

A COMPARATIVE STUDY OF METHODS FOR DRIVE TIME ESTIMATION ON GEOSPATIAL BIG DATA: A CASE STUDY IN USA

X. Fu^{1,2*}, D. Kakkar¹, J. Chen¹, K. M. Moynihan³, T. A. Hegland⁴, J. Blossom¹

¹ Center for Geographic Analysis (CGA), Harvard University, Cambridge Massachusetts
- xiaokang_fu@fas.harvard.edu, kakkar@fas.harvard.edu, junyi.chen064@gmail.com, jrblossom@cga.harvard.edu

² State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University, China
- whistle_ant@whu.edu.cn

³ Boston Children's Hospital and Harvard Medical School, Boston, Massachusetts - Katie.Moynihan@cardio.chboston.org

⁴ Agency for Healthcare Research and Quality, Rockville, Maryland - Thomas.Hegland@ahrq.hhs.gov

KEY WORDS: Drive Time, Big Data, Network Analysis, Geospatial, Open Source, High-Performance Computing

ABSTRACT:

Travel time estimation is crucial for several geospatial research studies, particularly healthcare accessibility studies. This paper presents a comparative study of six methods for drive time estimation on geospatial big data in the USA. The comparison is done with respect to the cost, accuracy, and scalability of these methods. The six methods examined are Google Maps API, Bing Maps API, Esri Routing Web Service, ArcGIS Pro Desktop, OpenStreetMap NetworkX (OSMnx), and Open Source Routing Machine (OSRM). Our case study involves calculating driving times of 10,000 origin-destination (OD) pairs between ZIP code population centroids and pediatric hospitals in the USA. We found that OSRM provides a low-cost, accurate, and efficient solution for calculating travel time on geospatial big data. Our study provides valuable insight into selecting the most appropriate drive time estimation method and is a benchmark for comparing the six different methods. Our open-source scripts are published on GitHub (<https://github.com/wybert/Comparative-Study-of-Methods-for-Drive-Time-Estimation>) to facilitate further usage and research by the wider academic community.

1. INTRODUCTION

Estimating drive times is critical for various disciplines, including urban planning, transportation engineering, business management, public health, and healthcare accessibility studies (Hu et al., 2020). In public health and medical service accessibility studies, it is often critical to know the travel time between patient locations and health services, clinics, or hospitals (Weiss et al., 2020). Accurate and efficient drive time estimation plays a pivotal role in informing decisions and understanding spatial relationships in these fields. Despite the availability of various drive time estimation methods, there is a noticeable lack of comprehensive comparative analysis to guide researchers and professionals in choosing the most suitable method for their specific needs, particularly when dealing with geospatial big data.

The use case for our project involves calculating driving times between ZIP code population centroids and pediatric hospitals, which is part of a larger project aimed at obtaining a better understanding of the quantity and quality of pediatric hospital capacity in the USA. The geospatial analytical goal of the project was to calculate driving times between 35,352 ZIP code population centroids and 928 hospitals, making for 32.8 million total calculations. Due to this massive amount of calculations, we wanted to evaluate available computation methods to identify the most efficient, cost-effective method to use. For this evaluation, we developed a sample dataset of 10,000 ZIP/hospital pairs to test with each method.

This paper presents a comparative study of six drive time estimation methods with respect to accuracy, cost, and scalability.

* Corresponding author

ity using a case study in the USA. The methods examined include the web service APIs Google, Bing, and Esri, Geographic Information System (GIS) based software ArcGIS Pro, and open-source solutions OpenStreetMap NetworkX (OSMnx) and Open Source Routing Machine (OSRM). Our case study encompasses over 32 million calculations, assessing the driving time between USA ZIP code population centroids and hospitals offering pediatric services. The primary objective of this research is to provide a benchmark comparison model and valuable guidance for selecting the most appropriate drive time estimation method for geospatial big data projects.

The paper is organized into 7 sections: Section 2 introduces the several methods, tools, and services which are commonly used for travel time distance calculations. Section 3 and 4 introduce the study area and describe the dataset which is used for comparative analysis. In Section 5, we outline the methodology used for calculating drive times using each of the six methods and discuss the data processing and visualization techniques. Section 6 presents the results of the comparative analysis, highlighting the key differences in accuracy, cost, and efficiency among the methods. Finally, Section 7 draws conclusions based on our findings, recommends the most suitable method for the given context, and suggests potential avenues for future research in drive time estimation on geospatial big data.

2. LITERATURE REVIEW

2.1 Drive Time Calculation Methods

In recent years, numerous methods for calculating drive times have emerged, each offering unique advantages and limitations. The different methods used are discussed below:

- (1) **Straight Line Distances:** The simplest way is to calculate the straight line distance or geodesic distance or the great-circle distance and divide it by an appropriate speed (Wang et al., 2019).
 - (2) **Graph Theory:** The more realistic method usually takes into account the actual road conditions (Wang et al., 2014). It allows the user to set different speeds for different levels of road networks to get a more accurate estimate. We can also set different speeds for different travel methods such as walking and driving according to our needs. The accuracy of this approach usually depends on the quality and availability of the road network data. Travel distance calculations based on road networks typically rely on routing algorithms from graph theory, such as Breadth-first Search, Dijkstra (Wu et al., 2019), Floyd-Warshall, A* (Pfooser et al., 2006), and the Bellman-Ford Algorithm.
 - (3) **Big Data Technology:** More accurate methods need to consider traffic conditions, which usually require the processing of geospatial big data. The key to these types of methods is to predict the state of the traffic. Attila and Vilmos summarize these methods as naive models such as the Instantaneous Travel Times (ITT) approach, parametric models such as Time Series Models, and non-parametric models such as Bayesian Networks and K-Nearest Neighbors Models (Nagy and Simon, 2018).
- (3) **Open-Source Packages:** These packages are based on programming languages such as C++, Java, Python, R and more. Most of the packages are usually based on the OSM road network data. OSMnx (Boeing, 2017) is a Python package that allows users to download the data and calculate the travel time. It supports calculations of work distances, bike distances, and drive distances using Python-based Networkx (Hagberg et al., 2008) and C++ based iGraph (Csardi et al., 2006). It can be easily implemented at the city level. Open Source Routing Machine (OSRM) (Luxen and Vetter, 2011) is a routing engine written in C++ that allows users to calculate the walk, bike, and drive routing and travel time distances. It allows for the calculation of the routing and travel time distances at the country and even continent levels.

3. STUDY AREA

The United States of America (USA) has been chosen as the study area for this research. The study area for this comparative analysis encompasses the 48 contiguous United States, providing a diverse range of urban and rural contexts to evaluate the performance of drive time estimation methods. Alaska and Hawaii were excluded from the analysis due to their lack of road network connectivity with the rest of the United States.

The USA also consists of a well-developed and extensive road network, which allows for a comprehensive analysis of drive time estimates across different regions and environments. Furthermore, the availability of detailed and up-to-date geospatial data, such as USA ZIP code population centroids and locations of hospitals, makes the USA a good choice for examining the accuracy, cost, and efficiency of drive time estimation methods across large geographic areas. By focusing on the USA as our main study area, we aim to provide valuable insights and guidance for researchers and professionals working with geospatial big data projects in similar contexts.

4. DATA

The dataset used in this study consists of two main components: USA ZIP code population centroids and hospital locations.

- (1) **Web APIs:** These services are provided by large technology companies such as Google, Microsoft, and Esri, and can also be used to calculate travel time. These services usually consider road networks, multi-modal traffic modes, and traffic conditions based on big data processing and machine learning algorithms. These methods usually only have a limited free number of calls for a single user and are difficult to meet a large number of calls. However, they are useful for cross-validation of the results of other methods or tools.
- (2) **Geographic Information System (GIS):** The tools such as ArcGIS, QGIS, and PostGIS can calculate straight-line distances, geodesic distances, great circle distances, and travel distances based on road networks. There are some plugins based on these GIS platforms developed by the researchers, such as Jonathan Chambers who use the 'st_closestpoint' function in PostGIS to find the closest point on the closest road for each building (Jonathan, 2020). With the development of large-scale open source data such as Open Street Map (OSM), more and more regions can easily obtain road network data. This makes it possible to calculate travel distances based on road networks for large-scale geographies such as countries and even continents.
- (1) **ZIP Code Centroids:** We obtained the geographic coordinates of population centroids for 35,352 ZIP codes from the US Department of Urban Development (Department of Housing and Urban Development, 2022). These centroids represent the central point of the ZIP code areas as determined by population distribution as opposed to geographic area. For our study, these locations represent the locations of pediatric residents in need of healthcare services.
- (2) **Pediatric Hospitals:** We compiled a list of 928 hospitals offering pediatric services across the USA using data from the American Hospital Association and other relevant sources. The dataset consists of the geographic coordinates and basic information about each hospital.

In addition to the primary data on ZIP code centroids and pediatric hospitals, we utilized the following supplementary data sources to support the analysis:

- (1) **Road Network Data:** We obtained road network data from ESRI for use with ArcGIS Pro and OpenStreetMap (Geofabrik GmbH and OpenStreetMap Contributors, 2018) for use with OSMnx and OSRM. These datasets provide the necessary information on road geometries, distances, and speed limits for calculating drive times.
- (2) **Traffic Data:** For the web service APIs (Google, Bing, and Esri), traffic data was automatically incorporated into the drive time estimations, while the open-source packages (OSMnx and OSRM) utilized default speed limits and travel speeds for their calculations.

The combination of this diverse study area, and comprehensive data sources allowed us to thoroughly evaluate the performance of the six drive time estimation methods. Further, we identify the most suitable method for accuracy, efficiency, and cost for calculating drive times on geospatial big data.

5. METHODS

To perform the comparative analysis of the six drive time estimation methods, we propose a systematic framework which is shown in Figure 1. The figure shows our comparison process, which includes selecting use cases, generating sample data from use cases, calculating driving time, comparing results, and analyzing. Each step of the framework is described in detail in the sub-sections below.

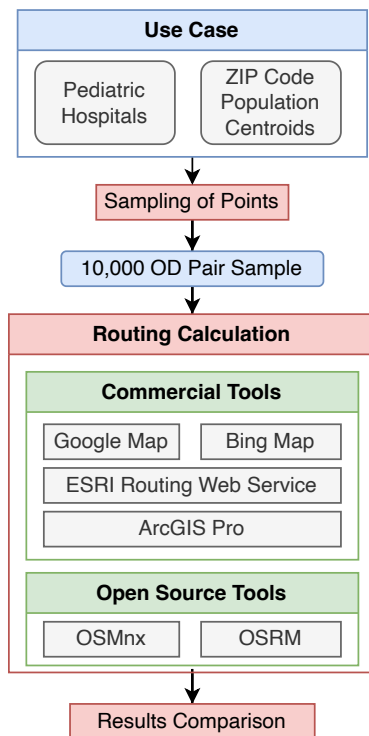


Figure 1. Framework for our comparative analysis.

5.1 Generation of Sample Data

The primary objective of this study is to conduct a comparative analysis of various methods for estimating driving time and to evaluate the accuracy, cost, and efficiency of each method. Therefore, it is essential to construct a realistic and feasible sample representative of the real world. This sample

mainly consists of carefully chosen Origin-Destination pairs (OD Pairs). The sample should strive to cover the entire study area. It should neither be too big, resulting in high computational costs nor be too small, leading to a lack of comprehensive representation in the results. Additionally, the sample should encompass point pairs of varying distances.

To address these issues, we generated random pairs of ZIP code centroids and pediatric hospitals to create a representative sample of 10,000 OD pairs for our comparative analysis. The steps of sample data generation are described below:

- (1) Generate OD pairs based on pediatric hospitals and ZIP code centroids.
- (2) Calculate the straight-line distance of different OD pairs.
- (3) Stratify the pairs into five bins based on their straight-line distances, which were used to approximate the drive time at an assumed average speed of 45 miles per hour:
 - (a) 0-15 minute drive time
 - (b) 16-30 minute drive time
 - (c) 31-60 minute drive time
 - (d) 61-120 minute drive time
 - (e) more than 120-minute drive time
- (4) From each bin, Randomly select 2,000 pairs, ensuring a total of 10,000 OD pairs for our analysis.
- (5) To guarantee comprehensive spatial representation and adequate coverage of the entire USA, we ensured that each of the 928 hospitals was represented at least twice per bin. For each hospital, a random origin was chosen within the group.

The spatial distribution of our 10,000 origin-destination pair sample is illustrated in Figure 2. As evident from the figure, our samples span across the entire conterminous USA, encompassing a diverse range of urban and rural areas, thereby ensuring good spatial representation. This comprehensive coverage helps us capture various road network characteristics and traffic conditions that are likely to influence drive time estimation in real life.

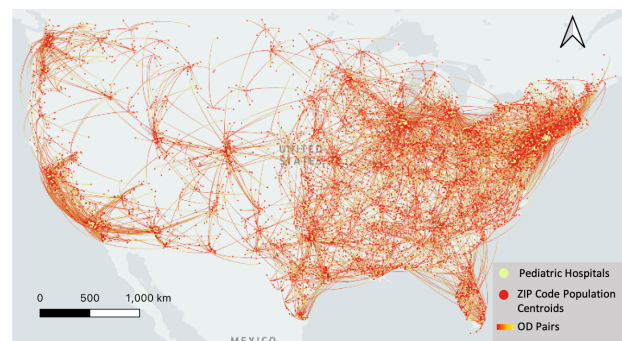


Figure 2. Geographic distribution of OD pairs.

Figure 3 below depicts the distribution of straight-line distances for these 10,000 OD pairs. We can observe that the distances vary widely, ranging from 0 to approximately 650 km. This wide range of distances helps to accommodate different travel

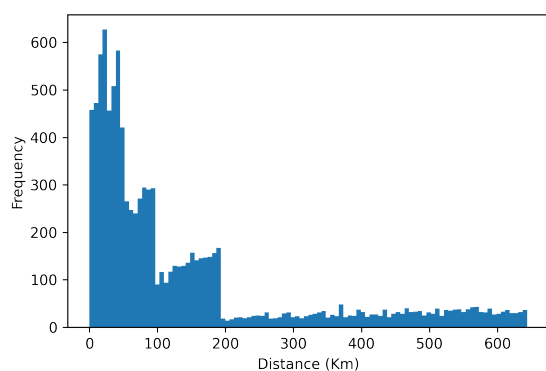


Figure 3. Distance distribution of the OD pairs.

scenarios, including short commutes within cities, medium-range travel between neighboring urban areas, and long-range travel between distant locations.

By considering such a diverse set of distances, our analysis takes into account potential variations in road network features and travel behavior across different geographical scales, ultimately providing more robust insights into the comparative performance of drive time estimation methods.

This approach allowed us to assess the accuracy, efficiency, and cost-effectiveness of each method in a real-world context. It also ensured that our results were representative of the diverse spatial and drive time characteristics (varying from a few minutes up to over 2 hours) in the USA.

5.2 Routing Calculation

In this section, we examine driving time calculations using various routing tools, including web service APIs (Google Maps API, Bing Maps API, Esri Routing Web Service), GIS desktop software (ArcGIS Pro Network Analyst), and open-source packages (OSMnx, OSRM). These tools offer distinct capabilities, enabling a thorough comparison of their performance. Our goal is to identify the most suitable method for our use case. In the following subsections, we briefly introduce each tool, emphasizing its features and the approaches used to obtain routing calculations.

- (1) Google Maps API: This widely-used web service API offers reliable routing calculations and robust geographic data, making it a popular choice for developers and researchers alike. We used the Python Requests package (Reitz et al., 2023) to submit requests and parse the results. We do not request routing calculations that consider real-time traffic to make the results comparable with other routing services.
- (2) Bing Maps API: Microsoft's mapping solution provides a user-friendly interface and accurate routing calculations, enabling seamless integration with various applications. The approaches used to obtain routing calculation here are the same as Google Maps API.
- (3) Esri Routing Web Service: Part of the ArcGIS suite, this web service API specializes in network analysis and routing, allowing users to harness Esri's powerful spatial capabilities. The approaches used to obtain routing calculation here are the same as Google Maps API.

- (4) ArcGIS Pro: A leading GIS desktop software, ArcGIS Pro version 3.0 offers advanced spatial analysis tools and the ability to calculate drive times using locally stored road network data. In ArcGIS Pro, we manually used the Network Analyst Route function to calculate drive times on the locally stored Esri road network data release from 2022.

- (5) OSMnx: This open-source Python package enables street network analysis and routing calculations using OpenStreetMap data, providing a versatile and customizable solution for various use cases. For OSMnx, we employed Python to perform street network analysis using OpenStreetMap data. It was too computationally intensive for us to load the entire US road network to OSMnx. To address this issue, we partition the OD pairs based on their respective states, disregarding those that span across state boundaries. Following this division, we perform routing calculations separately for each state. This approach ensures a more manageable analysis of the drive time estimations.

- (6) OSRM: As an open-source routing engine, OSRM leverages OpenStreetMap data to offer fast and accurate routing calculations, making it a valuable tool for developers and researchers. For OSRM, we utilized its C++ web API (Luxen and Vetter, 2011). We used both the demo server without loading road network data locally and a self-host server through Docker on our high-performance compute cluster by loading the OSM data.

By following the framework described above, we were able to compare the performance of the six drive time estimation methods and select the most appropriate method for our specific use case.

6. RESULTS AND ANALYSIS

We evaluate the processing speed, cost, and scalability of the six drive time estimation methods to determine their performance and suitability for large-scale applications. First, we examine whether each method can successfully complete the routing calculations for all samples. If yes, then we examine the time needed for these calculations (The results are shown in Figure 4). Further, we evaluate the cost needed to perform these calculations. The cost here primarily refers to the software fees required for the full computation. Finally, we investigate the scalability of these methods. Scalability refers to two aspects: whether the method can perform routing calculations for large geographic areas, such as at the countries or continents. The second is the ability to perform the calculations for a large number of routes with the support of increased resources and budgets. The results of each of the above-mentioned methods are described in the sections below:

(1) Google Maps

Google Maps took 3.05 hours to complete the calculations for 10,000 OD pairs. While it efficiently completed all calculations, Google enforces limits on request frequency. For instance, there is a 300 requests per minute per IP address limit (Google Development Team, 2019a) and a 1,000 elements per second limit, which encompasses both client-side and server-side queries (Google Development

Team, 2019b). To avoid exceeding these limits and risking suspension, we implemented a 1-second sleep interval between each request. Shortening this sleep time could potentially reduce the overall calculation time. Google provides a monthly \$200 credit for each account to access its services on Google Developers. This credit can support routing calculations for up to 40,000 OD pairs per month. This is enough for our samples so it was free of cost for our use case. Google Maps is capable of completing routing calculations on large geographies (such as countries) quickly and effortlessly. It does not require substantial computing resources from users. Nevertheless, the considerable cost associated with using Google Maps for routing calculations renders it less suitable for big data projects.

(2) Bing Maps

In our evaluation, Bing Maps API completed the calculations for 10,000 OD pairs in 3.63 hours, exhibiting comparable performance to Google Maps. For public and private Windows Apps, educational institutions, and non-profit organizations, Bing Maps permits a maximum of 50,000 cumulative billable transactions within any 24-hour period (Microsoft Development Team, 2020). Bing Maps API completed the calculation of our sample without any fees. But it is essential to note that exceeding this limit will result in additional costs, which may not be feasible for projects with tight budgets. In terms of scalability, Bing Maps is capable of successfully completing routing calculations on large geographies, such as an entire country. However, its transaction constraints limit its applicability for big data routing calculations. It is possible to overcome these limitations with an increased budget and appropriate hardware upgrades, but this might not be a practical solution for all research projects.

(3) ESRI Routing Service

ESRI's Routing Service completed the routing calculation for 10,000 OD pairs in 1.31 hours, showcasing impressive performance. ESRI Routing Service offers two authorization methods: using an ArcGIS Developer account or an ArcGIS Online account. Our organization has purchased a site-wide ArcGIS Online service that initially provides 2,000 credits per user. For non-site-wide users, the cost of credit equals 12 US cents. The credit-based system allows users to manage costs effectively while taking advantage of ESRI's powerful geospatial tools. We used 1,030 credits (\$123.60) for calculating our 10,000 samples. ESRI Routing Service provides a global road network, which can easily carry out routing calculations on a relatively large spatial scale. ESRI Routing Service is an attractive option for organizations with existing ESRI subscriptions or those seeking a cost-effective solution for large-scale routing calculations.

(4) OSMnx

OSMnx is an open-source Python package that allows users to access OpenStreetMap data and perform drive time calculations (Boeing, 2017). For our OSMnx calculation, we used a server running Ubuntu 20.04.3 LTS x86-64 operating system, with an Intel (Haswell, no TSX, IBRS) (24) @ 2.599GHz CPU. Our server was equipped with 100GB of Random Access Memory (RAM). On this system, we could not load the entire US road network into OSMnx. According to our method, we divided the OD

pairs into different states and then calculated them separately in each state. After removing the interstate OD pairs, we had 9,990 samples left. It took about 3 days to compute the drive time using an U As OSMnx is an open-source Python package, there is no fee for use. Even though OSMnx can handle city-level analyses efficiently, its speed and scalability may be limited when processing country or continent-level datasets.

(5) OSRM

OSRM offers two ways to calculate the drive time distances: OSRM demo server and OSRM Local server. The results of both these ways are discussed below:

(a) OSRM Demo Server offers a user-friendly interface for testing the OSRM routing engine like other methods based on Web APIs such as Google Maps and Bing Maps. When using the OSRM demo server, it took a total of 1.49 hours to complete the calculation. The demo server will have a service-wide rate limit of 5000 requests per minute (OSRM Development Team, 2023). But the demo server usage is restricted to reasonable, non-commercial use cases. The OSRM team suggests not exceeding 1 request per second. The OSRM demo server has no charge fee. The OSRM demo server provides support for large-scale (country-level or continent-level) routing calculations, but there are also limitations on the frequency of calls. The demo server is not designed for large-scale analyses and may experience performance issues when processing massive datasets.

(b) Running an OSRM local server on a high-performance computing cluster allows for faster processing and greater scalability compared to the demo server. To build one's own OSRM server, one needs to download and process OSM data and load it into OSRM. This step may require a server with large RAM depending on the size of the road network. Once that is done, there are no restrictions on using the local OSRM server. We launched a local OSRM server with 500GB RAM and 50 cores node on our high-performance compute cluster. OSRM Local Server was extremely fast and took less than 1 minute on our sample. As OSRM is open-source software, it is free of charge. If one has access to high-performance computing resources or a computer with hundreds of GB of RAM, OSRM Local Server can be used for extremely fast routing calculations. With optimized C++ implementation and the ability to handle large datasets, the OSRM local server is well-suited for country or continent-level analyses at no cost.

(6) ArcGIS Pro

ArcGIS Pro 3.0 required 1.5 hours of processing time to complete the calculation. However, the ArcGIS route calculation would crash after performing between 900 and 3,400 OD pairs. To perform the calculation of 10,000 pairs it was necessary to split the input into 5 batches for processing. A single license of ArcGIS Pro is \$1,500, with a maintenance cost of \$400 per year. These licensing costs may present a barrier for users without a site license, but

once purchased one can perform unlimited route calculations. Although it is a powerful platform, a major limitation is the chronic crashing of the route calculations as detailed above. This significantly increases the overall processing time and limits the scalability when dealing with large datasets and numerous OD pairs.

Figure 4 below compares the time required to process the 10,000 OD pairs using the 6 methods described above. It is worth noting that OSMnx takes the most (72 hours) and OSRM Local server takes the least time (0.01 hours).

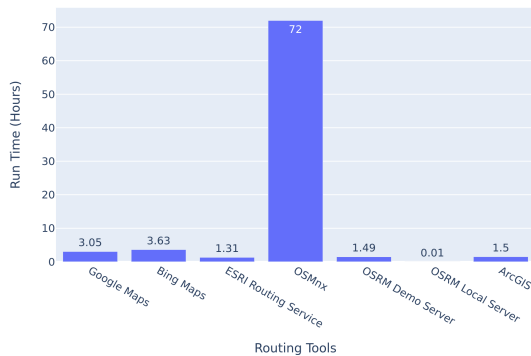


Figure 4. Time is taken to complete the calculations with different methods.

Finally, we compare the estimated drive durations generated by each method. Figure 6 illustrates the comparison of drive time for the different methods for 10,000 origin-destination pairs between USA ZIP code population centroids and pediatric hospitals. This evaluation allows us to gauge the differences between each method and identify potential discrepancies.

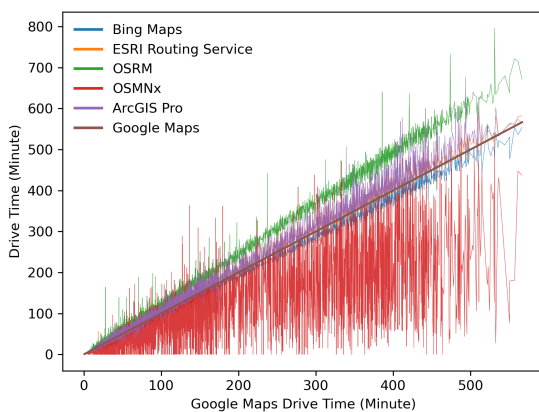


Figure 5. Comparison of estimated drive time by different methods. View this chart on an interactive website.

Our analysis revealed that the results obtained by all these methods exhibit a linear relationship. It is worth noting that, compared to the results from other tools, the overall drive time estimations from OSMnx are relatively shorter, and the fluctuations are more significant. As compared with other methods, the results from Google Maps fell in the middle in terms of estimated drive times. Because of this we chose to use Google

as the benchmark to compare all methods. The Google Maps results are closely aligned with those from the ESRI routing service and ArcGIS Pro. ArcGIS Pro's results show greater fluctuations compared to the ESRI routing service. This may be due to a slightly more accurate road network being utilized with the Esri routing service as compared to the ArcGIS Pro's static network which is from December of 2022. The results obtained by OSRM generally exhibit longer driving times than those of Google Maps, while the results from Bing Maps are typically shorter. To encourage

When examining Figure 5 it is apparent that as driving time increases, the gap between OSRM drive times and Google drive times also continues to increase slightly. To examine this difference in more detail, and to present a more nuanced look at shorter drive times, we present Figure 6 below, a zoomed-in look at just drive times less than 140 minutes. This graph shows a departure of the OSRM drive times from Google around the 50 - 60 minute mark. As 50 - 60 minute drives and longer usually involve highway travel and/or travel in rural areas, this difference may be due to how the OSRM and Google algorithms handle the computation of driving times on highways. Examining how the different methods compute highway vs. non-highway, and rural vs. urban driving environments could be a topic for future exploration.

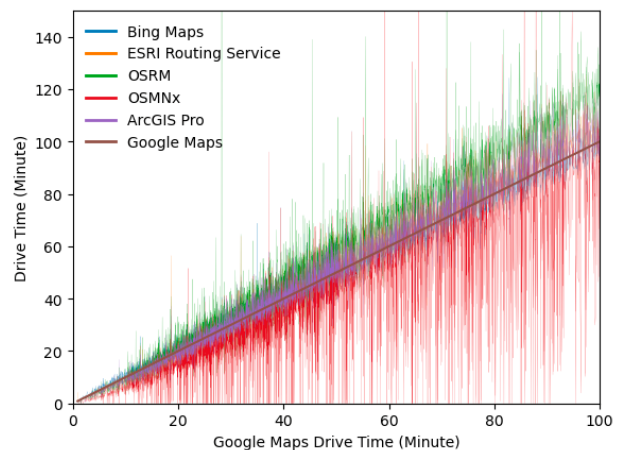


Figure 6. Detailed comparison of estimated drive times by different methods. View this chart on an interactive website.

It is important to highlight that the results calculated by these tools can be used for cross-validation, demonstrating the consistency and reliability of these methods. This also suggests that the calculations obtained using any tool can be employed for both spatial (analyzing results across different regions) and temporal (analyzing results over distinct time periods) comparisons, providing researchers and practitioners with a solid foundation for their geospatial analyses.

Based on our analysis, the speed, cost, and scalability of the various methods exhibit significant differences. While Google Maps, Bing Maps, and ESRI Routing Service offer fast and accurate solutions, their limitations in terms of daily quota and request rates may render them unsuitable for large-scale applications. Conversely, open-source or no-cost solutions like OSRM (local server) provide rapid processing, low cost, greater scalability, and consistent results, making them more suitable for geospatial big data projects. This insight is valuable for researchers and practitioners in selecting the most appropriate drive time es-

timization method for their specific needs and the scale of their projects.

7. CONCLUSIONS AND FUTURE WORK

Based on the results of our comparative study of these six drive time estimation methods using 10,000 OD pairs, we decided to use OSRM for the larger calculation of 32.8M OD pairs. OSRM based on the local server was able to do the 32.8M calculations in 6 minutes, providing an incredibly efficient, free, and accurate solution. The results of this calculation are currently being used to better understand and characterize the quantity and quality of pediatric hospital capacity in the USA.

Overall, Our study provides valuable guidance for the geospatial research community interested in performing drive time calculations. By analyzing a diverse sample of 10,000 ZIP/Hospital pairs, we were able to compare and contrast 6 drive time calculation methods. With the exception of the OSMnx solution, we found that all methods provide cost-effective, accurate results for drive time estimations of 10,000 pairs or less.

Our findings contribute to the broader understanding of drive time estimation methods and their performance in various contexts. This study serves as a benchmark for researchers and practitioners seeking to select the most appropriate method for their specific use case. It is worth noting that our analysis focused on the conterminous USA, and the performance of these methods may vary in different geographical regions. Future research should aim to replicate this comparative study in other countries, which will help to provide a more comprehensive understanding of the strengths and limitations of each method on a global scale. Additionally, to make this work more robust future studies could perform statistical comparisons between the different methods based on the time bin groupings, and explore the urban/rural differences in greater depth. By publishing our open-source scripts on GitHub, we encourage further exploration, adaptation, and application of these methods in other countries or for other purposes. Moreover, integrating empirical data on traffic conditions could further enhance the accuracy assessment of each method, supporting more informed decision-making for researchers and practitioners relying on accurate, efficient drive time calculations for their research.

ACKNOWLEDGEMENT

This work is partially funded by NSF award number 1841403 and the Society of Critical Care Medicine (SCCM) Weil Grant. We would like to thank Harvard's Faculty of Arts and Science Research Computing (FASRC) for providing computing resources for the work.

DISCLOSURE STATEMENT

The findings and conclusions in this article are those of the authors and do not necessarily represent the views of the Department of Health and Human Services or the Agency for Healthcare Research and Quality (AHRQ).

REFERENCES

Boeing, G., 2017. OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks.

Computers, Environment and Urban Systems, 65, 126–139. doi.org/10.1016/j.compenvurbsys.2017.05.004.

Csardi, G., Nepusz, T. et al., 2006. The igraph software package for complex network research. *InterJournal, complex systems*, 1695(5), 1–9. <https://igraph.org>.

Department of Housing and Urban Development, 2022. ZIP Code Population Weighted Centroids. <https://hudgis-hud.opendata.arcgis.com/datasets/HUD::zip-code-population-weighted-centroids> (23 May 2023).

Geofabrik GmbH and OpenStreetMap Contributors, 2018. Openstreetmap data extracts. Geofabrik GmbH, Amalienstraße 44, 76133 Karlsruhe, Germany. <http://download.geofabrik.de> (23 May 2023).

Google Development Team, 2019a. Maps JavaScript API usage and billing. Google Maps Platform. <https://developers.google.com/maps/documentation/javascript/usage-and-billing> (23 May 2023).

Google Development Team, 2019b. Maps JavaScript API usage and billing: other usage limits. Google Maps Platform. <https://developers.google.com/maps/documentation/distance-matrix/usage-and-billing/#other-usage-limits> (23 May 2023).

Hagberg, A. A., Schult, D. A., Swart, P. J., 2008. Exploring network structure, dynamics, and function using networkx. G. Varoquaux, T. Vaught, J. Millman (eds), *Proceedings of the 7th Python in Science Conference*, Pasadena, CA USA, 11 – 15.

Hu, Y., Wang, C., Li, R., Wang, F., 2020. Estimating a large drive time matrix between ZIP codes in the United States: a differential sampling approach. *Journal of transport geography*, 86, 102770. doi.org/10.1016/j.jtrangeo.2020.102770.

Jonathan, C., 2020. Computationally scalable geospatial network and routing analysis through multi-level spatial clustering. *MethodsX*, 7, 101072. doi.org/10.1016/j.mex.2020.101072.

Luxen, D., Vetter, C., 2011. Real-time routing with openstreetmap data. *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '11, ACM, New York, NY, USA, 513–516.

Microsoft Development Team, 2020. Maps api key: Create a map app. Microsoft. <https://www.microsoft.com/en-us/maps/create-a-bing-maps-key> (23 May 2023).

Nagy, A. M., Simon, V., 2018. Survey on traffic prediction in smart cities. *Pervasive and Mobile Computing*, 50, 148–163. doi.org/10.1016/j.pmcj.2018.07.004.

OSRM Development Team, 2023. Demo server. GitHub. <https://github.com/Project-OSRM/osrm-backend/wiki/Demo-server> (23 May 2023).

Pfoser, D., Tryfona, N., Voisard, A., 2006. Dynamic Travel Time Maps - Enabling Efficient Navigation. *18th International Conference on Scientific and Statistical Database Management (SSDBM'06)*, 369-378. doi.org/10.1109/SSDBM.2006.19.

Reitz, K., Cordasco, I., Prewitt, N., 2023. Requests: Http for humans. Read the Docs. <https://requests.readthedocs.io> (23 May 2023).

Wang, H., Tang, X., Kuo, Y.-H., Kifer, D., Li, Z. J., 2019. A Simple Baseline for Travel Time Estimation using Large-scale Trip Data. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10, 1 - 22. doi.org/10.48550/arXiv.1512.08580.

Wang, Y., Zheng, Y., Xue, Y., 2014. Travel time estimation of a path using sparse trajectories. *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, Association for Computing Machinery, New York, NY, USA, 25–34.

Weiss, D., Nelson, A., Vargas-Ruiz, C., Gligorić, K., Bavadekar, S., Gabrilovich, E., Bertozzi-Villa, A., Rozier, J., Gibson, H., Shekel, T. et al., 2020. Global maps of travel time to healthcare facilities. *Nature Medicine*, 26(12), 1835–1838. doi.org/10.1038/s41591-020-1059-1.

Wu, N., Wang, J., Zhao, W. X., Jin, Y., 2019. Learning to effectively estimate the travel time for fastest route recommendation. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, Association for Computing Machinery, New York, NY, USA, 1923–1932.