PERFORMANCE ANALYSIS OF GENETIC ALGORITHMS FOR ROUTE COMPUTATION APPLIED TO EMERGENCY VEHICLES IN UNCERTAIN TRAFFIC

Vlad CONSTANTINESCU[1,2], Monica PĂTRAȘCU[3]

Efficient transportation is an important requirement in today's world. As modern cities grow in size and complexity, the travel distances for people and goods increase while available time decreases. Routes must be computed dynamically and close to real-time, while taking into consideration various factors such as road congestion and maximum speed allowed. In this paper, we explore the use of evolutionary algorithms to solve this multiple criteria optimization problem. The performance analysis included in this study identifies the best configuration for the genetic routing algorithm which provides the best consistency of correct routes.

Keywords: intelligent transportation systems (ITS), genetic algorithms (GA), path planning, vehicle routing

1. Introduction

With the continuous increase in both size and complexity of modern cities, Intelligent Transportation Systems (ITS) play a more important role. ITS can be defined as the application of sensing, computing, electronics and communication technologies, together with management strategies, for improving the safety and efficiency of the transportation system [1]. The presence of ITS is an integral part of smart cities [2]. Traffic jams are an increasing problem in the modern cities. Infrastructure development is not always an option as real estate and financial limitations are often encountered. The development of ITS is important for decreasing travel times and pollution. Intelligent Transportation Systems provide a number of user services such as pre-trip travel information, incident management, traffic control or route guidance [3]. A study presented in [4] shows the use of route guidance systems for both local drivers and visiting drivers. With the increase in intelligence at vehicle level, more opportunities for increasing safety during transportation arise. These measures range from adaptive cruise control systems to collision avoidance and even driver monitoring. While applied to individual vehicles, they contribute to the overall safety and efficiency of travel through urban areas. To this respect, the European Union lists route guidance and

[1] PhD Student, Department of Automatic Control and Systems Engineering, University POLITEHNICA of Bucharest, Romania, e-mail: vlad.constantinescu@acse.pub.ro
[2] Institute of Space Science, Măgurele, Romania
[3] Associate Professor, Department of Automatic Control and Systems Engineering, University POLITEHNICA of Bucharest, Romania, e-mail: monica.patrascu@upb.ro

navigation as one of the core systems included in Vehicle Safety Systems as part of ITS [5].

According to the statistics of the Emergency Situations Department of the Romanian Ministry of Internal Affairs, from January to November 2018, emergency response crews answered to 1197 requests per day, in average [6]. The mean response time was between 13 minutes for fires and 12 minutes for medical emergencies (SMURD). More so, when the vehicles benefitting from optimum routes are emergency intervention crews, an efficient routing system through busy urban area is paramount to be integrated in any useful ITS. In 2017, for instance, the Brisbane City Council implemented a system for automatically turning traffic lights green when an ambulance or fire truck approaches the intersection [7]. This was done using Bluetooth and reduced the travel time of the emergency crews by 26%. A study made in Spain in 2010 [8] suggests than a 10-minute reduction in the response time of the medical crew can lead to a 30% decrease in the number of fatalities. From smartphones to field-specific tablets and devices designed to help professionals in their work, the development of mobile technologies has opened the gates to new uses when it comes to transportation. Combined with the rapid expansion of communication technologies which allows users to connect with each other and with various databases, drivers can nowadays access more and more accurate traffic information, while they travel. Emergency vehicles can particularly profit from real-time updates which support dynamic routing applications. GAs have been applied in control systems engineering, as described in [9]. The authors underline the benefits of evolutionary algorithms, benefits such as flexibility in representing the decision variables and robustness to difficult search environments. They are tolerant to discontinuities and noise. Evolutionary algorithms can also be coupled with neural and fuzzy control schemes and allow multi-objective optimization.

In this paper we evaluate the use of genetic routing algorithms for the path planning of emergency vehicles through urban traffic. In section 2 we present the genetic route computation method and the problems associated with computing a route. In section 3 we perform a comparative analysis of the algorithm for different parameters and determine the configurations which provide the best routing consistency. Finally, the last section contains the conclusions and future directions.

2. Genetic Algorithms for Route Computation

Urban traffic keeps increasing in density along with the development of urban conglomerates. Smart routing based on real-time data is preferred to offline modelling or prediction of traffic densities and flow. Computing the best route for a vehicle should take into consideration multiple criteria, such as route length, road occupancy, maximum speed allowed.

The route computation of emergency vehicles is a problem that is in part path planning and in part vehicle routing. For this study, we consider that an *emergency vehicle* is a vehicle taking part in urban traffic, actuated by a human driver, with law mandated maximum speed. The problem at hand is an optimization problem which must return a good and efficient route from a *specified starting point* to a *specified destination* for which time constraints are critical: the vehicle must reach the destination as soon as possible, i.e. with minimum waiting time, which translates into three requirements to be met concurrently:

$$\begin{aligned} I_P &= \min_{x}\{\sum_{i=1}^{n} x_i\} \\ I_C &= \min_{z(t)}\{\sum_{i=1}^{n} z_i(t)\} \\ I_N &= \min_{n}\{n\} \end{aligned} \tag{1}$$

where x is the length of a road segment delimited by two intersections, n is the total number of road segments, and $z(t)$ is the number of vehicles at any given time on a road segment. The criteria I_P, I_C and I_N refer to the minimization of route length, of road occupancy along the route, and number of route segments, respectively. Clearly, a good route must be comprised of few segments, of small total length, *and* with the least number of cars possible. This problem can be approached as a multi-objective optimization problem and because formal models of traffic are highly uncertain and usually non-linear, because the search space is not necessarily continuous, and because there are various constraints in play, a suitable optimization method is given by genetic algorithms (GAs), which have been successfully applied to this type of problem.

Genetic Algorithms (GA) [10] are metaheuristic search algorithms for optimization in large multi-dimensional non-smooth spaces. This class of algorithms was inspired by biological evolution mechanisms (Fig. 1). Each individual of an artificially generated population represents a possible solution to the optimization problem at hand. The members of the population are called chromosomes and each element of a chromosome represents a gene. A fitness function evaluates the performance of the individual and a fitness value is computed for each of them. For maximization problems, the greater the fitness value, the better the solution, and vice-versa. For minimization, the lower the fitness value, the better suited the individual. The design of the fitness function influences the search toward the global optimum. If some criteria or constraints are prioritized, the algorithm might first seek to meet the major criteria, thus contracting the search space and subsequently refining the search. If, however, all requirements are equally weighted, the search time might increase.

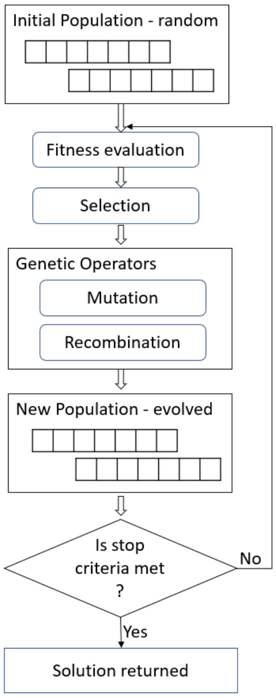

Fig.1. Genetic Algorithm

After the random set-up of the initial population, the algorithm uses mechanisms of selection, recombination and mutation (Fig. 1) for evolving the population. These operations are repeated until the stop condition—such as number of iterations or various quality criteria—is met.

Selection mechanisms are used to choose the individuals on which mutation or recombination is applied so that new individuals are created. Some widely used selection methods include roulette wheel and tournament.

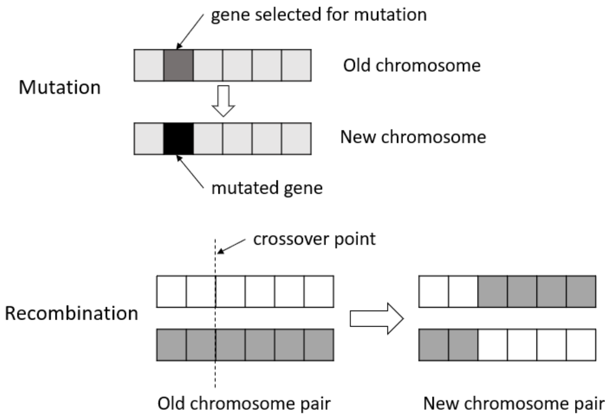

Fig.2. Mutation and single-point recombination

The mutation operator modifies one gene of a chromosome and so creates a new individual (Fig.2). The alteration of the gene is usually made according to a parameter known as mutation probability. Mutation is important to maintain genetic diversity in the population. This probability-based operator usually ensures that the algorithm does not end in local optima. Previous studies [11,12] have shown that GA's are highly sensitive to the mutation rate, which is the probability of an individual to suffer mutation. In its standard version, the mutation rate is defined as percentage of mutated individuals. When this rate is too small, the genetic diversity in the population cannot be maintained, but when too high, it slows down and even halts convergence. The recombination (or crossover) operator creates new chromosomes (children) by combining the features of other individuals(parents). The most basic recombination is called one-point-crossover and creates two children from two parents: a crossover point is selected and new individuals are created by taking the first part of one parent and the second from the other; the second new individual is created with the second part of the first parent and the first from the other (Fig.1).

The fitness function of the GA allows the integration of multiple criteria and so the obtained solution satisfies different objectives. The role of the fitness function is to compute a fitness value for each individual. Based on this value, a number of individuals from the population are selected and will be used for recombination. Depending on the type of selection, these individuals could be the best (elitist selection), or a combination of best or average (ranking selection), or even best and worst (separating selection). Termination criteria can be either one of several variations—a certain number of generations has been processed, a preset amount of time has passed, or the fitness value is within a desired range—or a combination of these.

Transferring the routing problem to genetic optimization raises three major questions:

- route encoding: how to best transpose the routing information into a chromosome when route lengths might vary and how does this representation affect the evolution operators;
- constantly changing search space due to vehicle moving throughout the road network and due to multiple constraints;
- returning a viable solution in a feasible amount of time: for evolutionary algorithms convergence is highly dependent on parameters that drive it toward or away from a global optimum, on mechanisms that tend to cause stagnation in local minima, etc., therefore properly configuring the algorithm is necessary to ensure success in the least possible amount of time.

The principle of genetic routing is illustrated in Fig. 3. The routing algorithm receives a destination point within the urban area and it knows the current position of the emergency vehicle. Based on current traffic data (received from external sources), the algorithm searches for a viable route which is then transmitted to the driver of the emergency vehicle. For a route to be viable, it needs to comply with the following restrictions: a route cannot contain duplicate segments, i.e. the emergency vehicle should not double back or be expected to travel the same segment twice; a route must contain at least one segment, routes cannot be null; a route must start and end at the proper positions on the map; a route must be composed of continuous route segments.

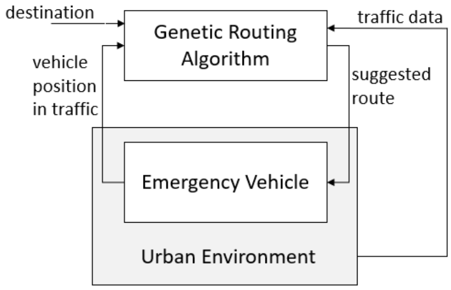

Fig.3. Genetic routing of emergency vehicles

The fitness function for the routing algorithm takes into account the route restrictions and determines which would be the route that offers the least amount of travel time based on length and occupancy [13]. For a route of *n* segments, the route fitness *F* is computed by analyzing each section in terms of degree of occupancy on the *i*-th segment D_i, length of route segments L_i, and a scaling function $\rho(P_0, i)$ which minimizes the importance given to segments farthest away from the current position of the vehicle P_0 (because their conditions might change by the time the driver reaches them):

$$F = \sum_{i=1}^{n} \frac{D_i}{\rho(P_0, i) L_i} \tag{2}$$

The algorithm rewards short routes with a small number of participants, focusing on routes with sparse traffic condition which could take less time to travel even though geographically they are longer. In a crowded urban area, sometimes the shortest distance in space does not ensure the smallest travel time due to congestions or slow-moving traffic. The start and end points of a route have been a main concern when we first explored the concept of genetic optimization applied to the vehicle routing problem [13]. We explored two options: a) compute the route once, at departure; or b) re-compute the route dynamically during travel. In [13] we determined that due to the ever-changing traffic conditions, recalculation of the route whenever possible is a better approach than computing the entire route from the start without adjustments. This choice, although it increases the computational resources the algorithm uses during travel, it also alleviates concerns regarding surety of convergence. GAs do not guarantee an optimal solution, but even though the current suggested route is correct, this assessment might change due to the other participants or other events that might cause a route to become unviable (for instance an accident). The same is true for the reverse situation, in which a route is not optimal: we only need the first few segments to be suitable because the route is recalculated as the vehicle advances through the city and traffic changes.

Another issue of genetic routing is the search space. In a large urban area with an interconnected network of streets, the fitness calculation in Eq. 2 is not enough to ensure that the emergency vehicle is driving toward the destination and not away or to ensure that all restrictions on route composition are met. We addressed these problems in [14] by designing a route encoding method which reduces the search space according to restrictions. Thus, we took an unnecessary computational strain off the fitness evaluation procedure and used it to gain an advantage in the search itself. This method is presented in Fig. 4. Given a simple map with 4 intersections (A,B,C,D), Fig. 4 illustrates the encoding of a route starting at A and ending at C: A→B→C, as shown with arrows in the figure.

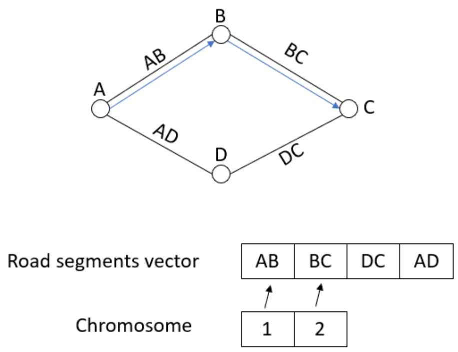

Fig.4 – Route encoding example

All the road segments presented in the map are stored in a vector. When encoding a route in a chromosome, we store the indexes of the corresponding road segments forming the route. For example, the chromosome associated to the route formed by the road segments AB and BCis the vector c = [1 2]. Using this method, we reduced the search space significantly by eliminating all the routes which did not comply with the restrictions regarding duplicates, non-null routes, proper start and destination, continuity, and occupancy. However, this is not enough to ensure correct routes are returned in a feasible amount of time, due to the very nature of genetic optimization. These algorithms might perform well in uncertain, non-smooth spaces, but they are also sensitive to the evolution mechanisms and probabilistic operators that drive the search. Therefore, in order to make sure that a valid and optimum route is found, we need to tune and test the implementation of the algorithm repeatedly until convergence times are satisfying and consistent.

3. Performance Analysis and GA Tuning for Route Computation

In [14] we have introduced a method of encoding routes into chromosomes and then applied a GA to compute the best route. The aim of the algorithm is to find the best route according to the specified criteria, such as least occupied and shortest distance travelled. A reduced search space ensures a more rapid convergence of the algorithm, so when encoding the routes into chromosomes we attempted to minimize the search space as much as possible. We used chromosomes with lengths similar to the possible routes lengths and each gene points to a road section that is part of the route. When computing the fitness function, only occupied genes (i.e. the segments part of the route, for instance segments AB and BC in Fig. 4) are taken into consideration. The value of the fitness function is increased if the encoded segments are continuous and if the route starts and ends at the start and destination point. If loops are found in the route or if the route contains no segments the value is decreased. The fitness function value is also adjusted according to the length of the route and to the number of cars present on the selected road segments. This ensures shorter and less occupied routes are found. The testing performed by the authors of [15] shows that tournament selection is the recommended selection mechanism for the basic traveling salesman problem. The tournament selection mechanism, with a tournament size of 2, is detailed in Fig. 5.

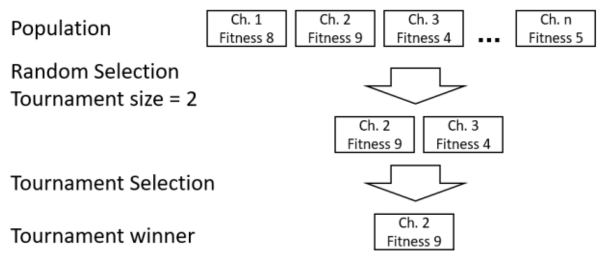

Fig.5 – Tournament selection

Tournament selection is a mechanism for selecting individuals from the population to create new members of the next generation using crossover and mutation. It has the following parameters: size of the tournament which represents the number of individuals selected from the population for the tournament, and probability which represents the probability with which the best individual is selected. The second-best individual is selected with a probability of $p\cdot(1-p)$ and so on. The Individuals are ranked by their fitness value. The best individual is the one with the highest fitness. For our tests, we used the SUMO (Simulation of

Urban Mobility) [16] software. The control of the traffic lights was implemented using agents [17]. For developing the agents, we used the JADE (JAVA Agent Development Framework) [18] environment. Our simulations ran on a system with 16GB RAM and an Intel Core I7 processor clocked at 2.9 GHz.

In order to find the shortest convergence time, we ran the algorithm with different values for population size and genetic operators, as shown in Table 1. The last column in the table lists the percentage of correct routes (continuous and least occupied) out of 10 runs. An epoch represents a generation (one cycle of the GA comprised of evaluation, selection, recombination).As we can see in Table 1, the tournament size affects the outcomes significantly as a high value for this parameter means the selection pressure is high [19] so individuals with an increased fitness value are selected for mutation and crossover.

A higher crossover probability increases the diversity in the population, but it can also decrease divergence by losing good individuals already found, as described in [20]. We also notice that an increased mutation probability tends to impede the gradual convergence to a solution and transform the progress of the algorithm to a random search in the solution space.

Table 1

Genetic algorithm analysis

No. of Epochs	Population size	Tournament size	Tournament probability	Crossover probability	Mutation probability	Run time [s]	% of valid routes
10000	500	20	80%	35%	8,3%	16	30%
		10				12	30%
		6				10	20%
		2				5	90%
					20%	6.5	40%
					5%	7	90%
					2%	9	10%
				10%	8.3%	4	100%
				20%		5	100%
				50%		6	100%

After running the tuning tests, we adjusted the algorithm with the best parameters and ran new tests for different population sizes to evaluate its consistency. The results are presented in Fig.6. We performed 100 runs for different numbers of epochs (between 5000 and 15000) with a population size of 500, a tournament of size 2 and probability 80%. The genetic operators are one-point crossover with 35% probability and mutation with 1/12~8.3% probability.

Although a run with 5000 epochs offers over 85% correct routes, we need over 9000 epochs in order to ensure our algorithm provides over 95% correct routes, as presented in Fig. 6. A run with over 12000 epochs provides close to 100% correct routes. For this implementation, we recommend an epoch number of at least 12000 in order to minimize the running time and to maximize the probability of finding correct routes.

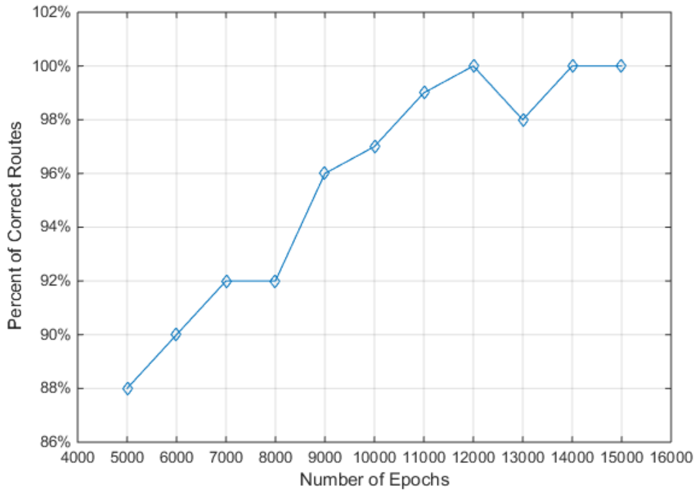

Fig.6 – Percentage of correct routes for 100 runs of the GA vs number of epochs

A computation with 5000 epochs takes 2-3 [sec] on our test system. A run with 15000 epochs takes 7-8 [sec] on the same system (average times computed across all simulations). With further code optimization, these computing times can be greatly reduced.

4. Discussion: Advantages and Limitations

If we compare the presented routing algorithm with the classical shortest path algorithms, we notice the advantage in computation speed of the classical algorithms, but these allow the use of a single criterion for selecting the route.

For testing purposes, we build an implementation of the Dijkstra algorithm in JAVA. The algorithm found the shortest route in less than 1 [sec]. However, a problem arises when trying to incorporate multiple criteria in the search. For Dijkstra, the multiple criteria must be combined into a single scalar weight value for each route segment. Such a value is difficult to compute adaptively, as it needs to contain pertaining to the position of the segment represented by the arc relative to the current position of the vehicle. When the vehicle advances through traffic, its position changes and thus the entire graph would need to be updated. When using Dijkstra, the scalar weights for the entire road network need to be computed prior to running the algorithm. One way to incorporate multiple criteria in the scalar weight is to assign different weights to different separate criteria and to combine them into a final scalar weight. These secondary weights are different for each road segment; for example, if taking into consideration the vicinity to a school and thus having reduced traffic requirement. Even so, there is no guarantee that a weighted sum is the best form of criteria combination. The computation cost of the arc weights would offset the small computation time of the optimization procedure, bringing it closer to the GA run times. Moreover, the GA implementation we used in this study is not optimized for real-time execution, in which case the run times can be significantly reduced.

On the other hand, when using the GA variant, multiple criteria are easily added to the fitness function without additional design effort. For instance, the uncertainty associated to types of roads (for example roads that pass near schools) does not require explicit formalization using only one scalar that might or might not offer an accurate representation of the uncertainty. With these remarks in mind, we conclude that the standard GA might not be the best, and thus we propose a new approach that uses the advantages of both methods, classical and evolutionary.

5. Conclusions

In this paper we explore how different parameters influence the performance and convergence of a GA based vehicle routing algorithm. We aim to tune the algorithm in order to minimize computation time and improve convergence, so we can find the shortest and least occupied route while maintaining a short run time. By running test on our implementation, we concluded that, for a specific set of GA parameters, our algorithm converges to a correct route if the number of epochs is large enough (12000 or more).

Our previous work has provided us with a proper encoding method, and we reduced the search space considerably. Therefore, further work will focus on improving the generic evolution mechanisms implemented here aiming to obtain a fully customized GA, specifically designed to solve the routing problem. By taking into account the encoding method, new operators might allow the reduction of the population size without losing consistency. Moreover, the search space and computation time can be further improved by introducing specialized mechanisms such as immunization, a powerful yet sensitive convergence driving tool. Given that the start and destination points are known, the GA can also start from a non-random initial population and so achieve better convergence times. The runtimes we obtained for generating consistent correct routes include both the search part of the algorithm and the computationally heavy fitness evaluation. With proper code optimization on a dedicated tablet or mobile device, this runtime can be further reduced, thus making it viable for real world implementation and usage.

R E F E R E N C E S

[1] *R. Nasim, A. Kassler*, "Distributed architectures for intelligent transport systems: A survey", in Second Symposium on Network Cloud Computing and Applications (NCCA), 2012.

[2] *Z. Xiong, H. Sheng, W. Rong, D. E. Cooper*, "Intelligent transportation systems for smart cities: a progress review.," Science China Information Sciences, **vol. 55**, no. 12, pp. 2908-2914,2012.
[3] *A. García-Ortiz, S. Amin, J. Wootton*, "Intelligent transportation systems—Enabling technologies," Mathematical and Computer Modelling, **vol. 22**, no. 4-7, pp. 11-81, 1995.
[4] *R. N. Fleischman, L. A. Thelen, D. Dennard*, "A preliminary account of TravTek route guidance use by rental and local drivers.," in Proceedings of the IEEE Vehicle Navigation and Information Systems Conference, 1993.
[5] European Union, "MOBILITY AND TRANSPORT," [Online]. Available: https://ec.europa.eu/transport/themes/its/road/application_areas/vehicle_safety_systems_en [Accessed 22.11.2018].
[6] *** Analiza statistica privind acţiunile de prevenire, pregătire şi răspuns în situaţii de urgenţă pentru perioada (Statistical analysis regarding prevention, preparation and response to emergency situations for the timeframe) 01.01.2018 – 30.11.2018 [Online] https://www.igsu.ro/documente/informare_publica/evaluari/2018/Analiza%20operativa%2001.01.2018%20-%2030.11.2018.pdf [Accessed 11.01.2019]
[7] *** Bluetooth technology turns red lights green for emergency vehicles [Online] https://www.brisbanetimes.com.au/national/queensland/bluetooth-technology-turns-red-lights-green-for-emergency-vehicles-20170221-guhs0y.html [Accessed 11.01.2019]
[8] *R. Sánchez-Mangas, A. Carcia-Ferrrer, A. de Juan, A.M. Arroyo,* "The probability of death in road traffic accidents. How important is a quick medical response?". Accident Analysis & Prevention, 2010, **vol. 42**, no.4, pp.1048-1056.
[9] *P. J. Fleming, R. C. Purshouse*, "Evolutionary algorithms in control systems engineering: a survey.," Control engineering practice, **vol. 10**, no. 11, pp. 1223-1241, 2002.
[10] *J. H. Holland,* Adaptation in Natural and Artificial Systems, Michigan: The University of Michigan Press, 1975.
[11] *M. Srinivas, L.M. Patnaik*, "Adaptive probabilities of crossover and mutation in genetic algorithms". IEEE Transactions on Systems, Man, and Cybernetics, 1994, **vol. 24**, no.4, pp. 656-667.
[12] *J. Smith, T.C. Fogarty,* "Self-adaptation of mutation rates in a steady state genetic algorithm." Evolutionary Computation, 1996., Proceedings of IEEE International Conference on, pp. 318-323. IEEE.
[13] *M. Pătrașcu, V. Constantinescu, A. Ion* "Controlling Emergency Vehicles in Urban Traffic with Genetic Algorithms." In Proceedings of The 9th EUROSIM Congress on Modelling and Simulation, EUROSIM 2016, The 57th SIMS Conference on Simulation and Modelling SIMS 2016, no. 142, pp. 243-250. Linköping University Electronic Press, 2018.
[14] *V. Constantinescu, M. Pătrașcu*, "Route encoding in evolutionary control systems for emergency vehicles". ITS Telecommunications (ITST), 2017 15th International Conference on (pp. 1-5). IEEE.
[15] *N. M.Razali, J. Geraghty* "Genetic algorithm performance with different selection strategies in solving TSP". In Proceedings of the world congress on engineering (**Vol. 2**, pp. 1134-1139). Hong Kong: International Association of Engineers, July 2011
[16] *D. Krajzewicz, J. Erdmann, M. Behrisch, L. Bieker* "Recent Development and Applications of SUMO - Simulation of Urban Mobility," International Journal on Advances in Systems and Measurements, **vol. 5**, no. 3&4, pp. 128-138, 2012.
[17] *A. Ion, C. Berceanu, M. Pătrașcu*, "Applying agent-based simulation to the design of traffic control systems with respect to real-world urban complexity," in Multi-Agent Systems and Agreement Technologies, Cham, Springer, 2015, pp. 395-409.
[18] *** "JAVA Agent Development Framework," [Online]. Available: http://jade.tilab.com/. [Accessed 25.012019].
[19] *B. L. Miller and D. E. Goldberg*, "Genetic algorithms, tournament selection, and the effects of noise," Complex systems, **vol. 3**, no. 9, pp. 193-212, 1995
[20] *M. Srinivas, L.M. Patnaik*, "Genetic algorithms: A survey", Computer, 27**vol. 6**, pp.17-26, 1994