

Controlling Emergency Vehicles in Urban Traffic with Genetic Algorithms

Monica Patrascu¹ Vlad Constantinescu^{1,2} Andreea Ion¹

¹Department of Automatic Control and Systems Engineering, University Politehnica of Bucharest, Romania

²Institute of Space Science, Bucharest, Romania

{monica.patrascu, vlad.constantinescu, andreea.ion}@acse.pub.ro

Abstract

Emergency officers could often benefit from a route planning system that is based on constant traffic monitoring and complex decision making, seeking to give victims another breath of hope by assisting emergency units with reaching them on time. The main challenge is providing responses in a continuously evolving environment within a prescribed time frame, while using limited resources and information that is often incomplete or uncertain. This paper presents a route control concept for emergency vehicles through urban traffic. The proposed genetic controller is designed to dynamically reassess the route while the vehicle passes through the road network, continuously generating new routes based on current traffic. The algorithm is tested in an agent based simulation model that includes both traffic participants and a distributed traffic control system.

Keywords: genetic algorithm, emergency response, control systems, distributed control, agent based simulation model

1 Introduction

In complex and distributed urban environments, the services that provide quality of life and safety have to deal with unpredictable events and incomplete data. Moreover, intelligent transport systems are becoming increasingly important as they aim to provide solutions to crucial issues related to transportation networks, such as congestion and various incidents. One of the most important activities in the protection of human life is the intervention of emergency responders, for which an important issue in the unpredictable urban road networks is the time required for an emergency vehicle to reach an event scene. Congestion and the various obstacles that may appear during the journey on the chosen path can increase travel time and therefore reduce the chances of ensuring the safety of human life (Blackwell et al, 2002; Pons et al 2005; Sladjana et al 2011; Rushworth et al 2014).

Thus, re-calculating the routes of emergency vehicle during their journey based on environmental changes is a way to avoid these obstacles.

Real-time decision problems are also playing an increasingly important role in transportation management, as advances in communication and

information technologies allow real-time information to be quickly obtained and processed. Therefore, dynamic vehicle route generation has become more and more efficient, especially in urban areas.

The problem of finding the most efficient routes for the quick access of the emergency vehicles in the current urban traffic is very important in terms of protecting and saving human lives. From an economic and social point of view, implementing the developed algorithms would increase the number of saved lives, reduce congestion and accident risk, would reduce fuel consumption and the time spent in traffic and by doing so, would also reduce the number of people affected by stress on the road.

The most important technological benefit regarding evolutionary computing is the possibility to integrate techniques typically associated with modeling complex systems in representing the possible solutions to optimization problems solved with the help of evolutionary algorithms. This opens the way to using these class of algorithms for solving problems that cannot be modeled using formal techniques and that can only be solved by using heuristic methods. The future applications of evolutionary computing are not restricted to vehicle routing; they include different other optimization problems, from designing control systems for processes affected by non-linearity and uncertainties, modeling complex and biological processes, algorithms for the optimization of sensor spreading over an area, to designing and tuning the command rules for distributed control systems applied to large-scale processes.

The initial route generation problem has been initially regarded as a variant of the travelling salesman problem (Dantzing et al 1959). Beside this classical formulation of the routing problem, a series of other approaches have been studied (Toth et al 2002). For route reconfiguration, the initial studies (Seguin et al 1997) have first taken into account the static routing problem (Psaraftis 1980; Madsen et al 1995), followed by more in depth analyses of the differences between dynamic and static routing (Psaraftis, 1988; Goel et al 2006).

Another perspective for solving the problem of dynamic vehicle routing takes into account evolutionary computing algorithms, either by using algorithms inspired from biology (Potvin, 2009) or

machine learning techniques such as supervised learning or genetic programming (Benyahia et al 1998). A technique based on hybrid genetic algorithms was used (Jih et al 1999) for solving the problem of routing a vehicle with size constraints. Another example relies on Dijkstra algorithms (Barrachina et al 2014) and evolutionary strategies for finding an optimum path in a short period of time. Another approach presented in the project Emergency Vehicle Priority implies controlling the traffic lights in favor of the emergency crews (White, 2012), but this solution affects the rest of the urban traffic. A way of solving the routing problem using genetic algorithms is by combining (Chand et al 2010) the Bin Packing (BPP) with the traveling salesman problem. In the most recent approaches, the solution to this problem was obtained using the Intelligent Water-drop algorithm (Kaur et al 2014).

There are multiple approaches to the application of multi-agent systems in dynamic reconfiguration of routes (Shah 2012), such as MARS, Jabatos or Ant System, but they were used for management problems of transport resources (e.g. assigning buses to routes by minimizing the number of vehicles required and maximizing the number of requests in the system).

One solution (Darbucha 2013) for the dynamic routing problem is a combination of agent-based systems and dispatcher-based routing strategies. Other similar approaches (Talbot et al 2010) using multi-criteria decision-making, but at a global level dispatcher. In this case, real time data is received by the dispatcher which computes the emergency vehicle route. If an obstacle is blocking this route, the dispatcher is informed and can take other decisions. This simulation is limited at dispatcher level by introducing an additional node between data processing and decision-making algorithm.

This paper is organized as follows. In section 2, the authors present the design of the proposed control system, along with the principles of genetic algorithms. In section 3 a case study is discussed, for which an agent based simulation model has been developed in order to simulate traffic in an urban area. Finally, section 4 contains the conclusions.

2 Proposed system design

The concept introduced in this paper focuses on obtaining routes for emergency crews in an urban environment. Urban conglomerates (especially in areas that had not been initially developed for the amount of vehicles that can be found in today's society) suffer from traffic congestions. Therefore, the shortest path between two points is not necessarily the fastest. In this paper, the objective is to find the shortest routes with the least amount of traffic at any moment, while also

considering that the occupancy of a road segment might change during travel.

In order to achieve this goal, we propose the use of a Genetic Algorithm (GA) for computing the shortest of the least occupied routes through a network of intersections. The algorithm takes into account the degree of occupation for each intersection and chooses the route with the least amount of traffic.

Genetic Algorithms (GAs) are optimization heuristics able to perform rapid searches in large amounts of uncertain or incomplete data, with an inherent structure that allows parallelization. Given how GAs most often offer sub-optimal solutions is not a deterrent for the considered routing problem, because the urban traffic is in constant evolution and change. Thus, it is unadvisable to spend too much time and/or computing power on searching for a route that might not be as viable in the next minutes.

GAs are designed to search in multiple directions at once through the solution space, starting even from the initialization phase, in which a random pool of solutions (called a population) is generated. These solutions, known as individuals, are then allowed to evolve based on the principles of the Darwinist natural selection. Using mechanisms of selection (choosing certain individuals to contribute to creating the next generation as parents), recombination (generating new individuals by combining the characteristics of the selected parent), and mutation (necessary to maintain population diversity), a GA runs continuously, evaluating at each step all the possible solutions of the considered problem until a stop criterion is met (for instance, after a certain number of generations, or after reaching a certain tolerance for the solution). The evaluation procedure is run through what is known as a fitness function, that returns to which degree an individual can be deemed fit or not to be a viable solution.

Due to their heuristic nature, GAs can integrate, within the fitness function, objectives and performance indexes that are not necessarily defined in a formal manner. Moreover, adding more and more conditions in order to declare a possible individual as a fit solution to the considered problem is not computationally taxing (for instance, due to objective fusion techniques that might be hard to do with non-heuristic descriptions).

In recent research, GAs have been applied to problems in various fields. Their popularity keeps increasing due to their ease of use, effectiveness, or applicability. In control systems design (Fleming et al 2002) GAs have been efficiently used with both feedback (Lewin 1994) and feedforward controllers (Lewin 1996), modelling (Huang 2012) and parameter estimation (Bush et al 2011). In other fields, GAs have been used for various problems, either as stand alone procedures (Patrascu et al 2016), or in combinations

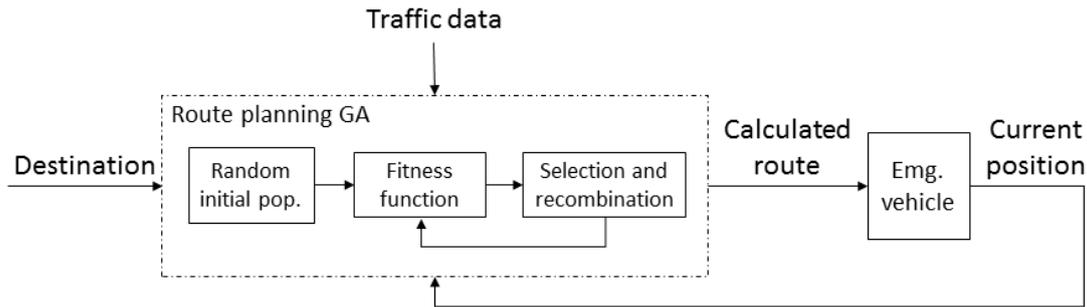


Figure 1. General principle of the dynamic route generation with genetic algorithms.

with other intelligent methods, like fuzzy systems (Alcala-Fdez et al 2009) or neural networks (Tao et al 2004).

In a real world implementation, the dynamic route generation algorithm would be developed to run independently onboard emergency vehicles, integrating traffic data from sensors throughout the city. It uses mobile computational resources available locally to the vehicle, thus eliminating the need for a central aggregation hub and offering scalability in case of major disasters, when more emergency crews are required in different parts of the city.

As the vehicle travels to the destination, the degree of occupation of the intersections along the route can change. In this case, the route is updated taking into account the new traffic data. The least occupied route is chosen based on the value of the corresponding fitness function (that computes a total degree of occupation for the entire suggested route).

Figure 1 presents the general principle of the proposed system. After a random initialization of the solution population, the GA computes the fitness function for each individual and, if necessary, genetic operators are applied to the population. The processes of evaluation, selection, and recombination are repeated until an individual with a high enough fitness is found or up to a maximum number of generations. Once a new route is generated by the GA, it is then transmitted to the driver.

At this point, whether the driver chooses to follow the suggested route or not, the GA starts its next run, in order to generate a new route by the time the vehicle reaches the next intersection along its path.

The entire system presented in figure 1 can be regarded as a control system in which the Genetic Algorithm (as *controller*) receives a destination (as a *setpoint*) and adjusts the trajectory (*controlled output*) of the emergency vehicle (*plant*) through a geographical area. The other traffic participants and the driver decision to ignore the suggested route are integrated into the control loop as *disturbances*. Each new suggested route is being transmitted to the driver

at each node of the road network (intersection) as a *command*.

Thus, the control problem of emergency crew routes can be defined as: an emergency vehicle (ambulance, fire truck, police car, etc.) receives a respond request to a given site while encountering as little traffic as possible during its journey. The emergency vehicle must travel from point A to point B, through an urban area comprised of a road network and traffic participants (regular vehicles and pedestrians).

The objective of the GA is to dynamically generate new routes for the emergency vehicle, supply these routes to the driver as the vehicle approaches each intersection, while taking into account traffic data that might be incomplete or inaccurate.

In what concerns the GA encoding of the vehicle routes, these are formed of a numeric representation in vectorial form, in which each position contains a route section identifier and the degree of occupation on the road segment leading to its associated intersection. The entire route is given by the position of each route section in the vector (its index).

Thus, an individual (or chromosome) is formed of genes that represent a route section each.

A route section is formed of that segment of a road between two intersections, on the traveling direction of the emergency vehicle. One road segment is associated with the intersection it travels towards, while the position of the emergency vehicle through a route is given by the road section & intersection pair it currently travels through. Thus, a path is formed of a starting point (either the initial position point A, or the current position), and as many route sections (road segment & intersection) as there can be delimited until the destination point B. For example, a path through 3 intersections will contain 4 genes.

A route section identifier can be defined as a coordinate pair or a unique identifier. This coordinate point can be, for instance, a physical geographical location for the beginning of each route section, a pair of coordinates for the start and finish of the route section, or even an ID number associated with each route section. For simplicity, in what follows, the

notation r will be used to denominate route section identifiers.

Thus, a route φ is comprised of n route sections defined by their position r_i in the urban area and their degree of occupancy D_i . Each pair $(r_i, D_i)_i$ has an associated index i that depicts their order in composing the route:

$$\varphi = [(r_0, D_0)_0 \ (r_1, D_1)_1 \ \dots \ (r_i, D_i)_i \ \dots \ (r_n, D_n)_n] \quad (1)$$

When computing the fitness of an individual, the algorithm takes into account both the degree of occupation of the road segments on the possible paths to the event site and the distance the destination.

For a path formed of n intersections, the route fitness F is computed by analyzing each section and cumulating the results for the entire route:

$$F^{-1} = \sum_{i=1}^n \frac{\delta \cdot D_i}{g(r_0, r_i)} \quad (2)$$

where δ is a scaling factor, D_i is degree of occupancy on the i -th section of the route, and $g(r_0, r_i)$ is a function that scales the penalty given to those sections of the route that are farthest from the starting section r_0 (either the initial section or the current section the emergency vehicle is traveling through).

The fitness value F of the route needs to be maximized, therefore the best routes are the ones with the lowest degree of occupancy.

The closer a road section (delimited by the intersections through which the vehicle must travel), the more it is contributing to the computed fitness value than the further ones. In this way, when calculating the fitness of a possible route, less importance is given to the degree of occupancy of farther intersections, as it may change by the time the vehicle is actually there.

3 Case Study

In order to test the proposed system, a simulation model needs to be designed that incorporated the complexity of urban traffic systems. In this respect, the most recent development in complex systems modelling belong to the field of Agent Based Modelling and Simulation (Patrascu et al. 2015), which provides engineers with Agent Based Simulation Models (ABSM) that are able to describe the behaviours of heterogeneous traffic participants, as well as their interactions and interdependencies.

In an ABSM, an agent is a persistent entity characterised by internal states. This entity interacts with its environment or with other agents, ultimately causing changes in the environment or in the states of the other agents. An ABSM is a collection of agents, their states, and the rules that govern interactions from

agents to agents, from agents to the environment, and from the environment to the agents.

Every agent of an ABSM has inputs and outputs, but it can also regulate their own internal states through local feedback loops. The internal state of an agent is governed by rules. Moreover, the behaviour of an agent is represented by the set of actions it performs (outputs) according to internal state and inputs.

As a whole, the ABSM has its own states, inputs, and outputs, but from a holistic perspective, the states of the entire simulation model integrates the complex interactions between the agents involved, by using rules to describe these behaviours. Most often, ABSM are evolving complex systems, used in generative experiments (Bertolotti, 2014), in order to test the viability of a solution that has little chance to be formally evaluated in realistic computing times (much like in the case of heuristics).

3.1 ABSM description

The agent-based simulation presented in this paper was implemented using the NetLogo (Wilensky et al 2015) environment, which offers access to the advantages of using agent based simulation models (ABSM) previously stated. Moreover, NetLogo offers way to implement user interfaces that is easy and accessible to researchers who don't necessarily have a strong programming background. NetLogo has been successfully used for proving and testing complex systems theorems and hypotheses. An in depth discussion on the advantages and disadvantages of using NetLogo for the simulation of complex control systems can be found in our previous work (Patrascu et al. 2015).

Thus, we illustrate the application of genetic algorithms in computing the quickest route (from the degree of occupation perspective) of an emergency vehicle through a simulated urban environment. The simulation application allows the user to set the number and types of crossroads present in the world model. Before starting the simulation, one can also chose the duration of different traffic light phases, the priorities for incoming traffic in a crossroad, as well as the rate at which new vehicles are inserted in the simulation. In this setup stage, the user can also chose the parameters that control the genetic algorithm responsible for planning the quickest route for the emergency vehicle.

For simulation purposes, the generated environment contains roads with four lanes (two lanes per travel direction) and specific traffic lights for turning left or going straight in order to increase the complexity of the road network model, as would be expected to happen in real urban areas: multiple lanes, directional traffic lights and so on. After starting the simulation, an emergency vehicle can be added to the environment. The parameters indicating the start and destination

position of the emergency vehicle, as well as its priority, can be set from the model's interface.

The traffic participants are simulated using mobile agents and consist of normal and emergency vehicles and pedestrians. The simulation environment, NetLogo, uses two types of agents: fixed and mobile. The fixed agents build the static parts of the simulation model, which, for the agent-based model compose their relevant environment, while the mobile agents are used to simulate dynamic entities. In this case, vehicles are mobile agents. Their behaviour is modelled using knowledge-based representations in a manner that most accurately describes the decision making processes of drivers. NetLogo offers the possibility of modelling these entities as complex or simplistic as the user desires. In this paper, the vehicle (and thus driver) behaviour is as close as possible to the behaviour of vehicles in real world traffic.

Emergency vehicles have a higher priority through traffic than civilian ones. The pedestrians behave similarly to the vehicles, with the exception that they can only move on the sidewalks. In figure 2 an example of simulated environment is presented, highlighting the intersection that the emergency vehicle (police) is currently traversing. The police car can be seen entering the inner crossroad area from the east and traveling west.

After creating the emergency vehicle, the user can chose to highlight and inspect the police car as it travels through the urban environment. This is implemented by opening a new window (presented in figure 2) that shows a zoom of the world model, centered on the emergency vehicle, and that follows it as it travels. In the zoom window presented here, we can see the state of the traffic lights in each intersection. The pedestrians obey the traffic lights placed in the corners of the intersection, while the vehicles follow the signals of the lights placed between the lanes. For vehicles, the traffic light closer to the center of the crossroad is for turning left and the other one for going straight. In figure 2, on the east-to-west road, we can see the traffic light for going straight showing a green light and the one for turning left showing red, thus allowing the police car to traverse the intersection and stopping the two civilian cars occupying the next lane from turning left. The route for the emergency vehicle is computed when the agent that models this vehicle is inserted in the simulation environment. The user can chose to have the route updated each time the vehicle crosses an intersection, or just request one calculation, at the beginning of the vehicle's journey.

During the simulation, at each time step, the phases of the traffic lights are computed for each crossroad, using a distributed traffic control system based on vehicle priority. Each traffic light phase (North, East, West, South) is computed using the corresponding road

occupancy data, which is obtained from sensors placed on each road section entering the intersection. When a civilian vehicle enters a road section, the occupancy is increased by 1 and when the vehicle leaves the road section, it is decreased by the same amount. In the case of an emergency vehicle, the occupancy is modified according to the priority of the vehicle. For more information regarding these types of control systems and their simulation using ABM, please consult our previous work (Patrascu et al 2015).



Figure 2. ABM world overview.

As the simulation runs, at each time step, new vehicles are inserted in the simulation environment. The user can set the frequency for inserting vehicles from the east and west or north and south side of the map. The vehicles can only be inserted on a lane connected to the edge of the map if there is available room. Depending on a user setting, the civilian vehicles can give way to the emergency vehicles by temporary switching lanes.

3.2 A simulation example

For a more facile visualization, the world model presented in this case study has the road sections (segment & intersection pairs) displayed in a grid. In what follows, we have chosen a 4x4 grid of intersections. Each route section is depicted by an identifier of the form (x, y) where x is the position on each line of the grid, while y is the position on each column. For example, $(1, 3)$ is the intersection marked A in figure 3. The starting point of the emergency vehicle (police car) is marked A in figure 3, while the final destination is B, at coordinates $(4, 2)$.

For the first experiment (Exp.1), the GA presented in the previous sections will be run as the vehicle departs from point A. In the second experiment (Exp.2), the GA will be run at the entry of each route section, re-generating the vehicle's route dynamically

as it travels through the urban area. The results of both experiments are presented in table 1 (where D is a matrix containing the degrees of occupancy for each route section, and φ is the route returned by the GA). For Exp.1, the GA is run only at point A. For Exp.2, the table shows the first 2 runs of the GA, first in point A and then in the next intersection, showing how the occupancy of the road segments has changed while the vehicle has travelled through the first route section, from (1, 3) to (2, 3).

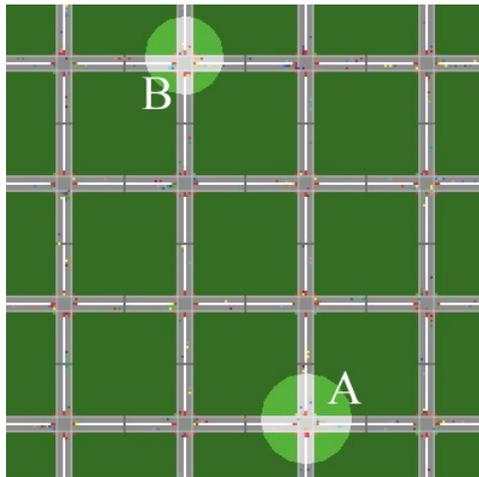


Figure 3. Starting point and destination of the emergency vehicle.

Table 1. Simulation run results

		Exp.1	Exp. 2																																
GA run at point A (1, 3)	D	<table border="1"> <tr><td>34</td><td>8</td><td>7</td><td>37</td></tr> <tr><td>40</td><td>10</td><td>12</td><td>46</td></tr> <tr><td>38</td><td>14</td><td>10</td><td>52</td></tr> <tr><td>18</td><td>16</td><td>9</td><td>36</td></tr> </table>	34	8	7	37	40	10	12	46	38	14	10	52	18	16	9	36	<table border="1"> <tr><td>34</td><td>11</td><td>14</td><td>44</td></tr> <tr><td>52</td><td>19</td><td>28</td><td>57</td></tr> <tr><td>53</td><td>17</td><td>15</td><td>63</td></tr> <tr><td>19</td><td>17</td><td>14</td><td>52</td></tr> </table>	34	11	14	44	52	19	28	57	53	17	15	63	19	17	14	52
	34	8	7	37																															
40	10	12	46																																
38	14	10	52																																
18	16	9	36																																
34	11	14	44																																
52	19	28	57																																
53	17	15	63																																
19	17	14	52																																
	φ	 [(1,3), (2,3), (3,3), (4,3), (4,2)] Total Cars: 46	 [(1,3), (2,3), (2,2), (3,2), (4,2)] Total Cars: 76																																
GA run at point (2, 3)	D	<table border="1"> <tr><td>44</td><td>18</td><td>14</td><td>58</td></tr> <tr><td>51</td><td>20</td><td>29</td><td>42</td></tr> <tr><td>50</td><td>18</td><td>28</td><td>61</td></tr> <tr><td>29</td><td>31</td><td>10</td><td>44</td></tr> </table>	44	18	14	58	51	20	29	42	50	18	28	61	29	31	10	44	<table border="1"> <tr><td>42</td><td>16</td><td>8</td><td>45</td></tr> <tr><td>55</td><td>22</td><td>18</td><td>66</td></tr> <tr><td>51</td><td>20</td><td>11</td><td>67</td></tr> <tr><td>25</td><td>11</td><td>18</td><td>43</td></tr> </table>	42	16	8	45	55	22	18	66	51	20	11	67	25	11	18	43
	44	18	14	58																															
51	20	29	42																																
50	18	28	61																																
29	31	10	44																																
42	16	8	45																																
55	22	18	66																																
51	20	11	67																																
25	11	18	43																																
	φ	 [(2,3), (3,3), (4,3), (4,2)] Total Cars: 89	 New route: [(2,3), (3,3), (4,3), (4,2)] Total Cars: 53 Old route: [(2,3), (2,2), (3,2), (4,2)] Total Cars: 69																																

In the first experiment, the vehicle's route is computed only once, at the beginning of its journey. As the vehicle travels through the area, the degree of occupancy might change on the previously selected routes. For instance, there is an alternate route starting at (2, 3) that has less vehicles traveling through it. To account for the changes in the environment, in the second experiment, the GA has been run at the beginning of each route section in order to determine if the route should be kept or dynamically changed. In table 1 a new and old route comparison is shown to illustrate this phenomenon.

The system we proposed in section 2 achieves its objectives of selecting the routes with the lowest degree of occupancy in a given urban area. Moreover, the system is capable of dynamically re-generating the emergency vehicle's route based on new traffic data and changes in the urban road network.

The preliminary testing conducted in this study shows great potential for the further development of evolutionary algorithms as controllers in closed loop systems. The next step in a thorough validation procedure is to develop simulation models that use real world maps, followed by real world testing. These are some of the main steps in the cycle of control systems development, which are designed to take a formal idea and make sure the final product works smoothly and within required performances.

4 Conclusions

In this study, a system for generating emergency vehicles routes has been presented. The system is able to account for changes in the environment or in urban traffic and dynamically supply new, better routes to the emergency vehicle drivers.

For this, a genetic algorithm has been designed that can recalculate the paths of the emergency responders while they are already on their journey to an event site. The entire route adjustment procedure is encapsulated in a control system scheme, in which the controlled variable is the route of the vehicle, while the other traffic participants are disturbances. The proposed system has been tested in a simulated environment specifically designed to emulate the complexity of the urban traffic.

Among the advantages of this type of system are: the possibility to run locally, on each emergency vehicle, thus eliminating the need for a central server type entity and overloaded communication networks; the possibility of implementation on devices that are already available (smartphones or tablets); not relying on traffic control systems, that can sometimes be overwhelmed, that can only manage an intersection at a time, or that can, at times, be offline (for maintenance or other reasons).

Moreover, for this type of system, context awareness is implicitly achieved. The intrinsic nature of control systems allows them to both collect information from the environment (either by direct sensing or by accessing real time traffic databases), and to interact with the it via the driver. Complete automation, although attractive from a point of view of vehicle autonomy, removes the driver from the decision making process. Thus, the middle ground we proposed seems the most reasonable.

Some of the limitations of this sort of system are related to the inherent heuristic nature of genetic algorithms, the proposed method requiring perhaps a sort of hybridization with formal methods, metaheuristics, memetic algorithms.

Further research endeavours in what concerns the proposed dynamic route generation using genetic algorithms include the design of specialized selection methods and chromosome encoding for routes and the test of the presented system, first in simulated environments with higher complexity, and then in real world scenarios.

Acknowledgements

We would like to thank our former student Eugen Marius Petre for his work within the EMAS (Emergent Multiscale Agents and Services) Research Group.

References

- S. Andjelic, G. Panic, and A. Sijacki. Emergency response time after out-of-hospital cardiac arrest. *European journal of internal medicine*, 22(4): 386-393, 2011.
- G. Asvin and G. Volker. Solving a dynamic real-life vehicle routing problem. *Operations research proceedings*: 367-372, 2006.
- B. O. Bush, J.-P. Hosom, A. Kain, and A. Amano-Kusumoto. Using a genetic algorithm to estimate parameters of a coarticulation model. In *Twelfth Annual Conference of the International Speech Communication Association*, pages 2677-2680, 2011.
- C.-F. Huang. A hybrid stock selection model using genetic algorithms and support vector regression. *Applied Soft Computing*, 12(2): 807-818, 2012.
- D. R. Lewin. A genetic algorithm for MIMO feedback control system design. *Adv. Control Chem. Process*, 101: 2014, 1994.
- D. R. Lewin. Multivariable feedforward control design using disturbance cost maps and a genetic algorithm. *Computers & chemical engineering*, 20(12): 1477-1489, 1996.
- D. Barbucha. A multi-agent approach to the dynamic vehicle routing problem with time windows. In *International Conference on Computational Collective Intelligence*, pages 467-476, 2013.
- G. B. Dantzig and J. H. Ramser. The truck dispatching problem. *Management science*, 6(1): 80-91, 1959.
- G. F. Rushworth, C. Bloe, H. L. Diack, R. Reilly, C. Murray, D. Stewart, and S. J. Leslie. Pre-hospital ECG e-transmission for patients with suspected myocardial infarction in the highlands of Scotland. *International journal of environmental research and public health*, 11(2): 2346-2360, 2014.
- H. N. Psaraftis. A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. *Transportation Science*, 14(2): 130-154, 1980.
- H. N. Psaraftis. Dynamic vehicle routing: Status and prospects. *Annals of operations research*, 61(1): 143-164, 1995.
- I. Benyahia and J.-Y. Potvin. Decision support for vehicle dispatching using genetic programming. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 28(3): 306-314, 1998.
- J. White. Emergency vehicle priority. *The Queensland Surveying and Spatial Conference*, Brisbane Australia, 2012.
- J. Barrachina, P. Garrido, M. Fogue, F. J. Martinez, J.-C. Cano, C. T. Calafate, and P. Manzoni. Reducing emergency services arrival time by using vehicular communications and Evolution Strategies. *Expert Systems with Applications*, 41(4): 1206-1217, 2014.
- J.-Y. Potvin. A review of bio-inspired algorithms for vehicle routing. *Bio-inspired algorithms for the vehicle routing problem*, 1-34, 2009.
- J. Alcalá-Fdez, R. Alcalá, M. J. Gacto, and F. Herrera. Learning the membership function contexts for mining fuzzy association rules by using genetic algorithms. *Fuzzy Sets and Systems*, 160(7): 905-921, 2009.
- M. M. Shah. Artificial Intelligence: Vehicle Routing Problem and Multi Agent System. *International Journal of Computer Applications*, DRISTI(1): 1-3, 2012.
- M. Patrascu and A. Ion. Evolutionary Modeling of Industrial Plants and Design of PID Controllers. In H. E. Ponce Espinosa, editor, *Nature-Inspired Computing for Control Systems*, volume 40 of *Studies in Systems, Decision and Control*, pages 73-119, 2016.
- M. Patrascu, A. Ion, and V. Constantinescu. Agent based simulation applied to the design of control systems for emergency vehicles access. In *ITS Telecommunications (ITST), 2015 14th International Conference on*, 50-54, 2015.
- O. B. G. Madsen, H. F. Ravn, and J. M. Rygaard. A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. *Annals of operations Research*, 60(1): 193-208, 1995.
- P. Chand, B. S. P. Mishra, and S. Dehuri. A multi objective genetic algorithm for solving vehicle routing problem. *International Journal of Information Technology and Knowledge Management*, 2(2): 503-506, 2010.
- P. Toth and D. Vigo. The vehicle routing problem, ser. SIAM monographs on discrete mathematics and applications. *Society for Industrial and Applied Mathematics*, 2002.
- P. J. Fleming and R. C. Purshouse. Evolutionary algorithms in control systems engineering: a survey. *Control engineering practice*, 10(11): 1223-1241, 2002.
- P. T. Pons, J. S. Haukoos, W. Bludworth, T. Cribley, K. A. Pons and V. J. Markovchick. Paramedic response time:

- does it affect patient survival?. *Academic Emergency Medicine*, 12(7): 594-600, 2005.
- Q. Tao, X. Liu, and M. Xue. A dynamic genetic algorithm based on continuous neural networks for a kind of non-convex optimization problems. *Applied mathematics and computation*, 150(3): 811-820, 2004.
- R. Kaur, R. Kaur, and N. Kaur. A Modified transmission Algorithm for Resolving Vehicle Routing Problem by Intelligent Water drop Algorithm. *International Journal on Recent and Innovation Trends in Computing and Communication*, 2(10): 3108-3112, 2014.
- R. Séguin, J.-Y. Potvin, M. Gendreau, T. G. Crainic, and P. Marcotte. Real-time decision problems: An operational research perspective. *Journal of the Operational Research Society*, 48(2): 162-174, 1997.
- T. H. Blackwell and K. S. Jay. Response time effectiveness: comparison of response time and survival in an urban emergency medical services system. *Academic Emergency Medicine*, 9(4): 288-295, 2002.
- T. Bertolotti. Generative and Demonstrative Experiments. In L. Magnani, editor, *Model-Based Reasoning in Science and Technology*, volume 8 of *Studies in Applied Philosophy, Epistemology and Rational Ethics*, pages 479-498, 2014.
- U. Wilensky and W. Rand. *An introduction to agent-based modeling: modeling natural, social, and engineered complex systems with NetLogo*, MIT Press, 2015.
- V. Talbot and I. Benyahia. Complex Application Architecture Dynamic Reconfiguration Based on Multi-criteria Decision Making. *International Journal of Software Engineering & Applications*, 1(4): 19-37, 2010.
- W.-R. Jih and J. Y.-J. Hsu. Dynamic vehicle routing using hybrid genetic algorithms. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation, Detroit, USA*, pages 453-458, 1999.