On the Performance of the Subtour Elimination Constraints Approach for the *p*-Regions Problem: A Computational Study

Juan Carlos Duque[1], Mario C. Vélez-Gallego[2], Laura Catalina Echeverri[1]

[1]Research in Spatial Economics (RiSE), School of Economics and Finance, Universidad EAFIT, Medellin, Colombia, [2]Department of Production Engineering, Universidad EAFIT, Medellin, Colombia

The p-regions is a mixed integer programming (MIP) model for the exhaustive clustering of a set of n geographic areas into p spatially contiguous regions while minimizing measures of intraregional heterogeneity. This is an NP-hard problem that requires a constant research of strategies to increase the size of instances that can be solved using exact optimization techniques. In this article, we explore the benefits of an iterative process that begins by solving the relaxed version of the p-regions that removes the constraints that guarantee the spatial contiguity of the regions. Then, additional constraints are incorporated iteratively to solve spatial discontinuities in the regions. In particular we explore the relationship between the level of spatial autocorrelation of the aggregation variable and the benefits obtained from this iterative process. The results show that high levels of spatial autocorrelation reduce computational times because the spatial patterns tend to create spatially contiguous regions. However, we found that the greatest benefits are obtained in two situations: (1) when $n/p \geq 3$; and (2) when the parameter p is close to the number of clusters in the spatial pattern of the aggregation variable.

Introduction

The *p*-regions model devised by Duque, Church, and Middleton (2011) is a mixed integer programming (MIP) model that solves problems related to the exhaustive clustering of a set of *n* geographic areas into *p* spatially contiguous regions while minimizing measures of intraregional heterogeneity.[1] Although problems of spatial clustering have been studied in the literature since the early 1960s (Vickrey 1961), the *p*-regions model is considered the first MIP formulation of the spatial clustering problem, and its appearance made the way for a series of *p*-regions models such as the *max-p-regions* model (Duque, Anselin, and Rey 2012), the *p*-functional regions model (Kim, Chun, and Kim 2013), the *p*-compact regions model (Li, Church, and Goodchild 2014b), and the network-*max-p*-regions (She, Duque, and Ye 2017).

Correspondence: Juan Carlos Duque, Department of Economics, Universidad EAFIT, Carrera 49 7 Sur – 50, Medellin 050022, Colombia, e-mail: jduquec1@eafit.edu.co

The spatial clustering problem is classified as nondeterministic polynomial-time hard (*NP*-hard) (Cliff et al. 1975; Keane 1975), which implies that the complexity of the problem, and therefore the solution time, increases drastically as the problem size increases. Such complexities have positioned heuristic solutions as the most appropriate solution techniques for addressing large instances (Duque, Ramos, and Surinach 2007; Li, Church, and Goodchild 2014a). However, as is already common for many *NP*-hard problems, constant research must be conducted to increase the size of instances that can be solved using exact optimization techniques.

Duque, Church, and Middleton (2011) proposed three different types of constraints to guarantee that each region is contiguous. One of these strategies, which is the focus of this article, is called the *Tree* model, where contiguity is achieved by connecting the areas in a region through an acyclic set of links (a tree). The authors prevent cycles by adapting the tour-breaking constraints devised by Miller, Tucker, and Zemlin (1960) (MTZ) for the Travelling Salesman Problem (TSP). The same article Duque, Church, and Middleton (2011) then showed the potential of using the subtour elimination constraint proposed by Dantzig, Fulkerson, and Johnson (1954) (DFJ constraints). In this case, the *Tree* model is solved without the MTZ constraints, and the resulting subtours are prevented from reappearing by adding the corresponding DFJ constraints. This process is repeated until no subtours appear in the solution. The results obtained from a small computational experiment with 10 instances show the potential of this strategy, with an average reduction of running time of 76.93% and an increase of 10.0% in the number of instances solved to optimality.

In this article, we explore the strategy proposed by Duque, Church, and Middleton (2011) in greater depth. The present work is not focused on proposing a new approach to reduce the computational time but rather aims to obtain a better understanding of the complexity of the spatial clustering problem and of the behavior of the DFJ constraints for the *p*-regions model. We attempt to answer questions such as is there a relationship between the level of spatial autocorrelation of the aggregation variable and the benefits obtained from the DFJ constraints? Intuitively, a higher level of spatial autocorrelation should correspond to a lower amount of the required DFJ constraints, because the spatial clusters would tend to emerge naturally. Some arguments in this vein, albeit without any computational evidence, have been presented by Openshaw and Wymer (1995) and Duque, Artís, and Ramos (2006) for non-exact spatial clustering. We also explore the relationship between the benefits derived from the DFJ constraints and the parameter *p* (number of regions) for a fixed number of areas *n*. In this regard, Duque, Church, and Middleton (2011) find that the lower the ratio n/p, the greater the benefits obtained from the DFJ constraints. We seek to test whether their finding holds for larger instances and different levels of spatial autocorrelation.

Obtaining a deeper understanding of this *NP*-hard problem will provide the academic community with new elements for solving larger instances not only of the *p*-regions model but also of other spatially constrained clustering models derived from the *p*-regions model. This also generates an interesting link between exploratory spatial data analysis (ESDA) and this MIP problem that can open the door to innovations in the area of spatially constrained clustering.

The remainder of the article is organized as follows. A literature review is presented in section "Literature review." The exact formulation of the *p*-regions model is introduced in section "The exact formulation of the *p*-regions model—*Tree* version." The solution approach is described in section "The subtour elimination constraints for the *p*-regions model," and our computational experiments are presented in sections "Computational experiment" and "Disentangling the relationship between the DFJ-p-regions and rho." Finally, conclusions are stated and avenues for future research are proposed in section "Conclusions."

Literature review

Problems related to aggregating areas into spatially contiguous regions have been given different names in the literature (e.g., 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, Haining, and Ma 1997], and clustering under connectivity constraints [Hansen et al. 2003]). All of these problems have two common factors: the spatial contiguity constraint and the exhaustive clustering. However, each problem includes additional conditions or characteristics depending on the area of application. For example, problems within the area of Electoral and School Districting impose additional constraints such as compactness or similarities to existing solutions (Browdy 1990; Macmillan and Pierce 1994; Williams 1995; Macmillan 2001; Caro et al. 2004; Ricca and Simeone 2008); other models of Sales Territory Alignment, Police Districting and Home Care Districting impose additional constraints to guarantee balanced workloads and minimum travel distances (Zoltners and Sinha 1983; D'amico et al. 2002; Blais, Lapierre, and Laporte 2003); and other models (e.g., Zone Design) seek to maximize the goodness of fit of a statistical model or the correlation between two variables (Openshaw and Rao 1995).

Regarding solution methods, due to the complexities of these models, heuristic methods are the most frequently used solution approaches. These methods typically decompose a solution into two phases: (1) construction of an initial feasible solution, usually by growing regions from seed areas and (2) local searches, which involve iterative modifications of a feasible solution while optimizing an aggregation criterion. For a literature review on these models, see Murtagh (1985), Gordon (1996) and Duque, Ramos, and Surinach (2007). More recently, Laura et al. (2015) explored the potential of combining heuristic solutions with parallelized computing to solve large spatially constrained clustering problems efficiently.

The use of exact formulations involving explicit spatial contiguity constraints is a relatively recent trend in the literature. To the best of our knowledge, the first MIP formulation to address the aggregation of n areas into p spatially contiguous regions was proposed by Duque, Church, and Middleton (2011) and is known as the p-regions problem.[2] In this work, the authors introduced three different strategies for ensuring contiguity: (1) an adaption of Miller, Tucker, and Zemlin (1960) tour-breaking constraints (or MTZ constraints) for the so-called Traveling Salesman Problem; (2) the use of ordered-area assignment variables based on an extension of an approach proposed by Cova and Church (2000) for the geographical site design problem; and (3) an extension of the work by Shirabe (2005) using flow constraints. These strategies are referred to as *Tree*, *Order*, and *Flow*, respectively.

The publication of the p-regions model motivated research on exact formulations of MIP models that require contiguity constraints along with additional conditions. The max-p-region, devised by Duque, Anselin, and Rey (2012), aggregates n areas into the maximum number of spatially contiguous regions such that the value of a spatially extensive regional attribute is higher than a predefined threshold value; the p-compact region, proposed by Li, Church, and Goodchild (2014b), seeks to aggregate n spatial units into p-compact, contiguous regions; the p-functional region, created by Kim, Chun, and Kim (2013), defines p functional regions by considering geographic flows; and the network-max-p-regions, devised by She, Duque, and Ye (2017), takes into account the influence of a given street network to generate two types of

regions: corridor regions, consisting on elongated regions along roads, and max-*p*-regions (i.e., shape-free regions) located either close to or far from roads.

The computational complexity associated with the contiguity constraints is one of the main challenges related to the treatment of spatially constrained clustering as an exact optimization problem. These constraints ensure that the areas assigned to the same region are spatially connected. One interesting aspect of this set of constraints is that it is too large, with only a fraction of these constraints actually required to guarantee optimality. Based on this characteristic, Duque, Church, and Middleton (2011) proposed a promising strategy to reduce the running time of the *Tree* version of the *p*-regions while guaranteeing optimality. Their proposal consisted of applying the subtour elimination constraint proposed by Dantzig, Fulkerson, and Johnson (1954) to solve the TSP problem.[3] Dantzig, Fulkerson, and Johnson (1954) introduced a means of solving the TSP in an iterative fashion. In their work, the authors present subtour elimination constraints as additional constraints on the TSP as a means of removing subtours that can arise when solving the problem in its relaxed form. They found that it is sufficient to begin with the relaxed form of the TSP as long as it is possible to determine when the solution of the relaxed problem contains subtours and, if so, to find these subtours in the current solution. They applied this idea to cope with the exponential number of constraints that arise when eliminating subtours. Their remarkable contribution served as a precursor to the development of the branch-and-cut algorithms that are used in practice to solve a broad range of optimization models (Grötschel and Nemhauser 2008). It also became the preferred approach to solve the TSP by integer programming (Orman and Williams 2006), and it has been successfully applied in other spatial optimization problems, such as the shortest covering path problem (Current, ReVelle, and Cohon 1984) and the Critical Cluster Problem (Church and Cova 2000).

In this article, we aim to explore the synergies that may result from combining the two research fields of exact optimization and ESDA (Anselin 2005; Ye and Rey 2013). The contribution by Duque, Church, and Middleton (2011) offers a good case study for this exploration. Specifically, we are interested in knowing how the spatial distribution of the aggregation variable (i.e., the variable used to measure the dissimilarity between the areas to be aggregated) can affect the complexity of the *p*-regions (in its *Tree* version) and the effectiveness of the subtour elimination constraints. Using aggregation variables with different levels of spatial autocorrelation, we can generate a range of spatial distributions that vary from a chessboard-like pattern (high negative spatial autocorrelation) to random spatial patterns (no spatial autocorrelation) and then to clustered spatial patterns (high spatial autocorrelation). By analyzing the performance of the subtour elimination constraints for the *p*-regions in these spatial patterns, we aim to be able to test whether high spatial autocorrelation levels can significantly reduce the number of subtour elimination constraints required to reach optimality. We can also evaluate whether the benefits of this solution approach depend on the relationship between the desired number of clusters, *p*, and the number of clusters generated by the data generation process. The issues addressed in this article can open new research avenues for this and other exact optimization problems.

The exact formulation of the *p*-regions model—*Tree* version

In this section, for the sake of completeness, we present the *Tree* version of the *p*-regions problem, *p*-regions^{Tree}, devised by Duque, Church, and Middleton (2011). This MIP model creates trees by introducing links between areas. A link between two areas denotes that the two areas are spatially connected (i.e., they are neighboring areas). Areas belonging to a given tree form

a region. The main condition for designing regions with trees is that no tree can have cycles. The p-regions^{Tree} prevents cycles via an adaption of Miller, Tucker, and Zemlin (1960) tour-breaking constraints (or MTZ constraints) of the TSP.

In the interest of ensuring readability, we present in this article a complete formulation of the p-regions^{Tree} as presented by Duque, Church, and Middleton (2011).

Parameters:

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

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

$N_i = \{j | c_{ij} = 1\}$, 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 $i < j$;

Decision variables:

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

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

$u_i =$ order assigned to each area i in a subnetwork or tree

Minimize:

$$Z = \sum_i \sum_{j|j>i} d_{ij} t_{ij}. \tag{1}$$

Subject to:

$$\sum_{i=1}^{n} \sum_{j \in N_i} x_{ij} = n - p; \tag{2}$$

$$\sum_{j \in N_i} x_{ij} \leq 1 \quad \forall i = 1, \cdots, n; \tag{3}$$

$$t_{ij} + t_{im} - t_{jm} \leq 1 \quad \forall i, j, m = 1, \cdots, n \text{ where } i \neq j, m \neq j; \tag{4}$$

$$t_{ij} - t_{ji} \leq 0 \quad \forall i, j = 1, \cdots, n; \tag{5}$$

$$x_{ij} - t_{ij} \leq 0 \quad \forall i = 1, \cdots, n; \forall j \in N_i; \tag{6}$$

$$u_i - u_j + (n-p) \times x_{ij} + (n-p-2) \times x_{ji} \leq n - p - 1 \forall i = 1, \cdots, n; \quad \forall j \in N_i; \tag{7}$$

$$1 \leq u_i \leq n - p \quad \forall i = 1, \cdots, n; \tag{8}$$

$$x_{ij} \in \{0, 1\} \quad \forall i = 1, \cdots, n; \forall j \in N_i; \tag{9}$$

$$t_{ij} \in \{0, 1\} \quad \forall i, j = 1, \cdots, n | i < j. \tag{10}$$

The objective function (1) minimizes the sum of pairwise dissimilarities between areas belonging to the same region. Constraint (2) holds that the sum of all selected links (i.e., $x_{ij} = 1$) equals n–p, which is the number of links required to form p trees. Constraints (3) holds that an area i can have

at most one link x_{ij} leaving area i. Constraints (4) generates triangulations between all areas belonging to the same region; that is, if areas i, j and m belong to the same region, then there must exist links of the type $t_{i,j}$ between those areas. Constraints (5) guarantees symmetry in the matrix $t_{i,j}$, so the objective function can be calculated with elements above the main diagonal. Constraints (6) holds that a link between areas i and j in the tree can be selected only if they assign to the same region, that is, $t_{ij} = 1$. Constraints (7) and (8) are tour-breaking MTZ constraints adapted from Miller, Tucker, and Zemlin (1960). These constraints assign an order u_i to each area such that a link of type $x_{i,j}$ must depart from an area i with an order u_i that is less than the order assigned to the arrival area u_j. Altogether this prevents a cycle of links x_{ij} in the tree of a given region. Finally, Constraints (9) and (10) guarantee variable integrity. In total, the number of constraints in expression (7) is given by $\sum_{i \in I} N_i$, whereas the number of constraints in expression (8) is n. The model without these MTZ constraints is hereafter referred to as the relaxed model.

The subtour elimination constraints for the *p*-regions model

The subtour elimination approach for the *p*-regions model, proposed by Duque, Church, and Middleton (2011), consists of an iterative process that begins by solving the relaxed version of the *p*-regions problem that removes the adapted MTZ constraints from its complete formulation (expression (7) and (8)). Then, subtour elimination constraints are incorporated iteratively, that is, the DFJ constraints of the type:

$$\sum_i \sum_j x_{ij} \leq |\Gamma| - 1 \quad \forall i \in \Gamma, \forall j \in N_i \cap \Gamma, \tag{11}$$

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

After the addition of the DFJ constraints, the model is solved again. This process is repeated until a solution is found that contains no cycles. The first solution with no subtours is the optimal solution of the complete *p*-regions problem. The steps above are systematized in Pseudocode 1 and are referred to as the DFJ-*p*-regions algorithm in the rest of this article.

In the *p*-regions^{Tree}, a region can be designed with different trees without affecting the objective function. Fig. 1 shows an example of such an equivalence: the region composed of

---

Algorithm. DFJ-*p*-REGIONS()

   Comment: Reach feasibility by adding DFJ constraints.
while true

do $\begin{cases} solution = \text{solve the relaxed model.} \\ cycles = \text{check for cycles in } solution \text{ and add them to } cycles. \\ \textbf{if } cycles \neq \varnothing \\ \quad \textbf{then} \begin{cases} \text{add } subtour\ elimination\ constraints \text{ to the model based on} \\ \text{Constraints (11).} \end{cases} \\ \text{else} \{\text{STOP.} \end{cases}$

return *solution*
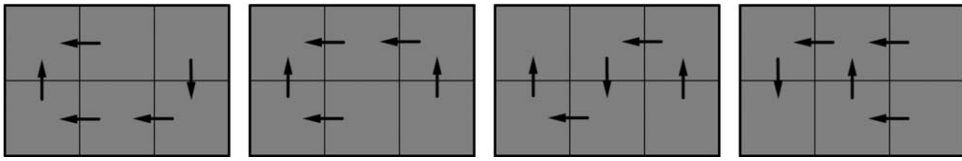
Figure 1. Equivalent feasible regions.

six areas can be obtained with different trees, and the objective function will be the same in all cases. This is because the objective function takes each of the pairwise distances, d_{ij} between the areas assigned to the same region (via the decision variable t_{ij}). Thus, the pairwise distances included in the objective function will be the same regardless of how the areas within the region are linked together.

This particularity of the p-regions^{Tree} model implies that when a region is infeasible (i.e., discontinuous) because of the presence of a subtour in the tree, the addition of the corresponding DFJ constraint, as presented in Constraints (11), may lead to the same infeasible solution. This is because the same infeasible region can remain intact merely by moving the subtour within the region. Fig. 2 shows an example of this situation: Fig. 2a contains an infeasible region, shown in light gray, composed of areas 3, 7, 8, 9, 10, 11, and 12. This region is infeasible because area 3 is disconnected from the other areas. This disconnection is caused by the subtour between areas 7, 8, 10, and 11. If we include the DFJ constraint to break this specific tour, the next solution will generate the same regions (therefore, the same objective function) merely by creating another tour somewhere within the same subregion containing areas 7, 8, 9, 10, 11, and 12. Figs. 2b–d are examples of other subtours that lead to the same solution. This characteristic will require several iterations in the algorithm DFJ-p-regions before solving this infeasibility.

In this article, we propose a minor modification of the DFJ constraints adapted by Duque, Church, and Middleton (2011) that solves the situation presented above. It consists of introducing a modification in the definition of the set Γ in the DFJ constraints (11). Our new definition of Γ is the following: a set of areas that belong to the subregion that contains the subtour. In the example described above, the set Γ would be {7, 8, 9, 10, 11, 12}, and the DFJ constraints for the case illustrated in 2a would be: $x_{7,8} + x_{7,10} + x_{8,7} + x_{8,9} + x_{8,11} + x_{9,8} + x_{9,12} + x_{10,7} + x_{10,11} + x_{11,8} + x_{11,10} + x_{11,12} + x_{12,9} + x_{12,11} \leq 5$. This constraint will also prevent any other subtour within the subregion with areas 7, 8, 9, 10, 11, and 12. Therefore, this specific infeasible solution will not appear again. In contrast, based on the definition of the set Γ proposed by
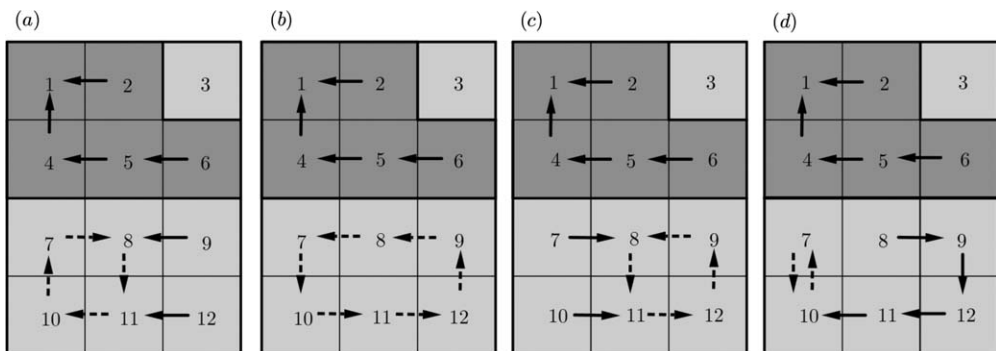
Figure 2. Equivalent infeasible solutions.

Duque, Church, and Middleton (2011), Γ and therefore the DFJ constraints would be different for each case in Fig. 2: for Fig. 2a, $\Gamma = \{7, 8, 10, 11\}$ and the DFJ constraint is $x_{7,8} + x_{7,10} + x_{8,7} + x_{8,11} + x_{10,7} + x_{10,11} + x_{11,8} + x_{11,10} \leq 3$; for Fig. 2b, $\Gamma = \{7, 8, 9, 10, 11, 12\}$ and the DFJ constraint is $x_{7,8} + x_{7,10} + x_{8,7} + x_{8,9} + x_{8,11} + x_{9,8} + x_{9,12} + x_{10,7} + x_{10,11} + x_{11,8} + x_{11,10} + x_{11,12} + x_{12,9} + x_{12,11} \leq 5$; for Fig. 2c, $\Gamma = \{8, 9, 11, 12\}$ and the DFJ constraint is $x_{8,9} + x_{8,11} + x_{9,8} + x_{9,12} + x_{11,8} + x_{11,12} + x_{12,9} + x_{12,11} \leq 3$; for Fig. 2d, $\Gamma = \{7, 10\}$ and the DFJ constraint is $x_{7,10} + x_{10,7} \leq 1$. In this article we will use the new definition of the set Γ when running the DFJ-*p*-regions algorithm.

Example

To illustrate how the DFJ-*p*-regions algorithm works, we present a simple example. Fig. 3 represents a regular lattice with nine areas and a grid grayscale coded according to variable y that is used to calculate dissimilarities between areas (i.e., $d_{ij} = |y_i - y_j|$).

As a benchmark, Fig. 4 shows an optimal solution of the complete problem. Decision variables u are explicitly shown in the corresponding order, whereas decision variables t and x are used to depict the different regions via links and the grayscale code.

Fig. 5 presents the solution and the subtour elimination constraints added during the execution of the DFJ-*p*-regions algorithm (with our new definition of the set Γ). The doted arrows denote the presence of a cycle in a region. The two subtours in Fig. 5 step 1 are good examples of the new definition of the set Γ; on the one hand, the subtour between areas 1 and 2 occurs within a small subregion composed of areas 1 and 2. In this case, $\Gamma = \{1, 2\}$ and the DFJ constraint is $x_{1,2} + x_{2,1} \leq 1$. On the other hand, the subtour between areas 5 and 8 occurs within a larger subregion composed of areas 4, 5, 7, 8 and 9. In this case, $\Gamma = \{4, 5, 7, 8, 9\}$ and the DFJ constraint is $x_{4,5} + x_{4,7} + x_{5,4} + x_{5,8} + x_{7,4} + x_{7,8} + x_{8,5} + x_{8,7} + x_{8,9} + x_{9,8} \leq 4$. This constraint will prohibit the current subtour and any other subtour that leads to the same subregion. Similar situations occur for the subtour between areas 1 and 4 in subregion $\{1, 2, 4\}$ in step 2 and for the subtour between areas 5 and 8 in subregion $\{5, 8, 9\}$ in step 3. When no cycle is present and the minimum gap criterion is achieved, we have found the optimal solution via the DFJ-*p*-regions algorithm. Fig. 6 shows the optimal solution that includes ten DFJ constraints.

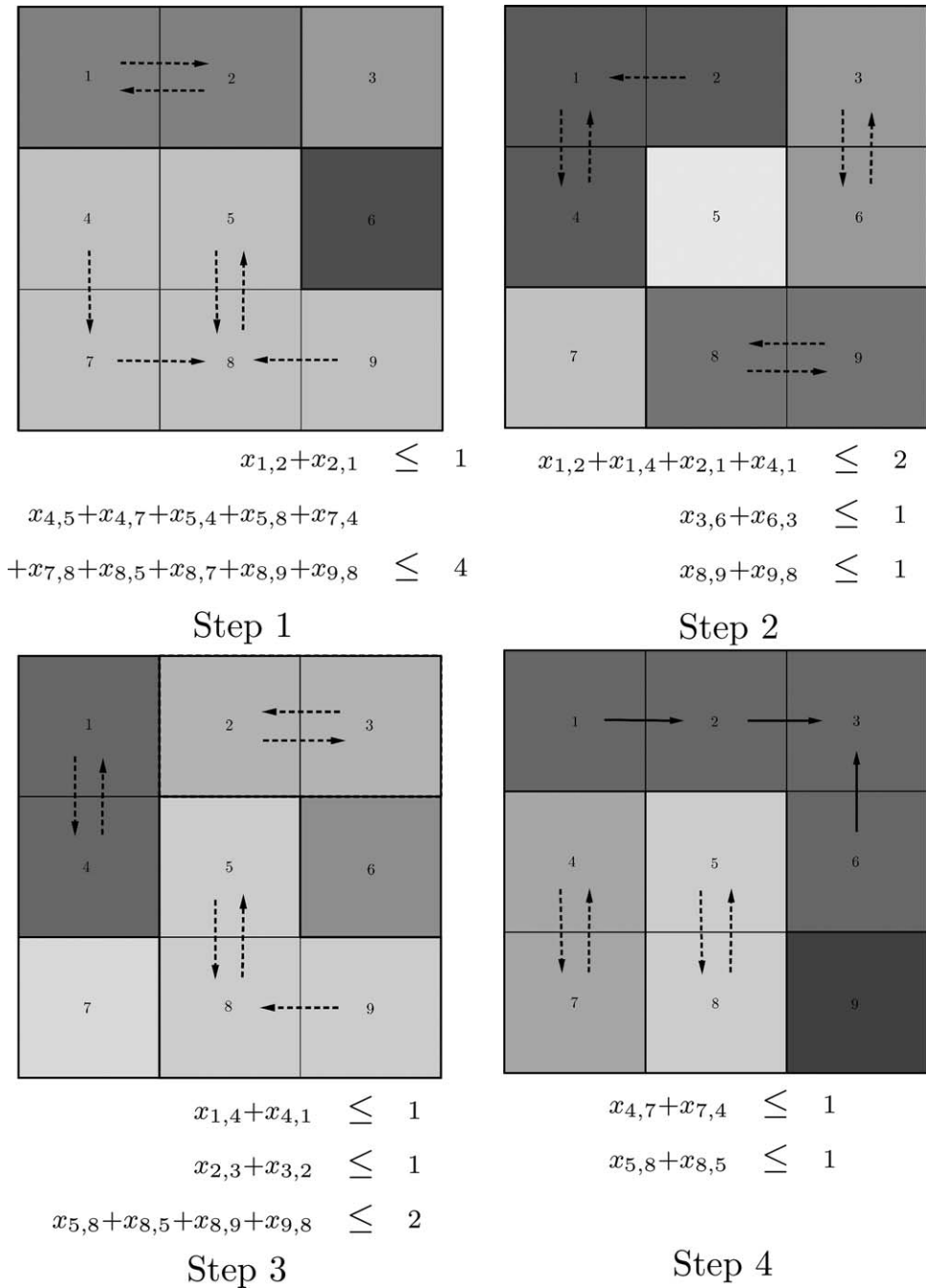|		
$y_1 = 4.857$	$y_2 = 6.164$	$y_3 = 7.907$
$y_4 = 5.753$	$y_5 = 2.034$	$y_6 = 6.308$
$y_7 = 8.367$	$y_8 = 1.137$	$y_9 = 0.778$

Figure 3. Example of input data.
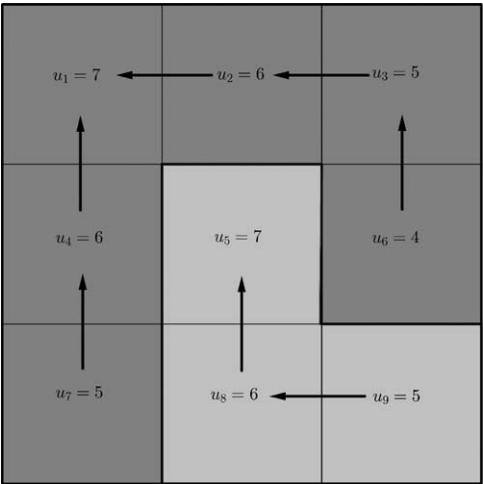
Figure 4. Optimal solution of the complete problem.

Computational experiment

In this section, we test the performance of the DFJ-p-regions algorithm and compare it with the classical solution approach, which involves solving the complete formulation of the p-regions^{Tree} model. Table 1 presents the characteristics of the data set used in the experiments. We created five regular lattices of different sizes, n. The attributes, from which dissimilarities d_{ij} were determined, were simulated as spatial autoregressive processes with six different values for the spatial autocorrelation parameter ρ, mean $=0$, and for spatial weights based on the rook criterion. We also defined three values for the parameter p associated with the number of regions. Finally, for each combination of $[n, p, \rho]$, we generated 30 instances. This gives us a total of 2,700 instances solved using both methods.

We implemented both methods in Python 2.7.9 and used Gurobi 5.6.0 to solve the MIPs problems. The connection between Gurobi and Python was carried out using the Gurobi Python Interface: Python was used to identify subtours, and Gurobi was used as an LP solver. For the DFJ-p-regions algorithm, once a MIP incumbent was found, we determined whether cycles were present, and the corresponding subtour elimination constraints were added using Gurobi lazy constraints. The solver was allowed to run for a maximum of 3 h. We ran the experiments on a Dell Power Edge 1950 GIII 8 cores, 2.33GHz Intel Xeon running Linux Rocks 6.1 at 64 bits of the Apolo Scientific Computing Center of Universidad EAFIT.

To identify if a subregion contains subtours, we proceeded as follows. We used the x_{ij} to construct directed graphs associated with each subregion. Then, recalling from graph theory, we checked for strongly connected components (SCC) in the directed graphs. A SCC is a maximal induced subgraph where for every pair of nodes there is a path back and forth, that is, a SCC is a subtour in the tree. When a graph has fewer SCCs than nodes, it means that it has subtours. In that case, we added the associated subtour elimination constraint, expression (11), where the members of the set Γ are the areas of the corresponding graph. We used the NetworkX Python package (Hagberg, Schult, and Swart 2008) to identify the SCC using Tarjan's algorithm (Tarjan 1972) with Nuutila's modifications (Nuutila and Soisalon-Soininen 1994).

$$x_{1,2}+x_{2,1} \leq 1$$

$$x_{4,5}+x_{4,7}+x_{5,4}+x_{5,8}+x_{7,4}$$
$$+x_{7,8}+x_{8,5}+x_{8,7}+x_{8,9}+x_{9,8} \leq 4$$

Step 1

$$x_{1,2}+x_{1,4}+x_{2,1}+x_{4,1} \leq 2$$

$$x_{3,6}+x_{6,3} \leq 1$$

$$x_{8,9}+x_{9,8} \leq 1$$

Step 2

$$x_{1,4}+x_{4,1} \leq 1$$

$$x_{2,3}+x_{3,2} \leq 1$$

$$x_{5,8}+x_{8,5}+x_{8,9}+x_{9,8} \leq 2$$

Step 3

$$x_{4,7}+x_{7,4} \leq 1$$

$$x_{5,8}+x_{8,5} \leq 1$$

Step 4

Figure 5. DFJ-*p*-regions: steps in branch and bound.

Table 2 reports, for each method, the number of instances that converged and those that did not. After 3 h, the DFJ-*p*-regions algorithm was able to solve 51.41% of the 2,700 instances, whereas the complete formulation solved 41.41% of the instances. This denotes an

41

Running time: 0.30 seconds
Objective Function: 55.630
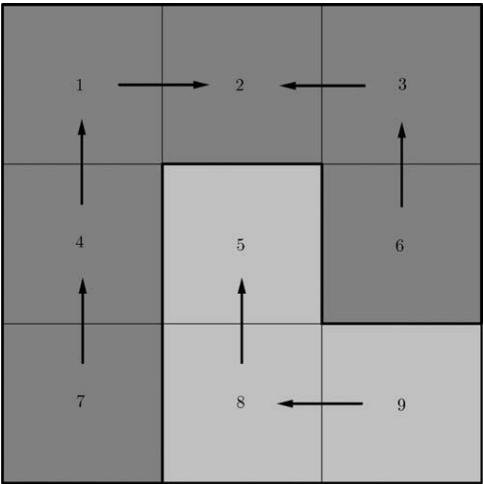
Figure 6. Optimal solution of the DFJ-*p*-regions.

improvement of 10% in the convergence rate and signifies an important achievement within the context of *NP*-hard problems.

To compare the results in terms of performance, the following indexes were calculated:

The *Speedup* is the relationship between the execution time required for the complete formulation, $CPUtime_C$, and the execution time required for the DFJ-*p*-regions algorithm, $CPUtime_{DFJ}$ (see equation (12)). Values greater than one denote that the DFJ-*p*-regions algorithm outperforms the complete formulation.

$$Speedup = \frac{CPUtime_C}{CPUtime_{DFJ}} \tag{12}$$

The best integer solution (*BIS*) is the best objective found by the end of the execution time, and the best bound (*BB*) is the best solution to the relaxed model found through a branch-and-bound search by the end of the execution time. The differences between the *BIS* and *BB* is calculated as percentages as follows:

$$\%\Delta BIS = \frac{(BIS_T - BIS_{DFJ})}{BIS_T} \times 100\% \tag{13}$$

Table 1. Experiment Design

Parameters	Values
n	16, 25, 36, 49, 64
p	3, 7, 11
ρ	0, 0.1, 0.3, 0.5, 0.7, 0.9
Instances per combination of [*n*, *p*, *ρ*]	30

Table 2. Performance Over 2,700 Instances

		DFJ-*p*-regions algorithm	
		Converged	Not converged
Complete	Converged	1,118	0
	Not converged	270	1,312

$$\%\Delta BB = \frac{(BB_{DFJ} - BB_T)}{BB_{DFJ}} \times 100\% \tag{14}$$

A positive value for $\%\Delta BIS$ denotes that the DFJ-*p*-regions algorithm has found a better feasible solution to the problem at hand relative to the complete formulation. Similarly, a positive $\%\Delta BB$ denotes that the DFJ-*p*-regions algorithm has found a better (i.e., higher) lower bound, implying that it is closer to converging than the complete formulation.

In the following subsections, we apply the indexes above depending on the convergence state of the two solution methods.[4] According to Table 2, there are three outcomes: both methods have converged (1,118 instances), only the DFJ-*p*-regions algorithm has converged (270 instances) and none of the methods have converged (1,312 instances).

Both methods have converged

Table 3 shows instances where both methods have converged, differentiating between *n*, *p*, and ρ. As expected, the higher the number of areas is, *n*, the lower the number of instances solved optimally is. The impact of parameter *p* is also as expected: the larger the *p* is, the smaller the average size of the clusters is, thus implying less complicated trees and fewer subtour possibilities.

Table 3. Number of Instances Where Both Methods Converged

n	*p*	ρ					
		0.0	0.1	0.3	0.5	0.7	0.9
16	3	30	30	30	30	30	30
	7	30	30	30	30	30	30
	11	30	30	30	30	30	30
25	3	2	2	4	3	3	1
	7	30	30	30	30	30	30
	11	30	30	30	30	30	30
36	3	0	0	0	0	0	0
	7	4	2	3	2	2	7
	11	29	30	30	29	30	30
49	3	0	0	0	0	0	0
	7	0	0	0	0	0	0
	11	0	0	1	1	1	2
64	3	0	0	0	0	0	0
	7	0	0	0	0	0	0
	11	0	0	0	0	0	0

Table 4. Geometric Mean of Speedup When Both Methods Converged

n	p	n/p	ρ					
			0.0	0.1	0.3	0.5	0.7	0.9
16	3	5.33	**2.58**	**2.44**	**2.08**	**3.20**	**3.21**	**5.64**
	7	2.29	0.40	0.30	0.32	0.87	0.65	0.86
	11	1.45	0.58	0.47	0.63	0.48	0.46	0.46
25	3	8.33	**4.99**	**8.45**	**4.18**	**5.92**	**2.76**	**8.37**
	7	3.57	**4.61**	**3.04**	**1.63**	**1.55**	**2.21**	**2.31**
	11	2.27	0.52	0.43	0.36	0.57	0.64	0.68
36	3	12.0	–	–	–	–	–	–
	7	5.14	**10.46**	**26.64**	**20.14**	**13.65**	**71.99**	**14.59**
	11	3.27	**5.08**	**4.98**	**4.50**	**4.28**	**3.79**	**3.20**
49	3	16.3	–	–	–	–	–	–
	7	7.00	–	–	–	–	–	–
	11	4.45	–	–	3.51	1.04	4.17	7.64

Because both methods found the optimal solution, we focus on comparing execution times. The geometric mean of *Speedup* is calculated among those instances that fall into this category. Table 4 shows this measure differentiated by n, p, and ρ; where the values in bold highlight the cases in which, on average, the DFJ-p-regions algorithm converged faster than the complete formulation. The results show that the DFJ-p-regions algorithm overcomes the complete formulation when the ratio n/p is greater than 3. With respect to the spatial dependence parameter ρ, no particular pattern was identified.

Only the DFJ-p-regions algorithm converged

Table 5 shows the number of instances where only the DFJ-p-regions algorithm converged, differentiated by n, p, and ρ. In this case, our interest is to determine how far the complete formulation was from the optimal solution. Tables 6 and 7 provide the average %ΔBIS and %ΔBB differentiated by n, p, and ρ. Values close to zero in Table 7 indicate that the complete formulation found an integer solution that is very close to the optimal, but Table 6 shows us that it is not close enough to satisfy the optimality criterion (values shown in Table 6 fall between 5.45% and 87.08%). None of the tables shows a clear effect of parameter ρ.

It is important to note that, to the best of our knowledge, this is the first time that an instance of the p-regions problem with 49 areas has been solved to optimality in less than 3 h using exact optimization techniques.

No method converged

Table 8 shows the instances where no method converged, differentiated by n, p, and ρ. The results show that spatial clustering remains a problem that is very difficult to solve optimally. Neither the complete formulation nor the DFJ-p-regions algorithm were able to solve a single instance with $n = 64$. In general, it seems that the higher the value of ρ, the easier to problem is to solve.

Table 9 presents the average % ΔBIS differentiated by n, p, and ρ. The symbol † shown in the table indicates that there was not data to compute % ΔBIS. The predominance of negative

Table 5. Number of Instances Where Only the DFJ-*p*-Regions Algorithm Converged

n	*p*	*ρ*					
		0.0	0.1	0.3	0.5	0.7	0.9
16	3	0	0	0	0	0	0
	7	0	0	0	0	0	0
	11	0	0	0	0	0	0
25	3	11	14	19	14	19	20
	7	0	0	0	0	0	0
	11	0	0	0	0	0	0
36	3	–	–	–	–	–	–
	7	15	18	18	21	26	21
	11	1	0	0	1	0	0
49	3	0	0	0	0	0	0
	7	0	0	0	0	0	0
	11	2	2	7	8	9	24

values indicate that, after 3 h, the complete formulation found a better integer solution than the DFJ-*p*-regions algorithm. We also found that for a fixed number or areas, *n*, the difference between the two approaches tends to increase, in favor of the complete formulation, as *p* decreases.

Disentangling the relationship between the DFJ-*p*-regions and rho

The computational experiments in section "Computational experiment" did not show a convincing evidence in favor of the direct relationship between the level of spatial autocorrelation and the benefits derived from the DFJ-*p*-regions approach. At this point, we know that the benefits of the DFJ-*p*-regions algorithm appear when $n/p \geq 3$, but the results did not show that

Table 6. Average %ΔBIS When Only the DFJ-*p*-Regions Algorithm Converged

n	*p*	*ρ*					
		0.0	0.1	0.3	0.5	0.7	0.9
16	3	–	–	–	–	–	–
	7	–	–	–	–	–	–
	11	–	–	–	–	–	–
25	3	0.00048%	0.00052%	0.21409%	0.00050%	0.00055%	0.00028%
	7	–	–	–	–	–	–
	11	–	–	–	–	–	–
36	3	–	–	–	–	–	–
	7	0.03590%	0.04871%	0.20044%	0.30229%	0.10595%	0.00027%
	11	0.00033%	–	–	0.00025%	–	–
49	3	–	–	–	–	–	–
	7	–	–	–	–	–	–
	11	0.00016%	0.00016%	0.00016%	0.00016%	0.00014%	0.14795%

Table 7. Average %ΔBB When Only the DFJ-p-Regions Algorithm Converged

n	p	ρ					
		0.0	0.1	0.3	0.5	0.7	0.9
16	3	–	–	–	–	–	–
	7	–	–	–	–	–	–
	11	–	–	–	–	–	–
25	3	53.44%	63.36%	62.14%	70.96%	87.08%	84.09%
	7	–	–	–	–	–	–
	11	–	–	–	–	–	–
36	3	–	–	–	–	–	–
	7	24.47%	31.96%	28.58%	38.58%	34.67%	38.05%
	11	5.45%	–	–	6.38%	–	–
49	3	–	–	–	–	–	–
	7	–	–	–	–	–	–
	11	9.07%	27.13%	20.62%	37.36%	31.94%	31.91%

spatial patterns with high spatial autocorrelation require a much lower number of subtour breaking constraints. These results appear counterintuitive in the light of our initial thoughts: in the presence of high spatial autocorrelation, the spatially constrained clusters will tend to appear naturally, and the relaxed p-regions will require a few number of subtour breaking constraints to reach optimality. We present in this section a last computational experiment with which we want to provide evidence in support of the following hypothesis: The maximal benefits from the DFJ-p-regions algorithm are obtained when the parameter p matches the number of natural clusters in the spatial pattern of the aggregation variable, p^*, that is, when $p = p^*$.

Table 8. Number of Instances Where No Method Converged

n	p	ρ					
		0.0	0.1	0.3	0.5	0.7	0.9
16	3	0	0	0	0	0	0
	7	0	0	0	0	0	0
	11	0	0	0	0	0	0
25	3	17	14	7	13	8	9
	7	0	0	0	0	0	0
	11	0	0	0	0	0	0
36	3	30	30	30	30	30	30
	7	11	10	9	7	2	2
	11	0	0	0	0	0	0
49	3	30	30	30	30	30	30
	7	30	30	30	30	30	30
	11	28	28	22	21	20	4
64	3	30	30	30	30	30	30
	7	30	30	30	30	30	30
	11	30	30	30	30	30	30

Table 9. Average of % ΔBIS When No Method Converged

n	p	ρ					
		0.0	0.1	0.3	0.5	0.7	0.9
16	3	–	–	–	–	–	–
	7	–	–	–	–	–	–
	11	–	–	–	–	–	–
25	3	0.32%	0.01%	0.68%	−0.50%	0.10%	0.02%
	7	–	–	–	–	–	–
	11	–	–	–	–	–	–
36	3	−12.73%	−6.18%	−6.67%	−7.08%	−9.47%	−11.25%
	7	0.22%	−0.16%	1.00%	−0.81%	1.97%	0.00%
	11	–	–	–	–	–	–
49	3	−101.35%	−2.36%	−38.67%	−59.24%	−71.89%	−4.54%
	7	−13.87%	−12.80%	−15.00%	−11.88%	−14.42%	−5.90%
	11	−0.67%	−1.32%	−1.11%	−0.06%	−1.28%	−0.34%
64	3	†	†	†	†	†	†
	7	−43.22%	−48.38%	−61.43%	−107.70%	−76.38%	−164.90%
	11	−11.17%	−13.61%	−17.68%	−100.07%	−17.50%	−29.56%

For this second experiment, a set of 30 regular lattices of size $n = 36$ were generated, each with a spatial pattern designed in such a way that each instance contains $p* = 7$ natural clusters. The areas within each cluster have a random value from an uniform distribution. The minimum and maximum limits of the uniform distribution change from one cluster to another so that each cluster contains values that do not overlap with values in other cluster. Fig. 7 presents an example of the spatial distribution obtained with this strategy. The seven clusters can be easily identified in the spatial pattern and the values within each cluster present small variations.

Our experiment consists of solving each instance using both approaches, the complete *p*-regions formulation and the DFJ-*p*-regions algorithm. In both cases we used three different values for *p*: {3, 7, 11}. Then, using the instances that where solved to optimality, we measured the execution times and the number of subtour elimination constraints required by the DFJ-*p*-region algorithm to terminate. We compared these results with the results from the experiment described in section "Computational experiment" with $n = 36$, *p*: {3, 7, 11}, and $\rho = 0$. Tables 10 and 11 present a summary of the results in terms of execution time (seconds) discriminated by the value of *p*. The columns labeled "Random" correspond to the instances generated with $\rho = 0$, and the columns labeled "Clustered" correspond to the instances that contain $p* = 7$ natural clusters. Even though the *p*-regions formulation is positively affected by the presence of natural clusters (see Table 10), the DFJ-*p*-regions algorithm significantly outperforms its counterpart both in terms of number of instances solved to optimality and computational cost, in particular when $p = p*$ (see Table 11). We also found that when $p* = p = 7$, the variance of the execution times is much lower in the DFJ-*p*-regions algorithm (with execution times between 6.82 and 16.25 s) than in the complete *p*-regions formulation (with execution times between 62.07 and 6,694.98 s).

Finally, the number of subtour elimination constraints required by the DFJ-*p*-regions algorithm is summarized in Table 12. The figures show two different effects: The first effect is related
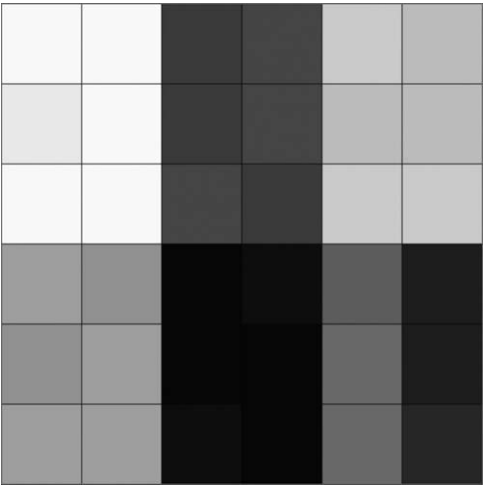
Figure 7. Example of instance with a spatial pattern with $p* = 7$ natural clusters.

Table 10. Execution Time Statistics for the Complete p-Regions Formulation

	$p = 3$		$p* = p = 7$		$p = 11$	
	Random	Clustered	Random	Clustered	Random	Clustered
Observations	0	0	4	28	29	30
Minimum	†	†	2,242.99	62.07	5.78	4.33
First quartile	†	†	2,398.26	75.60	82.21	6.82
Median	†	†	**3,389.78**	**96.63**	228.63	11.06
Mean	†	†	4,170.71	435.13	719.08	15.27
Third quartile	†	†	3,389.78	96.63	228.63	11.06
Maximum	†	†	7,660.30	6,694.98	5,396.95	51.14

Table 11. Execution Time Statistics for the DFJ p-Regions Algorithm

	$p = 3$		$p* = p = 7$		$p = 11$	
	Random	Clustered	Random	Clustered	Random	Clustered
Observations	0	15	19	30	30	30
Minimum	†	1,290.46	171.27	6.82	13.47	5.89
1st quartile	†	3,409.67	436.89	9.39	25.01	9.62
Median	†	4,665.22	**1,881.85**	**11.25**	57.06	12.93
Mean	†	4,893.66	3,198.48	11.16	65.98	12.33
3rd quartile	†	4,665.22	1,881.85	11.25	57.06	12.93
Maximum	†	8,755.03	10,688.46	16.25	207.05	21.58

Table 12. Subtour Elimination Constraints Added by the DFJ-*p*-Regions Algorithm

| | $p = 3$ | | $p^* = p = 7$ | | $p = 11$ | |
	Random	Clustered	Random	Clustered	Random	Clustered
Observations	0	15	19	30	30	30
Minimum	†	56.00	10.00	0.00	0.00	0.00
First quartile	†	84.50	48.00	0.00	0.00	0.00
Median	†	96.00	86.00	0.00	2.00	0.00
Mean	†	105.93	86.00	18.50	3.83	2.57
Third quartile	†	96.00	86.00	0.00	2.00	0.00
Maximum	†	176.00	190.00	152.00	16.00	74.00

to the low number of subtour elimination constraints as a consequence of a low ratio n/p. When $p = 11$ the ratio n/p is 3.3, which implies small regions and therefore fewer possibilities for subtours. In this case, both the random and the clustered patters are benefited. The second effect confirms the hypothesis formulated at the beginning of this section; when $p^* = p = 7$ the clustered pattern requires much less subtour elimination constraints than its counterpart. This last experiment lead us to the following conclusion: the greatest benefits of using the DFJ-*p*-regions algorithm emerge when (1) the instance exhibits a natural clustered pattern, and (2) the value of the parameter p is close to the number of natural clusters in the instance, p^*.

Conclusions

In this article, we implemented an extensive computational experiment to test the performace of the subtour elimination constraints proposed by Duque, Church, and Middleton (2011) for the *p*-regions problem (the DFJ-*p*-regions algorithm). We placed special attention to the impact of the levels of spatial autocorrelation of the aggregation variable on computational time.

The results show that the higher the spatial dependence, the lower the number of subtour breaking constraints required to optimally solve the problem and, thus, results in shorter execution time. This is because clustered spatial patterns tend to create spatially contiguous regions. We found that the greatest benefits are obtained in two situations: (1) when $n/p \geq 3$; and (2) when the parameter p is close to the number of clusters in the spatial pattern of the aggregation variable (p^*).

In this article we increased the convergence rate by roughly 12%, allowing us to optimally solve, with exact optimization, the largest instance of the *p*-region-problem from its first appearance in Duque, Church, and Middleton (2011). This constitutes an important step forward in the simplification of this *NP*-hard problem. In the future, we plan to explore other decomposition techniques that allow us to optimally solve larger instances of the problem (e.g., Lagrangian relaxation, Benders decomposition, and the so-called Dantzig–Wolfe decomposition, which is also known as column generation).

Notes

1 In this context, the term "exhaustive clustering" means that each area is assigned to one and only one region.

2 According to Duque et al. (2007), previous optimization models for the spatially constrained clustering did not include an explicit set of contiguity constraints: the linear optimization model proposed by Zoltners and Sinha (1983) requires prior selection of the set of areas that will be the region centers, which can lead to suboptimal solutions within the context of spatial clustering. Macmillan and Pierce (1994) formulated a nonlinear optimization problem that requires the recursive powering of the binary matrix of adjacencies until the resulting matrix does not exhibit any zero terms. Garfinkel and Nemhauser (1970) proposed a linear optimization model that requires the prior enumeration of all the feasible regions as input; however, because of the $NP-$ hard condition of this problem, it may be impossible to satisfy this requirement as the size of the problem instances increase. Other mathematical models proposed in the context of electoral or school districting do not incorporate an explicit constraint for guaranteeing spatial contiguity (Weaver and Hess 1963; Hess et al. 1965; Kaiser 1996; Mills 1967; Hess and Samuels 1971; Helbig, Orr, and Roediger 1972; George, Lamar, and Wallace 1997; Bacao, Lobo, and Painho 2005; Wise, Haining, and Ma 1997). These studies adapt location-allocation models to induce spatial contiguity by maximizing the compactness of the shape of the regions, which is a desirable regional characteristic in the field of districting. For this approach, the spatial contiguity of the final solution must be checked a posteriori.

3 The TSP. The TSP is a classical problem in the field of computer science. In its simplest form, the TSP is the problem that faces a salesperson who must visit *n* cities and thus needs to find the tour with the minimum total travel distance that visits each city exactly once and returns to the point of origin. A major challenge in solving the TSP is to ensure that the solution consists of a single tour that visits all the cities instead of multiple shorter tours (i., e., subtours) the union of which covers all of the cities. The constraints that prevents these subtours from appearing in the final solution are known as the so-called subtour elimination constraints.

4 The reader is referred to the supplementary material to see the generated instances and the database of the results presented here.

References

Anselin, L. (2005). *Exploring Spatial Data with GeoDa: A Work- Book*. Chicago, IL: Center for Spatially Integrated Social Science, University of Illinois at Urbana-Champaign.

Bacao, F., V. Lobo, and M. Painho. (2005). "Applying Genetic Algorithms to Zone Design." *Soft Computing* 9, 341–48.

Blais, M., S. Lapierre, and G. Laporte. (2003). "Solving a Home-Care Districting Problem in an Urban Setting." *Journal of the Operational Research Society* 54(11), 1141–47.

Browdy, M. (1990). "Simulated Annealing: An Improved Computer Model for Political Redistricting." *Yale Law & Policy Review* 8(1), 163–79.

Byfuglien, J., and A. Nordgard. (1973). "Region-Building: A Comparison of Methods." *Norwegian Journal of Geography* 27, 127–51.

Caro, F., T. Shirabe, M. Guignard, and A. Weintraub. (2004). "School Redistricting: Embedding GIS Tools with Integer Programming." *Journal of the Operational Research Society* 55(8), 836–49.

Church, R., and T. Cova. (2000). "Mapping Evacuation Risk on Transportation Networks Using a Spatial Optimization Model." *Transportation Research Part C: Emerging Technologies* 8(1), 321–36.

Cliff, A. D., P. Hagget, and J. K. Ord. (1975). *Elements of Spatial Structure : A Quantitative Approach*. New York: Cambridge University Press, Cambridge [Eng.].

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

Current, J., C. ReVelle, and J. Cohon. (1984). "The Shortest Covering Path Problem. An Application of Locational Constraints to Network Design." *Journal of Regional Science* 24(2), 161–83.

D'amico, S., S. Wang, R. Batta, and C. Rump. (2002). "A Simulated Annealing Approach to Police Districting Design." *Computer & Operations Research* 29(6), 667–84.

Dantzig, G., R. Fulkerson, and S. Johnson. (1954). "Solution of a Large-Scale Traveling-Salesman Problem." *Journal of the Operations Research Society of America* 2(4), 393–410.

Duque, J. C., L. Anselin, and S. Rey. (2012). "The Max-*p*-Regions Problem." *Journal of Regional Science* 52(3), 397–419.

Duque, J. C., M. Artís, and R. Ramos. (2006). "The Ecological Fallacy in a Time Series Context: Evidence from Spanish Regional Unemployment Rates." *Journal of Geographical Systems* 8(4), 391–410.

Duque, J. C., R. Church, and R. Middleton. (2011). "The p-Regions Problem." *Geographical Analysis* 43(1), 104–26.

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 Constraint." *Psychometrika* 47(4), 413–26.

Garfinkel, R., and G. Nemhauser. (1970). "Optimal Political Districting by Implicit Enumeration Techniques." *Management Science* 16(8), 495–508.

George, J., B. Lamar, and C. Wallace. (1997). "Political District Determination Using Large-Scale Network Optimization." *Socio-Economic Planning Sciences* 31, 11–28.

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

Grötschel, M., and G. Nemhauser. (2008). "George Dantzig's Contributions to Integer Programming." *Discrete Optimization* 5(2), 168–73.

Hagberg, A. A., D. A. Schult, and P. J. Swart. (2008). "Exploring Network Structure, Dynamics, and Function Using NetworkX." In *Proceedings of the Seventh Python in Science Conference (SciPy2008)*, 11–15, edited by G. Varoquaux, T. Vaught, J. Millman. Pasadena, CA. Available online at http://math.lanl.gov/~hagberg/Papers/hagberg-2008-exploring.pdf. Source of the cite http://math.lanl.gov/~hagberg/Publications/hagberg-2008-exploring.shtml

Hansen, P., B. Jaumard, C. Meyer, B. Simeone, and V. Doring. (2003). "Maximum Split Clustering Under Connectivity Constraints." *Journal of Classification* 20, 143–80.

Helbig, R., P. Orr, and R. Roediger. (1972). "Political Redistricting by Computer." *Communications of the ACM* 15, 735–41.

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

Hess, S., J. Weaver, H. Siegfeldt, J. Whelan, and P. Zitlau. (1965). "Nonpartisan Political Redistricting by Computer." *Operations Research* 13(6), 998–1006.

Kaiser, H. (1996). "An Objective Method for Establishing Legislative Districts." *Midwest Journal of Political Science* 10, 200–13.

Keane, M. (1975). "The Size of the Region-Building Problem." *Environment and Planning A* 7(5), 575–77.

Kim, H., Y. Chun, and K. Kim. (2013). "Delimitation of Functional Regions Using a p-Regions Problem Approach." *International Regional Science Review* 38(3), 1–29.

Laura, J., L. Wenwen, S. Rey, and L. Anselin. (2015). "Parallelization of a Regionalization Heuristic in Distributed Computing Platforms: A Case Study of Parallel-p-Compact-Regions Problem." *International Journal of Geographical Information Science* 29(4), 536–55.

Lefkovitch, L. (1980). "Conditional Clustering." *Biometrics* 36(1), 43–58.

Legendre, P. (1987). "Constrained Clustering." In *Developments in Numerical Ecology. NATO ASI Series*, Vol. G 14, 289–307, edited by P. Legendre and L. Legendre. Berlin: Springer.

Li, W., R. Church, and M. Goodchild. (2014a). "An Extendable Heuristic Framework to Solve the p-Compact-Regions Problem for Urban Economic Modeling." *Computers, Environment and Urban Systems* 43, 1–13.

Li, W., R. Church, and M. Goodchild. (2014b). "The p-Compact-Regions Problem." *Geographical Analysis* 46(3), 250–73.

Macmillan, W. (2001). "Redistricting in a GIS Environment: An Optimization Algorithm Using Switching-Points." *Journal of Geographical Systems* 3, 167–80.

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. S. Fotheringham and P. Rogerson. London: Taylor & Francis.

Maravalle, M., and B. Simeone. (1995). "A Spanning Tree Heuristic for Regional Clustering." *Communications in Statistics-Theory and Methods* 24, 625–39.

Miller, C. E., A. W. Tucker, and A. Zemlin. (1960). "Integer Programming Formulation of Traveling Salesman Problems." *Journal of the ACM* 7, 326–29.

Mills, G. (1967). "The Determination of Local Government Electoral Boundaries." *Operational Research Quarterly* 18, 243–55.

Murtagh, F. (1985). "A Survey of Algorithms for Contiguity-Constrained Clustering and Related Problems." *The Computer Journal* 28(1), 82–8.

Murtagh, F. (1992). "Contiguity-Constrained Clustering for Image Analysis." *Pattern Recognition Letters* 13, 677–83.

Nuutila, E., and E. Soisalon-Soininen. (1994). "On Finding the Strongly Connected Components in a Directed Graph." *Information Processing Letters* 49(1), 9–14.

Openshaw, S., and L. Rao. (1995). "Algorithms for Reengineering 1991 Census Geography." *Environment and Planning A* 27(3), 425–46.

Openshaw, S., and C. Wymer. (1995). "Classifying and Regionalizing Census Data." In *Census Users Handbook*, 239–68, edited by S. Openshaw. Cambridge, UK: GeoInformation International.

Orman, A., and H. Williams. (2006). "A Survey of Different Integer Programming Formulations of the Travelling Salesman Problem. Optimisation, Economics and Financial Analysis." *Advances in Computational Management Science* 9, 93–106.

Ricca, F., and B. Simeone. (2008). "Local Search Algorithms for Political Districting." *European Journal of Operational Research* 189, 1409–26.

She, B., J. C. Duque, and X. Ye. (2017). "The Network-Max-p-Regions Model." *International Journal of Geographical Information Science* 31(5), 1–20.

Shirabe, T. (2005). "A Model of Contiguity for Spatial Unit Allocation." *Geographical Analysis* 37(1), 2–16.

Tarjan, R. (1972). "Depth-First Search and Linear Graph Algorithms." *SIAM Journal on Computing* 1(2), 146–60.

Vickrey, W. (1961). "On the Prevention of Gerrymandering." *Political Science Quarterly* 76, 105–10.

Weaver, J., and S. Hess. (1963). "A Procedure for Nonpartisan Districting: Development of Computer Techniques." *The Yale Law Journal* 73(2), 288–308.

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

Wise, S., R. Haining, and J. Ma. (1997). "Regionalisation Tools for Exploratory Spatial Analysis of Health Data." In *Recent Developments in Spatial Analysis: Spatial Statistics, Behavioural Modelling, and Computational Intelligence*, 83–100, edited by Manfred M. Fischer and Arthur Getis. New York: Springer.

Ye, X., and S. Rey. (2013). "A Framework for Exploratory Space-Time Analysis of Economic Data." *The Annals of Regional Science* 50, 315–39.

Zoltners, A., and P. Sinha. (1983). "Sales Territory Alignment: A Review and Model." *Management Science* 29(11), 1237–56.