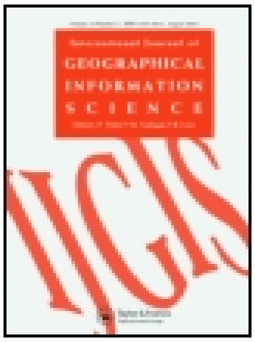

International Journal of Geographical Information Science

ISSN: 1365-8816 (Print) 1362-3087 (Online) Journal homepage: https://www.tandfonline.com/loi/tgis20

Efficient regionalization for spatially explicit neighborhood delineation

Ran Wei, Sergio Rey & Elijah Knaap

To cite this article: Ran Wei, Sergio Rey & Elijah Knaap (2020): Efficient regionalization for spatially explicit neighborhood delineation, International Journal of Geographical Information Science, DOI: 10.1080/13658816.2020.1759806

To link to this article: https://doi.org/10.1080/13658816.2020.1759806

Published online: 05 May 2020.

INTERNATIONAL JOURNAL OF GEOGRAPHICAL INFORMATION SCIENCE
https://doi.org/10.1080/13658816.2020.1759806

Taylor & Francis
Taylor & Francis Group

RESEARCH ARTICLE

Efficient regionalization for spatially explicit neighborhood delineation

Ran Wei, Sergio Rey and Elijah Knaap

Center for Geospatial Sciences, School of Public Policy, University of California, Riverside, CA, USA

ABSTRACT

Neighborhood delineation is increasingly relied upon in urban social science research to identify the most appropriate spatial unit. In problems of this type, the true number of neighborhoods (typically called the *k* parameter) is unknown and analysts often require algorithmic approaches to determine *k* endogenously. Existing approaches for neighborhood delineation that do not require pre-specification of a *k*-parameter, however, are either non-spatial or lead to noncontiguous or overlapping regions. In this paper, we propose the use of max-p-regions for neighborhood delineation so that the geographic space can be partitioned into a set of homogeneous and geographically contiguous neighborhoods. In addition, we developed a new efficient algorithm to address the computational challenges associated with solving the max-p-regions so that it can be applied for large-scale neighborhood delineation. This new algorithm is implemented in the open-source Python Spatial Analysis Library (PySAL). Computational experiments based on both simulated and realistic data sets are performed and the results demonstrate its effectiveness and efficiency.

ARTICLE HISTORY
Received 26 November 2019
Accepted 20 April 2020

KEYWORDS
Regionalization; optimization; neighborhoods

Introduction

An increasingly important technique in the field of GIScience is the identification of distinct sub-regions or neighborhoods within a study area using unsupervised learning methods. These clustering algorithms, generally categorized as regionalization methods, aim to partition the geographic space into a set of homogeneous and geographically contiguous regions (Openshaw and Rao 1995, Duque *et al*. 2007, Guo and Wang 2011, Garreton and Sánchez 2016). As the name implies, these techniques were deployed originally using large-scale geographic units such as counties, and the algorithms proceed to aggregate neighboring counties, subject to some optimization criteria, into large-scale regions. Today, regionalization algorithms might be more aptly described as 'spatially-constrained clustering algorithms' as their application enjoys continued and expanded use in neighborhood research that leverages much smaller-scale polygon data, such as census tracts, albeit with similarly sized study areas. In these newer contexts, then, the typical problem size for regionalization algorithms is increasing dramatically.

CONTACT Sergio Rey ✉ sergio.rey@ucr.edu

One important application of spatially constrained clustering is the identification of endogenous regions or neighborhoods where the total number, spatial configuration, and internal composition of neighborhoods are all unknown *a priori*. Here, the goal is to find the maximum number of neighborhoods, whose homogenous internal characteristics demarcate it from others. While the definition of neighborhood varies across disciplines, it typically refers to 'a contiguous territory defined by a bundle of social attributes that distinguish it from surrounding areas' (Spielman and Logan 2013), coinciding with the goal of regionalization approaches (Folch and Spielman 2014). This work is becoming increasingly prevalent thanks to two widespread applications.

The first applications are practical, in which regionalization is leveraged method of data processing used to develop new primitive spatial units that have better statistical reliability. The motivation, in this case, is that social surveys (e.g. the census) designed to represent small geographies are often drawn from small sample sizes, resulting in high error margins. A poignant example is provided by the U.S. Census American Community Survey (ACS) whose error margins can sometimes exceed the point estimate for tract-level geographies. To improve the reliability of these estimates, Spielman and Singleton (2015) have advocated the identification of 'bespoke' neighborhoods through multivariate clustering, which allows similar units to be grouped together until their survey estimates reach a more reliable threshold. Adopting the advice of Spielman and Singleton, including a connectivity constraint, and determining the appropriate number of resulting regions endogenously ensures that appropriate primitive units get aggregated together and helps ensure an optimal solution.

The second application is topical, in which regionalization is used to identify unique and discrete social neighborhoods according to their demographic composition. This body of work grows from the tradition of geodemographic analysis (e.g. Harris *et al*. 2005), where multivariate clustering algorithms are applied to census data to form neighborhood cluster types. In classic geodemographics, the resulting neighborhood types are mapped, however, there is no guarantee that neighborhood types are spatially compact or contiguous. As a result, neighborhood types can be spatially fragmented, which runs counter to the substantive understanding of neighborhoods as organizational units for human spatial behavior. For this reason, recent work on neighborhoods has sought to use spatially-constrained clustering to develop more realistic depictions of discrete neighborhoods. A prominent example is given by Rey *et al*. (2011) who use this approach to examine the dynamic footprint of social neighborhoods over time. However, one of the major difficulties in applying regionalization methods to neighborhood delineation is their significant computational complexity (Spielman and Logan 2013).

In both of the above applications, data volumes, problem sizes, and the range of substantive questions to be asked are increasing at a rapid pace. While existing regionalization algorithms have been applied with success, the current approaches are also computationally expensive and require long run times for modestly sized problems. As such, there is a clear need for exploration and development of novel approaches to regionalization that are scalable, efficient, and able to ingest vast amounts of data in short cycles. In this paper, we present one such approach. We focus on one of the most widely used regionalization methods, max-p-regions (Duque *et al*. 2012), and proposed a new efficient algorithm to address the computational challenges associated with solving it. In the next section, we provide a review of existing regionalization approaches with a particular focus on max-p-regions. Next, the new solution algorithm is presented. Finally, the proposed approach is applied to identifying neighborhoods in several simulated datasets and census datasets, highlighting the effectiveness and efficiency of the new regionalization approach.

Regionalization

The need to aggregate spatial units into a set of contiguous regions arises in many social and environmental contexts, such as political districting, school districting, police patrol districting, habitat delineation, and various zone aggregations for modeling purposes. Many regionalization algorithms have been developed to fulfill such needs. For instance, Duque *et al.* (2011) formulated a typical regionalization problem as a mixed-integer programming (MIP) model that can be solved using general MIP solver, like GUROBI (Gurobi Optimization 2019) or GLPK (GNU Linear Programming Kit 2012). Guo (2008) integrated contiguity constraints into hierarchical clustering and developed the regionalization algorithm with dynamically constrained agglomerative clustering and partitioning (REDCAP). Li *et al.* (2014) developed a heuristic method, memory-based randomized greedy and edge reassignment (MERGE), to aggregate spatial units into p compact and contiguous regions. A detailed review on regionalization algorithms can be found in Duque *et al.* (2007) and Garreton and Sánchez (2016).

Most of these regionalization algorithms require a prespecification of the number of regions identified (Folch and Spielman 2014, Garreton and Sánchez 2016). For example, the number of identified regions, *p*, is an input parameter for the p-regions model formulated in Duque *et al.* (2011), p-functional-regions formulated in Kim *et al.* (2015), and p-compact-regions formulated in Li *et al.* (2014). The users must select the level to cut for the hierarchical clustering-based method like REDCAP in Guo (2008) and Guo and Wang (2011). However, the users rarely know the number of regions *a priori*. Alternatively, the max-p-regions proposed in Duque *et al.* (2012) allows the users to specify criteria that define a region and a regionalization scheme that satisfies the criteria is identified by solving the model. Such endogenization of the number of regions based on user-specified criteria makes the max-p-regions approach ideally suited to identifying neighborhoods for further statistical modeling purposes (Folch and Spielman 2014). Here we reviewed max-p-regions model to highlight this and provide the basis for the solution algorithm developed. Consider the following notation (Duque *et al.* 2012):

Parameters

$$i, j = \text{index of spatial units, } i \in I$$

$$k = \text{index of potential regions, } k \in K$$

$$c = \text{index of contiguity order}$$

$$d_{ij} = \text{dissimilarity relationships between units } i \text{ and } j$$

$$l_i = \text{spatially extensive attribute value of unit } i$$

$$T = \text{minimum value for attribute } l \text{ at regional scale}$$

$$w_{ij} = \begin{cases} 1, \text{if unit i and j share aborder} \\ 0, \text{otherwise} \end{cases}$$

$$N_i = \{j | w_{ij} = 1\}$$

$$F = 1 + \left\lfloor \log\left(\sum_i \sum_j d_{ij}\right) \right\rfloor$$

Decision variables:

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

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

As the number of identified regions is unknown, the potential regions are indexed by k, which could range from 1 to the total number of spatial units. The contiguity order, indexed by c, is used to ensure contiguity within one region. Specifically, each region has only one root unit with a contiguity order $c = 0$. The other units that are assigned to the same region are either adjacent to the root unit or next to a unit that is contiguous to the root unit with a smaller order number. In addition to the attributes that are used to describe dissimilarity between units, the spatially extensive attribute, l_i, defines the size criteria that each region must satisfy, such as the number of population and number of housing units. The number of regions is endogenized by ensuring each region exceeds the threshold, T, on attribute l. The parameter w_{ij} defines whether units $y_{ij} \in \{0, 1\}, \forall i, j$ and j are adjacent, and the N_i is the set of units that are adjacent to unit i. Given this notation, the max-p-regions can be formulated as follows:

$$\min\left(-\sum_k \sum_i x_i^{k0}\right) * 10^F + \sum_i \sum_j d_{ij} y_{ij} \tag{1}$$

Subject to:

$$\sum_i x_i^{k0} \leq 1, \forall k \tag{2}$$

$$\sum_k \sum_c x_i^{kc} = 1, \forall i \tag{3}$$

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

$$\sum_i \sum_c x_i^{kc} l_i \geq T \sum_i x_i^{k0}, \forall k \tag{5}$$

$$y_{ij} \geq \sum_c x_i^{kc} + \sum_c x_j^{kc} - 1, \forall i, j, k \tag{6}$$

$$x_i^{kc} \in \{0, 1\}, \forall i, k, c \tag{7}$$

$$y_{ij} \in \{0, 1\}, \forall i, j$$

The objective, (1), has two main terms with one term maximizing the number of regions, $\sum_k \sum_i x_i^{k0}$, and the other term minimizing the total within-region dissimilarity, $\sum_i \sum_j d_{ij} y_{ij}$. The number of regions is multiplied by a scaling factor10^F so that the goal of maximizing the number of regions always dominates the goal of minimizing the total within-region heterogeneity. That is, a solution with larger number of regions will always be preferred over any other solutions with smaller number of regions; for solutions with the same number of regions, a solution with lower heterogeneity will be preferred. Constraints (2) ensure that each region has at most one root unit. Constraints (3) specify that each unit is assigned to exactly one region with one contiguity order. Constraints (4) require that unit *i* is assigned to region *k* at contiguity order *c* if and only if one of its adjacent unit *j* is assigned to region *k* at order $c - 1$. Constraints (5) ensure that the total value of spatially extensive attribute at each region exceeds the prespecified threshold. Constraints (6) link the decision variables. Constraints (7) impose binary restrictions on decision variables.

While only one spatially extensive attribute was included in this original formulation of max-p-regions, Folch and Spielman (2014) generalized it to enable multiple attributes to be the size constraints for identified regions. Such size constraints combined with the objective of maximizing the number of regions allow for the preservation of as much geographic detail as possible. In addition, the contiguity constraints and the other objective of minimizing the within-region heterogeneity ensure that the identified region is contiguous and as homogeneous as possible. These characteristics make the max-p-regions ideally suited for neighborhood delineation.

However, the max-p-regions are NP-hard and computationally expensive to solve (Duque *et al.* 2012). The largest-sized problem that can be solved optimally using exact MIP solution method is a problem with 16 units (Duque *et al.* 2012). We have also tried to solve the max-p-region problems using GUROBI, which is the most state-of-the-art commercial MIP solver, and the results are consistent with what was reported in Duque *et al.* (2012). To address its associated computational challenges, Duque *et al.* (2012) developed a two-phase heuristic method with the first phase constructing the feasible solution and the second phase improving the solution from the first phase through several different local search strategies (greedy, simulated annealing, and tabu search). While this heuristic method makes it computationally possible to solve practically sized problems, Duque *et al.* (2012) reported that it takes several hours to obtain the best quality solutions for problems with over 3000 units. There is a clear need to develop more efficient solution approaches for the max-p-regions in order to enable its application to large-scale neighborhood delineation.[1]

Solution approach

Given the computational complexity associated with solving the max-p-regions exactly and heuristically, a new solution approach is developed to efficiently solve max-p-regions for large-sized problems. This new solution approach is composed of three main stages: region growth, enclave assignment, and local search. The first stage focuses on growing regions in a way that can maximize the number of regions; the second stage assigns enclaves using a randomized greedy strategy; and the final stage iteratively improves the total within-region heterogeneity through a customized simulated annealing that integrates a tabu list. The overall design of the new solution approach for max-p-regions is summarized in Figure 1. After initialization, the procedure of growing regions is repeated for MI times as significant randomness is involved in the procedure and the resulting partition will be different from run to run. Next, the partitions leading to the maximal number of regions are passed to the following procedures for enclave assignment and local search. At the end, the partition with the least within-class heterogeneity is considered to be the best solution identified. Details of the three stages are now presented.

Region growth

The purpose of the region growth phase is to identify a set of contiguous regions whose total spatially extensive attribute exceeds the threshold. The flow chart for region growth is shown in Figure 2. It starts by randomly selecting an unassigned unit as the seed unit for a region and then iteratively adds the unassigned neighbors of the units in the region until it reaches the threshold or no unassigned neighbor can be found. If the region formed fails to reach the threshold, all the units assigned to the region are referred to as 'enclave' and are added to the enclave set. This process is repeated until all units have been either assigned to a region or included in the enclave set. At the end of this phase, we will identify a set of contiguous regions whose spatially extensive attribute exceeds the prespecified threshold and a set of enclaves.

This phase focuses on identifying as many regions as possible and does not account for the attribute dissimilarity between units, both significant design differences from the region growth algorithm proposed in Duque *et al*. (2012) that grows region by iteratively including the neighboring unit that minimizes the total within-class dissimilarity. As the

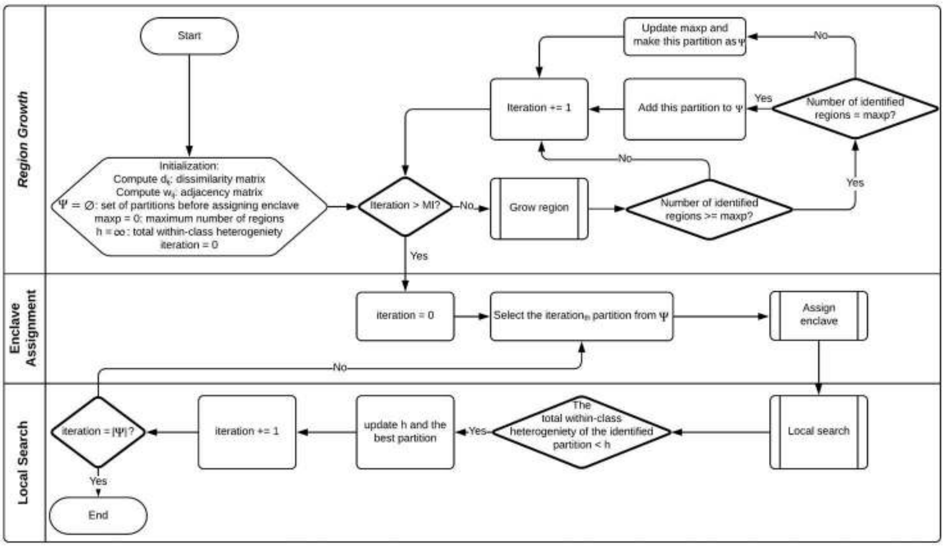
Region Growth
Enclave Assignment
Local Search
Start
Initialization:
Compute d_{ij}: dissimilarity matrix
Compute w_{ij}: adjacency matrix
$\Psi = \emptyset$: set of partitions before assigning enclave
maxp = 0: maximum number of regions
h = ∞: total within-class heterogeniety
iteration = 0
Iteration > MI?
No
Yes
Grow region
Number of identified regions >= maxp?
No
Yes
Number of identified regions = maxp?
Yes
No
Add this partition to Ψ
Update maxp and make this partition as Ψ
Iteration += 1
iteration = 0
Select the iteration_{th} partition from Ψ
Assign enclave
Local search
The total within-class heterogeneity of the identified partition < h
Yes
update h and the best partition
iteration += 1
iteration = |Ψ|?
No
Yes
End

Figure 1. Flow chart of the new solution approach.

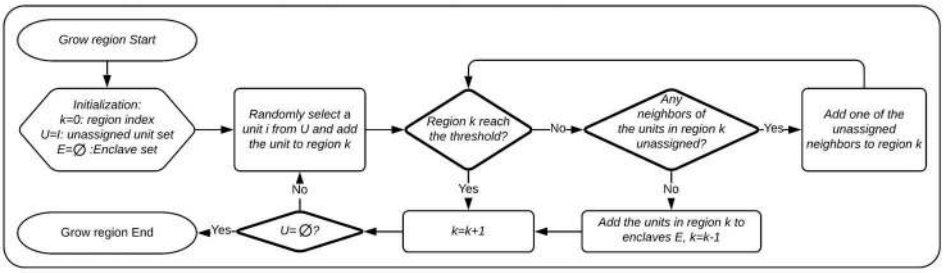
Grow region Start
Initialization:
k=0: region index
U=I: unassigned unit set
E=∅ :Enclave set
Randomly select a unit i from U and add the unit to region k
Region k reach the threshold?
No
Any neighbors of the units in region k unassigned?
Yes
Add one of the unassigned neighbors to region k
No
Yes
No
Grow region End
Yes
U=∅?
k=k+1
Add the units in region k to enclaves E, k=k-1

Figure 2. Flow chart of region growth.

number of identified regions is determined in this phase and will not be modified in the following two phases, it is important to devise the region growth strategy so that the number of regions can be maximized. The computational results in the next section show that this new algorithm can identify more compact regions and results in a much larger number of regions found.

Enclave assignment

The goal of the enclave assignment phase is to assign the enclaves to the regions identified in the region growth phase (Figure 3). It starts by randomly selecting a unit in the enclave set and then if any of its neighbors has been assigned to a region, the dissimilarity between the enclave and all neighboring regions is computed and the enclave will be randomly assigned to one of the neighboring regions with the N smallest dissimilarity. This process is repeated until all enclaves have been assigned to a region. At the end of this phase, we will identify a feasible solution for the max-p-regions problem where each region satisfies the contiguity and spatial threshold constraints and the identified regions are a complete partition for the spatial units.

This enclave assignment strategy is different from the greedy enclave assignment in Duque *et al*. (2012) where each enclave will be assigned to the neighboring region with the smallest dissimilarity. The strategy of randomly choosing one of the best candidates but not necessarily the top candidate is generally referred to as randomized greedy algorithm. It was first introduced by Feo and Resende (1995) in the Greedy Randomized Adaptive Search Procedure (GRASP) to increase solution diversity while not necessarily compromising the solution quality in the initial solution construction. Given such

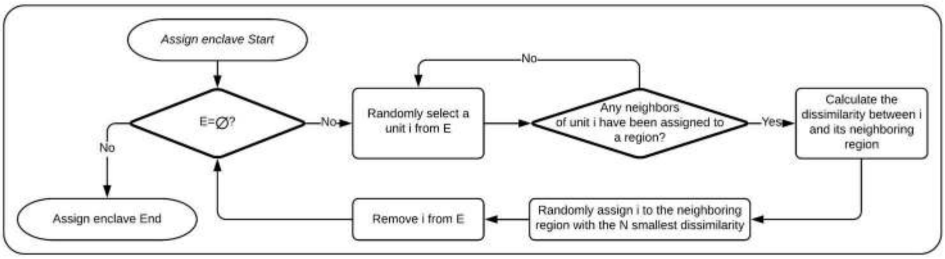
Assign enclave Start
No
E=∅?
No
Randomly select a unit i from E
Any neighbors of unit i have been assigned to a region?
Yes
Calculate the dissimilarity between i and its neighboring region
No
Assign enclave End
Remove i from E
Randomly assign i to the neighboring region with the N smallest dissimilarity

Figure 3. Flow chart of enclave assignment.

superiority to traditional greedy algorithm, this randomized greedy strategy has been applied in a range of regionalization problems (González-Ramírez *et al*. 2011; Cano-Belman et al. 2012, Li *et al*. 2014).

Local search

After identifying a good initial feasible solution in the first two phases, we design a local search algorithm to improve the solution's total within-class heterogeneity by iteratively moving a spatial unit from its current region (donor region) to a neighboring region (recipient region) while ensuring the solution's feasibility. The flow chart for the local search algorithm is depicted in Figure 4. This algorithm follows the general design of simulated annealing (SA) that simulates the process of heating a material and then slowly lowering the temperature to control defect. Duque *et al*. (2012) has implemented the SA to solve the max-p-regions problem. Specifically, given a feasible solution the SA algorithm identifies all candidate units that can move to a neighboring region without violating the contiguity and threshold constraints, and then randomly selects one candidate unit. If this move can reduce the total heterogeneity, it is accepted; otherwise, the nonimproving move is accepted with a probability defined by Boltzmann's equation, $p = e^{-\Delta H/t}$, where ΔH is the total heterogeneity change due to this move and t is the current temperature. This process is iterated with t gradually decreasing at a cooling rate α until t reaches a prespecified value.

Our new algorithm introduces several significant changes to the original SA algorithm. First, our algorithm dynamically updates a list of potential units that can move to a neighboring region without violating the contiguity and threshold constraints, rather than recompute the potential units at each iteration. Identifying movable units is

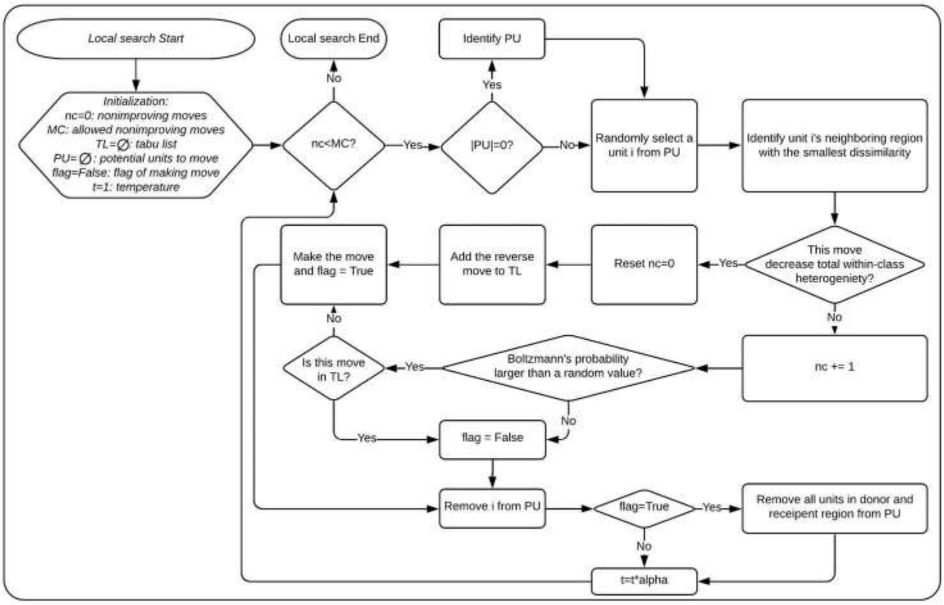
Local search Start
Initialization:
nc=0: nonimproving moves
MC: allowed nonimproving moves
TL=∅: tabu list
PU=∅: potential units to move
flag=False: flag of making move
t=1: temperature
Local search End
No
nc<MC?
Yes
Identify PU
Yes
|PU|=0?
No
Randomly select a unit i from PU
Identify unit i's neighboring region with the smallest dissimilarity
This move decrease total within-class heterogeneity?
Yes
Reset nc=0
Add the reverse move to TL
Make the move and flag = True
No
nc += 1
Boltzmann's probability larger than a random value?
Yes
Is this move in TL?
No
Yes
flag = False
No
Remove i from PU
flag=True
Yes
Remove all units in donor and receipent region from PU
No
t=t*alpha

Figure 4. Flow chart of local search.

computationally intensive because for each unit we need to check whether losing the unit will break the spatial threshold constraint and whether it will leave the remaining units in the region to be unconnected. Our algorithm recomputes the movable units only when the list of potential units is empty. Otherwise, the list is updated after each move by removing the moved unit, and all the units in the donor and recipient region. This will ensure the remaining units in the list are still able to move without violating the constraints. Second, once the potential unit is selected, only the best possible move is considered for further assessment, rather than any possible move. That is, only the neighboring region with the smallest dissimilarity could be the recipient region. As the solution diversity is maintained by randomly selecting a candidate unit, allowing the best move only could lead to faster convergence to high-quality solution. Third, a tabu list is integrated in the criteria for accepting nonimproving moves. The tabu list that represents a list of banned moves is used in tabu search algorithm to discourage the search from coming back to previously visited solution (Glover 1989). Li *et al*. (2014) show that once a nonimproving move is made near the algorithm completion, the search oscillates among a small set of solutions that consist of reverse moves of previous improving moves. We therefore construct the tabu list by iteratively adding the reverse moves of improving moves to prevent this and result in faster convergence. A nonimproving move is made only when it is not in the tabu list and the Boltzmann's probability is larger than a random value. The tabu list has a prespecified length limiting the number of moves that can be accommodated in the list and takes the queue strategy when the list is full. Finally, our algorithm allows for termination when all of the previous NC potential moves selected are nonimproving, rather than only in the case where the temperature t reaches a predefined value. This termination condition is consistent with the condition for tabu search in Duque *et al*. (2012). Computational experiments show this termination condition could lead to better-quality solutions.

In addition to the SA, Duque *et al*. (2012) also tested tabu search and greedy algorithms for local search. They reported that the tabu search can identify the best solutions in most scenarios but it is much more computationally expensive, whereas the simulated annealing and greedy algorithms are computationally efficient but lack the capacity to identify the best solutions. This new local search algorithm combines the strengths of tabu search and simulated annealing with the aim of identifying better-quality solutions and improving computational efficiency.

Results

We performed a series of computational experiments to assess the performance of the proposed approach for solving the max-p-regions problem. The data sets are retrieved from sample data in the ClusterPy library for regionalization research (Duque *et al*. 2011). The data include four simulated data sets, which are regular lattices with 100, 529, 1024, 2025 units, and two realistic datasets, which are 58 counties in California and 3106 connected counties in the U.S. The attribute value to measure the dissimilarity d_{ij} for the regular lattices is simulated using a spatial autoregressive process with $p = 0.9$, whereas the spatial extensive attribute value l_i is simulated using a uniform distribution of [10, 15]. Three different threshold values T = 100, 300, and 500 are tested for the simulated data set. The attribute dissimilarity d_{ij} is also simulated for the counties in

California but median household income is used for the counties in the U.S. The spatial extensive attribute value l_i is the population for the counties in California and the number of household units for the counties in the U.S. Three different threshold values T = 100,000, 300,000, and 500,000 are tested for the two realistic data sets. The algorithms are implemented using Python, and experiments are conducted on an Intel Core i7-6600 (2.60 GHz) computer running Windows with 16 GB RAM. The algorithm will be included in the next release of the open-source Python Spatial Analysis Library (PySAL).

As the number of identified regions is determined in the region growth phase, we first run the new region growth algorithm 999 times for each combination of dataset and threshold to compare the number of regions with what is found using the region growth approach in Duque *et al*. (2012). The results show that our new region growth algorithm identified larger number of regions for all datasets and thresholds except the dataset of counties in California (Figure 5). This is likely due to its small number of spatial units. For example, the number of regions identified by our new algorithm for the 2025 unit regular lattice with $T = 100$ ranges from 198 to 213 during the 999 runs, whereas that by the approach in Duque *et al*. (2012) ranges from 187 to 205. For the U.S. counties dataset with $T = 500,000$, the number of regions identified by the new algorithm varies from 148 to 165 during the 999 runs, whereas that by the approach in Duque *et al*. (2012) ranges from 137 to 154. Clearly, our new algorithm generally dominates the approach in Duque *et al*. (2012) in terms of number of regions identified. Next, for each partition with the maximum number of regions, we assign enclave using our new algorithm to generate initial feasible solutions. The computational time of region growth (999 runs) and enclave assignment for the new algorithm and the original algorithm is reported in Table 1. Clearly, the new region growth algorithm is more computationally efficient than that in Duque *et al*. (2012) because the new region growth algorithm does not account for the attribute dissimilarity while the original algorithm requires the identification of the neighboring unit that minimizes the total within-class dissimilarity at each iteration. The enclave assignment strategies for both algorithms are efficient and really depend upon the number of partitions that are identified with the maximum number of regions and are passed to the enclave assignment phase.

While several different local search algorithms are used in Duque *et al*. (2012), tabu search generally identified the best quality solutions. As a result, we only compare our local search algorithm with the tabu search in Duque *et al*. (2012). In order to make the results comparable, we run our local search algorithm and tabu search with the same feasible solution generated in previous stages. Each of the local search algorithms is run 10 times and the best solution is reported. The computational results are reported in Table 2. The column 'Total heterogeneity reduction' is defined as:

$$Total\ heterogeneity\ reduction = \frac{h(inital\ solution) - h(final\ solution)}{h(inital\ solution)} \tag{8}$$

where h represents the total within-class heterogeneity to evaluate the improvement of total within-class heterogeneity by local search algorithms. Duque *et al*. (2012) also employed the total heterogeneity reduction (8) to compare multiple local search algorithms. For datasets lattice 100, lattice 529, and California counties, our new local search algorithm leads to an average of 10.92%, 1.47%, 11.12% more total heterogeneity reduction for all three thresholds, respectively. For lattice 1024, our new local search algorithm results in 2.19% and 3.34% more

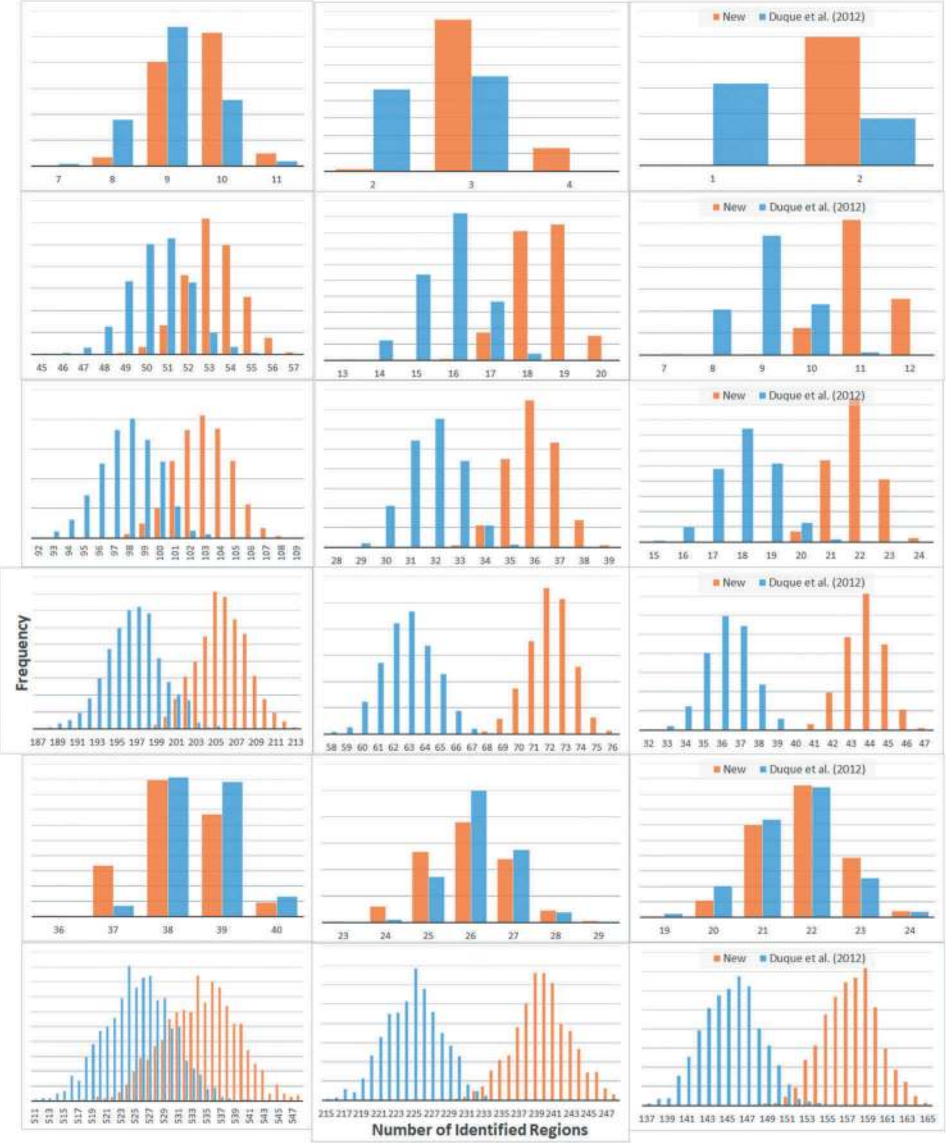

Figure 5. Distribution of the number of identified regions by the new region growth algorithm and the algorithm in.Duque *et al*. (2012).

total heterogeneity reduction for $T = 100$ and 500, respectively. For lattice 2025, it performs 0.18% and 0.39% better for $T = 300$ and 500, respectively. For US counties, tabu search performs better for all three thresholds with 0.72%, 1.62% and 0.55% more total heterogeneity reduction. This is not very surprising given the extensive search space of tabu search (Grover 1989; Edelkamp and Schrödl 2012). Column 'Running time' reports the computational time to run the local search algorithm. The tabu search takes more time in all scenarios except one for lattice 100 and one for California counties. The speedup of our new local search algorithm compared with tabu search is substantial for larger data sets. For example, the speedup for

Table 1. Computational time of region growth and enclave assignment for the new algorithm and the algorithm in Duque *et al.* (2012).

		Region growth time (seconds)		Enclave assignment (seconds)	
Dataset	Threshold	New algorithm	Duque *et al.* (2012)	New algorithm	Duque *et al.* (2012)
Lattice 100	100	0.37	1.87	0.07	0.02
Lattice 100	300	0.79	1.93	0.16	0.37
Lattice 100	500	1.02	2.06	1.36	0.24
Lattice 529	100	2.50	10.52	0.66	0.05
Lattice 529	300	3.33	11.26	1.44	0.16
Lattice 529	500	3.48	12.90	1.09	0.19
Lattice 1024	100	4.17	21.08	0.12	0.12
Lattice 1024	300	4.88	24.65	0.15	0.14
Lattice 1024	500	6.00	27.59	0.30	0.23
Lattice 2025	100	8.22	45.16	0.43	0.15
Lattice 2025	300	7.85	56.74	0.40	0.40
Lattice 2025	500	8.58	71.56	0.14	0.43
CA counties	100,000	0.80	0.72	0.03	0.03
CA counties	300,000	0.34	0.91	0.02	0.01
CA counties	500,000	0.28	0.91	0.01	0.01
US counties	100,000	12.69	99.05	0.36	0.22
US counties	300,000	12.36	245.52	0.29	1.72
US counties	500,000	15.25	473.18	0.32	1.43

Table 2. Computational results of new local search algorithm and tabu search algorithm.

		Total heterogeneity reduction		Running time (seconds)	
Dataset	Threshold	New algorithm	Tabu search	New algorithm	Tabu search
Lattice 100	100	30.99%	13.12%	0.24	2.67
Lattice 100	300	12.99%	11.37%	0.17	0.14
Lattice 100	500	32.32%	19.04%	0.83	4.66
Lattice 529	100	30.10%	29.66%	2.69	124.24
Lattice 529	300	23.17%	22.31%	3.09	17.50
Lattice 529	500	28.24%	25.12%	6.79	27.01
Lattice 1024	100	24.58%	22.40%	5.66	53.80
Lattice 1024	300	25.08%	26.87%	11.18	67.35
Lattice 1024	500	21.17%	17.82%	7.71	36.77
Lattice 2025	100	28.00%	28.46%	13.27	1560.19
Lattice 2025	300	24.12%	23.94%	20.39	240.00
Lattice 2025	500	25.84%	25.45%	28.90	1262.35
CA counties	100,000	17.82%	2.59%	0.09	0.04
CA counties	300,000	36.78%	29.79%	0.13	0.13
CA counties	500,000	42.64%	31.52%	0.11	0.88
US counties	100,000	28.54%	29.26%	39.66	3641.65
US counties	300,000	27.99%	29.62%	39.70	887.40
US counties	500,000	21.30%	21.85%	72.38	3383.63

lattice 2025 ranges from 12 to 118 and for US counties it ranges from 22 to 92. This suggests that our new local search algorithm is much more computationally efficient but can still guarantee the great solution quality in comparison with tabu search.

In addition to comparing the new algorithm and the original algorithm by phase to phase, we also conducted an end-to-end comparison by running the two complete algorithms with the same input data, and reported the final objective values (Equation 1), and total computational time in Table 3. Clearly, the new algorithm identifies better solutions with the same or lower objective value for all test datasets due to the much larger number of identified regions and generally smaller total within-region homogeneity. The

Table 3. Computational results of the new algorithm and the original algorithm in Duque *et al.* (2012).

		Objective value (Equation (1))		Total running time (seconds)	
Dataset	Threshold	New algorithm	Duque *et al.* (2012)	New algorithm	Duque *et al.* (2012)
Lattice 100	100	−1.10E+11	−1.10E+11	0.70	4.58
Lattice 100	300	−4.00E+10	−3.00E+10	1.13	2.46
Lattice 100	500	−2.00E+10	−2.00E+10	3.24	6.98
Lattice 529	100	−5.80E+14	−5.50E+14	5.99	134.93
Lattice 529	300	−2.00E+14	−1.80E+14	7.97	29.02
Lattice 529	500	−1.20E+14	−1.10E+14	11.47	40.20
Lattice 1024	100	−1.08E+17	−1.05E+17	10.10	75.26
Lattice 1024	300	−3.90E+16	−3.50E+16	16.47	92.37
Lattice 1024	500	−2.40E+16	−2.10E+16	14.29	64.81
Lattice 2025	100	−2.13E+18	−2.05E+18	22.43	1605.93
Lattice 2025	300	−7.70E+17	−6.80E+17	29.18	297.72
Lattice 2025	500	−4.70E+17	−4.00E+17	38.05	1334.72
CA counties	100,000	−4.10E+08	−4.00E+08	1.17	1.01
CA counties	300,000	−2.90E+08	−2.90E+08	0.58	1.23
CA counties	500,000	−2.50E+08	−2.40E+08	0.53	1.97
US counties	100,000	−5.49E+27	−5.39E+27	139.35	3819.14
US counties	300,000	−2.49E+27	−2.35E+27	139.52	1217.11
US counties	500,000	−1.66E+27	−1.53E+27	184.67	3949.14

computational gain in comparison with the original algorithm is significant and the average computational speedup of the new algorithm is 13. The identified neighborhoods for Lattice 100 with 100 threshold and US counties with 500,000 are presented in Figure 6.

Discussion and Conclusions

The last several decades have borne witness to three important trends in urban social science. The first–rapidly expanding data resources–is not limited to the urban context. Indeed, in recent years exploding volumes of data have led to the rapid development of techniques for both Big Data analysis and the data pipelining process. In urban research, however, this trend is also accompanied by (1) an increasing topical focus on neighborhoods and the important roles they play in human development and global sustainability, and (2) an increasing awareness of linked and multilevel spatial processes and the development of analytical techniques used to study them (Raudenbush and Bryk 2002, Raudenbush 2003, Harris *et al*. 2007, She *et al*. 2017, Zhong *et al*. 2019). In practice, these trends mean that the *problem size* in quantitative geography is increasing by orders of magnitude. Put differently, researchers today seek answers to questions about multiscalar neighborhood growth and change, persistent neighborhood inequality in high-performing economies, or neighborhood processes that link together places, actors and institutions within a single modeling framework. Addressing these challenges requires not only increasingly powerful computational platforms but also more efficient and performant implementations of the fundamental algorithms for urban neighborhood research. In this paper, we present one such advance.

The max-p-regions algorithm is designed to partition a study area into the largest possible set of mutually exclusive regions (or neighborhoods) that still satisfy an internal homogeneity constraint. Since its inception in 2012 (Duque *et al*. 2012a), the max-p-regions algorithm has been applied in a range of urban and social contexts including

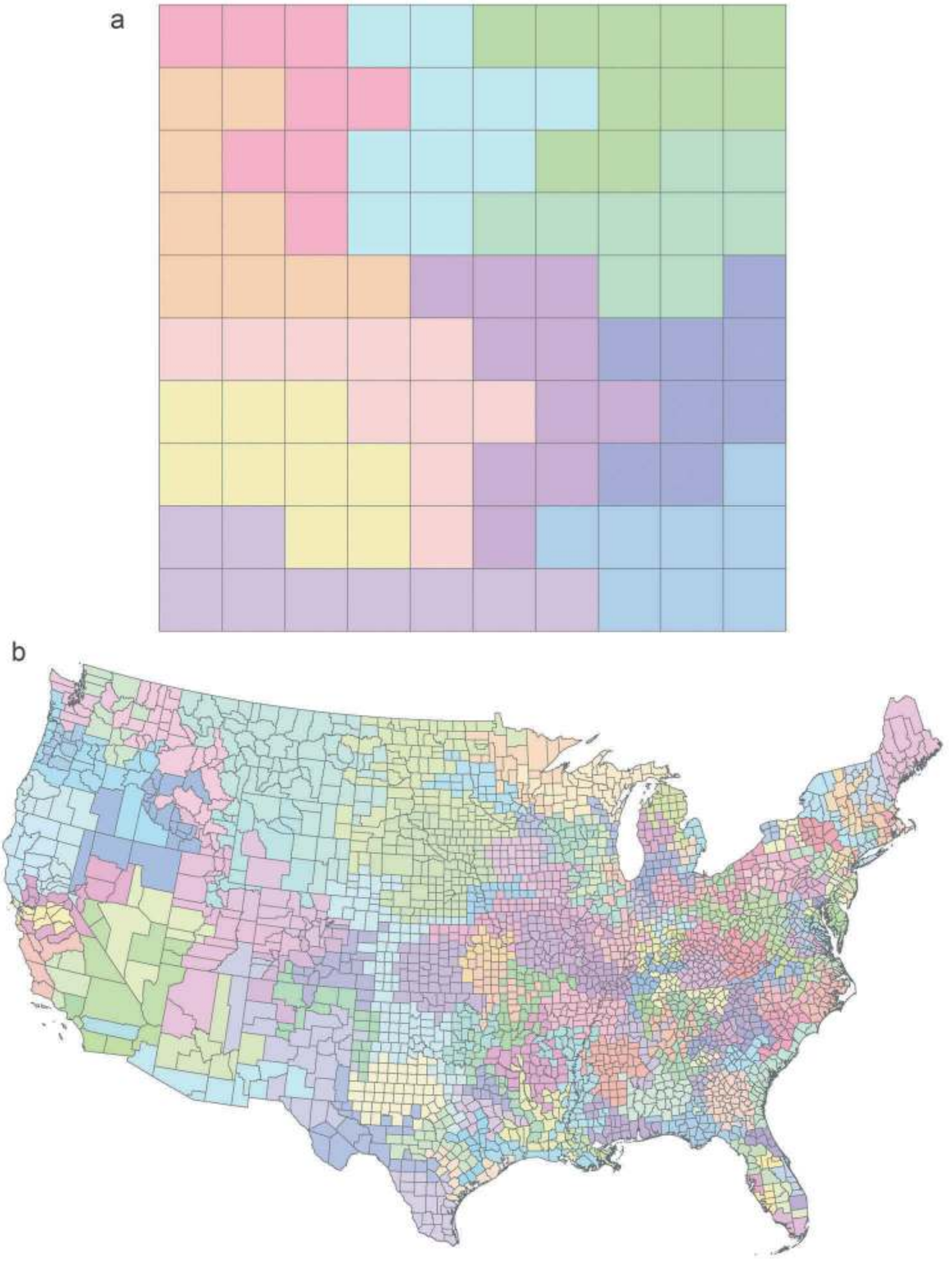

Figure 6. Neighborhood delineation; (a) Lattice 100 with 100 threshold; (b) US counties with 500,000 threshold.

urban slum delineation (Duque *et al.* 2012b), neighborhood dynamics (Rey *et al.* 2011), urban energy assessment (Reyna *et al.* 2016), regional inequality analysis, and more recently has been extended to problems that address network connectivity (She *et al.* 2017) and interregional comparisons (Rey and Sastré-Gutiérrez 2010). Despite the important findings advanced by these studies, we argue that the current implementation is waning in utility since it is unable to accommodate the massive data requirements inherent in modern urban scholarship.

In this paper, we have developed a new solution algorithm for max-p that can substantially reduce its computation time, and thus facilitates a much broader set of use-cases and larger volume of input data. For metropolitan-scale comparative work that typically requires several large datasets and potentially a dozen or more model runs to include multiple specifications, robustness checks (and potential for pilot errors), this means that scholars are now able to leverage max-p to address research problems in hours or days that would previously take weeks or months. Beyond its substantial improvement in runtime, however, our new algorithm also improves solution quality substantially by identifying much larger number of regions that also realize smaller within-region heterogeneity in comparison with the original algorithm in Duque *et al.* (2012).

Note

1. GeoDa contains a highly performant C++ implementation of max-p (https:https://geodacenter.github.io/workbook/8_spatial_clusters/lab8.html). We choose to implement our enhanced version in Python in order to compare it with the original PySAL implementation which was also implemented in Python. This lets us efficiently explore a prototype and alternative experimental designs.

Acknowledgments

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This material is based upon work supported by the National Science Foundation under grant [1831615].

Notes on contributors

Ran Wei is an Assistant Professor in the School of Public Policy and a founding member of the Center for Geospatial Sciences at the University of California, Riverside. Her areas of emphasis include GIScience, urban and regional analysis, spatial analysis, optimization, geovisualization, high performance computing and location analysis. Substantively, she has focused on a range of national and international issues, including urban/regional growth, transportation, public health, crime, housing mobility, energy infrastructure, and environmental sustainability.

Sergio Rey is Professor in the School of Public Policy and Founding Director of the Center for Geospatial Sciences at the University of California, Riverside. Rey's research interests focus on the development, implementation, and application of advanced methods of spatial and space-time data analysis. His substantive foci include regional inequality, convergence and growth dynamics as well as neighborhood change, segregation dynamics, spatial criminology and industrial networks.

Rey is the creator and lead developer of the open source package STARS: Space-Time Analysis of Regional Systems as well as co-founder and lead developer of PySAL: A Python Library for Spatial Analysis.

Elijah Knaap is the Associate Director of the Center for Geospatial Sciences at the University of California-Riverside. He holds a Bachelor's degree in sociology, and a Master's and PhD in Urban and Regional Planning, all from the University of Maryland.His work blends spatial data science with classic social theory to study issues of urban inequality, segregation, and neighborhood dynamics. In addition to his research, Eli is a core developer of the Python Spatial Analysis Library (PySAL)

ORCID

Sergio Rey http://orcid.org/0000-0001-5857-9762

Data and Codes Availability Statement

The data and codes that support the findings of this study are available with the identifier(s) at https://doi.10.6084/m9.figshare.11253152.v2.

References

Cano-Belmán, J., Ríos-Mercado, R.Z., and Salazar-Aguilar, M.A., 2012. Commercial territory design for a distribution firm with new constructive and destructive heuristics. *International Journal of Computational Intelligence Systems*, 5 (1), 126–147.

Duque, J.C., Anselin, L., and Rey, S.J., 2012. The max-p-regions problem. *Journal of Regional Science*, 52 (3), 397–419. doi:10.1111/j.1467-9787.2011.00743.x.

Duque, J.C., Church, R.L., and Middleton, R.S., 2011. The p-Regions Problem. *Geographical Analysis*, 43 (1), 104–126. doi:10.1111/j.1538-4632.2010.00810.x.

Duque, J.C., Ramos, R., and Surinach, J., 2007. Supervised regionalization methods: a survey. *International Regional Science Review*, 30 (3), 195. doi:10.1177/0160017607301605.

Edelkamp, S., and Schrödl, S., 2012. Chapter 2-basic searchalgorithms. *Heuristic search*, 47–87.

Feo, T.A. and Resende, M.G., 1995. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6 (2), 109–133. doi:10.1007/BF01096763.

Folch, D.C. and Spielman, S.E., 2014. Identifying regions based on flexible user-defined constraints. *International Journal of Geographical Information Science*, 28 (1), 164–184. doi:10.1080/13658816.2013.848986.

Garreton, M. and Sánchez, R., 2016. Identifying an optimal analysis level in multiscalar regionalization: A study case of social distress in Greater Santiago. *Computers, Environment and Urban Systems*, 56, 14–24. doi:10.1016/j.compenvurbsys.2015.10.007.

Glover, F., 1989. Tabu search—part I. *ORSA Journal on Computing*, 1 (3), 190–206. doi:10.1287/ijoc.1.3.190.

GNU Linear Programming Kit, (2012). *GLPK*. Available from: https://www.gnu.org/software/glpk/

González-Ramírez, R.G., *et al*., 2011. A hybrid metaheuristic approach to optimize the districting design of a parcel company. *Journal of Applied Research and Technology*, 9 (1), 19–35. doi:10.22201/icat.16656423.2011.9.01.441.

Guo, D., 2008. Regionalization with dynamically constrained agglomerative clustering and partitioning (REDCAP). *International Journal of Geographical Information Science*, 22 (7), 801–823. doi:10.1080/13658810701674970.

Guo, D. and Wang, H., 2011. Automatic region building for spatial analysis. *Transactions in GIS*, 15, 29–45. doi:10.1111/j.1467-9671.2011.01269.x.

Gurobi Optimization, 2019. *Gurobi optimizer reference manual*.

Harris, R., Johnston, R., and Burgess, S., 2007. Neighborhoods, ethnicity and school choice: developing a statistical framework for geodemographic analysis. *Population Research and Policy Review*, 26 (5–6), 553–579. doi:10.1007/s11113-007-9042-9.

Harris, R., Sleight, P., and Webber, R., 2005. *Geodemographics, GIS and neighbourhood targeting*. Vol. 8. Chichester, UK: John Wiley & Sons.

Kim, H., Chun, Y., and Kim, K., 2015. Delimitation of functional regions using ap-regions problem approach. *International Regional Science Review*, 38 (3), 235–263. doi:10.1177/0160017613484929.

Li, W., Church, R.L., and Goodchild, M.F., 2014. The p-compact-regions problem. *Geographical Analysis*, 46 (3), 250–273. doi:10.1111/gean.12038.

Openshaw, S. and Rao, L., 1995. Algorithms for reengineering 1991 census geography. *Environment & Planning A*, 27 (3), 425–446. doi:10.1068/a270425.

Raudenbush, S.W., 2003. The quantitative assessment of neighborhood social environments. *In*: *Neighborhoods and Health*. Oxford University Press, 112–131. doi:10.1093/acprof:oso/9780195138382.003.0005.

Raudenbush, S.W. and Bryk, A.S., 2002. *Hierarchical linear models: applications and data analysis methods*. Vol. 1. Thousand Oaks, USA: Sage.

Rey, S.J., *et al*., 2011. Measuring spatial dynamics in metropolitan areas. *Economic Development Quarterly*, 25 (1), 54–64. doi:10.1177/0891242410383414.

Rey, S.J. and Sastré-Gutiérrez, M.L., 2010. Interregional inequality dynamics in Mexico. *Spatial Economic Analysis*, 5 (3), 277–298. doi:10.1080/17421772.2010.493955.

Reyna, J.L., Chester, M.V., and Rey, S.J., 2016. Defining geographical boundaries with social and technical variables to improve urban energy assessments. *Energy*, 112, 742–754. doi:10.1016/j.energy.2016.06.091.

She, B., Duque, J.C., and Ye, X., 2017. The network-max-P-regions model. *International Journal of Geographical Information Science*, 31 (5), 962–981. doi:10.1080/13658816.2016.1252987.

Spielman, S.E., and Logan, J.R., 2013. Using high-resolution population data to identify neighborhoods and establish their boundaries. *Annals of the Association of American Geographers*, 103 (1), 67–84.

Spielman, S.E. and Singleton, A., 2015. Studying neighborhoods using uncertain data from the American community survey: a contextual approach. *Annals of the Association of American Geographers*, 105 (5), 1003–1025. doi:10.1080/00045608.2015.1052335.

Zhong, Q., *et al*., 2019. A multiobjective optimization model for locating affordable housing investments while maximizing accessibility to jobs by public transportation. *Environment and Planning B: Urban Analytics and City Science*, 46 (3), 490–510. doi:10.1177/2399808317719708.