The Green Paths route planning software for exposure-optimised travel

Paper Authors

- 1. Helle, Joose;
- 2. Poom, Age;
- 3. Willberg, Elias;
- 4. Toivonen, Tuuli

Paper Author Roles and Affiliations

- 1. Corresponding author; Digital Geography Lab, Department of Geosciences and Geography, University of Helsinki, Gustaf Hällströmin katu 2, 00560 Helsinki, Finland; Finnish Environment Institute SYKE, Latokartanonkaari 11, 00790 Helsinki, Finland
- 2. Digital Geography Lab, Department of Geosciences and Geography, University of Helsinki, Gustaf Hällströmin katu 2, 00560 Helsinki, Finland; Mobility Lab, Department of Geography, University of Tartu, Vanemuise 46, 50114 Tartu, Estonia
- 3. Digital Geography Lab, Department of Geosciences and Geography, University of Helsinki, Gustaf Hällströmin katu 2, 00560 Helsinki, Finland
- 4. Corresponding author; Digital Geography Lab, Department of Geosciences and Geography, University of Helsinki, Gustaf Hällströmin katu 2, 00560 Helsinki, Finland

Abstract

Green Paths is a prototype of route planning software for finding exposure-optimised routes for active travel. It incorporates external data on environmental exposures, including traffic noise levels, air quality, and street-level greenery into the street and paths network produced by the OpenStreetMap project. Written in the Python programming language, the software applies a novel environmental impedance function in the least cost path routing to find exposure-optimised routes. Routes for externally defined origin-destination pairs can be queried via a RESTful API. The API returns alternative routes equipped with rich exposure data. The published version of the software has been applied in population level environmental exposure assessment and in an end-user-oriented web-based route planner application designed for use in the Helsinki Metropolitan Area.

Keywords

Routing; environmental exposure; urban mobility; travel; walking; cycling; air quality; traffic noise; street-level greenery; spatial analysis

(1) Overview

Introduction

During urban trips, pedestrians and cyclists are exposed to the harms and benefits of the urban travel environment, such as traffic noise, air pollution, and greenery. As environmental conditions vary geographically in urban spaces, route choice becomes an important factor in determining environmental exposure during active travel [1,2]. In comparison to taking the shortest or fastest routes, environmentally

sensitive route planning is likely to reduce harmful exposures from travel and increase health benefits [3–6].

Transport, environmental and public health research has witnessed an emergence of exposure-optimised routing studies during the last decade [1]. The research field supports spatial and transport planning in creating sustainable and healthy urban space and travel options [2,7]. It also helps to advance conventional route planners that have mostly focused on minimising time, distance, or the monetary cost of travel [8]. Informed decision-making among end-users is supported by enabling access to environmental information about possible routes. Namely, pre-trip information about route qualities affects route choice if travellers perceive that the information is useful for their welfare [9,10].

Most routing studies on exposure optimisation have focused on finding low-exposure routes in respect to air pollution [e.g., 4,5,11–13]. Other studies that optimise routes also consider traffic noise levels or the presence of greenery [6,8,14,15], allergens [16], or extreme environmental conditions [17,18]. Several groups of researchers have developed new routing algorithms or prototype route planners for finding exposure-optimised routes [6,16,17,19].

Despite these efforts, real-world evidence on functional route planners that find exposure-optimised routes for active travel is limited. The few examples involve Clean Air Route Finder¹ that provides multiple choices for walking and cycling routes in London. The tool provides information about trip length, travel time, and the relative decrease in air pollution in comparison to the shortest route [20]. An open-source mobile application called *hRouting*, The Health-Optimal Route Planner, enables routes in Zürich with lower exposure to ultrafine particles to be found [21]. Some previous route planners in Canadian cities offered cyclists the opportunity to choose a route with better air quality [22–24]. Another temporarily available routing tool, a Clean Air Route Planner, was introduced in Helsinki, where walking routes were optimised according to precise air quality information [25].

To advance the research field with open-source tools, we have developed a prototype Green Paths route planning software for active travel. It enables evaluation of environmental exposure during walking and cycling, study access to healthy travel routes, and empowers citizens in making health-benefitting route decisions. The open-source routing software demonstrates how least cost path routing can be made sensitive to three environmental variables—traffic noise, air quality, and street-level greenery—in an urban context. It optimises either walking or cycling routes by combining data about environmental conditions with travel time. The Green Paths software provides multiple route suggestions between externally defined origin and destination pairs, from the fastest route to the ones with less noise and air pollution or more green views. The routing results are provided via a RESTful Application Programming Interface (API) to be used either in research or for end-user routing tools and other applications. The routing software has been developed to run in the Helsinki Metropolitan Area. For other urban areas, custom data acquisition and processing steps would be needed prior to running the application.

The Green Paths routing software shares some similarities with the previously developed Clean Air Route Planner for Helsinki [25]. Both applications use similar external data about air quality [26] and facilitate exposure-optimised routing for walking. However, there are significant differences in terms of the scope, functionality, and applicability of the Green Paths software. For example, both walking and cycling are available as travel modes and the routing functionality is based on an original environmental impedance function. The latter combines travel time, environmental cost, and sensitivity coefficients, and provides

_

¹ https://www.cleanairroutes.london/

multiple route suggestions with different trade-offs between exposure benefits and travel time costs as a result.

The aim of this paper is to describe the methodology and applicability of the Green Paths routing software. The paper provides a comprehensive view of the software, including its key features and technical architecture, and complements the complete technical documentation of the software available from GitHub. The paper does not cover the user interface of the Green Paths routing tool.

Implementation and architecture

Technical context

The Green Paths software is a prototype for exposure-optimised routing engines and applications. It is an open-source software written in the Python programming language. Python provided a suitable technical base for fast prototyping of different routing features and utilising several third-party libraries. The Green Paths software has been built as prototype for demonstrating path finding equipped with rich exposure metrics between selected origin-destination pairs. Thus, it has not been optimised to provide a routing service for extensive production use.

The complete technical documentation of the software has been published along with the source code and can be accessed via its primary README.md file. The technical documentation includes a straightforward installation guide for the software as well as other supporting technical information, such as sample graph data for testing and developing the software. As the development of the software continues, further modifications will be available from GitHub.

Table 1 lists the third-party Python libraries that the Green Paths software depends on, excluding the sub-dependencies of the libraries. In addition, the Green Paths software requires the use of the OpenTripPlanner (OTP) open-source routing software for pre-processing OpenStreetMap-based street network data. For further information, see the Graph build package section in this paper.

Table 1. The list of third-party Python libraries used in the Green Paths routing software.

Package	Link	Version (tested)	Description and use	
Flask	https://github.com/igrap h/python-igraph	1.1.2	A simple web framework for publishing the routing functionality via APIs	
GeoPandas	https://github.com/geop andas/geopandas	0.8.3	Data structures for spatial data processing and integration	
Gunicorn	https://github.com/beno itc/gunicorn	20.0.4	A Python WSGI HTTP Server for UNIX, used for running the APIs of the routing application as a web service	
igraph (python-igraph)	https://github.com/igrap h/python-igraph	0.8.3	Graph data structure and built-in algorithms for solving least cost path problems	
Shapely	https://github.com/Tobl erity/Shapely	1.7.1	Geometry objects and overlay analysis	

Features and architecture

The Green Paths software has three main features: (i) scripts and utilities for combining environmental data on noise, air quality, and greenery with OpenStreetMap-based street network graph; (ii) the main routing application for finding fastest and exposure-optimised routes for both walking and cycling; and (iii) RESTful API for publishing the environmentally sensitive routing functionalities as a web service. The architecture of the software, data processing flow, and the two use cases of the Green Paths routing software are illustrated in Figure 1.

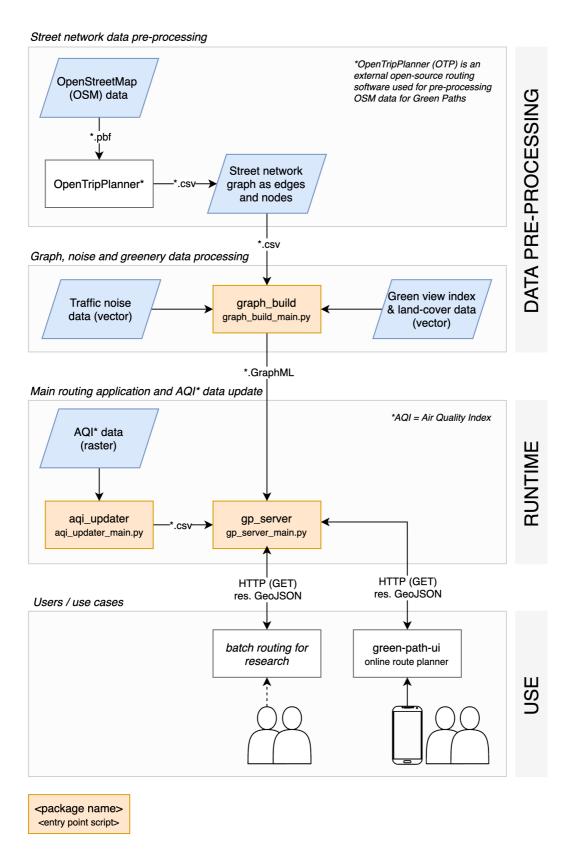


Figure 1. The architecture, data processing workflow, and use cases of the Green Paths routing software.

The software consists of four main packages that are described in more detail in the sections below:

1) graph_build: scripts and utilities for handling street network graph and environmental data;

- 2) aqi_updater: utilities for fetching, processing, and sampling either real-time or static air quality index data for the main routing application;
- 3) gp_server: the main routing application that handles routing requests;
- 4) common (omitted from the Figure 1): general utilities for handling geometry data and reading and writing street network graph data in the GraphML format (I/O).

External data

The Green Paths software uses external open-access data of street network, traffic noise, air quality as Air Quality Index (AQI), and street-level greenery as Green View Index (GVI) that are described in the Table 2.

Table 2. External data currently used in the Green Paths routing software for exposure-optimised routing.

Data	Content	Measure	Value	Format	Spatial extent	Source and references
			range			
Street	Street and path network	n/a	n/a	Standard	Global	OpenStreetMap,
network				compressed		www.openstreetmap.org,
data				Protocolbuffer		CC BY 2.0
				Binary Format		
				(PBF)		
Traffic	Day, evening and night A-	dB(A)	4075+	Vector,	European	[27], accessible from
noise	weighted equivalent			shapefile	urban areas	<pre>ckan.ymparisto.fi/dataset/</pre>
data	continuous sound pressure					<u>%7B0A7C0CF8-7BAA-</u>
	level (L _{den}) from road and					49FF-835A-
	rail traffic. Modelled every					<u>0B97DA89B9D4%7D</u> , CC
	five years according to the					BY 4.0
	EU Environmental Noise					
	Directive 2002/49/EC.					
Real-	Hourly composite air	Air Quality	15	Raster, mesh	Helsinki	FMI-ENFUSER 2.0 model,
time air	quality index covering NO ₂ ,	Index	(good	size 13x13m,	Metropolitan	[26,28], accessible from
quality	SO ₂ , O ₃ , PM _{2.5} , PM ₁₀ , BC	(AQI) 2.0	very	netCDF	Region	the Finnish Meteorological
data	and LDSA		poor)			Institute (FMI)
Static air	Annual average composite	Air Quality	15	Raster, mesh	Helsinki	FMI-ENFUSER 1.0 model,
quality	air quality index covering	Index	(good	size 13x13m,	Metropolitan	[26], accessible from the
data	NO_2 , SO_2 , O_3 , $PM_{2.5}$ and	(AQI) 1.0	very	netCDF or	Region	Helsinki Region
	PM ₁₀		poor)	TIFF		Environmental Services
						Authority (HSY)
Greenery	Proportion of green	Green	01	Vector,	Helsinki	[29], CC BY 4.0;
data	vegetation visible at street	View		shapefile	Metropolitan	hri.fi/data/fi/dataset/paak
	level based on street view	Index			Region	aupunkiseudun-
	panoramas and regional	(GVI)				maanpeiteaineisto, CC BY
	land cover data					4.0

Graph build package

The routing in the Green Paths software is based on a street network graph, which consists of nodes and edges between the nodes (representing street segments). Each edge is equipped with attributes about the geometry and distance of the edge, allowed travel modes, a biking safety factor, and environmental data. The nodes are characterised by only the geometry value. The required attributes and their value types are listed in the technical documentation of the software.

As illustrated in Figure 1, raw OpenStreetMap data in PBF format are first processed with OpenTripPlanner to create a street network graph data model. For this purpose, an executable jar-file of OpenTripPlanner is provided along with the source code of the Green Paths software. The resulting graph model with nodes,

edges, and respective attributes is imported via comma-separated values (CSV format) to the Green Paths software. The graph_build package uses the model to construct a street network graph that fits the internal graph format of the software. The Green Paths software can read and write graph objects via GraphML file format utilising custom I/O functions, which are written on top of the igraph library (module common.igraph).

During the later stages of the graph build, static external environmental data on traffic noise and greenery are sampled and attached to the edges of the graph via a spatial overlay analysis. These scripts are implemented using mostly procedural programming style.

AQI updater package

The aqi_updater package is needed for fetching, processing, and sampling Air Quality Index (AQI) values from the raster files to the street network graph. It is developed as an independent sibling package next to the gp_server package and can be used from the root of the project (aqi_updater_main.py). While the implementation of the package includes many generic functionalities, the current solution is adjusted to work in the Helsinki Metropolitan Area with the AQI data formats that are described in Table 2.

The aqi_updater reads graph data from the GraphML file and AQI data from a raster file (NetCDF or TIFF), and exports a CSV file that contains a value map of street IDs and respective AQI values. This value map can then be read by the gp_server package to update AQI values to its internal graph object at runtime.

There are two ways of using the aqi_updater, based on the definition in the configuration file of the gp_server package. The options involve either a procedural Python script for processing and sampling a local AQI raster file on demand (see the file aq_data_import.import_static_aqi_data) or a standalone application that fetches AQI data from a remote file server and processes and samples it on a scheduled manner. The latter use case is suitable for using real-time AQI values as demonstrated in the online proof-of-concept instance of the Green Paths routing software for the Helsinki Metropolitan Area.

GP server package

The gp_server package forms the main routing application of the Green Paths software and includes the following functionalities.

Configuration

Some of the runtime settings of the Green Paths software can be changed via a simple configuration module in the gp_server package (gp_server.conf). With the configurations, users can (i) define, which graph file is used, (ii) disable or enable either of the two travel modes (walking/biking), (iii) disable or enable exposure-optimised routing modes one by one, (iv) set the sensitivity coefficients that are used in exposure-optimised routing, (v) set default speed coefficients for travel modes, and (vi) disable or enable the research mode. The latter provides additional exposure information for each route. The differences between the research mode for batch routing and the end-user mode for routing single routes are explained in the technical documentation of the API.

Reading graph data

The Green Paths software is capable of finding five types of routes, including the fastest, safest (for bike, following the OpenTripPlanner capabilities), quietest, greenest routes, and routes with the freshest air. The three last environmental exposure optimised routing modes are optional. Each enabled routing mode requires the presence of corresponding edge attributes that are listed in the technical documentation. The gp_server package reads the necessary attribute values from the street network graph data in GraphML format, except for the air quality attribute values that are read from the CSV format output of the aqi_updater package.

Environmental impedance function

The software uses Dijkstra's least cost path routing algorithm, which by default is based on travel time. Travel time is derived from the distance of the edges of the graph and travel mode specific speed coefficients that are defined in the configuration file. To support exposure-optimised routing, we integrated an environmental impedance function to the travel-time-optimised routing algorithm. The function enables calculation of exposure-adjusted edge cost values (Eq. 1) that complement the travel time related base cost of edges:

$$C_e = C_t + C_t * c_e * s,$$
 (1)

where C_e is the total (composite) cost of the edge, C_t is the base cost of the edge that is proportional to travel time, c_e is an environmental cost coefficient, and s is an arbitrary sensitivity coefficient for assigning applicable weight(s) for the environmental cost component. The additional environmental cost $C_t * c_e * s$ is thus proportional to the value of the environmental variable and travel time. The definition of the environmental cost coefficient c_e depends on the exposure-optimised routing mode. It reflects either the traffic noise level, AQI value, or GVI value in the value range of respective environmental data (see Table 2).

The dynamic sensitivity coefficient *s* enables multiple alternative routes to be found for a single origin-destination pair. Multiple route suggestions enable responses to the variation of personal preferences, including varying sensitivity to environmental properties or different valuations of travel time. In other words, as it is challenging to define a single best route for everyone, the environmental impedance factor needs to include a dynamic sensitivity component that enables the environmental cost and travel time to be balanced. The gp_server package uses a list of sensitivity coefficients for routing several alternative routes. Higher sensitivity coefficients increase the weight of the environmental properties in routing, whereas lower sensitivities result in routes with travel times close to the fastest route. Routing results always include the fastest route by applying a sensitivity coefficient equal to zero. Ideally, exposure-optimised routing can find route(s) of significant improvement in environmental exposure yet with only a limited increase in travel time.

The appropriate list of sensitivity coefficients for each exposure-optimised routing mode varies depending on the value range and spatial variation of the environmental variable. Environmental variables with limited spatial variation require higher sensitivity coefficients to find alternative, exposure-optimised routes.

Start-up of the routing application and routing workflow

To find walking and cycling routes via the least cost path routing algorithm, the Green Paths software applies (edge) cost calculation principles to different travel modes and street types adopted and simplified from the OpenTripPlanner. We implemented a straightforward least cost path routing method: all cost attributes are pre-calculated and assigned to edges during the start-up of the application, i.e., prior to solving the least cost path problem with Dijkstra's algorithm. This approach makes executing Dijkstra's algorithm very simple. The downside is that the memory consumption of the application grows quickly along with the number of pre-calculated edge attributes, as an independent cost attribute is needed for each unique combination of travel mode, routing mode, and sensitivity coefficient. The basic edge cost calculation considers *length*, *allows_walking*, and *allows_biking* attributes parsed by OpenTripPlanner for assigning travel time-based costs (*base costs*) for the two travel modes. Walking with bike is also permitted, to allow streets to be crossed and reaching destinations in walk-only parts of the network (as in OpenTripPlanner). Next, exposure-adjusted edge costs are calculated from edge-level base costs and environmental exposure data using the environmental impedance function. Due to the pre-calculation of cost attributes, several advanced routing features that require dynamic cost calculation within Dijkstra's algorithm are not supported. These include turning costs for cycling and traffic light penalties.

The start-up of the routing application (gp_server_main.py) initiates the following sequence of actions to prepare the application for handling routing requests:

- 1) Graph data from an external GraphML file is read into an in-memory graph object;
- 2) Edge and node geometries are read from the graph into GeoPandas GeoDataFrames for fast lookup of the nearest nodes and edges for origins and destinations;
- 3) Travel time-based costs are calculated as edge attributes separately for both travel modes (walking & cycling);
- 4) Traffic noise-based costs are calculated for both travel modes at different sensitivities and set as edge attributes;
- 5) GVI-based costs are calculated for both travel modes at different sensitivities and set as edge attributes;
- 6) AQI update data are read and updated to the graph as edge attributes if the CSV-format output either for static or real-time AQI values from the aqi_updater application is available;
- 7) AQI-based costs are calculated for both travel modes at different sensitivities and set as edge attributes if the AQI values were updated to the graph under the step 6;
- 8) Steps 6 & 7 are repeated in a scheduled manner if real-time AQI data are available;

Steps 4–8 can be disabled from the configuration file of the gp_server package. After successfully executing these steps, the main application is ready to accept routing requests via the API endpoints defined in the module gp_server_main.py (start-up script of the routing application). The sequence of actions that is executed during handling a single routing request is visualised in Figure 2.

Prior to finding routes, the application finds the nearest edges and nodes to the user-defined origin and destination locations. If the nearest edge is closer than the nearest node, a new temporary node and linking edges are created on the nearest point of the nearest edge. The application aims to avoid creating new temporary nodes and edges, since they require re-indexing the graph prior to routing and thus they increase the routing time significantly. Optimising the resource use is achieved by using a distance difference threshold: if the nearest node is no more than 30 m further than the nearest edge, that node is used as the origin or destination location as opposed to creating a new temporary node.

Since the routing logic is embedded in the pre-calculation of edge costs, routing is executed simply by solving one least cost path problem per each pre-calculated cost attribute as per the user-defined routing options (OD, travel mode and exposure mode). Routing includes finding both fastest and several exposure optimised routes, as illustrated in Figure 2.

After routing, the identified routes are initially only known as sequences of edge IDs. Complete route information, including the length and environmental exposure properties, are collected and calculated from the graph data by edge IDs. Averaged environmental exposure properties such as mean traffic noise, AQI, and GVI values are then calculated from edge-level properties. Also, accumulated distances within each traffic noise, AQI, and GVI range are calculated and added as route properties.

The maximum number of exposure optimised routes for one origin-destination pair is bounded by the number of sensitivity coefficients that are in use. Since employing the sensitivity coefficients aims to result in routes of little variance in travel time but significant improvement in environmental exposure, many of the identified routes are almost identical by geometry. Therefore, prior to returning the routes to the user, identical and nearly identical routes are filtered out using a geometrical overlay analysis (with 50 m buffers).

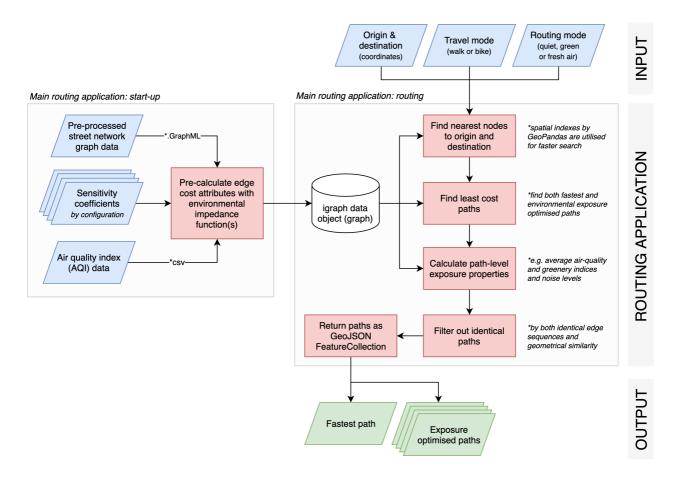


Figure 2. Flowchart of the start-up and the routing workflow of the main routing application (gp_server / gp_server_main.py).

API and response schema

The Green Paths software publishes its routing functionality as a RESTful API. The API accepts HTTP GET requests that contain the necessary routing settings as (URL) path variables.

In the case of a routing error, the response status code indicates the type of the error, and the response data contains a descriptive error message. After successful routing, the API returns all route information in GeoJSON FeatureCollection format (Figure 2). Each route features several descriptive properties, including exposure profiles for traffic noise, air quality, and greenery, and general information such as total distance. The API and its usage are documented in more detail in the technical documentation.

Quality control

The software code features several API tests and some unit tests that ensure the basic routing functionality. The testing approach focuses mainly on higher level regression testing (API tests), which allowed heavy refactoring of the architecture as well as implementation details whenever necessary. We automated the test runs as continuous integration pipelines using GitHub Actions. This ensured that the software remained functional before changes were pushed to the main branch.

(2) Availability

Operating system

Linux, OS X, and Windows.

Programming language

Python 3.8.

Additional system requirements

The CPU and memory requirements for the Green Paths software depend primarily on the size of the street network graph data, as the complete graph is loaded to memory at runtime. The size of the graph data file for HMA, including greenery and noise attributes, is roughly 500 MB. Hence, at least 500 + 100 MB memory is needed to run the application successfully. In practice, running several parallel instances of the application is recommended to allow a larger number and concurrent routing requests. Parallelisation can be achieved either with running the application in Docker containers, using multiple Gunicorn workers, or with a mixture of the two.

Dependencies

The Green Paths software requires installation of the following Python libraries: igraph, GeoPandas, Shapely, Flask, Gunicorn and the sub-dependencies of these. The complete Python environment is described as a YAML environment file and can be easily installed with the Conda package manager.

List of contributors

This software was developed and designed by Joose Helle (UH & SYKE) with support from Age Poom (UH, project management, conceptualisation, testing, and research), Elias Willberg (UH, testing and research), Tuomas Väisänen (UH, technical support), and Tuuli Toivonen (UH, conceptualisation and supervision).

Software location

Archive:

Name: Zenodo

Persistent identifier: DOI: 10.5281/zenodo.5409403

Licence: MIT licence

Version published: 0.8.0

Date published: 03/09/2021

Code repository:

Name: GitHub

Persistent identifier: https://github.com/DigitalGeographyLab/green-path-server

Licence: MIT licence

Version published: 0.8.0

Date published: 03/09/2021 (v0.8.0)

Language

The software and its documentation are written in English.

(3) Reuse potential

The published version of the proof-of-concept software for exposure-optimised routing has been developed to work in the Helsinki Metropolitan Area based on the locally available environmental data. A complete installation guide is provided along with the source code, including data download steps for the Helsinki Metropolitan Area. The ready-made sample street network graph for the region includes traffic noise and street-level greenery information.

To reuse the Green Paths software in spatial contexts other than in the Helsinki Metropolitan Area, both OpenStreetMap-based street network data and environmental data are needed. Provided that OpenStreetMap data are available for many cities in a standard format and the application is written with modular Python, the software can be adopted and applied elsewhere, both for research and route planner applications.

A substantial proportion of the software functionality of handling environmental data requires localisation, depending on the available data and their characteristics, such as file formats, value ranges, and spatial extents. For adopting the software to other locations, the need for modification or developing new implementation solutions depends on the similarity of the available data in those locations with the data applied in the current version of the software, specific to the Helsinki Metropolitan Area. Specifically, local implementation requires adaptation of the packages graph_builder and aqi_updater to match the available data formats and structures. The broad steps for data pre-processing and integration are described in the technical documentation.

The software is currently being applied to research to analyse multiple exposures during realised mobility and the accessibility of healthy travel environments in Helsinki. The software is also applied in a prototype end-user route planner Green Paths, accessible at https://green-paths.web.app/. The end-user web application was developed in the European Union funded project Healthy Outdoor Premises for Everyone (UIA HOPE)². It visualises the results returned by the API by suggesting multiple routes between the user-defined origin-destination pair. It presents the results separately for walking and cycling, and for quiet, green, and fresh air routes (Figure 3). The web application also visualises environmental data in the street network graph as optional background map layers.

_

² https://ilmanlaatu.eu/briefly-in-english/

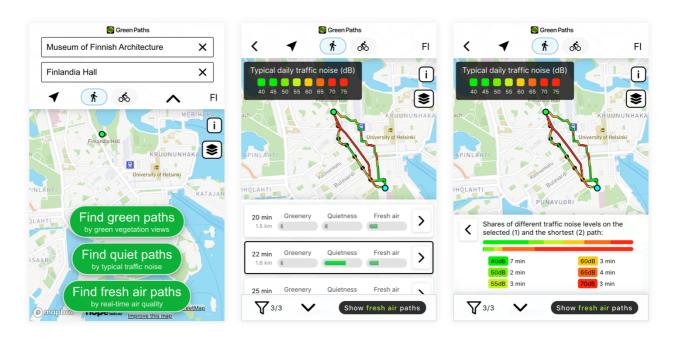


Figure 3. Selected screenshots of the Green Paths routing tool user interface for finding and comparing quiet routes for walking. Source: https://green-paths.web.app.

Acknowledgements

We would like to thank the Finnish Meteorological Institute, the Helsinki Region Environmental Services Authority, the City of Helsinki, Forum Virium Helsinki, and other partners from the UIA HOPE project, and the members of the Digital Geography Lab, University of Helsinki, for their input and support. We also wish to pass on our gratitude to all parties who made environmental data on street and path network, traffic noise, air quality, and street-level greenery in the Helsinki Metropolitan Area openly available and thus enabled the demonstration of exposure-optimised routing features of the Green Paths software: Akseli Toikka, Ville Mäkinen, Juha Oksanen, Lasse Johansson, the cities of Helsinki, Espoo, Vantaa, and Kauniainen, Finnish Meteorological Institute, Helsinki Region Environmental Services Authority, Finnish Environment Institute, and the OpenStreetMap project.

Funding statement

This work was supported by the European Regional Development Fund via the Urban Innovation Action – Healthy Outdoor Premises for Everyone (UIA HOPE), project ID UIA03-240, and the Amer Cultural Foundation in Finland via the Sporttia Stadiin/Urban Exerciser project.

Competing interests

The authors have no competing interests to declare.

References

[1] Poom A, Willberg E, Toivonen T. Environmental exposure during travel: A research review and

- suggestions forward. Health Place 2021;70:102584. https://doi.org/10.1016/j.healthplace.2021.102584.
- [2] Doorley R, Pakrashi V, Szeto WY, Ghosh B. Designing cycle networks to maximize health, environmental, and travel time impacts: An optimization-based approach. Int J Sustain Transp 2020;14:361–74. https://doi.org/10.1080/15568318.2018.1559899.
- [3] Brand VS, Kumar P, Damascena AS, Pritchard JP, Geurs KT, Andrade M de F. Impact of route choice and period of the day on cyclists' exposure to black carbon in London, Rotterdam and São Paulo. J Transp Geogr 2019;76:153–65. https://doi.org/10.1016/j.jtrangeo.2019.03.007.
- [4] Alam MS, Perugu H, McNabola A. A comparison of route-choice navigation across air pollution exposure, CO2 emission and traditional travel cost factors. Transp Res Part D Transp Environ 2018;65:82–100. https://doi.org/10.1016/j.trd.2018.08.007.
- [5] Mölter A, Lindley S. Influence of walking route choice on primary school children's exposure to air pollution A proof of concept study using simulation. Sci Total Environ 2015;530–531:257–62. https://doi.org/10.1016/j.scitotenv.2015.05.118.
- [6] Novack T, Wang Z, Zipf A. A system for generating customized pleasant pedestrian routes based on openstreetmap data. Sensors (Switzerland) 2018;18. https://doi.org/10.3390/s18113794.
- [7] Nieuwenhuijsen MJ. Urban and transport planning pathways to carbon neutral, liveable and healthy cities; A review of the current evidence. Environ Int 2020;140:105661. https://doi.org/10.1016/j.envint.2020.105661.
- [8] Ribeiro P, Mendes JFG. Route planning for soft modes of transport: Healthy routes. In: Pratelli A, Brebbia CA, editors. Urban Transp. XVII. WIT Trans. Built Environ., vol. 116, WIT PRESS; 2011, p. 677–88. https://doi.org/10.2495/UT110571.
- [9] Rapoport A, Gisches EJ, Daniel T, Lindsey R. Pre-trip information and route-choice decisions with stochastic travel conditions: Experiment. Transp Res Part B Methodol 2014;68:154–72. https://doi.org/10.1016/j.trb.2014.06.007.
- [10] Jiang G, Fosgerau M, Lo HK. Route choice, travel time variability, and rational inattention. Transp Res Part B Methodol 2020;132:188–207. https://doi.org/10.1016/j.trb.2019.05.020.
- [11] Davies G, Whyatt JD. A network-based approach for estimating pedestrian journey-time exposure to air pollution. Sci Total Environ 2014;485:62–70. https://doi.org/10.1016/j.scitotenv.2014.03.038.
- [12] Zou B, Li S, Zheng Z, Zhan BF, Yang Z, Wan N. Healthier routes planning: A new method and online implementation for minimizing air pollution exposure risk. Comput Environ Urban Syst 2020;80:101456. https://doi.org/10.1016/j.compenvurbsys.2019.101456.
- [13] Van den Hove A, Verwaeren J, Van den Bossche J, Theunis J, De Baets B. Development of a land use regression model for black carbon using mobile monitoring data and its application to pollution-avoiding routing. Environ Res 2020;183:108619. https://doi.org/10.1016/j.envres.2019.108619.
- [14] Caggiani L, Camporeale R, Ottomanelli M. A real time multi-objective cyclists route choice model for a bike-sharing mobile application. 5th IEEE Int. Conf. Model. Technol. Intell. Transp. Syst. MT-ITS 2017 - Proc., Institute of Electrical and Electronics Engineers Inc.; 2017, p. 645–50. https://doi.org/10.1109/MTITS.2017.8005593.
- [15] García-Díaz JA, Noguera-Arnaldos JÁ, Hernández-Alcaraz ML, Robles-Marín IM, García-Sánchez F, Valencia-García R. AllergyLESS. An intelligent recommender system to reduce exposition time to allergens in smart-cities. In: De La Prieta F, Omatu S, Fernández-Caballero A, editors. Distrib. Comput. Artif. Intell. 15th Int. Conf., vol. 800, Springer Verlag; 2019, p. 61–8.

- https://doi.org/10.1007/978-3-319-94649-8_8.
- [16] Rußig J, Bruns J. Reducing Individual Heat Stress through Path Planning. GI_Forum 2017;1:327–40. https://doi.org/10.1553/giscience2017_01_s327.
- [17] Opach T, Navarra C, Rød JK, Neset T-S. Towards a Route Planner Supporting Pedestrian Navigation in Hazard Exposed Urban Areas. In: Hughes AL, McNeill F, Zobel C, editors. ISCRAM 2020 Conf. Proc. 17th Int. Conf. Inf. Syst. Cris. Response Manag., Blacksburg, VA: Virginia Tech; 2020.
- [18] Mahajan S, Tang Y-S, Wu D-Y, Tsai T-C, Chen L-J. CAR: The Clean Air Routing Algorithm for Path Navigation with Minimal PM2.5 Exposure on the Move. IEEE Access 2019;7:147373–82. https://doi.org/10.1109/ACCESS.2019.2946419.
- [19] Grieve A. Clean Air Walking Routes. Monitoring Report for Cross River Partnership. London: 2017.
- [20] Hasenfratz D, Arn T, De Concini I, Saukh O, Thiele L. Health-optimal routing in urban areas. IPSN 2015 Proc. 14th Int. Symp. Inf. Process. Sens. Networks, New York, New York, USA: Association for Computing Machinery, Inc; 2015, p. 398–9. https://doi.org/10.1145/2737095.2737135.
- [21] Brauer M, Nunes M, Winters M, Su J. An Interactive Route Planner Incorporating Air Pollution and Cycling Determinants to Facilitate and Promote Cycling in Metro Vancouver, Canada. Epidemiology 2011;22:S78. https://doi.org/10.1097/01.ede.0000391902.48815.e4.
- [22] Su JG, Winters M, Nunes M, Brauer M. Designing a route planner to facilitate and promote cycling in Metro Vancouver, Canada. Transp Res Part A Policy Pract 2010;44:495–505. https://doi.org/10.1016/j.tra.2010.03.015.
- [23] Hatzopoulou M, Weichenthal S, Barreau G, Goldberg M, Farrell W, Crouse D, et al. A web-based route planning tool to reduce cyclists' exposures to traffic pollution: A case study in Montreal, Canada. Environ Res 2013;123:58–61. https://doi.org/10.1016/j.envres.2013.03.004.
- [24] Nurminen A, Malhi A, Johansson L, Framling K. A Clean Air Journey Planner for pedestrians using high resolution near real time air quality data. IEEE Xplore, Institute of Electrical and Electronics Engineers (IEEE); 2020, p. 44–51. https://doi.org/10.1109/ie49459.2020.9155068.
- [25] Karppinen A, Johansson L. Fusion of air quality information: Evaluation of the Enfuser-methdology in Finland and a case study in China. In: Mensink C, Kallos G, editors. Air Pollut. Model. its Appl. XXV. ITM 2016. Springer Proc. Complex., Chem: Springer; 2018, p. 213–8. https://doi.org/10.1007/978-3-319-57645-9_34.
- [26] Sito OY. Helsingin kaupungin meluselvitys 2017. Kaupunkiym. Helsinki: Helsingin kaupunki; 2017.
- [27] Karppinen A, Johansson L. Fusion of Air Quality Information: Evaluation of the Enfuser-Methodology in Finland and a Case Study in China. In: Mensink C, Kallos G, editors. Air Pollut. Model. its Appl. XXV. ITM 2016. Springer Proc. Complex., Cham: Springer; 2018, p. 213–8. https://doi.org/10.1007/978-3-319-57645-9_34.
- [28] Johansson L, Poom A, Niemi J, Timonen H, Karppinen A. On the utilization of real-time activity and air quality sensor data in a local-scale operative dispersion model in Helsinki. Proc. 38th Int. Tech. Meet. Air Pollut. Model. its Appl. 18-22 Oct. 2021, Barcelona, 2021.
- [29] Toikka A, Willberg E, Mäkinen V, Toivonen T, Oksanen J. The green view dataset for the capital of Finland, Helsinki. Data Br 2020;30:105601. https://doi.org/10.1016/j.dib.2020.105601.