The Max-p-Regions Problem

Juan C. Duque[1]
Research in Spatial Economics (RiSE).
Department of Economics, EAFIT University.
jduquec1@eafit.edu.co

Luc Anselin
GeoDa Center for Geospatial Anaysis and Computation
School of Geographical Sciences and Urban Planning.
Arizona State University.
luc.anselin@asu.edu

Sergio J. Rey
GeoDa Center for Geospatial Anaysis and Computation
School of Geographical Sciences and Urban Planning.
srey@asu.edu

October 26, 2010

[1]The authors thank the anonymous JRS reviewers and Boris Dev for their insightful and helpful comments during the review process. The usual disclaimer applies.

Abstract

In this paper, we introduce a new spatially constrained clustering problem called the max-p-regions problem. It involves the clustering of a set of geographic areas into the maximum number of homogeneous regions such that the value of a spatially extensive regional attribute is above a predefined threshold value. We formulate the max-p-regions problem as a mixed integer programming (MIP) problem, and propose a heuristic solution.

1 Introduction

According to Fischer (1980), a homogeneous region consist of a set of spatially contiguous areas which show a high degree of similarity regarding a set of attributes; e.g., degree of diversity, per capita income, level of quality of life, etc. This type of region is different from a functional region in the sense that the latter consists of spatially contiguous areas with a high degree of interdependence; e.g., high levels of commuting flows or commercial trade between them.[1]

The problem of aggregating areas into homogeneous regions is referred to by a host of different names, including region-building (Byfuglien and Nordgard, 1973), conditional clustering (Lefkovitch, 1980), clustering with relational constraints (Ferligoj and Batagelj, 1982), constrained clustering (Legendre, 1987), contiguity constrained clustering (Murtagh, 1992), regional clustering (Maravalle and Simeone, 1995), contiguity constrained classification (Gordon, 1996), regionalization (Wise et al., 1997), or clustering under connectivity constraints (Hansen et al., 2003).[2] The literature on this topic focuses on particular aspects of the problem such as strategies to ensure spatial contiguity of each region, ways to measure homogeneity, strategies to explore the solution space efficiently, and ways to check for solution feasibility.

From this basic problem (i.e., to aggregate areas into homogeneous regions) other sub-branches have emerged, which add new constraints with the aim to provide solutions to specific requirements in empirical applications. The most important constraints are: (a) shape of the regions (e.g., compactness, similarity to existing solutions); (b) equality of an attribute values across the regions (e.g., population equality); and (c) membership constraints (e.g., boundary integrity[3]). Each one of these additional constraints has generated a number of contributions suggesting different formulations and solution strategies.

Although models for solving either the problem of basic homogeneous regions or the extended versions of this problem have been under development for the past four decades, the dramatic increase in the availability of

[1] Semple and Green (1984) refers to these two types of regions as uniform and functional regions.

[2] For literature reviews on constrained clustering, see Murtagh (1985), Gordon (1996) and Duque et al. (2007). See also Legendre (1987) for a discussion about why constrained clustering is appropriate and necessary.

[3] This topic includes constraints that avoid solutions with regions being split by natural or artificial barriers. It also includes constraints that force a subset of areas to be assigned to the same homogeneous region.

highly disaggregated spatial data and computational resources provides the opportunity for regional scientists to explore new applications of spatial aggregation models. In this process, new challenges appear that need to be addressed with new formulations. One of those challenges is related to the definition of the number of homogeneous regions to be designed (the scale problem); many practitioners know that they need to aggregate areas into homogeneous regions but they do not know how many regions they should create.

While there is a wide range of methods for finding an appropriate level of aggregation,[4] choosing among these methods is complicated by a number of factors: (a) the performance of those methods is data dependent; (b) the choice of the number of regions is complicated due to a wide variety of methods available ; and, (c) the correct selection of the method requires a deep knowledge of the properties of each one of the available options. This situations has created a "barrier" for the use of the available spatial clustering techniques in practice.

Our experience with spatial aggregation models has shown us that in many empirical applications the researcher does not want to use spatial clustering as a tool for summarizing information or finding the real number of clusters in the data, but as a tool for designing suitable regions for analysis. In this scenario, although the researcher does not know how many regions (clusters) need to be designed, she may know a condition that must be satisfied by every region in order to make them suitable for the analysis. That information can then be used as a way to endogenize the number of regions.

This paper introduces the exact formulation and a solution method for a new type of spatially constrained clustering that we coined as the max-p-regions problem. In brief, the max-p-regions involves the aggregation of n areas into an unknown maximum number of homogeneous regions, while ensuring that each region satisfies a minimum threshold value imposed on a predefined spatially extensive attribute (e.g., number of households per region, area per region, population per region, etc.).

A unique feature of this model is that the number of regions is modeled as an endogenous parameter. Another important characteristic of this formulation is that, opposite to many existing approaches, the way the model satisfies the spatial contiguity constraint does not rely on imposing con-

[4]Milligan and Cooper (1985) evaluate 30 procedures for determining the number of clusters. The authors refer to this decision as "the dilemma of selecting the number of clusters". See also Gordon (1999) for a discussion on this topic.

straints on the shape of the regions (i.e., maximal compactness); instead, the max-*p*-regions model lets data dictate the shape of each region, which is a desirable characteristic in many empirical applications in regional science.

One of the most promising uses of the max-*p*-regions model is the definition of study regions. For example, in the statistical analysis of rates for small area estimation (i.e., crime rates, disease rates, unemployment rates) the precision with which the underlying rate can be measured is inversely related to the size of the population within the enumeration district. It is often desirable to combine small contiguous units so as to increase the precision of the rate estimation. In these cases, the max-*p*-regions algorithm can be used to design new study regions where (a) the loss of observations is minimized because it seeks to perform the minimum number of spatial aggregation; (b) the degree of aggregation bias is minimized, because intraregional homogeneity is maximized; and, (c) the new regions ensure valid statistical inference. It is also important to note that the max-*p*-regions model could be used as a way to avoid subjectivity in the definition of both scale (number of regions) and aggregation (shape of the regions) in applied analysis.

The remainder of the paper is organized as follows. A formal statement of the max-*p*-regions problem is formulated in the next section. A literature review is presented in Section 3. The exact formulation of the max-*p*-regions problem is introduced in Section 4. The heuristic algorithm for solving the max-*p*-regions problem, including some computational experience, is presented in Section 5. The article concludes with a summary and recommendations for future work.

2 Problem statement

Areas:

Let $A = \{A_1, A_2, ..., A_n\}$ denote a set $n = |A|$ areas.

Attributes:

Let A_{iy} denote the attribute y of area A_i, where $y \in Y = \{1, 2, ..., m\}$ with $m \geq 1$; and l_i denote a spatially extensive attribute of area A_i.

Relationship:

Let $d : A \times A \rightarrow \mathbb{R}^+ \cup \{0\}$ be the dissimilarity between areas based on the set of attributes Y such that $d_{ij} \equiv d(A_i, A_j)$ satisfies the conditions $d_{ij} \geq 0, d_{ij} = d_{ij}$ and $d_{ij} = 0$ for $i, j = 1, 2, \ldots, n$. Distance funtions can also be utilized; i.e., d_{ij} can also satify the subadditivity, or triangle inequality, condition: $d_{ij} \leq d_{ik} + d_{kj}$ for $i, j, k = 1, 2, \ldots, n$.

Let $W = (V, E)$ denote the contiguity graph associated with A such that vertices $v_i \in V$ correspond to areas $A_i \in A$ and edges $\{v_i, v_j\} \in E$ if and only if areas A_i and A_j share a common border. For the max-p-regions model W must be a connected graph.

Feasible Partitions of A:

Let $P_p = \{R_1, R_2, ..., R_p\}$ denote a partition of areas A into p regions with $1 \leq p \leq n$ such that:

$$
\begin{array}{l}
|R_k| > 0 \quad \text{for} \;\; k = 1, 2, ..., p; \\
R_k \cap R_{k'} = \emptyset \quad \text{for} \;\; k, k^{'} = 1, 2, ..., p \;\wedge k \neq k^{'}; \\
\bigcup_{k=1}^{p} R_k = A; \\
\sum\limits_{A_i \in R_k} l_i \geq \; threshold \quad \left\{ \begin{array}{l} \text{for} \;\; k = 1, 2, ..., p, \text{ and} \\ threshold \in \mathbb{R}^+ \cup \{0\} | 0 \leq threshold \leq \sum_{A_i \in A} l_i; \end{array} \right. \\
W(R_k) \text{ is connected} \quad \text{for} \;\; k = 1, 2, ..., p.
\end{array}
$$

Let Π denote the set of all feasible partitions of A.

Evaluation criterion for a feasible partition $P_p \in \Pi$:

$$
\begin{array}{lll}
h(R_k) & = \sum\limits_{ij: A_i, A_j \in R_k, i \leq j} d_{ij} & \text{Heterogeneity of region } k \text{ with } R_k \in P_p; \\
H(P_p) & = \sum\limits_{k=1}^{p} h(R_k) & \text{Total heterogeneity of partition } P_p \in \Pi.
\end{array}
$$

The max-p-regions problem may be formulated as:

Determine $P_p^* \in \Pi$ such that $|P_p^*| = max(|P_p| : P_p \in \Pi)$, and
$\nexists P_p \in \Pi : \; |P_p| = |P_p^*| \wedge H(P_p) < H(P_p^*)$

Next we present a basic example to illustrate an optimal solution for the max-p-regions problem. Figure 1 shows a regular lattice with nine square

areas which are grayscale-coded according to y, say the average price of a house in an area. We also have the number of houses per area as our spatially extensive attribute l. The objective is (1) to find the maximum number of contiguous regions, p, needed to group the nine areas in such a way that each region contains at least 120 houses (i.e., $threshold = 120$); and (2) to find, within all solutions with p regions, the solution with the least amount of regional heterogeneity based on y.[5]

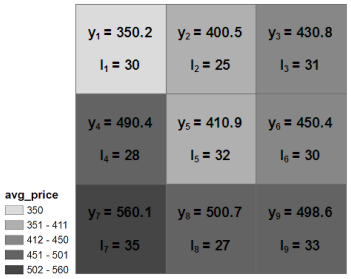

Figure 1: Example of input data: y = average price, and l = number of houses.

Table 1: Construction of the evaluation criterion $H\left(P_2^*\right)$

Expressions	Values
$h\left(R_1 = \{A_1, A_2, A_3, A_5, A_6\}\right)$	$d_{1,2}+d_{1,3}+d_{1,5}+d_{1,6}+d_{2,3}+d_{2,5}+d_{2,6}+d_{3,5}+d_{3,6}+d_{5,6} =$ $50.3+80.6+60.7+100.2+30.3+10.4+49.9+19.9+19.6+39.5 = 461.4$
$h\left(R_2 = \{A_4, A_7, A_8, A_9\}\right)$	$d_{4,7}+d_{4,8}+d_{4,9}+d_{7,8}+d_{7,9}+d_{8,9} =$ $69.7+10.3+8.2+59.4+61.5+2.1 = 211.2$
$H\left(P_2^*\right) = h(R_1)+h(R_2)$	$461.4+211.2 = 672.6$

Table 1 presents the components of the evaluation criterion for the optimal partition, $H(P_p^*)$. According to the definition of the max-p-regions problem, this optimal solution (P_p^*) implies the following in sequential order.

1. It is not possible to have more than two regions with at least 120 houses each.

2. There is not another feasible solution with two regions with a total regional heterogeneity, $H(P_p)$, lower than 672.6.

[5]For this example we assume $d_{ij} = |y_i - y_j|$; e.g., $d_{1,2} = |350.2 - 400.5| = 50.3$.

The bold borders in Figure 2 outline the resulting regions. The regions capture the spatial patterns by aggregating areas with similar values for variable y. Finally, both regions have more than 120 houses each: 148 houses in region R_1 and 123 in region R_2.

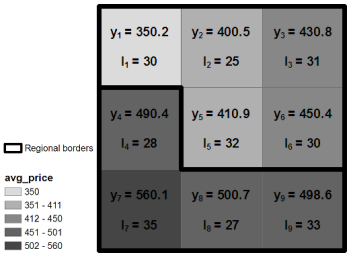

Figure 2: Optimal solution for a threshold of 120 houses per region.

3 Literature review

In the literature, there are three types methods for designing homogeneous regions. The first type of method designs the regions in two stages (Openshaw, 1973; Fischer, 1980). The first of the two stages starts by applying a conventional clustering algorithm to the areas without taking into account the geographical location of the areas being aggregated. In this stage, the focus is placed on creating clusters, not regions, of areas that are homogeneous in terms of a set of attributes, regardless of geography. The second of the two stages defines regions as subsets of spatially contiguous areas assigned to the same cluster. With this method the number of resulting regions heavily depends on the spatial patterns of the attributes used for calculating intraregional homogeneity (Openshaw and Rao, 1995).

The second type of method consists of constructing homogeneous regions by including the x and y coordinates of the centroids of the areas as two additional attributes in a conventional clustering algorithm (Webster and Burrough, 1972; Murray and Shyy, 2000). This is an indirect way to force geographically nearby areas to be assigned to the same cluster. In this case, the resulting regions will tend to be geographically compact and therefore spatially contiguous. Spatial contiguity in the final regional solution depends on the weight given to the geographical attributes (x and y coordinates) compared to the weights given to the other attributes (Wise et al., 1997). An increase in the weight of the geographic coordinate at-

tributes in the clustering procedure will increase the chances of obtaining spatially contiguous regions; As a trade-off, this increase in the geographic distance weighting compared to the weighting of the other attributes will detract from meeting the objective of obtaining intraregional homogeneity for the other attributes. One of the main challenges when applying this strategy it to decide how geographical and non-geographical attributes will be combined and weighted (Webster and Burrough, 1972; Cliff et al., 1975; Perruchet, 1983).[6]

For this paper, the key problem with the first two types of methods is that they do not include a procedure for ensuring the spatial contiguity of the regions. In both cases, this condition must be revised a posteriori. Because of the simplicity of their formulations, a key strength of these types of methods lies in their ability to handle large numbers of areas.[7]

A third type of method for clustering areas, our focus, guarantees spatial contiguity amongst the areas of each resulting region by explicitly including a spatial constraint within the regionalization procedure. The advantage of this strategy is that the objectives of spatial contiguity and intraregional homogeneity do not compete. Information about the neighboring structure of the set of areas is used only as an input for limiting the number of feasible solutions, and within this limited number of spatially contiguous solutions is intraregional homogeneity assessed. There is a wide range of strategies for guaranteeing spatial contiguity based on information about the neighbouring structure. They can be classified into five categories: (a) Adapted hierarchical clustering algorithms are where two clusters are merged only if they share a common border (Lankford, 1969; Byfuglien and Nordgard, 1973; Margules et al., 1985); (b) Seeded regions are where each region starts growing from an initial area from which other neighbouring areas are attached (Openshaw, 1977a); (c) Modification of an initial solution works by moving areas between regions while preserving spatial contiguity (Openshaw and Rao, 1995; Ferligoj and Batagelj, 1982); (d) Graph theory-based algorithms are where the areas and their neighborhood structure are represented as a connected graph that needs to be broken into connected subgraphs, while maximizing some intraregional homogeneity criterion (Maravalle et al., 1997; Hansen et al., 2003; Assunção et al., 2006); and, (e) Formulation of exact optimiza-

[6]Horn (1995) and Martin et al. (2001) point out that the final solution is also sensitive to the methodology applied to define the centroids of the areas.

[7]The number of areas to aggregate determines computational time cost and is an important factor when selecting the aggregation method. See Cliff and Hagget (1970), Cliff et al. (1975), and Keane (1975) for a discussion on the complexity of spatially constrained clustering.

tion models are where a subset of constraints are responsible for satisfying the spatial contiguity of each region (Murray and Shyy, 2000; Duque et al., 2010).[8]

The use of one method or another is not an arbitrary decision. For those problems where the shape of the regions should be guided by the spatial distribution of the variables, the use of conventional clustering with x and y coordinates are not appropriate because they always tend to generate circular (compact) regions. Also, problems that do not require nested aggregations at different scales will not ensure optimality by using adapted hierarchical clustering algorithms because with these methods the solution at one scale is conditioned to the solutions obtained at lower scales (Bunge, 1966). The method proposed in this paper satisfies the contiguity constraint in two ways. First, in the exact formulation we design constraints that borrow concepts from graph partitioning. And second, for the solution method, we design an algorithm that constructs feasible solutions, based on the seeded regions strategies, which are iteratively modified while searching for improvements on the evaluation criterion.

4 The exact formulation of the max-p-regions model

Parameters:

$i, I =$	Index and set of areas, $I = \{1, \cdots, n\}$;
$k =$	index of **potential** regions, $k = \{1, \cdots, n\}$;
$c =$	index of contiguity order, $c = \{0, \cdots, q\}$, with $q = (n-1)$;
$w_{ij} =$	$\begin{cases} 1, \text{ if areas } i \text{ and } j \text{ share a border, with } i, j \in I \text{ and } i \neq j \\ 0, \text{ otherwise;} \end{cases}$
$N_i =$	$\{j \vert w_{ij} = 1\}$, 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 $i < j$;
$h =$	$1 + \lfloor log(\sum_i \sum_{j \vert j>i} d_{ij}) \rfloor$, which is the number of digits of the floor function of $\sum_i \sum_{j \vert j>i} d_{ij}$, with $i, j \in I$;
$l_i =$	spatially extensive attribute value of area i, with $i \in I$;
$threshold =$	minimum value for attribute l at regional scale.

Decision variables:

[8] These methods for ensuring spatial contiguity are required in a wide range of related problems like political districting (Williams, 1995), school districting (Caro et al., 2004), sales districting (Zoltners and Sinha, 1983), among others.

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

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

Minimize:

$$Z = \left(-\sum_{k=1}^{n} \sum_{i=1}^{n} x_i^{k0} \right) * 10^h + \sum_{i} \sum_{j|j>i} d_{ij} t_{ij} \tag{1}$$

Subject to:

$$\sum_{i=1}^{n} x_i^{k0} \leq 1 \qquad \forall k = 1, \cdots, n \tag{2}$$

$$\sum_{k=1}^{n} \sum_{c=0}^{q} x_i^{kc} = 1 \qquad \forall i = 1, \cdots, n \tag{3}$$

$$x_i^{kc} \leq \sum_{j \in N_i} x_j^{k(c-1)} \qquad \forall i = 1, \cdots, n; \forall k = 1, \cdots, n; \forall c = 1, \cdots, q \tag{4}$$

$$\sum_{i=1}^{n} \sum_{c=0}^{q} x_i^{kc} l_i \geq threshold * \sum_{i=1}^{n} x_i^{k0} \qquad \forall k = 1, \cdots, n \tag{5}$$

$$t_{ij} \geq \sum_{c=0}^{q} x_i^{kc} + \sum_{c=0}^{q} x_j^{kc} - 1 \qquad \forall i, j = 1, \cdots, n | i < j; \forall k = 1, \cdots, n \tag{6}$$

$$x_i^{kc} \in \{0, 1\} \qquad \forall i = 1, \cdots, n; \forall k = 1, \cdots, n; \forall c = 0, \cdots, q \tag{7}$$

$$t_{ij} \in \{0, 1\} \qquad \forall i, j = 1, \cdots, n | i < j \tag{8}$$

In this formulation potential regions are represented by an index k. We call then "potential regions" because we do not know a priori how many regions will be created. When a region k is created it starts from a "root"

area i, which is an area assigned to region k in order zero (i.e., X_i^{k0}). Each region contains one and only one root area. The other areas are assigned to one root according to an ordering system that ensures that each area either is adjacent to the root area, or next to an area that is assigned to the same region with a smaller order number. 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 MIP model is formulated as a minimization problem with an objective function that comprises two terms, one term that controls the number of regions, p, and a second term that controls the total heterogeneity, $H(P_p)$. The first term is obtained by adding the number of areas designated as root areas (X_i^{k0}), and the second term adds the pairwise dissimilarities between areas assigned to the same region. Since the objective function is formulated as a minimization problem, we multiply the first term by minus one.

These two terms are merged into one single value, but not in the usual way (i.e., by multiplying each term by a weight). Instead, we merge them in such a way that there is an implicit hierarchy where the number of p regions comes first than the goal of reducing total heterogeneity. We achieve this hierarchy by multiplying the first term by a scaling factor $h = 1 + \lfloor log(\sum_i \sum_{j|j>i} d_{ij}) \rfloor$. For p regions the objective functions starts at $-p*10^h$. This value increases when we add the total heterogeneity, but h is big enough that, regardless the value of this heterogeneity, the objective function will never reach $-(p-1)*10^h$. This formulation has three implications:

- If the algorithm finds a feasible solution with a higher value of p, the improvement in the objective function will always be big enough that this new solution will be preferred over any other solution with a smaller value of p.
- For the same value of p, solutions with lower heterogeneity will be preferred over solutions with higher heterogeneity.
- The third implication is derived from the two first, and it is that we force the model to compare only total heterogeneities between solutions with the same number of regions. Comparing heterogeneities between solutions with different number of regions would be an unfair comparison.

Constraints (2) establish that a region k should not have more than one core area. A root area for a region has a defined order of zero ($c = 0$). Constraints (3) require that each area i be assigned to exactly one

region k and one contiguity order c. Constraints (4) require that area i be assigned to region k at order c if and only if an area j exist, in the adjacent neighborhood of i, that is assigned to the same region k in order $c-1$. Constraints (5) ensure that when a region is created, the value of the spatially extensive attribute in that region will be above the predefined threshold value. Constraints (6) select the pairwise dissimilarities that must be taken into account for calculating the total heterogeneity. Thus, the binary variable $t_{ij} = 1$ whenever areas i and j are assigned to the same region k, regardless of the order in which they are assigned. Finally, constraint (7) and (8) guarantee variable integrity.

In this formulation we do not impose any constraint on the shape of the regions. Our formulation even allows for regions in the solution that can appear as concentric rings around, for example, a Central Business District.

The MIP formulation of the max-p-regions model is computationally expensive. It has $3n + (n-1)n^2 + n\frac{n^2-n}{2}$ constraints and $(n-1)n^2 + \frac{n^2-n}{2}$ variables, which quickly make it intractable as the number of areas increases. However there are some options that can be considered to reduce the size of the problem:

1. Each area i with $l_i \geq threshold$ can be assigned to a different region k by adding constraints of the type $X_i^{k0} = 1$.

2. The upper limit of the indexes k and c can be reduced, because they were set for very extreme cases. Currently we do not have the decision rules to define how much the upper limits of k and c can be reduced without affecting optimality.

3. It is clear that, for a given solution, the objective function will not be affected if we modify the index of the region, or the order of assignment, as long as the set of areas per region is not modified. This implies that, when using the branch and bound method, the optimal solution will exist in multiple branches of the solution tree. Thus, a *Depth-first* branching direction may reduce the solution time.

4. If we take into account that any area can be the root of its region, then we can apply the "1 in 1" formulation proposed by Rosing and ReVelle (1986) within the context of flow capturing model. According to this formulation, a single area i can be arbitrarily assigned to one specific region without degrading the problem or the objective function. Thus, we can reduce computation time by adding the constraint $X_1^{1,0}$ without affecting optimality.

To illustrate the complexity of the max-p-regions we solved nineteen problems with different number of areas (n) and threshold values ($threshold$). The attributes y, from which the dissimilarities d_{ij} are calculated, were simulated as spatial autoregressive (SAR) processes with a spatial autocorrelation parameter $\rho = 0.8$, mean $= 0$, and the rook criterion of contiguity for constructing the spatial weights. The spatially extensive attributes l where generated from a discrete uniform distribution between 10 and 15. Table 2 summarizes computational results. Only four problems were solved to optimality, and feasible solutions were obtained for six problems. For the other nine problems CPLEX did not find a feasible solution after four hours. It is clear that with the commonly available computational power we currently need to use heuristics to solve meaningfully large problems.[9]

Table 2: Computational experience with CPLEX

problem	n	threshold	solution	p	time (sec.)
1	9	28	$-2.931 \cdot 10^2$	3	0.33
2	9	38	$-1.877 \cdot 10^2$	2	0.19
3	16	51	$-2.965 \cdot 10^3$	3	1257.25
4	16	68	$-1.948 \cdot 10^3$	2	198.86
5	25	52	$-6.069 \cdot 10^3$	6	†
6	25	79	$-3.984 \cdot 10^3$	3	†
7	25	105	$-2.920 \cdot 10^3$	3	†
8	36	53	$-9.094 \cdot 10^4$	7	†
9	36	68	$-7.087 \cdot 10^4$	5	†
10	36	120	–	–	†
11	49	54	–	–	†
12	49	65	–	–	†
13	49	82	–	–	†
14	49	109	$-6.027 \cdot 10^4$	6	†
15	64	52	–	–	†
16	64	60	–	–	†
17	64	64	–	–	†
18	64	84	–	–	†
19	64	140	–	–	†

* Optimal (by CPLEX).
† Run stopped after 4 h.
– No solution found.

[9]Results are based on using ILOG CPLEX 11.2 executed on a Dell Precision T3400 computer running the Windows XP-64bits operating system equipped with 8 GB RAM and a 2.99 GHz Intel Corel 2 Extreme processor.

5 Heuristic solution methods

In this section we propose a heuristic solution for the max-*p*-regions problem. The heuristic is presented in Pseudocode 1 and comprises two phases, a construction phase and a local search phase. The construction phase generates a set of feasible solutions, and the local search phase applies iterative modifications to those feasible solutions in order to improve the evaluation criterion. At the end, the heuristic returns the best solution found.

Pseudocode 1: MAX-P-REGIONS
A : Set of areas,
l : Spatially extensive attribute of areas,
d : Pairwise dissimilarities between areas,
W : Neighbourhoods,
$threshold$: Constraint on attribute l at regional level.

$P_p^{best} = \emptyset$, best partition.
$het = \infty$
$\Pi = \emptyset$, set of feasible partitions.
$\Psi = \emptyset$, set of partitions before enclaves assignment.
$maxP = 0$, maximum number of regions.
Construction Phase:
for $i = 1, 2, \cdots, maxitr$
  do $\begin{cases} \psi, \varepsilon, A' = \textbf{GrowRegions}(A, l, d, W, threshold) \\ p = |\psi|, \text{number of regions in partition } \psi \\ \textbf{if } p > maxP \\ \quad \textbf{then } \begin{cases} \Psi = \psi \\ maxP = p \end{cases} \\ \textbf{if } p = maxP \\ \quad \textbf{then } \{\Psi = \Psi \cup \psi \\ \textbf{if } p < maxP \\ \quad \textbf{then } \{pass \end{cases}$
for ψ in Ψ
  do $\begin{cases} P^{feasible} = \textbf{AssignEnclaves}(\psi, A^a, \varepsilon, d, W) \\ \Pi = \Pi \cup P^{feasible} \end{cases}$
Local Search Phase:
for $P^{feasible}$ in Π
  do $\begin{cases} P_p^{current} = \textbf{LocalSearch}(P^{feasible}) \\ \textbf{if } H(P_p^{current}) < het \\ \quad \textbf{then } \begin{cases} het = H(P_p^{current}) \\ P_p^{best} = P_p^{current} \end{cases} \end{cases}$
return P_p^{best}

5.1 Construction phase

The construction of a feasible solution is divided in two phases: growing phase (see Pseudocode 2), and enclaves assignment (see Pseudocode 3). During the growing phase the algorithm selects at random an unassigned area, which is the "seed area" of a growing region. Then, neighbouring unassigned areas are added to the initial seed until the region reaches the minimum threshold value.[10] Next, the algorithm selects a new seed area to start growing a new region. This process is repeated until it is not possible to grow new regions that satisfy the threshold value. Those areas that are not assigned to a region are known as "enclaves." At the end of the growing phase, the algorithm finished with a set of partial solutions where each solution is composed by a set of growing regions and a set of enclave areas.

The number of feasible growing regions may change from run to run. For this reason the algorithm repeats this procedure multiple times ($maxitr$) and keeps only those solutions where the number of growing regions is equal to the maximum number of regions obtained in prior iterations. Each partial solution is then passed to the process of enclaves assignment. In this phase each enclave area must be assigned to one neighbouring growing region according to a measure of similarity.[11] Once all the partial solutions have passed through the enclave assignment process, the algorithm has a set of feasible solutions, all of them with the same number of regions.

[10] The strategy of creating regions from the selection of an initial area appeared in the early 60s with Vickrey (1961) for solving districting problems. Variations of this methodology have been proposed by Thoreson and Littschwager (1967), Gearhart and Liittschwager (1969), Taylor (1973), Openshaw (1977a), Openshaw (1977b), and Rossiter and Johnston (1981).

[11] This implies that the enclave assignment process do not modify the number of regions. It just ensures the exhaustive assignment of areas to regions.

Pseudocode 2: GROWREGIONS
$A, l, d, W, threshold$

Comment: Grow regions from initial seeds such that the value of attribute l in each region is above *threshold*.

$\Psi = \emptyset$, set of partitions before enclaves assignment.
$\varepsilon = \emptyset$, set of enclave areas.
$A^u = A$, set of unassigned areas.
$A^a = \emptyset$, set of assigned areas.
while $A^u \neq \emptyset$
do
- $A_k =$ select, at random, one area from A^u.
- $A^u = A^u - \{A_k\}$, remove area A_k from set A^u.
- $A^a = A^a \cup \{A_k\}$, add area A_k to set A^a.
- **if** $l_k \geq threshold$
  - **then**
    - $R_k = \{A_k\}$, area A_k becomes a region by itself.
    - $\Psi = \Psi \cup \{R_k\}$, add region R_k to partition Ψ.
- **if** $l_k < threshold$
  - **then**
    - $R_k = \{A_k\}$, start a growing region seeded at area A_k.
    - $N = neigbours(A_k) - A^a$, set of neighbouring unassigned areas of A_k.
    - $L = l_k$, value of attribute l in area A_k.
    - $feasible = 1$
    - **while** $T < threshold$
      - **do**
        - **if** $N \neq \emptyset$
          - **then**
            - $A_i =$ area in N that minimizes the greedy adaptative function $g(A_i) = \sum_{j \in R_k} d_{ij}$
            - $R_k = R_k \cup \{A_i\}$
            - $N = (N - \{A_i\}) \cup neigbours(A_i) - A^a$
            - $T = T + l_i$
            - $A^u = A^u - \{A_i\}$
            - $A^a = A^a \cup \{A_i\}$
        - **if** $N = \emptyset$ and $T < threshold$
          - **then**
            - $\varepsilon = \varepsilon \cup R_k$
            - $feasible = 0$
            - $A^u = A^u \cup R_k$
            - $A^a = A^a - R_k$
            - *break*, leave the **while** loop.
    - **if** $feasible = 1$
      - **then** $\Psi = \Psi \cup \{R_k\}$
return Ψ, ε, A^a

Pseudocode 3: AssignEnclaves
$\psi, A^a, \varepsilon, d, W$

Comment: Assign each enclave in ε to one growing region in partition ψ.

while $\varepsilon \neq \emptyset$

do
- A_i = select an area A_i in ε that shares a border with at least one area in A^a.
- η = regions $\eta \subset \psi$ that share border with area A_i.
- R_k = region $R_k \subset \eta$ that minimizes the greedy adaptative function $g(A_i, R_k) = \sum_{j \in R_k} d_{ij}$.
- $R_k^{new} = R_k \cup \{A_i\}$
- $\psi = \psi - \{R_k\} \cup \{R_k^{new}\}$, update region R_k in ψ.
- $A^a = A^a \cup \{A_i\}$, update set of assigned areas.
- $\varepsilon = \varepsilon - A_i$, update set of enclaves.

$P^{feasible} = \psi$, at this point all the areas have been assigned to a region.
return $P^{feasible}$

5.2 Local search phase

Each one of these feasible solutions generated during the construction phase is then improved by applying a local search algorithm. The local search algorithm iteratively modifies the solution while seeking for improvements on the evaluation criterion. The set of new solutions that can be obtained from a current solution is known as the set of neighbouring solutions. There exist several ways to create this set: (a) moving one area from its regions to a neighbouring region, (b) swapping areas between two regions, (c) merging two regions and splitting them into two new regions, or (d) combining two feasible solutions into a new different feasible solution using genetic algorithms operators. Regardless of the strategy for creating neighbouring solution, the conditions is that each neighboring solution must generate a feasible solution.[12] In this paper, we define a neighbouring solution as the new feasible solution obtained by moving one area from its current region (donor region) to another neighboring region (recipient region). This neighbouring function has been applied by Bozkaya et al. (2003), Openshaw and Rao (1995), Ricca and Simeone (2008), Blais et al. (2003), and Bong and Wang (2004) for different types of spatial clustering problems.

We consider three different local search algorithms with the aim of determining which one performs better for the max-*p*-regions problem: Simulated

[12] See Nagel (1965), Sammons (1978) and Horn (1995) for a review on the different possibilities to generate neighbouring solutions within the context of spatial clustering.

Annealing (Kirkpatrick et al., 1983), Tabu Search (Glover, 1977) and Greedy Algorithm.

5.2.1 Simulated annealing

Simulated Annealing is described in Pseudocode 4. This algorithm starts from an initial feasible solution. Then, a neighbouring feasible solution is selected at random. If the neighbouring solution is better than the current solutions, then the move is accepted. If the neighbouring solution does not improve the current solution, then the transition to the new solution is allowed with an acceptance probability given by the Boltzmann's equation, $p = e^{-\Delta H/T}$, where ΔH is the change in the evaluation criterion, and T is the current temperature. At each iteration the temperature T gradually decreases at a given cooling rate α. Thus, as the algorithm progresses probability of accepting a non-improving move approaches to zero. The algorithm stops when T reaches a predefined value ϵ. The key parameter in this algorithm is the cooling rate α.

Pseudocode 4: LOCAL SEARCH: SIMULATEDANNEALING
$P^{feasible}, T_0, \alpha, \epsilon$

$P_p^{'} = P^{feasible}$, Best local optimum
$P_p^{current} = P^{feasible}$, Current solucion
$T = T_0$, Initial temperature
while $T \geq \epsilon$
do {
  Select at random a feasible neighbouring solution P_p^{new} of $P_p^{current}$
  if $H(P_p^{new}) < H(P_p^{'})$
    then { $P_p^{'} = P_p^{new}$; $P_p^{current} = P_p^{new}$
  else if $e^{-\Delta/T} > random$
    then { $P_p^{current} = P_p^{new}$
  $T = \alpha T$
}
return $(P_p^{'})$

5.2.2 Tabu search algorithm

The Tabu search algorithm is presented in Pseudocode 5. This metaheuristic is provided with a good capacity of escaping from local optimal solution by allowing a temporal worsening of the evaluation criterion with the hope of discovering a new solution better that the best solution obtained so far. It starts from an initial feasible solution. From this point the algorithm

moves to the best neighbouring solution even if this move causes a deterioration of the evaluation criterion (total heterogeneity). To prevent cycles, the reverse move is forbidden, or tabu, for a predefined number of iterations ($lengthTabu$). A tabu move is allowed only if the move yields a solution better than the best obtained so far (aspirational criterion). The algorithm stops when a total of $convTabu$ iterations have been performed without improving the aspirational criterion. According to the literature, the most critical parameter in this heuristic is the length of the tabu list, $lengthTabu$.

Pseudocode 5: LOCAL SEARCH: TABUSEARCH
$P^{feasible}, lengthTabu, convTabu$

$P_p^{'} = P_p^{current} = P^{feasible}$
$tabuList = \{\}$
$c = 1$
while $c \leq$ $convTabu$
do
- $N =$ Set of feasible neighbors of $P_p^{current}$
- **if** $N = \emptyset$
  - **then** $\{c = convTabu$
  - **else**
    - **for** P_p^{new} in N
      - **do**
        - **if** $P_p^{new} \notin tabuList$
          - **then**
            - **if** $H(P_p^{new}) < H(P_p^{'})$
              - **then**
                - $P_p^{'} = P_p^{new}$
                - $P_p^{current} = P_p^{new}$
                - $c = 1$
                - $N = \{\}$
                - $tabuList.add(P_p^{new})$
              - **else**
                - $P_p^{current} = P_p^{new}$
                - $c = c + 1$
                - $N = \{\}$
          - **else**
            - **if** $H(P_p^{new}) < H(P_p^{'})$
              - **then**
                - $P_p^{'} = P_p^{new}$
                - $P_p^{current} = P_p^{new}$
                - $c = 1$
                - $N = \{\}$
                - $tabuList.add(P_p^{new})$
              - **else**
                - $N = N - P_p^{new}$
                - $tabuList.pop()$

return $(P_p^{'})$

5.2.3 Greedy algorithm

The Greedy algorithm, described in Pseudocode 6, starts from an initial feasible solution, and selects a neighbouring solution at random. The neighbouring solution is allowed only if it improves the current solution. The algorithm stops when there is no neighbouring solution that improves the current solution. The Greedy algorithm is fast but it may easily get trapped into a local optimum.

Pseudocode 6: LOCAL SEARCH: GREEDY
$P^{feasible}$

$P_p^{'} = P^{feasible}$
$flag = 1$
while $flag$
do $\begin{cases} N = \text{Set of feasible neighbors of } P_p^{'} \text{ that improve the solution} \\ \textbf{if } N \neq \emptyset \\ \quad \textbf{then } \{P_p^{'} = \text{Randomly selects an element of } N \\ \quad \textbf{else } \{flag = 0 \end{cases}$
return $(P_p^{'})$

Two are the main challenges in the application of local search algorithms to the problem of spatial clustering: (a) to avoid getting trapped in a local optimal solution, and (b) to find feasible neighboring solutions efficiently. However, these techniques have been widely applied in other problems that impose spatial contiguity constraint. For example, simulated annealing has been applied in political districting by Browdy (1990), Macmillan and Pierce (1994), Macmillan (2001), and Ricca and Simeone (2008); in zone design by Openshaw and Rao (1995); and in police districting by D'amico et al. (2002). Tabu search as been applied in political districting by Bozkaya et al. (2003), Bong and Wang (2004), and Ricca and Simeone (2008); in zone design by Openshaw and Rao (1995); and in home care districting by Blais et al. (2003). And the greedy algorithm has been applied in constrained clustering by Bodin (1973), Fischer (1980), and Ferligoj and Batagelj (1982); in political districting by Nagel (1965), Liittschwager (1973), Moshman and Kokiko (1973), Horn (1995), Ricca and Simeone (2008), and Yamada (2009); and in zone design by Openshaw (1977a), and Openshaw and Rao (1995).

5.3 Computational experiments

In this section we study the performance of the three local search algorithms presented above. Table 3 presents the characteristics of the data set utilized

in the experiments. The irregular lattices were obtained from the sample data sets available at the GeoDa Center for Geospatial Analysis and Computation.[13] We used two different values for ρ, 0.6 and 0.9, in order to evaluate whether there is a change in the performance of the algorithms at different levels of spatial dependence.

Table 4 presents the parameters we use for the local search algorithms. In both algorithms we use different values for the key parameters: in Simulated Annealing (SA) we use two different values for the cooling rate (α), and in Tabu Search we use three different values for the length of the tabu list. This gives a total of six algorithms: Greedy, SA-0.9, SA-0.998, Tabu-10, Tabu-24, and Tabu-85. All the values for the parameters are based on previous experiments presented in Ríos-Mercado and Fernández (2009), Ricca and Simeone (2008), Bong and Wang (2004), Blais et al. (2003), Bozkaya et al. (2003), D'amico et al. (2002), Macmillan (2001), Openshaw and Rao (1995), Macmillan and Pierce (1994), and Browdy (1990).

Table 3: Characteristics of the data set

Characteristic	Values
Lattices	regular 20x20 ($n = 400$)
	regular 33x33 ($n = 1,056$)
	regular 55x56 ($n = 3,080$)
	Sacramento census tracks ($n = 403$)
	Colombian municipalities ($n = 1,068$)
	US census tracks ($n = 3,085$)
Neighbourhoods type	rook
y	SAR ($\rho = 0.6$) and SAR ($\rho = 0.9$)
l	Discrete Uniform [0,100]
threshold (TH)	100, 300, and 500

Table 4: Parameters for local search algorithms

Simulated Annealing	
Initial temperature (T_0)	1
Cooling rate (α)	0.9 and 0.998
Final temperature (ϵ)	0.0001
Tabu Search	
Tabu list length	10, 24, and 85
Maximum number of non-improving moves	$230 * sqrt(p)$

In order to make the results comparable, we generate an initial feasible

[13] http://geodacenter.asu.edu/sdata.

solution at random for each combination of lattice, ρ, and *threshold*. Then, we run the six local search algorithm with the same starting solution. This process is repeated ten times with different starting solutions. Thus, we solve a total of 2,160 problems (6 lattices $\times$ 2 values of ρ $\times$ 3 Threshold values $\times$ 6 algorithms $\times$ 10 repetitions).

Our results are presented in tables 5, 6, and 7. Each cell summarizes the results of solving the ten problems. Table 5 reports the number of times that each algorithm reached the best known solution. Table 6 reports the average reduction of the evaluation criterion (total heterogeneity), calculated as $[H(P^{initial}) - H(P^{final})]/H(P^{initial})]$, where $H(P^{initial})$ is the total heterogeneity of the initial feasible solution, and $H(P^{final})$ is the total heterogeneity at the end of the local search. Table 7 reports the average running times in seconds.

The results in Table 5 show that Tabu-85 reached the best known solution 71.11% of the cases, follow by Tabu-24 with 23.61%, Tabu-10 with 16.94%. The SA-0.998 and Greedy algorithms are significantly inferior with an 0.83% success rate, followed by SA-0.9 with 0.56%. Our results suggest that the larger the length of the tabu list, the higher the possibilities are to get the best solution. This finding is in line with Bozkaya et al. (2003) who found the best performance of Tabu Search for a list length of between 80 and 100. It is also important to note that longer tabu lists do not imply a significant change in the running times.

A comparison of Simulated Annealing with Tabu Search results in tables 6 and 7 shows that, on average, to get an additional reduction of 0.73% in the total heterogeneity using Tabu Search causes the running time to an increase by a factor of 4.84. Depending on the context of the application, this trade-off can be very expensive.

Contrary to our expectations, there is not a significant difference in the performance of the algorithms at different levels of spatial dependence; i.e., having clearer spatial patterns neither helps the algorithms to converge faster nor to reach a higher reduction of the initial objective function value. This finding implies that it is not necessary to consider the level of spatial dependence of the variables in y when calibrating the parameters of the algorithms.

Differences between regular and irregular lattices have a significant impact on the evaluation criterion and solution times. For irregular lattices, we found a 12.84% reduction in the capacity of the algorithms to reduce the evaluation criterion (Table 6). However, the algorithms converged 8.77% faster with irregular lattices when compared with regular (Table 7).

Table 7 shows that increasing the threshold value (TH) from 100 to 500,

yields a 34.78% reduction in running time for Tabu Search. This effect is the opposite for the other two algorithms: Simulated Annealing produces increases in running time by an average of 84.45%, and the Greedy algorithm multiplies the average running time by a factor of 2.13.

Table 5: Number of times that each algorithm reached the best known solution.

	$\rho = 0.6$			$\rho = 0.9$		
Heuristic	TH=100	TH=300	TH=500	TH=100	TH=300	TH=500
Regular lattice $n = 400$ (20x20)						
Greedy	0	0	0	0	1	1
SA-0.9	0	0	0	0	1	1
SA-0.998	0	0	0	0	1	1
Tabu-10	4	1	7	3	2	6
Tabu-24	6	4	3	4	5	3
Tabu-85	1	6	0	4	4	2
Sacramento census tracks $n = 403$						
Greedy	1	0	0	0	0	0
SA-0.9	0	0	0	0	0	0
SA-0.998	1	0	0	0	0	0
Tabu-10	4	3	4	6	3	1
Tabu-24	2	2	4	4	2	4
Tabu-85	6	6	4	3	6	6
Regular lattice $n = 1,056$ (33x33)						
Greedy	0	0	0	0	0	0
SA-0.9	0	0	0	0	0	0
SA-0.998	0	0	0	0	0	0
Tabu-10	0	0	0	1	0	2
Tabu-24	2	0	1	3	1	5
Tabu-85	8	10	9	9	9	6
Colombia municipalities $n = 1,068$						
Greedy	0	0	0	0	0	0
SA-0.9	0	0	0	0	0	0
SA-0.998	0	0	0	0	0	0
Tabu-10	1	2	0	0	1	2
Tabu-24	2	3	2	1	1	2
Tabu-85	7	9	9	9	8	8
Regular lattice $n = 3,080$ (55x56)						
Greedy	0	0	0	0	0	0
SA-0.9	0	0	0	0	0	0
SA-0.998	0	0	0	0	0	0
Tabu-10	3	0	1	2	1	0
Tabu-24	3	0	5	1	1	1
Tabu-85	7	10	7	10	10	9
US counties $n = 3,085$						
Greedy	0	0	0	0	0	0
SA-0.9	0	0	0	0	0	0
SA-0.998	0	0	0	0	0	0
Tabu-10	0	0	0	0	0	1
Tabu-24	2	1	2	2	1	0
Tabu-85	8	9	8	10	10	9

Table 6: Average reduction of the evaluation criterion (%).

	$\rho = 0.6$			$\rho = 0.9$		
Heuristic	TH=100	TH=300	TH=500	TH=100	TH=300	TH=500
Regular lattice $n = 400$ (20x20)						
Greedy	6.11	2.43	1.19	5.52	1.96	1.07
SA-0.9	6.58	2.55	1.21	5.62	1.98	1.08
SA-0.998	6.37	2.53	1.21	5.62	1.98	1.08
Tabu-10	8.67	3.13	2.00	6.52	2.54	1.93
Tabu-24	8.57	3.32	1.90	6.90	2.71	1.86
Tabu-85	7.32	3.46	1.29	6.53	2.52	1.59
Sacramento census tracks $n = 403$						
Greedy	4.25	1.22	0.90	4.37	1.85	1.10
SA-0.9	4.48	1.64	0.94	4.49	1.85	1.28
SA-0.998	4.37	1.52	0.93	4.46	2.02	1.26
Tabu-10	5.62	2.44	1.67	6.36	2.41	1.50
Tabu-24	5.77	2.56	1.85	6.42	2.44	1.89
Tabu-85	5.90	2.79	1.76	6.17	2.62	1.99
Regular lattice $n = 1,056$ (33x33)						
Greedy	4.29	1.01	0.97	4.48	0.62	1.20
SA-0.9	8.57	3.47	1.57	7.66	3.30	1.71
SA-0.998	5.78	2.36	1.21	5.92	1.87	1.41
Tabu-10	8.42	3.17	1.74	8.01	3.12	2.00
Tabu-24	8.57	3.41	1.84	8.20	3.18	2.11
Tabu-85	8.81	3.69	2.02	8.41	3.39	2.22
Colombia municipalities $n = 1,068$						
Greedy	3.82	0.73	0.51	3.50	0.97	0.77
SA-0.9	7.26	2.49	1.91	6.75	2.43	1.96
SA-0.998	4.92	1.61	1.16	4.76	1.41	1.24
Tabu-10	6.93	2.25	1.65	6.59	2.25	1.78
Tabu-24	7.04	2.44	1.82	6.76	2.29	1.85
Tabu-85	7.40	2.60	2.08	6.96	2.51	2.08
Regular lattice $n = 3,080$ (55x56)						
Greedy	2.93	0.41	0.22	3.30	0.60	0.30
SA-0.9	8.19	3.00	2.07	8.04	3.18	1.97
SA-0.998	6.32	1.92	1.35	6.14	2.14	1.19
Tabu-10	8.00	2.91	1.96	8.00	3.09	1.86
Tabu-24	8.06	2.95	2.00	8.01	3.15	1.88
Tabu-85	8.20	3.01	2.09	8.06	3.19	1.98
US counties $n = 3,085$						
Greedy	3.06	0.69	0.43	3.36	0.86	0.41
SA-0.9	7.07	2.53	1.51	6.78	2.54	1.47
SA-0.998	5.32	1.69	0.91	4.90	1.54	0.83
Tabu-10	6.95	2.46	1.41	6.71	2.44	1.40
Tabu-24	7.01	2.50	1.44	6.73	2.47	1.41
Tabu-85	7.08	2.54	1.52	6.79	2.55	1.48

Table 7: Average running time (seconds).

	$\rho = 0.6$			$\rho = 0.9$		
Heuristic	TH=100	TH=300	TH=500	TH=100	TH=300	TH=500
Regular lattice $n = 400$ (20x20)						
Greedy	1.73	2.10	2.90	1.42	2.06	2.70
SA-0.9	1.69	3.91	3.68	1.43	3.04	3.84
SA-0.998	60.80	194.91	188.34	73.28	156.81	194.54
Tabu-10	282.95	174.28	116.10	224.44	152.48	135.75
Tabu-24	191.81	149.28	150.33	147.16	108.75	115.17
Tabu-85	207.81	220.40	402.04	186.51	169.44	297.86
Sacramento census tracks $n = 403$						
Greedy	1.46	2.33	2.08	1.36	1.70	2.19
SA-0.9	2.63	4.03	3.93	2.19	3.68	4.02
SA-0.998	109.19	183.87	201.02	96.48	183.23	203.39
Tabu-10	197.17	115.26	86.97	206.24	130.08	93.65
Tabu-24	170.32	103.59	93.02	169.71	106.49	84.42
Tabu-85	148.63	167.09	153.94	214.57	115.52	149.73
Regular lattice $n = 1,056$ (33x33)						
Greedy	16.48	47.46	50.37	13.41	47.75	36.39
SA-0.9	41.56	48.43	30.21	28.71	60.32	29.60
SA-0.998	441.89	928.46	924.62	394.34	982.90	860.69
Tabu-10	3,530.05	2,509.21	1,952.74	3,147.65	2,672.26	2,018.99
Tabu-24	3,328.18	2,332.97	1,724.53	2,970.06	2,553.66	1,617.56
Tabu-85	2,973.35	1,931.83	1,262.03	2,227.80	2,207.33	1,114.55
Colombia municipalities $n = 1,068$						
Greedy	21.80	42.81	58.81	23.43	41.06	64.70
SA-0.9	31.54	37.33	39.56	30.25	39.09	40.61
SA-0.998	519.97	987.90	1,351.52	575.65	1,175.31	1,337.92
Tabu-10	2,090.06	1,434.37	1,196.22	2,183.32	1,530.76	1,350.57
Tabu-24	2,006.82	1,347.62	1,141.98	2,063.85	1,453.22	1,270.22
Tabu-85	2,197.85	1,311.72	1,117.74	1,823.41	1,418.37	1,150.04
Regular lattice $n = 3,080$ (55x56)						
Greedy	352.42	1,978.78	2,575.65	669.12	2,375.12	2,904.57
SA-0.9	670.17	742.92	774.82	646.77	781.64	768.19
SA-0.998	4,580.04	10,109.48	9,757.43	4,982.57	11,094.88	10,080.38
Tabu-10	65,327.64	46,601.99	40,360.95	62,786.81	47,136.13	38,793.53
Tabu-24	62,771.18	43,679.58	37,356.54	59,665.65	44,649.17	36,327.65
Tabu-85	61,225.69	42,049.43	35,714.13	59,791.21	45,265.58	39,431.93
US counties $n = 3,085$						
Greedy	661.42	2,893.20	3,118.18	860.95	3,390.53	2,905.46
SA-0.9	399.73	451.83	435.96	452.39	527.75	496.55
SA-0.998	5,812.57	12,911.45	12,637.12	6,837.01	14,380.63	13,403.45
Tabu-10	37,339.67	25,348.69	20,360.10	42,089.08	30,138.07	23,410.21
Tabu-24	35,755.91	24,305.28	19,247.84	40,838.94	29,216.97	22,186.28
Tabu-85	35,091.20	24,738.21	18,973.94	40,191.39	30,613.88	23,694.65

6 Conclusions and future research

In this paper we presented a new type of constrained clustering problem that we coined as the max-p-regions problem. This problem involves the aggregation of small areas into the maximum number of homogeneous regions such that the regional value of a spatially extensive attribute is above a minimum threshold value.

There are many potential applications of our model. For example, the max-p can be used in the design of study regions that allow valid statistical inference in the presence of spatial heteroskedasticity such as in spatial epidemiology studies that require a fair comparison of rate estimates across regions. In addition, our approach can be explored as a way to control for spurious spatial autocorrelation while minimizing the aggregation bias.

Classical problems in the literature can be also reformulated as a max-p-regions problem. For example, all the formulations on police districting and sales territory alignment assume that the headquarters or stores are already located in the territory. This may be an overly strict assumption. For instance, it is plausible that a researcher is confronted with a situation where those facilities do not yet exist or they need to be reallocated. Then, the max-p-regions model can be used to aggregate the areas into regions such that the regions are homogeneous in terms of customer characteristics or crime types, and each region contains a minimum amount of potential customers or emergency calls. Next, once the regions are designed, one can decide the best location of facility within each region at a subsequent stage.

References

Assunção, R. M., Neves, M. C., Câmara, G., and Freitas, C. D. C. (2006). Efficient regionalization techniques for socio-economic geographical units using minimum spanning trees. *International Journal of Geographical Information Science*, 20(7):797–811.

Blais, M., Lapierre, S., and Laporte, G. (2003). Solving a home-care districting problem in an urban setting. *Journal of the Operational Research Society*, 54(11):1141–1147.

Bodin, L. (1973). A districting experiment with a clustering algorithm. *Annals of the New York Academy of Sciences*, 219:209–214.

Bong, C. and Wang, Y. (2004). A multiobjective hybrid metaheuristic ap-

proach for GIS-based spatial zoning model. *Journal of Mathematical Modelling and Algorithms*, 3:245–261.

Bozkaya, B., Erkut, E., and Laporte, G. (2003). A tabu search heuristic and adaptive memory procedure for political districting. *European Journal of Operational Research*, 144:12–26.

Browdy, M. (1990). Simulated annealing: An improved computer model for political redistricting. *Yale Law & Policy Review*, 8(1):163–79.

Bunge, W. (1966). Gerrymandering, geography, and grouping. *Geographical Review*, 56(2):256–263.

Byfuglien, J. and Nordgard, A. (1973). Region-building: A comparison of methods. *Norwegian Journal of Geography*, 27:127–151.

Caro, F., Shirabe, T., Guignard, M., and Weintraub, A. (2004). School redistricting: embedding GIS tools with integer programming. *Journal of the Operational Research Society*, 55(8):836–849.

Cliff, A. and Hagget, P. (1970). On the efficiency of alternative aggregations in region-building problems. *Environment and Planning*, 2:285–294.

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

Cova, T. and Church, R. (2000). Contiguity constraints for single-region site search problems. *Geographical Analysis*, 32:306–329.

D'amico, S., Wang, S., Batta, R., and Rump, C. (2002). A simulated annealing approach to police districting design. *Computer & Operations Research*, 29(6):667–684.

Duque, J., Church, R., and Middleton, R. (2010). The *p*-regions problem. *Geographical Analysis*, forthcoming.

Duque, J., Ramos, R., and nach., J. S. (2007). Supervised regionalization methods: A survey. *International Regional Science Review*, 30:195–220.

Ferligoj, A. and Batagelj, V. (1982). Clustering with relational constraint. *Psychometrika*, 47(4):413–426.

Fischer, M. (1980). Regional taxonomy. A comparison of some hierarchic and non-hierarchic strategies. *Regional Science and Urban Economics*, 10:503–537.

Gearhart, B. and Liittschwager, J. (1969). Legislative districting by computer. *Behavioral Science*, 14(5):404–417.

Glover, F. (1977). Heuristic for integer programming using surrogate constraints. *Decision Science*, 8:156–166.

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

Gordon, A. D. (1996). A survey of constrained classification. *Computational Statistics & Data Analysis*, 21:17–29.

Hansen, P., Jaumard, B., Meyer, C., Simeone, B., and Doring, V. (2003). Maximum split clustering under connectivity constraints. *Journal of Classification*, 20:143–180.

Horn, M. (1995). Solution techniques for large regional partitioning problems. *Geographical Analysis*, 27(3):230–248.

Keane, M. (1975). The size of region-building problem. *Environment and Planning A*, 7:575–577.

Kirkpatrick, S., Gelatt, C., and Vecchi, M. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.

Lankford, P. (1969). Regionalization: Theory and alternative algorithms. *Geographical Analysis*, 1:196–212.

Lefkovitch, L. (1980). Conditional clustering. *Biometrics*, 36(1):43–58.

Legendre, P. (1987). *Developments in numerical ecology. NATO ASI Series, Vol. G 14*, chapter Constrained clustering, pages 289–307. Springer, Berlin.

Liittschwager, J. (1973). The Iowa redistricting system. *Annals of the New York Academy of Sciences*, 219:221–235.

Macmillan, W. (2001). Redistricting in a GIS environment: An optimization algorithm using switching-points. *Journal of Geographical Systems*, 3:167–180.

Macmillan, W. and Pierce, T. (1994). *Spatial Analysis and GIS*, chapter Optimization modelling in a GIS framework: the problem of political redistricting, pages 221–246. Taylor & Francis, London.

Maravalle, M. and Simeone, B. (1995). A spanning tree heuristic for regional clustering. *Communications in Statistics-Theory and Methods*, 24:625–639.

Maravalle, M., Simeone, B., and Naldini, R. (1997). Clustering on trees. *Computational Statistics & Data Analysis*, 24:217–234.

Margules, C., Faith, D., and Belbin, L. (1985). An adjacency constraint in agglomerative hierarchical classifications of geographic data. *Environment and Planning A*, 17:397–412.

Martin, D., Nolan, A., and Tranmer, M. (2001). The application of zone-design methodology in the 2001 UK census. *Environment and Planning A*, 33:1949–1962.

Milligan, G. and Cooper, M. (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50:159–179.

Moshman, J. and Kokiko, E. (1973). A redistricting algorithm applied to geographic reorganization of circuit courts. *Annals of the New York Academy of Sciences*, 219:236–245.

Murray, A. and Shyy, T. (2000). Integrating attribute and space characteristics in choropleth display and spatial data mining. *International Journal of Geographical Information Science*, 14(7):649–667.

Murtagh, F. (1985). A survey of algorithms for contiguity-constrained clustering and related problems. *The Computer Journal*, 28(1):82–88.

Murtagh, F. (1992). Contiguity-constrained clustering for image analysis. *Pattern Recognition Letters*, 13:677–683.

Nagel, S. (1965). Simplified bipartisan computer redistricting. *Stanford Law Review*, 17(5):863–899.

Openshaw, S. (1973). A regionalization program for large data sets. *Computer Applications*, 3(4):136–147.

Openshaw, S. (1977a). A geographical solution to scale and aggregation problems in region-building, partition and spatial modeling. *Transactions of the Institute of British Geographers*, 2(4):459–472.

Openshaw, S. (1977b). Optimal zoning systems for spatial interaction models. *Environment and Planning A*, 9:169–184.

Openshaw, S. and Rao, L. (1995). Algorithms for reengineering 1991 census geography. *Environment and Planning A*, 27(3):425–446.

Perruchet, C. (1983). Constrained agglomerative hierarchical classification. *Pattern Recognition*, 16(2):213–217.

Ricca, F. and Simeone, B. (2008). Local search algorithms for political districting. *European Journal of Operational Research*, 189:1409–1426.

Ríos-Mercado, R. and Fernández, E. (2009). A reactive GRASP for a commercial territory design problem with multiple balancing requirements. *Computers & Operations Research*, 36:755–776.

Rosing, K. and ReVelle, C. (1986). Optimal clustering. *Environment and Planning A*, 18:1463–1476.

Rossiter, D. and Johnston, R. (1981). Program GROUP-the identification of all possible solutions to a constituency-delimitation problem. *Environment and Planning A*, 13:231–238.

Sammons, R. (1978). *A simplistic approach to the redistricting problem*, chapter Spatial representation and spatial interaction, pages 71–94. M. Nijhoff Social Sciences Division, Leiden.

Semple, R. and Green, M. (1984). *Spatial statistics and models*, chapter Classification in human geography, pages 59–79. Springer, Berlin.

Taylor, P. (1973). Some implication of spatial organization of elections. *Transaction of the Institute of British Geographers*, 60:121–136.

Thoreson, J. and Littschwager, J. (1967). Legislative districting by computer simulation. *Behavioral Science*, 12:237–247.

Vickrey, W. (1961). On the prevention of gerrymandering. *Political Science Quarterly*, 76:105–110.

Webster, R. and Burrough, P. (1972). Computer-based soil mapping of small areas from sample data II. Classification smoothing. *European Journal of Soil Science*, 23(2):222–234.

Williams, J. (1995). Political redistricting: A review. *Papers in Regional Science*, 74(1):13–40.

Wise, S., Haining, R., and Ma., J. (1997). *Recent developments in spatial analysis: Spatial statistics, behavioural modelling, and computational intelligence*, chapter Regionalisation tools for exploratory spatial analysis of health data. Edited by Manfred M. Fischer and Arthur Getis, pages 83–100. Springer, New York.

Yamada, T. (2009). A mini-max spanning forest approach to the political districting problem. *International Journal of Systems Science*, 40(5):471–477.

Zoltners, A. and Sinha, P. (1983). Sales territory alignment: A review and model. *Management Science*, 29(11):1237–1256.