A Model of Contiguity for Spatial Unit Allocation

Takeshi Shirabe

Institute for Geoinformation, Technical University of Vienna, Vienna, Austria

We consider a problem of allocating spatial units (SUs) to particular uses to form "regions" according to specified criteria, which is here called "spatial unit allocation." Contiguity—the quality of a single region being connected—is one of the most frequently required criteria for this problem. This is also one that is difficult to model in algebraic terms for algorithmic solution. The purpose of this article is to propose a new exact formulation of contiguity that can be incorporated into any mixed integer programming model for SU allocation. The resulting model guarantees to enforce contiguity regardless of other included criteria such as compactness. Computational results suggest that problems involving a single region and fewer than about 200 SUs are optimally solved in fairly reasonable time, but that larger problems must rely on heuristics for approximate solutions. It is also found that a problem of any size can be formulated in a more tractable form when a fixed number of SUs are to be selected or when a certain SU is selected in advance.

Introduction

A spatial unit (SU) allocation problem can be cast as one of selecting subsets—here referred to as regions—of SUs such as census tracts, land parcels, and grid cells from a given set of SUs according to specified criteria. The problems encompass various applications ranging from political districting (Hess et al. 1965; Garfinkel and Nemhauser 1970; Hojati 1996; Mehrotra, Johnson, and Nemhauser 1998) and school districting (Yeates 1963; Belford and Ratliff 1972; Franklin and Koenigsberg 1973) to sales territory alignment (Hess and Samuels 1971; Shanker, Turner, and Zoltners 1975; Segal and Weinberger 1977; Marlin 1981; Zoltners and Sinha 1983; Fleischmann and Paraschis 1988) and timber harvest scheduling (Barahona, Weintraub, and Epstein 1992; Snyder and ReVelle 1996; Murray 1999; McDill, Rebain, and Braze 2002) to land allocation (Wright, ReVelle, and Cohon 1983; Gilbert, Holmes, and Rosenthal 1985; Diamond and Wright 1988; Tomlin and Johnston 1990; Benabdallah and Wright 1991, 1992; Crema 1996; Eastman, Jiang, and Toledano 1998;

Correspondence: Takeshi Shirabe, Institute for Geoinformation, Technical University of Vienna, Gusshausstr 27–29, 1040 Vienna, Austria
e-mail: shirabe@geoinfo.tuwien.ac.at

Submitted: December 4, 2003. Revised version accepted: May 18, 2004.

Cova and Church 2000; Aerts and Heuvelink 2002; Williams 2002, 2003; Aerts et al. 2003) and habitat reserve site selection (McDonnell et al. 2002; Önal and Briers 2002; Fischer and Church 2003; Nalle, Arthur, and Sessions 2003). Although required criteria highly vary from one application to another, they tend to relate to size, shape, or spatial relation (Shirabe and Tomlin 2002). For example, political districting has four essential criteria: equal population size, compact and contiguous shape, and mutually exclusive districts. While some regions, such as voting districts, are artificial and do not have any conspicuous physical existence, others may come to exist in the form of housing developments, airports, landfills, and so on. The latter type of region tends to have stricter shape requirements such as non-perforation, convexity, rectangularity, and similarity to a specific letter like "L" or "O."

Contiguity—the quality of a single region being connected—is one of the most frequently required SU allocation criteria, as fragmentation often affects the viability or influence of a region. It is also such a fundamental quality that many shapes can be realized only when a region is contiguous. Although an exact formulation of contiguity has repeatedly been called for in the literature (e.g., Wright, ReVelle, and Cohon 1983; Eastman, Jiang, and Toledano 1998; Cova and Church 2000), it has not been carried out until recently (Williams 2002) because of the complexity of articulating and operationalizing a statement of contiguity. Instead, contiguity has often been regarded as a property incidental to compactness—the quality of being circle-/square-like or consolidated rather than spread, because "the compactness driving force generally prevents noncontiguity" (Hess and Samuels 1971). Thus, many efforts have focused on generating a compact region, for example, by minimizing the total distance between each SU and the center of the region (e.g., Hess and Samuels 1971), by minimizing the boundary length of the region (Wright, ReVelle, and Cohon 1983), or by maximizing the number of neighboring SUs of the region (Aerts and Heuvelink 2002). A compact region, however, need not be contiguous (Cova and Church 2000; Aerts et al. 2003; Williams 2003).

More recently, sophisticated heuristics, such as "region growing" (Brookes 1997), have been developed for the task of forming a strictly contiguous region with other spatial criteria. One of the strongest advantages of such heuristics is their ability to handle a large number of SUs. It is not rare for practical problems to select a region from tens of thousands of SUs. Also, these problems tend to be ill-defined, that is, not all selection criteria can be enumerated or articulated in advance. Thus, finding many good solutions quickly is often of more practical value than finding one theoretical optimum with a considerable amount of computing time and cost (Crema 1996; Brookes 1997; Eastman, Jiang, and Toledano 1998; Aerts and Heuvelink 2002). One shortcoming, however, is that it is usually not known how good found solutions are. One may intuitively be able to tell if they are good, but cannot defend it. In general, for heuristic solutions to be properly evaluated, problems need to be formulated and solved exactly.

This article focuses on addressing the latter accuracy issue but without jeopardizing tractability. More specifically, it proposes a new exact formulation of contiguity constraints in terms that can handle small problems (in a practical sense) using current optimization algorithms. These constraints are not designed to address any specific problem whose criteria vary depending on context, but to be applied to a wide range of problems subject to contiguity requirements. The rest of this article is organized as follows. First, existing contiguity constraints are reviewed. Next, a new exact contiguity constraint set and its reduced versions for special cases are presented. Then the utility of the proposed constraints by addressing two sample problems is illustrated: one involving 179 irregular SUs and the other involving 100 regular SUs. The final section concludes the article.

Existing contiguity constraints

There have been attempts to model contiguity explicitly (if not exactly), particularly in the form of a mixed integer programming (MIP) model. Zoltners and Sinha (1983) formulated a sufficient contiguity condition for a sales territory alignment problem by utilizing what they called "hierarchical adjacency trees." A hierarchical adjacency tree is associated with an SU that is chosen as the center of a sales territory. It consists of a set of shortest paths (including the second shortest, third shortest, etc., if necessary) from the central SU to all other SUs, and defines predecessor–successor relationships along each shortest path. Contiguity is ensured if no SU is allocated to a territory unless at least one of its immediate predecessors along a shortest path is allocated to the same territory. This contiguity model is indeed innovative and useful, but is not without shortcomings. It is dependent on compactness and a predetermined center. In other words, it guarantees each territory to be contiguous, but at the same time it drives the territory to be compact around a selected SU. This may not be problematic as both compactness and contiguity are often desired properties for sales territories. However, this is not the case when one is relatively tolerant of less compact territories or simply wants to create contiguous territories regardless of the degree of compactness. More importantly, a central SU may not be so apparent in other contexts.

Cova and Church (2000) have made another significant step toward exact contiguity modeling by proposing a set of MIP constraints that guarantee to aggregate SUs to a contiguous region when a specially designated SU, called a root, is pre-assigned to that region. They found that every SU in a contiguous region has at least one predecessor on a (not necessarily shortest) path that is one unit closer to the root. They formulated this property in a way that associates each SU in a study area with a number of variables, each of which decides whether the distance from the root to that SU is a certain integer. Their contiguity model has enormous theoretical value as it proves that MIP formulation of contiguity is possible if a root is given. As they pointed out, however, the model is not efficient, because to cope with a fully general situation it might require as many binary (0–1) variables—a factor in determining the tractability of an MIP problem—as the product of the number of given SUs and the number of SUs to be selected. To remedy this drawback, they reduced their original contiguity constraints to what they called "shortest path contiguity-*n* (SPC-*n*) constraints," which allow each SU to be reached from the root only along one of its shortest *n* paths (*n* should be reasonably small). This heuristic is similar to Zoltners and Sinha (1983) in that a region grows along permissible shortest paths from a root, but is more independent of compactness because distance is measured in terms of the number of SUs passed rather than in travel time. SPC-*n* constraints are most tractable when they involve only first shortest paths—in fact, Mehrotra, Johnson, and Nemhauser (1998) applied this special version to a political districting problem. Unfortunately, tractability diminishes quickly as more higher-order shortest paths are taken into account, and the dependence on a fixed root and the bias toward a compact shape remain as long as shortest paths are used.

Williams (2002) has recently succeeded in modeling an exact contiguity condition in MIP format. His novelty is to consider in his contiguity model a (primal) graph that corresponds to a given set of SUs, as well as its dual graph. The model is formulated so as to search simultaneously for spanning trees of the two graphs (Williams 2001) and trim the primal spanning tree to one of a desired size. The resulting sub-tree represents a contiguous set of SUs. The model seems to have an efficient structure as its binary variables are only twice as many as the number of given SUs. His experiments imply, however, that the model may suffer from computational difficulties in dealing with a problem involving more than about 100 (particularly, unequally sized) SUs. Nevertheless, it is important to note that Williams' contiguity model has set a standard for evaluating other exact contiguity models that may follow.

In the next section, we present an alternative model of contiguity constraint. To do so, we too view a set of SUs as a graph, but formulate contiguity based on a theory of network flows rather than shortest paths or dual graphs.

Contiguity condition and its formulation

Contiguity can be defined in terms of a graph by equating each SU with a vertex and representing adjacency with an edge connecting a pair of SUs. In this setting, a set of SUs is said to be contiguous if there is at least one "path" (Ahuja, Magnanti, and Orlin 1993) between every pair of SUs in the set. Thus, contiguity is equivalent to the graph-theoretic notion of "connectedness" (Ahuja, Magnanti, and Orlin 1993). It follows that to check the contiguity (connectedness) of a set of SUs (a graph), *S*, one only needs to verify the following condition.

> Starting from an arbitrarily chosen SU (vertex) in *S*, one can reach every other SU (vertex) in *S* by following a sequence of adjacency edges.

Path finding between every SU and one specific SU in a contiguous set is analogous to fluid movement from multiple sources to a single sink in a connected network. In such a network, if one pours fluid into every source, all of it will reach the sink. To formulate this analogy in the context of SU allocation, we regard a given set of SUs as a network, in which each SU is represented by a node and each adjacency relationship between a pair of SUs is represented by two opposite directed arcs connecting that pair. A region is then defined as a sub-network (i.e., a portion of the entire network), in which only one (any) node serves as a sink and every other node provides one unit of supply. For a region to be contiguous, the supply sent from every source must ultimately arrive at the sink, without passing through the outside of the sub-network. Here, we are not concerned with how each unit of supply travels in the network but with whether it can reach the sink at least in one way (see Fig. 1). It is easy to see that a disconnected region violates the contiguity condition as more than one sink—one for each connected component of the disconnected region—is needed in order for all units of supply to be consumed.

Three sets of contiguity constraints are formulated below based on the contiguity condition described above.

General contiguity constraints

The first constraint set is general in that it guarantees to select a contiguous region from a given set of SUs regardless of any other criteria that may be additionally required. It is expressed as a set of linear equations as follows:

$$\sum_{\{j|(i,j)\in A\}} y_{ij} - \sum_{\{j|(j,i)\in A\}} y_{ji} \geq x_i - Mw_i \quad \forall i \in I \tag{1}$$

$$\sum_{i \in I} w_i = 1 \tag{2}$$

$$\sum_{\{j|(i,j)\in A\}} y_{ij} \leq (M-1)x_i \quad \forall i \in I \tag{3}$$

$$x_i \in \{0,1\} \quad \forall i \in I \tag{4}$$

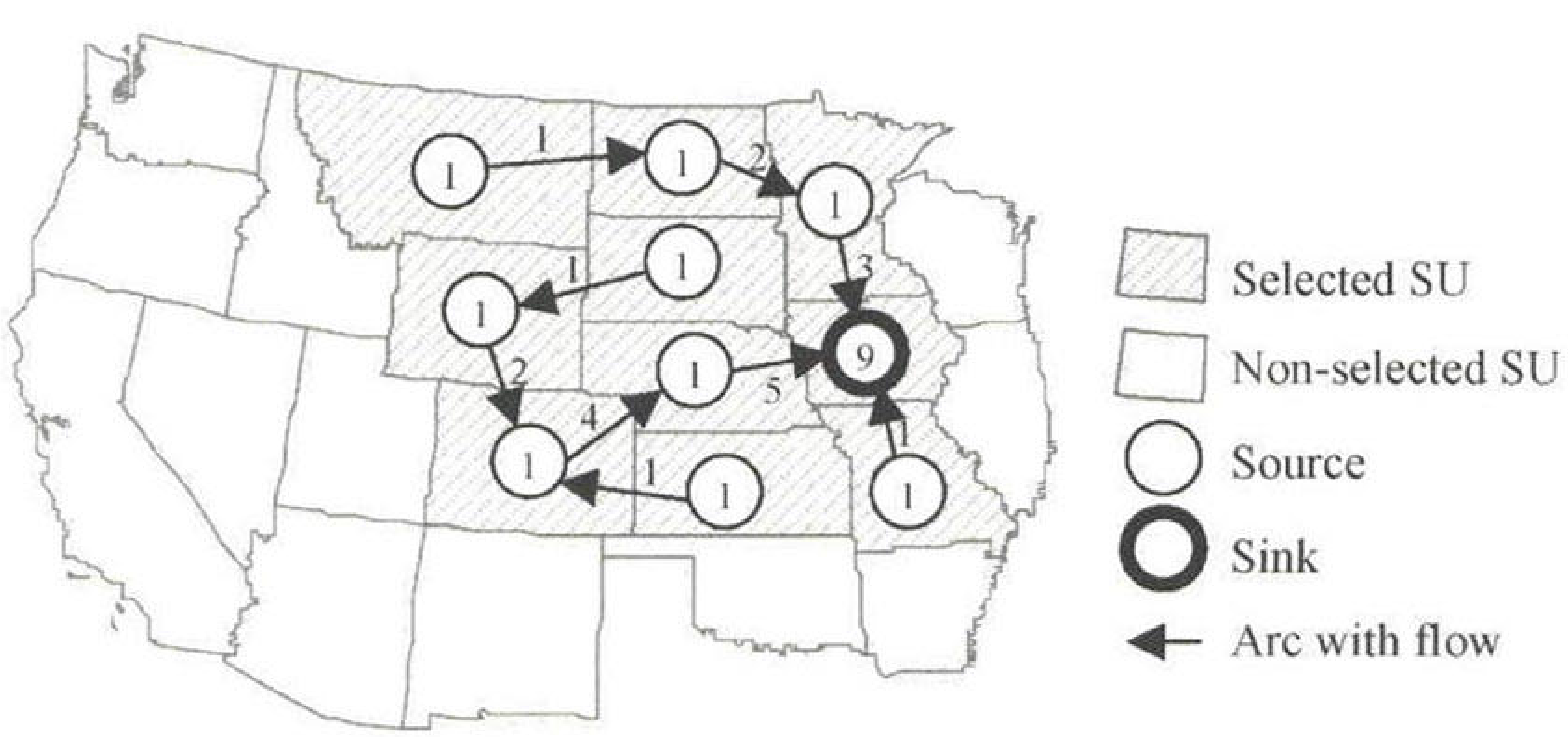

Figure 1. A network representation of a contiguous set of spatial units (SUs) (arcs without flow are suppressed) and a possible flow pattern. The numbers associated with a source, a sink, and an arc indicate the amounts of supply, demand, and flow, respectively.

$$w_i \in \{0, 1\} \quad \forall i \in I \tag{5}$$

$$y_{ij} \geq 0 \quad \forall (i, j) \in A \tag{6}$$

where I is the set of SUs, A the set of adjacent pairs of SUs, M a non-negative integer indicating the maximum allowable number of SUs to be chosen for inclusion in a region (set M to $|I|$ (or larger) in case there is no limit on the number of SUs to be chosen), x_i a binary decision variable indicating whether SU i is chosen for inclusion in a region ($x_i = 1$ if chosen, 0 otherwise), and w_i a binary decision variable indicating whether SU i is chosen as a sink ($w_i = 1$ if a sink, 0 otherwise), and y_{ij} a non-negative continuous decision variable indicating the amount of flow from SU i to SU j.

Constraints (1) represent the net outflow from each SU. The two terms on the left represent, respectively, the total outflow and total inflow of SU i. If SU i is included in a region but is not a sink, then we have $x_i = 1$, $w_i = 0$, and thus SU i must have supply ≥ 1. If SU i is included in a region and is a sink, then we have $x_i = 1$, $w_i = 1$, and thus SU i can have demand (negative net outflow) $\leq M - 1$. If SU i is not included in a region and is not a sink, then we have $x_i = 0$, $w_i = 0$, and thus SU i must have supply ≥ 0. If SU i is not included in a region but a sink, then we have $x_i = 0$, $w_i = 1$, and the rest of x_i's are forced to be 0, that is, no SUs are selected. Constraint (2) requires that one and only one SU be a sink. Constraints (3) ensure that there is no flow into any SU outside the region (where $x_i = 0$), and that the total inflow of any SU in the region (where $x_i = 1$) does not exceed $M - 1$. This implies that there may be flow from an SU outside the region to an SU in the region. Even in such a case, although a sink may have to receive an extra amount of flow, the supply from each SU in the region still must reach the sink and the contiguity condition holds. Note that it is assumed that a solution such that $x_i = 0$ for all i is feasible, that is, the empty set of SUs is a contiguous region. If a "region" is presupposed to include at least one SU, the following minimum-size constraint may be added:

$$\sum_{i \in I} x_i \geq 1 \tag{7}$$

The present set of contiguity constraints has a fairly efficient structure in that its size grows proportionally to $|I|$ and $|A|$. More precisely, the numbers of binary variables, continuous variables, and main constraints are, respectively, $2|I|$, $|A|$, and $2|I| + 1$.

Reduced contiguity constraints with fixed number of SUs to be selected

In the general contiguity constraints, it is unknown how many SUs are to be selected for a region until a solution is obtained. If it is known, however, they can be made more efficient by replacing constraints (1) with:

$$\sum_{\{j|(i,j) \in A\}} y_{ij} - \sum_{\{j|(j,i) \in A\}} y_{ji} = x_i - Mw_i \quad \forall i \in I \tag{8}$$

where M is the *exact* number of SUs to be selected.

These new constraints are interpreted in the same manner as constraints (1), except that the net outflow of each source SU and the net inflow of a sink SU are here fixed to one and $M-1$, respectively. Note that equalities are always binding constraints; thus, the actual number of variables, which includes slack variables for inequalities, has been reduced by $|I|$.

Reduced contiguity constraints with fixed sink

The general contiguity constraints involve two types of binary variables: one for the decision of whether to select a certain SU, and the other to designate a certain SU as a sink. A sink in these constraints corresponds to a root or a center in traditional shortest-path-based constraints (e.g., Zoltners and Sinha 1983; Cova and Church 2000). It is then obvious that, if a sink is fixed, the corresponding decision variables (i.e., all w_i's) will be unnecessary. Thus, given r as a sink, that is, $x_r = 1$ and $y_{rj} = 0$ for all j such that $(r, j) \in A$, the contiguity constraints can be reduced to the following form:

$$\sum_{\{j|(i,j)\in A\}} y_{ij} - \sum_{\{j|(j,i)\in A\}} y_{ji} = x_i \quad \forall i \neq r \tag{9}$$

$$\sum_{\{j|(j,i)\in A\}} y_{ji} \leq (M-2)x_i \quad \forall i \neq r \tag{10}$$

$$\sum_{\{j|(j,r)\in A\}} y_{jr} \leq (M-1) \tag{11}$$

$$x_i \in \{0,1\} \quad \forall i \neq r \tag{12}$$

$$y_{ij} \geq 0 \quad \forall (i,j) \in A \tag{13}$$

where M is once again the maximum allowable number of SUs to be selected.

Constraints (9) ensure that all selected SUs excluding a sink SU have exactly one unit of supply. Constraints (10) and (11) are interpreted in the same way as constraints (6): non-selected SUs have no inflow, and selected SUs may have some inflow. Note that if there is no upper limit on the number of SUs to be selected (i.e., M is greater than or equal to $|I|$), constraints (11) may be dropped.

Application

To illustrate how the contiguity constraints presented above are incorporated into optimization models for particular applications, this section addresses two simple SU allocation problems: one with an irregular set of SUs and the other with a regular set of SUs.

Irregular SUs

The first problem involves an irregular set of SUs, which is hypothetical but uses actual data. The study area is a small neighborhood (approximately $1{,}200 \times 1{,}800\ m^2$) called "Griffith" in Montgomery County, Maryland, and encompasses

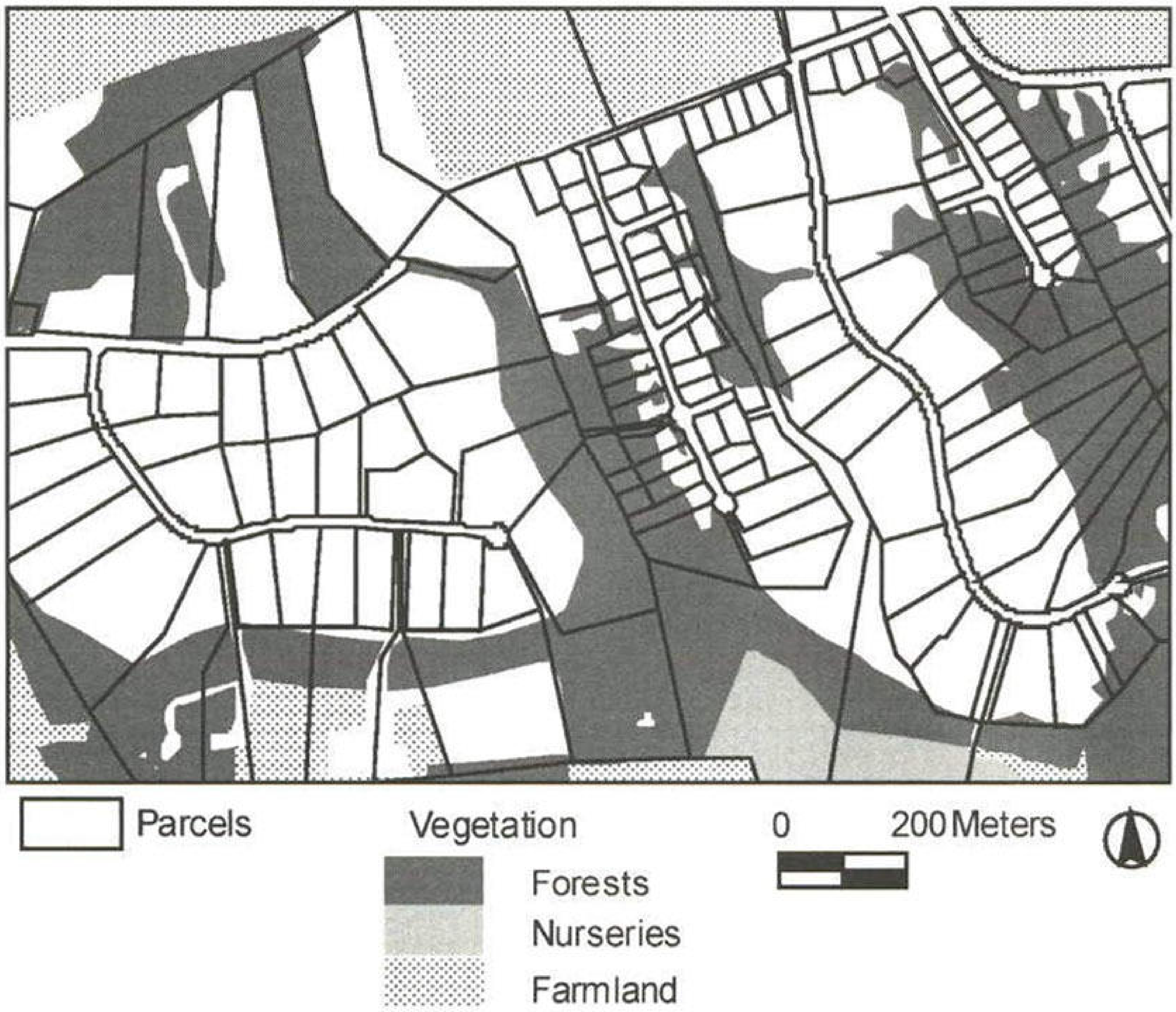

Figure 2. Study area.

179 land parcels (ranging in size from 392 to 116,226 m^2) characterized by vegetation type (Fig. 2). Each parcel has been assigned a value of expected forest benefit (ranging from 0 to 69,808) based on the amount of forest cover. The problem is rather contrived so that one can analyze exclusively the model's capacity to address contiguity. For more realistic land-allocation problems, see, for example, Wright, ReVelle, and Cohon (1983); Gilbert, Holmes, and Rosenthal (1985); Eastman, Jiang, and Toledano (1998); and Aerts and Heuvelink (2002).

Problem 1. A planning commission plans to purchase land for a forest conservation park. Therefore, select a contiguous region of land parcels in order to maximize derived benefit of specified total area to be acquired.

The problem is formulated as follows:

$$\max \sum_{i \in I} f_i x_i \tag{14}$$

subject to

$$\sum_{i \in I} a_i x_i \leq b \tag{15}$$

and (1)–(6) where a_i represents the area of parcel i, f_i represents the forest benefit value of parcel i, and b indicates the upper bound on the total acquired area specified by the commission.

Model (1)–(6), (14), (15), which contains 358 binary variables, 654 continuous variables, and 360 main constraints, was coded in GAMS and solved by an MIP solver, CPLEX 7.5, on an HPJ5000 workstation. We tested it with 20 different values of b ranging from 100,000 to 2,000,000 m^2. M is set to $|I|$ (= 179) in order not to rule out any possible selection. Note that the largest possible contiguous region encompasses 2,086,095 m^2 of 175 parcels (thus depending on the value of b, some parcels could be excluded in advance for reducing the problem size). Table 1 reports the optimal objective value, the number of branch-and-bound (BB) nodes, the central processing unit (CPU) time, and the number of iterations, for each value of b. All solutions were contiguous regions. As an example, Fig. 3 shows an optimal solution when $b = 100{,}000$.

The results indicate that all instances of Problem 1 are fairly tractable. Solution times of a few hundred seconds on a typical workstation should be acceptable for this scale (179 SUs) of SU allocation problem. It was also found that the model tends to be harder to solve as b (limit on the region's total area) increases until a certain level is reached, and then this tendency reverses. This indicates that the model is most computationally complex when a region is of a medium size—slightly smaller than half of the study area—relative to the study area. If the region's size is not strictly predetermined, one might want to drop the constraint and add it to the objective function with a certain weight to make the model more tractable.

Table 1 Computational Results for Problem 1

Value of b (m^2)	Objective value (m^2)	Number of BB nodes	CPU time (s)	Number of iterations
100,000	81,866	2,629	19.20	44,132
200,000	149,600	4,216	35.38	83,576
300,000	198,135	11,418	185.94	379,518
400,000	277,614	7,459	54.00	149,734
500,000	339,560	24,168	161.55	420,924
600,000	384,689	17,242	129.74	397,366
700,000	438,576	20,794	117.80	421,428
800,000	477,302	37,206	536.33	833,651
900,000	525,415	16,613	232.32	480,342
1,000,000	574,440	7,688	35.08	137,352
1,100,000	610,944	6,740	34.59	115,873
1,200,000	641,343	3,940	15.48	48,258
1,300,000	663,900	2,404	11.33	38,968
1,400,000	678,908	2,635	12.55	42,706
1,500,000	683,029	3,440	9.31	30,620
1,600,000	684,059	0	0.34	637

Note: Any b larger than 1,600,000 leads to the same objective value as for $b = 1{,}600{,}000$. BB, branch-and-bound; CPU, central processing unit.

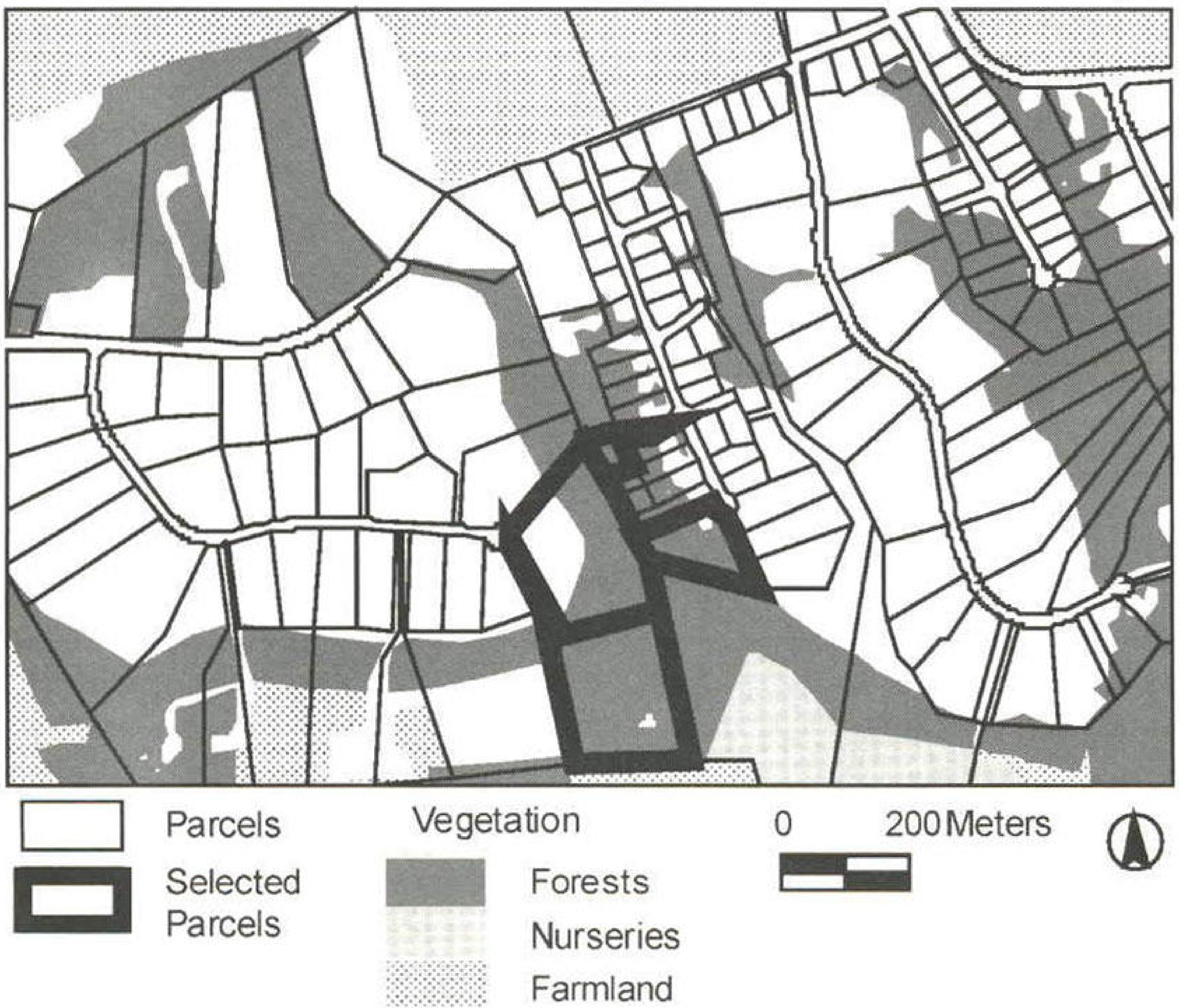

Figure 3. An optimal solution to Problem 1 with the upper bound on the total acquired area, *b*, equal to 100,000.

Now let us consider a special case where a fixed number of SUs are to be selected from the study area, to assess the reduced contiguity constraints described above. To do so, Problem 1 is modified such that the proposed forest conservation park must encompass exactly M land parcels instead of b m^2 or less. The revised problem is formulated as model (2)–(6), (8), (14), which appears similar in size to the previous one as it contains 358 binary variables, 654 continuous variables (no slack variables counted), and 359 main constraints. After testing it with 17 different values of M ranging from 10 to 160, however, the model turned out to be more tractable thanks partly to the binding nature of constraints (8). The improved tractability is also explained by the fact that constraints (15), which are "integer-unfriendly" in the sense of ReVelle (1993), were dropped. As seen in Table 2, the BB process was much less extensive than in the previous example and all except one solution were obtained within several seconds. Nevertheless, there is no guarantee that the selected M irregular parcels are of a desired size (if any). Thus, it would not normally be reasonable to use the present reduced constraints unless all SUs are of the same size (see the next section for one such example).

Finally, we reconsider Problem 1 with the assumption that one parcel has been pre-selected. This assumption makes useful the reduced contiguity constraints presented above. The resulting model (9), (10), (12–15) involves 178 binary variables,

Table 2 Computational Results for Problem 1 with a Fixed Number, *M*, of SUs to be Selected

Value of *M*	Objective value (m^2)	Number of BB nodes	CPU time (s)	Number of iterations
10	233,693	836	4.25	12,792
20	410,173	1,319	8.16	25,722
30	498,855	2,840	13.88	40,558
40	570,703	893	6.20	20,478
50	618,072	496	4.45	16,528
60	648,322	435	4.74	17,159
70	666,051	371	3.50	10,767
80	675,789	266	3.28	9,721
90	680,896	204	3.50	9,552
100	683,184	120	1.72	3,600
110	683,957	0	0.20	277
120	684,059	0	0.37	304
130	684,059	13	1.21	639
140	684,059	10	1.22	640
150	684,059	2	1.07	618
160	684,059	2	0.99	663

NOTE: SU, spatial unit; BB, branch-and-bound; CPU, central processing unit.

654 contiguous variables (no slack variables counted), and 357 main constraints. We tested this model by re-solving Problem 1,179 times, once for each land parcel designated as *r*. In these experiments, we set $b = 800{,}000$, because Problem 1 was most difficult at this value (see Table 1), which makes it most forgivable to sacrifice accuracy for tractability. Depending on *r*, CPU times varied from 0.07 to 24.93 s with a median of 1.69 s, a mean of 2.32 s, and a sum of 414.95 s. These results suggest that fixing a sink significantly improves tractability. This is largely because the number of binary variables has been halved and constraints (9) are binding. Then, like Cova and Church (2000), it might be worthwhile to try this sink-dependent model with every potential sink in search of a global optimum. For the present example, this technique worked as it saved about 20% total CPU time—536.33 versus 414.95 s. Experiments with several different values for *b*, however, have found that if the original problem is relatively easy (whose *b* is not close to 800,000), there is no benefit in solving 179 still easier sub-problems. Thus, the root-fixing technique should be used only when a problem is sufficiently difficult or when there are not many possible *r*'s.

Regular SUs

The second problem involves a regular set of SUs, which is reproduced from Williams (2002). The problem takes place on a 10-by-10 grid, where each cell is

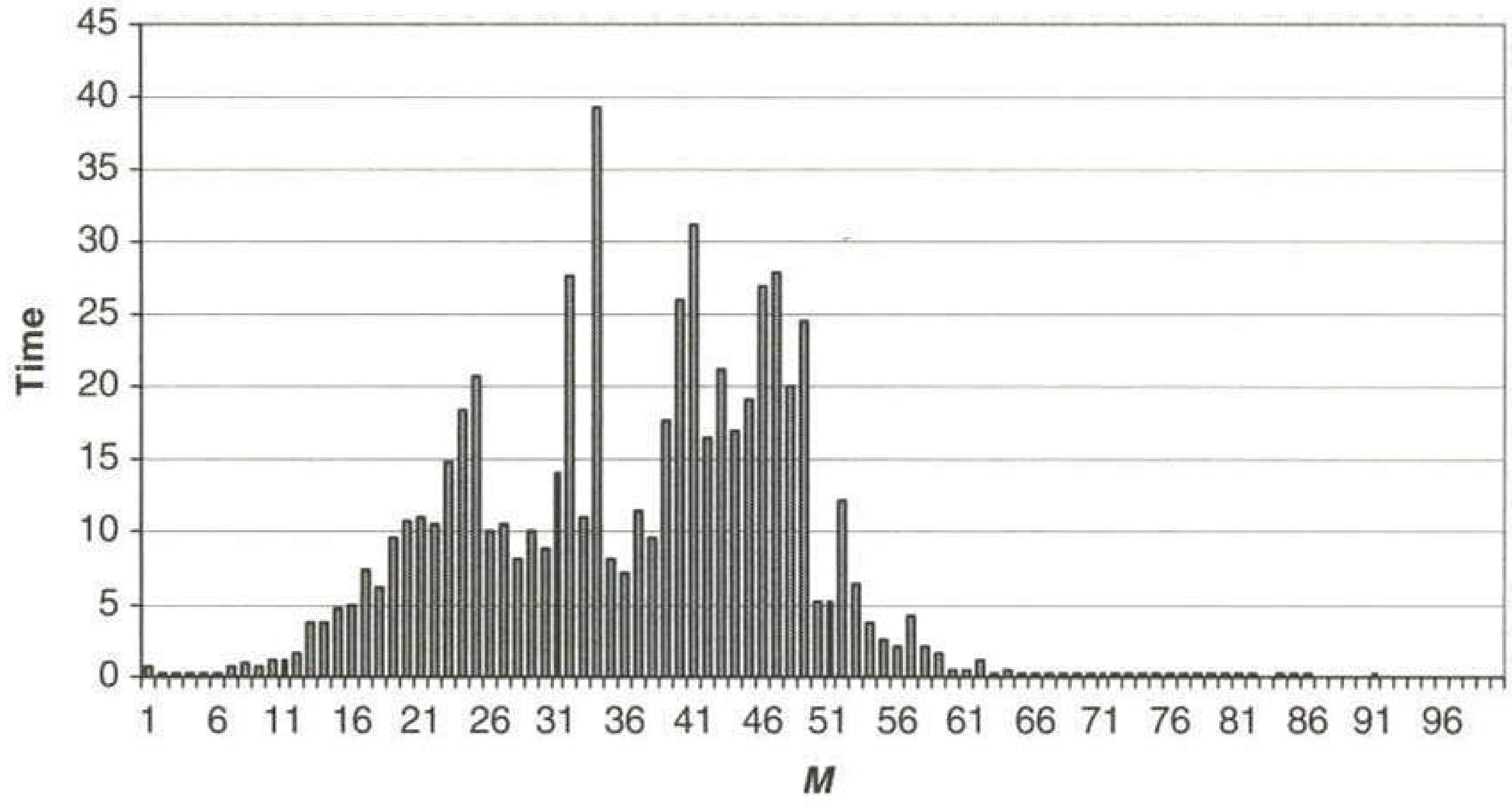

Figure 4. Distribution of solution times (in central processing unit seconds) for Problem 2 with the 10-by-10 grid. M indicates the number of selected spatial units.

assigned a random number taken from a uniform distribution of values ranging from 0.2 to 1.8 with a step size of 0.1 (see Williams (2002) for an illustration).

Problem 2. Select M contiguous cells from the grid to minimize the sum of the values of all selected cells.

The problem is formulated as follows:

$$\min \sum_{i \in I} c_i x_i \tag{16}$$

subject to (2)–(6), (8), and (16), where c_i indicates the value assigned to cell i.

We tested this model with 100 different values of M ranging from 1 to 100. All instances were solved optimally in less than 30 s, except for one case where $M = 34$, which took 39.3 s. The model's difficulty steadily grew as M increases until a first peak ($M = 26$) is reached. The model tends to be more difficult to solve when M is between about 40 and 50 (although the single most difficult case was when $M = 34$), but the solution becomes easy or trivial after M goes over this range (see Fig. 4).

Although this particular problem was small enough to be solved with little computational effort, it can easily become intractable as the number of SUs (or cells) increases. In fact, we failed to solve optimally a problem of selecting a medium-sized (160-cell) region from a similarly designed 20-by-20 grid. Five hours of computation only found a feasible solution with 26.7% optimality gap.

Conclusion

We have presented a new exact contiguity condition and have formulated it in terms of linear functions that can be incorporated into MIP models for SU allocation. Although SU allocation typically involves various criteria other than contiguity, the intention of this article is to provide a basic MIP framework to handle contiguity as such, which can then be extended to handle additional criteria. We have proved that any MIP model embedded with the general contiguity model guarantees to select a contiguous region from a given set of SUs regardless of any other requirement. We have also shown that the general contiguity constraints can be reduced to a significantly more efficient form, when a fixed number of SUs are selected and/or when a certain SU is selected in advance. Computational experiments suggest that MIP models using these constraints can handle small problems—such as one involving 100 regular grid cells or 179 irregular land parcels—in reasonable time on a typical workstation or personal computer. However, they will have computational difficulty solving larger problems. This is particularly true if additional complicating criteria are included. One such criterion is non-perforation, which the proposed constraints do not enforce.

Acknowledgements

The author would like to thank the Maryland-National Park and Planning Commission (M-NCPPC) and Norma J. Kawecki for providing cartographic data for this study, and Monique Guignard-Spielberg, Tony E. Smith, C. Dana Tomlin, Siqun (Sonia) Wang, and Mark D. Uyemura for their valuable comments on an earlier draft, and Simon McBride for his assistance in editing. The author is also indebted to the editor and anonymous reviewers for their suggestions, which have enhanced the structure and contents of this article. Any errors are, of course, the author's responsibility.

References

Aerts, J. C. J. H., E. Eisinger, G. B. M. Heuvelink, and T. J. Stewart. (2003). "Using Linear Integer Programming for Multi-Site Land-Use Allocation." *Geographical Analysis* 35, 148–69.

Aerts, J. C. J. H., and G. B. M. Heuvelink. (2002). "Using Simulated Annealing for Resource Allocation." *International Journal of Geographical Information Science* 16, 571–87.

Ahuja, R. K., T. L. Magnanti, and J. B. Orlin. (1993). *Network Flows: Theory, Algorithms, and Applications.* Englewood Cliffs, NJ: Prentice Hall.

Barahona, F., A. Weintraub, and R. Epstein. (1992). "Habitat Dispersion in Forest Planning and the Stable Set Problem." *Operations Research* 40, S14–21.

Belford, P. C., and H. D. Ratliff. (1972). "A Network-Flow Model for Racially Balancing Schools." *Operations Research* 20, 619–28.

Benabdallah, S., and J. R. Wright. (1991). "Shape Considerations in Spatial Optimization." *Civil Engineering Systems* 8, 145–52.

Benabdallah, S., and J. R. Wright. (1992). "Multiple Subregion Allocation Models." *ASCE Journal of Urban Planning and Development* 118, 24–40.

Brookes, C. J. (1997). "A Parameterized Region-Growing Programme for Site Allocation on Raster Suitability Maps." *International Journal of Geographical Information Science* 11, 375–96.

Cova, T. J., and R. L. Church. (2000). "Contiguity Constraints for Single-Region Site Search Problems." *Geographical Analysis* 32, 306–29.

Crema, S. C. (1996). "Comparison Between Linear Programming and a Choice Heuristic Approach to Multi-Objective Decision Making Using GIS." *Proceedings of GIS/LIS '96*, Denver, CO, 954–63.

Diamond, J. T., and J. R. Wright. (1988). "Design of an Integrated Spatial Information System for Multiobjective Land-Use Planning." *Environment and Planning B* 15, 205–14.

Eastman, J. R., H. Jiang, and J. Toledano. (1998). "Multi-Criteria and Multi-Objective Decision Making for Land Allocation Using GIS." In *Multi-Criteria Analysis for Land-Use Management, Environment and Management 9*, 227–51, edited by E. Beinat and P. Nijkamp. Dordrecht, the Netherlands: Kluwer Academic Publishers.

Fischer, D., and R. L. Church. (2003). "Clustering and Compactness in Reserve Site Selection: An Extension of the Biodiversity Management Area Selection Model." *Forest Science* 49, 555–65.

Fleischmann, B., and J. N. Paraschis. (1988). "Solving a Large Scale Districting Problem: A Case Report." *Computers and Operations Research* 15, 521–33.

Franklin, A. D., and E. Koenigsberg. (1973). "Computed School Assignments in a Large District." *Operations Research* 21, 413–26.

Garfinkel, R. S., and G. L. Nemhauser. (1970). "Optimal Political Districting by Implicit Enumeration Techniques." *Management Science* 16, B495–508.

Gilbert, K. C., D. D. Holmes, and R. E. Rosenthal. (1985). "A Multiobjective Discrete Optimization Model for Land Allocation." *Management Science* 31, 1509–22.

Hess, S. W., and S. A. Samuels. (1971). "Experiences with a Sales Districting Model: Criteria and Implementation." *Management Science* 18(4, Part II), 41–54.

Hess, S. W., J. B. Weaver, H. J. Siegfeldt, J. N. Whelan, and P. A. Zitlau. (1965). "Nonpartisan Political Redistricting by Computer." *Operations Research* 13, 998–1006.

Hojati, M. (1996). "Optimal Political Districting." *Computers and Operations Research* 23, 1147–61.

Marlin, P. G. (1981). "Application of the Transportation Model to a Large-Scale 'Districting' Problem." *Computers and Operations Research* 8, 83–96.

McDill, M. E., S. Rebain, and J. Braze. (2002). "Harvest Scheduling with Area-Based Adjacency Constraints." *Forest Science* 48, 631–42.

McDonnell, M. D., H. P. Possingham, I. R. Ball, and E. A. Cousins. (2002). "Mathematical Methods for Spatially Cohesive Reserve Design." *Environmental Modeling and Assessment* 7, 107–14.

Mehrotra, A., E. L. Johnson, and G. L. Nemhauser. (1998). "An Optimization Based Heuristic for Political Districting." *Management Science* 44, 1100–14.

Murray, A. T. (1999). "Spatial Restrictions in Harvest Scheduling." *Forest Science* 45, 45–52.

Nalle, D. J., J. L. Arthur, and J. Sessions. (2003). "Designing Compact and Contiguous Reserve Networks with a Hybrid Heuristic Algorithm." *Forest Science* 48, 59–68.

Önal, H., and R. A. Briers. (2002). "Incorporating Spatial Criteria in Optimum Reserve Network Selection." *Proceedings of the Royal Society of London: Biological Sciences* 269, 2437–41.

ReVelle, C. S. (1993). "Facility Siting and Integer-Friendly Programming." *European Journal of Operational Research* 65, 147–58.

Segal, M., and D. B. Weinberger. (1977). "Turfing." *Operations Research* 25, 367–86.

Shanker, R. J., R. E. Turner, and A. A. Zoltners. (1975). "Sales Territory Design: An Integrated Approach." *Management Science* 22, 309–20.

Shirabe, T., and C. D. Tomlin. (2002). "Decomposing Integer Programming Models for Spatial Allocation." In *Lecture Notes in Computer Science 2478: Proceedings of Second International Conference on Geographic Information Science*, 300–12, edited by M. J. Egenhofer and D. M. Mark. Berlin: Springer-Verlag.

Snyder, S., and C. ReVelle. (1996). "Temporal and Spatial Harvesting of Irregular Systems of Parcels." *Canadian Journal of Forest Research* 26, 1079–88.

Tomlin, C. D., and K. M. Johnston. (1990). "An Experiment in Land-Use Allocation with a Geographic Information System." In *Introductory Readings in Geographic Information Systems*, 159–69, edited by D. J. Peuquet and D. F. Marble. London: Taylor & Francis.

Williams, J. C. (2001). "A Linear-Size Zero-One Programming Model for the Minimum Spanning Tree Problem in Planar Graphs." *Networks* 39, 53–60.

Williams, J. C. (2002). "A Zero-One Programming Model for Contiguous Land Acquisition." *Geographical Analysis* 34, 330–49.

Williams, J. C. (2003). "Convex Land Acquisition with Zero-One Programming." *Environment and Planning B* 30, 255–70.

Wright, J., C. ReVelle, and J. Cohon. (1983). "A Multipleobjective Integer Programming Model for the Land Acquisition Problem." *Regional Science and Urban Economics* 13, 31–53.

Yeates, M. (1963). "Hinterland Delimitation: A Distance Minimizing Approach." *The Professional Geographer* 15(6), 7–10.

Zoltners, A. A., and P. Sinha. (1983). "Sales Territory Alignment: A Review and Model." *Management Science* 29, 1237–56.