= 0 \quad \text{if } d(i, k) > e(i, k) \quad (i \text{ and } k \text{ may not be in the same district}), \\ &= 1 \quad \text{otherwise.} \end{aligned}$$

The matrix Z may be modified in the course of the algorithm. (See Step VI in the Appendix.)

A "partial solution" to Phase I is a binary N-vector a_j which satisfies

$$P(j) \leq (1 + \alpha)\bar{p},$$

the units for which $a_{ij} = 1$ are contiguous and $z_{ik} = 0 \rightarrow a_{ij}a_{kj} = 0$. At any given time in the first phase we are able to partition the units into the following classes:

(1) Included units: units included in the current partial solution.
(2) Excluded units: units excluded from the current partial solution.
(3) Free units: units under consideration for inclusion with the units of the current partial solution.

As we test units for inclusion in the current partial solution, we represent the tested units by a vector V, with units appearing in order of testing. If unit k is included it is written as k, if excluded it is written as $-k$.

The units may now be further partitioned into:

(a) Non-tested units: units not appearing in V.
(b) Once-tested units: if $V = (a, b, \cdots, c, k, \cdots)$, so that k is included, then the vector $V = (a, b, \cdots, c, -k, \cdots)$ still remains to be tried.
(c) Twice-tested units: if $V = (a, b, \cdots, c, k, \cdots)$ and $V = (a, b, \cdots, c, -k, \cdots)$ has already been tried, we will write V as $(a, b, \cdots, c, \underline{k}, \cdots)$. Similarly if $V = (a, b, \cdots, c, -\underline{k}, \cdots)$, then the possibility of $V = (a, b, \cdots, c, k, \cdots)$ no longer exists.

As the algorithm progresses we need the following:

I: the set of included units, with $P(I) = \sum_{i \in I} p_i$.
E: the set of excluded units.
S_j: the set of units in district j.
F: the set of free units.
$B(i)$: the set of units such that $b_{ik} = 1$.
$Z(i)$: the set of units such that $z_{ik} = 0$.
$B(S) = \bigcup_{i \in S} B(i)$, where S is an arbitrary set.
$Z(S) = \bigcup_{i \in S} Z(i)$.
C: the set of free units contiguous to the included units[4] ($C = B(I) \cap F$).

N lists in which to file the generated districts. District j is filed in list i if i is the

[4] With one exception, see Step VI, in the Appendix.

lowest numbered unit such that $a_{ij} = 1$. The districts are filed in order of increasing cost with ties broken arbitrarily.

Once V is given, all sets described above are determined.

Enclaves

A set of contiguous units $S \subset \bar{I}$ forms an enclave if $P(S) < (1 - \alpha)\bar{p}$ and there do not exist units $i \varepsilon S$ and $k \varepsilon (\bar{I} \cap \bar{S})$ such that $b_{ik} = 1$ (i.e. $B(S) \cap \bar{I} \cap \bar{S} = \varnothing$).[5] As an input to the algorithm we use an integer EN which indicates how often an enclave check is performed. Clearly any district which causes an enclave cannot be in a plan.

Algorithm

Step I (Initialization). Set $V = (1)$, $j = 0$ (j counts the districts generated), $e = 0$ (when $e =$ EN an enclave check is performed), $D = \varnothing$ (the set of "fixed" districts), $U = \varnothing$ (the units of D).

Step II (Unit annexation). If $e \geq$ EN go to Step IV. Otherwise find the lowest numbered unit i such that $i \varepsilon C$ and $P(I) + p_i \leq (1 + \alpha)\bar{p}$. If no such unit exists go to Step V. Otherwise set $V = (V, i)$, $e = e + 1$. If $P(I) \geq (1 - \alpha)\bar{p}$ go to Step III. Otherwise repeat Step II.

Step III (File district). Compute c' for the units of I. If $c' > \beta$ go to Step II. Otherwise, let $j = j + 1$. We have created feasible district j. Let $c_j = | P(I) - \bar{p} |/\alpha\bar{p}$ (note that $0 \leq c_j \leq 1$). Let i be the lowest numbered unit of I. File district j in list i in order of nondecreasing cost (c_j). Go to Step II.

Step IV (Enclave check). Check to see if an enclave exists (as described above). If there is no enclaved set S, set $e = 0$ and go to Step II. If set S forms an enclave and if there exists a unit i such that $i \varepsilon (S \cap E)$, or if any unit of S is excluded from any other unit of S or if $P(I) + P(S) > (1 + \alpha)\bar{p}$, set[6] $e =$ EN $- 1$ and go to Step V. Otherwise set $e = 0$, let $S = (s_1, s_2, \cdots, s_p)$ and set $V = (V, \underline{s}_1, \underline{s}_2, \cdots, \underline{s}_p)$. Then, if $P(I) \geq (1 - \alpha)\bar{p}$ go to Step III, but if $P(I) < (1 - \alpha)\bar{p}$ go to Step II.

Step V (Backtrack). Let i be the rightmost unit of V which is not underlined. Replace i by $-\underline{i}$ and remove all entries to the right of i. If V contains N underlined entries terminate. Otherwise, set $e = e + 1$. If any entries of V are not underlined go to Step II. Otherwise go to Step VI. Step VI searches for reductions and because it is rather long it is given in the Appendix.

Example. Consider the fictional state having nine population units ($N = 9$) and four representatives ($M = 4$) in Figure 1. The circled number within each unit is its population.

The exclusion matrix has (somehow) been calculated as

$$
Z = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array}
\begin{array}{c}
\begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \\
\begin{bmatrix}
- & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
1 & - & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
1 & 1 & - & 1 & 1 & 1 & 1 & 0 & 0 \\
1 & 1 & 1 & - & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & - & 0 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 & 0 & - & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & - & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & - & 1 \\
0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & -
\end{bmatrix}
\end{array}.
$$

[5] The set $\bar{X}$ denotes the complement of the set X.

[6] Having just found an enclave, it is likely that more enclaves will occur in this part of the state. Setting $e =$ EN $- 1$ allows for another enclave check to be done at the next step.

Here $\bar{p} = 120/4 = 30$. Assume that $\alpha = 0.1$, so that the feasible district population range is (27, 33). Assume that β is very large (say 1000) so that shape compactness is not a factor. Let EN = 5 and

$$B = \begin{array}{cc} & \begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \\ \begin{array}{c} 1\\ 2\\ 3\\ 4\\ 5\\ 6\\ 7\\ 8\\ 9 \end{array} & \begin{bmatrix} - & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & - & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & - & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & - & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & - & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & - & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & - & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & - & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & - \end{bmatrix} \end{array}.$$

The districts generated,[7] along with their costs, and positions in appropriate lists are given in Table 1. In order to clarify the enclave concept, we note that if a check is done at $V = (-\underline{1}, -\underline{2}, 3, -\underline{4}, -\underline{6}, 7)$, the set (6) is an enclave and backtracking occurs.

4. Phase II—Optimization

In the second phase we find every optimal solution to the problem (5) – (8). This algorithm is also of the tree search variety. It is very fast, chiefly because the organization of the districts into lists allows STEP II below to be very effective. Specifically, it permits large numbers of districts to be dropped from consideration simply because they are not in the lists of interest. The algorithm is described in detail below.

Terminology

A "partial solution" in Phase II is a vector X satisfying:

$$\sum_{j=1}^{S} a_{ij} x_j \leq 1, \qquad i = 1, \cdots, N$$
$$\sum_{j=1}^{S} x_j \leq M,$$
$$x_j = 0, 1.$$

At any given time in the second phase we will be able to partition the variables x_j into the following classes:[8]

(1) fixed variables: those districts which must be in any plan (see Theorem 2 in the Appendix),

(2) trial positive variables: districts which are included in the current partial solution but which are not required to remain in later solutions,

(3) trial zero variables: districts which are excluded from the current partial solution but which may appear in later solutions,

(4) free variables: districts which are under consideration for inclusion in the current partial solution.

[7] The details of the example are given by Garfinkel [5].

[8] We could add a fifth class, namely districts which may not appear in any plan (see Theorems 2–4 in the Appendix), but these columns are simply deleted.

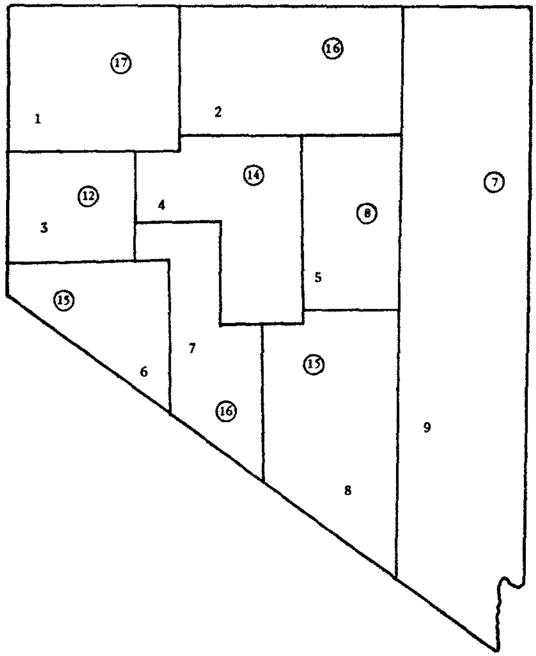

FIGURE 1. Example

TABLE 1

District #	Units	Cost	Filed in List	In Position
1	1, 2	1.00	1	3
2	1, 3	.33	1	2
3	1, 4	.33	1	1
4	2, 4	0.00	2	1
5	2, 5, 9	.33	2	2
6	3, 6	1.00	3	1
7	4, 5, 9	.33	4	3
8	4, 7	0.00	4	1
9	4, 8	.33	4	2
10	5, 8, 9	0.00	5	1
11	6, 7	.33	6	1
12	7, 8	.33	7	1

At the completion of Phase I, S districts (column vectors) have been generated. In addition each district j has been placed in one of N lists, where district j is in list i if the lowest numbered unit in the district is unit i. The districts have been filed in each list in order of nondecreasing cost (c_j) with ties broken arbitrarily. Some districts may also have been fixed to be in the plan by reduction techniques in Phase I. We call the set of fixed districts D and the set of units covered by these districts U. The number of districts in $\acute{D}$ is $N(D)$, and we must find $M - N(D)$ districts which cover the units of $\bar{U}$. Of course if $N(D) = M$ Phase II would not have been entered.

As the algorithm progresses we need the following:

f: $0 \leq f \leq M - N(D)$, the number of trial positive districts in the current partial solution,

Y: the set of districts in the current partial solution,

T: the units in the districts of Y,

S_j: the set of units in district j.

Algorithm

Step I (Initialization). Set $f = 0$, $Y = D$ and $T = U$. Go to Step II.

Step II (Choose next list). Find the lowest numbered unit i such that $i \varepsilon \bar{T}$. If there does not exist such a unit then we have covered all the units with fewer than M districts and go to Step IV. If we find a unit uncovered, set an indicator which tells us to begin at the "top" (lowest cost district) of list i and go to Step III.

Step III (District annexation). Begin at the indicated position in list i and examine in order of increasing cost the districts of the list. If we find a district j such that[9] $T \cap S_j = \varnothing$ then set $Y = Y \cup (j)$, $T = T \cup S_j$ and go to Step V. If no such district is found go to Step IV.

Step IV (Backtrack). There are no other plans containing the districts in the current partial solution (the districts whose units make up the set T). If $f = 0$ there are no other solutions and we terminate. Otherwise set $f = f - 1$. Subtract the units of the last trial positive district j in the partial solution from T. Set $Y = Y - (j)$. Set an indicator at the position in the list from which this last trial positive district was taken. Go to Step III.

Step V (Test for plan). Set $f = f + 1$. If $f = M - N(D)$ and the districts in the solution cover all the units, go to Step VI. If $f = M - N(D)$ and the districts in the solution do not cover all the units go to Step IV. If $f \neq M - N(D)$ go to Step II.

Step VI (Plan is found). A plan has been found whose districts make up the set Y and whose value V is the largest c_j of the districts j in Y. Delete from each list any district n for which $c_n > V$. If no districts are discarded go to Step IV. However, if any districts are discarded go to Step VII (given in the Appendix).

Example.[10] Continuing with the Phase I example, assume the given districts are as shown in Table 1. The A matrix is

$$
\begin{array}{cc}
 & \begin{array}{cccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} &
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0
\end{bmatrix}
\end{array}.
$$

Two optimal plans are found each with cost = .33. They are (2, 4, 10, 11) and (2, 5, 9, 11).

[9] This test and other logical tests are easily done using binary "AND"-"OR" operations.

[10] See Garfinkel [5] for the details of the example.

5. Extensions

None of the extensions given in this chapter have been implemented, although this would be easy to do for each of them.

Objective Function

There is no reason to restrict ourselves in Phase II to minimizing $\max_j c_j x_j$. Any of the following could easily be minimized:

(1) $\sum_j c_j x_j$—simply amend Step III in Phase II to stop the search of list i when a district is found which causes $\sum_j c_j x_j$ to exceed the current lower bound (value of the best solution found to date). In Step VI we let $\underline{V} = \sum_j c_j x_j$.

(2) $\max c'_j x_j$—the extension is trivial. Just keep track of c'_j instead of c_j.

(3) $\sum_j c'_j x_j$—identical to 1.

Note that if a summation objective function is used, the power of Step VI in Phase II is lost.

County Boundaries

We may wish to limit the crossing of county (or other political subdivision) boundaries by districts. Define Q_j to be the number of counties partly, but not completely, in district j. In Phase I we use the constraint

$$Q_j \leq \gamma. \tag{9}$$

Constraint (9) is implemented by simply keeping track of Q_j for the partial solution. If it exceeds γ we backtrack.

Political Victory

We may hope to guarantee one or more political parties (or any other segment of the population) at least a given number of representatives. Let p'_i be the number of voters of population unit i who are in the desired segment of the population. Let

$$\begin{aligned} t_j &= 1 \quad \text{if } \textstyle\sum_{i=1}^{N} a_{ij} p'_i > P(j)/2 \\ &= 0 \quad \text{otherwise} \end{aligned}$$

be an indicator of whether or not the given segment controls district j. We use the constraint

$$\textstyle\sum_{j=1}^{S} t_j x_j \geq \delta, \qquad 0 \leq \delta \leq M. \tag{10}$$

We implement constraint (10) by calculating t_j for every district j in Phase I. At Step V in Phase II we would then reject any tentative plan which violated (10). Of course, any number of constraints of this type could be invoked simultaneously, for different segments of the population.

6. Results

Three real problems[11] of varying degree of difficulty were attempted; Sussex County, Delaware with $N = 26$ (census tracts) and six districts required, the state of Washington with $N = 39$ counties and seven districts and West Virginia with $N = 55$ counties

[11] Problems suggested and data supplied by CROND, Inc.

TABLE 2

Sussex County

RUN	1	2	3
α	.1	.1	.1
β	∞	1.5	1.25
EN	3	3	3
Districts Generated	536	204	128
Phase I Time	21.5 sec.	17.8 sec.	16.3 sec.
Phase II Time	3.1 sec.	1.3 sec.	0.0 sec.
Maximum Population Deviation	.034	.069	No solution
Worst c'_j	1.92	1.44	

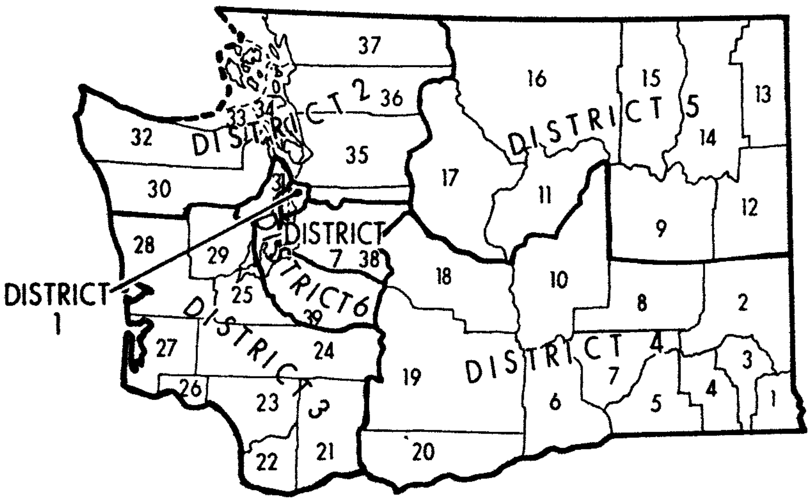

FIGURE 2. Washington, present plan

and five districts. Details of the results can be found in [5], and only a summary will be given here. The computer runs were on the IBM 7094.

Sussex County was very easy to district and no exclusion distances were needed. Some typical results are given in Table 2.

No successful runs were achieved in West Virginia. Phase I was either excessively long (with large exclusion distances) or terminated with no feasible solution (with small exclusion distances).

Washington proved to be the most interesting case and we will elaborate a bit on the results. The mean population $\bar{p}$ is 407,602. However, the population of county 38 (King County) is 935,014, so a single-member, seven district solution which maintains counties as population units is clearly infeasible. Since we restrict ourselves to single-member districts and since we wished to use counties as units we noted that County 39 (Pierce County) which is contiguous to 38 has a population of 321,590. Together the two counties have 1,256,604. If we make 3 districts out of these two counties, the mean population of each of the three would be 418,868 or about a 4% deviation from $\bar{p}$. We could then solve the remaining 37 county problem to obtain 4 districts.

Figure 2 shows the present plan, which does not preserve county boundaries. Its

TABLE 3
Washington

Run	1	2	3	4
α	.03	.006	.04	.1
β	∞	∞	2.0	1.5
EN	5	5	5	5
Districts Generated	1648	862	2766*	1466
Phase I Time	3.1 min.	5.5 min.	7.3 min.	9.0 min.
Phase II Time	14.0 sec.	5.5 sec.	38.2 sec.	—**
Max. Population Deviation	.02 (j = 4)	.003 (j = 3)	.038 (j = 3)	—
Worst c'_j	3.38 (j = 3)	2.62 (j = 2)	1.90 (j = 1)	—
Alternative optima	None	None	None	None

* Only the best 2495 of these were saved for Phase II.
** Termination occurred in Phase I as a unit was uncovered.

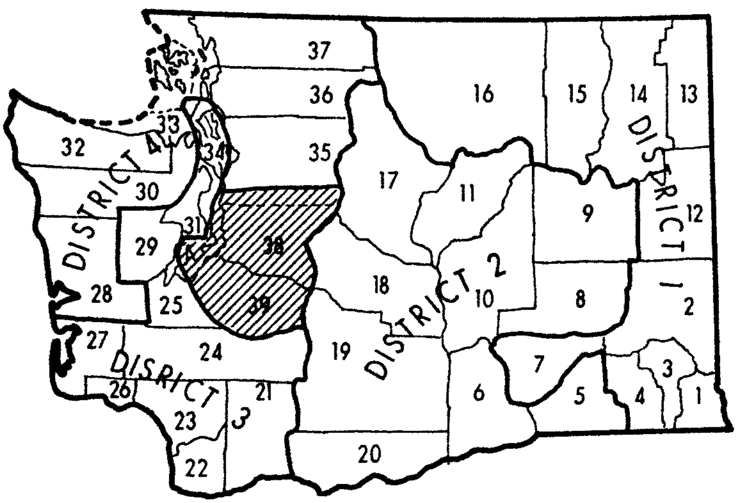

FIGURE 3. Washington, α = .03, β = ∞, $e(i, k)$ = 300 km.

worst district (#7) has a deviation of 25% from $\bar{p}$. Its worst district (#4) in shape compactness has value 1.79.

Results are given in Table 3. Exclusion distances are needed for computational feasibility. The drastic improvement in population equality of any of the computed plans over the present plan is clear. In run 3 we are able to achieve 3.8% population deviation with little loss in compactness from the present plan.

Finally, in Washington there exists a natural boundary, namely the Cascade mountains, which separates the present districts 4 and 5 from the rest of the state. The partition created by these mountains is more than just a geographical partition. On opposite sides of the mountain there are distinct social, political, industrial and other differences. Because of these considerations it may be politically realistic to consider this boundary inviolate. If we do so, the problem becomes computationally simpler. In fact, we can solve it as two problems, one with N = 20, M = 2 the other with N = 17, M = 2. In general, any additional constraints of this sort can only make

Phase I easier since it makes the contiguity matrix, B, less dense. Figure 3 shows a solution (run 1) which coincidentally preserves this boundary.

7. Conclusions

Although M (the number of districts) and the contiguity affect solution time, N (the number of population units) seems to be the most significant factor. It appears that problems with $N \leq 30$ can be solved very quickly. Somewhat larger problems $30 \leq N \leq 40$ can be solved with a moderate amount of computer time, but exclusion distances are required to keep Phase I manageable. Somewhere about at $N = 50$, problems become very difficult and require more computer time than we could afford. (A one-hour run failed in West Virginia.) Although computation time increases steeply with N, in the range of problem size we considered, eventually the growth will become approximately linear. This is true because the exclusions make a set of smaller problems out of very large ones. For very large problems storage of the A matrix in core will also become impossible. This should not be a major factor, however, with the availability of high speed external storage devices.

Finally, in light of the recent Supreme Court decisions, proportional representation must be achieved. In most states this will mean redistricting in 1970 (after the next census) and then not until 1980. Traditional redistricting methods having proved unacceptable, it seems clear that one should be willing to expend substantial computer funds every ten years to arrive at solutions. With the availability of such funds, problems much larger than those solved here will become feasible.

Appendix—Reductions

Theorems

Let the column partition of A be $a_1, \cdots, a_S$ and the row partition $r_1, \cdots, r_N$. Let $\emptyset = (0, \cdots, 0)$ and $\mu_n = (0, \cdots, 1, \cdots, 0)$ a unit vector with a 1 in the nth position. In giving the theorems we assume that the entire A matrix has been generated, but as indicated in the algorithms the theorems can be applied to a submatrix as well.

THEOREM 1.[12] *If $r_i = \emptyset$ for any i, (6) cannot be satisfied, and there does not exist a plan.*

THEOREM 2. *If $r_k = \mu_n$ for any k and for any n then $x_n = 1$. Furthermore, any row t for which $a_{tn} = 1$ may be deleted, and any column p for which $a_{tp} = 1$ may also be deleted, in order not to "overcover" row t.*

THEOREM 3. *If $r_k \geq r_t$ for any k and t, then row k may be deleted as well as any column n for which $a_{kn} = 1$ and $a_{tn} = 0$.*

Consider the A matrix

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

Were it not for column 3, row 2 would be greater than row 1 and columns 1, 2 and 5 could be deleted. If district (column) 3 is in the plan we must cover unit (row) 2 with

[12] Proofs for the theorems in this section may be found in [5] and [6].

district 1, 2 or 5. But districts 2 and 5 both have units in common with district 3. Therefore, whether district 3 is in the plan or not, districts 2 and 5 may be deleted. Stating this concept generally, we have Theorem 4.

THEOREM 4. *Assume for any k and t that it is not true that $r_k \geq r_t$ or that $r_t \geq r_k$. Let K be the index set associated with the columns n such that $a_{kn} > a_{tn}$, and T be the set corresponding to $a_{tn} > a_{kn}$. If there exists a row s with $a_{sn} = 1$ for all $n \varepsilon K$ and $a_{sj} = 1$ for some $j \varepsilon T$, column j may be deleted.*

Phase I *Algorithm—Step* VI

Let t denote the rightmost element of V and do the following four tests. If any columns are deleted during these tests start Step VI over again letting each element of V in succession be t. After the tests have been completed on all units of V, find the lowest numbered unit i not in V. Let $C = (i)$. Go to Step II.

Test 1 (Application of Theorem 1). If $r_t = \emptyset$ terminate, since no feasible solution exists.

Test 2 (Application of Theorem 2). If there exists a unit t such that $r_t = \mu_n$, set $D = D \cup (n)$ and $U = U \cup S_n$. For every unit i such that $i \varepsilon S_n$ and i is not in V, let $V = (V, -i)$. Also delete any column j, $j \neq n$, such that $a_{ij} = 1$, $i \varepsilon S_n$. If any columns are deleted, return to the beginning of Step VI.

Test 3 (Application of Theorem 3). If there exist units k, and t, such that $r_t \leq r_k$, delete any column n for which $a_{kn} = 1$ and $a_{tn} = 0$. If unit k is not in V, set $V = (V, -k)$. If any columns are deleted, return to the beginning of Step VI.

Test 4 (Application of Theorem 4).[13] For each unit k, $k \neq t$, find those columns n such that $a_{tn} = 1$ and $a_{kn} = 0$. Call the index set of all such columns G. Let $W = \bigcap_{n \varepsilon G} S_n$. If $G \neq \emptyset$ and if $W \neq (t)$, let Q be the index set of columns p such that $a_{tp} = 0$ and $a_{kp} = 1$. Any $p \varepsilon Q$ such that $S_p \cap W \neq \emptyset$ may be deleted. For any unit $i \varepsilon W$ we may set $z_{ik} = 0$. If any columns are deleted, return to the beginning of Step VI.

Phase II *Algorithm—Step* VII (*Application of Theorem 2*)

We may now be able to apply some of the reduction theorems, since the A matrix has been altered by the deletion of some columns. If any unit is covered only once, by Theorem 2, we fix the district j which covers it, and delete any district which has a unit in common with the fixed district. We may also delete the units of district j. Set $D = D \cup (j)$, $U = U \cup S_j$, $f = f - 1$, $N(D) = N(D) + 1$ and note that T remains unchanged. If any reduction is found repeat Step VII. Otherwise go to Step VIII.

Step VIII (Application of Theorem 3). See if there are two rows, k and t, such that $r_k > r_t$. If there are, by Theorem 3, we may delete all districts which contain unit k and not unit t. Also delete unit k. If any reduction is found go to Step VII. Otherwise go to Step IV.[14]

References

1. BALINSKI, M. L., "Integer Programming: Methods, Uses and Computation," *Management Science*, Vol. 12 (1965) pp. 253–313.
2. —— AND R. E. QUANDT, "On an Integer Program for a Delivery Problem," *Operations Research*, Vol. 12 (1964), pp. 300–304.

[13] Optional in the Computer Program. Test 4 should only be used for large problems.

[14] Theorem 4 could also be applied in Phase II, but since this is a time consuming test and since Phase II is relatively fast, it seems unnecessary.

3. Castellan, J., "Political Apportionment by Computer," *Brown University Computing Review*, Vol. 1 (1967), pp. 5–24.
4. Edmonds, J. and D. R. Fulkerson, "Bottleneck Extrema," Rand Memorandum RM-5375-PR, Jan. 1968.
5. Garfinkel, R. S., "Optimal Political Districting," Ph.D. Thesis, The Johns Hopkins University, Oct. 1968. Also as working Paper #6812, Coll. of Bus. Admin., Univ. of Rochester.
6. —— and G. L. Nemhauser, "The Set-Partitioning Problem: Set Covering with Equality Constraints," *Operations Research*, Vol. 17 (1969), pp. 848–856.
7. Geoffrion, A. M., "Integer Programming by Implicit Enumeration and Balas' Method," *SIAM Review*, Vol. 9 (1967), pp. 178–190.
8. Hess, S., Personal Communication, 1967.
9. ——, "Compactness—What Size and Shape," Workshop on Computer Application to Legislative Districting, 73rd National Conference on Government, National Municipal League (1967).
10. ——, Personal Communication, 1968.
11. ——, J. Weaver, H. Siegfeldt, J. Whelan and P. Zitlau, "Nonpartisan Political Redistricting by Computer," *Operations Research*, Vol. 13 (1965), pp. 993–1006.
12. McKay, R., *Reapportionment*, Twentieth Century Fund, New York, 1965.
13. Report #140, House of Representatives, 89th Congress, "Federal Standards for Congressional Redistricting."
14. Pierce, J. F., "Application of Combinatorial Programming to a Class of All-Zero-One Integer Programming Problems," *Management Science*, Vol. 15 (1968), pp. 191–209.
15. Thoreson, J. and J. Liittschwager, "Legislative Districting by Computer Simulation," *Behavioral Science*, Vol. 12 (1967), pp. 237–247.
16. Wagner, W. H., "An Application of Integer Programming for Legislative Redistricting," Presented to the Thirty-Fourth National Meeting of the Operations Research Society of America, November 1968.