

Healthy Route Generation and Recommendation

LAZAR PENDOV¹, ZHANLIN JI^{2,3}, IVAN GANCHEV^{1,3,4,*}

¹Department of Computer Systems,
University of Plovdiv “Paisii Hilendarski”,
236 Bulgaria Blvd., Plovdiv 4027,
BULGARIA

²College of Mathematics and Computer Science,
Zhejiang A&F University,
Hangzhou, Zhejiang 311300,
CHINA

³Telecommunications Research Center (TRC),
University of Limerick,
Limerick V94 T9PX,
IRELAND

⁴Institute of Mathematics and Informatics,
Bulgarian Academy of Sciences (IMI–BAS),
Sofia 1040,
BULGARIA

**Corresponding Author*

Abstract: - This paper presents the utilization of a developed pilot wireless-based Air Quality Index (AQI) monitoring system, reporting live geo-grid resolved air quality data, for the purposes of healthy route generation and recommendation to users. The generated routes are visualized on a map and recommended to users through a specially developed web-based application, as part of the client tier of the supporting IoT platform EMULSION. A distributed computing architecture is utilized for the generation of healthy (more precisely, ‘least air pollution exposure’) routes, performed in near real-time using the dynamic Dijkstra algorithm, based on the interpolated AQI values. In addition, the fastest and shortest routes for each journey, requested by a user, are generated as well. The importance of the presented work lies within the practical applicability of the proposed method for healthy route generation, either as a stand-alone version of the software application developed for the purpose or integrated into the existing popular navigation systems and applications alike.

Key-Words: - Internet of Things (IoT); IoT platform; EMULSION; Air Quality Index (AQI); healthy route; route generation; recommendation to users.

Tgegkxgf <O ctej "3: ."42460Tgxlugf <Ugr vgo dgt"37."42460Ceeegr vgf <P qxgo dgt"38."42460Rwdrkuj gf <F gego dgt"42."42460"

1 Introduction

The existing web-based route planners and real-time navigation systems, accessed by fixed or mobile personal devices, allow planning and adjusting journeys by providing comfort and a sense of safety to users, [1]. By supplying dynamic and integrated technological support tools to users, along with interactive planning and navigation features for various travel modes, these systems mainly find the

shortest, fastest, or cheapest traveling routes. However, people (especially urban residents) have begun to pay more attention to their quality of life (QoL), by considering environmental factors affecting their health, such as air quality, and this has become their new focus when traveling, [2]. Travel schemes with relatively low pollutant exposure can not only improve human health but can also benefit social stability and sustained progress. In contrast, path-based long-distance

outdoor activities, surrounded by poor air quality, have a negative effect on human health, especially when cycling, running, jogging, or walking, [3]. Therefore, an alternative approach should be taken in route planning, including health-related optimization criteria, based on the concept of “healthy route” (a.k.a. “green route”, “clean route”) and related concepts of “safe route” and “sustainable route”, [1].

A common indicator, used in many countries to measure air pollution, is the Air Quality Index (AQI), which was developed by the United States Environmental Protection Agency (US EPA), based on the following six major pollutants: ground-level ozone (O₃), particulate matter (PM_{2.5} and PM₁₀), carbon monoxide (CO), sulfur dioxide (SO₂), and nitrogen dioxide (NO₂). The US EPA AQI values run from 0 to 500, divided into six levels of concern, as shown in Table 1, [4]. The higher the AQI value, the greater the level of air pollution and the greater the health concern. The green level (AQI ≤ 50) represents satisfactory air quality, whereby air pollution poses little or no risk to humans. At the yellow level (51 ≤ AQI ≤ 100), the air quality is acceptable, but there may be a risk for some people, particularly those who are unusually sensitive to air pollution. At the orange level (101 ≤ AQI ≤ 150), sensitive groups (people with lung diseases, older people, and children) may experience air-quality-related health problems, whereas the general public is less likely to be affected. At the red level (151 ≤ AQI ≤ 200), part of the general public may experience health effects, whereas sensitive groups may experience more serious health effects. The purple level (201 ≤ AQI ≤ 300), signifies a health alert, whereby the risk of health effects is increased for everyone, and most people may experience increasingly severe adverse health effects. The maroon level (AQI ≥ 301) represents hazardous air quality and serves as a health warning of emergency conditions, so everyone is more likely to be affected.

The health-driven measurement of air quality with reasonable geo-grid resolution is in a growing demand across the world. As the current geo-grid resolved AQI is informative of environmental conditions related to personal health, in many countries, especially in urban areas, there are people needing to be able to daily check the current AQI value before going to work or doing outdoor activities. This can be integrated into mobile apps and such. To enable millions of simultaneous AQI requests, a server-side AQI monitoring and publishing system is required, operating with high throughput and high availability. It is easy to

visualize such a system, established on a corresponding Internet of Things (IoT) platform, e.g., functioning as an integral part of a smart city. Given the sizeable deployment and maintenance expense involved, the general public service thrust is to build a low-resolution AQI geo-grid network, e.g., with AQI monitoring points a kilometer apart. Figure 1 shows a sample low-resolution AQI network, established in Hebei Province (China) with just 62 AQI monitoring stations. The Chinese government’s national requirement would be for around 10,735 AQI monitoring stations deployed in such an area of 10,735.78 km². Hence there is a good motivation and market for the development of low-cost sensor-based AQI monitoring stations.

Table 1. The AQI levels of concern

AQI range	Level of concern	Color
0–50	good	green
51–100	moderate	yellow
101–150	unhealthy for sensitive groups	orange
151–200	unhealthy	red
201–300	very unhealthy	purple
301–500	hazardous	maroon



Fig. 1: The utilized low-resolution AQI network

The AQI monitoring stations in the established network function as part of the developed pilot AQI monitoring system, operating on top of the IoT platform EMULSION [5], which was successfully implemented and tested. EMULSION is a horizontal IoT platform of a combined type (hardware and software), built with low-cost electronics and open-source software, and consisting of seven tiers. In the sensor & actuator tier, different types of sensors, environment monitoring stations, location trackers, etc., operate to capture the changes occurring in the

physical world and send the corresponding information to the cloud tier through data/remote transfer units (D/RTUs) and smart communication gateways, through different wireless access networks. After analyzing the data sent by the sensor & actuator tier, the cloud tier makes appropriate decisions, generates suitable recommendations for users, and sends the necessary configuration information and/or commands to the controllers, actuators, guards, etc., located in the sensor & actuator tier, for enforcing the required actions needed for the realization of the imposed changes in the physical world.

Each AQI monitoring station includes a variety of pollution sensors connected with an ultra-low-power geo-grid identified D/RTU. Sensors communicate periodically (usually every 5 minutes) with the corresponding D/RTU. Each D/RTU sends the collected air-quality data, according to the scheduling algorithm used, to the cloud tier of EMULSION, which employs a distributed Redis database, a Hadoop cluster [6], and a web-based Geographic Information System (GIS) [7]. The cloud tier provides scaled GUI services for the mobile and desktop client applications' requests.

This paper demonstrates the use of the developed pilot AQI monitoring system for the generation of healthy routes for outdoor activity planning by users. The difference between the presented work with those published in the literature is that most of the published research suggests a specific application that does some specific things. In contrast, we offer here a more generic architecture that is technology-independent (i.e., w.r.t. hardware, operating system, programming language) and distributed in nature (i.e., hosted on multiple machines), which provides an opportunity for extending it almost without a limit. A clear explanation of the healthy route calculation is provided in the paper, along with a concrete implementation in the form of a web-based application, demonstrating the way it works in reality. The usefulness of the presented study relates to the possible integration of the proposed method for healthy route generation into the existing navigation systems and applications alike.

2 Related Work

In general, there are two primary types of methods for route cost calculation, [3], i.e., using: (1) *static* cost of paths serving as an input to the standard Dijkstra algorithm [8]; and (2) *dynamic* cost of paths, varying over space and time, such as the AQI value and the travel time which depends on the path

infrastructure's condition and traffic flow. As more advanced, multiple methods of the second type have been proposed, e.g., based on adaptive decision rules [9], genetic algorithms [10], [11], [12], probabilistic models [13], uncertainty [14], etc.

Previous studies on "healthy route" generation can be divided into two main groups [2], i.e., using: (1) monitoring stations to measure pollutant exposure on various types of roads, followed by classification of roads as healthy or unhealthy, based on the exposure levels; and (2) pollution distribution data obtained by different means, e.g., by a land use regression (LUR) [15], [16], an operational street pollution model (OSPM) [17], an interpolation method, [3], [18], etc., as an input to the (dynamic) Dijkstra algorithm for generating healthy routes [19], using different indicators (traffic volume, AQI, potential pollutant dose taken, etc.) as road network weights. If taking full advantage of modern pollutant retrieval technologies, the trustworthiness of the generated healthy routes could be significantly increased. For this, [2] proposes a short-distance healthy route planning approach, utilizing fine spatial resolution images, and meteorological and socioeconomic data to retrieve the spatial distribution of PM_{2.5} concentration in hourly intervals via a back-propagation neural network. The effectiveness of the approach is verified by comparing the PM_{2.5} potential dose reduction rate between the generated healthy route and the shortest route, reaching up to 20% reduction in some cases. As an important factor affecting the AQI values, PM_{2.5} concentration can be used also to predict the AQI, [3].

By utilizing an interpolation method from the second group, the current paper is focused on the 'least air pollution exposure' aspect of the "healthy route", computed in near real-time by applying the dynamic Dijkstra algorithm, whereby the potential exposure rate is calculated based on the AQI values, whereby the desired values of the main air pollutants (PM_{2.5}, PM₁₀, O₃, CO, SO₂, NO₂) serve as upper boundaries for reducing the number of possible routes.

3 Utilized Data

3.1 Permanent Data

This type of data rarely or never changes. The open data of the OpenStreetMap [20] are used as map data for the study area, shown in Figure 1. This is a rectangular area enclosed between the GPS coordinates (36.846000/114.345000) and (37.655772/115.684369) with a total coverage of

10735.78 km², which includes most of Hebei Province (China). We initially focused on the roads. In the data provided by the OpenStreetMap, the roads have their own type (motorway, primary road, etc.) and are described by an ordered list of nodes. The number of nodes in the study area is equal to 246,686. Each node has its own unique number and is described with GPS coordinates. Distances between nodes, and between nodes and the AQI monitoring stations deployed in the area, can be calculated, based on their GPS coordinates, using the haversine formula, as follows:

$$\begin{aligned}
 a &= \sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\Delta\lambda}{2}\right) \\
 c &= 2 \cdot \text{atan2}\left(\sqrt{a}, \sqrt{1-a}\right) \\
 d &= R \cdot c
 \end{aligned}
 \tag{1}$$

where φ_1 and φ_2 denote the latitudes of the two points (in radians), $\Delta\varphi$ denotes the difference between the latitudes of the two points, $\Delta\lambda$ denotes the difference between the longitudes of the two points (in radians), and R denotes the radius of the Earth.

The minimum, maximum, mean, and mode values of the distances between any two OpenStreetMap nodes in the study area are presented in Table 2.

Table 2. Distances between OpenStreetMap nodes in the study area

Distances (m)			
Min value	Max value	Mean value	Mode value
0.02	2633.79	96.14	11.00

We converted the OpenStreetMap node data into a graph, suitable for applying the Dijkstra algorithm. Each edge in the graph represents a connection between two nodes along a particular road existing in the area. The indicators used are the edge length and road type.

3.2 Refreshable Data

These data come from the AQI monitoring system, operating on top of the IoT platform EMULSION, which provides data on the main air pollutants (PM2.5, PM10, O₃, CO, SO₂, NO₂) and the AQI. There are 62 AQI monitoring stations in the study area (c.f., Figure 1). These data are updated on every hour.

3.3 Computable Data

To implement the Dijkstra algorithm, each OpenStreetMap node is assigned with a particular air quality value *AirQ*, obtained by interpolation, using the air quality values of the three nearest AQI monitoring stations, as follows:

$$\text{AirQ} = \frac{\frac{\text{AirQ}_1}{d_1} + \frac{\text{AirQ}_2}{d_2} + \frac{\text{AirQ}_3}{d_3}}{\frac{1}{d_1} + \frac{1}{d_2} + \frac{1}{d_3}}
 \tag{2}$$

where *AirQ*₁, *AirQ*₂, and *AirQ*₃ denote the air quality values of the three nearest AQI monitoring stations, and *d*₁, *d*₂, and *d*₃ denote the distance from the OpenStreetMap node to each of these three stations, respectively.

Table 3 presents statistics of the minimum, maximum, mean, and mode values of the distances between the OpenStreetMap nodes in the study area and the corresponding three nearest AQI monitoring stations (Station 1 is the nearest, and Station 3 is the farthest).

Table 3. Distances between OpenStreetMap nodes and the three nearest AQI monitoring stations in the study area

	Distances (m)			
	Min value	Max value	Mean value	Mode value
Station 1	9	44251	8585	6847
Station 2	507	45406	10269	8756
Station 3	939	47254	11784	10448

As can be seen from Table 3, there are worryingly large distances between some OpenStreetMap nodes and the nearest AQI monitoring stations, but these gaps can be compensated by deploying more stations in the area in the future.

4 Distributed Computing Architecture

The elaborated computing architecture is divided into four subsystems, as presented in Figure 2. Each subsystem is platform- and program-independent and can be implemented using different programming languages, operating systems, and hardware. Communication between subsystems adheres to open standard protocols. A major advantage of this architecture is the use of multiple computing servers, which makes it easily scalable depending on the expected number of users and workload.

4.1 Data Collection Subsystem

This subsystem consists of multiple AQI monitoring stations and an information server to which these stations periodically send their collected data. This subsystem is apart from the other three subsystems and is managed separately. The information server provides the aggregated collected data through an API in a JSON format.

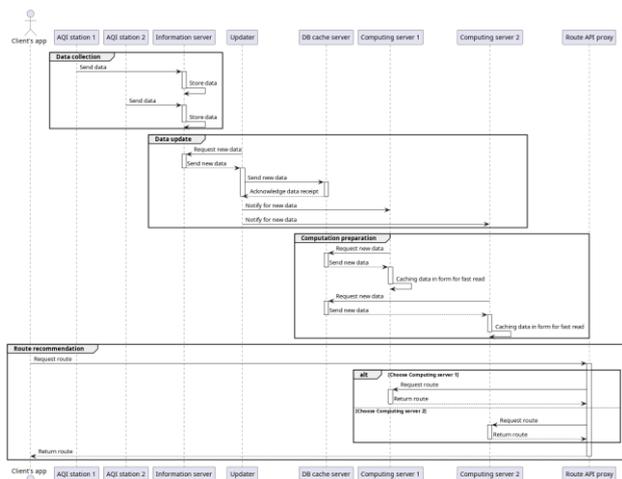


Fig. 2: The operation of the developed distributed architecture, utilized for route computation

4.2 Data Update Subsystem

This subsystem consists of an update script and a database server. Periodically, the script is activated, making a request to receive updates from the information server of the first subsystem. The received data are processed in an appropriate form and sent to the database server for caching. The update script then sends an update notification to each computing server, operating as part of the third subsystem, described next.

4.3 Computing Subsystem

When a computing server receives an update notification, it makes a request to the caching database server to receive the necessary new data, associated with its activity. Once the data are received, these are arranged in a form optimized for fast reading, preferably by caching in RAM.

4.4 Route Generation Subsystem

Through a corresponding client application, the user makes a request to the API of the proxy server to calculate a route (of a particular type) between two points on the map, according to her/his requirements. The proxy server selects one of the computing servers suitable for the request and forwards the request to it. After performing the required calculations, the computing server returns

the result to the proxy server, which in turn forwards it to the client application, which renders the result to the user.

The most recent (updated) data are used for route calculations. When the time to pass a route exceeds the updating time of data coming from the AQI monitoring stations, new requests can be made during the journey in order to regenerate the route with greater truthfulness. It is also possible to apply predictive algorithms based on historical data (such algorithms have not been implemented yet).

5 Implementation

The computing architecture proposed in the previous section is independent of the programming language, operating system, and hardware, but in order to have some implementation, concrete technologies must be chosen. In the solution presented here, the Linux Debian (trixie) operating system is used for all servers (except for the first subsystem) and PHP is used as a programming language (for console and web execution).

5.1 Data Collection Subsystem

This subsystem was developed before starting the work presented here; thus, it is not discussed further.

5.1.1 Data Update Subsystem

The update script is executed at a certain time interval (currently, each hour) via *cron*. The script makes an HTTP GET request to the information server of the first subsystem and receives a response in JSON format. The received data are processed and sent to the MySQL server for caching. Notification is then made via parallel GET requests to the computing servers. Security measures are taken by applying a unique key for each computing server so that this call cannot be made by an unauthorized participant.

5.1.2 Computing Subsystem

A major problem of the current implementation relates to the natural way PHP scripts work. When a PHP script is called, a new instance of it is created, the required resources are allocated, the necessary operations are performed, and a response is returned, followed by the release of all allocated resources. This continuous allocating and releasing of resources for each call, involving huge data structures, is extremely inefficient and cumbersome. To solve this problem, the data are cached in two stages:

- 1) **Stage 1:** Structuring the data in a jagged array format in a PHP file, suitable for importing via *include()* into the computation script;
- 2) **Stage 2:** Transferring the obtained data structure to RAM via the APCu extension for Apache. Thus, once the structure is allocated in the Apache server's RAM, it is accessible by all instances of the computing script without the need for multiple resource allocations and releases.

When the Apache server is restarted, the cache in RAM is lost, but the structure located in a file (created in the first stage) is usable and allows it to be cached in RAM again.

5.1.3 Route Generation Subsystem

This subsystem includes a client application, an API proxy server, and multiple computing servers, operating as follows:

- The developed *client application* is accessible through the official website of the EMULSION project [21], which is hosted on an Odroid-HC2 web server with a 32-bit ARM CPU: Samsung Exynos5422 ARM@ Cortex™-A15 Quad 2.0GHz/Cortex™-A7 Quad 1.4GHz. The web application sends requests to the API proxy using the HTTP GET commands and receives responses in a JSON format.
- The *API proxy* is hosted on the same web server and is implemented in PHP. When requesting a route computation, it chooses a random computation server, by taking into account the server capabilities (more powerful servers are called more often). The received response is forwarded to the client application. Necessary security measures are taken by applying a unique secret key for each computing server to prevent unauthorized requests.
- The *computing servers* include:
 - ✓ A PC with 2xCPU: Intel(R) Xeon(R) Gold 6134 CPU @ 3.20GHz (cores: 16, threads: 32). Only ¼ of its computing capabilities are reserved for the route computation task (threads: 8), whereas the remaining capabilities are used for performing other activities, as a demonstration of the ability to use a shared server.
 - ✓ A PC with Intel(R) Xeon(R) CPU E3-1220 v3 @ 3.10GHz (cores/threads: 4).
 - ✓ 15 single-board computers Odroid-N2+ with 64-bit ARM CPU: Amlogic

S922X, Quad Cortex-A73 2.4GHz and Dual Cortex-A53 2GHz (cores/threads: 4+2).

When the compute script is called, it looks for an available RAM cache. If such a cache is not available, the data are loaded from the cached file (Stage 1) and a new RAM cache is created.

During the trial experiments conducted with single-board servers Odroid MC1 (with 4 servers per unit) with 32-bit CPU (per server): Samsung Exynos5422 ARM® Cortex™-A15 Quad 2.0GHz/Cortex™-A7 Quad 1.4GHz (cores/threads per server: 8), the route computations often exceeded two minutes. Therefore, this type of single-board computer was excluded from the final version of this subsystem. However, with another software implementation of the computing process, it would probably be possible to utilize such inexpensive server hardware as well.

The used ARM computing servers (without the two non-ARM servers) are depicted in Figure 3.



Fig. 3: The utilized Odroid single-board computers (15x Odroid-N2+ and 5x Odroid-MC1)

6 Client Application

6.1 Setting up Maximum Values of Air Quality Parameters

The user can specify a preferred maximum value for each specific air pollutant (PM2.5, PM10, O3, CO, SO₂, NO₂) and for the AQI, which must be not exceeded in any case along the route.

6.2 Travel Modes

Different travel modes could be used by users, each with its own exposure to polluted air and speed of movement, which results in different inhaled doses. In [22], [23], the rate of exposure to a particular air pollutant on a route is calculated as follows:

$$\text{Exposure rate} = \text{MVR} \times \text{PC} \times \text{time} \quad (3)$$

where MVR (L/min) denotes the minute ventilatory rate (MVR), PC ($\mu g/m^3$) denotes the pollutant concentration, and $time$ (min) denotes the travel time on the route.

Based on (3), we calculate the rate of exposure to polluted air on a route, based on the AQI value, as follows:

$$Exposure\ rate = C_{mode} \times \sum_i (AQI_i \times time_i) \quad (4)$$

where AQI_i denotes the AQI value of the i^{th} section between two consecutive/neighborhood OpenStreetMap nodes on the route (calculated as the average value of the interpolated AQI values of the two nodes), $time_i$ denotes the travel time through that section (calculated by dividing the section length by the average travel speed on that section), and C_{mode} denotes the corresponding coefficient of the travel mode, calculated by comparing the MVRs for different travel modes, using their average values reported by ChatGPT. Coefficient C_{mode} takes values between 0 and 1, whereby a value of 1 means maximum exposure to air pollution, and a value closer to 0 implies minimum exposure. By comparing the MVRs of the considered travel modes, shown in Table 4, it can be seen that cycling leads to maximum exposure, so its coefficient C_{mode} is set to 1. The C_{mode} values of other travel modes are calculated by dividing their MVRs to the MVR of the cycling mode. More accurate values of coefficient C_{mode} can be obtained with more precise empirical studies investigated and taken into account in the future. If a user decides to wear a protection mask when traveling, this could be easily reflected in (4) by applying a corresponding protection coefficient $C_{protect}$.

In each travel mode, for each road type, there is a default value for the average speed, which can be adjusted in the application itself depending on the legal regulations for road traffic (e.g., according to maximum speed limits set), and can be further adjusted by the user depending on her/his abilities and habits related to the specific travel mode used. Some travel modes may be prohibited on certain road types (e.g., a car cannot be used within a pedestrian zone).

6.3 Routes' Start and End Points

In the current implementation, the selection of the start and end points of a route is done by the user by clicking on the map with the left (for the start point) and right (for the end point) mouse buttons. The specified coordinates will hardly coincide exactly with the coordinates of any OpenStreetMap node in

the dataset. Therefore, the closest node existing on the map is used for the route generation.

Table 4. Different travel modes and their corresponding minute ventilatory rates (MVRs) and C_{mode} coefficients

Travel mode	MVR (L/min)	C_{mode}
Driving a car (with closed windows)	7	0.200
Driving a car (with open windows)	8	0.229
Motorcycling	10	0.286
Cycling	35	1.000
Walking	15	0.429

6.4 Route Types

In addition to the 'least air pollution exposure' route, the developed web-based application can generate and recommend also the fastest route and shortest route for traveling. All route computations are performed by means of the Dijkstra algorithm, using, respectively, the 'exposure rate', defined in (4), as a cost for the 'least air pollution exposure' route, the 'time' for the fastest route, and the 'length' for the shortest route.

6.5 Results

The final result for each generated route type includes a route visualization, presented as a set of ordered GPS coordinates used for route visualization on the map, as shown in Figure 4(a) (each route type is drawn in a different color), and information about air quality parameter values on each route type, as presented in Figure 4(b). More specifically, the developed application generates three different types of routes for traveling between any two points in the study area. In the example presented in Figure 4, these are (i) the *healthiest* route with a minimum exposure rate of 1120.45 AQI.minutes, (ii) the *fastest* route with a minimum travel duration of 1:29 h, and (iii) the *shortest* route with a minimum travel distance of 133.069 km. Additional data include the GPS coordinates of the start and end points (respectively, (37.53357480/114.51482110) and (36.99950980/115.51223270) in this example), the exposure rate, and the name of the server used for the computation of each route type.

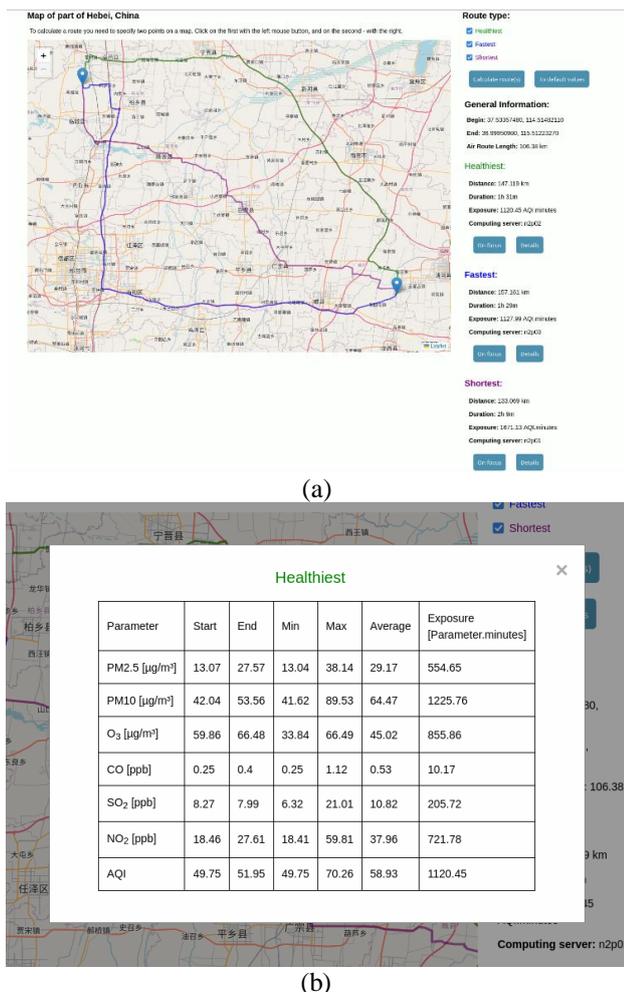


Fig. 4: Sample route generation: (a) different route types (i.e., healthiest, fastest, and shortest route) between two points in the study area; (b) the air quality parameter values on the healthiest route, generated in (a)

7 Conclusion

This paper has presented an elaborated distributed computing architecture for the generation of the healthiest routes (more precisely, the ‘least air pollution exposure’ routes), performed in near real-time by means of the dynamic Dijkstra algorithm, based on the air quality index (AQI). In addition, the fastest and shortest routes are generated as well. The generated routes are visualized on a map and recommended to users through a specially developed web-based application, as part of the client tier of the supporting IoT platform EMULSION. The generated healthiest routes, in particular, allow the users to avoid air-polluted areas posing particular health risks to them.

Future work will be focused on the development of suitable models to predict the future AQI, based on historical data, current meteorological data, and

weather forecasts, for the purposes of smart proactive “healthy routes” planning for outdoor activities of users. The routes will be initially preplanned, with the possibility to be dynamically changed later, if needed, depending on the current environmental conditions. The incorporation of such health-related criteria into existing navigation systems and applications for route generation and recommendation is envisaged as an important functionality extension of the latter, [5].

References:

- [1] Ribeiro, P. and J.F.G. Mendes, Healthy routes for active modes in school journeys. *International Journal of Sustainable Development and Planning*, 2013. 8(4): pp. 591-602. DOI: 10.2495/SDP-V8-N4-591-602.
- [2] Gao, L.-N., F. Tao, P.-L. Ma, C.-Y. Wang, W. Kong, W.-K. Chen, and T. Zhou, A short-distance healthy route planning approach. *Journal of Transport & Health*, 2022. 24: pp. 1-14. DOI: 10.1016/j.jth.2021.101314.
- [3] Zou, Z., T. Cai, and K. Cao, An urban big data-based air quality index prediction: A case study of routes planning for outdoor activities in Beijing. *Environment and Planning B: Urban Analytics and City Science*, 2019. 47(6): pp. 948-963. DOI: 10.1177/2399808319862292.
- [4] Air Quality Index (AQI) Basics, [Online]. <https://www.airnow.gov/aqi/aqi-basics/> (Accessed Date: September 20, 2024).
- [5] Ganchev, I., Z. Ji, and M. O'Droma, Horizontal IoT Platform EMULSION. *Electronics*, 2023. 12(8): pp. 1-21. DOI: 10.3390/electronics12081864.
- [6] Shafer, J., S. Rixner, and A.L. Cox. The Hadoop distributed filesystem: Balancing portability and performance. *Proc. of 2010 IEEE International Symposium on Performance Analysis of Systems & Software (ISPASS)*, pp. 122-133. White Plains, NY, USA, 2010. DOI: 10.1109/ISPASS.2010.5452045.
- [7] Goodchild, M.F., Geographic information system, in: Liu, L., Özsu, M.T. (eds) *Encyclopedia of Database Systems*. Springer, Boston, MA, 2009. pp. 1231-1236. DOI: 10.1007/978-0-387-39940-9_178.
- [8] Dijkstra, E.W., A note on two problems in connexion with graphs. *Numerische Mathematik*, 1959. 1(1): pp. 269-271. DOI: 10.1007/BF01386390.

- [9] Hall, R.W., The Fastest Path through a Network with Random Time-Dependent Travel-Times. *Transportation Science*, 1986. 20(3): pp. 182-188.
- [10] Pellazar, M.B. Vehicle route planning with constraints using genetic algorithms. *Proc. of National Aerospace and Electronics Conference (NAECON'94)*, vol. 1, pp. 111-118. Dayton, OH, USA, 1994. DOI: 10.1109/naecon.1994.333010.
- [11] Pattnaik, S.B., S. Mohan, and V.M. Tom, Urban bus transit route network design using genetic algorithm. *Journal of Transportation Engineering*, 1998. 124(4): pp. 368-375. DOI: 10.1061/(ASCE)0733-947X(1998)124:4(368).
- [12] Chien, S., Z.W. Yan, and E. Hou, Genetic algorithm approach for transit route planning and design. *Journal of Transportation Engineering*, 2001. 127(3): pp. 200-207. DOI: 10.1061/(ASCE)0733-947X(2001)127:3(200).
- [13] Boyan, J. and M. Mitzenmacher, Improved results for route planning in stochastic transportation. *Proc. of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 895-902. Washington, D.C., USA, 2001. DOI: 10.1145/365411.365803.
- [14] Nikolova, E., M. Brand, and D.R. Karger, Optimal route planning under uncertainty. *Proc. of the 16th International Conference on Automated Planning and Scheduling*, pp. 131-140. Cumbria, UK, 2006.
- [15] Hatzopoulou, M., S. Weichenthal, G. Barreau, M. Goldberg, W. Farrell, D. Crouse, and N. Ross, A web-based route planning tool to reduce cyclists' exposures to traffic pollution: a case study in Montreal, Canada. *Environmental Research*, 2013. 123: pp. 58-61. DOI: 10.1016/j.envres.2013.03.004.
- [16] Zou, B., S. Li, Z. Zheng, B.F. Zhan, Z. Yang, and N. Wan, Healthier routes planning: A new method and online implementation for minimizing air pollution exposure risk. *Computers, Environment and Urban Systems*, 2020. 80: pp. 1-11. DOI: 10.1016/j.compenvurbsys.2019.101456.
- [17] Hertel, O., M. Hvidberg, M. Ketzel, L. Storm, and L. Stausgaard, A proper choice of route significantly reduces air pollution exposure - a study on bicycle and bus trips in urban streets. *Science of The Total Environment*, 2008. 389(1): pp. 58-70. DOI: 10.1016/j.scitotenv.2007.08.058.
- [18] Zahmatkesh, H., M. Saber, and M. Malekpour, A New Method for Urban Travel Rout Planning Based on Air Pollution Sensor Data. *Current World Environment*, 2015. 10(Special-Issue1): pp. 699-704. DOI: 10.12944/CWE.10.Special-Issue1.83.
- [19] Mahajan, S., Y.-S. Tang, D.-Y. Wu, T.-C. Tsai and L.-J. Chen, CAR: The Clean Air Routing Algorithm for Path Navigation With Minimal PM2.5 Exposure on the Move. *IEEE Access*, 2019. 7: pp. 147373-147382. DOI: 10.1109/ACCESS.2019.2946419.
- [20] Arsanjani, J.J., A. Zipf, P. Mooney, and M. Helbich, OpenStreetMap in GIScience: Experiences, Research, and Applications, in *Lecture Notes in Geoinformation and Cartography*. 2015, Springer International Publishing: Imprint: Springer, Cham.
- [21] EMULSION project, [Online]. <http://emulsion.science> (Accessed Date: September 20, 2024).
- [22] Li, H.-C., P.-T. Chiueh, S.-P. Liu, and Y.-Y. Huang, Assessment of different route choice on commuters' exposure to air pollution in Taipei, Taiwan. *Environment Science Pollution Research*, 2017. 24(3): pp. 3163-3171. DOI: 10.1007/s11356-016-8000-7.
- [23] Ramos, C.A., H.T. Wolterbeek, and S.M. Almeida, Air pollutant exposure and inhaled dose during urban commuting: a comparison between cycling and motorized modes. *Air Quality Atmosphere and Health*, 2016. 9(8): pp. 867-879. DOI: 10.1007/s11869-015-0389-5.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed to the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

This publication has emanated from joint research conducted with the financial support of the Bulgarian National Science Fund (BNSF) under the Grant No. KP-06-IP-CHINA/1 (КП-06-ИП-КИТАЙ/1) and the National Key Research and Development Program of China under Grant No. 2017YFE0135700.

Conflict of Interest

The authors have no conflicts of interest to declare.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0

https://creativecommons.org/licenses/by/4.0/deed.en_US