SMP: Scalable Max-P Regionalization

Hussah Alrashid Yongyi Liu Amr Magdy
Department of Computer Science and Engineering
University of California, Riverside
Riverside, USA
{halra004, yliu786}@ucr.edu, amr@cs.ucr.edu

ABSTRACT

MP-regions is an NP-hard problem that groups spatial areas to produce a maximum number of regions by enforcing a user-defined constraint at the regional level. Existing approximate algorithms for MP-regions do not scale for large datasets due to their high computational cost. This paper introduces *SMP*; a scalable technique to support MP-regions on large datasets. *SMP* works on two stages. The first stage finds an initial solution through randomized search, and the second stage improves this solution through efficient heuristic search. *SMP* optimizes the region building efficiency and quality by tuning the randomized area selection to trade-off runtime with region homogeneity. The experimental evaluation shows the superiority of our technique to support an order of magnitude larger datasets efficiently compared to the state-of-the-art techniques while producing high-quality solutions.

CCS CONCEPTS

• **Information systems → Geographic information systems**.

KEYWORDS

Max p-regions, Regionalization, Spatial clustering

ACM Reference Format:
Hussah Alrashid Yongyi Liu Amr Magdy. 2022. SMP: Scalable Max-P Regionalization. In *The 30th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '22), November 1–4, 2022, Seattle, WA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3557915.3561011

1 INTRODUCTION

Regionalization is the process of clustering a set of spatial polygons into spatially contiguous regions that meet user-defined constraints [10]. An example of regionalization is clustering cities in California into regions so that each region has at least 30K COVID-19 infections. The traditional problem formulations [8, 10, 21] put a major hurdle on users to input the number of regions p. Such a hurdle introduced the challenging "spatial scale problem", as users fail to determine the appropriate spatial scale according to the underlying data. A new formulation, called *max-p-regions* (or MP-regions for short), has recently addressed the spatial scale problem [7] and became popular in different applications. Instead of forcing users to input p, the MP-regions automatically discovers the appropriate value of p based on a user-defined constraint on a certain attribute. This gives the user more tractable and flexible expressiveness. However, it suffers from severe scalability limitations and cannot support even moderate-sized datasets.

MP-regions is an NP-hard problem, therefore, finding an exact solution is prohibitively expensive and impractical in real applications. Besides, existing approximate algorithms support relatively small datasets. For ~3,000 US counties, the proposed techniques in [7, 30] consume from 2.5 minutes up to several hours, depending on the underlying constraints, for each single regionalization query. In fact, real applications already use datasets that are an order of magnitude larger than this dataset. Therefore, such inefficient time makes existing techniques incapable of supporting many real-world applications. For example, such limitation is crystal clear when MP-regions is applied to reduce the uncertainty of American Community Survey data [29], which help determine how more than $675 billion in USA federal and state funds are distributed each year [4] and is heavily used by community organizations, governmental agencies, social scientists in various disciplines [5]. It is explicitly stated in [29] that "*Using the algorithm (meaning MP-regions) requires a tradeoff that is not appropriate in all situations or for all audiences - one must be willing to reduce the number of geographic units of analysis.*" Similar examples repeat in studying neighborhoods [23, 30] and studying regional poverty [9]. This scalability limitation urges the need to build scalable regionalization algorithms to support large-sized data.

This paper proposes *SMP*; a system-level module that scales up MP-regions on large spatial datasets. *SMP* employs a two-phase approximate search. In the first phase, it employs novel heuristics to find an initial solution efficiently. Existing regionalization techniques grow as many randomized regions as possible in the first phase. This produces a large number of enclave areas as randomization cause enclave areas to remain unassigned. On the other hand, *SMP* is designed to reduce the number of enclaves and maximize the number of regions. It builds the initial regions with minimal inter-regional gaps by selecting new borders based on a criterion of spatial compactness. In the second phase, *SMP* optimizes existing heuristics to improve the initial solution. This approximate search avoids exhaustive exploration of the search space, so it efficiently supports large datasets that are not currently supported.

Our experiments on real datasets show significant improvements in regionalization scalability and quality by saving up to 97% of query time compared to existing competitors. The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 formulates the MP-regions problem. Section 4 introduces

This work is partially supported by the National Science Foundation, USA, under grants IIS-1849971, SES-1831615, and CNS-2031418.

SIGSPATIAL '22, November 1–4, 2022, Seattle, WA, USA

ACM ISBN 978-1-4503-9529-8/22/11.
https://doi.org/10.1145/3557915.3561011

our proposed technique. Finally, Section 5 provides experimental evaluation and Section 6 concludes the paper.

2 RELATED WORK

The regionalization literature studies various problems under spatial clustering [13, 20] and spatial graph partitioning [6, 32]. The latter is related to regionalization as spatial polygons can be modeled as a node-attributed spatial neighborhood graph, and regionalization can be expressed as a graph partitioning problem. However, the different objectives and constraints on the output sub-graphs make existing spatial graph partitioning techniques inapplicable for regionalization problems. The most related problems to our work are the *p-regions* [2, 8, 14, 16–19, 31] and the *max-p-regions* (MP-regions) [1, 7, 9, 11, 22, 25–30]. The *p-regions* problem is related in the sense that the MP-regions problem is a successor that overcomes its limitations. There is an existing literature on the *p-regions* problem that ranges from building compact regions [2, 16–18], network-constrained regionalization [31], and functional regions delimitation [14, 15]. Despite this literature, the fundamental change in MP-regions compared to *p-regions* makes it inapplicable to adopt existing *p-regions* techniques.

The literature on MP-regions [7] is more related to our work. MP-regions has variations with a single constraint [7, 28, 30], multiple constraints [11], and network-aware regionalization [27], in addition to a variety of applications that are built on top of them [1, 9, 22, 25, 26, 29]. Existing techniques build variations of the same framework, which is also used in our proposed technique in spirit. This framework first builds a set of regions as an initial solution, then uses a heuristic search to improve the solution quality and find a final approximate solution. The differences among these different techniques are in the way of performing the two steps, including different heuristics to build the initial regions, different types of heuristic search, processing optimizations of different steps, or the underlying spatial space, e.g., Euclidean space versus network-constrained space. Distinguished from all existing work, our work is the first to build an efficient technique that addresses the scalability issue of MP-regions. Our technique significantly beats all existing techniques in different performance measures and is able to support an order of magnitude larger datasets.

3 PROBLEM DEFINITION

This section provides a formal definition for the MP-regions problem. The problem input is a set of spatial areas $A = \{a_0, a_1, a_2, .., a_n\}$. Each area a_i is defined with four attributes: a_i=(i, b_i, s_i, d_i), where i is the area identifier, b_i is an arbitrary spatial polygon that defines the area's spatial boundaries, s_i is a spatially extensive attribute, and d_i is a dissimilarity attribute. A spatially extensive attribute s_i of an area is an attribute whose value is divided over the smaller sub-areas when the area is fragmented. For example, the *population* value of a county is divided over its cities, so the *population* is a spatially extensive attribute. This is the opposite of spatially intensive attributes, such as *temperature*, that are not divided when the spatial area is fragmented. The dissimilarity attribute d_i is used in computing output regions' heterogeneity. The MP-regions problem is defined as follows:

Input: (1) A set of n areas: $A = \{a_0, a_1, a_2, ..., a_n\}$. All the areas in A are spatially contiguous and form a single spatially connected component. (2) A threshold T. (3) An objective function H.

Output: A set of regions $R = \{r_1, r_2, ..., r_p\}$ of size p, where each region r_i is a non-empty set of spatially continuous areas satisfying the below constraints and objectives.

Constraints:

- $1 \leq p \leq n$
- $|r_i| \geq 1, \quad \forall r_i \in R$
- $r_i \cap r_j = \emptyset, \quad \forall r_i, r_j \in R \wedge \ i \neq j$
- $\bigcup_{i=1}^{p} r_i = A, \quad \forall r_i \in R$
- $$(\sum_{\forall a_{ij} \in r_i} s_{ij}) \geq T, \qquad \forall r_i \in R \tag{1}$$

Objectives:

- Maximizing p.
- Minimizing the total heterogeneity $H(R)$ in dissimilarity attributes belonging to the same region.

  In this paper, $H(R)$ is defined as follows:

$$H(R) = \sum_{\forall r_i \in R} \sum_{\forall a_{ij}, a_{ik} \in r_i} |d_{ij} - d_{ik}| \tag{2}$$

The first two constraints impose that the output has at least one region ($p \geq 1$), and each region has at least one area ($|r_i| \geq 1$). The third and fourth constraints ensure that each area is assigned to exactly one region, so the regions are both disjoint and covering all input areas. The last constraint ensures that each region has a total value of the spatially extensive attribute s equals to at least the input threshold T, e.g., total population of each region $\geq T$.

The objectives are two-fold. The first objective is maximizing the number of output regions p. This is the main objective of MP-regions problems, and it is prioritized over the second objective. This objective allows to eliminate the number of regions as a user-input, which addresses a major limitation in the previous regionalization problems. The second objective favors output regions to be as homogeneous as possible, measured as a function of a dissimilarity attribute. This attribute is not necessarily to be a spatial attribute. For example, a social scientist might need to produce regions that are homogeneous in average income level.

4 SCALABLE MAX-P REGIONALIZATION (*SMP*)

This section presents *SMP*; which builds on a prevalent two-stage framework to address NP-hard problems, finding an initial solution and then improving it. *SMP* consists of three phases: *region growing phase*, *region gluing phase*, and *optimization phase*.

The region growing phase efficiently builds a set of initial regions with a maximum size over multiple iterations. The region gluing phase finalizes building the initial regions by assigning any remaining areas that do not belong to any region. Finally, the optimization phase improves the quality of the solution.

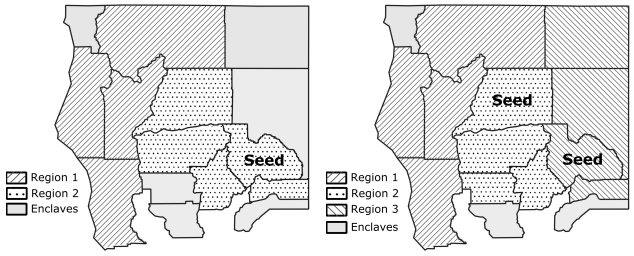
The main observation of our technique is that all existing techniques generate a large number of enclave areas while growing initial regions. These enclaves are scattered all over the space, so

they leave gaps between regions but cannot form regions themselves. *SMP* employs a *layered growing technique* that increases the number of initial regions through optimizing the area selection criteria for this objective while maintaining an efficient runtime. The *layered growing technique* changes the criteria of choosing the seed areas and the criteria of growing the regions from total randomness to constrained randomness. Existing techniques select a random seed area every time they grow a new region. Then, the region grows based on the spatial neighbors of this seed area. The criteria of selecting a total random seed for every region does not give any guarantee about the relative spatial distribution of regions. So, on average, it scatters regions all around the space and leaves a large number of enclaves in between. In state-of-the-art region growing, Figure 1a, six enclave areas are scattered in the space, leaving gaps between the two grown regions. The layered-growth instead groups these enclaves in one part of the space, enabling to grow three regions instead of two and reduces the number of enclaves as depicted in Figure 1b.

Region Growing Phase. At the beginning of the phase, the seed for the first region r_1 is chosen at random. Then all the neighbor areas of r_1 are stored in a queue q_{r_1}. While growing r_1, we add the first unassigned area from q_{r_1} to the region and update q_{r_1} accordingly. This ensures that the unassigned neighbor areas of r_1 are processed and added to the region layer by layer, which leads to a compact region shape that reduces the probability of forming gaps between r_1 and other regions to be grown later. If there exists no unassigned neighbor areas but the total extensive attribute of r_1 is still below the threshold, then all the areas are marked as enclaves. Once the region reaches the threshold, it is added to the list of regions and a new region starts growing. After the first region, for all subsequent regions, the seed area is chosen from the direct neighbor areas of existing regions. For region r_{i+1}, the seed area is the first unassigned area from the last grown region r_i's queue q_{r_i}. If all the areas in q_{r_i} are assigned, then the algorithm turns to $q_{r_{i-1}}$, $q_{r_{i-2}}$, up to q_{r_1}. This phase is repeated *MI* times to produce different initial solutions, and the solution with the largest number of regions p is kept for further processing.

Region Gluing Phase. The initial solution produced by the region growing phase might include enclave areas that are not assigned to any region. The region gluing phase assigns enclave areas to existing regions. To do this, we iterate over the list of enclave areas. For each enclave area, we list their neighbor areas from the spatial neighborhood graph, and in turn we list their neighbor regions. If it has no neighbor regions, we skip it in this step, and it is picked up again when at least one of its neighbors is assigned to a region. If it has one neighbor region, it is assigned to that region. If it has multiple neighbor regions, it is assigned to the region that gives the lowest heterogeneity value increase. We repeat this process until assigning all the enclaves.

Optimization Phase. The optimization phase optimizes the heterogeneity value *H(R)* of the initial solution. We use a heuristic search technique based on a modified simulated annealing (MSA) algorithm [30]. The technique moves areas from a region to another neighbor region so that the *H(R)* value is improved. The original simulated annealing algorithm [7] generates a set of movable areas by identifying all the areas that could be moved from one region (donor region) to the neighboring regions (recipient regions). The



(a) State-of-the-art regions **(b) Layered-growth regions**

Figure 1: Layered-growth impact on the number of regions

algorithm then selects an area to move randomly without violating the input constraints. The move is accepted if it improves the *H(R)* value of the current solution. Otherwise, the move is accepted with a probability calculated using the Boltzmann's equation: $e^{-\Delta H/TM}$ where $-\Delta H$ represents the heterogeneity variation on both donor region and recipient region, and *TM* represents the temperature. The temperature *TM* is decreased at each iteration at a fixed cooling rate *PH* until *TM* reaches a predefined value. The MSA algorithm [30] reduces the overhead of recomputing valid moves and introduces a tabu list, similar to tabu search [12], to prevent search cycles.

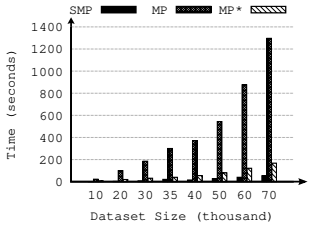
An area is considered movable if it satisfies two conditions: (1) the donor region's threshold remains above T after removing the area, (2) The area is not an articulation area (i.e. moving the area does not break the spatial contiguity of the donor region). Enforcing the second condition involves identifying the articulation areas which is a computational bottleneck. To solve this, we employ Tarjan's algorithm [24] to find all the articulation areas in a single graph traversal.

5 EXPERIMENTAL EVALUATION

This section provides an experimental evaluation for *SMP*.

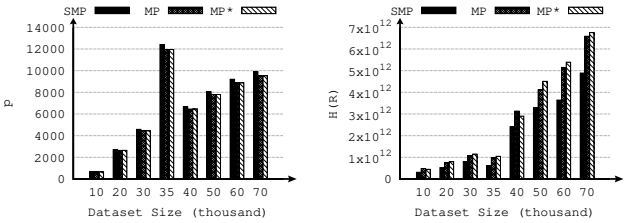
Experimental Setup. We evaluate our proposed technique ***SMP*** against two alternatives: (1) state-of-the-art technique for MP-regions problem [30], denoted as ***MP***, (2) An optimized version of *MP*, denoted as ***MP****, that uses a set instead of a list to maintain the regions' unassigned neighbors while growing the regions. We use three performance measures: *runtime* as a measure for scalability, *number of regions p* and *heterogeneity* as measures for solution quality. We use the dataset size (*DS*) to evaluate the performance with the following values: $\approx$ 10, 20, 30, 35, 40, 50, 60, 70 ($\times 10^3$) areas. The maximum iterations for the region growing phase *MI* is set to 40, the threshold T is set to 250 $\times 10^6$, and the number of iterations in the optimization phase *NI* is set to 50. The length of the tabu list (*LI*), the temperature of computing the Boltzmann's probability (*TM*), and its cooling rate (*PH*), are set to 100, 1, and 0.9, respectively, extending the optimal values from [7]. All the experiments are based on Java 14 implementation and run on Ubuntu 16.04 with a quad-core 3.5GHz processor and 128GB of memory.

Evaluation datasets. We evaluate all techniques on the TIGER Line shape files for (i) the census tracts of the US states, and (II) the county subdivisions [3]. In our experiment, we define the spatially extensive attribute over the *AWATER* attribute that represents the water area. The dissimilarity attribute is defined over the *ALAND* attribute that represents the land area. The county subdivisions dataset includes 35×10^3 areas. For the census tracts dataset, the



(a) Dataset Size (*DS*)

Figure 2: Runtime scalability varying *DS* values



(a) Impact on p **(b) Impact on the Heterogeneity**

Figure 3: Solution quality varying *DS* values

shape files of different neighboring states are merged together to form seven datasets of increasing sizes ranging from $10{\times}10^3$ to $70{\times}10^3$ areas.

Performance Evaluation. Figure 2a shows that *SMP* have linear runtime scalability with increasing dataset size ranging 0.9 to 54 seconds. While on the other hand, *MP* encounters an exponentially increasing runtime ranging from 22 to 1297 seconds. *MP** is faster than *MP* with a runtime ranging from 2 to 53 seconds but is still slower than *SMP*. Our technique is 10 to 29 times faster than *MP* for the largest dataset. The speedup ratio increases with dataset sizes. The changes introduced in *SMP*, either in the region growing phase or the optimization phase, are the dominating factors in speeding up the processing. Figure 3 shows that increasing *DS* increases both p (Figure 3a) and the heterogeneity (Figure 3b) for all alternatives. Since larger datasets have more areas, using the same default threshold value leads to growing more regions that adds to the heterogeneity score. *SMP* generates 26-370 regions more than *MP* and *MP**. *SMP* has a significantly lower heterogeneity than the other competitors. The better p value of *SMP* comes from growing the regions with small inter-regional gaps, which leads to smaller number of scattered enclaves and increases the number of regions. Dividing the same dataset into more regions leads to reducing the heterogeneity due to having a fewer number of area pairs that contribute to computing the heterogeneity.

6 CONCLUSIONS

This paper addresses the scalability issue of MP-regions. MP-regions is a regionalization problem that clusters spatial areas into homogeneous regions. MP-regions is an NP-hard problem. Many of the existing work experience performance degradation when solving MP-regions on large datasets. We propose *SMP* module to provide efficient and scalable approximate solutions for MP-regions. It employs multiple phases to find an initial solution then optimize it using heuristic search. Our experimental evaluation has shown the superiority of our technique against the state-of-the-art techniques in both scalability and solution quality.

REFERENCES

[1] Daniel Arribas-Bel and Charles R Schmidt. 2013. Self-Organizing Maps and the US Urban Spatial Structure. *EPB* 40 (2013), 362–371.
[2] Subhodip Biswas, Fanglan Chen, Zhiqian Chen, Andreea Sistrunk, Nathan Self, Chang-Tien Lu, and Naren Ramakrishnan. 2019. REGAL: A Regionalization Framework for School Boundaries. In *SIGSPATIAL*. 544–547.
[3] U.S. Census Bureau. 2019. TIGER/Line Shapefile, 2016, Series Information for the Current Census Tract State-based Shapefile. https://catalog.data.gov/dataset/tiger-line-shapefile-2016-series-information-for-the-current-census-tract-state-based-shapefile.
[4] U.S. Census Bureau. 2021. About the American Community Survey. https://www.census.gov/programs-surveys/acs/about.html.
[5] U.S. Census Bureau. 2021. ACS Data Stories–Stats in Action! https://www.census.gov/programs-surveys/acs/about/acs-data-stories.html.
[6] David Combe, Christine Largeron, Elöd Egyed-Zsigmond, and Mathias Géry. 2012. Combining Relations and Text in Scientific Network Clustering. In *ASONAM*. 1248–1253.
[7] Juan C Duque, Luc Anselin, and Sergio J Rey. 2012. The Max-P-Regions Problem. *JRS* 52 (2012), 397–419.
[8] Juan C Duque, Richard L Church, and Richard S Middleton. 2011. The P-Regions Problem. *Geographical Analysis* 43 (2011), 104–126.
[9] Juan C Duque, Jorge E Patino, Luis A Ruiz, and Josep E Pardo-Pascual. 2015. Measuring Intra-urban Poverty Using Land Cover and Texture Metrics Derived from Remote Sensing Data. *LUP* 135 (2015), 11–21.
[10] Juan Carlos Duque, Raúl Ramos, and Jordi Suriñach. 2007. Supervised Regionalization Methods: A Survey. *IRSR* 30 (2007), 195–220.
[11] David C Folch and Seth E Spielman. 2014. Identifying Regions Based On Flexible User-defined Constraints. *IJGIS* 28 (2014), 164–184.
[12] Fred Glover. 1989. Tabu Search—Part I. *ORSA* 1 (1989), 190–206.
[13] Diansheng Guo. 2008. Regionalization with Dynamically Constrained Agglomerative Clustering and Partitioning (REDCAP). *IJGIS* 22 (2008), 801–823.
[14] Hyun Kim, Yongwan Chun, and Kamyoung Kim. 2015. Delimitation of Functional Regions Using a P-Regions Problem Approach. *IRSR* 38 (2015), 235–263.
[15] Kamyoung Kim, Yongwan Chun, and Hyun Kim. 2017. p-Functional Clusters Location Problem for Detecting Spatial Clusters with Covering Approach. *Geographical Analysis* 49 (2017), 101–121.
[16] Jason Laura, Wenwen Li, Sergio J Rey, and Luc Anselin. 2015. Parallelization of a Regionalization Heuristic in Distributed Computing Platforms–a Case Study of Parallel-P-Compact-Regions Problem. *IJGIS* 29 (2015), 536–555.
[17] Wenwen Li, Richard L Church, and Michael F Goodchild. 2014. An Extendable Heuristic Framework to Solve the P-Compact-Regions Problem for Urban Economic Modeling. *CEUS* 43 (2014), 1–13.
[18] Wenwen Li, Richard L Church, and Michael F Goodchild. 2014. The p-Compact-Regions Problem. *Geographical Analysis* 46 (2014), 250–273.
[19] Yongyi Liu, Ahmed R. Mahmood, Amr Magdy, and Sergio Rey. 2022. PRUC: P-Regions with User-Defined Constraint. In *VLDB*. 491–503.
[20] Maurizio Maravalle and Bruno Simeone. 1995. A Spanning Tree Heuristic for Regional Clustering. *CSTM* 24 (1995), 625–639.
[21] Stan Openshaw. 1977. Optimal Zoning Systems for Spatial Interaction Models. *EPA* 9 (1977), 169–184.
[22] Jorge E Patino, Juan C Duque, Josep E Pardo-Pascual, and Luis A Ruiz. 2014. Using Remote Sensing to Assess the Relationship Between Crime and the Urban Layout. *Applied Geography* 55 (2014), 48–60.
[23] Ate Poorthuis. 2018. How to Draw a Neighborhood? The Potential of Big Data, Regionalization, and Community Detection for Understanding the Heterogeneous Nature of Urban Neighborhoods. *Geographical Analysis* 50, 2 (2018), 182–203.
[24] Tarjan R. 1971. Depth-first Search and Linear Graph Algorithms. *SICOM* 1 (1971), 114–121.
[25] Sergio J Rey, Luc Anselin, David C Folch, Daniel Arribas-Bel, Myrna L Sastré Gutiérrez, and Lindsey Interlante. 2011. Measuring Spatial Dynamics in Metropolitan Areas. *EDQ* 25 (2011), 54–64.
[26] Sergio J Rey and Myrna L Sastré-Gutiérrez. 2010. Interregional Inequality Dynamics in Mexico. *SEA* 5 (2010), 277–298.
[27] Bing She, Juan C Duque, and Xinyue Ye. 2017. The Network-Max-P-Regions Model. *IJGIS* 31 (2017), 962–981.
[28] V. Sindhu. 2018. *Exploring Parallel Efficiency and Synergy for Max-P Region Problem Using Python.* Master's thesis. Georgia State University.
[29] Seth E Spielman and David C Folch. 2015. Reducing Uncertainty in the American Community Survey through Data-driven Regionalization. *PloS ONE* 10 (2015), e0115626.
[30] Ran Wei, Sergio Rey, and Elijah Knaap. 2020. Efficient Regionalization for Spatially Explicit Neighborhood Delineation. *IJGIS* 35 (2020), 1–17.
[31] Xinyue Ye, Bing She, and Samuel Benya. 2018. Exploring Regionalization in the Network Urban Space. *JGSA* 2 (2018), 4.
[32] Yang Zhou, Hong Cheng, and Jeffrey Xu Yu. 2009. Graph Clustering Based on Structural/Attribute Similarities. *VLDB* 2 (2009), 718–729.