Project MUSE®
http://muse.jhu.edu

Justin C. Williams

A Zero-One Programming Model for Contiguous Land Acquisition

The land acquisition problem is a spatial partitioning problem that involves selecting multiple parcels to be acquired for a particular land use. Three selection criteria are considered: total cost, total area, and spatial contiguity. Achieving contiguity or connectivity has been problematic in previous exact methods for land acquisition. Here we present a new zero-one programming model that enforces necessary and sufficient conditions for achieving contiguity in discrete cell landscapes, independent of other spatial attributes such as compactness. Computational experience with several demonstration problems is reported, and results and extensions are discussed.

Spatial partitioning problems, in which the landscape is divided into distinct regions or zones, are ubiquitous in geography and planning. They have appeared as "regionalization" problems (Nutenko 1970), "region building" problems (Cliff and Haggett 1970; Keane 1975), "districting" problems (Horn 1995; Williams 1995), "clustering" problems (Rosing and ReVelle 1986), "land acquisition/ allocation" problems (Wright, ReVelle, and Cohon 1983; Gilbert et al. 1985; Diamond and Wright 1991; Cova and Church 2000), and "reserve design" problems (Williams and ReVelle 1996). In addition, facility siting problems such as the p-median problem (ReVelle and Swain 1970), as well as the delineation of market areas in central place theory (Dacey 1965), also involve partitioning space into a set of distinct regions.

In this paper we address a particular type of spatial partitioning problem, the land acquisition problem, which can be described in the following way. We are given a two-dimensional *landscape* that is represented as a set of n discrete parcels or *cells*. The landscape is to be partitioned into two *regions* by assigning each cell to one region or the other. One of the regions contains those cells selected for acquisition (for a particular land use, for example, green space) and the other region contains the remaining (unselected) cells.

The land acquisition problem was introduced as an optimization problem by Wright, ReVelle, and Cohon (1983) who formulated a zero-one programming model with three objectives: minimize the total cost of selected cells; maximize total area; and maximize compactness. The last objective was achieved by minimizing the total external border length of selected cells. The authors employed multiobjective linear programming together with a branch- and-bound routine to find all pareto-optimal or

Justin C. Williams is associate research professor of geography and environmental engineering at Johns Hopkins University. E-mail: jcwjr@hunix.hcf.jhu.edu

Geographical Analysis, Vol. 34, No. 4 (October 2002) The Ohio State University
Submitted: 9/26/01. Revised version accepted: 3/8/02

noninferior solutions for the special case of a regular grid. Other researchers have also developed mathematical programming approaches for land acquisition/ allocation problems.

Gilbert, Holmes, and Rosenthal (1985) considered four objectives: minimize total cost; minimize the distance between the selected region and the nearest "amenity" cell; maximize the distance between the selected region and the nearest "detractor" cell; and minimize the value of the shape index (maximize compactness) of the selected region. A regular grid landscape was used, and a requirement to select exactly p cells served as a constraint on total area. The authors developed an implicit enumeration algorithm to generate and evaluate alternative contiguous regions. An interactive multiobjective method was used to identify noninferior solutions.

Diamond and Wright (1991) also developed an implicit enumeration algorithm, but for a slightly different problem. They considered two objectives: minimize the total cost of the selected region; and maximize the suitability of the selected cell least suitable for the intended land use. This algorithm generated contiguous regions whose total area and compactness were constrained by upper and lower bounds. The method was applicable to irregular as well as regular grids.

Recently, Cova and Church (2000) developed a linear zero-one programming model for delineating a single compact and contiguous region in a regular or an irregular grid. This approach incorporated mathematical constraints that guarantee contiguity of the region and also allow compactness (length of external border) to be specified.

These land acquisition models follow earlier quantitative modeling approaches to land use planning that created regions by allocating one of multiple possible land uses to each parcel in the landscape (for example, Gordon and MacReynolds 1974; Hopkins 1977; Arad and Berechman 1978). Allocations were made in order to minimize interaction costs between regions or maximize interaction benefits, or to achieve acceptable land use compatibility. The problems were typically formulated as linear or quadratic assignment models, depending on complexity and the chosen criteria, and were solved using exact methods or, for large quadratic assignment problems, heuristics.

In this paper we present a new zero-one programming model for the land acquisition problem. The objective is to minimize the total cost of selecting a region of specified area. The primary focus of this paper, though, is on achieving spatial connectivity, which in previous models has been a limiting factor in finding exact solutions. Here we develop a model that guarantees contiguity in the set of selected cells. The model is applied to the problem of creating two regions (cells selected for acquisition; cells not selected). Computational experience with 25-cell and 100-cell demonstration problems is reported, and results and extensions are discussed.

1. CONTIGUITY IN LAND ACQUISITION

A single contiguous region may be desirable or even necessary in land acquisition for a number of reasons: facilitating communication, transportation, migration, construction and/or maintenance functions within the region; and facilitating perimeter enclosure, monitoring, and/or security operations for the region. In addition, contiguous regions may have stronger visual, spatial, or conceptual coherence than unconnected regions.

Spatial contiguity is fairly easy to define and there seems to be general agreement in the geography and planning literature on what it means. Informally, a region in discrete space is said to be contiguous if there exists, between every pair of cells in the region, a path that is composed of only successively *adjacent* cells of the region. Two cells are said to be adjacent if they share a common edge or boundary of positive length; cells with only a common corner point are not adjacent under this rule ("nearest neighbor" rule). Hence, it is possible to travel from any cell to any other cell in a contiguous region without leaving the region. As this definition implies, the question of contiguity is a yes or no question–either a region is contiguous or it is not. (It may also be useful to relax this yes/no definition by considering different degrees of connectivity; this issue is addressed in section 4.3.)

In order to find an optimal contiguous region, any solution algorithm would need to first distinguish between contiguous and noncontiguous regions, and then evaluate all of the contiguous regions in terms of feasibility and optimality criteria. To this end it would be useful to know how many contiguous regions exist in a given landscape. Cliff and Haggett (1970) and Keane (1975) investigated this question for discrete landscapes: in how many different ways (A) can a landscape of n cells be partitioned into regions, where the number of regions, k, can take on every value from 1 to n? These authors provide formulas for calculating bounds on A under alternative rules for creating regions. An upper bound is based on the rule that allows any two cells to be placed together in the same region (contiguity is not enforced under this rule). A lower bound is derived from the restrictive assumption that the landscape is a line of cells and regions must be contiguous line segments. Both the upper and the lower bound on A grow exponentially with problem size n, although there is wide divergence between these bounds.

In the land acquisition problem (where $k = 2$), the number of ways to select cells for acquisition is 2^n under no constraint on region size and no connectivity requirement. If we also say that exactly p cells must be selected, then the number of regions is n-choose-p $\binom{n}{p}$ under no connectivity requirement. The number of ways to select a *contiguous* p-cell region depends, additionally, on the shape and cell adjacency structure of the landscape. Consider, for example, the problem of selecting a contiguous region of two cells in a 100-cell landscape. If the 100 cells are arranged in a line, then 99 different contiguous regions are possible (the left selected cell may be any cell 1 through 99). Now, suppose that the 100-cell line is folded into a 10-by-10 grid (Figure 2); 180 contiguous two-cell regions are now possible. If the region size were increased to 25 cells, the line would admit only 76 contiguous regions, but the grid would admit many more than 180 contiguous arrangements, although the exact number is difficult to determine. In general, counting the number of possible contiguous regions is a difficult combinatorial exercise except in cases where the underlying landscape is very restrictive (for example, a line of cells) or the region size (p) is very small. There is evidently no general formula for determining the number of contiguous regions in a landscape of arbitrary dimensions and cell adjacency structure (or for deriving good bounds on this number). Methods have been developed for generating all possible contiguous regions systematically (see, for example, March and Matela 1974; Mills, Sherwell, and Van Rooyan 1995), although such enumeration processes become increasingly time consuming as p and n get larger.

Methods employed for enforcing contiguity in spatial partitioning problems include both heuristic (approximate) approaches and exact approaches. Effective and flexible heuristics have been developed by many researchers; two recent examples are the methods of Horn (1995) and Mehrotra, Johnson, and Nemhauser (1998) for political districting. Heuristics can typically find good and sometimes optimal solutions, but they can neither guarantee mathematical optimality nor determine the deviation from optimality in a solution.

Depending on the optimality criterion, the task of finding a mathematically optimal contiguous region can be challenging even for small problems, and difficult or impossible for medium-sized and large problems. Two successful approaches have been the implicit enumeration algorithms of Gilbert, Holmes, and Rosenthal (1985) and Diamond and Wright (1991). These algorithms have been most effective when applied to relatively small landscapes, or to larger landscapes in which the selected region is small or must have a very compact shape.

Gilbert, Holmes, and Rosenthal (1985) successfully applied their method to a 900-cell landscape but with a region size of only five cells. Diamond and Wright (1991) used their method to solve a variety of problems in which the landscape size ranged from 25 to 950 cells but the selected region was limited to ten or fewer cells. It is unclear how these approaches would perform in finding optimal regions larger than five to ten cells; but the rapid growth in the number of possible contiguous regions as region size increases beyond several cells suggests large increases in computing time. The two algorithms were also most effective under "min-max" or "max-min" (bottleneck) objectives. In contrast, under a "min-sum" objective, such as minimizing total cost, the problem was found to be NP-hard (Gilbert, Holmes, and Rosenthal 1985).

In implicit enumeration algorithms, contiguity can be guaranteed because the algorithm can be coded to generate only contiguous candidate regions. In some cases (for example, under bottleneck objectives) a contiguity requirement may even improve the algorithm's performance. In mathematical programming models, however, contiguity has been difficult to enforce. The difficulty lies in formulating constraints (ideally, a small set of linear constraints) that enforce both necessary and sufficient conditions for contiguity, that is, constraints that screen out all noncontiguous regions but no contiguous region. A necessary condition for achieving contiguity is that each cell in a multi-cell region must be adjacent to at least one other cell in the region. But for $p \geq 4$, this condition is not sufficient to guarantee contiguity (one or more disconnected pairs of adjacent cells may result). A sufficient condition for contiguity is to require that a region have a suitably high level of compactness (for example, require that the region be nearly circular or square in shape). But this requirement precludes regions that are contiguous but not extremely compact.

In the mathematical programming model of Cova and Church (2000), a large number of linear contiguity constraints was needed for the most general case of admitting any contiguous region as a possible solution. Their model became tractable only when the selected region was required to satisfy certain types of compactness conditions or when a "root" cell was preselected for the region. But these conditions excluded many contiguous regions from consideration.

Contiguity is often discussed together with compactness. As a result, there may be some confusion between the two as well as the tendency to use compactness as a substitute for contiguity. Although compactness may be defined and measured in different ways, a compact region can be thought of informally as a region whose cells are in close proximity to each other. Austin (1984) and Young (1988) review alternative measures of compactness. We note that contiguity and compactness are not equivalent, nor does one imply the other. A region can be contiguous but not compact (for example, a long line of cells), or compact but not contiguous (for example, two nearby but disconnected squares). Hence, using compactness as a substitute for contiguity may, on one hand, preclude what might otherwise be a desirable contiguous solution, or, on the other hand, result in a noncontiguous solution.

To the author's knowledge no general, practical mathematical programming method exists for enforcing contiguity in land acquisition and other spatial partitioning problems. There is a "need for a tractable constraint form that captures all contiguous site patterns regardless of spatial characteristics" (Cova and Church 2000, p. 328). The aim of this research is toward developing such a model. The model presented in the next sections enforces necessary and sufficient conditions for contiguity, independent of other spatial attributes such as compactness.

2. A GRAPH-THEORETICAL APPROACH TO LAND ACQUISITION WITH CONTIGUITY

In developing a model for land acquisition with contiguity, we make use of several concepts from graph theory and network optimization. In a previous paper, the author presented a zero-one programming model for finding minimum spanning trees (MST) in planar graphs (Williams 2001). That model provides a theoretical basis for the land acquisition model below. The basic idea is that the landscape, as a mosaic of cells, can be represented as a *planar graph* with vertices and edges, $G(V,E)$. A planar graph is a graph that can be constructed in the Cartesian plane so that no two edges intersect except at vertices (Chartrand and Oellermann 1993). Vertices in the graph represent cells in the landscape, and edges represent cell adjacencies (Figures 1a and 1b).

An important feature of planar graphs is that they have *dual graphs*. The edges and vertices of a planar graph (the "primal" graph) partition the plane into a set r of zones, all but one of which are enclosed or bounded. The dual graph is constructed by placing a dual vertex in each of the zones, including the unbounded zone, and then, for each edge in the primal graph, drawing a dual edge that joins the two dual vertices separated by the primal edge (Figure 1c). The dual of a planar graph is itself a planar graph. Note that the edges of the primal and dual graphs form *intersecting pairs*, with one primal edge and one dual edge in each pair.

In the MST model of Williams (2001), the special primal-dual structure of planar graphs was used to enforce contiguity in spanning trees of the graph (a spanning tree uses $n-1$ edges to connect all n vertices of a graph). The MST model generates both a (minimum) spanning tree in the primal graph and a second spanning tree in the dual graph. The two spanning trees are *complementary* in the sense that none of the edges in the primal tree intersect any of the edges of the dual tree, and the two trees together form a partition of the set of intersecting edge pairs (Figure 1d).

In formulating the MST model, the following further specifications were made in the primal and dual graphs. In the primal graph, an arbitrary vertex was designated the "terminal" vertex (vertex n, for convenience). In addition, each (undirected) edge of the primal graph was represented by two directed arcs: one arc (i,j) directed from vertex i to vertex j, and another arc (j,i) directed from j to i. Similarly in the dual graph, vertex r was specified arbitrarily as the dual terminus and each dual edge was represented by two directed arcs. For those edges incident to the terminal vertices, only one arc, directed into the terminus, was used (Figure 1e). The directed arcs were used to indicate the eligible (adjacent) successor vertices for each vertex.

The use of directed arcs and the complementary nature of the primal and dual spanning trees were exploited to enforce contiguity in the formation of both spanning trees. For each vertex in the primal graph, except the terminus, we required exactly one arc directed from the vertex to an adjacent successor vertex to be selected for the spanning tree (outflow requirement). This outflow requirement was also made for the dual graph. Hence, n-1 arcs in the primal graph and $r-1$ arcs in the dual graph were selected for the spanning trees, necessary conditions for trees with n and r arcs, respectively. Furthermore, the "interwoven" structure of the primal and dual graphs prevented cycles in both trees when this outflow requirement was satisfied. Any cycle in the primal tree would force one or more vertices in the dual tree to be disconnected from the dual terminus and violate the outflow requirement for the dual. Similarly, any cycle in the dual tree would disconnect the primal tree and violate the outflow requirement for the primal.

This method of enforcing contiguity in spanning trees is applied to the land acquisition problem in the following way. Suppose we have a spanning tree of an n-cell landscape. This n-vertex spanning tree can serve as the backbone for a p-vertex subset or "subtree" of the spanning tree, which represents a region of p cells in the landscape. The critical aspect is this: by requiring the subtree to contain $p-1$ edges (a

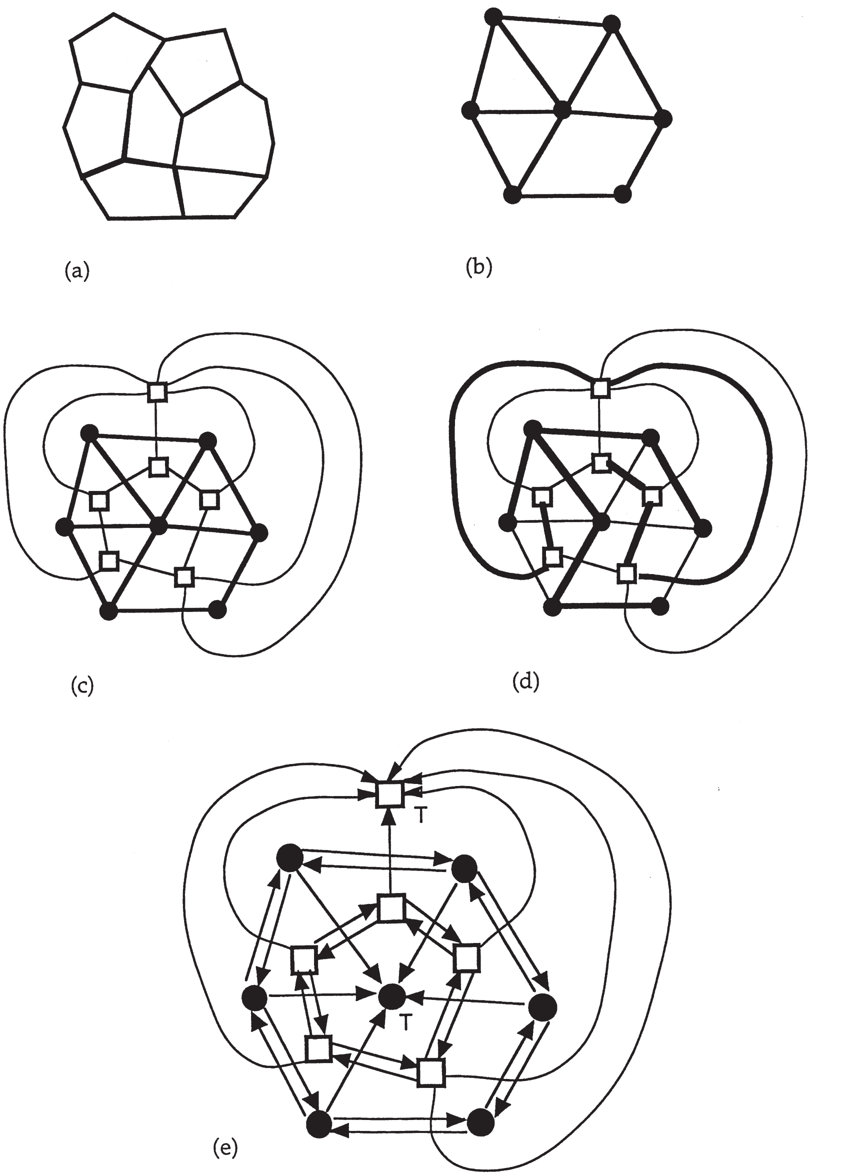

FIG. 1. Cellular Landscape and Corresponding Planar Graph. (a) A landscape of 7 cells. (b) A planar graph representation of the landscape; vertices represent cells and edges represent cell adjacencies. (c) Primal planar graph (circular vertices and bold edges) and its dual graph (square vertices and thin edges); $n = 7$, $r = 6$. (d) Complementary spanning trees (bold edges) in the primal and dual graphs. (e) Primal and dual graphs with edges drawn as directed arcs; "T" indicates the designated terminal vertices.

necessary condition for a p-vertex subtree), the subtree—and hence the cellular region it represents—is forced to be contiguous. The land acquisition problem, then, can be stated as a problem of finding an optimal subtree of a spanning tree within a planar graph.

In the model below, complementary spanning trees are created in both the primal planar graph (the cellular landscape) and the dual graph. The actual spanning trees formed depend on the optimality criterion, which seeks an optimal p-vertex subtree. That is, selection of an optimal subtree drives the choice of the spanning tree backbone. Contiguity of the subtree is enforced by requiring the subtree to be a subset of the spanning tree backbone in the primal graph, and by specifying that the difference between the number of vertices and number of edges in the subtree must equal 1. We note that a subtree representation of a contiguous p-cell region may not be unique (it may be possible to represent a region by different subtrees).

3. A DECISION MODEL FOR LAND ACQUISITION WITH CONTIGUITY

A zero-one programming model is formulated for identifying a minimum-cost contiguous region of specified area. The contiguity requirement is made only for the region of selected cells, and not for the remaining unselected cells. We assume that the landscape itself is contiguous. We first formulate the model for cellular landscapes that conform to a regular rectangular grid (raster system). In regular grids, all cells have equal area, which we specify as one unit. The total area of the region can then be represented by the total number of selected cells (p). Modifications for adapting the model to irregular grids, in which cells have varying sizes, shapes, and adjacencies, are discussed in section 4.2. The cost (Ci) of a cell i indicates the purchase price of the cell plus any improvement costs that would be needed. In the case of public-sector investment, cell cost might represent the opportunity cost of foregone land uses. Alternatively, cost might represent the level of unsuitability of a cell for the intended land use.

3.1 Model Notation

Indices, Sets, Parameters

- i, j, I are the indices and set of primal vertices (cells), where $i, j = 1, \ldots, n$;
- k, l, K are the indices and set of dual vertices, where $k, l = 1, \ldots, r$;
- Di is the set of primal vertices (cells) j that are adjacent to primal vertex (cell) i;
- Dk is the set of dual vertices l that are adjacent to dual vertex k;
- Ci is the cost of primal vertex (cell) i;
- p is the total number of vertices (cells) to select ($1 \leq p \leq n$);
- q is the number of cell clusters specified for the selected region under a relaxed connectivity requirement (see section 4.3); and
- q^* is the number of cell clusters in a selected region with no connectivity requirement (see section 4.3).

Decision Variables

- Xij = 1, if directed arc (i,j) in the primal graph is selected for the primal spanning tree *and is also* selected for the subtree, and
  = 0, otherwise;
- Yij = 1, if directed arc (i,j) in the primal graph is selected for the primal spanning tree *but is not* selected for the subtree, and
  = 0, otherwise;

Zkl	$=1$, if directed arc (k,l) in the dual graph is selected for the complementary dual spanning tree, and $=0$, otherwise;
Ui	$=1$, if primal vertex (cell) i is selected for acquisition, and $=0$, otherwise.

3.2 Formulation

$$\text{Minimize:} \quad \sum_{i \in I} Ci\,Ui \tag{1}$$

$$\text{Subject to:} \quad \sum_{j \in DI} Xij + \sum_{j \in Di} Yij = 1\,, \qquad \text{for all primal vertices } i = 1,\ldots,n-1 \tag{2}$$

$$\sum_{l \in Dk} Zkl = 1\,, \qquad \text{for all dual vertices } k = 1,\ldots,r-1 \tag{3}$$

$$Xij + Yij + Xji + Yji + Zkl + Zlk = 1, \quad \text{for all pairs of intersecting primal and dual arcs} \tag{4}$$

$$Xij + Xji \le Ui\,, \quad Xij + Xji \le Uj \qquad \text{for all primal arcs } (i, j),\ i < j \tag{5}$$

$$\sum_{j \in Di} Xij \le Ui\,, \qquad \text{for all primal vertices } i = 1,\ldots,n-1 \tag{6}$$

$$\sum_{i \in I} Ui = p \tag{7}$$

$$\sum_{i \in I} \sum_{j \in Di} Xij = p - 1 \tag{8}$$

$$Xij; Yij; Zkl; Ui \in \{0,1\} \tag{9}$$

3.3. Description of the Model

The objective function (1) minimizes the total cost (or average per-cell cost) of the selected region, which is measured as the sum of the costs of selected cells. Constraints (2) stipulate that for each vertex (cell) i in the primal graph, except the terminus n, exactly one primal arc directed from i to some adjacent vertex j must be selected for the primal spanning tree (outflow requirement). Furthermore, the selected primal arc must be either selected for the subtree (that is, the selected region), in which case $Xij = 1$, or not selected for the region, in which case $Ykl = 1$. A separate constraint is written for each primal vertex except vertex n. Constraints (3) are similar to (2) in that they stipulate that for each vertex k in the dual graph, except the dual terminus r, exactly one dual arc directed from k to some adjacent vertex l must be selected for the dual spanning tree. A separate constraint is written for each dual vertex except vertex r.

Constraints (4) force exactly one arc to be selected from each set of intersecting primal and dual arcs. (These sets correspond to the pairs of intersecting primal and dual edges.) These constraints guarantee the complementarity of the primal and dual spanning trees by ensuring that their respective edges do not intersect. A separate constraint is written for each set of intersecting primal and dual arcs. These constraints, in conjunction with (2) and (3), prevent cycles in the primal and dual trees.

Together, constraints (2), (3), and (4) create complementary spanning trees in the primal and dual graphs. Constraints (5) through (8), in turn, create a connected subtree of the primal spanning tree that represents a p-cell contiguous region. Constraints (5) stipulate that if either of the primal arcs (i,j) or (j,i) is selected for the subtree, then both of the incident vertices must also be selected for the subtree (that is, cells i and j must be selected for the region). A separate pair of constraints is written for each edge in the primal graph.

Constraints (6) stipulate that if any of the arcs directed out of vertex i are selected for the subtree, then vertex i must be selected for the subtree (that is, cell i must be selected for the region). A separate constraint is written for each primal vertex i except terminus n. These constraints are not strictly necessary in that they enforce no logical condition not already enforced by (2) and (5). However, they are included because they were found to be effective in improving the computational performance of the model.

Constraint (7) requires the subtree to contain p vertices (that is, the selected region must contain p cells). Constraint (8) requires the subtree to contain p-1 edges. The statements in (9) indicate that the decision variables are binary integer variables. (We note that the variables Xij, Yij, and Zkl may be specified as non-negative continuous variables and they will still take on binary values in an optimal solution in which all of the Ui variables are zero or one)

4. COMPUTATIONAL EXPERIENCE

Three computational experiments were performed to demonstrate the above model in applications to regular grids, irregular grids, and under conditions of relaxed connectivity.

4.1 Experiment 1, Contiguous Regions on a Regular Grid

In the first experiment, the model (1) through (9) was applied to a hypothetical landscape of 100 square cells arranged in a 10-by-10 grid. Cell costs were represented by random numbers taken from a uniform distribution with a range of [0.2, 1.8] and a step size of 0.1 (Figure 2). The problem was solved for 51 values of p, ranging from 5 to 95 (selected results appear in Table 1). For each value of p a minimum-

0.7	1.2	0.2	0.4	1.3	0.8	0.8	1.4	1.3	1.3
1.6	0.4	0.9	0.4	0.4	0.4	1.4	0.4	0.2	1.7
1.2	0.3	1.8	0.8	1.4	0.2	0.8	0.6	0.5	1.2
1.7	0.8	0.7	1.7	1.8	1.8	1.4	1.6	0.6	0.5
1.0	1.5	1.3	1.3	0.7	0.9	1.0	0.3	1.2	1.2
1.6	1.4	0.5	0.2	1.3	1.2	0.3	0.4	1.4	1.3
0.6	0.9	0.3	0.5	1.3	0.6	0.3	0.6	0.6	1.8
1.1	0.6	0.8	1.0	0.6	0.6	0.3	1.3	1.8	1.5
0.9	0.9	1.5	0.9	0.2	0.8	0.6	1.5	0.7	0.6
1.1	1.0	1.4	0.9	0.6	1.1	0.5	1.4	0.3	1.5

FIG. 2. Experiment 1, Cell Costs for 100-Cell Regular Grid

TABLE 1
Experiment 1, Contiguous Region on Regular Grid (n = 100), Selected Results

(A) p	(B) *Obj-R*	(C) *Obj-I*	(D) *Time (3.0)*	(E) *Time (7.1)*	(F) *Nodes (3.0)*	(G) *Nodes (7.1)*
95	85.10	85.10	0.81	0.83	0	0
90	76.80	76.80	1.19	0.19	0	0
85	69.20	69.20	1.18	0.27	0	0
80	62.20	62.20	2.25	0.21	3	0
75	55.42	55.50	2.05	1.21	16	0
70	48.92	49.00	3.66	0.66	18	0
65	42.65	42.80	3.43	0.81	24	3
60	36.93	37.10	5.12	2.30	29	21
55	31.93	32.40	12.11	8.06	101	97
50	27.43	28.00	58.04	16.50	447	313
45	23.35	24.00	69.47	18.69	686	593
40	19.51	20.30	293.93	86.85	2,606	2,335
35	16.01	16.80	224.74	37.53	2,531	1,301
30	12.75	13.90	1,044.62	95.45	11,669	4,013
25	9.91	11.30	1,258.95	67.02	13,262	3,499
20	7.52	8.90	629.79	99.81	7,155	3,722
15	5.39	6.60	617.32	78.99	4,715	3,251
10	3.46	4.00	20.88	9.79	211	274
5	1.60	1.60	1.66	0.12	0	0

Descriptions of Columns

(A) p is the number of cells (total area) selected for the contiguous region.
(B) *Obj-R* is the objective function value (cost) of the optimal solution to the relaxed linear program, that is, without the integer requirements on the decision variables.
(C) *Obj-I* is the objective function value (cost) of the integer-optimal solution.
(D) *Time (3.0)* is the computing time to find the integer-optimal solution in CPU seconds using the CPLEX 3.0 solver.
(E) *Time (7.1)* is the computing time to find the integer-optimal solution in CPU seconds using the CPLEX 7.1 solver.
(F) *Nodes (3.0)* is the number of branch and bound nodes needed to find the integer-optimal solution using CPLEX 3.0.
(G) *Nodes (7.1)* is the number of branch and bound nodes needed to find the integer-optimal solution using CPLEX 7.1.

cost contiguous region was found (Figure 3). Computing was performed on a Silicon Graphics workstation using CPLEX 3.0 mixed integer program solver.

Computational performance varied greatly, depending on the value of p. Run times for finding integer-optimal solutions were relatively fast when p was less than 10 or greater than 50, and were slowest for values of p between 20 and 30 (Table 1, columns D and F). The run times ranged from less than a second ($p = 95$) to a worst case of about 38 minutes ($p = 27$). Similarly, the amount of branching and bounding ranged from none ($p = 85$, 90, 95) to a worst case of 23,575 branch nodes ($p = 27$). Such relatively long computing times resulted even though strategies were employed to enhance computational performance. First, the constraints (6) proved to be effective in reducing the amount of branching and bounding (by up to 99 percent) relative to parallel trials conducted without these constraints. Second, an upper bounding strategy on the objective function (total cost) was used. The optimal objective function value for a particular value of p was used to derive an upper bound on the objective function for the next lower value of p.

For comparison, the model was applied to a larger problem ($n = 144$, 12-by-12 grid). For $p = 7$ in the 144-cell problem, the run time was roughly the same as for $p = 11$ in the 100-cell problem (about 24 seconds). However, for values of p between 14 and 69 in the larger problem, the run times exceeded the worst case time in the smaller problem. These results are consistent with prior results (Gilbert, Holmes, and Rosenthal1985; Diamond and Wright 1991), which indicate that computing time depends on the value of p as well as the value of n. As n increases, the range of values of p for which a problem remains tractable tends to decline.

We also had the opportunity to solve the 100-cell problem with newer software, CPLEX 7.1, run on a Dell Optiplex personal computer. Both solution times and the amount of branching and bounding declined significantly using this newer system

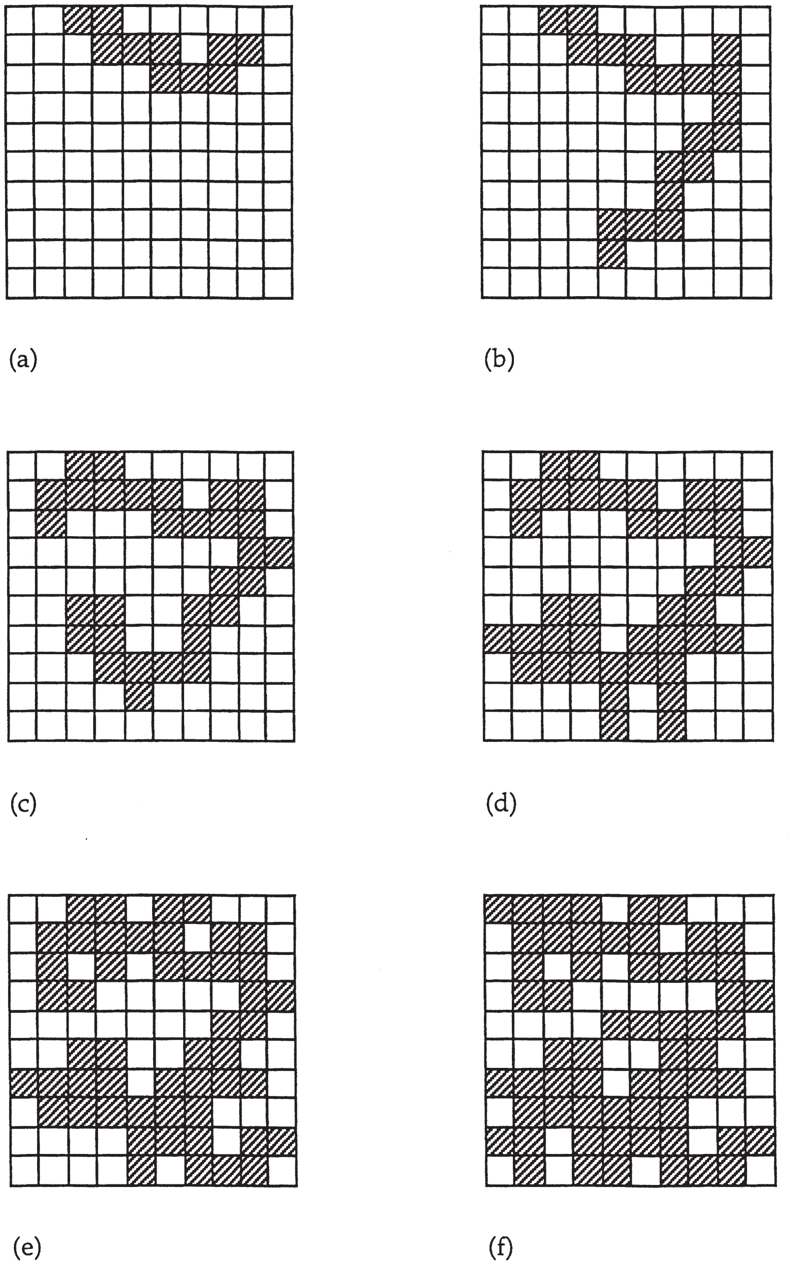

FIG. 3. Experiment 1, Least Cost Contiguous Regions. (a) 10 cells are selected for the region ($p = 10$), cost = 4.0. (b) $p = 20$, cost = 8.9. (c) $p = 30$, cost = 13.9. (d) $p = 40$, cost = 20.3. (e) $p = 50$, cost = 28.0. (f) $p = 60$, cost = 37.1.

(Table 1, columns E and G). The worst solution time of the 19 trials reported in Table 1 was about 100 seconds (for $p = 20$).

4.2 Experiment 2, Regular and Irregular Grid Comparisons

In the second experiment, we sought to compare the model's performance with respect to regular and irregular grids of 25 cells. Computing was performed using CPLEX 7.1 on the personal computer. In part (a) of this experiment, the model (1) through (9) was applied to a 5-by-5 regular grid. Cells were assigned random costs on the range [0.2, 1.8] (Figure 4a). The value of p was varied between 4 and 16. As shown in Table 2, columns B and C, optimal solutions were found in less than one second for each value of p (for example, Figure 4c).

In parts (b) and (c) of this experiment, we transformed the 5-by-5 regular grid into a 25-cell irregular grid in which the cells had different shapes and adjacencies (Figure 4b). In part (b), each of the cells was assigned an area of one unit, as in the regular grid. The primary difference between the two grids was in the structure of cell adjacencies. In applying models (1) through (9) to the irregular grid, computational performance declined relative to the regular grid (Table 2, columns D and E; Figure 4d). This indicates that problem tractability can be influenced by landscape configuration (graph structure) independent of landscape size.

In part (c), we used the irregular grid from part (b) but assigned the cells different areas, ranging from 0.5 to 1.5 (the total area of the grid remained 25 units). This further change required modifying the model because the region's specified area could no longer be expressed as the *number* of selected cells. Constraints (7) and (8) were replaced by a new constraint

$$\sum_{i \in I} Ui - \sum_{i \in I} \sum_{j \in Di} Xij = 1 \tag{10}$$

to ensure the correct relationship between the number of vertices and number of arcs in the selected subtree. We also added a lower bound constraint on the selected region's area,

$$\sum_{i \in I} Ai\, Ui \geq p \tag{11}$$

TABLE 2

Experiment 2, Contiguous Region, Regular Grid versus Irregular Grid ($n = 25$)

(A) p	(B) *Time (a)*	(C) *Nodes (a)*	(D) *Time (b)*	(E) *Nodes (b)*	(F) *Time (c)*	(G) *Nodes (c)*
16	0.81	6	0.72	0	0.78	0
14	0.02	0	0.17	0	0.33	15
12	0.11	1	0.15	14	0.11	0
10	0.14	6	0.37	25	0.40	20
8	0.17	5	0.57	73	1.43	132
6	0.19	7	0.27	3	0.21	0
4	0.07	2	0.70	77	0.51	36

Descriptions of Columns

(A) p is the total area specified for the selected region: it is the number of selected cells in the regular grid problem of part (a) and in the irregular grid, equal area problem of part (b); and it is a lower bound on region's area in the irregular grid, unequal area problem of part (c).
(B) *Time (a)* is the computing time to find the integer-optimal solution in CPU seconds for part (a), regular grid.
(C) *Nodes (a)* is the number of branch and bound nodes needed to find the integer-optimal solution for part (a), regular grid.
(D) *Time (b)* is the computing time to find the integer-optimal solution in CPU seconds for part (b), irregular grid, equal area.
(E) *Nodes (b)* is the number of branch and bound nodes needed to find the integer-optimal solution for part (b), irregular grid, equal area.
(F) *Time (c)* is the computing time to find the integer-optimal solution in CPU seconds for part (c), irregular grid, unequal area.
(G) *Nodes (c)* is the number of branch and bound nodes needed to find the integer-optimal solution for part (c), irregular grid, unequal area.

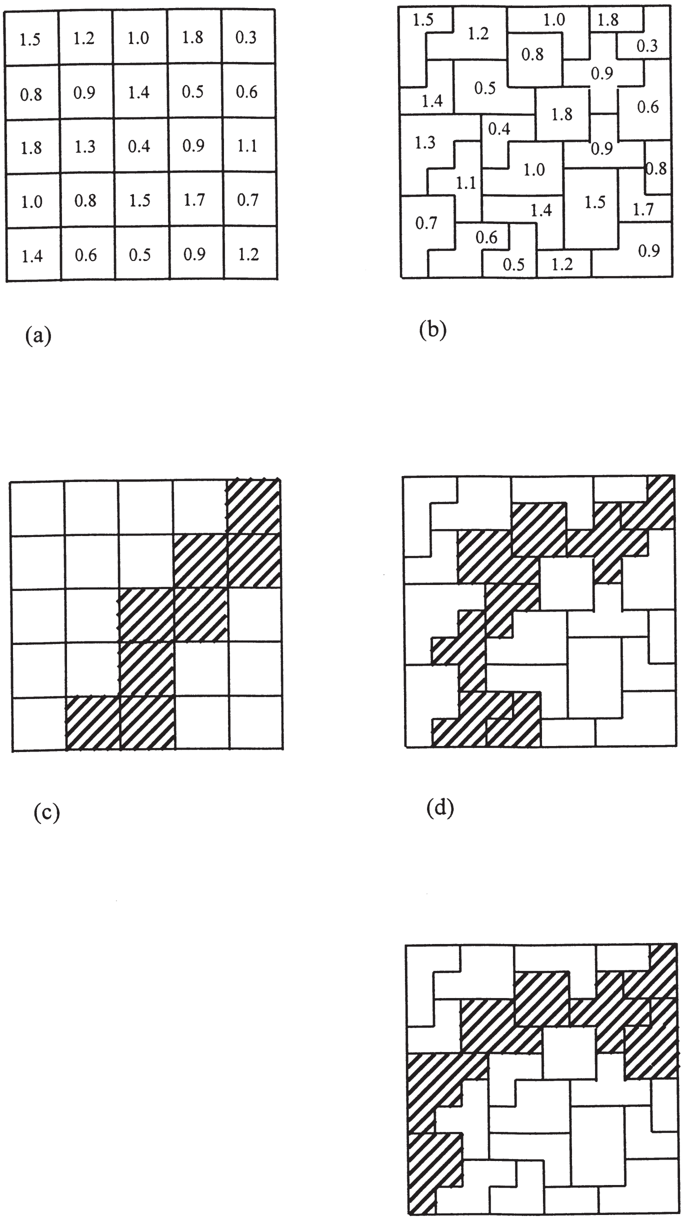

FIG. 4. Experiment 2, Regular Grid and Irregular Grid, 25 Cells. (a) Cell costs for regular grid (b) Cell costs for irregular grid. (c) Regular grid least cost contiguous region, $p = 8$, cost $= 5.3$. (d) Irregular grid, equal area cells, least cost contiguous region, $p = 8$, cost $= 5.1$. (e) Irregular grid, unequal area cells, least cost contiguous region, $p = 8$, area $= 8.25$, cost $= 5.1$.

where Ai is the area of cell (primal vertex) i. Overall, the computational performance was similar to that of part (b), although the worst-case run time (1.43 seconds for p = 8) exceeded every run time in (b) (Table 2, columns F and G). Longer run times tended to occur when the selected region's area turned out to be larger than the lower bound (p). For example, when the lower bound was specified as 8, the area of the least-cost contiguous region turned out to be 8.25 (Figure 4e). We note that constraints such as (11) are recognized as being "integer unfriendly" and may add to the computational burden of finding integer-optimal solutions (ReVelle 1993). One way around this drawback is to restate (11) as an objective of maximizing the region's area:

$$\text{Maximize:} \quad \sum_{i \in I} Ai\, Ui \tag{12}$$

Multiobjective optimization methods (for example, Cohon 1978) could then be applied to objectives (1) and (12) in order to generate solutions that represent efficient trade-offs between cost and area without using an area constraint.

4.3 Experiment 3, Relaxing Contiguity

Contiguity represents one extreme on a spectrum of connectivity. If a region contains p cells, the number of clusters (q) in the region may range from one (contiguous region) to p (every cell is disconnected from every other cell). We use the parameter q as a measure of the level of connectivity achieved; low values of q imply high levels of connectivity and vice versa (other measures of landscape connectivity are discussed by van Langevelde, van der Knaap, and Classen 1998). Because the cost of a p-cell region may depend upon this aspect of its configuration, it may be useful to consider the trade-off between the objectives of minimizing cost and maximizing the level of connectivity achieved. In this experiment we show how the model can be applied to the problem of controlling the level of connectivity when strict contiguity is not required.

If we wished to find a least-cost p-cell region under no requirement for connectivity, we would simply select the p least expensive cells. This solution would contain some number of clusters, q^*, where $1 \leq q^* \leq p$. We could also identify regions with higher levels of connectivity ($q \leq q^*$), although these regions would have a higher total cost. In selecting a least-cost region, then, we would like to be able to control the number of clusters (q) on the range $1 \leq q \leq q^*$.

Controlling the value of q in the model can be done in the following way. A p-cell region containing q contiguous clusters is analogous to p vertices distributed among q subtrees in a graph. As noted above, when $q = 1$ (contiguous region), the number of edges in the (single) subtree is $p-1$, one less than the number of vertices. In general, given q clusters, the total number of edges is $p-q$. Hence, in order to specify the number of clusters in the optimal solution, we would change the right-hand side of constraint (8) from $p-1$ to $p-q$, for regular grids. For irregular grids, we would change the right-hand side of (10) from 1 to q. Holding the region's area constant at p, the number of clusters q can be traded off against total cost. (Note that the specified value of q represents an upper bound on the number of clusters; the actual number of clusters will be less than q if two or more clusters happen to be adjacent.)

In this experiment the model (1) through (9) was applied to the 100-cell regular grid used in experiment 1. Computing was performed using CPLEX 7.1 on the personal computer. We sought to identify the trade-off between cost and connectivity, holding the region's total area (p) constant at 40. To begin the experiment, we imposed no connectivity requirement and simply selected the 40 least expensive cells; this yielded 11 clusters ($q^* = 11$) (Figure 6d). This solution was identified in less than a second and required no branching and bounding. The value of q was then varied between 1 and $q^* = 11$. In so doing, we identified eleven distinct least-cost regions, each 40 units in size and containing the specified number of clusters (q) (the solution for $q = 1$ was found in experiment 1). Solution times quickly declined as the connectivity requirement was relaxed (Table 3). Figure 5 shows how both the shape and the cost of the region change as the number of clusters comprising it increases from one to eleven (see also Figures 3d and 6d).

With these cost data, the value of q^* happened to be unique for $p = 40$, but q^* was not unique for other values of p. For $p = 30$, for example, 924 alternate regions existed, each having a (minimum) cost of 12.0 but a different number of clusters (q^* values ranged from 6 to 13). In such cases, the minimum of the q^* values (highest level of connectivity) would likely be of greatest interest. The (minimum) values of q^* for the 19 values of p in Table 1 are shown in Table 4. As p decreased from 95 to 5, the value of q^* increased to a maximum of 11 (at $p = 40$) and then fluctuated between 5 and 9.

5. DISCUSSION

Based on the above experiments, the computational burden for finding integer-optimal solutions (computing time, amount of branching and bounding) varied significantly with the values of parameters p (region area) and q (number of clusters in the region) for a given landscape size n. Computational burden was largest for $q = 1$ and $p = 0.2\,n$ to $0.3\,n$. We suggest that this variation is the result of three spatial factors and the interactions among these factors:

(a) The *combinatorial* factor. The number of possible ways to choose p cells from n cells (n-choose-p) reaches a maximum when $p = n/2$. Based on this factor alone, we might (erroneously) expect solution times to be highest when the value of p is near $n/2$ because a maximum number of potential solutions would need to be evaluated.

(b) The *number of clusters* factor. When no requirement for connectivity is made, the problem is trivial and can be solved by simply selecting the p least expensive cells (yielding q^* clusters). Except for relatively large values of p, this solution is unlikely to be contiguous, and would need to be altered to achieve higher levels of connectivity. When q^* is close to 1, only a small modification may be needed to make the solution contiguous. For example, two clusters might be turned into a single contiguous cluster by exchanging a few selected cells for a few (costlier) unselected cells. The num-

TABLE 3

Experiment 3, Regions with Varying Levels of Connectivity ($n = 100, p = 40$)

(A) q	(B) *Obj-R*	(C) *Obj-I*	(D) *Time*	(E) *Nodes*
1	19.51	20.30	86.85	2,335
2	19.10	19.80	19.53	592
3	18.92	19.20	4.81	96
4	18.80	18.90	0.58	6
5	18.73	18.80	0.51	4
6	18.65	18.70	0.62	4
7	18.60	18.60	0.17	0
8	18.55	18.60	0.29	0
9	18.50	18.50	0.22	0
10	18.45	18.50	0.25	0
11 (q^*)	18.40	18.40	0.17	0

Descriptions of Columns

(A) q is the number of clusters specified for the selected region.
(B) *Obj-R* is the objective function value (cost) of the optimal solution to the relaxed linear program, that is, without the integer requirements on the decision variables.
(C) *Obj-I* is the objective function value (cost) of the integer-optimal solution.
(D) *Time* is the computing time to find the integer-optimal solution in CPU seconds.
(E) *Nodes* is the number of branch and bound nodes needed to find the integer-optimal solution.

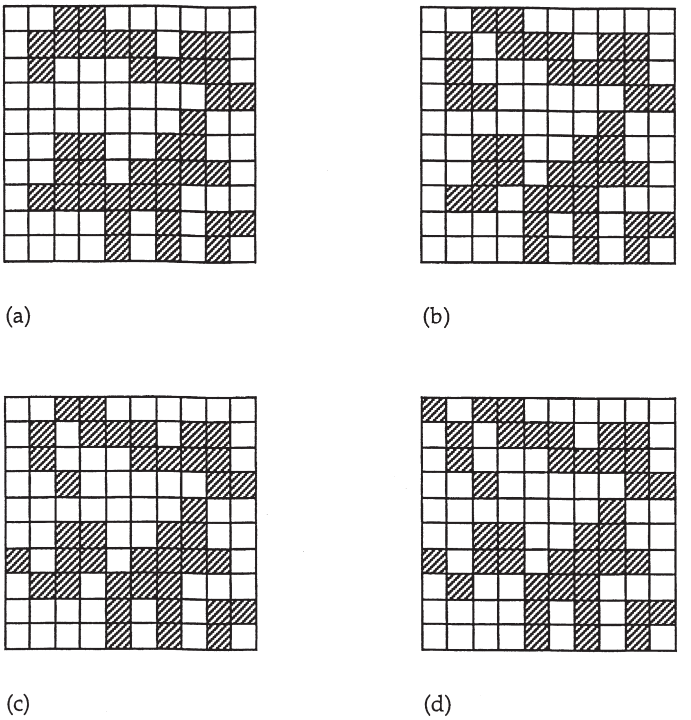

FIG. 5. Experiment 3, Least Cost Regions with Varying Levels of Connectivity. (a) The selected region contains 40 cells and 3 clusters ($p = 40$, $q = 3$), cost = 19.2. (b) $p = 40$, $q = 5$, cost = 18.8 (c) $p = 40$, $q = 7$, cost = 18.6. (d) $p = 40$, $q = 9$, cost = 18.5. For $p = 40$, $q = 1$ see Figure 3d. For $p = 40$, $q = q^* = 11$, see Figure 6d.

TABLE 4

Experiment 3, Regions with No Connectivity Requirement ($n = 100$), Selected Results

(A) p	(B) q^*	(C) *Obj-WoC*	(D) *Obj-C*	(E) *Obj-Gap*	(F) *Obj-Gap %*
95	1	85.10	85.10	- 0 -	- 0 -
90	1	76.80	76.80	- 0 -	- 0 -
85	1	69.20	69.20	- 0 -	- 0 -
80	2	62.10	62.20	0.1	0.16
75	3	55.20	55.50	0.3	0.54
70	3	48.70	49.00	0.3	0.62
65	3	42.40	42.80	0.4	0.94
60	5	36.50	37.10	0.6	1.64
55	5	31.30	32.40	1.1	3.51
50	6	26.70	28.00	1.3	4.87
45	8	22.40	24.00	1.6	7.14
40	11	18.40	20.30	1.9	10.33
35	8	15.00	16.80	1.8	12.00
30	6	12.00	13.90	1.9	15.83
25	8	9.00	11.30	2.3	25.56
20	7	6.40	8.90	2.5	45.31
15	9	4.30	6.60	2.3	53.49
10	8	2.50	4.00	1.5	60.00
5	5	1.00	1.60	0.6	60.00

Descriptions of Columns

(A) p is the number of cells (total area) selected for the region ($n = 100$).
(B) q^* is the (minimum) number of clusters in an integer-optimal solution with no connectivity requirement.
(C) *Obj-WoC* is the objective function value (cost) of an integer-optimal solution with no connectivity requirement.
(D) *Obj-C* is the objective function value of the integer-optimal contiguous solution (from Table 1, column C).
(E) *Obj-Gap* is the difference between *Obj-C* and *Obj-WoC*.
(F) *Obj-Gap %* is the percentage increase in *Obj-C* over *Obj-WoC*.

ber of computations needed to evaluate just a few exchanges would be relatively small. When q^* is large, however, more clusters would need to be joined, suggesting a greater number of cell exchanges and a greater computational effort to evaluate the different possible exchanges. Indicative of this factor is the difference between the objective function value of the optimal contiguous solution and that of the solution with no connectivity requirement (*Obj-Gap* in Table 4). In the demonstration problem, *Obj-Gap* tended to be widest when q^* was relatively large, and narrowest when q^* was small.

(c) The *distance between clusters* factor. As the spatial distances that separate clusters increase, the number of cell exchanges that would be needed to bridge these distances and achieve contiguity would probably increase, with a corresponding increase in computing time. (We refer to the distance between the nearest edges of clusters.) The average intercluster distance in the q^* solutions tend to increase as the density of selected cells declines, that is, as p decreases relative to n (Figure 6). Of course, average intercluster distance is highly dependent on the spatial distribution of cell costs, especially at low values of p.

For large values of p (for example, $p \geq 0.70\ n$) all three factors tend to promote computational efficiency: the number of possible combinations is relatively low; the value of q^* tends to be close to 1; and, because most cells in the landscape are selected, intercluster distances tend to be short. For values of p in the mid to high range (for example, $0.45\ n \leq p \leq 0.70\ n$), all three factors combine to drive up computing times as p declines. The number of combinations, the number of clusters, and intercluster distances are larger at the lower end of this range. For values of p in the low to mid range (for example, $0.15\ n \leq p \leq 0.45\ n$), the number of combinations starts to decline; but this is more than offset by the relatively large number of clusters and relatively large intercluster distances, which drive up computing time. For small values of p (for example, $p \leq 0.15\ n$), the relatively low number of combinations and relatively low number of clusters promote computational efficiency, but this is partially offset by relatively large intercluster distances.

In addition to these three factors, computing times were also influenced by whether or not the landscape conformed to a regular grid. In experiments on a small landscape ($n = 25$), less computing was needed to find integer-optimal solutions on regular grids than irregular grids. Both the pattern of cell adjacencies (graph structure) and whether irregular cells have equal or differing areas may impact computing time.

The shape attributes of solutions are also of interest. As indicated by Figure 3, the minimum-cost contiguous regions may not be particularly compact. At lower values of p the selected region may be sinuous, and at higher values of p it may be riddled with inlets, peninsulas, and holes. These spatial characteristics may also remain present when strict contiguity is relaxed to allow multiple clusters (Figures 5 and 6). These characteristics result from the particular spatial distribution of cell costs together with an absence of any compactness-promoting objective or constraint in the model.

Identifying an optimal contiguous region without regard for compactness may be desirable in some land acquisition problems (for example, in selecting parcels for stream corridor buffers)—and the model presented here would be appropriate for such cases. In other cases, some level of compactness may be needed in addition to achieving contiguity, and the ability to control compactness would be useful. Linear constraints for controlling external border length (one measure of compactness) were developed by Wright, ReVelle, and Cohon (1983), and could be added to the model. This would allow compactness to be achieved to the desired extent, without having to rely on it for contiguity. An exploration of this possibility is suggested as further research.

In the model presented here, two regions are created, but contiguity is required

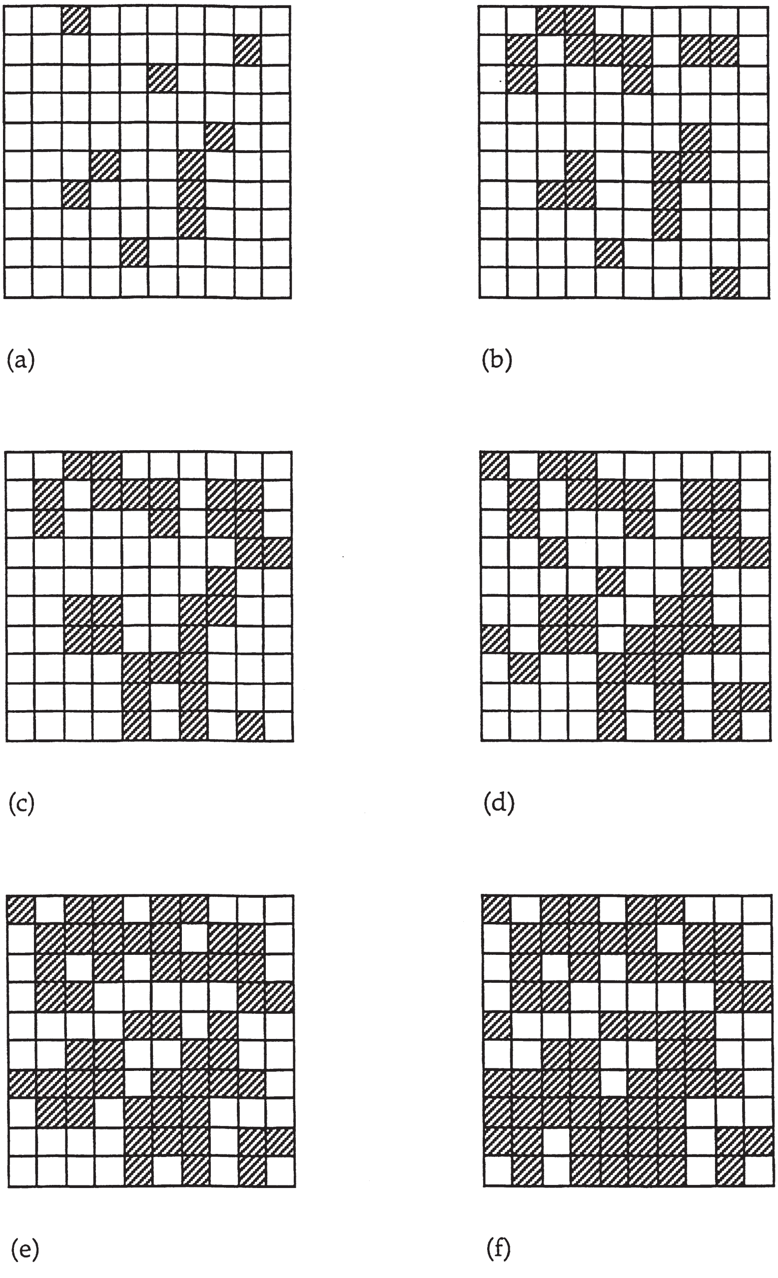

FIG. 6. Experiment 3, Least Cost Regions with No Connectivity Requirement. (a) The selected region contains 10 cells and 8 clusters, $p = 10$, $q^* = 8$, cost = 2.5. (b) $p = 20$, $q^* = 7$, cost = 6.4. (c) $p = 30$, $q^* = 6$, cost = 12.0. (d) $p = 40$, $q^* = 11$, cost = 18.4. (e) $p = 50$, $q^* = 6$, cost = 26.7. (f) $p = 60$, $q^* = 5$, cost = 36.5.

only for the region of selected cells. (Contiguity of the selected region does not guarantee contiguity of the unselected cells, as indicated by Figures 3c-f.) We expect that the model can be reformulated to enforce contiguity in both regions. A generalization of this two-region problem involves partitioning the landscape into k contiguous regions. It may also be possible to reformulate the model for the k-region problem—another area for further research.

6. SUMMARY AND CONCLUSIONS

In this paper a new zero-one programming model is presented for the land acquisition problem. The intent is to identify a contiguous region of discrete parcels or cells under the objective of minimizing total land cost. This model provides necessary and sufficient conditions for achieving contiguity in both regular and irregular parcel systems, independent of other spatial attributes such as compactness. No contiguous region is precluded a priori as a potential solution. The model also enables strict contiguity to be relaxed so that varying levels of connectivity may be achieved. The size of the model is "linear," that is, the number of variables and constraints grows linearly with the number of cells in the landscape (n). Computational experience indicates that exact solutions can be found in reasonable amounts of time (100 seconds or less) for medium-sized problems ($n = 100$) using commercially available software on a personal computer. However, solution times appear to be sensitive to values specified for the parameters n, p (number of cells in the selected region), and q (number of clusters in the selected region), as well as to whether or not the landscape is a regular grid. Possible extensions of the model include adding constraints to control compactness, as well as reformulating the model to delineate k contiguous regions. Successful solution of the 100-cell demonstration problem on a personal computer suggests that in more-powerful computing environments the model may be applicable to larger landscapes (for example, 1,000 or more cells). At this scale of application, GIS-based planning and decision support systems are typically used for analysis, and the model could potentially be included as part of such a system.

LITERATURE CITED

Arad, R. W., and J. Berechman (1978). "A Design Model for Allocating Interrelated Land-Use Activities in Discrete Space." *Environment and Planning A* 10, 1319–32.

Austin, R. F. (1984). "Measuring and Comparing Two-Dimensional Shapes." In *Spatial Statistics and Models*, edited by G. L. Gaile and C. J. Willmott, pp. 293–312. Dordrecht, Holland: D. Reidel Publishing Co.

Chartrand, G., and O. R. Oellermann (1993). *Applied and Algorithmic Graph Theory*. New York: McGraw-Hill.

Cliff, A. D., and P. Haggett (1970). "On the Efficiency of Alternative Aggregations in Region-Building Problems." *Environment and Planning* 2, 285–94.

Cohon, J. (1978). *Multiobjective Programming and Planning*. New York: Academic Press.

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

Dacey, M. F. (1965). "The Geometry of Central Place Theory." *Geografiska Annaler* 47B(2), 111–24.

Diamond, J. T., and J. R. Wright (1991). "An Implicit Enumeration Technique for the Land Acquisition Problem." *Civil Engineering Systems* 8, 101–14.

Gilbert, K. C., D. D. Holmes, and R. E. Rosenthal (1985). "A Multiobjective Discrete Optimization Model for Land Allocation." *Management Science* 31(12), 1509–22.

Gordon, P., and W. K. MacReynolds (1974). "Optimal Urban Forms." *Journal of Regional Science* 14(2), 217–31.

Hopkins, L. D. (1977). "Land-Use Design–Quadratic Assignment and Central Facility Models." *Environment and Planning A* 9, 625–42.

Horn, M.E.T. (1995). "Solution Techniques for Large Regional Partitioning Problems." *Geographical Analysis* 27 (July), 230–48.

Kean, M. (1975). "The Size of the Region Building Problem." *Environment and Planning A* 7, 575–77.

March, L., and R. Matela (1974). "The Animals of Architecture: Some Census Results on N-omino Populations for $N = 6, 7, 8$." *Environment and Planning B* 1, 193–216.

Mehrotra, A., E. L. Johnson, and G. L. Nemhauser (1998). "An Optimization Based Heuristic for Political Districting." *Management Science* 44(8), 1100–14.

Mills, G., D. Sherwell, and M. Van Rooyen (1995). "Integrated Housing Policy in South Africa: Cost Constraints on Built Form." *Environment and Planning B* 22, 5–20.

Nutenko, L. Y. (1970). "A Mathematical Method for Partitioning Territory." *Soviet Geography* 11(6), 480–9.

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

ReVelle, C. S., and R. Swain (1970). "Central Facilities Location." *Geographical Analysis* 2, 30–42.

Rosing, K. E., and C. S. ReVelle (1986). "Optimal Clustering." *Environment and Planning A* 18, 1463–76.

van Langevelde, F., W. G. M. van der Knaap, and G. D. H. Classen (1998). "Comparing Connectivity in Landscape Networks." *Environment and Planning B* 25, 849–63.

Williams, J. C. (1995). "Political Redistricting: A Review." *Papers in Regional Science* 74(1), 13–40.

______ (2001). "A Linear-Size Zero-One Programming Model for the Minimum Spanning Tree Problem in Planar Graphs." *Networks* 39(1), 53–60.

Williams, J. C., and C. S. ReVelle (1996). "A 0–1 Programming Approach to Delineating Protected Reserves." *Environment and Planning B: Planning and Design* 23, 607–24.

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

Young, H. P. (1988). "Measuring the Compactness of Legislative Districts." *Legislative Studies Quarterly* 13(1), 105–15.