An Iterated Local Search Algorithm for the Districting Problem

Authors: Kong Yunfeng, Kong Yunfeng

Date: 2022-03-29T17:34:07+00:00

Abstract

The regionalization problem involves partitioning a specific geographic region into several spatially contiguous zones to satisfy the fundamental principle of minimizing intra-zone differences while maximizing inter-zone differences, and finds wide applications in geography, environmental science, ecology, economics, agriculture, urban studies, and other fields. Over the past 60-plus years, scholars have attempted to establish various mathematical models for regionalization problems and have designed a series of solution algorithms, including exact algorithms, clustering-based algorithms, heuristic algorithms, and tree graph-based algorithms. To address the limitation that existing algorithms struggle to simultaneously achieve computational efficiency and solution quality, this paper proposes a regionalization algorithm based on Iterated Local Search (ILS). The main mechanisms of this algorithm include: improving zone quality through neighborhood unit movement; accelerating computational speed by rapidly calculating zone variance with reference to central units; employing a perturbation mechanism to escape local optima; updating zone centroids to enhance the objective value of zoning schemes; and utilizing population search to explore a larger solution space; throughout all algorithmic steps, spatial contiguity of zones is maintained through zone repair. Testing on 55 benchmark cases demonstrates that the ILS algorithm achieves superior solution quality compared to the ARISEL and SKATER algorithms, while requiring significantly less computational time than the ARISEL algorithm. A multi-criteria climate regionalization experiment further validates the practical utility of the ILS algorithm. The ILS algorithm presented in this paper balances both zone quality and computational efficiency, and permits a single zone to contain multiple spatially contiguous and relatively large regions, thereby offering flexibility and practicality.

An Iterated Local Search Algorithm for the Regionalization Problem

Yunfeng Kong
Key Laboratory of Geospatial Technology for the Middle and Lower Yellow River Regions, Ministry of Education, Henan University, Kaifeng, Henan, 475000

Abstract

Regionalization involves partitioning a specific geographic area into several spatially contiguous zones that minimize intra-regional variation while maximizing inter-regional differences. This fundamental principle finds extensive applications across geography, environmental science, ecology, economics, agriculture, urban planning, and related fields. Over the past six decades, scholars have attempted to establish various mathematical models for regionalization problems and have designed a series of solution algorithms, including exact methods, clustering-based approaches, heuristic algorithms, and tree-based methods. Addressing the limitation that existing algorithms struggle to balance computational efficiency with solution quality, this paper proposes a regionalization algorithm based on Iterated Local Search (ILS). The key mechanisms of this algorithm include: improving partition quality through neighboring unit movement; accelerating computation by rapidly calculating partition variance with reference to central units; employing perturbation mechanisms to escape local optima; updating partition centers to enhance objective values; utilizing population-based search to explore larger solution spaces; and maintaining spatial contiguity through partition repair operations at each step. Testing on 55 benchmark cases demonstrates that the ILS algorithm achieves superior solution quality compared to ARISEL and SKATER algorithms while requiring substantially less computational time than ARISEL. A multi-indicator climate regionalization experiment further validates the practical utility of the ILS algorithm. The proposed ILS algorithm balances partition quality and computational efficiency, and allows a single partition to contain multiple spatially contiguous and relatively large areas, offering both flexibility and practicality.

Keywords: regionalization problem; iterative local search; benchmark testing; case study

1 Research Background

Regionalization represents a fundamental problem in geography. It involves observing and studying regional complexes from a spatial perspective, exploring the formation, development, differentiation, combination, division, and merging of regional units, and synthesizing processes and types into a coherent framework [?]. Over the past century, significant advances have been made in regionalization theory, methodology, and applications, with widespread use in geography, environmental science, ecology, economics, agriculture, urban planning, cartography, and spatial statistics. With China’ s rapid socio-economic development, regionalization remains a foundational research area essential for strategic decision-making, planning, management, and policy formulation at national and regional levels.

Since the 1960s, regionalization research has focused on two major challenges: establishing theoretical foundations and determining partition boundaries. The former relies on understanding spatial patterns, structures, processes, mechanisms, and heterogeneity patterns of geographic phenomena to define regionalization objectives, principles, and indicators. The latter constitutes quantitative analysis, employing cartography, spatial analysis, spatial clustering, and spatial optimization to scientifically delineate boundaries. Liu et al. (2005) conducted an in-depth analysis of key scientific and technical issues in developing China’ s comprehensive regionalization scheme [?], while Zheng et al. (2008) elaborated on the connotation of physical geographic regionalization and proposed paradigms and key scientific questions [?], providing valuable guidance for regionalization across various domains.

The regionalization problem involves partitioning a specific geographic area into several spatially contiguous regions that minimize intra-regional differences and maximize inter-regional differences. Essentially, it is a clustering problem with added spatial contiguity constraints. Since clustering problems are already computationally complex, the spatial contiguity constraint makes regionalization even more challenging. Since the 1960s, scholars have developed various mathematical models and algorithms, including exact methods, clustering-based approaches, metaheuristic algorithms, tree-based methods, and hybrid heuristics [?]. However, these algorithms suffer from notable limitations: exact methods have excessive computational complexity; classical clustering algorithms struggle to handle spatial contiguity; tree-based methods rely solely on adjacency relationships, making it difficult to ensure regionalization quality after partitioning; and metaheuristic algorithms, while producing good solutions, require excessive computation time. To address these limitations, this paper proposes a new regionalization algorithm that ensures both solution quality and reduced computational complexity.

2 Literature Review

Regionalization problems are also known as spatial classification problems, spatial clustering problems, spatial aggregation problems, spatial districting problems, or zone design problems. Despite terminological differences, these concepts share the same essential goal: partitioning geographic space into regions that satisfy specific constraints to determine optimal zoning schemes.

The complexity of mathematical modeling and solving regionalization problems primarily stems from spatial contiguity constraints. Scholars have formally defined decision variables, constraints, and objective functions for various regionalization needs, proposing multiple mathematical models [?, ?, ?, ?, ?, ?, ?, ?, ?]. Keane (1975) proved that regionalization problems with spatial contiguity constraints are NP-Hard [?], making them extremely computationally demanding.

The p-regions problem, which partitions n spatial units into p contiguous regions, represents a classic regionalization problem. It can be expressed through three mixed-integer programming (MIP) formulations: tree models, order models, and network flow models [?]. However, exact algorithms based on mathematical models can only solve small-scale problems with few spatial units. For instance, CPLEX computations show that for benchmark cases with $n = 49$ and $p = 3 \sim 10$, optimal solutions cannot be obtained within three hours [?].

Clustering-based regionalization methods include classical clustering analysis, distance-weighted clustering, M-means clustering, and agglomerative hierarchical clustering. The first three approaches are conceptually simple but inadequate for handling spatial contiguity, often sacrificing partition quality to ensure contiguity. Classical hierarchical clustering has been more successfully applied to regionalization through the following process: (1) initially treat each spatial unit as a separate region; (2) compute similarity between regions; (3) merge the most similar contiguous regions; and (4) repeat steps (2) and (3) until reaching the target number of regions. Various methods exist for computing inter-regional similarity in step (2), such as minimizing variance (Ward), similarity between closest units (single linkage), similarity between most dissimilar units (complete linkage), and similarity based on mean or median values (average linkage). Step (3) restricts merging to adjacent regions to maintain contiguity. This bottom-up merging strategy suits problems with uncertain numbers of regions, though the similarity computation method and spatial adjacency constraints significantly influence the resulting cluster tree [?].

Heuristic regionalization algorithms operate by first constructing a feasible solution and then iteratively improving it through neighborhood operators. The AZP method, proposed by Openshaw (1977), represents a classic approach that initially partitions n spatial units randomly into k regions, then attempts to reassign units to different regions while respecting contiguity constraints [?]. Essentially a hill-climbing algorithm, AZP' s search process easily becomes trapped in local optima.

To avoid local optima in neighborhood search, scholars have continuously improved algorithms by incorporating metaheuristic mechanisms such as simulated annealing [?, ?] and tabu search [?] to enhance search diversity and obtain higher-quality solutions. Duque and Church (2004) enhanced the tabu search algorithm to create ARISEL, which generates multiple initial regionalization solutions and selects high-quality ones for tabu search [?].

To reduce computational complexity, researchers have proposed tree-based heuristic algorithms that abstract regions as network graphs, simplify them into trees, and obtain contiguous regions through tree partitioning. Tree nodes represent spatial units, while edges represent adjacency relationships [?].

Maravalle and Simeone (1995) proposed the MIDAS algorithm, which generates tree T from graph G, deletes $p-1$ connections to obtain p subtrees representing contiguous regions [?]. Since the solution space on tree T is limited, MIDAS continuously adjusts T to T^* to improve solutions. Subsequently, Assunção et al. (2006) developed the SKATER algorithm based on minimum spanning trees [?]. Guo (2008) improved this approach with the REDCAP algorithm, proposing six dynamic tree generation methods: First-Order-SLK, First-Order-CLK, First-Order-ALK, Full-Order-SLK, Full-Order-CLK, and Full-Order-ALK, finding that Full-Order-CLK and Full-Order-ALK outperformed other methods [?].

Overall, clustering algorithms are simple to implement but either fail to guarantee spatial contiguity or sacrifice global optimization quality to maintain it. Heuristic algorithms are numerous with straightforward improvement strategies but limited optimization performance. Metaheuristic methods achieve higher performance but involve complex designs and low computational efficiency. Tree-based methods significantly improve efficiency but drastically reduce the search space, compromising regionalization quality. Aydin et al. (2021) designed benchmark test cases to evaluate AZP, AZP-SA, AZP-Tabu, ARISEL, SKATER, and REDCAP algorithms, finding that ARISEL achieved the highest overall quality but with slow computation, while SKATER offered good solution quality with very high efficiency [?]. Given the expanding applications of regionalization and its significant impact on regional planning and decision-making, developing more effective algorithms that ensure both quality and rapid computation is essential.

3 Problem Definition

Consider a geographic region consisting of n spatial units, denoted as set $U = \{1, 2, 3 \ldots n\}$. Each unit has m attributes, denoted as set $A = \{1, 2, 3 \ldots m\}$, where unit i has attribute values $a_{i1}, a_{i2}, a_{i3} \ldots a_{im}$. The region is to be partitioned into p spatially contiguous regions, denoted as set $C = \{1, 2, 3 \ldots p\}$, where region i contains geographic units c_i, satisfying $c_i \cap c_j = \phi(i \neq j)$ and $c_1 \cup c_2 \cup c_3 \cup \ldots c_p = U$, meaning any two regions do not overlap and each spatial unit must be assigned to a specific region. The objective is to minimize the sum of within-region variance of unit attributes:

$$f(C) = \sum_{i \in C} \sum_{j \in c_i} \sum_{k \in A} (a_{jk} - \bar{a}_{ik})^2$$

where $\bar{a}_{ik}$ in Equation (1) represents the mean value of attribute k for all units in region i.

Several practical considerations arise in regionalization practice. First, given differences in meaning and dimensionality among attributes, standardized unit attribute values are typically employed. Common standardization methods include standard deviation normalization, min-max scaling, and linear proportion methods. Second, acknowledging that attributes may have varying importance, weights can be assigned to each attribute. Let $b_{i1}, b_{i2}, b_{i3} \ldots b_{im}$ denote the standardized attribute values for unit i, and let w_k represent the weight of attribute k. The regionalization objective function becomes:

$$f(C) = \sum_{i \in C} \sum_{k \in A} \sum_{j \in c_i} w_k (b_{jk} - \bar{b}_{ik})^2$$

Generally, an R^2 metric can be calculated for each attribute to evaluate partition quality:

$$R_k^2 = 1 - \frac{\sum_{i \in C} \sum_{j \in c_i} w_k (b_{jk} - \bar{b}_{ik})^2}{\sum_{j \in U} \sum_{k \in A} w_k (b_{jk} - \bar{b}_k)^2}$$

where $\bar{b}_k$ is the mean value of attribute k. If standard deviation normalization is used, the mean $\bar{b}_k = 0$. Additionally, an overall R^2 metric can be computed to evaluate partition quality:

$$R^2 = 1 - \frac{\sum_{i \in C} \sum_{k \in A} \sum_{j \in c_i} w_k (b_{jk} - \bar{b}_{ik})^2}{\sum_{j \in U} \sum_{k \in A} w_k (b_{jk} - \bar{b}_k)^2}$$

4 Algorithm Design

Solving regionalization problems presents several challenges. First, as defined herein, the regionalization problem is a clustering problem with added spatial contiguity constraints, making it considerably more complex. Spatial contiguity judgment and repair constitute two frequently performed operations in regionalization algorithms. Second, computing the mean $\bar{b}_{ik}$ for region i in objective function (2) is computationally expensive, potentially reducing algorithm efficiency. To accelerate computation, attribute values of region centers can replace the means $\bar{b}_{ik}$. Determining center points for each region facilitates rapid evaluation of solution quality. Based on this analysis, this paper adopts a center-based regionalization algorithm while maintaining spatial contiguity of each region.

The Iterated Local Search (ILS) algorithm is selected as the framework for solving the regionalization problem. The ILS algorithm is conceptually simple, easy to implement, and effective for discrete optimization problems [?]. Starting from an initial solution, the algorithm iteratively performs perturbation and local search. Since local search easily becomes trapped in local optima, perturbation of the current position enables escape from these optima. Initial solution generation, local search, and perturbation constitute the basic modules of ILS. To enhance optimization performance, this paper extends the single-solution ILS to a population-based ILS. The improved ILS algorithm proceeds as follows:

Parameters: population size (*psize*), perturbation strength (*strength*), maximum non-improving iterations (*mloops*).

```
1. $Pop = GenerateInitialSolutions(psize)$;
2. $s_{best} = Best(Pop)$;
3. $notImpr = 0$;
4. **while** $notImpr < mloops$ **do**
5. Select a solution $s$ from $P$ randomly;
6. $s' = Perturbation(s, strength)$;
7. $s'' = LocalSearch(s')$;
8. $s^* = updateCenters(s'')$;
9. **if** $f(s^*) < f(s_{best})$: $s_{best}=s^*$, $notImpr=0$;
10. **else**: $notImpr += 1$;
11. $Pop = UpdatePopulation(Pop, s^*)$;
12. **end while**
13. **Output** $s$.
```

Step (1) employs the classic K-medoids algorithm to generate initial solutions. This algorithm randomly selects p spatial units as region centers and iteratively performs unit assignment and center updates until no centers change. Assignment is computationally simple, as each unit is assigned to its nearest center. However, due to study area shape and spatial distribution of geographic features, the resulting partitions may not guarantee spatial contiguity, necessitating contiguity judgment and repair operations.

Step (7) uses boundary unit movement for local search. This method attempts to move a boundary unit to an adjacent region, updating the current solution if the move reduces the regionalization objective. This operation must consider spatial contiguity to ensure the contiguity constraint remains satisfied after unit movement.

Step (6) performs solution perturbation. Common perturbation methods include destroying several regions, destroying a contiguous area, or destroying a proportion of boundary units, followed by solution repair. If repaired partitions fail to maintain spatial contiguity, additional contiguity repair is performed.

Compared to single-solution search algorithms, the improved ILS maintains a population of solutions. First, step (1) generates a set of initial solutions. Second, in each iteration, a solution is randomly selected from the population as the current solution for search (step 5). Third, after searching, the population is updated with the new solution (step 11). In population updating, solution quality is prioritized, followed by solution diversity, maintaining sufficient differences among population members. The population-based ILS maintains a set of diverse elite solutions, expanding the search space and improving solution quality, though convergence speed typically decreases and computation time increases moderately.

Since the algorithm evaluates partition objectives based on center units, step (8) attempts to update center units after local search to further reduce the objective value.

Spatial contiguity judgment is a critical step in regionalization algorithms. This paper uses spanning trees to assess partition contiguity [**?**, **?**]. If all units in a partition can form a spanning tree, the partition is contiguous. Considering special cases, this paper allows a partition to contain two or more relatively large contiguous areas. In Figure 1 (left), the blue, brown, and green partitions each consist of two parts. All blue and brown components have sufficiently large areas, making them acceptable as spatially contiguous partitions. However, in the green partition, one component is too small and is considered a spatially discontinuous fragmented unit. The contiguity judgment method proceeds as follows: (1) for a given partition, construct a spanning tree starting from any unit; (2) if some units cannot connect to the spanning tree, construct a new spanning tree for the remaining units; (3) repeat step (2) until no units remain; (4) compute the number of units and area for each spanning tree—if any tree' s unit count or area falls below a specified threshold, the region is considered discontinuous, and the corresponding units are identified as fragmented. For discontinuous partitions, fragmented units must be repaired by reassigning them to the nearest adjacent partition. In the left panel of Figure 1, the green partition contains a small patch of 5 units that can be treated as fragmented and reassigned to the neighboring blue partition, with the repaired partition shown in the right panel.

5.1 Benchmark Testing

Algorithm testing employs the benchmark case set provided by Aydin et al. (2021) [**?**]. This case set is generated from three regular grid maps with sizes of 120 (10$×12), 300(15×20), *and*1200(30×40)*cells.Thesemapsarepre* − *partitionedintoregions*, *andattributevaluesforeachcellaresimulatedbasedonthepartitions.Regionalizationisp*
grid map, irregularly shaped region cases are simulated with 5 regions and a simulation parameter of 3. In total, 55 regionalization cases are generated, with cell attribute values randomly simulated 100 times for each case. The simulation method assigns a mean attribute value to each region, sets adjacent region mean differences using parameters 2, 3, or 4, and simulates cell attribute values from a normal distribution with variance 1. Table 1 summarizes the benchmark cases, with detailed descriptions available in Aydin et al. (2021) [**?**]. Data and results for six algorithms, along with quality metrics and computation times, are available at https://doi.org/10.6084/m9.figshare.14067239.

Table 1. Characteristics of the benchmark instances

Map Name	Grid Size	Region Shape	Region Count	Simulation Parameter	Simulation Runs
G120	10$×12|A, B|5, 10, 15|2, 3, 4|100||G300|15×20|A, B|5, 10, 15|2, 3, 4|100||G1200|30×40|A, B|5, 10, 15|2, 3	Irregular	3		150

To visually understand the cases, Figure 2 illustrates four regionalization schemes: G120-5A, G300-10B, G1200-15A, and Blob. Scheme names combine map name, region count, and region shape. Figure 3 shows simulated values for these four schemes using simulation parameters 4, 2, 3, and 3, respectively, where color intensity represents attribute magnitude. Larger simulation parameters create greater differences between regions, making the pre-defined regions easier to identify; conversely, smaller parameters produce more challenging cases.

For each of the 100 simulations per case, the ILS algorithm is applied to obtain 100 regionalization solutions, computing the Adjusted Rand Index (ARI) and R^2 metric for each solution. ARI measures similarity between the obtained partition and the true partition, with values closer to 1 being better. The R^2 metric measures the relative magnitude of within-region variance, with values closer to 1 indicating better performance.

Table 2 presents the mean ARI and R^2 values across 100 runs for each case, comparing ILS with SKATER and ARISEL algorithms (with ARISEL and SKATER results taken from [?]). The results show that ILS generally outperforms ARISEL, which in turn outperforms SKATER. For difficult cases with simulation parameter 2, ILS demonstrates particularly significant advantages.

Table 2. ARI and R^2 indexes from 55 benchmark instances

[Table content showing comparative metrics across all case configurations···]

For the Blob case shown in Figure 3, ILS advantages are even more pronounced. Figure 4 displays two ILS solutions (left) and two SKATER solutions from ArcGIS 10.3 (right). ILS nearly perfectly recovers the pre-defined partitions, while SKATER confuses some regions. ArcGIS computation time is approximately 2.5–2.7 seconds, while ILS requires 5.1–7.1 seconds. The four regionalization metrics in Figure 4 are 0.872, 0.891, 0.567, and 0.719.

Table 3 compares computation times across the three algorithms (with ARISEL and SKATER times from [?]). The results indicate: (1) SKATER implemented in ArcGIS is fastest, with modest time increases as case size grows; (2) ARISEL requires the most time, with rapid time increases as case size grows; (3) ILS computation time exceeds SKATER but is substantially lower than ARISEL. Note that different computational environments prevent direct time comparisons, but the times reflect general algorithm efficiency.

Table 3. Comparison of the computation times

[Table content showing time comparisons across algorithms and case sizes…]

5.2 Climate Regionalization of the Huang-Huai-Hai Region

To further test the algorithm, the Huang-Huai-Hai region is selected for climate regionalization. This region includes the Yellow River, Huai River, and Hai River basins plus the Shandong Peninsula. The study uses 30-year average annual precipitation and temperature data at 15-arcminute resolution. The case area comprises 2,478 spatial units, as shown in Figure 5. Each unit contains 60 attribute values: 30 annual precipitation values and 30 annual temperature values.

Both ArcGIS 10.3 SKATER and the proposed ILS algorithm are applied for climate regionalization, with region counts set to 3, 4, 5, 6, 7, 8, 9, 10, 12, and 15. All attributes are equally weighted at 1. Both algorithms apply standard deviation normalization to the 60 attributes. After regionalization, R^2 metrics are computed for each individual attribute and for the standardized data overall.

Table 4 presents R^2 metrics for both algorithms, including minimum (MinR2), average (AvgR2), and maximum (MaxR2) values across the 60 attributes, plus overall R^2 and computation time. ILS demonstrates significantly higher regionalization quality metrics than SKATER, though SKATER maintains a substantial lead in computational efficiency.

Table 4. The R^2 indexes from the case study area

[Table content showing comparative metrics for different region counts…]

Figure 6 shows regionalization results for six regions from both algorithms. The results differ substantially in region shape, size, and boundaries. SKATER produces blocky regions, while ILS generates more strip-like regions that align with the overall spatial patterns of temperature, precipitation, and topography. The R^2 metric for ILS (0.855) significantly exceeds that of SKATER (0.804). SKATER' s focus on similarity between adjacent units, without considering relationships between non-adjacent units, creates limitations. ILS overcomes SKATER' s local focus, thereby improving regionalization quality.

6 Conclusion

This paper proposes an improved ILS algorithm for solving regionalization problems. The algorithm comprises initial solution generation, local search, population-based search, solution perturbation, and center update components, with spatial contiguity judgment and repair operations ensuring all partitions remain contiguous. By evaluating partition objectives based on center points, the algorithm substantially reduces objective function computation and improves efficiency. Population search, perturbation, and center updates expand the solution space and enhance partition quality. Benchmark testing demonstrates that the improved ILS algorithm produces superior results compared to SKATER and ARISEL algorithms. For multi-attribute climate regionalization without clear boundaries, ILS achieves significantly better objective values than SKATER, with results consistent with regional patterns of topography, temperature, and precipitation.

The improved ILS algorithm design offers several distinctive features and advantages. First, compared to AZP, AZP-SA, AZP-Tabu, and ARISEL, ILS uses partition centers for objective computation, avoiding frequent calculation of within-region attribute means during local search and dramatically improving computational efficiency. Second, while AZP is a simple heuristic and AZP-SA/AZP-Tabu are metaheuristics that improve search strategy and quality, ARISEL uses multiple initial solutions and selects high-quality ones for tabu search to expand the search space. The improved ILS employs population-based search, perturbation, and center updates, distinguishing it from existing designs and leveraging established optimization mechanisms. Third, SKATER considers only similarity between adjacent units, drastically reducing search space and achieving high efficiency, while REDCAP considers only inter-regional similarity and performs bottom-up clustering. ILS overcomes the short-sightedness of SKATER and REDCAP through search and perturbation, facilitating discovery of high-quality partitions. These characteristics ensure both partition quality and reduced computational complexity.

Given the spatial gradualness of geographic phenomena, complexity of geographic systems, and scale-dependency of spatial differentiation patterns, application of this regionalization algorithm should be grounded in regional research: understanding geographic patterns and processes, comprehending regional characteristics, clarifying regionalization tasks and objectives, and selecting appropriate indicators. Future research directions include determining optimal region counts, selecting data standardization methods, choosing appropriate dissimilarity functions, and developing general regionalization methods and software tools based on this algorithm.

References

[1] Zheng D, Ge Q, Zhang X, et al. Review and prospect of regionalization work in China [J]. Geographical Research, 2005, 24(3): 330-344.

[2] Liu Y, Zheng D, Ge Q, et al. Understanding of key issues in China' s comprehensive regionalization research [J]. Geographical Research, 2005, 24(3): 321-329.

[3] Zheng D, Ouyang, Zhou C. Understanding and thinking on physical geographic regionalization methods [J]. Acta Geographica Sinica, 2008, 63(6): 563-573.

[4] Duque J C, Ramos R, Suriñach J. Supervised regionalization methods: A survey [J]. International Regional Science Review, 2007, 30(3): 195-220.

[5] Cliff A D, Haggett P, Ord J K, et al. Elements of Spatial Structure: A Quantitative Approach [M]. New York: Cambridge University Press, 1975.

[6] Keane M. The size of the region-building problem [J]. Environment and Planning A, 1975, 7(5): 575-577.

[7] Wright J, Revelle C, Cohon J. A multiobjective integer programming model for the land acquisition problem [J]. Regional Science and Urban Economics, 1983, 13(1): 31-53.

[8] Cova T J, Church R L. Contiguity constraints for single-region site search problems [J]. Geographical Analysis, 2000, 32(4): 306-329.

[9] Williams J C. A Zero-One Programming Model for Contiguous Land Acquisition [J]. Geographical Analysis, 2002, 34(4): 330-349.

[10] Shirabe T. A model of contiguity for spatial unit allocation [J]. Geographical Analysis, 2005, 37(1): 2-16.

[11] Duque J C, Church R L, Middleton R S. The p-Region Problem [J]. Geographical Analysis, 2011, 43, 104–126.

[12] Li W, Church R L, Goodchild M F. The p-compact-regions problem [J]. Geographical Analysis, 2014, 46(3): 250-273.

[13] Guo D. Regionalization with dynamically constrained agglomerative clustering and partitioning (REDCAP) [J]. International Journal of Geographical Information Science, 2008, 22(7): 801-823.

[14] Openshaw S. A geographical solution to scale and aggregation problems in region-building, partitioning and spatial modelling [J]. Transactions of the Institute of British Geographers, 1977: 459-472.

[15] Browdy M H. Simulated annealing: an improved computer model for political redistricting [J]. Yale Law & Policy Review, 1990: 163-179.

[16] Openshaw S, Rao L. Algorithms for reengineering 1991 Census geography [J]. Environment and planning A, 1995, 27(3): 425-446.

[17] Duque J C, Church R L. A new heuristic model for designing analytical regions[C]//North American Meeting of the International Regional Science Association, Seattle. 2004.

[18] Guo R. Two-dimensional ordered clustering method and its application in regional mapping [J]. Journal of Wuhan College of Geodesy and Cartography, 1985, (2): 21-29.

[19] Maravalle M, Simeone B. A spanning tree heuristic for regional clustering [J]. Communications in statistics-theory and methods, 1995, 24(3): 625-639.

[20] Assunção R M, Neves M C, Câmara G, et al. Efficient regionalization techniques for socio-economic geographical units using minimum spanning trees [J].

International Journal of Geographical Information Science, 2006, 20(7): 797-811.

[21] Aydin O, Janikas M V, Assunção R M, et al. A quantitative comparison of regionalization methods [J]. International Journal of Geographical Information Science, 2021, 35(11): 2287-2315.

[22] Lourenço H R, Martin O, Stützle T. Iterated Local Search: Framework and Applications [M] // Gendreau M, Potvin JY. eds. Handbook of Metaheuristics, 2nd. Edition. New York: Springer, 2010, 363-397.

[23] Xiao N. A unified conceptual framework for geographical optimization using evolutionary algorithms. Annals of the Association of American Geographers, 2008, 98(4): 795-817.

[24] Liu, Y., Cho, W. K., and Wang, S., 2016. PEAR: a massively parallel evolutionary computation approach for political redistricting optimization and analysis. Swarm and evolutionary computation, 30, 78-92.

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv —Machine translation. Verify with original.