引用格式：孔云峰.基于迭代局部搜索的区划问题算法研究[J].地球信息科学学报,2022,24(9):1730-1741. [Kong Y F. An improved iterative local search algorithm for the regionalization problem[J]. Journal of Geo-information Science, 2022,24(9):1730-1741.] **DOI:**10.12082/dqxxkx.2022.220139

基于迭代局部搜索的区划问题算法研究

孔云峰[1,2*]

1. 河南大学黄河中下游数字地理技术教育部重点实验室，开封 475000；2. 河南大学地理与环境学院，开封 475000

An Improved Iterative Local Search Algorithm for the Regionalization Problem

KONG Yunfeng[1,2*]

1. Key Laboratory of Geospatial Technology for the Middle and Lower Yellow River Regions, Ministry of Education, Henan University, Kaifeng 47500, China; 2. College of Geography and Environmental Science, Henan University, Kaifeng 475000, China

Abstract: Regionalization is to divide a large geographic area into a number of homogenous and spatially contiguous regions. It has been widely used in fields such as geography, cartography, ecology, environment management, socio-economy, and urban planning. Since the general regionalization problem has been proven to be NP-Hard, various models and solution methods for regionalization have been proposed since 1960s. The regionalization methods can be classified into four categories: exact, clustering-based, heuristic, and tree-based. However, the commonly used regionalization algorithms, such as AZP, AZP-SA, AZP-Tabu, ARISEL, SKATER, and REDCAP, are difficult to solve the problem in an effective and efficient manner simultaneously. An improved iterative local search algorithm is proposed in this paper for the regionalization problem. There are six key mechanisms in the new algorithm: the search of moving boundary units to improve the current solution, the center-based approach to accelerate the computation of solution objective, the solution perturbation to escape from the state of local optimum, the frequent update of regional centers to reevaluate the solution, the population-based search to explore larger solution space, and the region repair to keep spatially contiguous regions. The regionalization experimentations on 55 benchmark instances show that the proposed algorithms outperform ARISEL algorithm and SKATER algorithm in terms of sum-squared errors and adjusted Rand index. A case study of the climate regionalization using 60 attributes illustrates that the improved ILS is effective to delineate climate regions, and outperforms the well-known algorithms such as SKATER, REDCAP, and ARISEL.

Key words: regionalization; regionalization problem; objective function; iterative local search; benchmark test; case study

***Corresponding author:** KONG Yunfeng, E-mail: yfkong@henu.edu.cn

摘要：区划问题是将特定地理区域划分为若干空间连续的分区，满足分区内差异最小和分区间差异最大这一基本原则，广泛应用于地理、环境、生态、经济、农业、城市等领域。1960s以来，学者尝试建立各种区划问题数学模型，设计了一系列的求解算法，代表性的算法主要有：AZP、ARISEL、SKATER和REDCAP。本文提出了一个基于迭代局部搜索(ILS)的区划问题算法，进

收稿日期：2022-03-30；**修回日期：**2022-05-10.

基金项目：国家自然科学基金项目(41871307)。[**Foundation item:** National Natural Science Foundation of China, No.41871307.]

作者简介：孔云峰(1967—)，男，河南洛阳人，教授，主要从事空间分析、空间优化等研究。E-mail: yfkong@henu.edu.cn

一步提升算法性能。该算法主要机制包括：邻域单元移动搜索改进分区质量；参照中心单元快速计算分区方差，提升算法速度；使用扰动机制跳出当前解局部最优状态；更新分区中心点提升分区方案目标值；使用群搜索探索更大的解空间；以及算法各步骤中通过分区空间连续判断和破碎修复保持分区空间连续。55个基准案例测试表明：ILS算法求解质量优于ARISEL和SKATER算法。一个多指标气候分区实验也表明：ILS算法求解质量优于SKATER、REDCAP和ARISEL算法。

关键词：区划；区划问题；目标函数；迭代局部搜索；基准测试；案例研究

1 引言

区划(Regionalization)是地理学中的一个基本问题。区划是从区域角度观察和研究地域综合体，探讨区域单元的形成发展、分异组合、划分合并和相互联系，是对过程和类型综合研究的概括和总结[1]。上百年来，区划理论、方法与应用研究取得了显著的进展[1]，广泛应用于地理、环境、生态、经济、农业、城市等领域。随着我国社会经济的快速发展，为满足国家和地区各行业的战略决策、规划、管理和政策制定，区划仍是一个基础性的研究领域。

纵观20世纪60年代以来国内外区划研究进展，地理区划有两大重点和难点：区划的理论分析和分区边界的确定。前者针对区划需求，探索地理现象的空间格局、结构、过程、机理及其地域分异规律，确定区划目标、等级和原则，并遴选区划指标，指导地理区划实践。后者属于定量分析范畴，是在理论研究的基础上，运用地图制图、空间叠加、空间聚类、模型优化等手段，科学合理地划定分区边界。刘燕华等[2]对于编制中国综合区划方案所要解决的重要科学技术问题进行了深入分析。郑度等[3]深入阐述了自然地理区划的内涵，提出了自然地理区划范式及关键科学问题。在区划理论与方法研究取得长足进展的同时，也有学者指出：我国区划理论与方法研究仍存在薄弱环节，如指标体系的客观性不足、定量化程度不高、依靠主观经验划定重要边界等[4]。

本文重点关注区划工作中如何确定分区边界的问题。针对某一具体的区划工作，首先需要进行理论分析，确立区划目标、原则和指标体系。之后，在理论分析指导下，按照区域内基本地理单元的指标数据确定分区边界。区划问题(Regionalization Problem)是确定分区边界所涉及的数学问题。依据特定地理区域内基本空间单元的单个属性或多维属性，区划问题将该区域划分为若干空间连续的分区，满足分区内差异最小化和分区间差异最大化这一基本原则[5]。自20世纪60年代开始，学者尝试建立各种数学模型，并设计了一系列算法，包括精确算法、基于聚类的算法、元启发式算法、树图分割算法和混合启发算法[5]。代表性的区划算法主要有：AZP、AZP- SA、AZP- Tabu、ARISEL、SKATER 和REDCAP[6]。然而，这些算法存在以下局限：树图分割算法(SKATER和REDCAP)仅依据相邻单元生成树图，分割后难以保证区划质量；元启发式算法(AZP-Tabu、ARISEL)求解质量较好，但算法计算复杂度高，计算时间偏长。针对现有区划问题算法这些局限，本文提出一个新的区划算法，目标是进一步提升区划质量。

2 区划问题算法研究进展

区划问题本质上是一个增加了分区空间连续约束的聚类问题。早在20世纪60年代，学者就比较了地理分区与地理分类的关系，认为分区是分类的特殊形式[7]，开始形式化、定量化地描述区划问题，明确了区划目标和分区约束条件[8-10]。之后，针对各种区划需求，有学者提出了多个区划问题数学模型，如P分区(p-regions)问题、P紧凑分区(p-compact-regions)问题等。然而，空间连续性约束区划问题是一类NP-Hard问题[11]，模型求解的计算复杂度极高。P-regions问题将n个空间单元划分为p个连续区域，是一个经典的区划问题。该问题可表达为3种混合整型规划(MIP)模型：树模型、次序模型和网络流模型[12]。然而，基于数学模型的精确算法仅能够求解空间单元数量很少的小规模问题。CPLEX模型计算表明：针对n=49和p=3~10的基准案例，3 h的计算时间均不能获得最优解[12]。基于次序模型，Li等[13]提出了p紧凑分区(p-compact-regions)问题，将区划划分为p个空间连续的分区，目标是最小化分区内单元属性的差异。因模型计算复杂度过高，作者设计了一个带记忆的随机贪心与和边重连搜索(MERGE)算法。

早期的地理区划多采用聚类分析方法，如K-means方法、距离加权聚类分析方法和层次聚类方法。前2种方法，思路简单，但处理空间连续性分区的能力不足，以牺牲分区质量为代价保证分区空间

连续。经典的层次聚类方法较为成功地应用于区划，算法流程如下：① 首先将每一个空间单元作为一个分区；② 计算各分区间的相似度；③ 寻找近似度最接近且连续的2个分区，把他们合并为一个分区；④ 重复步骤②和③，直到分区数量满足区划目标。步骤②中分区相似度的计算有多种方法，如方差最小[14]、2个分区中最接近单元的相似度(single linkage)、2个分区中2个差异最大单元的相似度(complete linkage)、2个分区中单元均值或中值的相似度(average linkage)等[15]。步骤③限制相邻区域合并，保证分区的连续性。该方法采用自下而上的分区合并策略，适合于分区数量不确定的区划问题，但步骤②相似度计算方法和步骤③空间邻接约束对于聚类树的形成影响很大[16]。

启发式区划算法的基本原理是：先构造一个可行的区划方案，再使用邻域算子进行迭代搜索改进。第一步使用区域种子生长、聚类分析或其他简单算法构造一个可行的区划解，第二步根据当前的分区方案和空间单元间的空间关系，尝试进行空间单元的移动迭代地改进区划方案。AZP方法是Openshaw[17]提出的一个经典区划算法：将n个空间单元随机划分为k个区域，在顾及分区空间连续约束的前提下，尝试将某个单元重新分配到另一个区域，持续改进区划方案。本质上，该算法属于爬山算法，搜索过程容易陷入局部最优而过早停滞。

为避免邻域搜索过程陷入局部最优，学者不断改进算法，通过模拟退火[18-19]、禁忌[19]等元启发机制，提升搜索过程的多样性，从而获得较高质量的区划方案。Duque等[20]改进禁忌算法为ARISEL算法。该算法使用简单方法提供多个初始区划方案，选择高质量区划方案进行禁忌搜索，显著地提升了算法求解质量。

为降低邻域搜索的计算复杂度，学者提出了基于树图的启发式算法：先将区域抽象为网络图，再将网络图简化为树图，通过树分割获得空间连续的区域。树结点代表空间单元，树干表示空间单元间的邻接关系[21]。Maravalle等[22]提出了一个基于树图的区划问题算法(MIDAS)：根据图G生成树T，删除T的p-1条连接获得p棵子树，代表p个空间连续的区域。因树T上的解空间很有限，MIDAS算法尝试不断调整树T为T*获得更好的分区方案。此后，Assunção等[23]基于最小生成树概念提出了一个区划问题算法SKATER。Guo[16]改进该算法为RAD-CAP算法，提出了6种动态树生成方法：First-Order-SLK、First-Order-CLK、First-Order-ALK、Full-Order-SLK、Full-Order-CLK和Full-Order-ALK。实验发现：Full-Order-CLK和Full-Order-ALK优于其他生成树方法[16]。

总体上，聚类算法思路简单且容易实现，但这类方法有的难以保证分区连续性，有的虽顾及分区连续性但不能保证全局优化质量。启发式算法类型众多，启发改进方法设计思路简单，但优化性能有限；元启发方法性能较高，但这一类算法设计较为复杂，计算效率偏低。基于生成树的方法，计算效率大幅提升，但同时大幅降低了搜索空间，影响到区划质量。Aydin等[6]设计了基准测试案例，测试了AZP、AZP-SA、AZP-Tabu、ARISEL、SKATER和REDCAP算法。计算结果表明：ARISEL算法总体质量最高，但计算速度很慢；SKATER算法求解质量尚好，计算效率非常高。考虑到区划问题应用领域越来越广，在应用中对于区域规划、决策影响较大，有必要设计更有效地的区划算法，既保证区划质量，又能够快速计算。

3 区划问题定义

某一地理区域共有n个空间单元，记为集合U={1, 2, 3, …, n}。每个单元有m个属性，记为集合A={1, 2, 3, …, m}，单元i属性为a_{i1}, a_{i2}, a_{i3}, …, a_{im}。将地理区域划分为p个空间连续的分区，记为集合C={1, 2, 3, …, p}，分区i包含的地理单元集合为c_i，满足$c_i \cap c_j = \phi (i \neq j)$和$c_1 \cup c_2 \cup c_3 \cup \cdots c_p = U$，即任意两个分区无重叠，每个空间单元必须划分在特定分区中。分区目标是分区内单元属性方差之和最小[6,12,16]。

$$f(C) = \sum_{i \in C} \sum_{j \in c_i} \sum_{k \in A} \left(a_{jk} - \bar{a}_{ik} \right)^2 \quad (1)$$

式中：$\bar{a}_{ik}$为分区i中所有单元属性k的平均值。

在区划实践中，有几个实际问题需要考虑。首先，考虑到不同属性的含义和量纲存在差异，通常采用标准化后的单元属性值。常用的数据标准化方法包括：标准差标准化方法、最大最小极值标准化方法、线性比例标准化方法等。其次，考虑属性的重要性可能不同，可为每个属性设置权重。令单元i标准化属性值为b_{i1}, b_{i2}, b_{i3}, …, b_{im}，属性k的权重为w_k，区划目标函数为：

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

式中：$\bar{b}_{ik}$ 为分区 i 中单元属性 k 标准化值的平均值。

为评价分区质量，可统计每个属性的指标：

$$R_k^2=1-\sum_{i\in C}\sum_{j\in c_i}w_k\left(b_{jk}-\bar{b}_{ik}\right)^2/\sum_{j\in U}w_k\left(b_{jk}-\bar{b}_k\right)^2 \quad (3)$$

式中：$\bar{b}_k$ 为属性 k 标准化值的平均值。R_k^2 值处于0和1之间，数值越大，说明分区质量越好。ArcGIS中实现的K-means和SKATER算法，提供每个属性 k 的分区指标 R_k^2。

如采用标准差标准化方法，则式(3)中均值 $\bar{b}_k=0$。同时，可计算总体 R^2 指标评价分区质量：

$$R^2=1-\sum_{i\in C}\sum_{j\in c_i}\sum_{k\in A}w_k\left(b_{jk}-\bar{b}_{ik}\right)^2/\sum_{j\in U}\sum_{k\in A}w_k{b_{jk}}^2 \quad (4)$$

公式推导也可以证明：$R^2=\sum_{k\in A}w_kR_k^2/\sum_{k\in A}w_k$。

关于分区间差异性的表达，可使用分区属性均值 $\bar{b}_{ik}$ 与总体样本均值的差异来表达。知名的空间数据分析软件GeoDa (https://geodacenter.github.io)提供了AZP-SA、AZP-Tabu、ARISEL、SKATER和REDCAP算法，将分区间差异定义为：$\sum_{j\in U}\sum_{k\in A}{b_{jk}}^2-\sum_{i\in C}\sum_{j\in c_i}\sum_{k\in A}\left(b_{jk}-\bar{b}_{ik}\right)^2$。前半部分为总体样本的总方差，是一个恒值，后半部分为目标函数，随目标函数值减小，分区间差异增大。文献[16]将分区间差异表达为 $\sum_{i\in C}\sum_{k\in A}\left(\bar{b}_{ik}-\bar{b}_k\right)^2$，但未将其作为目标函数使用。一般地，随分区内的差异减小，而分区间的差异会增大。

4　算法设计

本文采用基于中心的区划算法，并保持每个分区空间连续。首先，目标函数(2)中分区 i 的均值 $\bar{b}_{ik}$ 计算较为复杂，容易造成算法计算效率偏低。为加快计算，本文采用分区中心点的属性值取代均值 $\bar{b}_{ik}$。在区划方案中，确定每个分区的中心点，有利于快速评估区划方案目标值。其次，本文将区划问题定义为一个增加了空间连续约束的聚类问题，满足空间连续条件使区划问题求解变得较为复杂。区划算法中，使用分区空间连续判断与空间连续修复保证分区的空间连续性。

本文选择迭代局部搜索(ILS)算法作为求解区划问题的算法框架。ILS算法思路简单、易于实现，对于离散优化问题行之有效[24]。该算法从一个初始解开始，迭代地进行扰动和局部搜索。局部搜索容易陷入局部最优，对当前位置的扰动能够使算法脱离局部最优。初始解生成、局部搜索和扰动使ILS算法的基本模块。为提升ILS算法优化性能，本文改造单解ILS算法为群解ILS算法。改进ILS算法流程如算法1所示。

算法1　改进**ILS**算法

参数:群大小 (*psize*)，破坏强度(*strength*)，连续未更新最好解循环数(*mloops*)。

1. P = GenerateInitialSolutions(*psize*);
2. s_{best}=Best(P);
3. *notImpr*=0;
4. While *notImpr* < *mloops*:
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. P =UpdatePopulation (P, s^*);
12. Output s_{best}

其中，步骤1采用经典K-medoids算法产生初始解。该算法随机选择 p 个空间单元作为分区中心，迭代进行单元指派和中心点更新，直到所有中心点不能更新为止。指派是将每个空间单元指派到最近的中心单元，计算简单；受研究区形状和地理要素空间分布的影响，指派形成的分区不能满保证其空间连续，为此算法需要判断分区的空间连续性，并进行连续性修复。

步骤6进行分区方案扰动。常用的扰动方法很多，例如，破坏若干分区、破坏一个连续区域、破坏一定比例的边界单元，然后进行解的修复。若修复后，分区不能保证空间连续，则继续进行空间连续修复。

步骤7采用分区边界单元移动方法进行局部搜索。该方法尝试移动某一个边界单元到相邻的分区，若该移动能够减少区划目标，则更新当前解。这一操作需要考虑分区的空间连续性，保证单元移动后分区连续性约束仍然得到满足。

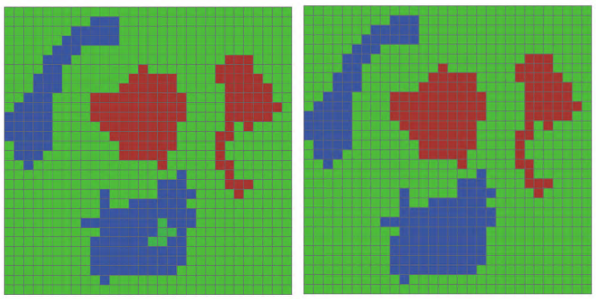
与基于单解的搜索算法相比，改进ILS算法维

护一组解。① 算法步骤1生成一组初始解;② 每一次迭代开始,从群解中随机选择一个解作为当前解进行搜索(步骤5);③ 搜索完成后,使用新解更新群解(步骤11)。群解更新中,优先考虑解的目标值,其次考虑解的差异程度,保持群解之间有一定的差异。基于群解的ILS算法维护一组具有差异度的精英解,扩大了解空间搜索范围,有利于改进求解质量;同时,算法收敛速度通常会变慢,计算时间有一定的增加。

因本文算法基于中心单元评估分区目标,局部搜索完成后,步骤8尝试更新中心单元,使分区目标值进一步降低。

空间连续性判断是区划算法中的一个关键步骤。本文使用生成树方法判断分区的连续性[25-26]。若一个分区中的所有单元能构成一个生成树,则该分区连续。考虑到一些特殊情形,本文算法也允许一个分区包含2个或多个面积较大的区域。图1(a)中,蓝色、红色和绿色分区均包括2个部分。所有蓝色或红色部分的面积均较大,可认为蓝色区域和红色区域是空间连续的分区。而绿色部分中,因其中一块过小,将其视为空间不连续的破碎单元。

空间连续判断方法如下:针对某一分区,从中心单元开始构造生成树;若所有单元能连接到生成树上,则该分区空间连续;若有单元不能连接到树上,该分区不是空间连续分区,剩余单元为破碎单元。当允许一个分区包含两个或多个面积较大的连续区域,空间连续判断方法如下:① 针对某一分区,从任意单元开始构造生成树;② 若有某些单元不能连接到生成树上,则针对剩余单元构造新的生成树;③ 重复步骤②直到没有剩余单元;④ 计算每一个生成树对应单元数和面积,若有某一生成树单元数或面积小于规定的阈值,则认为该区域不连续;同时,单元数或面积过小生成树对应的单元是破碎单元。针对空间不连续分区,需要对破碎的单元进行修复,将其指派到最近的相邻分区中。如图1左图中的绿色部分,有一块很小的斑块,共有5个单元,可将其作为破碎单元;可将其指派到邻近蓝色分区,修复后的分区如图1(b)所示。



(a) 不连续分区 (b) 连续分区

图1 分区空间连续性判断示意

Fig. 1 Illustration of the spatially contiguous regions

5 算法测试

5.1 基准案例测试

算法测试使用文献[6]提供的基准测试案例集。该案例集基于3个规则格网地图数据生成,网格数量分别为120(10×12)、300(15×20)和1200(30×40)。将这些地图事先划分为若干分区,并根据分区模拟每一个单元的属性数值。使用最终的模拟数据进行区划,测试算法的性能。基于每幅地图生成18个案例,即2类分区形状、3个分区数量和3个数值模拟参数的组合。分区形状为简单矩形为主(A)和较复杂图形(B),分区数量为5、10和15,相邻分区属性均值差异参数设置为2、3和4。另外,针对一幅900(30×30)网格地图,模拟了形状不规则分区案例,共5个分区,数值模拟参数为3。综上,共生成55个分区案例,每个案例的单元属性值分别随机模拟100次。模拟方法是:为每个分区设置一个属性均值,使用参数2、3或4设置相邻分区属性均值,以方差为1的正态分布随机模拟单元属性值。模拟案例基本情况见表1,案例详细介绍见文献[6],数据下载地址为 https://doi.org/10.6084/m9.figshare.14067239。该地址还提供了6种算法计算结

表1 基准测试案例特征

Tab. 1 Characteristics of The benchmark instances

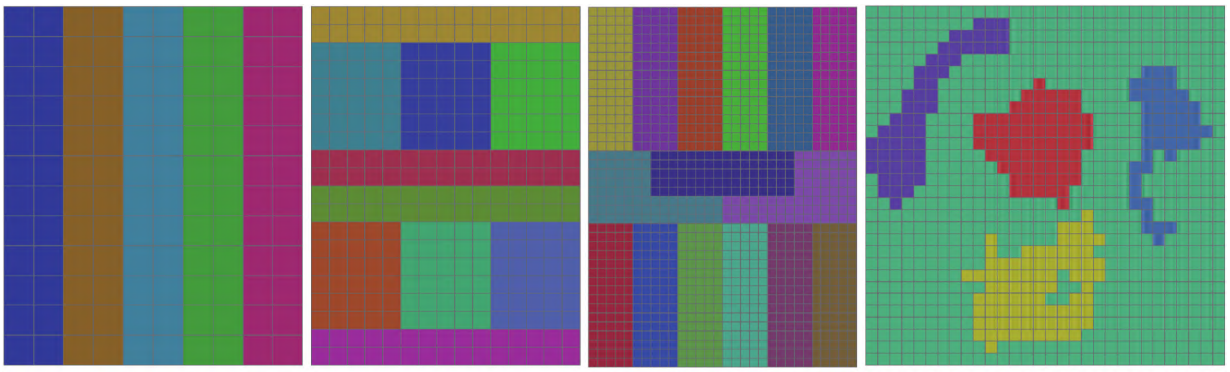
地图名称	网格大小	分区形状	分区数量/个	属性数值模拟参数	产生区划方案数量/件	数值模拟次数/次
G120	10×12	A, B	5, 10, 15	2, 3, 4	18	100
G300	15×20	A, B	5, 10, 15	2, 3, 4	18	100
G1200	30×40	A, B	5, 10, 15	2, 3, 4	18	100
Blob	30×30	不规则	4	3	1	100

果、质量评价指标和计算时间。

为直观地理解案例，图2展示了4种区划方案：G120-5A(图2(a))、G300-10B(图2(b))、G1200-15A(图2(c))和Blob(图2(d))。方案名称由地图名称、分区数量和分区形状组成。图3为图2中4个区划方案分别使用模拟参数4、2、3和3生成的模拟数值，色彩深浅代表数值的大小。模拟参数越大，不同分区单元的数值差异越大，相对容易区分事先设置的区域；反之，则较难辨认事先设置的区域。

案例测试的计算环境为：HP桌面计算机，配置Intel Core I7-6700 CPU 3.40-GHz和8 GB内存，Windows 10操作系统，安装有Python 2.7和ArcGIS 10.4。本文算法使用Python程序设计语言编程实现。为提升计算速度，算法在PyPy (https://www.pypy.org)环境中运行。算法参数*psize*、*strength*和*mloops*分别设置为10、5%和100。

针对每个分区方案的100个数值模拟，使用本文ILS算法进行计算，获得100个区划方案。案例测试中，本文算法采用严格的空间连续约束，使测试结果能够与其他算法进行质量比较。计算每个区划方案的调整兰德系数(ARI)指标[27]和R^2指标。ARI指标表示分区方案与真实分区的相似程度，越接近于1越好；R^2指标度量分区内单元属性方差的相对大小，越接近于1越好。表2列出了每个分区方案100个分区结果ARI指标和R^2指标的平均值，并与SKATER和ARISEL算法进行比较，其中，ARISEL和SKATER算法区划指标来自文献[6]。从表2可以看出，3个算法结果的ARI指数差异显著，ILS算法均值0.9494高于SKATER算法(0.8789)和ARISEL算法(0.8894)。而R^2指数差异并不大，ILS算法均值(0.9696)略高于SKATER算法(0.9618)和ARISEL算法(0.9686)。相对于ARISEL算法，ILS算法R^2指标提升了0.0010，等同于目标值降低了总方差的0.1%。尽管这一指标提升数

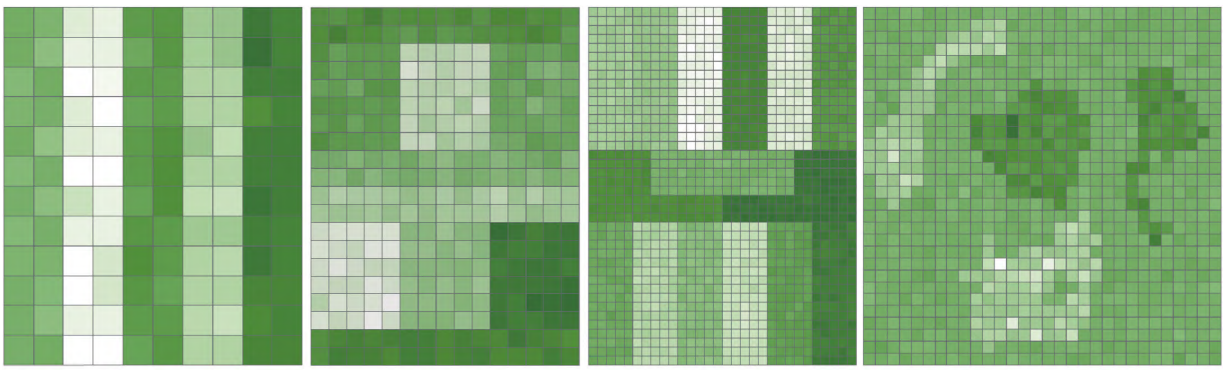


(a) G120_5A　(b) G300_10B　(c) 1200_15A　(d) Blob

注：不同颜色表示不同分区。

图2　分区示意

Fig. 2　Illustration of spatial units and regions (G120_5A, G300_10B, 1200_15A and Blob)



(a) G120_5A4　(b) G300_10B2　(c) 1200_15A3　(d) Blob3

注：不同颜色表示不同属性。

图3　单元属性数值模拟示意

Fig. 3　Simulated values of the spatial units

表2 基准案例ARI指数和R^2指数均值统计

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

案例名称	100个ARI指数均值			100个R^2指数均值		
	ILS	SKATER	ARISEL	ILS	SKATER	ARISEL
G120_5A2	0.8037	0.6650	0.7170	0.9010	0.8696	0.9014
G120_5A3	0.9442	0.7949	0.9160	0.9523	0.9151	0.9523
G120_5A4	0.9788	0.9123	0.9715	0.9711	0.9526	0.9708
G120_5B2	0.8609	0.7483	0.7348	0.9027	0.9022	0.9124
G120_5B3	0.9586	0.8842	0.9267	0.9520	0.9486	0.9529
G120_5B4	0.9866	0.9245	0.9791	0.9707	0.9658	0.9709
G120_10A2	0.8518	0.8007	0.7758	0.9718	0.9742	0.9734
G120_10A3	0.9627	0.8817	0.8849	0.9884	0.9854	0.9862
G120_10A4	0.9871	0.9258	0.9526	0.9932	0.9910	0.9923
G120_10B2	0.8515	0.7646	0.7564	0.9736	0.9743	0.9731
G120_10B3	0.9627	0.8839	0.8900	0.9883	0.9864	0.9868
G120_10B4	0.9912	0.9342	0.9537	0.9932	0.9915	0.9922
G120_15A2	0.8504	0.8063	0.7651	0.9864	0.9890	0.9860
G120_15A3	0.9493	0.8886	0.8685	0.9939	0.9940	0.9928
G120_15A4	0.9862	0.9313	0.9233	0.9966	0.9963	0.9961
G120_15B2	0.8505	0.8064	0.7789	0.9876	0.9879	0.9850
G120_15B3	0.9372	0.8813	0.8667	0.9940	0.9933	0.9925
G120_15B4	0.9865	0.9164	0.9242	0.9971	0.9959	0.9959
G300_5A2	0.8978	0.7394	0.8164	0.8971	0.8613	0.9004
G300_5A3	0.9692	0.8906	0.9561	0.9498	0.9258	0.9504
G300_5A4	0.9906	0.9444	0.9879	0.9707	0.9594	0.9708
G300_5B2	0.9015	0.8169	0.7978	0.8982	0.8917	0.9030
G300_5B3	0.9679	0.9176	0.9520	0.9499	0.9457	0.9504
G300_5B4	0.9907	0.9547	0.9884	0.9707	0.9665	0.9708
G300_10A2	0.8686	0.7593	0.8036	0.9682	0.9505	0.9606
G300_10A3	0.9616	0.8436	0.8913	0.9874	0.9735	0.9819
G300_10A4	0.9911	0.8728	0.9614	0.9929	0.9762	0.9909
G300_10B2	0.8674	0.8315	0.7849	0.9701	0.9691	0.9679
G300_10B3	0.9730	0.9151	0.8973	0.9875	0.9851	0.9848
G300_10B4	0.9942	0.9168	0.9748	0.9929	0.9881	0.9921
G300_15A2	0.8588	0.8351	0.7836	0.9853	0.9854	0.9814
G300_15A3	0.9621	0.8945	0.8686	0.9939	0.9924	0.9911
G300_15A4	0.9907	0.9242	0.9330	0.9967	0.9951	0.9954
G300_15B2	0.8771	0.8465	0.7951	0.9842	0.9868	0.9845
G300_15B3	0.9684	0.8995	0.8739	0.9940	0.9930	0.9921
G300_15B4	0.9931	0.9188	0.9241	0.9968	0.9949	0.9956
G1200_5A2	0.9602	0.7946	0.9015	0.8929	0.8544	0.8913
G1200_5A3	0.9863	0.9065	0.9817	0.9488	0.9218	0.9489
G1200_5A4	0.9949	0.9407	0.9944	0.9703	0.9449	0.9703
G1200_5B2	0.9548	0.8973	0.8878	0.8920	0.8854	0.8939
G1200_5B3	0.9876	0.9702	0.9755	0.9487	0.9469	0.9480

（转下页）

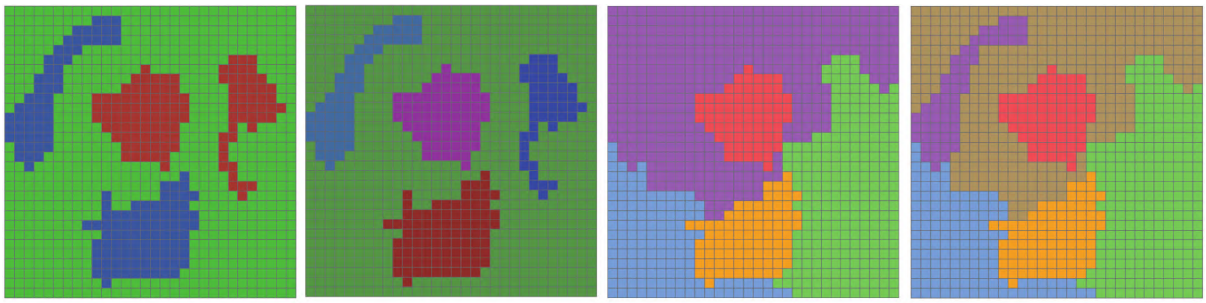
（接上页）

案例名称	100个ARI指数均值			100个R^2指数均值		
	ILS	SKATER	ARISEL	ILS	SKATER	ARISEL
G1200_5B4	0.9963	0.9849	0.9946	0.9702	0.9688	0.9702
G1200_10A2	0.8860	0.7842	0.8261	0.9618	0.9467	0.9564
G1200_10A3	0.9809	0.8533	0.9260	0.9865	0.9684	0.9801
G1200_10A4	0.9952	0.8702	0.9683	0.9926	0.9728	0.9904
G1200_10B2	0.9087	0.9008	0.8428	0.9677	0.9687	0.9681
G1200_10B3	0.9861	0.9499	0.9094	0.9866	0.9850	0.9839
G1200_10B4	0.9969	0.9620	0.9629	0.9926	0.9906	0.9912
G1200_15A2	0.8741	0.8814	0.8089	0.9832	0.9850	0.9811
G1200_15A3	0.9684	0.9153	0.8958	0.9932	0.9919	0.9906
G1200_15A4	0.9928	0.9319	0.9321	0.9964	0.9949	0.9950
G1200_15B2	0.8917	0.9148	0.8099	0.9834	0.9860	0.9824
G1200_15B3	0.9777	0.9322	0.8911	0.9934	0.9928	0.9917
G1200_15B4	0.9929	0.9392	0.9329	0.9964	0.9953	0.9953
Blob	0.9409	0.9357	0.8976	0.8700	0.8454	0.8646
平均值	0.9454	0.8789	0.8894	0.9696	0.9618	0.9686

值不大，但对分区边界产生了很显著的影响，导致ARI指数差异较大。从表2可以看出：总体上ILS算法结果优于ARISEL算法，而ARISEL算法优于SKATER算法。针对模拟参数为2的高难度案例，ILS算法优势更为显著。

针对图3中Blob模拟数值案例，因本文区域空间连续定义中一个分区可以包含两个或多个面积较大的连续区域，算法能够发现地理重复模式[28]。图4中，图4(a)和图4(b)区划方案为ILS算法结果，图4(c)和图4(d)区划方案为ArcGIS 10.4中SKATER算法结果。可以看出，ILS算法还原了事先设定的3个分区，而SKATER算法混淆了部分分区。ArcGIS的计算时间约为2.5~2.7 s，ILS算法需5.1~7.1 s。图4中，4个区划方案的R^2指标值分别为0.8726、0.8913、0.5680和0.7191。

表3为3个算法的计算时间比较，其中，ARISEL和SKATER算法计算时间来自文献[6]，时间单位均为秒。可以看出：① 在ArcGIS中实现的SKATER算法计算速度最快，随案例规模增大，计算时间增加较少；② ARISEL算法在Python环境中运行，计算时间最长，随案例规模增大，计算时间快速增长；③ 本文ILS算法计算时间高于SKATER算法，但远低于ARISEL算法。应当注意，3个算法的计算环境差异很大，表中计算时间仅做参考，不宜直接比较。



(a) ILS (p=3)　(b) ILS (p=5)　(c) SKATER (p=5)　(d) SKATER (p=6)

注：不同颜色表示不同分区。

图4　Blob案例分区结果

Fig. 4　Regionalization results from blob instance

表3 计算时间统计

Tab. 3 Statistics of the computation times

案例	ARISEL	SKATER	ILS	案例	ARISEL	SKATER	ILS
G120_05A	4.81	0.53	0.78	G120_05B	4.25	0.45	0.80
G120_10A	3.21	0.54	0.84	G120_10B	3.28	0.51	0.78
G120_15A	4.28	0.46	0.74	G120_15B	4.26	0.50	0.76
G300_05A	42.76	0.55	1.72	G300_05B	48.30	0.60	1.60
G300_10A	25.22	0.56	1.48	G300_10B	21.28	0.58	1.01
G300_15A	28.30	0.59	1.32	G300_15B	26.24	0.60	1.55
G1200_5A	1296.96	0.92	10.22	G1200_5B	1123.03	0.84	6.24
G1200_10A	740.46	0.94	7.16	G1200_10B	508.20	0.95	7.39
G1200_15A	481.21	0.95	5.49	G1200_15B	338.53	0.95	4.78

5.2 黄淮海地区气候分区

为进一步测试本文算法，选择黄淮海地区尝试进行气候区划。首先，本文黄淮海地区包括黄河、淮河、海河流域及山东半岛；其次，使用该区域15分网格30年年均降雨量和年均温度进行分区。案例区包括2478个网格点，如图5所示。研究区每个空间单元有60个属性数据，包括30个年均降雨量数据和30个年均气温数据。应当注意，本文案例仅用于测试算法，不是实际的气候区划。

针对黄淮海地区气候数据，分别使用本文ILS算法、GeoDa 1.20提供的SKATER、REDCAP和ARISEL算法进行气候区划。作者发现，GeoDa实现的SKATER算法在分区质量与计算时间方面均优于ArcGIS 10.4提供的SLATER算法，GeoDa实现的REDCAP算法(Full-Order Single linkage选项)，计算时间比SKATER略慢，但分区质量明显提升。GeoDa使用C++程序设计，实现的ARISEL算法比原创算法Python ClusterPy 0.9.9 (https://pypi.org/project/clusterPy)计算速度有几十倍的提升。因此，本文ILS算法与GeoDa提供的区划算法进行比较。采用这些算法，气候分区数量分别设置为3、4、5、6、7、8、9、10、12和15，并假定所有属性的权重相同，均为1。

表4提供了4个算法区划结果的R^2指标和计算时间。可以看出，ILS算法区划质量指标大幅领先SKATER，显著优于SKATER和ARISEL，其R^2指标

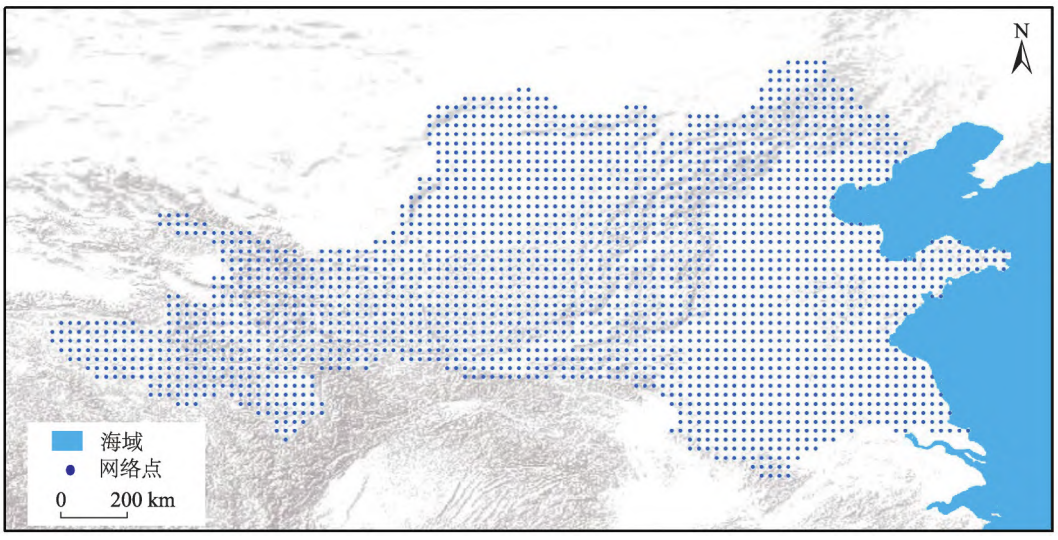


图5 黄淮海流域示意

Fig. 5 The Huang-Huai-Hai river basin

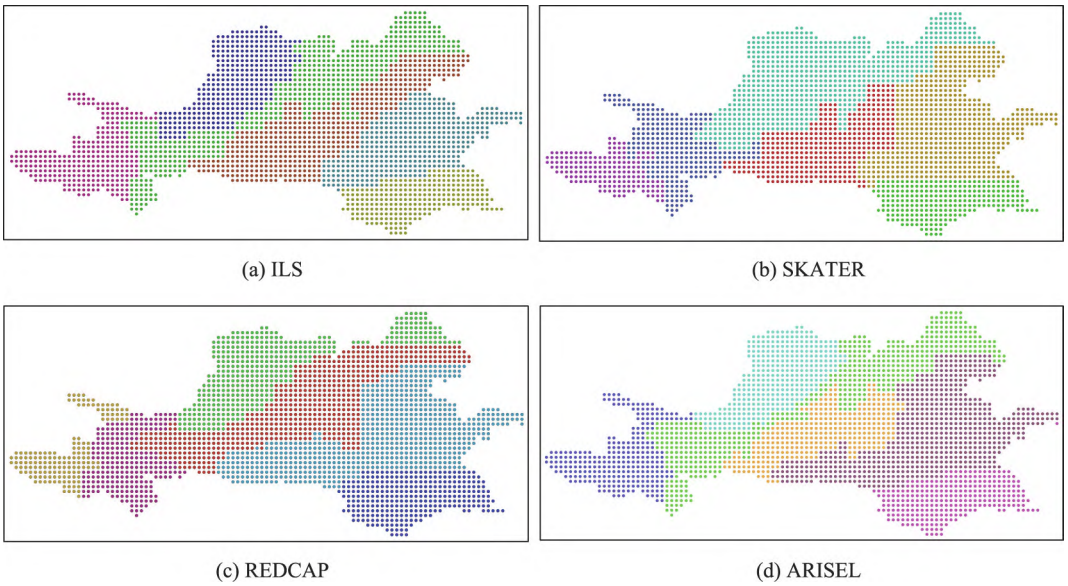
表4　研究区气候区划R^2指标统计

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

p	ILS		SKATER		REDCAP		ARISEL	
	R^2	时间/s	R^2	时间/s	R^2	时间/s	R^2	时间/s
3	0.6874	36.5	0.6713	约1.0	0.6754	约2.0	0.6826	89.6
4	0.7958	37.9	0.7551	约1.0	0.7794	约2.0	0.7966	94.5
5	0.8290	34.9	0.7833	约1.0	0.8126	约2.0	0.8223	62.2
6	0.8547	33.2	0.8084	约1.0	0.8443	约2.0	0.8498	60.9
7	0.8694	37.8	0.8300	约1.0	0.8597	约2.0	0.8668	58.8
8	0.8848	31.6	0.8489	约1.0	0.8721	约2.0	0.8828	51.8
9	0.8942	32.1	0.8643	约1.0	0.8831	约2.0	0.8893	38.9
10	0.9045	47.2	0.8732	约1.0	0.8956	约2.0	0.9034	49.4
12	0.9172	46.6	0.8899	约1.0	0.9102	约2.0	0.9080	41.1
15	0.9301	35.4	0.9124	约1.0	0.9230	约2.0	0.9275	37.8

平均值分别为0.8567、0.8237、0.8455和0.8529。因GeoDa未提供计算时间，表4中SKATER和REDCAP计算时间为估算，ARISEL算法时间为手工计时获得。本文ILS算法使用Python程序设计实现，若使用C/C++语言，计算效率会有大幅度提升，将显著地高于ARISEL算法。

图6为分区数量为6时，ILS、SKATER、REDCAP和ARISEL算法的区划结果。可以看出，4个区划结果差异较大，表现在区域形状、大小和边界的差异。从R^2指标看，ILS算法分区指标(0.8547)优于SKATER (0.8084)、REDCAP(0.8443)和ARISEL (0.8498)。SKATER算法基于相邻单元的相似性，不考虑非邻单元之间的关系，导致区划质量偏低。REDCAP算法改进SKATER算法，比较分区间均值的相似性，算法质量显著提高。ARISEL算法使用禁忌策略迭代地进行局部搜索，求解质量比REDCAP算法更高，但搜索时间偏长。ILS算法使用破坏重建的搜索策略，并更新分区中心，能够获得更高质量的分区。图6还可以看出ILS算法与ARISEL算法所划分气候分区的边界差异非常显著，但其R^2指标差异仅为0.0049。因此，R^2指标的轻微改善，可能对分区结果可能产生较为显著的影响。



(a) ILS　(b) SKATER　(c) REDCAP　(d) ARISEL

注：不同颜色表示不同分区。

图6　黄淮海流域气候区划示意

Fig. 6　The climate regions of the Huang-Huai-Hai River Basin

5 结论与讨论

本文改进ILS算法用于求解区划问题。该算法由初始解生成、局部搜索、群解搜索、解扰动、中心点更新等部分构成，且通过分区空间连续判断和修复操作保证当前解中所有分区空间连续。该算法基于分区中心点评价分区目标值，大幅降低了目标函数的计算，从而提升了算法效率。算法中，群搜索、解扰动和中心点更新扩大了解的搜索空间，从而有利于提升分区质量。基准案例测试表明：改进ILS算法区划结果优于SKATER算法和ARISEL算法。对于无明显气候分区边界的多属性气候分区案例，改进ILS算法分区目标值显著优于SKATER、REDCAP和ARISEL算法。

本文改进ILS算法设计具有几个显著的特点和优势。① 与AZP、AZP-SA、AZP-Tabu和ARISEL相比，ILS算法选择分区中心点进行目标函数计算，避免了局部搜索过程频繁地计算分区中单元属性均值，从而大幅地提高了算法计算效率。② AZP算法属于简单的启发式算法，AZP-SA、AZP-Tabu算法改进了搜索策略，属于元启发算法范畴，有效地提升了算法质量，ARISEL使用多个初始解，并选择高质量解进行禁忌搜索，有利于提升分区质量。改进ILS算法使用群解、扰动等方法，区别有现有算法设计，充分利用了成熟的优化算法机制。③ SKATER仅考虑相邻单元区之间的相似性，大幅降低了搜索空间，计算效率很高。REDCAP算法改进最小生成树构造方法，分区质量得到提升。ILS算法通过搜索和扰动克服了SKATER和REDCAP算法的过于短视的局限，有利于搜索到高质量分区。综上，ILS算法的这些特征，保证了分区质量，又有效降低了算法的复杂度。

考虑到地理现象的空间渐变性、地理系统的复杂性、空间分异规律的尺度依赖性，本文区划算法的使用应建立在区域研究基础上：把握地理现象格局与变化机理，理解特定区域的地理特征，明确区划任务与目标，进而选择合适的区划指标。进一步的研究方向包括：如何确定合适的分区数量，如何进行数据标准化处理，如何选择最适宜差异度函数，以及如何基于本文算法发展出通用的区划方法和软件工具。

参考文献(References):

[1] 郑度,葛全胜,张雪芹,等.中国区划工作的回顾与展望[J].地理研究,2005,24(3):330-344. [Zheng D, Ge Q S, Zhang X Q, et al. Regionalization in China: Retrospect and prospect[J]. Geographical Research, 2005,24(3):330-344.] DOI:10.3321/j.issn: 1000-0585.2005.03.002

[2] 刘燕华,郑度,葛全胜,等.关于开展中国综合区划研究若干问题的认识[J].地理研究,2005,24(3):321-329. [Liu Y H, Zheng D, Ge Q S, et al. Problems on the research of comprehensive regionalization in China[J]. Geographical Research, 2005,24(3):321-329.] DOI:10.3321/j.issn: 1000-0585.2005.03.001

[3] 郑度,欧阳,周成虎.对自然地理区划方法的认识与思考[J].地理学报,2008,63(6):563-573. [Zheng D, Ou Y, Zhou C H. Understanding of and thinking over geographical regionalization methodology[J]. Acta Geographica Sinica, 2008,63(6):563-573.] DOI:10.3321/j.issn:0375-5444.2008.06.001

[4] 高江波,黄姣,李双成,等.中国自然地理区划研究的新进展与发展趋势[J].地理科学进展,2010,29(11):1400-1407. [Gao J B, Huang J, Li S C, et al. The new progresses and development trends in the research of physio-geographical regionalization in China[J]. Progress in Geography, 2010,29(11):1400-1407.]

[5] Duque J C, Ramos R, Suriñach J. Supervised regionalization methods: A survey[J]. International Regional Science Review, 2007,30(3):195-220. DOI:10.1177/0160017607301605

[6] 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. DOI:10.1080/13658816.2021.1905819

[7] Grigg D. The logic of regional systems[J]. Annals of the Association of American Geographers, 1965,55(3):465-491. DOI:10.1111/j.1467-8306.1965.tb00529.x

[8] Lankford P M. Regionalization: Theory and alternative algorithms[J]. Geographical Analysis, 1969,1(2):196-212. DOI:10.1111/j.1538-4632.1969.tb00615.x

[9] Pocock D C D, Wishart D. Methods of deriving multi-factor uniform regions[J]. Transactions of the Institute of British Geographers, 1969(47):73. DOI:10.2307/621736

[10] Larson R C. The process of regionalization: An appropriate conceptual and methodological approach[J]. Socio-Economic Planning Sciences, 1981,15(5):199-205. DOI:10.1016/0038-0121(81)90040-9

[11] Keane M. The size of the region-building problem[J]. Environment and Planning A: Economy and Space, 1975,7(5):575-577. DOI:10.1068/a070575

[12] Duque J C,Church R L,Middleton R S.The p- regions problem.p[J].Geographical Analysis,2011,43(1):104- 126. DOI:10.1111/j.1538-4632.2010.00810.x

[13] Li W W, Church R L, Goodchild M F. The p-compact-regions problem[J]. Geographical Analysis, 2014,46(3):250-273. DOI:10.1111/gean.12038

[14] Ward J H. Hierarchical grouping to optimize an objective function[J]. Journal of the American Statistical Association, 1963,58(301):236-244. DOI:10.1080/01621459.1963.10500845

[15] Jain A K, Dubes R C. Algorithms for clustering data[M]. Englewood Cliffs, NJ: Prentice Hall, 1988

[16] Guo D. Regionalization with dynamically constrained agglomerative clustering and partitioning (REDCAP)[J]. International Journal of Geographical Information Science, 2008,22(7):801-823. DOI:10.1080/13658810701674970

[17] 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, 1977,2(4):459. DOI:10.2307/622300

[18] Browdy M H. Simulated annealing: An improved computer model for political redistricting[J]. Yale Law & Policy Review, 1990,8(1):163-179.

[19] Openshaw S, Rao L. Algorithms for reengineering 1991 census geography[J]. Environment and Planning A: Economy and Space, 1995,27(3):425-446. DOI:10.1068/a270425

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

[21] 郭仁忠.二维有序聚类方法及其在编制区划地图中的应用[J].武汉测绘学院学报,1985,10(2):21-29. [Guo R Z. Clustering method for 2—Dimensional ordered samples and its application to the compilation of maps of regional division[J]. Journal of Wuhan Institute of Surveying and Mapping, 1985,10(2):21-29.]

[22] Maravalle M, Simeone B. A spanning tree heuristic for regional clustering[J]. Communications in statistics - theory and methods, 1995,24(3):625- 639. DOI:10.1080/03610929508831512.

[23] 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. DOI:10.1080/13658810600665111

[24] Lourenço H R, Martin O C, Stützle T. Iterated local search: Framework and applications[M]//International Series in Operations Research & Management Science. Boston, MA: Springer US, 2010:363-397. DOI:10.1007/978-1-4419-1665-5_12

[25] Xiao N C. A unified conceptual framework for geographical optimization using evolutionary algorithms[J]. Annals of the Association of American Geographers, 2008,98(4):795-817. DOI:10.1080/00045600802232458

[26] Liu Y Y, Cho W K T, Wang S W. PEAR: a massively parallel evolutionary computation approach for political redistricting optimization and analysis[J]. Swarm and Evolutionary Computation, 2016,30:78- 92. DOI:10.1016/j.swevo.2016.04.004

[27] Rand W M. Objective criteria for the evaluation of clustering methods[J]. Journal of the American Statistical Association, 1971,66(336):846- 850. DOI:10.1080/01621459.1971.10482356

[28] Kang Y H, Wu K L, Gao S, et al. STICC: A multivariate spatial clustering method for repeated geographic pattern discovery with consideration of spatial contiguity[J]. International Journal of Geographical Information Science, 2022:1-32. DOI:10.1080/13658816.2022.2053980