geographical analysis

Geographical Analysis ISSN 0016-7363

The *p*-Regions Problem

Juan C. Duque,[1] Richard L. Church,[2] Richard S. Middleton[3]

[1]Research in Spatial Economics (RiSE), Department of Economics, EAFIT University, Medellin, Colombia, [2]Department of Geography, University of California at Santa Barbara, Santa Barbara, CA, [3]Earth and Environmental Sciences Division, Los Alamos National Laboratory, Los Alamos, NM.

The p-regions problem involves the aggregation or clustering of n small areas into p spatially contiguous regions while optimizing some criteria. The main objective of this article is to explore possible avenues for formulating this problem as a mixed integer-programming (MIP) problem. The critical issue in formulating this problem is to ensure that each region is a spatially contiguous cluster of small areas. We introduce three MIP models for solving the p regions problem. Each model minimizes the sum of dissimilarities between all pairs of areas within each region while guaranteeing contiguity. Three strategies designed to ensure contiguity are presented: (1) an adaptation of the Miller, Tucker, and Zemlin tour-breaking constraints developed for the traveling salesman problem; (2) the use of ordered-area assignment variables based upon an extension of an approach by Cova and Church for the geographical site design problem; and (3) the use of flow constraints based upon an extension of work by Shirabe. We test the efficacy of each formulation as well as specify a strategy to reduce overall problem size.

Introduction

The *p*-regions problem involves the aggregation of a finite set of *n* small areas into a set of *p* regions, where each region is geographically connected, while optimizing a predefined objective function. This problem is referred to by a host of different names, including the zonation, districting, and regionalization problem. It is related to a family of problems that are classified as nondeterministic polynomial-time hard (NP-hard) (Cliff et al. 1975; Keane 1975). Many spatial optimization models found in the regional science literature fall into this class; for example, the multiple facility location problem, the maximal covering location problem, and the land acquisition problem. Virtually all of these problems have been the subject of considerable research in terms of model formulation and algorithm design. The combinatorial complexity of such problems, including the *p*-regions problem, has led researchers

Correspondence: Juan C. Duque, Research in Spatial Economics (RiSE), Department of Economics, EAFIT University, Medellin, Colombia
e-mail: jduquec1@eafit.edu.co

Submitted: July 30, 2008. Revised version accepted: November 9, 2009.

to focus primarily on the use and development of heuristic solution methods (Openshaw 1988; Browdy 1990; Macmillan and Pierce 1994; Horn 1995; Openshaw and Rao 1995). For the *p*-regions problem, a heuristic approach typically starts with a nonoptimal but spatially contiguous regional configuration, and then iteratively moves or reassigns areas from a region to one of its adjacent regions in an attempt to improve objective quality while retaining spatial contiguity (Lankford 1969; Murtagh 1985; Gordon 1996; Duque, Ramos, and Surinach 2007). Unlike other computationally complex problems (e.g., the discrete multiple facility location problem), little or no emphasis has been placed on the formulation of an exact mixed integer-programming (MIP) model for the *p*-regions problem.

Even though most *p*-regions problems remain too large to consider solving using exact methods, general-purpose optimization software has improved over the last decade, and computational resources continue to be increased (Bixby 2002). This means that problem instances now are possible to solve optimally that were previously considered to be too big or complex to solve (Bixby et al. 2000).[1] Testing this boundary between problem size and the use of heuristics versus an optimal approach requires a model formulation. Unfortunately, few *p*-regions models have been formulated for this purpose (i.e., to be solved as a MIP problem). Thus, we pursue in this article three different formulations for this problem. This is not a straightforward task, as ensuring spatial contiguity among the defined regions is difficult, if not seemingly impossible.

Mixed integer-linear programming models for *p*-regions problem

Developing an exact model for the *p*-regions problem is dominated by one principal issue: the formulation of constraints needed to ensure spatial contiguity within each defined region.

Objective function

P-regions models ultimately attempt to optimize some objective function, typically maximizing a measure of overall intraregional homogeneity, which is functionally equivalent to minimizing heterogeneity. Gordon (1999) calculates heterogeneity for region k, C_k, as

$$H(C_k) \equiv \sum_{ij \in C_k | i < j} d_{ij}. \tag{1}$$

Equation (1) measures a region's heterogeneity as the sum of d_{ij} values associated with all possible area–area pairs within the region, where d_{ij} denotes a distance measure between areas i and j. In our formulation, d_{ij} can be relaxed to be a dissimilarity measure because it has to satisfy only the following properties: (a) $d_{ij} = d_{ji}$, (b) $d_{ij} \geq 0$, and (c) $d_{ij} = 0$ if $i = j$.[2]

Unlike other methodological approaches for spatial aggregation, the attribute variables that characterize each area, and that are utilized to calculate d_{ij}, can be nongeographical variables, such as socioeconomic profiles. This is possible because our models do not rely on geographical distances between areas to ensure spatial contiguity.

Given this definition of heterogeneity within a given region k, we can define the summed heterogeneity of p regions as the sum of each region's heterogeneity:

$$P(H, \Sigma) \equiv \sum_{k=1}^{p} H(C_k). \tag{2}$$

Accordingly, we can define one form of the p-regions problem as follows: ''Cluster n areas into p spatially contiguous regions, while minimizing the value of $P(H, \Sigma)$.''[3]

To illustrate how the objective function works, Fig. 1 shows a simple example using a regular lattice with nine areas. Figure 1a shows the grid grayscale coded according to the median housing price per area (y). Figure 1b shows a feasible way to aggregate nine areas into two regions.

In order to calculate the objective function that corresponds to this feasible solution, determination of the functional form that will be utilized to calculate the distance, or dissimilarity, between two areas is necessary. In this example, we apply a univariate Euclidean distance to measure how different two areas, i and j, are in terms of their mean housing price (d_{ij}). These distances are presented in Table 1.

Given a feasible solution and a distance function, Table 2 shows how to calculate both the heterogeneity of each region (i.e., $H(C_1)$ and $H(C_2)$) and the objective function $P(H, \Sigma)$.[4]

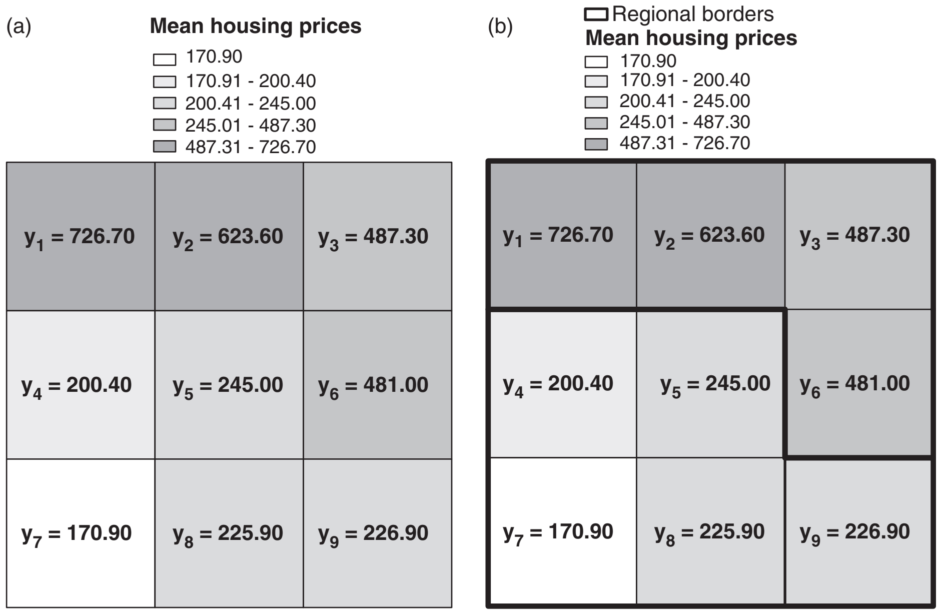

Figure 1. Example of input data and a feasible solution (a) Mean housing price per area (y); (b) Feasible solution for two regions.

Table 1 Pairwise Distances $d_{ij} = \sqrt{(y_i - y_j)^2} | i < j$

	2	3	4	5	6	7	8	9
1	103.1	239.4	526.3	481.7	245.7	555.8	500.8	499.8
2		136.3	423.2	378.6	142.6	452.7	397.7	396.7
3			286.9	242.3	6.3	316.4	261.4	260.4
4				44.6	280.6	29.5	25.5	26.5
5					236	74.1	19.1	18.1
6						310.1	255.1	254.1
7							55.0	56.0
8								1.0

Spatial contiguity

For a *p*-regions model solution to be considered feasible, the areas that form a given region must form a single contiguous region. Spatial contiguity poses the greatest obstacle in terms of producing exact methods for solving any *p*-regions problem. Although several methods exist that permit inclusion of constraints forcing spatial contiguity, all appear to require significant numbers of constraints and variables, resulting in a model of considerable size (Garfinkel and Nemhauser 1970; Macmillan and Pierce 1994; Mehrotra, Johnson, and Nemhauser 1998; Duque 2004).

Most MIP models for the *p*-regions problem are based on graph theory, where areas to be aggregated are represented by nodes on a network, and links are used to represent area adjacencies (Zoltners and Sinha 1983). Conceptually, given a graph $G(n, l)$ with n nodes and l links, the *p*-regions problem involves selecting a set of links to create *p*-disconnected subnetworks or trees, where each tree represents a connected set of areas (or nodes) representing a region. A tree can exist as an isolated node, because no a priori limit exists for the size of any specific region. The important feature is that each subnetwork or tree represents a region. To ensure that each subnetwork or region is connected, the subnetwork must be a tree; that is, contain no cycles.

The fundamental impediment in developing a MIP model is to design a constraint structure to ensure that a feasible solution contains no cycles within each identified tree. Typically, methods proposed to prevent cycles have been prohibitively costly to solve in terms of CPU run time. The literature is not clear about how this prevention can be accomplished, which, unfortunately, is a necessary condition for solving the *p*-regions problem as a MIP problem. In this article, we propose three new formulations, all quite different in structure, to identify an efficient model form for preventing cycles.

Table 2 Construction of the Objective Function $P(H, \Sigma)$

Expressions	Values
$H(C_1 = \{1, 2, 3, 6\})$	$d_{1,2}+d_{1,3}+d_{1,6}+d_{2,3}+d_{2,6}+d_{3,6} = 103.1+239.4+245.7+136.3+142.6+6.3 = 873.4$
$H(C_2 = \{4, 5, 7, 8, 9\})$	$d_{4,5}+d_{4,7}+d_{4,8}+d_{4,9}+d_{5,7}+d_{5,8}+d_{5,9}+d_{7,8}+d_{7,9}+d_{8,9}=44.6+873.4+29.5+25.5+26.5+74.1+19.1+18.1+55.0+56.0+1.0 = 349.4$
$P(H, \Sigma) = H(C_1)+H(C_2)$	$873.4+349.4 = 1222.8$

Formulating the *p*-regions problem in terms of a MIP model

The three MIP *p*-regions models (PRM) that are presented in this article have been inspired by different areas of spatial optimization research and are accordingly named:

- Tree^{PRM}: a forest of trees, with one tree per region. Cycles are prevented in each tree based upon the properties of three sets of constraints. This model is further constrained by redundant but effective cut constraints inspired by constraints developed by Miller, Tucker, and Zemlin (1960), known as the MTZ constraints, which were originally developed for the traveling salesman problem.
- Order^{PRM}: areas are added to a given region in a specified order, where order prevents cycles and ensures contiguity.
- Flow^{PRM}: a model inspired by the work of Shirabe (2005). This approach ensures contiguity by establishing a unit flow from each area (node) within a region to a selected sink of the region. The trace of flows for a given region represents a tree graph rooted at the designated regional sink. Flow can be routed only along arcs between nodes selected for the same region, thereby ensuring contiguity of the areas selected for the region.

Tree^{PRM}

In this model, links (represented by x_{ij} variables) are chosen to force each area i to appear in one of p-distinct trees or subnetworks. Cycles for each tree are prevented by a combination of several constraints. This first model is rather distinct in that regional membership is inferred from a set of variables, t_{ij}, whereas the subsequent two models (Flow^{PRM} and Order^{PRM}) are based upon an index that represents a given region. In general, we represent regions using the index k. To define a Tree model, consider the following problem parameters:

$$i, I = \text{index and set of areas, } i = \{1, \cdots, n\};$$

$$c_{ij} = \begin{cases} 1, \text{ if areas } i \text{ and } j \text{ share a border, with } i, j \in I \text{ and } i \neq j, \\ 0, \text{ otherwise;} \end{cases} .$$

$$N_i = \{j | c_{ij} = 1\}, \text{ the set of areas that are adjacent to area } i;$$

$$d_{ij} = \text{dissimilarity relationships between areas } i \text{ and } j \text{ with } i,\ j \in I \text{ and } i < j.$$

The model is based upon the following decision variables:

$$t_{ij} = \begin{cases} 1, \text{ if areas } i \text{ and } j \text{ belong to the same region,} \\ 0, \text{ otherwise;} \end{cases}$$

$$x_{ij} = \begin{cases} 1, \text{ if the arc or link between adjacent areas } i \text{ and } j \text{ is selected} \\ \text{ for a tree graph,} \\ 0, \text{ otherwise;} \end{cases} .$$

$$u_i = \text{ order assigned to each area } i \text{ in a subnetwork or tree.}$$

Consequently, we can define this first model as follows:
Minimize

$$Z = \sum_{i} \sum_{j|j>i} d_{ij} t_{ij}, \tag{3}$$

subject to

$$\sum_{i=1}^{n} \sum_{j \in N_i} x_{ij} = n - p, \tag{4}$$

$$\sum_{j \in N_i} x_{ij} \leq 1 \quad \forall i = 1, \cdots, n, \tag{5}$$

$$t_{ij} + t_{im} - t_{jm} \leq 1 \quad \forall i, j, m = 1, \cdots, n \text{ where } i \neq j, m \neq j, \tag{6}$$

$$t_{ij} - t_{ji} = 0 \quad \forall i, j = 1, \cdots, n, \tag{7}$$

$$x_{ij} - t_{ij} \leq 0 \quad \forall i = 1, \cdots, n; \forall j \in N_i, \tag{8}$$

$$u_i - u_j + (n - p) \times x_{ij} + (n - p - 2) \times x_{ji} \leq n - p1 \quad \forall i = 1, \cdots, n; \forall j \in N_i, \tag{9}$$

$$1 \leq u_i \leq n - p \quad \forall i = 1, \cdots, n, \tag{10}$$

$$x_{ij} \in \{0, 1\} \quad \forall i = 1, \cdots, n; \forall j \in N_i, \tag{11}$$

$$t_{ij} \in \{0, 1\} \quad \forall i, j = 1, \cdots, n \text{ where } j > i. \tag{12}$$

The objective function (3), which is identical for all of the models, minimizes the sum of dissimilarities between all pairs of areas within each region. The variables x_{ij} represent the selection of arcs or links for each tree graph, of which there are *p*. Each connected group of arcs represents a region and forms a tree graph (i.e., contains no cycles). If the entire problem space is divided into one region, the associated tree graph contains $n - 1$ arcs. In general, if the entire problem is divided into *p* regions, the sum of all arcs across all selected tree graphs represents the selection of $n - p$ arcs or links.

Constraint (4) establishes that the sum of all selected links (i.e., $x_{ij} = 1$) across all trees equals $n - p$. This condition by itself does not eliminate cycles from among the selected arcs. Constraints (5) require that any area *i* can have only one link x_{ij} selected for the tree that is directed away from node *i*. Constraints (6) ensure that if area *i* is part of the same region as *j* and *m*, then areas *j* and *m* also must be classified as being in the same region. Constraints (7) ensure symmetry in the matrix t_{ij}. Thus, if $t_{ji} = 1$ with $j < i$, then t_{ij} must equal 1; otherwise, the objective function remains unaffected. Constraints (8) require that if a link between adjacent areas *i* and *j* is selected for a tree graph, then areas *i* and *j* must be in the same region (i.e., $t_{ij} = 1$). Together constraints (4), (5), and (8) ensure that the arcs or links chosen are from adjacent areas grouped into the same region, but does not necessarily ensure that the subnetworks contain no cycles and therefore represent tree graphs. Constraints (9) and (10) are structured to prevent cycles among the selected trees. Essentially, this set of constraints represents a form of the tour-breaking constraints that can be found in Miller, Tucker, and Zemlin (1960). The u_i variables represent an integer value given to an area, and the value of each u_i variable represents an "order" that is assigned to an area of a region. Constraints (9) make a set of x_{ij} that are one in value impossible to form a cycle in any subgraph, because arcs chosen to lead away from an area i, $x_{ij} = 1$, must lead from an area assigned a u_i value that is less than the value assigned to area j. A cycle cannot exist because the arc chosen to complete a cycle would lead from an area with a higher u_i to an area with a lower u_j value. Thus, as structured, these constraints prevent any cycles among the selected subgraphs determined by the x_{ij} variables when $x_{ij} = 1$. Constraints (10) establish upper bounds on the values of u_i. Because each tree could contain at most $n - p$ arcs or links, we can specify that individual u_i values do no need to be any larger than $n - p$. If we wanted to maintain an upper limit on the number of areas that could be assigned to a given region, we could reduce the upper limit of the u_i values accordingly. Finally, constraints (11) and (12) ensure that the decision variables for x_{ij} and t_{ij} are either 0 or 1 in value.

The preceding model is a MIP problem and can be solved by using a general-purpose integer-linear optimization package. After we formulate all three models, we compare issues of model size and solvability.

Order^{PRM}

The basis for the Order^{PRM} model is quite different from the Tree model in that an explicit assignment of an area to a region exists. Each region is represented by an index k, where $k = 1, \ldots, p$, as well as by an area to serve as a "root" area. Any area can be chosen as a root, but one and only one root can exist per region. The other areas are assigned to one root according to an ordering system that ensures that the areas assigned to the same region are spatially connected. The contiguity conditions in this model represent an extension of the ordered-area assignment conditions proposed by Cova and Church (2000), who developed such conditions to enforce contiguity in a site design problem. The parameters of this model are

$i, I =$ index and set of areas, $i = \{1, \cdots, n\}$;

$$c_{ij} = \begin{cases} 1, & \text{if areas } i \text{ and } j \text{ share a border, with } i, j \in I \text{ and } i \neq j \\ 0, & \text{otherwise;} \end{cases}$$

$k, K =$ index and set of regions, $k = \{1, \cdots, p\}$;

$o, O =$ index and set of contiguity order, $o = \{0, \cdots, q\}$, with $q = n - p + 1$;

$N_i =$ $\{j | c_{ij} = 1\}$, the set of areas that are adjacent to area i;

$d_{ij} =$ dissimilarity relationships between areas i and j, with i, $j \in I$ and $i < j$.

The decision variables for this model are

$$t_{ij} \quad \begin{cases} 1, \text{ if areas } i \text{ and } j \text{ belong to the same region } k, \text{ with } i < j, \\ 0, \text{ otherwise;} \end{cases}$$

$$x_i^{ko} \quad \begin{cases} 1, \text{ if areas } i \text{ is assigned to region } k \text{ in order } o, \\ 0, \text{ otherwise.} \end{cases}$$

Although all models use the same set of variables, t_{ij}, the linking variables, x_i^{ko}, are now quite different in definition and function from the previously used variables x_{ij}. The variable x_i^{ko} represents the assignment of a given area *i* to region *k* based upon an order *o*. Contiguity is enforced for a region by ensuring that each area is either adjacent to the root area or next to an area that is assigned to the same region with a smaller order number. This model can be formulated as follows:

Minimize

$$Z = \sum_i \sum_{j|j>i} d_{ij} t_{ij}, \tag{13}$$

subject to

$$\sum_{i=1}^{n} x_i^{k0} = 1 \quad \forall k = 1, \cdots, p, \tag{14}$$

$$\sum_{k=1}^{p} \sum_{o=0}^{q} x_i^{ko} = 1 \quad \forall i = 1, \cdots, n, \tag{15}$$

$$x_i^{ko} \leq \sum_{j \in N_i} x_j^{k(o-1)} \quad \forall i = 1, \cdots, n; \forall k = 1, \cdots, p; \forall o = 1, \cdots, q, \tag{16}$$

$$t_{ij} \geq \sum_{o=0}^{q} x_i^{ko} + \sum_{o=0}^{q} x_j^{ko} - 1 \quad \forall i, j = 1, \cdots, n | i < j; \forall k = 1, \cdots, p, \tag{17}$$

$$x_i^{ko} \in \{0, 1\} \quad \forall i = 1, \cdots, n; \forall k = 1, \cdots, p; \forall o = 1, \cdots, q, \tag{18}$$

$$t_{ij} \in \{0, 1\} \quad \forall i, j = 1, \cdots, n | i < j. \tag{19}$$

The objective is exactly the same as in the Tree model, employing the same set of variables. Each region is explicitly referred to by index, *k*. Constraints (14) establish that each region *k* has a defined root area. A root area for a region has a defined order of zero. Essentially, these constraints require that each region *k* has one and only one root, $o = 0$. Constraints (15) require that each area *i* be assigned to exactly one region *k* and one contiguity order *o*. Constraints (16) require that area *i* be assigned to region *k* at order *o* if and only if an area *j* exists, in the adjacent neighborhood of *i*, that is assigned to the same region *k* in order $o - 1$. Altogether, constraints (14), (15), and (16) establish that each region *k* must be contiguous to the assigned root area. The structural constraints (17) force $t_{ij} = 1$ whenever areas i and j are assigned to the same region k, regardless of the order in which they are assigned. Next, constraints (18) restrict the x_i^{ko} variables to be 0-1 integer values. Finally, constraints (19) require that the t_{ij} variables be restricted in a similar manner. Like the Tree model, this also is a MIP, which can be solved by the use of off-the-shelf optimization software.

Flow^{PRM}

The final model was inspired by Shirabe's model for spatial unit allocation (Shirabe 2005), which uses an embedded network flow model in which all areas that are assigned must be connected with a flow route to the designated sink.

Our formulation allows for multiple networks, one per region. Thus, each region has an area that is designated as its sink. A flow network is defined along arcs connecting areas that are adjacent and share a portion of their boundary. If an area is assigned to region k, then this area must supply a unit to the flow that arrives to the sink of region k. A flow cannot be shared by two or more regions. Flow^{PRM} uses the following parameters:

$i, I =$ index and set of areas, $i = \{1, \cdots, n\}$;
$k, K =$ index and set of regions, $k = \{1, \cdots, p\}$;
$N_i =$ $\{j | c_{ij} = 1\}$, the set of areas that are adjacent to area i;
$d_{ij} =$ dissimilarity relationships between areas i and j, with $i, j \in I$ and $i < j$.

The decision variables for this model are as follows:

$$t_{ij} = \begin{cases} 1, & \text{if areas } i \text{ and } j \text{ belong to the same region } k, \text{ with } i < j \\ 0, & \text{otherwise;} \end{cases}$$

f_{ijk} nonnegative contiguous variable indicating the amount of flow from area i to j in region k;

$$y_{ik} = \begin{cases} 1, & \text{if area } i \text{ is included in region } k \\ 0, & \text{otherwise;} \end{cases}$$

$$w_{ik} = \begin{cases} 1, & \text{if area } i \text{ is chosen as a sink} \\ 0, & \text{otherwise.} \end{cases}$$

The Flow^{PRM} model can now be formulated as:
Minimize

$$Z = \sum_i \sum_{j|j>i} d_{ij} t_{ij}, \tag{20}$$

subject to

$$\sum_{k=1}^{p} y_{ik} = 1 \quad \forall i = 1, \cdots, n, \tag{21}$$

$$w_{ik} \leq y_{ik} \quad \forall i = 1, \cdots, n; \forall k = 1, \cdots, p, \tag{22}$$

$$\sum_{i=1}^{n} w_{ik} = 1 \quad \forall k = 1, \cdots, p, \tag{23}$$

$$f_{ijk} \leq y_{ik} \times (n - p) \quad \forall i = 1, \cdots, n; \forall j \in N_i; \forall k = 1, \cdots, p, \tag{24}$$

$$f_{ijk} \leq y_{jk} \times (n - p) \quad \forall i = 1, \cdots, n; \forall j \in N_i; \forall k = 1, \cdots, p, \tag{25}$$

$$\sum_{j \in N_i} f_{ijk} - \sum_{j \in N_i} f_{jik} \geq y_{ik} - (n - p) \times w_{ik} \quad \forall i = 1, \cdots, n; \forall k = 1, \cdots, p, \tag{26}$$

$$t_{ij} \geq y_{ik} + y_{jk} - 1 \quad \forall i, j = 1, \cdots, n | i < j; \forall k = 1, \cdots, p, \tag{27}$$

$$y_{ik} \in \{0, 1\} \quad \forall i = 1, \cdots, n; \forall k = 1, \cdots, p, \tag{28}$$

$$w_{ik} \in \{0, 1\} \quad \forall i = 1, \cdots, n; \forall k = 1, \cdots, p, \tag{29}$$

$$t_{ij} \geq 0 \quad \forall i, j = 1, \cdots, n | i < j, \tag{30}$$

$$f_{ijk} \geq 0 \quad \forall i = 1, \cdots, n; \forall j \in N_i; \forall k = 1, \cdots, p. \tag{31}$$

The objective here is exactly the same as in the first two models. Constraints (21) ensure that each area *i* is assigned to only one region *k*. Constraints (22) restrict the assignment of a sink in region *k* only to those areas that have been assigned to that region. Constraints (23) force each region to contain one and only one sink. Constraints (24) and (25) ensure that a flow can exist between areas *i* and *j* if and only if both areas have been assigned to the same region *k* and are adjacent areas. If area *i* is not a sink, constraints (26) ensure that area *i* must supply at least one unit of flow (net outflow ≥ 1). If area *i* is chosen to be a sink, then these constraints allow a negative net flow of up to $n - p - 1$ because sink areas do not have outflows. The value of $n - p - 1$ is because the largest possible region (in terms of the number of areas assigned to it) cannot be larger than $n - p$. If a sink does not generate a net outflow, then a total flow of no more than $n - p - 1$ units ends at a given regional sink. Constraints (27) force the t_{ij} variables to be one when both areas *i* and *j* are assigned to the same region. Constraints (28) and (29) ensure that the variables w_{ik} and t_{ij} are 0-1 integers in value. Finally, constraints (30) and (31) restrict the remaining decision variables to be nonnegative in value.

A deeper understanding of the three strategies to satisfy the spatial contiguity contraint

Each of the three models presented is based upon specific approaches: (1) defining each region with a subtree, where the subtrees cannot contain cycles, (2) defining each region to have an assigned root area, where contiguity is forced by special "order" conditions, and (3) representing regions by flow nets, where flow originates at each area and must flow between adjacent areas within the same region until it reaches the area designated as a sink. Depicting how contiguity is represented in each model allows for a better understanding of these three models.

Figure 2 shows six different ways to obtain the optimal solution for our example about mean housing prices introduced in the subsection "objective function." Each row reports two solutions per model (tree, order, and flow), and for each solution, we indicate the decision variables with values other than 0.[5] Note that the shape of the regions is the same in all of the solutions. The difference among these solutions is the way in which the decision variables are combined to guarantee the spatial contiguity of each region. These solutions are not the only ones; for each model, multiple ways exist to obtain the optimal solution with a proper recombination of decision variables.

For the Tree^{PRM} model, in Fig. 2a, the region connects areas 1–3, and 6 with a tree formed by links 1–2, 2–3, and 3–6 (i.e., $x_{1,2} = x_{2,3} = x_{3,6} = 1$). In Fig. 2b, the same region connects these areas with links 1–2, 3–2, and 6–3. Thus, the links can be arranged in different ways to construct a region. The decision variable u guarantees that the links do not create any tour that violates feasibility. The integer values of u require that an area i, from which a link x_{ij} is leaving, must have a value u_i less than the value assigned to the destination area j, u_j, to which the link is arriving.[6]

For the Order^{PRM} model, the solution in Fig. 2c selects area 3 as the root area for region 1 (i.e., $w_3^{1,0} = 1$), and area 7 as the root area for region 2 (i.e., $w_7^{2,0} = 1$). Given these root areas, all of the other areas are assigned following an ordering system that depends on the position of each area with respect to its region's root. Thus, in region 1, areas 2 and 6, which are first-order neighbors of the root area, are assigned in order 1 (i.e., $x_2^{1,1} = x_6^{1,1} = 1$), while area 1, which is two contiguity orders from its region's root, is assigned in order 2 (i.e., $x_1^{1,2} = 1$).[7] Figure 2d shows a different way to configure the same regions with different root areas. In conclusion, any area can be selected as the root of its region without affecting optimality.[8]

Finally, for the Flow^{PRM} model, the solution in Fig. 2e shows that region 1 (formed by areas 1, 2, 3, and 6) has its sink located in area 2 (i.e., $w_{2,1} = 1$). All of the other areas in this region are connected to this sink through flow strings represented by the decision variables f_{ijk}. Each area, apart from the sink, contributes a unit of flow; for example, in region 1 in Fig. 2f, a flow string starts in area 1 and arrives to area 2, $f_{1,2,1}$. Because no flow arrives to area 1, the flow $f_{1,2,1}$ accumulates only one unit of flow contributed by area 1 (i.e., $f_{1,2,1} = 1$). Next, a flow leaves from area 2 and arrives to area 3, $f_{2,3,1}$, accumulating the unit of flow coming from area 1, plus the unit of flow contributed by area 2; thus, $f_{2,3,1} = 2$. Finally, a third string connects area 3 to area 6 (the sink area) that deposits three units of flow into the sink.[9] As can be seen, because of the adaptability of the flow strings, any area can be the sink of a region without affecting optimality. Last, the role of variables y_{ik} is to ensure that no string flows leave from one region and arrive to a different region.

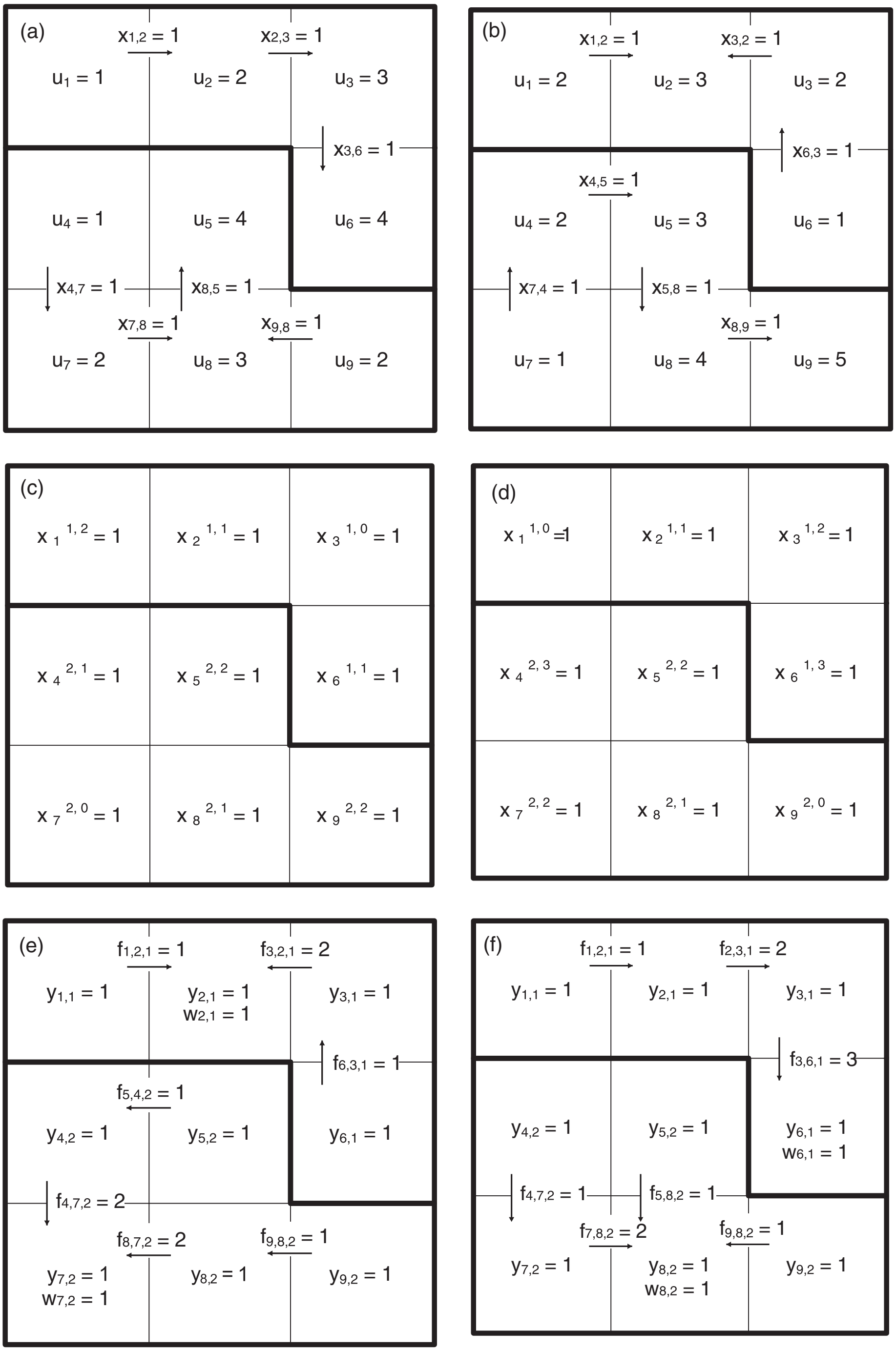

Figure 2. Examples of different ways to obtain the optimal solution (a) Tree^{PRM}: optimal solution 1; (b) Tree^{PRM}: optimal solution 2; (c) Order^{PRM}: optimal solution 1; (d) Order^{PRM}: optimal solution 2; (e) Flow^{PRM}: optimal solution 1; (f) Flow^{PRM}: optimal solution 2.

Table 3 Number of Constraints and Variables per Model

Model	Constraints	Variables
Tree^{PRM}	$1+n^3-n^2+3n+2\sum_{i=1}^{n}\lvert N_i\rvert$	$n^2+\sum_{i=1}^{n}\lvert N_i\rvert$
Order^{PRM}	$np(n-p+1)+n+p(1+\frac{n^2-n}{2})$	$np(n-p+2)+\frac{n^2-n}{2}$
Flow^{PRM}	$p(\frac{n^2-n}{2}+1)+2np+n+2p\sum_{i=1}^{n}\lvert N_i\rvert$	$2np+p\sum_{i=1}^{n}\lvert N_i\rvert+\frac{n^2-n}{2}$

Note: $|N_i|$ is the cardinality of N_i.

In conclusion, the *p*-regions problems have alternate optima, and this property can be potentially useful when searching for strategies for reducing run times.

Comparing model size and solution times

We have formulated three rather different models for the *p*-regions problem. Each model is a MIP, which conceivably can be solved by a general-purpose optimization software. The ease with which a given model can be solved is predicated on a number of factors, including model size (number of variables and constraints) and model structure.

Table 3 presents the functions that can be used to estimate the theoretical number of constraints and variables for each model. Figures 3 and 4 show the graphical representation of those functions for $n=4,\ldots,50$, and $p=2,\ldots,n-1$. Each model behaves differently when increasing the number of areas and/or regions.

Tree^{PRM} is the biggest model in terms of number of constraints. It grows rapidly when increasing the number of areas, mostly because of equation (6), which grows exponentially (n^3-2n^2+n). However, the simplicity of its formulation makes this model the smallest one in terms of the number of variables. An important characteristic is that Tree^{PRM} is not sensitive to changes in the number of regions because none of its variables include an index for the region.

Order^{PRM} is the second biggest model in terms of constraints, and the biggest in terms of variables. This model does not increase linearly with increases in the number of regions. For a given number of areas, Order^{PRM} tends to reach the maximum number of constraints when the proportion of regions/areas is 75.7% (± 1.3%), and the maximum number of variables when the proportion of regions/areas is 52.0% (± 3.4%).

The theoretical number of variables and constraints for Order^{PRM} was estimated for the worst-case scenario, where the index *o* is allowed to reach its theoretical maximum value; that is, $o=0,\ldots,(n-p+1)$. This maximum contiguity order $(n-p+1)$ occurs when $p-1$ regions have only one area, and the $n-p+1$ areas belonging to a single region are connected as a single chain with the variable x_i^{k0} located at one extreme of that chain.

Finally, Flow^{PRM} is the smallest model in terms of the number of constraints, and the second biggest model in terms of the number of variables. Different from

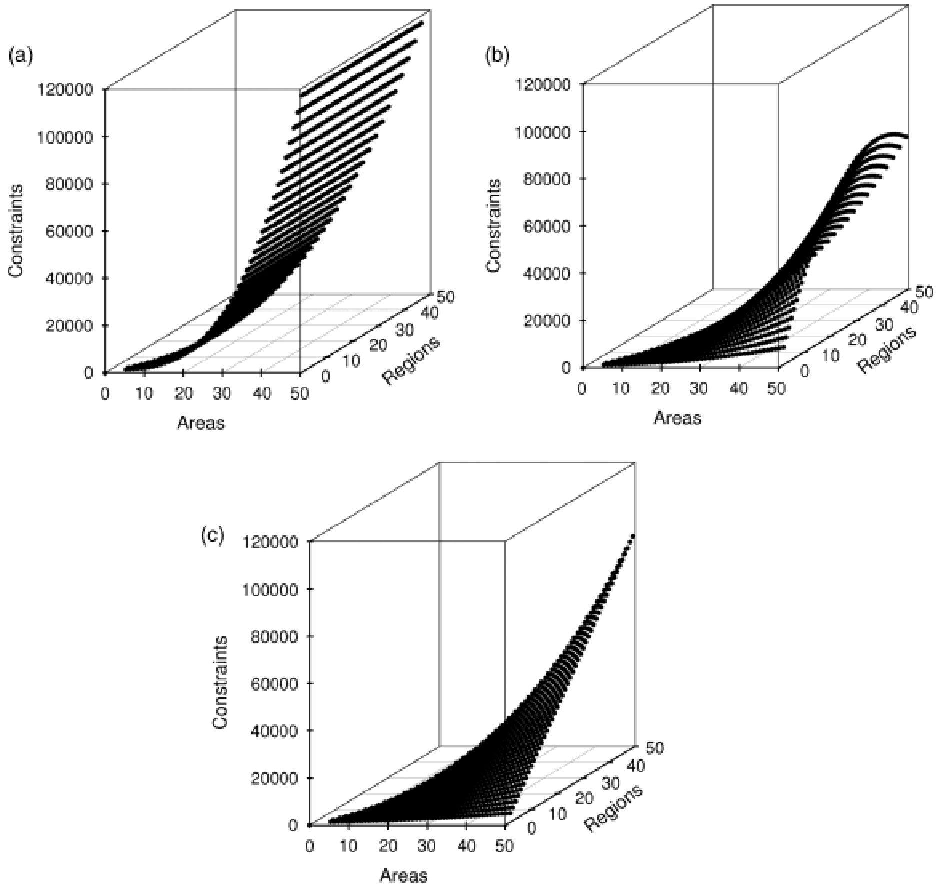

Figure 3. Number of constraints (with $\sum_{i=1}^{n} |N_i| = 5n$) (a) Tree^{PRM}; (b) Order^{PRM}; (c) Flow^{PRM}.

Tree^{PRM} and Order^{PRM}, the size of Flow^{PRM} increases with increases in either the number of areas or the number of regions.

Table 4 summarizes computational results from using ILOG CPLEX 11.2 executed on a Dell Precision T3400 computer running the Windows XP-64 bits operating system equipped with 8 GB RAM and a 2.99 GHz Intel Corel 2 Extreme processor. We solved 10 problems that combine different numbers of areas ($n = 16$, 25, and 49) and different numbers of regions ($p = 3$–7, and 10). The areas are organized as regular lattices with equal numbers of rows and columns (4 × 4, 5 × 5, and 7 × 7).

The aggregation variables (y), from which the dissimilarities d_{ij} are calculated, were simulated as spatial autoregressive (SAR) processes with $\rho = 0.7$.[10] A total of three spatial processes were simulated, one for $n = 16$, another for $n = 25$, and a third for $n = 49$. Thus, for a given value of n, the parameters d_{ij} are the same for the three models.[11]

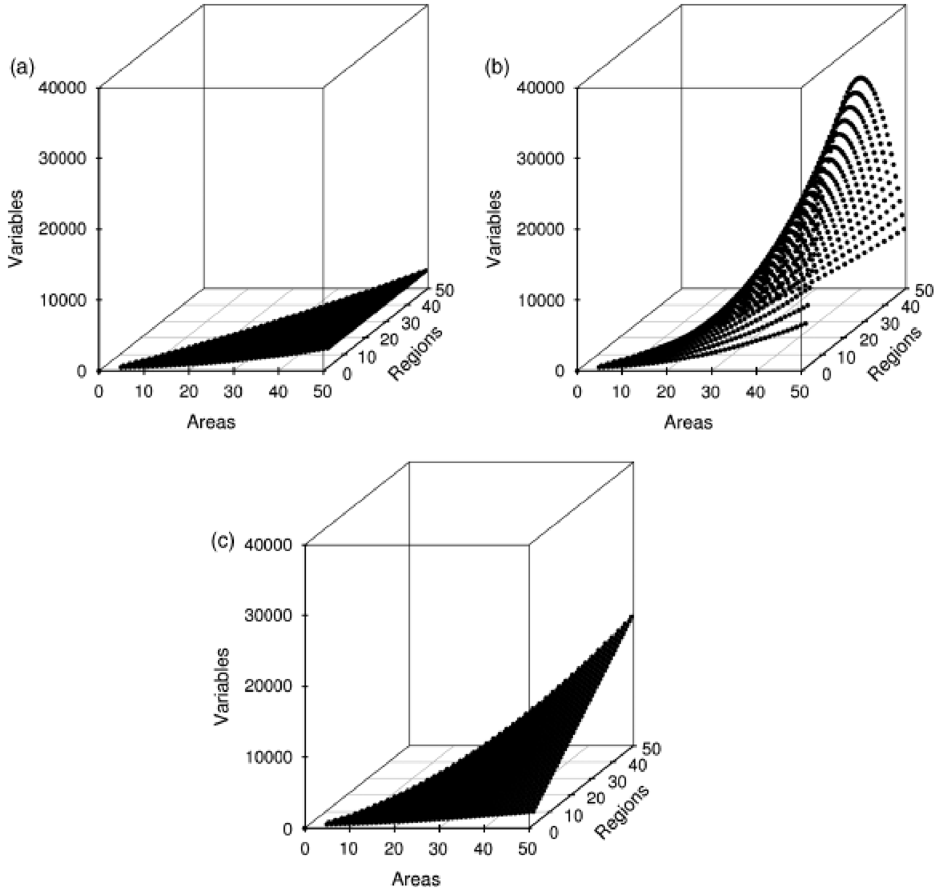

Figure 4. Number of variables (with $\sum_{i=1}^{n} |N_i| = 5n$) (a) Tree^{PRM} ; (b) Order^{PRM}; (c) Flow^{PRM}.

Regarding the parameter N_i, we use a simple rook contiguity criterion where area j is considered as adjacent to area i (i.e., $c_{ij} = 1$) if they share a common edge.

The Tree^{PRM} model optimally solved 40% of the problems, and the remaining 60% were stopped after 3 h. Problem 4 was stopped after 3 h but was optimally solved. The results also show that for a given value of n, the run times decrease when increasing the value of p.

The Order^{PRM} model produces the highest run times and optimally solved 30% of the problems. The run times increase dramatically for small values of p; this is predominantly due to the necessity of a higher range for the index of contiguity order, o.

The Flow^{PRM} model optimally solved 50% of the problems. After 3 h, this model did not find a feasible solution for problems 9 and 10. Compared with the other models, Flow^{PRM} incurs the lowest run times, but, unlike the Tree^{PRM} model, the run times increase with both n and p.

Table 4 Computational Experience with CPLEX

Problem	n	p	Best known	Tree^{PRM}		Order^{PRM}		Flow^{PRM}	
				Objective function	Time (sec)	Objective function	Time (sec)	Objective function	Time (sec)
1	16	3	27.42*	27.42	59.33	27.42	3131.03	27.42	0.84
2	16	4	17.34*	17.34	6.61	17.34	240.77	17.34	3.28
3	16	5	10.99*	10.99	0.61	10.99	324.06	10.99	3.59
4	25	3	59.72*	59.72	†	66.64	†	59.72	296.38
5	25	4	37.52*	38.60	†	39.05	†	37.52	4594.53
6	25	6	19.01*	19.01	1439.73	22.35	†	21.40	†
7	49	3	416.11	461.44	†	1075.54	†	416.35	†
8	49	5	226.32	226.32	†	334.22	†	328.17	†
9	49	7	101.36	101.36	†	237.47	†	—	†
10	49	10	55.32	55.32	†	123.30	†	—	†

*Optimal (by CPLEX). †Run stopped after 3 h.

In conclusion, the Tree^{PRM} and Flow^{PRM} models perform better than the Order^{PRM} model. However, a clear dominant model does not exist. All of the models require considerable computational resources, even when solving small p-regions problems.

Figure 5 presents the best-known solutions for each problem. The three sets of grids ($n = 16$, 25, and 49) are grayscale coded according to their respective values of y such that the lighter the polygon, the lower the value of y. The bold borders in the grids outline the resulting regions. As expected, the regions capture the spatial patterns by aggregating areas with similar values. Note that for a given value of n, the solutions for different values of p are not nested; that is, the solution for p regions cannot be obtained by merging two regions from the solution for $p+1$ regions. Two areas that are together in an optimal solution at a given scale do not

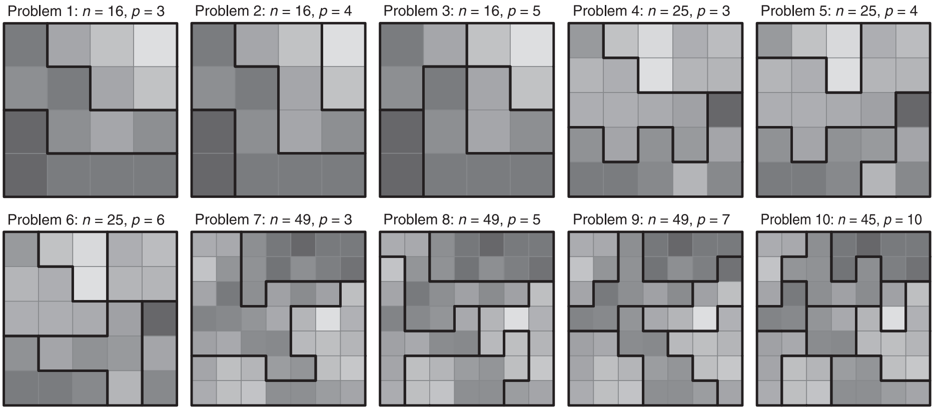
Problem 1: $n = 16$, $p = 3$
Problem 2: $n = 16$, $p = 4$
Problem 3: $n = 16$, $p = 5$
Problem 4: $n = 25$, $p = 3$
Problem 5: $n = 25$, $p = 4$
Problem 6: $n = 25$, $p = 6$
Problem 7: $n = 49$, $p = 3$
Problem 8: $n = 49$, $p = 5$
Problem 9: $n = 49$, $p = 7$
Problem 10: $n = 45$, $p = 10$

Figure 5. Best-known solutions of problems 1–10.

necessarily have to be together in an optimal solution at a different scale (Bunge 1966; Ferligoj and Batagelj 1982). Finally, unlike some algorithms proposed in the literature, our formulations do not force the regions to be compact; rather the shape of the regions are driven by the spatial distribution of the variables. This flexibility makes our models capable of capturing either compact or elongated spatial patterns.

Two methods to reduce the complexity of the *p*-regions models

As we stated in the introduction, the *p*-regions problem is an NP-hard problem, meaning that the time needed to solve a worst-case problem increases substantially as problem size increases. The results of the previous section indicate why researchers have concentrated on the development of heuristics to solve this problem, as even small problems may be difficult and time consuming to solve optimally. Nevertheless, the models proposed here are useful for various reasons. First, they can be used to establish and test the boundary between problem size and the use of heuristics versus the use of an optimal approach. Second, they may be streamlined, resulting in a potential reduction in solution time. In this section we present two strategies that offer the potential to reduce run times. The first technique involves the addition of one or more equality constraints that can be added without affecting optimality, and the second strategy involves solving a problem in a sequential manner.

Initial seeds

Solutions to the Order^{PRM} and Flow^{PRM} models involve explicit assignments to *k* regions. Given an optimal solution to either the Order^{PRM} or the Flow^{PRM} model, simply swapping a group's region number with another group's region number can generate *k*! equivalent optima. The region number indicates which areas belong to the same region, but the value of the solution is not dependent on the specific region number used for a specific region. Thus, there are *k*! ways in which the regional groups can be assigned a region number, yielding *k*! different optimal solutions. This first strategy consists of reducing the set of feasible solutions by establishing a priori one or a few regional seeds or roots. Without loss of generality, we can arbitrarily assign area 1 to region 1 and declare it a root or sink of the region. Because each area must be assigned to a region, and because the Order^{PRM} and Flow^{PRM} models do not distinguish any order between regional designations, we can assign any one area to the first region. This type of constraint is known as an *initial seed*. Initial seeds can be used in the Order^{PRM} and Flow^{PRM} (but not the Tree^{PRM}) model as follows:

1. Order^{PRM}: the seeds require that a specific area *i* be the root (i.e., order *o* is equal to 0) of a given region *k*. For example, we can add an additional constraint to the original formulation that sets area 1 to be the root of region 1:

$$x_1^{1,0} = 1. \tag{32}$$

2. Flow^{PRM}: this formulation has two ways to set the initial seed. The first one requires that a given area i be included in a given region k. The second consists of assigning area i to be the sink of region k. Thus, for area 1 and region 1, the constraints are as follows:

$$y_{1,1} = 1, \tag{33}$$

$$w_{1,1} = 1. \tag{34}$$

More than one seed can be added, depending on our knowledge of a problem. For example, we can assign two areas to different regions because they are located too far from each other, and they are not very similar, implying that the probability of an optimal solution assigning both areas to the same region essentially is 0.

Recursive cycle-breaking constraints

The second strategy for reducing computational time consists of solving the p-regions problem in an iterative fashion, where a relaxed model is solved first. The relaxed model does not contain all of the constraints needed to ensure contiguity. For instance, we can relax the Tree^{PRM} model by removing the constraints that eliminate cycles; that is, the modified MTZ constraints. This model is likely to take a small amount of computation time to solve but may not necessarily result in a feasible solution to the p-regions problem as one or more regions may not be contiguous. Rather than include all of the constraints necessary to ensure contiguity within a region, we can add constraints only when necessary. Regions in the Tree^{PRM} model are not contiguous when a cycle exists in one of the subtrees. We can prevent a specific cycle from existing in a region's subtree using the following constraints:

$$\sum_{i} \sum_{j \in N_i} x_{ij} \leq |\Gamma| - 1 \quad \forall i, j \in \Gamma, \tag{35}$$

where Γ is the set of areas involved in the cycle, and $|\Gamma|$ is the cardinality of Γ.

If we add these constraints to the relaxed problem and then resolve it, we identify a solution that groups areas into regions, and no regional subtree contains the cycle associated with Γ. If other cycles exist, we can then identify them and add cycle-breaking constraints to prevent them as well. We can continue this process of adding cycle-busting constraints and resolving the relaxed problem until no cycles exist in any subtrees. This final solution is optimal and presents a feasible solution to the p-regions problem. Pseudocode 1 describes the process for applying this strategy.

Pseudocode 1: Recursive cycle breaking (I, K, c_{ij}, N_i, d_{ij})

```
comment: Reach feasibility by adding cycle-breaking constraints
1: Solve Tree$^{PRM}$ without MTZ constraints (9) and (10)
2: Check for cycles that means the solution is infeasible
```

```
3: **if** infeasible solution
        then { add cycle − breaking constraints based on equation (35)
             { GOTO 1
        else STOP

return (optimal solution)
```

Table 5 show the results of solving the 10 problems introduced in the section "Comparing model size and solution times" when utilizing the strategies for reducing the complexity of the three models. As can be seen, the run times are significantly reduced.

The recursive cycle-breaking strategy proposed for the Tree^{PRM} model shows significant reductions in run times compared with the complete version of the models. The percentage of optimally solved problems is now 50%. This strategy performs better when the ratio n/p is small. In such a case, the number of areas per region tends to be small, and therefore the number of potential cycles also is small. The number of cycle-breaking constraints confirms this tendency.

An initial seed in the Order^{PRM} model increases the percentage of optimally solved problems from 30% to 50%, with a significant decrease in run times. Also, for those problems that were stopped after 3 h, the objective function value shows an improvement compared with the solutions obtained with the complete models.

The Flow^{PRM} model also shows improvement when using an initial seed. In this case, the percentage of optimally solved problems increases from 50% to 60%, and problems 9 and 10, for which the complete problem was not able to find an initial solution after 3 h, now have feasible solutions.

The results of the Order^{PRM} and Flow^{PRM} models show that the time taken to converge to optimality is a function of the number of alternative optima and that eliminating the existence of alternative optima with seed constraints reduces computational time substantially.

Conclusions

This article introduces three MIP model formulations for the p-regions problem, a generic name for any model that aggregates n small areas into p spatially contiguous regions. Each of our three models has identical objective functions but uses one of three different techniques to ensure that all areas within a region are spatially contiguous. The assurance of spatial contiguity is what makes the p-regions model difficult to solve using exact methods. However, increased computational resources and a wider availability of parallelized computing, ever-improving optimal solvers, and methods for reducing complexity allow increasingly larger p-regions problems to be solved optimally.

Table 5 Computational Experience with CPLEX Using Methods to Reduce the Complexity of the Models

Problem	n	p	Best known	Tree^{PRM} recursive			Order^{PRM} seed $x_1^{1,1} = 1$		Flow^{PRM} seed $w_{1,1} = 1$	
				Objective function	Time (s)	CBC	Objective function	Time (s)	Objective function	Time (s)
1	16	3	27.42*	27.42	5.78	20	27.42	3.20	27.42	0.78
2	16	4	17.34*	17.34	1.36	11	17.34	8.94	17.34	1.27
3	16	5	10.99*	10.99	0.36	3	10.99	25.81	10.99	2.03
4	25	3	59.72*	—	†	149	59.72	1796.72	59.72	71.89
5	25	4	37.52*	37.52	926.75	75	37.52	3065.75	37.52	598.30
6	25	6	19.01*	19.01	42.34	21	25.15	†	19.01	5697.75
7	49	3	416.11	—	†	17	416.11	†	420.09	†
8	49	5	226.32	—	†	28	236.94	†	391.15	†
9	49	7	101.36	—	†	51	129.57	†	141.85	†
10	49	10	55.32	55.32	†	45	68.50	†	58.39	†

CBC number of cycle-breaking constraints (see Pseudocode 1).*Optimal (by CPLEX).†Run stopped after 3 h.

Wide availability of disaggregated, georeferenced data give researchers the ability to customize their area of research, including varying the level of data aggregation and zoning scheme. This customization allows a focus on the appropriate scale for each study and potentially minimizes the effect of the modifiable areal unit problem. However, little research has been devoted to models and formal descriptions of how data should be aggregated to maximize the appropriateness of data aggregation and zoning. We believe that our three MIP *p*-regions models make a significant contribution in this respect.

The three MIP models retain a significant advantage over other *p*-regions problem methods in the sense that they do not make any assumptions about the shape of regions. Our approach can generate both compact and elongated regions, capturing the underlying spatial patterns within data. Other models typically rely on approximate measures for defining regions, such as generating regions based on minimizing the geographic distance to the centroid or minimizing a region's perimeter. Forcing compactness, for example, can be critically detrimental in many contexts (Duque, Ramos, and Surinach 2007).

Regarding the performance of the models, we found that Tree^{PRM} works better when the ratio n/p is small, whereas Order^{PRM} and Flow^{PRM} are recommended for those cases where the ratio n/p is large. Also, when a restriction on running time exists, the complete Tree^{PRM} model appears as the best option to obtain a good feasible solution.

Although during computational experiments, a substantial number of problems were stopped after their code executed for 3 h, in real applications, most of the *p*-regions solutions are meant to last for a long time (e.g., school districts, electoral districts, census output areas, among others). This longevity for solutions makes decision makers more willing to invest days, and even weeks, to find a good solution.

Further research can identify techniques to reduce the complexity and number of constraints to guarantee spatial contiguity in spatial MIP models. In general, examination of the spatial distribution of the input data could be used as a guide to reduce the problem size. Additionally, heuristics can be applied to a problem to minimize the number of variables and constraints required to represent a MIP model. Also, for the Tree^{PRM} model, a priori known cycles that lead to an objective value lower than the optimal feasible solution could be used to reduce the number of times a model needs to be solved in the recursive cycle-breaking procedure.

Acknowledgement

We would like to thank the editor and the three anonymous reviewers for their insightful and helpful comments during the review process.

Notes

[1] To illustrate the significant improvements in run times, Bixby (2002) solved different instances of a linear problem known as the patient distribution system (Carolan et al. 1990). The biggest instance of this model that was possible to be solved in 1988, the

pds70, has 114,944 rows, 422,356 columns, and 929,346 nonzeros. Taking CPLEX as a measure, CPLEX 1.0 (released in 1988) solved this model in 335,292.1 s, while CPLEX 7.1 (released in 2002) dual solved this model in 187.8 s on the same machine, or 1695.1 times faster.

[2] See Batagelj and Bren (1993) for more properties of distance and dissimilarity measures.

[3] See Fischer (1980) for other regional homogeneity measures.

[4] d_{ij} can be extended to include more than one variable (e.g., mean housing price and mean housing age), or more than one time period for a variable (e.g., gross domestic product for year $t, t+1, \ldots$).

[5] The decision variable t_{ij} takes the same values in the six solutions: $t_{1,2} = t_{1,3} = t_{1,6} = t_{2,3} = t_{2,6} = t_{3,6} = t_{4,5} = t_{4,7} = t_{4,8} = t_{4,9} = t_{5,7} = t_{5,8} = t_{5,9} = t_{7,8} = t_{7,9} = t_{8,9} = 1$. Therefore, the objective function value is the same: 1222.8.

[6] This condition would not work without the additional constraint requiring that each area cannot have more than one leaving node.

[7] In our examples, we use a rook criterion of contiguity where adjacent areas are neighbors if they share a common edge. This spatial contiguity relationship is captured by the parameter c_{ij} that appears in our three formulations.

[8] Note that this property can be further analyzed with the aim of finding a way to reduce the maximum value, $q = n - p - 1$, that the index o can take in our formulation. A reduction in this value can drastically reduce run times.

[9] Depending on the location of a sink, the region can have several flow strings arriving to the sink (this is the case of region 1 in the solution on the left) or a unique string that flows through all of the areas in a region before arriving to the sink (this is the case of region 1 in Fig. 2e).

[10] Simulating the values of y as SAR processes with $\rho = 0.7$ ensures that y shows spatial patterns;that is, areas with high (low) values tend to be surrounded by areas with high (low) values. Those patterns should be captured by our models because they seek to aggregate into a region those areas with similar values in order to minimize the heterogeneity within each region, $H(C_k)$.

[11] We used the Euclidean distance to calculate the d_{ij} values.

References

Batagelj, V., and M. Bren. (1993). ''Comparing Resemblance Measures.'' *Journal of Classification* 12, 73–93.

Bixby, R. E. (2002). ''Solving Real-World Linear Programs: A Decade and More of Progress.'' *Operations Research* 50, 3–15.

Bixby, R. E., M. Fenelon, Z. Gu, E. Rothberg, and R. Wunderling. (2000). ''MIP: Theory and Practice Closing the Gap.'' Working Paper. Available at http://citeseerx.ist.psu.edu/viewdoc/versions?doi=10.1.1.24.8303 (16 September 2007).

Browdy, M. H. (1990). ''Simulated Annealing: An Improved Computer Model for Political Redistricting.'' *Yale Law and Policy Review* 8, 163–79.

Bunge, W. (1966). ''Gerrymandering, Geography, and Grouping.'' *Geographical Review* 56, 256–63.

Carolan, W. J., J. E. Hill, J. L. Kennington, S. Niemi, and S. Wichmann. (1990). ''An Empirical Evaluation of the KORBX Algorithms for Military Airlift Applications.'' *Operations Research* 38, 240–8.

Cliff, A. D., P. Haggett, J. K. Ord, K. A. Bassett, and R. B. Davies, eds. (1975). *Elements of Spatial Structure: A Quantitative Approach*. New York: Cambridge University Press.

Cova, T. J., and R. L. Church. (2000). ''Contiguity Constraints for Single-Region Site Search Problems.'' *Geographical Analysis* 32, 306–29.

Duque, J. C. (2004). ''Design of Homogeneous Territorial Units: A Methodological Proposal and Applications.'' Ph.D. Dissertation, University of Barcelona.

Duque, J. C., R. Ramos, and J. Surinach. (2007). ''Supervised Regionalization Methods: A Survey.'' *International Regional Science Review* 30, 195–220.

Ferligoj, A., and V. Batagelj. (1982). ''Clustering with Relational Constra-Int.'' *Psychometrika* 47, 413–26.

Fischer, M. M. (1980). ''Regional Taxonomy: A Comparison of Some Hierarchic and Non-Hierarchic Strategies.'' *Regional Science and Urban Economics* 10, 503–37.

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

Gordon, A. D. (1996). ''A Survey of Constrained Classification.'' *Computational Statistics and Data Analysis* 21, 17–29.

Gordon, A. D. (1999). *Classification*, 2nd ed. London: Chapman & Hall/CRC.

Horn, M. E. T. (1995). ''Solution Techniques for Large Regional Partitioning Problems.'' *Geographical Analysis* 27, 230–48.

Keane, M. (1975). ''The Size of Region-Building Problem.'' *Environment and Planning A* 7, 575–7.

Lankford, P. (1969). ''Regionalization: Theory and Alternative Algorithms.'' *Geographical Analysis* 1, 196–212.

Macmillan, W., and T. Pierce. (1994). ''Optimization Modelling in a GIS Framework: The Problem of Political Redistricting.'' In *Spatial Analysis and GIS*, 221–46, edited by A. Fotheringham and T. Pierce. London: Taylor & Francis.

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

Miller, C. E., A. W. Tucker, and R. A. Zemlin. (1960). ''Integer Programming Formulations and Traveling Salesman Problems.'' *Journal of the ACM* 7, 326–9.

Murtagh, F. (1985). ''A Survey of Algorithms for Contiguity-Constrained Clustering and Related Problems.'' *The Computer Journal* 28, 82–8.

Openshaw, S. (1988). ''Building an Automated Modelling System to Explore a Universe of Spatial Interaction Models.'' *Geographical Analysis* 20, 31–46.

Openshaw, S., and L. Rao. (1995). ''Algorithms for Reengineering 1991 Census Geography.'' *Environment and Planning A* 27, 425–46.

Shirabe, T. (2005). ''A Model of Contiguity for Spatial Unit Allocation.'' *Geographical Analysis* 37, 2–16.

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