

# Using Evolution Strategies to Reduce Emergency Services Arrival Time in case of Accident

Javier Barrachina, Piedad Garrido, Manuel Fogue,  
Francisco J. Martinez  
University of Zaragoza, Spain  
Email: {barrachina, piedad, mfogue, f.martinez}@unizar.es

Juan-Carlos Cano, Carlos T. Calafate, Pietro Manzoni  
Universitat Politcnica de Valncia, Spain  
Email: {jucano, calafate, pmanzoni}@disca.upv.es

**Abstract**—A critical issue, especially in urban areas, is the occurrence of traffic accidents, since it could generate traffic jams. Additionally, these traffic jams will negatively affect to the rescue process, increasing the emergency services arrival time, which can determine the difference between life or death for injured people involved in the accident. In this paper, we propose four different approaches addressing the traffic congestion problem, comparing them to obtain the best solution. Using V2I communications, we are able to accurately estimate the traffic density in a certain area, which represents a key parameter to perform efficient traffic redirection, thereby reducing the emergency services arrival time, and avoiding traffic jams when an accident occurs. Specifically, we propose two approaches based on the Dijkstra algorithm, and two approaches based on Evolution Strategies. Results indicate that the Density-Based Evolution Strategy system is the best one among all the proposed solutions, since it offers the lowest emergency services travel times.

## I. INTRODUCTION

Traffic accidents represent a big problem for drivers and a serious burden for the economy of all the countries. A close look at traffic accidents shows that many of the casualties and serious medical conditions take place during the time elapsed between the accident occurrence and the arrival of the medical assistance. The so called ‘Golden Hour’ [1] after a car crash is the time within which medical or surgical intervention by a specialized trauma team has the greatest chance of saving lives. If more than 60 minutes have elapsed by the time the injured arrives to the operating table, the chances of survival fall sharply. Typical arrival of medical help takes about 15 minutes, but initial access and treatment starts 25 minutes after the accident. Transportation of the injured to the hospital usually takes place 50 minutes later. Therefore, time is critical for the survival of the injured in a severe crash incident, and any technology capable of providing a fast and efficient rescue operation after a traffic accident takes place will increase the probability of survival of the injured, and reduce the injury severity.

Cooperative vehicle systems have become an increasingly popular transportation paradigm in recent years. Wireless technologies, through vehicular networks, enable peer-to-peer mobile communications among vehicles (V2V), as well as communications between vehicles and infrastructures (V2I). Using these technologies, crashed vehicles are able to notify the emergency services about the occurrence of an accident. In addition, emergency services can dynamically redistribute

traffic by communicating or suggesting new routes to vehicles. These routes can be calculated using different methods such as Dijkstra-based algorithms, genetic algorithms, or evolution strategies. Evolutionary Algorithms imitate the principles of natural evolution as a method to solve parameter optimization problems. They have been successfully used to solve various types of optimization problems [2], since they provide an optimal solution without checking all the possible solutions, thereby reducing the execution time drastically. Evolution Strategies are a kind of Evolutionary Algorithm with the particularity that the mutation steps are included in the chromosome. This kind of Evolutionary Algorithms obtains very good results in numerical optimization problems, especially when working on continuous variables.

In this paper, we propose four different approaches to minimize the emergency services arrival time when an accident takes place in urban scenarios, also trying to avoid traffic jams. In particular, two of them are based on the Dijkstra algorithm, and the other two are based on Evolution Strategies. Additionally, we evaluated the four proposed solutions in three different scenarios with different topologies to determine the best solution in terms of travel times of the emergency services and the rest of vehicles.

So far, evolutionary algorithms have been widely used in the field of dynamic traffic distribution (e.g., [3], [4], and [5]). However, unlike our proposal, existing works do not focus on reducing the rescue time of the emergency services, or exploit the advantages of using vehicular communication capabilities to calculate vehicles routes. Additionally, in all these works, authors only consider a specific scenario for simulations to assess their proposal. From our point of view, simulating only one specific scenario is inadequate when presenting a vehicle routing model (even in real scenarios since it can lead to unrepresentative and inaccurate results). We consider that simulating different (and realistic) topologies is necessary, since the roadmap topology significantly affects the obtained results [6].

This paper is organized as follows: In Section II we present our four different re-routing systems (i.e., Dijkstra, Density-Based Dijkstra, Evolution Strategy, and Density-Based Evolution Strategy). Section III introduces the simulation environment used to assess our proposed schemes. Section IV shows the obtained results. Finally, Section V concludes this paper.

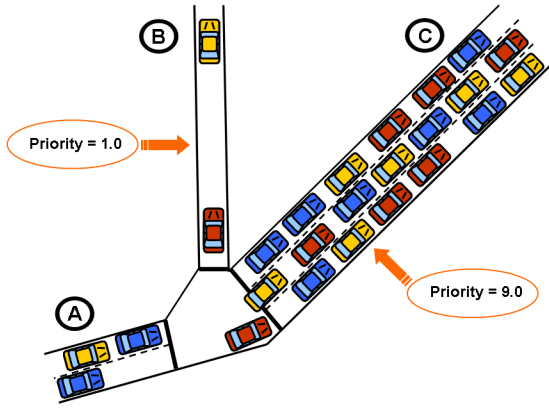


Fig. 1. Example of a traffic jam when the street priority is given by the number of lanes.

## II. OUR PROPOSED VEHICLE ROUTING SYSTEMS

In this Section, we propose four different vehicle routing approaches with the aim of ensuring that emergency services arrive at the place of the accident as soon as possible, whereas the rest of vehicles are not significantly affected, i.e., their travel times do not increase considerably, avoiding the possible traffic jams caused by the accident. Next, we present them in detail.

### A. Dijkstra

This system aims at obtaining the shortest route between two map locations by using the Dijkstra algorithm [7], specifically adapted to roads and streets, and taking into account the length and priority of the streets. The priority of each street indicates the preference it has with respect to the others for a vehicle when it arrives to a junction.

Specifically, in this approach, the street priority is calculated by using the number of lanes per street, assigning higher priority to the widest streets (i.e., with higher number of lanes). Figure 1 shows an example of this situation. As shown, vehicles arrive to the junction through street A. Using this system and considering the priorities shown in the figure (1.0 for street B and 9.0 for street C), the majority of vehicles continue their route through street C (90% of vehicles since this street has a greater number of lanes), collapsing it. However, street B has less traffic density, with a more fluid traffic.

This proposed system uses a static model for street priorities, where a priority is given to each street, and priorities do not change under any circumstance. This issue could generate two kind of problems when an accident occurs: (i) there could be traffic jams in specific areas of the scenario, whereas other areas present very low traffic, and (ii) the streets selected as routes for the emergency services do not present low priority for the rest of vehicles in order to reduce the number of potential vehicles blocking the streets.

The main advantage of this system is the low computational cost since it does not need to know the current traffic density

or the emergency service routes; in addition, when an accident occurs, this approach can be applied immediately.

### B. Density-Based Dijkstra

This proposed system is similar to the previous one, with the difference that, in this case, we take into account the traffic density in the area when the street priorities are assigned. To develop this method, those streets leading vehicles to high traffic density areas, are penalized. When an accident occurs, all the vehicles involved send a warning message using Vehicular Networks, and once the control systems are notified, they apply the vehicular density estimation approach proposed in [8]. In addition, the streets through which emergency services circulate to arrive at the accident site are penalized for the rest of vehicles. Specifically, in this proposed system, we proceed as follows:

- *Step 1:* we prioritize streets by normalizing the values (see Equation 1). As shown, the normalized values start in 1 and end in 10 ( $N_{min}$  and  $N_{max}$ , respectively).

$$N_x = \frac{(P_x - P_{min}) \cdot (N_{max} - N_{min})}{P_{max} - P_{min}} + N_{min}$$

where :

$$N_{min} = 1$$

$$N_{max} = 10$$

- *Step 2:* the normalized value for the rest of the areas ( $N_x$ ) is calculated by using a proportion between the minimum and the maximum traffic density percentages, and the traffic density of the area which we want to calculate the normalized value ( $P_{min}$ ,  $P_{max}$ , and  $P_x$ , respectively).
- *Step 3:* with the aim of penalizing streets with a high traffic density, we apply Equation 2. In this Equation, we obtain the inverse value calculated above ( $S_x$ ), since a higher priority value has more priority, and we multiply this value by the number of lanes of the street ( $L_x$ ).

$$S_x = (N_{max} - N_x + 1) \cdot L_x$$

- *Step 4:* with the aim of calculating the fastest route for the emergency services vehicle, this approach applies a simple Dijkstra algorithm for each one, calculating the shortest route between two map locations (accident site and hospital, police station, firehouse, etc.), regardless of traffic density. Note that, in this case, we do not take into account the street priorities since emergency vehicles always have to reach rapidly to the accident location.
- *Step 5:* as shown in Equation 3, we penalize these streets through which emergency services circulate ( $S_{x_e}$ ) by giving them a priority corresponding to the number of lanes (e.g., a street with four lanes has a priority of 4).

$$S_{x_e} = L_x$$

- *Step 6:* we calculate the new vehicle routes using a Dijkstra-Based algorithm taking into account the streets

priorities, since the shortest path could not be the fastest path.

Equation 4 shows an example of street priorities calculation. As shown, we have three different areas which contain the following percentage of traffic vehicles:  $P_{min} = 20\%$ ,  $P_{max} = 50\%$ , and  $P_x = 30\%$  of the total of vehicles. Also, we have three streets located in the aforementioned areas with these numbers of lanes ( $L_{min} = 3$ ,  $L_{max} = 2$ , and  $L_x = 1$ ). Since we have the maximum and minimum normalized values ( $N_{min}$  and  $N_{max}$ ), we calculate the other street normalized value ( $N_x$ ) by using Equation 1. Finally, we obtain the street priorities ( $S_{min}$ ,  $S_{max}$ , and  $S_x$ ) by using Equation 2, thereby obtaining street priorities of 30, 2, and 7 respectively.

$$\begin{aligned}
P_{min} &= 20, P_{max} = 50, P_x = 30 \\
N_{min} &= 1, N_{max} = 10 \\
L_{min} &= 3, L_{max} = 2, L_x = 1 \\
N_x &= \frac{(P_x - P_{min}) \cdot (N_{max} - N_{min})}{P_{max} - P_{min}} + N_{min} \\
N_x &= \frac{(30 - 20) \cdot (10 - 1)}{50 - 20} + 1 = 4 \\
S_x &= (11 - N_x) \cdot L_x \\
S_{min} &= (11 - 1) \cdot 3 = 30 \\
S_{max} &= (11 - 10) \cdot 2 = 2 \\
S_x &= (11 - 4) \cdot 1 = 7
\end{aligned} \tag{4}$$

### C. Evolution Strategy

Due to the high computational cost of calculating all possible combinations of street priorities to find the optimal solution, we consider interesting to apply an Evolution Strategy to address our problem, i.e., the reduction of the emergency services arrival time to the accident location. As in the previous proposed approaches, this scheme applies the Dijkstra algorithm for each emergency vehicle in order to calculate the emergency services routes, and also we penalize the streets selected for the emergency services vehicles.

Next, we present the main characteristics of our Evolution Strategy (i.e., definition of variables, fitness function, mutation, recombination, parents selection, and survivors selection).

1) *Definition of Variables*: An individual, i.e., a potential solution of our system, encodes a possible solution into a chromosome based structure (genotype) [9]. In this case, a vector of float point numbers which contains the priority value of each street (as shown in Figure 2) is considered. Street priorities are randomly selected in the vectors of the initial population for each street for the first time.

2) *Fitness Function*: Selection is a process in which solutions are selected for recombination based on their fitness values. Here, fitness refers to a measure of profit, utility, or goodness to be maximized while exploring the solution space. Our system has three different fitness functions designed to minimize the arrival time for the emergency vehicles and the

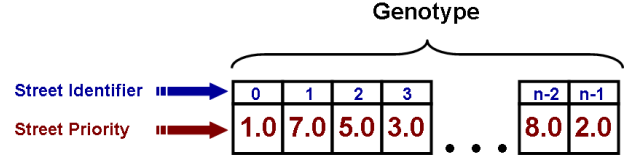


Fig. 2. Example of a genotype for street priorities.

travel time of the rest of vehicles: (i) Fitness Function 1 gives double importance to the arrival time of emergency services ('e' represents emergency services vehicles, and 'r' represents the rest of Regular vehicles) (see Equation 5), (ii) Fitness Function 2 assigns the same importance to both arrival times (see Equation 6), and (iii) Fitness Function 3 gives double importance to the arrival time of the rest of vehicles (see Equation 7). Although the latter should not perform well, since our main goal is to reduce the time required by the emergency vehicles to reach the accident location, we consider interesting to evaluate it to assess whether the system is able to significantly reduce the travel time of the rest of vehicles, while slightly increasing the the emergency services' arrival time.

$$FitnessFunction1 = 2 \cdot \frac{\sum_{i_e=0}^{n_e} t_{i_e}}{n_e} + \frac{\sum_{i_r=0}^{n_r} t_{i_r}}{n_r} \tag{5}$$

$$FitnessFunction2 = \frac{\sum_{i_e=0}^{n_e} t_{i_e}}{n_e} + \frac{\sum_{i_r=0}^{n_r} t_{i_r}}{n_r} \tag{6}$$

$$FitnessFunction3 = \frac{\sum_{i_e=0}^{n_e} t_{i_e}}{n_e} + 2 \cdot \frac{\sum_{i_r=0}^{n_r} t_{i_r}}{n_r} \tag{7}$$

3) *Mutation*: In an Evolution Strategy there is a strong emphasis on the mutation to create the offspring. Additionally, mutation is implemented by adding a random 'noise' obtained from a Gaussian distribution. Mutation parameters change during the execution of the algorithm. In our proposal, we use an Uncorrelated Mutation with  $n$  Step Sizes. The mutation mechanism applies the functions included in Equation 8, where  $\sigma$  is the mutation step size,  $\tau$  is the scale parameter for the mutation step sizes, and  $n$  is the number of individuals.

$$\begin{aligned}
\sigma'_i &= \sigma \cdot e^{\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)}, \\
x'_i &= x_i + \sigma'_i \cdot N_i(0,1) \\
\text{where :} \\
\tau' &\propto \frac{1}{\sqrt{2n}} \\
\tau &\propto \frac{1}{\sqrt{2\sqrt{n}}}
\end{aligned} \tag{8}$$

Using this kind of mutation, our genotype contains values  $x$  (street priority) and values  $\sigma$  (mutation step sizes), as shown in Figure 3.

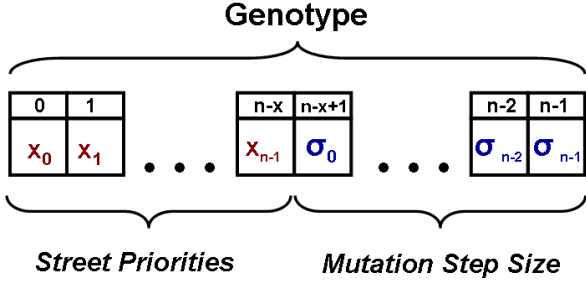


Fig. 3. Example of genotype formed by street priorities and mutation step sizes.

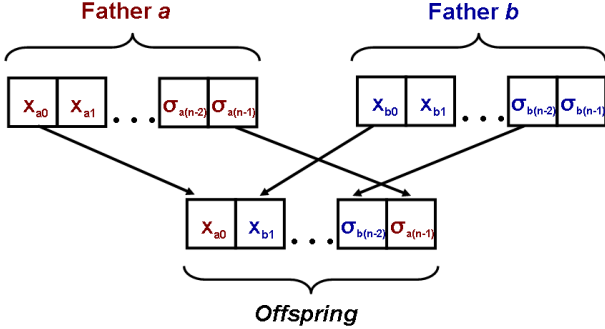


Fig. 4. Example of local discrete recombination.

To avoid too small standard deviations providing a negligible effect, we limit the value of the step sizes using a threshold ( $\varepsilon_0$ ), i.e.,  $\sigma' < \varepsilon_0 \Rightarrow \sigma' = \varepsilon_0$ .

4) *Recombination*: The basic recombination scheme in Evolution Strategies requires two parents to create a child. For  $\lambda$  descendants, the recombination process is performed  $\lambda$  times. There are two variants of recombination depending on how parental alleles are recombined:

- Discrete Recombination: one of the alleles of the parents is chosen with equal probability for both parents.
- Intermediate Recombination: the parental allele values are averaged.

Furthermore, two parents can be used, randomly obtained from the population of  $\mu$  individuals, for each component ( $i \in \{1..n\}$ ) of the offspring. This is known as *Global* recombination, and the variant in which only two parents are selected for the total of components is called *Local* recombination.

In our proposed system, we apply Local Discrete Recombination, since this method is one of the most widely used in this kind of algorithms, and it provides a good performance in most cases. As shown in Figure 4, each child allele is chosen with equal probability for both parents.

5) *Parents Selection*: The parents selection in Evolution Strategies does not depend on their fitness values. Parents are chosen randomly by using a uniform distribution from the population of  $\mu$  individuals.

6) *Survivors Selection*: The Survivors Selection consists on deterministically choosing the  $\mu$  best individuals, after creating

$\lambda$  descendants and calculating their fitness. There are two kinds of Survivor Selection:

- Selection ( $\mu, \lambda$ ): only the individuals of the offspring are considered to generate the next generation.
- Selection ( $\mu + \lambda$ ): survivors are selected from the union of parents and descendants.

Our proposed scheme uses Selection ( $\mu + \lambda$ ), since using Selection ( $\mu, \lambda$ ) descendants could produce worse results, delaying the achievement of the best solution.

#### D. Density-Based Evolution Strategy

With the aim of reducing the system runtime, we propose an Evolution Strategy with the same characteristics as the Evolution Strategy System (presented in the previous Subsection), but in this case we do not obtain the initial population randomly. We consider that by using the traffic density information, our system will be able to reduce the time required to find the optimal solution (by reducing the number of generations). Specifically, this approach combines both the Density-Based Dijkstra and the Evolution Strategy schemes.

Instead of getting the initial population randomly, we start the procedure by taking into account two different genotypes: (i) a genotype which contains street priorities based on the number of lanes, and (ii) a genotype which contains street priorities based on traffic density. The rest of individuals of the initial population are obtained by recombining these two genotypes. Street priorities based on the number of lanes are obtained by squaring the number of lanes of each street, and the street priorities based on traffic density and emergency vehicles routes are obtained by using the method proposed in the Density-Based Dijkstra approach. Then, we make a first recombination with them, selecting the  $n$  best descendants in order to generate a first offspring, so approaching to the best solution. This improvement will make the system reach the optimal solution in less time than using a random initial population.

Figure 5 shows an example of the objective of this solution. As shown, initializing the population accounting for the traffic density and the number of lanes could make it possible to obtain better solutions with a lower number of offsprings, thereby reducing the system runtime. In particular, while the non-density-based system would have created  $x_{ndb}$  generations to obtain the  $y_{db}$  fitness value, our density-based proposed system would obtain this value in its first generation. The initial executions would be avoided and, therefore, this approach would save crucial time.

### III. SIMULATION ENVIRONMENT

Regarding the traffic simulation, we use the Simulation of Urban MObility (SUMO), an open source, microscopic, continuous-space traffic simulator designed to handle large road networks [10]. To increase the level of realism of our simulations, we use real scenarios consisting of downtown areas from the cities of Rome (Italy), and New York (USA) imported directly from OpenStreetMap [11].

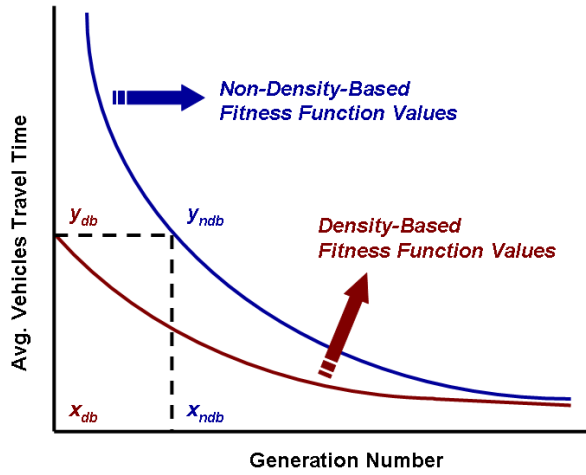


Fig. 5. Example of fitness function values using both proposed intelligent systems (i.e., Evolution Strategy and Density-Based Evolution Strategy).

TABLE I  
PARAMETERS USED FOR THE SIMULATIONS

Parameter	Value
number of simulations	100
roadmaps	Rome and New York
warm up time	60 seconds
roadmap size	2000m × 2000m
number of vehicles	500 and 1000
number of collided vehicles	1
warning message size	18KB [12]
beacon message size	512B
warning messages priority	AC3
beacon priority	AC1
interval between messages	1 second
RSU deployment policy	Uniform Mesh [13]
MAC/PHY	802.11p
radio propagation model	RAV [14]
mobility model	Krauss [15]
channel bandwidth	6Mbps
max. transmission range	400m

All simulation results consist of an average of over 100 runs with different scenarios, densities and fitness functions. Each simulation consist on vehicles circulating during 600 seconds. We simulate a car accident taking place at 60 seconds. We use the first 60 seconds as a warm up period to achieve a stable state. During this time, vehicles follow random routes. At the time of the accident we capture the current estimated location of all the vehicles and their target location. Then, we apply our proposed approaches to calculate the new vehicle routes, and to perform a comparison analysis. Additionally, we consider a non-static start and end position for the emergence vehicle, since an ambulance does not have to be always at the same place and the accident can occur in any location. Table I shows the parameters used for the simulations.

Table II shows the main features of each map for the cities under study. Specifically, we obtained the number of streets, the number of junctions, and the number of lanes per street. We also added a column labeled as *SJ Ratio*, which represents the result of dividing the number of streets between the number of junctions, thereby indicating the roadmap complexity. As shown, the first city (New York) presents an SJ ratio of 0.5130,

TABLE II  
MAP FEATURES

Map	Streets	Junctions	lanes/street	SJ Ratio
New York	257	500	1.573	0.5140
Rome	1655	1193	1.0590	1.3873

TABLE III  
PARAMETERS USED FOR THE EVOLUTION STRATEGY

Parameter	Value
number of simulations	100
population number	5
number of descendants	10
number of generations	20
fitness functions	Equations 5, 6, and 7
mutation	Uncorrelated Mutation with n Step Sizes
recombination	Local Discrete
parents selection	Randomly
survivors selection	( $\mu + \lambda$ )

which indicates that it has a simple topology, whereas Rome presents a greater SJ value, which indicates a more complex topology.

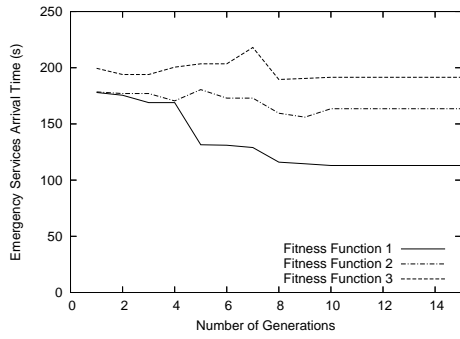
In order to obtain the real-time traffic density to provide this information to the system, we apply the Density Estimation Function presented in [8], which uses the number of beacons received by each Road side Unit (RSU) (parameter  $x$ ), and the *SJ Ratio* (parameter  $y$ ) to accurately estimate the vehicle density of a given area.

#### IV. SIMULATION RESULTS

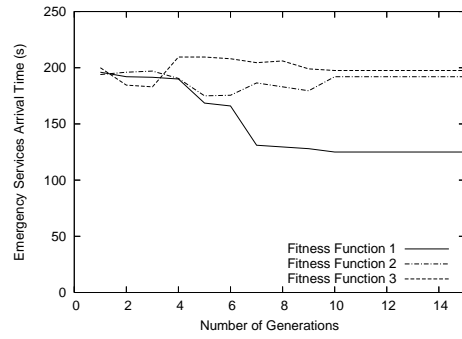
In this Section we present the simulation results of our four proposed approaches. First, we show the results obtained using the Evolution Strategy System. Our goal is to study the number of required generations to obtain the function convergence values. Then, we compare the Dijkstra, the Density-Based Dijkstra, and the Evolution Strategy Systems, demonstrating that by applying an evolution strategy we are able to obtain better results. Finally, we present a comparison between the Evolution Strategy and Density-Based Evolution Strategy Systems, with the aim of proving that adding traffic density information allows the evolution strategy to obtain better results using a smaller number of generations.

##### A. Evolution Strategy

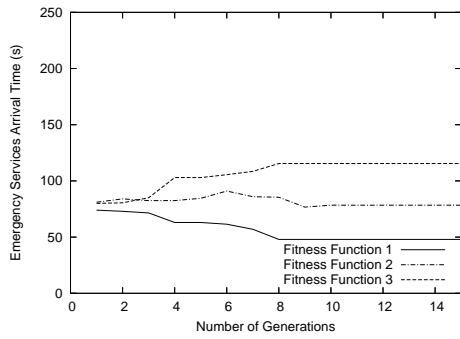
In this Subsection, we show the obtained results using our proposed Evolution Strategy and we analyze the number of generations required to obtain the function convergence value. Table III shows the parameters used for the Evolution Strategy used. Figures 6 and 7 present the obtained results. As expected, the system obtains the best emergency services arrival times when applying Equation 5 as a fitness function (i.e., the fitness function that gives doubled importance to the emergency services arrival time) in all simulated scenarios. Also, we can observe that, when using Equation 7 as a fitness function, our system is able to reduce the travel times of the rest of vehicles, although this solution slightly increases the emergency services arrival times. On the other hand, results indicate that when applying Equation 6 as a fitness function we are able to reduce both the emergency services arrival



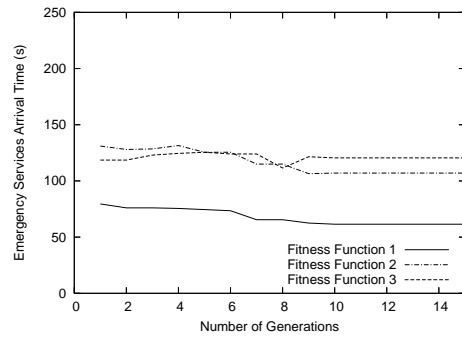
(a)



(b)

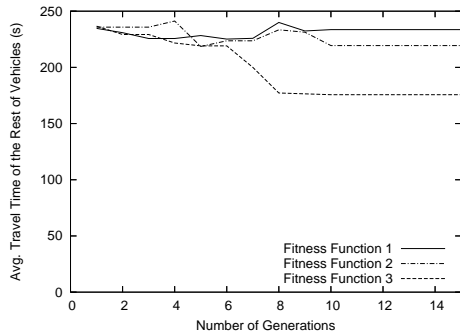


(c)

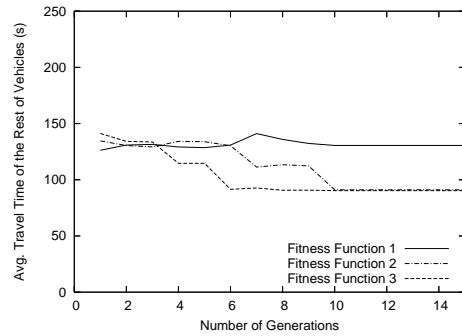


(d)

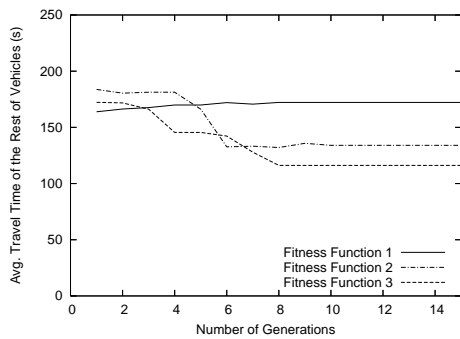
Fig. 6. Emergency services arrival times, using the Evolution Strategy in the scenarios of: Rome (Italy) (a) 125 vehicles/ $km^2$ , and (b) 250 vehicles/ $km^2$ , and New York (USA) (c) 125 vehicles/ $km^2$ , and (d) 250 vehicles/ $km^2$ .



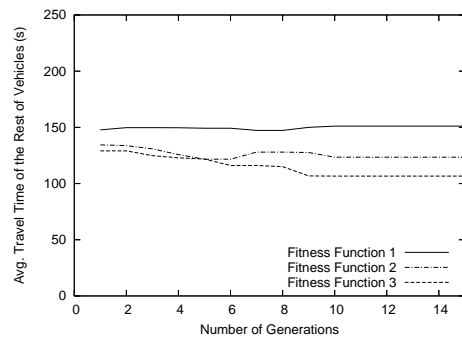
(a)



(b)



(c)



(d)

Fig. 7. Mean travel times of the rest of the vehicles, using the Evolution Strategy in the scenarios of: Rome (Italy) (a) 125 vehicles/ $km^2$ , and (b) 250 vehicles/ $km^2$ , and New York (USA) (c) 125 vehicles/ $km^2$ , and (d) 250 vehicles/ $km^2$ .

time and the rest of vehicles travel time, but they are not reduced in the same degree as when using the other two fitness functions. Since the main goal of our proposal is to reduce the emergency services arrival time as much as possible, we select Equation 5 as the best fitness function, which is able to minimize this time. In addition, as shown in Figure 6, by using this configuration the system obtains the function convergence values in 10 generations or less.

### B. Dijkstra, Density-Based Dijkstra, and Evolution Strategy Comparison

For the purpose of knowing which one is the best system, we analyze the results obtained with the configuration proposed in the previous Subsection (i. e., 10 number of generations, and Equation 5 as the fitness function), since they were the best parameter values when using the Evolution Strategy.

Table IV shows the average travel times of the emergency vehicles and the rest of vehicles (in seconds), when varying the roadmap scenario, the vehicle density, and the traffic re-routing approach. As shown, when using the Density-Based Dijkstra system we improve in all scenarios compared with the application of pure Dijkstra. In particular, we reduce emergency services travel times by 13.92% on average (i. e., 19.33% in Rome, and 8.51% in New York). Also, we reduce the rest of vehicles travel time by an average of 8.53% (i.e., 5.45% in Rome, and 11.61% in New York).

On the other hand, the Evolution Strategy significantly reduces the emergency services arrival time, although it increases the travel time for the rest of the vehicles. Specifically, this system reduces emergency services travel times by an average of 34.12% (40.36% in Rome, and 27.88% in New York). However, it increases the travel time for the rest of the vehicles by 10.03% on average (10.53% in Rome, and 9.53% in New York). Although this intelligent system increases the travel time for the rest of the vehicles (a maximum of 16.27%), it can significantly reduce the emergency services travel time (a minimum of 26.35%).

### C. Comparison Between Evolution Strategy and Density-Based Evolution Strategy Systems

In this Subsection we compare our two proposed intelligent algorithms (i.e., Evolution Strategy and Density-Based Evolution Strategy). Simulations were performed using the parameters showed in Table III, but, in order to simplify the comparison, we only simulate our systems using Equation 5 as the fitness function. As shown in Figure 8, the results obtained when applying the Density-Based Evolution Strategy system are better than when using the Evolution Strategy. Also, we can observe that the Density-Based approach allows obtaining smaller emergency services arrival times with fewer generations, since we consider traffic density when initializing the population.

In addition, we compare the Density-Based Evolution Strategy system results with those obtained when using the Dijkstra system. As shown in Table V, we reduce the emergency services travel times by 52.37% on average (53.58% in Rome,

TABLE V  
SIMULATION RESULTS

Scenario	Vehicles/ $km^2$	Dijkstra		Density-Based Evolution Strategy	
		Vehicles Avg. t.	Emgcy. Serv.	Vehicles Avg. t.	Emgcy. Serv.
Rome	125	222.91	190	249.36	<b>89</b>
	250	112.27	209	126.51	<b>93</b>
New York	125	151.43	68	171.06	<b>31</b>
	250	143.46	83.5	157.44	<b>43.5</b>

and 51.16% in New York). However, this system increases the rest of vehicles travel time by 11.82% on average (12.27% in Rome, and 11.36% in New York). Although this intelligent system increases the travel time for the rest of vehicles (a maximum of 12.96%), it can significantly reduce the emergency services arrival time (a minimum of 47.9%).

Since one of the most important goals of our approach is reducing the emergency services travel times, the Density-Based Evolution Strategy system is the best one among all the proposed solutions. Once again, we demonstrate that traffic density is a key factor in vehicular scenarios.

## V. CONCLUSIONS

In this paper we propose four different approaches to reduce the emergency services arrival time when an accident occurs, trying to avoid traffic jams that could result from this particular situation. Specifically, we present two systems based on Evolution Strategies which obtain a sub-optimal solution in a reduced time.

Results show that the Density-based Evolution Strategy performs better than the rest. In particular, this approach reduces the emergency services arrival time by a minimum of 47.9%, increasing the travel time of the rest of vehicles by just 12.96% in the worst case when compared to the rest of our proposed algorithms that obtain an improvement of 5.99% (Density-Based Dijkstra), and 26.35% (Evolution Strategy), respectively.

## ACKNOWLEDGMENTS

This work was partially supported by the *Ministerio de Ciencia e Innovación*, Spain, under Grant TIN2011-27543-C03-01, as well as by the *Fundación Universitaria Antonio Gargallo*, the *Obra Social de Ibercaja*, the *Government of Aragón* and the *European Social Fund (T91 Research Group)*.

## REFERENCES

- [1] M. Fogue, P. Garrido, F. J. Martinez, J.-C. Cano, C. T. Calafate, and P. Manzoni, "A novel approach for traffic accidents sanitary resource allocation based on multi-objective genetic algorithms," *Expert Systems with Applications*, vol. 40, no. 1, pp. 323–336, 2013.
- [2] G. Greenwood, C. Lang, and S. Hurley, "Scheduling tasks in real-time systems using evolutionary strategies," in *Proceedings of the Third Workshop on Parallel and Distributed Real-Time Systems*, Apr. 1995, pp. 195–196.
- [3] K. Ohara, Y. Nojima, and H. Ishibuchi, "Comparison between Centralized Global Optimization and Distributed Local Optimization for Traffic Jam Avoidance," in *Proc. of 2006 Genetic and Evolutionary Computation Conference Late Breaking Papers*, Jul. 2006.

TABLE IV  
SIMULATION RESULTS

Scenario	Vehicles/ $km^2$	Dijkstra		Density-Based Dijkstra		Evolution Strategy	
		Vehicles Avg. t.	Emgcy. Serv.	Vehicles Avg. t.	Emgcy. Serv.	Vehicles Avg. t.	Emgcy. Serv.
Rome	125	222.91	190	205.09	162	233.59	<b>113</b>
	250	112.27	209	109.01	159	130.54	<b>125</b>
New York	125	151.43	68	134.04	60.5	172.19	<b>48</b>
	250	143.46	83.5	126.61	78.5	151.14	<b>61.5</b>

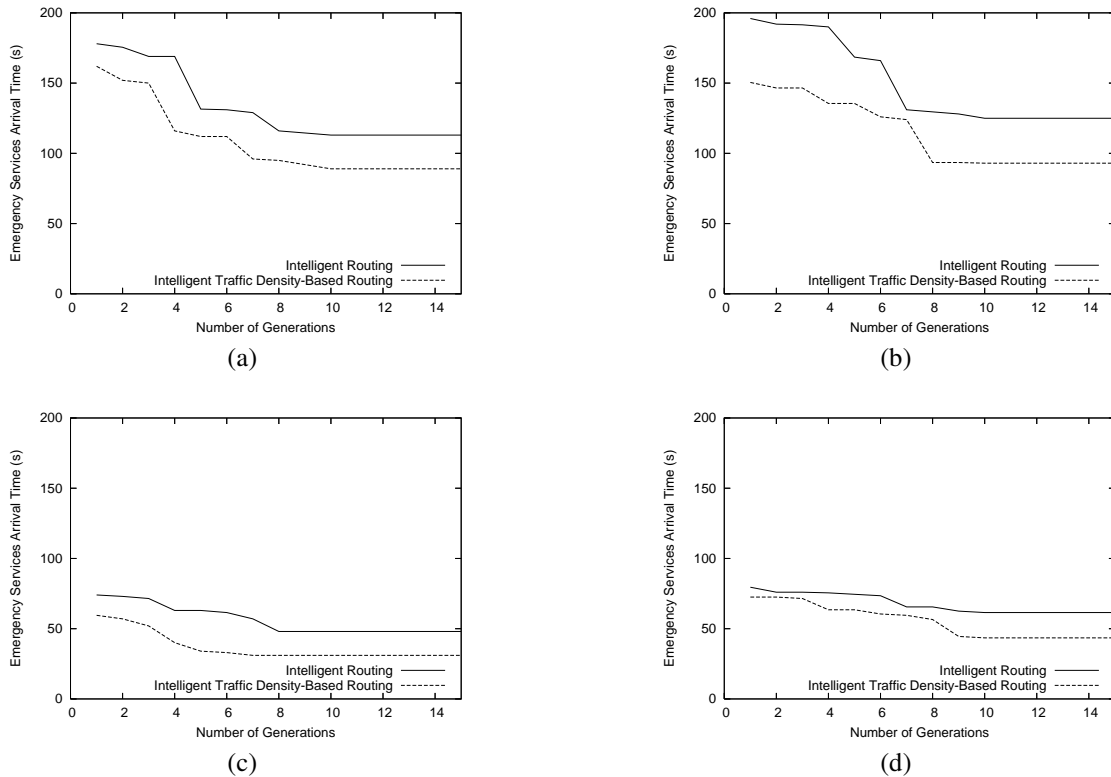


Fig. 8. Evolution Strategy and Density-Based Evolution Strategy systems emergency services arrival times on average after 100 simulations in the scenarios of: Rome (Italy) (a) 125 vehicles/ $km^2$ , and (b) 250 vehicles/ $km^2$ , and New York (USA) (c) 125 vehicles/ $km^2$ , and (d) 250 vehicles/ $km^2$ .

- [4] J. Sanchez-Medina, M. Galan-Moreno, and E. Rubio-Royo, "Traffic Signal Optimization 'La Almozara' District in Saragossa Under Congestion Conditions, Using Genetic Algorithms, Traffic Microsimulation, and Cluster Computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 1, pp. 132–141, Mar. 2010.
- [5] H. Dezani, L. Gomes, F. Damiani, and N. Marranghello, "Controlling traffic jams on urban roads modeled in Coloured Petri net using Genetic Algorithm," in *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, Oct. 2012, pp. 3043–3048.
- [6] M. Fogue, P. Garrido, F. J. Martinez, J.-C. Cano, C. T. Calafate, and P. Manzoni, "Analysis of the most representative factors affecting Warning Message Dissemination in VANETs under real roadmaps," in *19th annual meeting of the IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, Singapore, July 2011, pp. 197–204.
- [7] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.
- [8] J. Barrachina, M. Fogue, P. Garrido, F. J. Martinez, J.-C. Cano, C. T. Calafate, and P. Manzoni, "I-VDE: A Novel Approach to Estimate Vehicular Density by Using Vehicular Networks," in *The 12th International Conference on Ad Hoc Networks and Wireless (ADHOC-NOW 2013)*, July 2013.
- [9] D. Mester and O. Bräysy, "Active guided evolution strategies for large-scale vehicle routing problems with time windows," *Computers & Operations Research*, vol. 32, no. 6, pp. 1593–1614, 2005.
- [10] D. Krajzewicz and C. Rossel, "Simulation of Urban MObility (SUMO)," Centre for Applied Informatics (ZAIK) and the Institute of Transport Research at the German Aerospace Centre, 2007, available at <http://sumo.sourceforge.net/index.shtml>.
- [11] OpenStreetMap, "Collaborative project to create a free editable map of the world," 2012, available at <http://www.openstreetmap.org>.
- [12] J. Barrachina, P. Garrido, M. Fogue, F. J. Martinez, J.-C. Cano, C. T. Calafate, and P. Manzoni, "VEACON: A Vehicular Accident Ontology designed to improve safety on the roads," *Journal of Network and Computer Applications*, vol. 35, no. 6, pp. 1891–1900, Nov. 2012.
- [13] —, "D-RSU: A Density-Based Approach for Road Side Unit Deployment in Urban Scenarios," in *International Workshop on IPv6-based Vehicular Networks (Vehi6), collocated with the 2012 IEEE Intelligent Vehicles Symposium*, Jun. 2012, pp. 1–6.
- [14] F. J. Martinez, M. Fogue, C. K. Toh, J.-C. Cano, C. T. Calafate, and P. Manzoni, "Computer simulations of VANETs using realistic city topologies," *Wireless Personal Communications*, vol. 69, no. 2, pp. 639–663, 2013.
- [15] S. Krauss, P. Wagner, and C. Gawron, "Metastable states in a microscopic model of traffic flow," *Physical Review E*, vol. 55, no. 5, pp. 5597–5602, 1997.