DOI: https://doi.org/10.26565/2304-6201-2023-57-04

UDC 656.11

Nikolyuk P.K. *doctor of physical and mathematical sciences, professor of information technologies department*
Vasil' Stus Donetsk National University, 600-richchia str., 21, Vinnytsia-21, Ukraine, 21021
e-mail: p.nikolyuk@donnu.edu.ua
https://orcid.org/0000-0002-0286-297X

Routing a vehicle through city traffic by the time-optimal dynamic path

The key goal of this study is to synchronize traffic flows, optimize the use of the transport arteries throughout the city, prevent congestion, and follow each vehicle to its destination to minimize time spent on the trip. As a result, the total time spent by cars on the road will be significantly reduced, and environmental conditions will improve accordingly.
The object of the study is a city's transportation network, represented as a weighted oriented nonplanar multigraph (WONM). The key advantage of using the graph theory to build optimal routes is based on the following considerations: 1) the graph theory has developed many algorithms for finding optimal routes (Dijkstra algorithm, Floyd-Warshall algorithm, A-star algorithm, etc.); 2) the graph theory is used as the theoretical and practical basis of logistical systems, including urban traffic. To build a route in such a multigraph an A-star algorithm has been used, which establishes an optimal time (t-optimal) route between two selected vertices of graph.
The study offers a realistic prospect for solving the problem of congestions through the use of a special software algorithm oriented towards establishing optimal routes and using graphs to represent the city's transportation network.
The fundamental issue is the representation of the city's transport network in the form of an electronic map and the display of GPS-identifiers of vehicles involved in traffic on this map. The "city traffic → electronic map" representation makes it possible to obtain data as to the level of congestion in the transport network. The use of an electronic city map allows the GPS coordinates of each vehicle to be projected onto it. Thus, the city's transport network is under the full control of the transportation management center (TMC), which has a real opportunity to interact with each vehicle and constantly adjust its route, choosing the t-optimal one. The route adjustment is carried out via General Packet Radio Service (GPRS) channel in the form of voice commands as in conventional GPS navigation. However, the specifics are as follows: 1) the navigation is carried out online; 2) t-optimal routes are plotted, taking into account the traffic situation at any given time.
Thus, a large-scale transportation urban traffic network and an associated computer program has been developed. This is an applied project, and its results can be used to effectively regulate traffic in megacities in order to minimize the travel time of each vehicle along a selected route.
Keywords: *weighted oriented nonplanar multigraph, GPS-navigation, A-star algorithm, t-optimal route, intelligent traffic.*

How to quote: P.K. Nikolyuk, "Routing a vehicle through city traffic by the time-optimal dynamic path", *Bulletin of V.N. Karazin Kharkiv National University, series Mathematical modelling. Information technology. Automated control systems*, vol. 57, pp. 42-54, 2023.
https://doi.org/10.26565/2304-6201-2023-57-02

Як цитувати: Nikolyuk P.K. Routing a vehicle through city traffic by the time-optimal dynamic path. *Вісник Харківського національного університету імені В. Н.Каразіна, серія Математичне моделювання. Інформаційні технології. Автоматизовані системи управління*. 2023. вип. 57. С.42-54. https://doi.org/10.26565/2304-6201-2023-57-02

1. Introduction

In the graph theory many algorithms have been developed to build optimal routes, including t-optimal ones [1] and this is the key advantage of the multigraph model over other traffic simulation models. Another key advantage of the multigraph model is the possibility to simulate the city's transportation network, depicting all lanes, intersections, traffic junctions, and pedestrian crossings. These features allow implementing the procedure of programmatic regulation of urban traffic, which is a particularly important result of the study. In other words, a basic framework for automating urban traffic processes has been created. The model of the urban transport network in the form of a WONM allows using graph theory algorithms to find optimal routes for vehicles participating in traffic. It is most appropriate to use the A-star algorithm, since it ensures the optimal routes between two selected positions of the graph (city transport network) and has a number of advantages over other similar algorithms.

The fundamental issue is the creation of bijective mapping principles of the "city traffic → WONM" type. In other words, the location of vehicles on urban transport arteries should be reproduced as objectively as possible on the urban electronic map used by the city TMC. In addition, constant interaction between each vehicle, which is identified on the electronic map by a pair (O_i, D_i) is required. Each such pair is a specific vehicle identifier which can be tracked on the city's electronic map. Since each car i is under the control of TMC, tracking it along the selected route is equivalent to tracking the identifier (O_i, D_i) on the electronic city map. In turn, vehicle tracking involves online interaction with the vehicle driver via a GPRS channel. Therefore, each driver can receive accurate and timely instructions on the best way to travel. Importantly, such instructions correlate with the dynamics of urban traffic and are the result of implementing the A-star algorithm, which creates the optimal route in the graph and constantly updates the route in accordance with the traffic situation at a particular time. In other words, the route of each car is synchronized with the dynamical changes in the traffic situation.

Practically, regulating the traffic of tracked vehicles is carried out by automating traffic control at each intersection and road sections between neighboring intersections. This allows TMC to control all traffic flows in the metropolis in real time, as well as to track each vehicle along the route where the initial point is established by using a GPS navigator and the destination point is set.

The process of controlling and regulating urban road traffic is based on the fact that TMC transmits voice commands to each driver via GPRS channel highlighting the route to the destination point as in conventional GPS navigation. The distinct feature of the program is the analysis of the dynamic traffic situation throughout the city by using interactive road sensors located on the electronic map of the city in the vicinity of intersections.

The practical application of the described process of regulating the urban traffic will lead to the disappearance or significant reduction of traffic jams. The main results of this study are: 1) optimization of vehicle routes, which minimizes the travel time between origin and destination points; 2) synchronization of traffic flows; 3) the uniform redistribution of traffic flows within the city, which will lead to optimal use of transport arteries throughout the city; 4) absence or significant reduction in the duration and frequency of congestions.

2. Analytical review of the related works

The previous investigation [2] addresses the problem of traffic control in a metropolis by using the Dijkstra algorithm. The authors consider the Intelligent Transportation System (ITS) application in order to analyze the causes of congestion on urban transport arteries. ITS makes it possible to provide innovative services and enable users to be better informed and make safer, more coordinated, and 'smarter' usage of transportation networks.

A map-matching algorithm [3] allows for the correct comparison of the GPS-trajectory of the vehicle with the corresponding road segment. In particular, the comparison of GPS data with the electronic map of the city is carried out by using Automatic Vehicle Location (AVL) systems that use vehicle tracking system to track the movement of a vehicle. In turn, the information collected by AVL can be compared with electronic maps via the Internet or with help of special software. A map-matching algorithm is a significant part of any navigation system because it aligns data from the GPS with a digital road network. The map-matching algorithms are divided into simple, weight-based, and advanced ones. The advanced algorithms use different mathematical models such as probability theory, fuzzy logic, multihypothesis road tracking, a hidden Markov model, a hybrid Bayersian network and a neural network. The cited research proposes the weight-based algorithm to find a best segment of road network. In [3,5], the problem of map-matching of GPS markers (points) of vehicles with electronic maps is considered in detail. For this purpose, the so-called device-based sensors are used, which constantly record and transmit space-time information about the location and movement of the vehicle [6]. The latest applications of urban vehicle mobility visualization systems are based on the usage of Geographic Information Systems (GIS). This system allows linking data to an electronic map, in particular displaying the geolocation position of the vehicle on such a map. The study [7] shows that finding the shortest route by using an A-star algorithm facilitates the procedure of comparing a vehicle position on electronic map with road segments. That means that Dijkstra or Floyd algorithms are less effective in this case.

City traffic flows need recording in real time, since knowing the level of congestion on city streets is the starting point of solving a traffic flow regulation problem. The next step is for the TMC to select the best routes in the urban transportation network. This is key problem in terms of urban traffic strategy.

Thus, in [8], a new control system using the heuristic approach is used to solve this problem. The authors propose a new integrated control algorithm that combines actions of dynamic traffic routing with on-ramp metering to optimize the traffic flow. The algorithm can manage the system in the case of a capacity constraint in the highway network. The investigation focuses on the development of integrated traffic management strategies in road networks. The highway network is modeled by using the Lighthill-Whitham-Richards traffic flow model, which introduces the terms of flow density and average speed. An integrated control algorithm based on the methodology of feedback control and variable structure control has been developed to solve the proposed problem. Three options for selecting alternative routes to avoid congestion have been tested. The results show that the proposed algorithms can offer to the user a balance between the selected alternative routes.

A number of studies have been aimed at implementing the principles of urban traffic forecasting. Studying networks of different nature, such as neural networks, allows making forecasts. In particular, short-term traffic forecasting [9] is one of the most important elements of all active traffic control. The development of a forecasting model based on neural networks allows for short-term forecasting based on the selection of the best combination of forecast parameters. In this work, a self-correcting neural network based on genetic sorting has been used. The NSGA-II algorithm is used as a multi-objective optimizer for short-term forecasting.

The Kalman filter [10] for traffic forecasting on urban highways based on data obtained from connected vehicles is used. That allows for real-time forecasting, since the data of the connected vehicle is analyzed immediately before the forecast period. To analyze traffic data, the Vissim simulator, which registers vehicles at different speeds is used. The performance of the algorithm for different traffic situations is evaluated by using statistical methods.

The short-term forecasting of traffic scenarios [11] uses a set of specific tools and models. In particular, such forecasting models as the non-parametric k-Nearest Neighbor (kNN) regression model, the Gaussian probability maximization model, and the double seasonal exponential HolteWinters smoothing model are important components. Real highway traffic data is used to test the theoretical conceptions. The research allows predicting weekly and monthly fluctuations in average daily traffic with varying degrees of precision while being ease of use. The article uses the information entropy method and the less common Shapley method.

Forecasting of traffic processes can be carried out both for the day and for a longer-term range [12]. For this purpose, a traffic management system based on predictive information through the use of static and mobile objects is used. These objects (agents) use a methodology for collecting and transmitting data on traffic flow parameters (speed and density), spatial and temporal information on urban traffic modes in order to monitor and predict expected traffic density patterns. All that permits the vehicle driver to choose the best routes and thus ensure uninterrupted traffic flow and reduce the frequency of traffic jams. Traffic situation monitoring and forecasting is performed through the integration of NS2, Simulation of Urban Mobility (SUMO), OpenStreatMap (OSM) and the MOVE tool.

Urban traffic is a function of the capacity of a particular urban intersection or a place where traffic flows intersect. The fundamental question is how to organize an effective process of regulating the passage of vehicles through this intersection. Therefore, the main task is to create a mode of optimizing the capacity of each urban intersection [13]. For this purpose, an algorithm to control traffic signals at isolated intersections is used. In this approach, a central controller is used to collect the real-time location of communicating vehicles at certain intervals. The controlling of traffic lights is focused on the correct choice of the phase of the traffic signal, which maximizes the average vehicle speed while limiting the maximum delay that any individual vehicle can experience. The proposed algorithm is also effective when the level of interaction between connected vehicles is imperfect, when such objects account for more than 40% of the traffic flow. The results show that the proposed strategy can help to significantly reduce long delays. Intersection coordination is carried out through the implementation of a computer program for regulating vehicle flows [14]. The presented computer program allows optimizing the passage of cars through each individual intersection.

The invention of real-world applied algorithms has played an important role in creating an intelligent intersection control [15]. In particular, an intelligent traffic light can be implemented on the basis of an intelligent terminal. The intelligent terminal has a bidirectional communication with the TMC, which is used to manage urban roads and provide information services for cars, as well as provide electronic maps and route maps for the intelligent terminal. The intelligent traffic light system automatically detects the number of vehicles on the roads to adapt the travel time for each road. The system and method provided by the invention can help drivers optimize route maps and improve the efficiency of vehicle traffic.

The adaptive traffic light control system [16] operates in real time. The authors propose to consider each intersection in the urban transport network as an independent, autonomous object that can change the length of the green phase of the traffic light depending on the characteristics of local traffic. This will significantly improve the capacity of each individual intersection.

Robot cars are promising objects that are receiving special attention from researchers. Such cars operate in an autonomous mode – without drivers [17]. The advantages of this innovative approach and the real prospects for its use are becoming apparent today. This problem has recently received considerable attention from researchers around the world, and the number of publications has been growing sharply in recent years.

3. Presentation of the problem and the results of the work

It is convenient to regulate urban traffic on an electronic map of the city by means of a bi-active display of the type "urban traffic → electronic map". In this case, urban traffic can be managed by TMC. However, the implementation of the transformation "urban traffic → electronic map" is a significant problem [6]. To solve this problem, the following tasks have been set (Fig. 1):

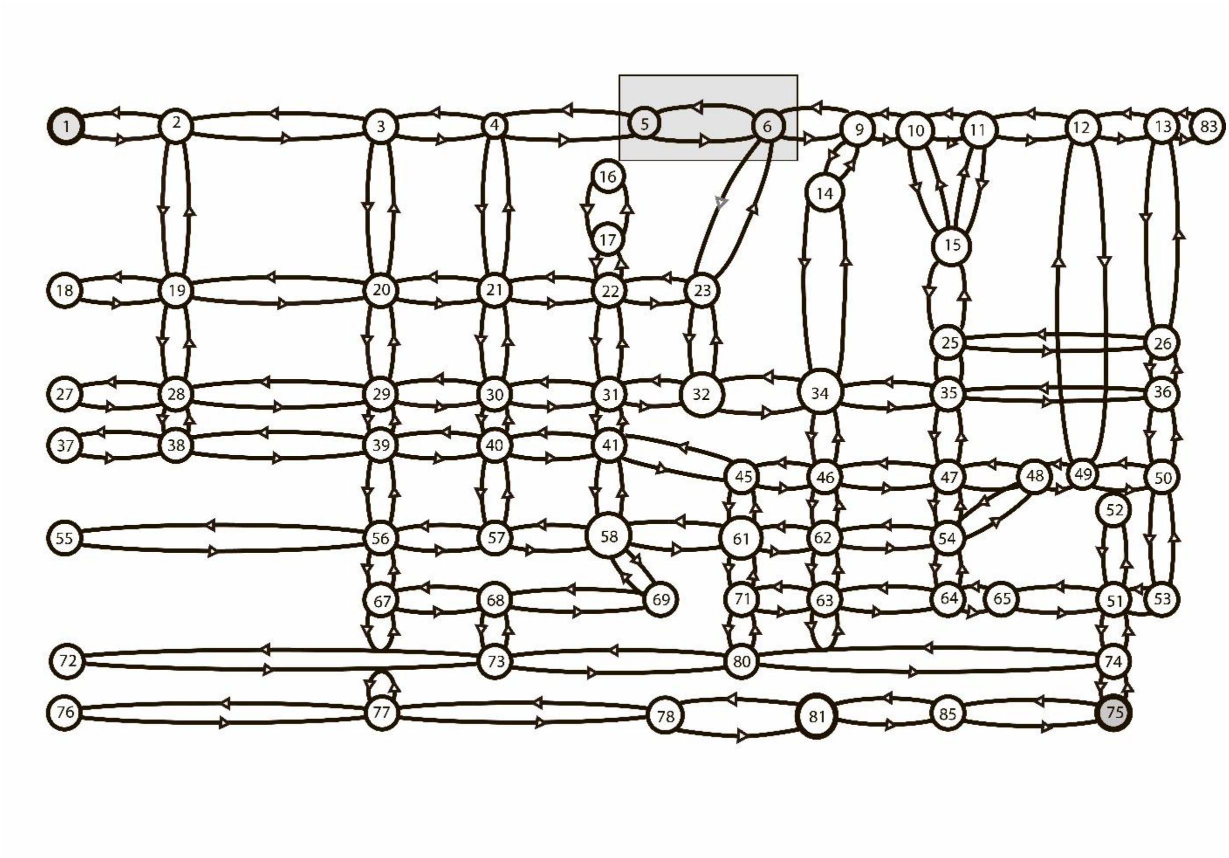

Fig.1. Weighted oriented non-planar multigraph model of the city's transport network fragment. The highlighted gray square is shown in detail on Fig. 2 below. The origin and destination points of a particular vehicle are located at nodes 1 and 75, respectively.

- to create a real-time model of the urban transport network in the form of an oriented weighted nonplanar multigraph with dynamically loaded arcs;
- to activate the electronic map of the city in TMC and use this map to track each vehicle;
- to create optimal time routes for all vehicles that have ordered such routes;
- to implement in real time the software algorithm for constant route updates for each guided vehicle. This will allow synchronizing city traffic with the traffic management procedure on the TMC and making adjustments to the route for each accompanied vehicle.

The schema of the principle of traffic organization is shown as the diagram on Fig. 1. Urban traffic management is relied upon by the TMC, which consists of three interconnected subdivisions: the city road electronic map, which receives geo-positioning data from each vehicle in the city road network. After processing, the data is transferred to the information and computing center. The routes calculated here send as instructions via a GPRS channel to each vehicle by traffic control center that manages each vehicle.

The creation of a model of the city's transport network by using the WONM makes the operational management of traffic flows possible. The fundamental point is that in such a graph, each arc is matched with a weight - a dynamic value correlated with the actual traffic load of each city road lane. In the terms of graph theory, this graph is a bijective (as objective as possible) representation of the distribution of vehicles on the city's transport arteries. WONM is the basic representation of an electronic traffic map of the city. GPS identifiers of vehicles involved in urban traffic are projected onto such a map [3].

The optimal routes in the graphs are constructed by using special algorithms, among which the Dijkstra and Floyd-Warschall algorithms are the most well-known [18]. The first one finds the optimal route between the selected graph vertex and all other vertices, and the second one finds the optimal route between all vertices of the graph. However, for this study, we need to build optimal routes between two selected vertices of the graph, since each driver wants to get from a certain position A to position B. In this regard, it is advisable to use the A*-algorithm to build optimal routes between two specific vertices of the graph.

The implementation of a software algorithm that builds optimal routes has a number of features, including high traffic dynamics and different levels of traffic congestion in individual lanes on the same road section and in individual streets. It is known that city central highways are usually congested during peak hours, while secondary transport arteries operate in normal or even underloaded traffic mode [6]. Therefore, one of the fundamental tasks of the proposed algorithm is to organize a predominantly uniform distribution of vehicles along the city's road arteries.

One of the issues is interacting with each vehicle in order to transfer data on the calculated optimal route. The registration of vehicles participating in traffic is carried out through the use of special sensors located on the electronic map of the city in the vicinity of intersections. These sensors interact (read information) with an electronic vehicle marker O_i, D_i when it crosses the electronic sensor. The data from the sensors act as input values to the information and computing center that implements the A-star algorithm (Fig. 1). The center plans the optimal route for each vehicle, taking into account the traffic situation in real time. For this purpose, each route is constantly corrected. Let us consider the specific details of the algorithm by using the example of the model shown in Fig. 2.

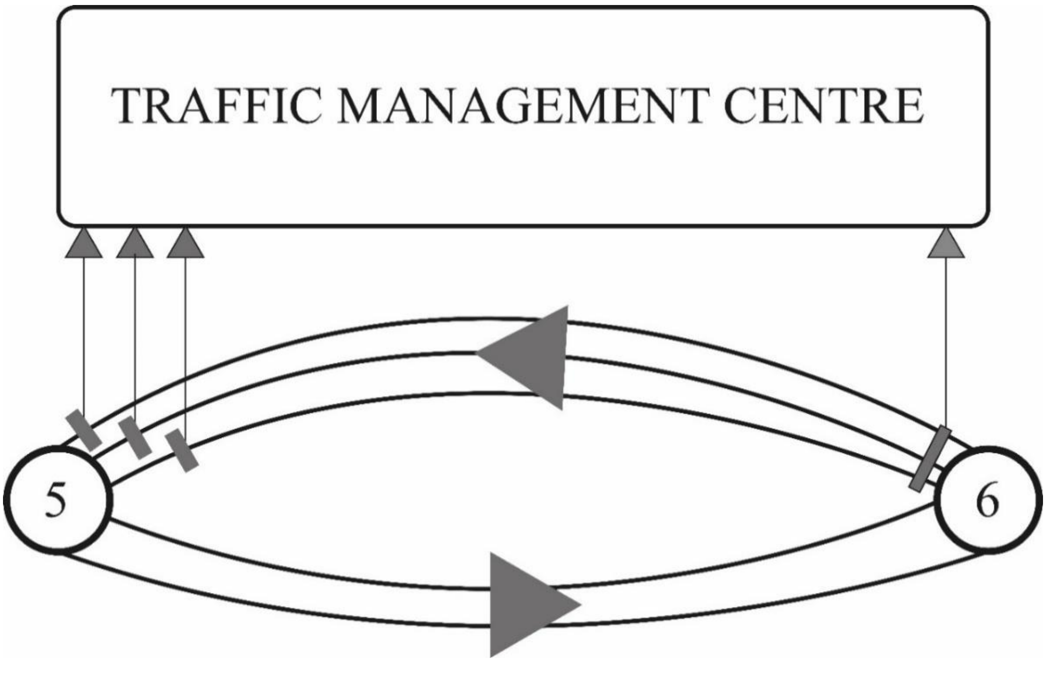

Fig. 2. Scheme of an interaction between electronic map sensors and TMC

This simple graph is an actual representation of a section of the city's transportation network. This means that the lanes between intersections are reproduced – there can be 2, 3 or more such lanes, which corresponds to the situation with the real urban transport network. In addition, each arc of the graph is assigned a time-varying weight that corresponds to real city traffic. The graph (Fig. 1) also contains two types of traffic interchanges – the first is vertex 17, where circular traffic is organized, and the second is organized between intersections 2 and 8, which are connected by a road that runs over intersection 5. The graph (Fig. 2) models not just the road between intersections, but also reproduces individual lanes, which is a fundamentally important point, since the congestion of vehicles in different lanes, as a rule, varies significantly. Therefore, it is necessary to record the congestion of each lane separately. The fundamental issue is to build the time-optimal route for each vehicle. For this purpose, the following definition should be introduced:

Definition 1. The intersection traffic resistance as a function of the traffic observation time t in the direction of the intersection for cars located along lane h is defined as the time interval for which a car is delayed at the intersection j:

$$R_h^j(t) = S_h^j(t) \cdot T_h^j(t), \tag{1}$$

where $T_h^j(t) = \tau_j^{red}(h) + \tau_j^{green}(h)$ – the length of the traffic light switching cycle at the intersection j (the length of the yellow signal is included in $t_j^{red}(h)$); $S_h^j(t)$ – a delay factor – a dimensionless coefficient is set as follows

$$S_h^j(t) = \begin{cases} \delta \\ \tau_h^j / T_h^j + \delta \\ \tau_h^{red} / T_h^j + C_h^j + \delta \\ t_h^{green} / T_h^j + C_h^j + \delta \end{cases}, \tag{2}$$

where the value $\delta << 1$ corresponds to the ratio (vehicle travel time through the intersection)/(the length of the traffic light cycle T_h^j). Thus, in the case of the first term, the vehicle crosses the intersection j without any delay. The value $\tau_h^j / T_h^j + \delta$ in the second term is obtained when the vehicle approaches the intersection j when the red light is on, delays for a while $\tau_h^j \le \tau_h^{red}$, and passes the intersection immediately after the green phase turns on. The value $t_h^{red} / T_h^j + C_h^j$ in the third term consists of the fraction T_h^j plus the number of traffic light switching cycles ($C_h^j \ge 1$) during which the car is delayed at the intersection. The value $t_h^{green} / T_h^j + C_h^j$ is a part of the value of the fourth term, consisting of the fraction T_h^j plus the integer number of traffic signal cycles (C_h^j).

The delay time (intersection resistance) of a car at an intersection is given as

$$R_h^j(t) = \begin{cases} \delta \cdot T_h^j \\ \tau_h^j + \delta \cdot T_h^j \\ t_h^{red} + C_h^j \cdot T_h^j + \delta \cdot T_h^j \\ t_h^{green} + C_h^j \cdot T_h^j + \delta \cdot T_h^j \end{cases} \tag{3}$$

The weight $W^h(l_{ij}, t)$ of each arc of the graph (Fig. 2) is the sum of

$$W_h(l_{ij},t) = R_h^j(t) \cdot V_{ij} + L_h . \quad (4)$$

Where l_{ij} is a lane directed from the intersection i to the intersection j; l is a designation for the lanes named starting from the centerline as $a,b,c,d,...$ (for example, the entry c_{6_5} symbolizes the third lane from the dividing line of the road, which is directed from intersection 6 to intersection 5 – Fig. 2). In the program algorithm [19], which presents the complete procedure for calculating the optimal route the antipode of expression (4) is the value of $double\ Wli\_j = R() \cdot 10 + Li\_j$. We assume that the average vehicle speed (in m/s) on urban transport arteries is 10. The principal task of the study is to find the minimum value of the function

$$\sum_{h=1}^{H} (R_h^j(t) \cdot V_{ij} + L_h) \to \min . \quad (5)$$

Where, H is the set of lanes along the route that a vehicle passes while moving from the start to the end position. Expression (5) consists of a variable component (the first term) and a constant component (the second term). The variable component is a function of time, since traffic in the city is highly dynamic.

The fundamental issue is to build a route synchronized with real, highly dynamic urban traffic. The procedure of such synchronization is implemented through the process of constantly recalculating the optimal route for each vehicle by a program. Specifically, the route is recalculated to take into account the state of city traffic at a particular moment every time the vehicle crosses the next intersection. In other words, we are talking about the route that corresponds to the traffic situation at a given time. The features of the process of implementing a dynamic software process that allows for route adjustment will be described below.

Expression (5) represents the objective function of the problem: the total weight of all consecutive lanes along the route should be minimal. The minimization of this function is performed by the A*-algorithm. The code for the graph (Fig. 1) is presented in [19].

Definition 2. The weight of the route (5) is the sum of the real and virtual components of the path: ($\sum_{h=1}^{H} L_h$) is the real component of the path that the car physically travels along the route; ($\sum_{h=1}^{H} R_h^j(t) \cdot V_{ij}$) is the virtual component of the path – this is the path that the vehicle would have traveled during the delay at the intersection $R_h^j(t)$, moving at an average speed V_{ij}.

Expression (5) is equivalent to the following

$$\tau^{trip} = \sum_{h=1}^{H} (R_h^j(t) + L_h / V_{ij}) \to \min , \quad (6)$$

where τ^{trip} is the travel time of the vehicle from its starting position to the destination. In expression (6) $R_h^j(t)$ there is a delay time of the vehicle (which was in the traffic lane h) at the intersection j, during which the vehicle is in the standby mode. To travel such a route (6) takes a minimum time.

According to (6), there is reason to talk about optimizing the car's travel along the selected route in terms of time. Accordingly, the total time spent by all vehicles participating in the traffic will also be optimal (minimal).

An important value that characterizes urban traffic is the coefficient of dynamics, which characterizes the capacity of each individual intersection

$$D_h^j(t) = m_h^j(t) / (N_h^j(t)) . \quad (7)$$

Where $N_h^j(t)$ represents the number of vehicles in the lane h in the direction of the intersection j and the value $m_h^j(t)$ is the number of vehicles that passed through the intersection j from the lane h

during the green phase t_h^{green} of the traffic light. Thus, the traffic dynamics on a lane h along a road section $i \to j$ is determined by the ratio between the number of vehicles leaving this road section during the time equal to the green phase t_h^{green} of the traffic light to the number of vehicles on the road section $i \to j$ at a given time.

The throughput $D_h^j(t)$ is related to the intersection resistance $R_h^j(t)$ as follows

$$D_h^j(t)=k / S_h^j(t) , \quad (8)$$

where the coefficient k has a time dimension.

The objective function is written in the form

$$D_h^j(t) \to \max . \quad (9)$$

The analysis of the relation (8) shows that the dynamics of traffic directly depends on the capacity of the city's intersections. A number of works have been devoted to the issue of improving the throughput of urban intersections [20, 21]. In our study, the process of creating the intelligent intersection is based on the following principle.

Definition 3: Intersection intelligence is a traffic signal control process in which the length of the green light phase t_h^{green} in a direction $i \to j$ is proportional to the number of vehicles in that direction.

Creating the intelligent intersection allows maximizing both its capacity (9). In order to illustrate the operation of such object, it is advisable to conduct a model experiment by using the AnyLogic North America LLC 8.8.0 program. The purpose of the modeling procedure is to show how creating the intelligent intersection affects the value of $D_h^j(t)$.

To control traffic, it is necessary to control the values included in (6). The process of fixing the described values is organized as follows: a sensor i_h (input sensor), located on an electronic map in the vicinity of the intersection i in the direction of the intersection j (Fig. 2), registers the geoposition of vehicles in accordance with their code identifier. The spectrum of such identifiers is given by the relation

$$S=\{(O_1,D_1),(O_2,D_2),...(O_k,D_k),...(O_N,D_N)\} . \quad (10)$$

Where N is the number of vehicles participating in urban traffic. Each car gets its own unique token consisting of an origin-destination pair (O_k, D_k). This pair is a specific recorder of each tracked vehicle, which allows it to be identified when it crosses the input or output sensors of the electronic map. The set of output sensors (Fig. 2) located in the vicinity of the intersection j on the lanes directed in the direction of $i \to j$ registers vehicles that have left the road section $i \to j$. The input sensors, on the opposite, register vehicles that have entered the road section $i \to j$. The ratio $D_h^j(t)$ of these values according to (7) indicates the dynamics of traffic on a particular road section in a particular direction $i \to j$. It is clear that the closer $D_h^j(t)$ is to unity, the higher the traffic dynamics on the road section in the direction of $i \to j$. The task is to find a sequence of lanes in the urban transportation network whose total weight is minimal (in terms of vehicle travel time) compared to all possible options – a t-optimal route.

Relation (7) characterizes the dynamics of vehicle movement on the road section $i \to j$. If

$$1 \le N_h^j(t)/m_h^j(t) < 2, \quad (9)$$

we can talk about high dynamics on this section of the road.

On the contrary, if

$$N_h^j(t)/m_h^j(t) \gg 1 \quad , \tag{10}$$

it indicates low dynamics of vehicle movement in the corresponding lane h, which causes a large weight of this lane. Thus, it is necessary to determine the throughput for each individual lane separately. That is taken into account in developed software by implementing the A*-algorithm.

The functional relationship between the relations (10) and (11) and the intersection resistance $R_h^j(t)$ is represented as

$$R_h^j(t) \sim 1/D_h^j(t) \; . \tag{11}$$

Improving the throughput of each individual intersection is a separate problem, which has been addressed in a number of studies [4,7,22,23]. If we improve the capacity of each individual intersection, then the terms $R_h^j(t)$ in the objective function (6) will be minimized. Thus, the procedure for minimizing the objective function (6) consists of two stages: 1) minimization of the total weight of the route, which is associated with the choice of the route in the graph/urban transport network; 2) minimization of the travel time through each individual intersection.

In order to simulate the weights of graph arcs, the program [19] uses a random variable generator that generates a variable part of the weight of a graph arc by using the method of $public\,static\,double\,R()$. In fact, this method returns a spectrum of real numbers in the range 0÷500. This range corresponds to the real traffic situation caused by the delay of a vehicle at an intersection in the range of 0÷10 times of traffic light switching cycles, where an average duration of one cycle is 50 seconds. The weight of the graph arc $W_h(l_{ij}, t)$ is represented as the sum of (4), the value of which correlates with the congestion of urban transport arteries. In the program, the weights of the graph arcs are given as follows $double\; Wa_{2\_1} = R() \cdot 10 + L_{2\_1}$.

For example, for the lane a_{2_1} we get $double\; Wa_{2\_1} = R() \cdot 10 + L_{2\_1}$. Thus, each arc of the graph (Fig. 1) receives a time-varying weight, the value of which is constantly changing, which corresponds to the realities of urban transport arteries. Fig. 3 shows a mix of neighboring intersections. When the geopositioning tags of the vehicles cross the input sensors of the electronic map, the signal is sent to the ICC and the re-routing procedure [19] for vehicles with markers $(O_i, D_i,\; i \in N)$ is initiated.

The re-routing procedure is necessary to respond to changes in urban traffic, in other words, to synchronize the traffic dynamics with the routing procedure. For example, in the case of the specific vehicle (Fig. 3), the TMC program calculates a new route as soon as the specified vehicle crosses the intersection i (the geopositioning mark of the vehicle on the electronic map crosses the input sensor i_h). This process initiates the transmission of the control signal via GPRS channel and at the very beginning of the road section (that is important!) the driver of the vehicle will receive the program-calculated instructions for the further route selection. Thus, the driver has the time and opportunity to shift to the desired lane in order to proceed to the next intersection, where the re-routing procedure will be repeated. And so on until the vehicle reaches its destination.

Study [1] shows that finding the shortest route by using the A-star algorithm facilitates the procedure of comparing the geoposition of cars on an electronic map with road segments. This means that the Dijkstra or Floyd algorithms are less effective in this case.

A compact representation of the congestion of urban transport networks can be made by using matrix analysis, as shown in [22]. The elements of such a matrix are values that represent the ratio of the average speed of traffic on a road segment j to the speed of free movement on that segment.

The real-time interaction between each vehicle and the TMC is crucial for the continuous programmatic route correction. The program [19] creates specific individual routes for each vehicle: each vehicle is tracked by an individual program, as the A-star algorithm creates an optimal route between two specified graph vertices. In other words, this algorithm is tied to the geometric coordinates of the origin-destination pair.

If you run the program [19], the results will usually be different each time you run it. One of the possible variants of the program is presented below:

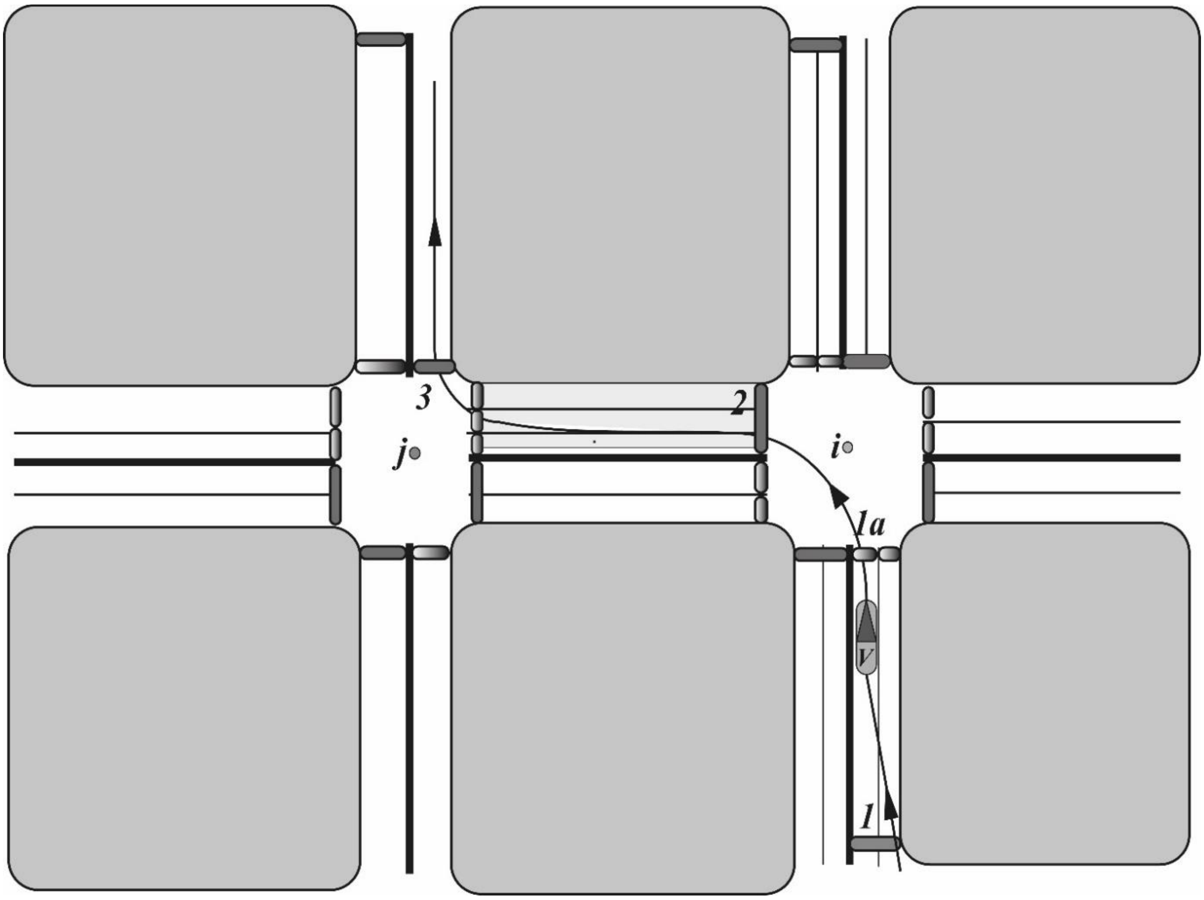

Fig.3. The trajectory of the car's route is shown. The data from the electronic map sensors are sent to the TMC (Fig. 2). The numbers 1, 2 and 3 on the input sensors mark the positions where the vehicle routes are recalculated in order to be synchronized with the new traffic realities.

Listing 1
Course-1: [*a2_3-Broad Street*, a3_2-Broad Street, a2_5-Heroes Street, a5_4-Short Street, a4_7-Long Street, b7_9-Trump Street, a9_13-Welly Street, a13_14-Cherry Street]
Course-2: [*a3_2-Broad Street*, a2_5-Heroes Street, a5_4-Short Street, a4_7-Long Street, b7_9-Trump Street, a9_13-Welly Street, a13_14-Cherry Street]
Course-3: [*a2_5-Heroes Street*, a5_4-Short Street, a4_7-Long Street, b7_9-Trump Street, a9_13-Welly Street, a13_14-Cherry Street]
Course-4: [*a5_4-Short Street*, a4_7-Long Street, b7_9-Trump Street, a9_13-Welly Street, a13_14-Cherry Street]
Course-5: [*a4_7-Long Street*, b7_9-Trump Street, a9_13-Welly Street, a13_14-Cherry Street]
Course-6: [*b7_9-Trump Street*, a9_13-Welly Street, a13_14-Cherry Street]
Course-7: [*a9_13-Welly Street*, a13_14-Cherry Street]
Course-8: [*a13_14-Cherry Street*]

In general, the specific appearance of Listing 1 depends on the outcome of the *public static double R*() function that simulates the weights of the lanes.

The real route looks like this: a2_3-Broad Street → a3_2-Broad Street → a2_5-Heroes Street → a5_4-Short Street → a4_7-Long Street → b7_9-Trump Street → a9_13-Welly Street → a13_14-Cherry Street.

Between consecutive Courses there is a delay phase, which simulates the time required for a car to travel between two adjacent input sensors located along the vehicle route. In [19], the delay phase is programmatically implemented by using the *Thread.sleep((long) (t2-t1))* method. For a visual interpretation, let us refer to Fig. 3, where the car crosses two neighboring input sensors – 1,2 and 3 on its route. The time between the moments of crossing these sensors is the delay phase. Until the vehicle reaches the new lane, the program "waits". As soon as the vehicle crosses the input sensor of this new lane, the program immediately plots an updated route, which, in turn, is calculated anew according to a changing traffic situation.

The tracked driver is given verbal instructions (as in conventional GPS navigation) to move to the next lane along the route (these lanes are highlighted in Listing 1). Thus, the route previously calculated is connected to the new one (that is very important!), since the new route calculation uses the second position in the route previously calculated. Thus, dynamic software approximation of urban traffic is used. That allows the program to account for the dynamics of city traffic. Therefore, the routes for all vehicles in the metropolis will be plotted, which is time-optimal. As a result, even if traffic jams do not disappear, their probability will be significantly reduced.

Let us now describe some of the features of the program presented in [19], which consists of three classes – Astar, Node, and Edge. The first class is very large, because it contains the constructors of all nodes of the graph of a type

$$static\, Node\; li\_j = new\, Node("li\_j - Liberty\, avenue", 1250)\,. \tag{12}$$

Number 1250 represent the heuristic distance measured in a straight line from the intermediate node of the graph to the final node of the route. Each node of the graph is extended, i.e., it covers the adjacent lane; for example, a node $a1\_2$ includes node 1 itself and lane 1_2, which comes from this node. All that is needed in order to register cars, which, of course, are located in lanes, not at intersections. And when a real route is plotted, the starting position of the car, as well as its finish position, are set on the start position of the graph (Fig. 1).

In the *public static void initGraph()* method of the Astar class, constructors of the type *ni_j.adjacencies = new Edge[]* are specified, which form the interaction between the edges of the graph. When forming incident edges by using such constructors, traffic rules are necessarily taken into account. For example, you cannot turn from the lane *a6_5* (Fig. 1) to the lane *a5_2* as it is prohibited by traffic rules. Therefore, those constructors include only such elements of the route that are not prohibited by traffic rules.

The *public static void AstarSearch (Node source, Node goal)* method of the Astar class implements the main mission – the implementation of the *A-star* algorithm. This algorithm uses the heuristic function $f(n) = g(n) + h(n)$, where n is the current node, $g(n)$ is the distance from the starting position of the vehicle to the current node, calculated along the arcs of the graph, and $h(n)$ is the distance from the current node of the graph to the end position, calculated along a straight line. Heuristic distances are set in the program [18] by constructors of the type (12). The peculiarity of this algorithm is that it deliberately cuts off unpromising routes by introducing a heuristic distance. Additionally, that algorithm has a minor, namely linear, algorithmic complexity.

In the field of the *Node* class, class variables are declared, and the public *Node (String val, double hVal)* method returns the value of the *value* variable.

The *Edge* class returns the values of the *target* and *cost* variables, which are used when plotting a route in the *public static void AstarSearch (Node source, Node goal)* method of the *A-star* class.

Thus, the study creates a realistic prospect for solving the problem of congestion through the use of a special software algorithm focused on laying optimal routes and graphs, which, in turn, model the city's transportation network. The use of an electronic map of the city allows the GPS coordinates of each vehicle to be projected onto it. Thus, the city's transport network is under the full control of the TMC, which has a real opportunity to interact with each vehicle and constantly adjust its route, choosing the t-optimal one.

4. Conclusions

1. A multigraph model that reproduces the transport network of a city simulates the real-life lanes. Each arc of the graph receives a dynamic weight that synchronously changes in accordance with changes in traffic.
2. The use of sensors located near each intersection allows for the registration of traffic flows on the electronic map, therefore, the entire transportation network of the city is under the control of the TMC. That makes it possible to track all traffic changes and calculate dynamic optimal routes for each vehicle. As a result, urban traffic enters an equilibrium state – the Nash equilibrium state making it possible to eliminate traffic jams.

3. The working program module [19] that paves the time-optimal routes in the graph, and hence in the real transportation network has been developed. It uses the heuristic A-star algorithm, a powerful computational method of graph theory. That makes it possible to synchronize traffic flows and bring them to a qualitatively new level.
4. The study operates only with electronic services, which allows implementing the proposed method of creating optimal urban routes without significant economic costs.
5. Since all cars participating in traffic at a certain point in time will move under the control of the TMS along t-optimal routes, there will be a complete synchronization of traffic flows, which will lead to a qualitatively new state of urban traffic. That, in turn, will lead to the disappearance of traffic jams on city highways. Moreover, each driver will arrive at their destination in the shortest possible time.

REFERENCES

1. T. Xu , B. Ran , Y. Cui, Dynamic user optimal route choice problem on a signalized transportation network Transportation Engineering 13, 2023. https://doi.org/10.1016/j.treng.2022.100153
2. P. Nikolyuk, T. Neskorodieva, E. Fedorov, E. Chioma. Intellectual algorithm implementation for megacity traffic management, CEUR Workshop proceedings, Information Technology and Interactions, 2021, V.2845, https://cutt.ly/WwAwZZAp
3. P. Dutta, S. Khatua, S. Choudhuri. DB-Corouting: Density Based Coordinated Vehicle Rerouting in Smart Environment, International journal of intelligent transportation systems research V.19, 2021. doi:10.1007/s13177-021-00261-6 .
4. M. Hashemi, H. A.Karimi,. A weight-based map-matching algorithm for vehicle navigation in complex urban networks. Journal Intelligent Transportation Systems: Technology, Planning, and Operations V. 20, 2016, DOI:10.1080/15472450.2016.1166058
5. J. Pan, I. S. Popa, K. Zeitouni, C. Borcea, Proactive vehicular traffic rerouting for lower travel time. IEEE Transactions on Vehicular Technology, 62((2013). doi:10.1109/TVT.2013.2260422
6. T. Sobral, T. Galvao, J. Bordes, Visualization of Urban Mobility Data from Intelligent Transportation Systems. Sensors, V.19, 2019, https://doi.org/10.3390/s19020332.
7. M. Quddus, S. Washington, Shortest path and vehicle trajectory aided map-matching for low frequency GPS data. Transportation Research Part C: Emerging Technologies, V.55, 2015, https://doi.org/10.1016/j.trc.2015.02.017
8. H.Majid, C. Lu, H. Karim, An integrated approach for dynamic traffic routing and ramp metering using sliding mode control. Journal of Traffic and Transportation Engineering (English Edition), V.5, 2018, https://doi.org/10.1016/j.jtte.2017.08.002
9. S. Rahimipour, R. Moeinfar, S. M. Hashemi, Traffic prediction using a self-adjusted evolutionary neural network, J. Mod. Transport. V.27, 2019, DOI: https:// 10.1007/s40534-018-0179-5
10. A. Emami, M. Sarvi, S.A. Bagloee, Using Kalman filter algorithm for short-term traffic flow prediction in a connected vehicle environment, J. Modern Transport. 27, 2019, https://doi.org/10.1007/s40534-019-0193-2 .
11. A. Pompigna, F. Rupia,. Comparing practice-ready forecast models for weekly and monthly fluctuations of average daily traffic and enhancing accuracy by weighting methods, Journal of Traffic and Transportation Engineering (English Edition), V.5, 2018, http://dx.doi.org/10.1016/j.jtte.2018.01.002 .
12. S. Chavhan, P. Venkataram, Prediction based traffic management in a metropolitan area, Journal of Traffic and Transportation Engineering (English Edition), V.7, 2020). https://doi.org/10.1016/J.JTTE.2018.05.003 .
13. X. Liang, S. Ilgin Guler, V. Gayan,. An equtable traffic signal control scheme at isolated intersections using Connected Vehicle technology, Transp. Research Part C, V.110, 2020, https://doi.org/10.1016/j.trc.2019.11.005 .
14. C. Cintrano, J. Ferrer, M. López-Ibáñez, E. Alba. Hybridization of Evolutionary Operators with Elitist Iterated Racing for the Simulation Optimization of Traffic Lights Programs, Evolutionary computation, 31, 2023, https://doi.org/10.1162/evco_a_00314
15. C. Xu, J. Xu,. Intelligent terminal based intelligent traffic light system and method. Pat. CN104575066, 2015, https://patents.google.com/patent/CN104575066A/en.
16. S. S. Smith, G. G. Barlow, X.-F. Xie (2017). Smart and scalable urban signal networks: methods and systems for adaptive traffic signal control. Pat. US 9,830,813 B2.

https://www.wiomax.com/team/xie/schic/
17. H. Yu, R. Jiang, Z. Zheng Li, R. Liu, X. Chen, Automated vehicle-involved traffic flow studies: A survey of assumptions, models, speculations, and perspectives. J. of Intel. Transp. Systems, V.127, 2021, https://doi.org/10.1016/j.trc.2021.103101
18. S. Panda, A. M. Patki, K. Hushing, Traffic Management Using Swarm Intelligence and Route Selection Using Android Application// International Journal of Engineering and Innovative Technology (IJEIT), V.5, 2015, https://www.ijeit.com/Vol%205/Issue%206/IJEIT1412201512_11.pdf
19. P.K. Nikolyuk,. A-Star algorithm, 2023, https://github.com/pknikolyuk/A-Star_algorithm/blob/master/src/Astar.java
20. A. M. T. Emtenan, A. Haghighat , M. Shields, J. Shaw, P. Hawley, A. Sharma , C. M. Day, Exploratory Regression Models for Estimating Right-Turn-on-Red Volume on Exclusive Right-Turn Lanes at Signalized Intersections. Transportation Research Record, V.2677, 2022, https://doi.org/10.1177/03611981221116370
21. J. Tan, X. Shi, Z. Li, K. Yang, N. Xie, H. Yu, L. Wang, Z. Li, Continuous and Diskrete-Time Optimal Controls for an Isolated Signalized Intersection. Journal of Sensors, Article ID 6290248, 2017, https://doi.org/10.1155/2017/6290248
22. X. Ma, Y. Li, P. Chen. Identifying spatiotemporal traffic patterns in large-scale urban road networks using a modified nonnegative matrix factorization algorithm. Journal of Traffic and Transportation Engineering (English Edition), V.7, 2020, http://dx.doi.org/10.1016/j.jtte.2018.12.002 .

Маршрутизація транспортного засобу з оптимальним за часом динамічним міським шляхом руху

Ніколюк Петро Карпович *доктор фізико-математичних наук, професор кафедри інформаційних технологій Донецького національного університету імені Василя Стуса, вул. 600-річчя, 21, Вінниця-21, Україна, 21021*
e-mail: *p.nikolyuk@donnu.edu.ua* **https://orcid.org/0000-0002-0286-297X**

Ключова мета цього дослідження - синхронізувати транспортні потоки, оптимізувати використання транспортних артерій по всьому місту, запобігти заторам і супроводжувати кожен транспортний засіб до місця призначення так, щоб час, витрачений на поїздку, був мінімальним. В результаті загальний час, проведений автомобілями в дорозі, значно скоротиться, а екологічні показники відповідно покращяться.

Об'єктом дослідження є транспортна мережа міста, представлена у вигляді зваженого орієнтованого неплоского мультиграфа (ЗОНМ). Ключова перевага використання теорії графів для побудови оптимальних маршрутів базується на наступних обставинах: 1). теорія графів розробила багато алгоритмів побудови оптимальних маршрутів (алгоритм Дейкстри, алгоритм Флойда-Уоршалла, алгоритм А-зірки та ін.); 2). теорія графів є теоретичною і практичною основою логістики, в тому числі міського транспорту. Для побудови маршруту в такому мультиграфі використовується алгоритм А-зірка, який прокладає оптимальний за часом (t-оптимальний) маршрут між двома обраними вершинами графа.

Дослідження створює реальну перспективу вирішення проблеми заторів за рахунок використання спеціального програмного алгоритму, орієнтованого на прокладання оптимальних маршрутів та використання графів, які, в свою чергу, моделюють транспортну мережу міста.

Принциповим питанням є представлення транспортної мережі міста у вигляді електронної карти та відображення на ній GPS-ідентифікаторів транспортних засобів, що беруть участь у дорожньому русі. Відображення "транспортний рух міста → електронна карта" дає можливість отримати дані про рівень завантаженості транспортної мережі. Використання електронної карти міста дозволяє спроектувати на неї GPS-координати кожного транспортного засобу. Таким чином, транспортна мережа міста знаходиться під повним контролем центру управління транспортом (ЦУТ), який має реальну можливість взаємодіяти з кожним транспортним засобом і постійно коригувати його маршрут, обираючи t-оптимальний. Коригування маршруту здійснюється через канал General Packet Radio Service (GPRS) у вигляді голосових команд, як у звичайній GPS-навігації. Однак специфіка полягає в наступному: 1). навігація здійснюється в режимі онлайн; 2). прокладаються t-оптимальні маршрути з урахуванням дорожньої ситуації в кожен момент часу.

Таким чином, розроблено великомасштабну транспортну міську дорожню мережу у поєднанні з комп'ютерною програмою. Робота має прикладний характер, а її результати можуть бути використані для ефективного регулювання дорожнього руху в мегаполісах з метою мінімізації проїзду кожного транспортного засобу за обраним маршрутом.

Ключові слова: *зважений орієнтований неплоский мультиграф, GPS-навігація, алгоритм A-star, t-оптимальний маршрут, інтелектуальний трафік.*

Надійшла у першій редакції 11.03.2023, в останній - 07.04.2023.
The first version has been received on 11.03.2023, the final version - on 07.04.2023.