APPLICATIONS OF SET COVERING THEORY TO THE PARTITIONING OF POLITICAL ELECTORAL CONSTITUENCIES

BY

JOSEPH OKEY ELLAH.

Being a Thesis submitted to the Faculty of Science,
University of Durham for the fulfilment of the M.Sc.
degree.

Durham, England

May 1980.

To the memory of my late brother.

ACKNOWLEDGEMENT

I am very grateful to my supervisor Dr. D.M. Greig for her continuous help throughout this work.

ABSTRACT

APPLICATIONS OF SET COVERING THEORY TO THE PARTITIONING OF POLITICAL ELECTORAL CONSTITUENCIES

BY

JOSEPH OKEY ELLAH.

The present work reviews recent computer techniques to the constituency boundary problem. A computer technique based on the set-covering theory is developed and it is shown how the computer results based on the choice of objective can help decision making as regards the optimal plan with respect to equitable apportionment. Data based on the Northern Counties of England was used for the European Assembly Constituency apportionment.

CONTENTS

CHAPTER 0

INTRODUCTION.

0.1

In recent times the birth of democracy and the development of parliaments and parliamentary representation has given rise to the search for techniques that will ensure equitable apportionment with respect to political electoral constituencies.

This work has tried to develop a computer technique that could be used for allocating population units to constituencies in an attempt to achieve equitable apportionment free from human bias. Furthermore techniques on how different objectives could be incorporated into the problem were also developed.

The first chapter of this work is devoted to a coverage of the mathematical techniques used. The second chapter is a survey of other computer techniques that have been developed in an attempt to solve this problem. The third chapter covers the development and application of my approach to the European Assembly Constituencies for the northern counties of England. Different objectives including those that have political considerations are used. The fourth chapter gives the computer results, associated plans, observations and recommendations.

CHAPTER ONE

MATHEMATICAL PROGRAMMING AND THE PARTITIONING OF POLITICAL DISTRICTS.

1.1 General Programming

Generally problems dealing with the maximization or minimization of a function are classified as optimization problems. These problems generally deal with the optimum allocation of some scarce commodities. A general programming problem can therefore be stated as follows:-

maximize or minimize $z = f(x_1, x_2, \text{-----}, x_n)$ (1)

for a set of n variables $x_1, x_2, \text{----}, x_n$ which satisfy m inequalities or equations

$$g_i(x_1, \text{----}, x_n) \left\{ \leq, =, \geq \right\} b_i, \ (i = 1, \text{---}, m) \qquad (2)$$

where the b_i are known or assumed to be known constants while the $g_i(x_1, \text{----}, x_n)$ are assumed to be specified functions.

There are different techniques for solving some special cases of the above functions and I shall cover a few of these techniques later in this chapter.

The general programming problem is divided into different groups according to the specification of the functions as follows, Linear programming and Non Linear programming. These are in turn divided into different classes. I shall restrict myself to the linear programming type because the problem that I solved made use of the linear programming theory.

1.2 Linear programming

This is a type of the general programming problem where all the functions are linear or assumed to be linear.

For example:

When (1) becomes $f(x_1, \text{----}, x_n) = \sum_{j=1}^{n} C_j x_j$ ----- (3)

Where C_j are known constants and $x_j \geqslant 0$ (4)

and (2) becomes $g_i(x_1, ----, x_n) = \sum_{j=1}^{n} a_{ij}x_j \{\leq, =, \geq\} b_i$ (5)

we have a linear programming problem. The exact form depends on the specifications of the functions.

The above linear programming problem could specify a situation where there are n(competing) activities with the

- i representing the number of limited resources;
- j representing the number of activities;
- Z representing the overall measure of effectiveness or penalty for making a particular choice;
- X_j representing the decision variable which specifies the level of activity of j;
- C_j specifies the increase in Z due to a unit increase in j;
- b_i represents the available amount of resource i while a_{ij} stands for the amount of resource i consumed by each unit of activity j; Then $X_j \geqslant 0$ implies that none of these should be operated at a negative level.

A few specific linear programming terminologies are worthy of mention. With reference to the above linear programming problem, (3) is the objective function, (5) are the functional constraints, while (4) stands for the nonnegativity constraints and a_{ij}, b_i, C_j are the parameters.

Some linear programming problems which are of interest in the political partitioning problem are the transportation problem and Integer linear programming problem.

1.3 The transportation problem

This is a special type of the linear programming problem and it is formulated thus:

Determine X_{ij} $(i = 1, \text{-----}m,\ j = 1, \text{-----}, n)$ such as to minimise

$$Z = \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij} X_{ij}$$

such that $\sum_{j=1}^{m} X_{ij} = S_i$

$$\sum_{i=1}^{n} X_{ij} = d_j$$

and $X_{ij} \geq 0,\ \forall_i, j$

$$i = 1, \text{----}, m$$

$$j = 1, \text{----}, n.$$

For this special case, the objective Z is to minimise the total distribution cost; m is the number of sources, n is the number of destinations, S_i is the supply from source i, d_j is the demand at destination j. C_{ij} stands for the cost per unit distribution from source i to destination j; while X_{ij} is the number of units to be distributed from source i to destination j.

It is interesting to note that most linear programming problems can be framed this way irrespective of their physical meaning.

The assignment/allocation problem deserves a brief description.

1.4 The assignment/allocation problem

The objective here is usually to assign some specific supply to a demand point. It is generally stated thus:

Determine X_{ij} $(i = 1, \text{---}, m;\ j = 1, \text{----}, n)$ such as to minimise

$Z = \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij} X_{ij}$ such that

$$\sum_{j=1}^{n} X_{ij} = 1 \qquad i = 1, \text{---}, m$$

$$\sum_{i=1}^{m} X_{ij} = 1 \qquad j = 1, \text{---}, n$$

$$X_{ij} \geq 0 \qquad \forall_i,\ j.$$

For this specific case therefore Z is the total cost and the aim/objective is to minimise the total cost of assigning some supply to a destination that has a demand for one.

1.5 Integer programming

Integer programming problems are special types of the general linear programming problems with an additional constraint which restricts the X_j to integer values only.

In general therefore, integer programming problems amount to finding all

$X_1, X_2, ----, X_n$ such as to minimise/maximise

$$Z = \sum_{j=1}^{n} C_j X_j \text{ ----- } \quad (j = 1, ---, n) \tag{1}$$

subject to

$$\sum a_{ij} X_j \begin{Bmatrix} \leq , =, \geq \end{Bmatrix} b_i \tag{2}$$

provided that

$$X_j = 0, 1, 2, 3, ----, n \quad \text{(integers)} \tag{3}$$

Furthermore if (3) is replaced by

$$X_j \leq 1 \text{ --------(integers)} \tag{4}$$

then the problem becomes a "zero-one" problem. "Zero-one" problems will be covered later in the "set-covering" section.

Integer programming has been widely used in an attempt to solve actual problems since most of the problems that confront us in our everyday life demand integer solutions. In solving the political boundary partitioning problem the theory of 'set-covering' was applied to the problem. "Set-covering" problems are themselves integer programming problems. I shall look at the 'set-covering' theory in detail, but, I think that I should look first at the more general theory of linear programming and then methods of solving linear programming problems restricted to integers before returning to the 'set-covering' theory.

1.6 Theory of linear programming

Let $f(x) = cx$ (1)

be the objective function to be maximised or minimised subject to

$$S = \{x / Ax = b,\ x \geqslant 0\} \quad (2)$$

In this particular situation

A is an mxn matrix,

b is an m - vector

c is an n - vector while

0 is an n - vector of zeroes.

The set S is a convex set since for every x, y $\in$ S imply $\alpha x + (1 - \alpha)y \in S$ for all $0 \leq \alpha \leq 1$. A vector x satisfying $Ax = b$, $x \geqslant 0$ is a solution to the linear problem and a feasible solution in fact.

Given any linear problem there exist three possibilities.

1) No feasible solution exists.

2) There could exist vectors x and y such that for $Cy > 0$; $x + \alpha y$ is a feasible solution for every nonnegative scalar α. In such a case X^o can be made arbitrarily large and thus rendering the problem unbounded.

3) There exists a feasible solution X^o with $\alpha > C\,X^o \geqslant C\,X$ for all feasible solutions of X, X^o offers an optimal solution.

Let us assume that we are dealing with the third case because in the optimization phase of the political electoral constituency problem we shall be meeting the third case mainly and at times the first.

If a feasible solution exists then a basic feasible solution must exist in the first instance. Let A = (B, N) be a permutation of the columns of A in (2) where B = m x m is a non singular matrix, i.e. det. $B \neq 0$, then the matrix B is a basis matrix.

Let $X = (X_B, X_N)$ where X_B = vector of basic variables associated with the columns of B, and X_N = vector of non-basic variables associated with the columns of N. Then AX = b -----(2) could be written as

$$BX_B + NX_N = b \quad (3)$$

Since B is non-singular, then B^{-1} exists

Hence

$$X_B = B^{-1}b - B^{-1} NX_N \quad (4)$$

In particular

$$X_B = B^{-1}b, \ X_N = 0 \quad (5)$$

X_B is therefore a basic solution and if furthermore $X_B \geqslant 0$, then X_B is a basic feasible solution. If the linear programming problem has an optimal solution, it has a basic optimal solution and since there exist at most $\binom{n}{m}$ bases one can therefore arrive at the basic optimal solution of the convex set by solving these. Also a feasible solution corresponds to an extreme point if and only if it is basic. For $S = \{ x/Ax = b, x \geqslant 0$, a point $x \in S$ is an extreme point of the convex set if $\nexists$ distinct $y_1, y_2 \in S$ and a scalar α, $0 < \alpha < 1$ such that $x = \alpha y_1 + (1 - \alpha)y_2$.

In the absence of degeneracy a basic feasible solution can certainly be improved as we move from one extreme point of the convex set S to another until we obtain the optimal solution, and techniques are available also for the degenerate case.

1.7 How to improve a basic feasible solution

Let

$$C = (C_B, C_N) \text{ and hence } X_o = C_B X_B + C_N X_N \quad (6)$$

and using (4) to eliminate X_B in (6) we have

$$X_o = C_B B^{-1} b - (C_B B^{-1}N - C_N) X_N \qquad (7)$$

Let $X_N = 0$, then (7) ---$X_o = C_B B^{-1} b$ and the basic feasible solution given by (4), (5) and (7) are therefore

$$\begin{bmatrix} X_o \\ X_B \end{bmatrix} = \begin{bmatrix} C_B B^{-1} b \\ B^{-1} b \end{bmatrix} - \begin{bmatrix} C_B B^{-1} b - C_N \\ B_N^{-1} \end{bmatrix} X_N$$

Letting $X_o \equiv X_{Bo}$, $X_B \equiv (X_{B1}, \ldots\ldots, X_{B_m})$ and R the index set of columns N.

Also let

$$Y_o \equiv \begin{bmatrix} C_B B^{-1} b \\ B^{-1} b \end{bmatrix} \equiv \begin{bmatrix} Y_{00} \\ Y_{10} \\ \vdots \\ Y_{MO} \end{bmatrix} \quad \text{and}$$

if $j \in R$ such that a_j is a column of N, then

$$Y_i \equiv \begin{bmatrix} C_B B^{-1} a_j - C_j \\ B^{-1} a_j \end{bmatrix} \equiv \begin{bmatrix} Y_{ij} \\ Y_{ij} \\ \vdots \\ Y_{mj} \end{bmatrix}$$

Then (4) and (7) can be written as

$$X_{Bi} = Y_{io} - \sum_{j \in R} Y_{ij} X_j, \qquad (8)$$
$$i = 0, 1, \ldots, m$$

Setting $X_j = 0 \; \forall_{j \in R}$ then solution (5) is obtained. Let X_B be non-degenerate and $Y_{oj} < 0$ for some $j \in R$, say $j = K$, then by increasing X_K while keeping all other non-basic variables fixed at zero. It is noticed that X_o increases linearly with slope $-Y_{OK}$. And X_{Bi} is a linear function of X_K with slope $-Y_{iK}$. If $Y_{iK} > 0$, then $X_{Bi} \geqslant 0$ and this holds as long as $X_K \leq Y_{io}/Y_{ik} = \theta_{ik}$. Note that when $X_K = \theta_{ik}$, $X_{Bi} = 0$.

Finding a new basic feasible solution corresponds to solving the γ^{th} equation (8) for X_K to obtain

$$X_K = \frac{Y_{ro}}{Y_{rk}} - \sum_{j \in R - \{K\}} \left(\frac{Y_{rj}}{Y_{rk}} \right) X_j - \frac{1}{Y_{rk}} X_{Br} \tag{9}$$

and eliminating X_K from other equations of (8) to obtain

$$X_{Bi} = Y_{io} - \frac{Y_{ik} Y_{ro}}{Y_{rk}} - \sum_{J \in R - \{K\}} \left(Y_{ij} - \frac{Y_{ik} Y_{rj}}{Y_{rk}} \right) X_j + \frac{Y_{ik}}{Y_{rk}} X_{Br} \tag{10}$$

where $i = r$

Setting $X_{Br} = 0$ and $X_j = 0$ for all $J \in R - \{K\}$ so that whenever $\exists$ a non-degenerate basic feasible solution with $Y_{oj} < 0$, for some $J \in R$, say $j = M$ and $Y_{im} > 0$ for at least one i, then a basic feasible solution can be found by exchanging one column of N for one of B.

Let B = a basis matrix with a basic feasible solution X^o with $Y_{oj} \geqslant 0$ $\forall$ $J \in R$, then from (8) the objective function becomes

$$X_o = C_B B^{-1} b - \sum_{J \in R} Y_{oj} X_j \tag{11}$$

With the absence of X^o_B in (11), $Y_{oj} \geqslant 0$ and $X_j \geqslant 0$ $_{J \in R}$, hence $C_B B^{-1} b$ is an upper bound on X_o. Since $X^o_B = B^{-1} b$ and $X^o_N = 0$ is feasible and achieves the upper bound hence the solution is optimal. Thus a basis solution is optimal if (i) $Y_{io} \geqslant 0$, $(i = 1, ----, m)$ is feasible and $Y_{oj} \geqslant 0$ $\forall J \in R$.

These procedures are formalised in what is described as the Simplex Algorithm. (See T.C. HU (1969)[1], Hadley, G (1969)[2]).

It is interesting to note that every linear program has its dual and both of them have some characteristics that are worthy of note.

1.8 The dual concept

Gale, Kuhn and Tucker (1971)[3] said the following about the dual problem.

> "It turns out that for each linear programme, there is another linear programme associated with it, such that the two linear programmes have many interesting relations. We call the first linear program, the primal linear program, and the second, the dual linear program."

Let us have the following as a linear problem.

$$\text{Min } Z = CX$$

subject to $AX \geq b$

$X \geq 0$ the dual would be

$$\max W = Yb$$

Subject to $YA \leq C$

$Y \geq 0$

or stated in another way, let the following be the primal linear problem:

$$\text{Min } Z = CX$$

subject to

$$AX = b$$

$$X \geq 0$$

The dual would be the following

$$\text{Max } W = \pi b$$

$$\pi A \leq C$$

$$\pi \gtrless 0.$$

Also the following relationships exist for any primal and its dual.

(1) Both the Primal and the dual problems have optimal solutions.

(2) The dual is unbounded and the Primal is infeasible.

(3) The Primal is unbounded and the dual is infeasible.

(4) The Primal and the dual are both infeasible. Garfinkel, R.S. and Nemhauser (1972)[4] give a proof of these relationships.

For a standard linear programming problem which is framed thus:

Maximise $Z = \sum_{j=1}^{n} C_j X_j$ subject to

$$\sum_{j=1}^{n} a_{ij} X_j \leq b_i$$

and

$$X_j \geqslant 0$$

$i = 1, ----, m$

$j = 1, ----, n.$

The dual would amount to finding all values of Y_i, ----, Y_m such as to

minimise $Y_o = \sum_{i-1}^{m} b_i y_i$

Subject to

$$\sum_{i-1}^{m} a_{ij} Y_i \geqslant C_j$$

and

$$Y_i \geqslant 0$$

$i = 1, -----, m$

$j = 1, ---, n$

Despite the fact that the formulation of the dual problem arises during the computation period, it has a geometric interpretation as well as a much more interesting economic interpretation. The dual variables Y_i ($i = 1, -----, m$) would represent the current unit contribution of all resources that would perhaps be consumed by one unit of activity j.

$Y_o = \sum_{i=1}^{m} b_i Y_i$ would represent the total implied value of these resources consumed by the different activities while the constraints

$\sum_{i=1}^{m} a_{ij} Y_i \geqslant C_j$, ($j = 1, -----, n$) would indicate that the contribution of the resources to the criterion of effectiveness must be at least equal to the unit contribution of activity j.

The methods for solving these are similar to those of the Primal except that while in the simplex we decide about which vector to enter the basis first, in the dual we rather decide about which vector should leave the basis first.

A summary of solution techniques to Integer linear programming problems will be given. One can obtain details about the solution techniques in Beal, E.M.L. (1954)[5]; Balinski, M.L. (1965)[6]; Dantzig, G.B. (1960)[7].

1.9 Methods of solving Integer linear programming problems

There are three main approaches to solving integer linear programming problems. They are as follows; Implicit Enumeration approach, Branch and Bound procedure and Gomory's Cutting plane method.

1.9.1 Gomory's cutting plane method

Given an integer linear programming problem say,

$$\max X_o = a_{oo} - a_{o_1} X_1 - a_{o_2} X_2 - \dots - a_{on} X_n$$

Subject to

$$X_{n+1} = a_{n+1,0} - a_{n+1,1} X_1 - a_{n+2,2} X_2 - \dots - a_{n+2,n} X_n$$

$$\vdots$$

$$X_{n+m} = a_{n+m,o} - a_{n+m,1} X_1 - a_{n+m,2} X_2 - \dots - a_{n+m,n} X_n$$

$$X_j \geqslant 0 \quad (j = 1, \dots, n, n+1, \dots, n+m)$$

$$X_j \ (j = 1, \dots, n) \text{ integers} \qquad (1)$$

Here $X_{n+1}, \dots, X_{n+m}$ are the slack variables while $X_1, \dots, X_n$ are the original variables of (1). The above problem can be solved by Gomory via either the simplex technique or the dual simplex technique and if the results are restricted to integers, then the solution has been found and the optimal solution has been reached but if otherwise, then extra constraints known as Gormory's Cuts are inserted to generate strictly integer solutions.

'Gomory's Cuts' are derived as follows:

Rewriting (1) in tableau form we have

$$
\begin{array}{rc|cccc|}
\hline
X_o & = & a_{oo} & a_{o1} & --- & a_{on} \\
X_1 & = & 0 & -1 & & \\
\vdots & & \vdots & & \ddots & \\
X_n & = & 0 & & & -1 \\
\hline
X_n+1 & = & a_{n+1,0} & a_{n+1,1} & & a_{n+1,n} \\
\vdots & & \vdots & & & \\
X_{n+m} & = & a_{n+m,1} & a_{n+m,1} & ------ & a_{n+m,n} \\
\hline
\end{array}
\qquad (2)
$$

Then go through the following procedure;

1) Search through the final simplex tableau or dual simplex tableau and select a variable which produced a non integer solution.

2) Examine the row that specifies the non-integer solution value for that variable, then replace each coefficient in that row by the smallest possible positive number which is congruent to that coefficient.

3) Set the resulting expression greater than or equal to the fractional part of the constant of that row and add this to the tableau, then invoke the simplex or dual simplex again.

The rule therefore is to find a λ large enough to produce a pivote of -1 and at the same time give the biggest decrease in a_o which should be the lexicographically smallest column. The process thus goes as follows:

Step I: Let V = Source row

Step II: Let α_s = the lexicographically smallest column with $a_{vj} < 0$

Step III: For each $a_{vj} < 0$, let μ_j be the largest integer such that $\alpha_s < \left(\frac{\alpha_j}{\mu_j}\right)$

Step IV: Let $\left[\frac{-a_{vj}}{\lambda_j}\right] = \mu_j$ or $\lambda_j = \frac{-a_{vj}}{\mu_j}$ (row V is the source row).

This λ_j is the λ that will make $\alpha_s < \alpha_j / \left[{}^{-a_{vj}}/\lambda_j\right]$

Step V: Let $\lambda = \max_j \lambda_j$ for $a_{vj} < 0$. Hence the selection of λ is to make the pivot -1, keep the tableau dual feasible and cause the greatest decrease lexicographically in the 0^{th} column.

The choice of λ min of λ is very important. Hence assuming the pivot column has been determined by choosing the column which is lexicographically smallest with $a_{vj} < 0$. Then the cut is determined by constructing a cut from an equation by taking it modulo a number determined by the condition that the resulting pivot element is -1. Assuming the source row is $x = a_o + a_j(-x_j)$ ---------------------(3), whether a_o and a_j are real numbers (or integers), the Gemory Cut is

$$S = \left[{}^{a_o}/\lambda\right] + \Sigma \left[{}^{a_j}/\lambda\right] (-x_j^t)$$

where 't' stands for the tableau in question. The result will be that we shall have interger coefficients with pivot -1, and the first n+1 rows will be integers; and S a non-negative integer variable and $\lambda > 1$.

The choice of λ therefore decreases or increases the strength of the cut.

One of the optimisation techniques used in the phase 2 of my program employed the so called Wilson's Cut which could be stronger than the Gomory's Cut, but certainly not weaker.

Wilson, R.B. (1966)[8] said the following about his cut:

> "Stronger cuts are available and easily calculated, than those obtained by Gomory in the construction of his all - integer integer programming algorithm".

1.9.2 Wilson's Cut

To derive Wilson's stronger cuts let us consider the following:

$$S_1 = a_o + \sum_{j=1}^{n} a_j(-X_j), \quad a_o < 0$$

and let it be the equation from which the cut is to be derived, then a stronger choice of λ would be

$$\lambda = \lambda^* = \max(\lambda_{min}, \min_j \in J \lambda^*)$$

where

$$\lambda^* = a_o / (1 + [a_o/\lambda_{min}]) - \varepsilon \quad \text{and for} \quad j \geq 1$$

$$\lambda_j^* = a_j / [a_j/\lambda_{min}].$$ If all $a_j < 0$, so that $\sim J$ is empty,

and $[a_o/\lambda_{min}] = -1$, then λ^* may be taken indefinitely large.

To illustrate the strength of this cut consider the following example given by Wilson, R.B. (1966)[9]

		$-X_1$	$-X_2$	$-X_3$	$-X_4$
Z =	20	1	2	3	4
S_1 =	-20	-7	-8	-15	18

X_1 - Column is the pivote column since it is the lexicographically smallest column from among those in set J with negative entries. Hence $J = 1, 2, 3$.

For Gomory the cut would have been considered thus:

$$\mu_1 = 1, \quad \mu_2 = 2, \quad \mu_3 = 3,$$

$$\lambda_1 = 7, \quad \lambda_2 = 4, \quad \lambda_3 = 5$$

Hence

$$\lambda_{min} = \max_{j \in J} \lambda = 7.$$

Thus, the cut is

$$S = [^{-20}/_7] \quad [^{-7}/_7] \quad [^{-8}/_7] \quad [^{-15}/_7] \quad [^{18}/_7],$$

hence

$$S_2 = -3 + X_1 + 2X_2 + 3X_3 - 2X_4$$

Thus

$$S_2^* = -3 \quad -1 \quad -2 \quad -3 \quad 2$$

But Wilson's Cut would have been considered thus

$$\lambda^* = \max (\lambda_{min}, \min_{j \in J} \lambda^*)$$

where

$$\lambda_o^* = {}^a o/(1 + [{}^a o/\lambda_{min}]) - \varepsilon = \min(9, 10 - \varepsilon)$$

and for

$$J \geqslant 1, \lambda_j^* = a_j/[{}^a j/\lambda_{min}] = 9;$$

Thus making

$$S_2^{**} = -3 \quad -1 \quad -1 \quad -2 \quad 2 = X_1 + X_2 + 2X_3 - 2X_4 \geqslant 3$$

which is stronger than S_2^* by $X_2 + X_3$.

Hence

$$S_2^{**} = [S_2^* - (X_2 + X_3)]$$

From my experience the use of Wilson's Cut greatly enhanced the all integer algorithm and it facilitated the running time for the phase 2 of the algorithm that will be presented in chapter 3. For a detailed analysis of this cut, see Wilson, R.B. (1966)[10].
Greenberg, H. (1971)[11] and Langmaack, H. (1965)[12] have a good coverage of Gomory's cutting plane method.

It may also be possible to bring about early convergence by combining the constraints using methods of e.g. Surrogate Constraints and Aggregate Constraints. Garfinkel, R.S. and Nemhauser (1971)[13], Taha, H.A, (1976)[14] and Plane, D.R; and McMillan, C; Jrn (1971)[15] have a detailed analysis of them.

I shall now look at some of the other methods of solving integer linear programming problems.

1.9.3 Enumeration Techniques

Enumeration techniques take advantage of the fact that in a bounded integer linear programming problem or in a mixed integer linear programming problem, the set of values of the integer variables is finite and the task is therefore to find all such values. Usually there are 2^n solutions. The methods employed to solve problems via the enumeration procedure are designed to limit the enumeration process and converge to the optimal solution without going through all the 2^n solutions.

Enumeration techniques are well covered in most mathematical programming books but I shall endeavour to summarise some of them; for details see Garfinkel, R.S. and Nemhauser, G.L. (1972)[16].

1.9.4 Theory of enumeration

Given that

$$\sum_{j=1}^{n} jXj = B$$

Subject to

$$X_j = \begin{cases} 1 - \text{If } j \text{ is one of the integers allowed} \\ 0 - \text{otherwise.} \end{cases}$$

and

$$X_j = 0, 1 \text{ for all } j.$$

The solution to the above are given by unique paths from Vertex $O(V_o)$ to each of the vertices of the enumeration tree $(V_o, \ldots, V_n)$. Here, each edge imposes a constraint while each vertex j represents the constraint solution j.

This is usually illustrated using a tree, while the search for the solution is illustrated by trying out all solutions from the vertex taking one path at a time.

If for example, we are to find all $X \in S$, then vertex j restricts X to S_j, where S_j is the intersection of the set of points satisfying the constraints given by the edges of $P_{(j)}$.

If $P_{(j)}$ has K + 1 vertices denoted by

$$V_o = V_{j(0)}, V_{j(1)} \ldots\ldots, V_{j(K)} = V_j$$

Then

$$S = S_{j(0)} \supseteq S_{j(1)} \supseteq \ldots\ldots \supseteq S_{j(K)} = S_j.$$

It is noted that $V_{j(K)}$ is the predecessor of V_j and V_j is the successor of its predecessor. A vertex would therefore have a unique predecessor but would generally have more than one successor.

The constraints of the edges from V_j to its successors determine a finite path say S_j^* of subset S_j such that $\underset{T \in S_j}{\cup T} = S_j$. The set S_j^* is referred to as a separation of S_j. Each of the $|S_j^*|$ edges emanating from any V_j corresponds to a constraint restricting X to only one of the elements of S_j^*. In many cases S_j^* is a partition of S_j.

A vertex that has not been fathomed and whose corresponding constraints have not been separated is regarded as a live vertex and it is usually to these live vertices that we divert while 'branching' in search of a solution or a better solution. In most cases 'branching' is made to one of the successor vertices of the vertex currently being considered.

'Bounding' is accomplished by checking out the objective function value for any 'branching' made. Where the current vertex has been fathomed; that is, all necessary completions have been made then we 'backtrack' along $P_{(j)}$ until a live vertex is encountered. Where none exists, the enumeration terminates.

To accelerate enumeration some interesting techniques are employed. For example, fathoming a vertex by bound. To illustrate this; assume we are at vertex j; so that the problem is as follows.

$$\text{Max } Z(x),\ X \in S_{(j)} \qquad (1)$$

$$Z_j^* = \begin{cases} Z(X^*_{(j)}) & \text{If } X^*_{(j)} \text{ solves (1)} \\ -\infty & \text{If } S_{(j)} = \emptyset \\ \infty & \text{If (1) is unbounded.} \end{cases}$$

By relaxing (1) an upper bound $\bar{Z}_j \geqslant Z_j^*$ could be considered, say

$$\text{Max } Z(x),\ X \in T_j \supseteq S_j \qquad (2)$$

$$Z_j^{**} = \begin{cases} \infty & \text{If S is unbounded} \\ -\infty & \text{If } T_j = \emptyset \\ Z_j^o = Z(X^o_{(j)}) & \text{If } X^o_{(j)} \text{ solves (2).} \end{cases}$$

Note that an upper bound at a vertex is also valid for the successors, since if V_K is a successor of V_j, then $T_j \supseteq S_j \supseteq S_k$.

$$S_j = \{X / A^j x = b^j,\ X \geqslant 0 \text{ integer}\}$$

$$T_j = \{X / A^j x = b^j,\ X \geqslant 0\},$$ $\bar{Z}_j$ is thus calculated by solving the corresponding linear programme. To calculate lower bounds; suppose for example Z_j satisfies $Z_j \leqslant Z_j^*$, then find $x^1 \in S_j$ and let $\underline{Z}_j = Z(x^1)$. If V_k is a predecessor of V_j, then $\underline{Z}_j \leqslant Z_k^*$ which yields $\underline{Z}_j \leqslant Z_o^*$. A vertex is therefore fathomed by bounds if:

(a) $\bar{Z}_j = \underline{Z}_j \Longleftrightarrow$ No better solution exists.

(b) $\bar{Z}_j \leqslant \underline{Z}^o \longrightarrow$ No successor of V_j can yield a solution that improves on the best solution.

set F_k, completion is accomplished.

The partition $S_k^* = \left\{ S_k \cap \{x/x_j = 0\}, \; S_k \cap \{x/x_j = 1\} \right\}$ satisfies $S_k \in S_j^* \Rightarrow |H_k| < |H_j|$, thus guaranteeing finiteness. For details about finiteness see, HU, T.C. (1969)[17].

'Bounding is accomplished as follows:

Consider vertex V_k, and consider the problem at V_k.

$$\text{Max } Z_k = \sum_{j \in F_k} C_j X_j + \sum_{j \in S_k} C_j$$

where

$$\sum_{S=F_k} a_{ij} X_j \leq b_i - \sum_{j \in S_k} a_{ij} = S_i \qquad (1)$$

$$i = 1, \ldots, m.$$

$$X_j = 0, 1$$

$$j \in F_k.$$

Let $T_k = H_k$, since $C_j \leq 0$, X_k^o is obtained by setting $X_j = 0$, $j \in F_k$; thus $\bar{Z}_k = Z_k^o = \sum_{j \in S_k^+} C_j$.

If $S = (S_1, \ldots, S_m) \geq 0$, then X_k^o is feasible to (1) and $\underline{Z}_k = Z_k^o$.

Fathoming is accomplished by

(a) $\bar{Z}_k = \underline{Z}_k \Longrightarrow X_k^o$ is feasible to (1)

(b) $\bar{Z}_k \leq Z_o \Longrightarrow$ A sufficient condition for this is as follows;

Suppose for some $t_i = \sum_{j \in F_k} \min\{0, a_{ij}\} > S_i$ in which case no completion of W_k can satisfy i. Hence $\bar{Z}_k = -\infty$ and V_k is fathomed. See Garfinkel, R.S. and Nemhauser, G.L. (1972)[18] for a discussion of the techniques for choosing partitioning variables and branching.

1.9.6 Set covering problem

The set covering problem is a class of binary problems that requires that one searches for the minimum number of edges that would cover all the nodes on a graph. For the purpose of this topic, nodes stand for points, while edges stand for connecting lines.

Generally the set covering problem is stated thus:

Consider a set, $I = \{1, \ldots\ldots, m\}$ and

a set $P = \{P_1, \ldots\ldots, P_n\}$ where

$P_j \subseteq I$, and for $j \in J = \{1, \ldots\ldots, n\}$;

a subset $J^* \subseteq J$ defines a cover of I, if

$$\bigcup_{j \in J^*} P_j = I \qquad (1)$$

and $$j, K \in J^*, \; j \neq K \Longrightarrow P_j \cap P_k = \phi \qquad (2)$$

J^* defines a partition of I or a cover for I. Now let $C_j > 0$ be associated with every $j \in J$, then the total "cost" of the cover $J^* = \sum_{j \in J} C_j$.

The problem therefore would be to find a cover of minimum cost and this can be translated into a linear programming problem as follows:

$$\text{Min } X_o = \sum_{j=1}^{n} C_j X_j \qquad (3)$$

Subject to the following:

$$\sum_{j=1}^{n} a_{ij} X_j \geqslant 1 \qquad (4)$$

$$X_j = 0, 1 \qquad (5)$$

$i = 1, \ldots, m$

$j = 1, \ldots\ldots, n$:

where

$$X_j = \begin{cases} 1 & \text{If } j \text{ is in the cover} \\ 0 & \text{otherwise} \end{cases}$$

and

$$a_{ij} = \begin{cases} 1 & \text{If } i \in P_j \\ 0 & \text{otherwise.} \end{cases}$$

The above can be reduced to a set partitioning problem thus: instead of (4) we now have

$$\sum_{j=1}^{n} a_{ij} X_j = 1 \qquad (6$$

Any X satisfying (4) and (5) or (5) and (6) is a solution (cover /partition).

Any problem in the above set-up can then be solved via implicit enumeration discussed in section 1.9.5 or via Gomory's cutting plane method discussed in section 1.9.1.

This can be viewed with respect to the political parliamentary constituency boundary problem as follows:

Consider a set q_i, i = 1, ..., m to be indivisible population units which are to be grouped into "K" districts. Given (1) matrix "a_{ij}", defining acceptable group (2) "Cost" for each group and (3) an objective function related to (2); then the problem could be framed into a linear programming problem as follows:

$$\text{Min} \sum C_j X_j \qquad (1)$$

where $C_j = \left| P_j - \bar{p} \right|$, defines the acceptability of a district.

Subject to

$$\sum_{j=1}^{n} a_{ij} X_j = 1 \qquad (2)$$

$$\sum_{j=1}^{n} X_j = K \qquad (3)$$

$$j = 1, \ldots, S \quad i = 1, \ldots, n$$

$$X_j = \begin{cases} 1 & \text{if district } j \text{ is accepted} \\ 0 & \text{otherwise} \end{cases}$$

Set covering theory has many applications and for details of these see Geoffrion, A.M. (1971)[19]. For a detailed discussion of zero-one problems see Glover, F. (1965)[20].

The mathematical discussions in this chapter will be constantly referred to in chapter 3. At this point it is necessary to survey other approaches to computer techniques in political boundary problems and this will be done in the next chapter.

CHAPTER TWO

ELECTORAL POLITICAL BOUNDARY PROBLEMS

2.1 Origin of problem

For many decades since the birth of democracy and the subsequent acceptance of the equality of man; the principle of one-man one-vote has been incorporated into the system and the world's nations have been searching for ways to satisfy this ideal. Many states therefore face lots of difficulties in an attempt to produce an equitable electoral plan for their parliamentary elections as well as local government elections. This difficulty arises because there are literally thousands of ways in which a country can be subdivided for electoral purposes. Partisan bias helps to compound the problem and makes it impossible in most cases to decide on the most equitable electoral plan for a political area.

There has nevertheless been a constant quest for equitable electoral plans. Countries like Britain, and some European countries appoint non-partisan boundary commissioners to help draw up an equitable plan for the nation. These boundary commissioners seldom succeed in their duties because of all the influences that they would usually be subjected to apart from their personal political bias. Thus, the plans produced by these commissioners are usually subjected to disputes and representations as a result of malapportionment, and it would normally take many months and at times years to produce an acceptable plan where possible. In quite a large number of cases no acceptable plan is reached, the nation would normally accept and abide by whatever plan the ruling party is satisfied with. As a result plans keep changing with each change of government. This led Vickrey, V. (1961)[21] in his paper entitled, "On the Prevention of Gerrymandering", to propose the use of an automatic and impersonal procedure for drawing up constituency boundaries. By 'automatic' and 'impersonal procedure' he precisely means 'the computer'. In agreement with him, Forrest, E. (1965)[22], maintains that:

> "since the computer doesn't know how to gerrymander ....the electronically generated map cannot be anything but unbiased".

I shall at this point define some terms which will be constantly used throughout this work. They are as follows:

a) UNIT: This implies the smallest indivisible part of a population. Each unit has a position P_i, $\underline{x}_i$, corresponds to the position of unit P_i, $i = 1, \ldots, n$.

b) GROUP: A 'group' means a set of $P_{i's}$ making up a constituency or legislative district (Grouping).

c) PLAN: An electoral plan defines a Group for each P.

2.2. Necessity for a computer technique

A few people have used computer techniques to generate group plans. Some of the well known techniques are those by the following people: Weaver, J.B. and Hess, S.W. (1963)[23] and their group of 'civic-minded engineers' known as the 'CROND. INC. (Computer Research In Non-Partisan Districting) have developed a technique which accomplishes this task by measuring the acceptability of a group on its compactness measure.

Thoreson, J. and Liittschwager, J. (1967)[24], have developed a heuristic approach in which computer simulation techniques are used for forming electoral groups.

Garfinkel, R. (1968)[25], has a computerised 'tree-search' approach to this problem.

G.Mills (1967)[26], used a heuristic based on Weaver's technique and applied this to a British local Government Area - Bristol County Borough.

Wagner, W.H. (1968)[27], used an integer programming technique for solving the political grouping problem.

From the short list above, it is certain that the problem of grouping units together for electoral purposes has been approached from different dimensions as a result of its necessity. We must all appreciate the fact that the possibility of having a fair government depends almost entirely on the type of group plan a country has. In fact nearly all civil wars have as a direct cause or a remote cause political malapportionment, which is the result of accepting to choose a biased group plan.

I shall analyse some of the work done by the above named computer pioneers in electoral grouping but I shall first systematically survey the general consideration in an electoral grouping excerise.

2.3 General considerations

There are a number of criteria that have to be considered in the excercise of partitioning a political entity into electoral groups. They are as follows; population equality, contiguity, compactness, homogeneity, state law, singularity of representation, preservation of political boundaries and natural geographical features.

2.3.1 Population equality

Electoral groups are required in practically every case to be of equal or nearly equal population. This is in fact the most important consideration because it in effect preserves the one-man one-vote principle which is implied in the acceptance of democracy as a system of government.

Population equality is defined slightly differently by different parliaments and a few of the common definitions are as follows:

(a) 'Electoral Quota'. Some parliaments, like the British House of Commons define 'electoral quota' to mean the number derived by dividing the 'electorate' by the number of available seats. 'Electorate', in this sense means the list of registered voters.

Population equality in this case would then demand that the total population of a group must be as near as possible to the electoral quota.

(b) 'Population Variance Ratio'. Some states in the United States define 'population variance ratio' to be the ratio of the largest population to the smallest population per representative. The smaller the ratio, the nearer the electoral groups are to absolute equality.

(c) 'Minimum Percentage Test'. In an attempt to achieve population equality some states try to determine the minimum percentage of the state's citizens that reside in an electoral group electing a controlling majority. The electoral groups are ranked on the basis of their population per representative from smallest to largest. The population of each successively larger group is added up until a majority of legislators is accounted for and at that point the summed population is divided by the state's total population giving the figure for the minimum population from which the minimum percentage is derived.

The analysis above has population as its central theme, yet different measurements of population exist and the pattern of representation can be affected greatly by the population measure chosen. The following therefore are some of the available population measurements:

i) Total Population.

ii) Population of voting age.

iii) Population excluding aliens.

iv) Population of registered voters, etc.

The first step in this excercise would then be to choose the exact population measurement to use, then the population equality desired is defined and finally a decision as to the 'allowable percentage deviation' from 'perfect equality' is taken. The final step is usually not fixed by law but a few suggestions have been made in this regard. The American Political Science Committee (1951)[28] recommended 10% and stated further that on no account should it be more than 15%.

2.3.2 Contiguity

In order to preserve territorial continuity in electoral groups it is desirable to generate an electoral plan whose groups are made up of contiguous units. Although the concept of 'best' contiguity is absent, yet an electoral plan would be more acceptable and feasible if the principle of equal population is combined with that of territorial continuity.

A group is contiguous, if it is possible to travel between any two locations within the group without leaving its boundaries, hence movement within a group without crossing the boundaries of another group is guaranteed.

This criterion is very necessary for campaign purposes and for the allocation of resources in terms of electoral groups.

Contiguity can be rigorously defined as follows: let $B = \{b_{ik}\}$ be a symmetric N X N matrix, where $b_{ik} \begin{cases} = 1, & \text{if units of } i \text{ and } k \text{ have a common boundary,} \\ = 0 & \text{otherwise.} \end{cases}$

Let us consider an electoral group as an undirected graph with the vertices as the units of the group, an arc exists between i and k if and only if $b_{ik} = 1$. A group is contiguous if the graph is connected: that is to say that a path exists between every pair of vertices.

2.3.3 Compactness

Geographically, compactness means being closely united so as to economize space. Mathematically it could be conceived for example as requiring the maximization of the ratio of a group's area to its perimeter.

If the above are the accepted definitions then a group would be more desirable if it was circular or square in shape.

The idea of compactness in the context of an electoral plan should be seen as the desire to create electoral groups that reflect at least to some extent, popular interest patterns, since in fact a representative should be seen as someone representing the interests of those who elected him and to whose class he should be identified with.

Compactness defined as a measure of population as well as Geographical Concentration would be preferable in this sense.

Mathematically compactness of a group composed of units P_i at $\underline{X}_i$ could be defined as follows:

Assuming that the distance between the centres of units i and K is $d(i,K) = |\underline{X}_i - \underline{X}_K|$. For each pair of units in a group, let e(i,K) = exclusion distance, then group j is feasible if $d(i, K) > e(i, K) \Rightarrow a_{ij} \cdot a_{Kj} = 0$. For a particular group, say j, let $d_j = \max \{d(i, K), a_{ij} \cdot a_{Kj}\}$ (i, K = 1, ..., N where there exists n units in group j) be defined as the distance between units of j which are farthest apart. Then $d_j = 0$ if group j contains only one unit. Let A'_j = the area of group j. It follows that $C_j = d^2_j / A'_j$ would then be a dimensionless measure of group j. As C_j decreases, group j becomes more compact. As stated above, if $C_j = 0$ then group j defines a single unit. There has been no generally accepted definition of compactness.

Weaver, J.B. and Hess, S.W. (1963)[29] in an attempt to combine the mathematical and geographical definitions of compactness, stated that "moment of inertia provides a possible measure of compactness in legislative districting, involving both area and population".

To illustrate their definition, consider the following: let figure one represent an electoral group. By dividing the group into thirty six rectangular blocks, it would be possible to make calculations similar to moment of inertia about any point in the plane of the figure say X. For each block the moment of inertia would be the product of the block's population times the square of the distance between the block and that point. The moment of inertia about point X for the whole group would be the summed moments of intertia of all the blocks. This sum would be smallest if X corresponds to the population centre of the group, that is, 'centre of gravity' of the population. For computational purposes, let moment of inertia about point X be defined as the weight of say $S_1 = W_1$, times A_1^2, where A_1 = the distance between S_1 and X = $W_1A_1^2$. Hence the moment of inertia about X for the whole group is $(W_1A_1^2 + W_2A_2^2 + \ldots + W_{36}A_{36}^2) = \sum_{i=1}^{36} W_iA_i^2$ = Population moment of inertia at X.

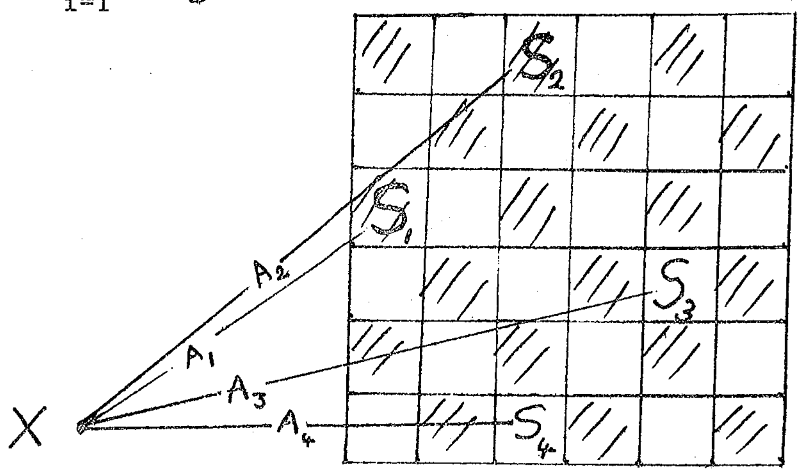


fig (1)

Hence, compactness which is a measure of diffuseness is thus the sum of the squared distances of each voter from the centre.

A comparison of the summed moments of inertia for different electoral plans would enable one to determine the plan that is most compact.

A heuristic based on this technique but mostly used for warehouse location problems was used by Weaver, J.B. and Hess, S.W. (1963)[30] for producing an electoral plan.

Harris, C.C. (1964)[31] looks at compactness slightly differently. He would rather think of it in terms of the length and width of an electoral group. In order to generate a compact electoral plan he decided to minimize the difference between the lengths and widths of all the electoral groups under consideration.

Hence,

$$\text{Min} \sum_{i=1}^{n} \left| L_i - W_i \right|$$

n = number of groups
L_i = max. length of group i
W_i = max. width of group i.

He initially generated all the possible groups somehow before minimizing as above.

Cellar, E. saw compactness differently. He rather considered it as a measure of the quotient of the length and the width. Hence he defined compactness as follows:

$$\sum_{i=1}^{n} \left(\frac{L_i}{W_i} \right)$$

n = number of groups
L_i = max. length of group i
W_i = max. width of W_i.

He tried to minimize the above in order to determine the most compact plan.

The American, House of representatives in Report No. 140 of the 89th Congress in discussing compactness in their "Federal standards for Congressional Redistricting" defined compactness as the absence of gerrymandering.

From the above discussion, it is clear that compactness as a criterion for electoral planning is very desirable yet its lack of a vivid and an exact definition renders its exact application highly impracticable. Compactness is nevertheless, important but certainly less important than population equality and contiguity.

2.3.4 Homogeneity

In consideration of the economic and social interests of a community one is in effect considering their homogeneity. Homogeneous groups therefore preserve the communities economic interests. A representative is expected to represent the social as well as the economic interests of his electorate. This task is only possible if the representative is representing an electorate with similar interests but if his electorate is a conglomeration of people from practically every economic class and interest, it is unlikely that he can represent his electorate adequately. At the best he would be representing a minority class, usually the upper class to which he would most likely belong. His voice would then be that of the minority few and such representation is highly undesirable.

In practice, it is hard to apportion groups on the basis of homogeneity yet when boundaries are drawn in such a way as not to put those who live in slums together with those who live in palaces or have strictly industrial areas and strictly agricultural areas together, then homogeneity of electoral grouping is preserved.

The advent of electronically controlled techniques makes the inclusion of homogeneity very difficult since a precise mathematical definition of homogeneity is not readily available.

2.3.5 State law

Parliaments usually pass laws which guide those concerned with forming units into groups for electoral purposes.

These laws usually define the population measure to use. They also define some other terms, like, 'electorate', electoral quota, etc. Most importantly, the laws must state the number of seats allocated to the country or a sub-region of the country.

In general, only the basic criteria are laid down. They are usually loosely phrased, thus, they fail to guarantee equitable apportionment even when they are strictly followed as they demand; nevertheless, whatever strict regulations they define must be adhered to or else the whole apportionment excercise could be in vain.

The loose nature of these state laws resulting in their lack of guidance, often lead to malapportionment and gerrymandering, and thus, law courts step in to settle disputes arising from malapportionment; especially in the United States where the courts can over-rule a parliamentary act as unconstitutional.

I would like to state that in as much as it is desirable to abide by state laws, it is more desirable to have, precise, clear and unambiguous state laws. I would therefore recommend a review of the existing apportionment acts since they hardly serve any purpose presently other than guaranteeing possible gerrymandering freedom.

2.3.6 Singularity of representation

It is a general practice to have a single member representing one electoral group. Recently there has been a desire to create multi-member groups whereby more than one person would be elected to represent an electoral group. Some european countries have actually adopted this method for some of their elections. In fact, Northern Ireland of Great Britain is a multi-member group since it is a single assembly group and yet elects three members to the European Assembly (European Assembly Elections Act (1978))[32].

Whatever may be the reason for the creation of these multi-member groups, I believe strongly that a single member group has more merits and therefore more desirable. Where this is not practicable, and I do not see why not, the electorate must be informed about the demerits and merits of a multi-member constituency. I doubt whether it has any merits whatsoever.

The only reason for the emergence of multi-member groups must have been due to the difficulty of providing equitable apportionment with regards to single member groups. This difficulty could be overcome by using my computer technique which will be presented in the next chapter. I therefore affirm that singularity of representation is more desirable than plurality of representation.

2.3.7 Preservation of political boundaries

It is desirable to maintain existing political boundaries in order not to break up useful, ancient traditional ties. In the event of creating electoral groups it is therefore necessary to preserve as much as practicable the boundaries of cities, townships and counties.

It is a good practice to include cities, townships, counties and a few other minor local government areas in one and only one electoral area.

The preservation of these boundaries could lead to some minor population differences between electoral groups yet such minor differences in population would have little adverse effect on a state as compared to the problem that could be created by breaking up ancient historic and economic ties.

In some cases the preservation of these boundaries are stipulated by law. In the European Assembly Act (1978)[32], it is stipulated that the boundaries of the British House of Commons constituencies be preserved. The preservation of such large political boundaries could lead to large differences in the population of European Assembly Groups yet the relationship between the European Assembly and the House of Commons makes it more desirable to have such large entities together in one group than to have small portions of, say, twelve House of Commons groups in one European Assembly Group.

2.3.8 Natural geographical features

It is highly desirable to use geographical and topological features such as, rivers, lakes, mountains and valleys etc., as political electoral boundaries. Naturally, these geographical and topological features preserve the identity and 'mini' culture of a community. They determine in some respects the profession and hence the economic class of a community, thus, they could be useful both as natural 'barriers' as well as political 'barriers'.

Their importance as useful political boundaries should be foreseen during the initial apportionment exercise, hence, they could be used as the boundaries of the smallest population units that a nation would always strive to preserve in subsequent apportionment exercise for larger political groups.

Harris, C.C. (1964)[34] and Hess, W. and Weaver, J.B. (1963)[30] have some discussion on general considerations.

I shall now discuss in some detail a few of the computer techniques that have been used by some of the forerunners of computer techniques for creating electoral groups.

2.4 Previous work

The general problem of malapportionment and the search for an automatic and impersonal medium has led to the development of computer techniques for partitioning a state into electoral groups. People have therefore sought to develop computerised mathematical models for this purpose. Generally the problem is split into two parts, phase one and phase two, phase one generates all possible groupings. Phase two determines the best plan for the area using the groups generated in phase one.

Most of the techniques developed so far merge phase one with phase two but the determination of the best plan is done via human scanning.

I shall discuss the work done by the following: Hess, S.W. and Weaver, J.B. et al; James D. Thoreson and John M. Liittschwager; Garfinkel, R.S. and Nemhauser, G.L.; G.Mills. Except for G.Mills, the others named above worked on areas in the United States.

2.4.1 S.W.Hess, J.B.Weaver et al; "Computer Techniques for Non-partisan Political Redistricting". (U.S.).

Their work was centred on the use of compactness as a measure of the equitability of an electoral plan. They defined compactness as a measure of geographical and population concentration.

Hess, S.W. and his four 'civic-minded' engineers of the CROND. INC. (Computer Research on Non-Partisan Districting); in one of their papers published by Weaver, J.B. and Hess, S.W., titled 'A Procedure for Non-Partisan Districting: Development of Computer Techniques' (Yale Law Journal, Vol. 73) they said the following about compactness:

> "Moment of inertia provides a possible measure of compactness in legislative districting, involving both area and population".

In support of this they developed a technique which used compactness as an effective measure of the superiority of an electoral plan over another (Operational Research Journal, Vol. 13, 1965). See section 2.3.3 for reference.

Their technique merged phase one with phase two and finally employed human scanning to reject non contiguous groups. The choice of the best plan was based on the plan that produced the minimum moment of inertia.

2.4.2 Moment of interia - compactness

In an attempt to study the properties of rotating bodies physicists try to have a measure of the dispersion of a body's weight about an axis of rotation. This measure is called moment of inertia.

The moment of inertia of a mass about an axis of rotation can be defined as the product of the mass and the square of the distance to the axis. Where a body has only two dimensions and has an axis of rotation perpendicular to its plane then one would speak of the moment of inertia of the body about the point where the axis intersects the plane.

For electoral purposes and in relation to geographical and population concentration (density), let us consider the following: let $M(\bar{\underline{x}})$ for a group be defined as $M(\bar{\underline{x}})_{G_i} = \sum_{i=1}^{s} P_i(\underline{x}_1 - \bar{\underline{x}})^2$ = moment of inertia of group i. $i = 1, \ldots, s$ (units in group i). P_i, $i = 1, \ldots, s$ (population unit i, $i = 1, \ldots, s$; in group i). $\underline{X}_1$, $i = 1, \ldots s$ (Position of unit i). $\bar{\underline{X}}$ = The point of rotation (Point at which moment of inertia is being calculated).

It will be noticed that this will be smallest with respect to $\bar{\underline{X}}$ when $\underline{X} = \bar{\underline{X}}$ since $M(\bar{\underline{X}}) = \sum_{i=1}^{s} P_i(\underline{x}_i - \bar{\underline{X}}) = 0$. This implies in fact that the population is concentrated at the point at which the moment of inertia is being calculated. To determine the most compact therefore, the moment of inertia $M(\bar{\underline{X}})_{G_i} = \sum_{i=1}^{s} P_i (\underline{X}_i - \bar{\underline{X}})^2$ is calculated then

$$M_j = \sum_{i=1}^{T} M(\bar{\underline{X}})_{G_i}$$

$i = 1, \ldots, T$ (where there are T groups in the plan)

M_j = (moment of inertia of plan j).

When the moments of inertia are calculated from different points, they represent different plans. The choice would then be

$$\text{Min} \sum_{i=1}^{T} (M(\bar{X})_{G_i}, \Sigma M(\bar{X})_{G_i} \in M').$$

where M' is a set of M_j (the total moments of inertia for all plans) $j = 1, \ldots, K$ (where there are K plans).

2.4.3 Transportation algorithm

In theory the above is what Weaver and his friends intend to do. In their paper entitled 'A procedure for Non Partisan Districting: Development of Computer Techniques", (Y. LJ[illegible])[35] they noted as follows:

> "Since districting by minimizing moment of Inertia involves numerous calculations, application of this procedure by hand would require considerable time and introduce significant probability of arithmetic error. To overcome these problems, we have used electronic computers which very quickly perform the necessary calculations.....to the data supplied them".

They went further to add that,

> "No available programs or computer techniques are known which will give a single, best answer to the districting problem.....",

and they concluded by saying that,

> "The chosen measure of compactness makes it possible to take advantage of certain mathematical similarities between the redistricting problem and a problem already programmed on computers - That of assigning customer orders to specific warehouse locations so as to minimize freight costs. This program, supplemented for this specific use by various additional steps and sub-calculations assigns E.Ds (ED = Enumeration (unit) districts)(Customers), to L.D. Centres (Warehouses) (L.D. = Legislative (Group) district), in a manner minimizing moment of inertia (freight cost)".

They in fact used a transportation algorithm as confirmed by one of their papers on "Non-Partisan political redistricting by Computer" (O.R. 1965)[36] they commented as follows:

> "Other warehouse - location techniques were unsatisfactory; either warehouse capacities were unconstrained, or codes

> were unavailable or too small. We resorted to an approach built around existing transportation codes. While not the ultimate in districting programs, it worked".

They in effect first formulated the problem as an integer programming problem as follows:

Let K = Lesilative districts and

Let n = number of population units.

Let P_j = population of the j^{th} population unit, ($j = 1, 2, \ldots, n$)

Let d_{ij} = distance between centres of population units i and j for ($i, j = 1, \ldots, n$).

Then $X_{ij} = \begin{cases} 1 & \text{if the } j^{th} \text{ population is assigned to the } i^{th} \text{ centre.} \\ 0 & \text{otherwise.} \end{cases}$

a = minimum allowable group population as a percentage of the average group population.

b = maximum allowable group population as a percentage of average group population.

The objective is to determine the n^2 values X_{ij} that would minimise moment of inertia $\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij}^2 P_j X_{ij}$ subject to the following 3n + 1 constraints.

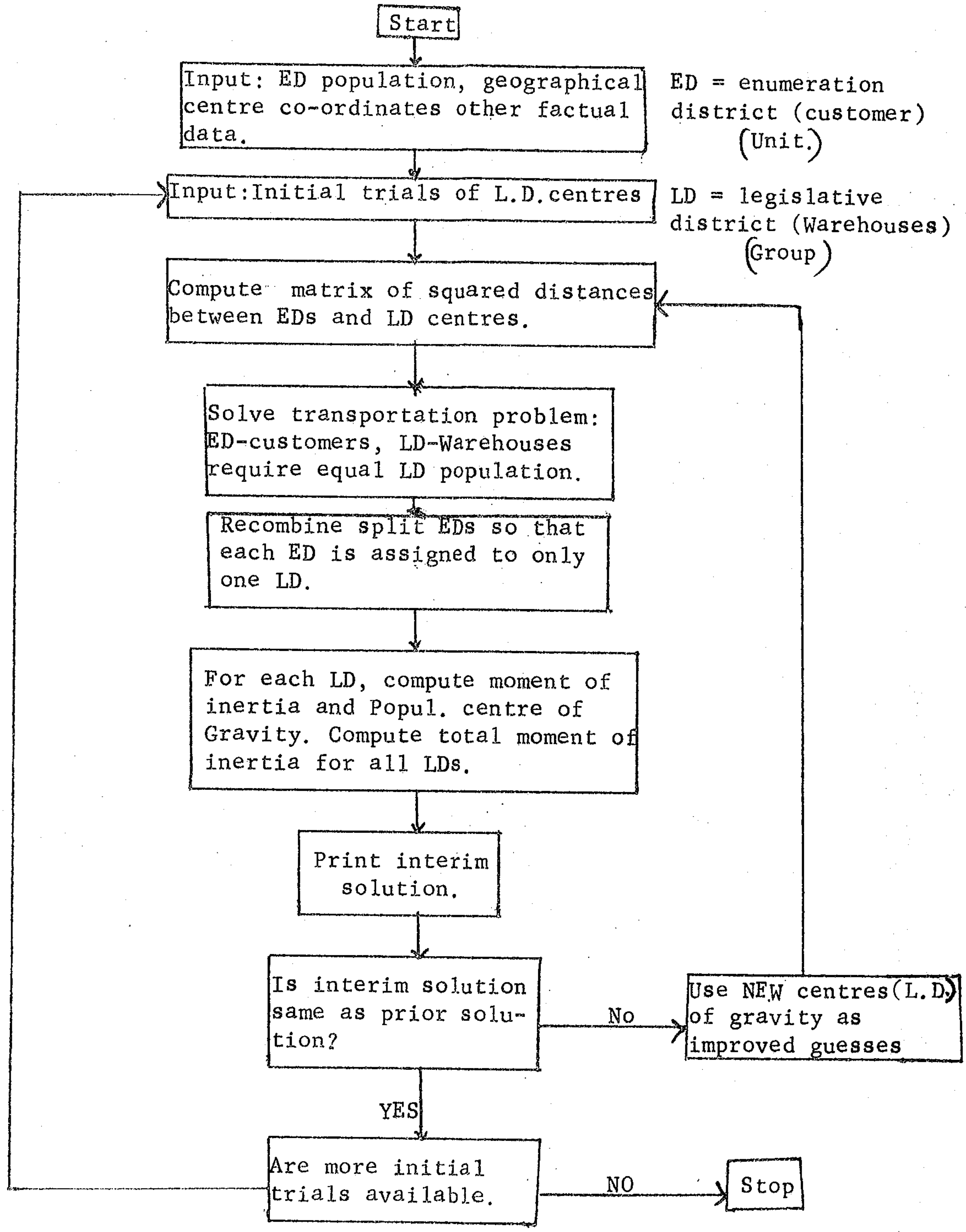
$$\sum_{i=1}^{n} X_{ij} = 1 \qquad (j = 1, \ldots, n)$$

$$\sum X_{ii} = K$$

$$\sum_{j=1}^{n} P_j X_{ij} \geqslant (a/100) \left(\sum_{j=1}^{n} P_j^1/K\right) X_{ii} \quad (i = 1, \ldots, n)$$

$$\sum_{j=1}^{n} P_i X_{ij} \leqslant (b/100)\left(\sum_{j=1}^{n} P_{j/K}\right) X_{ii} \quad (i = 1, \ldots, n)$$

The resultant plan from the above problem would then be checked manually for contiguousness. The application of this technique was not very satisfactory all through. They in fact had problems with states as small as Deleware State (O.R. Vol. 13,1965)[37] so they resorted to the use of a transportation heuristic which I intend to present diagramatically as follows.



Phase One:

The heuristic essentially goes through the following procedures:

1. An initial guess as to the geographical centres of both the units and the groups are made. These initial guesses are supplied as input data so also are the exact population figures for the units.

2. The transportation algorithm is used to accomplish the assignment

of the units to different groups on an equal population basis resulting in the splitting of the population figures of nearly all the indivisible units.

3. Adjustments are made so that each unit is assigned to one and only one group.

4. The moment of inertia as well as the new centre of gravity of each group is computed.

5. Steps (2) to (4) are repeated until the solution converges, although this is not guaranteed, nevertheless the local minima for each set of guessed centres is recorded for the selection of the minimum.

6. More initial guesses for both the units and group centres are made and the program is executed all over. The selection of the best plan is then made by inspection.

2.4.4 Phase two

In the above technique attempt is made to merge phase one with phase two. Although phase two is not completed since the final allocation still goes through human inspection for the selection of the plan that is contiguous with a small overall moment of inertia. The absence of a guarantee for convergence makes this technique a trial and error technique.

To illustrate the method of selection, consider the following.

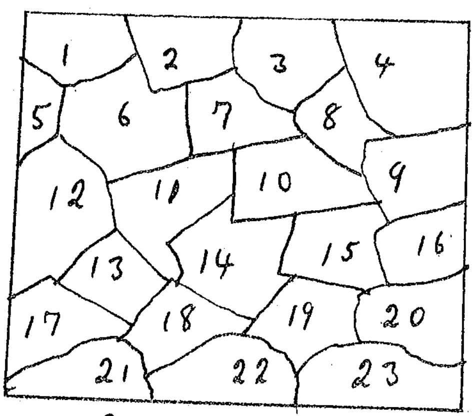
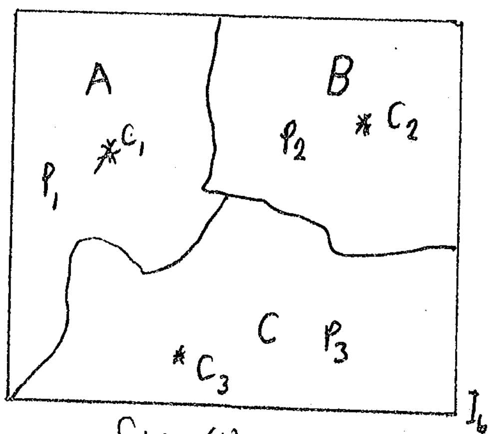


fig (4) (a)



fig(4) (b)

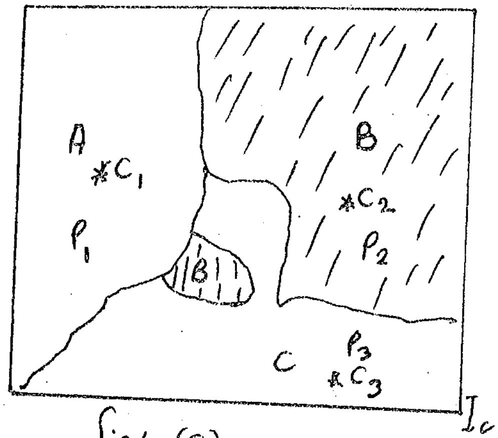


fig.4 (c)

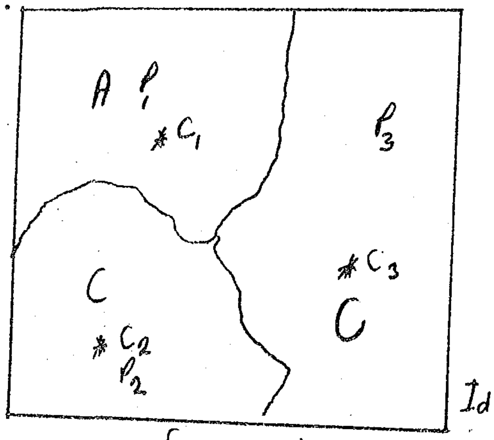


fig. 4 (d)

Let figure 4(a) be a set of units. If it is desired to divide the 23 units into three groups and after phase one accompanied by its splitting and readjustment of units. Let figure 4(b), 4(c) and 4(d) be the resultant plans with their populations P_1, P_2, P_3 and the associated moments of inertia I_b, I_c, and I_d. Figure 4(c) will be rejected on the grounds of the absence of contiguity while it will be left for the choice to be made between figure 4(b) and figure 4(d) on the basis of the moments of inertia and population equality. Weaver, J.B. et al (1965)[23] and Weaver, J.B. et al (1963)[35] have some more details.

2.4.5 <u>"Legislative districting by computer simulation: by J.D. Thoreson and John M. Liittschwager. (U.S.).</u>

They developed a heuristic approach to the partitioning of electoral areas problem. They accomplished the task of producing an equitable plan on the basis of population equality as their main objective. They selected total population as their major population measure and achieved equality of population by using the 'population variance ratio' and the 'minimum percentage test' as discussed in section 2.3.1(b) and 2.3.1(c) respectively.

For compactness they used Harris' measure of sum of maximum lengths minus minimum widths. They finally calculated the statistics for the moment of inertia although that was not included in the program as a compactness measure.

They developed two methods, method one and method two. Each of the methods accomplished the production of a plan in one phase and the best plan was determined by human scanning.

2.4.5.1 Method one

The first method is divided into two sections: One compulsory section and the other voluntary depending on the result of the first section. The area to be partitioned is arbitrarily divided into units (Pop) and grouping is accomplished by forming combinations of these units.

2.4.5.2 Section One

(A). The state is arbitrarily partitioned into regions with their corresponding populations. Arbitrary co-ordinate system is defined approximating the geographical centre of these regions. A region can take any shape. They are divided up in such a way that the most densely populated would contain a lower population than the state average per member. In a case where some regions for some reasons have to contain more people than the state's average per member, the following can be done.

i) If a region has more population than the states' average and if such a region cannot be split up, then that regions population must be divided by the state's average and the result rounded up to an integer; and consequently seats would be assigned to the region. Where the above is not followed then the following can apply.

ii) Let P_{min} = Pre-assigned lower limit and let P_{max} = Pre-assigned upper limit. Also let $\bar{P}_i$ = Population of region i. Then if $P_{min} \leq \bar{P}_i < P_{max}$, $\bar{P}_i$ must be assigned one seat. $(i=1,\ldots\ldots,N)$.

iii) If $2 P_{min} \angle \bar{P}_i \overset{or}{=} P_{max}$, then $(\bar{P}_i + P_{i+1})$ must be assigned two members. (P_{i+1}, implies other regions).

iv) If $2 P_{min} \angle \bar{P}_i$, then $\bar{P}_i$ should be assigned a number of members equal to the nearest integer value of $\bar{P}_i/P_K$ where P_K = state average.

Assignment continues as above until the remaining regions say P_{j+1} all have their populations below the lower limit.

(B). With all the unassigned regions having populations, each below the state's average and consequently lesser than the preassigned average, a new system of assignment is initiated. (Unassigned implies that no seats have been assigned to those regions). One of the unassigned regions say A_1 is selected as a reference region.

(C). B_1 is chosen, where B_1 is the region that is farthest away from A_1.

(D). With B_1 as a starting point form a group around it. Regions are assigned to B_1 on the basis of distance-contiguous criteria where P_{i+1} is assigned to B_1 if it is capable of forming a contiguous group and also nearer to B_1 than other unassigned regions. This combination on the basis of distance-contiguous criteria continues until an additional combination results in the 'population quota' being exceeded. 'Population quota' in this respect refers to the ratio of unassigned population to unassigned seats. At that point then, the distance criteria is abandoned and contiguous regions are added in such a way as to minimize the absolute deviation from population quota. The addition is continued until an electoral group is formed.

(E). Steps (C) and (D) are repeated. This involves finding the current unassigned region which is farthest from A_1 say B_2 and new electoral group is formed around B_2 and including B_2. Generally therefore electoral groups would be formed about B_i, i = 1, 2, ..., N. until such a time that all the regions and members are assigned. The completion of the above results in a plan. Different plans can be prepared by using different reference regions. The most compact with the least deviation from population quota is selected as the best. The selection which should have formed part of phase two is done via human scanning.

2.4.5.3 Method one: section two

It happens at times that due to some state law it could be required that some earlier political boundaries such as county boundaries should be preserved. The preservation of such boundaries could lead to large population disparities. Where that results then section two comes in as an attempt to equalise the population further more.

A) The single member group procedure is abandoned for two member groupings. The single member groups of section one are the data for this section.

B) Successively two single member groups of section one are added together. Each set of additions reduces the ratio term for the population variance ratio and this reduction is desirable. This section presumes that the two members are elected at large, hence the average population per member can be used as the population of people represented by each member. This process would terminate at the point when further reduction in population variance ratio would only be as a result of the combination of three section one single member groups.

2.4.5.4 Method two

This second method by Thoreson and Liittschwager is strictly meant for single member groups. The computer program is written in such a way that the computer assumes that cells with adjacent subscript notations are contiguous. For example, element (2.2), (2.3), (4.2) and (2.4) are supposed to be contiguous.

A) A grid is placed over a map to cover the whole area that needs to be partitioned into electoral groups. Geometric uniformity among the cells of the grid is maintained. The population of each grid is noted. Proper care is taken to ensure that the populations within each of the cells is not in excess of the average population per group.

B) The populations within each cell are estimated as accurately as possible. See Harris, C.C. (1964)[34] for different methods of estimating population densities.

C) Associate population densities to each of the grid cells and form a population matrix by assigning subscripts to each element. See figure 5 for a sample cell arrangment.

D) Choose an arbitrary element similar to method one and use this for reference. Call this A_1.

E) Determine B_1, the element farthest away from A_1 as was done in method one.

F) The first step is to form groupings about B_i say B_1 of the unassigned elements of the grid cells whose populations are less than the average for an electoral group.

With B_1 as a starting point, the units of the grid cells are grouped around it in a concentric counterclockwise manner until the summed population of the unassigned units exceeds the 'population quota'. 'Population quota' in this sense is the ratio of the unassigned population to the assigned members. Also the grouping is done in such a way that combining each successive vectors results in a contiguous rectangular electoral group.

The process therefore involves the determination of B_1 and the consideration of the next successive vector in the order of increasing distance from B_1 until a further addition would result in a greater deviation from the population quota than the previous one. The first unit assigned is usually the single element immediately above B_1. The next vector is then considered and further additions are made in the order of increasing distance from B_1. When it is likely that a further addition would result in a greater deviation from the population quota; stop, and an electoral group has been formed.

G). Steps (E) and (F) are repeated. This involves the determination of B_i say B_2, farthest away from A_1 selected from the unassigned units of the grid cells. A new group is formed about say, B_i, i = 1,....., K until all the units of the cell have been allocated to all the preassigned seats. Completion of step (G) results in a plan.

Below is an example of the cell arrangement for one electoral group and the order of assignment of the population units (cells) to the group.

29 IX	28 IX	27 IX	26 IX	25 IX
12 VI	11 V	10 V	9 V	24 VIII
13 VI	2 II	1 I	8 IV	23 VIII
14 VI	3 II	0	7 IV	22 VIII
15 VI	4 III	5 III	6 IV	21 VIII
16 VII	17 VII	18 VII	19 VII	20 VIII

<u>Assignment Sequence</u>

Vector	Order of Assignement
I	1
II	3, 2
III	5, 4
IV	7, 8, 6
V	10, 9, 11
VI	14, 15, 13, 12
VII	18, 19, 17, 16
VIII	22, 21, 23, 24, 20
IX	27, 26, 28, 25, 29

FIGURE 5. Sample Cell Arrangement

Like method one, a plan depends to a great extent on the choice of reference point A_i. Different reference points would normally produce slightly different plans. A choice as to the best is made via human scanning on the basis of population equality and to some extent compactness.

2.4.6 Computer apportionment technique; used by G.Mills (G.B.)

G.Mills in his paper entitled "The Determination of Local Government Electoral Boundaries" (O.R. Vol. 18, 1967)[38], in applying the technique used by Weaver et al (1965)[23], made some very useful observations and comments which can serve as a summary to the earlier discussion on the location-allocation technique. He applied this technique in partitioning the Bristol County Borough into local government electoral groups. He noted that,

> "The assignment of indivisible population groups (of varying sizes) to wards is a problem in integer programming. In practice, however, the size of the problem would render this approach computationally infeasible. Accordingly, the procedure used was a standard, 'Warehouse location heuristic' of a general kind already used in the electoral-boundary context by Hess......".

A summary of the procedure he used is as follows:

i) Arbitrary ward centres are chosen. Wards in this context refer to the local Government electoral groups.

ii) The transportation problem is solved and split units are combined.

iii) New centres are computed for the wards (groups), namely the centre of gravity of the set of units assigned to the wards (groups) in step (ii).

iv) Where any of the new centres differs from the old, return to step (ii); otherwise the procedure has terminated at what is in some sense, a local optima.

The above procedure is in fact similar to that used by Hess, Weaver et al.

G.Mills noted that

> "different local optima may be generated by starting from different arbitrary centres".

These local optima should be similar to the different plans produced by Thoreson and Liittschwager by altering their points of reference A_i.

Mills finally recommended that,

> "non-contiguous solutions must be rejected or adapted after human scanning of local optima".

This human intervention at the tail end of the program could possibly result to gerrymandering, which the whole computer technique was designed to eliminate and overcome.

I must mention that I have tried these techniques and made similar observations. What distressed me most about the techniques was the amount of time used in computing new centres, and the frequent occurence of non-contiguous plans that invariably have to be eliminated from consideration.

His initial observation that

> "The assignment of indivisible population groups (of varying sizes) to wards is a problem in integer programming",

is indeed very interesting. I saw the problem that way and solved it as such.

I shall next summarize the work done by Garfinkel, R.S. and Nemhauser, G.L. They both saw the problem as an integer programming problem and hence tackled it with their own integer programming technique.

For a detailed survey of Mills' work see (O.R. Vol. 18, 1967)[26].

2.4.7 R.S.Garfinkel and G.L. Nemhauser: "Optimal political districting"

Garfinkel and Nemhauser as stated above saw the partitioning of electoral groupings as an integer problem and solved it as such via an enumeration technique. I shall give a summary of their work but for a detailed survey see Garfinkel, R.S. and Nemhauser, G.L. (1970)[39], Garfinkel, R.S. (1968)[25] and Garfinkel, R.S. and Nemhauser, G.L. (1969)[40].

The first three surveys in this topic lacked a defined objective function and therefore had no function to be minimized or maximized and as a result convergence was not guaranteed and the production of an 'optimal plan' was also not guaranteed. An 'optimal plan' refers to the most acceptable plan with respect to equitability in apportionment.

On the contrary Garfinkel and Nemhausers' technique had an objective function and a guaranteed optimal plan with the termination of any program where such was possible.

They executed their program as follows:- They tried to consider equality in population, contiguity and compactness.

For population equality they considered all groupings $|P_{(j)} - \bar{P}| \leq \alpha\bar{P}$ (1) where $P_{(j)}$ was the population of a feasible group, $\bar{P}$ = the mean population for the area; and $\alpha 100$ $(0 \leq \alpha \leq 1)$.

For contiguity they defined a symmetric N X N - matrix with $B = \{b_{iK}\}$ and $b_{iK} = 1$ if units i and K have a common boundary greater than a point, $b_{iK} = 0$ otherwise.

For compactness they considered an exclusion matrix defined thus: $d_{(i,K)} > e(i,K) \rightarrow a_{ij} \cdot a_{Kj} = 0$ where for group j, $d_j = \max_{i,K} \{d(i,K)\, a_{ij} \cdot a_{Kj}\}, i,K = 1, \ldots, N$ is defined as the distance between the units of j which were farthest apart. They did not state exactly how this exclusion matrix was calculated.

In applying this technique to a fictional nine units state requiring four electoral groups. They calculated the exclusion matrix "somehow" (Management Science, Vol. 16, No.8, 1970, pp. B-499).

$Z_{i,K} = 0$ if $d(i,K) > e(i,K)$ (i and K may not be in the same group)
$= 1$ Otherwise (2)

	1	2	3	4	5	6	7	8	9	
1	-	1	1	1	1	1	0	0	0	1
2	1	-	1	1	1	0	0	0	1	2
3	1	1	-	1	1	1	1	0	0	3
4	1	1	1	-	1	1	1	1	1	4
5	1	1	1	1	-	0	1	1	1	5
6	1	0	1	1	0	-	1	1	0	6
7	0	0	1	1	1	1	-	1	0	7
8	0	0	0	1	1	1	1	-	1	8
9	0	0	0	1	1	0	0	1	-	9

They, then calculated the contiguity matrix. This matrix limited the number of population units to be considered on the grounds of being non-contiguous, where contiguity for B was defined as follows.

$B = \{b_{i,K}\}$, $b_{i,K} = 1$ if units i and K have a common boundary greater than a point (are contiguous).

= 0 Otherwise (3)

	1	2	3	4	5	6	7	8	9	
1	-	1	1	1	0	0	0	0	0	1
2	1	-	0	1	1	0	0	0	1	2
3	1	0	-	1	0	1	1	0	0	3
4	1	1	1	-	1	0	1	1	0	4
5		1	0	1	-	0	0	1	1	5
6	0	0	1	0	0	-	1	0	0	6
7	0	0	1	1	0	1	-	1	0	7
8	0	0	0	1	1	0	1	-	1	8
9	0	1	0	0	1	0	0	1	-	9
	1	2	3	4	5	6	7	8	9	

By summing the population of the different units in a 'tree-search' manner and determining those combinations that satisfied (1 - 3) plus a possible fourth, B' which was supposed to define a dimensionless measure of shape compactness. Assuming the following fourteen groupings were obtained:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1	1	1	0	0	0	0	0	0	0	0	0	1	1
2	1	0	0	1	1	0	0	0	0	0	0	0	0	1
3	0	1	0	0	0	1	0	0	0	0	0	0	0	0
4	0	0	1	1	0	0	1	1	1	0	0	0	1	1
5	0	0	0	0	1	0	1	0	0	1	0	0	1	0
6	0	0	0	0	0	1	0	0	0	0	1	0	0	0
7	0	0	0	0	0	0	0	1	0	0	1	1	0	0
8	0	0	0	0	0	0	0	0	1	1	0	1	0	0
9	0	0	0	0	1	0	1	0	0	1	0	0	0	0

Then the second phase which was a 'tree-search' technique as well would then try to determine the best four groups or the optimal grouping that would result in a feasible plan. The second phase was set up as follows:

$$\text{Min} \; \underset{j=1}{\overset{S}{\text{Max}}} \; C_j X_j \quad \text{where} \quad C_j = \left| P_{(j)} - \bar{P} \right| \Big/ \alpha \bar{P} \quad \text{subject to}$$

$$\sum_{j=1}^{S} a_{ij} X_j \leq 1 \qquad \begin{array}{l} i = 1, \ldots\ldots, N. \\ j = 1, \ldots\ldots, S \text{ (total number of groups)} \end{array} \qquad (5)$$

$$\sum_{j=1}^{S} X_j \leq M = 4 \qquad (6)$$

$$X_j = 0, 1, . \qquad (7)$$

a_{ij} = pop. unit i, in group j = 1 if

group j is in the optimal plan, = 0, otherwise.

The result would be (2, 4, 10, 11) and (2, 5, 9, 11) since they both give the same cost.

From the foregoing it is clear that in agreement with G.Mills, the problem is an integer programming one, and Garfinkel and Nemhauser have treated it as such.

My approach to the problem which will be discussed in the next chapter as applied to the European Assembly Groups (Constituencies) is an integer programming approach and my technique therefore has a lot in common with the technique that has just been presented.

The example given by Garfinkel and Nemhauser (Management Science, Vol. 16, No. 8, 1970)[39] is based on a fictional nine units state. They claim to have applied this technique to areas with larger population units than nine. For example, Washington State; nevertheless they applied it without success to West Virginia and also they were very unsuccessful with 55 county states.

CHAPTER THREE

PARTITIONING THE EUROPEAN CONSTITUENCIES

3.0

This chapter describes a computer procedure for (a) sorting given units into acceptable groups according to some specified criteria and, (b) finding the optimum plan, that is, the optimum selection of such groups which include all the units and also optimise some specified objective. The two stages are described respectively as Phase 1 and Phase 2, and the method is applied to a particular problem, namely the determination of the best plan for grouping the Northern Counties of England, (Cleveland, Cumbria, Durham, Tyne and Wear, Northumbria) into 5 European Constituencies. The approach however, is quite general and provides in principle a procedure for solving any such problem without the necessity for manual intervention or scanning once the criteria for Phase 1 and the objective(s) for Phase 2 have been defined.

In section 3.1 the conditions governing this particular problem are specified. Section 3.2 then describes the procedure used for Phase 1, the determination of acceptable groupings, using the criteria of (i) Contiguity of grouped units, (ii) sufficiently small population deviation, (Compactness has not been used since the total area considered is small, but could be included at the cost of considerable extra calculation). Section 3.3 defines the conditions for obtaining an optimum plan. Here, these are that all units must be allocated and that the number of groups is fixed. Subject to these, I define first a population objective and then a more general objective including in addition political considerations.

3.1 Elections Act

The European Assembly Elections Act (1978)[41] provided for the following seats with reference to the United Kingdom, 66 members for England, 8 members for Scotland, 4 members for Wales and 3 members for Northern Ireland.

The Act directed as follows:-

> "Each assembly constituency shall consist of an area that includes two or more parliamentary constituencies and (b) no parliamentary constituency shall be included partly in one and partly in another".

It further stipulated that

> "The electorate of any assembly constituency in Great Britain shall be as near the electorate quota as is reasonably practicable having regard, where appropriate to special Geographical Consideration".

The Act went on to define "electoral quota" and "electorate". It stated thus

> ".... in their application to a part of Great Britain for which there is a Boundary Commission - 'electorate quota' means the number obtained by dividing the electorate of that part of Great Britain by the number of assembly constituencies specified for that part....".

The Act defined electorate as follows:

> "....'electorate' means - (a) in relation to an assembly constituency, the number of persons whose names appear on the relevant registers for that assembly constituency in force on the enumeration date".

With the above extracts from the Act as a guide the relevant data for the problem were provided as follows:

3.1.2 (1) A map showing the geographical positions of the House of Commons Constituencies. Appendix 1 has an extract of the map relevant to this work. The map was from the Times Guide to the House of Commons.
(2) A comprehensive list of the Parliamentary Constituencies, their respective populations and their political complexion computed from the percentage of votes cast for the three major political parties during the last election. (The Times Guide to the House of Commons 1979)[42] and (Report, European Assembly Constituencies)[43].

The following is a list of the Parliamentary Constituencies, Populations and Percentages of votes, May, 1979.

		Pop.	Con.	Lib.	Lab.	Units considered.
1.	Scarborough	60,535	53.2	20.3	25.5	(1)
2.	Cleveland and Whitby	65,651	51.6	11.2	37.8	
3.	Teeside, Middlesbrough	64,573	30.4	9.1	56.2	
4.	Teeside, Redcar	63,249	36.7	8.9	53.7	
5.	Teeside, Stockton	88,181	36.2	9.2	53.1	(2)
6.	Teeside, Thornaby	62,518	39.1	9.2	51.1	
7.	Hartlepool	65,968	38.4	6.5	55.1	(3)
8.	Richmond (Yorks.)	64,669	61.5	21.2	17.4	(4)
9.	Bishop Auckland	75,134	37.9	13.3	48.7	(5)
10.	Darlington	63,408	43.4	10.2	45.5	
11.	Easington	65,416	24.7	14.4	60.9	(6)
12.	Durham	77,382	33.3	14.5	52.3	(7)
13.	N.W.Durham	63,329	29.6	9.1	61.3	(8)
14.	Houghton-le-Spring	60,609	20.7	10.2	65.5	(9)
15.	Chester-le-Street	79,588	25.2	14.4	60.4	(10)
16.	Consett	58,320	24.9	13.9	61.3	(11)
17.	Sunderland North	78,009	32.1	10.3	57.5	(12)
18.	Sunderland South	75,801	37.9	9.0	53.1	
19.	Jarrow	55,991	29.1	9.1	55.8	(13)
20.	South Shields	71,437	31.0	12.0	57.1	(14)
21.	Gateshead East	63,904	29.9	8.9	61.2	(15)
22.	Gateshead West	30,180	25.9	5.9	67.2	
23.	Blaydon	58,316	35.0	11.6	53.4	(16)
24.	Tynemouth	75,801	51.6	9.9	38.5	(17)
25.	Newcastle-on-Tyne Central	23,683	19.3	13.4	67.3	
26.	Newcastle-on-Tyne East	45,463	36.4	8.5	55.1	
27.	Newcastle-on-Tyne North	39,898	47.6	11.2	41.2	(18)
28.	Newcastle-on-Tyne West	81,410	35.9	9.6	54.5	
29.	Wallsend	90,179	31.3	12.3	51.1	(19)
30.	Blyth	77,687	22.7	8.3	40.1	(20)
31.	Morpeth	49,764	25.7	18.0	56.3	(21)
32.	Berwick-on-Tweed	42,703	38.4	54.3	7.3	(22)
33.	Hexam	66,846	48.0	20.1	31.9	(23)
34.	Carlisle	53,183	39.1	11.2	49.7	(24)
35.	Penrith and the Border	56,974	61.2	16.5	22.4	
36.	Workington	55,134	40.7	6.1	53.2	(25)
37	Whitehaven	52,224	39.8	5.9	52.4	(26)
38.	Westmorland	58,189	56.6	28.8	14.6	(27)
39.	Morecambe and Lonsdale	68,597	55.4	19.5	25.3	(28)
40.	Barrow in Furness	54,421	35.1	11.7	53.2	
41.	Lancaster	51,183	47.6	14.6	37.3	(29)
42.	North Fylde	77 528	60.8	14.4	24.0	

3.2 <u>Phase 1 - The determination of acceptable groups</u>:

Each state or country has some peculiar features that must be reflected in a partitioning exercise of this nature, so, the general considerations outlined in chapter II have to be applied with due regard to the specific requirements of the area in question.

The following considerations were made for the execution of Phase 1.

3.2.1 Considerations

(i) State laws

The European Assembly Act quoted in section 3.1 is the law guiding the apportionment exercise created, for the benefit of the boundary commissioners. A computer program executing the duties of a human agent must be bound by the same laws as the human agent.

The Act stipulates that the House of Commons constituency boundaries should be preserved and that two or more of these constituencies should be merged to form one European Assembly constituency. Hence the population units used are the House of Commons constituencies.

ii) Contiguity

It has been observed that there can hardly be any precise mathematical definition of contiguity. The absence of the concept of 'best' contiguity makes this rather more difficult. A group is either contiguous or not. It was therefore desirable in order to ensure the generation of contiguous groupings; I had to define a 0 - 1 matrix manually from the map in Appendix 1, and this was supplied to guide the computer programme against generating non-contiguous groups. This matrix was defined thus:, let j and K be units of T (T = state), = N X N matrix. Then $T_{j,K}$ = 1 if units of j and K have a common boundary =0 otherwise.

Furthermore some obvious combinations were made according to the following principles.

a) Units whose geographical areas are completely surrounded by another are merged together; for example Carlisle and Penrith and Border.

Why not automatic?

Again why not automate?

b) Areas that must be together due to their location at extreme points of the state were merged, for example, Scarborough, Whitby and Cleveland. It is impossible to attach these units to separate groups while maintaining contiguity. This may be confirmed by reference to the map on Appendix 1.

It must be noted that the program can still be executed without these initial mergers and yet produce an acceptable plan but for efficient execution, these steps are highly recommended. From the map therefore, areas that must be together were isolated, hence 29 units remained to be assigned to five groups and a (0 - 1) matrix could then be defined specifying which units were contiguous.

iii) Population The European Assembly Act 1978 further stipulated that each European Assembly Constituency population should be as close as possible to the 'electorate quota' where 'electorate quota' is as defined above: hence it is the number found by dividing the electorate by the number of seats for any particular area under consideration. To ensure the above I made the following considerations.

Let K = number of seats and let

Pi, (i = 1, ..., n) = population of each House of Commons constituency, then $\bar{P} = {}^{1}/_{K} \sum_{i=1}^{n} Pi$ = the electorate.

Since a European Assembly Constituency (group) was required to be as close to the 'electorate quota' as practicable, I decided to define the percentage deviation of a possible group as α 100 ($0 \leq \alpha \leq 1$) and to consider the effect of allowing only groups whose deviations were smaller than this.

The values used were α = 0.15, 0.10, 0.08, 0.05 and 0.02 respectively.

This idea of allowable percentage deviation meant that a feasible group was one which had a population within the allowable deviation. Hence let P_j = Population of group j, then group j was acceptable if

$(\bar{P} = \alpha 100\ \bar{P}) \leq P_j \leq (\bar{P} + \alpha 100\ \bar{P})$.

This deviation principle enabled phase one to generate more possible groups for α = 15% than α = 2%. It is desirable to have a good number of possible groupings because that allows for a wide choice in phase 2. It may in fact be impossible to satisfy the conditions for a possible plan if α is too small.

iv) Other considerations: A few other criteria as discussed in chapter 2 Section 2.3 could be included.

Compactness could be included as an additional requirement by rejecting any group which failed some criterion - it would require some geographical information also and would invariably involve a great deal of calculation thereby increasing cost due to computer time, in this case the compact nature of the country did not warrant its inclusion.

3.2.2 Description of phase one algorithm

With the following as the data, phase one was executed as described below.

i) Population units (House of Commons Parliamentary Constituencies.)

ii) Names of the population units. The above data were from the "Boundary Commission for England, report...." as listed in section 3.1.2.

iii) Percentages of votes for the different parties for the 3rd May, 1979 election.

iv) 0 - 1 matrix produced from map for contiguity.

The percentages as recorded in section 3.1.2 were from the Times Guide to the House of Commons ' May, 1979. There were some minor differences between the list of registered voters in the "Boundary... Report" as compared to the "Times Guide..." also the percentages were for the exact number of those who voted but I assumed that the same trend would be followed and hence used them for the electorate as listed in section (3.1.2). I therefore designed an additive algorithm for this purpose and execution was in seven steps.

Step 1. Phase one algorithm

The units (population) are summed up in this case the British House of Commons Constituencies: thus

Let $P_{Tot} = \sum_{j=1}^{n} P_j$ where j = units of population (House of Commons Constituencies).

$P_j \subseteq I \qquad j = 1, \ldots, n$

I = set of all British Parliamentary Constituencies.

Step 2.

The 'electorate quota' is calculated as follows

Let 'electorate quota' = $\bar{P}$ then

$\bar{P} = 1/_K \sum_{j=1}^{n} P_j$ where K = 5 = preassigned number of seats for the area under consideration.

Step 3.

The minimum and maximum acceptable population for a given α are determined as follows:

$P_{min} = \bar{P} - \alpha\ 100\ \bar{P}$ and

$P_{max} = \bar{P} + \alpha\ 100\ \bar{P}$

where P_{min} = minimum acceptable population.

and P_{max} = maximum acceptable population.

Hence P_j is possible if $(\bar{P} - \alpha\ 100\ \bar{P}) \leq P_j \leq (\bar{P} + \alpha\ 100\ \bar{P})$

Step 4.

The addition of individual units starts. This is carried out linearly starting with the first unit in the list and changing to the next according to the order of listing.

The following calculations are carried out for each initial unit.

i) Names are associated with the initial unit.

ii) The population deviation, is calculated thus $\left|\bar{P} - P_i\right|$ (j=1,,,,,n) for the initial unit.

iii) Calculations as to the political complexion are carried out as follows:

Let α^{**} = percentage of votes for party M in unit i, then

$$\beta_j = \left(\frac{P_i\alpha_i^{**}}{P_i}\right)_{M_L}\left(\frac{100}{1}\right)$$ = The percentage of votes for party M in the initial unit of group j.

(i = 1,....,N) = initial units in group j.

(L = 1, ..., Z) = Number of major parties under consideration.

The winning potential for the initial units are determined after testing the percentage obtained above as to whether a party was very likely, likely or not very likely to win should such set of units remain in the grouping of the final plan.

Step 5.

i) More contiguous units are added to the initial unit of (4) above and similar calculations like those of (4) are carried out for each set of units under consideration including the intial unit thus:

i) The unit to be added is tested for contiguity with the initial unit of (4) by reference to the 0-1 matrix, if contiguous it is added, if not it is dropped and another unit is considered.

ii) The population deviation is determined for the set of units in the group thus $\left|\bar{p} - \sum_{j=1}^{S} p_j\right|$ (j = 1, ..., S) for the set (P_j).

iii) Names are associated with all the untis.

iv) The political complexion is calculated thus:

Let α^{**} = percentage of votes for party M in unit i(i = 1,....,N)

then $$\alpha_j = \left(\frac{\sum P_i\alpha_i^{**}}{\sum_{j=} P_j}\right)_{M_L}\left(\frac{100}{1}\right)$$ = the percentage of votes for party M in group j.

(i = 1, ..., N for all the units.

(j = 1, ..., S) for the step P_j

(L = 1, ...,Z) for the major parties.

Step 6.

The contiguous group P_j generated in (5) is tested to determine whether it lies within the (a) population range thus

$P_{min} \leqslant P_j \leqslant P_{max}$.

If yes it is recorded with all the associated statistics.

i) Population deviation from electorate quota.

ii) Exact population of the group, populations and individual names of the units of the group.

iii) Winning potential index or percentage whichever is desired.

b) If $P_j < P_{max}$ and $P_j \geqslant P_{min}$, after the above cacalculations control returns to (5) for more contiguous units to be added.

Why?

c) If $P_j > P_{max}$ no records are taken. The first unit is dropped and it returns to (5).

d) If $P_j = P_{max}$ records are taken and it returns to (5). This process continues until all the units in the data set are considered then it returns to (4) and starts with another initial unit.

? What's linear about it

Step 7.

When all the linear combinations have been made and all the remaining P_j are together less than P_{min} termination occurs.

....

The result is a matrix a_{ij}, where i = 1, ..., n (number of units) and j = 1, ..., S (number of possible groups). This is rewritten into a 0 - 1 matrix where $a_{ij} = 1$ if unit i is in possible group j and $a_{ij} = 0$ otherwise.

This matrix can be associated with any desired objective for the execution of the second and final phase in the next algorithm.

The program for phase one can be found in Appendix 2.

3.2.3 Comments

The resultant number of groupings generated in phase one could be more than necessary and present some problems as regards computer time and space for the efficient and economic execution of phase 2. In this regard the algorithm has provision for limiting the search in phase one to a straight line and by starting from two extreme points of the population and at most three it could be possible to generate enough groups for a successful execution of phase 2 with a guaranteed optimal plan.

It is also desirable to set up the data cards in such a way that units are placed next to their nearest neighbours. This saves time and helps the scanning process of the 0-1 exclusion contiguity matrix during the execution of step 5.

As noted above the 'a_{ij}' = group matrix is an S X N matrix where S corresponds to the number of different possible groupings and N is the number of units.

Some of the groupings could in fact be subsets of others and it is the function of the next phase to determine the best combination of groupings that would constitute an optimal plan.

The following (figure 1) illustrates the formation of these groups.

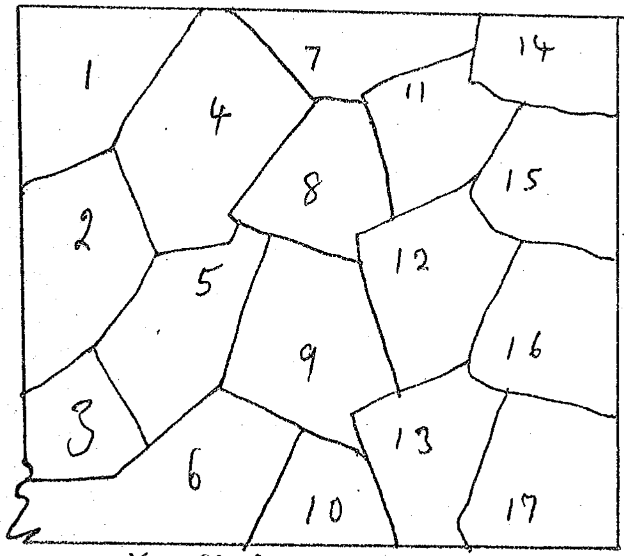


Fig. (1a)

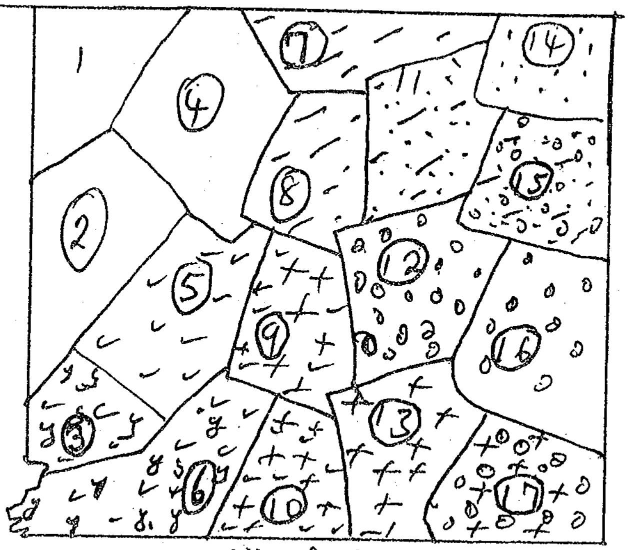


Fig. (1b)

Figure (1a) shows the individual units (population).

Figure (1b) illustrates the groupings formed thus:

If P_j, $j = 1, \ldots . T$ is a group then the next would be

(a) $P_{jT}, j_T = 1, \ldots\ldots, T + i, (i = T + 1, \ldots, n)$

or

(b) $P_{jm}, j_m = 1 + j, \ldots\ldots, (T + i), + (T + i + K), +, \ldots, n$

(a) The first case would occur if $P_{min} \leq P_j < P_{max}$ and $P_{min} < P_{jT} \leq P_{max}$.

(b) The second case would occur if for every j, $P_{min} < P_{jT} > P_{max}$ or P_{jT} is not contiguous then the first unit is dropped and combination starts from the next unit of the previous group.

Termination occurs if for all the units left $P'_j < P_{min}$.

3.3 Phase 2

Phase two determines the optimal plan with the results of phase one rewritten in a 0-1 format as its data and the necessary objective function generated in phase one as its objective. A necessary and sufficient condition for a successful execution of phase two is that all units are allocated. This phase would therefore terminate when all units have been optimally allocated or where such allocation is not possible due to perhaps the use of a very small α in phase one. Termination will occur after an exhaustive effort for a possible feasible allocation has failed.

The following considerations are applied for the execution of phase 2.

3.3.1 i) Constraints

Phase two is principally concerned with the application of the set covering theory discussed in section 1.9.6 of chapter one.

The following constraints are therefore necessary for the set up of the problem based on the requirements of the partitioning problem.

a) Preservation of Political boundaries:

Using the notation given there to express the present problem, the set I is the set of units (constituencies), the set P is the set of possible groups, that is the set of all groups satisfying the criteria which have been described. Each P_j is then a subset of the set I, P_j is most easily defined by a matrix A_j the elements of this could be defined by

$A_{ij} = P_i$, the population of unit i, if i is in the set P_j.

$A_{ij} = 0$ otherwise.

So that the total population of the i^{th} group, defined by the set P_j is

$$\sum_{i \in P_j} A_{ij}$$

However it is easier to consider the equivalent matrix produced by dividing the i^{th} row by P_i, that is

$a_{ij} = 1$ if i is in the set P_j

$a_{ij} = 0$ otherwise.

The requirement for a cover is the determination of X_j such that

$X_j = 1$ if j is in the cover

$X_j = 0$, Otherwise

where we now require

$$\sum_{j} a_{ij} X_j = 1$$

for all i (so that every unit i is allocated to exactly one set P_j).

$$i = 1, \ldots, n$$

$$j = 1, \ldots, S$$

Because of the equality requirement of the constraint and the $\{\leq\}$ and $\{\geq\}$ nature of the all integer and implicit enumeration algorithms respectively, the above was substituted with the following for the all integer and vice versa for the implicit enumeration.

$$\sum_{j=1}^{S} A_{ij} X_j = 1 = \begin{cases} \sum_{j=1}^{S} a_{ij} X_j \leq 1 & (1) \\ \sum_{j=1}^{S} -a_{ij} X_j \leq -1 \quad \text{i.e. } \sum a_{ij} X_j \geq 1 & (2) \end{cases}$$

b) Singularity of representation

Furthermore, single member groups were considered, hence each group must have one and only one representative and there must be only 5 representatives for the area under consideration. Hence the following constraint was necessary to guarantee that

$$\sum_{j=1}^{S} X_j = K = 5.$$

This was also converted to two inequality constraints thus:

$$\sum_{j=1}^{S} X_j \leq 5 \tag{3}$$

and

$$\sum_{j=1}^{S} -X_j \leq -5 \tag{4}$$

With these two most important requirements guaranteed it was certain that depending on the objective, whatever cover that resulted from the solution of the above would certainly be feasible and optimal.

To show how large this seemingly small problem could be in practice, $\alpha = 0.15$ yielded a 60 x 100 matrix while $\alpha = 0.10$ yielded a 60 x 74 matrix corresponding to 60 constraints and 100 variables, and 60 constraints and 74 variables respectively. The number of groupings and hence variables could be twice as much but as mentioned in section 3.2.3 the search could be limited and hence the number of groupings.

The next chapter has an example of a phase 2 set-up for groupings with $\alpha = 0.15$.

Having considered the major constraints, the next task is to consider the available objectives and choose that which we intend to use.

ii) Objective - population only

Since it is desirable to have a plan that ensures equitable apportionment, hence the populations of the group in the optimal plan should be as near the 'electoral quota' as much as practicable.

The following objective should then be adequate.

$$\text{Min} \sum_{j=1}^{S} C_j X_j \qquad (5)$$

$j = 1,\ldots.S$ (set of possible groups)

where $C_j = \left| P_{(j)} - \bar{P} \right|$

P_j = population of group j

$\bar{P}$ = electorate quota.

This acceptability measure which is in fact the deviation of group j from the electorate quota derived absolutely would certainly guarantee the production of the optimal plan strictly based on population nearness. The calculation of this acceptability measure is a function of phase one. The result obtained with population nearness as the main objective is adequately discussed in the next chapter.

iii) Objective - including political information

The program was designed among other things to determine the 'winning potential' of each of the major political parties. 'Winning potential' in this sense means the probability that a party would desire a particular arrangement as opposed to another because of the likelihood of winning more seats with one arrangement than the other.

Some of the advocates of bi-partisan 'districting' recommend that boundary commissioners be drawn from members of the major parties that have been holding a balance of power in the particular country. Furthermore they are the ones who would approve or disapprove any proposed plan.

An objective which included political information was applied. The objective was derived as follows: Let $P_j, (j=1,\ldots,n)$ be pop. units.

Let $\alpha_i, (j=1,\ldots,n)$ be the percentage of people who voted for party M in the previous election. Then

$$\left(\frac{\sum_{j=1}^{S} P_j\alpha_j}{\sum_{j=1}^{S} P_j}\right)_M \left(\frac{100}{1}\right)$$ would be the percentage of voters for party M in group j. Let

$$\mu_{j_m} = \left(\frac{\sum_{j=1}^{S} P_j\alpha_j}{\sum_{j=1} P_j}\right)_M \left(\frac{100}{1}\right)$$

Hence by maximising $$\sum_{j=1}^{S} \mu_{j_m} X_j \qquad (13)$$

one is in effect ensuring party M the optimal number of seats. Clearly, a plan reulting from this type of objective would in most cases reflect a vivid example of gerrymandering in practice.

By including this gerrymandering objective it is possible to predict the type of plans each of the parties would desire most.

The other objective could be a combination of both the nearness in population objective and the political bias objective. Hence

let $$\lambda_j = C_j X_j$$

and $$\mu_{j_m} = \left(\frac{\sum_{j=1}^{S} P_j\alpha_j}{\sum_{j=1} P_j}\right)\left(\frac{100}{1}\right)$$

$j = 1, \ldots, S$ (set of groups)

Then $$\max \sum_{j=1}^{S} k_{j_m} + \min \sum_{j=1}^{S} \lambda_j \qquad (7)$$

$$= \min \left\{ - \sum_{j=1}^{S} k_{j_m} + \sum_{j=1}^{S} \lambda_j \right\} \qquad (8)$$

This objective would produce a parameter map, one that considered both a parties winning potentials and the nearness of the populations of these groups to the electorate quota.

With a corresponding winning potential index as described in phase one step 5. The second phase of this algorithm was executed using the above two types of objectives. The results are in the next chapter.

3.3.2 Phase two algorithm

At the conclusion of phase one there resulted an (N X S) matrix a_{ij}, where $a_{ij} = 1$ if unit i is in group j and $a_{ij} = 0$, otherwise. The problem was therefore constrained as discussed in section 3.3.1 and the appropriate objective function chosen. Both the objective function and the constraint coefficients are functions of phase one.

Consider the following set up:

$$\text{Min} \sum_{j=1}^{S} C_j X_j \qquad (1)$$

where $C_j = \left| P_{(j)} - \bar{P} \right|$ $\qquad j = 1,\ldots,S\approx 100$ (for $\alpha=0.15$)

$P_{(j)}$ = Pop. of group j.

$\bar{P}$ = electorate quota.

Subject to

$$\sum_{j=1}^{S} X_j \{\leq, \geq\} 5 \qquad (2)$$

$$\sum_{j=1}^{S} -X_j \{\leq, \geq\} -5 \qquad (3)$$

(one of each)

$$\sum_{j=1}^{S} a_{ij} X_j \{\leq, \geq\} 1 \qquad \text{(29 of these)} \qquad (4)$$

and

$$\sum_{j=1}^{S} -a_{ij} X_j \{\leq, \geq\} -1 \qquad \text{(29 of these)} \qquad (5)$$

$X_j = 1, 0$ --- (6) $j = 1, \ldots, S$ (number of groups, 100 for α=0.15, etc.)

$X_j = 1$ if group j is accepted $i = 1, \ldots, N$ (number of units involved).

$X_j = 0$ otherwise.

$\{\leq, \geq\}$ the exact sign used depends on the method employed.

The above was for the population objective function, each objective function that was used called for a slightly different set up with respect to the objective function. Also for each α, α = 0.15, 0.10, 0.8, 0.5, and 0.2, a totally different set up resulted since the number of groupings changed, hence the X_j representing the variables changed, thus for α = 0.15 there were 100 X_js (variables), and 74 X_js (variables for α = 0.10 and so on. The number of constraints for this particular set of units remained at 60. Two constraints were to ensure that exactly the desired number of seats were allocated, while the remaining 58 ensured that each unit was allocated to only one group.

The problem was solved via two techniques which took advantage of the set covering set up as discussed in section 1.9.6 of chapter one.

The techniques were the implicit enumeration technique and Gomory's cutting plane method enhanced by the stronger Wilson's cut.

Method One:

(i) Implicit enumeration

As can be seen, the set up of the problem portrays it as an integer linear programming problem covered in section 1.5 of chapter one. It is in fact a bounded integer programming problem of the set partitioning type hence it yields itself to the solution technique known as implicit enumeration.

Sections 1.9.3 to 1.9.5 of chapter one have the theory behind this technique. I shall present the technique as used computationally.

In order to apply this technique to the problem, I adapted a program originally written by Plane D.E. and McMillan, C, Jr. (1971) and conditioned it to solve problems of this size and nature since it was originally intended for very small problems.

To arrive at feasibility in a good time the following techniques were employed.

a) All constraints were put in the form $G_i \geq$ a constant say 0 for this case.

b) All coefficients in the objective function were restricted to zero or a positive number.

Consider the following sets:

S = set containing subset of all variables completed by the assignment of 0, and 1. Any variable not assigned a value at S is a free variable. Also at S_K say, $X_j = 1$ appears as j while $X_j=0$ appears as -j hence for n = 4, and S = $\{1, -3\}$, then $x_1 = 1$, $x_3 = 0$ while x_2 and x_4 are free variables.

V = set of violated constraints, hence V_K = set of violated constraints when partial solution S_K is completed by setting to zero all free variables.

T = set of free positive variables (have posit. coefficients in one or more constraints). Hence T_K in V_K.

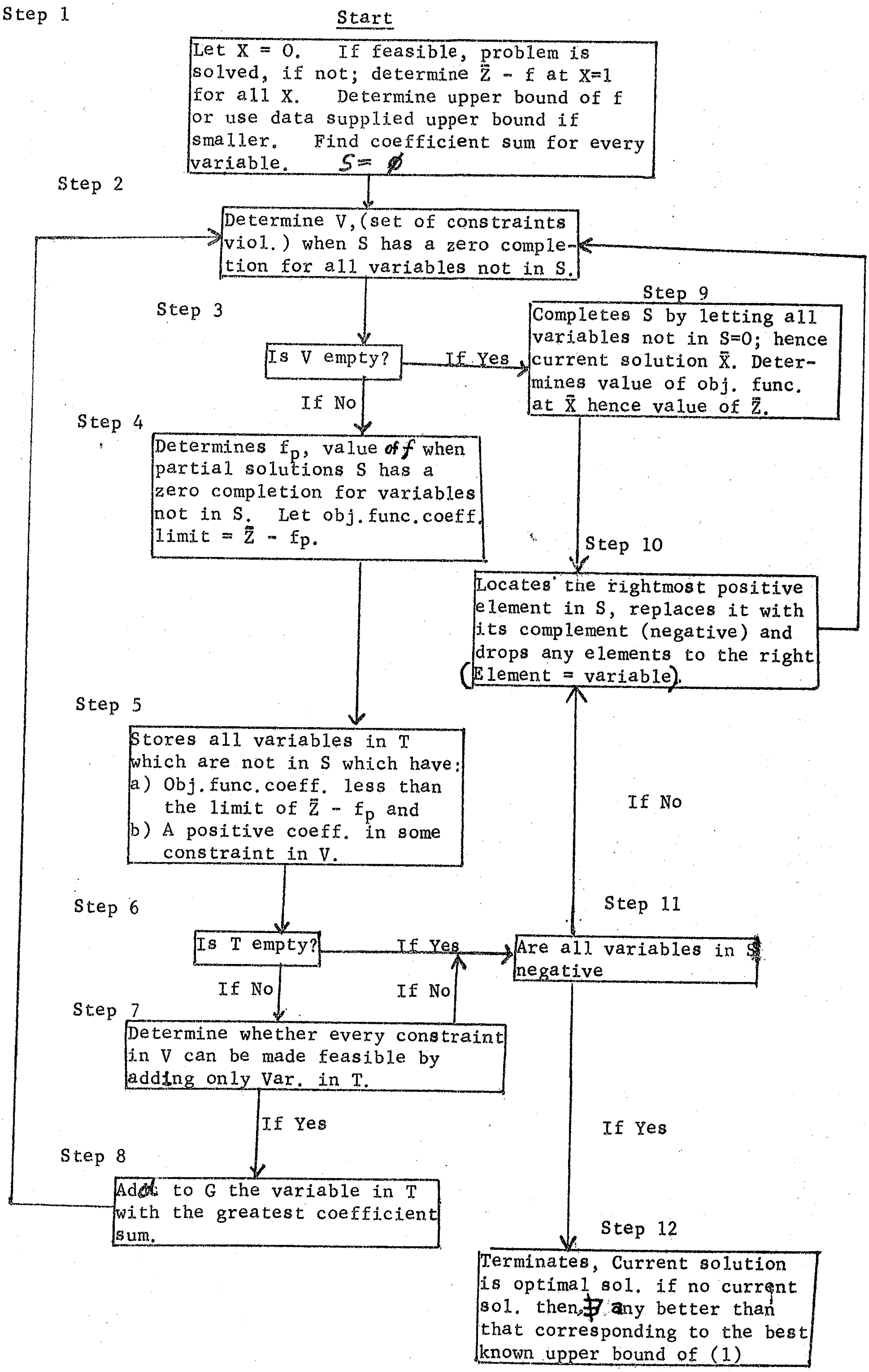
$\bar{X}$ = set of current optimal sol.

$\bar{Z}$ = value of f when obj. function is evaluated at $\bar{X}$.

The algorithm specifically compares feasible solutions as they are enumerated in search of the optimal.

With the above sets in mind I shall now present the algorithm diagramatically in the twelve steps that it goes through. At the end of the steps below, the problem of allocating all the units to their respective groups must have been accomplished.

This method is highly recommended for the partitioning problem. The reasons will be discussed in the next chapter.


Step 1
Start
Let X = 0. If feasible, problem is solved, if not; determine Z̄ - f at X=1 for all X. Determine upper bound of f or use data supplied upper bound if smaller. Find coefficient sum for every variable. S= ∅
Step 2
Determine V,(set of constraints viol.) when S has a zero completion for all variables not in S.
Step 3
Is V empty?
If Yes
Step 9
Completes S by letting all variables not in S=0; hence current solution X̄. Determines value of obj. func. at X̄ hence value of Z̄.
If No
Step 4
Determines f_p, value of f when partial solutions S has a zero completion for variables not in S. Let obj.func.coeff. limit = Z̄ - fp.
Step 10
Locates the rightmost positive element in S, replaces it with its complement (negative) and drops any elements to the right (Element = variable).
Step 5
Stores all variables in T which are not in S which have:
a) Obj.func.coeff. less than the limit of Z̄ - f_p and
b) A positive coeff. in some constraint in V.
If No
Step 6
Is T empty?
If Yes
Step 11
Are all variables in S negative
If No
If No
Step 7
Determine whether every constraint in V can be made feasible by adding only Var. in T.
If Yes
If Yes
Step 8
Add to G the variable in T with the greatest coefficient sum.
Step 12
Terminates, Current solution is optimal sol. if no current sol. then ∄ any better than that corresponding to the best known upper bound of (1)

The above description covers the 12 steps that are involved in the execution of phase 2 via implicit enumeration.

I shall add that at step 3 if $V = \emptyset$, then step 9 associated with its objective function value defines a possible cover with all the units allocated. It may not necessarily be the optimal cover.

The program for this technique is in Appendix 3.

ii) Gomory's method:

The problem was also solved via Gomory's method (section 1.9.1) and enhanced by the use of Wilson's cut as discussed in chapter one, section 1.9.2.

The problem was set up as discussed in section 3.3.2.

The a_{ij} and b_i are all integers while the C_jX_j are non-negative integers. C_j corresponds to the coefficients of the objective function while the a_{ij} correspond to the constraint coefficients. $(j = 1,\ldots.,S)$ $(i = 1,\ldots.,n)$
In particular the a_{ij} are 0's and 1's.
The resulting (MX1) X N matrix must be lexicographically positive, hence the first non-zero element in each column must be positive.
After the above set-up which is similar to the set up in section 3.3.2 execution was carried out in the following steps.

Step 0.

Starts with an all integer matrix A^o.

Step 1.

It selects from $a_{io} < 0$ $(i=1,\ldots.n+m)$ the row that has the smallest i, hence the generating row. But if $a_{io} \geqslant 0$ $(i=1,\ldots.,n+m)$ then the problem is solved.

Step 2.

Select $\lambda > 0$ according to the technique discussed in section 1.9.2, chapter one about deriving Wilson's cut and add the derived row to the bottom of the tableau.

Step 3.

Invoke the dual simplex method. This is similar to the simplex technique discussed in chapter one, sections 1.6 and 1.7. There relationship is as discussed in section 1.8 of the same chapter. The only difference is that it first decides what variable is to leave the basis, and then decides on the variable to enter the basis. It drops the pivote row after performing the dual simplex step and returns to step 1.

iii) Both techniques are good but I recommend the implicit enumeration technique because of a few advantages enumerated which makes it very suitable for a problem of this nature.

From the discussion in section 1.7 and section 1.9.5 of chapter one and also section 3.3.2 (i) of this chapter it is noted that when once an optimal solution is found, that is, if it is impossible to improve the objection function value both methods terminate, whereas it is desirable to have more than one optimal solution for a problem of this type. It is possible in fact to have two optimal plans but this next plan cannot be picked up since the programs terminate if the next available solution (plan) is not better than the current, whereas it is our intention to consider all plans of equal optimal value; nevertheless, the existence of a solution of equal importance can be investigated by perturbing the objective function values for the current optimal solution. Hopefully, if another optimal plan exists except the one already available it will be picked up.

I shall discuss briefly some other considerations before giving the results in the next chapter.

3.4 Other considerations

Compactness:

The compactness objective was considered but was not specifically used for this program. Consider the following:

$$\gamma_j = \left| L_j - W_j \right| \quad (9)$$

Where L_j = maximum length of group j

and W_j = maximum of width of group j

also consider $\beta = \left(L_j / W_j \right)$ (10)

where L_j and W_j are as defined above.

It is possible that γ_j and β_j would provide a possible measure of the compactness of group j, hence a possible objective that could take care of compactness would be the following.

$$\min \left\{ \sum_{j=1}^{S} \gamma_j X_j + \sum_{j=1}^{S} C_j X_j \right\} \quad (11)$$

or

$$\min \left\{ \sum_{j=1}^{S} \beta_j X_j + \sum_{j=1}^{S} C_j X_j \right\} \quad (12)$$

The above objectives are certain to produce a plan with an assured equitable arrangement. It is indeed very useful for countries that have very wide geographical areas like the U.S.A.

ii) Planning for political victory

Those who are working strictly for political victory can use (13) as the objective where $\max \sum_{j=1}^{S} \mu_{j_m} X_j$ (13)

or for a clearer assurance of victory the phase one algorithm should have the following:

let d_i' = population of voters of unit i who are in a desired party.

Then let $W_j = 1$ if $a_{ij} d_i' > P_{j/2}$ where P_j is the pop.of group j and $W_j = 0$ otherwise.

In phase one therefore, each time a group is generated, a calculation as to the controlling power of a party is made via the following constraint,

$$\sum_{j=1}^{S} W_j X_j \geq Z, \quad 0 \leq Z \leq M.$$

This will have an effect similar to the index calculated in this program for phase one but would differ in that it is not a constraint to guarantee the generation of any particular set of group complexion.

iii) Alternate phase one technique:

The efficient execution of the algorithm depends on the generation of enough possible groupings in phase one that could result in an optimal plan for phase 2.

An efficient technique for generating all possible groupings from a given set of units P_i, i=1,....n in phase 1 would be as follows:

Let $P_t \subset I = P_i$; $(t = 1,\ldots,M) \subseteq (i = 1,\ldots,n)$ such that

$\exists\ P_{t_K} \in P_t \quad (t = 1,\ldots,M)$

and $\forall\ P_{t_i} \exists$ a path between P_{t_i} and P_{t_K} which implies that P_{t_K} is connected to every $P_{t_i} \in P_t$.

With P_{t_K} as centre generate all possible combinations of P_{t_K} and other elements of P_t which satisfy the population criteria. In practice every element in I will in turn be treated as a P_{t_K} i.e. as a centre.

All the possible combinations generated would in effect form groups which would be within a defined range of population, contiguous and above all as compact as practicable. Phase two would then make the most equitable allocation with respect to the above criteria. The resulting optimal plan would in effect be the best possible.

The above arrangement is highly recommended for apportioning the usual small size constituencies of the House of Commons and the Councils.

The units un this case should be the census enumeration districts. A better result will be guaranteed if a grid is placed over a map and the grid cells used as the population units. The size of the units for the European Assembly problem discouraged the use of the above technique, nevertheless it was tried out.

The above could be used for any size of state. There will nevertheless be a lot of manual work supplying the original input data otherwise it is as straightforward as the algorithm just presented.

A slight adjustment of the phase one algorithm in Appendix 2 could also be used for this alternate technique.

I shall now present the solutions to the earlier problems in the next chapter.

CHAPTER FOUR

RESULTS

The results of the work discussed in chapter 3 will now be given and discussed.

In section 4.1 the performance of the phase one algorithm described in sections 3.2.2 and 3.3.2 is analysed; in particular the variation of the number of acceptable groups with α is demonstrated.

In section 4.2 is described the results of the two methods of implementing phase two. In section 4.3 the optimum grouping for the population objective is given. In section 4.4 is shown the effect of using a more general objective including political considerations.

4.1 Phase one results

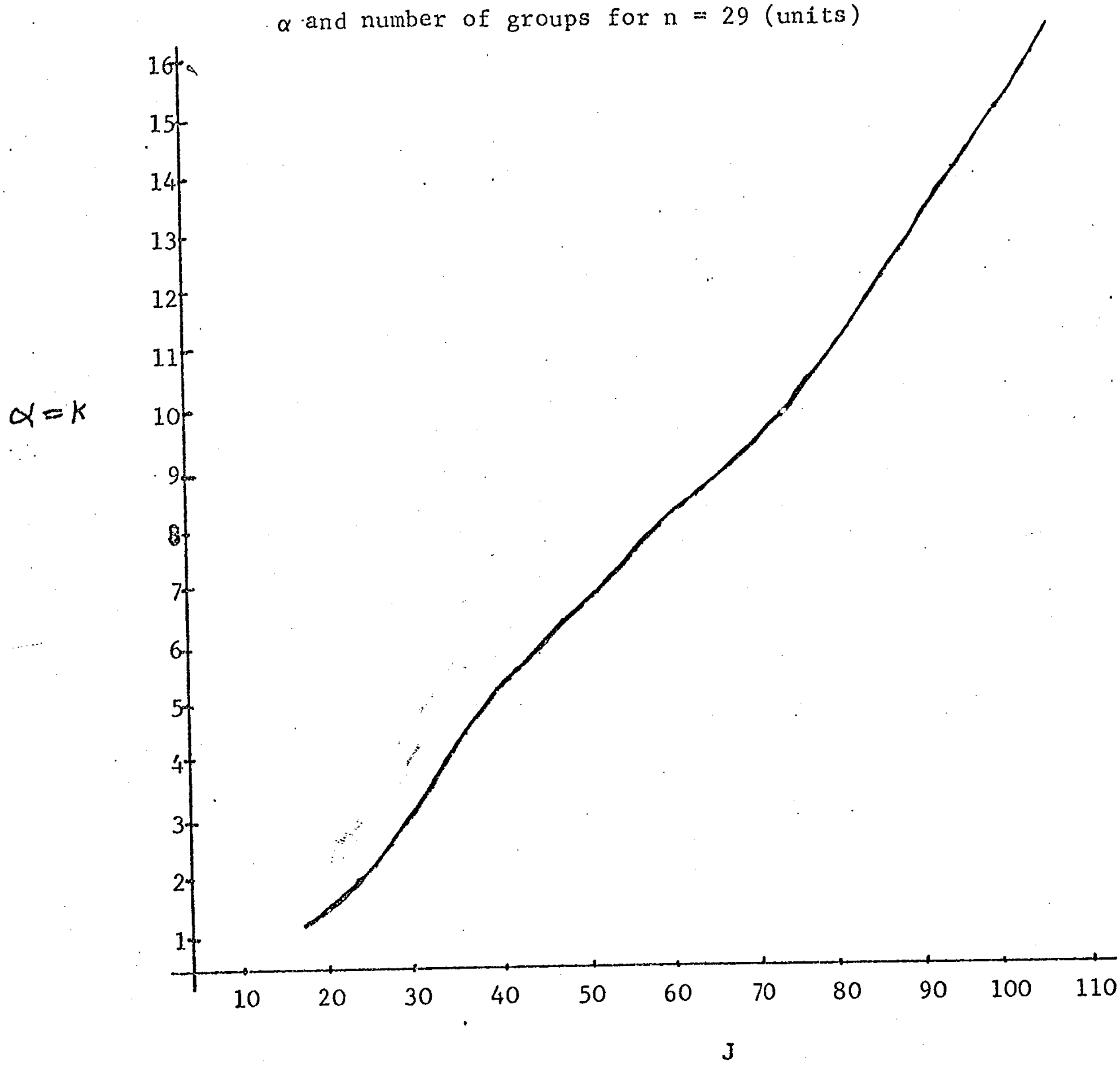
α = allowable percentage deviation	$P_j = a_{ij}$ groups formed. (possible)	Time in seconds
15%	100 distinct possible groups	1.715 seconds
10%	74 distinct possible groups	1.562 seconds
8%	60 distinct possible groups	1.6 seconds
5%	41 distinct possible groups	1.459 seconds
2%	26 distinct possible groups	1.37 seconds

The result of phase one was S (groups) column vectors satisfying a set of criteria. $S = a_{ij}$, unit i is in group j (or column vector j). The above is a brief analysis of the performance of phase one algorithm.

It is clear therefore that as α increases the number of possible groups formed increase because of the wider range of acceptable population deviation allowed and the time also increases. The result of the above would be that for α large a greater chance for gerrymandering exists than for α small.

Also for $\alpha = K \subseteq K^*$, $\alpha = K$ would contain all the groupings that are in $\alpha = K^*$, hence a larger α gives a wider choice of selection in phase two. For small α it may not be possible to obtain a solution to phase two at all, see section 4.1.1. It is therefore recommended to use a large α and then use phase two to select the best plan from it.

α and number of groups for n = 29 (units)



where P_j, j = 1, 2, ...J is an acceptable group satisfying a given set of criteria.

Figure 1.

4.1.1

Population objective	α = K%	No. of groups P_j	Time for optimal cover via implicit enumeration
	15%	100	247.325 seconds
	10%	74	61.721 seconds
	8%	60	50.53 seconds
	5%	41	30.721 seconds
	2%	26	No cover

The time varies for different objectives, hence different objective function values will involve slightly different paths. It was observed that the optimal plan for the population objective was the same for all the αs that produced a cover. It was different for the political objective which involved gerrymandering, hence more scope for gerrymandering with a large α than a small α.

4.2 Phase Two: Performance of the two techniques.

(i) The results obtained from the 2 methods of implementing phase two show that as mentioned in chapter 3 the implicit enumeration technique is preferable for the following reasons.

1) Addition is the only arithmetic operation, and therefore it would be possible to handle the allocation of a very large area with very many units.

2) Assuming the programme terminates prematurely due to computer time or otherwise, perhaps one of the partial solutions could be adequate and satisfactory since as it progresses it lists all the partial solutions and for the adaptation made only possible solutions are listed.

3) The possibility of monitoring the implicit enumeration calculations exist and this could help one to carry out informed stopping rules and opportunistic implementation should the number of variables and constraints be very large.

4) Groups of solutions are considered at any one time and enumeration is done implicitly instead of explicitly in which case the 2^n solutions are not enumerated and a lot of time is thus saved.

5) Because of the possibility of having more than one possible solution at any one run it has an added advantage over Gomory's method which has only one solution, the optimal solution. It

is therefore very much suited to this type of problem.

6) Where termination occurs without a cover hence no possible solution, implicit enumeration technique can provide a possible direction towards a manageable plan by reference to the different partial solutions enumerated during the search for a possible solution whereas Gomory's technique will provide none.

ii) Gomory's method enhanced by the stronger Wilson's cut generally carried out the allocation in a much shorter time than the implicit enumeration technique. The result was usually restricted to the optimal plan and was therefore a single solution.

Although the time was generally much shorter by using this method than the former yet the addition of one or more constraints in search of some other solution could increase the time greatly and in most cases execution could terminate without a solution.

It was not possible to monitor the steps during execution or carry out informed stopping rules and opportunistic implementation.

It seems therefore that despite the fastness of the above technique yet the implicit enumeration technique is preferable for this type of problem.

iii) The result is a vector X, (x_j = 1, if group j is selected to be in the plan and x_j = 0 otherwise). The implicit enumeration technique would usually give the solution as a set of vectors X, X_j = 1 if group j is in the plan and x_j = 0, otherwise; where the first vector refers to the first possible plan, then the next, until the optimal plan is reached. Each plan has the associated objective function values which indicate the superiority of one plan over the other.

An example of the solution after an implicit enumeration approach to the problem with population as the only objective is given.

MAIL: 1 MESSAGE, TOTAL LINES 9

@RUN 91 5=001
EXECUTION BEGINS 06:39:44

OBJECTIVE FUNCTION

X 1	X 2	X 3	X 4	X 5	X 6	X 7	X 8	X 9	X 10
X 11	X 12	X 13	X 14	X 15	X 16	X 17	X 18	X 19	X 20
X 21	X 22	X 23	X 24	X 25	X 26	X 27	X 28	X 29	X 30
X 31	X 32	X 33	X 34	X 35	X 36	X 37	X 38	X 39	X 40
X 41	X 42	X 43	X 44	X 45	X 46	X 47	X 48	X 49	X 50
X 51	X 52	X 53	X 54	X 55	X 56	X 57	X 58	X 59	X 60
X 61	X 62	X 63	X 64	X 65	X 66	X 67	X 68	X 69	X 70
X 71	X 72	X 73	X 74	X 75	X 76	X 77	X 78	X 79	X 80
X 81	X 82	X 83	X 84	X 85	X 86	X 87	X 88	X 89	X 90
X 91	X 92	X 93	X 94	X 95	X 96	X 97	X 98	X 99	X100

56.0	1.0	61.0	75.0	68.0	8.0	73.0	21.0	50.0	9.0
67.0	56.0	62.0	16.0	75.0	30.0	34.0	21.0	55.0	15.0
73.0	47.0	45.0	14.0	43.0	72.0	14.0	16.0	72.0	19.0
35.0	53.0	17.0	60.0	33.0	15.0	58.0	42.0	[illegible]	67.0
8.0	1.0	50.0	17.0	32.0	25.0	37.0	10.0	53.0	62.0
[illegible]	52.0	11.0	51.0	1.0	75.0	21.0	34.0	55.0	53.0
45.0	47.0	12.0	39.0	41.0	8.0	57.0	45.0	51.0	6.0
55.0	42.0	15.0	71.0	52.0	13.0	69.0	53.0	13.0	74.0
71.0	15.0	78.0	32.0	27.0	61.0	2.0	63.0	72.0	12.0
65.0	6.0	57.0	62.0	1.0	45.0	33.0	24.0	22.0	56.0

CONSTRAINTS

	CONSTANT										
G 1	1.0	-1.0	0.0	0.0	0.0	0.0	-1.0	-1.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
G 2	1.0	-1.0	0.0	0.0	0.0	0.0	-1.0	-1.0	-1.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0	0.0
G 3	1.0	-1.0	0.0	0.0	0.0	0.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	0.0
G 4	1.0	0.0	-1.0	0.0	0.0	0.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0
		0.0	-1.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	-1.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
G 5	1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	-1.0	-1.0	0.0
G 6	1.0	0.0	-1.0	0.0	0.0	0.0	0.0	-1.0	0.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	0.0
G 7	1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	-1.0	-1.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	0.0	0.0	0.0
G 8	1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	-1.0	0.0	-1.0	0.0
		-1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	-1.0	0.0	-1.0	-1.0	-1.0	-1.0	0.0	0.0
G 9	1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0

		-1.0	-1.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	-1.0	0.0	0.0	-1.0	0.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	0.0	0.0	0.0
G10	1.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		-1.0	-1.0	-1.0	-1.0	-1.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	0.0	0.0	0.0	0.0
G11	1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		-1.0	0.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	-1.0	-1.0	-1.0	0.0	0.0	0.0	0.0	0.0
		-1.0	0.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
G12	1.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	0.0	0.0	0.0
G13	1.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	-1.0	0.0	0.0	-1.0	0.0	0.0	-1.0
		0.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
G14	1.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
G15	1.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	0.0	0.0	0.0	0.0	0.0
G16	1.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		-1.0	0.0	0.0	0.0	-1.0	0.0	-1.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	0.0	0.0	0.0	0.0	-1.0	0.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
G17	1.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	-1.0	-1.0	-1.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
G18	1.0	0.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
G19	1.0	0.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	-1.0	-1.0	-1.0	-1.0	-1.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
G20	1.0	0.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0

		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	-1.0	-1.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
G21	1.0	0.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	0.0	0.0	0.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	-1.0
		-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
G22	1.0	0.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	-1.0	0.0	-1.0	-1.0
		-1.0	0.0	0.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
G23	1.0	0.0	0.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0
		-1.0	0.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
G24	1.0	0.0	0.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0	0.0	-1.0
G25	1.0	0.0	0.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0
G26	1.0	0.0	0.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0
G27	1.0	0.0	0.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0	0.0	0.0
G28	1.0	0.0	0.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	-1.0	-1.0	-1.0	-1.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0
G29	1.0	0.0	0.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0
G30	-1.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

G31	-1.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
G32	-1.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0
		1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0
G33	-1.0	0.0	1.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0
		1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
		0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
G34	-1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0
		1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0
		1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	0.0
G35	-1.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	1.0
		1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0
G36	-1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
		1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0
G37	-1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
		1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
		1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0
		1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	1.0	0.0	1.0	1.0	1.0	1.0	0.0	0.0
G38	-1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
		1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
		1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
		1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	1.0
		1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0
G39	-1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
		1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
		1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0
G40	-1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
		1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0
		1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
		1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
G41	-1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0
		1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
		1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0
		1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0

		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
		1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0
G54	-1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
		1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
G55	-1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
G56	-1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
G57	-1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
G58	-1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
G59	5.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
G60	-5.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
		1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
		1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
		1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
		1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
		1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
		1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
		1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
		1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
		1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

```
STEP
NUMB                                         PARTIAL SOLUTION

  535     39 -51   2 -42 -55 -95 -87 -70 -92  -6 -41 -66 -10 -48 -53 -63 -90 -76 -24  27
         -20 -36 -73 -82 -14 -33 -44 -28 -79 -30  -8 -18 -57 -99 -98 -46 -85 -16 -45 -84
         -35 -60 -97 -17 -58 -31 -47 -64 -65 -38 -72 -25 -96 -23 -61 -22 -62 -68  -9 -54
         -69 -52 -32 -49 -78 -19 -59   1 -12-100 -67 -93 -37 -34 -43  -3 -86 -13 -50 -75
         -94 -88 -71 -91 -11 -40   5 -77 -74   0   0   0   0   0   0   0   0   0   0   0

39319    -39 -51   2  42 -55 -95  87 -70 -92  -6 -41 -66 -10 -48 -53 -63 -90 -76 -24 -27
         -20 -36 -73 -82 -14 -33 -44 -28 -79 -30  -8 -18 -57 -99 -98 -46 -85 -16 -45 -84
         -35 -60 -97 -17 -58 -31 -47 -64 -65 -38 -72 -25 -96 -23 -61 -22 -62  68  -9 -54
         -59 -52 -32 -49 -78 -19 -59   1 -12-100 -67 -93 -37 -34 -43  -3 -86 -13 -50 -75
         -94 -88 -71 -91 -11 -40  -5 -77 -74 -81 -26 -29 -89  -7   0   0   0   0   0   0

67879    -39 -51  -2  42 -55 -95 -87 -70 -92   6  41 -66 -10 -48 -53 -63 -90 -76 -24 -27
         -20 -36 -73 -82  14 -33 -44  28 -79 -30  -8 -18 -57 -99 -98 -46 -85 -16 -45 -84
         -35 -60 -97 -17 -58 -31 -47 -64 -65 -38 -72 -25 -96 -23 -61 -22 -62 -68  -9 -54
         -69 -52 -32 -49 -78 -19 -59  -1 -12-100 -67 -93 -37 -34 -43  -3 -86 -13 -50 -75
         -94 -88 -71 -91 -11 -40  -5 -77 -74 -81 -26 -29 -89  -7 -21 -80  -4 -15   0   0


POSSIBLE      PLAN, STEP   535
                1   1   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   1   0
                0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0


POSSIBLE      PLAN, STEP 39319
                1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0
                0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0


POSSIBLE      PLAN, STEP 67879
                0   0   0   0   0   1   0   0   0   0   0   0   0   1   0   0   0   0   0   0
                0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0
                1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0


              BEST PLAN   FOUND
                0   0   0   0   0   1   0   0   0   0   0   0   0   1   0   0   0   0   0   0
                0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0
                1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                         OPTIMAL VALUE OF OBJECTIVE FUNCTION =      51.0000
EXECUTION TERMINATED   07:39:24  T=247.325  RC=0     $28.31

$SIG
```

4.3 Results with population objective only

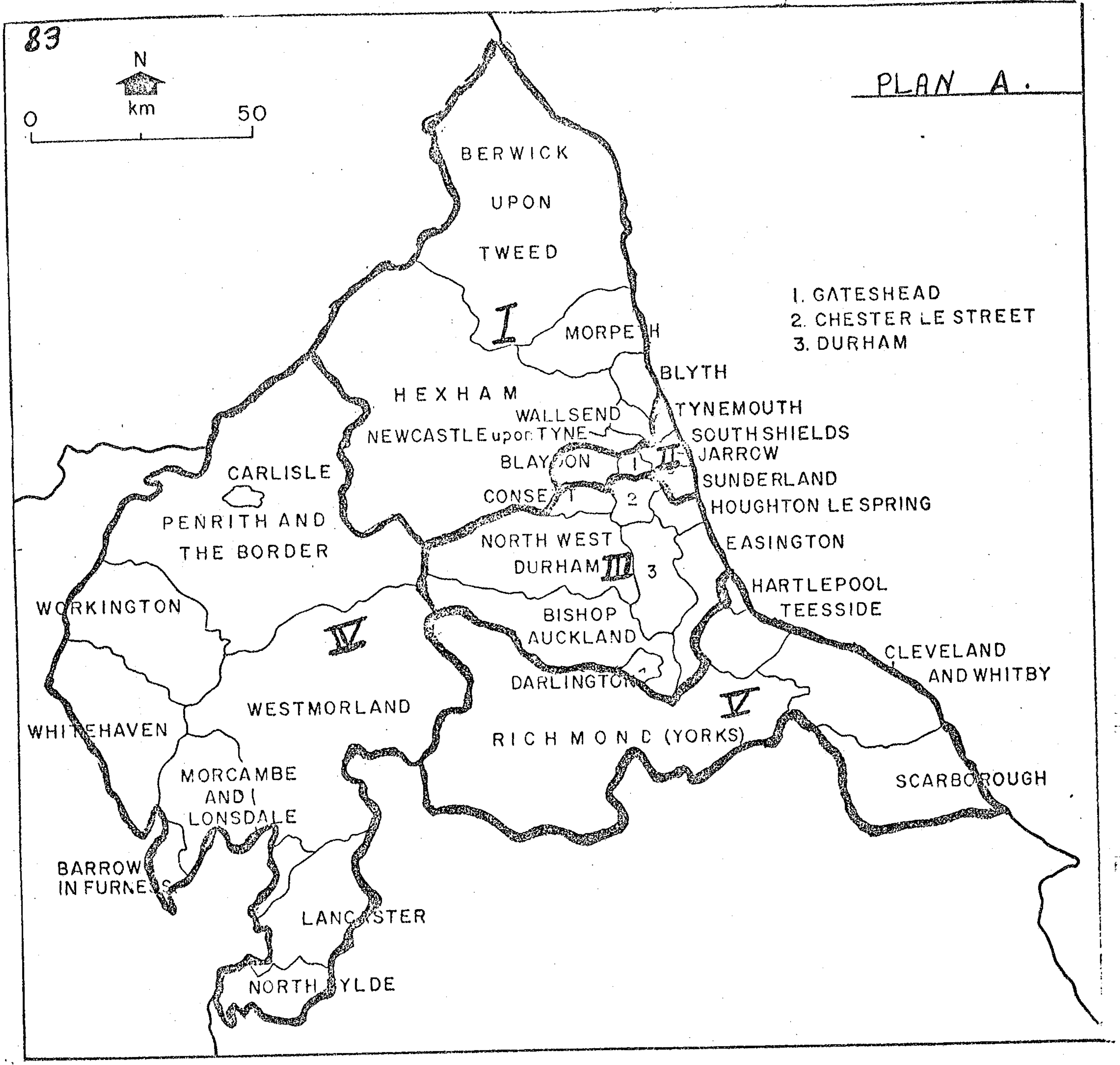
The optimum plan for the population objective is shown on Plan A, for the above result.(For Plan A see pp. 83.)

It is interesting to note that the maximum percentage population deviation from the electorate quota is 3.44% while the minimum is 0.34% and the average percentage deviation is 2.03%.

Contrasting the above with Plan A1 which was the initial plan provided via the enumeration technique with a maximum deviation of 13.04% and a minimum of 0.05% and an average of 5.38%. This improved to the plan shown on Plan A2, with a maximum deviation of 10.75%, and a minimum of 0.34% and an average of 4.29%. And finally the above optimal plan with a maximum deviation of 3.44%, minimum deviation of 0.34% and an average deviation of 3.44% as mentioned above. (For Plans A1 and A2 see pages 85 and 87 respectively.

It was observed that for a given objective there could exist more than one optimal plan with exactly the same objective function value yet the two techniques whose performance were discussed in the last section were unable to pick up alternative and yet equally good plans from a set of possible groupings. Usually termination occurred whenever an optimal plan was found and if there did not exist any other plan that could improve upon the objective function value of the current optimal plan. As mentioned briefly in chapter 3, section 3.3.2 (iii); the theory of implicit enumeration and the set covering theory of chapter one and the discussion on the methods of solving integer linear programming problems of the same chapter, section 1.9 gave rise to that problem. This could be overcome by perturbing the objective function values of the current optimal plan to check for the existence of another plan with exactly the same objective function values.

Also since I used α = 15%, 10%, 8%, 5% and 2% respectively one would have been tempted to conclude that α = 5% is the minimum that can produce the optimal cover yet the result shows that α = 4% can produce a cover; in particular α = 3.44% can produce a cover. Hence minimum α to produce an optimal cover is α = 3.44%.

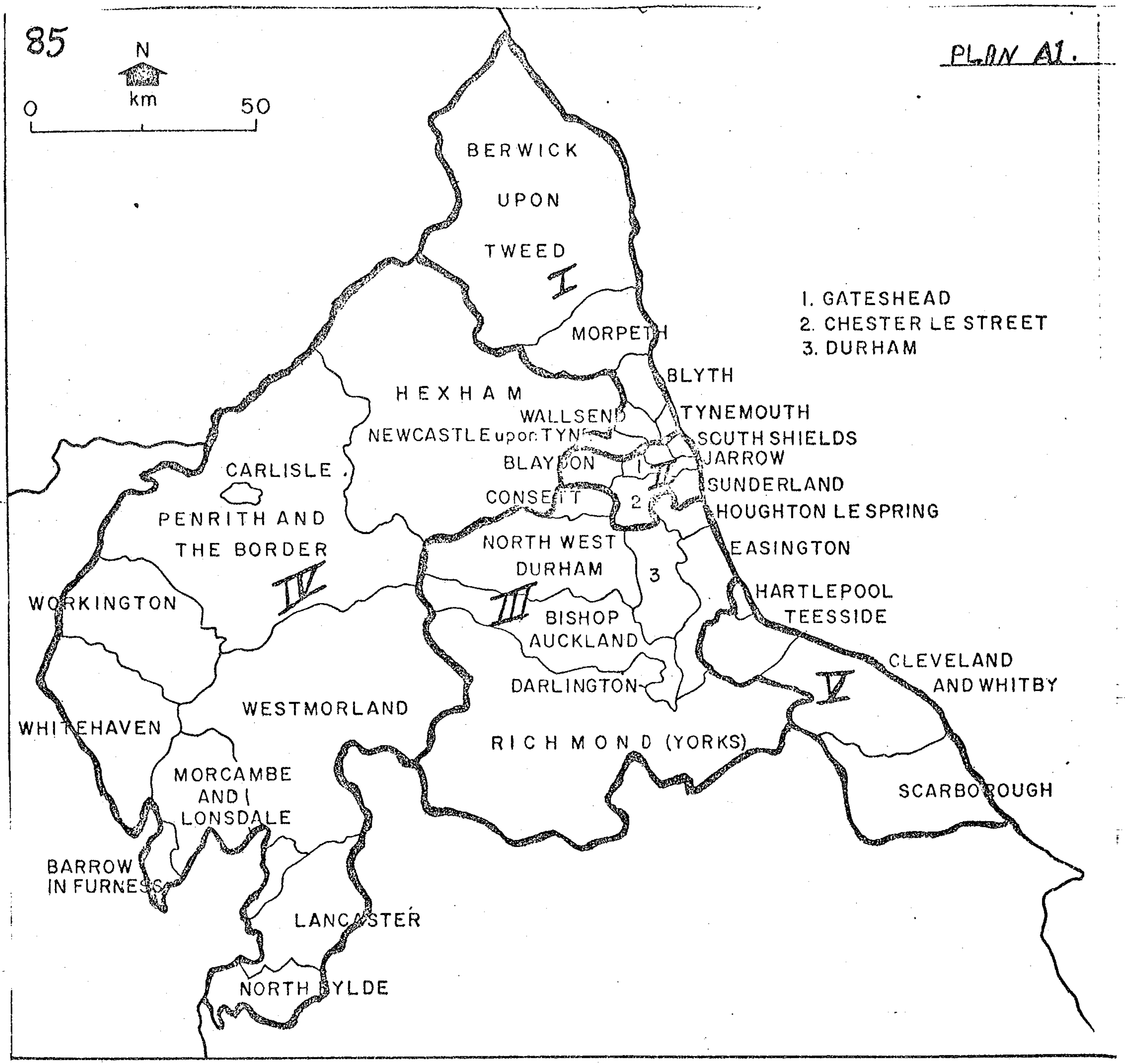

N
0
km
50
PLAN A.
BERWICK
UPON
TWEED
I
MORPETH
1. GATESHEAD
2. CHESTER LE STREET
3. DURHAM
BLYTH
HEXHAM
WALLSEND
TYNEMOUTH
NEWCASTLE upon TYNE
SOUTH SHIELDS
JARROW
II
CARLISLE
SUNDERLAND
HOUGHTON LE SPRING
PENRITH AND
THE BORDER
NORTH WEST
DURHAM
III
EASINGTON
HARTLEPOOL
TEESSIDE
WORKINGTON
IV
BISHOP
AUCKLAND
CLEVELAND
AND WHITBY
DARLINGTON
V
WESTMORLAND
WHITEHAVEN
RICHMOND (YORKS)
MORCAMBE
AND
LONSDALE
SCARBOROUGH
BARROW
IN FURNESS
NORTH FYLDE

Final Plan - Population Objective Only. α = 15% **PLAN A.**

Group = P_j	Population of Group - P_j	Percentage deviation of Group - P_j from electorate quota - $\bar{P}$
I	517,633	1.64%
II	508,140	3.44%
III	543,186	3.21%
IV	528,065	0.34%
V	534,344	1.53%

Average percentage deviation from the electorate quota for all groups = 2.03%.

PLAN A1.


N
km
0
50
BERWICK UPON TWEED
I
MORPETH
HEXHAM
BLYTH
TYNEMOUTH
WALLSEND
NEWCASTLE upon TYNE
SOUTH SHIELDS
JARROW
BLAYDON
1
II
SUNDERLAND
CONSETT
2
HOUGHTON LE SPRING
CARLISLE
PENRITH AND THE BORDER
NORTH WEST DURHAM
EASINGTON
3
IV
WORKINGTON
III
HARTLEPOOL
TEESSIDE
BISHOP AUCKLAND
CLEVELAND AND WHITBY
DARLINGTON
V
WESTMORLAND
WHITEHAVEN
RICHMOND (YORKS)
MORCAMBE AND LONSDALE
SCARBOROUGH
BARROW IN FURNESS
LANCASTER
NORTH FYLDE
1. GATESHEAD
2. CHESTER LE STREET
3. DURHAM

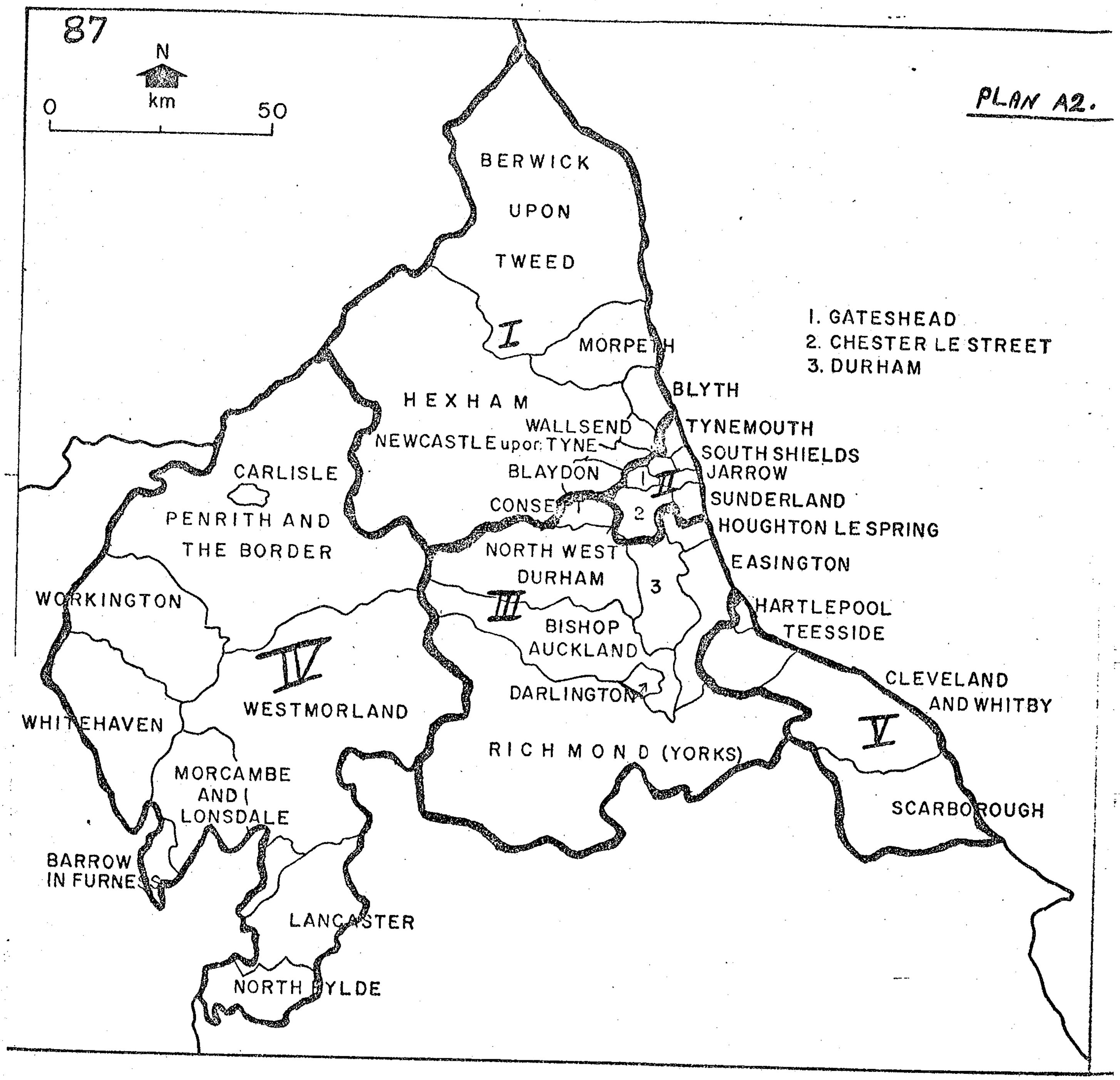
1st Plan - Population Objective Only. $\alpha = 15\%$

PLAN A1.

Group P_j	Population of Group - P_j	Percentage deviation of Group P_j from the electorate quota
I	526,588	0.05%
II	511,927	2.72%
III	528,267	0.37%
IV	594,911	13.04%
V	469,675	10.75%

Average Percentage deviation from the electorate quota for all groups = 5.38%

PLAN A2.


N
0
km
50
BERWICK
UPON
TWEED
I
MORPETH
HEXHAM
BLYTH
WALLSEND
TYNEMOUTH
NEWCASTLE upon TYNE
SOUTH SHIELDS
CARLISLE
BLAYDON
JARROW
1
II
CONSETT
2
SUNDERLAND
HOUGHTON LE SPRING
PENRITH AND
THE BORDER
NORTH WEST
DURHAM
EASINGTON
3
WORKINGTON
III
HARTLEPOOL
BISHOP
AUCKLAND
TEESSIDE
IV
CLEVELAND
AND WHITBY
DARLINGTON
WESTMORLAND
WHITEHAVEN
V
RICHMOND (YORKS)
MORCAMBE
AND
LONSDALE
SCARBOROUGH
BARROW
IN FURNESS
LANCASTER
NORTH FYLDE
1. GATESHEAD
2. CHESTER LE STREET
3. DURHAM

2nd Plan - Population Objective Only. $\alpha = 15\%$

PLAN A2.

Group P_j	Population of Group - P_j	Percentage deviation of Group P_j from the electorate quota - $\bar{P}$
I	575,949	9.43%
II	529,412	0.59%
III	528,267	0.37%
IV	528,065	0.34%
V	469,675	10.75%

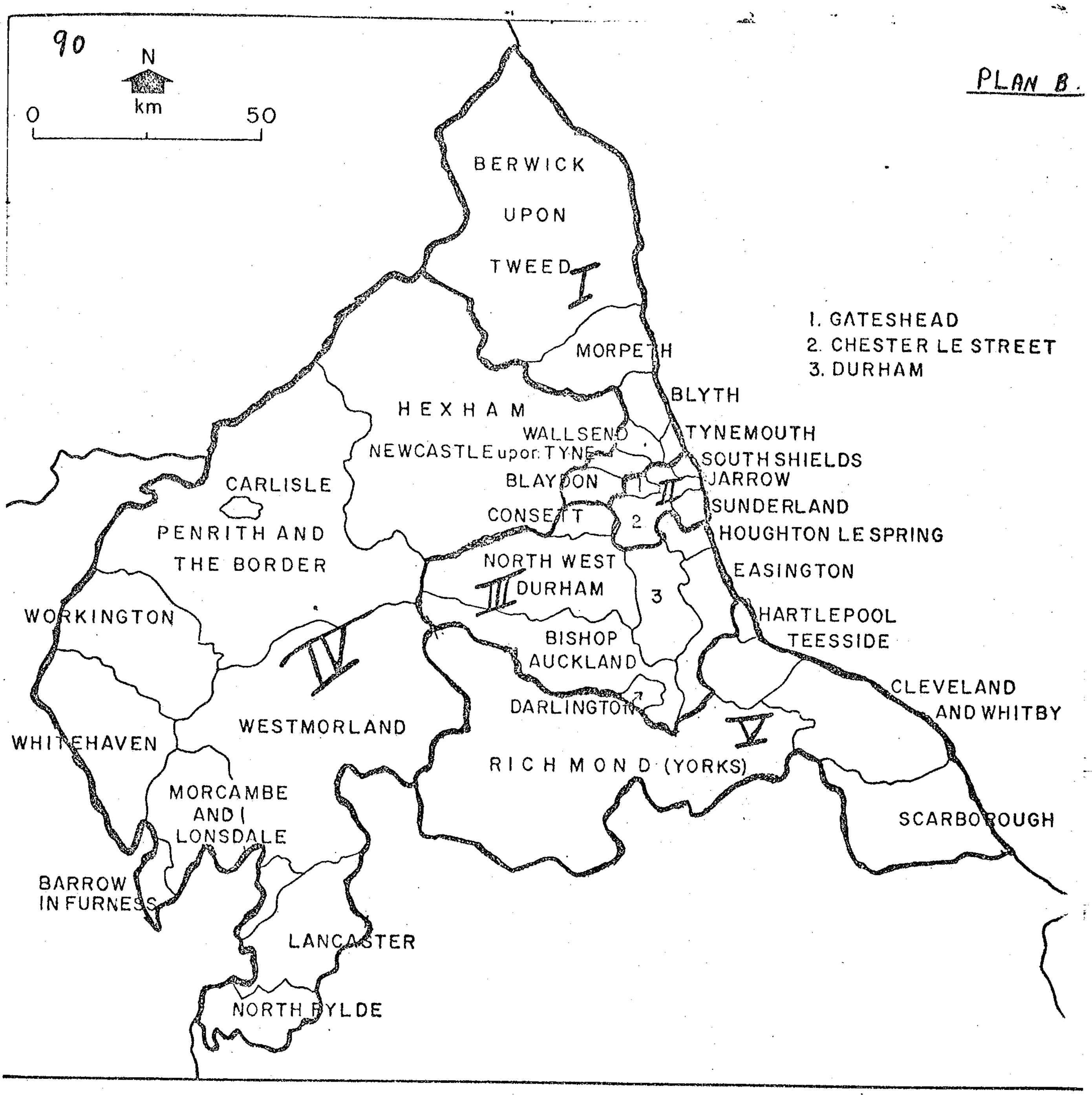
Average Percentage deviation from the electorate quota for all the groups = 4.29%.

4.4 Political objective only

The affect of using a political objective was observed for $\alpha = 15\%$. The following plans, Plan C (see pp. 95), and Plan B (see pp. 90) summarises the effects when apportioning on strictly political considerations.

Plan B (see pp. 90) shows the effect of allocating units for the benefit of the labour party, hence guaranteeing labour a maximum of 46% overall votes. This would guarantee labour two groups with over 50%, another two with over 40%. This grouping is possible at the detriment of the population objective which will cause overall 10.28% population deviation with a maximum deviation of 13.80% and a minimum of 1.53% deviation.

Letting μ = population objective and λ = political objective. For $\mu = 0$ the above holds and for $\lambda = 0$ the condition in the previous plan, Plan A (see pp. 83) holds.


PLAN B.
N
km
0
50
BERWICK
UPON
TWEED
I
MORPETH
BLYTH
HEXHAM
WALLSEND
TYNEMOUTH
NEWCASTLE upon TYNE
SOUTH SHIELDS
BLAYDON
JARROW
CARLISLE
SUNDERLAND
CONSETT
2
HOUGHTON LE SPRING
PENRITH AND
THE BORDER
NORTH WEST
DURHAM
EASINGTON
III
3
WORKINGTON
HARTLEPOOL
TEESSIDE
BISHOP
AUCKLAND
IV
CLEVELAND
AND WHITBY
DARLINGTON
V
WESTMORLAND
WHITEHAVEN
RICHMOND (YORKS)
MORCAMBE
AND
LONSDALE
SCARBOROUGH
BARROW
IN FURNESS
LANCASTER
NORTH FYLDE
1. GATESHEAD
2. CHESTER LE STREET
3. DURHAM

Final Plan - Labour Party Consideration. α = 15%

PLAN B.

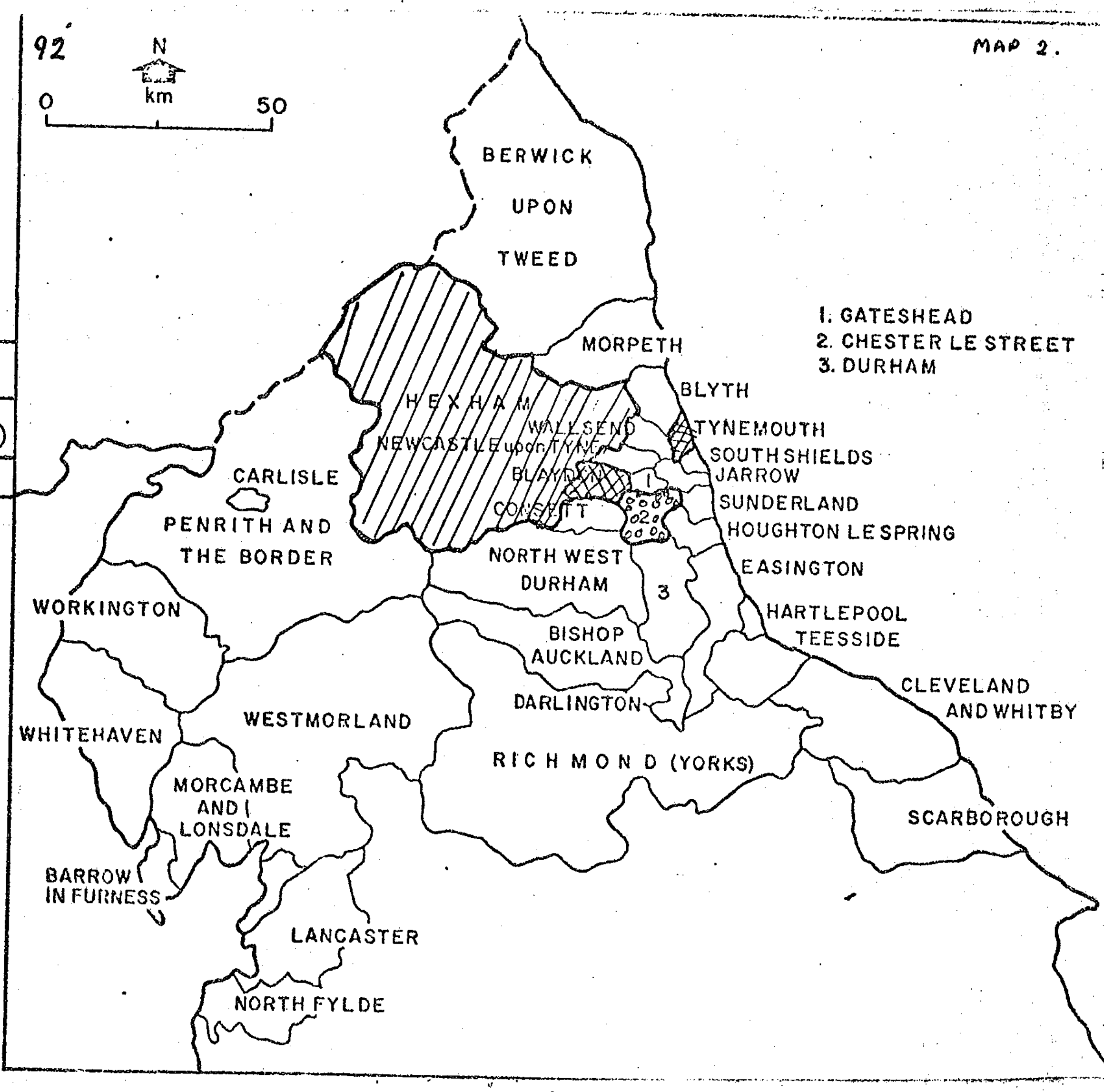
Group P_j	Population of Group - P_j	Expected Percentage of voters for Labour.	Percentage deviation of Group P_j from the electorate quota - $\bar{P}$.
I	584,904	45%	11.14%
II	453,611	58%	13.80%
III	463,588	51%	11.90%
IV	594,911	33%	13.04%
V	534,344	43%	1.53%

Average Percentage of Votes expected from all the groups = 46%.

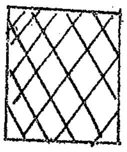
Average Percentage population deviation from the electorate quota for all the groups = 10.28%.

MAP 2.

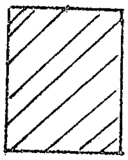
SHIFTS DUE TO LABOUR PARTY CONSIDERATION. (OBT.)
(See pp. 93)



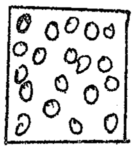
The following units shifted from their allocation in Plan A (see pp. 83) to the allocation in Plan B (see pp. 90) due to change in the objective from strictly population consideration to labour party consideration. See map 2. pp. 92.



Tynemouth and Blaydon shifted from II to I.



Hexam shifted from I to IV.

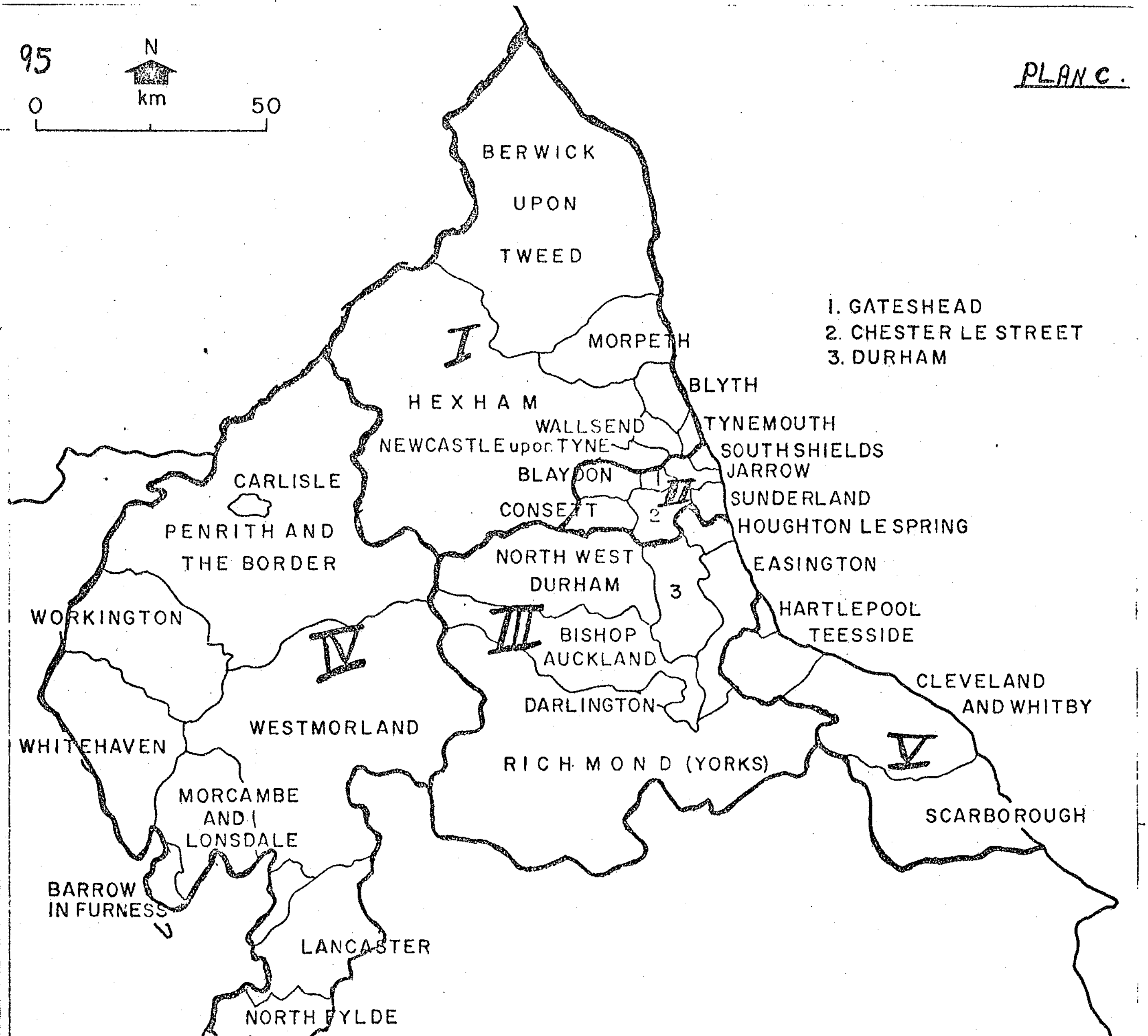


Chester-le-Street shifted from III to II.

ii) Plan C (see pp. 95), shows the effect of using a conservative party oriented objective and the consequent plan therefore. The results show that with such a plan the conservative party would have two groupings with about 40% - 49% in this labour dominated area and two groupings with about 36% - 38%. The plan would guarantee the conservative party an overall 38.6% votes to the detriment of the population objective which will have an overall 8.58% shift with a maximum of 12.76% and a minimum of 0.34% deviation. The effect of this objective is contrasted with the population objective in Plan A (see pp. 83).

iii) The above discussions are not the ultimate in the procedure but it is desirable to see the plan from the point of view of the major political organisations; nevertheless the population objective would normally provide the most equitable apportionment.

95


N
km
0
50
PLAN C.
BERWICK
UPON
TWEED
I
MORPETH
HEXHAM
BLYTH
WALLSEND
TYNEMOUTH
NEWCASTLE upon TYNE
SOUTH SHIELDS
JARROW
BLAYDON
SUNDERLAND
CONSETT
HOUGHTON LE SPRING
CARLISLE
PENRITH AND
THE BORDER
NORTH WEST
DURHAM
EASINGTON
3
HARTLEPOOL
TEESSIDE
WORKINGTON
IV
III
BISHOP
AUCKLAND
CLEVELAND
AND WHITBY
DARLINGTON
WESTMORLAND
WHITEHAVEN
RICHMOND (YORKS)
V
MORCAMBE
AND
LONSDALE
SCARBOROUGH
BARROW
IN FURNESS
LANCASTER
NORTH FYLDE
1. GATESHEAD
2. CHESTER LE STREET
3. DURHAM

Final Plan - Conservative Party Consideration. $\alpha = 15\%$

PLAN C.

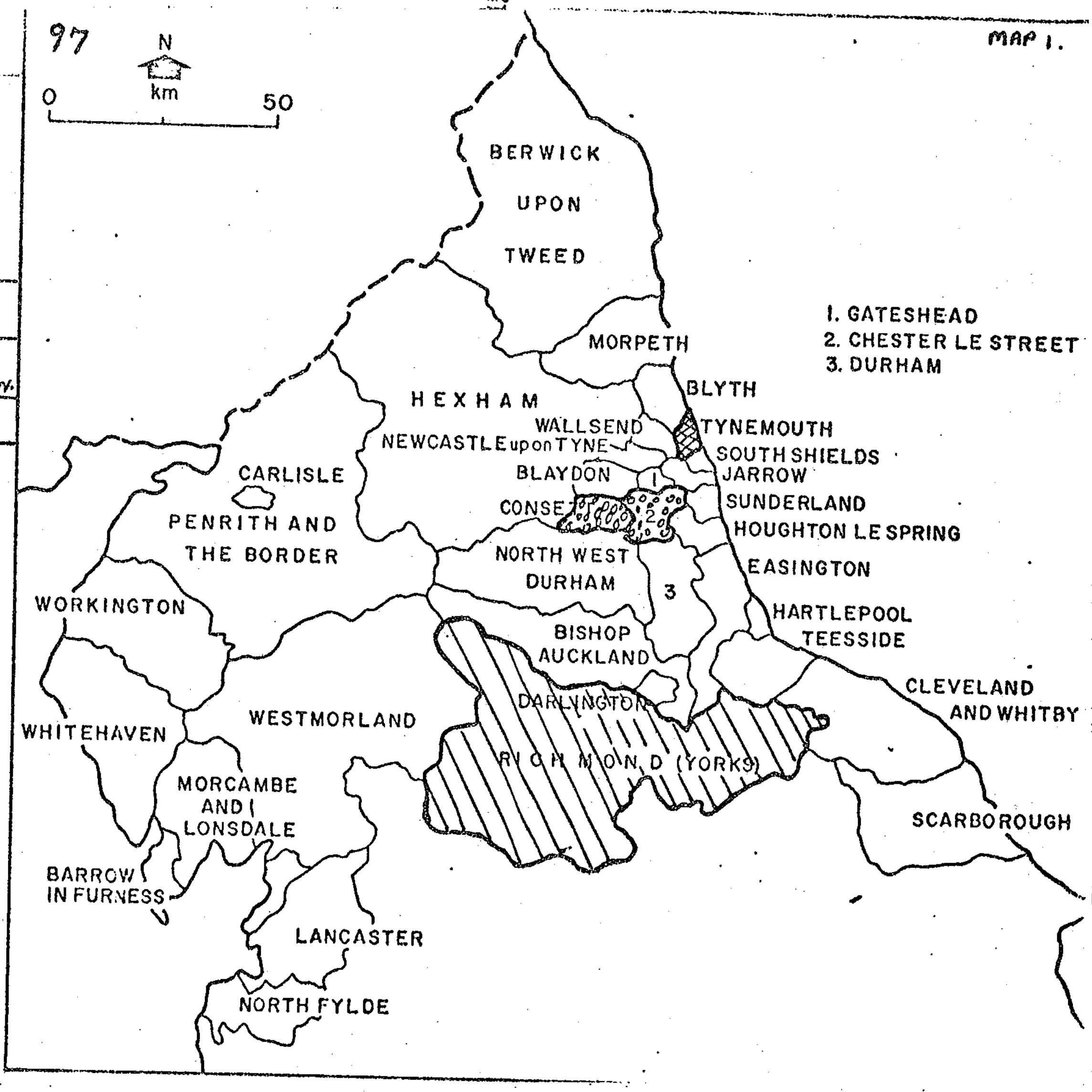
Group P_j	Population of Group - P_j	Expected Percentage of voters for Conservatives.	Percentage deviation of Group P_j from the electorate quota - $\bar{P}$.
I	593,434	36%	12.76%
II	570,247	30%	8.35%
III	469,947	38%	10.70%
IV	528,065	49%	0.34%
V	469,675	40%	10.75%

Average Percentage of votes expected from all the Groups = 38.6%.

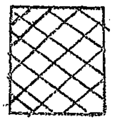
Average Percentage population deviation from the electorate quota for all the Groups = 8.58%.

MAP 1.

SHIFTS DUE TO CONSERVATIVE PARTY (OBJECTIVE) CONSIDERATION.
(See pp. 98)

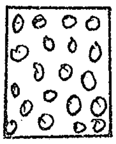


The following units shifted from their allocation *in* Plan A (see pp. 83) to the allocation in Plan C (see pp. 95) due to change in the objective from strictly population consideration to Conservative party consideration. See MAP 1. pp. 97



Tynemouth shifted from II to I.

Richmond (Yorks.) shifted from V to III.



Consett and Chester-le-Street shifted from III to II.

4.5 Conclusions

Non-partisan political constituency apportionment has been a major problem whose solution is necessary for the legitimacy of a democratic system of government.

A survey of the previous approaches to this problem clearly shows that a lot has been done and a lot still needs to be done in order to solve the problem totally. A few of the earlier techniques have the disadvantage of being inexact. A few of them also fail to optimise any objective function and hence do not guarantee the production of an optimal plan. The problem of getting rid of non-contiguous plans seem to be general and hence a few of the earlier works include human scanning for the elimination of non-contiguous groups. They nevertheless have an advantage of running fast in the computer.

The technique that I have presented in this work is a practicable method. It involves little time and effort and after the initial input, it does not involve human scanning to determine an optimal plan or a non-contiguous plan. It can be applied to a state of any size when once the necessary data for phase one has been supplied no more human intervention is necessary. The data for phase two is a function of phase one and the production of the optimal plan is a function of phase two.

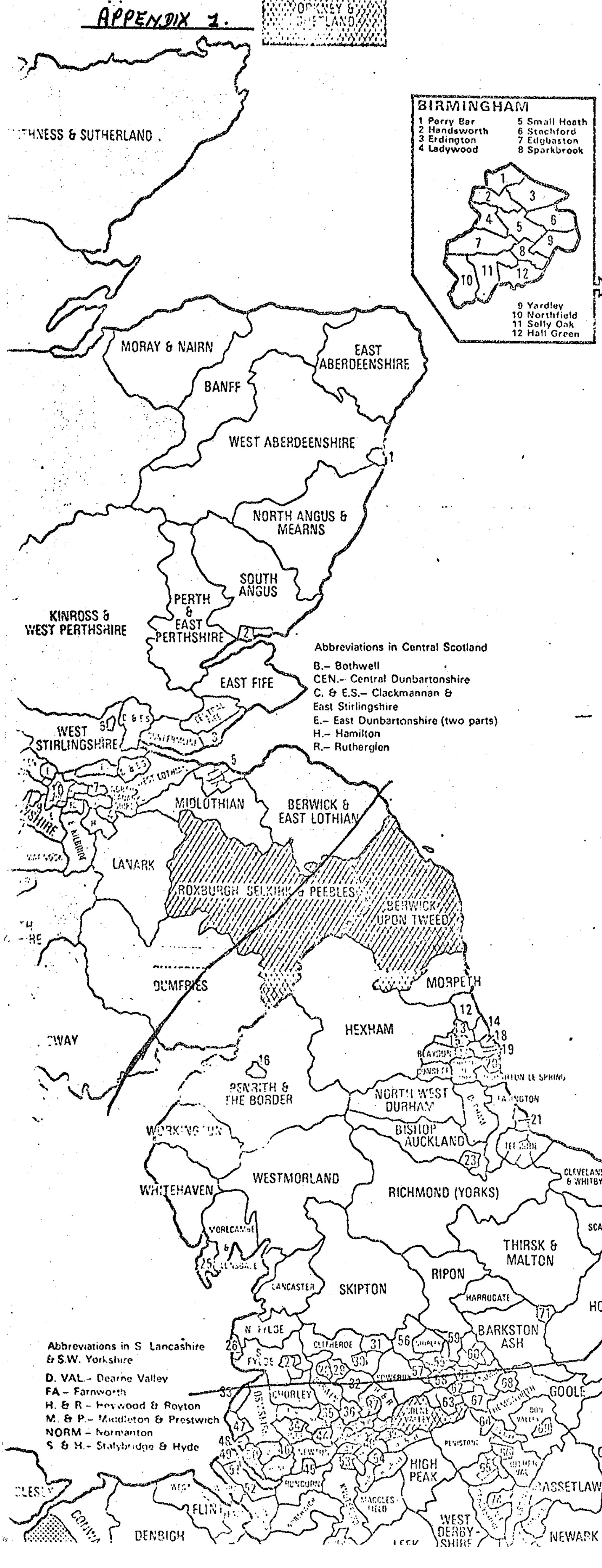
I shall like to suggest that using population as a criteria for acceptability, the lower limit of percentage population deviation to be used should be 5% while 15% should be the upper limit. The groups generated in phase one using such α_S^s would certainly provide a satisfactory plan for most areas.

The algorithm would provide a much more interesting result if smaller population units are used for the allocation of say the Council wards or House of Commons constituencies.

The algorithm could still be improved with regards to the automatic merging of phase one with phase two; when that is accomplished I claim that the electoral boundary partitioning problem can be satisfactorily and objectively solved by this method.

APPENDIX 1.
Map.

APPENDIX 1.



its position on the map

29 Accrington
85 Aldridge & Brownhills
53 Altrincham & Sale
39 Ashton-under-Lyne
64 Barnsley
25 Barrow-in-Furness
Basildon
106 Bath
61 Batley & Morley
52 Bebington & Ellesmere Port
51 Birkenhead
90 Birmingham
28 Blackburn
26 Blackpool
Blaydon
12 Blyth
35 Bolton
48 Bootle
122 Bournemouth
55 Bradford
58 Brighouse & Spenborough
120 Brighton
105 Bristol
30 Burnley
36 Bury & Radcliffe
95 Cambridge
16 Carlisle
Cheadle
103 Cheltenham
111 Chertsey & Walton
74 Chesterfield
112 Christchurch & Lymington
91 Coventry
47 Crosby
23 Darlington
76 Derby
62 Dewsbury
69 Doncaster
83 Dudley
42 Eccles
Epsom & Ewell
Esher
99 Eton & Slough
108 Exeter
124 Fareham
Farnworth
17 Gateshead
117 Gillingham
104 Gloucester
113 Gosport
72 Grimsby
86 Halesowen & Stourbridge
57 Halifax
21 Hartlepool
118 Hastings
114 Havant & Waterloo
Hazel Grove
119 Hove
63 Huddersfield
Ince
96 Ipswich
19 Jarrow
56 Keighley
70 Kingston-upon-Hull
60 Leeds
92 Leicester
44 Leigh
73 Lincoln
50 Liverpool
97 Luton
40 Manchester
Middleton & Prestwich
31 Nelson & Colne
78 Newcastle-under-Lyme
15 Newcastle-upon Tyne
93 Northampton
94 Norwich
75 Nottingham
Nuneaton
38 Oldham
100 Oxford
Peterborough
110 Plymouth
68 Pontefract & Castleford
107 Poole
125 Portsmouth
27 Preston
59 Pudsey
101 Reading North
Reigate
37 Rochdale
116 Rochester & Chatham
32 Rossendale
66 Rotherham
46 St Helens
41 Salford
65 Sheffield
Solihull
123 Southampton
115 Southend-on-Sea
33 Southport
18 South Shields
Spelthorne
54 Stockport
77 Stoke-on-Trent
43 Stretford
20 Sunderland
89 Sutton Coldfield
102 Swindon
Teesside
Thanet
Thurrock
109 Torbay
14 Tynemouth
67 Wakefield
49 Wallasey
13 Wallsend
81 Walsall
84 Warley
45 Warrington
98 Watford
82 West Bromwich
34 Wigan
80 Wolverhampton
88 Worcester
121 Worthing
71 York

APPENDIX 2.

Phase one Computer Program.

APPENDIX 2.

```
$SIGNON MAL2 T=5 P=8
CHARGING RATE = UNIVERSITY, BATCH
**LAST SIGNON WAS: 14:52:10   TUE 13-MAY-80
 USER "MAL2" SIGNED ON AT  12:40:58 ON FRI 16-MAY-80
MAIL:  1 MESSAGE,  TOTAL LINES  9


$LIST ELL1
          DIMENSION IX(29),IY(29),IB(29,29),IT(29),IM(29),IN(29),
        1 IC(60),ILI(60),ILA(60),INNAME(29),INAME(29)
          REAL*8  INNAME,INAME
C           STEP 0. READ  IN  ALL  VALUES: UNITS POP-
C           LATION,0-1 MATRIX,PERCENTAGE OF VOTERS  IN
C           EACH  UNIT AS AT THE LAST ELECTION.
        6 READ(5,7,END=100)N,IP
        7 FORMAT(2I4)
          READ(5,1)(IX(I),I=1,N)
        1 FORMAT(10I7)
          DO 20 I=1,N
       20 READ(6,30)(IB(I,J),J=1,N)
       30 FORMAT(20I4)
          DO 50 I=1,N
       50 READ(9,40)IT(I),IM(I),IN(I)
       40 FORMAT(3I4)
          READ(10,65)(INAME(I),I=1,N)
       65 FORMAT(10A7)
C           STEP 1 BEGINS THE TOTAL IS CALCULATED.
          ITOT=0
          DO 12 I=1,N
          ITOT=ITOT+IX(I)
C           STEP 2  THE ELECTORATE QUOTA IS CALCULATED.
       12 CONTINUE
          MEAN=ITOT/5
          IPAH=(0.15*MEAN)
C           STEP 3 THE MAXIMUM AND MINIMUM ALLOWABLE POPULATION
C          DEVIATION FOR THE AREA IS CALCULATED FROM THE
C           PERCENTAGE DECLARED.
          MAX=IPAH+MEAN
          MIN=MEAN-IPAH
          IK=100*IP
          ICON=0
          ILAB=0
          ILIB=0
          ISUM=0
          ICONS=0
          ILIBP=0
          ILABP=0
          ID=0
          LK=15
          WRITE(6,15)
          WRITE(6,14)ITOT,MEAN,MAX,MIN,IK
          WRITE(6,77)
          WRITE(6,70)IX
          WRITE(6,38)
          DO 160 I=1,N
      160 WRITE(6,150)(IB(I,J),J=1,N)
          WRITE(6,90)
          WRITE(6,990)
          DO 165 I=1,N


      165 WRITE(6,155)IT(I),IM(I),IN(I)
          WRITE(6,109)
          WRITE(6,180)INAME
          WRITE(6,880)
          WRITE(6,260)
C           STEP 4  BEGINS - THE ADDITION OF INDIVIDUAL UNITS
C           LINEARLY STARTING WITH THE FIRST UNIT AND
C           CHANGING TO THE NEXT ACCORDING TO THE ORDER
C           OF LISTING.
          KK=0
          DO 3 I=1,N
          ICOUNT=1
          ISUM=ISUM+IX(I)
          INNAME(I)=INAME(I)
          IC(I)=(IX(I)*IT(I))/100
          ICONS=ICONS+IC(I)
          ILI(I)=(IX(I)*IM(I))/100
          ILIBP=ILIBP+ILI(I)
          ILA(I)=(IX(I)*IN(I))/100
          ILABP=ILABP+ILA(I)
          ID=IABS(MEAN-ISUM)/IK
          ICON=(ICONS*100)/ISUM
          ILIB=(ILIBP*100)/ISUM
          ILAB=(ILABP*100)/ISUM
          IF(ICON.LT.40)ICON=2
          IF(ICON.GE.40.AND.ICON.LT.50)ICON=1
          IF(ICON.GE.50)ICON=0
          IF(ILIB.LT.40)ILIB=2
          IF(ILIB.GE.40.AND.ILIB.LT.50)ILIB=1
          IF(ILIB.GE.50)ILIB=0
          IF(ILAB.LT.40)ILAB=2
          IF(ILAB.GE.40.AND.ILAB.LT.50)ILAB=1
          IF(ILAB.GE.50)ILAB=0
          INC=(ICON*LK)+ID
          LNB=(ILAB*LK)+ID
          IF(ISUM.GT.MAX)GO TO 3
          IY(ICOUNT)=IX(I)
          INNAME(ICOUNT)=INAME(I)
C           STEP 5  BEGINS -  THIS INVOLVES THE ADDITION
C           OF MORE CONTIGUOUS UNITS TO THE INITIAL UNIT
C           OF 4 ABOVE.
```

```
      KK=I+1
      IF(KK.GT.N)GO TO 3
      DO 2 J=KK,N
      DO 9 K=I,N
      IF(IB(I,J).EQ.0)GO TO 2
      ISUM=ISUM+IX(J)
      IC(J)=(IX(J)*IT(J))/100
      ICONS=ICONS+IC(J)
      ILI(J)=(IX(J)*IM(J))/100
      ILIBP=ILIBP+ILI(J)
      ILA(J)=(IX(J)*IN(J))/100
      ILABP=ILABP+ILA(J)
      IF(ISUM.GT.MAX)GO TO 4
      ICOUNT=ICOUNT+1
      IY(ICOUNT)=IX(J)
      INNAME(ICOUNT)=INAME(J)
      ID=IABS(MEAN-ISUM)/IK
      ICON=(ICONS*100)/ISUM
      ILIB=(ILIBP*100)/ISUM
      ILAB=(ILABP*100)/ISUM
      IF(ICON.LT.40)ICON=2
      IF(ICON.GE.40.AND.ICON.LT.50)ICON=1
      IF(ICON.GE.50)ICON=0
      IF(ILIB.LT.40)ILIB=2
      IF(ILIB.GE.40.AND.ILIB.LT.50)ILIB=1
      IF(ILIB.GE.50)ILIB=0
      IF(ILAB.LT.40)ILAB=2
      IF(ILAB.GE.40.AND.ILAB.LT.50)ILAB=1
      IF(ILAB.GE.50)ILAB=0
      INC=(ICON*LK)+ID
      LNB=(ILAB*LK)+ID
C       STEP 6 . THE CONTIGUOUS GROUP OF 5 IS TESTED
C       TO DETERMINE WETHER IT IS WITHIN THE
C       POPULATION RANGE OF 3 ABOVE AND RECORD OF
C       THE GENERATED GROUP WITH ITS ASSOCIATED
C       STATISTICS IS MADE OR A RETURN TO
C       STEP 5 OR STEP 4 DEPENDING ON THE
C       APPROPRIATE STEP DETERMINED BY FURTHER
C       TESTS BELOW.
      IF(ISUM.GE.MIN.AND.ISUM.LE.MAX)WRITE(6,5)ID,INC,LNB,ILIB,
     1 ICON,ILAB,ILIBP,ICONS,ILABP,ISUM,(IY(L),L=1,ICOUNT)
      IF(ISUM.GE.MIN.AND.ISUM.LE.MAX)WRITE(6,17)(INNAME(L),L=1,ICOUNT)
      IF(ISUM.LE.MAX)GO TO 2
    4 IF(ISUM.GT.MAX)ISUM=ISUM-IX(J)
      ICONS=ICONS-IC(J)
      ILIBP=ILIBP-ILI(J)
      ILABP=ILABP-ILA(J)
    9 CONTINUE
    2 CONTINUE
      ICON=0
      ILIB=0
      ILAB=0
      ICONS=0
      ILIBP=0
      ILABP=0
      ID=0
      ISUM=0
    3 CONTINUE
C       STEP 7 . WHEN ALL LINEAR SEARCHES HAVE BEEN MADE
C       FOR POSSIBLE GROUPINGS TERMINATION OCCURS.
      GO TO 6
   15 FORMAT(/,4X,'TOTAL',3X,'MEAN',4X,'MAX',5X,'MIN',5X,'R/FACTOR',/)
   14 FORMAT(10I8,//)
   77 FORMAT(//,'POPULATION FIGURES FOR DIF.UNITS.',/)
   73 FORMAT(/,(10I7),//)
   88 FORMAT(//,'NEARNESS AND CONTIGUITY MATRIX',/)
  150 FORMAT(29I3)
   99 FORMAT(/,'PERCENTAGE OF PARTY MEMBERS IN EACH POP.UNIT',/)
  990 FORMAT(1X,'CON',1X,'LIB',1X,'LAB',/)
  155 FORMAT(3I4)
  199 FORMAT(//,'NAMES OF POP. UNITS/PARLIAMENTARY CONSTITUENCIES'/)
  180 FORMAT(/,(10A7))
  888 FORMAT(///,'POSSIBLE       DISTRICTS',/)
  200 FORMAT(//,1X,'DEVIA',1X,'INDCO',1X,'INDLA',1X,'WIN. POTENTIAL',
     1 1X,'LIB.P',2X,'CON.P',2X,'LAB.P',2X,'TOTAL',10X,'DISTRICTS',
     1 /,19X,'LIB',2X,'CON',2X,'LAB',20X,/)
    5 FORMAT(/,1X,6I5,13I7,/)
   17 FORMAT(/,59X,10A7,/)
  100 STOP
```

APPENDIX 3.

Phase two Computer Program.

APPENDIX 3.

```
 USER "MAL2" SIGNED ON AT  12:41:30 ON FRI 16-MAY-80
MAIL:  1 MESSAGE,  TOTAL LINES  9


$LIST ELL2
            DIMENSION A(110,110),C(110),B(110),CS(110),W(110,110),IX(110),
          1 IS(110)
            DIMENSION IV(110), IT(110), NOTT(110), SUMS(110)
            DIMENSION IPRINT(110),ISAVE(110,110),ISTEP(110),INUM(110)
      C
            EPS = 0.000001
      C
      C     INITIALIZATION-READ IN ALL VALUES;BEGIN STEP 1
      C     TRY SOLUTION X'S=0;IF FEASIBLE,STOP,OBJECTIVE
      C     FUNCTION VALUE=0.0;IF NOT THEN CALCULATES UPPER
      C     BOUND ON F OR USES DATA SUPPLIED UPPER BOUND,
      C     DEPENDING ON WHICH IS THE SMALLER.
      C
            DO 11 I = 1,100
         11 INUM(I) = I
          1 CONTINUE
            ITPCK = 0
            IFEAS = 0
            ICOUNT = 0
            READ(5,500)M,N,INT
        500 FORMAT(20I4)
            IF (M-101)4,9000,9000
          4 DO 2 II=1,110
            B(II) = 0.0
            C(II) = 0.0
            IS(II) = 0
            IV(II) = 0
            IT(II) = 0
            IX(II) = 99
            NOTT(II) = 0
            SUMS(II) = 0.0
            DO 2 JJ=1,100
            A(II,JJ) = 0.0
            W(II,JJ) = 0.0
          2 CONTINUE
            READ(5,510)(C(J),J=1,N)
        510 FORMAT (20F4.0)
            DO 10 I = 1, M
         10 READ(5,510)B(I),(A(I,J),J=1,N)
            READ(5,510)ZBAR
            FZBAR = ZBAR
            DO 20 J    = 1, N
            CS(J) = 0.
            DO 20 I = 1, M
         20 CS(J) = CS(J) + A(I,J)
      C     PRINT MATRIX INPUT
            WRITE(6,12)
         12 FORMAT (1H1,19X,18HOBJECTIVE FUNCTION,/)
            WRITE(6,76)(INUM(J),J=1,N)
         76 FORMAT (12X,10(4X,1HX,I3))
            WRITE(6,77)(C(K),K=1,N)


         77 FORMAT (1H0,12X,10F7.1,/,(13X,10F7.1))
            WRITE(6,81)
         81 FORMAT(1H0,/,20X,11HCONSTRAINTS,//,6X,8HCONSTANT,/)
            DO 84 I = 1,M
            WRITE(6,83)I,B(I),(A(I,J),J=1,N)
         83 FORMAT(1H0,1X,1HG,I2,2X,F6.1,10F7.1,/,(13X,10F7.1))
         84 CONTINUE
            DO 17 I = 1,M
            IF(B(I))19,17,17
         17 CONTINUE
            DO 18 I = 1,100
            IX(I) = 0
         18 CONTINUE
            ZBAR = 0.0
            WRITE(6,86)
         86 FORMAT(1H0,//,16X,26HALL CONSTANTS ARE POSITIVE,/)
            GO TO 1750
         19 WRITE(6,5)
          5 FORMAT(1H1,/,6H  STEP,122X,/,5H NUMB,49X,16HPARTIAL SOLUTION,
          1 55X,4HZBAR,/)
            NUMB = 0
            NS = 0
      C
      C     STEP 2-FINDS SET OF VIOLATED CONSTRAINTS(V)
      C     WHEN PARTIAL SOLUTION(S) HAS A ZERO COMPLETION.
      C     FINDS VALUE FOR OBJECTIVE FUNCTION(FP) FOR
      C     THE CURRENT PARTIAL SOLUTION.
      C
         45 IF (NUMB)645,645,639
        639 IP=100
            IF (NS-100)640,640,642
        640 IP = NS
        642 DO 1001 I = 1,100
            IPRINT(I) = IS(I)
       1001 CONTINUE
        645 FP = 0.
            NW = 0
            IF (NS)51,51,52
         52 DO 50 J = 1, NS
            IF(IS(J)) 50, 50, 55
         55 NW = NW + 1
            JJ = IS(J)
            DO 60 I = 1, M
         60 W(I,NW ) = A(I,JJ)
            FP = FP + C(JJ)
         50 CONTINUE
         51 NW = NW + 1
            DO 65 I = 1, M
         65 W(I, NW) = B(I)
            MV = 0
            DO 70 I = 1, M
            SUMS(I) = 0.
            DO 80 J = 1, NW
         80 SUMS(I) = SUMS(I) + W(I, J)
            IF (SUMS(I) + EPS) 85, 70, 70
         85 MV = MV + 1
            IV(MV) = I
      C
      C     STEP 3-IF NO VIOLATED CONSTRAINTS,THEN
      C     GO TO STEP 9AND RECORD CURRENT SOLUTION.
```

```
C     ARE VIOLATED CONSTRAINTS THEN GO TO
C     STEP 4.
C
   70 CONTINUE
      IF(MV) 200, 200, 90
   90 IP = 100
      IF (MV-100)92,92,94
   92 IP = MV
   94 CONTINUE
C
C     STEP 4-LET ZBAR=CURRENT OBJECTIVE
C     FUNCTION LIMIT.
C
      CLIM = ZBAR - EP
      NW = 0
      NT = 0
      IT(1) = 0
C
C     STEP 5-STORE IN SET T THE GROUP(S) THAT IS NOT
C     IN THE CURRENT PARTIAL SOLUTION WHICH HAS,
C     A) AN OBJECTIVE FUNCTION COEFFICIENT LESS THAN
C     THE UPPER LIMIT.
C     B) A POSITIVE COEFFICIENT IN SOME OF THE
C     VIOLATED CONSTRAINTS OF SET V.
C
      DO 100 J = 1, N
  100 NOTT(J) = 0
      IF (NS)104,104,101
  101 DO 105 J = 1, NS
      ITEMP = IS(J)
      IF (ITEMP) 102,105,105
  102 ITEMP = -ITEMP
  105 NOTT(ITEMP) = 1
  104 DO 110 J = 1, N
      IF(NOTT(J)) 115, 115, 110
  115 IF (CLIM - C(J)) 110, 110, 120
  120 DO 125 I = 1, MV
      ITEMP = IV(I)
      IF(A(ITEMP, J))125, 125, 130
  125 CONTINUE
      GO TO 110
  130 NT = NT + 1
      IT(NT) = J
      NW = NW + 1
      DO 135 I = 1, M
  135 W(I, NW) = A(I,J)
  110 CONTINUE
      IP = 100
      IF (NT-100)106,106,108
  106 IP = NT
  108 CONTINUE
C
C     STEP 6-CHEKS WHETHER SET T IS EMPTY.
C     IF IT IS EMPTY , SETS ITPCK=1 AND GOES TO
C     STEP 11 WHICH IMPLIES BACKTRACKING.
C     IF IT IS NOT EMPTY , GOES TO STEP 7.
C
      IF(NT)1400,1400,138



 1400 ITPCK = 1
      JMAX = 0
      GO TO 1600
C
C          STEP 7
C     CAN THE VIOLATED CONSTRAINTS BECOME
C     FEASIBLE BY THE ADDITION OF ONLY
C     THE GROUUPS IN THE CURRENT SET T?
C     IF YES -SET ITPCK TO 1 AND BACKTRACK VIA
C     STEP 11.
C     IF NO- GO TO STEP 8.
C
  138 DO 140 I = 1, MV
      ITEMP = IV(I)
      DO 145 J = 1, NW
      IF(W(ITEMP, J)) 145, 145, 150
  150 SUMS(ITEMP) = SUMS(ITEMP) + W(ITEMP, J)
  145 CONTINUE
      IF (SUMS(ITEMP) + EPS) 152, 140, 140
  152 CONTINUE
      ITPCK=1
      JMAX = 0
      GO TO 1600
  140 CONTINUE
C
C     S(T)EP 8-SEARCHES THROUGH SET T TO
C     DETERMINE THE  GROUP WITH THE
C     GREATEST COEFFICIENT SUM AND ADDS
C     SAME TO THE NEXT PARTIAL SOLUTION (S),
C     THEN BACKTRACKS.
C
      JMAX = IT(1)
      CSMAX = CS(JMAX)
      IF (NT-2) 156,146,146
  146 DO 155 J = 2, NT
      JTEMP = IT(J)
      IF(CS(JTEMP ) - CSMAX) 155, 160, 170
  160 IF (C(JTEMP) - C(JMAX)) 170, 155, 155
  170 JMAX = JTEMP
      CSMAX = CS(JTEMP)
  155 CONTINUE
  156 CONTINUE
      GO TO 1600
  157 CONTINUE
      NS = NS + 1
      IS(NS) = JMAX
      NUMB = NUMB + 1
      GO TO 45
C
C     STEP 9--THE CURRENT PARTIAL SOLUTION (S) IS
C     COMPLETED BY SETTING ALL GROUPS NOT
C     IN THE CURRENT SOLUTION EQUAL TO ZERO;
C     THIS THEN PROVIDES A PLAN BUT NOT
C     NECESSARILY THE OPTIMAL PLAN.THE
C     OBJECTIVE FUNCTION VALUE FOR THIS
C     SOLUTION X-BAR BECOMES THE NEW ZBAR
C     WHICH IS THE CURRENT UPPER LIMIT FOR THE OBJ. FUNCT.
C
  200 DO 210 J = 1,N
  210 IX(J) = 0
```

```
      ZBAR = 0
      DO 215 J = 1, NS
      JTEMP = IS(J)
      IF(JTEMP) 215, 215, 217
  217 IX(JTEMP) = 1
      ZBAR = ZBAR + C(JTEMP)
  215 CONTINUE
C
C     SETS IFEAS=1, IF A SOLUTION IS FOUND
C     IN STEP 9  SO THAT THE CURRENT
C     POSSIBLE PLAN CAN BE PUT INTO ISAVE
C     FOR RECORDING.
C
      IFEAS = 1
      JMAX = 0
      CLIM = 0
C
C     PRINTS THE POSSIBLE PLANS IN THEIR
C     ORDER OF PRODUCTION.
C
 1000 ICK = (NUMB/INT)*INT - NUMB
      IF(ICK)1600 ,1510,1600
 1510 WRITE(6,1500)NUMB,(IPRINT(I),I=1,20),ZBAR,
     1 (IPRINT(I),I=21,40),(IPRINT(I),I=41,60),
     1 (IPRINT(I),I=61,80),(IPRINT(I),I=81,100)
 1500 FORMAT(1X,I6,6X,20I4,20X,F6.1,/,13X,20I4,33X,/,13X,20I4,33X,/,
     1 13X,20I4,33X,/,13X,20I4,33X,/)
 1600 CONTINUE
      IF (IFEAS-1) 1605,300,300
 1605 IF (ITPCK-1) 157,300,300
C
C     STEPS 10 AND 11.
C     CHECK WHETHER ALL THE GROUPS IN SET (S) ARE
C     NEGATIVE,THAT IS FATHOMED.IF THERE EXIST
C     GROUPS NOT YET FATHOMED THEN LOCATE THE
C     RIGHTMOST GROUP IN SET (S), REPLACE IT WITH
C     ITS COPLEMENT(NEGATIVE) AND THEN DROP ALL
C     THE GROUPS  TO THE RIGHT;BACKTRACK TO
C     STEP 2.IF ALL ARE FATHOMED , TERMINATE..
C
  300 NEWS = NS
      DO 220 J = 1, NS
      JJ = NS - J + 1
      IF(IS(JJ)) 225, 225, 230
  225 NEWS = NEWS - 1
  220 CONTINUE
      GO TO 400
  230 IS(JJ) = -IS(JJ)
      NS = NEWS
      IF (IFEAS-1) 1512,1508,1508
 1508 IF (ITPCK-1) 1511,1512,1512
 1511 IF (100-ICOUNT) 1512,1512,1509
 1509 ICOUNT = ICOUNT + 1
      ISTEP(ICOUNT) = NUMB
      DO 1510 I = 1,N
      ISAVE(ICOUNT,I) = IX(I)
 1510 CONTINUE
 1512 IFEAS = 0
      ITPCK = 0
      NUMB = NUMB + 1


      GO TO 45
C
C     STEP 12-TERMINATE AND RECORD THE FINAL SOLUTION
C     AS THE OPTIMAL PLAN,IF THERE IS NO SOLUTION
C     THEN THERE IS NO POSSIBLE PLAN WITH THE ALPHA
C     USED OR THE UPPER LIMIT OF THE OBJECTIVE
C     FUNCTION DEFINED IS TOO LOW, HENCE PERCENTAGE
C     DEVIATION SHOULD BE REDEFINED .
C
  400 WRITE(6,1619)
 1619 FORMAT (1H0)
      IF(IX(1)-100)1630,1615,1615
 1615 WRITE(6,1620)FZBAR
 1620 FORMAT(1H ,4X,80HTHERE IS NO FEASIBLE SOLUTION WITH A VALUE FOR TH
     2E OBJECTIVE FUNCTION LOWER THAN,F7.1,24H, THE INITIAL ZBAR VALUE)
      GO TO 1
 1630 DO 1700 I = 1,ICOUNT
      WRITE(6,1650)ISTEP(I),(ISAVE(I,J),J=1,20),(ISAVE(I,J),J=21,40),
     1 (ISAVE(I,J),J=41,60),(ISAVE(I,J),J=61,80),(ISAVE(I,J),J=81,100)
 1650 FORMAT (1H0,////,4X,23HPOSSIBLE         PLAN, STEP,I5,2X,/,
     1 17X,20I4,/,17X,20I4,/,17X,20I4,/,17X,20I4,/,17X,20I4,/)
 1700 CONTINUE
 1750 WRITE(6,1800)(IX(I),I=1,20),(IX(I),I=21,40),(IX(I),I=41,60),
     1 (IX(I),I=61,80),(IX(I),I=81,100)
 1800 FORMAT (1H0,/, 15X,16HBEST PLAN  FOUND,2X,/,
     1 17X,20I4,/,17X,20I4,/,17X,20I4,/,17X,20I4,/,17X,20I4,/)
      WRITE(6,1900) ZBAR
 1900 FORMAT (1H0,28X,38HOPTIMAL VALUE OF OBJECTIVE FUNCTION = , F10.4)
      GO TO 1
 9000 STOP
      END
END OF FILE.
```

$SIG

BIBLIOGRAPHY

AMERICAN POLITICAL SCIENCE ASSOCIATION, Committee on Congressional Reapportionment. The reapportionment of Congress. American Political Science Review, (1951)[28], 45, pp. 153-157.

BALINSKI, M.L., "Integer Programming: Methods, Uses, Computation", Management Science, 12 (3), pp. 253-313 (Nov. 1965)[6].

BEALE, E.M.L., "An Alternative Method of Linear Programming", Proc. Cambridge Phil. Soc., 50 (4), pp. 513-523 (1954)[5].

DANTZIG, G.B., "On the Significance of Solving Linear Programming Problems with some Integer Variables", Econometrica, 28 (1), pp. 30-44. (Jan. 1960)[7].

EUROPEAN ASSEMBLY ELECTIONS ACT (1978)[32]. pp. 1-5.

EUROPEAN ASSEMBLY ELECTIONS ACT (1978)[33]. pp. 11.

EUROPEAN ASSEMBLY ELECTIONS ACT (1978)[41], pp. 1-11.

FORREST, E. (1965)[22]. "Apportionment by Computer', American Behavioral Scientist, 7, 23 and 35.

GALE, D., H.W.KUHN, and A.W.TUCKER; "Linear Programming and the Theory of Games", in T.C.Koopmans (ed.), Cowles Commission Monograph, No. 13 (19)[3], pp. 317-329.

GARFINKEL, R.S., and NEMHAUSER, G.L., (1972)[4]: Integer Programming. John Wiley, pp. 49-53.

GARFINKEL, R.S., and NEMHAUSER, G.L. (1972)[13]: Integer Programming. John Wiley, pp. 130-143.

GARFINKEL. R.S. and NEMHAUSER, G.L. (1972)[15]: Discrete Optimization, Integer Programming and Network Analysis for Management Decisions. Prentice Hall. pp. 64-65, pp. 218-247.

GARFINKEL, R.S. and NEMHAUSER, G.L. (1972)[16]: Integer Programming. John Wiley. pp. 108-152, pp. 312-318.

GARFINKEL, R.S., and NEMHAUSER, G.L. (1972)[18]: Integer Programming. John Wiley. pp. 123-124.

GARFINKEL, R.S.: "Optimal Political Districting". Ph. D.Thesis, The John Hopkins University, Oct. (1968)[25]. Also as Working Paper 6812 College of Business Administration, University of Rochester.

GARFINKEL, R.S., and NEMHAUSER, G.L. "Optimal Political Districting by Implicit Enumeration Techniques", Management Science Vol. 16, No. 8, pp. B495-B508 (1970)[39].

GARFINKEL, R.S., and NEMHAUSER, G.L. "The Set Partitioning Problem with Equality Constraints", Operational Research, Vol. 17 (1969)[40], pp. 848-856.

GEOFFRION, A.M. (1972)[19]: Perspectives on Optimization. Addison Wesley. pp. 167-169.

GLOVER, F. "A Multiphase-Dual Algorithm for the Zero-One Integer Programming Problem". Operations Research Journal, Vol. 13, (1965)[20]. pp.879-917.

GREENBERG, H.: Integer Programming: Mathematics in Science and Engineering, 76, Academic Press, London and New York, pp. 66-67, (1971)[11].

HADLEY, G., (1962)[2]: Linear Programming. Addison Wesley. pp. 108-170.

HARRIS, C.C. Jr.: "A Scientific Method of Districting". Behavioral Science Quarterly, (1964)[31], 9, pp. 219-225.

HARRIS, C.C. Jr.: "A Scientific Method of Districting". Behavioral Science Quarterly, (1964)[34], 9, pp. 219-225.

HU, T.C., (1969)[1]: Integer Programming and Network Flows, Addison Wesley, pp. 26-45.

HU, T.C. (1969)[17]: Integer Programming and Network Flows, Adison Wesley, pp. 39.

LANGMAACK, H.: Algorithm 263A, Gomory I. C.A.C.M., 8, (1965)[12].

MILLS, G. "Determination of Local Government Electoral Boundaries", Operations Research Journal, Vol. 18, (1967)[26]. pp. 243-255.

MILLS, G. "Determination of Local Government Electoral Boundaries", Operations Research, Vol. 18 (1967)[38], pp. 247.

PLANE, D.R., and MCMILLAN, C. Jrn. (1971)[15]: Discrete Optimization, Integer Programming and Network Analysis for Management Decisions. Prentice Hall. pp. 64-65, pp. 218-247.

REPORT, EUROPEAN ASSEMBLY CONSTITUENCIES, England (1978)[43].

TAHA, H.A. (1976)[14]: Operations Research, An Introduction, MacMillan. pp. 269-270.

THE TIMES GUIDE TO THE HOUSE OF COMMONS (1979)[42].

THORESON, J. and LIITTSCHWAGER, J.M., "Computers in Behavioural Science: Legislative districting by Computer Simulation", Behavioral Science, Vol. 12. (1967)[24], pp. 237-247.

VICKREY, V. (1961)[21]: "On the Prevention of Gerrymandering", Political Science Quarterly, 76, 105-111.

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)[27].

WEAVER, J.B., HESS, S.W., SIEGFELDT, H.I., WHELAN, J.N. and ZITLAN, P.A.: "Non Partisan Political Redistricting by Computer". Operations Research Journal Vol. 13 (1965)[23], pp. 998-1006.

WEAVER, J.B., and HESS, S.W.: "A Procedure for Non Partisan districting: Development of Computer Techniques", Yale Law Journal, (1963)[29], 73, pp. 299.

WEAVER, J.B., and HESS, S.W.: "A Procedure for Non Partisan districting: Development of Computer Techniques", Yale Law Journal, (1963)[30], 72, pp. 288-308.

WEAVER, J.B., and HESS, S.W. "A Procedure for Non Partisan Districting: Development of Computer Techniques". Yale Law Journal, (1963)[35], 73, pp. 301.

WEAVER, J.B., et al. "Non Partisan Redistricting by Computer". Operations Research Journal Vol. 13 (1965)[36], pp. 999-1003.

WEAVER, J.B., et al. "Non Partisan Redistricting by Computer". Operations Research Journal Vol. 13 (1965)[37], pp. 999.

WILSON, R.B.: "Stronger Cuts in Gomory's All Integer Programming Algorithm". Operations Research Journal, Vol. 15, pp. 155. (1967)[8].

WILSON, R.B.: "Stronger Cuts in Gomory's All Integer Programming Algorithm". Operations Research Journal, Vol. 15, pp. 156. (1967)[9].

WILSON, R.B.: "Stronger Cuts in Gomory's All Integer Programming Algorithm". Operations Research Journal, Vol. 15, pp. 155-157. (1967)[10].