UDC 004.94

DEVELOPMENT OF INNOVATIVE APPROACHES FOR NETWORK OPTIMIZATION USING GEOSPATIAL MULTI-COMPONENT SYSTEMS

Boyko N. I. – PhD, Associate Professor, Associate Professor of the Department of Artificial Intelligence Systems, Lviv Polytechnic National University, Lviv, Ukraine.

Salanchii T. O. – Student, Department of Artificial Intelligence Systems, Lviv Polytechnic National University, Lviv, Ukraine.

ABSTRACT

Context. Developing a geospatial multi-agent system for optimizing transportation networks is crucial for enhancing efficiency and reducing travel time. This involves employing optimization algorithms and simulating agent behavior within the network.

Objective. The aim of this study is to develop a geospatial multi-agent system for optimizing transportation networks, focusing on improving network efficiency and minimizing travel time through the application of advanced optimization algorithms and agent-based modeling.

Method. The proposed method for optimizing transportation networks combines foundational structure with advanced refinement in two stages: pre-processing and evolutionary strategy optimization. In the first stage, a Minimum Spanning Tree is constructed using Kruskal's algorithm to establish the shortest, loop-free network that connects all key points, accounting for natural obstacles and existing routes. This provides a cost-effective and realistic baseline. The second stage refines the network through an evolutionary strategy, where agents representing MST variations are optimized using a fitness function balancing total path length, average node distances, and penalties for excessive edges. Optimization employs crossover to combine solutions and mutation to introduce diversity through edge modifications. Repeated over multiple epochs, this process incrementally improves the network, resulting in an optimized design that minimizes costs, enhances connectivity, and respects real-world constraints.

Results. The results of applying the evolutionary strategy and minimum spanning tree methods were analyzed in detail. Comparing these methods to benchmarks like Tokyo's railway network and the Slime Mold algorithm revealed the advantage of using the evolutionary approach in generating optimal paths. The findings emphasize the need for integrating advanced algorithms to further refine path optimization and network design.

Conclusions. The research successfully developed a geospatial multi-agent system for optimizing transportation networks, achieving its objectives by addressing key challenges in transport network planning. A detailed analysis of existing solutions revealed the dynamic and complex nature of transportation systems and underscored the need for adaptability to environmental changes, such as new routes or obstacles. The proposed approach enhanced the minimum spanning tree with an evolutionary strategy, enabling flexibility and rapid adaptation. Results demonstrated the system's effectiveness in planning optimal intercity transport networks. Future work could refine environmental assessments, improve route cost evaluations, expand metrics, define new performance criteria, and integrate neural network models to further enhance optimization capabilities, particularly for urban networks.

KEYWORDS: geospatial multi-agent system, optimization of transportation networks, evolution strategy.

ABBREVIATIONS

MST is a minimum spanning tree;
CS is a capital particle;
TF is a target food;
ES is an Evolution Strategies.

NOMENCLATURE

$E' \subseteq E$ is the condition for satisfying the goal
$dG'(u,v)$ is the shortest path distance between nodes u and v in G';
λ is a penalty weight;
$d(e)$ is the length of the edge;
$l(p_i, p_j)$ is the shortest route between points p_i and p_j within the network;
S is the set of all shortest paths between all points;
w is the weighting factor that controls the importance;
L is the total graph distance;
D_{avr} is the average value of the minimum distances between any pair of points in the network;
S is the penalty;
$T(\nu)$ is the number of triangles that include the vertex ν;
k_ν is the number of neighbors of the vertex;
$N_{triangles}$ is the number of triangles in the graph;
N_{groups} is the number of groups of size 3.

INTRODUCTION

The optimization of transportation networks is increasingly crucial as cities grow and demand for efficient systems rises. A geospatial multi-agent system offers a promising solution to streamline these networks, improving efficiency and reducing travel times. Research has made progress in optimizing transportation, but challenges remain. Traditional methods struggle with the complexity of real-world systems, where factors like traffic density, geography, and transport types interact. Data preprocessing is another hurdle. Integrating geospatial information from various sources requires significant effort, delaying implementation and reducing effectiveness. Another key issue is adaptability. Transportation networks are dynamic, constantly influenced by infrastructure changes and shifting traffic

DOI 10.15588/1607-3274-2025-2-16

patterns. An effective system must adapt quickly to these changes. However, many theoretical models fail to translate into practical, operational systems, limiting their real-world application. This work proposes an evolutionary strategy to transform the MST into an optimized network. The algorithm's effectiveness was tested against the Slime Mold algorithm [23] and Tokyo's railway system, showing promising results. Though the approach holds potential, more research is needed to refine the system's adaptability, streamline data processing, and enhance scalability for broader use.

Object of the study: interaction process and dynamics of vehicle movement within transportation networks, considering the topology of urban areas and optimal routes.

Subject of the study: algorithms for analyzing and optimizing resource allocation in geospatial multi-agent systems aimed at improving the efficiency of transportation networks in cities.

The aim of this study is to develop a geospatial multi-agent system that optimizes transportation networks by addressing key challenges such as data preprocessing, adaptability, and complexity. By applying evolutionary strategies to transform the minimum spanning tree into an optimal network, this research seeks to improve transportation efficiency and flexibility. The findings aim to contribute to the development of more effective and scalable solutions for real-world transportation systems.

Tasks of the research:

– Analyze existing methods for optimizing transportation networks and geospatial systems.

– Develop a new multi-agent system algorithm that accounts for the geospatial features of transportation networks and agent capabilities for movement optimization.

– Model and simulate the system using test data to assess its effectiveness and reliability.

– Analyze the results and compare them with existing transportation optimization methods to confirm the advantages of the developed system.

The advantages of modeling transportation network optimization lie in the ability to predict traffic flow, identify critical congestion points, and propose effective strategies to improve overall system efficiency. Such models can account for various factors, including traffic density, infrastructure, and agent behavior, allowing for a deeper understanding of the dynamic interactions within transportation systems. They are invaluable tools for decision-making, offering insights that can guide the design and implementation of smarter, more efficient transport solutions.

The relevance of this study is explained in the growing need to optimize transportation networks due to increasing urbanization, traffic congestion, and environmental concerns. As transportation systems evolve, traditional methods of management become less effective, highlighting the importance of developing innovative approaches like geospatial multi-agent systems. This research is timely, as it addresses both the practical need for improved traffic management and the scientific challenge of simulating complex, real-world transportation dynamics.

Understanding and optimizing transportation networks is critical not only for reducing travel time but also for minimizing environmental impact and for enhancing public well-being. The outcomes of this research will have direct implications for urban planning, traffic management, and sustainability efforts, contributing to the development of smarter cities with more efficient, adaptive, and environmentally friendly transportation systems.

The focus of the research is on identifying the relationships between various elements of urban transportation networks, such as traffic patterns, infrastructure, and agent behavior, as well as determining the factors that contribute to inefficiencies or congestion in these systems. The object and subject of the research reflect the core aspects investigated within this study to optimize transportation networks and enhance their efficiency in urban environments.

This research is significant as it aims to enhance the optimization of transportation networks, a vital element of modern infrastructure that facilitates the efficient movement of people and goods. Addressing the challenges associated with these networks is essential for promoting economic growth and safeguarding public health. This study proposes the development of a geospatial multi-agent system as a new solution for transportation network optimization. Such systems enable the modeling of complex agent interactions while accounting for the unique characteristics of each network.

Transport networks are inherently dynamic and complex, making traditional management methods increasingly ineffective. The application of geospatial multi-agent systems is particularly relevant in this context, as they offer the ability to simulate and analyze multiple factors simultaneously. For instance, traffic density, diverse modes of transportation, and geographic constraints can all be incorporated into the system's framework.

Despite notable progress in the field, as highlighted in recent studies [5, 7], significant challenges remain. One major issue is the high complexity involved in system development, stemming from the need to integrate numerous variables and adapt to real-world conditions. Additionally, substantial efforts are required for data preprocessing, including the aggregation and transformation of geospatial data from various sources. This often results in inefficiencies and complicates the practical implementation of these systems.

Another key challenge is the adaptability of the system. Transportation networks are constantly evolving - new roads are built, traffic patterns change, and weather affects old roads. Thus, the system must be flexible enough to respond swiftly to these changes. Furthermore, many studies tend to focus on theoretical frameworks

DOI 10.15588/1607-3274-2025-2-16

without addressing practical considerations, which limits their applicability in real-world scenarios.

To address these challenges, this research introduces an evolutionary strategy to transform a minimal spanning tree into an optimized transportation system. The proposed approach is evaluated by comparing its performance to the Slime Mold algorithm [23] and Tokyo's railway network, demonstrating its potential as an effective and innovative solution. While the development of such systems presents several hurdles, it holds immense promise for creating adaptable, efficient, and practical transportation networks.

1 PROBLEM STATEMENT

The problem of optimizing a transportation network can be described using graph theory, where the network is represented as a graph $G=(V,E)$. Here, V denotes the set of nodes (e.g., cities or junctions), and E represents the set of edges (e.g., potential routes) with weights $w(e)$, which correspond to costs such as distance, time, or construction expenses.

The goal is to determine a subgraph $G'=(V,E')$ that satisfies several objectives:

1. Minimizing Total Path Length: the network should have the lowest possible total cost, calculated as the sum of weights of all selected edges in E.

$$\min(\sum_{e\in E'} w(e)).$$

2. Ensure Connectivity: G' must form a connected subgraph such that every pair of nodes $(u,v)\in V$ is reachable.

3. Optimize Average Shortest Path Length: Minimize the average shortest path length between all pairs of nodes:

$$\min(\frac{1}{|V|(|V|-1)}\sum_{(u,v)\in V} dG'(u,v).$$

4. Restrict Excessive Edge Addition: Impose a penalty P for adding extra edges beyond a defined threshold k:

$$P(E')=\lambda * \max(0,|E'|-|V|+k).$$

These goals are expressed in a single optimization function that balances the total path length, connectivity efficiency, and penalties for excessive edges. Additionally, constraints are applied to ensure the network respects real-world factors such as geographical obstacles, infrastructural limitations, and dynamic changes in environmental conditions.

This formulation allows for the adaptive and efficient design of transportation networks under dynamic and real-world constraints.

2 LITERATURE REVIEW

Optimization methods are critical in addressing modern challenges in network systems, transportation, and urban planning. This summary highlights the most relevant recent studies on optimal transport and network optimization, focusing on their practical applications, advancements, and limitations.

In research [2], "Imitation-regularized optimal transport on networks: provable robustness and application to logistics planning", the authors address disruptions in network systems with a method called Simulation-Regularized Optimal Transport (I-OT). This approach enhances system resilience and provides practical applications in logistics planning using real-world data. However, the study assumes that networks are Markovian, a simplification that might not always hold true. Additionally, it does not examine the stability of I-OT solutions concerning Schrödinger's bridge problems, nor does it compare I-OT with other transport planning methods, leaving questions about its relative performance unanswered.

Study [3], "Heuristic Optimal Transport in Branching Networks" introduces an efficient heuristic algorithm for large-scale transport problems. This algorithm reduces computation time significantly, adapts well to various network topologies, including those with multiple sources, and sinks. Nonetheless, the reliance on heuristic approximations can compromise solution accuracy, creating a trade-off between speed and precision. Furthermore, the method may struggle with transport tasks involving nonlinear cost functions or complex constraints, limiting its applicability.

Research [4], "Optimal intervention in traffic networks" proposes a topological optimization-based method for route planning in construction projects, focusing on reducing congestion and improving real-time infrastructure planning. The study demonstrates its efficiency through use cases but highlights a critical limitation: the high computational resources required for the optimization process. This drawback could result in complicated practical adoption, especially in scenarios with limited computational infrastructure.

In article [5], "Network centrality guided multi-objective particle swarm optimization for transport optimization on networks", the authors present a multi-objective particle swarm optimization algorithm incorporating Gaussian mutation to balance exploration and exploitation effectively. The algorithm performs well in achieving convergence to Pareto fronts and identifying optimal solutions. However, its high computational complexity raises concerns for large-scale applications. Additionally, the lack of comprehensive comparisons with other state-of-the-art methods and real-world validations limits its generalizability and practical relevance.

These studies illustrate the advancements and trade-offs in applying optimization techniques to real-world problems. While promising, future research must address

DOI 10.15588/1607-3274-2025-2-16

scalability, robustness, and broader applicability to realize their full potential.

Table 1 provides research analysis of related publications that were selected from few sources: Scopus and ArXiv.

Table 1 – Review of related papers

Link	Methodolo-gy	Pros of the methodology	Cons of the methodology
[2]	Simulation-regularized optimal transport	Enhances resilience of network systems; practical in logistics planning with real-world data	Assumes Markovian networks (not always valid); lacks comparative analysis with other methods; limited stability insights
[3]	Heuristic algorithm for branching networks	Efficient for large-scale applications; adaptable to varied topologies	Compromises accuracy for speed; less effective for nonlinear costs or complex constraints
[4]	Topological optimization for real-time planning	Reduces congestion; effective in infrastructure development	Requires significant computational resources, limiting practicality in resource-constrained scenarios
[5]	Particle swarm with Gaussian mutation	Balances exploration and exploitation; efficient Pareto convergence	High computational complexity; lacks broader comparisons and real-world validation

Summing up the results presented in Table 1, the analysis highlights the significance of optimizing transportation networks and logistics systems using innovative methodologies. The reviewed studies emphasize the relevance of resilient and efficient approaches to address modern challenges in network planning. While the methodologies demonstrate notable advancements in efficiency, adaptability, and real-world applicability, certain limitations persist, such as computational complexity, reliance on specific assumptions, and the need for comprehensive comparative analyses. These findings underscore the importance of further research to refine these methods and enhance their scalability and robustness for practical applications.

3 MATERIALS AND METHODS

This study explores a promising approach to optimizing transportation networks using bio-inspired algorithms and geospatial data. The focus is on developing efficient, resilient, and adaptive systems capable of addressing the complexities of urban environments. To evaluate the proposed method, the Slime Mold Algorithm [23] was employed as a benchmark due to its ability to model self-organizing network behavior.

To test these methodologies, we used a modified dataset derived from the open-source Slime Mould project [23]. This dataset includes geographic coordinates (latitude and longitude) and Cartesian coordinates (x, y) of Tokyo subway stations, formatted in GeoJSON. For convenience, the data was converted into a pandas DataFrame, facilitating integration with analytical tools and evolutionary algorithms. Converted dataset contains the geographic coordinates (latitude, longitude) and coordinates (x, y) of subway stations, enabling efficient modeling and analysis of Tokyo's transportation infrastructure. An example of the transformed data is displayed on Table 2.

Table 2 – Example of a modified data set

node	lon	lat	*x*	*y*
0	118, 86	32, 04	201	163
1	118, 78	32, 05	116	174
2	118, 98	32, 09	320	208
4	118, 79	32, 04	127	166

A map of the subway system, with stations marked as nodes and connections represented as edges is provided on Figure 1.

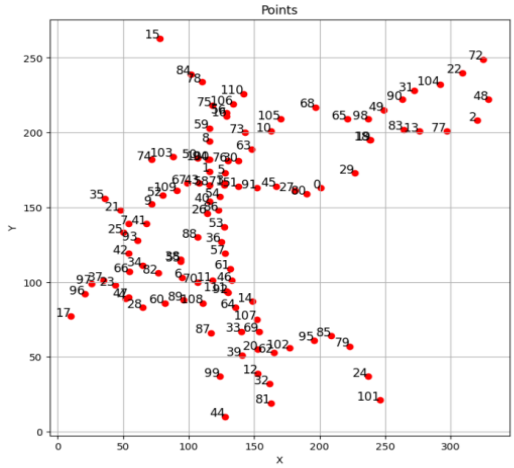


Figure 1 – Map of subway station locations in Tokyo

The slime mold algorithm is inspired by the natural behavior of Physarum polycephalum and simulates the organic growth of transport networks. This method mimics how slime molds optimize paths to connect resources efficiently. By balancing efficiency and resilience, the algorithm generates adaptive road networks capable of withstanding environmental or infrastructural changes.

The networks designed using the slime algorithm are characterized by their speed and cost-effectiveness, as well as their ability to self-regulate. This makes them particularly well-suited for dynamic environments that require ongoing optimization.

When it comes to road network design, the slime algorithm excels in achieving a balance between efficiency and resilience. It prioritizes minimizing distances while also ensuring that the network can withstand failures and adapt to shifts in the environment or infrastructure.

As a result, these networks are not just fast and economical; they also possess the capability to self-regulate, even in adverse conditions. This unique combination of features makes the slime algorithm a powerful tool for creating robust road networks.

DOI 10.15588/1607-3274-2025-2-16

Utilizing the self-organizing principles found in slime mold, this algorithm facilitates the development of transportation networks that can adjust to evolving conditions, such as rising traffic levels or new construction initiatives. This adaptability makes it particularly suitable for dynamic environments that require ongoing enhancement and optimization of transport infrastructure [17].

In this study, the slime algorithm [23] was employed to assess and compare the effectiveness of the proposed method. Therefore, it is pertinent to provide a brief overview of how the algorithm operates.

Slime simulation involves several factors. According to the literature, when Physarum is placed in a medium, such as a petri dish filled with nutrients like oatmeal, it creates a network of protoplasmic tubes to link all available food sources. In this context, the slime sample, its environment, and the nutrients present are the primary elements to consider when modeling the nutrient transport system. The slime model presented in [23] utilizes agent-based modeling to replicate the decision-making processes involved in the movement of the slime sample. This model consists of four essential components: the grid, the food sources, the slime itself, and the slime particles or agents.

A grid serves as the foundational structure that represents the environment for slime modeling. Each element of the grid is initialized as a cell, creating a comprehensive framework for the simulation. This grid consists of two distinct informational layers: the first layer is dedicated to nutrients and slime, while the second layer is composed of pheromones that play a crucial role in guiding the movement of the slime.

In this modeling approach, nutrients are represented as food sources (FS). Each food source is initialized with the highest constant pheromone value in the second layer of the grid, ensuring that the slime can effectively locate and connect to these resources. The primary objective of the slime is to cover and connect all food sources throughout the entire grid, thereby optimizing nutrient transport.

The slime is represented as a "population" of Physarum cells or agents. It maintains various global states that assist each agent, or slime particle, in making movement decisions based on the second layer of the grid. A key component in this process is the capital particle (CS), which designates the target food (TF) for all agents. The CS is randomly chosen from slime particles positioned at the four corners of the slime's "covered area". Once the CS is selected, it seeks out the nearest unconnected food source, which becomes the TF until it is linked.

Since the slime's ultimate aim is to capture and connect all food sources on the grid, the CS is dynamically updated in real-time. Additionally, the slime continuously replicates agents (slime particles) to move toward the TF. This ongoing adaptation is referred to as the evolutionary process. To illustrate this process clearly, a map of food sources (FS) is provided, along with a pheromone map that reflects the number of epochs, as shown in Figure 2.


Figure 2 – Pheromone distribution map across epochs

Each slime particle functions as an individual "agent" within the structure of the slime mold. These particles possess local states that enable them to make decisions regarding their movement. To navigate their environment effectively, a slime particle typically goes through two primary phases: the sensory phase and the diffusion phase. During the sensory phase, a slime particle identifies a food path (FP), which represents the shortest route from the nearest connected food source to the target food (TF). Each food source along this path is referred to as a step food (SF) leading to the TF. Once the FP is established, the agent transitions into the diffusion phase. In this stage, it expands its reach while assessing the conditions of adjacent cells. A schematic illustration of this process can be found in Figure 3.

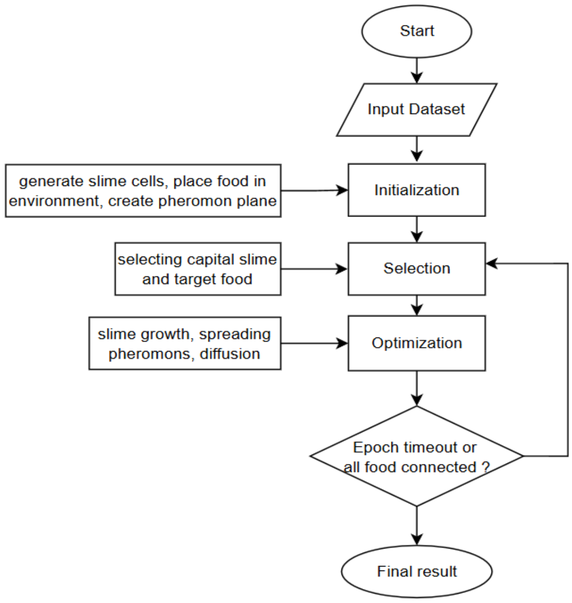


Figure 3 – Schematic representation of Slime Mould algorithm

The diffusion process contains six key features:

1. The pheromone level of a slime particle decreases with each diffusion step.

2. Diffusion is directed toward the nearest step food (SF).

3. If sufficient pheromone is present, diffusion can extend to other neighboring cells.

4. A slime particle will refrain from spreading if it is too distant from the nearest connected food source.

5. During the diffusion stage, a new replicated slime particle may be generated, potentially exhibiting mutated traits such as altered movement direction or pheromone emission.

6. Different conditions of neighboring cells will influence how the pheromone level of the slime particle changes.

DOI 10.15588/1607-3274-2025-2-16

As a conclusion – at each stage, the algorithm improves the system using the physical properties of slime. Pheromones allow the algorithm to estimate the value of each segment: the higher the value of pheromones, the higher the probability that the segment will remain in the final decision. Diffusion, on the other hand, helps to “dry out” suboptimal segments, eliminating routes that do not contribute to efficient communication between stations.

Evolution Strategies represent a method for addressing optimization problems, drawing inspiration from the principles of natural selection and evolution. Similar to other evolutionary computation techniques, this approach utilizes a population of potential solutions that are iteratively improved through mutation, selection, and crossover processes. In this context, each decision is represented as an “individual” or “agent” characterized by a specific set of parameters that define its attributes. The primary goal of evolutionary strategies is to refine these parameters, ultimately identifying the most effective options through continuous improvement.

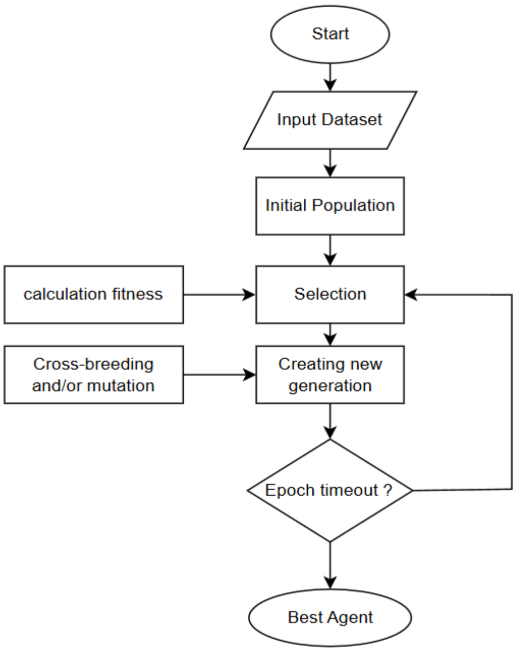


Figure 4 – Schematic representation of Evolution Strategy

One notable characteristic of evolutionary strategies is their capacity to adjust to intricate, multidimensional settings that involve numerous variables. These strategies can be utilized across various fields, particularly in optimization tasks related to technical systems, economics, and biological applications.

A fundamental component of evolutionary strategies is the mutation process, which creates new potential solutions by altering the parameters of existing individuals. Additionally, the selection process plays a crucial role, as it guarantees that only the most effective individuals contribute their traits to the subsequent generation. This combination of mutation and selection fosters continuous improvement in the search for optimal solutions.

This approach is widely used in designing complex systems, such as road networks and communication infrastructures, where maintaining a balance between efficiency and sustainability is essential. In such systems, changes in the environment, resource constraints, and unpredictable disruptions often pose significant challenges. Evolutionary strategies are particularly valuable in these cases, as they can generate solutions that remain effective despite fluctuating conditions. Their adaptability makes them well suited for optimizing dynamic systems that require resilience and long-term viability.

The developed transport network optimization method consists of two stages:

1. Input data is pre-processed: using the Kruskal algorithm, a minimum spanning tree (MST) is created, which forms a basic network with the minimum total length of paths between key points (for example, cities).

2. Using evolution strategy to improve MST by adding “usefull” segments to enhance connectivity, reduce travel time, and increase network resilience while maintaining cost efficiency.

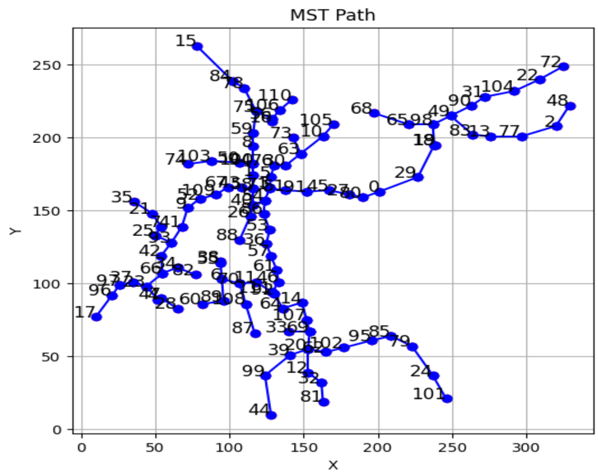


Figure 5 – Example of MST using Kruskal’s method

Kruskal’s algorithm is a well-known greedy algorithm designed to find a minimum spanning tree (MST) within a weighted graph. This algorithm operates on the principle of incrementally adding edges that possess the least weight, all while ensuring that no loops are formed. The process continues until every nodes in the graph is connected, resulting in a robust structure. Schematic representation of algorithm is on Fig. 6.

By utilizing the minimum spanning tree, we can establish a network that guarantees the connectivity of all points at the lowest possible cost. This foundational network serves as a solid base for future additions to the transport infrastructure. As new edges are added, the efficiency of the network can be significantly improved, paving the way for a more effective and responsive transportation system. In essence, Kruskal’s algorithm not only provides a solution to the problem of connectivity but also lays the foundation for ongoing development in the next stage.

DOI 10.15588/1607-3274-2025-2-16

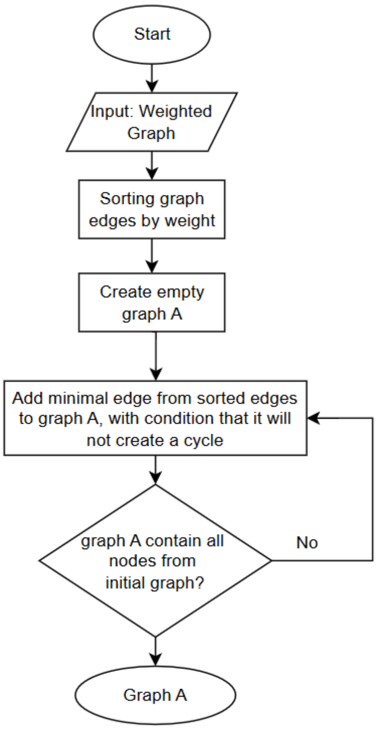


Figure 6 – Schematic representation of Kruskal algorithm

The second stage of the method uses an evolutionary strategy to gradually improve the network structure. It begins with forming an initial population of agents, where each agent represents a set of additional edges added to the minimal spanning tree. These extra edges help refine the network, making it more efficient.

Next, each agent is scored using a fitness function that considers several factors. Let us have a set of vertices $P = \{p_1, p_2, p_3 ... p_n\}$, representing key nodes in the network, and a list of edges E, representing the paths between points in the agent. To form an optimal network, we can set the following fitness function f, that considers listed metrics:

1. Total Graph Length: Total sum of edges, used to reduce the cost of building new roads.

For instance, let $L(E)$ be the total length of all edges in the set E. This metric can be represented into the fitness function (1):

$$L = \sum_{e \in E} d(e), \tag{1}$$

the goal is to find a combination E that L is minimal while fulfilling other conditions.

Average distance between any pair of points: Minimizing this value ensures that the points in the network are not too far apart, improving overall efficiency. Let D_{avr} represent the average of the minimum distances between any pair of points in the network (2):

$$D_{avr} = \frac{1}{|S|} \sum_{p_i, p_j \in P} l(p_i, p_j), \tag{2}$$

is a formalized representation of the definition of the set of all shortest paths between all points.

2. Extra Edge Penalty: To prevent excessive network complexity, the algorithm imposes a penalty when the number of extra edges surpasses a predefined threshold. This threshold is set as a permissible ratio of the total number of points—for instance, 1.1, which allows up to 10% additional edges. When this limit is exceeded, a penalty is introduced (3):

$$S = \begin{cases} \lambda * (|E| - \alpha * |P|), if\ |E| > \alpha * |P| \\ 0 \end{cases}, \tag{3}$$

in the experiments performed $\lambda = 0.01$.

Then the final fitness function can be expressed as (4):

$$F = w * L + D_{avg} + S\ . \tag{4}$$

During the third step, agents undergo changes via crossover and mutation. Crossover in this method is performed by selecting common additional edges that are present in two parent agents. This strategic choice is designed to preserve the fundamental structure of each solution, ensuring that the strengths of both parents are retained in the child.

Following this, the algorithm introduces an element of diversity and potential for further optimization. This is achieved by randomly selecting unique edges from each parent and incorporating them into the shared edges. To illustrate, consider a scenario where one agent possesses four extra edges while another has six, with three of those edges being common to both. In this case, the new agent will inherit the three common edges, but it will also receive three randomly chosen unique edges from each parent. This approach guarantees that the total number of additional segments in the offspring matches the maximum number found in either parent. This process is visually represented in Figure 7.

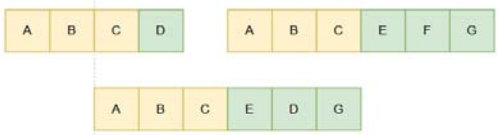


Figure 7 – Crossover operation

By employing this method, the algorithm effectively maintains a stable foundation derived from both parent agents. At the same time, it explores new possibilities by integrating unique elements. This dual approach not only preserves the valuable characteristics of the parent solutions but also enhances the search for optimal solutions, making the overall process efficient and robust.

Despite the effectiveness of this approach, a complete mutation remains essential. Without it, there is a risk that parents sharing the same genome may stagnate in their evolutionary journey. In this method, mutation is executed through a random selection among three distinct actions: removing, replacing, or adding an edge.

DOI 10.15588/1607-3274-2025-2-16

In cases where addition is the chosen action, a new edge is introduced to the list of additional edges. This new edge is selected from those that have not yet been utilized. However, it is important to note that this addition can only occur if the total number of additional edges does not surpass the predetermined maximum limit.

When the deletion option is chosen, one edge is randomly eliminated from the agent's set of additional edges. This action not only simplifies the path but also encourages the exploration of more streamlined solutions.

On the other hand, if the replacement option is selected, a randomly chosen additional edge is substituted with a new one. This new edge is drawn from the pool of available edges, deliberately excluding those already incorporated into the agent's structure. This strategy ensures variability while maintaining a constant total number of additional edges.

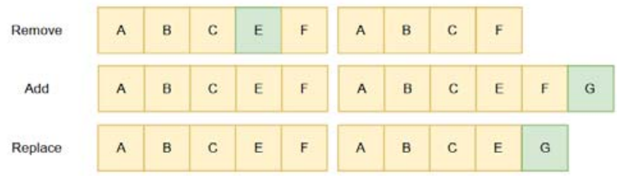


Figure 8 – Visualization of mutation types

This implementation of mutation strategy is designed to maintain balance of solution stability and exploration, ensuring that the agent maintains path connectivity and optimality while simultaneously discovering new possible solutions.

The fourth step of the algorithm involves iterative improving through evaluation, crossover. This process is repeated for a predetermined number of epochs, allowing the system to gradually enhance the quality of solutions.

With each iteration, the population of agents undergoes continuous improvement, driven by the selection of the most promising candidates. Over time, this iterative process minimizes the gap between the best and worst performing agents, moving toward an optimal solution.

As the algorithm progresses (See Fig. 9), weaker solutions are gradually eliminated, while stronger ones propagate, ensuring that each generation is more refined than the last. By the final epoch, the selection process has filtered the population to its most effective configuration. Ultimately, the agent exhibiting the highest performance is chosen as the best solution.

The clustering coefficient is a metric that measures the likelihood that neighbors of a node in a graph are also connected to each other, forming a triangle. This coefficient indicates the local density of connections for each node and helps to understand the structure of the graph at a local level. Figures 10 (a) and 10 (b) illustrate the differences in the structure of graphs with different clustering coefficients: in a graph with high clustering.

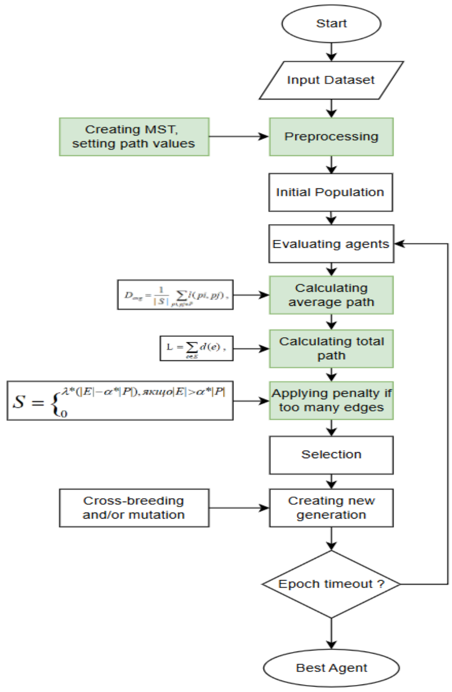


Figure 9 – Schematic representation of proposed algorithm

a b

Figure 10 – Example of graphs with different clustering coefficient

The clustering coefficient for an individual node is defined as the ratio of the number of existing connections between its neighbors to the maximum possible number of such connections. The local clustering coefficient for a vertex indicates how much its neighbors are also neighbors with each other. It is defined as follows (5):

$$C_{local}(V) = \frac{2T(v)}{k_v(k_v - 1)}. \quad (5)$$

The higher the clustering coefficient, the more the node is part of a tightly connected group where its neighbors also interact closely with one another.

High values of the clustering coefficient indicate that the graph tends to form strongly connected local groups, or "clusters". The clustering coefficient for a graph (global) is given by (6):

$$C_{global} = \frac{3N_{triangles}}{N_{groups}}, \quad (6)$$

a triangle in a graph is a group of three vertices where each one is connected to the other two, while groups of three vertices are simply all possible triplets of vertices, regardless of whether they are connected.

The clustering coefficient for the entire graph allows for the assessment of the overall tendency to form such

DOI 10.15588/1607-3274-2025-2-16

local clusters. For a random graph, this coefficient is always equal to 0, as nodes are only connected to nodes with which they have the shortest paths, without forming triangles, which are the basis for clustering. For an agent, the clustering coefficient is equal to 0, meaning that the agent maintains separate lines of communication but does not engage in close connections with other nodes.

In the process of solving optimization problems, it is important to consider and compare several approaches to achieve effective results. This study examines two methods: the slime mold method, which simulates the behavior of slime mold in finding the optimal path between food sources, and the evolutionary strategy, which is based on improving the MST. Both methods have distinct mechanisms for search, self-organization, and adaptation, which define their unique advantages and limitations. Comparative Tables 3–6 presents an analysis of the stages of these methods, from initialization to optimization. The goal of this comparison is to identify the most suitable method for solving optimization problems under the given conditions.

Table 3 – Initialization stage comparison

Initialization	
Slime Mould	Slime particles are created, and a grid of food and pheromones is generated
Evolution Strategy	Combinations of all possible paths between vertices are created, and an initial population of agents is formed, where each agent is a list of paths.
Evolution strategy based on MST improvement (Proposed method)	MST is generated, combinations of all possible paths between vertices are created, and an initial population of agents is formed, where each agent is an addition to the MST.

Table 4 – Search stage comparison

Search	
Slime Mould	A leader particle is selected to indicate the general direction for others; agents move and leave pheromone hints for other particles.
Evolution Strategy	Each agent is evaluated through a fitness function.
Evolution strategy based on MST improvement (Proposed method)	Each agent is evaluated through a fitness function; the average path, total path is calculated, and a penalty is added if the number of paths exceeds the allowed limit.

Table 5 – Self organizing stage comparison

Self-organizing	
Slime Mould	Using the diffusion process, the slime improves the system.
Evolution Strategy	Agents are sorted by fitness value (agents that could not build a connected path/graph are excluded).
Evolution strategy based on MST improvement (Proposed method)	Agents are sorted by fitness value.

Table 6 – Information processing stage comparison

Information Processing	
Slime Mould	Due to the physical properties of pheromones, non-optimal segments are "dried up" and important segments remain.
Evolution Strategy	The top 50% of the best-fitted agents are selected. Agents randomly form pairs that undergo classical crossover (with the preservation of graph connectivity) with a chance of random mutation.
Evolution strategy based on MST improvement (Proposed method)	The top 50% of the best-fitted agents are selected. Agents randomly form pairs that undergo crossover with a chance of random mutation. This allows for the reproduction of the initial population size while maintaining fitness and diversity.

Based on the analysis of the stages of work of each method, certain conclusions can be drawn regarding their effectiveness. The evolutionary strategy based on improving the MST has proven to be more adaptable for situations where flexibility and the ability to adapt are important, as its mechanisms of crossover and mutation allow for the preservation of a diversity of solutions within the population of agents. This strategy promotes a higher adaptability of agents to changing conditions, which positively affects the effectiveness of optimization in complex networks.

The slime method also has advantages in terms of the speed of self-organization due to the physical properties of pheromones, which allow for the identification of important segments and the "drying out" of non-optimal ones. Therefore, for tasks that require a quick convergence to an optimal solution, the slime method is also appropriate. Given the advantages of the evolutionary strategy for adaptive environments, it is advisable to use it as the primary method for further experiments.

4 EXPERIMENTS

Conducting experiments is of great importance, as it allows us to evaluate how effectively the system improves transportation efficiency and reduces travel time. These experiments measure the performance of the optimization algorithms and assess their ability to handle real-world challenges such as congestion and disruptions. Moreover, the results help validate the evolutionary strategy's transformation of minimum cost distance into optimal paths while extending the minimal spanning tree for practical applications, offering valuable insights into network optimization.

To implement the evolutionary algorithm system, Python was chosen as the primary programming language due to its versatility, readability, and extensive ecosystem of libraries. Python is widely recognized for its simplicity and cross-platform compatibility, making it an ideal choice for developing diverse software solutions. In particular, the following libraries and tools were used to develop the application:

– NetworkX: This library specializes in analyzing and visualizing complex networks and graphs. It was used for constructing minimum spanning trees, checking graph connectivity, and visualizing transportation networks.

DOI 10.15588/1607-3274-2025-2-16

– NumPy is a library for scientific computing in Python. Essential for numerical computations, NumPy excels at handling large, multi-dimensional arrays with performance optimizations through C-based implementation.

– Pandas: Known for its robust data manipulation capabilities, Pandas offers the DataFrame structure, which facilitates efficient data cleaning, filtering, grouping, and aggregation.

– Matplotlib & Seaborn: These libraries provided tools for creating high-quality visualizations, from basic plots to more advanced data representations.

This section focuses on identifying the optimal parameters for an evolutionary strategy, which are critical for ensuring the algorithm's efficiency and performance. Specifically, we examine key parameters such as the number of epochs and population size. These parameters significantly influence both the quality of the solutions obtained and the convergence speed of the algorithm.

The number of epochs determines the iterations of the optimization process. A higher number of epochs allows the algorithm to fine-tune its search for the best solutions. However, it also increases the execution time. Striking a balance between learning efficiency and execution speed is essential for parameter optimization.

Population Size Population size directly impacts the diversity of solutions explored during optimization. Larger populations offer more options for evolutionary selection, increasing the likelihood of finding a global optimum. On the other hand, excessively large populations can complicate the optimization process and increase computational costs.

To identify the optimal algorithm parameters, a series of experiments were conducted. Two primary metrics were used for evaluation:

– Execution time: How long the method takes to complete.

– Fitness of the best agent: Calculated using the objective function described earlier.

The goal was to balance execution speed and result accuracy, maximizing system resource efficiency.

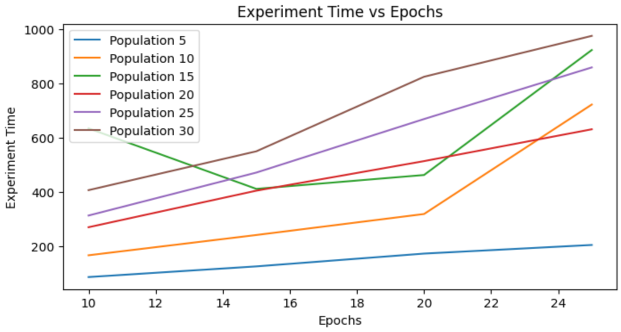


Figure 11 – Relationship between execution time and the number of epochs for different population sizes

Analysis reveals that increasing both population size and the number of epochs gradually extends the execution time. This is because larger populations require more processing time.

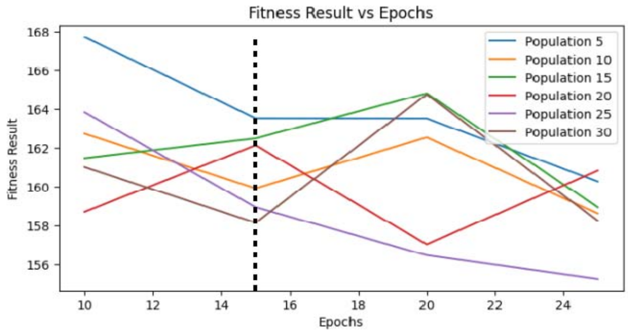


Figure 12 – Relationship between fitness results and the number of epochs for different population sizes

The Fig. 11–12 shows that increasing epochs does not always lead to consistent fitness improvements. For example, populations of 15 and 20 exhibit instability in later stages.

Based on the experiments, the optimal parameter combination is a population size of 25 agents and 15 epochs. This choice is supported by a clear trend of fitness improvement with increasing epochs, as observed in the fitness-epochs graph. This balance ensures efficient resource utilization while maintaining high solution quality.

Using the optimal parameters (25 agents and 15 epochs), a comparison was conducted to evaluate the efficiency of the developed method against the slime mold algorithm. For the slime mold algorithm, 350 epochs were used, as this provided sufficient time for the agents to grow and form optimal paths. The Tokyo subway system was also included as a benchmark example for comparison.

The evaluation was based on the criterion of the average distance between each pair of points (2).

This approach allowed us to determine how well the optimal paths identified by the algorithms align with real-world transportation routes and whether they could provide more efficient connections between stations compared to the existing subway network. MST is displayed on Figure 13. The calculations and results are presented in Tables 7–9. For better visualization, each table includes comparison between MST and generated system (red edges are common for MST and generated system, green are present only in generated system).

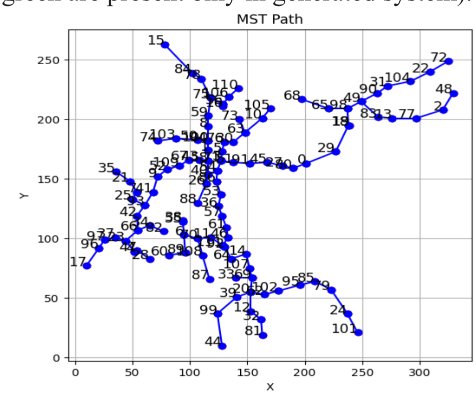


Figure 13 – Generated MST

© Boyko N. I., Salanchii T. O., 2025
DOI 10.15588/1607-3274-2025-2-16

Table 7 – Calculation and analysis of Tokyo railway system

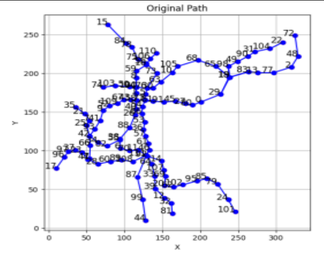
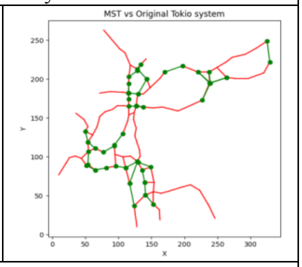
Tokyo railway	
	
Average distance	$AVG = \frac{1}{\lvert S \rvert} \sum_{a \in S} a = \frac{1971289.6}{12544} = 157.15$
Total distance	$L = \sum_{t \in E} t = 1738.6$
Clustering coefficient	$C_{global} = \frac{3N_{triangles}}{N_{groups}} = 0$ as there is no triangle in the system

Table 8 – Calculation and analysis of Slime Mould system

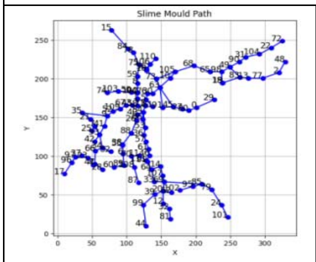
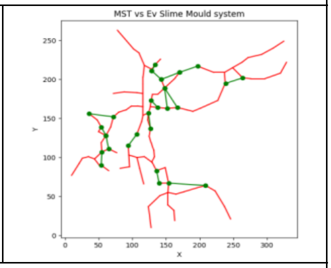
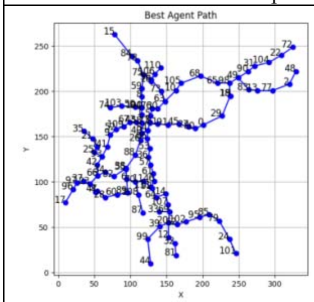
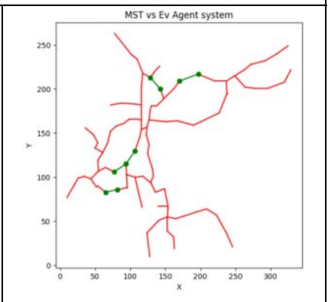
Slime Mould method	
	
Average distance	$AVG = \frac{1}{\lvert S \rvert} \sum_{a \in S} a = \frac{2488781.6}{12544} = 198.40$
Total distance	$L = \sum_{t \in E} t = 1676.67$
Clustering coefficient	$C_{global} = \frac{3N_{triangles}}{N_{groups}} = \frac{3}{75} = 0.04$

Table 9 – Calculation and analysis of system generated by proposed method

Proposed method	
	
Average distance	$AVG = \frac{1}{\lvert S \rvert} \sum_{a \in S} a = \frac{1930396.16}{12544} = 153.89$
Total distance	$L = \sum_{t \in E} t = 1527.66$
Clustering coefficient	$C_{global} = \frac{3N_{triangles}}{N_{groups}} = 0$ as there is no triangle in the system

5 RESULTS

In this chapter, we will evaluate the performance of the various algorithms employed to optimize transportation networks. It is crucial to assess not only the visual representations of the results but also to quantify those using relevant metrics. As described in the previous section, the average distance, total distance, and clustering coefficient were estimated for each algorithm. The results of the calculations are given in Table 10.

Table 10 – Comparison of each system

	Tokyo system	Slime Mould	Proposed method
Average distance	157.15	198.40	153.88
Total distance	1738.65	1676.667	1527.66
Clustering coefficient	0	0.04	0

Let us compare the effectiveness of the systems: each algorithm demonstrates different results regarding the average distance between pairs of points. The actual Tokyo railway system, which served as the benchmark, showed a score of 157.15. The slime algorithm, while it identified some optimal paths, had a worse score of 198.40, indicating lower efficiency compared to the benchmark. The developed method achieved the best result of 153.88.

Based on the comparative studies conducted, the developed method (using optimal parameters: 25 agents and 15 epochs) demonstrated the highest efficiency. It provided a shorter average distance between stations compared to the actual Tokyo railway system and achieved this improvement alongside a reduction in the overall path length, indicating its potential for further use in optimizing transportation networks.

Moreover, the total distance metric further emphasizes the advantages of the proposed method. With a total distance of 1527.66, it outperformed both the Tokyo system and the Slime Mould algorithm, which recorded total distances of 1738.65 and 1676.67, respectively. This reduction in total distance not only signifies a more efficient routing of transportation but also suggests potential cost savings and reduced travel times for users.

The clustering coefficient, while not as critical in this context, also provides insight into the connectivity of the network. The proposed method maintained a clustering coefficient of 0, similar to the Tokyo system, while the Slime Mould algorithm achieved a coefficient of 0.04. This indicates that the proposed method retains a level of simplicity in its structure, which can be beneficial for implementation and scalability.

In conclusion, the results indicate that the proposed method is not only effective in optimizing transportation networks but also demonstrates a significant improvement over existing systems. Future work should focus on refining the algorithm further, potentially incorporating more complex models or hybrid approaches that could enhance performance even more. By exploring advanced techniques such as machine learning or multi-agent systems, we can continue to push the boundaries of transportation network optimization.

DOI 10.15588/1607-3274-2025-2-16

6 DISCUSSION

Conducted research aimed at developing a geospatial multi-agent system for optimizing transportation networks. A comprehensive analysis of the subject area was performed, focusing on existing solutions and identifying promising research directions.

The study revealed several critical challenges in planning optimal transportation systems, particularly the dynamic nature and complexity of transportation networks. An additional issue is the necessity for the system to adapt to environmental changes, which is vital due to the constant evolution of routes and the emergence of obstacles. To address these challenges, a novel approach was proposed that enhances the minimum spanning tree (MST) using evolutionary strategies.

The results indicate that the developed system can effectively plan optimal transportation connections between cities. This system has the potential to significantly reduce travel time and costs, benefiting both passengers and transportation operators. Further research should aim to refine the methods for assessing environmental factors and the costs associated with selected routes, as well as improve existing metrics and develop new criteria for evaluating efficiency.

Additionally, integrating the system with neural network models could lead to a more in-depth analysis and optimization of transportation routes, particularly in complex urban environments where traditional methods may be inadequate. By utilizing neural networks to process geographical maps and spatial data, the system can generate structured input data that enhances route planning and urban development strategies. This integration has the potential to revolutionize our approach to urban transportation planning, making it more adaptive and efficient in response to ever-changing conditions.

CONCLUSIONS

The conducted research on developing a geospatial multi-agent system for optimizing transportation networks has yielded significant insights into the complexities and dynamic nature of transportation planning. The study's findings underscore the practical significance of the proposed system, which demonstrates the capability to effectively plan optimal transportation connections between cities. This advancement has the potential to substantially reduce travel time and costs, providing tangible benefits for both passengers and transportation operators.

Scientific Novelty: The research contributes to the scientific community by introducing a novel approach that combines evolutionary strategies with traditional transportation planning methods. This innovative perspective not only addresses existing challenges but also opens new avenues for exploration in the field of transportation optimization.

Practical Significance: The results of this research are particularly relevant for urban planners and transportation authorities, as they offer a novel solution to the pressing challenges of optimizing transportation networks. By enhancing the minimum spanning tree (MST) with evolutionary strategies, the system can adapt to environmental changes and evolving routes, thereby improving the overall efficiency of transportation systems.

Prospects for Further Research: The prospects for further research are promising, particularly in the areas of algorithm refinement and the development of new evaluation metrics for transportation efficiency. Investigating the application of the proposed system across various practical scenarios, including urban and rural settings, could yield valuable insights. Additionally, exploring the potential of machine learning and artificial intelligence in conjunction with the multi-agent system may lead to breakthroughs in adaptive transportation planning, ultimately revolutionizing how we approach urban mobility in response to changing demands.

Recommendations for Further Research: To build upon the findings of this study, it is recommended that future research focus on refining methods for assessing environmental factors and the associated costs of selected routes. Additionally, exploring the integration of neural network models into the system could provide deeper insights into route optimization, especially in complex urban environments. This integration could facilitate the processing of geographical maps and spatial data, leading to more structured input data that enhances route planning and urban development strategies.

ACKNOWLEDGEMENTS

The study was created within research topic "Methods and means of artificial intelligence to prevent the spread of tuberculosis in wartime" (№0124U000660), which is carried out at the Department of Artificial Intelligence Systems of the Institute of Computer Sciences and Information of technologies of the National University "Lviv Polytechnic".

REFERENCES

1. Moher D., Liberati A., Tetzlaff J. Preferred Reporting Items for Systematic Reviews and Meta-Analyses, *The PRISMA Statement. PLoS Medicine*, 2020, Vol. 6(7), P. e1000097, Mode of access: https://doi.org/10.1371/journal.pmed.1000097.
2. Oishi K., Hashizume Y., Jimbo T., Kaji H., Kashima K. Imitation-regularized optimal transport on networks: provable robustness and application to logistics planning, *arXiv preprint arXiv*, 2024, P. 1906.07114, Mode of access: https://doi.org/10.48550/arXiv.2402.17967.
3. Andrecut M. Heuristic Optimal Transport in Branching Networks, *arXiv preprint arXiv*, 2021, P. 1906.07115, Mode of access: https://doi.org/10.48550/arXiv.2311.06650.
4. Cominetti L., Como G., Ozdaglar A., Parise F. Optimal intervention in traffic networks, *arXiv preprint arXiv*, 2021, P. 1906.07116, Mode of access: https://doi.org/10.48550/arXiv.2102.08441.
5. Wu J., Pu C., Ding S., Cao G., Pardalos P. M. NC-MOPSO: Network centrality guided multi-objective particle swarm optimization for transport optimization on networks, *arXiv preprint arXiv*, 2021, P. 906.07117, Mode of access: https://doi.org/10.48550/arXiv.2009.03575.
6. Aslani B., Mohebbi S. Learn to decompose multiobjective optimization models for large-scale networks, *arXiv preprint arXiv*, 2021, P. 1906.07118, Mode of access: https://doi.org/10.1111/itor.13169.

DOI 10.15588/1607-3274-2025-2-16

7. Sbayti O., Housni K., Hicham Hanin M., El Makrani A. Comparative analysis of proactive and reactive routing protocols in VANET environment, *International Journal of Electrical and Computer Engineering (IJECE)*, 2023, Vol. 13(5), P. 5374–5387, Mode of access: https://doi.org/10.11591/ijece.v13i5.pp5374-5387.
8. Rafael H., Pereira M., Herszenhut D., Saraiva M., Farber S. Ride-hailing and transit accessibility considering the trade-off between time and money, *Cities*, 2024, Vol. 144, P. 104663, Mode of access: https://doi.org/10.1016/j.cities.2023.104663
9. Baptista D., De Bacco C. Convergence properties of optimal transport-based temporal networks, *arXiv preprint arXiv*, 2021, Mode of access: https://doi.org/10.1007/978-3-030-93409-5_48
10. Wen W., Zhang W., Deng H. Research on Urban Road Network Evaluation Based on Fractal Analysis, *Journal of Advanced Transportation*, 2023, P. 1–8, Mode of access: https://doi.org/10.1155/2023/9938001.
11. Choi J., Kang M. Analyzing and Improving Optimal-Transport-based Adversarial Networks, *arXiv:2310.02611*, 2023, Mode of access: https://doi.org/10.48550/arXiv.2310.02611.
12. VANET (Vehicular Ad-hoc Network) [Electronic resource]. Access mode: https://en.wikipedia.org/wiki/Vehicular_ad-hoc_network (access date: 11.12.2024). Title from the screen.
13. AODV (Ad-hoc On-Demand Distance Vector routing) [Electronic resource]. Access mode: https://en.wikipedia.org/wiki/Ad-hoc_On-Demand_Distance_Vector_routing (access date: 11.12.2024). Title from the screen.
14. DSDV (Destination-Sequenced Distance Vector routing) [Electronic resource]. Access mode: https://en.wikipedia.org/wiki/Destination-Sequenced_Distance_Vector_routing (access date: 11.12.2024). Title from the screen.
15. OLSR (Optimized Link State Routing) [Electronic resource]. Access mode: https://en.wikipedia.org/wiki/Optimized_Link_State_Routing_Protocol (access date: 11.12.2024). Title from the screen.
16. Yang X.-S. Nature-Inspired Algorithms in Optimization: Introduction, Hybridization and Insights, *arXiv:2401.00976*, 2023, Mode of access: https://doi.org/10.1007/978-981-99-3970-1.
17. Wei Y., Othman Z., Mohd Daud K., Luo Q., Zhou Y. Advances in Slime Mould Algorithm: A Comprehensive Survey, *Biomimetics*, 2024, Vol. 9, P. 31, Mode of access: https://doi.org/10.3390/biomimetics9010031.
18. Bellman-Ford Algorithm [Electronic resource]. Access mode: https://brilliant.org/wiki/bellman-ford-algorithm (access date: 11.12.2024). Title from the screen.
19. A* Algorithm [Electronic resource]. Access mode: https://en.wikipedia.org/wiki/A*_search_algorithm (access date: 11.12.2024). Title from the screen.
20. Network flow problem [Electronic resource]. Access mode: https://en.wikipedia.org/wiki/Network_flow_problem (access date: 11.12.2024). Title from the screen.
21. Network flow problem, Ford–Fulkerson algorithm [Electronic resource]. Access mode: https://www.networkflowalgs.com/book.pdf (access date: 11.12.2024). Title from the screen.
22. Wagdy A., Hadi A. A, Khater A. Gaining-sharing knowledge based algorithm for solving optimization problems: a novel nature-inspired algorithm, *International Journal of Machine Learning and Cybernetics*, 2023, Mode of access: https://doi.org/10.1007/s13042-019-01053-x.
23. SlimeMould [Electronic resource]. Access mode: https://github.com/MoeBuTa/SlimeMould (access date: 11.12.2024). Title from the screen.
24. Boyko N., Pytel A. Aspects of the study of genetic algorithms and mechanisms for their optimization for the travelling salesman problem, *International Journal of Computing*, 2021, Vol. 20(4), P. 543–550, Mode of access: https://doi.org/10.47839/ijc.20.4.2442.
25. Vyklyuk Ya., Nevinskyi D., Boyko N. GeoCity – a New Dynamic-Spatial Model of Urban Ecosystem, *J. Geogr. Inst. Cvijic*, 2023, Vol. 73(2), P. 187–203, Mode of access: https://doi.org/10.2298/IJGI2302187V.

Received 16.01.2025.
Accepted 29.04.2025.

УДК 004.94

РОЗРОБКА ІННОВАЦІЙНИХ ПІДХОДІВ ДЛЯ ОПТИМІЗАЦІЇ МЕРЕЖ ЗА ДОПОМОГОЮ ГЕОПРОСТОРОВИХ БАГАТОКОМПОНЕНТНИХ СИСТЕМ

Бойко Н. І. – канд. економ. наук, доцент, доцент кафедри Систем штучного інтелекту, Національний університет «Львівська політехніка», Львів, Україна.

Саланчій Т. О. – студент кафедри Систем штучного інтелекту, Національний університет «Львівська політехніка», Львів, Україна.

АНОТАЦІЯ

Актуальність. Розробка геопросторової багатагентної системи для оптимізації транспортних мереж є важливою для підвищення ефективності та зменшення часу подорожі. Це передбачає використання алгоритмів оптимізації та моделювання поведінки агентів у межах мережі.

Мета роботи є розробка геопросторової багатагентної системи для оптимізації транспортних мереж, зосереджуючи увагу на покращенні ефективності мережі та мінімізації часу подорожі шляхом застосування передових алгоритмів оптимізації та моделювання на основі агентів.

Метод. Запропонований метод оптимізації транспортних мереж поєднує базову структуру з розширеним уточненням у два етапи: попередня обробка та оптимізація еволюційної стратегії. На першому етапі будується мінімальне остовне дерево за допомогою алгоритму Крускала для встановлення найкоротшої мережі без петель, яка з’єднує всі ключові точки, враховуючи природні перешкоди та існуючі маршрути. Це забезпечує економічно ефективну та реалістичну базову лінію. Другий етап удосконалює мережу за допомогою еволюційної стратегії, де агенти, що представляють варіації мінімального остового дерева, оптимізуються за допомогою функції пристосування, яка балансує загальну довжину шляху, середню відстань до вузлів і штрафи за надмірні краї. Оптимізація використовує кросовер для поєднання рішень і мутацію для введення різноманітності через модифікації країв. Цей процес, повторюється протягом багатьох епох, поступово покращує мережу, в результаті чого створюється оптимізований щлях, який мінімізує витрати, покращує підключення та поважає обмеження подані в режимі реального часу.

Результати. Результати застосування еволюційної стратегії та методів мінімальної вартості відстані були детально проаналізовані. Для еволюційної стратегії були оцінені такі метрики, як ефективність шляхів і обчислювальні витрати, що

DOI 10.15588/1607-3274-2025-2-16

продемонструвало значні покращення в оптимізації мережі. У випадку MST, хоча метод надав базову структуру для вибору шляхів, візуальні та числові оцінки підкреслили обмеження в розв'язанні складних реальних обмежень. Порівняння цих методів з еталонами, такими як залізнична мережа Токіо та алгоритм слизової цвілі, виявило перевагу еволюційного підходу в генерації оптимальних шляхів. Висновки підкреслюють необхідність інтеграції передових алгоритмів для подальшого вдосконалення оптимізації шляхів і проектування мереж.

Висновки. Дослідження успішно розробило геопросторову багатоагентну систему для оптимізації транспортних мереж, досягнувши поставлених цілей шляхом вирішення ключових проблем у плануванні транспортної мережі. Детальний аналіз існуючих рішень виявив динамічний і складний характер транспортних систем і підкреслив необхідність адаптації до змін навколишнього середовища, таких як нові маршрути або перешкоди. Запропонований підхід розширив мінімальне охоплююче дерево за допомогою еволюційної стратегії, забезпечивши гнучкість і швидку адаптацію. Результати продемонстрували ефективність системи в плануванні оптимальних міжміських транспортних мереж. Майбутня робота може вдосконалити екологічні оцінки, покращити оцінку вартості маршруту, розширити показники, визначити нові критерії продуктивності та інтегрувати моделі нейронних мереж для подальшого підвищення можливостей оптимізації, особливо для міських мереж.

КЛЮЧОВІ СЛОВА: геопросторова мультиагентна система, оптимізація транспортних мереж, еволюційна стратегія.

ЛІТЕРАТУРА

1. Moher D. Preferred Reporting Items for Systematic Reviews and Meta-Analyses / D. Moher, A. Liberati, J. Tetzlaff // The PRISMA Statement. PLoS Medicine. – 2020. – Vol. 6(7). – P. e1000097. – Mode of access: https://doi.org/10.1371/journal.pmed.1000097.
2. Imitation-regularized optimal transport on networks: provable robustness and application to logistics planning / [K. Oishi, Y. Hashizume, T. Jimbo et al.] // arXiv preprint arXiv. – 2024. – P. 1906.07114. – Mode of access: https://doi.org/10.48550/arXiv.2402.17967.
3. Andrecut M. Heuristic Optimal Transport in Branching Networks / M. Andrecut // arXiv preprint arXiv. – 2021. – P. 1906.07115. – Mode of access: https://doi.org/10.48550/arXiv.2311.06650.
4. Optimal intervention in traffic networks / [L. Cianfanelli, G. Como, A. Ozdaglar, F. Parise] // arXiv preprint arXiv. – 2021. – P. 1906.07116. – Mode of access: https://doi.org/10.48550/arXiv.2102.08441.
5. NC-MOPSO: Network centrality guided multi-objective particle swarm optimization for transport optimization on networks / [J. Wu, C. Pu, S. Ding et al.] // arXiv preprint arXiv. – 2021. – P. 906.07117. – Mode of access: https://doi.org/10.48550/arXiv.2009.03575.
6. Aslani B. Learn to decompose multiobjective optimization models for large-scale networks / B. Aslani, S. Mohebbi // arXiv preprint arXiv. – 2021. – P. 1906.07118. – Mode of access: https://doi.org/10.1111/itor.13169.
7. Comparative analysis of proactive and reactive routing protocols in VANET environment / [O. Sbayti, K. Housni, M. Hicham Hanin, A. El Makrani] // International Journal of Electrical and Computer Engineering (IJECE). – 2023. – Vol. 13(5). – P. 5374–5387. Mode of access: https://doi.org/10.11591/ijece.v13i5.pp5374-5387.
8. Ride-hailing and transit accessibility considering the trade-off between time and money / [H. Rafael, M. Pereira, D. Herszenhut et al.] // Cities. – 2024. – Vol. 144. – P. 104663. – Mode of access: https://doi.org/10.1016/j.cities.2023.104663
9. Baptista D. Convergence properties of optimal transport-based temporal networks / D. Baptista, C. De Bacco // arXiv preprint arXiv. – 2021. – Mode of access: https://doi.org/10.1007/978-3-030-93409-5_48
10. Wen W. Research on Urban Road Network Evaluation Based on Fractal Analysis / W. Wen, W. Zhang, H. Deng // Journal of Advanced Transportation. – 2023. – P. 1–8. – Mode of access: https://doi.org/10.1155/2023/9938001.
11. Choi J. Analyzing and Improving Optimal-Transport-based Adversarial Networks / J. Choi, M. Kang // arXiv:2310.02611. – 2023. – Mode of access: https://doi.org/10.48550/arXiv.2310.02611.
12. VANET (Vehicular Ad-hoc Network) [Electronic resource]. – Access mode: https://en.wikipedia.org/wiki/Vehicular_ad-hoc_network (access date: 11.12.2024). – Title from the screen.
13. AODV (Ad-hoc On-Demand Distance Vector routing) [Electronic resource]. – Access mode: https://en.wikipedia.org/wiki/Ad-hoc_On-Demand_Distance_Vector_routing (access date: 11.12.2024). – Title from the screen.
14. DSDV (Destination-Sequenced Distance Vector routing) [Electronic resource]. – Access mode: https://en.wikipedia.org/wiki/Destination-Sequenced_Distance_Vector_routing (access date: 11.12.2024). – Title from the screen.
15. OLSR (Optimized Link State Routing) [Electronic resource]. – Access mode: https://en.wikipedia.org/wiki/Optimized_Link_State_Routing_Protocol (access date: 11.12.2024). – Title from the screen.
16. Yang X.-S. Nature-Inspired Algorithms in Optimization: Introduction, Hybridization and Insights / X.-S. Yang // arXiv:2401.00976. – 2023. – Mode of access: https://doi.org/10.1007/978-981-99-3970-1.
17. Advances in Slime Mould Algorithm: A Comprehensive Survey / [Y. Wei, Z. Othman, K. Mohd Daud et al.], // Biomimetics. – 2024. – Vol. 9. – P. 31. – Mode of access: https://doi.org/10.3390/biomimetics9010031.
18. Bellman-Ford Algorithm [Electronic resource]. – Access mode: https://brilliant.org/wiki/bellman-ford-algorithm (access date: 11.12.2024). – Title from the screen.
19. A* Algorithm [Electronic resource]. – Access mode: https://en.wikipedia.org/wiki/A*_search_algorithm (access date: 11.12.2024). – Title from the screen.
20. Network flow problem [Electronic resource]. – Access mode: https://en.wikipedia.org/wiki/Network_flow_problem (access date: 11.12.2024). – Title from the screen.
21. Network flow problem, Ford–Fulkerson algorithm [Electronic resource]. – Access mode: https://www.networkflowalgs.com/book.pdf (access date: 11.12.2024). – Title from the screen.
22. Wagdy A. Gaining-sharing knowledge based algorithm for solving optimization problems: a novel nature-inspired algorithm / A. Wagdy, A. A Hadi, A. Khater // International Journal of Machine Learning and Cybernetics. – 2023. – Mode of access: https://doi.org/10.1007/s13042-019-01053-x.
23. SlimeMould [Electronic resource]. – Access mode: https://github.com/MoeBuTa/SlimeMould (access date: 11.12.2024). – Title from the screen.
24. Boyko N. Aspects of the study of genetic algorithms and mechanisms for their optimization for the travelling salesman problem / N. Boyko, A. Pytel // International Journal of Computing. – 2021. – Vol. 20(4). – P. 543–550. – Mode of access: https://doi.org/10.47839/ijc.20.4.2442.
25. Vyklyuk Ya. GeoCity – a New Dynamic-Spatial Model of Urban Ecosystem / Ya. Vyklyuk, D. Nevinskyi, N. Boyko // J. Geogr. Inst. Cvijic. – 2023. – Vol. 73(2). – P. 187–203. – Mode of access: https://doi.org/10.2298/IJGI2302187V

DOI 10.15588/1607-3274-2025-2-16