Algorithms for the School Districting Problem

By

Hisashi MINE*, Katsuhisa OHNO*, and Isao MIYAJI**

(Received September 30, 1981)

Abstract

This paper presents two algorithms for finding solutions to the problem of school districting, which is the dividing of an administrative area into some school districts consisting of several population units. The problem is formulated as a set partitioning problem, after having enumerated the feasible districts satisfying all the given requirements. An algorithm for finding an exact optimal solution is first proposed. Using the population units as indivisible elements, the first phase enumerates all the feasible districts which satisfy the given requirements, such as contiguity, capacity, and so on. The second phase determines the optimal school districting that minimizes the sum of the distances traveled by all students.

Since the computation time of the exact algorithm increases very quickly as the number of population units increases, an improved algorithm is derived for finding an optimal or near-optimal solution within a reasonable computation time. This algorithm constructs the core of each school district before enumerating the feasible districts. The core of each school district is composed of the population units which are assigned to the school, with the minimal distances traveled until the given bound on the population is satisfied. Computation results show that the improved algorithm can find an optimal or near-optimal solution for a problem having 122 units within one minute.

1. Introduction

The problems of school districting have arisen from the viewpoint of racial balance in the United States. On the other hand, school administrators in Japan have been facing changing enrollments that require new school redistricting. A problem on school redistricting for a junior high school actually occurred several years ago in the city where one of the authors lived. It was due to the remarkable differences in the number of students in the schools.

Many of the school districting problems in the United States are taken into consideration such as attending school by bus and assigning students in a certain

* Department of Applied Mathematics and Physics, Faculty of Engineering. Kyoto University. Kyoto 606, JAPAN.

** Tsuyama Technical College. Tsuyama 708, JAPAN.

population unit (for short, unit) to some schools [2], [3], [8]. On the other hand, in Japan it is significant to consider the problem of assigning all of the students in a certain unit to only one school [10]. However, the problem where all the students in a certain unit are assigned to only one school and where school districts are constructed from contiguous units does not seem to have yet been considered. The purpose of this paper is to propose exact and improved algorithms for solving the school districting problem with all the given requirements, including those mentioned above. The political districting problem which Garfinkel and Nemhauser considered is similar to the school districting problem [6]. A school district differs from a political district in the existence of a district center, that is, a school.

In Section 2, the school districting problem is explained. The problem is formulated as a set partitioning problem in Section 3. Section 4 gives an exact algorithm by combining an approach for a set partitioning problem with an implicit enumeration method. Section 5 proposes an improved algorithm where the construction of a core of a school district is annexed to the exact algorithm. In Section 6, computation results using the two proposed algorithms are shown for several numerical examples. The effectiveness of the concept of core is shown by the fact that an actual problem having 122 population units could be solved within one minute by the improved algorithm. The possibility of using other optimality criteria and other requirements is discussed in Section 7.

2. School Districting Problem

An administrative area such as a prefecture, a sub-prefecture or a city is assumed to be divided into school districts consisting of several units. Each school district has one and only one pre-assigned schoolhouse which is located in one of its units. To distinguish this unit from other units without any schoolhouse, it is called a school unit (for short, s-unit). School districting is considered as a process of partitioning an administrative area having N units into M school districts, each of which contains one and only one s-unit where $M<N$.

Various requirements should be taken into consideration in the school districting problem. This paper introduces the requirements mentioned in the preceding section: (i) Each district should contain one and only one s-unit; (ii) Each unit should belong to one and only one district; (iii) Each district should consist of contiguous units, that is, no district should contain isolated units; (iv) The number of students in each school should be restricted within specified capacity limits; (v) The distance of each unit from the s-unit should not be greater than a specified exclusion distance, in order to restrict the distances traveled by the students.

In this paper, an optimality criterion for school districting is the minimization of the sum of the distances traveled by all the students. Therefore, the school districting problem (for short, SD problem) is a minimization of the sum of the distances traveled by all the students subject to the above mentioned five requirements (i)-(v).

3. Formulation as a Set Partitioning Problem

The SD problem is one of the combinatorial optimization problems whose computation time tends to increase rapidly as the size of the problems becomes larger. To find the optimal solution to the SD problem, our exact algorithm consists of two phases. In Phase I, the feasible districts are enumerated by an implicit enumeration method. In Phase II, the optimal school district is obtained by regarding the feasible districts as subsets.

Let the elements of the matrix A for the feasible districts be a_{ij}, where for fixed j and for $i=1, \cdots, N$, $a_{ij}=1$ if the feasible district j contains unit i; otherwise $a_{ij}=0$. A feasible district j can correspond to a column vector a_j of the matrix A. Let a set of the feasible districts for each school k be G_k. The set G_k can be expressed by the set of column vectors a_j, such that $a_{kj}=1$. The structure of matrix A is shown in Fig. 1, where the entries of the first to the Mth row are assumed to be the value of unity at the position specified by the symbol $\boxed{1}$. It takes the value of zero at the other positions. The entries of the other rows (i.e. $M+1, \cdots$, Nth rows) take either zero or unity. It is assumed that the matrix A of the feasible districts with N rows and J columns has been obtained, where J is the number of the feasible districts. Matrix A gives the coefficient matrix in a set partitioning problem. Let the labels of s-units be 1 to M, and let the labels of the other units be $M+1$ to N.

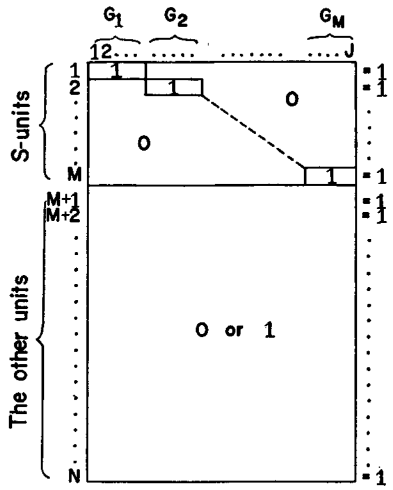


Fig. 1. Matrix A.

Thus, the SD problem means selecting exactly the M column vectors from matrix A, so that the row-wise sum of the entries of the column vectors selected be equal to unity. The first to the Mth rows in the structure of matrix A show that the constraint $\sum_{j=1}^{J} x_j = M$ is not required because one and only one column vector should be selected from among each set G_k in order to satisfy this restriction of the row-wise sum. Then, SD problem can be formulated as a set partitioning problem and expressed as follows:

$$\text{(3.1)} \quad \text{minimize} \quad \sum_{j=1}^{J} c_j x_j ,$$

subject to

$$\text{(3.2)} \quad \sum_{j=1}^{J} a_{ij} x_j = 1 \quad \text{for } i = 1, \cdots, N ,$$

$$\text{(3.3)} \quad x_j \in \{0, 1\} \quad \text{for } j = 1, \cdots, J ,$$

where the weight c_j for $j \in G_k$ is expressed as $\sum_{j=1}^{N} a_{ij} p_i d_{ik}$ in terms of the population p_i of unit i and its distance d_{ik} to school k. The value of the decision variable x_j is as follows: $x_j=1$ if the column vector a_j is selected for the school districting, otherwise $x_j=0$. It is noted that the problem (3.1)$\sim$(3.3) is *NP*-complete [1], [4]. From the structure of matrix A, equation (3.2) can be rewritten as follows:

$$\text{(3.2a)} \quad \sum_{j \in G_k} x_j = 1 \quad \text{for } k = 1, \cdots, M ,$$

$$\text{(3.2b)} \quad \sum_{k=1}^{M} \sum_{j \in G_k}^{J} a_{ij} x_j = 1 \quad \text{for } i = M+1, \cdots, N .$$

These formulations indicate that the optimization can be done by modifying the algorithm for the set partitioning problem [5], so as to use the feature of matrix A.

4. Exact Algorithm

In this section, an algorithm is proposed for solving the SD problem exactly. The exact algorithm consists of two phases: (I) the enumeration of the feasible districts and (II) the optimization. The first phase consists of a systematic method for enumerating all the feasible districts by taking advantage of the structure of the requirements (i) to (v), explained in Section 2. In particular, the contiguity requirement (iii) and the distance requirement (v) simplify the enumeration effort. This phase uses a modified procedure of an implicit enumeration method [7] with the features of the given requirements. This is suggested from the algorithm for the political districting problem [6]. With these enumerated feasible districts, a feasible district matrix A is formed by adjoining together the column vectors associated with each feasible district. The SD problem has been formulated as a set partitioning problem in the preceding section. The second phase is the search for the optimal solution, using a modified form of the search algorithm for the set partitioning problem [5]. In this section, the outline of each phase is only sketched.

4.1 Enumeration of feasible districts

This phase gives a listing of all the feasible districts by the tree search algorithm for each s-unit. In particular, the set of all the feasible districts is partitioned into subsets G_k, the set containing the s-unit k for $k=1,\cdots,M$. Before enumerating set G_k, the labels of s-unit $k(=1,\cdots,M)$ and units $i(=M+1,\cdots,N)$ are renumbered according to the decreasing order of population so as to reduce the computation time.

Let the sets of all s-units and the other units be denoted by S and T, respectively. For each G_k, the procedure starts with the set $B_k=\{i\,|\,d_{ik}\leq e, i\in T\}=\{b_{ik}\}$ of all the units within the exclusion distance e from the s-unit k (requirement (v)), where b_{ik} is the unit number i. From the elements of this set B_k, a contiguous unit to the s-unit k is selected. A candidate set I is constructed from the s-unit and the selected unit for the district. Next, from the elements of set B_k is selected a unit contiguous to some unit of the resulting candidate set I (requirement (iii)). Then, the unit is included to form an augmented candidate set I.

This process is repeated until, the lower bound L_k of the capacity requirement (iv) is satisfied. If this lower bound requirement is satisfied, then it is examined whether the augmented set I satisfies the upper bound U_k (requirement (iv)). If requirements (i), (iii), (iv) and (v) are satisfied, then the augmented candidate set I is a feasible district. Further, from the elements of set B_k, a unit contiguous to some unit of the above augmented candidate set I is chosen, and a new augmented set I' which satisfies requirement (iv) is formed.

When the augmented set I' does not satisfy some of the requirements, a backtrack to the previous set I is made. That is, the unit which has been added lastly to set I' is deleted, and a new unit from set B_k is selected. The new unit is added to the deleted set I. It is examined whether the new augmented set satisfies all the requirements. The procedure is repeated until all the feasible districts are constructed.

Finally, the feasible district matrix A with N rows, described in Section 3, is constructed from the set of all the feasible districts enumerated. First, to each feasible district corresponds an N dimentional column vector, whose components indicate whether corresponding units are included in the feasible district. Next, these column vectors are adjoined to form the feasible district matrix A. The column vectors in each set G_k are rearranged in an increasing order of the corresponding weights c_j.

4.2. Optimization

In Phase II, an optimal school districting is found by using matrix A, shown in Fig. 1. Before performing this phase, the column vector set G_k of matrix A is rearranged and renumbered in an increasing order of the number of elements of set G_k for s-unit k, in order to reduce the computation time. First, the first column vector is selected from set G_1 for the first s-unit. Next, choose a column vector with as small a weight as possible from among each set G_k for each s-unit $k(=2, \cdots, M)$ so as to satisfy equations (3.2a) and (3.2b). In this procedure, it is assumed that one feasible solution has been found. The same procedure leads to the optimal solution by enumerating implicitly a potential set of the feasible solutions which have less weight than the weight of the latest feasible solution obtained in the preceding stages.

5. Improved Algorithm

The numerical examples given in the next section show that the computation time using the exact algorithm presented in the preceding section increases very quickly as the number of units increases. For instance, a problem with 30 units could not be solved within ten minutes by the exact algorithm on the FACOM M-200 computer at the Data Processing Center of Kyoto University. Since the number of units in a practical school districting problem seems to be at least thirty, a more efficient algorithm should be devised so that optimal or near-optimal solutions of practical problems can be obtained within reasonable computation time.

It is noted that the exact algorithm requires the enumeration of all the feasible districts (Phase I). For example, the number of feasible districts in the example with 30 units is 2802, which is the number of decision variables in Phase II. Consequently, in order to devise an efficient algorithm, the number of feasible districts to be considered should be as reasonably few as possible. An improved algorithm based on the following new idea is devised. The units near an s-unit are expected to be in the school district corresponding to the s-unit. The idea forms the heart of the procedure in the improved algorithm. A district composed of those units and the s-unit is called the core of a school district. The core of a school district is constructed by annexing one by one the units having a minimal weight from among all units which satisfy the contiguity requirement (iii). Next, the feasible districts are enumerated, using the first phase in the exact algorithm, by regarding the s-unit in the exact algorithm as the core of a school district. Finally, the second phase in the exact algorithm searches for the optimal solution from the feasible districts. The effectiveness of the concept of core in solving the SD problem is shown by the fact that an actual problem with 122 units could be solved within one minute by the improved algorithm.

The elements of sets S and T are renumbered in a decreasing order of population, respectively. First, for each school $k \in S$, the set $B_k = \{b_{ik}\}$ of units $i \in T$ which satisfy the distance requirement (v) is selected. The elements of set B_k are rearranged in an increasing order of weight w_{ik}, where $w_{ik} = p_i d_{ik}$. Next, to the school $k \in S$, unit $i \in B_k$ with the minimal weight w_{ik} is assigned one by one from the units which satisfy the contiguity requirement (iii) and the upper bound U_k of population (iv). The assignment of units i to school k is repeated until the specified population $L'_k = r_k L_k$ $(0 < r_k \leq 1)$ is satisfied for all schools $k \in S$. It is noted that L'_k is not greater than the lower bound L_k of population (iv). It may occur that a unit is assigned to two schools. In such a case, the unit is assigned to the school with the smaller weight. This procedure constructs the core of a school district for each school. The unit constructing the core is called a core unit. The other unit is called a remaining unit.

The enumeration of feasible districts starts with the set H of all the remaining units $i \in B_k$ for each school k. That is, the remaining units, not belonging to any of the cores of school districts, are annexed to the cores by regarding the cores as the s-units. A feasible district matrix A is typically shown in Fig. 2. Next, the optimal solution for the feasible districts is found by optimization (Phase II), as in the exact algorithm. These procedures result in a rapid decrease in the number of feasible districts, and an appreciable reduction in computation time. The improved algorithm can quickly find the optimal solution if the value of r_k can be properly determined, otherwise a near-optimal solution. In this section, the outline of the improved algorithm has been only sketched. The algorithm will be described in detail in the Appendix.

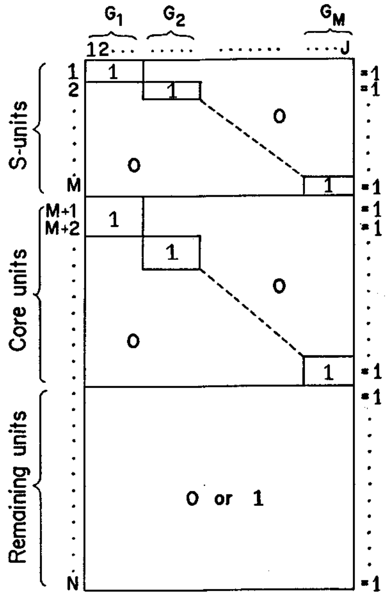


Fig. 2. Feasible district matrix A.

6. Numerical Examples

The algorithms were coded in FORTRAN. Several numerical examples were solved on the FACOM M-200 computer at the Data Processing Center of Kyoto University. The results are given in Table 1. The examples with 70

Table 1. Summary of numerical examples.

No.	No. of units	No. of schools	Values			Improved algorithm		Exact algorithm	
			ε	α	r	Value of obj. func.	Computation time (sec)	Value of obj. func.	Computation time (sec)
1	20	3	40	0.1	0.8	1.82×10^4	0.09	1.81×10^4	10.32
2				0.2		1.82	0.10	1.81	56.18
3	30	4	40	0.1	0.8	1.82×10^5	0.15	—	>600
4				0.2		1.84	0.11	—	—
5	40	4	40	0.1	1.0	3.12×10^5	0.16	—	—
6				0.2		3.11	0.19	—	—
7	70	4	50	0.1	0.8	5.22×10^4	5.88	—	—
8				0.2	0.9	5.06	2.27	—	—
9	122	4	200	0.1	0.9	1.58×10^5	16.99	—	—
10				0.2		1.54	41.70	—	—

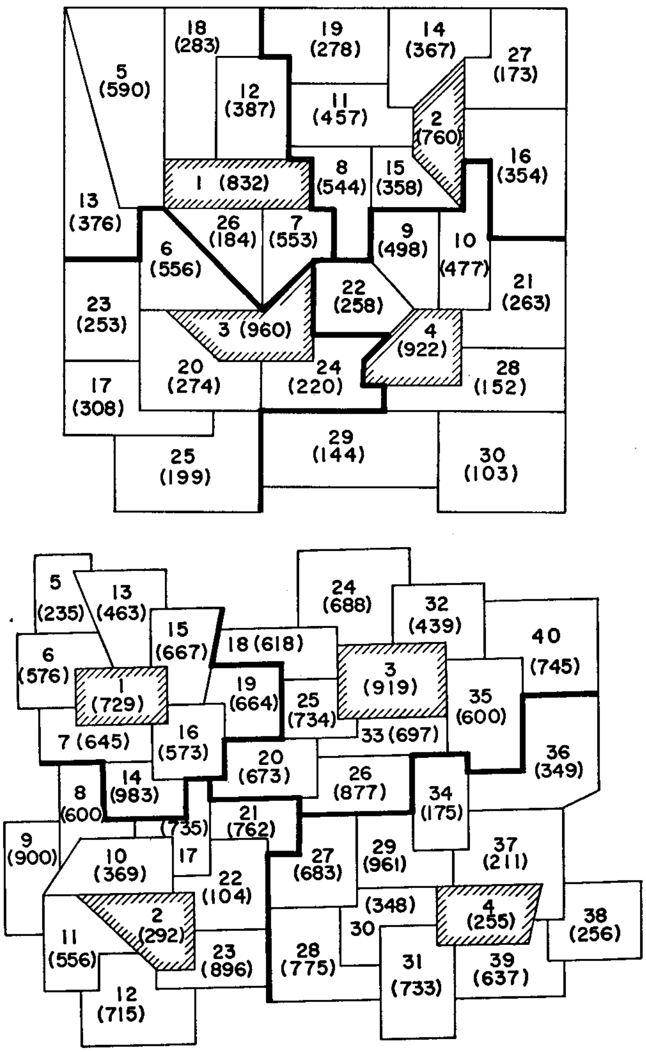
units used data similar to those of Holloway et al. [9]. The last examples with 122 units used actual data taken from the school redistricting problem of the city where one of the authors lived, while the data for the rest of the examples were randomly generated.

In all the numerical examples, the lower bound L_k and the upper bound U_k in the requirement (iv) were set as $L_k=(1-\alpha)\bar{p}$ and $U_k=(1+\alpha)\bar{p}$, respectively. Here, the value $100\alpha(0<\alpha\leq1)$ is the maximum allowable percentage deviation of the population Q_k of each school district from the average district population $\bar{p}$ $(=\sum_{i=1}^{N} p_i/M)$. Moreover, r_k was fixed as $r_k=r$. The exact algorithm in Section 4 found the optimal solutions for examples 1 and 2. The last column of Table 1, however, shows that the computation time increases very quickly as the number of population units increases. The exact algorithm could not solve a numerical example with 30 units within ten minutes. The number of units in practical school districting problems seems to be at least thirty.

The improved algorithm in Section 5 can find the optimal solution if a suitable value of r_k is given. The results of Table 1 show that the improved algorithm takes a very short computation time. For example, it solved the actual school districting problem within one minute. Moreover, the computation time increases very slowly as the number of units increases. The computational experiments show that the improved algorithm is extremely efficient. Since the construction of cores plays a very important role in the improved algorithm, the newly introduced concept of cores has a marked effect on the algorithm efficiency.

Table 1 shows that the number N of population units seems to more significantly affect the computation time than the number M of schools or the values e and α. It seems desirable that the value of r_k used in constructing cores should be about 0.8～0.9 for a satisfactory reduction in the computation time. Since the same values of r_k for all schools k in the these examples are used, the optimal solutions may not be obtained for all the examples. As shown in examples 1 and 2, however, the solutions obtained are quite near-optimal.

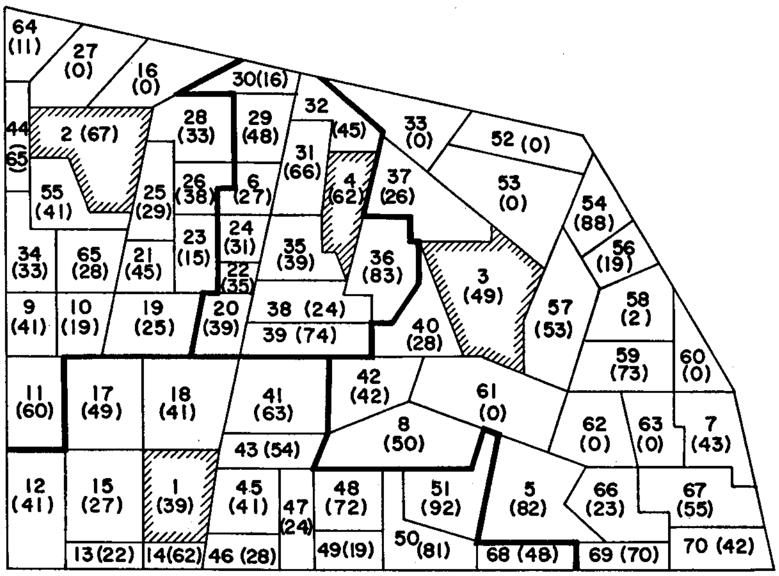
Fig. 3 will give a better understanding of the improved algorithm. The data of examples 3, 6 and 8 are shown in Fig. 3. The shaded areas represent the s-units. Each unit is identified by an integer together with its population in paren-



(1) Example 3

(2) Example 6

Fig. 3.



(3) Example 8

Fig. 3.

thesis. The school districtings obtained by the improved algorithm are shown by heavy boarders.

7. Discussion and Conclusion

In this paper, the optimality criterion is the minimization of the sum of the distances traveled by all the students. That is, weight c_j in equation (3.1) is given by $c_j=\sum_{i=1}^{N} a_{ij}p_i d_{ik}$ for $j\in G_k$, where d_{ik} means the distance between the centers of unit i and school k, and G_k means a set of feasible districts for the school k. The distance d_{ik}, however, may be specified by the actual travel distance or time between unit i and school k. Moreover, weight c_j may also be given by: $c_j=\sum_{i=1}^{N} a_{ij}p_i d_{ik}^2$ or, more generally, $c_j=\sum_{i=1}^{N} a_{ij}f(p_i, d_{ik})$, where f is any real valued function. In these cases, the algorithms can be applied without any modification. When some units should belong to the same district because of a social background, those units can be regarded as one unit. Consequently, the data can be combined as if it were of one unit. It is also noted that some other requirements, if they exist, can be easily incorporated in the proposed algorithms.

In this paper, the SD problem has been formulated as the set partitioning problem. An exact algorithm has been proposed for finding an optimal school districting in Section 4. In Section 5, the improved algorithm has been proposed for a quicker way of finding an optimal or near-optimal school districting. The improved algorithm gives sufficiently suitable solutions to ten numerical examples in a very short time. The new idea of cores played an important role, because the enumeration of feasible districts is materially reduced in number. The results of Table 1 show that an actual school districting problem with 122 units can be solved by the improved algorithm within one minute. The computation time by the improved algorithm increases very slowly as the number of units increases. Moreover, the improved algorithm is suitable for an interactive procedure checking various solutions by changing the parameters. It is noted that thirty elements of the feasible district matrix are stored within one word. Therefore, the improved algorithm requires about $J \cdot N/30$ words. These facts show the efficiency of the improved algorithm.

The improved algorithm uses the parameter r_k in constructing cores of school districts. It is not easy to determine the best value of the parameter r_k. Therefore, the solutions obtained by the improved algorithm are usually approximate ones. It is also quite difficult to estimate the differences between optimal solutions and approximate ones obtained by the algorithm. However, it seems reasonable to believe that the solutions obtained are near-optimal.

Appendix

Details of the Improved Algorithm

As noted in Section 5, the improved algorithm for the school districting problem consists of three phases. The exact algorithm is derived by deleting Phase I in the improved algorithm.

(1) Phase I: Construction of a Core

Step 1 (*Enumeration of assignable units*): The labels of s-units $k(=1,\cdots,M)$ and units $i(=M+1,\cdots,N)$ are renumbered according to the decreasing order of population. Set $S=\{1,2,\cdots,M\}$. For each school $k\in S$, find the set $B_k=\{b_{ik}\}$, as defined in Section 4. If some units do not belong to any set B_k, then there is no feasible solution and the algorithm is terminated. Otherwise, for each set B_k, rearrange the element b_{ik} in an increasing order of the weight w_{ik}.

Step 2 (*Assignment of s-units*): Assign each s-unit $k\in S$ to the corresponding school k, that is, $S_k=\{k\}$, where S_k is the set of units assigned to school k. Set $j=1$.

Step 3 (*Assignment of units*): Let k be the jth element of set S. For an assignment to school $k\in S$, select unit $i\in B_k$ with the minimal weight w_{ik} from among those units which have the positive element b_{ik}, and which are contiguous to any unit of set S_k, and which also satisfy the upper bound U_k, that is,

$Q_k+p_i \leq U_k$, where $Q_k=\sum_{i\in S_k} p_i=p(S_k)$. Set $b_{ik}=-b_{ik}$, where a negative element of b_{ik} means that the unit i had been checked already. If there is not such a unit, then set $j=j+1$ and repeat this step until $j>|S|$. If the unit i selected above has been already assigned to another school $k'(\neq k)$, then go to Step 5 in order to resolve the conflict. Otherwise, set $S_k=S_k \cup \{j\}$ and $j=j+1$. If $j>|S|$, then go to Step 4. Otherwise, repeat this step.

Step 4 (*Test for the specified population*): Examine whether the population $Q_k=P(S_k)$ for each school k is greater than the specified population $L'_k(=r_kL_k)$. If $Q_k \geq L'_k$ for all schools $k\in S$ or all the elements of the set B_k for any school $k\in S$ are exhausted, then go to Phase II. Otherwise, replace S by the set of schools for which the relation $Q_k<L'_k$ holds. If S is empty, then go to Phase II. Otherwise, return to Step 3.

Step 5 (*Comparison between two weights*): Compare the weight w_{ik} with the weight $w_{ik'}$ for unit $i(=-b_{ik}=-b_{ik'})$. If $w_{ik}>w_{ik'}$, then unit i is left in the set $S_{k'}$ and return to Step 3 in order to find another unit for an assignment to school k. Otherwise, remove the unit i from set $S_{k'}$ and add it to set S_k. Check the contiguity constraint for all the units which are added to set $S_{k'}$ after unit i has been assigned to school k'. If any units in set $S_{k'}$ do not satisfy the contiguity constraint, then remove all such units from set $S_{k'}$ and set $b_{ik'}=-b_{ik'}$ for all of them. Set $j=j+1$ and return to Step 3.

(2) Phase II: Enumeration of Feasible Districts

Step 6 (*Initialization*): From each set B_k exclude the core units (i.e. $b_{ik}<0$). The tested units are represented by a vector V, with the labels of the units as elements appearing in the order of testing. The vector $V=(v_i)$ is used to indicate the units included in the current partial solution, that is, $v_i=i$ if unit i is to be included, otherwise $v_i=-i$. First, set $k=1$, $I=S_1$, $G_k=\phi$ and $j=1$. Next, set $V=(S_1)$, which means to give all the elements of set S_1 to the vector V as its elements. If $P(I)\geq L_k$, then go to Step 8.

Step 7 (*Unit annexation*): From set B_k, find the unit i with the minimal index of set B_k in the set of units which satisfies the upper bound U_k of population, and which is contiguous to any unit of set S_k, and is not included as an element of the vector V. A rearrangement of set B_k in Step 1 results in a reduction of computation time in this step. If such unit i does not exist, then go to Step 9. Otherwise, set $V=(V,i)$ and $I=I\cup\{i\}$. If the relation $P(I)\geq L_k$ is satisfied, then go to Step 8, otherwise repeat Step 7 in order to annex other units.

Step 8 (*Registration of feasible district*): A feasible district has been found. If district j agrees with any district $n \in G_k$, then set $v_i = -i$, $I = I - \{i\}$ and return to Step 7. Otherwise, record it in the matrix A as a column vector, such that the elements a_{ij}, corresponding with units i which are included in the feasible district, take unity. Then, set $G_k = G_k \cup \{j\}$ and $j = j+1$, and return to Step 7.

Step 9 (*Backtracking*): Let m be the rightmost non-negative unit of the vector V, Replace m by $-m$, and remove all entries to the right of m from vector V, and all the corresponding units from set I. If $m \notin S_k$, then return to Step 7. Otherwise, as every vector a_j for s-unit k has been enumerated, rearrange the elements of set G_k for s-unit k so that $c_j \leq c_l$ for $i, l \in G_k$, and then set $k = k+1$. If $k > M$, then go to Step 10. Otherwise, set $I = S_k$, $V = (S_k)$ and $j = 1$. If $P(I) \geq L_k$, then return to Step 8; otherwise, return Step 7.

Step 10 (*Test for the zero row vector*): If the ith row vector $\boldsymbol{r}_i$ of the matrix A is not a null vector, set $D = \phi$ where D is the set of districts in the optimal solution, and go to Step 11. Otherwise, since no feasible solution exists, from each set S_k exclude one by one the recently annexed core unit, as some core units seem to have to be assigned to another school, and return to Step 6 with the constructed set G_k. If any set S_k is empty, then stop, since there is no feasible solution.

Step 11 (*Test for unit row vector*): If the ith row vector $\boldsymbol{r}_i$ of the matrix A is a unit vector with the jth element unity, then $x_j = 1$ in every feasible solution. The partial solution D should include the feasible district j, that is, set $D = D \cup \{j\}$. Then, let k be the s-unit that is included in the column vector $\boldsymbol{a}_j$. Every column vector $\boldsymbol{a}_l$ with $l \in G_k$ $(l \neq j)$ can be deleted from the matrix A, and every column vector $\boldsymbol{a}_q$ for column $q \notin G_k$ such that $a_{pq} = a_{pj} = 1$ $(q \neq j)$ for some row p can be also deleted. In addition, the row vector $\boldsymbol{r}_i$ and every row vector $\boldsymbol{r}_p$ such that $a_{pj} = 1$ for some row p may be deleted. If some $\boldsymbol{r}_i$ results in the unit vector, then repeat Step 11. Otherwise, go to Phase III.

(3) Phase III: Optimization

Step 12 (*Initialization*): Rearrange all the feasible district sets so that $|G_l| \leq |G_k|$ for $l < k$, where $|\cdot|$ denotes the number of elements of the set. Set $z_0 = \infty$ and $d_1 = 0$, where d_k is an indicator. Compute $d_k = d_{k-1} + |G_{k-1}|$ for $k = 2, \cdots, M$. Set $k = 0$.

Step 13 (*Choosing next s-unit*): Set $k = k+1$ and $n = d_k$. If $D \cap G_k \neq \phi$, then return to the top of this step. Otherwise, go to Step 14.

Step 14 (*Constructing the partial solution*): Find the column where $n<j\in G_k$, $Q(D)\cap H_j=\phi$, and $z(D)+c_j<z_0$, where $Q(D)$ is the set of units included in all districts belonging to the set D, where H_j is the set of units included in district j, and where $z(D)$ is the weight $\sum_{j\in D} c_j$ of the partial solution D. If there is such a column j, then go to Step 15. Otherwise, go to Step 16.

Step 15 (*Backtracking*): Set $D=D\cup\{j\}$. If $k\neq M$, then return to Step 13. Otherwise, if $Q(D)\neq K$, then go to Step 16, where K is the set of all units. If $Q(D)=K$, then a new and better solution has been obtained. Set $z_0=z(D)$. Let l be the feasible district included in the set $D\cap G_M$. Exclude the feasible districts that need not be considered as follows: Set $G_m=G_m-\{j\mid z_0\leq c_j, j\in G_m\}$ for $m=1,\cdots,M-1$ and $G_M=G_M-\{j\mid j\geq l, j\in G_M\}$ and $D=D-\{l\}$, as the elements of set G_k are rearranged in an increasing order of weight c_j. Go to Step 16.

Step 16 (*Test solution*): If $D=\phi$, then go to Step 17. Otherwise, let $\{d\}$ be the element added most lately to set D, put $D=D-\{d\}$, and put $k=k-1$. Return to Step 14.

Step 17 (*Determination of solution*): If $z_0=\infty$, then no feasible solution exists. If $z_0<\infty$, then the solution that attains z_0 is optimal.

Acknowledgement

The authors wish to thank D. Norberto Navarrete Jr. of Kyoto University for several improvements in the use of English for this manuscript.

References

1) Aho, A.V., Hopcroft, J.E. and Ullman, J.D.: *The Design and Analysis of Computer Algorithms.* Addison-Wesley Co., 1974.
2) Belford, P.C. and Ratliff, H.D.: A Network-Flow Model for Racially Balancing Schools. *Operations Research,* Vol. 20 (1972), 619-628.
3) Clarke, S. and Surkis, J.: An Operations Research Approach to Racial Desegregation of School Systems. *Socio-Economic Planning Science,* Vol. 1 (1968), 259–272.
4) Garey, M.R. and Johnson, D.S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman & Company, San Francisco, 1979.
5) Garfinkel, R.S. and Nemhauser, G.L.: *Integer Programming.* John Wily & Sons, 1972.
6) Garfinkel, R.S. and Nemhauser, G.L.: Optimal Political Districting by Implicit Enumeration Techniques. *Management Science,* Vol. 16 (1970), B495–508.
7) Geoffrion, A.M.: An Improved Implicit Enumeration Approach for Integer Programming. *Operations Research,* Vol. 17 (1969), 437–454.
8) Heckman, L.B. and Taylor, H.M.: School Rezoning to Achieve Racial Balance: A Linear Programming Approach. *Socio-Economic Planning Science,* Vol. 3 (1969), 127–133.
9) Holloway, C.A., Wehrung, D.A. and Zeitlin, M.P.: An Interactive Procedure for the School Boundary Problem with Declining Enrollment. *Operations Research,* Vol. 23 (1975), 191–206.
10) Liggett, R.S.: The Application of an Implicit Enumeration Algorithm to the School Desegregation Problem. *Management Science,* Vol. 20 (1973), 159–168.