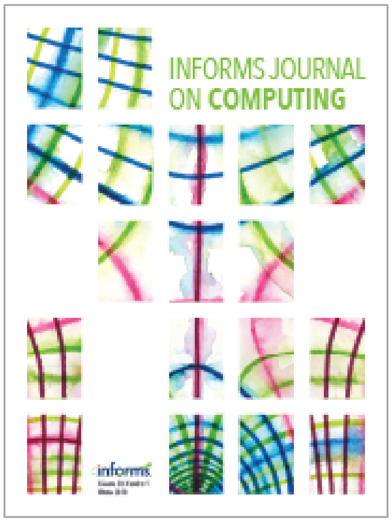


INFORMS Journal on Computing

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

VeRoViz: A Vehicle Routing Visualization Toolkit

Lan Peng, Chase Murray

INFORMS JOURNAL ON COMPUTING
Vol. 34, No. 4, July–August 2022, pp. 1842–1848
ISSN 1091-9856 (print), ISSN 1526-5528 (online)
http://pubsonline.informs.org/journal/ijoc

VeRoViz: A Vehicle Routing Visualization Toolkit

Lan Peng,[a] **Chase Murray**[a,*]

[a] Department of Industrial & Systems Engineering, University at Buffalo, Buffalo, New York 14260
*Corresponding author
Contact: lanpeng@buffalo.edu (LP); cmurray3@buffalo.edu, https://orcid.org/0000-0001-8506-5556 (CM)

Received: December 30, 2020
Revised: January 18, 2021; June 18, 2021; October 9, 2021; October 28, 2021; November 24, 2021
Accepted: December 2, 2021
Published Online in Articles in Advance: March 17, 2022

https://doi.org/10.1287/ijoc.2022.1159

Abstract. VeRoViz is an open-source vehicle routing visualization package consisting of both Python and web-based components. It was developed to streamline the workflow for vehicle routing researchers by simplifying and automating many of the tedious tasks associated with generating realistic test problems. VeRoViz also provides new functionality to produce customizable visualizations of complex vehicle routing problems. These visualization tools—including Gantt charts, static maps, and dynamic 3-D videos—assist researchers in validating models and communicating results. Additionally, a comprehensive collection of utility functions within VeRoViz provides researchers with useful tools to assess features of their problems. This paper provides an overview of VeRoViz and highlights the flexibility and ease-of-use of the toolkit.

Summary of Contribution: This paper describes an open-source software package designed to assist vehicle routing researchers. The software package, named VeRoViz (vehicle routing visualization), streamlines the process of collecting and visualizing data relevant to vehicle routing research. This includes capturing road network travel times (and distances) between locations, displaying turn-by-turn vehicle routes, generating Gantt charts of vehicle assignments, and producing 3-D "movies" of vehicle routing solutions. The VeRoViz suite consists of three components. First, a web-based interface allows researchers (and instructors) to quickly "sketch" elements of a vehicle routing problem, providing visual representations of nodes and arcs on a map. Second, the VeRoViz Python package provides an extensive collection of functions that assist operations researchers in collecting data and displaying solutions. Finally, an HTML/JavaScript plugin is available for displaying time-dynamic 3-D visualization of vehicle routes. VeRoViz is not a vehicle routing solver; it is a software suite that complements and supports researchers who are developing solution approaches to increasingly realistic vehicle routing problems (e.g., problems related to drone delivery, electric vehicles, or dynamic delivery requests).

History: Accepted by Ted Ralphs, Area Editor for Software Tools.
Supplemental Material: The software that supports the findings of this study is available within the paper and its Supplementary Information [https://pubsonline.informs.org/doi/suppl/10.1287/ijoc.2022.1159] or is available from the IJOC GitHub software repository (https://github.com/INFORMSJoC) at [http://dx.doi.org/10.5281/zenodo.5595093].

Keywords: vehicle routing • visualization • road networks • open source

1. Introduction

VeRoViz is an open-source suite of tools designed for researchers investigating problems in vehicle routing, logistics, and transportation. These tools were developed with the following user-centric aims in mind:

1. Simplify the process of collecting relevant data from a variety of disparate sources, each with different protocols and data structures. This includes travel time/distance matrices for different transportation modes (e.g., pedestrian, cycling, or automobile), turn-by-turn routing on road networks (again, with different travel modes), weather (which may be useful for drone routing), elevation changes (of importance for fuel consumption considerations or electric vehicles), and roadway classes (e.g., highways versus gravel roads). Such data are readily available but are often cumbersome to collect and format into consistent data structures.

2. Easily create maps and other visuals to validate routing algorithms, gain insights into system performance, and communicate findings. VeRoViz provides numerous visualization functions, ranging from simple Gantt charts to dynamic 3-D movies that show vehicle movements over time.

3. Leverage existing open-source libraries by providing "wrappers" that facilitate interoperability among these libraries. For example, there are libraries for geocoding (translating an address to GPS coordinates) and other libraries that plot GPS coordinates on a map. VeRoViz uses consistent data structures that enable the output from one library to seamlessly serve as input to a different library.

Table 1. A Summary of Relevant Project Links

VeRoViz homepage	https://veroviz.org
Sketch	https://veroviz.org/sketch
Installation guide	https://veroviz.org/gettingstarted.html
Detailed system documentation	https://veroviz.org/documentation.html
GitHub repository with all source code	https://github.com/optimatorlab/veroviz
Python package index (PyPI)	https://pypi.org/project/veroviz/
IJOC GitHub permanent repository (Peng and Murray 2021)	https://github.com/INFORMSJoC/2020.0340

4. Provide utility functions for commonly calculated attributes, such as the distance between two locations, conversion of measurement units (e.g., miles to meters), the heading of a vehicle, or the convex hull of a set of nodes. VeRoViz also has functions to determine the location of a vehicle at a particular time, the point on a path that is closest to a given location, or the *k*-nearest nodes to a location.

5. Generate and share realistic test problems and their solutions using road network data. By using well-documented data structures for storing node and arc data, VeRoViz makes it easy to disseminate test problems that can be imported by other researchers.

VeRoViz features three components that support these aims. First, `Sketch` is a website for manually drawing nodes and vehicle routes on a map. The second component is the veroviz Python package, which is the core tool of the VeRoViz suite. Finally, a `Cesium` (2020) plugin is provided, which allows users to view 3-D movies generated by the `veroviz` Python package. `Cesium` (2020) is an open-source JavaScript library for visualizing and analyzing temporal 3-D geospatial data in a web browser.

Table 1 provides a list of links for some key elements of VeRoViz. In addition to serving as a permanent repository of the VeRoViz source code, the IJOC GitHub page (Peng and Murray 2021) also contains additional code examples.

The remainder of this article is organized as follows. Section 2 provides an overview of `Sketch`. Next, the `veroviz` Python package is described in Section 3, followed by an overview of the `Cesium` plugin in Section 4. Concluding remarks are provided in Section 5.

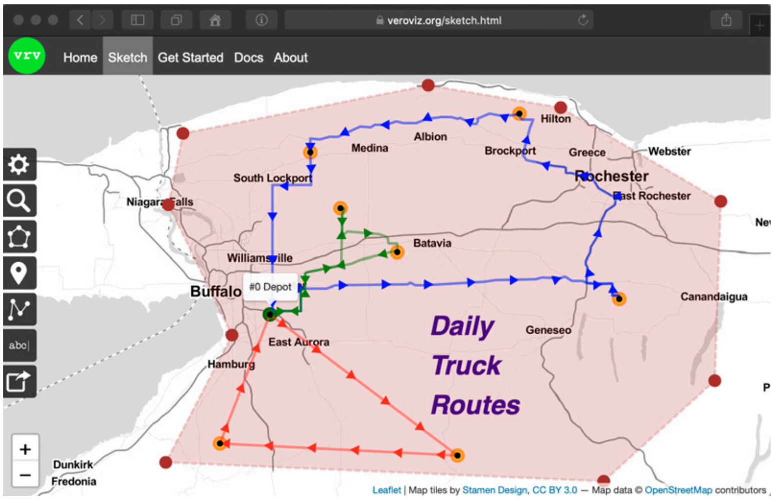
2. Sketch

Sketch, shown in Figure 1, is a web-based tool that allows users to quickly draw elements of a vehicle routing problem on a map. Using intuitive point-and-click interactions, users may place nodes (markers) and draw arcs (routes). The "snap to road" feature automatically moves the nodes to the nearest road. When drawing routes, users have the option of snapping the routes to the nearest nodes and may choose to draw the routes along the road network. Polyline "bounding regions" may be drawn to represent areas of interest. `Sketch` also allows users to add text annotations to the map. All objects (e.g., nodes, arcs, boundaries, and annotations) can be easily added, edited, or removed.

Users may select from 18 map backgrounds, including satellite imagery, topographical maps, and grayscale maps. A geocoding functionality allows users to define node locations based on an address or landmark. No downloads or installation steps are required to use `Sketch`; simply visit https://veroviz.org/sketch.

`Sketch` is an effective tool for quickly creating figures, suitable for presentations or reports, that help to describe a particular vehicle routing problem or logistic system. In the classroom, Sketch can be used to

Figure 1. (Color online) Screenshot of the Sketch Website, Showing Three Vehicle Routes



demonstrate sample solutions to traveling salesperson or vehicle routing problems. By toggling the routes between straight lines and routes restricted to road networks, students can visualize the differences between solutions based on Euclidean travel and those based on actual roadways. For researchers looking to test vehicle routing algorithms on problems with specific features (e.g., clustered distributions of customers or depot placements in varying locations), `Sketch` makes it easy to manually create such instances.

Another key feature of `Sketch`, which is not found in other online tools, such as Google Maps, is the ability to export the map objects to Python. Thus, all of the data specifying the node placements, vehicle routes, boundary regions, and annotations can be readily available for use by a vehicle routing solver.

3. The Veroviz Python Package

The veroviz Python package is the primary tool in the VeRoViz suite. It features an extensive collection of functions that assist researchers in collecting data and displaying solutions. The package is available on the Python Package Index (PyPI 2020) website and may be installed via the terminal command `pip install veroviz`.

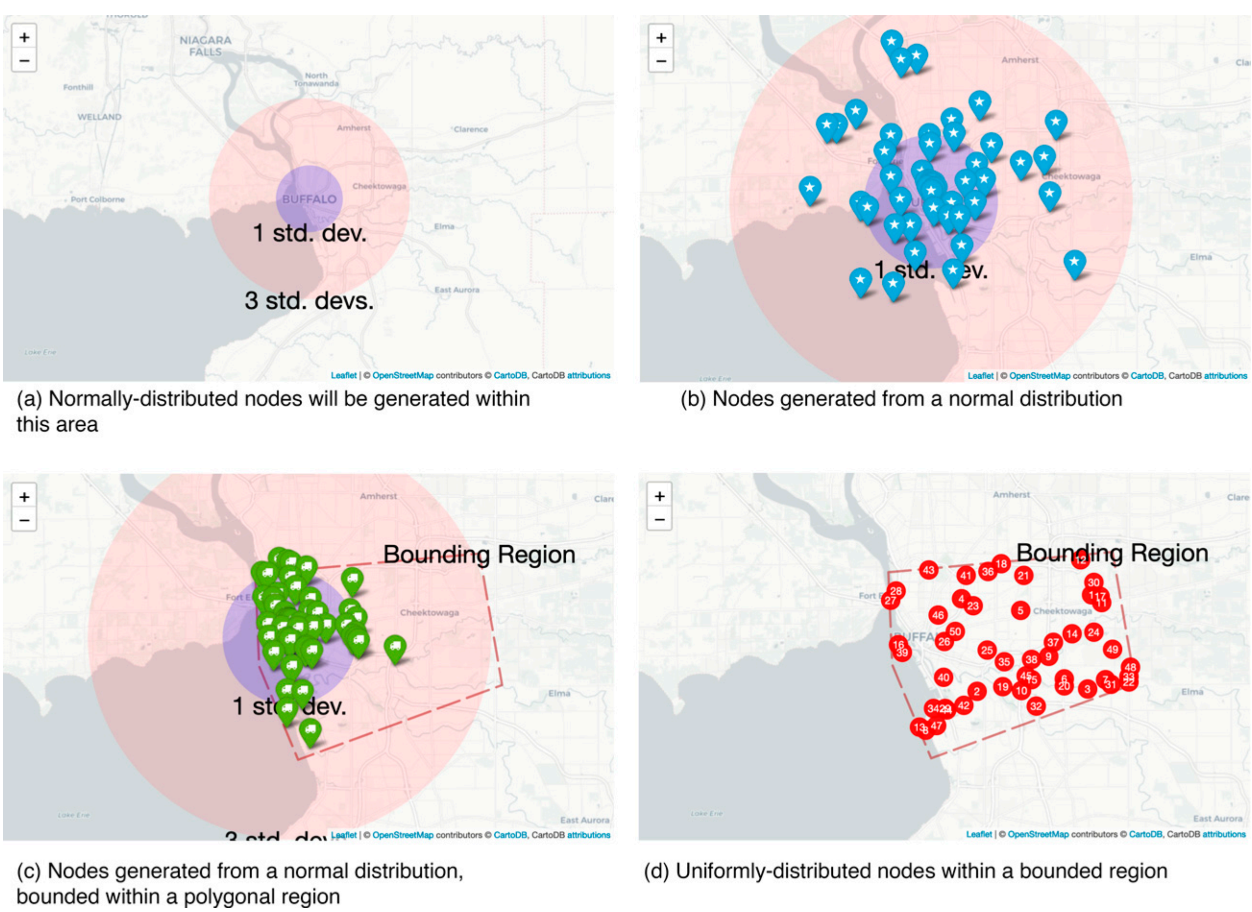
The online supplement accompanying this paper provides a more comprehensive demonstration of several VeRoViz functions via a simple vehicle routing example involving a truck and a drone in a last-mile delivery problem. A Python Jupyter notebook for the accompanying demonstration is available at https://veroviz.org/documentation.html, which also contains additional examples and details.

Some of the key applications of the `veroviz` Python package are summarized as follows.

1. Generating test problems via the `generateNodes()` function. Although there are numerous existing libraries of benchmark instances for traveling salesperson and vehicle routing problems (cf., tsplib 2013, cvrplib 2017, VRP-REP 2017, vrplib 2020), the vast majority of these problems ignore actual road networks and use Cartesian coordinates. Aside from the arc-routing instance creation software described in Lum et al. (2018), we are not aware of any other software packages for building new realistic test problems. In the case of VeRoViz, this includes generating nodes according to a variety of distributions (Figure 2) and snapping nodes to the road network (Figure 3).

2. Get travel time and distance matrices from a variety of data providers and for numerous travel profiles

Figure 2. (Color online) Example Outputs from the `GenerateNodes()` Function with 50 Nodes



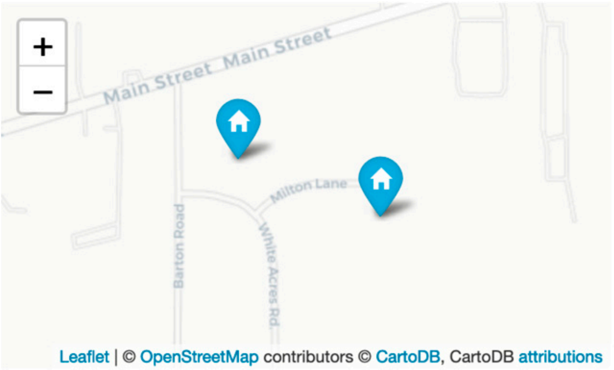
(a) Normally-distributed nodes will be generated within this area

(b) Nodes generated from a normal distribution

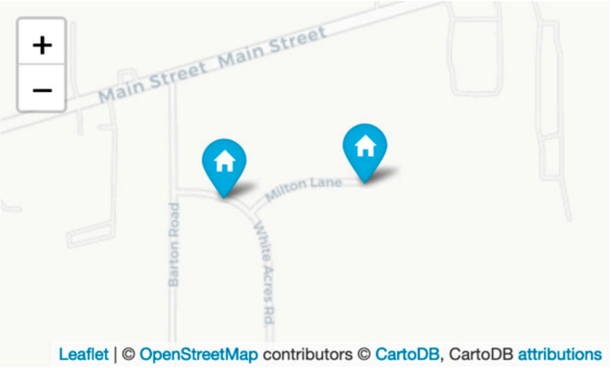
(c) Nodes generated from a normal distribution, bounded within a polygonal region

(d) Uniformly-distributed nodes within a bounded region

Figure 3. (Color online) Snapping Nodes to the Road Network, Using the `SnapNodesToRoad()` Function.



(a) Before snapping



(b) After snapping

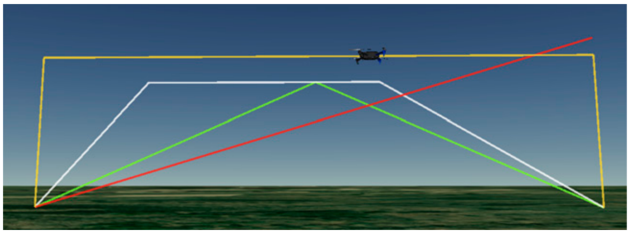
(e.g., fastest, shortest, pedestrian, or wheelchair). The `getTimeDist2D()` function accommodates multiple application programming interfaces (APIs) that provide access to such data for ground-based travel. The `getTimeDist3D()` function provides similar data for airborne vehicles, such as drones. Flight profiles for aircraft are shown in Figure 4.

3. Get turn-by-turn routes on road networks with the `getShapepoints2D()` function, which returns not only a collection of polylines that correspond to roadways but also the time at which the vehicle reaches each vertex of these polylines.

4. View nodes and arcs (VRP solutions) on interactive and customizable leaflet maps, via the `createLeaflet()` function. For example, Figure 5 shows routes for two different vehicles (a drone represented by dotted lines and a truck represented by solid lines). In Figure 5(a), the truck route is shown with simple straight-line arcs, whereas the truck's turn-by-turn route on the road network is shown in Figure 5(b). In addition to the `createLeaflet()` function, VeRoViz provides other functions for adding text, circles, polylines, and markers to maps. A variety of map backgrounds may be selected as shown in Figure 6.

5. Create dynamic 3-D movies of solutions via the `createCesium()` function. This function automatically generates data files that are imported by the VeRoViz `Cesium` plugin, described in Section 4.

Figure 4. (Color online) Available Drone Flight Profiles Include "Square", "Trapezoidal", "Triangular", and "Straight"



3.1. Data Sources

VeRoViz supports the use of numerous online sources for querying travel directions, travel time/distance matrices, elevation, weather, and isochrones (as shown in Figure 7).

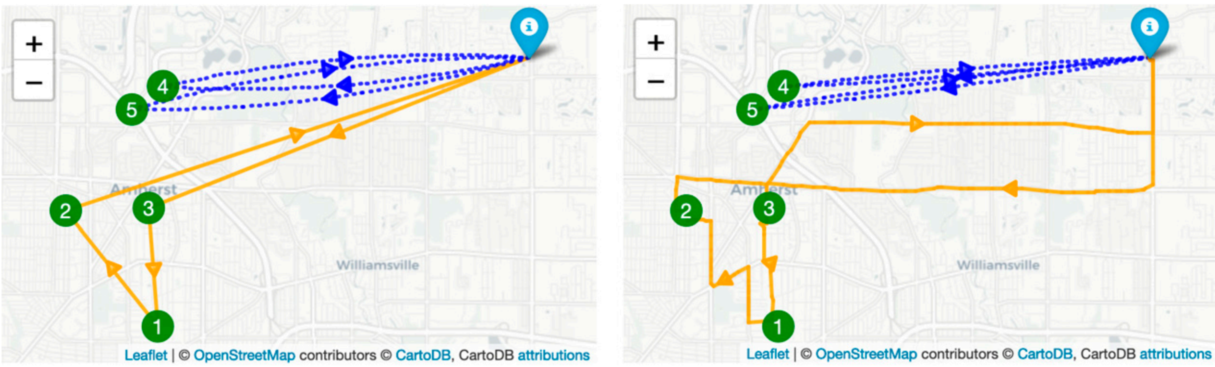
A summary of the data available from each source is given in Table 2. In the case of simple Euclidean and Manhattan routing, no external data providers are required. Integration with additional data providers is planned for future VeRoViz releases.

MapQuest (2020) is a commercial data source but offers a free API plan that allows users up to 15,000 transactions per month (a single distance matrix request counts as one transaction for up to 25 nodes at a time). The transaction limit is higher for paid plans. MapQuest generally provides more detailed routing results than the open-source data providers described subsequently, which are based OpenStreetMaps data.

The Open Source Routing Machine (2020) is available via both an online API (in which case no installation is required) or as a local installation. Currently, however, VeRoViz only supports the online API. Although no API key is required by the online API, it is hosted on a "demo" server, which is not intended for high-volume user requests. The OSRM option was included in VeRoViz to allow users the opportunity to explore VeRoViz functionality on a small scale without registering for an API key; it is not intended for use on large numerical studies.

OpenRouteService (2020) is currently the suggested data provider for VeRoViz. The online version of OpenRouteService offers a generous query limit (500 matrix requests and 2,000 direction requests per day at 40 requests per minute) with a free API key and provides geocoding services, elevation data, and isochrones. OpenRouteService users making frequent API calls may need to pause their code between requests to avoid the per-minute limitations. The open-source OpenRouteService routing engine can also be installed locally, which VeRoViz supports. The local installation has the advantage of not imposing query limits but does require

Figure 5. (Color online) The `CreateLeaflet()` Function Makes It Easy to View Vehicle Routes on a Map



(a) Simple straight arcs can be used to show the sequence of node visits, while curved arcs can show outand-back routes.

(b) Turn-by-turn road routes can be used to show ground vehicle trips.

significant hard drive space if large geographic regions are downloaded.

pgRouting (2020) is another open-source database application that must be installed locally on the user's computer. As with the local OpenRouteService installation, this has the advantage of eliminating query limits but does require downloading large quantities of road network data prior to use.

OpenWeatherMap (2020) provides current and forecast weather data via a free API key. Elevation-API (2020) provides elevation data for practically any location on the globe. Elevations at 5-km resolutions are available for free; higher resolution data may be obtained for a nominal fee. Elevation data from the U.S. Geological Survey (2020) is also available without an API key.

A key feature of VeRoViz is that it does not require separate functions for each data provider. For example, when requesting a travel time matrix, the user employs the generic `getTimeDist2D()` function with the choice of data provider given as an input argument. The output data structure is the same regardless of data provider.

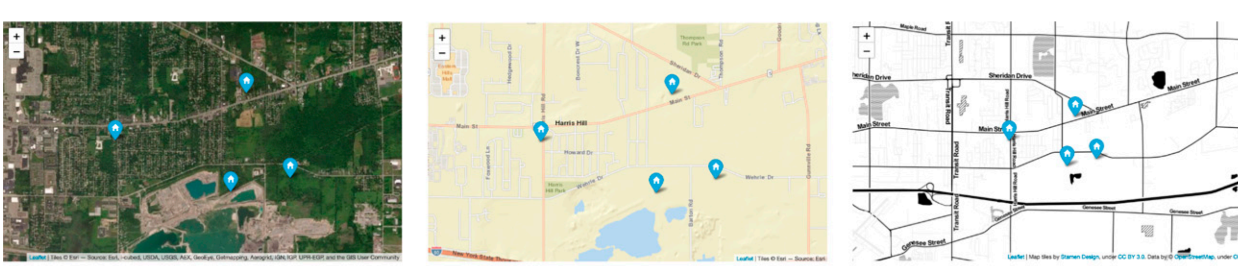
3.2. Python Libraries Used by VeRoViz

The veroviz package makes use of numerous Python libraries. A complete list is available at https://veroviz.org/credits.html. We focus on just a few of the key libraries here.

First, VeRoViz leverages `pandas` (McKinney 2010) dataframes to store and manipulate data. In particular, VeRoViz has defined three core types of pandas dataframes: "nodes," "arcs," and "assignments." The nodes dataframe houses information about single locations (vertices), including GPS coordinates, altitude, and elevation. It also includes information related to the display of nodes on maps, such as names and colors. The arcs dataframe stores information regarding pairs of origin–destination locations, which can be used to draw straight arcs on maps. The assignments dataframe extends arcs data to also include the start and end times. This dataframe is typically used to store the detailed turn-by-turn navigation returned by data providers. Details on those dataframes can be found on the VeRoViz documentation website (https://veroviz.org/docs/dataframes.html).

Second, VeRoViz uses the `Folium` (2020) library to generate interactive `Leaflet` (2020) maps. `Leaflet` is a popular open-source JavaScript library for displaying maps in web browsers. It enables users to overlay objects, such as markers, polylines, and polygons, on a variety of map backgrounds. `Leaflet` styling options are documented at https://veroviz.org/docs/leaflet_style.html. Finally, the GeoPy (2020) library provides utility functions for geocoding and for calculating distances based on latitude–longitude coordinates.

Figure 6. (Color online) A Sample of the 13 Available Map Background Tiles

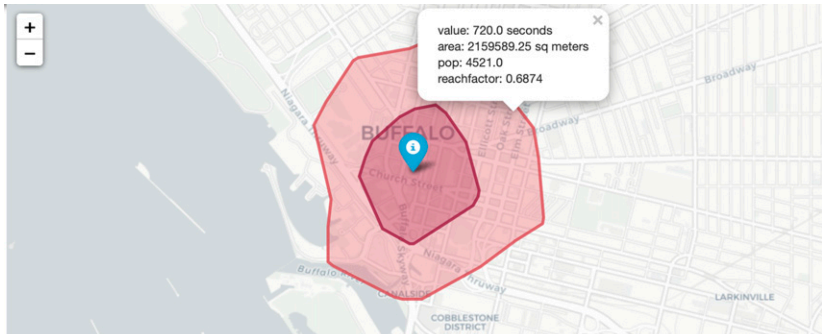


(a) ESRI Aerial

(b) ESRI Roadmap

(c) Stamen Toner

Figure 7. (Color online) Isochrones Are Lines Representing Equal Distance or Travel Time from (or to) a Given Point



Notes. The VeRoViz `isochrones()` function makes it easy to obtain this data from the Open Route Service data provider. Isochrones may be generated for various travel modes, including pedestrian, cycling, car, and heavy trucks. Here, we see isochrones for 6- and 12-minute walking times from the location marked with the pin.

4. Cesium Plugin

The VeRoViz `Cesium` viewer extension consists of an HTML file, a collection of JavaScript files, and numerous 3-D models (see Figure 8(b)) that can be displayed within `Cesium`. The `createCesium()` function generates Cesium markup language (CZML) files, a JSON-based format that describes the properties of objects (e.g., vehicles) that change positions over time. Figure 8(a) provides a screenshot of the VeRoViz `Cesium` viewer. Users may change background maps, adjust the size of the objects, and toggle visibility of vehicle routes. This is a powerful tool for analyzing routing problems in which vehicles move in three dimensions or multiple vehicles must be present at the same location simultaneously. A publicly viewable example is available at https://veroviz.org/cesiumdemo.html.

To make use of this feature, users must first install `Cesium` and `Node.js` (2020), which creates a local (private) web server on the user's computer. Next, the VeRoViz `Cesium` extension should be installed alongside the main `Cesium` application. Within Python, VeRoViz's `createCesium()` function exports information from nodes and assignments dataframes to a `Cesium`-compatible format. The user may then open the VeRoViz `Cesium` extension in a web browser to interact with a 3-D movie of the vehicle routes.

5. Discussion

This paper provides an overview of a new software toolbox, VeRoViz, that helps researchers in the vehicle routing community to sketch problems, generate test instances, visualize routing results, and calculate various metrics of interest. With the capabilities and flexibility of VeRoViz, users can easily acquire road network data and generate test instances. VeRoViz provides the research community with an open-source toolbox that can create interactive routing maps and videos with minimal code writing. In addition to providing helpful tools for researchers, we also hope that the VeRoViz data structures will facilitate the dissemination of test instances and their best-known solutions.

We look forward to feedback and contributions from the user community to identify bugs and add functionality. In the event that any of the links referenced in this

Table 2. Data Providers Used in VeRoViz

		[none]	MapQuest	OSRM	ORS-online	ORS-local	pgRouting	USGS	Elevation-API	OpenWeatherMap
	Data location		online	online	online	local	local	online	online	online
	Requires API Key		✓		✓				✓	✓
	Fastest		✓	✓	✓	✓	✓			
	Shortest		✓		✓					
	Pedestrian		✓		✓	✓				
	Cycling				✓	✓				
Route types	Truck				✓	✓				
	Wheelchair				✓					
	Euclidean	✓								
	Manhattan	✓								
	Snap nodes to road?		✓	✓	✓	✓	✓			
	Isochrones				✓					
	Elevation				✓			✓	✓	
	Geocode	✓	✓		✓					
	Weather									✓

Figure 8. (Color online) Screenshots of the VeRoViz `Cesium` Plugin

(a) An example view of vehicle routes in Cesium.

(b) 3D models included with the plugin.

paper are broken, please consult the IJOC GitHub site (https://github.com/INFORMSJoC/2020.0340) for the permanent archive (Peng and Murray 2021).

Acknowledgments

The authors are grateful to area editor Ted Ralphs for his overall management of the review process and to the two anonymous reviewers whose helpful suggestions have improved the quality of this manuscript.

References

Cesium (2020) Cesium. Accessed February 17, 2022, http://cesium.com/index.html.

cvrplib (2017) Capacitated vehicle routing problem library. Accessed February 17, 2022, http://vrp.atd-lab.inf.puc-rio.br/index.php/en/.

Elevation-API (2020) Elevation-api. Accessed February 17, 2022, https://elevation-api.io.

Folium (2020) Folium. Accessed February 17, 2022, https://python-visualization.github.io/folium/.

GeoPy (2020) GeoPy. Accessed February 17, 2022, https://geopy.readthedocs.io/en/stable/.

Leaflet (2020) Leaflet: An open-source JavaScript library for mobile-friendly interactive maps. Accessed February 17, 2022, https://leafletjs.com.

Lum O, Golden B, Wasil E (2018) An open-source desktop application for generating arc-routing benchmark instances. *INFORMS J. Comput.* 30(2):361–370.

MapQuest (2020) MapQuest. Accessed February 17, 2022, https://www.mapquest.com/.

McKinney W (2010) Data structures for statistical computing in python. van der Walt S, Millman J, eds. *Proc. Ninth Python Sci. Conf.*, June 28–July 3, Austin, TX, 45(1):56–61. 10.25080/Majora-92bf1922-00a.

Node.js (2020) Node.js. Accessed February 17, 2022, https://nodejs.org/.

Open Source Routing Machine (2020) Open source routing machine. Accessed February 17, 2022, http://project-osrm.org/.

Openrouteservice (2020) Accessed February 17, 2022, https://openrouteservice.org/.

OpenWeathermap (2020) Openweathermap. Accessed February 17, 2022, https://openweathermap.org.

Peng L, Murray C (2021) VeRoViz: A vehicle routing visualization toolkit. Accessed February 17, 2022, https://github.com/INFORMSJoC/2020.0340.

pgRouting (2020) pgrouting. Accessed February 17, 2022, http://pgrouting.org.

PyPI (2020) Python package index. Accessed February 17, 2022, https://pypi.org.

Tsplib (2013) Tsplib. Accessed February 17, 2022, http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/.

U.S. Geological Survey (2020) Elevation point query service. Accessed February 17, 2022, https://nationalmap.gov/epqs/.

Vrp-rep (2017) VRP-REP. http://www.vrp-rep.org/datasets.html.

Vrplib (2020) Vehicle routing problem library. Accessed February 17, 2022, http://or.dei.unibo.it/library/vrplib-vehicle-routing-problem-library.