

Document downloaded from:

<http://hdl.handle.net/10251/38084>

This paper must be cited as:

Barrachina, J.; Garrido, P.; Fogue, M.; Martínez, F.J.; Cano Escribá, J.C.; Tavares De Araujo Cesariny Calafate, C.M.; Manzoni, P. (2014). Reducing emergence services arrival time by using vehicular communications and Evolution Strategies. *Expert Systems with Applications*. 41(4):1206-1217. doi:10.1016/j.eswa.2013.08.004.



The final publication is available at

<http://dx.doi.org/10.1016/j.eswa.2013.08.004>

Copyright Elsevier

# Reducing Emergence Services arrival time by using Vehicular Communications and Evolution Strategies

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 Politècnica de València, Spain*

*Email: {jucano, calafate, pmanzoni}@disca.upv.es*

---

## Abstract

Nowadays, traffic jams in urban areas have become a problem that keeps growing every year since the number of vehicles in our cities is continuously increasing. One of the most common causes producing traffic jams are vehicle accidents. Moreover, the arrival time of the emergency services could be raised due to traffic congestion. Intelligent Transportation Systems (ITS) have a key role in order to reduce or mitigate this problem. 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, so reducing the emergency services arrival time, and avoiding traffic jams when an accident occurs. Specifically, we propose two approaches based on Dijkstra algorithm, and two approaches based on Evolution Strategies. Notice that, when an accident occurs, time is a critical issue, and strategies here proposed contribute to optimal solution within a short period of time.

*Keywords:* Vehicular Networks; Traffic Accidents Assistance; Evolution Strategies;

---

## 1. 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 injured take place during the time elapsed between the accident occurrence and the arrival of the medical assistance. The so called ‘Golden Hour’ (Fogue et al., 2013) 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.

Additionally, urban traffic congestion affects most cities around the world. This scenario is getting even worse since the number of vehicles circulating in our cities grows every year. Vehicle accidents are one of the most common causes generating traffic jams in urban scenarios, which yield a higher cost of fuel, increasing air pollution.

Intelligent Transportation Systems (ITS) are among those newly introduced technologies that promise a cure-all remedy to the ever increasing traffic congestion problem (Jawad and Ozbay, 2005). In the near future, ITS will help the city traffic to be safer and more comfortable, redistributing traffic to avoid traffic jams (Ma et al., 2012), communicating real-time information when an accident occurs (Barrachina et al., 2012b), and using intelligent systems for parking search (Lu et al., 2009).

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 (Greenwood et al., 1995), since they provide an optimal solution without checking all the possible solutions, so 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.

There are several works where intelligent systems are used to avoid traffic jams (e.g., Ohara et al. (2006), Sanchez-Medina et al. (2010), and Dezani et al. (2012)). However, they do not focus on reducing the rescue time of the emergency services, or exploit the advantages of using vehicular communication capabilities. Additionally, in all these works, authors only consider a specific scenario for simulations to assess their proposal, which might lead to unrepresentative results and wrong conclusions.

In this paper, we propose four different approaches to minimize the emergency services arrival time when an accident occurs in urban scenarios, also trying to avoid traffic jams scenarios. 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.

This paper is organized as follows: In Section 2 we present our four different

Table 1: Features of our proposals

	Dijkstra	Density-Based Dijkstra	Evolution Strategy	Density-Based Evolution Strategy
Deterministic	✓	✓	✗	✗
Nondeterministic	✗	✗	✓	✓
Considering traffic density	✗	✓	✗	✓

re-routing systems (i.e., Dijkstra, Density-Based Dijkstra, Evolution Strategy, and Density-Based Evolution Strategy). Section 3 introduces the simulation environment used to assess our proposed schemes. Section 4 shows the obtained results, and Section 5 reviews the related work regarding intelligent systems used to avoid traffic jams and minimize vehicle travel times. Finally, Section 6 concludes this paper.

## 2. 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. Specifically, they are: (i) Dijkstra, (ii) Density-Based Dijkstra, (iii) Evolution Strategy, and (iv) Density-Based Evolution Strategy.

Table 1 presents the main features of these proposed approaches. As shown, the first two proposed approaches are simple and deterministic. The first one accounts for the number of lanes of each street to find the solution, and the second system additionally takes into account the traffic density. The other two proposed approaches are implemented using evolution strategies, and additionally, our last mechanism uses a real-time traffic density estimation to get better solutions.

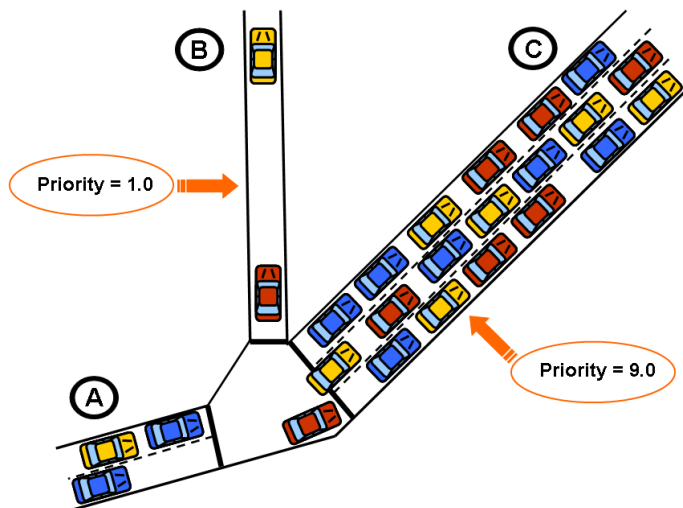


Figure 1: Example of a traffic jam when the street priority is given by the number of lanes.

### 2.1. Dijkstra

This system aims at obtaining the shortest route between two map positions by using the Dijkstra algorithm (Dijkstra, 1959), 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 an intersection.

In this system, 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). The main disadvantage of this system is noticeable when there is a high number of vehicles in a specific area, since it might produce traffic jams even in the widest streets. 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 prior-

ity 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.

## 2.2. 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 Communications. When control systems are notified, they apply the vehicular density estimation approach presented in Section 3. 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 :

$$N_{min} = 1$$

$$N_{max} = 10$$

(1)

- *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 ( $L_x$ ).

$$S_x = (N_{max} - N_x + 1) \cdot L_x \quad (2)$$

- *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 positions (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 more priority than the rest of vehicles, regardless of the street they are circulating in.
- *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).

$$S_{x_e} = L_x \quad (3)$$

- *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.

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

This system requires from the estimated traffic density. In Section 3 we present a system which needs to receive beacons during 30 seconds to estimate the traffic density. To reduce this 30 seconds period, control units could continuously execute the aforementioned estimation system in order to know immediately the traffic density estimation, assuming an error of non-real-time estimation with a maximum threshold of 30 seconds. Using this approximation, our system would only require calculating the emergency services routes.

### 2.3. Evolution Strategy

Evolutionary algorithms are based on Darwinian theories of evolution to explain the origin of species (Eiben and Smith, 2003). Natural selection favors

---

**Algorithm 1: Evolutionary Algorithm Scheme**

---

```
BEGIN
  Initialize Population
  Evaluation
  REPEAT UNTIL ( Finish Condition ) DO
    Parents Selection
    Recombination
    Mutation
    Evaluation
    Survivor Selection
  END LOOP
END
```

---

those individuals competing for resources in a more effective way, i.e., better adapted to the environmental conditions. Although there are different variants of evolutionary algorithms, such as genetic algorithms, evolution strategies, evolutionary programming, and genetic programming, all of them have the same essence: an individual population generates descendants, and the best individuals are selected to obtain the next generation. All evolutionary algorithms have the same methodology, presented in Algorithm 1.

Evolution strategies are a variant of evolutionary algorithms with the following features:

- They are typically used for conditions parameter optimization.
- There is a strong emphasis on mutation for creating offspring.
- Mutation is implemented by adding some random noise drawn from a Gaussian distribution.
- Mutation parameters are changed during a run of the algorithm, achieving faster results.

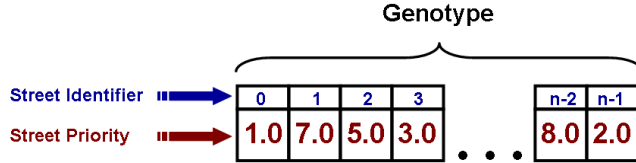


Figure 2: Example of a genotype for street priorities.

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. Evolution Strategies are typically used to solve optimization problems of continuous variables. As in the previous proposed approaches, this scheme applies the Dijkstra algorithm for each emergency vehicle in order to calculate the emergency services routes. In this case we do not take traffic density into account, but we penalize the streets selected for the emergency services vehicles. Then, we calculate new routes for vehicles using a priority-based Dijkstra algorithm (with the same aims of the previously proposed system).

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

### 2.3.1. Definition of Variables

An individual, i.e., a potential solution of our system, encodes a possible solution into a chromosome based structure (genotype) (Mester and Bräysy, 2005). 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.

### 2.3.2. 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 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. Next, we compare these functions to determine which one provides better results when simulating the testbed.

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

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

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

### 2.3.3. 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.

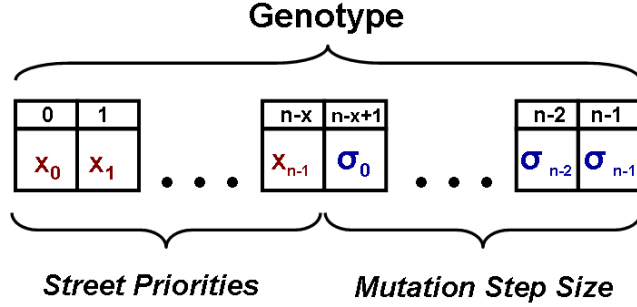


Figure 3: Example of genotype formed by street priorities and mutation step sizes.

$$\sigma'_i = \sigma \cdot e^{\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)},$$

$$x'_i = x_i + \sigma'_i \cdot N_i(0,1)$$

*where :*

$$\tau' \propto \frac{1}{\sqrt{2n}}$$

$$\tau \propto \frac{1}{\sqrt{2\sqrt{n}}}$$
(8)

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

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.,  $\sigma' < \varepsilon_0 \Rightarrow \sigma' = \varepsilon_0$ .

#### 2.3.4. 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.

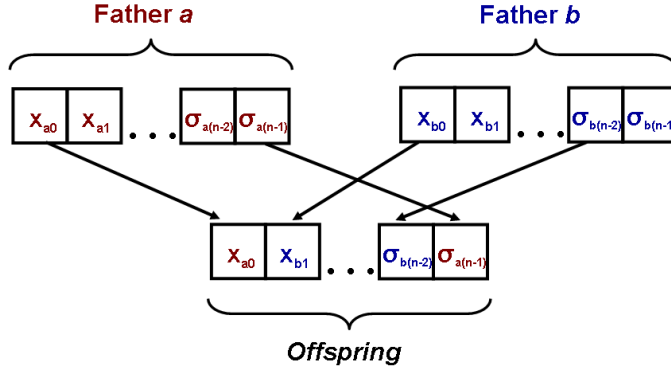


Figure 4: Example of local discrete recombination.

Furthermore, two parents can be used, randomly obtained from the population of  $\mu$  individuals, for each component ( $i \in \{1 \dots n\}$ ) 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.

### 2.3.5. 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.

### 2.3.6. 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.

#### 2.4. 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. As shown, while the

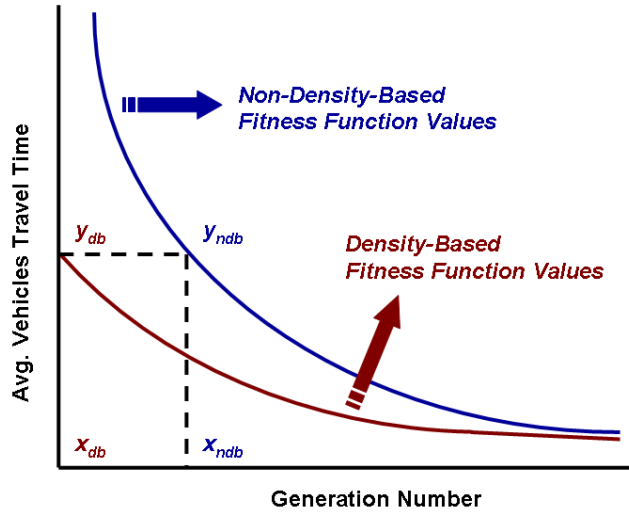


Figure 5: Example of fitness function values using both proposed intelligent systems (i.e., Evolution Strategy and Density-Based Evolution Strategy).

non-density-based system would have created  $x_{db}$  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.

### 3. Simulation Environment

Traffic simulation is known to be a very complex issue. One of the main reasons is due to the fact that traffic simulators must model the discrete dynamics that arise from the interaction among individual vehicles (Benjaafar et al., 1997). The Simulation of Urban MObility (SUMO) is an open source, microscopic, continuous-space traffic simulator designed to handle large road networks, and it is mainly developed by employees of the Institute of Transportation Systems at the German Aerospace Center<sup>1</sup> (Krajzewicz and Rossel, 2007).

<sup>1</sup><http://www.dlr.de/fs/en/desktopdefault.aspx>



Table 2: Attributes of SUMO Streets

Attribute	Description
id	The unique id of the street
from	The id of the starting junction
to	The id of the final junction
priority	Street weight regarding the rest of the streets

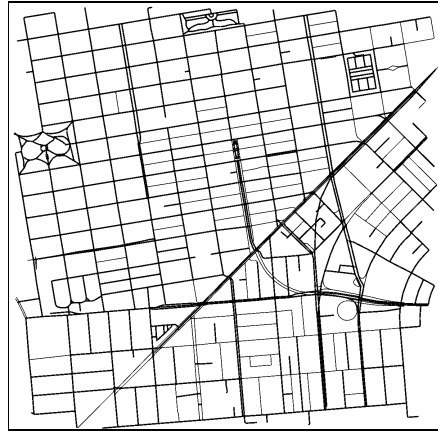
The SUMO mobility generator supports several mobility models, such as the Krauss mobility model (Krauss et al., 1997). In addition, SUMO allows customizing a wide variety of parameters including the initial and final position of the vehicles, the type of vehicles, the maximum speed of each street, or the street’s priority. Table 2 shows the SUMO street attributes that we use in our system. Moreover, each SUMO lane has an attribute indicating the street to which it belongs. This allows us to obtain the number of lanes at every street. We use the attributes *from* and *to* in order to determine the heading of the street, the attribute *id* to link lanes with streets, and the attribute *priority* to implement our proposed schemes.

To increase the level of realism of our simulations, we use real scenarios consisting of downtown areas from the cities of Rome (Italy), San Francisco (USA), and New York (USA) imported directly from OpenStreetMap (OpenStreetMap, 2012). OpenStreetMap is a project which aims at creating and providing free geographic data, such as street and road maps. According the SJ Ratio (i.e., a variable calculated from dividing the number of the streets and the number of junctions), these cities are examples of the roadmaps with the highest SJ Ratio, an intermediate SJ Ratio, and the lowest SJ Ratio, respectively (see Figure 6). So, we assess our proposal under different and representative roadmap profiles.

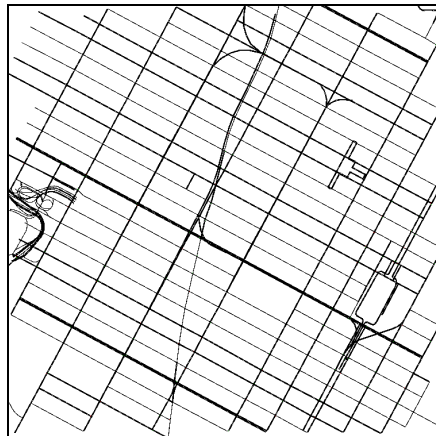
All simulation results consist of an average of over 100 runs with different scenarios, densities and fitness functions. Each simulation consist on vehicles circulating during 600 seconds. We simulate a car accident taking place at 60 seconds. We use the first 60 seconds as a warm up period to achieve a stable state. During this time, vehicles follow random routes. At the time of the



(a)



(b)



(c)

Figure 6: Scenarios used in our simulations. Fragments of the cities of: (a) Rome (Italy), (b) San Francisco (USA), and (c) New York (USA).

Table 3: Parameters used for the simulations

Parameter	Value
number of simulations	100
roadmaps	Rome, San Francisco, and New York
warm up time	60 seconds
roadmap size	$2000m \times 2000m$
number of vehicles	500 and 1000
number of collided vehicles	1
warning message size	18KB (Barrachina et al., 2012b)
beacon message size	512B
warning messages priority	AC3
beacon priority	AC1
interval between messages	1 second
RSU deployment policy	Uniform Mesh (Barrachina et al., 2012a)
MAC/PHY	802.11p
radio propagation model	RAV (Martinez et al., 2012)
mobility model	Krauss (Krauss et al., 1997)
channel bandwidth	6Mbps
max. transmission range	400m

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 3 shows the parameters used for the simulations.

Table 4 shows the main features of each map for the cities under study. Specifically, we obtained the number of streets, the number of junctions, the average distance of segments, 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, which indicates that it has a simple topology, whereas the last cities in the table present 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 (Barrachina

Table 4: Map Features

Map	Streets	Junctions	avg. distance of segment (m.)	lanes/street	SJ Ratio
New York	257	500	45.8853	1.0590	0.5140
Minnesota	459	591	102.0652	1.0144	0.7766
Madrid	628	715	83.0820	1.2696	0.8783
San Francisco	725	818	72.7065	1.1749	0.8863
Amsterdam	1494	1449	44.8973	1.1145	1.0311
Sydney	872	814	72.1813	1.2014	1.0713
Liverpool	1758	1502	49.9620	1.2295	1.1704
Valencia	2829	2233	33.3653	1.0854	1.2669
Rome	1655	1193	45.8853	1.0590	1.3873

Table 5: Coefficients of our Proposed Density Estimation Equation

Coeff.	Value
a	2.30375E+02
b	1.90696E+01
c	-4.29461E+02
d	3.18809E+01
f	1.87953E+02
g	-6.81258E+01

et al., 2013). The proposed function (see Equation 9 and the coefficients showed in Table 5) needs the number of beacons received by each RSU (parameter  $x$ ), and the *SJ Ratio* (parameter  $y$ ) to accurately estimate the vehicle density of a given area.

$$f(x, y) = a + b \cdot \ln(x) + \frac{c}{y} + d \cdot \ln(x)^2 + \frac{f}{y^2} + \frac{g \cdot \ln(x)}{y} \quad (9)$$

#### 4. 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

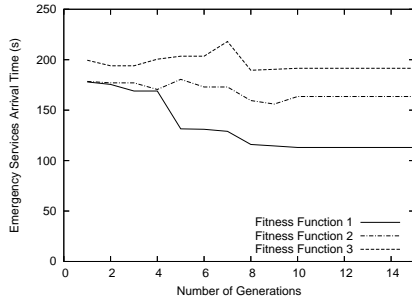
Table 6: Parameters used for the Evolution Strategy

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

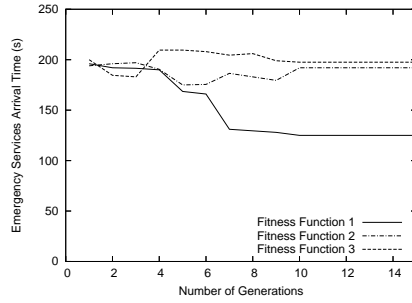
an evolution strategy we are able to obtain better results. Later, we present a comparison between the Evolution Strategy and Density-Based Evolution Strategy Systems, with the aim of proving that adding traffic density information allows the evolution strategy to obtain better results using a smaller number of generations. Finally, we study the impact of reducing the population size and the number of descendants on the obtained results; our goal is to reduce the system runtime, while reducing the needed time for emergency services to arrive.

#### 4.1. Evolution Strategy

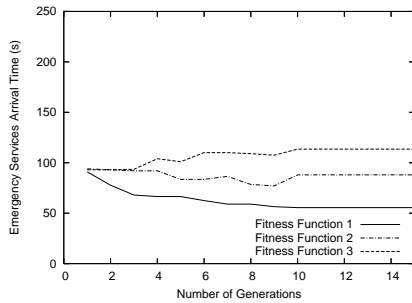
In this Subsection, we show the obtained results using our proposed Evolution Strategy and we analyze the number of generations required to obtain the function convergence value. Table 6 shows the parameters used for the Evolution Strategy used. Figures 7 and 8 present the obtained results. As expected, 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



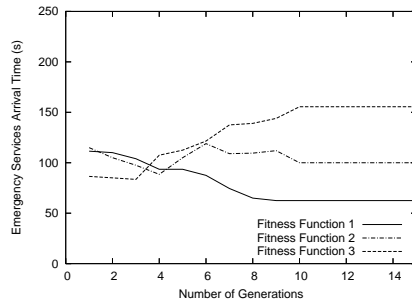
(a)



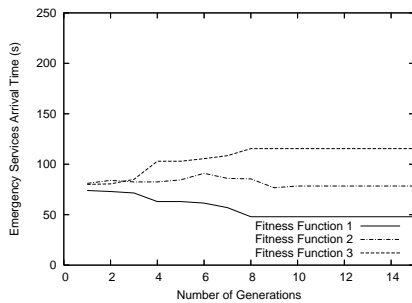
(b)



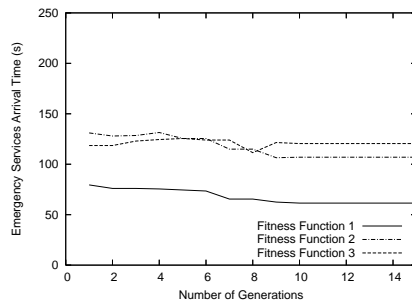
(c)



(d)

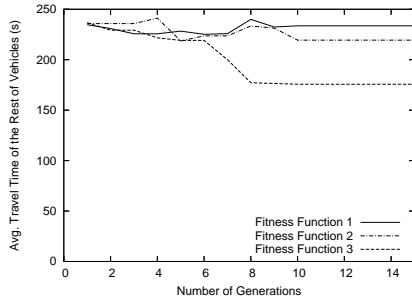


(e)

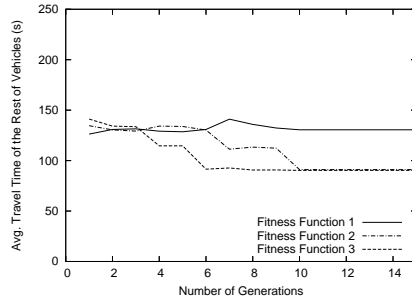


(f)

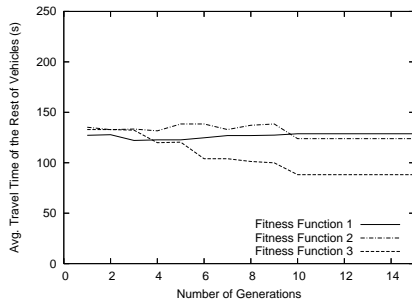
Figure 7: Emergency services arrival times, using the Evolution Strategy in the scenarios of: Rome (Italy) (a) 125 vehicles/ $km^2$ , and (b) 250 vehicles/ $km^2$ , San Francisco (USA) (c) 125 vehicles/ $km^2$ , and (d) 250 vehicles/ $km^2$ , and New York (USA) (e) 125 vehicles/ $km^2$ , and (f) 250 vehicles/ $km^2$ .



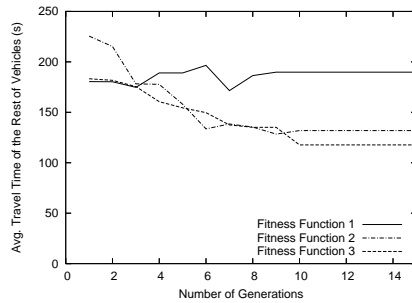
(a)



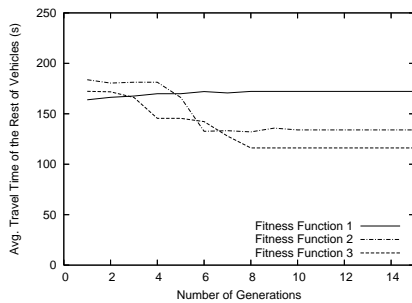
(b)



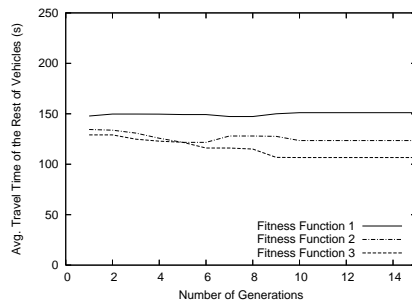
(c)



(d)



(e)



(f)

Figure 8: 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/ $km^2$ , San Francisco (USA) (c) 125 vehicles/ $km^2$ , and (d) 250 vehicles/ $km^2$ , and New York (USA) (e) 125 vehicles/ $km^2$ , and (f) 250 vehicles/ $km^2$ .

Table 7: 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	113
	250	12.27	209	109.01	159	130.54	125
San Francisco	125	112.02	92.5	106.86	82.5	128.74	55.5
	250	148.08	126	145.14	82.5	189.81	62.5
New York	125	151.43	68	134.04	60.5	172.19	48
	250	143.46	83.5	126.61	78.5	151.14	61.5

are able to reduce both the emergency services arrival 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 7, by using this configuration the system obtains the function convergence values in 10 generations or less.

#### 4.2. 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 7 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 16.84% on average (i. e., 19.33% in Rome, 22.67% in San Francisco, and 8.51% in New York). Also, we reduce the rest of vehicles travel time by an average of 6.79% (i.e., 5.45% in Rome, 3.3% in San Francisco, 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



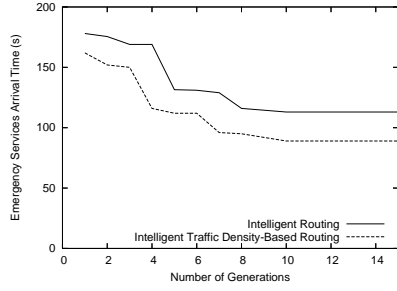
the vehicles. Specifically, this system reduces emergency services travel times by an average of 37.81% (40.36% in Rome, 45.2% in San Francisco, and 27.88% in New York). However, it increases the travel time for the rest of the vehicles by 13.87% on average (10.53% in Rome, 21.55% in San Francisco, and 9.53% in New York). Although this intelligent system increases the travel time for the rest of the vehicles (a maximum of 28.18%), it can significantly reduce the emergency services travel time (a minimum of 26.35%).

#### *4.3. Comparison Between Evolution Strategy and Density-Based Evolution Strategy Systems*

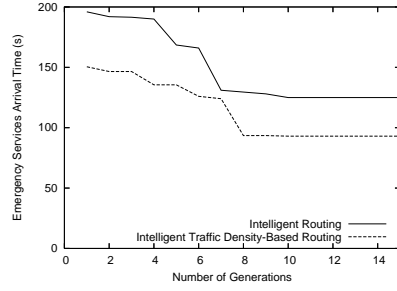
In this Subsection we compare our two proposed intelligent algorithms (i.e., Evolution Strategy and Density-Based Evolution Strategy). Simulations were performed using the parameters showed in Table 6, but, in order to simplify the comparison, we only simulate our systems using Equation 5 as the fitness function. As shown in Figure 9, 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 8, we reduce the emergency services travel times by 54.33% on average (53.58% in Rome, 55.26% in San Francisco, and 51.16% in New York). However, this system increases the rest of vehicles travel time by 11.49% on average (12.27% in Rome, 10.85% in San Francisco, and 11.36% in New York). Although this intelligent system increases the travel time for the rest of vehicles (a maximum of 14.39%), it can significantly reduce the emergency services arrival time (a minimum of 47.9%).

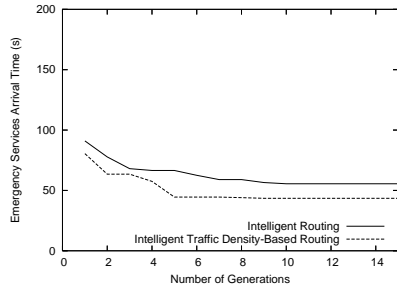
Since one of the most important goals of our approach is reducing the emergency services travel times, the Density-Based Evolution Strategy system is the best one among all the proposed solutions. Once again, we demonstrate that



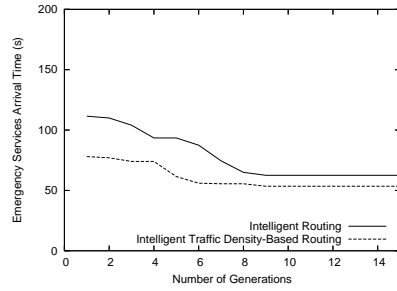
(a)



(b)



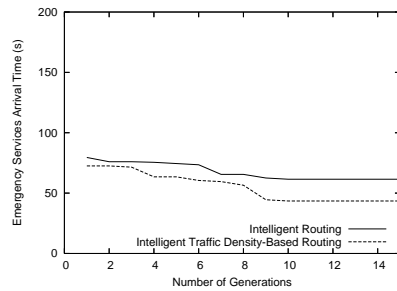
(c)



(d)



(e)



(f)

Figure 9: 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/ $km^2$ , San Francisco (USA) (c) 125 vehicles/ $km^2$ , and (d) 250 vehicles/ $km^2$ , and New York (USA) (e) 125 vehicles/ $km^2$ , and (f) 250 vehicles/ $km^2$ .

Table 8: Simulation Results

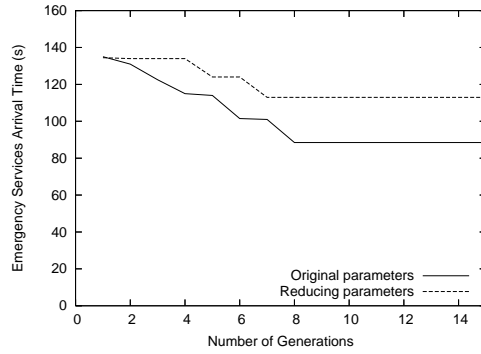
Scenario	Vehicles/ $km^2$	Dijkstra		Density-Based Evolution Strategy	
		Vehicles Avg. t.	Emgey. Serv.	Vehicles Avg. t.	Emgey. Serv.
Rome	125	222.91	190	249.36	<b>89</b>
	250	12.27	209	126.51	<b>93</b>
San Francisco	125	112.02	92.5	120.21	<b>43.5</b>
	250	148.08	126	169.38	<b>53.5</b>
New York	125	151.43	68	171.06	<b>31</b>
	250	143.46	83.5	157.44	<b>43.5</b>

traffic density is a key factor in vehicular scenarios.

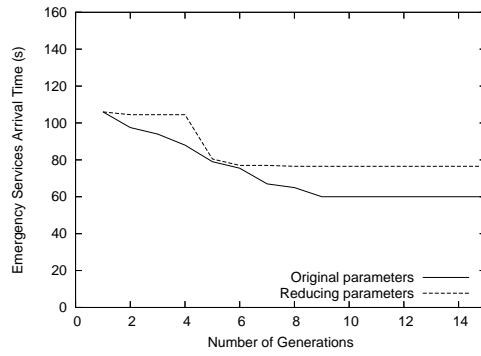
#### 4.4. Density-Based Evolution Strategy System Reducing Population and Number of Descendants

As stated above, the emergency services arrival time is a critical factor when an accidents occurs. Simulations performed by using Evolution Strategies require a high computational cost, increasing its application time. Hence, reducing the necessary simulations would decrease the system action time which directly affects the time required by emergency services to arrive at the accident location. For this reason, in this Subsection we assess our best proposed system's performance (i.e., the Density-Based Evolution Strategy) but reducing the population size and the number of descendants. Table 9 presents the parameters used in these simulations. As shown, we reduce the number of population individuals from 5 to 3, and the number of descendants from 10 to 5. Note that we only use the Density-Based Evolution Strategy system in conjunction with Equation 5, since we obtained the best results using this configuration.

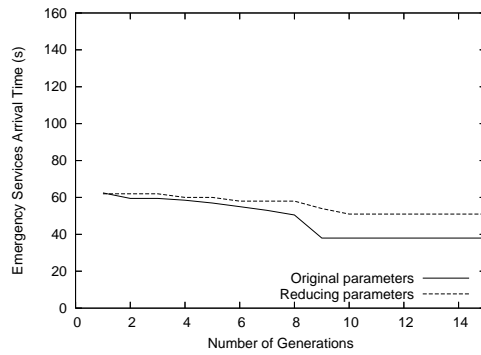
Figure 10 shows the obtained results. As can be seen, when reducing the number of population individuals and descendants, the emergency services arrival time increases: 27.68% in Rome, 27.5% in San Francisco, and 34.21% in New York. This occurs because we generate a smaller number of possible population individuals in each generation, thereby restricting the probability to achieve better individuals.



(a)



(b)



(c)

Figure 10: Emergency services arrival times simulating 250 vehicles/ $km^2$  varying the number of population individuals and descendants numbers in the scenarios of: (a) Rome (Italy), (b) San Francisco (USA), and (c) New York (USA).

Table 9: Parameters used for the Density-Based Evolution Strategy System

Parameter	Value
number of simulations	100
population number	5 and 3
number of descendants	10 and 5
number of generations	20
fitness function	Equation 5
mutation	Uncorrelated Mutation with n Step Sizes
recombination	Local Discrete
parents selection	Randomly
survivors selection	$(\mu + \lambda)$

## 5. Related Work

Genetic algorithms have been widely used in the field of dynamic traffic distribution. However, unlike our proposal, existing works do not focus on reducing the rescue time of the emergency services. In this section, we present some of the most relevant works related to our work.

Ohara et al. (2006) examined two routing methods to reduce the average vehicle travel time: one of them used a centralized system, and the other provided drivers some selection agents, but each driver had to select his route. Since the number of combinations of vehicles routes exponentially increases as the number of vehicles grows, authors employed a genetic algorithm to search for a near-optimal route combination for all vehicles.

Yoshikawa and Terai (2009) discussed a route selection algorithm, particularly focused on a hybrid technique which combines genetic algorithms with the Dijkstra algorithm (Dijkstra, 1959) to achieve high quality route guidance. They presented a solution similar to *The Traveling Salesman Problem* (Johnson and McGeoch, 1997). Specifically, their proposal is based on an individual vehicle which has an order of the passing points as genes. Authors estimated distances between nodes based on Manhattan street distances, although the topology of real urban areas are usually quite different from regular and simple Manhattan-style roads. In addition, they only took into account route distances for each individual vehicle without using vehicle density information to develop

their genetic algorithm.

More recently, Dezani et al. (2012) presented an application for real-time traffic lights control in congested urban traffic environments, taking as input the locations and routes of the vehicles in the involved areas. Authors used V2I communications to gather the location of vehicles in order to calculate traffic density. Additionally, they developed a genetic algorithm to solve traffic jams by controlling traffic lights. With respect to their proposed traffic density estimation system, we consider that it is not realistic since the vehicles circulating outside the infrastructure coverage area cannot communicate their position.

In all these previous works, authors validate their proposals just by considering a theoretical scenario. However, we consider that these simulations are not realistic, since real-world roads do not follow a general pattern, especially in urban scenarios.

Other authors proposed intelligent systems for traffic distribution using real scenarios to assess their proposal. Collins and Muntean (2008) presented a novel adaptive vehicle routing algorithm enabled by wireless vehicular networks. Their system was based on the client-server architecture, where clients are vehicles. They used a genetic algorithm to select the best route for each vehicle, using a fitness function taking into account road congestion, vehicle travel time, and fuel consumption. Specifically, they used four different kinds of simulations: (i) the shortest route is selected, but it does not vary during the travel, (ii) each vehicle drives towards its own destination according to the route management solution, but without adaptation during the travel, (iii) each vehicle drives towards its own destination according to the route management solution with dynamic adaptation during the travel, and, (iv) the hypothetical ‘ideal’ solution based on traffic saturation and able to dynamically re-route vehicles is selected. However, the only scenario used in their simulations was a fragment of the city of Boston (USA).

Sanchez-Medina et al. (2010) developed a model for traffic signal optimization based on the combination of three key techniques: (i) genetic algorithms for the optimization task, (ii) cellular-automata-based microsimulators for eval-

uating every possible solution for traffic-light programming times, and (iii) a Beowulf Cluster, which used a multiple-instruction-multiple-data (MIMD) price/performance ratio. They tested the genetic algorithm with four different fitness functions: (i) number of vehicles that reach their destination point easily, (ii) mean travel time, (iii) time of occupancy and state of occupancy, and (iv) global mean speed. Authors used a traffic model based on both Krauss (Krauss et al., 1997), and Schadschneider and Chowdhury (Schadschneider et al., 1999) mobility models. However, they focused their simulations on a specific scenario, i.e., ‘La Almozara’ district in Zaragoza.

To the best of our knowledge, although there are several works where intelligent systems are used to avoid traffic jams, none of them neither is focused on reducing the arrival time of the emergency services to the accident location, nor uses a street priority scheme to calculate vehicles routes. Additionally, in all previous studies, authors only consider a specific scenario for simulations in order 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 nonrepresentative and inaccurate results). We consider that simulating different (and realistic) topologies is necessary, since the roadmap topology significantly affects the obtained results (Fogue et al., 2011).

## 6. 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. Moreover, we demonstrate that traffic density is a key factor to distribute traffic in an efficient manner.

Our proposals have been tested in three different scenarios with different topologies and traffic densities. Results show that the best solution is to combine

an Evolution Strategy with the traffic density information collected at the time of the accident, which is used to initialize the population. The improvement obtained with this approach reduces the emergency services arrival time by a minimum of 47.9%, increasing the travel time of the rest of vehicles only a 14.39% in the worst case, 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, and the Obra Social de Ibercaja, under Grant 2013/B010, and by the Government of Aragón and the European Social Fund (T91 Research Group).

### References

- Barrachina, J., Fogue, M., Garrido, P., Martinez, F. J., Cano, J.-C., Calafate, C. T., Manzoni, P., July 2013. 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).
- Barrachina, J., Garrido, P., Fogue, M., Martinez, F. J., Cano, J.-C., Calafate, C. T., Manzoni, P., Jun. 2012a. 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. pp. 1–6.
- Barrachina, J., Garrido, P., Fogue, M., Martinez, F. J., Cano, J.-C., Calafate, C. T., Manzoni, P., Nov. 2012b. VEACON: A Vehicular Accident Ontology designed to improve safety on the roads. *Journal of Network and Computer*



Applications 35 (6), 1891–1900.

URL <http://dx.doi.org/10.1016/j.jnca.2012.07.013>

Benjaafar, S., Dooley, K., Setyawan, W., 1997. Cellular automata for traffic flow modeling. Tech. rep., Center for Transportation Studies University of Minnesota.

Collins, K., Muntean, G.-M., Sept. 2008. An adaptive vehicle route management solution enabled by wireless vehicular networks. In: IEEE 68th Vehicular Technology Conference (VTC). pp. 1–5.

Dezani, H., Gomes, L., Damiani, F., Marranghello, N., Oct. 2012. 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. pp. 3043–3048.

Dijkstra, E. W., 1959. A note on two problems in connexion with graphs. *Numerische Mathematik* 1, 269–271.

Eiben, E., Smith, J. E., 2003. *Introduction to Evolutionary Computing*. Springer-Verlag.

Fogue, M., Garrido, P., Martinez, F. J., Cano, J.-C., Calafate, C. T., Manzoni, P., July 2011. 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, pp. 197–204.

Fogue, M., Garrido, P., Martinez, F. J., Cano, J.-C., Calafate, C. T., Manzoni, P., 2013. A novel approach for traffic accidents sanitary resource allocation based on multi-objective genetic algorithms. *Expert Systems with Applications* 40 (1), 323 – 336.

- Greenwood, G., Lang, C., Hurley, S., Apr. 1995. Scheduling tasks in real-time systems using evolutionary strategies. In: Proceedings of the Third Workshop on Parallel and Distributed Real-Time Systems. pp. 195–196.
- Jawad, D., Ozbay, K., Sept. 2005. The cost behavior of intelligent transportation systems over time. In: IEEE Intelligent Transportation Systems. pp. 1039–1044.
- Johnson, D., McGeoch, L., 1997. The travelling salesman problem: A case study in local optimization. In: Local Search in Combinatorial Optimization. Wiley. pp. 215–310.
- Krajzewicz, D., Rossel, C., 2007. Simulation of Urban MObility (SUMO). Centre for Applied Informatics (ZAIK) and the Institute of Transport Research at the German Aerospace Centre, available at <http://sumo.sourceforge.net/index.shtml>.
- Krauss, S., Wagner, P., Gawron, C., 1997. Metastable states in a microscopic model of traffic flow. *Physical Review E* 55 (5), 5597–5602.
- Lu, R., Lin, X., Zhu, H., Shen, X., Apr. 2009. Spark: A new vanet-based smart parking scheme for large parking lots. In: INFOCOM 2009, IEEE. pp. 1413–1421.
- Ma, J., Huang, X., Jiang, Y., Jun. 2012. Applicable research of urban road traffic congestion based on rough set theory and genetic algorithm. In: World Automation Congress (WAC). pp. 1–4.
- Martinez, F. J., Fogue, M., Toh, C. K., Cano, J.-C., Calafate, C. T., Manzoni, P., 2012. Computer simulations of VANETs using realistic city topologies. *Wireless Personal Communications*, 1–25.
- Mester, D., Bräysy, O., 2005. Active guided evolution strategies for large-scale vehicle routing problems with time windows. *Computers & Operations Research* 32 (6), 1593–1614.

- Ohara, K., Nojima, Y., Ishibuchi, H., Jul. 2006. 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. p. 6p CD ROM.
- OpenStreetMap, 2012. Collaborative project to create a free editable map of the world. Available at <http://www.openstreetmap.org>.
- Sanchez-Medina, J., Galan-Moreno, M., Rubio-Royo, E., Mar. 2010. 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* 11 (1), 132–141.
- Schadschneider, A., Chowdhury, D., Brockfeld, E., Klauck, K., Santen, L., Zittartz, J., 1999. *Traffic and Granular Flow: Social, Traffic, and Granular Dynamics*. Springer-Verlag.
- Yoshikawa, M., Terai, H., Apr. 2009. Car navigation system based on hybrid genetic algorithm. In: 2009 WRI World Congress on Computer Science and Information Engineering. Vol. 5. pp. 62–65.