J Geogr Syst (2011) 13:355–372
DOI 10.1007/s10109-010-0137-1

ORIGINAL ARTICLE

A computationally efficient method for delineating irregularly shaped spatial clusters

Juan C. Duque · Jared Aldstadt · Ermilson Velasquez · Jose L. Franco · Alejandro Betancourt

Received: 24 March 2010 / Accepted: 2 September 2010 / Published online: 29 September 2010

Abstract In this paper, we present an efficiency improvement for the algorithm called AMOEBA, A Multidirectional Optimum Ecotope-Based Algorithm, devised by Aldstadt and Getis (Geogr Anal 38(4):327–343, 2006). AMOEBA embeds a local spatial autocorrelation statistic in an iterative procedure in order to identify spatial clusters (ecotopes) of related spatial units. We provide an analysis of the computational complexity of the original AMOEBA and develop an alternative formulation that reduces computational time without losing optimality. Empirical evidence is provided using georeferenced socio-demographic data in Accra, Ghana.

Keywords AMOEBA · Cluster detection · Local G statistic · Ecotope

JEL Classification C02 mathematical methods · C4 econometric and statistical methods: special topics

J. C. Duque (✉)
Research in Spatial Economics (RISE-group), Department of Economics,
EAFIT University, Carrera 49 7 Sur-50, Medellin, Colombia
e-mail: jduquec1@eafit.edu.co

J. Aldstadt
Department of Geography, University at Buffalo,
117 Wilkeson Quad, Buffalo, NY 14261-0055, USA
e-mail: geojared@buffalo.edu

E. Velasquez · J. L. Franco · A. Betancourt
Research in Spatial Economics (RISE-group), Department of Fundamental Sciences,
EAFIT University, Carrera 49 7 Sur-50, Medellin, Colombia

1 Introduction

Since the early 1990s, spatial analysts shifted their focus toward the study of the non-stationarity of spatial relationships (Getis and Ord 1992; Anselin 1995; Ord and Getis 1995). Ignoring differences in distribution across space by solely using global statistics can lead to the ecological inference fallacy (Robinson 1950). The need for identifying local irregularities in spatial data emerged, in part, as a consequence of the rise of highly disaggregated spatial data together with an increase in computational capabilities (Fotheringham et al. 2000).

Spatial clusters are one of the forms of non-stationarity in space. They can be defined as "a geographically bounded group of occurrences of sufficient size and concentration to be unlikely to have occurred by chance" (Knox 1989). Spatial clusters identification techniques are applied in many areas of inquiry, and are most frequently used in epidemiology and criminology research (Aldstadt 2010).[1]

One of the most recent algorithms developed for this purpose is the AMOEBA (A Multidirectional Optimal Ecotope-Based Algorithm) (Aldstadt and Getis 2006). In brief, this algorithm starts with an initial area to which neighboring areas are iteratively attached until the addition of any neighboring area fails to increase the magnitude of the local G_i^* of Getis and Ord (1992) and Ord and Getis (1995). The resulting region is considered an ecotope. This procedure is executed for all areas, and final ecotopes are defined after resolving overlaps and asserting non-randomness.

Most existing cluster identification techniques make the implicit assumption that clusters are circular and compact regions. This assumption may be invalid. Spatial clustering of mapped variables may result from a large set of related phenomena. The natural environment, the built environment, and a complex set of human interactions are responsible for the spatial aggregation of extreme values. There is no reason to believe that the resulting spatial patterns should result in circular hotspots.

Assuming that clusters are circular may lead to incorrect cluster size and false-positive determinations (Jacquez 2009). In simulation studies, the circular spatial scan statistic tends to detect clusters that are larger than the true simulated cluster (Tango and Takahashi 2005; Aldstadt and Getis 2006). Circular clusters of high value also may include spatial units with below-average values. This situation arises when peripheral low-value units are absorbed into a high-value cluster or worsen when disjointed units with high values are captured at the edges of a single circular cluster. In the latter case, a circular cluster of high values may be centered on a low-valued unit. The AMOEBA does not permit the inclusion of low-value spatial units in identified clusters of high values and vice versa.

The computational complexity of the AMOEBA implies that the time needed to solve a problem will increase substantially as its size increases.[2] Also, computational experiments show that the combinatorial approach of this algorithm

[1] A collection of statistical tests for cluster detection is available in a software package named *GeoSurveillance*. See Yamada et al. (2009) for more information about this software.

[2] In general, the problem of spatial clustering is related to a family of problems that are classified as N-P hard (Wu et al. 2007; Gaudart et al. 2005). The combinatorial complexity of this type of problems has led researchers to primarily focus on the development of algorithmic solutions.

makes execution times highly sensitive to the size of the clusters. These two characteristics make AMOEBA difficult, if not impossible, to apply to large problems.

In this paper, we propose an alternative formulation for the AMOEBA that significantly reduces its computational complexity without losing optimality. The main characteristic of our approach is that we take advantage of some properties of both the empirical distribution of the variable and the formulation of the G_i^* statistic to guide the algorithm toward an optimal solution, avoiding the need for combinatorial evaluations of the solution space, which are exceedingly costly from a computational perspective. This new formulation makes possible the application of AMOEBA to problem instances that involve very large numbers of areas.

The rest of the article is organized as follows: first, the original AMOEBA and its supporting concepts are discussed. Next, the proposed variant is presented, and a proof is provided that shows the equivalence of both algorithms. Afterward, experimental results are presented that show performance comparisons between both algorithms. Then, a brief empirical application based on data from Accra (Ghana) is presented. The final section concludes.

2 Preliminaries

The G_i^* statistic measures the association between the values of an attribute at a given area and its neighbors. In other words, it measures the level of clustering of an attribute x around an area. For a given area i, G_i^* is defined as follows:

$$G_i^* = \frac{\sum_{j=1}^{N} w_{ij}x_j - \bar{x}\sum_{j=1}^{N} w_{ij}}{S\sqrt{\frac{N\sum_{j=1}^{N} w_{ij}^2 - (\sum_{j=1}^{N} w_{ij})^2}{N-1}}} \tag{1}$$

where elements w_{ij} are the spatial weights that reflect the proximity between areas i and j, N is the number of areas, x_j is the value of the attribute at area j, $\bar{x}$ represents the mean of the x values, and

$$S = \sqrt{\frac{\sum_{j=1}^{N} x_j^2}{N} - \bar{x}^2}.$$

Although the G_i^* statistic can be calculated using different representations of the weights matrix (i.e., w_{ij} can be a binary variable or non-binary function, such as, inverse distance), within the context of the AMOEBA the w_{ij} values are restricted to be binary, with $w_{ij} = 1$ for those areas i and j included in the same ecotope.

The numerator of the statistic can be seen as the sum of the divergences with respect to the mean of attribute x. This result occurs because, if we take only the x_j where $w_{ij} = 1$, we are left with

$$\sum_{j=1}^{N} w_{ij}x_j - \bar{x}\sum_{j=1}^{N} w_{ij} = \sum_{j=1}^{N} (x_j - \bar{x}).$$

The denominator is a strictly positive number that varies according to the number of areas in the analyzed region.

G_i^* is asymptotically distributed as a normal $N(0, 1)$. A positive (negative) and statistically significant value of this statistic indicates the presence of a cluster of high (low) values of attribute x around area i. Thus, Aldstadt and Getis' AMOEBA identifies high-valued, or low-valued, ecotopes by looking for subsets of geographically connected areas with a high absolute value of the G_i^* statistic.

The algorithm starts by taking an area i and computing its G_i^* value. When performed for a single unit, this amounts to calculating the standard score of the value for unit i. A positive (negative) value of the statistic indicates that the value of the attribute at area i is greater (lower) than the mean. Next, an exhaustive evaluation is carried out by calculating the G_i^* statistic for each region that includes the initial unit i and every possible combination of neighboring areas of area i. The set that results in the maximum absolute G_i^* value with the same sign as the G_i^* value for unit i alone is recorded. If this G_i^* value for unit i and a set of its neighbors is greater than the G_i^* value for unit i alone; then, this set becomes the ecotope. The procedure continues by examining each possible combination of areas contiguous to the newly identified ecotope. This iterative process of identifying sets of neighboring areas that maximize the value of G_i^* is repeated until it is not possible to increase the absolute value of the G_i^* statistic by addition of a set of contiguous units. The steps of growing an ecotope from an initial area are summarized in Algorithm 1, which we call exhaustive AMOEBA.

After examining each area as the initial seed of an ecotope, the algorithm keeps the non-overlapping ecotopes with the highest G_i^* values. For each remaining ecotope, Aldstadt and Getis (2006) propose to perform a Monte Carlo-type permutation test to calculate the statistical significance of each ecotope. This test performs a large number of random spatial permutations for the attribute x and records the times that the sum of the attribute values in the ecotope is larger than the sum of the values in the original ecotope. The p-value for the ecotopes is then calculated as the ratio between this number plus one and the total number of permutations plus one. Those ecotopes with p-values below some predesignated level of significance are considered as true clusters.

The problem with the original formulation of AMOEBA lies in the extensive evaluation that must be realized to optimize G_i^* at each step. AMOEBA computes the statistic for all combinations of non-excluded neighbors of the ecotope. Thus, if a given ecotope has c neighbors, the number of iterations required to optimize the statistic is

$$\sum_{i=1}^{c} \binom{c}{i} = \sum_{i=1}^{c} \frac{c!}{i!(c-i)!}.$$

Needless to say, this number can become very large even for relatively small numbers of neighbors. For example, an ecotope with 20 neighbors requires 1,048,575 iterations to fully explore the search space.

Algorithm 1 Exhaustive AMOEBA

1: $R = a_0$ // *a_0 is the seed area for this iteration*
2: $G_i^{*opt} = G_i^*(R)$ // *The function G_i^* calculates* (1) *for the given region*
3: N = Neighbors (R) // *Neighbors gives a list of all areas contiguous to R*
4: T = [] // *Begins an empty list of discarded areas*
5: **if** $G_i^{*opt} \geq 0$ **then**
6: **do**
7: $G_i^{*aux} = G_i^{*opt}$
8: N = setDifference (N, T) // *Removes all areas in T from N*
9: C = Combination (N // *Combinations gives a list of all possible combinations of all sizes of the given list of areas*
10: C_{opt} = [] // *Sets C_{opt} to begin as an empty list*
11: **for** i = 1 to length (C) **do**
12: **if** $G_i^*(R \cup C(i)) > G_i^{*opt}$ **then**
13: $C_{opt} = C(i)$
14: $G_i^{*opt} = G_i^*(R \cup C(i))$
15: **end if**
16: **end for**
17: $R = R \cup C_{opt}$
18: N = Neighbors (R)
19: **while** $G_i^{*aux} \neq G_i^{*opt}$
20: **return** R
21: **else**
22: **do**
23: $G_i^{*aux} = G_i^{*opt}$
24: N = setDifference (N, T)
25: C = Combinations (N)
26: C_{opt} = [] // *Sets C_{opt} to begin as an empty list*
27: **for** i = 1 to length (C)
28: **if** $G_i^*(R \cup C(i)) < G_i^{*opt}$ **then**
29: $C_{opt} = C(i)$
30: $G_i^{*opt} = G_i^*(R \cup C(i))$
31: **end if**
32: **end for**
33: $R = R \cup C_{opt}$
34: N = Neighbors (R)
35: **while** $G_i^{*aux} \neq G_i^{*opt}$
36: **return** R
37: **end if**

3 The improved algorithm

To introduce the improved method, let us revisit the formulation of the G_i^* statistic as shown in (1). Note that a region or ecotope is essentially a geographically linked group of areas. From this perspective, it is only natural to define a region as a spatially contiguous set of areas. That being said, in the AMOEBA, the G_i^* statistic is essentially a function that assigns real values to sets of spatially contiguous areas.

Suppose we run AMOEBA on a study region with N areas and an attribute x with elements x_i, indicating the value of x at area i. Let us denote this set of areas as M, and $\bar{x}$ and S as the mean and the standard deviation of the attribute x.

Now, let R be a subregion of M with n areas. As such, it is also a set of areas that is contained in M. n corresponds to the *cardinality* of R, the number of elements in it. In (1), n is exactly $\sum_{j=1}^{N} w_{ij}$. Also, because the term $\sum_{j=1}^{N} w_{ij}x_j$ adds only the data from the areas that are in the region R, it can be equivalently expressed as $\sum_{i\in R} x_i$.

It is clear as well that having w_{ij} constrained to take values of either 0 or 1, then $w_{ij} = w_{ij}^2$ and $\sum_{j=1}^{N} w_{ij}^2 = \sum_{j=1}^{N} w_{ij} = n$. Because of this, $\left(\sum_{j=1}^{N} w_{ij}\right)^2 = n^2$.

With all this in mind, (1) can then be rewritten as follows:

$$G_R^* = \frac{\sum_{i\in R} x_i - n\bar{x}}{S\sqrt{\frac{Nn-n^2}{N-1}}}. \tag{2}$$

This notation implies that G_R^* is a function that goes from the power set of M to the real numbers,[3] and depends on the areas that are in the region R and the parameters $N, \bar{x}$, and S that are obtained from the areas in M.

Suppose we are trying to maximize this statistic for a region with a positive value of G_R^*. Intuitively, the areas that would contribute the most to its growth should be those with the highest values above the mean, that is, an area with a higher attribute value would contribute more to the statistic than one with a lower value.

The efficiency improvement presented in this paper results from the way the statistic is maximized in each iteration. Instead of doing an exhaustive search of all possible combinations of neighbors, we take a constructive approach where the areas are sorted such that those that contribute most to the growth in absolute value of the statistic come first; then, they are added one by one until no further improvement is made upon the statistic. At that point, the resulting region maximizes the statistic and the algorithm proceeds with the next iteration. We coined this modified algorithm the constructive AMOEBA and it is presented in Algorithm 2.

To guarantee that the results obtained from Algorithms 1 and 2 are equivalent, we have to prove that the G_i^* statistic is indeed maximized by applying this process to it. To prove the equivalence of constructive and exhaustive AMOEBA, the following proposition is put forth:

Proposition *Let H be a region such that $G_H^* > 0$ with a set V of neighboring areas. Let E be a set of neighbors such that $G_{H\cup E}^*$ is maximal, and n* is the*

[3] The power set of a set M is the set of all subsets of M.

Algorithm 2 Constructive AMOEBA

1: R = a_0 // *a_0 is the seed area for this iteration*
2: $G_i^{*opt} = G_i^*(R)$ // *The function G_i^* calculates* (1) *for the given region*
3: N = Neighbors (R) // *Neighbors gives a list of all areas contiguous to R*
4: T = [] // *Begins an empty list of discarded areas*
5: **if** $G_i^{*opt} \geq 0$ **then**
6: **do**
7: $G_i^{*aux} = G_i^{*opt}$
8: N = setDifference (N, T) // *Removes all areas in T from N*
9: Sort N according to area data in descending order
10: **for** $i = 1$ to length (N) **do**
11: **if** $G_i^*(R \cup N(i)) > G_i^{*opt}$ **then**
12: $R = R \cup N(i)$
13: $G_i^{*opt} = G_i^*(R)$
14: **end if**
15: **end for**
16: N = Neighbors (R)
17: **while** $G_i^{*aux} \neq G_i^{*opt}$
18: **return** R
19: **else**
20: **do**
21: $G_i^{*aux} = G_i^{*opt}$
22: N = setDifference (N, T) // *Removes all areas in T from N*
23: Sort N according to area data in ascending order
24: **for** $i = 1$ to length (N) **do**
25: **if** $G_i^*(R \cup N(i)) < G_i^{*opt}$ **then**
26: $R = R \cup N(i)$
27: $G_i^{*opt} = G_i^*(R)$
28: **end if**
29: **end for**
30: N = Neighbors (R)
31: **while** $G_i^{*aux} \neq G_i^{*opt}$
32: **return** R
33: **end if**

cardinality of E. Then the union of H and the set of n areas with the highest data values in V generate exactly the same value of G_i^*.*

This proposition states that when maximizing the G_i^* statistic for a region, it is sufficient to take the first n^* neighboring areas with the highest values. Although this number is unknown, by sequentially adding areas to the cluster ordered by the highest values, the problem is reduced to choosing the number of added neighbors such that the statistic is maximized. The preceding proposition guarantees that this

is indeed the optimum. Analogously, when the intention is to minimize the statistic for an area with a negative statistic, the areas with the lowest values are added and the number of areas that yields the minimum statistic is chosen. Both the proposition and the proof for this second case are analogous to the first.

Proof Let H be a region such that $G^*_{H\geq 0}$ and V the set of neighbors of H. Let ϑ be the amount of areas in V and P the power set of V, that is, the set of all subsets of V. Every area in V has an assigned value.

Consider the order of the elements of V given by $a_1, a_2,\ldots, a_\vartheta$, where $x_1 \geq x_2 \geq\cdots\geq x_\vartheta$. x_i is the value assigned to the area a_i, $i = 1, 2,\ldots, \vartheta$. This can be expressed as $\forall a_i, a_j \in V(i < j \rightarrow x_i \geq x_j)$.

Because the set $\{g \in \mathbb{R} : g = G^*_{H\cup Q}, Q \in P\}$ is a finite set of real numbers, it has a maximum. Thus, there exists $E \in P$ such that

$$G^*_{H\cup E} = \max_{Q\in P} G^*_{H\cup Q}.$$

In other words, $\forall Q \in P(G^*_{H\cup E} \geq G^*_{H\cup Q})$.

Let n^* be the number of elements in E and Γ the set of the n^* areas with the highest associated values, that is, $\Gamma = \{a_1, a_2, \ldots, a_{n^*}\}$

Reasoning to the absurd, suppose $G^*_{E\cup H} \neq G^*_{\Gamma\cup H}$. Then, $E \neq \Gamma$ (because $H \cap E = H \cap \Gamma = \emptyset$). Then, there exist Ψ, Φ with $\Psi = \{x_{n_1}, x_{n_2}, \ldots, x_{n_p}\} \subseteq \Gamma$ and $\Phi = \{x_{s_1}, x_{s_2}, \ldots, x_{s_p}\} \subseteq V - \Gamma$ such that $E = (\Gamma - \Psi) \cup \Phi$. This basically means that Γ and E differ by a number of elements that are in these sets.

Now, because $\Psi \subseteq \Gamma$ and $\Phi \cap \Gamma = \emptyset$, $n_1, n_2, \ldots, n_p \leq n^*$ and $s_1, s_2, \ldots, s_p > n^*$. Because of this, for all $0 \leq i \leq p$, it holds that $n_i \leq s_i$ and $x_{n_i} \geq x_{s_i}$. Adding all these inequalities, we have

$$\sum_{x_i\in\Phi} x_i \leq \sum_{x_i\in\Psi} x_i. \tag{3}$$

Adding $\sum_{x_i\in\Gamma-\Psi} x_i + \sum_{x_i\in H} x_i$ to each side of (3),

$$\sum_{x_i\in\Phi} x_i + \sum_{x_i\in\Gamma-\Psi} x_i + \sum_{x_i\in H} x_i \leq \sum_{x_i\in\Psi} x_i + \sum_{x_i\in\Gamma-\Psi} x_i + \sum_{x_i\in H} x_i$$

Due to the construction of the sets, this is equivalent to

$$\sum_{x_i\in E\cup H} x_i \leq \sum_{x_i\in\Gamma\cup H} x_i.$$

Subtracting $n^*\bar{x}$, and then dividing both sides of the inequality by $S\sqrt{\frac{Nn^*-(n^*)^2}{N-1}}$, we are left with

$$\frac{\sum_{x_i\in E\cup H} x_i - n^*\bar{x}}{S\sqrt{\frac{Nn^*-(n^*)^2}{N-1}}} \leq \frac{\sum_{x_i\in\Gamma\cup H} x_i - n^*\bar{x}}{S\sqrt{\frac{Nn^*-(n^*)^2}{N-1}}},$$

which is equivalent to:

$$G^*_{E\cup H} \leq G^*_{\Gamma\cup H}.$$

However, by hypothesis, we have that $G^*_{E\cup H} \neq G^*_{\Gamma\cup H}$. This restricts the last expression to

$$G^*_{E\cup H} < G^*_{\Gamma\cup H}. \tag{4}$$

But $\Gamma \in P$, and according to our hypothesis $G^*_{H\cup E} \geq G^*_{H\cup\Gamma}$, which contradicts (4). We thus conclude that $G^*_{E\cup H} = G^*_{\Gamma\cup H}$, that is, Γ also maximizes the G^*_i statistic. □

4 Computational experiments

4.1 The data

The data for the experiments consist of square grids of different sizes on which spatial processes were automatically generated taking into account the following requirements:

- The number of spatial clusters is predefined, and it is a even number such that half of the clusters are clusters of high values, and the other half are clusters of low values. Because the location of the clusters is defined at random, there are cases in which groups of clusters collapse into a larger single cluster.
- The percentage of areas that are assigned to clusters is 20% of the total number of areas on the grid. This areas are evenly distributed among the clusters.
- The shape of the clusters is controlled by a compactness factor (c.f.) that allow us to create either compact or elongated clusters. This factor varies between 0 and 1, with 0 being the maximum level of elongation (forcing the areas in the cluster to be arranged as a chain), and 1 being the maximum level of compactness of the cluster.
- For all intents and purposes, the rook contiguity criterion defines the neighborhood of each area.
- The values of the areas that belong to a cluster are extracted from the tails of a standard normal distribution. The values for the remaining areas are assigned by using the complete distribution. In the last case, because the complete distribution is considered, it is possible to have additional clusters, usually of small size, apart from the clusters that where deliberately constructed.

It is worth noting that the distribution of the generated attribute do not conform to a normal distribution. This is because of the previously discussed non-stationarities present in the generated ecotopes.

A more detailed description of the steps to generate artificial clusters is presented in Algorithm 3.[4]

Figure 1 shows two examples of instances of the problem. They consist of a 500 × 500 grid with a spatial process that contains six compact cluster, Fig. 1a, and

[4] Other options to generate different spatial clustering patterns, like the spiral or linear clustering pattern, can be found in Jackson et al. (2010).

Algorithm 3 Procedure to generate spatial clustering patterns

Require: P = Number of clusters, c = Compactness factor ($c \in [0, 1]$)

1: Generate a large amount of normally distributed ($\mu = 0$, $\sigma = 1$) random numbers and store them in the set D

2: Let T_l be the left tail of the generated numbers, and T_r the right tail

3: **for** $i = 1$ to P **do**

4: Calculate S, the size of the cluster, as $S = \text{Round}\left(0.2 \cdot \frac{\text{Total areas}}{P}\right)$

5: Calculate L, the "length" of the cluster, as $L = \text{Round}\left((1 - c) \cdot S\right)$

6: Select a seed area in the map that has not been assigned to a cluster. This area conforms a new cluster

7: Until the cluster has L areas, randomly choose an unassigned area from the last added area's neighbors. Assign this area to the current cluster. This forms the "backbone" of the cluster

8: Iteratively choose an area from the neighbors of the cluster that has not yet been assigned to a cluster, and append it to the cluster. Do this until the cluster has S areas

9: If creating a low-valued cluster, randomly assign values from T_l to each area in the cluster. If creating a high-valued cluster, randomly assign values from T_r

10: **end for**

11: Randomly assign values from D to each area that is not in a cluster

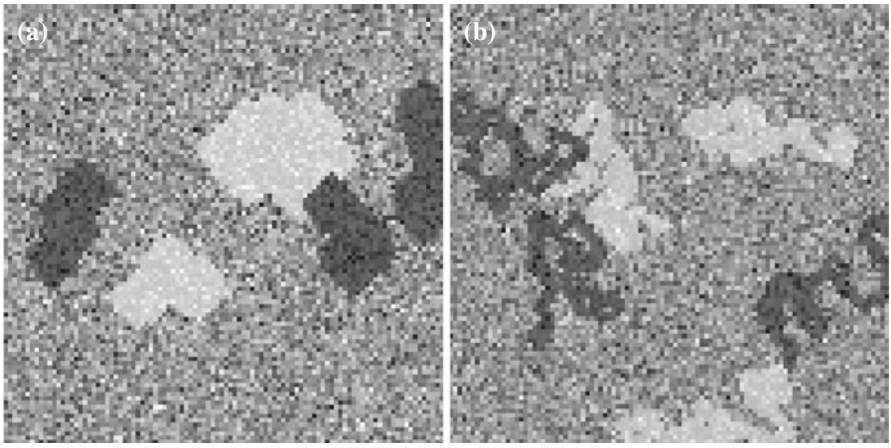

Fig. 1 Examples of instances of the problem. **a** Compact clusters (c.f. = 0.9). **b** Elongated clusters (c.f. = 0.55)

six elongated clusters, Fig. 1b. In the case of compact clusters, two cluster of low value collapsed into a single cluster.

4.2 Software and hardware

In order to ensure a fair comparison, both exhaustive and constructive AMOEBA were implemented in Python 2.6, including the Numerical Python (NumPy)

Table 1 Execution times of AMOEBA (mean ± standard deviation)

Grid size	Exhaustive (s)	Constructive (s)
4 × 4	0.12 ± 0.03	0.09 ± 0.03
5 × 5	0.31 ± 0.22	0.13 ± 0.03
6 × 6	1.63 ± 4.52	0.19 ± 0.04
7 × 7	6.47 ± 32.27	0.25 ± 0.05
8 × 8	31.7 ± 99.56	0.33 ± 0.06
9 × 9	131.97 ± 481.75	0.43 ± 0.07
10 × 10	1,800.44 ± 11053.35	0.57 ± 0.1

package.[5] Python was also used to implement the data generation process and the data and cluster visualization (with TkInter). The constructive version of AMOEBA and the algorithm to generate artificial clusters are available in GeoGrouper, an open source software written in Python that offers a selection of algorithms for region design and spatial cluster identification. This software is developed in RiSE-group (Research in Spatial Economics) at EAFIT University, and it is available from http://geogrouper.appspot.com.[6]

Regarding the hardware, we executed the algorithm on a Dell Precision T3400 computer running the Windows XP-64bits operating system equipped with 8GB RAM and a 2.99 GHz Intel Corel 2 Extreme Quad-Core processor. To optimize computational resources, we dedicated one core and 2GB RAM to each instance of the problem, which allowed us to solve four instances in parallel.

4.3 Experiment 1: Comparing the performance of exhaustive AMOEBA and constructive AMOEBA

The first experiment is intended to compare the performance of Algorithms 1 and 2. To do this, we generated grids of sizes ranging from 4 × 4 to 10 × 10, each one with 100 realizations of clustered spatial processes. These 700 instances were solved with both the exhaustive and constructive AMOEBA.

As expected, both algorithms identified the same clusters.[7] The running times and standard deviations are reported in Table 1.

The results show how the constructive AMOEBA provides a solution in a considerably smaller amount of time than the exhaustive AMOEBA. This difference grows larger as the size of the problem increases. The solution times for the constructive AMOEBA also has a much lower variance than the observed with the exhaustive AMOEBA.

In Fig. 2, we report the results for grids up to 20 × 20. In this case, we present results only for constructive AMOEBA because the execution times of exhaustive AMOEBA were too large to large to be analyzed. As the grids became larger, the

[5] NumPy provides a wide variety of mathematical functions needed for scientific computing with Python (Oliphant 2006).

[6] The algorithm as a Python module is also available from the authors on request.

[7] For each instance we generated 1,000 random permutations to perform the Monte Carlo-type permutation test.

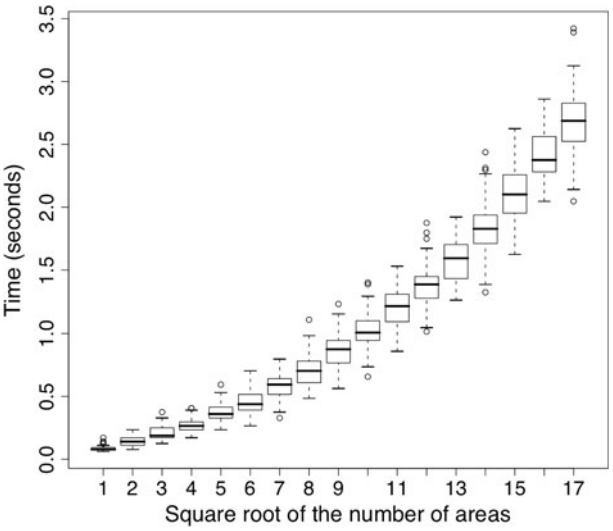

Fig. 2 Performance of constructive AMOEBA

execution times for constructive AMOEBA grew as well, but in no case did the algorithm require more than 3.5 s to detect the ecotopes. The quadratic growth of the execution times is due to the quadratic increase in the number of areas as the sides of the grid augment.

4.4 Experiment 2: Shape and size of clusters and performance of constructive AMOEBA

In this experiment, we explore the changes in running times due to changes in the size and shape of the clusters. For this, we generated 100 instances of sizes 30×30, 50×50, and 100×100 with spatial processes containing 4, 6, and 8 elongated and compact clusters. In total, 1,800 instances were solved with constructive AMOEBA. The results of these experiments are presented in Fig. 3a–c. Each boxplot summarizes the running times obtained after solving 100 instances of a given grid size, number of clusters, and compactness factor.

The first finding is that the larger the number of clusters, which implies clusters of smaller size,[8] the smaller the average and standard deviation of running times. The size of the clusters affects the performance because for big clusters there is, at each iteration, a larger number of neighboring areas that are candidates for joining the cluster. In exhaustive AMOEBA, this phenomenon has a much larger impact on running times because of its combinatorial approach.

There is a direct relationship between the cluster size and the standard deviation of running times because when, during the data generation process, a group of clusters collapses, the greater the size of the individual clusters collapsed, the

[8] This is because the percentage of areas that are assigned to be part of clusters is fixed, which implies that a larger number of clusters results in smaller cluster size.

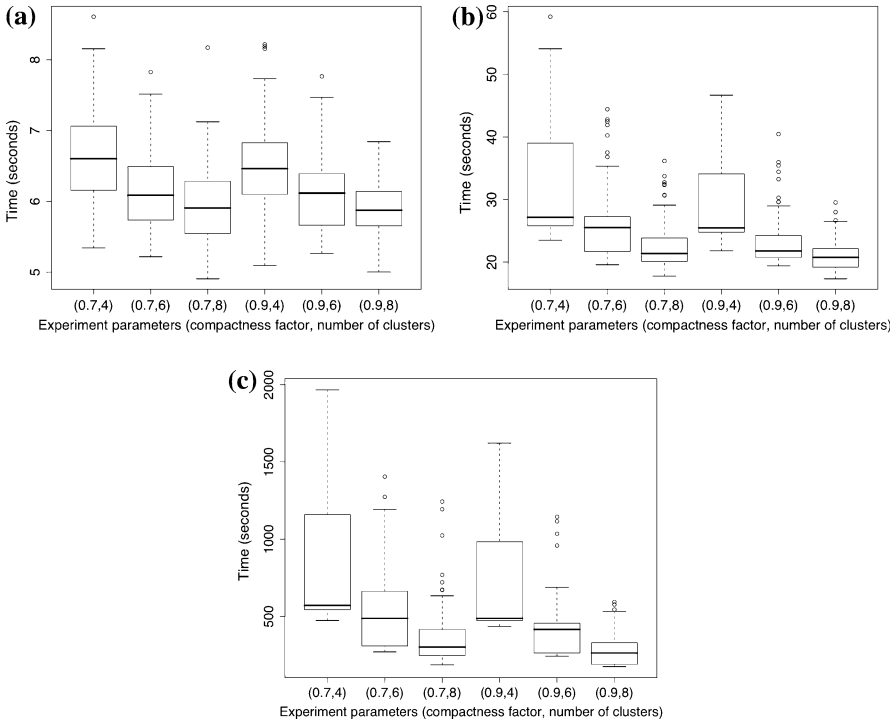

Fig. 3 Running times of constructive AMOEBA for different shapes and sizes of the clusters. **a** 30 × 30. **b** 50 × 50. **c** 100 × 100

greater the impact on running times. This relationships explains why the box plots show in most cases a significantly longer tail toward larger values, which suggests that the running times are positively skewed.

The results show that the shape of the cluster has a minimal impact on the execution time. At each iteration, the optimization process is solely dependent on the attribute value of the neighboring areas of the ecotope, so the time required to obtain the solution is linear instead of factorial.

4.5 Experiment 3: Spatial distribution and performance of AMOEBA 2

Finally, in the third experiment, we measure how the running time of the constructive algorithm is affected by the spatial distribution of the data. To approach this question, we carried out the experiment presented in Algorithm 4.

The resulting distribution of the running times is presented in Fig. 4. The figure shows an outlier of 22.81 s that corresponds to the time required for AMOEBA to solve the unaltered generated map. The running times corresponding to the permuted data are significantly lower, with a symmetrical distribution. The instances that correspond to the highest and lowest execution times are shown in Fig. 5. Figure 5a, c present the spatial distribution of the variables, and Fig. 5b, d show the true clusters

Algorithm 4 Spatial distribution experiment

1: Generate 50 × 50 grid with 3 high valued and 3 low valued-clusters
2: Execute constructive AMOEBA and store the execution time
3: **for** $i = 1$ to 200
4: Randomly distribute the data in the grid
5: Execute constructive AMOEBA and store the execution time
6: **end for**

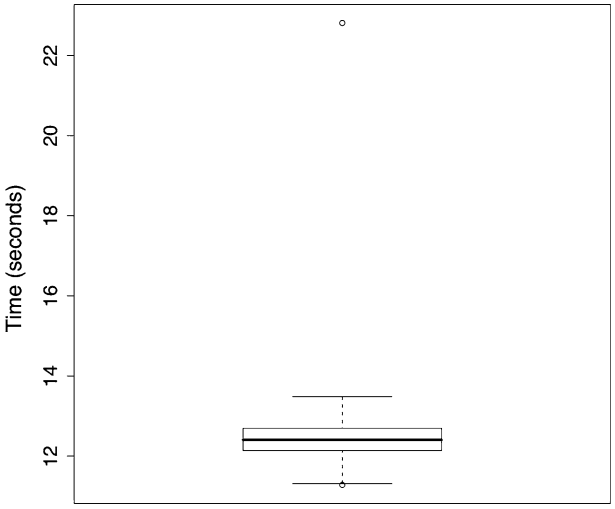

Fig. 4 Execution times recorded from Algorithm 4

(i.e., those clusters that are statistically significant when applying the Monte Carlo-type permutation test). Clusters of high values are color coded with black, clusters of low values are color coded with white, and the areas outside of cluster are color coded with gray. It can be derived from this experiment that the execution times of AMOEBA depend on the spatial distribution of the data in the grid.

5 Empirical application

The data used for this empirical application were provided by Professor John Weeks, director of the International Population Center at San Diego State University.[9] The region of study is the Metropolitan Area of Accra divided into 1,717 Enumeration Areas, each of which was assigned a value corresponding to the proportion of adults whose profession is administrative, clerical, or professional. The map in Fig. 6 is color coded according to this variable: the darker the color of

[9] http://geography.sdsu.edu/Research/Projects/IPC/ipc2research.html.

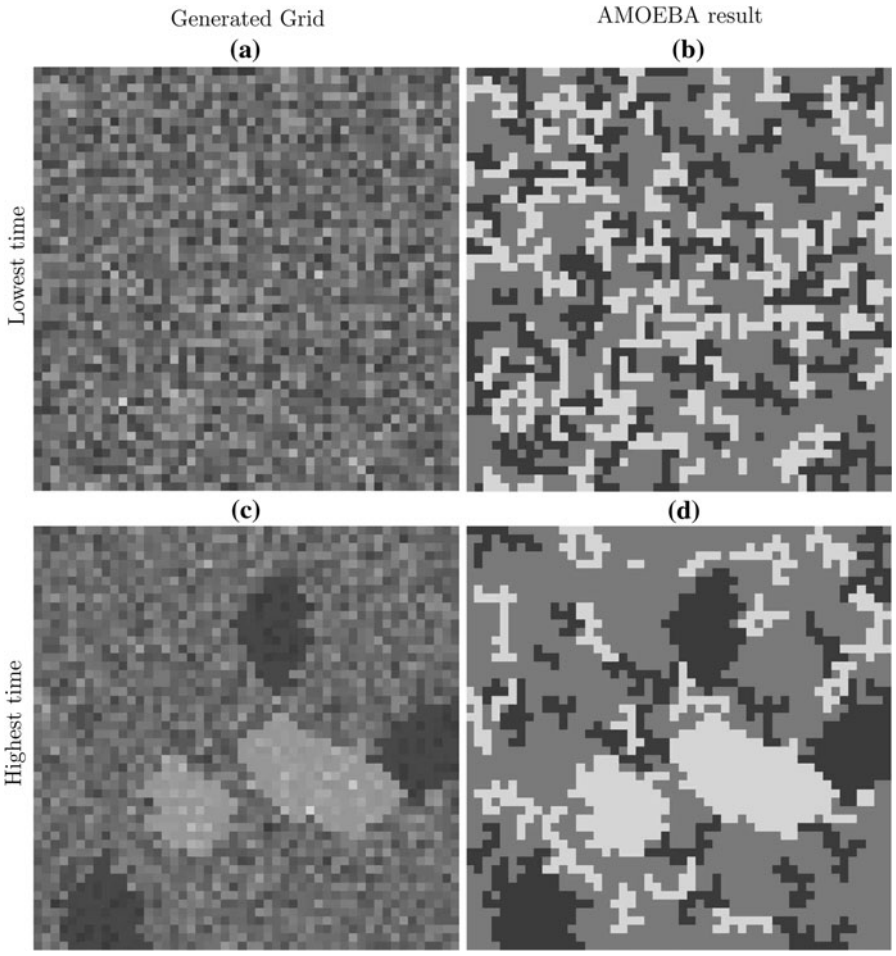

Fig. 5 Grids corresponding to the highest and lowest times from Algorithm 4

the spatial unit, the higher the proportion of adults whose profession is administrative, clerical, or professional in that particular area.

A study of this map using AMOEBA can give the researcher insight into the socio-economic patterns present on the city. Employment in the selected sectors requires higher levels of education than the omitted categories, which include sales, service, mining, fishing, and agriculture. This variable, therefore, is a proxy for social and human capital in an enumeration area.

The clusters obtained from AMOEBA are shown in Fig. 7. The resulting ecotopes resemble closely the regions that are visually apparent in Fig. 6. Table 2 presents some summary statistics to provide a better understanding of the results obtained from AMOEBA. The last column of the table provides those statistics for the 1,717 Enumeration Areas in Accra.

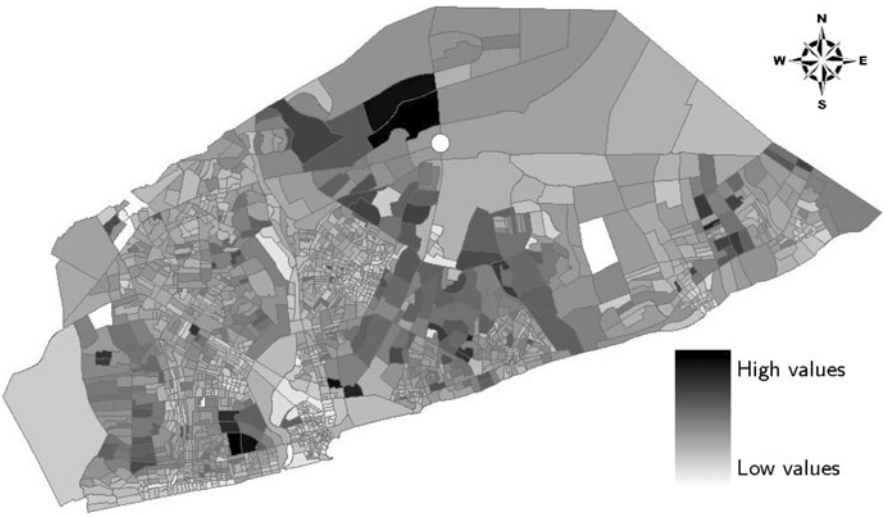

Fig. 6 Map of Accra color coded according to the percentage of adults whose occupation is administrative, clerical or professional

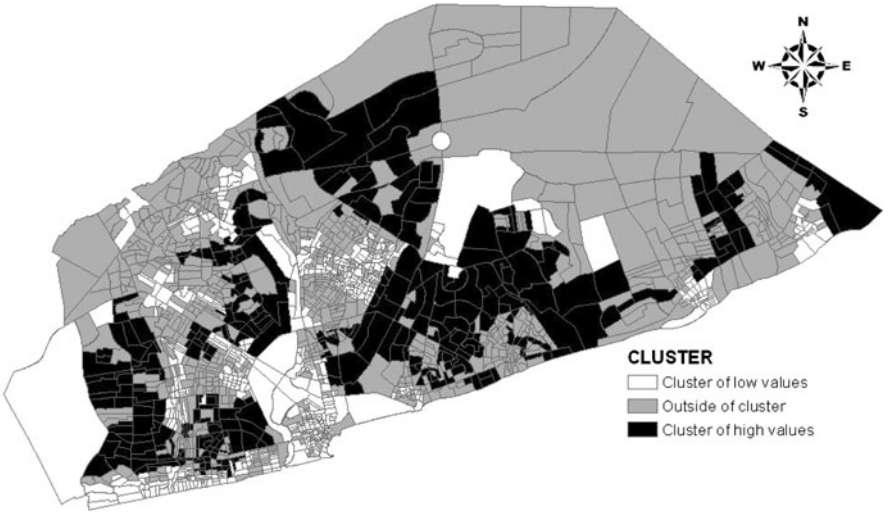

Fig. 7 AMOEBA clusters

6 Conclusions

This paper presents an alternative formulation for the AMOEBA that drastically reduces run times, while yielding exactly the same results as the original formulation. This methodological improvement makes it possible to use AMOEBA to solve larger problems.

Table 2 Summary statistics for the results obtained from AMOEBA

	Clusters of high values	Clusters of low values	Areas outside of cluster	Total
Number of areas	336	556	825	1,717
% of population (n = 1,645,584)	17.78%	30.94%	51.28%	100.00%
Mean (x)	38.58%	15.94%	25.54%	24.98%
Standard deviation (x)	8.15%	4.37%	4.81%	9.65%
Max (x)	72.00%	24.00%	53.00%	72.00%
Min (x)	25.00%	0.00%	8.00%	0.00%

x: Proportion of adults whose profession is administrative, clerical or professional

After we proved the equivalence of the original exhaustive AMOEBA and the new constructive AMOEBA, we carried out different experiments to get a deeper understanding of the performance of the new algorithm. Those experiments showed that the spatial distribution of data affects running times; thus, the bigger the ecotopes the larger the running time, and the larger the variance. Finally, an empirical example using socio-economic data in Accra was presented, showing AMOEBAS' capability to identify irregular ecotopes.

Further research will be conducted in two directions. First, we will study the applicability of this constructive approach to other spatial clustering statistics. Second, we will formulate a new version of AMOEBA that can be applied to spatial panel data.

Acknowledgments The authors thank Professor Dr. John Weeks, director of the International Population Center at San Diego State University, for providing us with the data for our empirical application. The usual disclaimer applies.

References

Aldstadt J (2010) Spatial clustering. In: Fischer M, Getis A (eds) Handbook of applied spatial analysis. Springer, Berlin, pp 279–300

Aldstadt J, Getis A (2006) Using AMOEBA to create a spatial weights matrix and identify spatial clusters. Geogr Anal 38(4):327–343

Anselin L (1995) Local indicators of spatial association-LISA. Geogr Anal 27(2):93–115

Fotheringham S, Brunsdon C, Charlton M (2000) Quantitative geography: perspectives on spatial data analysis. Sage Publications, London

Gaudart J, Poudiougou B, Ranque S, Doumbo O (2005) Oblique decision trees for spatial pattern detection: optimal algorithm and application to malaria risk. BMC Medical Research Methodology. doi:10.1186/1471-2288-5-22

Getis A, Ord J (1992) The analysis of spatial association by use of distance statistics. Geogr Anal 24(3):189–206

Jackson MC, Huang L, Xie Q, Tiwari RC (2010) A modified version of Moran's I. Int J Health Geograp. doi:10.1186/1476-072X-9-33

Jacquez G (2009) Cluster morphology analysis. Spat Spattemporal Epidemiol 1(1):19–29

Knox E (1989) Detection of clusters. In: Elliot P (eds) Methodology of enquiries into disease clustering. Small Area Health Statistics Unit, London, pp 17–22

Oliphant T (2006) Guide to NumPy. Trelgol Publishing, USA

Ord J, Getis A (1995) Local spatial autocorrelation statistics: Distributional issues and application. Geogr Anal 27(4):286–306

Robinson W (1950) Ecological correlations and the behavior of individuals. Am Sociol Rev 15(3):351–357

Tango T, Takahashi K (2005) A flexibly shaped spatial scan statistic for detecting clusters. Int J Health Geogr. doi:10.1186/1476-072X-4-11

Wu J, Kendrick K, Feng J (2007) A novel approach to detect hot-spots in large-scale multivariate data. BMC Bioinformatics. doi:10.1186/1471-2105-8-331

Yamada I, Rogerson P, Lee G (2009) GeoSurveillance: a GIS-based system for the detection and monitoring of spatial clusters. J Geogr Syst 11(2):155–173