Zio Enrico

Baraldi Piero

Popescu Irina Crenguta
Department of Nuclear Engineering, Polytechnic of Milan, Milan, Italy

Optimising a fuzzy fault classification tree by a single-objective genetic algorithm

Keywords

fault classification, decision tree, fuzzy logic, genetic algorithm

Abstract

In this paper a single-objective Genetic Algorithm is exploited to optimise a Fuzzy Decision Tree for fault classification. The optimisation procedure is presented with respect to an ancillary classification problem built with artificial data. Work is in progress for the application of the proposed approach to a real fault classification problem.

1. Introduction

In recent years, many efforts have been devoted to the development of automatic diagnostic techniques based on statistical or geometric methods, neural networks, expert systems, fuzzy and neuro-fuzzy approaches [22], [11], [15], [5], [7]. These techniques have proven to be very effective but often remain "black boxes" as to the interpretation of the physical relationships underpinning the fault classification.

In an effort to overcome this limitation, a systematic approach to fault classification has been introduced by the authors leading to a Fuzzy Decision Tree (FDT) [25], [26]. The main advantages of the proposed approach from the operator point of view are the transparency of the resulting classification model and its visualization in the form of a DT [14], [20].

The construction of the Decision Tree (DT) is pursued starting from the fuzzy rules of a Fuzzy Rule Base (FRB) derived from a clustering algorithm tailored to fault classification [25]. To do this, every Fuzzy Set (FS) representing a deviation of the monitored signals in the respective ranges of variability (Universes of Discourse, UODs, in Fuzzy Logic terminology) is associated to a symptom of a fault class and the FRB of the model is translated into a Symptom Table in which the relationships between fault classes and symptoms are explicitly laid out.

In practice, however, it is often difficult to attribute the detected symptoms to a given fault class, given that one fault may cause several symptoms and dually a symptom may describe more than one possible fault.

To solve this problem, the relationships between fault classes and symptoms contained in the Symptoms Table are systematically represented in a DT, which is then quantified by applying the rules of Fuzzy Logic.

The design of the DT entails the successive consideration of the symptoms. These can be considered in different orders, leading to different structures of the DT and thus different classification performances. Hence, a combinatorial optimisation problem arises with regards to the DT design.

In this paper, a single-objective genetic algorithm search is devised to find the sequence of symptoms leading to the optimal configuration of the DT, i.e. that which achieves the maximum classification performance.

The paper is organized as follows. Section 2 illustrates the procedure adopted for the construction of the DT and its fuzzy quantification. In Section 3, the results related to its application to an artificial case study regarding the classification of data randomly extracted from six different Gaussian distributions are reported. Section 4 presents the optimisation of the FDT by a single-objective genetic algorithm maximizing the percentage of correct classifications. A synthetic

discussion of the findings of the work is provided in the last Section.

2. From a Fuzzy Classification Model to a Fuzzy decision Tree

Let us consider an industrial system or plant whose "state of health" is monitored by a set of sensors, which collect the relevant parameters data at a given frequency. These data (also called "signals") provide a picture of the health state of the plant. A particular picture corresponds to the plant functioning in nominal conditions, with all the signals within their design envelope. Deviations from the nominal states are due to faults of different types (classes), which may occur to the components of the plant, leading to different "pictures" of the monitored signals.

When a generic fault of class Γ_j, $j=1,...,c$, occurs in the plant, corresponding representative symptoms are observed by the monitoring system, in terms of variations in the signal values. A symptom associated to the fault of class Γ_j is a deviation of a monitored signal from its reference value, outside of the allowed design envelope. The objective of fault identification is to build a system capable of recognizing the fault as of class Γ_j on the basis of the measured symptoms, i.e. the monitored signals.

In this work, we assume that the classification of the fault is performed by applying a previously built FRB (for example, a possible method for building an FRB from available pre-classified data is presented in [25]). Such FRB has one fuzzy rule for each fault class: the generic rule j associates the symptoms of the monitored signals (input data) to the fault class Γ_j. In the fuzzy rules, each one of the FSs of the antecedents describes a deviation of a monitored signal, i.e. a symptom, except those FSs representing the still nominal conditions of those monitored signals which are unaffected by the particular fault. Correspondingly, the generic FS X_{pj} associated to the p-th antecedent in rule j, $p=1,...,n$, $j=1,...,c$, represents a symptom for the class of faults Γ_j.

Notice that the relations between fault classes and symptoms (signals deviations) are not univocal: the faults of a given class may initiate several symptoms and in turn one symptom may be a legitimate representative of several possible fault classes.

On the other hand, an adequately designed monitoring system should be capable of associating to each fault class a unique set of symptoms (signal deviations). This leads to a Symptom Table such as the one reported in *Table 1*, where S_r, $r=1,...,s$, denotes the generic symptom.

The binary vector $\sigma_j=[I_{j1},I_{j2},...,I_{js}]$ represents the reference symptoms vector for fault class Γ_j, $j=1,...,c$. Each element I_{jr} is a binary value that corresponds to the presence or absence of symptom r when a fault of class Γ_j has occurred, $r=1,...,s$, $j=1,...,c$.

Table 1. Symptom Table: Reference relations between fault classes and symptoms [13]

Fault Class	Symptom Type					
	S_1	S_2	⋯	S_r	⋯	S_s
Γ_1	I_{11}	I_{12}	⋯	I_{1r}	⋯	I_{1s}
Γ_2	I_{21}	I_{22}	⋯	I_{2r}	⋯	I_{2s}
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Γ_j	I_{j1}	I_{j2}	⋯	I_{jr}	⋯	I_{js}
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Γ_c	I_{c1}	I_{c2}	⋯	I_{cr}	⋯	I_{cs}

During operation, the monitored signals could then be translated into an observation vector $\sigma'=(I_1',I_2',...,I_s')$, which carries the information on the presence or absence of the symptoms. As explained earlier, a symptom is present in the system if its representative measured signal has deviated from its nominal value beyond the design envelope. For example, a patient has the symptom "fever" if his or her monitored temperature rises to a "high" value, i.e. above 37°C.

However, in practice often the presence or absence of a symptom remains uncertain and ambiguous due to the complexity of the non-linear signal behaviours associated to the various faults, to the measurement errors of the monitoring sensors and to the imprecise and ambiguous definition of the signal deviation ranges and the associated linguistic labels [25]. To reflect this uncertainty, a fuzzy observation vector $\sigma_f'=(\mu_1',\mu_2',...,\mu_s')$ is associated to a pattern of

deviations of the monitored signals measured in correspondence of a given fault, where $\mu_r^{'}$, $r=1,...,s$, is the value of the membership of the FS corresponding to the symptom S_r and gives the degree with which it is present in the monitored situation being examined.
The fault identification problem is then to identify which fault class is occurring in the plant on the basis of the fuzzy observation vector $\sigma_f^{'}$. To tackle this problem a systematic procedure for constructing a DT is presented below [26].

2.1. Decision Tree

Based on the Symptom *Table 1*, a complete DT can be diagrammed by examining all s symptoms one by one [7]. Taking into consideration all possible combinations of symptoms, the DT will have 2^s branches given that each of the s symptoms can be either present or absent. On the other hand, only one combination of symptoms corresponds to a given fault: thus, only c of the 2^s tree branches correspond to a class while the remaining $2^s - c$ combinations of symptoms cannot be associated to a class.
For building a smaller, more transparent and easier to interpret DT, two main hypotheses are assumed [26], [13]:
- if a symptom is indicated as present in the measured observation vector $\sigma^{'}$, it is certainly present in the system;
- the presence of a single symptom characteristic of a fault suffices to conclude that the measured pattern of signals belongs to that fault class.
In this context, defining an "unwanted" symptom as a symptom that, although not present in the system, somehow is present by mistake in the observation vector and a "missing" symptom as a symptom that is not observed although it is present in the system [13], the first hypothesis can be called of "impossibility of unwanted symptoms" and the second of "possibility of missing symptoms".
The procedure for building the DT proceeds as follows:
1. A root node is placed at the top of the tree. This node refers to all possible fault classes identified for the system under analysis.
2. A symptom from the Symptom Table is associated to this node.
3. The root node is split into two branches: the left corresponding to the presence of the symptom, the right to the absence of the symptom.
4. The fault classes for which the symptom is present are associated to a node under the left branch. If only one fault class is found to contain the symptom, then the associated node is a terminal leaf of the branch and its identification is guaranteed by the fact that it has been assumed that a symptom that is absent in the system cannot be indicated as present (impossibility of unwanted symptoms hypothesis). The fault class associated to the identified leaf may be also associated to other leaves, at the end of other branches in the tree. This accounts for the possibility that a symptom is not indicated as present by the monitoring system although it actually is (possibility of missing symptoms hypothesis). If more than one fault class are associated to the node characterized by the identified symptom, a new symptom is searched in the Symptom Table and associated to the node in order to differentiate between the identified fault classes. To select the new differentiating symptom, the previous procedure is applied, starting from step 2.
5. The right branch from the root node is further developed by first adding a node associated to all possible fault classes. This node is then treated as a local root node to which the branching procedure is applied starting from step 2.
6. The tree development terminates when all symptoms have been considered and their associated branches developed down to the distinguishing leaves of the individual fault classes.
A path through the branches of the tree, from the root node to a leaf, identifies a crisp observation vector $\sigma^{'}$ of symptoms representative of the fault class associated to the corresponding leaf. As pointed out above, different paths may lead to different leaves associated to the same fault class, due to the possibility of missing symptoms.
In operation, the DT gives the correct diagnosis when the measured symptom vector matches completely with the reference symptom vector of a fault class; on the contrary, the diagnosis is conservative in case of a missing symptom, i.e. it is not necessary to have all the symptoms to diagnose the fault.
Finally, in case of unwanted symptoms, the classification is driven by the structure of the tree and the classification will be wrong if the first symptom considered is an unwanted symptom.
From the above it appears that an issue of adequate DT design arises with respect to the order with which the successive symptoms are considered for optimal classification performance.

2.2. Classification by the FDT

In the realistic case of ambiguity in the actual presence or absence of a symptom, in correspondence of a given pattern of signal deviations the degree of activation of

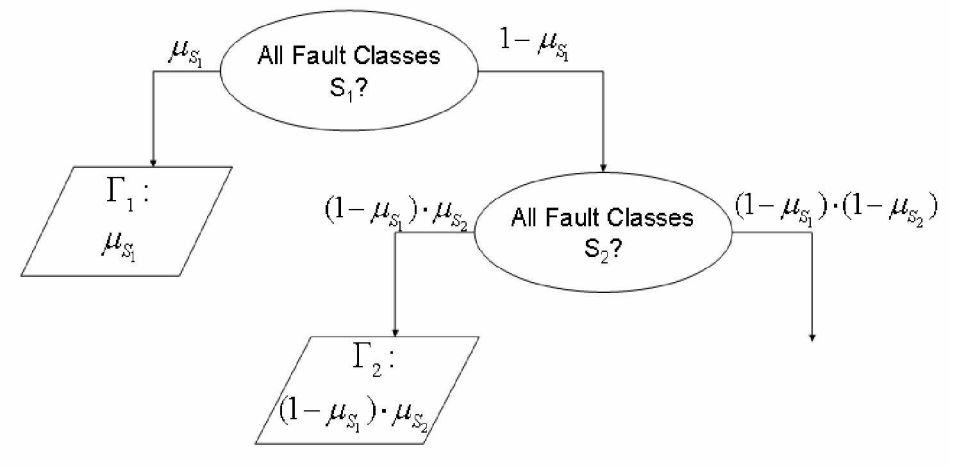

Figure 1. Propagation of fuzzy information through the DT

each symptom S_r, $r=1,...,s$, is computed from the MF of the corresponding FS. The DT then becomes a FDT and the possibility classification of a given pattern of measured signal deviations is performed by proceeding through all the branches of the tree and computing the MFs to each fault class, at the tree leaves.

The symptoms degrees of activation are then propagated through the DT according to the rules of FS theory. In particular, the logic operator of negation of a symptom S_r is implemented by $\left(1-\mu_{S_r}\right)$ in the right branch corresponding to the absence of the symptom whereas its complement μ_{S_r} is propagated along the left branch associated to its presence (*Figure 1*). The connection between two nodes of the tree represent a logic operator of intersection (*and*), implemented as the algebraic product of the membership values in this work.

Finally, since more than one terminal leaf can indicate the same class, the final membership to a given class is computed through the logic operation of union (*or*) of all the leaves associated to that class. The logic operator *or* is here implemented as the MFs sum limited to 1, accordingly to the rules of FS arithmetic.

As mentioned at the end of Section 2.1, different sequences of symptoms lead to different DTs and, implicitly, to different classification performances. For realistic problems, the number of possible sequences of symptoms for building the DT is combinatorial, so that a trial and error process for finding the optimal structure of the tree, i.e. that which allows obtaining the maximum classification performance, would not be practical. For example, the number of possible sequences of a group of 15 symptoms, would be approximately 10^{11} and to each sequence corresponds a different DT whose classification performance must be evaluated.

3. Application of the FDT to an artificial case study

The classification approach has been applied to the artificial four-dimensional, six-classes data set of *Figure 2*. The data have been obtained by random sampling from 6 different Gaussian distributions and can be assumed to represent the system response signals resulting from 6 different types of system faults.

The previously illustrated procedure for classifying the data into the six classes (Section 2) consists of two main steps: the first one is the building of the FRB, i.e. a set of transparent and accurate fuzzy rules and the second one is the construction and quantification of the corresponding FDT.

A fuzzy clustering – based method has been used for obtaining the FRB from available pre-classified data [25]. The resulting FRB is composed of $c=6$ rules, one for each class *(Table 2*).

To build the associated DT, first each antecedent of the rules in the FRB is associated to a symptom, resulting in 15 possible symptoms, indicated as S_i, $i=1,2,...,15$, in *Table 2*. This allows the translation of the FRB in the Symptom *Table 3*.

Then, by applying the steps 1.– 6. of the procedure for building the DT (Section 2.1) on the sequence of symptoms $\Sigma_0=[S_1;S_2;\ ...\ ;S_{15}]$, one obtains the DT reported in *Figure 3*.

The possibility quantification of the degree of membership to different classes can be performed as described in Section 2.2, i.e. propagating through the branches of the tree the degree of activation of each symptom according to the rules of Fuzzy Logic. The final assignment of an incoming pattern of signals to a class is conservatively realized as follows:

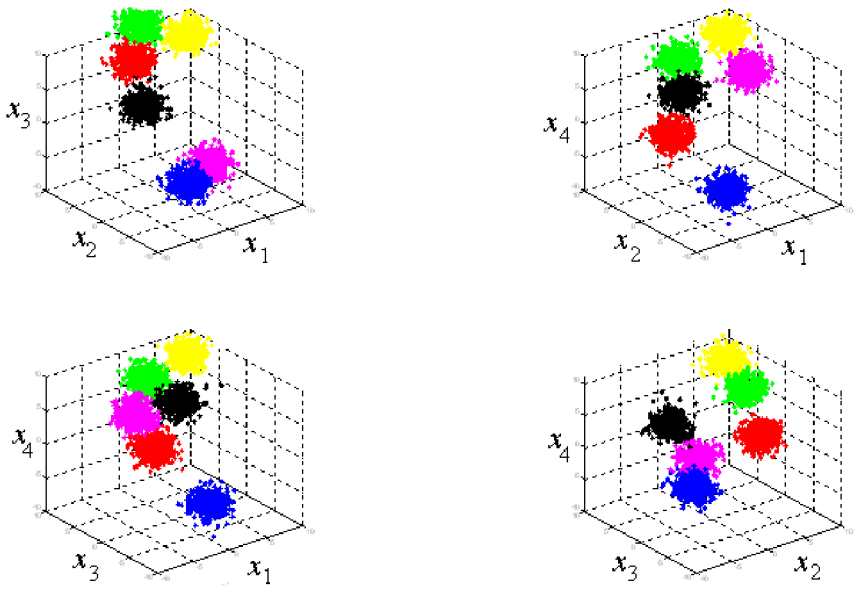

Figure 2. Four-dimensional data set comprised of six classes

Table 2. The Table of rules of the FRB

Rule		x_1	x_2	x_3	x_4		Γ_1	Γ_2	Γ_3	Γ_4	Γ_5	Γ_6
1		Low S_1	Low S_4	Low S_9	Medium S_{12}		Yes	No	No	No	No	No
2		High S_2	Medium S_5	Medium S_{10}	High S_{13}		No	Yes	No	No	No	No
3	IF	High S_2	High S_6	Medium S_{10}	High S_{13}	THEN	No	No	Yes	No	No	No
4		High S_2	Low S_4	Medium S_{10}	Low S_{14}		No	No	No	Yes	No	No
5		High S_2	Higher S_7	Medium S_{10}	Medium S_{12}		No	No	No	No	Yes	No
6		Higher S_3	Highest S_8	High S_{11}	Higher S_{15}		No	No	No	No	No	Yes

Table 3. Symptom Table

	x_1 low	x_1 high	x_1 higher	x_2 low	x_2 medium	x_2 high	x_2 higher	x_2 highest	x_3 low	x_3 medium	x_3 high	x_4 medium	x_4 high	x_4 low	x_4 higher
	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}
Γ_1	1	0	0	1	0	0	0	0	1	0	0	1	0	0	0
Γ_2	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0
Γ_3	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0
Γ_4	0	1	0	1	0	0	0	0	0	1	0	0	0	1	0
Γ_5	0	1	0	0	0	0	1	0	0	1	0	1	0	0	0
Γ_6	0	0	1	0	0	0	0	1	0	0	1	0	0	0	1

- the pattern is declared assigned to a class (in possibility terms), if the membership grade to the respective class is larger than a confidence threshold γ (here chosen equal to 0.6);
- the pattern is declared 'atypical', if none of the membership grades is larger than γ ;
- the pattern is declared 'ambiguous', if more than one membership grade is larger than γ .

A test on a set of 600 data has resulted in only 40.67% correct classifications to the six fault classes, while 10.5% of the data are considered as atypical, 2.33% as ambiguous and 46.5% are assigned to the wrong class. The obtained performance is obviously unacceptable and motivates the search for an optimal or near-optimal sequence of symptoms upon which to build the DT. The objective of the optimisation algorithm is to find the sequence of symptoms that leads to the DT with the best classification performance in terms of percentage of correct classifications. The number of possible sequences of symptoms is 15! (¼10^{11}), in Section 4, this combinatorial optimisation problem is tackled by a single-objective genetic algorithm.

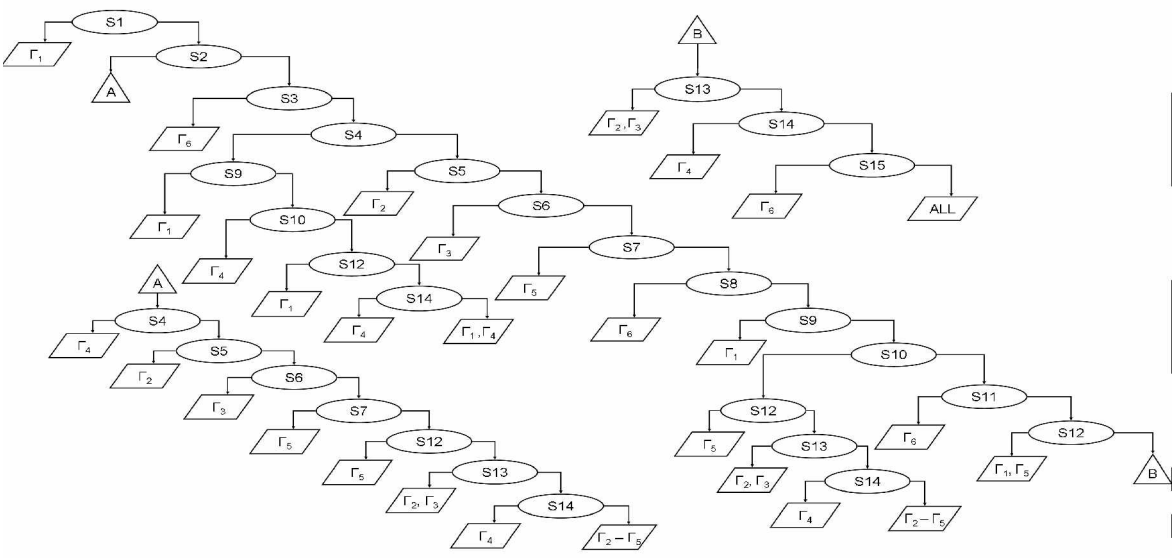

Figure 3. DT for classification, built with the ordered sequence of symptoms Σ_0

4. Genetic algorithm optimisation of the decision tree design

, a procedure based on a single-objective genetic In this Section algorithm (Appendix A) is carried out for determining the sequence of symptoms to which corresponds the FDT with the maximum classification performance. The genetic algorithm can be seen as performing a wrapper search [12] around the classification algorithm (*Figure 4*): the symptoms sequence selected during the search is evaluated using as criterion (fitness) the percentage of correct classified data achieved by the FDT itself.

The data and rules of the genetic algorithm search are given in *Table 4*. These parameters have been established through a systematic procedure of experimentation. The objective (fitness) function to be maximized is the percentage of correct data classifications; the decision variable is the symptoms sequence.

Table 4. GA run parameters

Number of chromosomes in the population	100
Number of generations (termination criterion)	50
Selection	Standard Roulette
Replacement	Children - Parents
Mutation probability	0.01
Crossover probability (one-site)	1

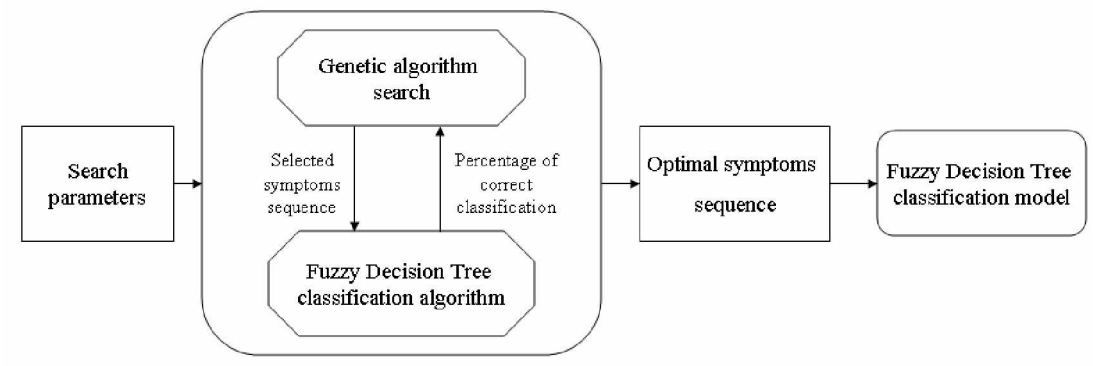

Figure 4. Single-objective genetic algorithm "wrapper" search

Each chromosome is made up by 15 genes, one gene for each symptom. The single gene can assume any integer value in [0,15] that encodes the "swap" position of the symptom along the sequence. An

example of a chromosome coding a particular sequence is given in *Figure 5*. To decode the chromosome in its corresponding symptom sequence, a 15 – steps procedure is performed, one for each gene. At the generic step $i = 1, 2, ..., 15$, the ordered sequence Σ_{i-1} and the value k contained in the i-th gene are considered: the symptom in the i-th position of Σ_{i-1} is swapped with the symptom in the k-th position of the sequence. For example in the first step of Figure 5, the value 7 in gene 1 means that the symptom S_1 is placed in position 7 of the sequence and simultaneously the symptom that occupied position 7 is swapped to position 1. This operation is carried out until the 15th gene of the chromosome is worked out, leading to the final sequence:

$$\Sigma_{15} = [S_3; S_{11}; S_5; S_{12}; S_6; S_8; S_7; S_2; S_1; S_{10}; S_{13}; S_9; S_{14}; S_4; S_{15}]$$

Note that this original chromosome random design leads to a coherent symptom sequence, i.e. without repetition of symptoms, thus avoiding computationally burdensome chromosome coherence checking a posterior.

The optimal sequence found at convergence of the genetic algorithm is:

$$\Sigma_1 = [S_4; S_6; S_7; S_3; S_{12}; S_{10}; S_{13}; S_{15}; S_5; S_1; S_{14}; S_{11}; S_8; S_9; S_2]$$

The FDT built following this sequence ends into 46 leaves and achieves a classification performance of 91.34%, while 5.33% of the data are considered as atypical, 0.33% as ambiguous and only 3% are assigned to the wrong class.

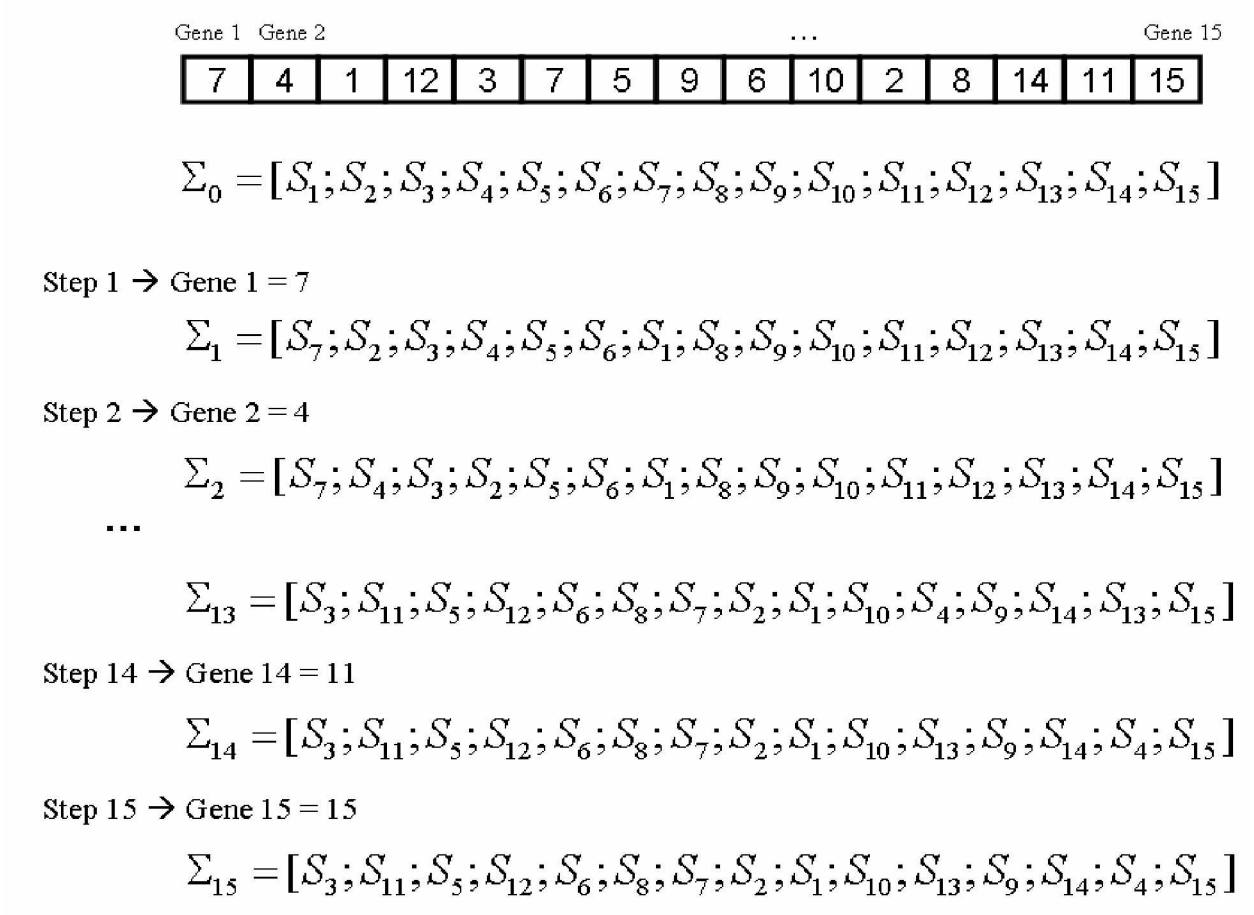

Figure 5. Example of a chromosome and the corresponding sequence

5. Conclusion

In realistic applications, fault classification is usually based on ambiguous information, which can be effectively handled within a fuzzy logic framework. A Fuzzy Decision Tree can then be built to logically structure the uncertain information available. To each fault class corresponds a classification rule, with mono-dimensional Fuzzy Sets representing the characteristic symptoms for the corresponding fault class.

The classification performance by the resulting FDT is dependent on the order in which the symptoms are considered in the building procedure of the DT. This leads to an optimisation problem with respect to the construction of the tree. In this work, this problem has been tackled by means of a single-objective genetic algorithm in which the different sequences of symptoms are coded into the chromosomes of the genetic population by an original procedure, which guarantees coherence, i.e. no repetition of symptoms in the sequence.

The genetic algorithm-based optimisation procedure developed has been successfully applied to a test case regarding the development of Fuzzy Decision Trees for the classification of artificial data. Undergoing research concerns the application of the developed classification procedure to a real diagnostic problem.

Appendix A: A brief recall of Genetic Algorithms

In the following, only a concise snapshot is provided on the basics of genetic algorithms optimisation. For an extensive and detailed presentation of this computational paradigm, the interested reader is invited to consult the available copious literature [18], [17], [3], [6], [9], [4], [2].

Genetic Algorithms (GAs) are optimisation methods aiming at finding the global optimum of a set of real objective functions, $F \equiv \{f(\cdot)\}$, of one or more decision variables, $U \equiv \{u\}$, possibly subject to various linear or non linear constraints. Their main properties are that the search is conducted i) using a population of multiple solution points or candidates, ii) using operations inspired by the evolution of species, such as breeding and genetic mutation, iii) using probabilistic operations, iv) using only information on the objective or search function and not on its derivatives [16].

GAs owe their name to their operational similarities with the biological and behavioural phenomena of living beings. After the pioneering theoretical work by John Holland [10], in the last decade a flourishing literature has been devoted to their application to real problems. The basics of the method may be found in Goldberg [8]; some applications in various contexts are included in Chambers [1].

The terminology adopted in GAs contains many terms borrowed from biology, suitably redefined to fit the algorithmic context. Thus, GAs operate on a set of (artificial) chromosomes, which are strings of numbers, generally sequences of binary digits 0 and 1. If the objective function of the optimisation has many arguments (typically called control factors or decision variables), each string is partitioned in as many substrings of assigned lengths, one for each argument and, correspondingly, we say that each chromosome is partitioned in (artificial) genes. The genes constitute the so-called genotype of the chromosome and the substrings, when decoded in real numbers, constitute its phenotype. When the objective functions are evaluated in correspondence of a set of values of the control factors of a chromosome, its values are called the fitness of that chromosome. Thus, each chromosome gives rise to a trial solution to the problem at hand in terms of a set of values of its control factors.

The GA search is performed by constructing a sequence of populations of chromosomes, the individuals of each population being the children of those of the previous population and the parents of those of the successive population. The initial population is generated by randomly sampling the bits of all the strings. At each step, the new population is then obtained by manipulating the strings of the old population in order to arrive at a new population hopefully characterized by increased mean fitness. This sequence continues until a termination criterion is reached. As for the natural selection, the string manipulation consists in selecting and mating pairs of chromosomes in order to groom chromosomes of the next population. This is done by repeatedly performing on the strings the four fundamental operations of reproduction, crossover, replacement and mutation, all based on random sampling: the parents' selection step determines the individuals which participate in the reproduction phase; reproduction itself allows the exchange of already existing genes whereas mutation introduces new genetic material; the substitution defines the individuals for the next population. This way of proceeding enables to efficiently arrive at optimal or near-optimal solutions.

With regards to their performance, it is acknowledged that GAs takes a more global view of the search space than many other optimisation methods. The main advantages are i) fast convergence to near global optimum, ii) superior global searching capability in complicated search spaces, iii) applicability even when gradient information is not readily achievable.

References

[1] Chambers, L. (1995). Practical handbook of genetic algorithms: applications Vol. I; new frontiers Vol. II. CRC Press, 1995.

[2] Coello, C. A. (2000). Treating Constraints as Objectives for Single-Objective Evolutionary Optimisation. *Engineering Optimisation*. Vol. 32, 275-308.

[3] De Jong, K. (1990). *Genetic-algorithm-based learning, Machine learning: an artificial intelligence approach*. Morgan Kaufmann Publishers Inc., San Francisco.

[4] Deb, K. (1999). Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation Journal.* Vol. 7, 205-230.

[5] Du, J. & Er, M. J. (2004). Fault Diagnosis in Air-Handling Unit System Using Dynamic Fuzzy Neural Network. *Proceedings of the sixth International FLINS Conference*, 483-488.

[6] Eiben, D. A. E. & Smith, J. E. (2003). *Introduction to Evolutionary Computing*. Springer.

[7] Goddu, G., Li, B. Chow, M. Y. & Hung, J. C. (1998). Motor Bearing Fault Diagnosis by a Fundamental Frequency Amplitude Based Fuzzy Decision System. *IEEE Trans. on Systems, Man, Cybernetics*, 1961-1965.

[8] Goldberg, D. E. (1989). *Genetic algorithms in search, optimisation, and machine learning"*, Addison-Wesley Publ. Co.

[9] Haupt, R. L. & Haupt, S. H. (1998). *Practical genetic algorithms.* J. Wiley & Sons, New York.

[10] Holland, J. H. (1975). Adaptation in natural and artificial systems: an introductory analysis with application to biology. *Control and Artificial Intelligence.* MIT Press, 4-th edition.

[11] Isermann, R. & Ballé, P. (1997). Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice.* No. 5, 709-719.

[12] Keller, J.M., Gray, M. R. & Givens J. A. (1985). A fuzzy k-nearest neighbor algorithm. *IEEE Trans. Syst., Man, Cybern.* SMC-15 (4), 580-585.

[13] Kim, K. O. & Zuo, M. J. *Two Fault Classification Methods for Large Systems when Available Data is Limited.*

[14] Lee, C., Alena, R. L. & Robinson, P. (2005). Migrating Fault Trees To Decision Trees For Real Time Fault Detection On International Space Station. *Aerospace, 2005 IEEE Conference,* 1-6.

[15] Leonhardt, S. & Ayoubi, M. (1997). Methods of fault diagnosis. *Control Engineering Practice*. No. 5, 683-692.

[16] Marseguerra, M., Zio, E. & Podofillini, L. (2003). *Model parameters estimation and sensitivity by genetic algorithms*. *Annals of Nuclear Energy* 30, 1437-1456.

[17] Michaelwicz, Z. (1999). *Genetic Algorithms + Data Structures = Evolution Programs.* Springer, New York.

[18] Mitchell, M. (1996). *An introduction to genetic algorithms,* MIT Press, London.

[19] Puska, E. & Noemann, S. (2002). 3-d core studies for Hambo simulator. *Proceedings of Presentation on Man-Machine System Research, Enlarged Halden Programme Group Meeting*. Vol. 2, September.

[20] Rovnyak, S., Kretsinger, S., Thorp, J. & Brown, D. (1994). Decision trees for real-time transient stability prediction. Power Systems. *IEEE Transactions on Systems*, Vol. 9, No.3, 1417-1426.

[21] Roverso, D. (2003). Fault diagnosis with the Aladdin transient classifier. *Proceedings of System Diagnosis and Prognosis: Security and Condition Monitoring Issues III, AeroSense2003, Aerospace and Defence Sensing and Control Technologies Symposium*, Orlando, FL, USA.

[22] Uhrig, R. E. & Tsoukalas, L. H. (1999). *Soft computing technologies in nuclear engineering applications.* Progress in Nuclear Energy, Elsevier Science, 34, 13-75.

[23] Zio, E., Baraldi, P. & Roverso, D. (2005). An extended classificability index for feature selection in nuclear transients. *Annals of Nuclear Energy.* No. 32, 1632-1649.

[24] Zio, E., Baraldi, P. & Pedroni, N. (2006). *Selecting Features for Nuclear Transients Classification by means of Genetic Algorithms. IEEE Transaction on Nuclear Science*. Vol. 53, No. 3, 1479-1493.

[25] Zio, E., Baraldi, P. & Popescu, I. C. From fuzzy clustering to a rule-based model for fault classification.

[26] Zio, E. Baraldi, P. & Popescu, I. C. A Fuzzy Decision Tree for Fault Classification.