UDC 519.17:338.48:004.8 DOI 10.36910/775.24153966.2026.85.14

О. В. Бондарук, А. С. Осадчий, М. С. Вавдіюк

Луцький національний технічний університет

ОПТИМІЗАЦІЯ МАРШРУТІВ КУЛЬТУРНОЇ СПАДЩИНИ МЕТОДАМИ ТЕОРІЇ ГРАФІВ: МЕРЕЖА МУЗЕЇВ ТА "ЛУЦЬКІ КЛИКУНИ"

Анотація. У статті розглядається задача алгоритмічної оптимізації туристичних маршрутів на макрорівні (регіон) та мікрорівні (місто) на основі теорії графів. Дослідження базується на двох практичних кейсах у Волинській області: мережі з 7 музеїв та тематичного пішохідного маршруту з 21 скульптури «Луцьких Кликунів». Культурні об'єкти змодельовано у вигляді вершин зважених графів, де вагами виступають відстані або витрати на подорож. Програмна реалізація на мові Python об'єднує різні алгоритмічні підходи: для побудови ефективних циклів музейної мережі застосовано алгоритми Дейкстри, Крускала та евристику найближчого сусіда, а для оптимізації міського маршруту (задача комівояжера) - комбінацію жадібного алгоритму з локальною оптимізацією 2-opt. Результати підтверджують універсальність розробленого підходу: сформовано надійну модель для регіонального планування та розраховано оптимальний пішохідний маршрут довжиною 11,5 км, доступний для одноденної екскурсії.

Ключові слова: *теорія графів, задача комівояжера, оптимізація маршруту, алгоритм Дейкстри, алгоритм Крускала, логістика туризму, культурна спадщина Волині, Python, Луцьк, Луцькі Кликуни, алгоритм 2-opt, матриця Лапласіана.*

O.V. Bondaruk, A. S. Osadchyi, M. S. Vavdiiuk

OPTIMIZATION OF CULTURAL HERITAGE ROUTES USING GRAPH THEORY METHODS: MUSEUM NETWORK AND "LUTSK KLIKUNS"

Abstract. This paper addresses the problem of mathematical modeling and algorithmic optimization of tourist routes at macro- (regional) and micro- (urban) levels using graph theory. The research is based on two case studies in the Volyn region: a logistic network of seven key museums and a themed walking route featuring 21 "Lutsk Klikuns" bronze sculptures. The methodology involves representing the spatial distribution of cultural objects as weighted undirected graphs, where edges reflect real logistical costs (time, fuel, or pedestrian distance). The software implementation, performed in Python (Jupyter Notebook, NumPy), integrates distinct algorithmic strategies for different scales. For the regional museum network, Dijkstra's algorithm, the Nearest Neighbor heuristic, and Kruskal's algorithm were applied to construct a minimum spanning tree and cost-efficient cycles. For the urban pedestrian route, a combination of the Greedy algorithm and 2-opt local optimization was employed to solve the Traveling Salesman Problem. The results demonstrate the universality of the proposed approach: a logistically sound itinerary minimizing transit costs was created for automobile tourism, while an optimal pedestrian path of approximately 11.5 km (starting from point A1) was calculated for the city tour. The practical significance of the work lies in automating the planning of cultural expeditions and enhancing the tourist experience through scientifically grounded navigation.

Keywords: *graph theory, Dijkstra's algorithm, Kruskal's algorithm, tourism logistics, cultural heritage of Volyn, Python, Traveling Salesman Problem (TSP), route optimization, tourism, Lutsk, Lutsk Klikuns, 2-opt algorithm, Laplacian matrix.*

Introduction. Efficient planning of tourist routes between cultural heritage objects is an important practical task in modern urban navigation and decision-support systems [3, 7]. Museums and historical landmarks can be naturally represented as vertices of a weighted graph, where edges correspond to travel distances or time costs between locations [8]. Within this framework, classical graph algorithms provide a reliable mathematical and computational foundation for solving route optimization problems related to tourism and cultural infrastructure, such as the Traveling Salesman Problem (TSP) or Vehicle Routing Problem (VRP) [1, 9]. Fundamental approaches, including Dijkstra's algorithm and local search heuristics like 2-opt, remain relevant for creating practical navigation solutions [2, 6].

Lutsk, the historic center of the Volyn region, serves as an ideal case study for applying these methods due to its unique cultural heritage. One of the modern tourist attractions of the city is the "Lutsk Klikuns" (town criers). Historically, these were guards who patrolled the castle walls, announced the time, and warned of danger. Today, the memory of this profession is preserved in the form of 21 bronze sculptures installed in various parts of the city. While the cultural value of these objects is undeniable, their geographical distribution presents a logistical challenge. The sculptures are scattered across the city - from the central park and the Old Town to remote railway areas. Similarly, on a regional level, the network of museums in Volyn requires efficient logistical planning to minimize travel costs between distant settlements.

For a tourist unfamiliar with the topography, visiting these locations can turn into a chaotic and exhausting experience, leading to wasted time and missed objects. Therefore, the task of building optimal routes - both for pedestrian city tours and regional vehicle travel - using mathematical modeling is an urgent scientific and practical task.

The scientific novelty of this study lies in the integrated application of classical graph algorithms to a small-scale regional cultural heritage network based on real geographic data, combined with comparative evaluation using MST as a lower bound.

Analysis of recent research and publications. The problem of finding the shortest route through a set of points is known in mathematics as the Traveling Salesman Problem (TSP). Since finding an exact solution for a large number of vertices is an NP-hard problem, heuristic algorithms are usually employed in tourism applications. Scientists such as Applegate and Cook [1] have significantly advanced the methods of solving TSP.

For determining shortest paths in weighted graphs with non-negative edge weights, Dijkstra's algorithm remains one of the most widely used methods, as stated in [2]. In applied routing and navigation problems, greedy heuristics such as the Nearest Neighbor method are often combined with shortest-path computations to construct feasible visiting sequences within acceptable computational time. To further optimize the route, local search methods like k-opt (Lin-Kernighan) are used, as described in the works of Helsgaun [6].

In the field of logistics and vehicle routing (VRP), significant contributions have been made by Toth and Vigo [9]. However, most existing navigation solutions focus on automobile transport, ignoring the specifics of pedestrian infrastructure (sidewalks, crossings, park paths) and the unique requirements of urban cultural heritage sites, a gap highlighted in recent studies by Gavalas [3, 4] and Rodriguez [8].

Unsolved aspects of the general problem. Despite the extensive theoretical development of graph algorithms, their application to small-scale cultural heritage networks based on real geographic data remains insufficiently documented. In particular, the combined use of shortest-path algorithms and greedy route construction methods for museum networks has not been thoroughly analyzed in the context of applied and educational research. This creates a gap between theoretical algorithmic studies and practical route-planning solutions for regional cultural tourism. Furthermore, in practical terms, existing general navigation applications (such as Google Maps or Waze) typically focus on building a route between two points. Creating a multi-point itinerary (e.g., involving 21 distinct locations) with automatic sequence optimization remains a non-trivial task for standard users. Currently, there is no scientifically grounded, optimized route for specific cultural products like the "Lutsk Klikuns" quest that takes into account the real topology of the city's pedestrian network.

Paper objective. The goal of this paper is to develop and implement a unified algorithmic approach for optimizing tourist routes at both regional and urban scales using classical graph theory and heuristic methods within the Python programming environment. The study targets two distinct practical applications: constructing an efficient logistic network between museums in the Volyn region and generating an optimal pedestrian route for visiting all 21 "Lutsk Klikuns" sculptures. In this research, optimality is defined in a practical sense - as the construction of closed routes where total length is minimized subject to real geographic data and infrastructure constraints. The proposed approach does not aim to find a global exact solution to the Traveling Salesman Problem (TSP), but rather to obtain near-optimal, computationally efficient itineraries suitable for real-world tourism. The effectiveness of the developed solutions is evaluated by minimizing total travel distance and comparing the results against theoretical lower bounds, such as the minimum spanning tree of the network.

Main material. Case Study: Regional Museum Network. *1. Problem formulation.* Mathematical modeling of a tourist route requires representing the inter-city space as a weighted graph G = (V, E), where the set of vertices V represents museums, and the set of edges E represents transport routes between them (visualized in Fig. 1). $\forall e \epsilon E$ has a weight $w(e)$ corresponding to the geographical distance in kilometers.

In this study, we consider the problem of constructing an optimal tourist cycle that starts and ends at the same vertex. A key feature of this problem in a real environment is that the optimal sequence of visits to all museums does not always allow for a simple Hamiltonian cycle. Due to the specific location of roads, the shortest path between two non-adjacent attractions in the sequence may require passing through already visited vertices or edges more than once. Therefore, the tourist route is defined as a closed walk of minimum total weight $L = \Sigma\, w(e_i)$ which covers all target vertices of the graph. This formulation reflects realistic transport constraints and allows intermediate transit through previously visited locations when necessary.

To ensure consistency, several formal assumptions have been made. We assume that the graph is connected, which guarantees that every museum is accessible from any starting point. Distances are considered symmetric, $w(v_i, v_j) = w(v_j, v_i)$. In addition, the model relies on triangle inequality, i.e., the weight of an edge between two vertices is always less than or equal to the sum of the weights of any alternative path between them. These assumptions allow us to consider the distances between landmarks as a metric space.

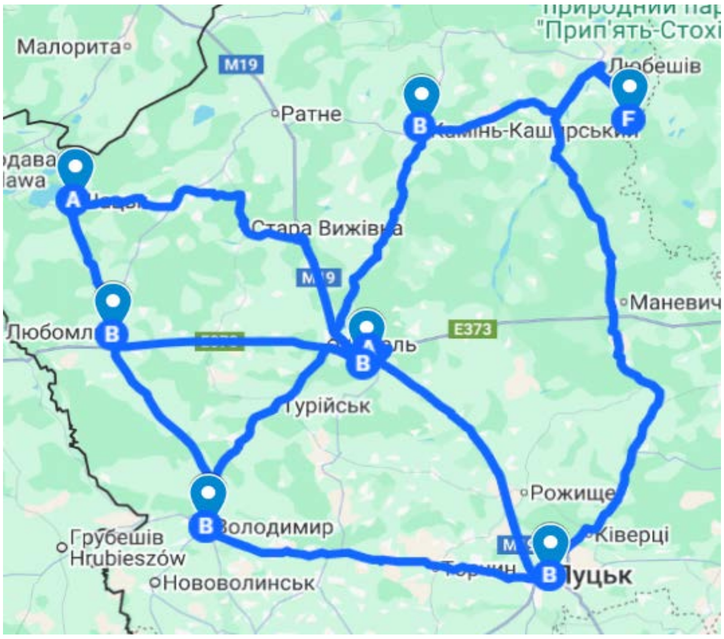


Fig. 1. **Graph representation of the Volyn museum network.**

Description of algorithms. The problem of constructing an optimal tourist route is a practical application of the traveling salesman problem (TSP), which is classified as an NP-hard problem. For the purposes of this study, we use a combination of classical graph algorithms to achieve an effective routing strategy for various urban cultural sites.

Dijkstra's algorithm. Dijkstra's algorithm is used to determine the shortest path between any two vertices in a basic transport network. In our study, we apply this algorithm to calculate the distance matrix nxn for selected sets of objects, ensuring that each pairwise distance used in subsequent steps is the absolute minimum allowed by the urban infrastructure.

Nearest Neighbor (NN) heuristic. To build a sequence of visits, we use the nearest neighbor heuristic. This “greedy” algorithm starts from a selected location and repeatedly visits the nearest unvisited object until all locations are included, finally returning to the starting point. This approach is particularly effective for tourist navigation because it mimics natural human decision-making.

Kruskal's algorithm. We use Kruskal's algorithm to construct the minimum spanning tree (MST) of the network. The MST represents the minimum connection cost and serves as a theoretical benchmark for evaluating the efficiency of routes generated by the nearest neighbor method.

Data description. The effectiveness and accuracy of route optimization directly depend on the quality of the input spatial data. For case study 1, a group of seven key cultural sites in the Volyn region was selected: Kamin-Kashyrskyi Folk Museum of Local History, Luboml Local History Museum, The Kosach Estate, Volyn Regional Museum of Local History, Lobnensky Museum of Partisan Glory, Museum of Flora and Fauna of Shatsk Forest College, Volodymyr Historical Museum named after O. M. Dvernitsky.

Information about these institutions, their exact geographical location, and status was collected based on data from open government and specialized sources. In particular, materials from the information portal “Museum Space of Volyn,” data from the Museum Fund of Ukraine, and official web resources of territorial communities were used. A 7x7 distance matrix was created to represent the distances between these objects mathematically (Table 1). Data was collected by analyzing routes on Google My Maps online maps. This made it possible to obtain real distance measurements in kilometers, taking into account the public road network of the Volyn region and the optimal routes between settlements. It should be noted that some elements of this matrix are undefined. This is due to the geographical location of objects in different areas of the region and the peculiarities of the road infrastructure: in certain cases, direct road connections between two museums may be irrational. In such situations, the algorithm considers these routes to have infinite weight ($w = \infty$), which encourages the construction of a route through key logistics hubs and main roads in the region.

1- Kamin-Kashyrskyi Folk Museum of Local History
2 - Luboml Local History Museum
3 - The Kosach Estate

4 - Volyn Regional Museum of Local History
5 - Lobnensky Museum of Partisan Glory
6 - Museum of Flora and Fauna of Shatsk Forest College
7 - Volodymyr Historical Museum named after O. M. Dvernitsky

Table 1

Distance matrix between museums in kilometers

	1	2	3	4	5	6	7
1	0	-	61.8	-	60.6	-	-
2	-	0	59	-	-	32	52
3	61.8	59	0	66	-	93	60
4	-	-	66	0	151.1	-	77
5	60.6	-	-	151.1	0	-	-
6	-	32	93	-	-	0	-
7	-	52	60	77	-	-	0

Experimental implementation. To verify the theoretical models described for case study 1, a software implementation was carried out in Python. Below is a description of the development environment, a detailed analysis of the code construction logic, and the stages of calculating the optimal route for the selected museum network.

The algorithms were developed and tested in the interactive Jupyter Notebook environment. This tool was chosen because it allows step-by-step execution of computational blocks, which is critical for verifying the intermediate results of the Dijkstra algorithm. The environment allows for isolated debugging of individual modules - from the preparation of the initial matrix to the final reconstruction of the path - and instant verification of the state of data arrays without the need to restart the entire program. The NumPy library was used for manipulating numerical data and working efficiently with the distance matrix.

The process of finding the optimal path is based on the sequential execution of the main stages of data processing, which are implemented in the form of interconnected functions in the Jupyter Notebook environment.

Module for finding the shortest paths (Dijkstra’s function). The program uses a graph adjacency matrix that was manually initialized based on collected geographic data, where pairs of museums without a direct connection are assigned an infinite value. The dijkstra function calculates the minimum distances from each of the 7 points to all others. This iterative process allows the program to select the nearest unvisited vertex each time and update the weights of the paths through it. In parallel with this, a list of predecessors is formed - a data structure that remembers the node from which we arrived at the current point, which is key for further visualization of real movement.

Tourist cycle formation module (tsp function). At the next stage, using the already formed complete matrix of shortest distances dist_matrix, the “nearest neighbor” strategy is implemented within the tsp function. Starting from the selected starting point start_node_index, the algorithm sequentially selects the nearest unvisited museum until it covers all 7 locations in the area. An important feature of the developed logic is automatic route closure: after visiting the last object, the program independently calculates the return path to the starting point, forming a complete cycle.

Transit route reconstruction module (reconstruct_path function). The final stage is devoted to the complete reconstruction of transit routes. Since direct connections between individual museums in Volyn are often impossible, the program uses the reconstruct_path function. It works on the basis of an array of predecessors and recursively unfolds each segment of the route, adding all the necessary intermediate nodes. This transforms an abstract sequence of museums into a detailed chain of indices, a path list suitable for navigation.

Connectivity analysis using Kruskal's algorithm (kruskal function). A separate block of the program is dedicated to constructing a minimum spanning tree (MST) using the kruskal function. The process involves sorting all available road connections by length and gradually adding the shortest edges to the overall network. To manage connectivity components and prevent the formation of cycles in the code, a system of auxiliary functions is used, where get_root searches for the root for the current set using the path compression method, while join_sets and join_groups provide logical merging of components, with join_groups interacting directly with root elements to establish connections between different groups of vertices. This complex algorithm allows us to determine the theoretical minimum length of roads required to connect all seven museums into a single network, which becomes a benchmark for comparison with the results of the tourist route obtained. The complete implementation of the features discussed above can be accessed in our GitHub repository [11].

Case study: Cultural Heritage Trail "Lutsk Klikuns". *1. Problem formulation.* To solve the problem of optimizing the excursion route, the apparatus of graph theory was used. The city road network is modeled as a weighted graph $G = (V, E)$, where $V = \{v_1, v_2, \dots, v_n\}$ is the set of vertices corresponding to the locations of the "Lutsk Klikuns" sculptures ($n = 21$), and E is the set of edges connecting them. Each edge (i, j) is assigned a weight w_{ij}, which corresponds to the pedestrian distance between points i and j in meters.The task is reduced to finding the Hamiltonian path (or circuit) of minimum length that passes through all vertices of the graph exactly once. Mathematically, the objective function for the Traveling Salesman Problem (TSP) is minimized:

$$L = \sum_{i=1}^{n-1} w_{p_i, p_{i+1}} + w_{p_n, p_1} \rightarrow min$$

where p is the permutation of vertices representing the order of visiting the objects.

Description of algorithms. The computational experiment was conducted in the interactive environment Jupyter Notebook. The software implementation was performed using the Python 3.13.5 programming language [10]. A distinctive feature of the developed solution is the refusal to use heavy external frameworks for graph processing. Instead, a native implementation of algorithms [2] was created using standard Python libraries, which ensures high code performance and ease of deployment.

1. Microsoft Excel was used to collect, structure, and store the initial dataset (coordinates and distance matrix).

2. Google My Maps service was used to visualize the route nodes and verify the pedestrian accessibility of the paths.

3. The complete source code and dataset are available at GitHub repository [12]

Two algorithmic approaches were implemented in the program code to find the solution:

a) *Greedy Algorithm (Nearest Neighbor)*. The basic algorithm selects the nearest unvisited node as the next step. It is computationally fast but often leads to suboptimal solutions due to "local traps", creating excessive loops at the end of the route.

b) *2-opt Heuristic Optimization*. To improve the route obtained by the greedy method, the 2-opt local search algorithm was applied manually. The essence of the method is to iteratively check for intersections in the route graph. If the inequality holds:

$$dist(A, C) + dist(B, D) < dist(A, B) + dist(C, D)$$

then edges (A, B) and (C, D) are removed, and the route is reconnected through edges (A, C) and (B, D). This procedure is repeated until no further improvements are possible, eliminating inefficient "zigzags" in the path.

Research results. Case Study: Regional Museum Network. To verify the theoretical models described above, a software implementation was developed in the Python programming language. This section outlines the computational environment, the structure of the implemented algorithms, and the main stages involved in constructing an efficient tourist route for the selected museum network.

The algorithms were implemented and tested in the interactive Jupyter Notebook environment. This environment was chosen because it supports step-by-step execution of computational blocks, which is essential for monitoring intermediate results, particularly during the execution of Dijkstra's algorithm. Such an approach enables isolated debugging of individual modules, ranging from the initialization of the distance matrix to the final reconstruction of the tourist route. Numerical computations and matrix operations were performed using the NumPy library [5].

Analysis of the routes obtained. The use of the Dijkstra function made it possible to form a complete distance matrix, eliminating the problem of the absence of direct roads between distant objects. Visualization of one of the constructed routes (Fig. 2), starting from the Volyn Regional Museum of Local History (vertex 3, Lutsk), demonstrates the algorithm's ability to construct a complete logistical trajectory. The total distance of the calculated tourist cycle was 574.50 km, which is an acceptable indicator for a multi-day car trip through the region.

During the experiment, the optimal route was obtained with the start and end at vertex 3 (Lutsk): 3 → 2 → 1 → 5 → 1→ 6 → 2 → 0 → 4 → 3. Analysis of this sequence shows that although tourists have to pass through certain nodes (for example, vertices 1 - Luboml and 2 - Kolodyazhne) more than once, such a route is mathematically and practically the most optimal for this road configuration. Repeated visits are due to the tree-like structure of the road network in certain parts of the region, where certain locations have a limited number of connecting highways. Returning through already visited points in such conditions is a necessary condition for minimizing the total distance compared to using significantly longer detours. The efficiency of the constructed route was assessed by comparing its total length (574.50 km) with the weight of the minimum spanning tree (MST), which was 331.4 km. In this study, the MST acts as the lower limit of theoretical connectivity. Although the tourist cycle exceeds the MST in length, this is explained by the need to close the route and return to the starting point, as well as the specifics of transit routes. The small gap between these indicators, given the actual topology of the roads, indicates the high quality of the solution found.

The developed approach has direct practical significance for the development of domestic tourism, since automated calculations allow for the optimization of fuel and time costs for organized tour groups and the creation of dynamic itineraries that can be adjusted to the selected starting point (e.g., Lutsk, Volodymyr, or Luboml), and provide effective logistical support for regional cultural events.

Despite its successful implementation, the study revealed certain limitations of the nearest neighbor heuristic used. The main drawback is the algorithm's “short-sightedness”: in the final stages of cycle construction, it may be forced to choose very long edges to return to the remaining points, which sometimes significantly increases the final distance. In addition, the model is based on static geographical distances and does not take into account dynamic factors such as current road conditions, traffic jams, or seasonal traffic restrictions.

The experiments confirmed the viability of the mathematical model. The resulting route is logistically justified and demonstrates a balance between the complexity of calculations and the quality of the final result. Further improvement of the system may consist in the integration of local search algorithms (e.g., 2-opt) to minimize route intersections and take into account the opening hours of museums.

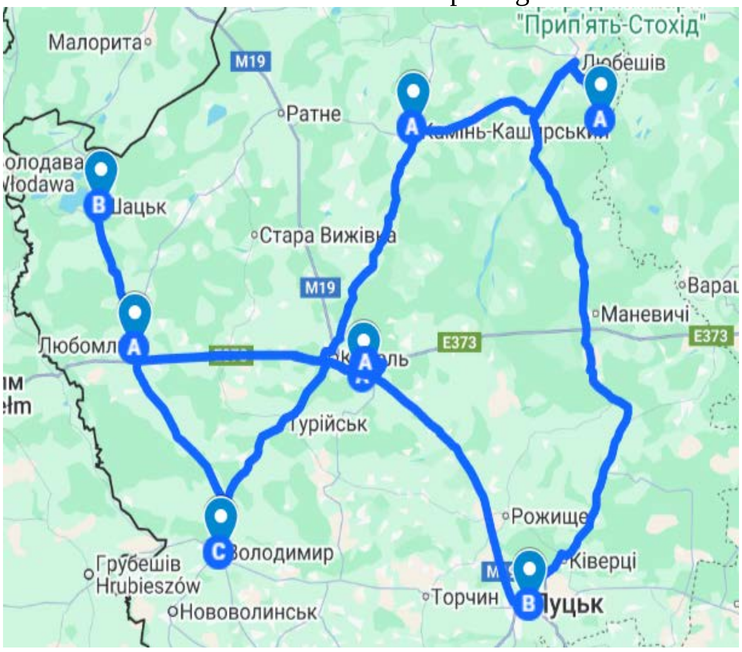


Fig. 2. **Visualization of one of the routes (starting point - Volyn Regional Museum of Local History), constructed based on the application of algorithms.**

Case study: Cultural Heritage Trail "Lutsk Klikuns". The starting point was chosen as A1, where the sculpture "Radyk Zadovolenyi" is located. To interpret the matrix, the correspondence of codes to the first 7 sculptures is as follows: A1 - Radyk Zadovolenyi, B2 - Zustrichayko, C3 - Hnat, D4 - Vasyl «Soloveiko», E5 - Franyo, F6 - Khvatskó and Prudkó, G7 - Knyzhko. The first result of the study is the distance matrix calculated for these points. A fragment of this data is presented in Table 2. For better visual perception, a color gradient is applied to the cells: the darker the shade, the greater the distance between the objects. The full distance matrix is provided in the “Distance matrix” worksheet of the “Table for the graph_Lutsk Klikuns_.xlsx” file, available in the repository [12].

Before applying the routing algorithms, the structural properties of the graph were analyzed using the Laplacian matrix. Given the large dimension of the network (n = 21), the full matrix is provided in the “Laplacian matrix” worksheet of the “Table for the graph_Lutsk Klikuns_.xlsx” file, available in the repository [12]. Below is the principal fragment (Table 3) corresponding to the first 7 vertices (7x7).

Table 2

Fragment of the distance matrix between the first 7 locations of "Lutsk Klikuns" in meters

	A1	B2	C3	D4	E5	F6	G7
A1	-	2850	3260	4390	5040	5390	4470
B2	2850	-	1010	2080	2850	3200	2200
C3	3260	1010	-	1150	1910	2300	1360
D4	4390	2080	1150	-	865	1450	779
E5	5040	2850	1910	865	-	572	1060
F6	5390	3200	2300	1450	572	-	962
G7	4470	2200	1360	779	1060	962	-

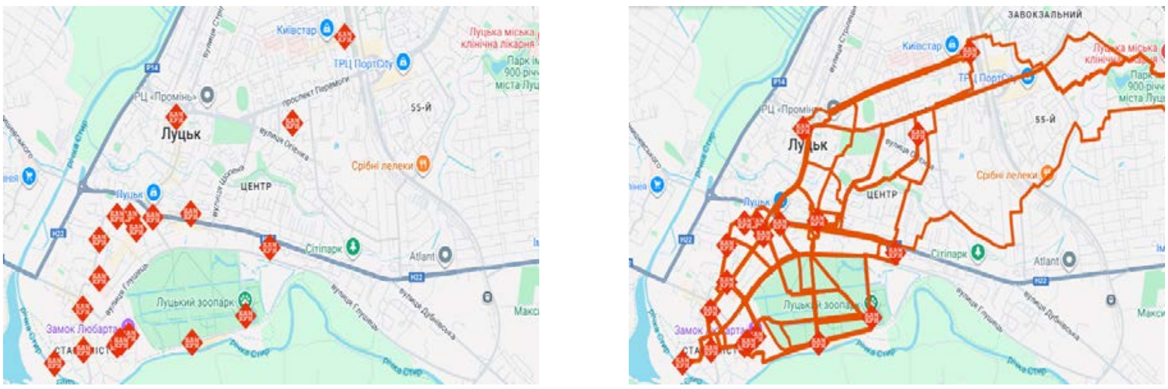
Table 3

Fragment of the Laplacian matrix (7x7)

	A1	B2	C3	D4	E5	F6	G7
A1	20	-1	-1	-1	-1	-1	-1
B2	-1	20	-1	-1	-1	-1	-1
C3	-1	-1	20	-1	-1	-1	-1
D4	-1	-1	-1	20	-1	-1	-1
E5	-1	-1	-1	-1	20	-1	-1
F6	-1	-1	-1	-1	-1	20	-1
G7	-1	-1	-1	-1	-1	-1	20

The algebraic properties of this matrix (specifically the Fiedler value $\lambda_2 > 0$) confirm that the graph is fully connected, allowing for the construction of a valid Hamiltonian cycle.

The geographical distribution of the 21 historical sculptures ("vertices") across the city map is shown in Fig. 3 (a). The same figure also illustrates the constructed pedestrian graph (Fig. 3 (b)), including all potential edges between the tourist objects, which demonstrates the high density and complexity of the route network.



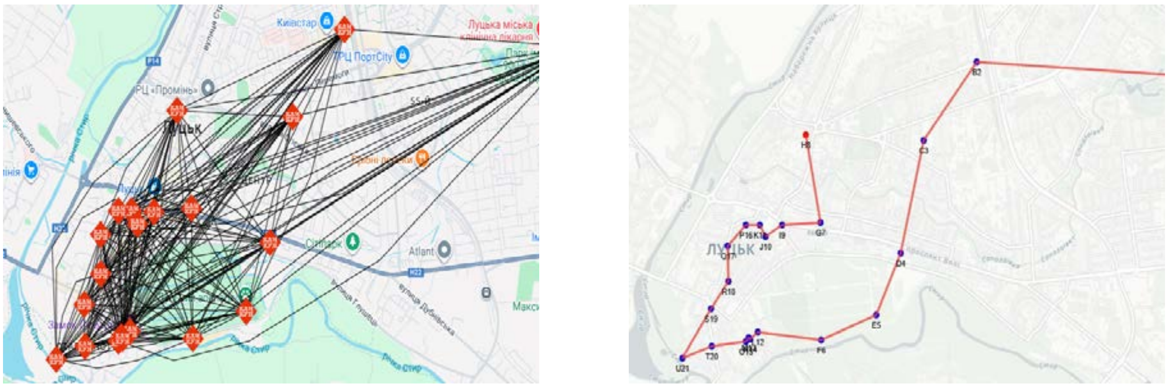
a) b)

Fig. 3. **Location of the "Lutsk Klikuns" vertices and visualization of the pedestrian graph edges on the city map.**

For the mathematical analysis, the pedestrian network was represented as an abstract weighted graph (Fig. 4 (a)) suitable for algorithmic processing. An initial route was constructed using a Greedy heuristic and then refined using the 2-opt optimization algorithm. The resulting Hamiltonian path and the corresponding structural graph model are shown in Fig. 4 (b).

Route sequence: A1 (Radyk Zadovolenyi) → B2 (Zustrichayko) → C3 (Hnat) → D4 (Vasyl «Soloveiko») → E5 (Franyo) → F6 (Khvatskó and Prudkó) → L12 (Vertun) → N14 (Klikun Andrii) →

M13 (Vartko and Vartko) → O15 (Stephan) → T20 (Vogniar) → U21 (Mykytovych) → S19 (Bratko and Bratko) → R18 (Providnyk) → Q17 (Kliuchnyk) → P16 (Kavus) → K11 (Muzyka) → J10 (Trylinko) → I9 (Zirko) → G7 (Knyzhko) → H8 (Semen Gust).



a) b)

Fig. 4. **Graph model of connections between tourist objects and the optimized pedestrian route obtained using the Greedy + 2-opt algorithm.**

Quantitative indicators:

1. Total number of vertices: 21.
2. Number of unique edges analyzed: 210.
3. Total length of the optimized route: 11,503 meters.

Visual analysis of the built route (Fig. 4 (b)) shows that the algorithm successfully avoided large "zigzags" and backtracking. For a tourist moving at an average speed of 4 km/h, the pure walking time will be approximately 3 hours. Taking into account stops for photos and sightseeing (10-15 minutes per point), the total duration of the excursion will be 6-7 hours, which fits perfectly into a one-day tourist program. Without optimization, a random search for these locations could exceed 15-18 km, which is physically difficult for an unprepared person.

Conclusions. The study confirms that classical graph algorithms can be effectively applied to practical route planning problems in cultural tourism, bridging the gap between theoretical models and real-world navigation. The obtained results demonstrate the feasibility of combining shortest-path algorithms with heuristic methods for both regional and urban navigation tasks.

Case Study: Regional Museum Network.

1. *Algorithmic Efficiency*. The integration of Dijkstra's algorithm with the Nearest Neighbor heuristic successfully generates a complete and mathematically valid tourist cycle, effectively managing missing direct connections by calculating optimal transit paths. The calculated total route distance of 574.50 km demonstrates a reasonable and acceptable deviation from the theoretical lower bound established by the Minimum Spanning Tree (331.4 km). However, while the computational approach is highly efficient for small-scale networks, the Nearest Neighbor heuristic exhibits a "short-sightedness" limitation in the final stages of cycle construction, which can occasionally lead to the selection of longer return edges.

2. *Practical Feasibility*. The proposed model successfully fills the gap between abstract graph theory and practical tourism logistics by using real geographic data and accounting for the actual road infrastructure of the Volyn region. The algorithm effectively accounts for the tree-like topology of regional roads, proving that traversing previously visited nodes (e.g., Luboml and Kolodyazhne) is a practically justified strategy to avoid significantly longer detours. Therefore, the generated route is highly practical for real-world application, offering an optimized, ready-to-use itinerary suitable for car tourism and organized cultural expeditions.

3. *Scalability and Flexibility*. The Python-based implementation offers a flexible tool for recalculating routes based on different starting locations (e.g., Lutsk, Volodymyr, or Luboml). The program's structure allows for easy scaling, meaning new cultural heritage sites can be added to the network without altering the core algorithms. Additionally, the model can be further improved by integrating local search methods (e.g., 2-opt) to resolve path intersections, or by incorporating real-world constraints such as traffic conditions and museum operating hours.

Case study: Cultural Heritage Trail "Lutsk Klikuns".

1. *Algorithmic Efficiency*. The implementation of the Greedy algorithm combined with 2-opt local optimization in Python successfully solved the routing problem for the "Lutsk Klikuns" thematic pedestrian

route. The proposed method reduced the chaotic set of 21 locations into a structured, logistically optimal route with a total length of 11.5 km.

2. *Practical Feasibility*. A key advantage of the study is the use of real pedestrian distances rather than theoretical straight lines (Euclidean distance). This ensures that the generated route is physically feasible for an average tourist within a single-day excursion (approximately 3 - 4 hours of pure walking time). Similarly, for the regional museum network, the approach allowed for the construction of logistically sound loops that minimize travel costs.

3. *Scalability and Flexibility*. The developed software tool (script in Jupyter Notebook environment) is universal. It allows for easy scaling of the model: adding new sculptures, changing the starting point (e.g., from A1 to any other location), or adapting the algorithm for other cities without significant code modifications. This provides a flexible technological foundation for the future development of smart tourism in the Volyn region.

References

1. Applegate D. L., Bixby R. E., Chvatal V., Cook W. J. *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, 2006. 608 p.

2. Cormen T. H., Leiserson C. E., Rivest R. L., Stein C. *Introduction to Algorithms*. MIT Press, 2022. 1312 p.

3. Gavalas D., Konstantopoulos C., Mastakas K., Pantziou G. Mobile recommender systems in tourism. *Journal of Network and Computer Applications*. 2014. Vol. 39. P. 319–333.

4. Gavalas D., Sylla C., Kasapakis V., Pantziou G. Algorithms for the orienteering problem in tourism. *Tourism Management Perspectives*. 2022. Vol. 41. P. 100–115.

5. Harris C. R., Millman K. J., van der Walt S. J. et al. Array programming with NumPy. *Nature*. 2020. Vol. 585. P. 357–362.

6. Helsgaun K. An effective implementation of the Lin–Kernighan traveling salesman heuristic. *European Journal of Operational Research*. 2000. Vol. 126, No. 1. P. 106–130.

7. Liu Y., Wang S., Zhang X. Graph theory applications in smart tourism route optimization. MDPI, 2023. P. 112–128.

8. Rodriguez A., Smith J. Graph-based route optimization for urban cultural heritage sites. *Journal of Heritage Tourism*. 2023. Vol. 18, No. 2. P. 145–160.

9. Toth P., Vigo D. *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics, 2014. 466 p.

10. VanderPlas J. *Python Data Science Handbook: Essential Tools for Working with Data*. O'Reilly Media, 2016. 548 p.

11. Vavdiiuk M. S., Bondaruk O. V. *volyn-route-optimizer*. GitHub, 2026. [Online]. Available: https://github.com/4marchello/volyn-route-optimizer.

12. Vavdiiuk M. S., Osadchyi A. S. *lutsk-route-optimizer*. GitHub, 2026. [Online]. Available: https://github.com/4marchello/lutsk-route-optimizer.

Reviewers

1. **Syasky Andriy** - Doctor of Technical Sciences, Professor of the Department of Applied Mathematics and Mechanics at Lutsk National Technical University.

2. **Mikulich Olena** - Doctor of Technical Sciences, Professor of the Department of Applied Mathematics and Mechanics at Lutsk National Technical University.