Hierarchical Fuzzy Rule-Based System Optimized with Genetic Algorithms for Short Term Traffic Congestion Prediction

Xiao Zhang[a,b], Enrique Onieva[a], Asier Perallos[a], Eneko Osaba[a], Victor C.S. Lee[b]

[a]*Deusto Institute of Technology, University of Deusto, Bilbao 48007, Spain.*
email: {xiao.zhang, enrique.onieva, perallos, e.osaba}@deusto.es
[b]*Department of Computer Science, City University of Hong Kong, Hong Kong.*
email: {xiao.zhang@my.cityu.edu.hk, csvlee@cityu.edu.hk}

Abstract

Taking practical and effective traffic prediction and control measures to ease highway traffic congestion is a significant issue in the research field of Intelligent Transportation Systems (ITS). This paper develops a Hierarchical Fuzzy Rule-Based System (HFRBS) optimized by Genetic Algorithms (GAs) to develop an accurate and robust traffic congestion prediction system employing a large number of input variables. The proposed system reduces the size of the involved input variables and rule base while maintaining a high degree of accuracy. To achieve this, a hierarchical structure composed of FRBSs is optimized by a Steady-State GA, which combines variable selection and ranking, lateral tuning of the membership functions, and optimization of the rule base. We test the capability of the proposed approach on short term traffic congestion problems, as well as on benchmark datasets, and compare the outcomes with representative algorithms from the literature in inferring fuzzy rules, confirming the effectiveness of the proposed approach.

Keywords: Intelligent transportation systems; Traffic congestion prediction; Hierarchical fuzzy rule-based systems; Genetic Algorithms; Congestion Forecasting; Genetic Fuzzy Systems

1. Introduction

Nowadays, traffic congestion on highways is a global issue: in fact, almost all of nations suffer from it, to varying degrees. Among other problems, it causes business losses due to increased travel time, requires increases in public infrastructure investment, and threatens urban environmental quality due to the extra emissions involved. It has been estimated by the Transport White Paper (March 2011) that the costs caused by congestion will increase by approximately 50% by 2050. For these reasons, traffic control to ease highway traffic congestion is a significant issue in the research field of Intelligent Transportation Systems (ITS).

Therefore, the prediction of traffic congestion is a vital component of ITS, which aims to influence travel behavior, improve mobility, and save energy. Traffic predictive information can be used either by drivers directly to avert potential traffic blocks, or by traffic management and control systems to ensure traffic flow.

Over the last decades, some of the most used algorithms for traffic forecasting have been based on the Kalman Filter (KF) ([28, 36]) and the Autoregressive Integrated Moving Average (ARIMA) ([2, 54]). Although these methods have improved traffic modeling and prediction, there are still some shortcomings in dealing with predictions. For instance, KF tends to generate overestimation or underestimation that deteriorates the prediction accuracy when the traffic condition undergoes very significant changes; on the other hand, ARIMA mainly targets single variable time-series data, instead of using a richer dataset.

In studies with richer datasets ([17, 43, 47, 50]), traffic flow, occupancy and speed have been found suitable for predicting traffic conditions in the short run. In particular, the results in ([17]) indicate that the traffic flow and occupancy prediction accuracy was higher than the speed. Other studies like ([6, 34]) imply either that predictions based on traffic flow are more reliable through information theory to analyse a subset of real data from Austin, Texas; or that the use of occupancy, which is proportional to density, is a better indicator of traffic condition ([34]). However, there are some other contradictory findings, where the speed was found to be favoured over flow and occupancy, because they are more efficient and meaningful for the users ([43]).

In general, four input variables suitable for capturing and predicting traffic conditions have dominated the literature: mainline flow, occupancy, speed, and ramp flow.

In recent years, several soft computing techniques have received much attention and enabled an encouraging level of performance for traffic prediction purposes. These techniques can be broadly classified into Support Vector Machines (SVM) ([44, 52, 55]), Neural Networks (NN) ([10, 25, 29, 46]), Fuzzy Rule-Based Systems (FRBS) ([16, 60]), and Genetic Algorithms (GA) ([1]). Furthermore, combinations or hybridizations of these have brought about promising results, such as the genetic optimization of NN ([51]). There are some other studies like ([41, 61]).

Among these techniques, the large number of variables involved may produce inaccurate prediction on the part of SVMs, due to the choice of the appropriate kernel function for the practical problem [52]. NN has obtained relatively better performance at traffic forecasting and has attracted more attention. However, the local optimum and generalization ability limit its effectiveness. The reason for this is that the synaptic weights and thresholds of NN are random and time-consuming to initialize ([25]). In addition, their performance mainly lies on the training process, and is especially affected by a large number of data with high quality and defined parameters ([40]).

Fuzzy logic ([57]) is usually adopted to deal with the complexity derived from traffic situations ([35, 37]). It allows processing uncertain information to be simply represented as simple rules, e.g. for the traffic congestion prediction problem. Traffic states, such as speed, occupancy and flow, are grouped into finite categories, such as high, medium, and low. Then, rules are created to relevant traffic states with congestion detection. For example, a rule can be: *if the occupancy is high, speed is low, and flow is medium, then congestion is positive*. Among Fuzzy Logic systems, FRBS is the most representative case, which has been applied in many kinds of complex real-world problems.

GAs are stochastic search algorithms, which are inspired by natural evolution principles of species in nature([26]). GAs have often been used in the literature, since they can deal with extremely complex problems which are hard to solve by traditional methods ([30]). GAs have also been widely applied for FRBS learning and tuning. A number of papers have been dedicated to the automatic generation of the knowledge base of a FRBS ([5]). One of the most successful applications is Genetic FRBS (GFRBS), where a GA is used to learn or tune the components of a FRBS ([13, 37]).

The design of a FRBS is generally a time consuming and complex process. It involves knowledge acquisition, the definition of the controller structure, the rules, and the other parameters. When a traditional FRBS is faced with a large scale problem (i.e., with a high number of input variables), the number of rules increases exponentially while the obtained FRBS is barely accurate or interpretable. Up to now, one of the most important issues in FRBS is how to reduce the size of the rule base involved while maintaining an adequate accuracy. One feasible way to achieve this goal is to arrange the input variables hierarchically, which is known as *hierarchical FRBS* (HFRBS) ([8, 42]). HFRBS consists of a number of low-dimensional fuzzy systems arranged hierarchically. This way, the total number of rules grows only linearly with the number of input variables.

In order to implement these systems, several proposals have been presented ([8, 9, 14, 32, 48]). Some of them identify common parts of the set of rules and create submodules that generate these common parts ([9, 49]). In other schemas, the hierarchical level of each module refers to an increase in the granularity of the variables ([14]). The authors in ([32]) propose an approach of mapping rule-based schemes, referred to as a limpid-hierarchical fuzzy system, which aims at overcoming the problem that the outputs of the intermediate layers do not possess physical meaning. The work presented in ([48]) gives an introduction to hierarchical fuzzy control along with several examples by using evolutionary algorithms. The authors in ([8]) suggest the use of a multi-objective GA to learn HFRBS, especially dedicated to reducing the size of the rules and to improving the interpretability of the system. However, the curse of dimensionality is still an unsolved, and difficult, problem in fuzzy logic control theory ([58]).

This paper is motivated by the circumstance that previous methods have their own advantages and disadvantages, and a particular previous approach may not be able to accurately predict traffic congestion with a high number of variables over the durations of the prediction period. For purpose of achieving more accurate and robust traffic prediction, we here propose a Genetic Hierarchical FRBS (GHFRBS) capable of predicting traffic congestion in multiple prediction horizons.

The original contributions of the proposal are as follows:

- The number of variables generated is small, and the selected variables are distributed in a set of serial hierarchical modules.
- GHFRBS significantly reduces the complexity of the fuzzy rules.

- GHFRBS performs a lateral tuning process of membership functions (MFs) and optimization of the rule bases of the systems in the hierarchy.
- This model allows the system to identify and rank input variables from among a large number of input variables.

A learning process based on a Steady-State GA for extracting the rule base is adopted by the proposed HFRBS. With the use of a hierarchical structure, a smaller number of variables are automatically selected to be the set of serial hierarchical modules. GHFRBS performs a lateral tuning of the data base of the fuzzy systems to improve the accuracy. It allows large scale variables to be treated with low complexity, due to its hierarchical structure and the use of its variable selection and ranking mechanism.

The present paper is structured as follows: Section 2 describes some preliminary concepts and gives the details of the problem to be solved. This section also establishes the problem model with data support. Section 3 explains the implemented genetic learning based hierarchical fuzzy system, Section 4 describes the experimentation carried out and analyses the results. Finally, Section 5 draws some conclusions and outlines further research.

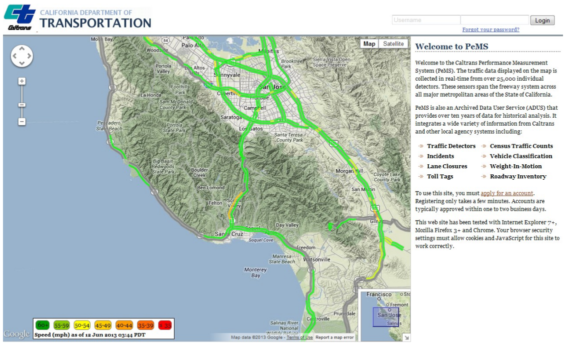
2. Preliminaries

This section introduces the data support of the proposed problem and gives the corresponding model. Moreover, a brief discussion of HFRBS is included.

2.1. Data Support

The data used is accessible on *Performance Measurement System* (PeMS[1]). Figure 1 shows a screen capture of the application, which is composed of a large scale highway data collection, storage, and analysis system ([11]). This database gathers data from over 30,000 miles of highway in the state of California, USA. The information is taken every 30 seconds or 5 minutes, and is used to assemble a history of traffic measurements for every loop detector station in the site. This is publicly available for statistical and academic purposes.

The aggregate detector dataset is the cornerstone of PeMS. This dataset allows the users to examine information from loop detectors at several levels of spatial and temporal aggregation, average over the hour of the day (or days of the week), as well as quantity relations during 5-minute to one-month data points. The real dataset collected from different detectors are highly correlated among themselves. A detailed description of the dataset features is available in ([11]). According to the real system (PeMS), its result indicates, for first time, that vehicles travel at 60 miles/hour when a highway section is operating most efficiently (when there is maximum flow on the highway section). As reported in most of works like CoTEC ([7]), by characterizing highway traffic congestion in several US cities, the traffic state with speed [30-50] miles/hour is categorized as the slight level of congestion. In addition, we rationalize the distribution rate of congestion incidents in the preprocessed datasets through considering the characteristics of the data of PeMS and analyzing the congestion cases in the collected datasets, in order to test our system more effectively. In view of above considerations, a mean speed below 45 miles/hour is considered to be congestion in this study.



Welcome to PeMS

Welcome to the Caltrans Performance Measurement System (PeMS). The traffic data displayed on the map is collected in real-time from over 25,000 individual detectors. These sensors span the freeway system across all major metropolitan areas of the State of California.

PeMS is also an Archived Data User Service (ADUS) that provides over ten years of data for historical analysis. It integrates a wide variety of information from Caltrans and other local agency systems including:

- Traffic Detectors
- Incidents
- Lane Closures
- Toll Tags
- Census Traffic Counts
- Vehicle Classification
- Weight-In-Motion
- Roadway Inventory

To use this site, you must apply for an account. Registering only takes a few minutes. Accounts are typically approved within one to two business days.

This web site has been tested with Internet Explorer 7+, Mozilla Firefox 3+ and Chrome. Your browser security settings must allow cookies and JavaScript for this site to work correctly.

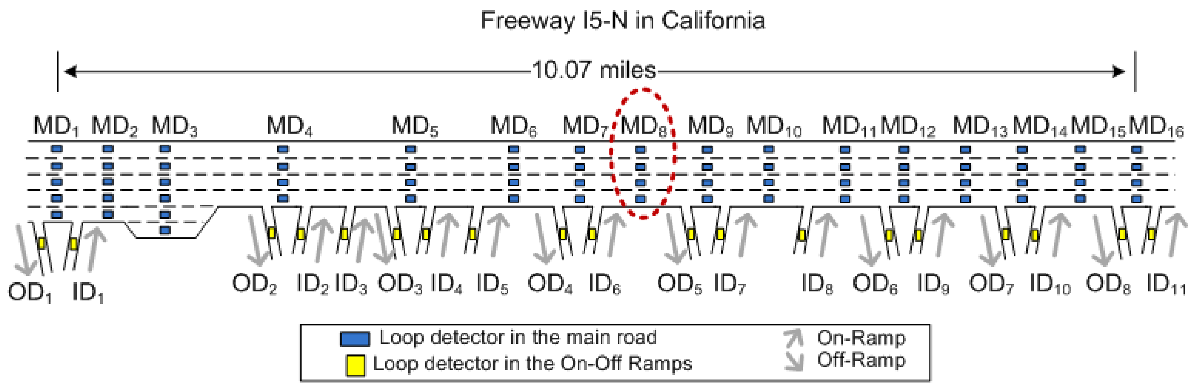
Figure 1: Performance Measurement System in California.

The study was conducted on a segment of the *I-5* highway, in San Diego, California, 10.07 miles long and with changes from four to six lanes on the northbound side. Loop detectors were placed on the highway mainline approximately every one-third of a mile, as well as in each on-ramp and off-ramp segment. The traffic measures collected by the detector stations for every 5 minutes are the average speed, occupancy, and flow (number of vehicles). The data were collected from April 1 06:00 until April 30 22:59 2013.

A schematic graphic of the area site is provided in Figure 2. In this figure, a total of 68 loop detectors are distributed in 16 points along the main road ($MD_i, \{i = 1 \dots 16\}$). In addition, 8 loop detectors are situated in the off-ramps ($OD_i, \{i = 1 \dots 8\}$) and 11 loop detectors in the on-ramps ($ID_i, \{i = 1 \dots 11\}$).

The purpose of this paper is to predict when conges-

[1] http://pems.dot.ca.gov/



Freeway I5-N in California
10.07 miles
MD_1 MD_2 MD_3 MD_4 MD_5 MD_6 MD_7 MD_8 MD_9 MD_{10} MD_{11} MD_{12} MD_{13} MD_{14} MD_{15} MD_{16}
OD_1 ID_1 OD_2 ID_2 ID_3 OD_3 ID_4 ID_5 OD_4 ID_6 OD_5 ID_7 ID_8 OD_6 ID_9 OD_7 ID_{10} OD_8 ID_{11}
Loop detector in the main road
Loop detector in the On-Off Ramps
On-Ramp
Off-Ramp

Figure 2: Graphical representation of the arrangement of the detectors along the highway.

tion is going to occur at point MD_8 (marked in Figure 2) with time horizons of 5, 15 and 30 minutes.

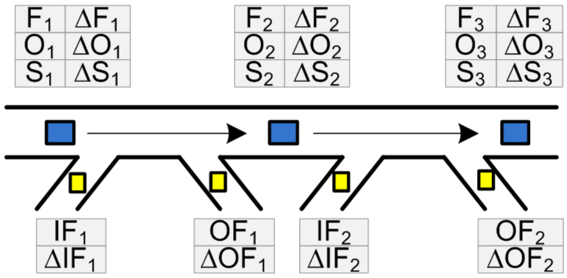
2.2. Data Simplification and Model

At first, it is important to note that loop detectors located in the main road report three different values:

1. $flow$: the number of vehicles per time period of granularity (5 minutes).
2. $occupancy$: the percentage of time that the detector is switched on (%).
3. $speed$: the vehicle speed measured and calculated at the detector (miles/hour).

On the other hand, the loop detectors located at the on-ramps and off-ramps only report the $flow$. In total, 223 variables are collected ($68 \cdot 3 + 11 \cdot 1 + 8 \cdot 1$). In this work, the derivatives of the variables may take on some importance. These derivates are calculated as the difference between two consecutive samples, divided by the time step. This way, finally, 446 variables would be considered.

With the aim of simplifying the number of variables to be processed, the values reported by $MD_{(2...7)}$ and $MD_{(9...11)}$ are obviated. Since the traffic state will be influenced by its Up-stream, and Down-stream point with abnormal traffic state will affect the traffic condition of Up-stream point, one should consider this information when studying the state of congestion at the location of interest. As for prediction purposes, in this study, Upstream, Center and Down-stream traffic measurements at time t will influence, to a certain extent, the point of interest traffic at time $t + h$ minutes approximately. The collection interval of the measures of the traffic is 5 minutes, while the aforementioned most efficient speed on the highway is 60 miles/hour. Therefore, data from points at about 5 miles before and after



F_1 ΔF_1 O_1 ΔO_1 S_1 ΔS_1
F_2 ΔF_2 O_2 ΔO_2 S_2 ΔS_2
F_3 ΔF_3 O_3 ΔO_3 S_3 ΔS_3
IF_1 ΔIF_1
OF_1 ΔOF_1
IF_2 ΔIF_2
OF_2 ΔOF_2

Figure 3: Graphical representation of the highway after simplification of the variables.

MD_8 are used. In addition, values coming from the same point of the road are averaged in order to obtain only 9 variables.

- F_1, F_2 and F_3 represent the averaged $flow$ at points MD_1, MD_8 and MD_{12}, respectively.
- The same procedure is followed with the $speed$ and $occupancy$, obtaining respectively, variables named $\{O_1, O_2, O_3\}$ and $\{S_1, S_2, S_3\}$.

As to the on-ramps and off-ramps, $flow$ values from the ramps before and next to MD_8 are aggregated, yielding $\{IF_1, IF_2, OF_1, OF_2\}$ as the result.

The derivatives of the previous variables are also calculated and included in the dataset. They are denoted by $\{\Delta F_1, \Delta F_2, \ldots \Delta OF_1, \Delta OF_2\}$.

After the simplification process, the 26 variables obtained are illustrated graphically in Figure 3, which can potentially be used to predict the highway congestion at the point of interest at the next time interval.

Thus, the congestion prediction problem is modelled as follows:

$$C(t + h) = f(F_1(t), F_2(t), \cdots, \Delta OF_2(t)) \quad (1)$$

where $C(t+h)$ represents the predicted congestion state at time $t+h$, $h = \{5, 15, 30\}$ minutes. As mentioned before, the state is considered as congestion when $speed < 45$ miles/hour.

The prediction outcomes are used to provide effective warning of heavy traffic or congestion, properly integrated with the ramp meter control system in order to ensure traffic flow. For instance, if the algorithm predicts that a possible congestion may occur in a point, it may reduce metering rates of the upstream entrance ramps to cope with that situation.

2.3. Hierarchical Fuzzy Rule-Based System (HFRBS)

In HFRBS, the number of rules is reduced by decomposing the FRBS into a set of simpler fuzzy sub-systems linked hierarchically. In this hierarchical structure, the first level of the FRBS obtains an approximate output, then tuned by the second level FRBS. This procedure can be iterated in subsequent levels.

An HFRBS with serial distribution ([8, 59]) is used for methodology development and discussions throughout this study, in order to predict congestion at a desired point.

Figure 4 presents an example of an HFRBS with 4 input variables (b), in comparison with a classical FRBS with 4 input variables (a). Assuming the use of three MFs to fuzzify each input variable, and using a complete rule base that covers each combination of variables, it is found that the conventional FRBS (Figure 4(a)) must be composed of $3^4 = 81$ rules. But, using the HFRBS (Figure 4(b)), each low-dimensional fuzzy system consists of $3^2 = 9$ rules. Therefore, the total number of rules is $3 \cdot 3^2 = 27$. This indicates that the system enjoys a significant reduction in the total number of rules due to its hierarchical structure.

This way, supposing a fuzzy system with N input variables and M MFs for each variable, the rule base would be composed of:

- M^N rules in a conventionally structured FRBS.
- $M^2 \cdot (N-1)$ rules in a hierarchically structured FRBS.

Thus, given the 26 input variables in this study: $3^{26} = 2.5419e + 12$ rules would be necessary in the non-hierarchical FRBS, while only $9 \cdot (26-1) = 225$ rules would be used in the HFRBS.

In addition, in a hierarchical structure, typically, the most influential input variables are chosen as the system variables in the first level, the next most important variables are chosen in the second level, and so forth ([42, 59]).

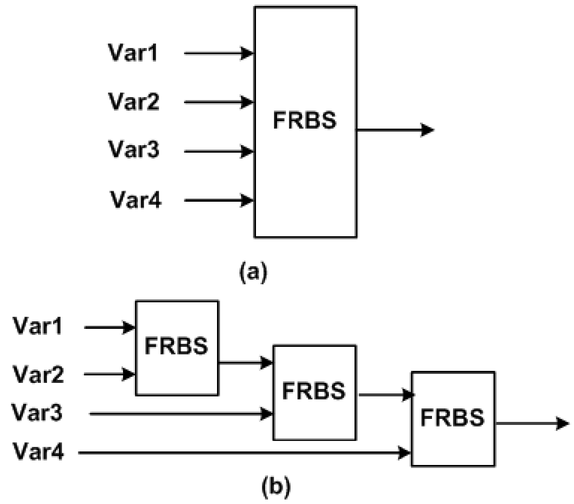


Figure 4: Fuzzy system examples. (a) Non-Hierarchical FRBS, (b) Hierarchical FRBS.

In many traffic congestion studies, different variables, such as traffic flow, occupancy, speed, and their derivatives, have been used in a variety of prediction models. However, it is often debated which variables are better suited for this purpose ([6, 33, 56]). Therefore, another great advantage of HFRBS for traffic prediction is the ability of obtaining a better understanding of the usability of the available dataset, and determining which of its attributes are the most valuable. After the variable selection process (Section 3.1), the number of chosen variables will probably be smaller than 26.

3. Genetic Optimized Hierarchical Fuzzy System

This section explains the algorithm proposed for dealing with the problem of defining and tuning an HFRBS. As mentioned in Section 1, a serial distribution of modules will be introduced in this paper, where each FRBS in the hierarchy only uses two input variables. The first FRBS employs two external variables and the remaining ones use an internal and an external variable (Figure 4(b)).

In this proposal, an evolutionary process which evolves the HFRBS incorporates a set of functions that allow us to tackle traffic prediction problem with a large number of variables. These functions are:

- Evolution of the hierarchical structure, including variable selection and ranking.
- Lateral tuning of the MFs used to codify the input variables in each fuzzy system in the HFRBS.

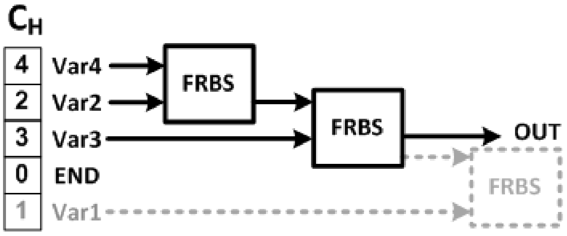


Figure 5: Permutation representation of the hierarchical model with {4, 2, 3, 0, 1}.

- Optimization of the rule base of each fuzzy system in the hierarchy.

In the following subsections, some important aspects of the algorithm are mentioned. Then the mechanisms of the algorithm framework and specific characteristics are described.

3.1. Triple Coding Scheme

A triple coding scheme for hierarchical structure (C_H), tuning (C_T), and rule-base consequences optimization (C_R) is used:

$$Chromosome = \{C_H - C_T - C_R\}$$

In the hierarchical structure part (C_H), a permutation encoding is used to represent the system. A hierarchical structure with N variables is encoded into a permutation of $N + 1$ elements. The representation is as follows:

$$C_H = \{h_1, h_2, \ldots, h_i, \ldots, h_{N+1}\} \qquad (2)$$

where $h_i \in \{0, 1, \ldots, N\}$.

With this encoding, a value j in the ith position means the use of the jth variable in the ith position of the hierarchy. In addition, 0 means no use of variables from this point.

Assuming that $C_H = \{4, 2, 3, 0, 1\}$ ($N = 4$), Figure 5 illustrates the permutation-structural representation of the hierarchical fuzzy model.

As regards the lateral tuning part (C_T), a real matrix with dimensions $I \cdot M \cdot N$ is adopted in the proposed model, where I is the number of input variables per module, M is the number of MFs to codify each input variable, and N is the total number of variables in the dataset. It could be represented more formally as follows:

$$C_T = \{t_{(1,1,1)}, \cdots, t_{(1,1,M)}; \cdots, t_{(I,N-1,M)}\} \qquad (3)$$

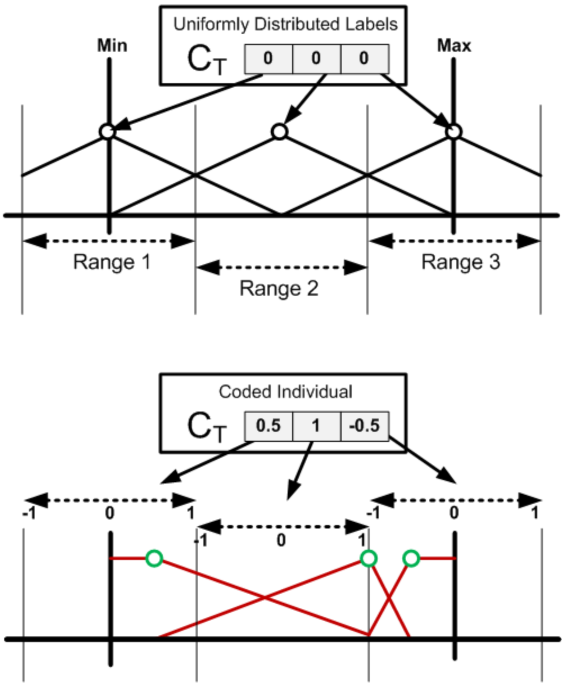


Figure 6: Example of lateral tuning codification. Uniformly distributed labels (top) and resulting after codification with lateral tuning (bottom).

where $t_{i,j,k}$ sets the value to use to modify the kth membership function of the ith input variable of the jth module in the hierarchy.

Instead of using the absolute position of the core of the MFs, the genetic lateral tuning process proposed in ([3]) is employed: for each set of MFs, a real value makes the core of the MF move within a predefined range centered on the position it would occupy in an equally distributed configuration. Figure 6 shows an example of how the codification works. In the top part, the initial uniformly distributed labels and the ranges in whose they can move are shown. In the bottom part, the final distribution obtained for an individual coded as {0.5,1,-0.5} is presented. Each value represents the displacement along the range in which the MF moves.

A real coded matrix is used for the rule-base consequences part (C_R), where we consider the number of labels per variable M, the total number of variables N, and the number of input variables for each FRBS in level I. The complete consequence base encoded as a part of real coded chromosome C_R is

$$C_R = (r_{(1,1)}, r_{(1,2)}, \cdots, r_{(M^I, N-1)}), \qquad (4)$$

where $r_{(i,j)} \in [0, 1]$ indicates the value to use as the consequent of the ith rule of the jth module in the hierarchy. No rule selection is carried out in this paper. For this reason, a rule base is composed by M^I and the composed rules.

In this paper, $I = 2$ inputs are used per module, and $M = 3$ MFs per input variable. Figure 7 illustrates the triple coding scheme for the example with $N = 4$ possible input variables. In this figure, it can be seen that

- C_H determines the number of modules by positioning the special character 0, so only the first three variables are used. C_H also determines the order in which the variables are processed by the hierarchy; in this case $\{4, 2, 3\}$.
- Since only two inputs are processed by each module, and they are codified by 3 MFs, C_T is codified as a matrix with 6 ($I \cdot M$) rows and three (N) columns. Each column sets the MFs of a module in the hierarchy.
- C_R is a matrix with 9 (M^I) rows and three (N) columns. Again, the ith column represents the rule base to implement by the ith module in the hierarchy.

3.2. Genetic operators

The chromosomes that make up the initial population are randomly generated. Each individual is initialized with a random permutation of the values $\{0, 1, \ldots N\}$ in the C_H part, an $(I \times M \times (N-1))$ matrix of values in $[-1, 1]$ in the C_T part, and an $(M^I \times N)$ matrix with values in $[0, 1]$ in the C_R part.

For the whole system, the selection procedure is done by roulette wheel, where each chromosome is represented by a space that proportionally corresponds to its fitness. This way, better individuals get higher chances.

The ordered two-point crossover ([22]) is used as the crossover function for the hierarchical part (C_H). Interchange mutation is used for the mutation operator.

The ordered two-point crossover works as follows. Given two parent chromosomes, firstly, two random crossover points are selected to cut each parent into a left, a middle, and a right part. One offspring inherits its left and right parts from the first parent, and its middle part is filled by the sequence of genes in the second parent. The second offspring is generated the other way round: using the left and right parts from the second parent and the middle of the first one. This is one of the most popular and effective crossovers for dealing with permutations ([19, 21]). After the crossover operation, the implementation of the mutation is done by the interchange operator. It selects randomly two genes within a specific range (a relatively small interval), in order to be further improved by fine-tuning the chromosome.

For the real coded parts of the individual (C_T and C_R), $BLX - \alpha$ crossover ([24]) and BGA mutation ([45]) are used. Both are widely employed in real-coded GA ([20, 23, 31]).

Given two parents $X = (x_1 \ldots x_m)$ and $Y = (y_1 \ldots y_m)$, for each i, $BLX - \alpha$ crossover creates two offspring by generating random values in the interval:

$$[min(x_i, y_i) - \alpha \cdot |x_i - y_i|,\\ max(x_i, y_i) + \alpha \cdot |x_i - y_i|]$$

with $\alpha \in [0, 1]$.

For the same assumed $X = (x_1 \ldots x_m)$ as above, the x_i obtained from the BGA mutation operator is

$$x_i^{'} = x_i \pm range_i \cdot \sum_{k=0}^{15} (\alpha_k 2^{-k})$$

where $range_i$ defines the mutation range: it is set to $0.5 \cdot (b_i - a_i)$ in this paper. The sign ($+$ or $-$) is chosen with a probability of 0.5 and $\alpha_k \in \{0, 1\}$ is randomly generated with $p(\alpha_k = 1) = \frac{1}{16}$.

3.3. Chromosome Evaluation

In this paper, we will use the well-known mean absolute error (MAE):

$$\text{MAE} = \frac{1}{N_t} \sum_{i=1}^{N_t} |O_i - E_i| \tag{5}$$

where E_i and O_i are the desired and the obtained outputs for the ith sample in the dataset, and N_t represents the number of training data.

In order to apply the roulette wheel operator for the crossover selection process, the value $(1 - MAE)$ is used.

3.4. Mechanism and characteristics of the algorithm framework

The procedure of construction of the GHFRBS model is shown in Figure 8. The characteristics of this model allow the system to identify and rank the input variables by learning the intrinsic rules individually and mutually from the highly correlated datasets. The selection of the input variables and the ranking mechanisms are as follows.

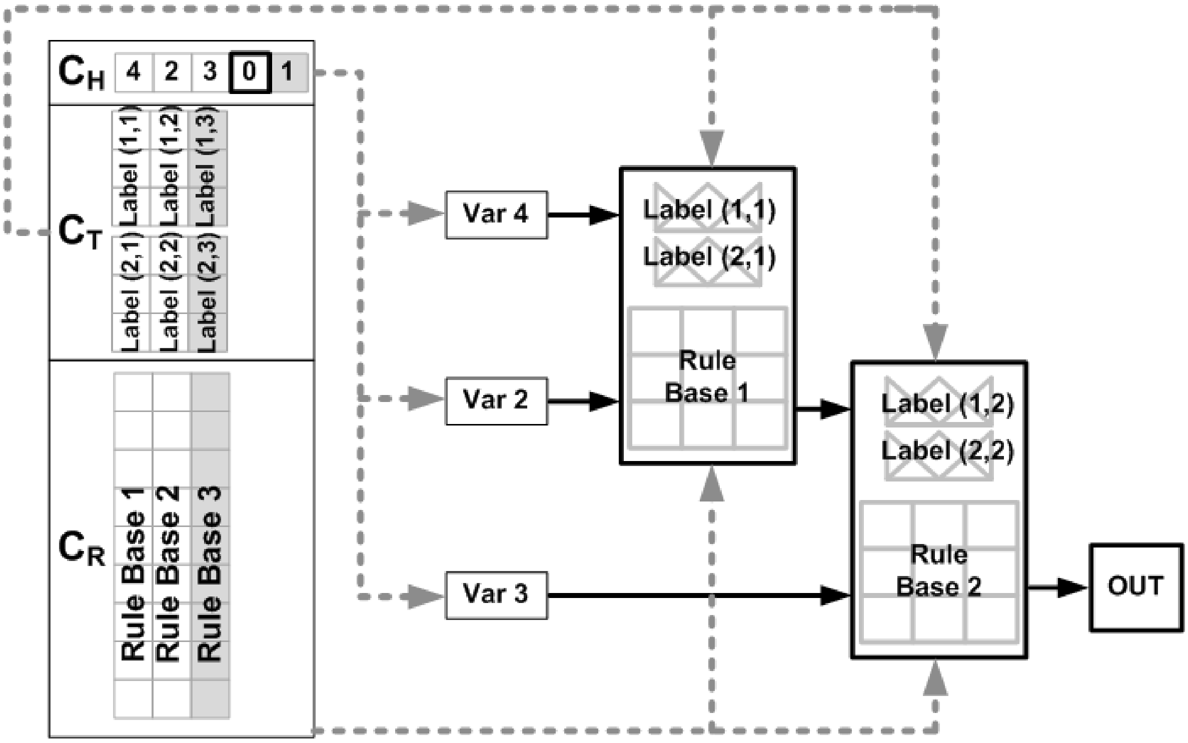


Figure 7: A triple coding scheme example for a system with $N = 5$ variables, $I = 2$ inputs per module and $M = 3$ MFs per input.

- Initially, each input variable is selected to formulate the hierarchical fuzzy model randomly and equiprobably;
- The input variables with greater contribution to the objective function will be enhanced and get a higher opportunity to be selected in the next evolution;
- The genetic operators, i.e., crossover and mutation, provide a selection and ranking scheme by which the modules select the appropriate input variables automatically.

The proposed evolutionary algorithm framework is based on the well-known steady-state GA ([53]), shown in Algorithm 1. The steady-state GA benefits from selecting two individuals and combining them to obtain two offspring by crossover and mutation operators. Then, if these two new individuals are better adapted than the two worst individuals of the population, the former are included in the population by replacing the latter. Thanks to the advantage of this elitist strategy and elimination mechanism, the use of the steady-state GA allows a faster convergence and smaller evaluation number.

4. Experimentation

To evaluate the usefulness of the proposed method, namely, GHFRBS, an experimental study was carried out, which was organized as follows:

- Section 4.1 presents the experimental setup, including its datasets, techniques and parameters.
- Section 4.2 details the obtained results.
- Section 4.3 analyses the obtained results in detail, as to the accuracy and complexity of the obtained models.

4.1. Experimental Setup

This section presents all the datasets, techniques and parameters used during the experimentation, with the purpose of validating the performance of the implemented GHFRBS.

4.1.1. Datasets

As mentioned in Section 2.1, the proposed GHFRBS was implemented for a traffic congestion prediction

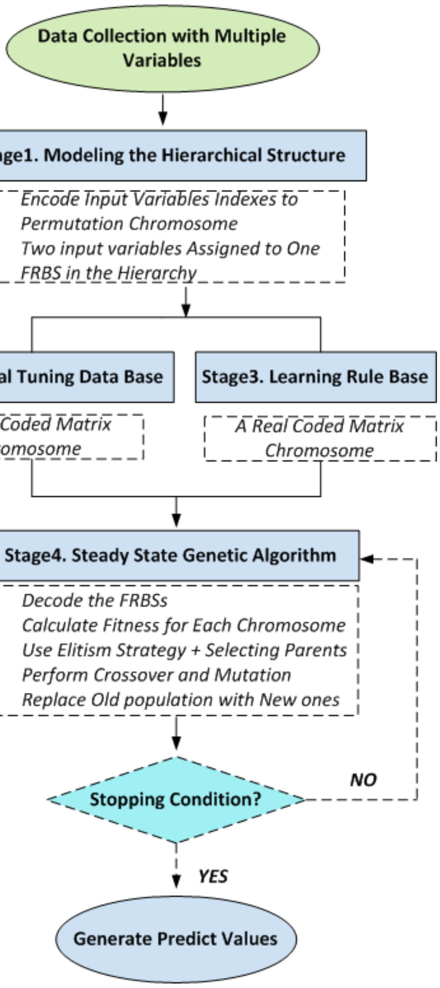

Data Collection with Multiple Variables
Stage1. Modeling the Hierarchical Structure
Encode Input Variables Indexes to Permutation Chromosome
Two input variables Assigned to One FRBS in the Hierarchy
Stage2. Lateral Tuning Data Base
A Real Coded Matrix Chromosome
Stage3. Learning Rule Base
A Real Coded Matrix Chromosome
Stage4. Steady State Genetic Algorithm
Decode the FRBSs
Calculate Fitness for Each Chromosome
Use Elitism Strategy + Selecting Parents
Perform Crossover and Mutation
Replace Old population with New ones
Stopping Condition?
NO
YES
Generate Predict Values

Figure 8: The construction procedures of the GHFRBS model.

problem using traffic data obtained from *PeMS*. Descriptions of the 26 involved variables were also explained in Section 2.2. Finally, three datasets are generated in order to test the implemented system for congestion prediction, each one of them is designed to predict whether congestion is going to occur in the next 5, 15 and 30 minutes, and they are referred to as $TRAFFIC_5$, $TRAFFIC_{15}$ and $TRAFFIC_{30}$, respectively.

In addition, with the aim of studying whether the introduced algorithm is feasible for some of the most used benchmark datasets, 4 two-class classification problems with numbers of real-valued variables from low to high have been tested. They are available in *KEEL*[2] ([4]). Table 1 summarizes the main features for each problem,

```
Data: $N_P$ (population size)
$M_E$ (maximum number of evaluations)
Result: Best individual from $P$
$e \leftarrow 0$
$P \leftarrow$ Initialize_Population
Evaluate individuals in $P$
$e \leftarrow$ e + $N_P$
while $e < M_E$ do
    Select two individuals $P_1$ and $P_2$ from $P$
    Apply crossover and mutation over $P_1$ and $P_2$
    Evaluate $P_1$ and $P_2$
    $e \leftarrow$ e + 2
    Find the two worst individuals $Worst_{(1,2)}$
    from $P$
    if $P_{(1,2)}$ are better than $Worst_{(1,2)}$ then
        Replace $Worst_{(1,2)}$ by $P_{(1,2)}$
    else
        return $P$
    end
end
```

Algorithm 1: Steady-State Genetic algorithm

where $\#Attributes$ stands for the number of input variables, $\#Cases$ for the total number of instances, and $\#Source$ indicates the web site where the data sets are available or where the original data come from.

Dataset	$\#Attributes$	$\#Cases$	$\#Source$
$PIMA$	8	768	KEEL
$RING$	20	7400	KEEL
$SPAMBASE$	57	4597	KEEL
$SONAR$	60	208	KEEL
$TRAFFIC_5$	26	6119	PeMS
$TRAFFIC_{15}$	26	6119	PeMS
$TRAFFIC_{30}$	26	6119	PeMS

Table 1: Data sets considered in the experimental analysis.

4.1.2. Compared Techniques

In order to study whether the behavior of GHFRBS is comparable to some of the most used soft computing techniques, its performance will be compared with the following techniques in a variety of aspects:

- *Fuzzy AdaBoost* ([15]) is an improved version of the boosting algorithm AdaBoost ([18]) to deal with fuzzy rules.

- *Fuzzy LogitBoost* ([38]) employs a GA to extract fuzzy rules iteratively, which are then combined to decide the output.

[2] www.keel.es

- *Fuzzy Chi-RW* ([12, 27]) identifies the relations between the variables of the problem and makes an association between the feature space and the space of classes, with the aim of generating the rule base.

All of the above methods are run by using *KEEL*, and the parameter values are configured according to the recommended setting.

4.1.3. Parameter Setup

The proposed algorithm has been run with the following parameter values: $M_E = 100,000$ as the maximum number of evaluations, $N_P = 100$ as the population size, $p_c = 0.8$ as the crossover probability, $p_m = 0.2$ as the mutation probability per chromosome. Since an analysis of the parameter sensitivity is not a major concern of this study, we did not perform any previous analysis to fix these values, and the parameters were selected in accordance with the standards of evolutionary techniques. Better results might be obtained by tuning them sufficiently. We set $M = 3$ labels per variable for all the problems and methods in order to make the rule base not grow excessively.

All the experiments reported in this paper have been performed by a 5×2 cross-validation model. That is, the data set is randomly split into two parts of (approximately) equal size. The algorithm is then applied five times to each problem.

4.2. Results

The outputs generated by the GHFRBS are continuous values within the interval [0, 1], indicating the levels of congestion. In order to evaluate the prediction performance and compare it with other methods, the same measure of performance should be employed. The real outputs are rounded to and compared with the actual congestion indicator, i.e., 0 or 1. If errors occur, they are categorized as false alarm (false predictions of *congestion* or *no congestion*).

In this case, the model produces the final binary output: *congestion* or *no congestion*. It can be considered as a binary classification task. For the purpose of evaluating the classification accuracy of GHFRBS and comparing it with other techniques, the MAE (Equation 5) measure is chosen.

Table 2 collects the results, where the $\overline{MAE_{tra}} \pm \sigma$ and $\overline{MAE_{tst}} \pm \sigma$ columns are the average error rate values with standard deviation over the training and test data set, respectively; $\#R$ and $\#V$ stand for the average number of fuzzy rules and the average number of input variables used by the system, respectively; $\#R/V$ is the number of variables per rule. In the case of the presented GHFRBS, the number of variables ($\#V$) also gives us the number of modules in the system, which can be calculated as $\#V - 1$ (see Figure 4 (b)).

$\overline{MAE_{tra}} \pm \sigma$ and $\overline{MAE_{tst}} \pm \sigma$ reflect the performance of the algorithms over the datasets, while $\#R$, $\#V$ and $\#R/V$ indicate the obtained systems' complexity.

Moreover, Table 4 shows the 26 variables that compound the $TRAFFIC$ datasets and the order in which they have been included in the hierarchy for each one of the executions (when no number is shown, the variable has not been selected). With this, it is expected to be able to identify the most important variables in the dataset.

4.3. Analysis of the results

In this section, a detailed analysis of the obtained results is presented in terms of prediction accuracy, size and complexity of the obtained rule sets. Most importantly, a multi-perspective analysis of the results for the *TRAFFIC* problems is given.

4.3.1. Comparison analysis of the obtained prediction accuracy

With respect to the accuracy of the algorithms ($MAE_{(tra|tst)}$ in Table 2), it can be highlighted that:

- With respect to *PIMA* (8 variables), *Fuzzy LogitBoost* clearly outperforms the other approaches in general in the training set, but it does not show any clear advantages over the proposed algorithm in the test.

- For the *RING* dataset (20 variables), the accuracy of solution obtained by GHFRBS doesn't outperform Boost algorithms.

- In the *SPAMBASE* and *SONAR* datasets (57 and 60 variables), the accuracy of GHFRBS is higher than the other algorithms.

- With the *TRAFFIC* datasets (26 variables), The results clearly show that the proposed approach is capable of predicting traffic congestion up to 30 minutes into the future with a high degree of accuracy (94%–96%). Meanwhile, it has obtained the highest degree of accuracy in all three *TRAFFIC* cases when compared with the rest of techniques.

We complement the statistical study of the obtained test results by comparing each pair of techniques. In

Dataset	Method	$\overline{MAE_{tra}} \pm \sigma$	$\overline{MAE_{tst}} \pm \sigma$	$\#V$	$\#R$	$\#V/R$
$PIMA$	Fuzzy AdaBoost	0.2182 ± 0.0200	0.2745 ± 0.0252	8	8	5.25
	Fuzzy LogitBoost	**0.1385 ± 0.0217**	**0.2557 ± 0.0144**	8	25	5.02
	Fuzzy Chi-RW	0.2406 ± 0.0063	0.2740 ± 0.0206	8	72.8	8
	GHFRBS	0.2135 ± 0.0190	**0.2589 ± 0.0156**	4.4	30.6	2
$RING$	Fuzzy AdaBoost	0.1462 ± 0.0070	0.1550 ± 0.0071	20	8	12.47
	Fuzzy LogitBoost	**0.0250 ± 0.0029**	**0.0359 ± 0.0027**	20	25	15.07
	Fuzzy Chi-RW	0.4196 ± 0.0030	0.4500 ± 0.0033	20	359.6	20
	GHFRBS	0.2290 ± 0.0059	0.2924 ± 0.0474	7.2	55.8	2
$SPAMBASE$	Fuzzy AdaBoost	0.6063 ± 0.0099	0.6053 ± 0.0099	57	8	48.25
	Fuzzy LogitBoost	0.3936 ± 0.0099	0.3947 ± 0.0099	57	25	48.68
	Fuzzy Chi-RW	0.2741 ± 0.0070	0.2942 ± 0.0139	57	253	57
	GHFRBS	**0.2108 ± 0.0215**	**0.2219 ± 0.0090**	34.8	304.2	2
$SONAR$	Fuzzy AdaBoost	0.5404 ± 0.0292	0.5250 ± 0.0323	60	8	50.28
	Fuzzy LogitBoost	0.4288 ± 0.0376	0.4769 ± 0.0357	60	25	50.53
	Fuzzy Chi-RW	**0.0000 ± 0.0000**	0.5039 ± 0.0251	60	104	60
	GHFRBS	**0.1769 ± 0.0221**	**0.3461 ± 0.0461**	27.2	235.8	2
$TRAFFIC_5$	Fuzzy AdaBoost	0.0456 ± 0.0059	0.0476 ± 0.0038	26	8	18.93
	Fuzzy LogitBoost	**0.0292 ± 0.0026**	0.0474 ± 0.0030	26	25	19.94
	Fuzzy Chi-RW	0.0457 ± 0.0015	0.0535 ± 0.0028	26	511.4	20
	GHFRBS	0.0313 ± 0.0031	**0.0367 ± 0.0025**	10	81	2
$TRAFFIC_{15}$	Fuzzy AdaBoost	0.0571 ± 0.0083	0.0643 ± 0.0098	26	8	19.38
	Fuzzy LogitBoost	0.0394 ± 0.0041	0.0560 ± 0.0024	26	25	19.46
	Fuzzy Chi-RW	0.0494 ± 0.0012	0.0604 ± 0.0027	26	506.2	26
	GHFRBS	**0.0313 ± 0.0031**	**0.0533 ± 0.0025**	11.8	97.2	2
$TRAFFIC_{30}$	Fuzzy AdaBoost	0.0679 ± 0.0106	0.0730 ± 0.0127	26	8	18.93
	Fuzzy LogitBoost	0.0471 ± 0.0037	0.0635 ± 0.0039	26	25	19.84
	Fuzzy Chi-RW	0.0539 ± 0.0029	0.0626 ± 0.0053	26	503.6	26
	GHFRBS	**0.0423 ± 0.0045**	**0.0533 ± 0.0110**	11.4	93.6	2

Table 2: Comparison of the performance of Fuzzy AdaBoost, Fuzzy LogitBoost and Fuzzy Chi-RW with GHFRBS in terms of $\overline{MAE_{tra}} \pm \sigma$, $\overline{MAE_{tst}} \pm \sigma$, $\#V$, $\#R$ and $\#V/R$.

Table 3, Student's t-test for each problem used in Table 2 is shown. The symbols + and - indicate that the method in the row significantly improves/degrades the performance obtained with the method in the column. The symbol * denote a nonsignificant difference, and the symbol () means equivalence. The t statistic is calculated as presented in Equation 6.

$$t = \frac{\overline{X_1} - \overline{X_2}}{\sqrt{\frac{(N_1-1)\sigma_1+(N_2-1)\sigma_2}{N_1+N_2-2}\frac{N_1+N_2}{N_1 N_2}}} \qquad (6)$$

where $\overline{X_i}$, σ_i and N_i represent, respectively, the mean, standard deviation and number of samples for each one of the techniques to be compared.

Using a 95% confidence interval ($t_{0.05}$ = 2.021), the test indicated that the GHFRBS significantly outperformed all the other techniques on *SPAMBASE*, *SONAR* and $TRAFFIC_{5,15,30}$ datasets. In addition, the proposed technique outperforms the other techniques in half the cases with a smaller number of input variables. One possibility is that the specific Boost technique is advantageous when solving problems with a small number of input variables.

Summarizing, the proposed GHFRBS obtains, in all cases, comparable performance to the most-used techniques, the best results in *TRAFFIC* problems.

4.3.2. Analysis of the obtained fuzzy models

When analysing the size and complexity of the obtained models, it is important to note that, given the parameters recommended for their use, *Fuzzy AdaBoost* and *LogitBoost* use a fixed number of rules (preestablished), while *Fuzzy Chi-RW* uses a variable number of rules and GHFRBS uses a variable number of modules (used variables) with a fixed number of rules.

First, the incorporation of a variable selection mechanism by the GHFRBS allows reducing the complexity of the datasets by using only less than one-half (on average) of the number of variables in the original datasets. In particular, in the *TRAFFIC* datasets, the number of variables is reduced to 40% approximately. More specifically, as to the complexity of the models, it is worth highlighting that:

- *Fuzzy AdaBoost* and *LogitBoost* always obtain less rules than GHFRBS, since the number of rules is pre-established as a parameter of the method.
- Furthermore, the complexity of the obtained rules is much higher than the one from the rules obtained by GHFRBS. Rules from Boost methods are formed by about 5 clauses in the antecedent part for *PIMA*, 13 for *RING*, 50 for *SPAMBASE* and *SONAR*, and 19 for *TRAFFIC* datasets.
- As to *Fuzzy Chi-RW*, the GHFRBS obtains fewer rules in all cases, with the exception of the *SPAMBASE* and *SONAR* datasets. In particular, for *TRAFFIC* datasets, *Fuzzy Chi-RW* obtains more than 500 rules, while GHFRBS gets less than 100, and with higher precision.
- It is also important to note that the rules obtained by *Fuzzy Chi-RW* always make use of all the variables in the antecedent part, while the ones derived from GHFRBS only use two.

To sum up, it is worth noting that in problems with a low number of variables, the precision does not show clear advantages over the other algorithms when the number of input variables decrease. Nevertheless, in problems with a higher number of variables as well as real traffic data, where there is greater dependence between them, a hierarchical structure and a lower number of input variables help to improve the classification accuracy and decrease the complexity of the fuzzy systems.

Figure 9 illustrates an example of the hierarchical structures obtained by GHFRBS in a dataset of $TRAFFIC_5$. It also includes information about the obtained solution, such as the number of rules, the fitness function, and the values obtained for each of the measures. The average values are: $MAE_{tra} = 0.0307$, $MAE_{tst} = 0.0333$, $\#V = 4$, $\#R = 27$, and $\#V/R = 2$. The problem has 26 input variables. They have been reduced, by selection, to 4 variables (ΔF_1, ΔF_2, O_3, O_2) and hierarchically ranked in three modules, where the output of the last module (Out3) denotes the prediction of the congestion state in the next 5 minutes.

4.3.3. Analysis of the input variables for traffic congestion prediction

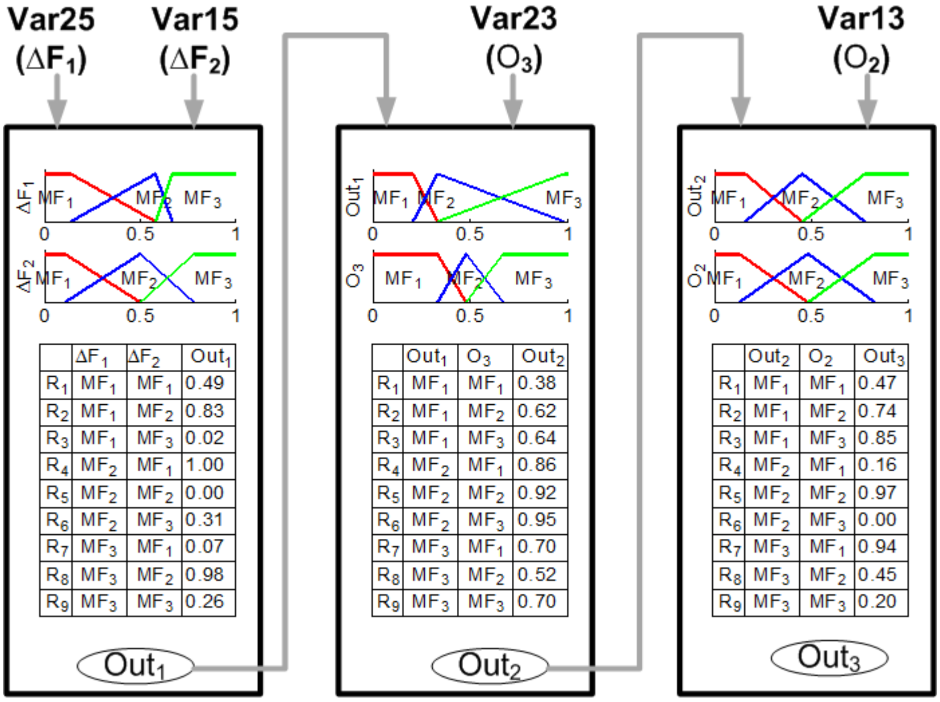
In order to analyse the importance of each input variable involved in the congestion prediction problem, Table 4 shows the order in which each variable is selected for each one of the executions of the GHFRBS. Analysing the table, it can be concluded that:

- The variables which are selected with the highest frequency are O_2, ΔO_2, O_3, ΔF_1, S_2, ΔS_2, ΔOF_2.
- The variables chosen with the lowest frequency are IF_2, OF_2, S_3 and ΔF_3.
- In the $TRAFFIC_5$ problem, it is worth noting that variable O_2 is selected in each execution (with ranking positions 13, 7, 12, 14, and 4). Then, the variable O_3 is selected in four executions with ranking positions 5, 3, 4, and 3.
- As to the $TRAFFIC_{30}$ problem, it is worth noting that the variable O_2 is selected in each execution (with ranking positions 1, 10, 11, 23, and 10), and the variable S_2 is selected in four executions with ranking positions 2, 1, 2, and 6.

O_2, ΔO_2, O_3, ΔF_1, S_2, ΔS_2, and ΔOF_2 are found to be the variables with the highest correlation with traffic congestion at the target point, while IF_2, OF_2, S_3 and ΔF_3 are the least correlated with traffic congestion. From the point of view of input variables, the results indicate that occupancy's prediction performance is better than that of the speed or the flow. The same level of speed or flow may correspond to two distinct traffic states (free-flow or congested). This is most probably because it is affected by traffic composition and the vehicles' length. From the perspective of location, the input variables at the point of interest are mostly selected, because traffic conditions are mostly associated with their own past values. In summary, the usefulness of an input variable for prediction depends on its ranking in the fuzzy systems and the frequency of its being selected.

5. Conclusions and Future Research

In this paper, we have proposed a novel approach, based on hierarchical fuzzy rule base systems and genetic algorithms, to build traffic congestion prediction systems from a high number of input variables.



	ΔF_1	ΔF_2	Out_1
R_1	MF_1	MF_1	0.49
R_2	MF_1	MF_2	0.83
R_3	MF_1	MF_3	0.02
R_4	MF_2	MF_1	1.00
R_5	MF_2	MF_2	0.00
R_6	MF_2	MF_3	0.31
R_7	MF_3	MF_1	0.07
R_8	MF_3	MF_2	0.98
R_9	MF_3	MF_3	0.26

	Out_1	O_3	Out_2
R_1	MF_1	MF_1	0.38
R_2	MF_1	MF_2	0.62
R_3	MF_1	MF_3	0.64
R_4	MF_2	MF_1	0.86
R_5	MF_2	MF_2	0.92
R_6	MF_2	MF_3	0.95
R_7	MF_3	MF_1	0.70
R_8	MF_3	MF_2	0.52
R_9	MF_3	MF_3	0.70

	Out_2	O_2	Out_3
R_1	MF_1	MF_1	0.47
R_2	MF_1	MF_2	0.74
R_3	MF_1	MF_3	0.85
R_4	MF_2	MF_1	0.16
R_5	MF_2	MF_2	0.97
R_6	MF_2	MF_3	0.00
R_7	MF_3	MF_1	0.94
R_8	MF_3	MF_2	0.45
R_9	MF_3	MF_3	0.20

Figure 9: Example of a solution obtained by GHFRBS for the prediction of traffic congestion 5 minutes later.

The proposed GHFRBS has the following novel features: (i) thanks to its hierarchical structure, a smaller number of input variables are generated, and the selected input variables are distributed in a set of serial modules; (ii) the complexity of the fuzzy rules is reduced significantly because the number of input variables per modules is set to two; (iii) a Steady-State Genetic Algorithm and a lateral tuning process of the MFs are employed to automatically construct the FRBSs.

In order to test the performance of the new approach, we have applied it to the prediction of traffic congestion with real traffic data collected from PeMS, as well as benchmark datasets from KEEL. The obtained results are promising, and have demonstrated the simplicity of the fuzzy rules of the obtained models, and the effectiveness of GHFRBS for predicting traffic congestion with time horizons of 5, 15 and 30 minutes.

Last but not least, due to its serial hierarchical structure and its automatic ranking and selection of the input variables, our technique may explore a new research topic in the field of data mining, in obtaining a better understanding of the usefulness of the available datasets. This will be regarded as our future research.

ACKNOWLEDGMENT

The authors would like to thank the EU *Intelligent Cooperative Sensing for Improved traffic efficiency* (ICSI) project (FP7-ICT-2011-8) for its support in the development of this research.

References

[1] Abdulhai, B., Porwal, H., Recker, W., 2002. Short-term traffic flow prediction using neuro-genetic algorithms. Journal of Intelligent Transportation Systems: Technology, Planning, and Operations 7 (1), 3–41.

[2] Ahmed, M., Cook, A., 1979. Analysis of freeway traffic time-series data by using Box–Jenkins techniques. Transportation Research Record 722, 1–9.

[3] Alcalá, R., Alcalá-Fernández, J., Herrera, F., 2007. A proposal for the genetic lateral tuning of linguistic fuzzy systems and its interaction with rule selection. IEEE Transactions on Fuzzy Systems 15 (4), 616–635.

[4] Alcalá-Fernández, J., Sánchez, L., García, S., del Jesús, M. J., Ventura, S., Garrell, J., Otero, J., Romero, C., Bacardit, J., Rivas, V. M., et al., 2009. Keel: A software tool to assess evolutionary algorithms for data mining problems. Soft Computing 13 (3), 307–318.

[5] Badie, A., 2010. Genetic fuzzy self-tuning PID controllers for antilock braking systems. Engineering Applications of Artificial Intelligence 23 (7), 1041–1052.

[6] Balke, K. N., Chaudhary, N., Chu, C.-L., Kuchangi, S., Nelson, P., Songchitruksa, P., Swaroop, D., Tyagi, V., 2005. Dynamic

traffic flow modeling for incident detection and short-term congestion prediction: Year 1 progress report. Tech. rep., Texas Transportation Institute, Texas A&M University System.

[7] Bauza, R. and Gozalvez, J., 2013. Traffic congestion detection in large-scale scenarios using vehicle-to-vehicle communications. Journal of Network and Computer Applications 36 (5), 1295–1307.

[8] Benítez, A. D., Casillas, J., 2013. Multi-objective genetic learning of serial hierarchical fuzzy systems for large-scale problems. Soft Computing 17 (1), 165–194.

[9] Cala, S., Moreno-Velo, F. J., 2010. XFHL: A tool for the induction of hierarchical fuzzy systems. In: IEEE International Conference on Fuzzy Systems. pp. 1–6.

[10] Chan, K., Dillon, T., Singh, J., Chang, E., 2012. Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and Levenberg–Marquardt algorithm. IEEE Transactions on Intelligent Transportation Systems 13 (2), 644–654.

[11] Chen, C., 2003. Freeway performance measurement system (PeMS). California PATH Research Report.

[12] Chi, Z., Yan, H., Tuan, P., 1996. Fuzzy Algorithms: With Applications to Image Processing and Pattern Recognition. Vol. 10. World Scientific, Singapore.

[13] Cordon, O., Gomide, F., Herrera, F., Hoffmann, F., Magdalena, L., 2004. Ten years of genetic fuzzy systems: Current framework and new trends. Fuzzy Sets and Systems 141 (1), 5–31.

[14] Cordón, O., Herrera, F., Zwir, I., 2003. A hierarchical knowledge-based environment for linguistic modeling: Models and iterative methodology. Fuzzy Sets and Systems 138 (2), 307–341.

[15] del Jesus, M. J., Hoffmann, F., Navascués, L. J., Sánchez, L., 2004. Induction of fuzzy-rule-based classifiers with evolutionary boosting algorithms. IEEE Transactions on Fuzzy Systems 12 (3), 296–308.

[16] Dimitriou, L., Tsekeris, T., Stathopoulos, A., 2008. Adaptive hybrid fuzzy rule-based system approach for modeling and predicting urban traffic flow. Transportation Research Part C: Emerging Technologies 16 (5), 554–573.

[17] Dougherty, M. S., Cobbett, M. R., 1997. Short-term inter-urban traffic forecasts using neural networks. International Journal of Forecasting 13 (1), 21–31.

[18] Freund, Y., Schapire, R. E., et al., 1996. Experiments with a new boosting algorithm. In: International Conference on Machine Learning. pp. 148–156.

[19] Fung, R. Y., Liu, R., Jiang, Z., 2013. A memetic algorithm for the open capacitated arc routing problem. Transportation Research Part E: Logistics and Transportation Review 50, 53–67.

[20] Gacto, M. J., Alcalá, R., Herrera, F., 2012. A multi-objective evolutionary algorithm for an effective tuning of fuzzy logic controllers in heating, ventilating and air conditioning systems. Applied Intelligence 36 (2), 330–347.

[21] Goldberg, D., Lingle, R., 1985. Alleles loci, and the traveling salesman problem. In: Proceedings of the 1st International Conference on Genetic Algorithms. Lawrence Erlbaum, Hillsdale, NJ, USA, pp. 154–159.

[22] Goldberg, D. E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Boston, MA, USA.

[23] Hadavandi, E., Shavandi, H., Ghanbari, A., 2010. Integration of genetic fuzzy systems and artificial neural networks for stock price forecasting. Knowledge-Based Systems 23 (8), 800–808.

[24] Herrera, F., Lozano, M., Verdegay, J., 1998. Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. Artificial Intelligence Review 12 (4), 265–319.

[25] Hodge, V., Krishnan, R., Jackson, T., Austin, J., Polak, J., 2011. Short-term traffic prediction using a binary neural network. In: 43rd Annual UTSG Conference, Open University, Milton Keynes, UK.

[26] Holland, J. H., 1992. Adaptation in Natural and Artificial Systems. MIT Press.

[27] Ishibuchi, H., Yamamoto, T., 2005. Rule weight specification in fuzzy rule-based classification systems. IEEE Transactions on Fuzzy Systems 13 (4), 428–435.

[28] Jin, S., Wang, D.-H., Xu, C., Ma, D.-F., 2013. Short-term traffic safety forecasting using Gaussian mixture model and Kalman filter. Journal of Zhejiang University Science A 14 (4), 231–243.

[29] Karlaftis, M., Vlahogianni, E., 2011. Statistical methods versus neural networks in transportation research: Differences, similarities and some insights. Transportation Research Part C: Emerging Technologies 19 (3), 387–399.

[30] Konar, A., 2005. Computational Intelligence: Principles, Techniques and Applications. Springer-Verlag, Berlin.

[31] Lacroix, B., Molina, D., Herrera, F., 2012. Region based memetic algorithm with LS chaining. In: IEEE Congress on Evolutionary Computation. pp. 1–6.

[32] Lee, M.-L., Chung, H.-Y., Yu, F.-M., 2003. Modeling of hierarchical fuzzy systems. Fuzzy Sets and Systems 138 (2), 343–361.

[33] Lee, Y., Wei, C.-H., 2010. A computerized feature selection method using genetic algorithms to forecast freeway accident duration times. Computer-Aided Civil and Infrastructure Engineering 25 (2), 132–148.

[34] Lin, W.-H., Lu, Q., Dahlgren, J., 2002. Dynamic procedure for short-term prediction of traffic conditions. Transportation Research Record: Journal of the Transportation Research Board 1783 (1), 149–157.

[35] Liu, K., Fei, X., 2010. A fuzzy-logic-based system for freeway bottleneck severity diagnosis in a sensor network. Transportation Research Part C: Emerging Technologies 18 (4), 554–567.

[36] Okutani, I., Stephanedes, Y., 1984. Dynamic prediction of traffic volume through Kalman filtering theory. Transportation Research Part B: Methodological 18 (1), 1–11.

[37] Onieva, E., Milanés, V., Villagra, J., Pérez, J., Godoy, J., 2012. Genetic optimization of a vehicle fuzzy decision system for intersections. Expert Systems with Applications 39 (18), 13148–13157.

[38] Otero, J., Sánchez, L., 2006. Induction of descriptive fuzzy classifiers with the LogitBoost algorithm. Soft Computing 10 (9), 825–835.

[39] Park, B., 2002. Hybrid neuro-fuzzy application in short-term freeway traffic volume forecasting. Transportation Research Record 1802 (1), 190–196.

[40] Posawang, P., Phosaard, S., Polnigongit, W., Pattara-Atikom, W., 2010. Perception-based road traffic congestion classification using neural networks and decision tree. In: Electronic Engineering and Computing Technology. Springer-Verlag, Berlin, pp. 237–248.

[41] Quek, C., Pasquier, M., Lim, B., 2006. Pop-traffic: A novel fuzzy neural approach to road traffic analysis and prediction. IEEE Transactions on Intelligent Transportation Systems 7 (2), 133–146.

[42] Raju, G., Zhou, J., Kisner, R., 1991. Hierarchical fuzzy control. International Journal of Control 54 (5), 1201–1216.

[43] Samoili, S., Dumont, A.-G., 2012. Framework for real-time traffic forecasting methodology under exogenous parameters. In: 12th Swiss Transport Research Conference. pp. 512–522.

[44] Sapankevych, N., Sankar, R., 2009. Time series prediction using support vector machines: A survey. IEEE Computational Intelligence Magazine 4 (2), 24–38.

[45] Schlierkamp-Voosen, D., 1993. Predictive models for the breeder genetic algorithm. Evolutionary Computation 1 (1), 25–

49.

[46] Smith, B. L., Demetsky, M. J., 1994. Short-term traffic flow prediction: Neural network approach. Transportation Research Record 1453, 98–104.

[47] Stathopoulos, A., Karlaftis, M. G., 2003. A multivariate state space approach for urban traffic flow modeling and prediction. Transportation Research Part C: Emerging Technologies 11 (2), 121–135.

[48] Stonier, R. J., Mohammadian, M., 2004. Multi-layered and hierarchical fuzzy modelling using evolutionary algorithms. In: International Conference on Computational Intelligence for Modelling, Control and Automation. University of Canberra, Canberra, Australia, pp. 321–344.

[49] Torra, V., 2002. A review of the construction of hierarchical fuzzy systems. International Journal of Intelligent Systems 17 (5), 531–543.

[50] Vlahogianni, E. I., Geroliminis, N., Skabardonis, A., 2008. Empirical and analytical investigation of traffic flow regimes and transitions in signalized arterials. Journal of Transportation Engineering 134 (12), 512–522.

[51] Vlahogianni, E. I., Karlaftis, M. G., Golias, J. C., 2005. Optimized and meta-optimized neural networks for short-term traffic flow prediction: A genetic approach. Transportation Research Part C: Emerging Technologies 13 (3), 211–234.

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

[53] Whitley, D., Kauth, J., 1988. GENITOR: A different genetic algorithm. Colorado State University, Department of Computer Science.

[54] Williams, B. M., Hoel, L. A., 2003. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. Journal of Transportation Engineering 129 (6), 664–672.

[55] Wu, C.-H., Ho, J.-M., Lee, D., 2004. Travel-time prediction with support vector regression. IEEE Transactions on Intelligent Transportation Systems 5 (4), 276–281.

[56] Yang, S., 2013. On feature selection for traffic congestion prediction. Transportation Research Part C: Emerging Technologies 26, 160–169.

[57] Zadeh, L. A., 1965. Fuzzy sets. Information and control 8 (3), 338–353.

[58] Zajaczkowski, J., Verma, B., 2010. An evolutionary algorithm based approach for selection of topologies in hierarchical fuzzy systems. In: IEEE Congress on Evolutionary Computation. pp. 1–8.

[59] Zajaczkowski, J., Verma, B., 2012. Selection and impact of different topologies in multi-layered hierarchical fuzzy systems. Applied Intelligence 36 (3), 564–584.

[60] Zhang, Y., Ye, Z., 2008. Short-term traffic flow forecasting using fuzzy logic system methods. Journal of Intelligent Transportation Systems 12 (3), 102–112.

[61] Zheng, W., Lee, D.-H., Shi, Q., 2006. Short-term freeway traffic flow prediction: Bayesian combined neural network approach. Journal of Transportation Engineering 132 (2), 114–121.

Dataset	Method	Fuzzy AdaBoost	Fuzzy LogitBoost	Fuzzy Chi-RW	GHFRBS
$PIMA$	Fuzzy AdaBoost	( )	-	*	-
	Fuzzy LogitBoost	+	( )	+	*
	Fuzzy Chi-RW	*	-	( )	-
	GHFRBS	+	*	+	( )
$RING$	Fuzzy AdaBoost	( )	-	+	+
	Fuzzy LogitBoost	+	( )	+	+
	Fuzzy Chi-RW	-	-	( )	-
	GHFRBS	-	-	+	( )
$SPAMBASE$	Fuzzy AdaBoost	( )	-	-	-
	Fuzzy LogitBoost	+	( )	-	-
	Fuzzy Chi-RW	+	+	( )	-
	GHFRBS	+	+	+	( )
$SONAR$	Fuzzy AdaBoost	( )	-	*	-
	Fuzzy LogitBoost	+	( )	*	-
	Fuzzy Chi-RW	*	*	( )	-
	GHFRBS	+	+	+	( )
$TRAFFIC_5$	Fuzzy AdaBoost	( )	*	+	-
	Fuzzy LogitBoost	*	( )	+	-
	Fuzzy Chi-RW	-	-	( )	-
	GHFRBS	+	+	+	( )
$TRAFFIC_{15}$	Fuzzy AdaBoost	( )	-	*	-
	Fuzzy LogitBoost	+	( )	+	-
	Fuzzy Chi-RW	*	-	( )	-
	GHFRBS	+	+	+	( )
$TRAFFIC_{30}$	Fuzzy AdaBoost	( )	-	-	-
	Fuzzy LogitBoost	+	( )	*	-
	Fuzzy Chi-RW	+	*	( )	-
	GHFRBS	+	+	+	( )

Table 3: Student't t-test, performed over values obtained in the test set ($\overline{MAE_{tst}}$, in Table 2).

		F_1	O_1	S_1	ΔF_1	ΔO_1	ΔS_1	IF_1	OF_1	ΔIF_1	ΔOF_1	S_2	F_2	O_2	ΔS_2	ΔF_2	ΔO_2	IF_2	OF_2	ΔIF_2	ΔOF_2	S_3	F_3	O_3	ΔS_3	ΔF_3	ΔO_3
$TRAFFIC_5$	Ex1	10	8	-	-	-	9	6	-	-	2	-	-	13	11	-	7	3	12	-	5	-	4	-	1	-	-
	Ex2	-	-	-	2	-	-	-	4	1	-	-	-	7	-	-	-	-	-	-	-	-	-	5	6	-	3
	Ex3	-	-	4	5	-	1	6	-	-	-	-	7	12	-	-	2	-	-	10	9	8	11	3	-	-	-
	Ex4	-	1	8	-	6	-	-	-	-	11	2	5	14	3	7	12	10	-	9	-	-	-	4	-	-	13
	Ex5	-	-	-	-	-	-	-	-	-	-	-	-	4	-	2	-	-	-	-	-	-	-	3	-	1	-
$TRAFFIC_{15}$		F_1	O_1	S_1	ΔF_1	ΔO_1	ΔS_1	IF_1	OF_1	ΔIF_1	ΔOF_1	S_2	F_2	O_2	ΔS_2	ΔF_2	ΔO_2	IF_2	OF_2	ΔIF_2	ΔOF_2	S_3	F_3	O_3	ΔS_3	ΔF_3	ΔO_3
	Ex1	-	-	-	-	-	-	-	-	3	-	-	5	-	2	-	-	-	-	-	1	4	-	-	-	-	-
	Ex2	4	19	11	8	9	-	17	20	2	3	7	23	25	22	5	13	16	24	14	18	15	6	1	12	10	21
	Ex3	14	13	-	1	7	-	11	10	2	16	12	8	20	5	15	19	9	-	3	4	-	-	18	17	6	-
	Ex4	-	-	-	1	-	4	-	2	-	-	-	-	6	-	-	5	-	-	-	-	-	-	3	-	-	-
	Ex5	-	-	-	-	-	-	-	-	-	-	1	-	3	-	-	2	-	-	-	-	-	-	-	-	-	-
$TRAFFIC_{30}$		F_1	O_1	S_1	ΔF_1	ΔO_1	ΔS_1	IF_1	OF_1	ΔIF_1	ΔOF_1	S_2	F_2	O_2	ΔS_2	ΔF_2	ΔO_2	IF_2	OF_2	ΔIF_2	ΔOF_2	S_3	F_3	O_3	ΔS_3	ΔF_3	ΔO_3
	Ex1	-	-	-	-	-	-	-	-	-	-	2	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-
	Ex2	-	4	1	3	-	6	-	-	-	8	-	-	10	-	-	9	-	-	2	5	-	-	7	-	-	-
	Ex3	-	-	-	6	-	10	5	-	-	-	1	-	11	3	-	-	-	-	8	-	-	9	4	7	-	2
	Ex4	21	18	8	11	5	19	10	9	13	1	2	14	23	6	7	4	-	3	17	12	16	22	-	-	20	15
	Ex5	5	-	-	-	9	-	-	-	-	-	6	11	10	4	3	-	-	7	-	8	-	1	-	-	-	2

Table 4: Order in which each input variable has been selected in each execution of the GHFRBS. (-) means *not used*.