A hybrid method for short-term traffic congestion forecasting using genetic algorithms and cross-entropy

Pedro Lopez-Garcia, Enrique Onieva, Eneko Osaba, Antonio D. Masegosa, and Asier Perallos

***Abstract*—This paper presents a method of optimizing the elements of a hierarchy of Fuzzy Rule-Based Systems (FRBSs). It is a hybridization of a Genetic Algorithm (GA) and the Cross Entropy (CE) method, named GACE. It is used to predict congestion in a 9-km-long stretch of the I5 freeway in California, with time horizons of 5, 15, and 30 minutes. A comparative study of different levels of hybridization in GACE is made. These range from a pure GA to a pure CE, passing through different weights for each of the combined techniques. The results prove that GACE is more accurate than GA or CE alone for predicting short-term traffic congestion.**

***Index Terms*—Intelligent Transportation Systems; Genetic Algorithms; Cross Entropy; Hierarchical Fuzzy Rule-Based Systems; Traffic Congestion Prediction; Curse of Dimensionality; Fuzzy Logic; Fuzzy Systems**

I. Introduction

In today's society, the number of vehicles in cities and freeways is continually increasing. In the USA, for example, there were 798 vehicles per 1000 persons in 2013. There are many problems related with this increase, such as noise, pollution, and traffic jams, which lead to time being wasted in travel and productivity loss.

Therefore, traffic congestion detection is a fundamental issue in the field of Intelligent Transportation Systems (ITSs). Successful prediction could result in noise reduction (not only in urban environments but also on freeways), energy savings (resulting in a decrease of pollution), increased effectiveness and performance of transport systems, and savings in public infrastructure [1], [2].

Two of the most frequently used methods of traffic forecasting in the last decade are the Kalman Filter (KF) [3] and the Autoregressive Integrated Moving Average (ARIMA) [4]. Both are regressive models that find patterns in order to predict a value in the future. The use of these methods has been extended to other fields. For example, ARIMA has been used for water quality prediction [5], electrical demand forecasting [6], and tourism demand forecasting[7]. KF has been used for air quality [8] and project duration forecasting [9]. In addition, these two methods can be combined, as in [10].

In recent years, other alternatives have been developed. In [11], Vehicle-to-Vehicle (V2V) communications have been used in large-scale highway scenarios. In [12], different machine-learning techniques are used to predict the average speed on a given street and given routes to allow individual cars to avoid traffic jams.

Another type of scheme is Time of Day (TOD). In [13], for example, TOD control divides a day into several intervals and tries to find optimal controls for each interval. SSoft computing techniques such as the Support Vector Machine (SVM), Neural Network (NN), Genetic Algorithm (GA), or Fuzzy Rule- Based System (FRBS) have been used separately [14], [15] and in combination [16], [17]. These methods have been extensively used in traffic forecasting in the last decade. For example, in [18], the authors proposed the use of Linear Genetic Programming, NN, and Fuzzy Logic for estimating 5- and 30-minute flow rates on a highway. A variant of ARIMA is combined with SVM in [19] to forecast the time series of traffic flow. In [20], statistical models such as the Support Vector Regression Model are combined with a chaotic immune algorithm to predict inter-urban traffic flow. Univariate and multivariate NN and autoregressive time series models are compared and used for short-term prediction of freeway speeds in [21]. A detailed survey on short-term traffic forecasting can be found in [22].

Among these techniques, SVMs may not perform well when dealing with large numbers of variables because of the choice of the appropriate kernel function for the practical problem [23]. NNs obtain better results and have attracted more attention. However, due to the local optimum problem and their generalizability of NNs, their effectiveness is limited. Besides, the result of an NN depends mainly on the network training process and is affected by the large amount of high quality data and the defined parameters [24].

The research presented in this paper is motivated by the intention of predicting short-term congestion on a 9-km long stretch of freeway. To do that, the congestion is predicted not only at one point on the road but along the whole section, offering a new approach to this kind of problem. For this purpose, the use of a Hierarchical Fuzzy Rule-Based System (HFRBS) for the prediction of traffic jams on a freeway is proposed. To optimize the parameters concerning each of the units in the hierarchy, a Cross Entropy (CE) algorithm is combined with a generational GA. The combination of

The authors are with the Deusto Institute of Technology (DeustoTech), University of Deusto, Bilbao 48007, Spain. email:{p.lopez, enrique.onieva, e.osaba, ad.masegosa, perallos}@deusto.es

Antonio D. Masegosa is also with IKERBASQUE, Basque Foundation for Science, Bilbao, Spain

This work has been partially funded by the TIMON Project *Enhanced real time services for an optimized multimodal mobility relying on cooperative networks and open data*. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 636220.

techniques is common in recent years [16], [21]. Although GAs are used in this kind of problem, the hybridization of this technique with others and its application to the optimization of FRBSs and their hierarchies is the main novelty of this study. Also, results obtained by HFRBS are easier to treat by the operator due to the linguistic information they provide.

The rest of this paper is structured as follows. Section II explains the different methods used in this paper. A wide view of the behaviour of the proposal and operators used is presented in Section III. Section IV contains the experimentation and the analysis of the results. Finally, Section V presents the conclusions and further research.

II. Preliminaries

Since the proposal is based on hybridization between GA and CE, both methods are described in Sections II-A and II-B respectively. In addition, Section II-C presents some of the basics of fuzzy logic in general, and HFRBS in particular.

A. Genetic Algorithms

GAs are search heuristics that mimic the process of natural selection. Since their appearance in the 1970s in Holland [25], their adaptability to hard problems has led GAs to appear in the literature both on their own [26], [27], and in combination with different techniques, in order to solve a wide variety of problems [28]–[30].

In its classical version, a GA maintains a population of candidate solutions, or individuals. Four procedures (selection, crossover, mutation, and replacement) are applied iteratively to this population.

GAs have been used in many cases to learn or to tune different components of FRBSs. In [31], a GA is used to obtain the best rule base possible for a controller for vehicle management at intersections. In [32], a wide explanation of GAs used for optimizing different parts of FRBSs is given.

Algorithm 1 is an example of a GA structure where $P_{crossover}$ and $P_{mutation}$ are the crossover probability and mutation probability, respectively.

Data: $Pop_{size},\ P_{crossover},\ P_{mutation}, T_{max}$
Result: $Best\ individual\ found$
$t \leftarrow 0$
$P_0 \leftarrow$ Initialize Population
Evaluate P_0
while $t < T_{max}$ **do**
  $Parents \leftarrow$ Select parents from P_t
  $Offspring \leftarrow$ Crossover($Parents, P_{crossover}$)
  $Offspring \leftarrow$ Mutate($Offspring$,$P_{mutation}$)
  Evaluate $Offspring$
  $P_{t+1} \leftarrow$ Replacement process with actual Population P_t and $Offspring$
  $t \leftarrow t + 1$
end

Algorithm 1: Pseudocode of workflow followed by the GA.

B. Cross Entropy

The CE method was proposed by Rubinstein in 1997 [33]. It was created as an adaptive method for rare-event probabilities and combinatorial optimization.

CE has three main phases:

1) *Generation* of $Samples_{num}$ random samples obeying a normal distribution with mean and variance of $\overline{x}$ and σ, respectively.
2) *Selection* of the $Samples_{selected}$ best samples from the set generated in the previous phase.
3) *Updating* of the $\overline{x}$ and σ values according to the fitness obtained by the best of the samples.

CE can be applied to estimation or optimization problems [34] and can be used in different fields. In [35], CE tunes fuzzy control systems for a drilling process. Another case is presented in [36], where CE is used for decision making, determining the optimal weights of attributes.

As the algorithm proceeds, $\overline{x}$ values are located at the points with the best results and σ become smaller until both are focused on the area of the best solutions found in the domain. CE counts with a parameter, $Learn_{rate}$. This parameter is used to update the means and variances during the execution of the algorithm with the means and variances of the new selected samples. Algorithm 2 presents the whole process. It is important to note that the algorithm is presented for a one-dimensional problem; in the case of more dimensions, $\overline{x}$ and σ must be vectors and each of their dimensions should be treated separately.

Data: $Samples_{num}, Update_{samples}, Learn_{rate}, T_{max}$
Result: $Best\ individual\ found$
$\overline{x} \leftarrow$ Initialize Means
$\sigma \leftarrow$ Initialize Variances
$t \leftarrow 0$
while $t < T_{max}$ **do**
  $Samples \leftarrow$ Generate $Samples_{num}$ Samples under N($\overline{x}$,σ)
  Evaluate $Samples$
  $Samples_{selected} \leftarrow$ Select the $Update_{samples}$ best from $Samples$
  $\overline{x} \leftarrow (1 - Learn_{rate}) \cdot \overline{x} + Learn_{rate} \cdot$ Mean($Samples_{selected}$)
  $\sigma \leftarrow (1 - Learn_{rate}) \cdot \sigma + Learn_{rate} \cdot$ Variance($Samples_{selected}$)
  $t \leftarrow t + 1$
end

Algorithm 2: Pseudocode of workflow followed by the CE.

C. Fuzzy Logic

Fuzzy logic, introduced by Zadeh in 1965 [37], allows uncertain information to be processed by using simple IF–THEN rules. There are many applications that use Fuzzy Logic [38]. In traffic problems, such as [39], fuzzy logic has often been adopted since it allows the information and decisions involved to be described by this kind of rule.

One of the most frequently used kinds of fuzzy systems is the FRBS. FRBSs are used in different kinds of real-world problems like energy management [40], remote health monitoring [41], and weather prediction [42].

A variant of the basic FRBS is the Hierarchical FRBS (HFRBS) [43] which is made up of several FRBSs, joined to each other in such a way that the output of one of them is the input of another. Thanks to their structure, these systems are useful for the improvement of the accuracy–interpretability balance in problems with a large number of variables, reducing the number of rules needed to process them, for example to find a possible solution to the *curse of dimensionality* problem [44]. Depending on the way the systems are structured [44], three different types of HFRBS can be found:

1) *Serial HFRBS*: The output of an FRBS is the input of the next one.
2) *Parallel HFRBS*: The structure is organized as layers. Outputs from the first layer are used as inputs of the second layer of the FRBSs, and so on.
3) *Hybrid HFRBS*: A combination of the previous cases.

This paper is focused on Parallel HFRBS (PHFRBS). With this organization, all the variables to be considered can be processed at the same time and with similar relevance. These PHFRBSs are applied with a constraint: each FRBS is restricted to two inputs. Therefore, if there are an odd number of variables, the last one will be one of the inputs of the last FRBS. Figure 1 shows the obtained hierarchies for three and four variables. In a formal way, a PHFRBS codified in this way to work with X input variables needs $X-1$ single FRBSs.

Besides, in this paper, the Takagi-Sugeno-Kang (TSK) type of FRBS is used. It employs trapezoids for codification of inputs and constant singletons for the outputs. For inference, the minimum T-norm has been used. TSK systems allow fast calculation of the output of the system. On the other hand, since the congestion levels considered in this work are consecutive (Sec. IV-A), non-discrete output values can be used.

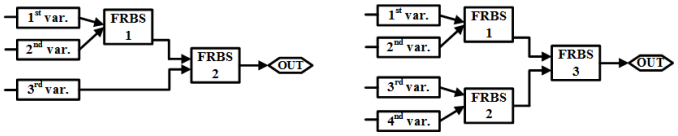

Fig. 1. Parallel HFRBS structured for three (a) and four (b) variables

Several authors of this paper have previously published works based on HFRBS and congestion prediction. In [16], a study using HFRBS in a similar way to this work is performed. The differences are that the authors use only a GA for the optimization of the systems and it is structured in a serial way.

III. Genetic Algorithm with Cross Entropy: GACE

In this paper, a PHFRBS optimized with a hybrid GA and CE method (GACE) is used to forecast short-term traffic congestion. On the one hand, the use of a PHFRBS helps with the problem of dimensionality. On the other hand, GACE improves the selection of variables for each system and the calculation of rules and fuzzy labels. The explanation of the algorithm is extended in the present section.

A. Chromosome Structure and Fitness Calculation

First of all, to be able to explain several aspects of the algorithm and calculate the values from the chromosome in later sections, its structure is introduced here. The chromosome codifies the three parts of the PHFRBS:

1) *Hierarchy*: This defines the subset of variables selected to be processed by the PHFRBS, as well as the order in which they are included in the system.
2) *Membership Functions* (MFs): Codify the location of the labels used to encode each of the input variables for each FRBS in the hierarchy.
3) *Rule Bases* (RB): These codify the positions of the singletons used as consequents of the rule bases of the FRBSs in the hierarchy.

In this paper, these parts are called $C_{hierarchy}$, C_{labels} and C_{rules}, respectively. $C_{hierarchy}$ consists of a permutation vector in which a value i in the position j denotes that the ith variable is inserted in the PHFRBS in the jth position. In addition, an ending character (denoted by 0), determines from which point of the vector no more variables are used. Therefore, the size of the permutation is $N_{variables}+1$. The number of modules ($N_{modules}$) is related to $N_{variables}$, due to the fact that the maximum of $N_{modules}$ is $N_{variables}-1$. Figure 1 proves this assertion.

C_{label} is composed of two real valued matrices in the $[-1,\ 1]$ interval, named MF_1 and MF_2, that codify the location of the MFs for the first and second input variables of each of the single FRBSs, respectively. In a formal way, C_{label} is formed by two $N_{modules}\cdot N_{labels}$ matrices. Therefore:

$$C_{label} = MF_i(j,k)\ \forall i \in \{1,2\} \\ \forall j \in \{1 \dots N_{modules}\},\ \forall k \in \{1 \dots N_{labels}\} \tag{1}$$

where each value denotes the kth MF used to codify the ith input of the jth FRBS in the hierarchy.

In addition, the lateral tuning technique designed in [45] for MF codification has been used. Originally, the values of the MFs are normalized to lie within the interval $[-1,\ 1]$. First, lateral tuning calculates the different equal-longitude divisions based on the number of labels used by the problem. After that, the method calculates the positions of the MFs in these divisions and converts them into a $[min,\ max]$ codification. Figure 2 shows how the codification works in a four-label structure. The equal-longitude divisions have been drawn in grey. On the other hand, final codification has been displayed in green. X_i values, previously normalized in a $[-1,\ 1]$ interval, are converted and placed in green divisions. Their final positions are marked as red circles.

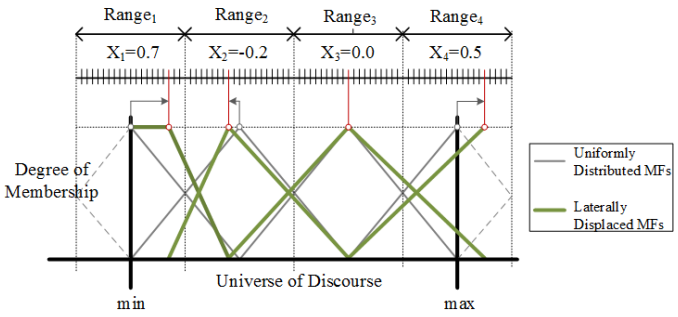

Fig. 2. Lateral tuning codification for an example with four MFs. $Range_i$ denotes the interval $[-1,\ 1]$ in which each X_i ('high' point of $Label_i$) can be located.

Finally, C_{rules} is presented as a real valued matrix within the interval $[0,\ 1]$ ($Rules$). The RB of each single FRBS has a size of ${N_{labels}}^2$.

$$C_{rules} = Rules_i(j)\ \forall i \in \{1 \dots N_{modules}\},\ j \in \{1 \dots N_{labels}^2\} \tag{2}$$

where each value denotes the consequent of the jth rule of the RB of the ith FRBS in the hierarchy.

As an example, Figure 3 presents a codification of a PHFRBS with six variables. As can be seen, $C_{hierarchy}$ determines the order in which the variables enter into the hierarchy, as well as the variables to be excluded (by means of the use of the ending character). On the other hand, the MFs and RB of the ith FRBS in the hierarchy are denoted by $MF_1(i,m)$, $MF_2(i,m)$ and $Rules_i(r)$, where $m \in \{1, \dots N_{labels}\}$ and $r \in \{1, \dots N_{labels}^2\}$.

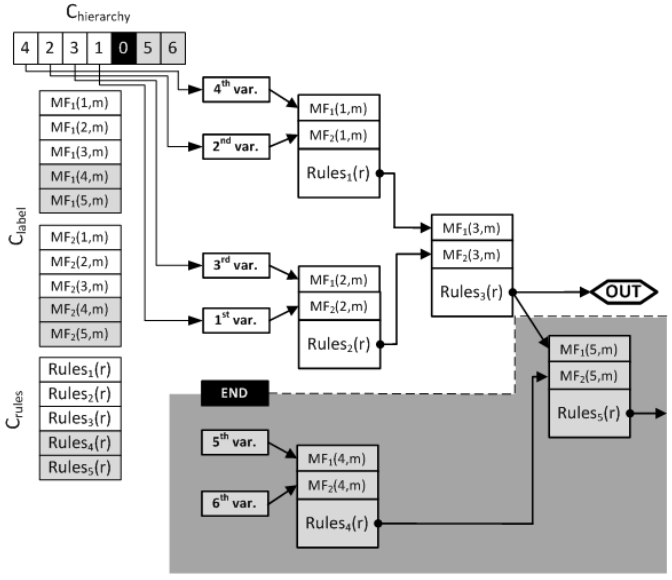

Fig. 3. Example of codification for a PHFRBS to process up to six input variables.

It is important to note that since the C_{label} and C_{rule} are limited, respectively, to lying within $[-1,\ 1]$ and $[0,\ 1]$, all the implemented operators must take values within the corresponding intervals.

For the fitness function, the mean absolute error (MAE) [46] is calculated. This value takes the difference between the obtained and expected values and divides it by the number of samples. Equation 3 presents its calculation.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\overline{Y_i} - Y_i| \tag{3}$$

where n is the number of instances, Y is the predicted value and $\overline{Y}$ is the expected one.

B. GACE: Genetic Algorithm and Cross Entropy

GACE is designed with the idea of taking advantage of the exploration ability of a GA and the exploitation ability of a CE in a given optimization problem, with the aim of optimizing the inputs, labels, and rules of a PHFRBS.

For this purpose, first, the initialization of the PHFRBSs constituting the initial population is done. In each iteration, the population of solutions (POP_t) is divided into two sub-populations, GA_{pop} and CE_{pop}, with GA_{size} and CE_{size} individuals, respectively. GA_{pop} is chosen by a selection method and used for applying the GA operators, while CE_{pop} is formed by the CE_{size} best individuals in the population POP_t and used in the CE part of the algorithm. Both GA_{size} and CE_{size} are chosen by the user, and their sum equals POP_{size}.

Once both populations have been chosen, each is used in a different way:

- GA_{pop}: GA operators are applied to this population to generate GA_{size} new individuals. The specifications of the operators used are given in detail in Section III-C.
- CE_{pop}: In this case, the CE algorithm is applied to the individuals of the population. First, $\overline{x}$ and σ are updated by employing Algorithm 2 (lines 7 and 8). Then, CE_{size} individuals are randomly generated, obeying a normal distribution, using $\overline{x}$ and σ. Section III-D presents the implementation of this process in detail.

A new population is formed of the individuals generated by the last two operations. It contains GA_{size} individuals resulting from GA operators and CE_{size} individuals from the CE method. This new population replaces the old one. Therefore, GACE is a generational algorithm. Figure 4 shows the described working.

It is important to note that for the optimization of a PHFRBS, two different codifications are used in the same individual: permutation for $C_{hierarchy}$ and real-valued for C_{label} and C_{rules}. For this reason, specific operators are used to work with the corresponding codifications. They are explained in detail in Sections III-C and III-D.

C. Genetic Algorithm Operators

In this section, explanations of the different operators used in the GA part are presented.

Binary tournament [47] is used as the selection algorithm. It chooses two random individuals in the population. The winner of the tournament is the fittest individual. The process

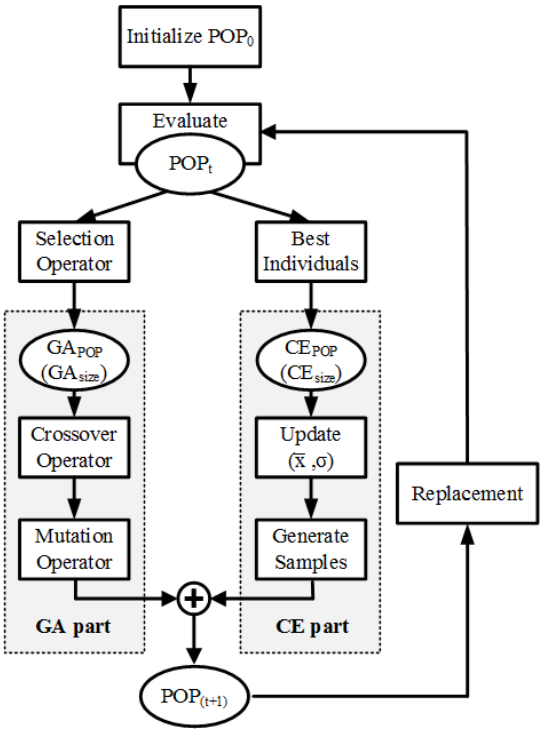

Fig. 4. Steps of GACE algorithm: A population POP_t is divided into two sub-populations GA_{pop} and CE_{pop}. After applying the operators, GA_{size} and CE_{size} individuals form the new population POP_{t+1} and replace the old one.

is repeated as often as desired. With this method, GA_{size} individuals are taken.

Although the label and rules parts of our individuals are real valued matrices, the hierarchy part is a permutation. So, two crossover and mutation operators are needed.

For the permutation part, a variant of the order crossover has been chosen [48], where only one point is selected to perform the operation. This function randomly chooses a cutpoint c. The content of each parent from the first position to c is preserved in the offspring. The rest of the offspring are sorted according to the parent from which the first segment is not inherited. The decision to use only one point is imposed in order to keep the first variables in their positions in the offspring $C_{hierarchy}$ since they have more possibility of entering into the PHFRBS (since they would be before the ending character). Figure 5 provides an example. Considering two parents (P_1 and P_2), the offspring are formed in the first instance by $\{2, 3, 4\}$ and $\{4, 5, 2\}$ (light boxes). Then, starting from the cut point c, values are chosen in the order fixed by the other parent. For each parent, values placed before the cut point are not taken into account in the offspring of the other parent. These values are marked with X's in their boxes. Therefore, the sequence for O_1 is $\{1, 6, 5\}$ and the sequence for O_2 is $\{6, 1, 3\}$. These sequences are placed after the cut point c in their corresponding children.

For the real part of the individuals (label and rules), BLX-α [49] crossover is adopted. Given two parents $A = (a_1...a_m)$ and $B = (b_1...b_m)$ for each i, BLX-α crossover creates two offspring by generating random values in the interval presented

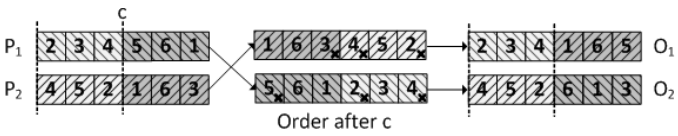

Fig. 5. Variant of the order crossover used in the present work. First, the cut point is chosen. After that, the order is selected, and finally the new individuals are created.

TABLE I
INITIALIZATION VALUES FOR EACH OF THE PARTS IN $\overline{x}$ AND σ.

Individual	$\overline{x_0}$	σ_0
$C_{hierarchy}$	$0.5 \cdot N_{variables}$	$0.5 \cdot N_{variables}$
C_{labels}	0	1
C_{rules}	0.5	0.5

in Equation 4, with $\alpha \in [0,\ 1]$. This crossover is chosen by the authors because of its good synergy between exploration and explotation of the individual [50].

$$[min(a_i, b_i) - \alpha|a_i - b_i|,\ max(a_i, b_i) + \alpha|a_i - b_i|] \quad (4)$$

For mutation operators, the distinction between hierarchy, labels and rules is also made. For the hierarchy, a swap mutation operator [51] is applied: two random positions in the permutation are exchanged for each other. For labels and rules, a BGA mutation [52] is used. Given $A = (a_1 \dots a_m)$, the BGA operator returns a_i, calculated as presented in Equation 5. The movements performed by this operator are small, supposing a small change in the individual. This is the reason for the choice of this operator.

$$a_i' = a_i \pm \beta \sum_{k=0}^{15} (\alpha_k 2^{-k}) \quad (5)$$

where β defines the mutation range. The sign (+ or -) is chosen with probability 0.5, and $\alpha_k \in \{0, 0.33, 0.66, 1\}$ is randomly generated with $p(\alpha_k = \{0, 0.33, 0.66, 1\}) = \frac{1}{4}$.

D. Cross Entropy Operators

As mentioned before, two individuals, one representing the average $\overline{x}$ and another representing the variance σ, must be kept by CE. These individuals have the same structure as a normal individual (Figure 3), but they have to be initialized in such a way that generating new samples from them (in the first iteration) is equivalent to randomly generating individuals. For this purpose, the initialization of each of the parts is carried out as presented in Table I. These initial values will be updated during the execution of the algorithm.

The individuals to be processed by CE are selected in a deterministic way. The CE_{size} best individuals in the population are chosen to constitute CE_{pop}. Once they have been obtained, the individuals $\overline{x}$ and σ update their values and new samples are generated.

The updating is done by directly applying the equations in lines 7 and 8 from Algorithm 2 to the C_{label} and C_{rule} parts of the individuals. For these parts, the generation of new samples is done by generating random values obeying the normal distribution with $\overline{x}$ and σ.

For the $C_{hierarchy}$ part of the individuals, the updating of the mean and variance and the generation of new samples follow a different process. The vectors representing the hierarchy are converted into vectors that store the order of each of the variables (the last position is used to represent the position of the ending character). Then $\overline{x}$ and σ are updated by using these order vectors. Finally, the samples generated are subjected to the inverse transformation (from order to hierarchy). This process is illustrated in Figure 6 and works as follows: the given selected individuals (a) are converted into order vectors (b); then the mean and variance values are calculated and used to update $\overline{x}$ and σ (c). Finally, new samples are generated (d) and converted into hierarchy vectors (f).

The individuals generated are combined with the ones coming from the GA part of the method in order to generate the population to be processed in the next iteration of the algorithm.

IV. Experimentation

In this section, the performed experimentation is presented. In Section IV-A, the data used in this work are described. The organization of those data is explained in IV-B. Section IV-C contains every different configuration used to carry out the experimentation. Finally, Section IV-D presents the results of each experiment carried out in test data and analysis of them.

A. Data support

The data used in this work were provided by the Caltrans Performance Measurement System (PeMS[1]). PeMS is a real-time database from the California Department of Transportation that offers over 10 years of traffic data for historical analysis. A 9-km-long section of highway I5 in Sacramento, California, is used for this research. A total of 13 points where sensors are situated on the main road and eight ramp sensors were chosen. Traffic measures were collected by the detector stations every 5 minutes. The information collected by the main road sensors monitored the flow (the number of vehicles), the occupancy (the percentage of the time during which the sensor was switched on) and the speed (in miles per hour). The information taken from the ramp sensors only concerns the flow. The data were collected from 0:00 on September 1, 2013 to 23:55 on September 30, 2013. Figure 7 shows a schema of the fragment of road used as well as the sensors located on it.

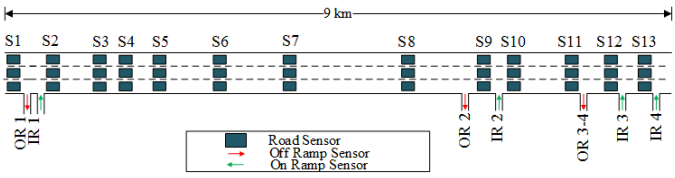

Fig. 7. Segment of highway I5 used in this study. Sensors are denoted by S, Off Ramps by OR, and On Ramps by IR.

A congestion variable is added at the end of the datasets. It is calculated using the intervals shown in Table II, which are adopted from [53]. Although the density is not given by PeMS, its calculation is carried out by using the values of the flow and speed: $density = flow/speed$. Therefore, congestion can take one of four values: free, slight, moderate and severe. These values are presented as universal units of the metric system (km).

TABLE II
Values of congestion and their calculus.

Level of Congestion	Traffic Density (ve/km/ln)	Vehicle Speed (km/h)
Slight	[29–37]	[48–80]
Moderate	[37–50]	[24–64]
Severe	> 50	< 40
Free	Other cases	

B. Datasets

With the data obtained in the previous section, four different types of datasets were created:

- Point Dataset (PD): This dataset contains all sensors. The congestion column is calculated only for a point on the road. In this case, the middle point corresponding to $S7$ is chosen.
- Simplified Point Dataset (SPD): This dataset is a simplified version of PD. It only contains data from the first ($S1$), middle ($S7$), and last ($S13$) sensors and a combined value of the off-ramp and on-ramp flows. The congestion values are calculated in the same way as in PD. Figure 8 shows the way in which the data were collected. ΔIR and ΔOR represent the combined flow value of the in-ramps and off-ramps, respectively, located before and after the point of interest.
- Section Dataset (SD): This dataset contains all sensors, and the congestion is calculated as the maximum level of congestion that occurs at any of the the sensors on the main road.
- Simplified Section Dataset (SSD): this is a simplified version of SD. It contains only the first ($S1$), middle ($S7$), and last ($S13$) sensors and an average flow of the off-ramps and on-ramps. The congestion is calculated in the same way as in SD.

In summary, PD and SPD aim at predicting the congestion at a single point $S7$, while SD and SSD aim at predicting the maximum level of congestion appearing in the whole road section. On the other hand, PD and SD use all the available information while SPD and SSD use a version of the dataset with a smaller number of attributes. Besides, the proposed technique takes into account what happens before and after the point of interest to make the prediction in $S7$ more accurate in the PD and SPD datasets. In addition, for each type of dataset, in order to make a congestion prediction, the desired output will be the value of congestion that occurs within a time horizon of 5, 15, or 30 minutes ahead. From now on, 12 generated datasets will be used and denoted by the acronym and the time horizon. For instance, PD_5 will denote the point dataset with a time horizon of 5 minutes.

In order to provide a preview of the datasets collected, Figure 9 shows the density of examples for each level of

[1]www.pems.dot.ca.gov

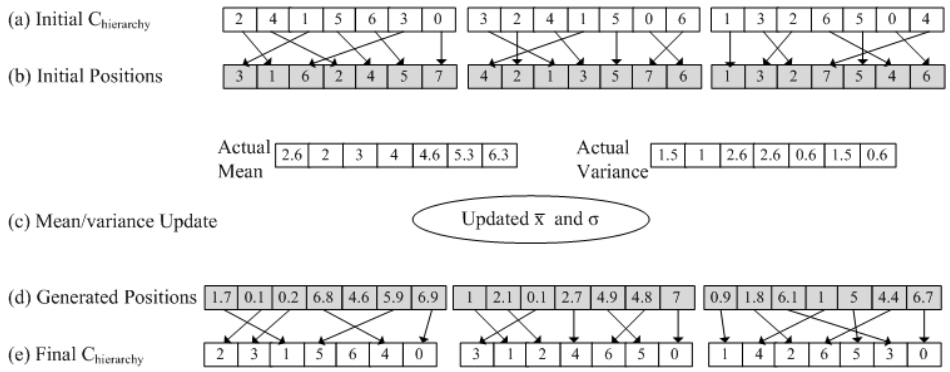

Fig. 6. Creation phases of hierarchy part in a new individual

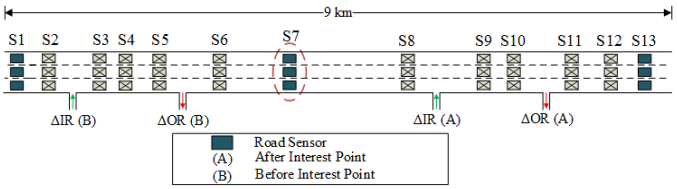

Fig. 8. Simplified Point Dataset, taking into account only $S1$, $S7$, and $S13$ and a combination of OR and IR before and after the point of interest.

congestion considered in this work. In this figure, densities are shown with respect to the day of the week (left) and hour of the day (right) for datasets regarding data related to the point of interest (top) and the section of the road (bottom). These figures show graphically the probability of occurrence of each of the traffic states in different datasets with respect to the temporal variable. With the exception of the SD dataset, where a state of severe congestion is highly possible between 7:00 and 9:00 hours, during the rest of the time, it is hard to properly estimate the state of the road in the datasets.

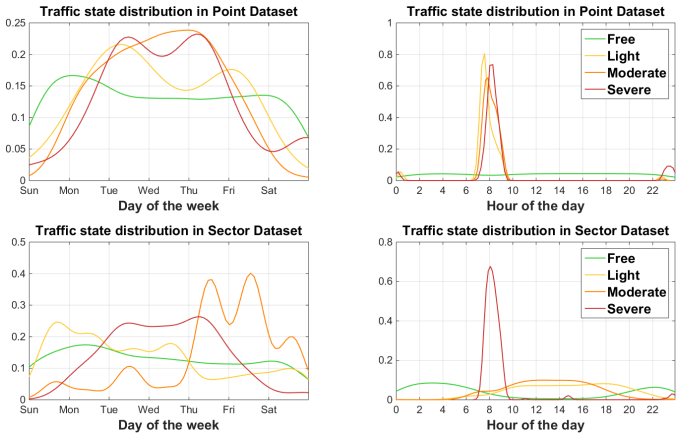

Fig. 9. Traffic state distribution in PD_5 and SD_5 datasets depending on the day of the week and hour of the day.

The datasets used in this paper contain a larger number of free flow examples than any other class of congestion. For this reason, these datasets are highly unbalanced. In order to validate this affirmation, the imbalance ratio (IR) [54] is used. Equation 6 shows how the calculation has been done;

TABLE III
DATASETS USED IN THE PRESENT WORK. VALUES BEFORE THE ARROW INDICATE THE ORIGINAL VALUE; THOSE AFTER THE ARROW WERE OBTAINED AFTER REDUCTION.

Datasets	Vars	Free	Slight	Moderate	Severe	IR
PD_5	47	8277 → 850	61→ 61	172 → 172	129 → 129	135.6→ 13.9
PD_{15}	47	8275 → 866	61 → 61	172→ 172	129→ 129	135.6→ 14.1
PD_{30}	47	8272→ 847	61→ 61	172 → 172	129→ 129	135.6→ 13.8
SPD_5	13	8277→ 113	61→ 55	172 → 120	129→ 97	135.6→ 2.1
SPD_{15}	13	8275→ 139	61 → 45	172 → 102	129→ 103	135.6→ 3.0
SPD_{30}	13	8272→ 88	61 → 58	172 → 108	129 → 96	135.6→ 1.8
SD_5	47	4157→ 293	2776 → 473	1402 → 453	304 → 304	13.6→ 1.6
SD_{15}	47	4155 → 290	2776 → 549	1402 → 444	304 → 304	13.6→ 1.8
SD_{30}	47	4152→ 293	2776 → 464	1402 → 442	304→ 304	13.6→ 1.5
SSD_5	11	4157→ 46	2776 → 58	1402 → 65	304 → 108	13.6→ 2.3
SSD_{15}	11	4155→ 56	2776 → 57	1402 → 85	304 → 110	13.6→ 1.9
SSD_{30}	11	4152 → 54	2776 → 43	1402 → 76	304 → 114	13.6→ 2.6

in addition, Table III shows the number of instances of each of the types of congestion considered and other values such as the number of variables in each dataset and the number of instances of each class.

$$IR = MAC/MIC \tag{6}$$

where MAC is the number of instances of the majority class and MIC is the number of instances of the minority class.

With the aim of balancing the datasets, a simplification procedure inspired by the K-nearest neighbours method [55] is used. The reduction is based on two parameters: k and u, where k indicates the chosen number of neighbours, and u is a threshold chosen by the user that cannot be exceeded for the distance between the actual node and any of the chosen neighbours. If this threshold distance between actual node and nth neighbour is exceeded, the nth neighbour is deleted from the dataset. The method is iterative and stops when no nodes can be deleted, that is, when any of the combinations of a node and its neighbours surpasses the indicated threshold. The results obtained by applying this reduction method to the datasets used in this paper are shown in Table III after the arrows. This table also contains the IR values of the datasets so as to be able to compare it with the complete ones.

Reduced datasets are used as training data for GACE in order to avoid over-fitting of the majority classes, while the complete datasets are used for testing the results. In this work, the forecasting problem has been treated like a classification one. States of the road are replaced by numbers to provide proper forecasting measures [56]. In particular, following

assignations of congestion states will be used for the calculation of the MAE (Eq. 3): {Free=1, Slight=2, Moderate=3, and Severe=4}. With this change, accuracy metrics can be used in forecasting in a proper way. Finally, these datasets are provided[2], so that they can be downloaded for use.

C. Experimental Setup

Different combinations of the number of individuals in GA_{pop} and CE_{pop} are tested and compared. In addition, GACE configurations are compared with pure GA ($CE_{size} = 0$) and pure CE ($GA_{size} = 0$) to test the benefits of using the proposed hybridization, in comparison with using each of the methods alone. Eight experiments with different numbers of individuals in the populations were carried out; each execution was repeated 10 times in order to obtain average results.

For the experiments, several values of number of generations and population size have been tested. Finally, the number of generations was set to 500 and the population size to 50. The executions are referred to by their population sizes: that is, $\{GA_{size} - CE_{size}\}$; for example, 50-0 refers to a GA with the operators explained in this paper, and 0-50 is the same as a CE execution. $GA_{size} \in \{50, 45, 40, 35, 25, 15, 10, 0\}$ is used, and the remaining $CE_{size} = 50 - GA_{size}$.

The number of MFs to be used in each of the single FRBSs that compose the hierarchy was set to $N_{labels} = 3$, so the number of rules in each FRBS is $N_{rules} = N_{labels}^2 = 9$. The value of N_{labels} has been chosen in order to establish three MFs (low, middle, and high) for each input variable of a system. The increase in N_{labels} involves an increase in N_{rules}, and it can worsen the performance and accuracy of the systems. The size of the hierarchy part of the codification depends on the number of variables in each dataset plus one (the ending character), and varies from 12 to 48 (Table III).

For the GA part of the experimentation, the probability of crossover was set to 0.8, and the probability of mutation was set to 0.2. These probabilities were set to these values due to the use of a high crossover probability and low mutation probability in most applications of GAs. For BLX crossover and BGA mutation, $\alpha = \beta = 0.5$. While the use of $\alpha = 0.5$ is the default value for BLX, $\beta = 0.5$ is chosen for BGA to keep the probabilities of exploration and exploitation at the same level. In the CE part of the algorithm, it is recommended that $Learn_{rate}$ be in range of $[0.7,\ 0.9]$ [57]. Following this recommendation, $Learn_{rate} = 0.7$ was used in order to update $\overline{x}$ and σ.

A cross-validation method for testing the model is used. Cross-validation divides the dataset into n sub-datasets with the same number of instances; $n - 1$ sub-datasets are used for training the model, and the last one is used for testing it. In this case, $n = 10$. The instances in each sub-dataset are chosen randomly from the whole dataset.

D. Results

The experimentation was done using Matlab Software. Table IV shows the average symmetric mean absolute percentage error (sMAPE) [58] in the test datasets, considering previous state assignations ({Free=1, Slight=2, Moderate=3 and Severe=4}). The calculation of sMAPE is shown in Eq. 7, where $\overline{Y}$ is the expected value, Y is the predicted one, and n is the number of examples. By using sMAPE, the problem of large errors when the actual value, $\overline{Y}$, is close to zero and the large difference between the absolute percentage errors when $\overline{Y}$ is greater than Y and vice versa is avoided [59].

TABLE IV
AVERAGED SMAPE IN TEST SETS FOR EACH OF THE TECHNIQUES.

Dataset	GA	GACE $45-5$	GACE $40-10$	GACE $35-15$	GACE $25-25$	GACE $15-35$	GACE $10-40$	CE
PD_5	0.023	**0.022**	**0.020**	**0.020**	0.023	0.045	0.028	0.039
PD_{15}	0.017	**0.011**	**0.011**	**0.013**	0.015	0.023	0.023	0.067
PD_{30}	0.044	**0.017**	**0.016**	**0.017**	0.019	0.018	0.026	0.042
SPD_5	0.020	**0.019**	**0.018**	0.025	**0.018**	0.023	0.027	0.303
SPD_{15}	0.017	0.016	**0.011**	**0.012**	0.015	0.060	0.109	0.434
SPD_{30}	0.031	0.027	**0.021**	**0.021**	0.026	0.028	0.040	0.298
SD_5	0.199	0.199	0.202	0.204	**0.198**	**0.197**	0.340	0.484
SD_{15}	0.333	**0.240**	0.251	**0.217**	**0.240**	0.365	0.463	0.545
SD_{30}	**0.202**	**0.186**	0.244	0.210	0.205	0.318	0.378	0.445
SSD_5	0.237	**0.201**	**0.234**	0.296	0.327	0.375	0.396	0.355
SSD_{15}	**0.301**	**0.256**	0.322	0.344	0.311	0.374	0.375	0.361
SSD_{30}	0.306	**0.221**	0.328	**0.299**	0.346	0.343	0.387	0.387

$$sMAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|\overline{Y_i} - Y_i|}{(|\overline{Y_i}| + |Y_i|)/2} \cdot 100 \qquad (7)$$

The boldfaced values in the tables indicate the two best values for each dataset. In most cases, a GACE with $GA_{size} \in [35,\ 45]$ obtains values of higher accuracy than other algorithms such as $GACE_{25-25}$, GA, or CE. Therefore, a better performance is obtained by the algorithms that use a bigger GA_{size} than CE_{size}. $GACE_{45-15}$ obtains one of the two best results in 9 out of 12 cases, while $GACE_{40-10}$ obtains one of the best results in 7 out of 12 cases. On the other hand, the results obtained by $GACE_{10-40}$ and CE are not among the best ones in any case. Comparing GA with CE, GA obtains the best result in all cases and CE obtains the worst result in 9 out of 12 cases.

Figures 10-13 show the percentage of correctly classified instances for each of the types of congestion predicted in this research. Each bar indicates the mean and variation values for each execution carried out; thus, eight bars are shown per plot. Each bar is denoted by its GA_{size} value on the X axis. From left to right are the values from pure GA to CE.

In most cases, the GACE configurations achieve performance better than or equal to that of GA or CE, with pure GA obtaining values closer to those obtained by GACE in most of the datasets. The best prediction values are obtained when the time horizon is low (5 minutes) and when congestion prediction is done for a point (PD and SPD). When predicting congestion in a segment (SD and SSD), a 30-minute forecast is possible with good accuracy values. Regarding the use of simplified or complete datasets, it is easy to come to the conclusion that in the point prediction cases (PD and SPD), simplified datasets give practically the same values for any congestion type. The best results are obtained with the SPD_5 dataset. For the segment prediction cases, good prediction values for Severe congestion are obtained with all of the datasets, significantly improving the prediction of Free

[2]http://research.mobility.deustotech.eu/media/publication_resources/I5_Congestion_Datasets_GACE2015.rar

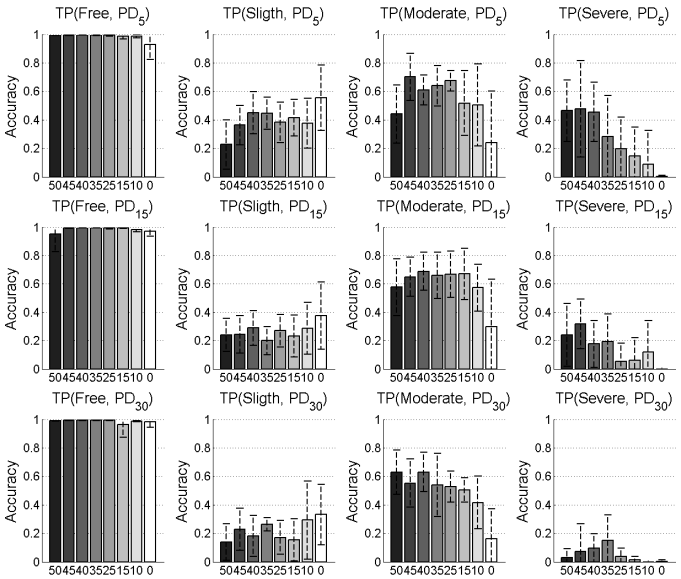

Fig. 10. Percentage of correctly classified instances for each class in the datasets PD_5 (top), PD_{15} (center), and PD_{30} (bottom).

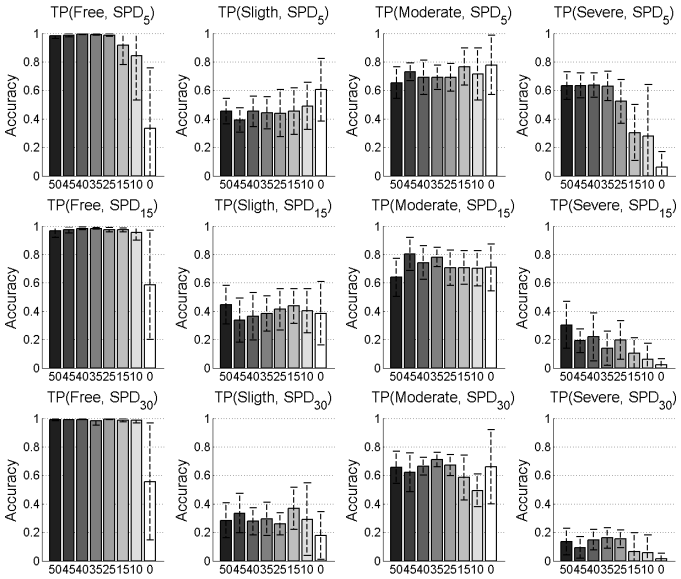

Fig. 11. Percentage of correctly classified instances for each class in the datasets SPD_5 (top), SPD_{15} (center), and SPD_{30} (bottom).

congestion values' with the simplified datasets, specifically SSD_5 and SSD_{15}, where the results between the different congestion levels are more stable.

Table V shows the average ranking for each technique in all the datasets, that is, the average position occupied when they are sorted by fitness in each dataset. The results show that $45-5$ is the best in 50% of the cases (2 out of 4 congestion cases) while $40-10$ and $35-15$ obtain the best position in the remaining cases. This can be seen in Table VI, where the best techniques for each dataset and congestion case are shown. GACE techniques have the best performance in 67% of cases (32 of 48), mostly in the Free (11 of 12) and Severe (10 of 12) congestion cases, while in Slight and Moderate cases the GACE techniques have the best performance in only 6 and

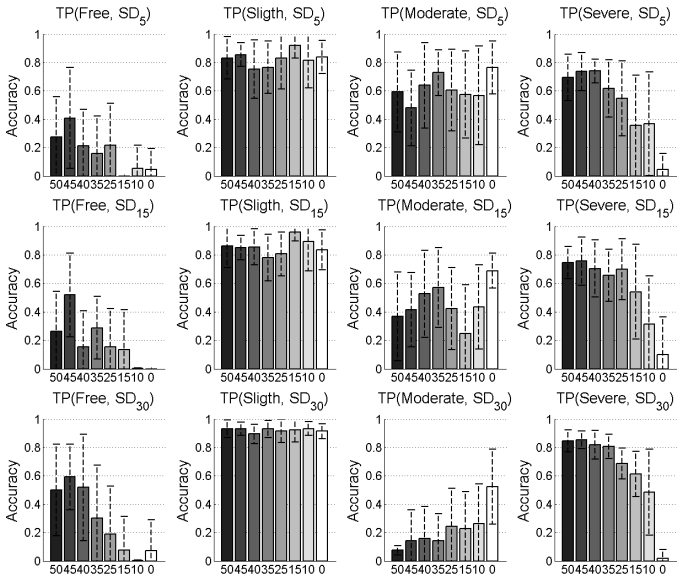

Fig. 12. Percentage of correctly classified instances for each class in the datasets SD_5 (top), SD_{15} (center), and SD_{30} (bottom).

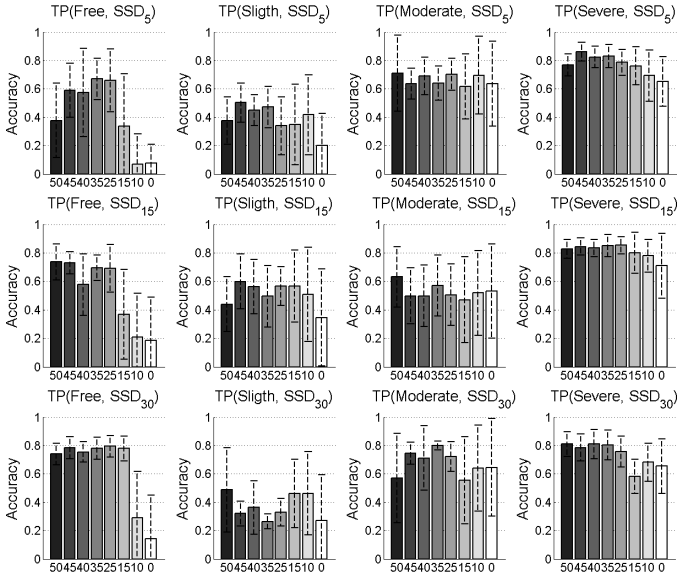

Fig. 13. Percentage of correctly classified instances for each class in the dataset: SSD_5 (top), SSD_{15} (center), and SSD_{30} (bottom).

TABLE V
AVERAGE RANKING FOR EACH TECHNIQUE WHEN PREDICTING DIFFERENT LEVELS OF CONGESTION. THE TWO BEST VALUES ARE BOLDFACED FOR EACH CASE.

	GA	GACE $45-5$	GACE $40-10$	GACE $35-15$	GACE $25-25$	GACE $15-35$	GACE $10-40$	CE
Free	4.16	**2.66**	**2.91**	3.08	3.08	5.75	6.75	7.58
Slight	4.50	4.25	4.83	5.33	5.33	**3.66**	**3.41**	4.66
Moderate	5.33	4.66	**3.58**	**3.16**	3.91	5.75	5.41	4.16
Severe	2.91	**2.33**	**2.58**	3.16	4.16	6.25	6.75	7.83
Total	4.22	**3.47**	**3.47**	3.68	4.12	5.35	5.58	6.06

TABLE VI
BEST TECHNIQUE FOR PREDICTING CONGESTION IN EACH DATASET.

	Free	Slight	Moderate	Severe
PD_5	$40-10$	CE	$45-5$	$45-5$
PD_{15}	$35-15$	CE	$40-10$	$45-5$
PD_{30}	$25-25$	CE	GA	$35-15$
SPD_5	$40-10$	CE	CE	$40-10$
SPD_{15}	$40-10$	GA	$45-5$	GA
SPD_{30}	$25-25$	$15-35$	$35-15$	$35-15$
SD_5	$45-5$	$15-35$	CE	$40-10$
SD_{15}	$45-5$	$15-35$	CE	$45-5$
SD_{30}	$45-5$	$10-40$	CE	$45-5$
SSD_5	$35-15$	$45-5$	GA	$45-5$
SSD_{15}	GA	$45-5$	GA	$25-25$
SSD_{30}	$25-25$	GA	$35-15$	GA

TABLE VII
COMPARISON OF GACE TECHNIQUES WITH C4.5, LDWPSO, AND SGERD ALGORITHMS

Dataset	GACE $45-5$	GACE $40-10$	GACE $35-15$	C4.5	LDWPSO	SGERD
PD_5	0.022	**0.020**	**0.020**	**0.002**	0.100	0.043
PD_{15}	**0.011**	**0.011**	0.013	**0.004**	0.100	0.043
PD_{30}	0.017	**0.016**	0.017	**0.004**	0.089	0.043
SPD_5	0.019	**0.018**	0.025	**0.018**	**0.013**	0.604
SPD_{15}	0.016	**0.011**	**0.012**	0.033	0.027	0.108
SPD_{30}	**0.027**	**0.021**	**0.021**	0.052	**0.027**	0.029
SD_5	**0.199**	0.202	0.204	**0.048**	0.606	0.202
SD_{15}	0.240	0.251	**0.217**	**0.140**	0.604	0.396
SD_{30}	**0.186**	0.244	0.210	**0.176**	0.602	0.395
SSD_5	**0.201**	**0.234**	0.296	0.362	0.362	0.684
SSD_{15}	**0.256**	**0.322**	0.344	0.328	0.423	0.766
SSD_{30}	**0.221**	0.328	0.299	0.310	0.400	**0.270**

5 cases, respectively.

In addition to the experimentation carried out, a comparison between the three best GACE configurations and three algorithms from the literature is made. The algorithms are executed using KEEL Software [60] with their default values. These techniques are C4.5 [61], Linear Decreasing Weight - Particle Swarm Optimization (LDWPSO) [62], and Steady-State Genetic Algorithm for Extracting Fuzzy Classification Rules (SGERD) [63].

Table VII contains the sMAPE for each dataset and technique. Boldfaced values indicate the best result for each dataset.

In the PD cases, C4.5 is better than GACE in 3 out of 6 cases but GACE is between the best values obtained in all of the datasets. On the other hand, in SD cases, GACE performs better than methods from the literature in 3 out of 6 cases (SSDs). The biggest difference between two best values is found for the SD_5 dataset, with $GACE_{45-5}$ and C4.5. LDWPSO obtains one of the best values in two cases, while SGERD obtains the second best value in only one dataset.

Looking at the results of the experimentation, it can be said that the proposed technique is a competitive technique. Even so the algorithm could improve in PDs and SDs in order to be competitive with this type of technique or achieve a better performance on this kind of datasets.

V. CONCLUSIONS

In this paper, a combination of a Genetic Algorithm (GA) and the Cross Entropy (CE) method for the optimization of a Parallel Hierarchical Fuzzy Rule-Based System (PHFRBS) is presented. The objective of the combination is to obtain a synergy between exploration and exploitation in order to improve the system's parameters. The method divides the population into two sub-populations. In one of them, a GA is used, while in the other, the CE method is executed. After that, both sub-populations are joined, and they replace the current population. The final aim is to predict congestion at a point or in a segment of the I5 freeway in California for a time horizon of 5, 15, or 30 minutes. The data were obtained from a governmental source and were used to prove the efficiency of the systems. Four different datasets were created with these data.

Different weights, denoted as GA_{size} and CE_{size}, are used as the population sizes for the experimentation. GACE configurations are compared with pure GA ($CE_{size} = 0$) and pure CE ($GA_{size} = 0$) to test the benefits of using the proposed hybridization in comparison with the two pure methods alone. It was found that GACE showed good performance. The algorithm can predict short-term congestion at a point with a very low error. GACE obtains better results with a bigger GA_{size} than CE_{size}. $GACE_{45-15}$ is, in 9 of 12 cases, one of the best performers. On the other hand, CE obtains the worst values in 10 out of 12 cases. When analysing datasets, the best errors are obtained with complete datasets. Despite this issue, the errors obtained with simplified datasets are small enough to consider using only this kind of dataset in future works.

Although the proposed technique has proved its good performance on this kind of problem, it would be interesting to demonstrate whether combining GA with other techniques like Differential Evolution or PSO would improve the optimization of these systems. Besides, GACE can be used in other research areas due to the adaptability of its parts. The population size can be an obstacle, because it can change depending on the kind of problem being addressed. In future research, improvements can be made in the optimization of the PHFRBS by applying different methods. Also, a model that automatically selects the best combination of sizes of the GA and CE parts or improves the results in the cases of moderate and severe congestion can be sought.

REFERENCES

[1] A. Abadi, T. Rajabioun, and P. Ioannou, "Traffic flow prediction for road transportation networks with limited traffic data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 653–662, 2015.

[2] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.

[3] L. Zhang, J. Ma, and C. Zhu, "Theory modeling and application of an adaptive kalman filter for short-term traffic flow prediction," *Journal of Information and Computational Science*, vol. 9, no. 16, pp. 5101–5109, 2012.

[4] M.-C. Tan, L.-B. Feng, and J.-M. Xu, "Traffic flow prediction based on hybrid arima and ann model," *Zhongguo Gonglu Xuebao/China Journal of Highway and Transport*, vol. 20, no. 4, pp. 118–121, 2007.

[5] D. Ámer Faruk, "A hybrid neural network and arima model for water quality time series prediction," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 4, pp. 586–594, 2010.

[6] J. Taylor, L. de Menezes, and P. McSharry, "A comparison of univariate methods for forecasting electricity demand up to a day ahead," *International Journal of Forecasting*, vol. 22, no. 1, pp. 1–16, 2006.

[7] C.-F. Chen, M.-C. Lai, and C.-C. Yeh, "Forecasting tourism demand based on empirical mode decomposition and neural network," *Knowledge-Based Systems*, vol. 26, pp. 281–287, 2012.

[8] K. De Ridder, U. Kumar, D. Lauwaet, S. Van Looy, and W. Lefebvre, "Kalman filter-based air quality forecast adjustment," *NATO Science for Peace and Security Series C: Environmental Security*, vol. 137, pp. 177–181, 2013.

[9] B.-C. Kim and K. Reinschmidt, "Probabilistic forecasting of project duration using kalman filter and the earned value method," *Journal of Construction Engineering and Management*, vol. 136, no. 8, pp. 834–843, 2010.

[10] H. Liu, H.-Q. Tian, and Y.-F. Li, "Comparison of two new arima-ann and arima-kalman hybrid methods for wind speed prediction," *Applied Energy*, vol. 98, pp. 415–424, 2012.

[11] R. Bauza and J. Gozalvez, "Traffic congestion detection in large-scale scenarios using vehicle-to-vehicle communications," *Journal of Network and Computer Applications*, vol. 36, no. 5, pp. 1295–1307, 2013.

[12] J. Rzeszótko and S. Nguyen, "Machine learning for traffic prediction," *Fundamenta Informaticae*, vol. 119, no. 3-4, pp. 407–420, 2012.

[13] L. Jia, L. Yang, Q. Kong, and S. Lin, "Study of artificial immune clustering algorithm and its applications to urban traffic control," *International Journal of Information Technology*, vol. 12, no. 3, pp. 1–6, 2006.

[14] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, and L. D. Han, "Online-svr for short-term traffic flow prediction under typical and atypical traffic conditions," *Expert systems with applications*, vol. 36, no. 3, pp. 6164–6173, 2009.

[15] R. Li and G. Rose, "Incorporating uncertainty into short-term travel time predictions," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 6, pp. 1006–1018, 2011.

[16] X. Zhang, E. Onieva, A. Perallos, E. Osaba, and V. Lee, "Hierarchical fuzzy rule-based system optimized with genetic algorithms for short term traffic congestion prediction," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 127–142, 2014.

[17] C. Sicre, A. Cucala, and A. Fernández-Cardador, "Real time regulation of efficient driving of high speed trains based on a genetic algorithm and a fuzzy model of manual driving," *Engineering Applications of Artificial Intelligence*, vol. 29, pp. 79–92, 2014.

[18] S. A. Zargari, S. Z. Siabil, A. H. Alavi, and A. H. Gandomi, "A computational intelligence-based approach for short-term traffic flow prediction," *Expert Systems*, vol. 29, no. 2, pp. 124–142, 2012.

[19] N. Zhang, Y. Zhang, and H. Lu, "Seasonal autoregressive integrated moving average and support vector machine models: prediction of short-term traffic flow on freeways," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2215, pp. 85–92, 2011.

[20] W.-C. Hong, "Application of seasonal svr with chaotic immune algorithm in traffic flow forecasting," *Neural Computing and Applications*, vol. 21, no. 3, pp. 583–593, 2012.

[21] E. Vlahogianni and M. Karlaftis, "Testing and comparing neural network and statistical approaches for predicting transportation time series," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2399, pp. 9–22, 2013.

[22] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: Where we are and where we're going," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 3–19, 2014.

[23] J. Wang and Q. Shi, "Short-term traffic speed forecasting hybrid model based on chaos-wavelet analysis-support vector machine theory," *Transportation Research Part C: Emerging Technologies*, vol. 27, pp. 219–232, 2013.

[24] P. Posawang, S. Phosaard, W. Polnigongit, and W. Pattara-Atikom, "Perception-based road traffic congestion classification using neural networks and decision tree," *Lecture Notes in Electrical Engineering*, vol. 60, pp. 237–248, 2010.

[25] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[26] E. Osaba, F. Diaz, and E. Onieva, "Golden ball: a novel meta-heuristic to solve combinatorial optimization problems based on soccer concepts," *Applied Intelligence*, vol. 41, no. 1, pp. 145–166, 2014.

[27] E. Osaba, E. Onieva, R. Carballedo, F. Diaz, A. Perallos, and X. Zhang, "A multi-crossover and adaptive island based population algorithm for solving routing problems," *Journal of Zhejiang University SCIENCE C*, vol. 14, no. 11, pp. 815–821, 2013.

[28] J. Qiao, N. Yang, and J. Gao, "Two-stage fuzzy logic controller for signalized intersection," *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, vol. 41, no. 1, pp. 178–184, 2011.

[29] V. Bevilacqua, N. Costantino, M. Dotoli, M. Falagario, and F. Sciancalepore, "Strategic design and multi-objective optimisation of distribution networks based on genetic algorithms," *International Journal of Computer Integrated Manufacturing*, vol. 25, no. 12, pp. 1139–1150, 2012.

[30] T.-C. Lu, "Genetic-algorithm-based type reduction algorithm for interval type-2 fuzzy logic controllers," *Engineering Applications of Artificial Intelligence*, vol. 42, pp. 36–44, 2015.

[31] E. Onieva, V. Milanés, J. Villagrá, J. Pérez, and J. Godoy, "Genetic optimization of a vehicle fuzzy decision system for intersections," *Expert Systems with Applications*, vol. 39, no. 18, pp. 13 148–13 157, 2012.

[32] O. Cordón, F. Gomide, F. Herrera, F. Hoffmann, and L. Magdalena, "Ten years of genetic fuzzy systems: Current framework and new trends," *Fuzzy Sets and Systems*, vol. 141, no. 1, pp. 5–31, 2004.

[33] R. Rubinstein, "Optimization of computer simulation models with rare events," *European Journal of Operational Research*, vol. 99, no. 1, pp. 89–112, 1997.

[34] ——, "The cross-entropy method for combinatorial and continuous optimization," *Methodology and computing in applied probability*, vol. 1, no. 2, pp. 127–190, 1999.

[35] R. E. Haber, R. M. del Toro, and A. Gajate, "Optimal fuzzy control system using the cross-entropy method. a case study of a drilling process," *Information Sciences*, vol. 180, no. 14, pp. 2777–2792, 2010.

[36] M. Xia and Z. Xu, "Entropy/cross entropy-based group decision making under intuitionistic fuzzy environment," *Information Fusion*, vol. 13, no. 1, pp. 31–47, 2012.

[37] L. A. Zadeh, "Fuzzy sets," *Information and control*, vol. 8, no. 3, pp. 338–353, 1965.

[38] J. Ropero, C. León, A. Carrasco, A. Gómez, and O. Rivera, "Fuzzy logic applications for knowledge discovery: A survey," *International Journal of Advancements in Computing Technology*, vol. 3, no. 6, pp. 187–198, 2011.

[39] E. Onieva, J. Alonso, J. Pérez, V. Milanés, and T. De Pedro, "Autonomous car fuzzy control modeled by iterative genetic algorithms," 2009, pp. 1615–1620.

[40] R. Ghorbani, E. Bibeau, and S. Filizadeh, "On conversion of hybrid electric vehicles to plug-in," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 4, pp. 2016–2020, 2010.

[41] J. Iglesias, P. Angelov, A. Ledezma, and A. Sanchis, "Human activity recognition based on evolving fuzzy systems," *International Journal of Neural Systems*, vol. 20, no. 5, pp. 355–364, 2010.

[42] M. Awan and M. Awais, "Predicting weather events using fuzzy rule based system," *Applied Soft Computing Journal*, vol. 11, no. 1, pp. 56–63, 2011.

[43] A. Fernández, M. del Jesus, and F. Herrera, "Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets," *International Journal of Approximate Reasoning*, vol. 50, no. 3, pp. 561–577, 2009.

[44] A. Benítez and J. Casillas, "Multi-objective genetic learning of serial hierarchical fuzzy systems for large-scale problems," *Soft Computing*, vol. 17, no. 1, pp. 165–194, 2013.

[45] R. Alcalá, J. Alcalá-Fdez, and F. Herrera, "A proposal for the genetic lateral tuning of linguistic fuzzy systems and its interaction with rule selection," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 4, pp. 616–635, 2007.

[46] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance," *Climate research*, vol. 30, no. 1, p. 79, 2005.

[47] D. E. Goldberg and K. Deb, "A comparative analysis of selection schemes used in genetic algorithms," *Foundations of genetic algorithms*, vol. 1, pp. 69–93, 1991.

[48] K. Deep and H. M. Adane, "New variations of order crossover for travelling salesman problem," *IJCOPI*, vol. 2, no. 1, pp. 2–13, 2011.

[49] L. J. Eshelman and J. D. Schaffer, "Real-coded genetic algorithms and interval-schemata," 1992.

[50] F. Herrera, M. Lozano, and J. Verdegay, "Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis," *Artificial Intelligence Review*, vol. 12, no. 4, pp. 265–319, 1998.

[51] P. Larraniaga, C. Kuijpers, R. Murga, I. Inza, and S. Dizdarevic, "Genetic algorithms for the travelling salesman problem: A review of representations and operators," *Artificial Intelligence Review*, vol. 13, no. 2, pp. 129–170, 1999.

[52] D. Schlierkamp-Voosen and H. Mühlenbein, "Predictive models for the breeder genetic algorithm," *Evolutionary Computation*, vol. 1, no. 1, pp. 25–49, 1993.

[53] L. M. C. Skycomp, Inc. in association with Whytney Bailey, "Major highway performance ratings and bottleneck inventory," *State of Maryland, Spring 2008*, 2009.

[54] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Information Sciences*, vol. 250, pp. 113–141, 2013.
[55] J. M. Keller, M. R. Gray, and J. A. Givens, "A fuzzy k-nearest neighbor algorithm," *IEEE Transactions on Systems, Man and Cybernetics*, no. 4, pp. 580–585, 1985.
[56] O. Duru, "The role of predictions in transport policy making and the forecasting profession: Misconceptions, illusions and cognitive bias," *Illusions and Cognitive Bias (October 1)*, 2013.
[57] P.-T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Annals of operations research*, vol. 134, no. 1, pp. 19–67, 2005.
[58] R. Hyndman and A. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, no. 4, pp. 679–688, 2006.
[59] S. Makridakis and M. Hibon, "The m3-competition: Results, conclusions and implications," *International Journal of Forecasting*, vol. 16, no. 4, pp. 451–476, 2000.
[60] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *Journal of Multiple-Valued Logic and Soft Computing*, vol. 17, no. 2-3, pp. 255–287, 2011.
[61] J. R. Quinlan, *C4. 5: programs for machine learning*. Elsevier, 2014.
[62] T. Sousa, A. Silva, and A. Neves, "Particle swarm based data mining algorithms for classification tasks," *Parallel Computing*, vol. 30, no. 5, pp. 767–783, 2004.
[63] E. G. Mansoori, M. J. Zolghadri, and S. D. Katebi, "Sgerd: A steady-state genetic algorithm for extracting fuzzy classification rules from data," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 4, pp. 1061–1071, 2008.

Pedro Lopez-Garcia Pedro Lopez-Garcia was born in Almeria, Spain, in 1991. He obtained his BSc in Technical Engineer in Computer Systems at the University of Almeria, Almeria, Spain (2012), a MSc in Soft Computing and Intelligent Systems at the University of Granada, Granada, Spain (2013) and a MSc in Advanced Artificial Intelligent in National University of Distance Education (UNED), Spain (2014). He is currently a PhD student in the field of Artificial Intelligent applied to mobility in DeustoTech Mobility research team in Deusto Institute of Techonology. His main research interest are: Fuzzy Logic, Metaheuristics, Soft Computing, Intelligent Transportation Systems among others.

Enrique Onieva Enrique Onieva was born in Priego de Cordoba, Spain, in 1983. He received the B.E., M.E. and PhD degrees in computer science with specialization in Soft Computing and Intelligent Systems from the University of Granada, Spain, in 2006, 2008 and 2011, respectively. From 2007 to 2012, he was with the AUTOPIA Program at the Centre for Automation and Robotics, Madrid. In 2012 he was with the Models of Decision and Optimization group from the University of Granada. From the beginning of 2013, he is with the Mobility Unit at the Deusto Insitute of Technology (DeustoTech), where he carries out cutting-edge research in the application of soft computing techniques to the field of intelligent transportation systems. His research interest is based on the application of Soft Computing Techniques to Intelligent Transportation Systems, including fuzzy-logic based decision and control and evolutionary optimization.

Osaba Eneko Osaba is a PhD student at the Deustotech research center at the University of Deusto. He holds a B.Sc. in Computer Science since 2010 by the University of Deusto, and he received in 2011 the title of Master Degree in Development and Integration of Software Solutions, having studied at the same university. At the same time, in September 2014 he began his career as a lecturer, teaching statistics at the University of Deusto. His doctoral thesis is focused on artificial intelligence, specifically in the field of combinatorial optimization, studying and developing heuristics and meta-heuristics solving routing problems. During the develop of his thesis Eneko made a 3-months-stay at Middlesex University (London). Since 2010, he has participated in several research projects, and has also contributed to scientific production, publishing several papers in international conferences and journals. In addition, Eneko has participated as program committee member of some conferences as GECCO or HAIS, and he usually he acts as a reviewer in conferences and journals such as Soft Computing, Evolving Systems and Computers in Industry.

Antonio D. Masegosa Antonio D. Masegosa currently works at the Mobility Unit of the Deusto Institute of Technology as IKERBASQUE Research Fellow. He took his University degree in Computer Engineering in 2005 and his PhD in Computer Sciences in 2010, both from the University of Granada, Spain. From June 2010 to November 2014 he was a post-doc researcher at the Research Center for ICT of the University of Granada. He has published three books and more than 20 papers in leading scientific journals and in both international and national conferences. He is member of the program committee of international conferences as IEEE CEC, ECAL or NICSO. He has served as reviewer in international journals as Information Sciences, NeuroComputing and Memetic Computing and international conferences as GECCO. His main research interests are: Intelligent Systems, Soft Computing, Cooperative Hybrid Metaheuristics, Dynamic Optimization Problems, Fuzzy Systems, and Intelligent Transportation Systems among others.

Asier Perallos Dr. Asier Perallos holds a PhD, MSc and BSc in Computer Engineering from the University of Deusto (Spain). Since 1999 he has been working as a lecturer in the Faculty of Engineering at the University of Deusto, being now accredited by Spanish Government as Associate Professor. His teaching focuses on software design and distributed systems, having taught several BSc, MSc and PhD courses. He is currently the Head of the Computer Engineering Department and in the past has been the director of several MSc in Software Engineering. He is also Head Researcher at the DeustoTech Mobility Unit at the Deusto Institute of Technology (DeustoTech), where he coordinates the research activities of around 25 researchers. This research unit promotes the application of ICT to address smarter transport and mobility. In particular, Perallos' research background is focused on telematic systems, vehicular communication middleware and intelligent transportation systems. He has over a decade of experience in R&D management, with tens of projects and technology transfer actions led, more than 25 JCR indexed publications and more than 50 other contributions in the area of intelligent transport systems and 2 PhD dissertations supervised.