Developing of AI assistant for route optimization intelligent system

Victoria Vysotska[1,†], Kirill Smelyakov[2,*,†], Anastasiya Chupryna[2,†], Maksym Derenskyi[2,*,†], Vadim Repikhov[2,†], and Mykyta Hvozdiev[2,†]

[1] *Kharkiv National University of Internal Affairs, L. Landau Avenue 27 61080 Kharkiv, Ukraine*
[2] *Kharkiv National University of Radio Electronics, 14 Nauky Ave., Kharkiv, 61166, Ukraine*

Abstract

This paper is devoted to the research and development of an intelligent assistant for constructing optimal routes and improving the final generation result. Rules and methods were developed aiming to find generation artifacts and eliminate them to improve the result produced by Ant Colony Optimization. These tasks were solved: analysis of modern approaches and solutions for calculating optimal routes, software implementation of the optimal route search algorithm, setting up rules for finding artifacts, software implementation of post-processing, application of LLM models to improve the result, conducting experiments, and comparing results. The methodology is based on analyzing the process of route generating and highlighting artifacts that arise because of generation inaccuracies and methods for correcting them. The following results were achieved: the software was developed that allows post-processing of generation results and compared with original generation results. The software implementation showed high efficiency and speed of route generation result correction, allowing the final route to be improved by 3–5%. At the same time, the use of AI assistant significantly improves the performance of daily operations when working with the information system, making interaction with system more convenient. The results obtained are satisfactory, but the direction of modifying the ACO algorithm and applying parallel software implementation looks promising.

Keywords

paper template, paper formatting, CEUR-WS

1. Introduction

Nowadays, solving transport logistics problems and finding optimal routes is widespread [1]. Logistics takes significant place in the global economy. Every day, a human faces various variants of logistics problems, from using applications with maps to ordering goods on delivery services. In a broader sense, at the global level, logistics is built into supply chains. With a high level of globalization, the importance of solving the long supply chains issue is becoming more actual, and their optimization is an important task from an economic point of view.

Global logistics costs are constantly growing. It is a big market with big expenses. Thus, in 2023, the logistics market was valued at 5.35 trillion euros, with road transport accounting for the largest share. By 2029, logistics spending is expected to increase to 14.4 trillion euros.

Thus, in China, logistics companies face major challenges and believe that in the context of economic globalization, the key to creating a company's competitive advantage has shifted from the first source of profit (raw materials economy) and the second (increasing labor productivity) to the third source of profit – creating an effective logistics system [2]. One of the important tasks is to choose the optimal route, for example, to transport goods from city to city or deliver to addresses. The shorter the route is chosen and such factors as road surface quality, weather

CIAW-2025: Computational Intelligence Application Workshop, September 26-27, 2025, Lviv, Ukraine

[1*] Corresponding author.
[†] These authors contributed equally.
victoria.a.vysotska@lpnu.ua (V. Vysotska); kyrylo.smelyakov@nure.ua (K. Smelyakov); anastasiya.chupryna@nure.ua (A. Chupryna); maksym.derenskyi@nure.ua (M. Derenskyi); vadym.repikhov@nure.ua (V. Repikhov); mykyta.hvozdev@nure.ua (M. Hvozdiev)
0000-0001-6417-3689 (V. Vysotska); 0000-0001-9938-5489 (K. Smelyakov); 0000-0003-0394-9900 (A. Chupryna); 0000-0003-1539-8018 (M. Derenskyi); 0000-0002-1274-4205 (V. Repikhov); 0000-0003-4022-9777 (M. Hvozdiev)

conditions, etc. are considered, the more logistic costs will be reduced. A big advantage is the automation of solving transport problems using mathematical and software algorithms to find the optimal route. To automate the problem, it is necessary to determine what exactly needs to be automated. This work describes the solution and improvement of the results for the Traveling Salesman Problem (TSP). There are many algorithms for solving such a problem, and they all come down to either a complete enumeration of all possible combinations and direct comparison, or to random enumeration and finding a quasi-optimal option. Tests were carried out with such algorithms as Brute Force, Monte Carlo and Ant Colony Optimization.

The next step of this work is to improve the results of optimal route generation using an intelligent assistant. The improvement of the optimal route generation result is done automatically in the post-processing format, which allows for the route to be analyzed again and artifacts, the so-called "loops", to be removed. This ultimately allows for an additional 3-5% improvement in the result on average for various indicators, depending on the optimization criterion, be it distance, fuel costs or time.

Additionally, the use of an intelligent assistant allows for the implementation of an improved model of user interaction with the system. Nowadays, voice processing and text-to-text technologies have become widespread [3]. In the future, this can be used by large test models to generate a response based on a user request [4]. Thus, the user gets the opportunity to perform normal work with the system using a chat bot or voice commands, which in the long term allows for accelerating and improving the user experience and reducing the amount of user interface required to perform everyday tasks. At the same time, it is necessary to take care of the confidentiality of user data. For this purpose, such data is usually stored in encrypted form and decrypted only using a private key-token and only on the server side.

2. Related works

Finding the optimal route is the most popular problem in logistics. Usually, the term "optimal route" means a route that is the shortest in distance. It is worth adding that the criteria for the optimality of a route can also be time and resources, fuel, which are spent on passing the route. The field of logistics is very promising and popular for researchers in our time. In the work [5], a study of Pareto-optimal shortest paths in stochastic transport networks is conducted, considering the variability of travel time. The authors explore the methodology of finding reliable and efficient routes under conditions of uncertainty of travel time. Finding the shortest route is also necessary in Internet networks for the highest data transfer rate. In the work [6], routing is considered as a means of increasing the performance of computer systems, with the focus on the efficiency of arithmetic calculations in different calculation systems when solving problems on graphs. or time.

The paper [7] investigates the improvement of heuristic intelligent algorithms for route finding in 3D space, with an emphasis on improving the accuracy, search speed and stability of the algorithm. A new intelligent algorithm based on Beetle Antennae Search (BAS) is developed with three additional, non-trivial mechanisms aimed at solving the problems of low efficiency and convergence accuracy. Here, the Ant Colony Optimization algorithm is used to generate an initial route, which serves as a guideline for further optimization. The paper demonstrates how complex of traditional methods with modern heuristics can significantly improve the quality of solutions in navigation problems.

Recent studies also highlight the potential of large language models (LLM) to improve user interaction with intelligent systems. In works [8-9], LLM is explored for applications in user assistance and software version control, showing efficiency in transforming textual queries into structured executable commands. Such methods can be effectively applied in developing of an intelligent assistant within the service. Furthermore, study [10] describes the implementation of LLM architectures for natural language processing tasks. In the paper [6], alternative techniques for solving the shortest path problem are analyzed, while classification-based methods are discussed in [4, 11-12].

The proposed assistant is designed to be capable to identify and resolve route generation artifacts, such as “loops”, thus improving the result of generation. Study [13] presents an approach to process natural language which, presumably, demonstrates the potential to help in optimizing computational processes involved in routing problem solving. Consequently, the developed system should integrate an intelligent assistant with active use of ACO algorithm for initial route generation followed by further improvement via loop detection and correction.

3. Methods and materials

The aim of this work is to implement methods for constructing optimal routes for solving the TSP problem and further application of methods for post-processing the result for greater optimality. Post-processing will be built using both mathematical methods for determining intersections of route sections, "artifacts", "loops" and the use of a trained LLM model for analyzing the entire route and improving its optimality. The idea is to develop a model that will determine the intersections of route sections consisting of any number of segments and having any number of sections with intersections that make the route less optimal.

The optimization model for route colculation result must meet the following requirements:

- the ability to find intersections along the entire length of a route, identifying the beginning and end of an intersection
- low algorithmic complexity, ensuring fast calculations and low resource consumption
- ability to recalculate non-optimal areas
- versatile in software implementation

3.1. TSP solving algorithms

During the research, we analyzed and implemented several algorithms for finding the optimal route to solve the TSP problem, among which we can highlight such as brute force, Monte Carlo and Ant Colony Optimization.

The TSP might be described as following. Let π be an order of placement of points ($\pi \in S_n$), where S_n is the set of all options for placing points of length n. Then the route model will be

$$v_{\pi(1)} \to v_{\pi(2)} \to \dots \to v_{\pi(n)} \to v_{\pi(1)}. \quad (1)$$

The total length of the route will be

$$L(\pi) = \sum_{i=1}^{n-1} d\left(v_{\pi(i)}, v_{\pi(i+1)}\right) + d\left(v_{\pi(n)}, v_{\pi(1)}\right). \quad (2)$$

Brute force algorithm allows finding the shortest path by enumerating all possible combinations but is the least optimal in terms of resource consumption and time. Since the complexity grows exponentially with an increase in the number of points, then, when calculating the shortest path using 8 points, the algorithm shows an acceptable speed of operation, and with 10 points, the calculation time becomes unacceptably long. The model for solving the TSP problem using the Brute Force method is a direct search for such an option, in which the length of the entire route will be minimal

$$L_{min} = \min_{\pi \in S_n} L(\pi). \quad (3)$$

Monte Carlo is a heuristic algorithm that works by randomly searching through the available solutions to a problem. As described in [14], this algorithm is highly dependent on input parameters such as the number of available solutions and the number of iterations to find a solution. For each random permutation $\pi_k = [\pi_k(1), \dots, \pi_k(n)]$ the length of the route will be determined by the formula:

$$L(\pi)=\sum_{i=1}^{n-1} d\left(v_{\pi_k(i)}, v_{\pi_k(i+1)}\right)+d\left(v_{\pi_k(n)}, v_{\pi_k(1)}\right). \tag{4}$$

The optimal approximate solution is described by the following formula

$$L L_{min} \approx \min_{k=1\ldots M} L(\pi_k). \tag{5}$$

Further experiments showed that this algorithm allows processing a larger number of points, but with an increase in the number of available solutions, the accuracy of calculations decreases, and it is necessary to increase the number of iterations, which ultimately leads to greater complexity and resource consumption.

Ant colony optimization algorithm is also heuristic but works differently. It randomly tries available route options from one point to another, but at the same time, the proximity of the next point and the fact that this segment of the path has already been visited before have a strong influence, which allows choosing already known routes with a higher probability.

The ant algorithm, as experiments have shown, turned out to be optimal in the case of calculating the optimal path among many points. Further experiments showed that this algorithm allows you to effectively process 50 points and get a good result when processing 100 points. This class of algorithms has a polynomial complexity, which allows you to use a relatively small number of resources to calculate the path for many points. The algorithm also lends itself well to various modifications and additions. So, in the work [15] the algorithm is used to plan the route of mobile robots based on an improved version of the ant algorithm. The work solves its main shortcomings: slow convergence and tendency to get stuck in local minimums and dead ends. The following options are proposed to solve these shortcomings:

- adaptive heuristic function that depends on the number of iterations;
- adaptive pheromone parameters, which allows dynamic configuration of the amount of pheromone, accelerating convergence and making it less likely to get stuck in local minima;
- integration of the Monte Carlo algorithm, which allows to avoid tortuous trajectories, making the path more direct and smoother.

The essence of the ant algorithm is that over a certain number of iterations, virtual agents – “ants” randomly pass through sections of the route, gradually forming the optimal path. With each pass of the “ant”, a certain trace is left – a “pheromone”, which, when the next “ant” passes, affects the probability of choosing an already familiar section of the route and thereby further distinguishing it from the rest. The algorithm model is described by a set of formulas. The probability estimate of the transition to the next point is expressed by the formula

$$L P_{i,j;k}=\left(\tau_{i,j;k}^{\alpha} \cdot \eta_{i,j;k}^{\beta}\right) / \left(\sum_{m} \tau_{i,m;k}^{\alpha} \cdot \eta_{i,m;k}^{\beta}\right). \tag{6}$$

The right side of the formula is the probability of moving to a certain point where $\eta_{i,j;k}$ – the attractiveness of a road is proportional to the distance, and $\tau_{i,j;k}$ is pheromone amount, left by previous "ants". α, β are coefficients to configure the influence of these parameters during calculations, by default set to 1. The right side of the formula is the sum of all probabilities of transition from the current point to all available next ones. That is, the probability of transition to the next point is inversely proportional to the sum of the probabilities of transition to all available points. The sum of all probabilities is 1.

After calculating the probabilities, a numerical roulette is constructed, where all probabilities of transition to the next point are distributed in the range from 0 to 1 and a number from 0 to 1 is randomly generated, thus choosing a point for transition. The "ant" moves to city j from city i. City i is marked as unavailable. At the current iteration ξ, it is no longer possible to move to it. We assume $i=j$. This continues until all points have been visited.

The amount of pheromone is recalculated after the completion of the inner loop, when all the ants of the group have laid their paths. The addition of pheromone that ant number k makes on the road between cities i and j at iteration number t is expressed by the system

$$\Delta\tau_{i,j;k}^{(t)}=\begin{cases}Q/L_k^{(t)}, if\,(i,j)\in T_k^{(t)},\\ 0, otherwise.\end{cases} \tag{7}$$

After calculating the pheromone additions for each road, a new amount of pheromone is calculated at the next iteration for each road using the following formula

$$\tau_{i,j}^{(t+1)}=(1-p)\cdot\tau_{i,j}^{(t)}+\frac{1}{m}\sum_{k=1}^{m}\Delta\tau_{i,j;k}^{(t)}. \tag{8}$$

where p is the pheromone evaporation coefficient. It is needed to balance the paths and if a certain path is not visited during the iteration, it becomes less attractive.

This parameter can be set to 0, and then the path will maintain a constant level of attractiveness. Thus, for a certain number of iterations, the optimal path is calculated. The high convergence rate of the problem solution exists at the first iterations, at later ones the convergence rate decreases. The path calculated at the last iteration is the result of the algorithm. Visualization is shown in Figure 1.

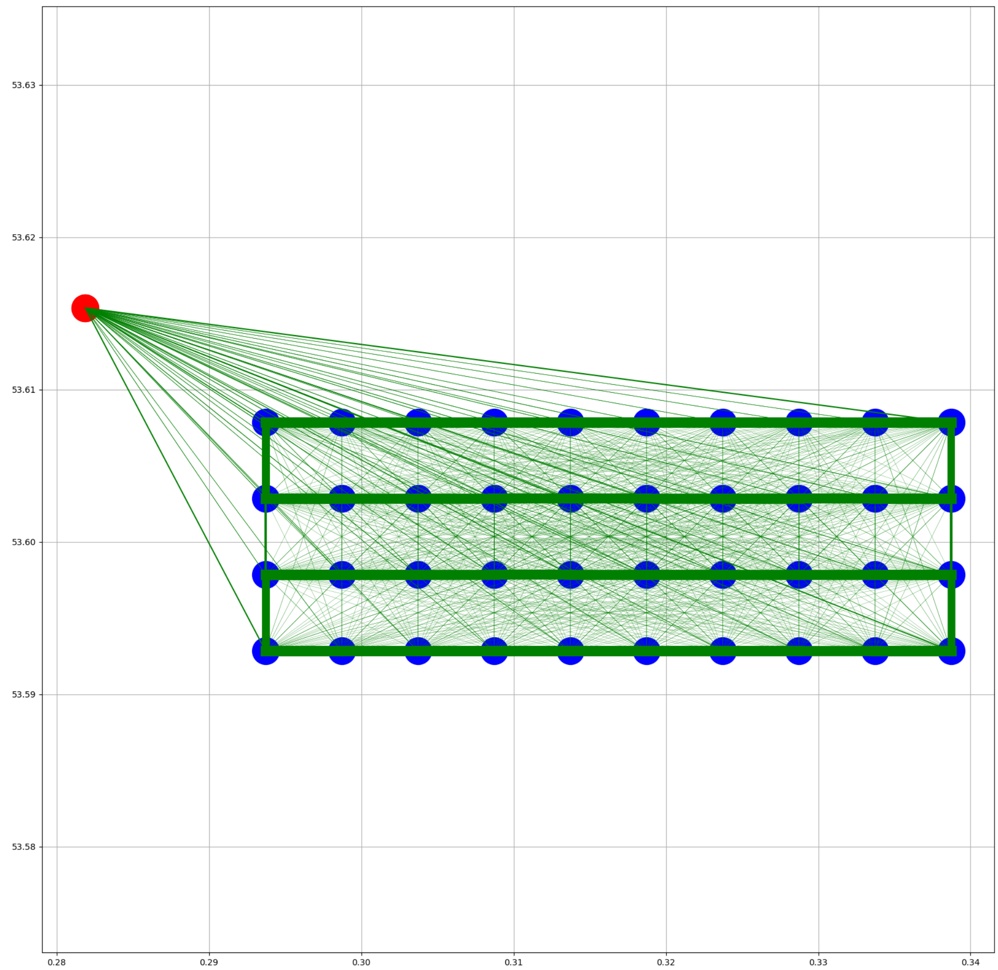

Figure 1: Visualization of ant colony optimization.

After calculating the optimal route using the Monte Carlo algorithm or AOC, intersections of segments - "loops" - may occur, which negatively affect the result. Their solution allows for a more optimized route. The general scheme for finding the optimal route is shown in Figure 2.

Figure 2: General pipeline of optimal route generation.

The application implements the logic in which the user sends a request to find the optimal route, providing a set of points among which the calculation should be made, marking the "start" and "finish". It is also possible to manually select which algorithm to use for calculations. When the algorithm completes its work, subsequent processing of the result is performed. Post-processing of the result is performed automatically according to pre-set rules for finding intersections - "loops". After determining the start and finish of the "loop", the algorithm for finding the optimal route on this section is launched. The calculation result replaces the non-optimal section.

3.2. Loops problem solving

When calculating the optimal path using the ant colony algorithm, route generation artifacts often arise in which some path segments intersect and form “loops” as in Figure 3.

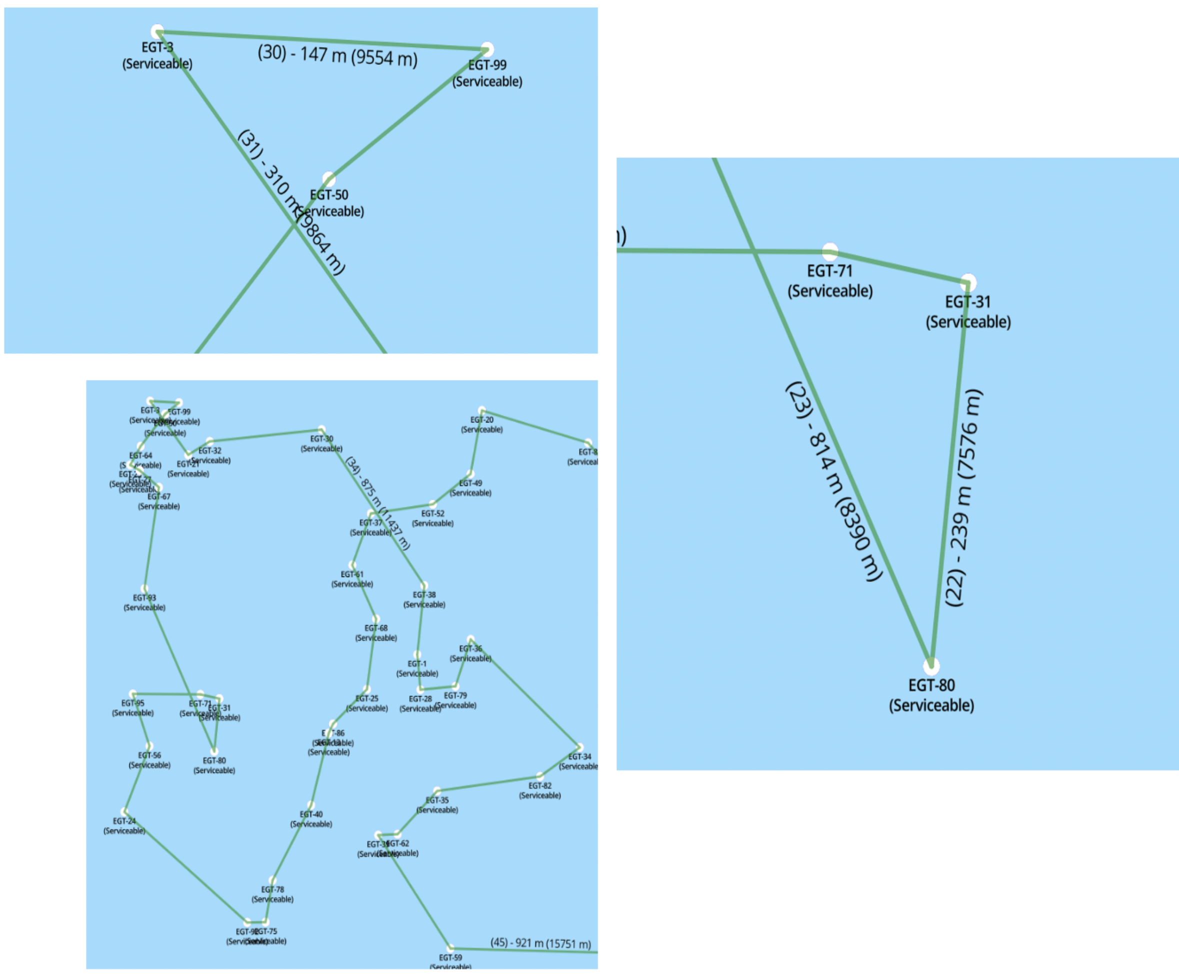

Figure 3: Loops visualization.

Such loops make the path longer, and their resolution can make the result more optimal. To determine the intersecting segments, we can use parameterization and represent the two segments in a system of equations. Let's say we have a segment AB and A_1B_1, where: $A=(X_a, Y_a)$, $B=(X_b, Y_b)$, $A_1=(X_{a1}, Y_{a1})$, $B_1=(X_{b1}, Y_{b1})$. The parametric representation of the segments:

$$P(t_1)=(x, y)=t_1(X_b, Y_b)+(1-t_1)(X_a, Y_a), \quad (9)$$

$$Q(t_2)=(x, y)=t_2(X_{b1}, Y_{b1})+(1-t_2)(X_{a1}, Y_{a1}) \quad (10)$$

Segments intersect if such a value exists $t_1, t_2 \in [0,1]$:

$$t_1 X_b+(1-t_1) X_a=t_2 X_{b1}+(1-t_2) X_{a1}, \quad (9)$$

$$t_1 Y_b+(1-t_1) Y_a=t_2 Y_{b1}+(1-t_2) Y_{a1}. \quad (10)$$

The segments intersect if after solving the system the following results were obtained for t_1, t_2:

$$0<t_1<1 \; and \; 0<t_2<1. \quad (11)$$

After finding the intersecting segments, we determine the starting and ending points of the loop. Then, on the resulting section, we determine the optimal route and replace the original section with the newly calculated one. Thus, we remove all intersections and "loops".

3.3. Software development

Before implementing the algorithms for finding the optimal route, we developed a software system for entering and storing data that are used in the experiments. The system was developed using web technologies according to the principles of "client-server" architecture. The software architecture is shown in Figure 1. The scope of the system is servicing offshore wind electric turbines.

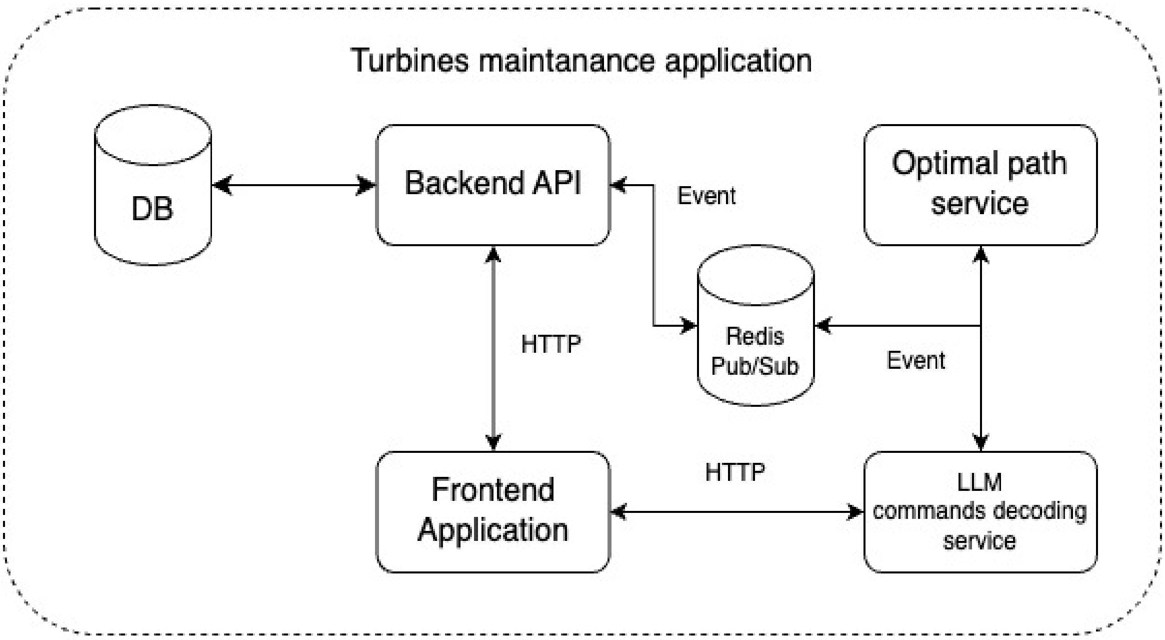

Figure 4: Application architecture.

The developed application consists of a set of services developed following the microservice architecture principles described in [16–19]. It involves several components: a frontend application that provides the user interface, a backend API responsible for business logic, a database (DB) serving as a data storage, an optimal route service that implements algorithms for route generation and optimization, and an LLM-based command decoding module, which processes textual information using a large language model, as described in [20].

The backend API is implemented in a robust way following the approaches shown in papers [21–22]. It is structured as a universal API, logically divided into modules responsible for entity management, turbine maintenance operations, and communication between services. The system is built with Node.js as the runtime environment, and the code is written in TypeScript, a statically typed superset of JavaScript. The core server functionality is implemented using the Express framework. The PostgreSQL is used as the relational database management system.

The application is aimed at automating the planning of offshore wind turbine maintenance. During maintenance, a marine vessel with a crew on board sails through a certain number of turbines and planning the optimal route plays a very important role in ensuring optimal fuel and time costs. The application interface is shown in Figure 5.

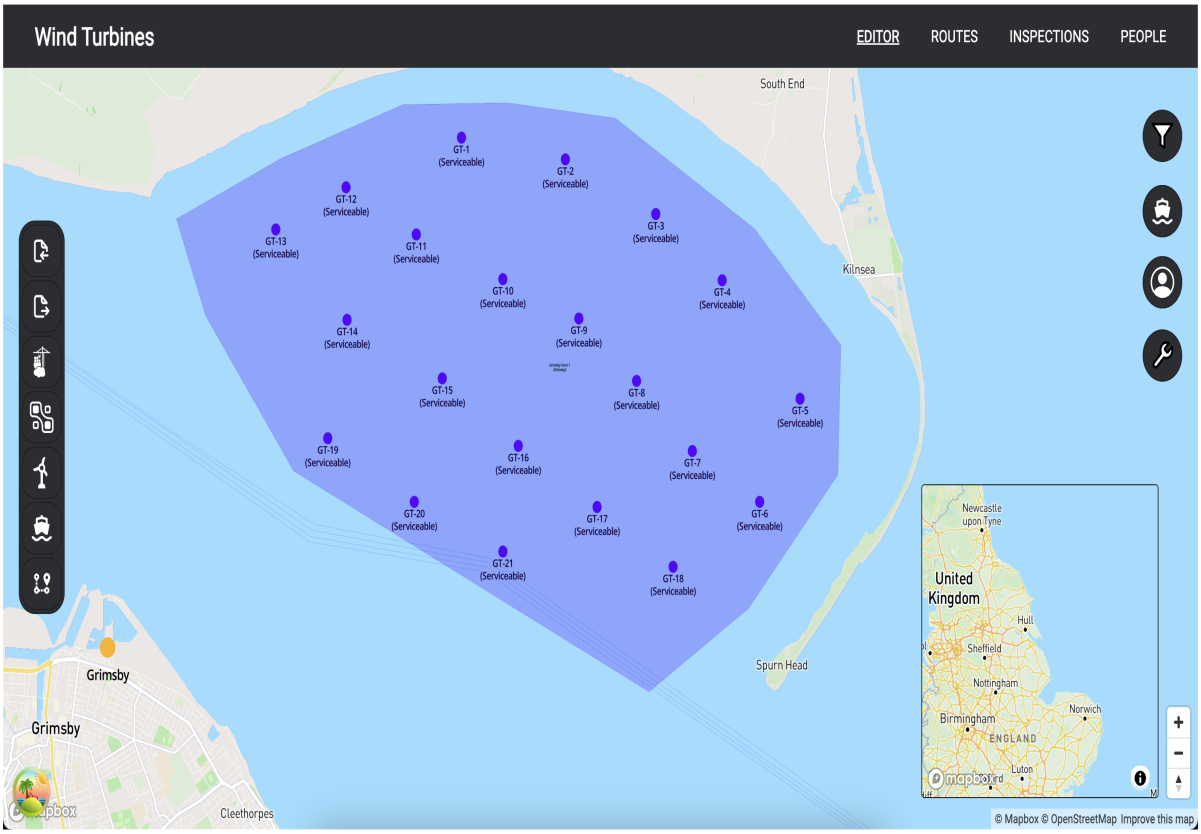

Figure 5: Application interface.

Figure 6 shows an example of the optimal route generation interface. If necessary, the user can manually change the route generation result using the interface as in Figure 7. For example, if, after generating a route, artifacts – “loops” – are detected, the user can manually resolve them independently, as shown in Figure 8.

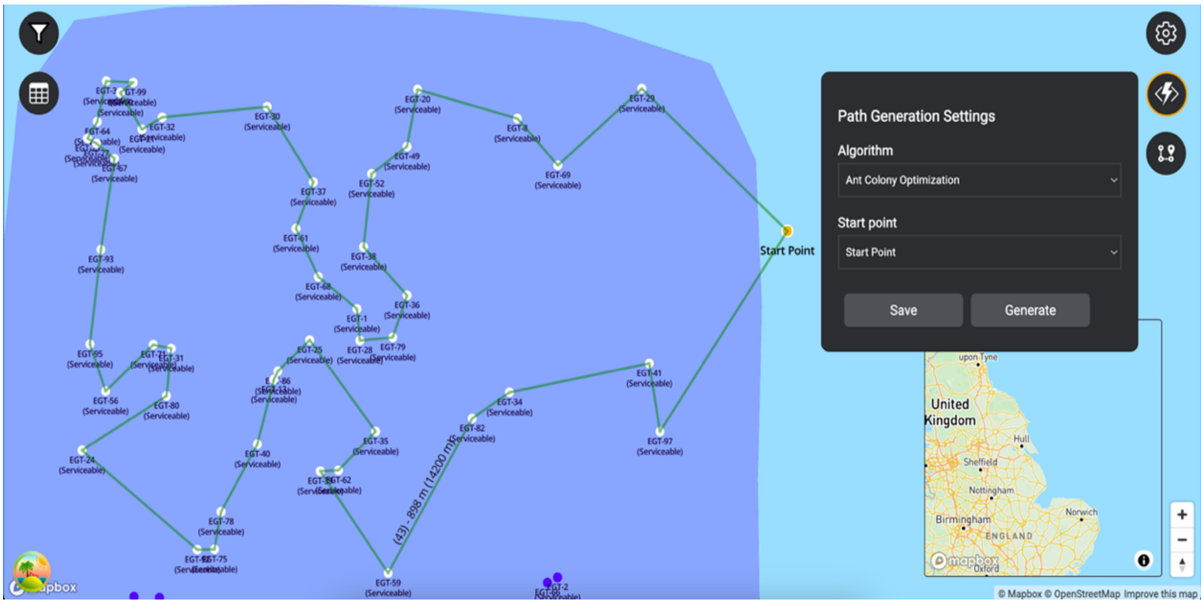

Figure 6: Optimal route calculation result.

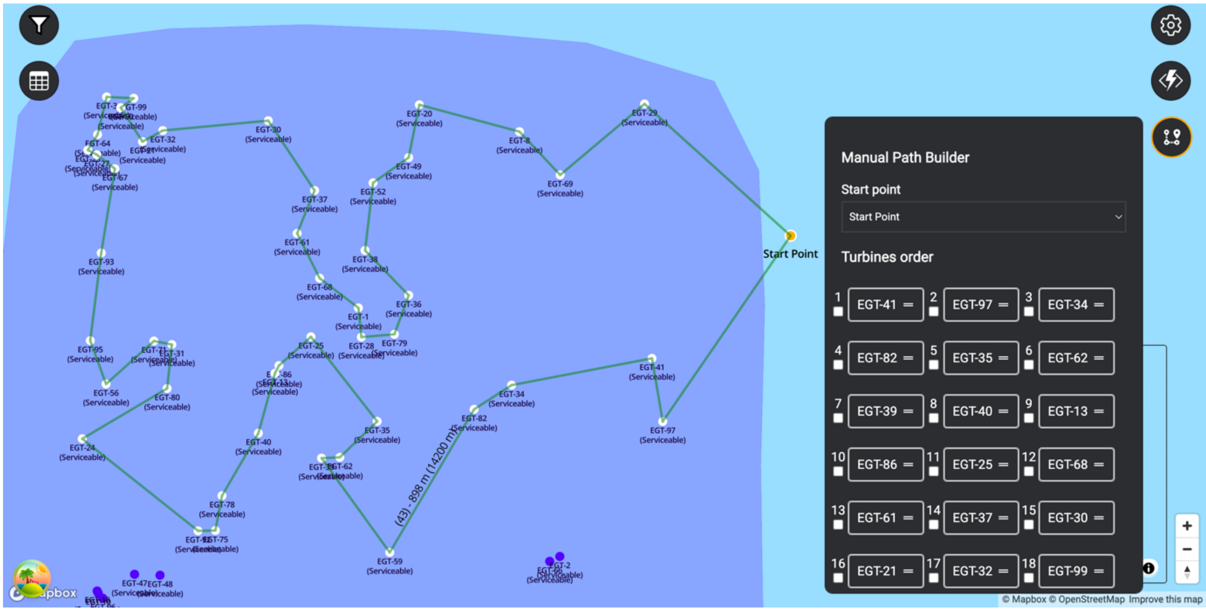

Figure 7: Route manipulation interface.

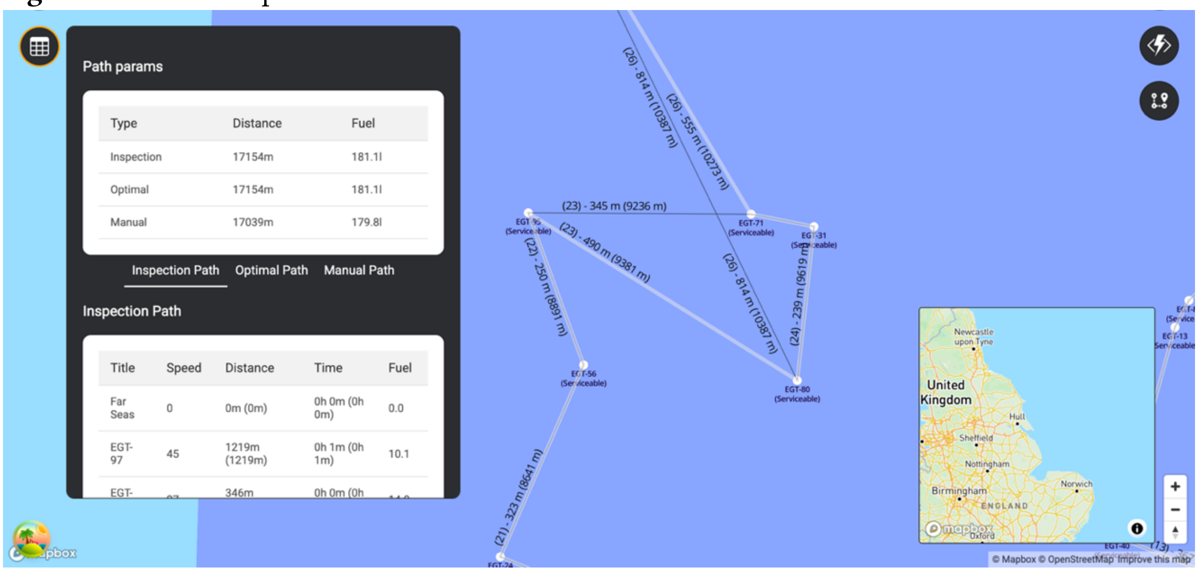

Figure 8: Loop correction result.

4. Experiment

The main way to conduct experiments in the decision-making information system is to simulate the real work of the operator. To evaluate the effectiveness of the proposed approach, a series of experiments were conducted. At the first stage, routes were generated using the Ant Colony Optimization algorithm for different sets of points. The obtained solutions demonstrated that ACO provides quasi-optimal paths with acceptable computation time. Another experiment of optimal path generation involved generating a route for 50 points using the ACO. In this scenario, all points were distributed in a random sequence a situation that is rarely happening in practice but is effective for testing the algorithm effectiveness. The obtained result is shown in Figure 9.

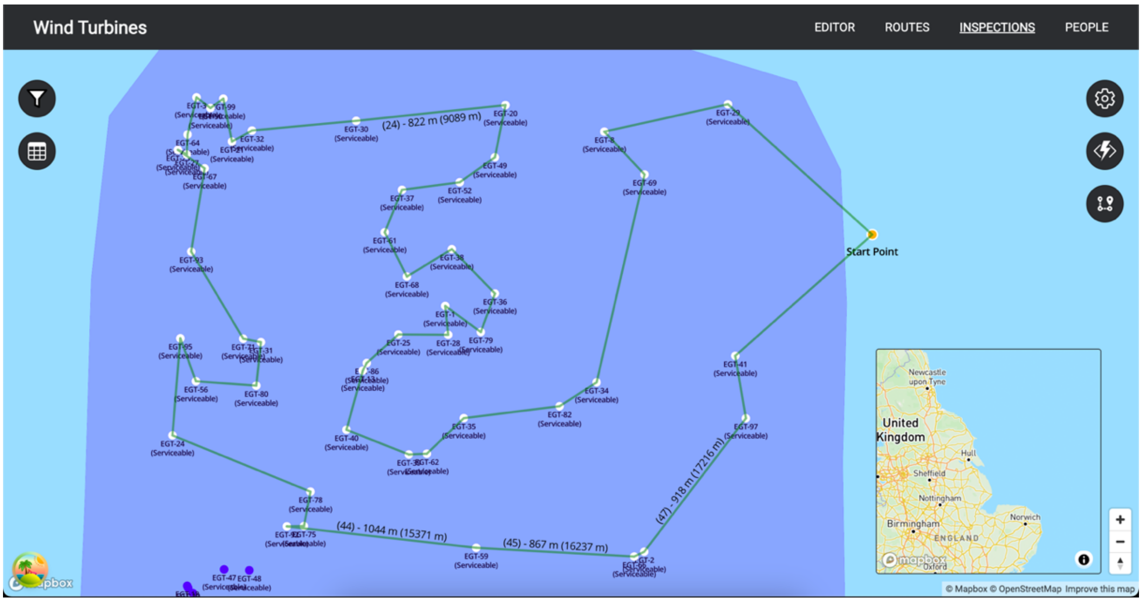

Figure 9: Result of calculating the optimal path for 50 points.

The ant colony optimization algorithm demonstrates high performance in determining optimal paths. However, it is not entirely perfect. In certain cases, route generation artifacts appear, so-called “loops” which are illustrated in Figure 10.

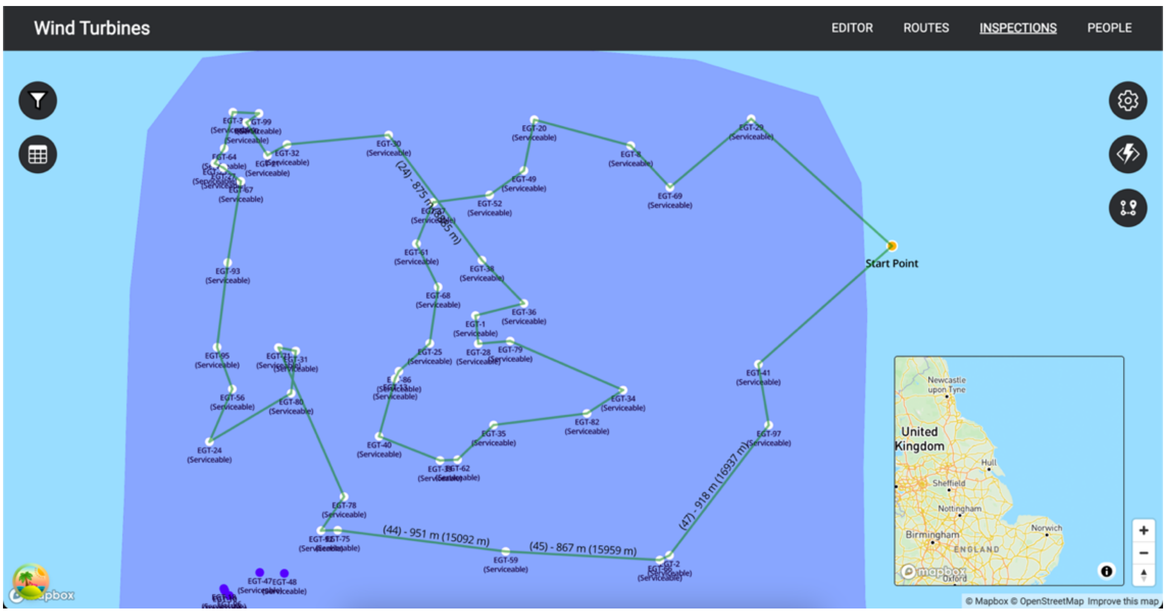

Figure 10: Formation of "loops" when calculating the optimal route.

To address these issues there were developed specific rules for detecting intersections within the generated route. Intersecting segments can be identified through direct enumeration and by solving a system of linear equations, which can be done within an acceptable time. Once an intersection is detected, the algorithm extracts all points located between the intersecting route segments and detects points of both the start and end of the intersection area to enhance the reliability of calculations. For this set of points, the optimal path is recalculated using the ACO algorithm. Significant aspect of this process is the accurate definition of the starting and ending points of the recalculated route segment to avoid getting of new artifacts. Given the limited number of points involved, the recalculation requires only a little amount of computation time. The corrected route is shown in Figure 11. The loop-correction procedure gives a measurable improvement, reducing the total route length by approximately 3%.

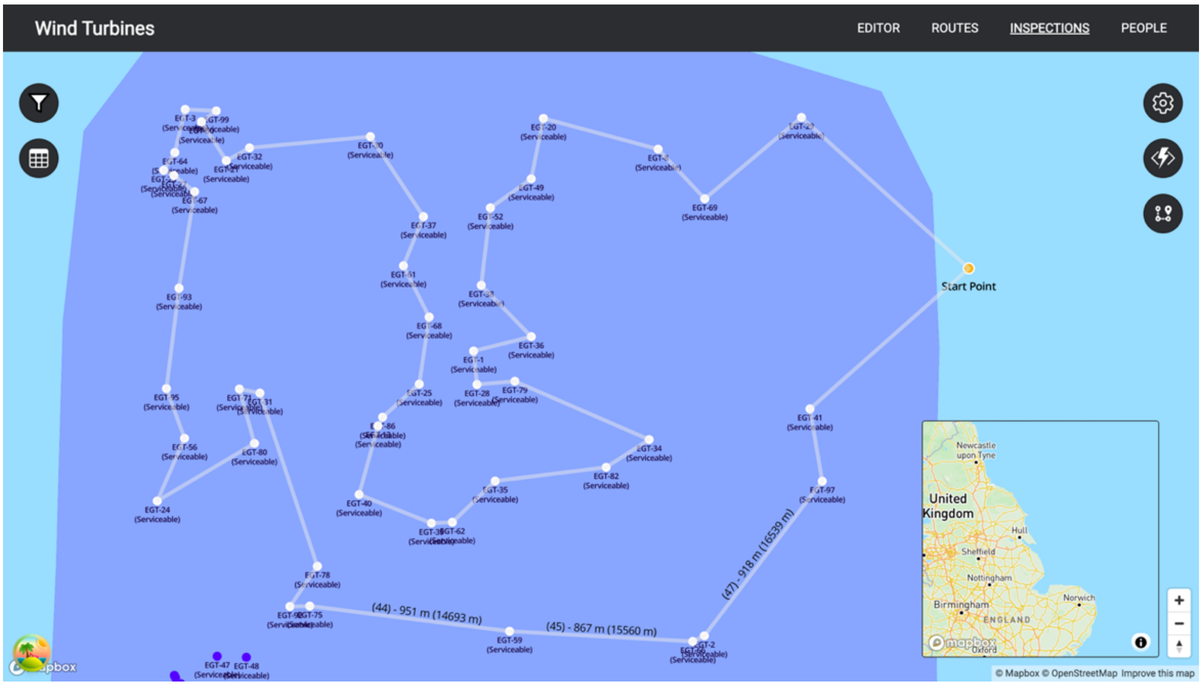

Figure 11: Corrected route.

5. Results

A series of experiments were carried out to simulate the real wok of a system operator. Initially, several configurations of point placement were manually created, followed by the execution of an algorithm developed to calculate the optimal path between them. The integration of an AI assistant significantly reduced the number of manual operations required, such as points selection, configuration, and algorithm selection. Next experiments were done testing the enhancing of the results of route generation using the ACO algorithm produced visible results that are very promising. The developed rules for detecting intersections and recalculating the route segments involved in loop made it possible to significantly improve the results of generation. It effectively eliminates “loops” and achieve results that more close to the most optimal route. A group of experts validated the effectiveness of this approach. One of the experiment results is shown in Figure 12, and in Table 1.

Table 1
Comparison of path generated, and path corrected by AI assistant

#	Route type	Distance	Fuel spent
1	Generated	18285 m	191.8 l
	Corrected	18003 m	186.4 l

2	Generated	18663 m	201.8 l
	Corrected	18204 m	189.4 l
3	Generated	18499 m	196.7 l
	Corrected	17984 m	182.1 l
4	Generated	18142 m	190.9 l
	Corrected	18120 m	189.6 l
5	Generated	18918 m	212.5 l
	Corrected	18106 m	190.4 l

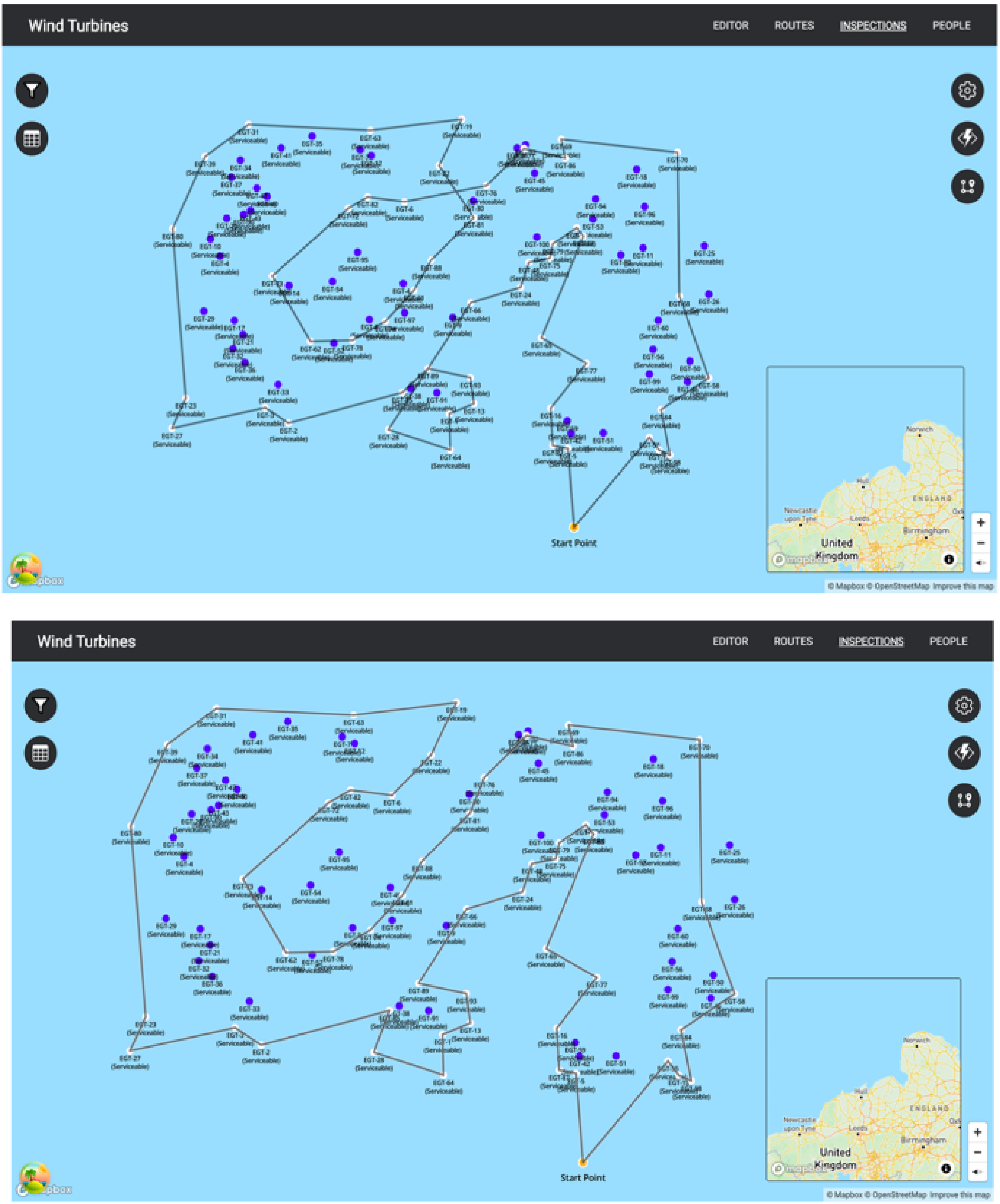

Figure 12: Intersections solving visualization.

6. Discussions

The main goal of this paper was to explore the potential of integrating an intelligent assistant into the software of an information system developed for optimal route finding. The implementation of intelligent assistant helps to minimize the time system operators spend interacting with the software and performing routine operations. The assistant allows to improve the quality of optimal route calculations which is the core functionality of the information system and can also be used for training of personal working with system.

Experimental results demonstrated that the intelligent assistant allows operators to efficiently and quickly perform their work and effectively build optimal routes with the help of it. The assistant can identify the necessary algorithm and parameters and run optimal route generation, then ensuring its quality and improve if necessary.

The route optimization module based on the ant colony algorithm also proved its effectiveness. Notably, the developed set of rules for detecting intersections, or “loops,” within the generated route and performing localized recalculations made it possible to optimize the initial path generation result by an average of 3–5%.

The obtained experimental results confirm that the developed software achieves adequate accuracy in generation of optimal paths and allows for further improvements of results through post-processing and correction of generation artifacts, compared to the initial calculations. The solution received positive evaluations from industry experts.

Future system development will focus on improving and refining the optimal route generation algorithm to ensure greater accuracy and computational efficiency when processing a larger number of points, along with additional experimental validation. Such advancements are expected to improve the performance of the ACO algorithm, allowing its practical usage. The next step involves implementing the system as a minimum viable product (MVP) in a production environment to begin testing by professionals in their work.

Another promising direction of research involves close collaboration with logistics specialists to integrate real-time sensor data streaming from operating vehicles, transmitting geo coordinates, speed, and direction, for subsequent data processing and visualization. This integration will support both the continuous training of the AI model and the real-time representation of operational data with the user interface.

7. Conclusions

The current stage of research and development of intelligent information systems for automating technological processes has been thoroughly analyzed. The analysis of algorithms and technologies applied to optimal route generation, system functionality, interface design, and data visualization were carried out. Based on the research, a set of requirements was defined and then there was developed an information system module that helps in decision making for generating optimal routes for solving the Traveling Salesman Problem (TSP). This module allows for optimal route generation using the Ant Colony Optimization (ACO) algorithm and includes the feature of subsequent route improvements with additional recalculations.

Emphasis was placed on developing of an intelligent assistant necessary for reducing operator work and minimizing decision making time. The assistant supports both text and voice-based interaction via text-to-speech technology, following the approaches presented in [23–24]. The integration of text-to-speech functionality provides a more friendly and intuitive interface, allowing the system to inform the operator about the status of route generation, system performance, or detected artifacts. This notably increases the efficiency and accessibility of operator interaction, especially in high-load production environment.

The principles of optimal route generation algorithms are presented in detail, including the mathematical models and equations of the ACO algorithm. A method for analyzing generated routes has been introduced, based on identifying intersecting segments and recalculating localized route fragments between intersection points. This approach allows the system to improve initially generated routes, achieving results closer to the optimal. To implement this enhancement, an algorithm of searching for intersections was developed that automatically detects intersections and performs recalculation of the loops found.

The software implementation of the system is discussed, highlighting its modular architecture and the technologies applied for data processing, visualization, and secure web-based interaction. Given that the system operates with sensitive data including object coordinates, user information, and operational parameters particular attention has been paid to data protection and security. Modern solutions for authentication, encrypted communication, and secure credential and log storage are planned to be implemented [25]. The database schema is designed to manage dynamic optimization data while ensuring strict access control and compliance with current information security standards [26-33].

The experiments were carried out, and they demonstrated the key functional capabilities of the system, showing the operator's interaction with the software during route generation and further improvement. Experimental results confirmed that the improved algorithm provides more accurate route generations, improving results by approximately 3–5% on average. Moreover, experiments involving the intelligent assistant showed that its integration significantly reduces the time required to configure and execute optimization tasks while improving the intuitiveness of user interaction.

Future work will be directed toward further improvement of route optimization algorithms, more integration of the intelligent assistant, and the application of LLM based technologies for predictive route analysis. Planned research includes the development of multimodal interaction methods that combine textual, auditory, and visual components, as well as the strengthening of data protection and cloud synchronization mechanisms. The next phase will focus on the adoption of advanced text-to-speech technologies to create a fully voice-operated interface, enabling operators to manage optimization and monitoring tasks more efficiently It's also planned to implement parallel computations to reduce the complexity and time needed for performing route generations and allowing to generate the optimal path among larger amount of points.

Declaration on Generative AI

During the preparation of this work, the authors used Grammarly in order to: Grammar and spelling check. After using these tools/services, the authors reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] I. Kozin, G. Kozina, U. Narzullayev, Fragmented metaheuristics for combined transport logistics problems, in: Proc. 3rd Int. Conf. on Problems of Logistics, Management and Operation in the East-West Transport Corridor (PLMO), IEEE, Baku, Azerbaijan, 2024, pp. 1–3. doi:10.1109/PLMO62307.2024.10887156.

[2] M. Lili, Research on the relationship between regional logistics demand and service innovation based on optimal allocation of factors in digital economy, in: Proc. 2021 IEEE 2nd Int. Conf. on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), IEEE, Nanchang, China, 2021, pp. 825–828. doi:10.1109/ICBAIE52039.2021.9389872.

[3] V.M. Reddy, T. Vaishnavi, K.P. Kumar, Speech-to-text and text-to-speech recognition using deep learning, in: Proc. 2023 2nd Int. Conf. on Edge Computing and Applications (ICECAA), IEEE, Namakkal, India, 2023, pp. 657–666. doi:10.1109/ICECAA58104.2023.10212222.

[4] A.M. Alshahrani, H.F. Ahmad, J. Hussain, Detecting health misinformation on social networking sites using large language models and deep learning-based natural language processing, in: Proc. 2024 2nd Int. Conf. on Foundation and Large Language Models (FLLM), IEEE, Dubai, UAE, 2024, pp. 561–568. doi:10.1109/FLLM63129.2024.10852503.

[5] M. Owais, A. Alshehri, Pareto optimal path generation algorithm in stochastic transportation networks, IEEE Access 8 (2020) 58970–58981. doi:10.1109/ACCESS.2020.2983047.
[6] V. Krasnobayev, A. Kuznetsov, A. Yanko, T. Kuznetsova, Solving the shortest path problem using integer residual arithmetic, in: Proc. 2020 IEEE Int. Conf. on Problems of Infocommunications, Science and Technology (PIC S&T), IEEE, Kharkiv, Ukraine, 2020, pp. 563–566. doi:10.1109/PICST51311.2020.9467947.
[7] X. Jiang, Z. Lin, T. He, X. Ma, S. Ma, S. Li, Optimal path finding with beetle antennae search algorithm using ant colony optimization initialization and different searching strategies, IEEE Access 8 (2020) 15459–15471. doi:10.1109/ACCESS.2020.2965579.
[8] W. Wang, et al., Development of an adaptive user support system based on multimodal large language models, in: Proc. 2024 IEEE Symp. on Visual Languages and Human-Centric Computing (VL/HCC), IEEE, Liverpool, UK, 2024, pp. 344–347. doi:10.1109/VL/HCC60511.2024.00044.
[9] J. Niu, et al., Design and implementation of intelligent software version management platform for power communication network based on large language models, in: Proc. 2024 IEEE 4th Int. Conf. on Information Technology, Big Data and Artificial Intelligence (ICIBA), IEEE, Chongqing, China, 2024, pp. 1451–1555. doi:10.1109/ICIBA62489.2024.10868397.
[10] K. Qiu, S. Bakirtzis, I. Wassell, H. Song, J. Zhang, K. Wang, Large language model-based wireless network design, IEEE Wireless Commun. Lett. 13(12) (2024) 3340–3344. doi:10.1109/LWC.2024.3462556.
[11] Y. Zdorenko, O. Lavrut, T. Lavrut, V. Lytvyn, Y. Burov, V. Vysotska, Route selection method in military information and telecommunication networks based on ANFIS, in: CEUR-WS Proceedings, vol. 2917, 2021, pp. 514–524.
[12] V. Vysotska, K. Przystupa, N. Sharonova, M. Derenskyi, G. Pliekhova, V. Repikhov, Information system for monitoring and planning maintenance of offshore wind farms, in: CEUR-WS Proceedings, vol. 3723(2), 2024, pp. 63–82.
[13] M. Li, Exploring the application of large language models in spoken language understanding tasks, in: Proc. 2024 IEEE 2nd Int. Conf. on Sensors, Electronics and Computer Engineering (ICSECE), IEEE, Jinzhou, China, 2024, pp. 1537–1542. doi:10.1109/ICSECE61636.2024.10729345.
[14] Y.-W. Xia, C. Yang, B.-Q. Chen, A path planning method based on improved single player-Monte Carlo tree search, IEEE Access 8 (2020) 163694–163702. doi:10.1109/ACCESS.2020.3021748.
[15] Z. Farisi, B. Zhu, J. Wang, Z. Wang, An improved ant colony algorithm for mobile robot path planning strategy, in: Proc. 2024 5th Int. Conf. on Computer, Big Data and Artificial Intelligence (ICCBD+AI), IEEE, Jingdezhen, China, 2024, pp. 334–338. doi:10.1109/ICCBD-AI65562.2024.00063.
[16] O.M. Bakhriddinuali, A.J. Samadillaugli, Z.S. G'olibugli, Microservice approach in designing a scalable architecture on the example of an electronic invoice system, in: Proc. 2021 Int. Conf. on Information Science and Communications Technologies (ICISCT), IEEE, Tashkent, Uzbekistan, 2021, pp. 1–3. doi:10.1109/ICISCT52966.2021.9670337.
[17] I.A. Kautsar, M.R. Maika, A.N. Budiman, A.B. Setyawan, J.Y. Awali, Microservice-based architecture: development of rapid prototyping supportive tools for project-based learning, in: Proc. 2023 IEEE World Engineering Education Conference (EDUNINE), IEEE, Bogota, Colombia, 2023, pp. 1–6. doi:10.1109/EDUNINE57531.2023.10102884.
[18] K. Adrio, C.N. Tanzil, M.C. Lianto, Z.E. Rasjid, Comparative analysis of monolith, microservice API gateway and microservice federated gateway on web-based application using GraphQL API, in: Proc. 2023 10th Int. Conf. on Electrical Engineering, Computer Science and Informatics (EECSI), IEEE, Palembang, Indonesia, 2023, pp. 654–660. doi:10.1109/EECSI59885.2023.10295809.
[19] I. Kyrychenko, D. Pronina, Comparison of Redux and React Hooks methods in terms of performance, in: Proc. 6th Int. Conf. on Computational Linguistics and Intelligent Systems (COLINS 2022), CEUR-WS, vol. 3171, 2022, pp. 791–800.

[20] C. Curto, D. Giordano, D.G. Indelicato, V. Patatu, Can a Llama be a watchdog? Exploring Llama 3 and Code Llama for static application security testing, in: Proc. 2024 IEEE Int. Conf. on Cyber Security and Resilience (CSR), IEEE, London, UK, 2024, pp. 395–400. doi:10.1109/CSR61664.2024.10679444.
[21] E. John, M. Siddique, Efficient semantic web services development approaches using REST and JSON, in: Proc. 2021 Int. Conf. on Decision Aid Sciences and Application (DASA), IEEE, Sakheer, Bahrain, 2021, pp. 231–235. doi:10.1109/DASA53625.2021.9682324.
[22] V. Tagdiwala, A. Bharoliya, P. Patel, D. Shah, M. Aibin, Robust client and server state synchronisation framework for React applications: react-state-sync, in: Proc. 2023 IEEE Canadian Conf. on Electrical and Computer Engineering (CCECE), IEEE, Regina, Canada, 2023, pp. 475–481. doi:10.1109/CCECE58730.2023.10289106.
[23] M. Kim, J. Choi, D. Kim, Y.M. Ro, Textless unit-to-unit training for many-to-many multilingual speech-to-speech translation, IEEE/ACM Trans. Audio, Speech, Lang. Process. 32 (2024) 3934–3946. doi:10.1109/TASLP.2024.3444470.
[24] S. Liu, A.S. Hussain, C. Sun, Y. Shan, Music Understanding LLaMA: Advancing text-to-music generation with question answering and captioning, in: Proc. ICASSP 2024 – IEEE Int. Conf. on Acoustics, Speech and Signal Processing, IEEE, Seoul, South Korea, 2024, pp. 286–290. doi:10.1109/ICASSP48485.2024.10447027.
[25] Y. Ling, C. Yang, X. Li, D. Bin, S. Han, M. Xie, Web security protection technology based on honeypot technology, in: Proc. 2022 IEEE Conf. on Telecommunications, Optics and Computer Science (TOCS), IEEE, Dalian, China, 2022, pp. 112–115. doi:10.1109/TOCS56154.2022.10016100.
[26] A. Berko, V. Vysotska, O. Naum, N. Borovets, S. Chyrun, V. Panasyuk, Big data analysis for startup of supporting Ukraine Internet tourism, in: Proc. 2023 IEEE 5th Int. Conf. on Advanced Information and Communication Technologies (AICT), IEEE, 2023, pp. 164–169.
[27] N. Antonyuk, V. Vysotska, V. Lytvyn, Y. Burov, A. Demchuk, I. Lyudkevych, I. Bobyk, Consolidated information web resource for online tourism based on data integration and geolocation, in: Proc. 2019 IEEE 14th Int. Conf. on Computer Sciences and Information Technologies (CSIT), vol. 1, IEEE, Lviv, Ukraine, 2019, pp. 15–20.
[28] D. Peleshko, T. Rak, J.R. Noennig, V. Lytvyn, V. Vysotska, Drone monitoring system DROMOS of urban environmental dynamics, in: CEUR-WS Proceedings, vol. 2565, 2020, pp. 178–193.
[29] A. Vysotsky, N. Antonyuk, A. Vysotskyi, V. Lytvyn, V. Vysotska, D. Dosyn, O. Slyusarchuk, Online tourism system for proposals formation to user based on data integration from various sources, in: Proc. 2019 IEEE 14th Int. Conf. on Computer Sciences and Information Technologies (CSIT), vol. 2, IEEE, 2019, pp. 92–97.
[30] Y. Matseliukh, M. Bublyk, V. Vysotska, Development of intelligent system for visual passenger flows simulation of public transport in smart city based on neural network, in: CEUR-WS Proceedings, vol. 2870, 2021, pp. 1087–1138.
[31] V. Lytvyn, A. Kowalska-Styczen, D. Peleshko, T. Rak, V. Voloshyn, J.R. Noennig, H. Pryshchepa, Aviation aircraft planning system project development, in: Proc. Conf. on Computer Science and Information Technologies, Springer, Cham, 2019, pp. 315–348.
[32] D. Uhryn, O. Naum, N. Antonyuk, I. Dyyak, L. Chyrun, A. Demchuk, T. Batiuk, Tourist itineraries plan design based on the behavior of bee colonies, in: CEUR-WS Proceedings, vol. 2631, 2020, pp. 516–539.
[33] I. Krislata, A. Katrenko, V. Lytvyn, V. Vysotska, Y. Burov, Traffic flows system development for smart city, in: CEUR-WS Proceedings, vol. 2565, 2020, pp. 280–294.