The Journal of Open Source Software

spopt: a python package for solving spatial optimization problems in PySAL

Xin Feng[1, 2], **James D. Gaboardi**[3], **Elijah Knaap**[1], **Ran Wei**[1], **and Sergio Rey**[1]

1 Center for Geospatial Sciences, University of California Riverside **2** Department of Geography and Environmental Sustainability, University of Oklahoma **3** Department of Geography, Pennsylvania State University

DOI: 10.21105/joss.03330

Software

- Review
- Repository
- Archive

Editor: Vissarion Fisikopoulos

Reviewers:

- @ArikaLZ
- @samuel-rosa

Submitted: 21 May 2021
Published: 17 June 2021

Summary

Spatial optimization is a major spatial analytical tool in management and planning, the significance of which cannot be overstated. Spatial optimization models play an important role in designing and managing effective and efficient service systems such as transportation, education, public health, environmental protection, and commercial investment among others. To this end, spopt (spatial optimization) is under active development for the inclusion of newly proposed models and methods for regionalization, facility location, and transportation-oriented solutions. Spopt is a submodule in the open-source spatial analysis library PySAL (Python Spatial Analysis Library) founded by Dr. Serge Rey and Dr. Luc Ancelin in 2005 (Sergio J. Rey et al., 2015; Sergio J. Rey et al., 2021; Sergio J. Rey & Anselin, 2010). The goal of developing spopt is to provide management and decision-making support to all relevant practitioners and to further promote the appropriate and meaningful application of spatial optimization models in practice.

Statement of need

Spatial optimization methods/algorithms can be accessed in many ways. ArcGIS (https://www.esri.com/en-us/home) and TransCAD (https://www.caliper.com/) are two well-known commercial GIS software packages that provide modules designed for structuring and solving spatial optimization problems. The optimization functions they offer focus on a set classical single facility location methods (e.g., Weber, Median, Centroid, 1-center), routing and shortest path methods (e.g., shortest path on the network, least cost path over the terrain), and multi-facility location-allocation methods (e.g., coverage models, p-median problem). They are user-friendly and visually appealing, but the cost is relatively high (Murray, 2019).

Open source software is another option to access spatial optimization. Although it may require users to have a certain level of programming experience, open source software provides relatively novel and comprehensive methods, and more importantly, it is free. This is particularly true for regionalization methods. They are very limited in commercial GIS software, and may only have grouping analysis for vector data and region identification for raster data. On the contrary, there are many application-oriented open source packages that facilitate the implementation of regionalization methods in various fields, including climate (e.g., HiClimR (https://cran.r-project.org/web/packages/HiClimR/index.html), synoptReg (https://cran.r-project.org/web/packages/synoptReg/index.html)), biography (e.g., Phyloregion (https://cran.r-project.org/web/packages/phyloregion/index.html), regioneR (http://bioconductor.org/packages/release/bioc/html/regioneR.html)), hydrology (e.g., nsRFA(https://cran.r-project.org/web/packages/nsRFA/index.html)), agricultural (e.g., OpenLCA (https://www.openlca.org/)), and so on. The functions of graph regionalization with clustering and partitioning have been provided by several packages such as Rgeoda, maxcut: Max-Cut Problem, RBGL: R Boost Graph Library, and grPartition. They are probably the most closely related projects to the regionalization section of spopt, however, they are written in R and MATLAB. Therefore, it is necessary to develop an open source optimization package written in Python that focuses on regionalization.

Current functionality

Originating from the region module in PySAL, spopt is under active development for the inclusion of newly proposed models and methods for regionalization. Six models are developed for aggregating a large set of geographic units (with small footprints) into a smaller number of regions (with large footprints). They are: 1. Max-p-regions: the clustering of a set of geographic areas into the maximum number of homogeneous and spatially contiguous regions such that the value of a spatially extensive regional attribute is above a predefined threshold (Duque et al., 2012; Wei et al., 2020). 2. Spatially-encouraged spectral clustering (spenc): an algorithm to balance spatial and feature coherence using kernel combination in spectral clustering (Wolf, n.d.). 3. Region-K-means: K-means clustering for regions with the constraint that each cluster forms a spatially connected component. 4. Automatic Zoning Procedure (AZP): the aggregation of data for a larger number of zones into a prespecified smaller number of regions based on a predefined type of objective function (Openshaw, 1977; Openshaw & Rao, 1995). 5. Skater: a constrained spatial regionalization algorithm based on spanning tree pruning. Specifically, the number of edges is prespecified to be cut in a continuous tree to group spatial units into contiguous regions (Assunção et al., 2006). 6. WardSpatial: an agglomerative clustering (each observation starts in its own cluster, and pairs of clusters are chosen to merge at each step) using ward linkage (the goal is to minimize the variance of the clusters) with a spatial connectivity constraint (sklearn.cluster.AgglomerativeClustering).

Take the functionality of Max-p-regions as an example. Other methods can be applied in a similar process, including importing the needed packages, imputing and reading data, defining the parameters, solving the model, and plotting the solution.

```
from spopt.region import MaxPHeuristic as MaxP
import matplotlib.pyplot as plt
import geopandas as gpd
import libpysal
# input the data on regional incomes for Mexican states as an example.
mexico = gpd.read_file(libpysal.examples.get_path('mexicojoin.shp'))
# specify a number of parameters that will serve as input to the Max-p-regions mod
attrs_name = [f'PCGDP{2000}']
w = libpysal.weights.Queen.from_dataframe(mexico)
threshold, top_n, mexico['count'] = 6, 2, 1
threshold_name = 'count'
# solve the Max-p-regions model.
model = MaxP(mexico, w, attrs_name, threshold_name, threshold, top_n)
model.solve()
# plot the model solution.
mexico['maxp_new'] = model.labels_
mexico.plot(column='maxp_new', categorical=True,  edgecolor='w')
```

The corresponding solution of Max-p-regions running the above code is shown in Figure 1 It results in five regions, three of which have six states, and two with seven states each. Each region is a spatially connected component, as required by the Max-p-regions problem.

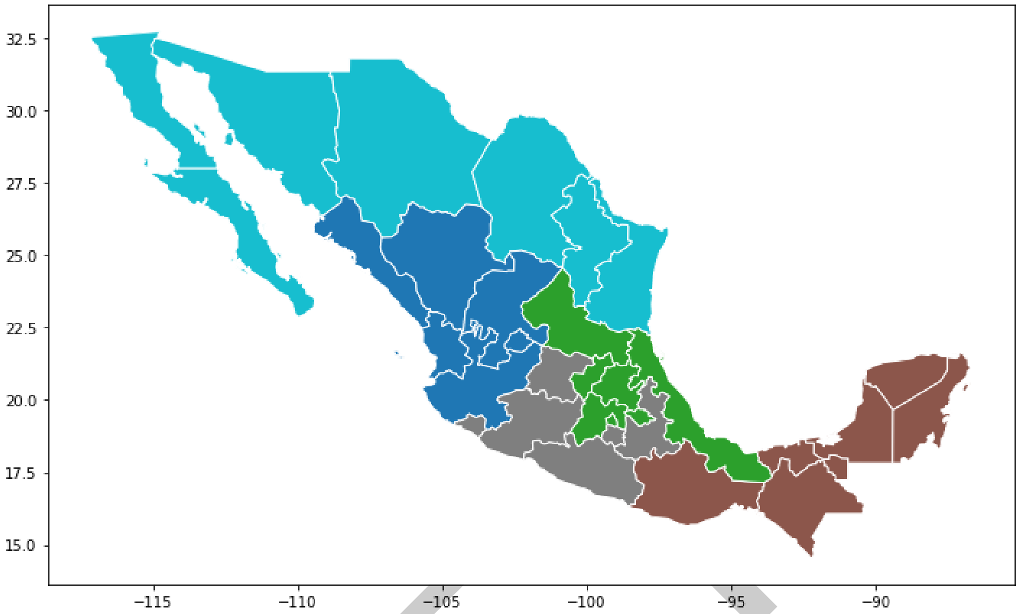

Figure 1: The solution of Max-p-regions when 32 Mexican states are clustered into the maximum number of regions such that each region has at least 6 states and homogeneity in per capita gross domestic product in 2000 is maximized.

Planned Enhancements

Spopt is under active development and the spopt developers look forward to your extensive attention and participation. In the near future, there are three major enhancements we plan to pursue for spopt:

1. The first stream will be on the enhancement of regionalization algorithms by including several novel extensions of the classical regionalization models, such as the integration of spatial data uncertainty and the shape of identified regions in the max-p-regions problem.
2. The second direction involves providing an open source version of classical single and/or multi-facility location and location-allocation methods which have been included in commercial GIS software packages.
3. We anticipate adding functionality for solving traditional routing and transportation-oriented problems. Initially, this will come in the form of integer programming formulations of the Travelling Salesperson Problem (Miller et al., 1960) and the Transportation Problem (Koopmans, 1949).

Acknowledgements

We would like to thank all the contributors to this package. Besides, we would like to extend our gratitude to all the users for inspiring and questioning this package to make it better. Spopt development was partially supported by National Science Foundation Award #1831615 RIDIR: Scalable Geospatial Analytics for Social Science Research.

References

Assunção, R. M., Neves, M. C., Câmara, G., & Costa Freitas, C. da. (2006). Efficient regionalization techniques for socio-economic geographical units using minimum spanning trees. *International Journal of Geographical Information Science*, *20*(7), 797–811. https://doi.org/10.1080/13658810600665111

Duque, J. C., Anselin, L., & Rey, S. J. (2012). THE MAX-p-REGIONS PROBLEM*. *Journal of Regional Science*, *52*(3), 397–419. https://doi.org/10.1111/j.1467-9787.2011.00743.x

Koopmans, T. C. (1949). Optimum utilization of the transportation system. *Econometrica: Journal of the Econometric Society*, 136–146. https://doi.org/10.2307/1907301

Miller, C. E., Tucker, A. W., & Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. *Journal of the ACM (JACM)*, *7*(4), 326–329. https://doi.org/10.1145/321043.321046

Murray, A. T. (2019). Contemporary optimization application through geographic information systems. *Omega*, 102176. https://doi.org/10.1016/j.omega.2019.102176

Openshaw, S. (1977). A geographical solution to scale and aggregation problems in region-building, partitioning and spatial modelling. *Transactions of the Institute of British Geographers*, 459–472. https://doi.org/10.2307/622300

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

Rey, Sergio J., & Anselin, L. (2010). PySAL: A python library of spatial analytical methods. In *Handbook of applied spatial analysis* (pp. 175–193). Springer. https://doi.org/10.1007/978-3-642-03647-7_11

Rey, Sergio J., Anselin, L., Amaral, P., Arribas-Bel, D., Cortes, R. X., Gaboardi, J. D., Kang, W., Knaap, E., Li, Z., Lumnitz, S., Oshan, T. M., Shao, H., & Wolf, L. J. (2021). The PySAL Ecosystem: Philosophy and Implementation. *Geographical Analysis*. https://doi.org/10.1111/gean.12276

Rey, Sergio J., Anselin, L., Li, X., Pahle, R., Laura, J., Li, W., & Koschinsky, J. (2015). Open geospatial analytics with PySAL. *ISPRS International Journal of Geo-Information*, *4*(2), 815–836. https://doi.org/10.3390/ijgi4020815

Wei, R., Rey, S., & Knaap, E. (2020). Efficient regionalization for spatially explicit neighborhood delineation. *International Journal of Geographical Information Science*, 1–17. https://doi.org/10.1080/13658816.2020.1759806

Wolf, L. (n.d.). *Spatially-encouraged spectral clustering: A technique for blending map typologies and regionalization*. https://doi.org/10.31219/osf.io/yzt2p