An Optimization Based Heuristic for Political Districting

Anuj Mehrotra • Ellis L. Johnson • George L. Nemhauser
Department of Management Science, University of Miami, Coral Gables, Florida 33124-8237
School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332
School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332

Redistricting, the redrawing of congressional district boundaries within the states, may occur every 10 years on the basis of the population census. Many redistricting plans are designed with partisan politics in mind, resulting in disputes and forcing judges to intervene. We address this problem from a nonpolitical viewpoint and present an optimization based heuristic incorporating universally agreed upon characteristics. We model the problem as a constrained graph partitioning problem and develop a specialized branch-and-price based solution methodology. We demonstrate the feasibility of our methodology by showing how to satisfy the one-person, one-vote principle with compact and contiguous districts for the state of South Carolina.
(*Graph Partitioning*; *Branch-and-price*; *Column Generation*; *Clustering*; *Districting*)

1. Introduction

Every 10 years, the results of a population census in the U.S. may require a redistribution of the House seats among the states. This process is known as *reapportionment*. Following reapportionment, each state with more than one representative may have to accommodate these shifts in population by redrawing the political boundaries between districts. This process, whether necessitated by a gain or loss of House seats, or by mere shifts of population within the state, is known as *redistricting*.

Historically, the Constitution contained provisions for the reapportionment of the U.S. House seats but did not specify how the members should be elected. While most states formed districts with the purpose of electing one representative from each district, some states allowed the candidates to run at large, with voters able to cast as many votes as there were seats to be filled. For states partitioned into districts, the practice of drawing district lines to maximize the advantage of a political party became known as "gerrymandering." The term originated in 1812 from a salamander-shaped congressional district created by the Massachusetts Legislature when Elbridge Gerry was governor. Gerrymandering can be of various types. A partisan gerrymandering is one in which a single party draws lines to its advantage. A bipartisan gerrymandering is one in which the lines are drawn to protect incumbents. A racial or ethnic gerrymandering attempts to dilute or preserve the strength of minorities. Several constitutional amendments were passed in the nineteenth century to prevent gerrymandering and ensure fairness of congressional district lines.

There are three essential characteristics of districts: the districts should have nearly equal populations to adhere to the one-person, one-vote principle; the districts should be contiguous; and the districts should be geographically compact. We explain these characteristics in the next section in detail.

Many states have constantly violated one or more of the above characteristics in their plans. The courts can intervene. In nearly half of the states that underwent redistricting based on the 1990 reapportionment, federal or state courts played a significant role in the redistricting debate and judges actually issued new lines in ten states. Population equality has been deemed by the courts to be very important and states must prove the legitimacy of deviations from precise population

0025-1909/98/4408/1100$05.00

equality. To avoid this, states have gone to extremes to ensure population equality at the expense of non-contiguous and non-compact districts with little regard to jurisdictional boundaries of counties and cities. Following some court rulings that forced states to redraw district lines because of excessive population variance, most states in the 1990s came very close to precise population equality (Preimesberger and Tarr 1993).

In many states, particularly in the South, the driving force was the Justice Department demand that majority-minority districts be drawn wherever possible. Several states (including Georgia, Florida, North Carolina, and South Carolina) went to extremes to form such districts without regard to compactness. Most plans were challenged in court and after several, sometimes conflicting court rulings, and several re-drawing of the districts, the states of Georgia and Florida are still working on their districting plans. In a recent ruling, the federal courts have rejected districting plans that were deemed to have districts that were gerrymandered to increase the chances of electing a minority candidate. In every case, the offending districts were deemed by the courts to be gerrymandered because *they were either not contiguous or not compact*. The courts never used a quantitative measure of compactness to declare the plans unsuitable. Instead, the courts have simply disallowed plans with long and thin or snakelike districts. In other words it appears that the courts have evaluated compactness only visually.

There are some mathematical and computerized approaches in the literature. Such methods can eliminate gerrymandering by providing clearly defined deterministic steps that do not permit user discretion. On the other hand such methods can be used to provide alternative plans by adjusting parameters in the algorithm. The latter alternative may be of greater interest to politicians and the courts who have to produce and evaluate plans.

Local search based methods were used in Kaiser (1966) and in Nagel (1965) to improve an existing districting plan by swapping population units that improved some measure of fitness such as population equality. An iterative facility location model that allocates population units to legislative district centers was developed in Weaver and Hess (1963). An implicit enumeration technique was developed (Garfinkel and Nemhauser 1970). The problem of determining New Zealand's electoral districts has been studied in George et al. (1993), in which a location-allocation based iterative method is used in conjunction with a geographic information system (GIS). A similar approach is used in designing sales territories in Fleischmann and Paraschis (1988). The problem of school districting has been studied in Ferland and Guénette (1990) and an interactive decision support system has been developed. The primary objectives are to assign contiguous sectors to a school and to ensure that the students attend the same school from year to year while meeting the capacity constraints of the school.

Our goal is to develop a districting method that provides population equality and contiguous and compact districts while retaining jurisdictional boundaries of counties or other political subunits insofar as possible. Contiguity is achieved by treating it as a hard constraint. Population equality is considered in the optimization phase and can be tightened, if necessary, in the post-processing phase. The retention of subunit boundaries results from the definition of population units in the optimization phase and by minimizing the number of units to be split in the postprocessing step. Since the entire procedure ignores the political data, it is free from possible accusations of gerrymandering. However, it is possible to incorporate additional constraints that can achieve, for example, minority representation. One can then evaluate the resulting degradation in other desirable characteristics. Our contributions include development of appropriate mathematical models that capture the essential features of a desirable districting plan, and implementation of a branch-and-price based method that helps determine a plan that needs little or no subjective intervention to yield the final set of compact, contiguous, and equally populated districts.

In the next section, we develop a mathematical model of the districting problem, which is similar to the model used in Garfinkel and Nemhauser (1970), except for the fact that our model is capable of considering many more potential districts. In §3, we give details of how our branch-and-price based methodology can be practically implemented. We are not aware of any other studies that have applied branch-and-price to districting problems. In §4, we present results of using our methodology to develop districting plans based on the 1990 census for the state of South Carolina. We discuss the results and present some other features that can be incorporated in our approach in §5.

2. A Mathematical Framework

DEFINITION. A districting plan is a partitioning of indivisible population units (for example, counties) into a predetermined number of districts such that the units in each district are contiguous, each district is geographically compact and the sum of the populations of the units in any district lies within a predetermined range.

The problem can be modeled as a graph partitioning problem, see Mehrotra (1992) for example, by associating a node with every population unit and connecting two nodes by an edge whenever the corresponding population units are geographical neighbors. The weight on a node is equal to the population of the corresponding unit. A plan is represented by a partitioning of the nodes such that the nodes in any set of the partition induce a connected subgraph (to ensure contiguity of the districts) and the sum of the node weights lies within the prespecified range (to satisfy the population requirements). A plan is good if each resulting district is geographically compact.

Suppose that we assign a penalty cost to every potential district that measures its ''non-compactness'' from an ideal district (we will develop such a cost in the next subsection). If all possible districts that satisfied the population and contiguity requirements were enumerated, the problem would reduce to determining a set of K districts, where K is known, that satisfies an upper bound on the total penalty cost while making sure that every population unit is included in exactly one of the selected districts. Since it is not obvious how to determine an acceptable upper bound, we choose to minimize the total penalty cost; but it should be understood that compactness is in reality a loose constraint rather than an objective.

Let δ_{ij} equal 1 if population unit i is contained in district j, and 0 otherwise and let x_j equal 1 when district j is selected in the districting plan, and 0 otherwise. The districting problem can then be stated as the following set partitioning problem with an additional cardinality constraint:

$$\begin{aligned}
\text{minimize } & \sum_{j\in J} c_j x_j \\
\text{subject to } & \sum_{j\in J} \delta_{ij} x_j = 1 \quad \text{for } i = 1, \dots, n, \\
& \sum_{j\in J} x_j = K, \\
& x_j \in \{0, 1\}, \quad \forall j \in J, \qquad (1)
\end{aligned}$$

where c_j is the cost of district j, n is the number of population units in the state, and J is the set of all possible districts that satisfy the contiguity and population requirements. We will refer to this model as PLAN(J) in the sequel.

An alternative objective function is the bottleneck measure $\min \max_{j\in J} c_j x_j$. We have chosen to evaluate the overall compactness of a plan using the contribution of each district rather than the compactness of the worst district. However, it would not be difficult to use the bottleneck objective instead of the sum in our methodology.

2.1. District Cost

Although most states impose some form of compactness standard on their districting plans, there is no uniformly acceptable definition of compactness. Eight different measures of compactness are studied in Young (1988). The study shows that each measure fails to give satisfactory results on certain geographic configurations. It suggests that any good measure of compactness must apply both to the districting plan as a whole and to each district individually, and that it should treat census tracts as indivisible building blocks whose shape is irrelevant to the measure. Additionally, such a measure should not discriminate between large rural and small urban districts and that it should be conceptually simple and should use easily collected and verifiable data.

Here, we develop a penalty cost that measures non-compactness of a district. Our goal is to get a proxy for the visual notion of non-compactness that is easy to use in our optimization model. Noting that a district would tend to be compact if the population units in the district are not far from each other, we first considered using the pairwise sum of Euclidean distances between population units in a district as the penalty cost for the district. Early experiments suggested that this measure gives a high cost to districts comprising of several rural population units as these sparsely populated areas are spread over large geographic areas. To remove this bias, we modified the distance between two units by multiplying it by the sum of the populations of the two units. This tended to bias the measure against the densely populated areas. Several experiments, including scaling of the populations, indicated that the deficiency in this measure continued because of Euclidean distances. This happens because the actual distance between units A and B might be lower than the distance between units A and C even though A and C are geographic neighbors while A and B are not. Hence, as detailed next, we finally selected a distance function that measures the proximity in terms of how many other population units one must go through to get from one population unit to another. After making this adjustment, empirical results suggested that it was unnecessary to include population weights in the measure.

Let $G(V, E)$ be a graph with V defined to be the set of population units and E the pairs of units that share a common border. A district is a node induced subgraph $G'(V', E')$ that is connected and satisfies population bounds. We will measure the non-compactness of G' by how far units in the district are from a central unit. The length of a path from i to j is defined to be the number of edges in the path. Let s_{ij} be the number of edges in a shortest path from i to j in G. We define the center of G' to be a node $u \in V'$ such that $\Sigma_{j \in V'} s_{uj}$ is minimized. Alternatively, we could use $\max_{j \in V'}\{s_{uj}\}$ to define the center. Although the max criterion is the usual definition in graph theory, the sum criterion works better for our methodology. We define the cost of the district to be $\Sigma_{j \in V'} s_{uj}$ where u is a center of the district. The smaller the cost, the more compact a district is.

In §4, we demonstrate that this cost function yields visually compact districts, that is, on the basis of existing court decisions, the plans it yields surely would be acceptable with respect to contiguity and compactness. Since we treat each population unit as a node in a graph (that ignores the actual geographic shape and size) and consider number of edges in a shortest path (rather than the actual distance), our measure also seems to be consistent with the recommended principles in the study in Young (1988).

2.2. District Generation

Since the number of possible districts, J, is exponentially large in the number of population units, model PLAN(J) has an exponential number of columns or variables. Thus it is not practical to enumerate all of the districts. Instead, we generate them on an "as needed" basis. This methodology, called *column generation* is discussed in detail in Barnhart et al. (1998). A brief outline of the methodology is presented here.

Begin with a subset $\bar{J}$ of feasible districts. Solve the linear relaxation (replace the integrality constraints on x_j with $0 \le x_j \le 1$) of PLAN(J) restricted to $j \in \bar{J}$. The optimal solution to the linear relaxation of PLAN($\bar{J}$), denoted by LP_PLAN($\bar{J}$) is a feasible solution for the unrestricted linear program LP_PLAN(J) and provides a dual value π_i for each constraint in LP_PLAN(J). Now, determine if it would be useful to expand $\bar{J}$, i.e. determine if the current solution to the LP relaxation is optimal or if there are columns in $J \backslash \bar{J}$ that price out favorably. This is done by solving the following subproblem called SP: $\min_{u \in V} \{S(u)\}$

$$\text{where } S(u) = -\pi_{n+1} - \pi_u + \min \sum_{i \in \{V \backslash u\}} (s_{ui} - \pi_i) y_i,$$

$$p_{\min} - p_u \le \sum_{i \in \{V \backslash u\}} p_i y_i \le p_{\max} - p_u,$$

$$y \text{ satisfies contiguity constraints,}$$

$$y_i \in \{0, 1\}, \quad i \in \{V \backslash u\}, \tag{2}$$

where p_i is the population of the ith unit, $p_{\min}$ and $p_{\max}$ are the lower and upper bounds on the population of a district, and $y_i = 1$ if unit i is in the district and $y_i = 0$, otherwise. Let $\bar{p} = \Sigma_{i \in V} p_i / K$ be the mean population of a district. Typically, $p_{\min} = (1 - \alpha)\bar{p}$ and $p_{\max} = (1 + \alpha)\bar{p}$ where α is the maximum deviation allowed from the mean in district population. Determining $S(u)$ requires solving a two-sided knapsack problem with additional constraints, and SP is solved by finding $S(u)$ for every $u \in V$. We will discuss the contiguity constraints in the next subsection. If the optimal objective value of SP is negative, then a district that yields the minimum value is added to the set $\bar{J}$ and LP_PLAN($\bar{J}$) is re-solved; else the current solution to LP_PLAN($\bar{J}$) is also an optimal solution of LP_PLAN(J). In the latter case, if the solution is integral, then we have solved PLAN(J). If it is not integral, we branch as explained in §3.6. More details on the column generation are given in §3.5.

2.3. Ensuring Contiguity

A district is contiguous if it is possible to travel from any point in the district to any other point in the district without having to go through any other district. In graphical terms, G' must be connected, i.e. there exists a path from each node in the district to every other node in the district. While the definition of c_k in PLAN(J) tries to ensure that the districts generated are contiguous, the contiguity of a generated district can not be guaranteed by cost alone. To ensure that a district is contiguous, we need to add more restrictions.

We could add linear inequalities to (2) to enforce contiguity precisely, i.e. exclude all subgraphs except those that are connected and contain u. However, this requires an exponential number of constraints and would be very demanding computationally. Instead, we take a simpler approach by requiring the district to be a subtree of a shortest path tree rooted at u. This guarantees contiguity but eliminates some contiguous districts. However districts that are not subtrees of a shortest path tree are unlikely to be compact. To enforce the shortest subtree requirement, we add constraints that permit j to be selected only if at least one of the nodes that is adjacent to it and closer to u is also selected. Specifically, if $S_j = \{i \in V \mid s_{ui} = s_{uj} - 1 \text{ and } (i, j) \in E\}$, then we add the contiguity constraint: $y_j \leq \Sigma_{i \in S_j} y_i$, ensuring that node j is selected only if all nodes along some shortest path from u to j are also selected.

2.4. Relationship with Other Models

We have modeled the districting problem as a constrained graph partitioning problem that is similar to constrained facility location problems (see Mirchandani and Francis 1990) in that the centers of the districts to be generated can be viewed as facilities that are located to serve the population units in the district. The population restriction is similar to the capacity of each facility. The major differences are that all nodes in the network are potential sites for the facilities, that the cost of serving a customer from a given facility is also dependent on other customers served by the facility, that there are no fixed costs of opening the facilities, the location of a facility (or a center) is not necessarily unique for a district and that the nodes served by a facility should induce a connected subgraph. There are also some similarities with other clustering and graph partitioning models described in the literature (Mehrotra 1992).

3. Implementation Issues

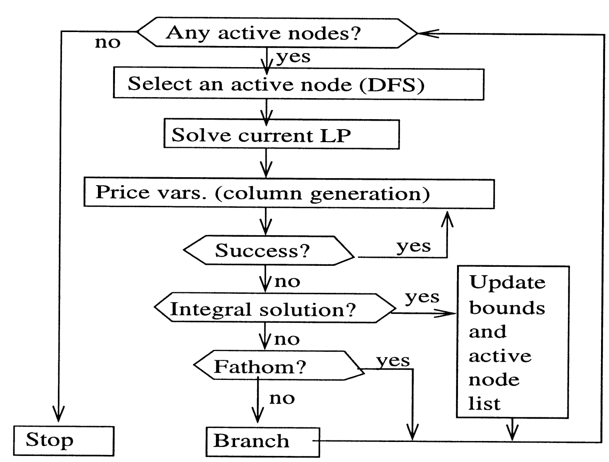
In this section, we discuss a practical implementation of the ideas presented in the previous sections. The methodology consists of four phases as discussed in this section.

The data required by our methodology consist of a state map that shows the boundaries of the counties (or other political units), the population of each unit, and the number of districts to be generated. The preprocessing phase determines the population units and their adjacencies from this map. The start phase generates a starting solution. The optimization phase (Figure 1) uses a branch-and-price approach to determine an improved, near optimal plan in which the districts formed may differ slightly in population. Finally, the postprocessing phase modifies the resulting plan to ensure population equality.

3.1. Basic Population Units

The models developed in the previous section use basic population units that we assume are indivisible.

Figure 1 **Optimization Phase**



Typically, we try to use counties as population units but this is not always possible. For example, a county may have more population than the target population of a district. If a county's population is more than $\bar{p}$, then a district (or districts) consisting of only part of the county with population equal to $\bar{p}$ is extracted and the number of districts to be formed from the remaining counties is reduced by 1. Extracted districts are assumed to comprise the central part of the county and hence do not require modifying adjacencies in the graph. This simplifying assumption is removed in the postprocessing step.

Population equality of districts may not be possible to achieve unless counties are split into two or more population units. We split counties with populations greater than $0.25\bar{p}$ and less than $\bar{p}$ so that the resulting units have population equal to or less than $0.25\bar{p}$. Splitting of a county will typically require re-establishing adjacencies between population units.

Sometimes, the geographic location of counties implies that they need to be together in a district. For example, if a county is adjacent to only one other county, it must belong in the same district as the adjacent county. In this case we coalesce them into a single population unit. This reduces the size of the graph G over which the column generation problem is solved and makes the problem easier to solve. We also combine counties with very little population. Specifically, a county with population less than $0.02\bar{p}$ is combined with the smallest county to which it is adjacent. As a result of these steps, all units will have a population between 2% and 25% of $\bar{p}$.

3.2. District Population Range

If we were to restrict the population of a district to be within extremely narrow limits, typically there would be no feasible plan in which the population units are not split and the districts are contiguous. Even if there existed such a plan, it is very unlikely that the districts would be compact. Moreover such a restrictive constraint would increase the computational burden of solving the subproblems for generating improving districts and make the method impractical.

There are two ways to overcome this difficulty: either define the population units to be much smaller subdivision of the counties, such as census block groups or census tracts, or have a postprocessing step to adjust the populations in the districting plan generated with a slightly relaxed population constraint. We choose the latter method for two reasons: the idea of keeping larger units together insofar as possible would otherwise be defeated and the difficulty of the problems to be solved would increase substantially by increasing the number of population units. Postprocessing will attempt to minimize the number of units to be split in modifying the plan to achieve population equality.

We initially set the district population to lie within 2% to 5% of $\bar{p}$. The postprocessing procedure modifies the resulting plan to ensure strict population equality if necessary.

3.3. Preprocessing

Our procedure uses the adjacency graph and the populations of the counties to coalesce two or more counties. Because of the odd shapes of population units, it is not enough for two units to share a boundary to be called adjacent units. Instead, we place an edge between two nodes only if the convex hull of the two corresponding units does not include a large portion of other population units.

The preprocessing step does the following.

1. Coalesce the nodes corresponding to counties that are fixed to be together due to geographic considerations. These units are marked not to be split into smaller units (even if the total population exceeds $0.25\bar{p}$). The coalesced unit is adjacent to nodes that any of the original nodes were adjacent to.

2. If $p_i > \bar{p}$, let $p_i = r\bar{p} + p_i^*$, where $p_i^* < \bar{p}$ and r is a positive integer. Extract r districts of population $\bar{p}$ and reduce the number of districts to be constructed by r. This step does not require modifying the adjacencies in the graph.

3. If $p_i < 0.02\bar{p}$, unit i is coalesced with the least populated neighboring unit. The coalesced unit is adjacent to any node that either of the two nodes were adjacent to.

4. If $0.25\bar{p} < p_i < \bar{p}$, unit i is split into k equally populated units such that the population of each new unit is at most $0.25\bar{p}$. The adjacencies of these split units needs to be reestablished by locating them geographically on the state map. This is done outside of our procedure.

3.4. Initial Plan: A Clustering Heuristic

Our methodology typically performs much better when it is initiated by a set of columns corresponding to a good plan. When the number of districts to be formed has not changed, past districting plans may provide a good starting point. Alternatively, we can build plans from scratch using heuristics.

A simple heuristic to generate a district starts at a node and accumulates neighboring nodes until the population restrictions are met. This greedy approach of growing a district usually generates a contiguous and compact district. Then the nodes in the district are removed from further consideration and the process is repeated until K districts are formed. The difficulty with this approach is similar to the difficulty with most greedy heuristics where the initial clusters are good, but they leave behind a set of nodes towards the end that can not form a compact and contiguous district. To overcome this difficulty, we start by choosing a reference node and then growing districts starting from nodes that are far away from the reference node. The idea is that the units left behind at the end, which are all near the reference unit, will comprise a contiguous and compact district. We describe this heuristic next.

Let $u \in V$ be the reference node. We apply the heuristic once for every node serving as a reference node and then select the best of the resulting solutions. Let L be a list of nodes that have not been assigned to any district. Initially L consists of all the nodes in V. Let $\deg_L[i]$ be the number of nodes in L that are adjacent to node i. The heuristic has two steps: the first step is used to start the generation of a new district by choosing a starting node. The second step is used to grow the district being generated. These two steps are used to generate $K - 1$ districts. The nodes not assigned to these $K - 1$ districts correspond to the units that comprise the Kth district.

1. *Choose a starting node*: Choose a node $v \in \text{argmax}_j\{p_j : j \in \text{argmax}_{i \in L}\{s_{ui}\}\}$. Remove v from L. Let M be the list of nodes in the district being generated. The first element in M is v. Let $\deg_M[i]$ be the number of nodes in M adjacent to node $i \in L$. Let $\text{pop}(M) = \Sigma_{i \in M}\, p_i$.

2. *Complete the district started*:

(a) *Termination test*: If $\text{pop}(M) \geq p_{\min}$, form a district comprising of the units that correspond to the nodes in M, calculate the cost c_k of the district and stop.

(b) *Choose the next node to add*: If possible, out of all unassigned nodes, which can still be included in the district being generated without exceeding the maximum population, choose a node that is closest to the start node. If there is a tie, choose a node that is adjacent to more nodes already included in the district (break ties arbitrarily in this case). That is, let

$$w \in \text{argmax}_j\{\deg_M(j) : j \in \text{argmin}_{i \in L} \times \{s_{iv} : \text{pop}(M) + p_i \leq p_{\max},\ \deg_M(i) > 0\}\}.$$

If there is no such node, form an (infeasible) district that corresponds to the nodes already in M and let the cost of this district be very high (to show infeasibility) and stop.

(c) *Update Lists*: Remove w from L and add w to M. Update $\text{pop}(M)$, $\deg_M$, and $\deg_L$. Repeat 2.

We have experimented with replacing the termination condition $\text{pop}(M) \geq p_{\min}$ in Step 2 with the condition $\text{pop}(M) \geq \bar{p}$, but the results are slightly inferior.

If the heuristic still fails to generate K districts that meet the contiguity and the population requirements, we assign a very high cost to the districts that do not meet the population or contiguity requirements and go on to the column generation procedure without a feasible starting solution.

3.5. Column Generation

The integer programming subproblem SP to generate a column for the master problem is both theoretically and practically hard. Here we describe some techniques for making this approach computationally efficient. We have included all of these ideas in our implementation.

A district with units that are farther than distance 3 from the center of the district rarely gets generated. In any case, such districts are usually not compact enough to be selected in the final plan. Hence, to reduce the difficulty of solving SP, while solving for $S(u)$, we fix $y_i = 0$ whenever $s_{ui} > 3$.

Typically, column generation procedures display a *tailing-off* effect: as optimality is approached, columns are generated that give very little or no improvement in the objective value. Additionally, the subproblems to generate a column get harder. Since it is not critical to optimize LP_PLAN, we terminate the column generation using a prespecified tolerance. We stop the column generation procedure if the subproblem optimization can not yield a column with reduced cost at most −0.01. This target value (−0.01) helps in pruning the branch-and-bound tree for solving (2).

Instead of solving SP to optimality, the first solution that provides an improving column is used. After obtaining an improving column from $S(u)$, the center of the generated district is determined. If the center of this district is different from u, then the real cost of this district is even smaller than the cost determined from $S(u)$.

While solving SP as (sub)subproblems $S(u)$, we solve for $S(u)$ in order of nonincreasing values of the dual variables π_u. Empirically, this helps in finding an improving column after solving fewer (sub)subproblems.

3.6. Branching

3.6.1. Branching Rule. The usual branching schemes are inappropriate for integer programs where the entire set of columns is not explicitly available. Consider, for instance, the rule of branching on a fractional variable, where the variable is set to 1 in one subproblem and set to 0 in the other. The former subproblem causes no problem: setting a district variable to 1 corresponds to forming a district from the units corresponding to that variable. Those units can therefore be removed from further consideration. The other subproblem is more difficult. Setting a variable to 0 corresponds to not permitting the use of that district. To prevent this district from being regenerated involves finding the second, third, and so on best solutions to the already difficult subproblem (2). In the case of our model, this difficulty is overcome by using the so called Ryan-Foster branching (Ryan and Foster 1981), also see Barnhart et al. (1998), Mehrotra and Trick (1996) and Vance et al. (1993). Define the following operations: SAME(S) requires that the population units in the set S all belong to the same district and DIFFER(i, j) requires that units i and j belong to different districts.

SAME(i, j) implies that $y_i = y_j$ and DIFFER(i, j) is implemented by adding the constraint $y_i + y_j \le 1$ to the subproblem.

Consider a fractional solution to LP_PLAN($\bar{J}$). It is easy to see (Barnhart et al. 1998, Mehrotra and Trick 1996, Vance et al. 1993) that there exist two districts S_1 and S_2, and units i, j, such that $i \in S_1 \cap S_2$, and $j \in S_1 \backslash S_2$, and both x_{s_1} and x_{s_2} are fractional. Create the subproblems: DIFFER(i, j) and SAME(i, j).

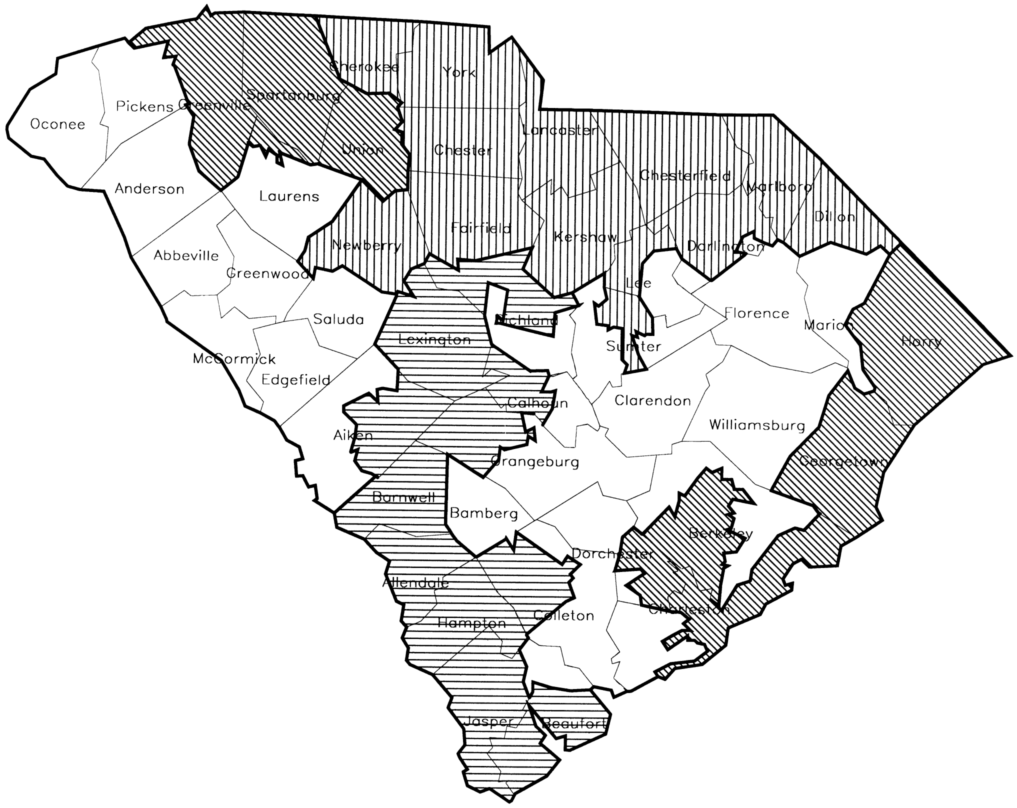
Any feasible districting plan must occur in exactly one of the two sets. Furthermore, the current fractional solution is not feasible for the two subproblems, since x_{s_1} gets fixed equal to zero in DIFFER(i, j) and x_{s_2} gets fixed equal to zero in SAME(i, j). Like traditional branching schemes, this approach creates only 2 subproblems. On the SAME branch, the subproblem is slightly easier since we have eliminated a variable. On the DIFFER branch, the subproblem is slightly harder since we have added an additional constraint to (2).

3.6.2. Implementation. We use a depth-first-search (DFS) strategy in choosing the node to evaluate. We have also experimented with choosing the node with the best bound and with switching from a DFS strategy to the best bound strategy after updating the upper bound (from the one provided by the initial solution). The DFS strategy seems to work best, both from a point of view of number of nodes explored and in CPU time overall.

For implementing our branching, we first find the most fractional column s_1, that is, s_1 is a column in the LP solution with value closest to 0.5. Then we find the row i covered by this column that corresponds to the most populated unit and choose the most fractional column (s_2) in the LP solution out of the remaining columns that cover row i. Then we find row j such that only one of the columns s_1 or s_2 cover row j. We have also experimented with choosing the first fractional column as s_1. This tends to increase the overall effort. We branch to create two sub (master) problems SAME(i, j) and DIFFER(i, j). In the depth first search, we always follow the branch SAME(i, j) before the branch DIFFER(i, j). This tends to reduce the number of nodes searched.

We do not wait for the linear program at a node to be optimized before branching. Rather, we optimize the linear program only at the root node of the branch-and-bound tree to obtain a lower bound. Then at any other node of the branch-and-bound tree, we stop generating columns as soon as the restricted linear programming relaxation objective value goes below the lower bound determined by rounding up the objective value at the root node. This tends to reduce both the number of

Figure 2 S.C. Actual Plan



columns generated and the size of the tree. The resulting decrease in overall computation time by not solving the linear programming relaxation to optimality before branching has also been experienced on other combinatorial problems (Barnhart et al. 1998, Mehrotra and Trick 1996, Vance et al. 1993).

We have implemented our methodology on a DEC ALPHA 3000 (Model 300) workstation using CPLEX version 2.1 as the linear programming solver and MINTO version 1.5 [13] as the integer programming solver for the optimization phase.

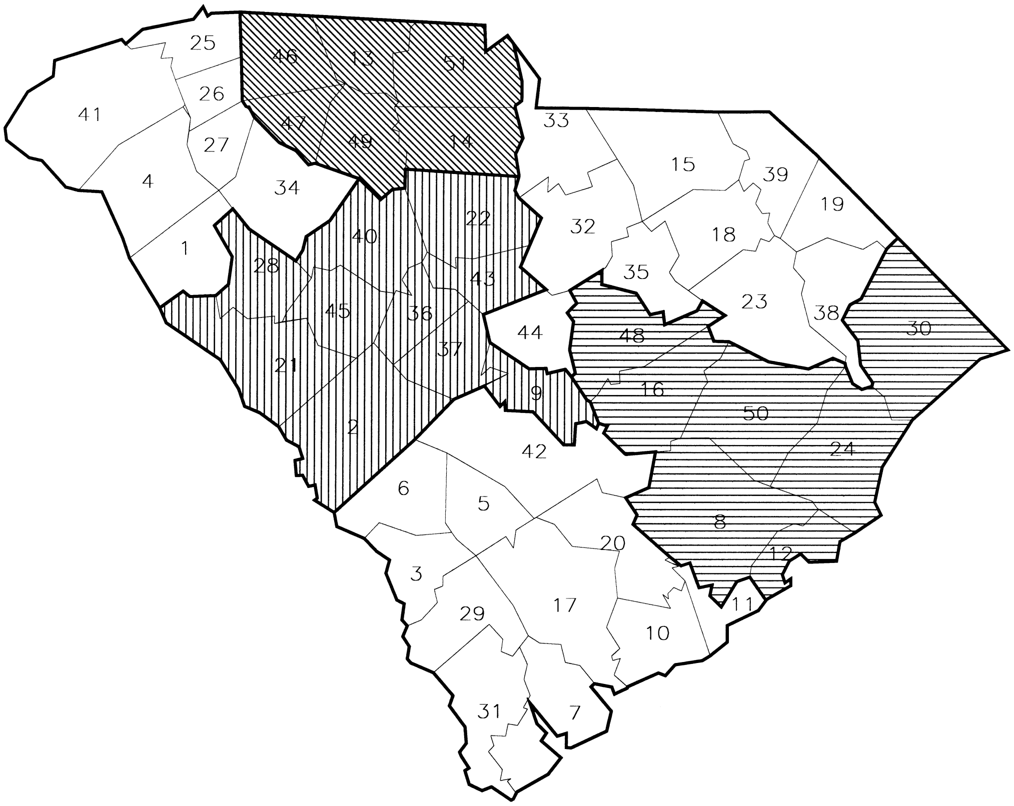
Table 1 Initial 5% South Carolina Plan

District	Population	Cost	Units
1	567952	15	3 5 6 7 10 11 17 20 29 31 42
2	591982	7	1 4 25 37 34 41
3	555670	8	8 12 16 23 24 48 50
4	596267	17	15 18 19 30 32 33 35 38 39 44
5	572032	9	13 14 26 46 47 49 51
6	602797	13	2 9 21 22 28 36 37 40 43 45

3.7. Postprocessing

The districts generated by using the previous models are not (exactly) equal in population. We suggest attaining a required balance in populations by shifting district boundaries. Let each district be represented as a node on a graph H with an edge between two nodes i and j if districts i and j are neighboring districts in the resulting plan, that is, i and j contain population units that are adjacent nodes in G. We determine the populations to be shifted between districts by solving a transshipment problem. Let P_i be the population of district i. District i is a source node if $P_i > \bar{p}$ and its supply is $P_i - \bar{p}$. District j is a sink if $P_j < \bar{p}$ and its demand is $\bar{p} - P_j$. If $P_i = \bar{p}$, then district i is neutral. Direct shipments between i and j are possible only if i and j are adjacent

Figure 3 **S.C. Five Percent Plan**



in H. Because of adjacencies, some flow may be necessary between pairs of overpopulated districts and/or between pairs of underpopulated districts. But these flows can then be minimized by using non-zero penalty costs on arcs that connect two overpopulated or two underpopulated districts. The costs on arcs that connect an overpopulated district with an underpopulated district are set to zero.

Table 2 **5% South Carolina Plan**

District	Population	Cost	Units
1	567952	15	3 5 6 7 10 11 17 20 29 31 42
2	572032	9	13 14 26 46 47 49 51
3	585379	9	8 12 16 24 30 48 50
4	591982	7	1 4 25 27 34 41
5	566558	15	15 18 19 23 32 33 35 38 39 44
6	602797	13	2 9 21 22 28 36 37 40 43 45

The actual shifts of populations are accomplished by moving part of population unit(s) between districts. This can be accomplished by considering appropriate sub-units of a unit and then shifting sub-units while maintaining compactness and contiguity of the districts. A more detailed study of this shift process can be achieved by predividing the districts into small units such as census tracts and then restricting the shifts only to whole census tracts.

4. South Carolina Case Study

In this section, we present results for the state of South Carolina which consists of 46 counties and six congressional districts. The county map and the 1990 districting plan are given in Figure 2. Note that 13 counties are divided between two districts. More details about the constitution of each district can be found in Preimesberger and Tarr (1993).

Figure 4 Postprocessing for South Carolina

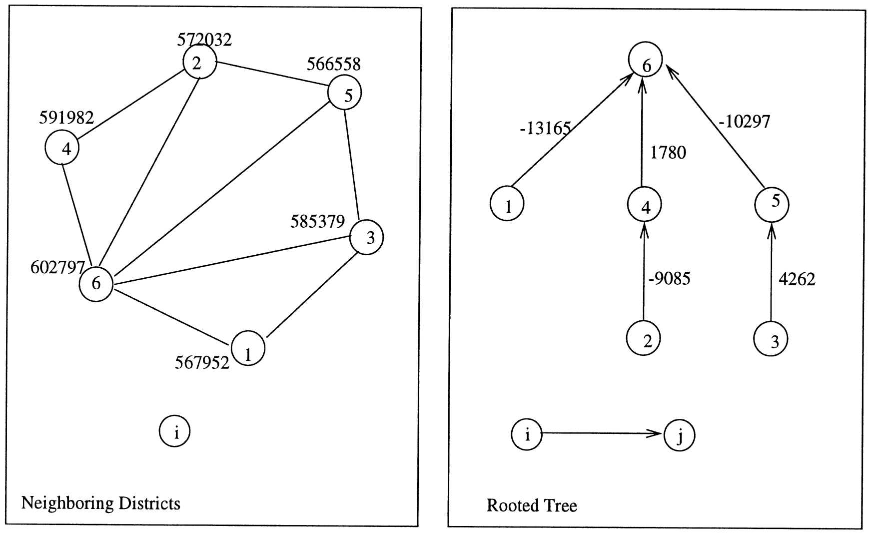
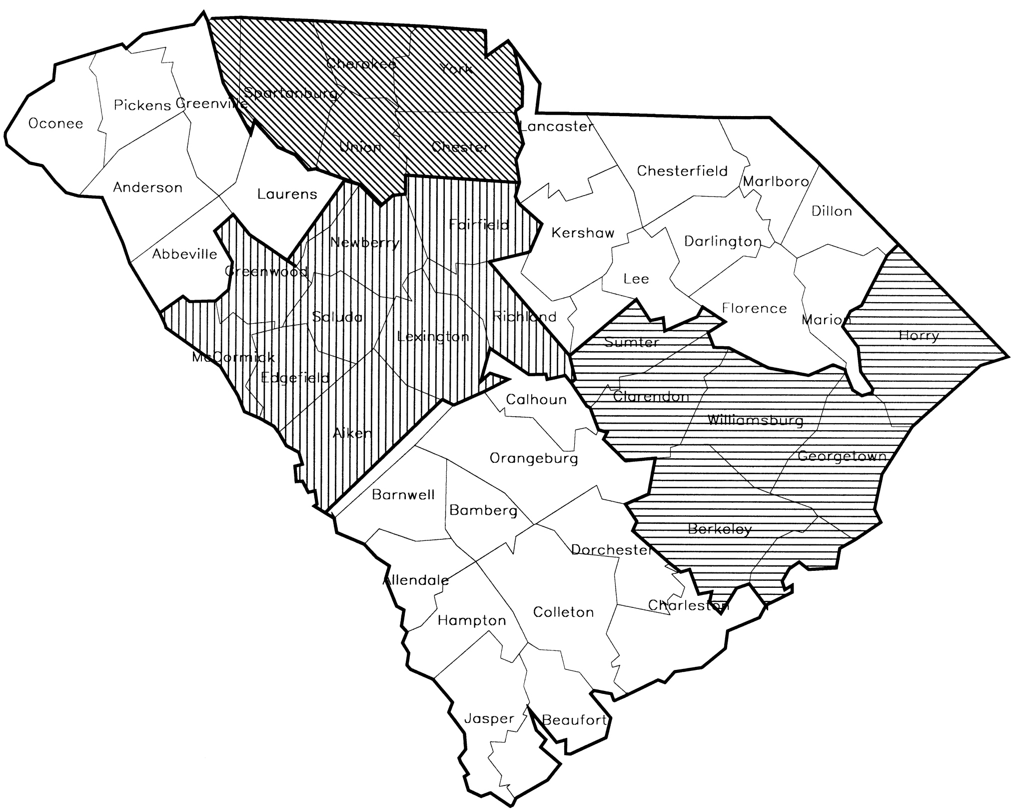


Figure 5 S.C. Plan after Postprocessing



Here we study the plans generated by using our methodology. The optimization phase in each of the experiments discussed here took less than 5 minutes of CPU time.

4.1. Preprocessing

Some counties are combined and others are split for reasons explained previously. In particular, Oconee and Pickens were combined because of geographical considerations to form unit Oconee. Edgefield and McCormick were combined because of geographical considerations to form unit Edgefield. Charleston and Greenville were split to form 3 units each. Lexington, Richland and Spartanburg were split to form 2 units each. Appendix A lists the resulting population units and their populations. The adjacency graph used is given in Appendix B.

4.2. Starting Solution and the Optimization Phase

We first generated plans where the population deviation was limited to 2%. The clustering heuristic was unable to produce a feasible solution. The optimization phase, however, generated a plan with objective value 71. The maximum deviation from $\bar{p}$ in this plan was 1.86%. A total of 1581 columns were generated and a total of 61 nodes were explored in the branch-and-bound tree. We omit other details of this plan and instead describe the 5% deviation plans that are more compact.

When the population deviation was permitted to be up to 5%, the heuristic generated a plan with objective value of 69. The maximum deviation from $\bar{p}$ in this solution is for district 3 (4.37%). This plan is summarized in Table 1.

Figure 3 and Table 2 show the plan generated in the optimization phase with objective value 68. The maximum deviation from $\bar{p}$ in this solution is for district 6 (3.73%). A total of 802 columns were generated and a total of 25 nodes were explored in the branch-and-bound tree.

The districts generated are geographically contiguous and compact. Additionally, in contrast with the existing plan, very few counties are split between two or more districts.

4.3. Postprocessing

Consider the 5% plan shown in Figure 3. The adjacency graph H for the plan is shown in Figure 4 along with population shifts between districts that comprise a solution to the transshipment problem as explained in Section 3.7.

The final plan for South Carolina which additionally satisfies the population equality constraint is shown in Figure 5.

The following adjustments are made to get the final suggested plan:

- Move Calhoun (12753) and part of Aiken (412) to district 1.
- Move an additional part of Greenville (9085) to district 2.
- Move part of Sumter (4262) to district 5.
- Move an additional part of Richland (10297) to district 5.
- Move part of Laurens (1780) to district 6.

Only 3 new counties (for a total of 6) are split in the final plan. The results indicate that it is not critical to start with very strict population restrictions. Instead, starting with slightly relaxed population restrictions, we are able to obtain a plan that satisfies compactness and contiguity requirements without splitting many counties. This plan, modified by postprocessing, yields a plan that also satisfies the population requirement strictly without splitting many additional counties.

5. Concluding Remarks

In the South Carolina case study, we have shown that our optimization based methodology provides an effective way of generating high quality districting plans. Our plans are superior to existing plans with respect to compactness and splitting of counties. Additionally, our plans are free of gerrymandering since they do not consider any political, ethnic or racial data.

5.1. Other States

We have also used our methodology to generate districting plans for the state of North Carolina (Mehrotra et al. 1995). The results are very similar, but even stronger since the North Carolina plan is extremely noncompact and not contiguous. We have discussed with officials of Georgia the development of a new plan based upon our methodology. Georgia's 1994 plan was declared unconstitutional and after the state legislature could not agree on a plan, the court imposed its own plan for the 1996 election.

5.2. Modifying Existing Plans

Although we concentrated on generating districting plans without regard to existing district boundaries, our column generation methodology can be used to produce districts similar to existing districts by modifying the subproblem that needs to be solved. For example, if the centers of the districts are fixed, we can restrict our attention to solving for improving columns for each of the centers. Preprocessing can be used to fix counties that are desired to be together. Penalty costs for moving a population unit to a new district can be incorporated in the subproblem and the distances between units that are in the same district in the existing plans can be scaled down.

5.3. Incorporating Other Criteria

Although, our basic model provides a non-political method for districting, it can be modified to incorporate some other criteria that are deemed important by the courts. For example, if we wanted to enforce at least h districts to be a minority districts, we would add the constraint $\Sigma_{j\in J}\ \omega_j x_j \geq h$ to (1) where ω_j is 1 if district j is a minority district and 0 otherwise. In the column generation procedure, we would add a constraint that ensures the generation of some minority districts. For example, if $p_i = p_i^1 + p_i^2$, where p_i^1 is the minority population in population unit i, we would add the constraint $\Sigma_{i\in V}\ p_i^1 y_i \geq \Sigma_{i\in V}\ p_i^2 y_i$ to the subproblem (2) for column generation.

Other features also are easy to model. Consider, for example, the requirement that two incumbents can not belong to the same district in the new plan. For every pair of counties i and j such that i and j are counties of incumbent candidates, the constraint DIFFER(i, j) can be forced in the column generation procedure.[1]

[1] The first author was partially supported by the James W. McLamore Research Award in Business and Social Sciences. The second and third authors were partially supported by NSF grant DDM-9115768 to Georgia Institute of Technology. We are grateful to three anonymous referees and an associate editor for providing many suggestions for improving the earlier drafts of this paper.

Appendix A. South Carolina data after preprocessing

1	Abbeville	23862	2	Aiken	120940	3	Allendale	11722
4	Anderson	145196	5	Bamberg	16902	6	Barnwell	20293
7	Beaufort	86425	8	Berkeley	128776	9	Calhoun	12753
10	Charleston	98346	11	Charleston	98346	12	Charleston	98346
13	Cherokee	44506	14	Chester	32170	15	Chesterfield	38577
16	Clarendon	28450	17	Colleton	34377	18	Darlington	61851
19	Dillon	29114	20	Dorchester	83060	21	Edgefield	27243
22	Fairfield	22295	23	Florence	114344	24	Georgetown	46302
25	Greenville	106722	26	Greenville	106722	27	Greenville	106722
28	Greenwood	59567	29	Hampton	18191	30	Horry	144053
31	Jasper	15487	32	Kershaw	43599	33	Lancaster	54516
34	Laurens	58092	35	Lee	18437	36	Lexington	83805
37	Lexington	83805	38	Marion	33899	39	Marlboro	29361
40	Newberry	33172	41	Oconee	151388	42	Orangeburg	84803
43	Richland	142860	44	Richland	142860	45	Saluda	16357
46	Spartanburg	113400	47	Spartanburg	113400	48	Sumter	102637
49	Union	30337	50	Williamsburg	36815	51	York	131497

Appendix B. South Carolina adjacency graph after preprocessing

The first entry is the node number, the second entry is the number of adjacencies to nodes with a higher node number followed by the adjacent nodes.

Node	No.	Adjacencies	Node	No.	Adjacencies
1	3	4 21 28	2	5	6 21 36 37 42
3	3	5 6 29	4	3	26 27 41
5	3	6 17 42	6	1	42
7	2	17 31	8	6	11 12 16 20 42 50
9	5	16 37 42 44 48	10	3	11 17 20
11	2	12 20	12	1	24
13	4	46 47 49 51	14	4	22 33 49 51
15	4	18 32 33 39	16	3	42 48 50
17	2	20 29	18	3	23 35 39
19	2	38 39	20	1	42
21	2	28 45	22	3	32 40 43
23	2	38 50	24	2	30 50
25	3	26 41 46	26	3	27 41 46
27	2	34 47	28	2	34 45
29	1	31	30	1	38
32	5	33 35 43 44 48	33	1	51
34	3	40 47 49	35	1	48
36	3	37 43 45	37	1	44
40	2	45 49	43	1	44
44	1	48	46	1	47
47	1	49			

References

Barnhart, C., E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, P. H. Vance 1998. Branch-and-price: column generation for huge integer programs. *Operations Research*. to appear.

Ferland, J. A., G. Guénette 1990. Decision support system for a school districting problem. *Operations Research*. **38** 15–21.

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

Garey, M. R., D. S. Johnson 1979. *Computers and intractability: A guide to the theory of NP-Completeness*. W. H. Freeman and Company, New York.

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

George, J. A., B. W. Lamar, C. A. Wallace 1993. Determining New Zealand electoral districts using a network-based model. *Operational Research Society of New Zealand*, *Proceedings*. **29** 276–283.

Kaiser, H. 1966. An objective method for establishing legislative districts,'' *Midwest Journal of Political Science*. X.

Mehrotra, A. 1992. Constrained graph partitioning: decomposition, polyhedral structure and algorithms. Ph.D. thesis, Georgia Institute of Technology, Atlanta, GA.

——, E. Johnson, G. L. Nemhauser 1996. Taking the politics out of districting (earlier version of the current paper). Department of Management Science, School of Business Administration, University of Miami, Coral Gables, FL.

——, M. A. Trick 1996. A column generation approach for graph coloring. *INFORMS Journal on Computing*. **8** 344–354.

Mirchandani, P. B., R. L. Francis (Eds.) 1990. *Discrete Location Theory*. John Wiley & Sons, Inc., New York.

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

Nemhauser, G. L., M. W. P. Savelsbergh, G. C. Sigismondi 1994. MINTO, a mixed integer optimizer. *Operations Research Letters*. **15** 47–58.

——, L. A. Wolsey 1988. *Integer and Combinatorial Optimization.* John Wiley & Sons, Inc., New York.

Preimesberger, J., D. Tarr, (Eds.) 1993. *Congressional Districts in the 1990s*. Congressional Quarterly Inc., Library of Congress Cataloging-in-Publication Data, Washington, D.C.

Ryan, D. M., B. A. Foster 1981. *An integer programming approach to scheduling*. A. Wren, (Ed.). *Computer Scheduling of Public Transport Urban Passenger Vehicle and Crew Scheduling*. North Holland, Amsterdam. 269–280.

Slater, C. M., G. E. Hall, (Eds.) 1992. 1993 *County and city extra*. Bernan Press, Lanham, MD.

Vance, P. H., C. Barnhart, E. L. Johnson, G. L. Nemhauser 1993. Solving binary cutting stock problems by column generation and branch-and-Bound. *Computational Optimization and Applications*, **III** 111–130.

Weaver, J. B., S. W. Hess 1963. A procedure for nonpartisan districting: development of computer techniques. *Yale Law Journal*. **73** 288–308.

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

Accepted by Bruce Golden; *received August* 21, 1995. *This paper has been with the authors* 6 *months for* 3 *revisions*.