Multimedia Tools and Applications (2019) 78: 4693–4721
https://doi.org/10.1007/s11042-018-6714-x

Intelligent route planning system based on interval computing

Wojciech Chmiel[1] · Iwona Skalna[2] · Stanisław Jędrusik[2]

Received: 1 September 2017 / Revised: 8 August 2018 / Accepted: 21 September 2018 /
Published online: 9 October 2018

Abstract

We investigate the problem of vehicle route planning in a dynamic environment. In order to better reflect real-life situations, we assume that travel times are not known exactly, but bounded from below and from above, i.e., they are given as interval quantities. Accordingly, we develop algorithms for effective route replanning in a highly dynamic road network environment that combines traffic image processing with interval data for dynamic path optimisation. The developed algorithms are integrated into a larger system for traffic management. The efficiency of the proposed algorithms and their ability to support the dynamics of road traffic is verified using real data. The experimental research was also conducted using the microscopic, time-discrete, space-continuous traffic simulator.

Keywords Route optimisation · Video surveillance · Dynamic map · Interval arithmetic · Traffic image processing

1 Introduction

Nowadays, vehicle users demand efficient route planning in a dynamically changing environment. However, existing navigation systems consist mainly in the analysis of static

This work has been co-financed by the National Centre for Research and Development (NCBiR), Smart Infrastructure Road Innovation (RID) project no. DZP/RID-I-68/14/NCBIR/2016.

✉ Wojciech Chmiel
wch@agh.edu.pl

Iwona Skalna
skalna@agh.edu.pl

Stanisław Jędrusik
jedrusik@zarz.agh.edu.pl

[1] Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering, AGH University of Science and Technology, Kraków, Poland

[2] Faculty of Management AGH University of Science and Technology, AGH University of Science and Technology, Kraków, Poland

parameters (e.g., total route length or road type), and even recent solutions rarely use uncertain data (e.g., average driving speed in selected hours). In modern navigation systems the dynamic properties of the traffic conditions are still not considered enough. For example, Google Maps consider current traffic information only partially. Therefore, in many cases the driver must recognize traffic conditions in advance or in the place of an accident, which is highly inconvenient. Thus, in congested metropolises, the efficiency of route planning, which is particularly important in daily activities of public services such as police, fire brigades, medical emergency services, etc., is far from what is expected. When traffic conditions change, the navigation system should be able to compute the new route in a few seconds and send it to a user or decide wheatear this change has a significant impact on the travel time.

Route planning problems are often subject to various time-varying factors, such as equipment failures, traffic accidents, traffic congestion and uncertainties in road networks. Most of the existing navigation systems use the average values of travel times to reflect the dynamics of the road traffic. However, in our opinion interval values describe the dynamic nature of the road traffic better than the average values, because they take into account both the shortest and the longest travel time measured in the predefined time intervals. For example, in some cases the average value can remain unchanged even if the minimum and the maximum travel time changes. So, the interval quantity is more informative than the average. Moreover, an appropriately parametrized system based on the interval arithmetic allows better reaction to a chaotic change of the road congestion by "capturing" relevant and "filtering" irrelevant changes, i.e., by taking into account changes that indeed influence travel times and neglecting an usually large amount of data that does not influence significantly travel times. Additionally, the interval arithmetic offers the relatively easy computational model, much simpler than the stochastic approach. The main contributions of this paper are as follows:

- The system for real-time optimisation of a route in a dynamic (highly uncertain and unpredictable) environment, such as city road traffic has been proposed. The presented solution employs algorithms that use information on current traffic parameters delivered continuously from various types of sensors. The amount of available data is quite large, so we had to face the problem of their appropriate use in traffic management or drivers supporting systems. The developed solution gives the possibility of efficient use of available data and meeting the expectations of road network users.
- We have formulated theorems that are based on interval arithmetics which would enable us to optimise the decision to change the route, in the presence of uncertainty.
- We have demonstrated the usefulness of interval algebra for effective description and analysis of dynamic traffic parameters, in particular for optimising decisions in the presence of uncertainty.

Thanks to the use of interval arithmetics, the proposed algorithm enables us to determine if the route should be changed to effectively reach the goal – because the traffic in the entire network has changed (e.g., travel times in some sections) so that the current route is no longer optimal. Moreover, the developed solution would allow as to prevent situations in which the drivers use routine routes, despite the fact that their travel time changes dynamically depending on the time of day, becoming far from optimal. This, in turn, allows preventing traffic accumulation, making it easier to maintain the equilibrium in the road network.

So, the main issue addressed in this paper is to develop intelligent tools that will help the system to decide whether it should send or not a new route to a driver, based on uncertain (due to fluctuations) travel times. The developed solutions, presented in Section 5.4, are integrated with the INSIGMA (Intelligent Information System for Global Monitoring, Detection and Identification of Threats) system. The main objective of the INSIGMA system is to improve the public security and life quality by providing advanced tools that are able to detect dangerous events as well as monitor and optimise road traffic. This objective can be achieved by using dynamic data obtained from different types of sensors, such as cameras and inductive loops. The detailed description of the INSIGMA system is presented in Section 3. In the presented system the traffic parameters are determined by using the traffic image processing.

The rest of the paper is organized as follows. Section 2 presents an overview of methods for classical route optimisation in a static environment. Section 4 describes methods for modelling uncertainty. Section 5 presents the theory and algorithms developed for the Route Planning System. Section 6 is devoted to experimental tests of the proposed navigation system. The paper ends with some concluding remarks.

2 Related works

2.1 Fast route algorithms

There exist quite a great number of algorithms for fast route planning. Below we present an overview of the best-known such algorithms.

Bidirectional Dijkstra algorithm [31] searches a graph forward from a starting vertex and backward from an ending vertex until they meet. The bidirectional search is very important, since it can be used in combination with other speed-up techniques.

Separator-based multilevel method [48] defines the *shortest-path overlay graph*. The main idea of this approach is to partition the graph into smaller sub-graphs, where the shortest-path distance in the origin graph is equal to the shortest-path distance in the subgraphs. This approach is extended to multiple levels ([4, 21]). In the dynamic case (when the weight of the edge within some components changes), the complete construction of the subgraph is not needed. However, due to the fast increase of the processing time and required space, this method can be impractical for large graphs.

Highway hierarchies (HHs) algorithm [44], which is used by several commercial systems, is based on the assumption that only the main roads are used to cover the distance from an origin to a target. This method requires a special type of preprocessing and tuning, taking into account the quality of computed routes and the calculation speed. This algorithm is very similar to the heuristics used in commercial systems. Nevertheless, this method is much faster, because it finds the optimal route in milliseconds. Some modification of the *Highway Hierarchies* [44] method relies on using *distance tables* [43]. The total distance for nodes in a certain level can be precomputed and stored in a table. This algorithm can be two times faster than the pure HHs search.

Reach-based routing [18] method resembles the HHs method and has some variants (including the dynamic ones), that use the geometric distance or the shortest-path distance.

Because of the high computing cost(data preprocessing), the *Reach-based Routing* method is not preferred for large networks.

Transit-based routing [39] method is based on the rule that when driving to a distant place, the origin place is left through a limited number of significant traffic junctions (called *transit nodes*) hence the calculation of the shortest path is limited to some table lookups.

Edge labels [48] method precomputes information for an edge that specifies a set of nodes which are a superset of all the nodes that lie on the shortest path starting within this edge. In [48] this superset is specified by an angular range. In this method many types of geometric containers [29, 54] can be used; however, it requires computation the shortest path for each pair of the nodes.

Landmark-A* method [16] uses precomputed distances to the set of nodes, uniformly distributed over far ends of a network. This approach enables the algorithm to gain a considerable speed-up for route planning as well as much less preprocessing time.

Precomputed cluster distances (PCD), is based on partitioning of a network into disjoint clusters [34]. The PCD method can be used in prune search, because it provides upper and lower bounds for distances. The prune search gives the speed-up comparable to *Landmark-A**; however, using smaller memory space.

Schultes and Sanders [47] introduced a dynamic technique for the fast route planning in a large road networks, called the *Dynamic Highway-Node Routing* (DHN). This method uses a special type of data preprocessing. Thanks to this, if the travel times change, the already preprocessed data will be updated in milliseconds. This method is based on the generalization of the *Separator-based Multilevel* method [48] for the *highway-node routing* and allows finding the shortest path in one microsecond on average. The most interesting and important for the problem presented in this paper is that the authors proposed two types of settings in a dynamic case i.e. when the travel time changes because of the traffic fluctuation. The former is the *server setting*, where the cover graphs are updated so that the algorithm could once more produce correct results. The latter one is a *mobile setting*, where changes are not introduced. It should be mentioned that in the server setting case, updating of the preprocessed data requires a very small amount of time. In the urban route case, this time practically does not depend on the number of the changed edges. The method based on the DHN algorithm will be used in the system presented in this paper for calculating new routes.

The most interesting recent advances in the route planning and the traffic optimisation, based on both the exact and approximation methods, such as the dynamic programming, linear programming and the approximation methods, are presented below. In the future, some of these methods can be implemented in the INSIGMA system to improve its robustness.

Dinitz et al. in [14] proposed a hybrid algorithm based on the Bellman–Ford and Dijkstra algorithms, improving the running time bound of the Bellman–Ford algorithm for graphs with a sparse distribution and negative cost edges. Delling et al. in [12] proposed the shortest path algorithm which supports turn costs, enables real-time queries, and can incorporate a newly defined cost function in a few seconds. It enables supporting real-time traffic updates and personalized optimisation functions.

Buchhold et al. in [7] studied the equilibrium flow patterns in road networks. They consider the problem known as the traffic assignment, where a flow on each road segment is at a equilibrium. This approach enables developing interactive transportation and urban planning applications and in the future, real-time demand-responsive public transit systems. Angelelli et al. in [1] proposed the linear programming model for a proactive route guidance approach that minimises both congestion and the user travel inconvenience. In this approach the algorithm assigns paths other than the shortest (least-duration) to vehicles in order to reduce the congestion in the road network while keeping the user travel inconvenience on a reasonable level. Rahmani et al. in [42] investigated the sensitivity of the estimated travel times and proposed a fixed point formulation of the simultaneous path inference and the travel time estimation problem. Lee et al. in [30] proposed the Gini coefficient, which is a well-known measure of the statistical dispersion, for measuring and evaluating the travel time reliability.

Zeng et al. in [57] proposed the optimisation model for the energy-efficient vehicle driving from the origin to the destination within a certain period of time. In order to solve this problem, the dynamic programming method was used. In the proposed model the stop signs, traffic lights, turns and curved segments, roads with different grades and speed limits, and torque operation are defined as constrains of the decision variables.

Ma et al. in [32] proposed a generalized Markov chain approach for estimation of the trip travel time distributions between arbitrary origin-destination pairs at arbitrary times, from a link or segment travel time distributions. The proposed approach can be used under the situation when data sample is limited, especially for urban road networks. Chen et al. in [8] proposed a copula-based approach that incorporates the stochastic characteristics of the segments' travel time to model an urban arterial travel time distribution (TTD). It serves as a basis for the travel time reliability estimating and can be implemented in various applications in practice including the real time dynamic control of traffic systems, the congestion and the incident detection. Sever et al. in [51] proposed a hybrid approximate dynamic programming (ADP) with a deterministic look-ahead policy and value function approximation for the dynamic shortest path problem with travel time-dependent stochastic disruptions. The considered problem is formulated as a discrete time finite horizon Markov decision process.

Chen et al. in [9] proposed the moment–matching–based hybrid genetic algorithm (MHGA) to search for the reliable shortest path (RSP) in stochastic road networks with link correlations. In the presented approach the link travel times are characterized by lognormal distributions and the correlations between links are considered in the form of the correlation coefficient. Dib et al. in [13] proposed the meta-heuristics based on the genetic algorithm (GA) and the variable neighbourhood search (VNS) for the shortest path problem. Zhang et al. in [58] proposed the hybrid algorithm based on combining the artificial immune and the ant colony optimisation algorithms (IACO) to solve the multi-objective emergency grain distribution vehicles routing problem. This algorithm uses the Pareto non-dominated sorted population and calculates the degree of crowding which preserves the density of population by determining whether a solution is located in a crowded area of the known Pareto solutions or not. Miao et al. in [35] proposed the optimisation algorithm based on the evolution paradigm and along with an adaptive real-time optimisation strategy to solve the vehicle macroscopic motion planning (VMMP) problem. This problem optimise simultaneously a vehicle route and a speed using both traffic data and vehicle characteristics to improve the fuel economy for a given expected trip time. The optimisation model utilises the current traffic flow data. Marinakis et al. in [33] proposed a hybridized version of the particle swarm optimisation algorithm (PSO) with a variable neighborhood search (VNS) for solving the constrained shortest path problem. Huang et al. in [24] proposed a time-delay neural network (TDNN) framework for the time-dependent shortest path problem.

Nowadays, the importance of the route optimisation in the dynamic environment is on the increase. Han et al. in [19] considered the problem of the path replanning in a dynamic environment. This problem is similar to the one considered in this paper, but refers to the obstacles moving on the triangular grid instead of the road traffic. The authors proposed a decision-making methodology for a path regeneration which enables making intelligent decisions to reduce the path length without updating the path over every time unit interval. Hu et al. in [22] proposed the ripple-spreading algorithm (RSA) for the path optimisation problem. The RSA algorithm in contrast to the most deterministic top-down centralized path optimisation methods, such as the Dijkstra's algorithm, is a bottom-up decentralized agent-based simulation method which guarantees the global optimal solution with a very good scalability. For this problem Hu et al. in [23] proposed the RSA algorithm as a realization of the co-evolutionary path optimisation which achieves in a single run the optimal actual travelling trajectory in a given dynamic routing environment.

Idri et al. in [25] proposed the shortest path algorithm for the time-dependent and multi-modal networks. Sarbazi-Azad et al. in [45] proposed for the all-pairs shortest path problem and the single-pair shortest path problem the method which exploits the GPU computational power. All the shortest paths for the million vertex graphs can be computed in 360 seconds and a single-pair shortest path query on the same graph is answered in a quarter of a millisecond. Huang et al. in [24] considered a path selection in the road network as a integrated decision in the time-dependent vehicle routing problem, denoted as the path flexibility (PF). In the proposed approach the path selection decision depends upon the departure time at the customers and the congestion levels in the relevant road network.

Nowadays, the number of methods for path planning in the dynamic enviroment considerably increases ([2, 40, 41, 55, 59]). But, they generally refer to the problem of path planning in the dynamic environment in the context of autonomous mobile robots navigation or unmanned aerial vehicles. A detailed survey of the recent advances in algorithms for the route planning in transportation networks is presented in [5].

3 Centralized system for route planning

The Intelligent Route Planning System (RPS) is a part of the INSIGMA (Intelligent Information System for Global Monitoring, Detection and Identification of Threats) project [10, 11]. The main objective of the latter system is detection, identification, monitoring of objects and threats in a dynamic environment, such as road networks and public areas. Solutions developed in the INSIGMA project present a holistic approach to the broadly understood security and ecology in the sense of energy savings. The project includes solutions from several areas, such as people recognition, suspicious behaviour recognition, vehicles identification, traffic monitoring, threats identification, data security, route optimisation and multimedia data mining.

The important objective of the INSIGMA project is calculation of the best route for individual drivers in the dynamic road traffic. On the basis of the parameters describing the state of the current traffic obtained by several types of the detectors, such as video-detectors, acoustic arrays, microwave sensors, GPS trackers and many others, the RPS system (Fig. 1) provides the drivers with the best route from origin to destination. This route is calculated using various types of the optimisation algorithms, which take into consideration both the user preferences and availability of the traffic information.

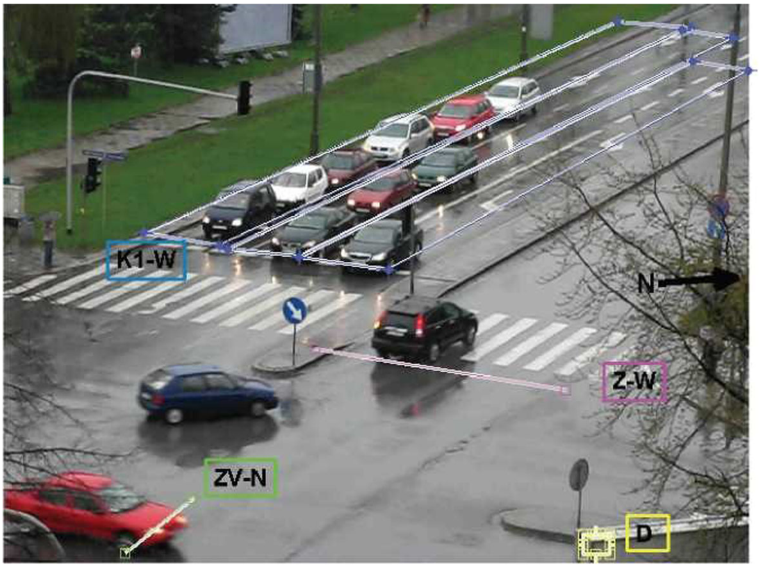


Fig. 1 Traffic analysis using video-detectors in INSIGMA project

Figure 2 presents the basic elements of the Route Planning System [53]. The most important element of the RPS system is the *Decision Module* which takes a decision if the route contained in the *Calculated Routes* should be recalculated or/and sent to the drivers using the *Client Interface*. If some travel times change the routes in the *Calculate Routes* module are updated using, i.e., the DHN algorithm (see Section 5.4). The *Algorithms* module embeds

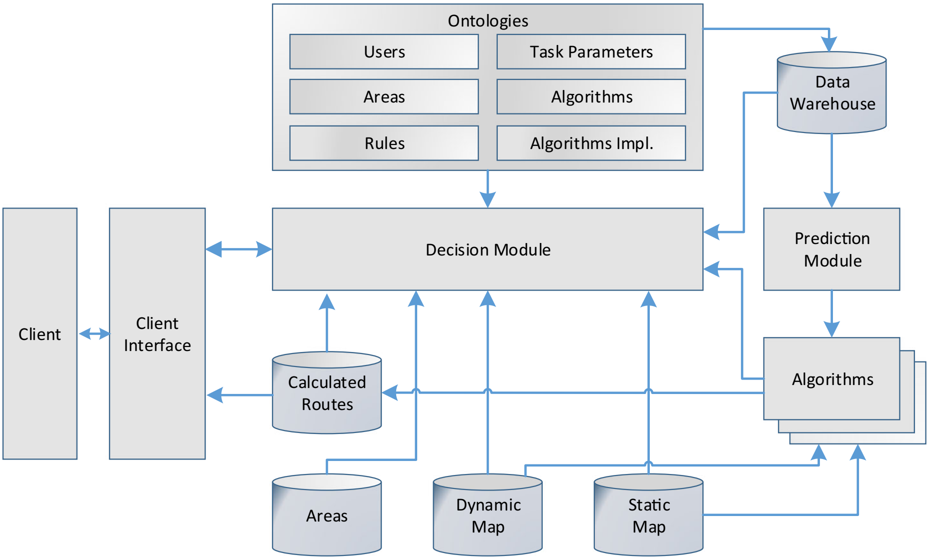


Fig. 2 Most important modules of Route Planning System

both exact and approximation algorithms for the path optimisation (e.g., DHN) and algorithms for prediction of traffic parameters (e.g., neural network, decision tree). The *Static Map* stores the physical description of the roads and roadsides while the dynamic parameters, such as the average speed or the travel time, are stored in the *Dynamic Map*. The data gathered by these modules and the *Data Warehouse* is accessible through the web service-based interfaces included in the *Client Interface*. Additionally, the *Dynamic Map* and the *Data Warehouse* provide an OLAP interface allowing for an *ad-hoc* multi-dimensional analysis. The semantic descriptions of the data stored in the RPS system, such as detectors and monitoring parameters, roads preferred by the users, tasks, algorithms, users and rules are included in the ontologies defined on the basis of the Web Ontology Language (OWL) are stored in the *Ontologies* module. The *Data Warehouse* stores the historical data for the future analysis by the *Prediction Module*. This module enables predicting the future traffic state on the basis of the multi-dimensional data analysis using the machine learning methods. The *Areas* module stores the special areas of the city road network e.g. those characterised by an unexpected traffic fluctuation or available only for a special type of users (police, medical services, etc.).

4 Route Planning System under uncertainty

Significant acceleration of the response time of the shortest path queries is obtained by using precalculated paths and specially prepared data stored in the RPS database (see, e.g., [3, 15, 46]). However, unexpected traffic fluctuations can affect big areas of the road network. If those fluctuations are frequent, travel times become uncertain. In order to avoid slowing down of the traffic, it is crucial to be able to answer, on the basis of uncertain information, whether a new shortest path should be recalculated and sent to the car driver. The main issue addressed in this paper is determining the conditions under which a new route should be calculated and sent to the user. It enables road traffic smoothing, reducing the computational effort required to the shortest path calculation and additionally minimizes the network traffic between RPS application servers (the *Client Interfaces* in Fig. 2) and thousands of customers. In order to be able to address this issue, the appropriate way of modelling uncertain travel times, which are stored in the *Data Warehouse*, must be selected. In general, there are three most popular ways of modelling uncertainty: stochastic approach, interval approach and fuzzy approach.

The stochastic time dependent route planning uses time-dependent road-graphs, where the average and the standard deviations of the travel time are stored. He et al. [20] showed that the travel time distribution is best described by a left-centred distribution (with a larger probability of deviations to high realizations of the travel times rather than to low travel times). In the stochastic models, the uncertainty of the travel time of one edge progresses through a route and influences the cost of other edges. Calculating the impact of the variance of the edge travel time on the variance of the whole path is not practically feasible. The main deficiency of this approach is that computing the variance of a path requires for all edges in the path a comprehensive calculation of all potential combinations of the travel times. This approach is time and space costly for large road-graphs unless the travel time profiles have a simple form [60] and travel times are Erlang$_k$ or exponentially distributed.

In the fuzzy approach, the travel times are modeled using fuzzy numbers [56] or fuzzy random variables. Initially, fuzzy numbers were used in a unified system for planning route based on uncertain information. Finally, interval numbers were used instead, despite the fact that the fuzzy numbers enable a more precise description of the traffic. The fuzzy numbers give too excessive computational and memory overhead. As mentioned earlier, the response time has fundamental significance because the Route Planning System works nearly as a real-time system.

5 Main result

This section introduces the preliminary theory of interval route planning and presents algorithms developed for the RPS system integrated within the INSIGMA system.

5.1 Interval arithmetic

When dealing with real-world problems, one can rarely avoid uncertainty. At the empirical level, uncertainty is an inseparable companion of any measurement. The difference between the measured and the actual values is called a *measurement error*. The absolute value of the measurement error can usually be bounded. It is therefore guaranteed that the actual (unknown) value of the desired quantity belongs to the interval with the midpoint being the measured value, and the radius being the upper bound for the (absolute value) of the measurement error. Interval arithmetic is also very useful when dealing with rounding errors that are inherent in a floating-point computation. Therefore, imprecision, approximation, or uncertainties in the knowledge of the exact values of physical and technical parameters can be modelled conveniently by means of intervals. The role of compact intervals as independent objects is continuously increasing in numerical analysis when verifying or enclosing solutions of various mathematical problems or when proving that such problems cannot have a solution within a given domain. Interval arithmetic has been successfully used in chemical engineering [52], in the domain of spacecraft systems [28], in economics [26], in the simulation of physical and diagnostic reasoning, in the analysis and control of structural systems [17]. Nowadays, application areas of interval methods include electrical engineering, industrial engineering, control theory, remote sensing, experimental and computational physics, chaotic systems, celestial mechanics, signal processing, computer graphics, robotics, computer-assisted proofs, power systems, finance, route planning etc.

In the interval approach to route planning, an uncertain travel time $\widetilde{t}$ is represented by a closed interval $\mathbf{t} = [\underline{t}, \overline{t}]$. No further information which values are more likely to occur is available. It needs to be remarked that throughout this paper, interval quantities will be denoted, according to the interval standard, by bold small cap letters, and $\mathbb{IR}$ will denote the set of all closed intervals.

Elementary arithmetic operations $\circ \in \{+, -, *, /\}$ are defined on the set of intervals using the endpoints [38] in the following way:

$$\begin{aligned}
\boldsymbol{x} + \boldsymbol{y} &= [\underline{x} + \underline{y}, \overline{x} + \overline{y}],\\
\boldsymbol{x} - \boldsymbol{y} &= [\underline{x} - \overline{y}, \overline{x} - \underline{y}],\\
\boldsymbol{x} \cdot \boldsymbol{y} &= [\min\{\underline{x}\,\underline{y}, \underline{x}\,\overline{y}, \overline{x}\,\underline{y}, \overline{x}\,\overline{y}\}, \max\{\underline{x}\,\underline{y}, \underline{x}\,\overline{y}, \overline{x}\,\underline{y}, \overline{x}\,\overline{y}\}],\\
\boldsymbol{x}/\boldsymbol{y} &= \boldsymbol{x} \cdot [1/\overline{y}, 1/\underline{y}], 0 \notin \boldsymbol{y}. \qquad (1)
\end{aligned}$$

In order to be able to employ interval quantities in the route planning system, the ordering of intervals is required. Despite the fact that there is no natural ordering among the intervals, several comparison methods have been developed. Ordering of intervals is of great interest in practical modelling and optimisation of real world cases. Below, the probabilistic approach proposed in [49] is presented.

Let $\boldsymbol{a} = [\underline{a}, \overline{a}]$ and $\mathbf{b} = [\underline{b}, \overline{b}]$ be intervals and let $a \in \boldsymbol{a}$ and $b \in \mathbf{b}$ be independent uniform random variables. If $\boldsymbol{a}$ and $\mathbf{b}$ overlap, then some disjoint subintervals can be distinguished. The fall of random variables a and b in the subintervals $[a_1, b_1]$, $[b_1, a_2]$, $[a_2, b_2]$ may be treated as a set of independent random events. Let $H_k : a \in \boldsymbol{a}_i, b \in \mathbf{b}_j$, for $k = 1$ to n, be random events, where $\boldsymbol{a}_i$ and $\mathbf{b}_j$ are certain subintervals of intervals $\boldsymbol{a}$ and $\mathbf{b}$ in accordance with $\boldsymbol{a} = \mathbf{b}igcup_i\boldsymbol{a}_i$ and $\mathbf{b} = \mathbf{b}igcup_i\mathbf{b}_i$. Let $P(H_k)$ be the probability of an event H_k, and $P(\mathbf{b} > \boldsymbol{a}|H_k)$ be the conditional probability of $\mathbf{b} > \boldsymbol{a}$ given H_k. Then, the composite probability may be expressed as follows:

$$P(\mathbf{b} > \boldsymbol{a}) = \sum_{k=1}^{n} P(H_k)P(\mathbf{b} > \boldsymbol{a}|H_k) \tag{2}$$

The resulting formula for the case of overlapping is as follows:

$$P(\mathbf{b} > \boldsymbol{a}) = 1 - \frac{1}{2}\frac{(a_2 - b_1)^2}{(a_2 - a_1)(b_2 - b_1})) \tag{3}$$

The results obtained in [49] for all possible interval overlapping are shown in Table 1, where obvious cases (without overlapping) are omitted.

The probability based approach can be treated as a framework for elaboration of constructive methods of interval comparison in various special situations. Some aspects of the interval comparison and ordering group of intervals, based on this approach, are presented, e.g., in [50].

5.2 Interval approach to the shortest path problem

If it is possible to find a route not affected by travel time changes, special assumptions about the searched route must be made. Montemanni et al. [37] define two types of optimisation problems: the absolute robust shortest path problem and the robust deviation shortest path problem.

Let us assume that some scenario r is a feasible realization of edge costs. The *absolute robust shortest path* (*ABS*) is the path which for all possible scenarios r has the smallest (among all paths from the origin to the destination) maximum cost.

The difference between the cost of the path from origin to destination and the cost of the shortest path from origin to destination in the scenario r is called the *robust deviation*.

Table 1 Typical cases of interval comparison [49]

Case	$P(\boldsymbol{a} > \mathbf{b})$	$P(\boldsymbol{a} = \mathbf{b})$
$b_1 < a_1 = a_2 < b_2$	$\frac{b_1-a_1}{a_2-a_1}$	0
$a_1 < b_1 = b_2 < a_2$ and $b1 < a2$ and $b1 = b2$	$\frac{b_2-a_1}{b_2-b_1}$	0
$a_1 \leqslant b_1 \wedge b_2 \leqslant a_2$	$\frac{b_1-a_1}{a_2-a_1} + \frac{1}{2}\frac{a_2-a_1}{b_2-b_1}$	$\frac{b_2-b_1}{a_2-a_1}$
$b_1 \leqslant a_1 \wedge a_2 \leqslant b_2$	$\frac{b_2-a_2}{b_2-b_1} + \frac{1}{2}\frac{a_2-a_1}{b_2-b_1}$	$\frac{a_2-a_1}{b_2-b_1}$
$a_1 \leqslant b_1 \wedge a_2 \leqslant b_2 \wedge b_1 \leqslant a_2$	$1 - \frac{1}{2}\frac{(a_2-b_1)^2}{(a_2-a_1)(b_2-b_1)}$	$\frac{(a_2-b_1)^2}{(a_2-a_1)(b_2-b_1)}$
$b_1 \leqslant a_1 \wedge b_2 \leqslant a_2 \wedge a_1 \leqslant b_2$	$1 - \frac{1}{2}\frac{(b_2-a_1)^2}{(a_2-a_1)(b_2-b_1)}$	$\frac{(b_2-a_1)^2}{(a_2-a_1)(b_2-b_1)}$

The *robust deviation of the shortest path (RD)* from origin to destination is the path, which for all possible scenarios r has the smallest (among all paths from origin to destination) *maximum* robust deviation.

A branch and bound algorithm was proposed for ABS and RD problem in [37] and a mixed integer program was defined for these problems in [36]. Both ABS and RD problems are *NP-hard*, and hence finding a solution to bigger graphs is practically impossible. In the early version of the RPS system, this algorithm was used for finding the shortest path; however, it provides too pessimistic results.

5.3 The travel time change effect under interval uncertainty

Let the model of a road network be defined by a graph $G = (V, E)$ with a weight function $w : E \to \mathbb{R}$, which assigns a real value (a weight) to each edge $(u, v) \in V \times V$ of the graph G. It can be assumed that edge weights, which defines the travel time, can change in time (i.e., they are functions of time).

Let us consider a typical situation occurring in a city traffic. A vehicle drives from a start point s (vertex in the graph) to a target point u, using the shortest route calculated on the basis of the instantaneous travel time. If during the journey, the travel time changes for some road (modelled as an edge of the graph), the shortest path may change as well. The following theorems allow us to state whether the new shortest path needs to be recalculated and sent to a driver.

Theorem 1 (Edge elongation) *Let a graph $G = (V, E)$ be given. Let p be the shortest path from a start vertex s to a target vertex u in G, such that the length of p is $L(s; u; p; G)$ and p contains an edge $a = \{v_1, v_2\}$, i.e., $p = (s, \ldots, v_1, v_2, , \ldots, u)$. Let then p' be the shortest path of length $L(s; u; p'; G')$ from s to u in the graph $G' = (V, E \backslash \{a\})$. Now assume that the edge a was elongated by $\Delta a > 0$. Then, the path p remains the shortest path in G if*

$$\Delta a < L(s, u, p', G') - L(s, u, p, G).$$

Proof (Theorem 1) The following two cases can be distinguished:

i. The edge a is not a part of the path p. Then, the path p remains the shortest path from s to u in G, regardless of the length of the edge a.
ii. The edge a is a part of the path p. Then, the elongation of the edge a by Δa causes the respective elongation of the path p, and hence the path p might no longer be the shortest path in G. This will be the case if there exists a path p' that does not contain the edge a and is shorter than p, i.e., $L(s, u, p, G) + \Delta a > L(s, u, p', G)$. To determine whether such a path exists, it is enough to compute the shortest path in the graph $G' = (V, E \setminus \{a\})$.

□

Theorem 2 (Edge shortening) *Let a graph $G = (V, E)$ be given. Let p be the shortest path from a start vertex s to a target vertex u in G, such that the length of p is $L(s, u, p, G)$ and let $G' = (V, E \backslash \{a\})$, where $a = \{v_1, v_2\}$. Let as assume that the edge a was shortened by $\Delta a > 0$. Then, the following holds:*

(i) *If a is a part of p, then p remains the shortest path in G, regardless of the size of Δa.*

(ii) *If a is not a part of p, then p remains the shortest path if the following inequalities are fulfilled:*

$$L\left(s, v_1, G'\right) + l(a) - \Delta a + L\left(v_2, u, G'\right) > L(s, u, G), \tag{4}$$

$$L\left(u, v_1, G'\right) + l(a) - \Delta a + L\left(v_2, s, G'\right) > L\left(s, u, G\right), \tag{5}$$

where $l(a)$ is the length of the edge a before shortening, $L(s, v_1, G')$ $(L(u, v_1, G'))$ is the length of the shortest path between s and v_1 (u and v_1), and $L(v_2, u, G')$ $(L(v_2, s, G'))$ is the length of the shortest path between v_2 and u (v_2 and s) in G'.

Proof (Theorem 2)

(i) The edge a is a part of the path p. Since p is the shortest path from s to u, hence $L(s, u, p, G) < L(s, u, p', G)$ for each path p' from s to u in G. The shortening of the edge a will shorten the path p and other paths that contain the edge a (paths that do not contain the edge a can be neglected). The new lenght of the path p and other considered paths are, respectively, $L(s, u, p, G) - \Delta a$ and $L(s, u, p', G) - \Delta a$. Now, since

$$L(s, u, p, G) < L(s, u, p', G),$$

thus

$$L(s, u, p, G) - \Delta a < L(s, u, p', G) - \Delta a,$$

which means that p is still the shortest path from s to u in G.

(i) The edge a is not a part of the path p. Then, p may no longer be the shortest path from s to u in G. A path that contains the edge a can become the new shortest path. Therefore, it must be checked if the two paths depicted in Fig. 3 are not shorter than path p. However, since every subpath of the shortest path is the shortest path between the two of any vertices in this path(according to Bellman's Principle of Optimality [6]), it is enough to calculate the shortest path in the graph $G' = (V, E \setminus \{a\})$ from the start vertex s to v_1, the shortest path from v_2 to u (see Fig. 3(1)) and, respectively, the shortest path from the vertex s to v_2, the shortest path from v_1 to u (see Fig. 3(2)).

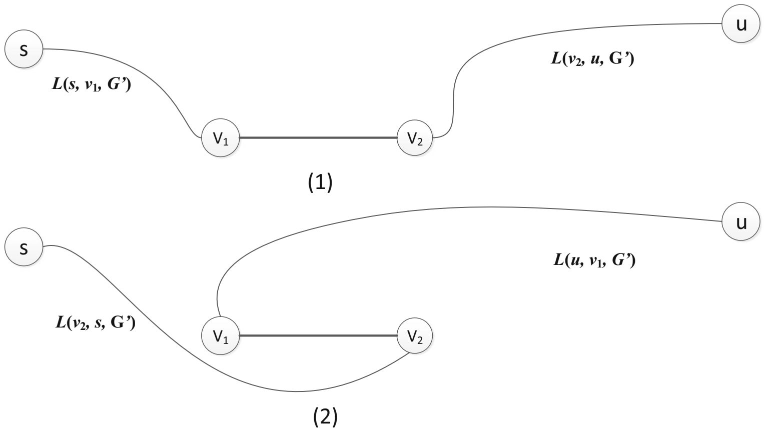


Fig. 3 Two cases for path that includes edge $a = \{v_1, v_2\}$: (1) when the route first visits the node v_1 or (2) when the route first visits the node v_2

By an appropriate combination of both path pairs with the edge a, the shortest path between s and u, which contains the edge a, is obtained. If one of these two paths are shorter than p, then this path is the shortest path between s and u after shortening the edge a.

□

5.4 Interval route planning algorithm

The developed Route Planning System obtains information about the current traffic state from video-sensors. On the basis of the parameter values from RPS sensors, the travel time for each route (edge (i, j)) is calculated and stored in the *Dynamic Map*. The structure of roads, e.g., junction topology, is included in the *Static Map*. For each route in the *Static Map*, an interval number describing minimal and maximal measured travel time in the last $(n-1)\delta$ seconds is stored:

$$\mathbf{T}_{ij} = [\min(T_{ij}(t_s - i\delta)), \max(T_{ij}(t_s - i\delta))], i = 0, 1, \ldots, n-1 \tag{6}$$

Apart from this, the simple moving average measured travel times in the last $(n-1)\delta$ seconds $\overline{T}_{ij}(t_s)$ are stored too:

$$\overline{T}_{ij}(t_s) = \frac{1}{n}\sum_{i=0}^{n-1} T_{ij}(t_s - i\delta) \tag{7}$$

there $t_s - (n-1)\delta, \ldots, t_s - \delta, t_s$ are the moments of the last n travel time measurement. It is typically assumed that the measurement period δ ranges from 20 to 30 seconds and n ranges from 10 to 15.

The first version of the RPS system used a branch and bound algorithm for the *robust shortest path* (RD) (see Section 5.2) proposed by Montemanni et al. in [37]. However, because of the too pessimistic approximation of the shortest-path, in the current version of the RPS system for computing the shortest route, the *Dynamic Highway-Node Routing* (DHN) algorithm, proposed by Schuters and Sanders [47] in the server scenario version, is used. This algorithm has an ability to update fast and easily the travel time of the shortest path when one or a few edges are changed (see Section 2.1).

The simple moving average time (the (7)) is used for computing the shortest path. In order to make the system more robust for small travel time changes, the decision about sending a new shortest path to the user is made on the basis of *interval travel times* and Theorems 1 and 2. For clarity, to represent an interval values bold characters will be used. The values defined in these theorems are modeled by interval values in the following way:

(i) Let $p = (a_1, a_2, \ldots, a_k)$ is the shortest path (list of edges a_i) obtained using DHN algorithm, where each edge travel time takes the value of simple moving average (7).

(ii) In the case of elongation of the edge (see Theorem 1), the values in equation $\Delta = \mathbf{L}(s, u, p', G') - \mathbf{L}(s, u, p, G)$ are treated as the interval values (the 6). Particularly, $\mathbf{L}(s, u, p', G')$ and $\mathbf{L}(s, u, p, G)$ are interpreted as the sum of the interval weight of the edges included in the shortest path obtained using the algorithm DHN. The theory of interval algebra is used for determining whether the conditions defined in Theorem 1 are met or not.

(iii) In case of shortening (Theorem 2) all values are interpreted as interval values (the (6)). Particularly $\mathbf{L}(s, v_1, G')$, $\mathbf{l}(v_1, v_2)$, $\Delta\mathbf{a}$, and $\mathbf{L}(v_2, u, G')$, $\mathbf{L}(s, u, G)$, $\mathbf{L}(u, v_1, G')$,

$\mathbf{l}(v_1, v_2)$, $\Delta\mathbf{a}$, $\mathbf{L}(v_2, s, G')$, $\mathbf{L}(s, u, G)$, $\mathbf{L}(s, u, p', G')$ and $\mathbf{L}(s, u, p, G)$ are interpreted as intervals. The theory of interval algebra is used for determining whether the conditions defined in Theorem 2 are met or not.

For comparison and ordering of the intervals values the method presented in Section 5 was used. This method is based on the probabilistic approach to interval ordering (see Table 1) and was proposed by Sevastjanov and Róg [49]. The probability used during verification of conditions defined in Theorems 1 and 2 is a parameter which should be specified before running the algorithm.

5.5 Update path decision algorithm

The algorithms presented in Figs. 4 and 5 describe the decision flow between the parts of the Route Planing System (Fig. 2) to make the decision about sending (or not) the new path to the customers.

The variables used in Fig. 4 refer to the variables defined in Theorem 1. Interval value $\Delta\mathbf{a}$ determines the travel elongation time on some edge, the interval value $\mathbf{L}$ ($\mathbf{L'}$) is the shortest path in the graph G (G') and interval value Δ is the length difference between the two shortest paths in the graph G and the graph G' between the same pair of nodes.

The variables used in Fig. 5 refer to the variables defined in Theorem 2. Interval value $\Delta\mathbf{a}$ determines the travel decreasing time on some edge, where $\mathbf{l}(v_1, v_2)$ is the length (the interval value) of the edge a before shortening, $\mathbf{L}(s, v_1, G')$ ($\mathbf{L}(u, v_1, G')$) is the length (the interval value)of the shortest path between s and v_1 (u and v_1), and $\mathbf{L}(v_2, u, G')$

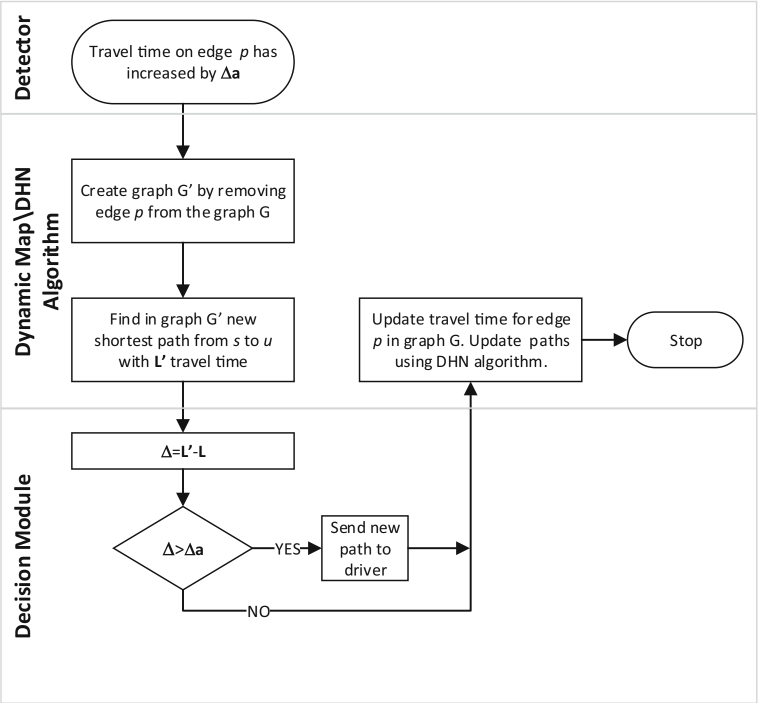

Detector
Travel time on edge p has increased by Δa
Dynamic Map\DHN Algorithm
Create graph G' by removing edge p from the graph G
Find in graph G' new shortest path from s to u with L' travel time
Update travel time for edge p in graph G. Update paths using DHN algorithm.
Stop
Decision Module
Δ=L'-L
Δ>Δa
YES
Send new path to driver
NO

Fig. 4 Decision flow between components of Route Planing System in case of travel time elongation

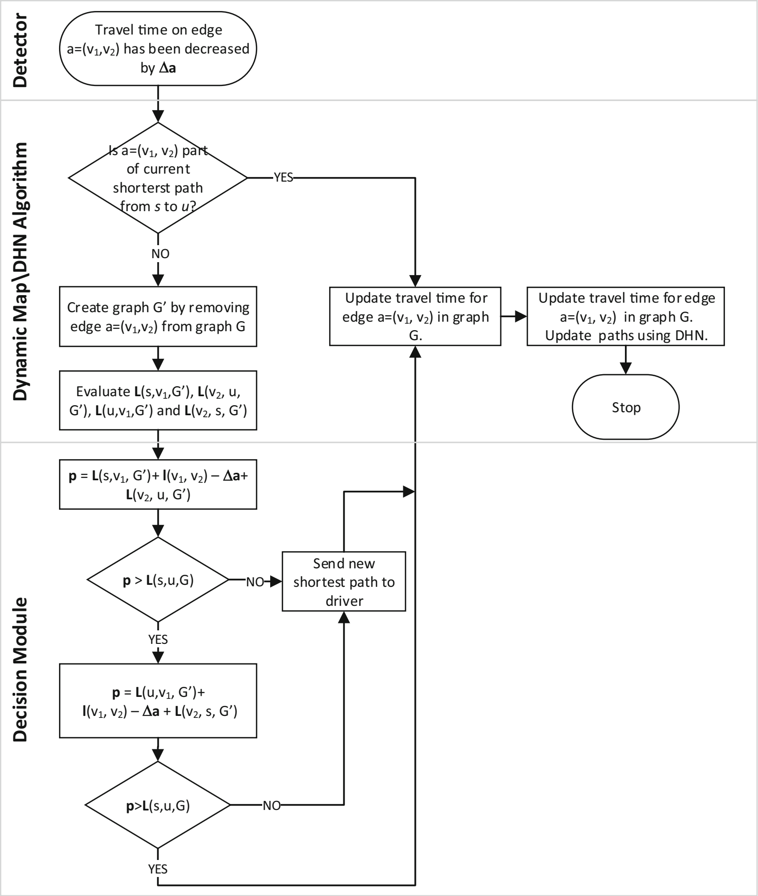

Detector
Travel time on edge a=(v1,v2) has been decreased by Δa
Dynamic Map\DHN Algorithm
Is a=(v1, v2) part of current shorterst path from s to u?
YES
NO
Create graph G' by removing edge a=(v1,v2) from graph G
Update travel time for edge a=(v1, v2) in graph G.
Update travel time for edge a=(v1, v2) in graph G. Update paths using DHN.
Stop
Evaluate L(s,v1,G'), L(v2, u, G'), L(u,v1,G') and L(v2, s, G')
Decision Module
p = L(s,v1, G')+ l(v1, v2) – Δa+ L(v2, u, G')
p > L(s,u,G)
NO
Send new shortest path to driver
YES
p = L(u,v1, G')+ l(v1, v2) – Δa + L(v2, s, G')
p>L(s,u,G)
NO
YES

Fig. 5 Decision flow between components of Route Planing System in case of travel time shortening

($\mathbf{L}(v_2, s, G')$) is the length (the interval value) of the shortest path between v_2 and u (v_2 and s) in G'.

The figures are split into rectangles to indicate a place of the decision algorithm execution, i.e., *Detector*, *Dynamic Map*, DHN algorithm or *Decision Module*.

In the first case (elongation, see Fig. 4) information about the change of the edge travel time to the *Dynamic Map* is sent by the detectors. The *Dynamic Map* determines (using methods from the *Algorithms Module*) the shortest path in the graph obtained through removing the changed edge. If the difference between lengths of these two paths exceeds the edge increase, the *Decision Module* sends the new shortest path to the customers. Then, the algorithm DHN recalculates the subset of the shortest paths kept in memory to accelerate further calculations.

In the next case (shortening, Fig. 5), the situation is a little more complicated and requires considering three sub-cases. If the shortened edge is a part of the shortest path followed by customers, this path always remains the shortest path and nothing is sent to the customers. The information in the Dynamic Map is updated and the DHN algorithm recomputes the subset of the shortest paths kept in memory. If the shortened edge is not a part of the shortest paths followed by the customers, the two next subcases have to be considered. In both subcases the two shortest paths from the graph obtained by removing the shortened edge are computed. These paths are determined by the *Algorithm Module* and passed to the *Decision Module*. The *Decision Module* checks if the conditions (i) and (ii) from Theorem 2 are fulfilled. If so, nothing is sent to the customers. Otherwise, a new path is sent to them. Then, the DHN algorithm recomputes the subset of the shortest paths kept in memory. It should be emphasized that all values used in the presented above comparison are interval values and are compared using the rules defined in Table 1.

6 Computational experiments

Several tests of the parameters controlling the Route Planning System were designed and performed. The efficiency of path predictions was verified on the basis of the real traffic network of the city of Kraków. The traffic flow was modelled using the data registered by detectors belonging to the Board of Infrastructure and Transport located in the streets of the city of Kraków. The experimental research was conducted using the space-continuous traffic simulator - SUMO (Simulation of Urban MObility) developed by the Institute of Transportation Systems in German Aerospace Center. SUMO models the movement of every single vehicle in the street, assuming that the behaviour of the vehicle depends on both: the vehicle's physical abilities to move and the driver's controlling behaviour.

Two scenarios for management of a travel time fluctuation, presented in Fig. 6, were investigated during experiments. The first scenario that does not use optimisation is as follows: if the travel time changes, then a new shortest path is determined and sent immediately to the driver (Fig. 6b). The second scenario uses *Decision Module* which, on the basis of the Theorem 1 (Fig. 4) and Theorem 2 (Fig. 5), determines the decision about the change of the travel route by the driver (Fig. 6a). In both cases SUMO Road Traffic Simulator plays the role of the detectors located in the streets, which provide information about the current road traffic.

The proposed test suite for the RPS system uses:

(i) different size of the traffic networks,
(ii) different values of the traffic flow intensity (rate),
(iii) different values of probability used to compare the interval values from Theorems 1 and 2.

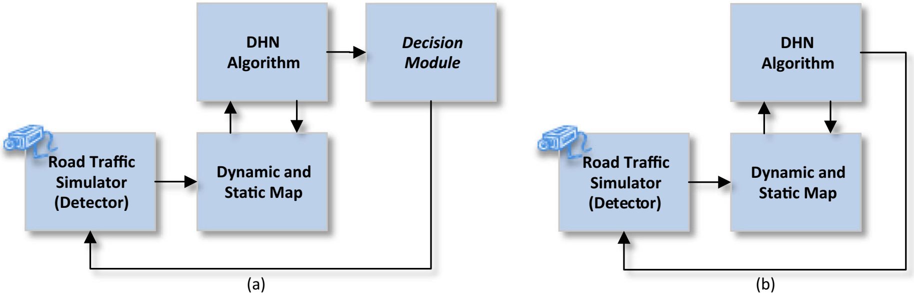


Fig. 6 Two investigated scenarios: **a** with optimisation and **b** without optimisation

For ordering the intervals two methods, presented in the Section 5, were considered. The first method is based on *possibility degree-based ranking* proposed by Jiang et al. [27]. The second one, proposed by Sevastjanov and Róg, is based on the probabilistic approach to interval ordering (see Table 1). After conducting the preliminary tests, the decision was made to apply the method proposed by Sevastjanov and Róg. The probability used during verification of the conditions defined in Theorems 1 and 2 ($P(a > b)$) takes the value: 0.6, 0.65, 0.68, 0.8, 0.85, 0.89, 0,9, 0.95, 0.98 and 1. The number of edges of the road during the test changes between 20 and 4000. The overall number of cars travelling in the streets is proportional to the number of the graph edges and is based on the data registered by detectors in the streets of the city of Kraków. The following parameters are used below to evaluate the results of the experiments:

- $\bar{t}_{opt\_avr}$ - average travel time with optimisation,
- $\bar{t}_{avr}$ - average travel time without optimisation,
- $E_{avr} = \frac{\bar{t}_{avr} - \bar{t}_{opt\_avr}}{\bar{t}_{avr}} \cdot 100\%$ - relative percentage difference between average travel times with and without optimisation,
- Avr - average value of values in column.

6.1 Value of probability in interval comparison

In the tests presented in Tables 2 and 3 two graphs were used. The first test uses the graph with 1000 edges and the second - the graph with 4000 edges with the moderate traffic. The first column of the tables contains the probability p, which is used to compare two intervals (see Section 5.1).

A relative percentage difference between travel time in the case of the graph with 1000 edges is about 0.3% and in the case of the graph with 4000 edges about - 0.25%. The E_{avr} (Fig. 7) has one maximum and one minimum value. The minimum -20% (and -6% for the larger graph) appears if the probability has the value between 0.65 and 0.75. In this case too many cars very frequently changes a travel path due to small change of the travel time. It results in the tendency to obtain a more frequent new path which has negative impact on overall traffic, leading to being blocked by each other. The maximum E_{avr} (greatest shortening traveling time: 17.7% for the smaller graph and 7.5% for the bigger graph) is

Table 2 Comparison of travel times with and without optimisation for graph with 1000 edges

p	$\bar{t}_{opt\_avr}[s]$	$\bar{t}_{avr}[s]$	$E_{avr}[\%]$
0.6	502.1	502.6	0.1
0.65	448.9	455.4	1.4
0.7	489.9	455.4	-7.6
0.75	589.9	490.8	-20.2
0.8	522.1	523.8	0.3
0.85	488.7	593.6	17.7
0.9	545.1	552.6	1.4
0.92	438.7	457.6	4.1
0.95	512.1	530.7	3.5
0.98	578.9	585.7	1.2
1	578.9	585.7	1.2
Avr	517.8	521.3	0.3

Table 3 Comparison of travel times with and without optimisation for graph with 4000 edges (based on Kraków's roads network)

p	$\bar{t}_{opt\_avr}[s]$	$\bar{t}_{avr}[s]$	$E_{avr}[\%]$
0.6	647.8	648.1	0.1
0.65	616.1	619.7	0.6
0.7	640.5	619.7	-3.3
0.75	686.5	641.1	-6.6
0.8	659.1	660.0	0.1
0.85	639.8	687.5	7.5
0.9	670.4	673.7	0.5
0.92	610.8	621.0	1.7
0.95	653.6	663.6	1.5
0.98	683.3	685.3	0.3
1	683.3	685.3	0.3
Avr	653.8	655.0	0.25

obtained when the probability used in the decision algorithm has the value between 0.82 and 0.9 (Fig. 8). When the decision algorithm uses the probability between 0.9 and 1, the algorithm reacts only to bigger changes and the number of cases of shortening travel time is smaller.

6.2 Average travel time dependence on road network size

Nineteen graphs of the size ranging between 10 to 4000 with the moderate traffic were used in the test (Table 4). The values in the table are the average travel time in optimisation or without it, where the probability p equals $p = 0.95$. The first column of the table contains N, which defines the number of edges in the graph.

For graphs with a small number of edges (20-400), the optimisation algorithm works very well. If the travel time changes, the algorithm determines new optimal paths which shorten travel time about 1% to 7%. When the size of the graph increases, the average income decreases from 3% to 1%. When the number of edges is relatively small, the number of the alternated paths is small too. If the algorithm finds an optimal path, the income is relatively high. If the number of routes in the road network increases, the number of the alternated paths increases too (see Fig. 8). The difference between the travel time is relatively smaller

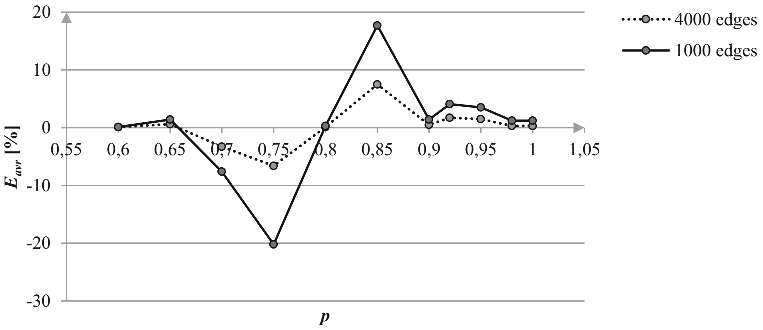


Fig. 7 Relative percentage difference between average travel times with and without optimisation

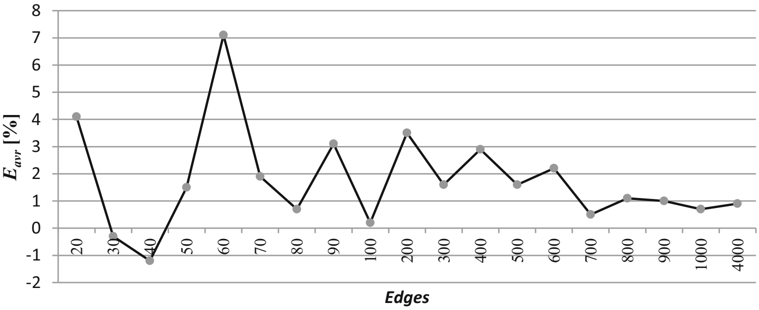


Fig. 8 Relative percentage difference between average travel times with and without optimisation for graphs with 20 to 4000 edges

and the income using the optimal path is smaller too. It should be mentioned that the number of routes (edges) in a average European city is about 1000-8000. So, we should expect the travel time income to be about 1%. However, this is still a significant improvement taking into account a big traffic volume.

6.3 Average travel time dependence on the road traffic volume

In Tables 5 and 6 the average error for different size of the graphs (500 and 4000) and for 10 different traffic volumes (rates) are compared. Traffic volume R determines how many cars

Table 4 Comparison of travel times in the case of optimisation and without optimisation for the graph with 20 edges to 4000 edges

N	$\bar{t}_{opt\_avr}[s]$	$\bar{t}_{avr}[s]$	$E_{avr}[\%]$
20	137.8	143.7	4.1
30	122.5	122.1	-0.3
40	173.5	171.4	-1.2
50	213.2	216.4	1.5
60	213.9	230.2	7.1
70	174.8	178.2	1.9
80	214.1	215.7	0.7
90	250.4	258.3	3.1
100	219.4	219.8	0.2
200	355.2	368.0	3.5
300	373.7	379.8	1.6
400	411.1	423.3	2.9
500	493.6	501.6	1.6
600	465.3	475.7	2.2
700	420.0	422.3	0.5
800	599.7	606.2	1.1
900	659.3	666.0	1.0
1000	514.0	517.7	0.7
4000	652.6	658.7	0.9
Avr	350.8	356.6	1.7

Table 5 Comparison of travel times with and without optimisation for different volume of traffic and graph with 500 edges

R	$\bar{t}_{opt\_avr}[s]$	$\bar{t}_{avr}[s]$	$E_{avr}[\%]$
1	225.9	221.9	-1.8
2	197.8	206.1	4.2
3	201.5	216.5	6.9
4	212.1	212.5	0.2
5	224.5	224.5	0.0
6	243.7	244.2	0.2
7	254.8	257.5	1.1
8	195.9	201.6	2.9
9	173.3	173.3	0.0
10	248.9	250.9	0.8

in each simulation step are generated in each network entry point. In each experiment the value of the probability p was set to 0.85. The first column of the table contains R, which defines the route traffic volume.

Analysis of Fig. 9 leads to the conclusion that if traffic increases, the positive impact of optimisation on the travel time decreases. The greater number of vehicles in the streets leads to a bigger road congestion. If the travel time changes, the optimisation algorithm cannot find much better optimal paths.

6.4 Average travel time with respect to street length

It is interesting to observe how the travel time is affected by the length of streets. For this propose the tests with different average length of streets were carried out (Tables 7 and 8). In each experiment the value of the probability p was set to 0.85. The first column of the tables contains L_{avr}, which defines the average length of streets.

From Fig. 10, it can be seen that if the average length of streets increases (the graph with 4000 edges), the usage of the optimisation algorithm leads to better results. This is because there exist more promising paths between the actual position and the destination of the vehicle. This effect appears because longer streets can buffer more vehicles and in doing so, they prevent the vehicles which would enter them from being blocked. The effect of

Table 6 Comparison of travel times with and without optimisation for different volume of traffic and graph with 4000 edges (based on Kraków’s roads network)

R	$\bar{t}_{opt\_avr}[s]$	$\bar{t}_{avr}[s]$	$E_{avr}[\%]$
1	578.0	582.1	0.71
2	434.1	434.9	0.18
3	519.6	525.4	1.12
4	548.3	572.9	4.29
5	460.8	471.1	2.19
6	587.2	594.4	1.21
7	539.8	543.9	0.75
8	526.3	530.1	0.72
9	522.7	522.7	0.00
10	553.6	553.6	0.00

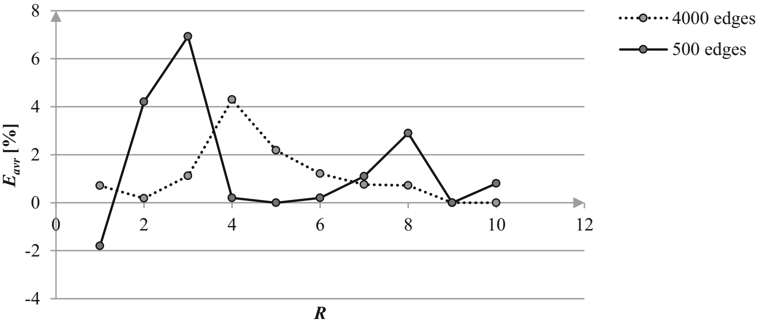


Fig. 9 Relative percentage difference between average travel times with and without optimisation for the different value of the traffic rate

improvements does not appear when the size of the traffic networks is smaller. Otherwise, if the length of the street increases, the traffic decreases and the impact of the optimisation on the travel time decreases too.

6.5 Efficiency of RPS system

This section presents the impact of the traffic parameters, the road network parameters and the *Decision Module* control parameters on the query time. The query time T_{QT} is defined as the difference between the time when the travel time change is detected and the time when the new shortest path is received by the SUMO simulator.

In order to assess the efficiency of the considered scenarios, we compare the average time T_{QT} (in milliseconds) for all simulated cars ($\overline{T}_{QT}$). The experiments were conducted using the road network, where the number of edges varies between 20 and 4000, whereas the number of cars varies between 200 and 40,000. The experiments were performed on a computer with Intel Core i7-870, 2930 MHz, four cores, 8 MB cache processor and 16GB main memory. The program was implemented in C# on .Net platform, Python and Progress Data Base.

An analysis of Table 9 leads to the conclusion that the greater probability causes smaller computation overload. This phenomenon results from less frequent computation of the new

Table 7 Comparison of travel times in the case with and without optimisation for different average length of street graph with 500 edges

L_{avr}	$\bar{t}_{opt\_avr}[s]$	$\bar{t}_{avr}[s]$	$E_{avr}[\%]$
200	152.1	154.7	1.68
300	241.1	243.2	0.86
400	183.2	195.1	6.10
500	299.9	306.2	2.06
700	245.2	245.2	0.00
1000	390	390.6	0.15
1200	368	368	0.00
1500	669.7	670.7	0.15
1700	698.7	698.8	0.01
2000	844.5	845.4	0.11

Table 8 Comparison of travel times with and without optimisation for different average length of street graph with 4000 edges (based on Kraków's roads network)

L_{avr}	$\bar{t}_{opt\_avr}[s]$	$\bar{t}_{avr}[s]$	$E_{avr}[\%]$
200	348.6	347.5	−0.32
300	392.3	391.7	−0.15
400	402.6	404.6	0.49
500	513.2	512.5	−0.14
700	658.9	674.3	2.28
1000	787.8	815.4	3.38
1200	1080.1	1095.6	1.41
1500	1030.5	1029.2	−0.13
1700	1369.9	1394.3	1.75
2000	1352.4	1369.8	1.27

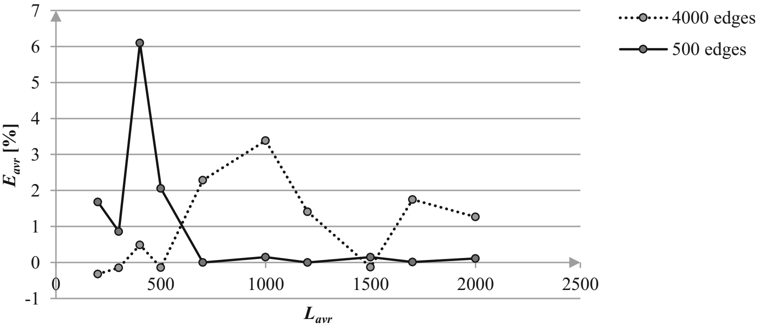


Fig. 10 Relative percentage difference between average travel times with and without optimisation for a different value of the average streets length

Table 9 Average query time $\overline{T}_{QT}$ and standard deviation of average query time $\sigma(\overline{T}_{QT})$ for RPS system with and without optimisation for different network size and different probability value

p	without optimisation				with optimisation			
	$N = 1000$		$N = 4000$		$N = 1000$		$N = 4000$	
	$\overline{T}_{QT}$	$\sigma(\overline{T}_{QT})$	$\overline{T}_{QT}$	$\sigma(\overline{T}_{QT})$	$\overline{T}_{QT}$	$\sigma(\overline{T}_{QT})$	$\overline{T}_{QT}$	$\sigma(\overline{T}_{QT})$
0.60	146.77	13.57	197.52	21.44	148.60	14.21	207.55	24.77
0.65	147.05	14.17	212.36	20.46	144.93	14.32	202.75	21.76
0.70	142.11	13.93	202.25	21.10	151.49	15.53	199.14	21.58
0.75	138.02	13.20	196.86	20.82	152.61	14.62	200.06	22.94
0.80	135.49	13.73	207.68	20.08	138.46	16.57	195.95	21.84
0.85	133.77	13.18	200.69	21.37	150.10	15.33	190.92	24.11
0.90	134.47	13.25	201.50	20.03	142.26	13.92	188.88	21.11
0.92	140.51	14.01	188.66	20.92	144.36	13.90	185.25	21.77
0.95	133.33	12.98	193.52	19.88	134.03	15.63	184.42	21.53
0.98	129.56	13.50	193.79	20.27	145.61	15.20	185.14	21.14
1.00	136.80	13.89	199.01	19.83	142.17	13.28	184.34	21.20

Table 10 Average query time $\overline{T}_{QT}$ in milliseconds and standard deviation of average query time $\sigma(\overline{T}_{QT})$ for RPS system with and without optimisation for different network size and probability value $p = 0.75$

N	without optimisation		with optimisation	
	$\overline{T}_{QT}$	$\sigma(\overline{T}_{QT})$	$\overline{T}_{QT}$	$\sigma(\overline{T}_{QT})$
20	42.03	4.09	44.98	4.06
30	47.06	4.58	49.27	4.12
40	49.85	4.56	49.92	4.25
50	48.80	4.70	51.60	4.61
60	51.45	4.36	51.74	4.37
70	51.60	4.74	53.04	4.64
80	50.71	4.54	54.49	4.83
90	52.91	4.65	53.60	4.85
100	54.86	4.88	54.74	5.07
200	53.80	5.02	58.37	4.91
300	55.41	5.49	57.44	5.29
400	60.78	5.30	61.12	5.50
500	62.93	5.71	64.00	5.94
600	71.80	7.47	76.09	6.50
700	83.14	8.99	90.18	7.28
800	135.88	11.20	139.51	12.20
900	134.21	6.00	140.31	11.38
1000	136.85	14.06	150.11	16.66
4000	208.78	20.83	212.83	23.11

Table 11 Average query time $\overline{T}_{QT}$ and standard deviation of average query time $\sigma(\overline{T}_{QT})$ for RPS system with and without optimisation for different network size, different route traffic volume R and probability value $p = 0.75$

R	without optimisation				with optimisation			
	N=500		N=4000		N=500		N=4000	
	$\overline{T}_{QT}$	$\sigma(\overline{T}_{QT})$	$\overline{T}_{QT}$	$\sigma(\overline{T}_{QT})$	$\overline{T}_{QT}$	$\sigma(\overline{T}_{QT})$	$\overline{T}_{QT}$	$\sigma(\overline{T}_{QT})$
1	39.61	3.76	137.25	11.34	41.43	3.80	137.47	11.31
2	42.47	3.87	139.49	12.01	42.21	3.86	138.89	12.13
3	46.19	4.44	143.73	12.79	51.61	4.39	149.27	13.13
4	48.02	4.56	162.45	13.63	49.51	4.16	168.05	13.57
5	63.13	5.58	203.26	17.67	66.94	5.56	219.05	17.47
6	80.08	7.55	240.81	22.51	86.21	7.36	245.36	22.92
7	90.26	8.71	254.92	24.97	94.83	8.38	262.33	25.79
8	96.81	8.54	272.91	27.23	99.00	8.83	284.48	27.71
9	102.77	9.18	308.34	28.68	107.55	9.37	311.16	28.98
10	109.14	9.51	324.08	30.89	119.77	10.37	328.42	30.77

shortest path. It should be noted that the computation time of the *Decision Module* changes between 1 ms to 16 ms and in a small degree affects the query time in comparison to the scenario without optimisation. Moreover, the computation time of the *Decision Module* does not depend on the size of the road network and if the size of the network increases four times, then the query time increase maximally about 50%.

An analysis of Table 10 shows that as the road network increases 200 times, then the query time increases no more than five times. The standard deviation of the query time is relative small. The overhead of the optimisation is relative small and varies between 5 ms to 10 ms.

Analysis of Table 11 leads to the conclusion that the traffic volume has a significant impact on the query time. If the traffic volume increases ten times then the query time increases about three times. The presented results show that the application of the *Decision Module* in the RPS System has a small impact on the query time.

7 Conclusions

Nowadays there exist a great number of systems which enable the driver to reach his destination, but a few of them use the information about the current situation on a road. This paper studies the application of a new method based on interval numbers for the traffic management in a city. By using this method and the information about the current traffic state, the proposed centralized navigation system helps the driver to navigate more effectively in a volatile environment of the city. The presented system optimises the path and also decides if the current shortest path should be changed to the new one because of traffic fluctuation. The proposed theorems enable effective path updating without recalculation of the best path in each time interval. Moreover, the test shows that using the *Decision Module* does not increase the query time value. To find an optimal path in the static case is relatively easy and a lot of methods for this case were developed on the basis of $A*$ or *Dijkstra* algorithms. Generally, in order to accelerate all these methods the precalculated data is used. But, the static data do not properly describe the properties of the city traffic, where the parameters, such as the travel time are unsteady. That is why the routing methods should take into consideration uncertainty of traffic parameters. There exist several theoretical methods for modelling uncertainty. For example, a stochastic or interval version of the shortest path problem which was studied by Montemanni [37] and Flinsenberg [15].

In this paper a part of the wider research conducted under the INSIGMA project is presented, where the methodology is based on the continuous supply of the information from the streets and the on-line reaction to the changes. As traffic parameters are subject to fluctuations (also due to the accuracy of the measurement), a mechanism for preventing frequent path changing was proposed. On the one hand, it enables the system to implement the assessment of uncertainty. On the other hand, it determines the conditions when the new shortest path to the customer should be sent.

The important results of the performed research are the implementation of the interval algebra in the control of city traffic and the provision of the theoretical basis for determining the conditions that increase the chance of obtaining the optimal path under traffic fluctuation. In order to confirm the presented theoretical results a large number of tests was carried out on the basis of the real traffic network of the city of Kraków. During the tests there were compared different methods for ordering interval numbers, different size of the road-graphs, different values of traffic intensity and different values of probability used to compare the interval values from Theorems 1 and 2. The test results show that implementation of the proposed “intelligent” method under the INSIGMA system can decrease the average travel time by more than 1%. The lesson learned is that information from a great number of sensors installed in the streets enables their users to improve the city traffic and in doing so can contribute to cleaning the air in the city.

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

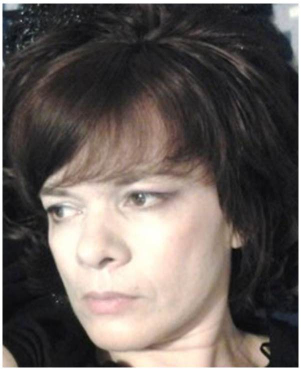
References

1. Angelelli E, Arsik I, Morandi V, Savelsbergh M, Speranza M (2016) Proactive route guidance to avoid congestion. Transp Res B Methodol 94:1–21. https://doi.org/10.1016/J.TRB.2016.08.015
2. Arvanitakis I, Tzes A, Giannousakis K (2017) Mobile robot navigation under pose uncertainty in unknown environments. IFAC-PapersOnLine 50(1):12,710–12,714. https://doi.org/10.1016/J.IFACOL.2017.08.2267
3. Bader R, Dees J, Geisberger R, Sanders P (2011) Alternative route graphs in road networks. In: Marchetti-Spaccamela A, Segal M (eds) Theory and practice of algorithms in (computer) systems, lecture notes in computer science, vol 6595. Springer, Berlin, pp 21–32
4. Bast H, Funke S, Matijevic D, Sanders P, Schultes D (2007) In transit to constant time shortest-path queries in road networks
5. Bast H, Delling D, Goldberg A, Müller-Hannemann M, Pajor T, Sanders P, Wagner D, Werneck RF (2016) Route planning in transportation networks. Springer International Publishing, Cham, pp 19–80. https://doi.org/10.1007/978-3-319-49487-6_2
6. Bellman R (1957) Dynamic programming, 1st edn. Princeton University Press, Princeton
7. Buchhold V, Sanders P, Wagner D (2018) Real-time traffic assignment using fast queries in customizable contraction hierarchies. In: 17th international symposium on experimental algorithms, SEA 2018, June 27-29, 2018, L’Aquila, Italy, pp 27:1–27:15. https://doi.org/10.4230/LIPIcs.SEA.2018.27
8. Chen M, Yu G, Chen P, Wang Y (2017) A copula-based approach for estimating the travel time reliability of urban arterial. Transp Res Part C: Emerg Technol 82:1–23. https://doi.org/10.1016/J.TRC.2017.06.007
9. Chen P, Tong R, Lu G, Wang Y (2018) The α-reliable path problem in stochastic road networks with link correlations: a moment-matching-based path finding algorithm. Expert Syst Appl 110:20–32. https://doi.org/10.1016/J.ESWA.2018.05.022
10. Chmiel W, Szwed P (2015) Learning fuzzy cognitive map for traffic prediction using an evolutionary algorithm. In: Proceedings of the 8th international conference on multimedia communications, services and security, MCSS 2015, November 24, 2015. Springer International Publishing, Kraków, Poland, pp 195–209
11. Chmiel W, Dańda J, Dziech A, Ernst S, Kadłuczka P, Mikrut Z, Pawlik P, Szwed P, Wojnicki I (2016) Insigma: an intelligent transportation system for urban mobility enhancement. Multimed Tools Appl 75(17):10,529–10,560
12. Delling D, Goldberg AV, Pajor T, Werneck RF (2011) Customizable route planning. In: Pardalos PM, Rebennack S (eds) Experimental algorithms. Springer, Berlin, pp 376–387
13. Dib O, Manier MA, Moalic L, Caminada A (2017) Combining VNS with genetic algorithm to solve the one-to-one routing issue in road networks. Comput Oper Res 78:420–430. https://doi.org/10.1016/J.COR.2015.11.010
14. Dinitz Y, Itzhak R (2017) Hybrid Bellman–Ford–Dijkstra algorithm. J Discrete Algo 42:35–44. https://doi.org/10.1016/J.JDA.2017.01.001
15. Flinsenberg I (2004) Route planning algorithms for car navigation. PhD thesis, Technische Universiteit Eindhoven

16. Goldberg AV, Harrelson C (2004) Computing the shortest path: A* search meets graph theory. Tech. Rep. MSR-TR-2004-24 Microsoft Research, Vancouver, Canada
17. Grosso AD, Zucchini A (1993) Applications of interval arithmetic to the analysis and control of structural systems. Smart Mater Struct 2(2):103
18. Gutman R (2004) Reach-based routing: a new approach to shortest path algorithms optimized for road networks. In: Proceedings 6th workshop on algorithm engineering and experiments (ALENEX), SIAM, pp 100–111
19. Han J, Seo Y (2018) Path regeneration decisions in a dynamic environment. Inform Sci 450:39–52. https://doi.org/10.1016/J.INS.2018.03.035
20. He R, Liu H, Kornhauser A, Ran B (2002) Temporal and spatial variability of travel time. Tech rep. Center for Traffic Simulation Studies, UC Irvine
21. Holzer M, Schulz F, Wagner D (2009) Engineering multilevel overlay graphs for shortest-path queries. J Exp Algorithmics 13:5:2.5–5:2.26
22. Hu XB, Wang M, Leeson MS, Di Paolo EA, Liu H (2016) Deterministic agent-based path optimization by mimicking the spreading of ripples. Evol Comput 24(2):319–346. https://doi.org/10.1162/EVCO_a_00156
23. Hu XB, Zhang MK, Zhang Q, Liao JQ (2017) Co-evolutionary path optimization by Ripple-Spreading algorithm. Trans Res Part B: Method 106:411–432. https://doi.org/10.1016/J.TRB.2017.06.007
24. Huang W, Yan C, Wang J, Wang W (2017) A time-delay neural network for solving time-dependent shortest path problem. Neural Netw 90:21–28. https://doi.org/10.1016/j.neunet.2017.03.002
25. Idri A, Oukarfi M, Boulmakoul A, Zeitouni K, Masri A (2017) A new time-dependent shortest path algorithm for multimodal transportation network. Procedia Comput Sci 109:692–697. https://doi.org/10.1016/J.PROCS.2017.05.379
26. Jerrel M (1997) Interval arithmetic for input-output models with inexact data. Comput Econ 10(1):89–100
27. Jiang C, Han X, Liu G, Liu G (2008) A nonlinear interval number programming method for uncertain optimization problems. Eur J Oper Res 188:1–13
28. Kampen EV, Chu P, Mulder B (2010) Optimization of spacecraft rendezvous and docking using interval analysis. Optimization of Spacecraft Rendezvous and Docking using Interval Analysis AIAA Guidance, Navigation, and Control Conference 2-5 August 2010, American Institute of Aeronautics and Astronautics
29. Lauther U (2004) An extremely fast, exact algorithm for finding shortest paths in static networks with geographical background. In: Geoinformation und mobilität – von der forschung zur praktischen anwendung, vol 22, pp 219–230
30. Lee SB, Lee SM, Lee KY (2017), A Gini coefficient based evaluation on the reliability of travel time forecasting. Journal of King Saud University - Engineering Sciences. https://doi.org/10.1016/J.JKSUES.2017.12.001
31. Luby M, Ragde P (1989) A bidirectional shortest-path algorithm with good average-case behavior. Algorithmica 4(1-4):551–567
32. Ma Z, Koutsopoulos HN, Ferreira L, Mesbah M (2017) Estimation of trip travel time distribution using a generalized Markov chain approach. Trans Res Part C: Emerg Technol 74:1–21. https://doi.org/10.1016/j.trc.2016.11.008
33. Marinakis Y, Migdalas A, Sifaleras A (2017) A hybrid particle swarm optimization – variable neighborhood search algorithm for constrained shortest path problems. Euro J Oper Res 261(3):819–834. https://doi.org/10.1016/J.EJOR.2017.03.031
34. Maue J, Sanders P, Matijevic D (2010) Goal–directed shortest-path queries using precomputed cluster distances. J Exp Algorithmics 14:2:3.2–2:3.27
35. Miao C, Liu H, Zhu GG, Chen H (2018) Connectivity-based optimization of vehicle route and speed for improved fuel economy. Trans Res Part C: Emerg Technol 91:353–368
36. Montemanni R, Gambardella L (2008) Robust shortest path problems with uncertain costs. Tech. rep., IDSIA-03-08
37. Montemanni R, Gambardella L, Donati A (2004) A branch and bound algorithm for the robust shortest path problem with interval data. Oper Res Lett 32(3):225–232
38. Moore R (1966) Interval analysis. Prentice-Hall series in automatic computation. Prentice-Hall, Upper Saddle River
39. Naghawi H, Wolshon B (2010) Transit-based emergency evacuation simulation modeling. J Trans Safety Secur 2(2):184–201
40. Patle B, Pandey A, Jagadeesh A, Parhi D (2018) Path planning in uncertain environment by using firefly algorithm. Defence Technology. https://doi.org/10.1016/J.DT.2018.06.004

41. Pshikhopov V, Guzik V, Pyavchenko A, Pereverzev V, Saprykin R, Pshikhopov V (2017) Neural networking path planning based on neural-like structures. Path Planning for Vehicles Operating in Uncertain 2D Environments, pp 25–96
42. Rahmani M, Koutsopoulos HN, Jenelius E (2017) Travel time estimation from sparse floating car data with consistent path inference: a fixed point approach. Trans Res Part C: Emerg Technol 85:628–643. https://doi.org/10.1016/J.TRC.2017.10.012
43. Sanders PDS (2006) Engineering highway hierarchies. In: Proceedings of the 14th conference on annual European symposium, ESA'06, vol 14. Springer-Verlag, London, UK, pp 804–816
44. Sanders P, Schultes D (2005) Highway hierarchies hasten exact shortest path queries. In: Proceedings of the 13th annual European conference on algorithms, ESA'05. Springer-Verlag, Berlin, Heidelberg, pp 568–579
45. Sarbazi-Azad H, Chapuis G, Djidjev H, Lavenier D, Andonov R (2017) GPU-accelerated shortest paths computations for planar graphs. Advances in GPU Research and Practice, pp 273–306. https://doi.org/10.1016/B978-0-12-803738-6.00011-2
46. Schultes D (2008) Route planning in road networks. PhD thesis, Universität Fridericiana zu Karlsruhe
47. Schultes D, Sanders P (2007) Dynamic highway-node routing. In: Demetrescu C (ed) Experimental algorithms, lecture notes in computer science, vol 4525. Springer, Berlin, Heidelberg, pp 66–79
48. Schulz F, Wagner D, Weihe K (1999) Dijkstra's algorithm on-line: an empirical case study from public railroad transport. In: Proceedings of the 3rd international workshop on algorithm engineering, WAE '99. Springer-Verlag, London, UK, pp 110–123
49. Sevastjanov P, Róg P (2003) A probabilistic approach to fuzzy and crisp interval ordering. Task Quarterly 7(1):147–156
50. Sevastjanov P, Róg P, Karczewski K (2002) A probabilistic method for ordering group of intervals. Comput Sci 2(2):45–53
51. Sever D, Zhao L, Dellaert N, Demir E, Van Woensel T, De Kok T (2018) The dynamic shortest path problem with time-dependent stochastic disruptions. Transp Res Part C: Emerg Technol 92:42–57. https://doi.org/10.1016/J.TRC.2018.04.018
52. Stadtherr M, Xu G, Burgos-Solorzano G, Haynes W (2007) Reliable computation of phase stability and equilibrium using interval methods. Int J Reliab Saf 1(4):465–488
53. Szwed P, Kadluczka P, Chmiel W, Glowacz A, Sliwa J (2012) Ontology based integration and decision support in the insigma route planning subsystem. In: Proceedings of federated conference on computer science and information systems - FedCSIS, Wroclaw, Poland 2012, pp 141–148
54. Wagner D, Willhalm T (2003) Geometric speed-up techniques for finding shortest paths in large sparse graphs. In: Battista GD, Zwick U (eds) Proceedings of the 13th annual European conference on algorithms, ESA'03, LNCS, pp 776–787
55. Yang CH, Tsai MH, Kang SC, Hung CY (2018) UAV path planning method for digital terrain model reconstruction – A debris fan example. Autom Construc 93:214–230. https://doi.org/10.1016/J.AUTCON.2018.05.024
56. Zadeh L (1965) Fuzzy sets. Inf Control 8(3):338–353
57. Zeng X, Wang J (2018) Globally energy-optimal speed planning for road vehicles on a given route. Transp Res Part C: Emerg Technol 93:148–160
58. Zhang Q, Xiong S (2018) Routing optimization of emergency grain distribution vehicles using the immune ant colony optimization algorithm. Applied Soft Computing
59. Zhao Y, Zheng Z, Liu Y (2018) Survey on computational-intelligence-based UAV path planning. Knowl-Based Syst 158:54–64. https://doi.org/10.1016/J.KNOSYS.2018.05.033
60. Zhu SQ (2004) Stochastic time-dependent route planning in car navigation systems. Master's thesis, Technische Universiteit Eindhoven

Wojciech Chmiel received his Ph.D. in Automatics from the AGH University of Science and Technology in Krakow. He is associated with the Department of Automatics and Biomedical Engineering in the Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering. He research focuses on the area of operation research, approximation algorithms, machine learning, multimedia analysis and processing, modelling discrete optimization problems, fuzzy optimization and decision making. He has taken part in several international research projects in optimization and artificial intelligence area.



Iwona Skalna obtained Master degree in Mathematics, Master degree in Computer Science, and Ph.D. degree from Faculty of Civil Engineering at the Silesian University of Technology in Gliwice, Poland. She is holding a post-doctoral position in the Faculty of Management at the AGH University of Science and Technology in Cracow. She is a Lecturer in the Department of Applied Computer Science. Her research focuses on modeling uncertainties with intervals and fuzzy numbers, and solving parametric linear systems with uncertain parameters.

Stanisław Jędrusik, Ph.D is currently a Senior Lecturer in the Department of Applied Computer Science in the Faculty of Management at AGH University of Science and Technology in Cracow. In 1991 he received master degree in electronics and automatics and then in 2002 he obtained PhD in economy from AGH University of Science and Technology. His research interest lies in the area of artificial intelligence and include rule based systems, machine learning, knowledge representation and business process modeling.