Short-term traffic congestion forecasting using hybrid metaheuristics and rule-based methods: A comparative study

Pedro Lopez-Garcia[1], Eneko Osaba[1], Enrique Onieva[1,2], Antonio D. Masegosa[1,2,3], and Asier Perallos[1,2]

[1] DeustoTech-Fundacin Deusto, Deusto Foundation, 48007, Bilbao, Spain.
[2] Faculty of Engineering, University of Deusto, 48007, Bilbao, Spain.
[3] IKERBASQUE, Basque Foundation for Science, 48011, Bilbao, Spain.
e-mail: {p.lopez, e.osaba, enrique.onieva, ad.masegosa, perallos}@deusto.es

Abstract. In this paper, a comparative study between a hybrid technique that combines a Genetic Algorithm with a Cross Entropy method to optimize Fuzzy Rule-Based Systems, and literature techniques is presented. These techniques are applied to traffic congestion datasets in order to determine their performance in this area. Different types of datasets have been chosen. The used time horizons are 5, 15 and 30 minutes. Results show that the hybrid technique improves those results obtained by the techniques of the state of the art. In this way, the performed experimentation shows the competitiveness of the proposal in this area of application.

Keywords: genetic algorithms, cross entropy, classification, machine learning, hybrid optimization, fuzzy rule-based systems, intelligent transportation systems

1 Introduction

According to the Eurobarometer [2], road congestions are one of the problems that citizenship are more worried about regarding road transport. Therefore, traffic congestion prediction is a fundamental issue in the field of Intelligent Transportation Systems (ITSs). If congestion is predicted successfully, it could help to take decisions can result in noise reduction and energy savings. Also, it could increase the effectiveness and the performance of transport systems, and lead to savings in public infrastructure.

While two of the most frequently used methods for this task in the last decade are the Kalman Filter and the Autoregressive Integrated Moving Average (ARIMA), other alternatives have been developed in recent years. Among them, Soft Computing techniques as Support Vector Machines (SVM), Neural Networks (NN), Genetic Algorithms (GA) or Fuzzy Rule-Based Systems (FRBS) have been used in traffic forecasting tasks in particular [22], and in ITS field in general [17, 18].

The research developed in this paper aim at extending the analysis done in [13] by comparing the method presented in that paper with state of the art classification algorithms in order to evaluate the competitiveness of our proposal versus high-performance methods for traffic congestion prediction over different scenarios in terms of data available. In this way, we intend to offer an analysis about the advantages of our method depending on the characteristics of the data at hand. For this purpose, a hybrid algorithm which combines a Genetic Algorithm and a Cross Entropy, called GACE, is used with the aim of optimizing the different parts of a hierarchy of Fuzzy Rule-Based Systems (FRBS). This hierarchy was applied for predicting the congestion in several points and sectors of a road. The article is structured as follows. In Section 2, we give some background information about GA, cross entropy and hybrid metaheuristics. The definition of the proposed algorithm is detailed in Section 3. Then, the experimentation, the datasets used, and the comparative study are described in Section 4. Finally, conclusions are pointed out in Section 5.

2 Background

In this section, a brief summary about recent literature related to the different algorithms that compose the proposal is done, i.e. GA (Section 2.1) and Cross Entropy (CE) (Section 2.2). Besides, some examples about hybrid algorithms are shown in Section 2.3.

2.1 Genetic Algorithm

GA is a well-known metaheuristic introduced by Holland in [8]. Their objective is to mimic some of the processes observed in natural evolution. They have widely used since their proposal. For example, in [23], GA optimizes a fuzzy controller in order to improve the regenerative braking energy recovery rate of an electric vehicle. In that study, several road conditions are simulated and analysed. Results indicated that the use of a fuzzy logic control strategy based on the GA could improve the energy recovery and prolong the endurance mileage of the electric vehicle. Another example can be found in [6], where a GA is introduced to optimize the signal cycle length, split ratio, and phase difference for a traffic signal control in a district of Shangai. Interested readers are referred to [11] and [10] for extensive reviews of GAs in the literature.

2.2 Cross Entropy

CE method is an adaptive method proposed by Rubinstein [20] for rare-event probabilities and combinatorial optimization. The technique is divided in three phases:

1. Generate random samples from a normal distribution with given mean and standard deviation.

2. Select the best individuals from the previous samples.
3. Update mean and standard deviation according to the best individuals.

The aim of this algorithm is to focus the search in the area that contains the best samples found. CE has been used in the last years in different fields.

In the field of ITS, CE has been used in [19] to solve a Vehicle Routing Problem with weight coefficients and stochastic demands. In [16], CE is applied to optimize fuzzy logic controllers. These controllers are designed to command the orientation of an unmanned aerial vehicle to modify its trajectory for avoiding collisions.

2.3 Hybrid Algorithms

The hybridization of different metaheuristics is an important topic in the literature [7]. In hybridization, two or more techniques are combined in order to create synergies among them and cover the lacks that they can have separately. The combination is made with the aim of obtaining a good performance, improving the results obtained by the techniques for its own. Metaheuristics and their hybridizations have been widely used in the literature for different problems. In [1], a GA with a restricted search is hybridized with Extreme Learning Machine for reducing traffic noise and improve ITS. A method to design an intelligent suspension system with the objective of overcoming the trade-off barrier using the smallest actuator is presented in [9]. A hybrid genetic algorithm is used to tune the system, and performed good scenarios previously used in literature.

3 GACE: Genetic Algorithm with Cross Entropy

The method used in this study is an hybridization of the two methods described in Section 2, and it has been called GACE (Genetic Algorithm with Cross Entropy). The lack of exploitation ability in population-based algorithms as GA and the high probability of CE to become stuck in local optima, especially when a high learning rate is assigned, motivated the creation of this hybridization. The proposed technique tries to cover the lack of exploitation of GA using CE, focusing the search in the promising areas. On the other hand, the low exploration balance of CE is compensated by GA. Therefore, GACE is created with the aim of taking advantage of the exploration ability of GA and the exploitation ability of the CE. GACE works as follows: first of all, the initial population is created randomly with a given number of individuals $Size_{POP}$. The population is divided into two sub-populations in each generation. These sub-populations are POP_{GA} with $Size_{GA}$ individuals and POP_{CE} with a size of $Size_{CE}$ individuals.

In this application of the algorithm, $Size_{GA}$ and $Size_{POP}$ are established by the user, while $Size_{CE}$ is calculated as $Size_{CE} = Size_{POP} - Size_{GA}$. While the individuals that form POP_{GA} are chosen by the given selection method, POP_{CE} individuals are the best individuals in the current population POP_t. After the creation of both sub-populations, each one is used in a different way:

– In POP_{GA}, GA operators are applied in order to create $Size_{GA}$ new individuals with p_c and p_m as crossover and mutation probability respectively.
– In case of POP_{CE}, a total of $Size_{CE}$ individuals are randomly generated applying a normal distribution $\mathcal{N}(\overline{M}, S)$, where $\overline{M}$ is the mean and S the standard deviation. Both parameters, $\overline{M}$ and S, are updated employing the CE method with the n_{up} best individuals of POP_{CE}, and a parameter called Learning Rate L_r, used to update $\overline{M}$ and S during the execution of the algorithm with the means and deviations of the new selected samples.

After both algorithms are applied to its sub-population, POP_{t+1} is created using the offsprings generated in the last two steps, i.e. POP_{t+1} is formed by the GA_{size} individuals generated with POP_{GA} and the $Size_{CE}$ individuals created using CE method. Therefore, the total population size is the sum of the number of individuals in each sub-populations, i.e. $POP_{size} = Size_{GA} + Size_{CE}$. In addition, elitism is applied, i.e. if the best individual found so far is not part of the actual population, it is inserted on it, replacing the worst individual. The whole process is presented in Algorithm 1.

Data: $Size_{POP}$, $Size_{GA}$, p_c, p_m, L_r, n_{up}, T_{max}
Result: *Best individual found*
$Size_{CE} \leftarrow Size_{POP} - Size_{GA}$
$t \leftarrow 0$
$POP_0 \leftarrow$ Initialize($Size_{POP}$)
$\overline{M} \leftarrow$ Initialize Means vector
$S \leftarrow$ Initialize Standard Deviation vector
Evaluate POP_0
while $t < T_{max}$ **do**
    $POP_{GA} \leftarrow$ SelectionOperator($POP_t, Size_{GA}$)
    $POP_{CE} \leftarrow$ SelectBestSamples($POP_t, Size_{CE}$)
    $Offspring_{GA} \leftarrow$ Crossover(POP_{GA}, p_c)
    $Offspring_{GA} \leftarrow$ Mutation($Offspring_{GA}$,p_m)
    $Offspring_{CE} \leftarrow$ Generate($POP_{CE}, Size_{CE}, \overline{M}, S$)
    $\overline{M} \leftarrow UpdateMeans(L_r, \overline{M}, Offspring_{CE}, n_{up})$
    $S \leftarrow UpdateDeviation(L_r, S, Offspring_{CE}, n_{up})$
    $POP_{t+1} \leftarrow Offspring_{GA} \bigcup Offspring_{CE}$
    Evaluate POP_{t+1}
    Add the best individual found to POP_{t+1} if it is not in the population
    $t \leftarrow t + 1$
end

Algorithm 1: Pseudocode of the workflow followed by the proposed method GACE

Focusing on congestion forecasting, the algorithm was used to optimize the different parts of a hierarchical FRBS. In this work, we extend the experimentation done in [13] by comparing our proposal with state of the art classification algorithms for traffic congestion prediction over different scenarios in terms of data available. In this way, we intend to offer an analysis about the advantages of our method depending on the characteristics of the data at hand.

3.1 Application of the proposal to optimize FRBS hierarchy

In this section, we explain the structure and the parts used for optimizing the Hierarchical Fuzzy Rule-Based Systems (HFRBS). First of all, the HFRBS counts with three parts:

1. Hierarchy ($C_{hierarchy}$): the subset of variables selected to be used by the HFRBS and the order in which they are included in the system. An ending character is included in this part. After this point, no more variables are used.
2. Membership Functions (C_{label}): codification of the location of the labels used to encode each variable for each FRBS in the hierarchy.
3. Rules (C_{rules}): positions of the singletons used as consequence of the rule bases of the FRBSs in the hierarchy.

These three parts are optimized by the proposed algorithm. These variables are used by the HFRBS as showed in Figure 1. As it can be seen in this figure, systems are structured in a parallel way. The reason to use this organization is to consider each variable at the same time and with similar relevance. Also, a constraint is imposed: each FRBS has only two inputs. For more details, the interested reader is referred to [13].

4 Experimentation and Results

In this section, information about the datasets used in the experimentation is showed in Section 4.1, which also includes the configuration used by the algorithm in this experimentation. Besides, the results of the performend tests are showed in Section 4.2.

4.1 Datasets and Configuration

With the aim of forecasting traffic congestion in a road, different types of datasets are considered in the present experimentation. These datasets represent a 9-km sector from the I5 highway in California, and they can be downloaded from the link [4]. The data is taken from the platform PeMS[5]. There are a total of 13 sensors in the road which take three values: flow (number of vehicles), occupancy (percentage of time during which the sensor was switched on) and speed (in miles per hour). Besides these values, a congestion variable is added using the intervals applied in the previous work and showed in Table 1. Density is also calculated using the values of the flow and the speed: $density = flow/speed$. In addition,

[4] https://www.researchgate.net/publication/287771448_I5_Congestion_Datasets_GACE2015
[5] http://pems.dot.ca.gov/

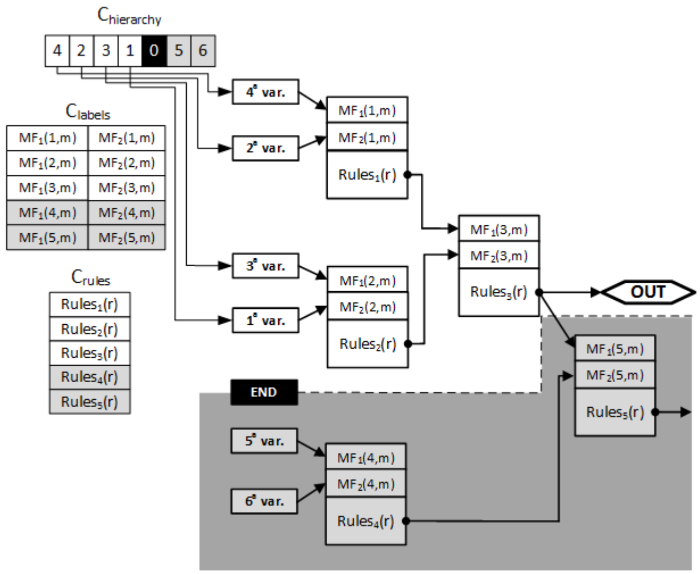

Fig. 1. Example of a hierarchy with six variables. The variables after the ending character (remarked as 0) have not count for its use in the hierarchy.

there are a total of 8 ramps (4 in-ramps and 4 out-ramps) in the section of the road. These ramps provide flow values, which is the number of vehicles that get in (or out) the road. Following the levels showed in the table, congestion can take Free, Slight, Moderate or Severe values. These values are presented as universal units of the metric system (km). The different types of congestion are replaced by numbers in order to provide proper forecasting measures. Therefore, congestion states will be changed to use them for the calculation of the error: $\{$Free $= 1$, Slight $= 2$, Moderate $= 3$ and Severe $= 4\}$.

Table 1. 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	

Figure 2 shows a schema with the section of road used as well as the sensors located on it. Two types of datasets have been used: Point (PD) and Sector (SD) datasets. In the first one, the point of interest (S7), that is, the point in which the congestion is predicted, is found in the middle of the road, while in the second one, the point of interest is the complete road segment. A total of 49 variables are contained in PD and SD datasets (39 variables for the sensors in the road, 8 variables in ramps, and one congestion variable). The time-horizons used are 5, 15, and 30 minutes, and they are indicated as a subscript in each dataset.

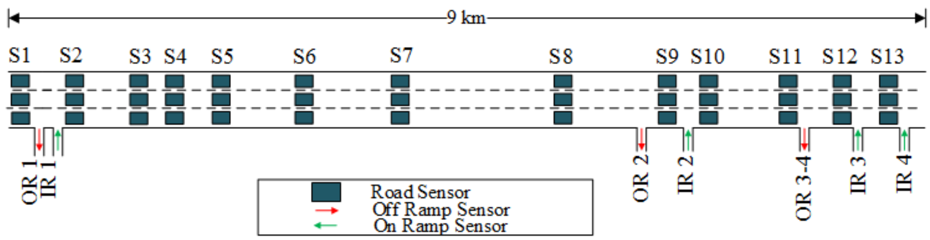

Fig. 2. Sector of the I5 highway used in this work. Sensors are denoted by S. Off and On Ramps are denoted by OR and IR respectively

For the GA part of the proposal, binary tournament is used as the selection operator. In case of crossover operator, BLX-α has been applied to C_{labels} and C_{rules} parts, while a variant of the order crossover has been chosen for $C_{hierarchy}$. In case of mutation operators, BGA has been used for C_{labels} and C_{rules} and a swap mutation operator has been applied to $C_{hierarchy}$.

About the configuration used by GACE, the values are summarized in Table 2. The number of labels and rules of each system of the hierarchy are defined as six (three per input variable) and nine respectively. Besides, a 10-fold cross validation method for testing the model is used.

Parameter	Value
T_{max}	500
$Size_{pop}$	50
$Size_{GA}$	{35,40,45}
p_c	0.8
p_m	0.2
L_r	0.7
n_{up}	$Size_{CE}$

Table 2. Values of the parameters used in the experimentation

4.2 Comparative and Results

In this experimentation, we compare the combination of GACE and HFRBSs for congestion prediction, with six other state-of-the-art data mining techniques in order to get more insights about the performance and competitiveness of this method. KEEL software [6] has been used for the execution of the algorithms with their default values:

- **Adaboost** [3] is a boosting algorithm, which repeatedly invokes a learning algorithm to successively generate a committee of simple, low-quality fuzzy classifiers. Each time these classifiers are added to the compound one, the weights of the examples in training set are changed, and a voting strength, which depends on its accuracy, is given to the classifier. In this algorithm, each of the weak hypothesis is a fuzzy rule extracted from data.
- **Grammar-based Genetic Programming algorithm** (GP) [21] is used to learn a fuzzy classifier by means of learning fuzzy rules throught Genetic Programming algorithms.
- **INNER** [14] tries to extract a small set of suitable rules to represent the training set, achieving an aceptable accuracy.
- **PART** [4] is a classification model by covering rules based on decision trees. The aim is to determine a decision list of rules that predicts correctly the value of the target attribute. PART is based on C4.5 algorithm, due to in each iteration, a partial C4.5 Tree is generated and its best rule is extracted. The method ends when all the examples are covered.
- **Real Encoding - Particle Swarm Optimization** (REPSO) [12] is a PSO-based classifier that perform a classification task by means of a PSO algorithm. It uses a real encoding approach.
- **Tree Analysis with Randomly Generated and Evolved Trees** (Target) [5] is a hybrid decision tree with the aim of obtain a forest of rules that better suits the training data by means of a GA search.

The error used for the comparative is the symmetric mean absolute percentage error (sMAPE) [15]. The calculation of sMAPE is showed in Eq. 1, where $\bar{Y}$ is the expected value, Y the predicted one, and n is the number of examples. The aim of using SMAPE is to take into account that the error between two close types of congestion is smaller than the error between two values of congestion . For example, the error between $Y = Free$ and $\bar{Y} = Severe$ is greater than the error between $Y = Free$ and $\bar{Y} = Slight$.

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

Experimentation is made with these techniques and the best configurations obtained in [13], i.e. $GACE_{45-5}$, $GACE_{40-10}$ and $GACE_{35-15}$. Subscript indicates the size of both subpopulations, i.e. $GACE_{45-5}$ means that $Size_{GA} = 45$ while $Size_{CE} = 5$. The results obtained and the comparison between them are showed in Table 3. Bold values indicate the two best techniques in each dataset.

Results show that GACE obtains one of the two best values in all the datasets, and the two best values so far in 4 out 6. The best configuration, $GACE_{45-5}$, achieves one of the best sMAPE value in all datasets. $INNER$ and $Target$, together with GP are the techniques that obtain an error closer to the proposal.

[6] http://sci2s.ugr.es/keel/

Dataset	GACE $45-5$	GACE $40-10$	GACE $35-15$	AdaBoost	GP	INNER	PART	REPSO	Target
PD_5	**0.009**	**0.009**	**0.010**	0.042	0.014	0.028	0.032	0.013	0.016
PD_{15}	**0.013**	**0.013**	**0.013**	0.042	0.017	0.027	0.042	0.020	**0.016**
PD_{30}	**0.017**	**0.016**	**0.016**	0.042	0.020	0.036	0.043	0.022	0.021
SD_5	**0.149**	0.184	0.195	0.433	0.333	**0.169**	0.409	0.386	0.176
SD_{15}	**0.130**	0.182	**0.169**	0.386	0.2539	0.227	0.392	0.386	0.397
SD_{30}	**0.123**	**0.140**	0.169	0.392	0.409	0.205	0.394	0.387	0.404

Table 3. Average errors of the different techniques in each dataset

5 Conclusions

In this paper, we aimed at studying more in depth the performance of Fuzzy Rule-Based Systems evolved by GACE to predict traffic congestion. The goal of this study is to check the competitiveness of this approach when it is used for this purpose. To this end, we compared this method versus other six literature techniques with data obtained in a 9-km stretch of highway. In this new comparative, the best configurations obtained previously have been used. The results confirm the good performance of GACE and HFRBS against well-known techniques of the state of the art in this kind of problems. In future works, other datasets are planned to be used. Besides, new time horizons can be interesting to use. Also, a exhaustive study about computational time of the proposal and the literature methods would be interesting to make. Finally, congestion datasets from other sources could be created, or found, in order to prove the performance of different metaheuristics in this theme.

Acknowledgment

Authors work was supported by TIMON Project. This project has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 636220.

References

1. E. Alexandre, L. Cuadra, S. Salcedo-Sanz, A. Pastor-Snchez, and C. Casanova-Mateo. Hybridizing extreme learning machines and genetic algorithms to select acoustic features in vehicle classification applications. *Neurocomputing*, 152:58–68, 2015.
2. European Commission. Special eurobarometer 422a, quality of transport, 2014.
3. M.J. del Jesus, F. Hoffmann, L. Junco, and L. Sánchez. Induction of fuzzy-rule-based classifiers with evolutionary boosting algorithms. *IEEE Transactions on Fuzzy Systems*, 12(3):296–308, 2004.
4. E. Frank and I.H. Witten. Generating accurate rule sets without global optimization. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 144–151, 1998.

5. J.B. Gray and G. Fan. Classification tree analysis using target. *Computational Statistics and Data Analysis*, 52(3):1362–1372, 2008.
6. Y. Han, B. Xing, J. Yao, and J. Liu. Optimal model of regional traffic signal control under mixed traffic flow condition. *Jiaotong Yunshu Gongcheng Xuebao/Journal of Traffic and Transportation Engineering*, 15(1):119–126, 2015.
7. S.A. Hernández, G. Leguizamón, and E. Mezura-Montes. Hybridization of differential evolution using hill climbing to solve constrained optimization problems. *Revista Iberoamericana de Inteligencia Artificial*, 16(52):3–15, 2013.
8. J. H. Holland. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. Univ. of Michigan Press, 1975.
9. S. Kanarachos and A. Kanarachos. Intelligent road adaptive suspension system design using an experts' based hybrid genetic algorithm. *Expert Systems with Applications*, 42(21):8232–8242, 2015.
10. S. Karakatic and V. Podgorelec. A survey of genetic algorithms for solving multi depot vehicle routing problem. *Applied Soft Computing*, 27:519–532, 2015.
11. T.Y. Lim. Structured population genetic algorithms: A literature survey. *Artificial Intelligence Review*, 41(3):385–399, 2014.
12. Yu Liu, Zheng Qin, Zhewen Shi, and Junying Chen. Rule discovery with particle swarm optimization. In *Content Computing*, pages 291–296. Springer, 2004.
13. P. Lopez-Garcia, E. Onieva, E. Osaba, A. D. Masegosa, and A. Perallos. A hybrid method for short-term traffic congestion forecasting using genetic algorithms and cross entropy. *IEEE Transactions on Intelligent Transportation Systems*, 17(2):557–569, 2016.
14. O. Luaces. Inflating examples to obtain rules. *International Journal of Intelligent Systems*, 18:1113–1143, 2003.
15. S. Makridakis and M. Hibon. The m3-competition: Results, conclusions and implications. *International Journal of Forecasting*, 16(4):451–476, 2000.
16. M.A. Olivares-Mendez, C. Fu, S. Kannan, H. Voos, and P. Campoy. Using the cross-entropy method for control optimization: A case study of see-and-avoid on unmanned aerial vehicles. pages 1183–1189, 2014.
17. E. Onieva, V. Milanes, J. Villagra, J. Perez, and J. Godoy. Genetic optimization of a vehicle fuzzy decision system for intersections. *Expert Systems with Applications*, 39(18):13148–13157, 2012.
18. E. Osaba, F. Diaz, and E. Onieva. Golden ball: a novel meta-heuristic to solve combinatorial optimization problems based on soccer concepts. *Applied Intelligence*, 41(1):145–166, 2014.
19. Y. Qiu. Vehicle routing problem with weight coefficients and stochastic demands based on the cross-entropy method. pages 159–162, 2009.
20. R.Y. Rubinstein. Optimization of computer simulation models with rare events. *European Journal of Operational Research*, 99(1):89–112, 1997.
21. L. Sánchez, I. Couso, and J.A. Corrales. Combining gp operators with sa search to evolve fuzzy rule based classifiers. *Information Sciences*, 136(1-4):175–192, 2001.
22. Eleni Vlahogianni and Matthew Karlaftis. Testing and comparing neural network and statistical approaches for predicting transportation time series. *Transportation Research Record: Journal of the Transportation Research Board*, (2399):9–22, 2013.
23. M. Zhou, S. Bi, C. Dong, and C. He. Regenerative braking system for electric vehicles based on genetic algorithm fuzzy logic control. *ICIC Express Letters, Part B: Applications*, 5(3):689–695, 2014.