# Location-allocation and public transit: An update on UCL student teacher placements

Nick Bearman*[1], James D. Gaboardi[2], Patrick J. Roddy[3], Qunshan Zhao[4], Huanfa Chen[5] and Levi Wolf[6]

[1]Independent Scholar and Department of Geography, University College London
[2]Geospatial Science and Human Security Division, Oak Ridge National Laboratory, USA
[3]Centre for Advanced Research Computing (ARC), University College London
[4]Urban Big Data Centre (UBDC), University of Glasgow
[5]Centre for Advanced Spatial Analysis (CASA), University College London
[6]School of Geographical Sciences, University of Bristol

GISRUK 2025

**Summary**

Location-allocation is a key application of GIS, with many varied applications. We present further development of a location-allocation tool for a real-world case study with UCL using public transport. The location of UCL in Greater London (UK) means that the inclusion of public transport is vital for this case study. The location-allocation is implemented as a capacitated $p$-median location-allocation model, using `spopt`, a library in the Python Spatial Analysis Library (PySAL) ecosystem. The initial results of this work are promising, with calculation times reduced by up to 60%, and the majority of students allocated shorter journey times, with further testing ongoing.

**KEYWORDS:** location-allocation, public transport, student placement, case study.

## 1 Introduction

Location-allocation is a key element of GIS and network analysis and has many different applications within private business, the public sector, and academia. Within location-allocation, there are many different requirements for specific applications, which has resulted in the development of varied modelling approaches. This proliferation of approaches, with varied parameters and objectives, means the novice user may not be aware of the full potential of location-allocation. Equally, drive-time analysis for walking or driving is a very common application of location-allocation, but applications to public transport are much less common (Kotavaara et al., 2018). This extended abstract presents the development and application of a capacitated $p$-median location-allocation

*nick@nickbearman.com

model for a specific case study involving student teachers travelling to their placement schools via public transport across Greater London, UK. It builds on a previous publication at the AGILE conference series, adding new developments and real world application of the model (Bearman et al., 2023).

## 2 Location-Allocation

A classic application of location-allocation is the siting of facilities near to their users. In the public sector, a typical application of this is siting a new hospital. We can use location-allocation to calculate the optimal site for the new hospital, so as many residents as possible are within the shortest possible travel time (Fredriksson, 2017). At one level, this could be thought of as running a series of drive-time routes from every home to the potential hospital site and then picking the site with the total lowest travel time for all residents. The complexity increases when considering multiple sites or facilities (for example, pick the best three sites for hospitals out of a possible six), and again when you have existing sites that are predefined (we already have three hospitals, where should be build the new one out of a possible choice of six options)? This is when the location-allocation model allows us to determine the optimal location for a facility such as a hospital (Daskin, 1997). This tool can be applied in business for new stores or distribution centres (Venkatesan and Kumar, 2004), urban planning (Syam, 2008; Rahman and Smith, 2000) and many other situations.

Classically, location problems are classified into three categories (Daskin, 2008):

- covering-based problems
- median-based problems
- other problems.

The covering-based problem can be further divided into:

- Location Set Covering Problem (LSCP)
- Maximal Covering Location Problem (MCLP)
- $p$-centre location problem.

Median-based problems include two main models: $p$-median location problem, and fixed charge facility location problem (Daskin, 2008). Among these models: LSCP, MCLP, $p$-centre, and $p$-median are the most widely used in public facilities (ReVelle, 1989; Coskun and Erol, 2010).

For this case study from UCL, where trainee teachers are assigned to schools throughout Greater London, the goal is minimising travel time for trainee teachers to reach their assigned schools while accounting for a range of criteria. To meet the requirements, we implemented a variant of the $p$-median model, which is designed to minimise the total travel distance or time of customers to the nearest facility (Serra and Marianov, 1998). This will provide an optimal allocation of trainee teachers to schools, whilst minimising their overall travel time via public transport.

## 3 Current status of spopt – spatial optimisation package for PySAL

The `spopt` – spatial optimisation – Python package is a federated project within the PySAL – the Python Spatial Analysis Library – ecosystem (Feng et al., 2022; Rey et al., 2022). Early releases of the package centred on functionality consisting of a suite of heuristic regionalisation methods, which was later expanded to include optimal facility location modelling. The classic $p$-median problem was among those initial models, which was supplemented with facility capacity constraints in the initial work for this project described in Bearman et al. (2023). That functionality was merged into the code base with the release of version `v0.6.0`[1], which included another variant of the $p$-median problem: the $k$-nearest $p$-median. Formulations of the problem, both classic and variants, are fully described in available tutorials[2] and Bearman et al. (2023), as well as comparisons of the resultant solutions[3].

`spopt` is available for installation[4] via the Python Package Index, Conda-Forge, or from source, with the recommended method being from Conda-Forge.

## 4 Case Study: UCL IOE Teacher Placement Updates

Each year, the UCL IOE (Institute of Education) teaches about 800 trainee secondary school teachers (students), who will teach students aged 11-18. Each trainee teacher has to take two 60-day placements at a school teaching their specialist subject. IOE currently have about 800 students to place each year, with 500 schools offering around 800 placements. They also have a similar setup for primary school teacher training (teaching students aged 4/5-11) with approximately 150 students and primary placements. A priority system is in place, with students with caring commitments being allocated shorter journeys than students without caring commitments. Currently a manual process is undertaken to allocate students to schools, a time-consuming process. Various location-allocation options were investigated, with a custom development of `spopt` chosen. Patrick J. Roddy at UCL created a Python script that preprocessed the IOE placement data, using the TfL (Transport for London) API to provide the list of origin-destination pairs for the public transport journeys. This was then used to solve a capacitated $p$-median problem with predefined facilities in `spopt`, detailed in Bearman et al. (2023).

Alongside the development with `spopt` for the UCL case study, a series of software development projects have been undertaken at UCL to allow the library to be used for trainee-teacher allocation for IOE. A further round of development has also taken place to create a front-end tool to allow the users in IOE to perform the location-allocation themselves, and allow multiple staff to use the tool at the same time. This is under current development, and the presentation will include further updates.

[1] https://github.com/pysal/spopt/releases/tag/v0.6.0
[2] https://pysal.org/spopt/notebooks/p-median.html
[3] https://pysal.org/spopt/notebooks/p-median_variations.html
[4] https://pysal.org/spopt/installation.html

Currently the software has six main stages:

1. Upload
2. Input Data
3. Compute Journeys
4. Allocate
5. Review
6. Export

**Upload** is where the user provides the school and student information, as CSV files. These are the current output from SITS and InPlace, the software used for student management across the whole of IOE. They are in a specific structured format, providing information on the student name, number, home postcode, preferences about possible transport modes, any school restrictions, and priority information relating to caring commitments.

**Input Data** is where these files are checked and any missing data highlighted where possible. Currently, this checks postcode validity and will be extended to include the presence of a correct school code, and a complete set of data for each student.

**Compute Journeys** is where the TfL API is used to calculate all possible combinations of journeys from each student to each school. This is currently quite time-consuming, and with a full data set of 800 students and 500 schools will take around 26 hours to process. UCL developers are currently working on optimising this, however this is not a major issue because this calculation only needs to be done once.

**Allocate** is the main area of this software, where an initial allocation of students to schools is performed, and shown in a grid with students listed as rows and schools listed as columns. Students allocated to a specific school are outlined with black square, with journey times for all potential school student combinations shown in minutes. Each journey has a pop-up with the journey summary (time, mode in colour and options to manually allocate) and a pop-up map is also available. Within this is the option to leave allocation to automatic, or manually allocate or exclude a specific student to or from a specific school. This stage is quick to calculate, with a set of around 100 students taking less than 5 seconds to reallocate. This is currently being optimised by UCL developers, and the plan is that users could manually reallocate specific students and the automatic allocation would update in real time.

Each school student combination cell has a background colour, green for journeys less than 60 minutes, blue for journeys between 60 and 90 minutes and purple for journeys over 90 minutes. By default, the students are listed in priority order and then alphabetically. The intention is that each age group or subject will be processed separately, with a maximum of 200-300 students each. The allocation can be altered, in which case all other students will be reallocated accordingly. Student allocations can also be locked to prevent reallocation. *Unfortunately we are not currently* *able to share screenshots of the UCL tool. We hope we will be able to share some in April at the conference.*

The **Review** screen summarises allocated journeys, and can allow the user to review the allocated journeys, select specific journeys and find overly long journeys. Figure 1 shows a mock up of how this might look. It is expected that the user will switch between the Allocate and Review screens regularly, to help allocate the "edge cases" that the algorithm is unable to solve.

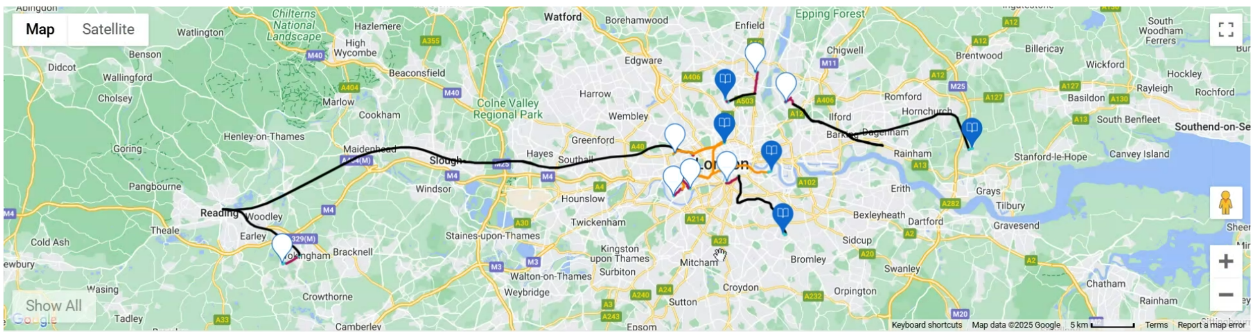


Figure 1: How part of the **Review** view might look, showing a map and and the allocated journeys. *This is not real data.* Below this would be a list summary of the allocated journeys, showing time, and a breakdown by transport mode (not shown).

The **Export** screen shows the allocation data and allows it to be exported to a CSV file. These files can then be read into InPlace, and/or shared with subject leads for review.

## 5 Initial thoughts from Users

From an initial feedback session with IOE users of the software, a number of topics were highlighted. Feedback was on a very small subset of data. The interface was easy to use, and reasonably clear to interact with. An initial setup user guide was provided, but once the data was input and the calculations completed, the Allocate and Review views were easy to use without direction.

When working in the Allocate view, staff had a number of queries about why specific students were allocated to specific schools, which did not match what they would do. Understanding the mathematically optimal allocations the software made was complicated by the test data being a subset, compared to the previous manual allocation with the complete data. The logic of the allocation process to minimise the total travel time was explained, but it was difficult for users to understand. However, most users understood and agreed with most initial allocations. We aim to make the allocation process more transparent in the software, potentially by showing the range of options for each student, and the per-student impact of altering an allocation. We intend to do a complete comparison with a full data set, both "by hand" and "by software".

Additionally, there were a number of additional restrictions that needed to be added to the software that were not present. This is due to the process being relatively complex, and while many of the requirements are recorded, a number of small edge case requirements are not, and only stored in the heads of the current users!

## 6 Data and Software Availability

We will create a workbook that demonstrates the use of this software with completely open-source data and public transport routing. There is an open example from Bearman et al. (2023), which is a Jupyter notebook that demonstrates the features of `spopt` and the techniques discussed[5].

The development of the UCL software is in active development, and future progress will be reported at the conference. Longer term, UCL have plans to roll out the use of the tool within different groups from September 2025, and consider external users after this.

## 7 Results and Conclusion

Results from this prototype are encouraging and have allowed users in IOE to take advantage of the power of location-allocation to speed up and improve their allocation of students to schools. From the early stages of the project, we knew some manual reallocation would be required, as it is infeasible to meet all possible requirements and variations. We have a number of new elements to include to improve the allocation model, as mentioned previously.

James Grindrod and the IOE team are currently testing the tool. In February and March they will be able to provide a more detailed comparison of previous allocations and new allocations using the tool, which will be reported on at the conference. Initial feedback from James has been that with the original manual allocation system, it would take approximately one week to allocate students to their placements. With the new system, he anticipates the initial automated calculation for the journeys with the TfL API will take about 26 hours to run followed by 2-3 days of manually tweaking allocations for a final result. This is clearly a massive time saving on their previous methods, of at least 60%. In addition, previous testing showed that the automated allocation reduced average student travel time by 9.6 minutes per student, with an initial sample of 93 students.

## 8 Acknowledgements

The authors wish to express their thanks to Rongbo Xu for her contributions to the development of `spopt`. We would also like to thank James Grindrod (& team), Nick Mulliss in UCL IOE, Naved Kadri, Arianna Franculacci and Tim Machin in UCL ISD and Jonathan Cooper in UCL ARC for their involvement in the project and feedback on the app.

Dr Qunshan Zhao has received the ESRC's ongoing support for the Urban Big Data Centre [ES/L011921/1 and ES/S007105/1].

The following acknowledgement applies to James D. Gaboardi (affiliation 2) only:

[5]https://github.com/UCL/ioe-student-school-allocation

- This manuscript has been authored in part by UT-Battelle LLC under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

## 9 Biography

- *Nick Bearman* is a GIS Trainer and Consultant working freelance for a variety of academic and non-academic clients, as well as a Visiting Research Fellow at UCL Geography. He is interested in showing how GIS can be used in new and interesting areas.
- *James D. Gaboardi* is an Associate Research Scientist in the Geospatial Science and Human Security Division at Oak Ridge National Laboratory. He bridges the gap between pure R&D and research software engineering with a background in scientific software development and spatio-temporal modelling, specifically, network-based population and facilities.
- *Patrick J. Roddy* is a Senior Research Software Engineer at the Centre for Advanced Research Computing, UCL. Prior to this, he undertook a PhD in novel signal processing techniques in cosmology. These days he is interested in open-source software.
- *Qunshan Zhao* is currently a Professor in Urban Analytics in Urban Studies and Social Policy and Urban Big Data Centre (UBDC) in the School of Social and Political Sciences, University of Glasgow. His research interest focuses on creating a sustainable urban future and tackling related social, economic, and environmental problems by using new forms of data and related analytical approaches including GIScience (geographic information systems, remote sensing, and spatial analysis), machine learning/deep learning/statistics, operations research, sensor networks, and urban climate modeling and instrumentation.
- *Huanfa Chen* is an Associate Professor in Spatial Data Science, at the Bartlett Centre for Advanced Spatial Analysis (CASA), University College London. His research draws on GI-Science, machine learning, and spatial optimisation to address contemporary challenges in the planning and operations of urban services, incl. policing, fire services, public health, and transport.
- *Levi Wolf* is an Associate Professor of Quantitative Human Geography at the School of Geographical Sciences, University of Bristol. He works on new methods and concepts to understand segregation and sorting in economic, political, and demographic behaviour, including problems in redistricting, elections, segregation and income inequality.

## References

Bearman, N., Xu, R., Roddy, P. J., Gaboardi, J. D., Zhao, Q., Chen, H., & Wolf, L. (2023). Developing capacitated p-median location-allocation model in the spopt library to allow ucl student teacher placements using public transport. *AGILE: GIScience Series*, *4*, 20, https://doi.org/10.5194/agile-giss-4-20-2023.

Coskun, N. & Erol, R. (2010). An Optimization Model for Locating and Sizing Emergency Medical Service Stations. *Journal of Medical Systems*, *34*, 43–49, https://doi.org/10.1007/s10916-008-9214-0.

Daskin, M. (1997). Network and Discrete Location: Models, Algorithms and Applications. *Journal of the Operational Research Society*, *48*, 763–764, https://doi.org/10.1057/palgrave.jors.2600828.

Daskin, M. S. (2008). What you should know about location modeling. *Naval Research Logistics (NRL)*, *55*(4), 283–294, https://doi.org/https://doi.org/10.1002/nav.20284.

Feng, X., et al. (2022). spopt: a python package for solving spatial optimization problems in PySAL. *Journal of Open Source Software*, *7*(74), 3330, https://doi.org/10.21105/joss.03330.

Fredriksson, A. (2017). Location-allocation of public services — Citizen access, transparency and measurement. A method and evidence from Brazil and Sweden. *Socio-Economic Planning Sciences*, *59*, 1–12, https://doi.org/10.1016/j.seps.2016.09.008.

Kotavaara, O., Pohjosenperä, T., & Rusanen, J. (2018). Integrated location-allocation of private car and public transport users — Primary health care facility allocation in the Oulu Region of Finland. *AGILE 2018*, (pp.$\tilde{2}$3)., https://doi.org/10.5194/agile-giss-3-23-2022.

Rahman, S. & Smith, D. K. (2000). Use of location-allocation models in health service development planning in developing nations. *European Journal of Operational Research*, *123*, 437–452, https://doi.org/10.1016/S0377-2217(99)00289-1.

ReVelle, C. (1989). Review, extension and prediction in emergency service siting models. *European Journal of Operational Research*, *40*, 58–69, https://doi.org/10.1016/0377-2217(89)90272-5.

Rey, S. J., et al. (2022). The PySAL Ecosystem: Philosophy and Implementation. *Geographical Analysis*, *54*(3), 467–487, https://doi.org/https://doi.org/10.1111/gean.12276.

Serra, D. & Marianov, V. (1998). The *p*-median problem in a changing network: the case of Barcelona. *Location Science*, *6*, 383–394, https://doi.org/10.1016/S0966-8349(98)00049-7.

Syam, S. S. (2008). A multiple server location-allocation model for service system design. *Computers & Operations Research*, *35*, 2248–2265, https://doi.org/10.1016/j.cor.2006.10.019.

Venkatesan, R. & Kumar, V. (2004). A Customer Lifetime Value Framework for Customer Selection and Resource Allocation Strategy. *Journal of Marketing*, *68*, 106–125, https://doi.org/10.1509/jmkg.68.4.106.42728.