View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Servicio de Coordinación de Bibliotecas de la Universidad Politécnica de Madrid

Evolutionary Generation of Fuzzy Knowledge Bases for Diagnosing Monitored Railway Systems

A. Carrascal [1], A. Díez [1], J.M. Font [2], D. Manrique [2]

[1] Fundación Fatronik-Tecnalia, Paseo Mikeletegi 7, Parque Tecnológico, 20009 Donostia, Spain
[2] Departamento de Inteligencia Artificial, Facultad de Informática, UPM, Madrid, Spain

ABSTRACT

Classical approaches when building diagnosis and monitoring systems are rule-based systems, which allow the representation of existing knowledge by using rules. There are several techniques that facilitate this task, such as fuzzy logic, which allows knowledge to be modeled in an intuitive way. Nevertheless, sometimes it is not easy to define the fuzzy rule set that represents the knowledge from a certain domain. To overcome this drawback, an evolutionary system based on a grammar guided genetic programming technique for the automatic generation of fuzzy knowledge bases has been employed in diagnosing monitored railway networks. This system employs a grammar-based initialization method and both, grammar-based crossover and mutation operators, to achieve well balanced exploitation and exploration capabilities of the search space, assuring high convergence speed and low chance of getting trapped in local optima. Tests have been carried out in a real-world train monitoring domain, in which a sensor network is periodically monitoring critical train components. Results achieved show that this evolutionary system accomplishes an automatic knowledge discovery process, which is able to build a fuzzy rule base that represents the expert knowledge extracted from the domain of the detection of abnormal train conditions.

Keywords: Monitoring and Diagnosis Systems, Grammar Guided Genetic Programming, Fuzzy Logic, Rule-Based System, Evolutionary computation, Knowledge Discovery.

1. INTRODUCTION

Because of the recent technological revolution occurred in the industrial sector, raising an appropriate manual maintenance process has turned out increasingly difficult. Thus, the amount of information about the state of a monitored system is continuously increasing, quickly exceeding the capacity of maintenance technicians. While the industry is undergoing a technological revolution, new reactive, proactive and predictive maintenance approaches are being developed.

The success of monitoring and intelligent diagnosis systems relies on the use of the knowledge regarding existing domains (Knowledge Based Systems, or KBS) [1]. In this kind of domains, the main difficulty regarding failure and anomaly detection is how to make the expert knowledge explicit and how to model it. Rule-based knowledge modeling is one of the most common approaches [2]. Nevertheless, there are several domains where this approach cannot be easily applied, due to either, non-existing previous expert knowledge or overly complex knowledge base management [3].

Fuzzy logic allows to model quantitative concepts in a qualitative manner so that knowledge can be stored in a way close to the human reasoning. Knowledge underlying in control processes, condition monitoring and diagnosis systems can be also modelled into a set of fuzzy rules. Thus, there are many industrial applications which have successfully employed fuzzy logic [4].

If no prior knowledge exists, the rules and membership functions can be directly extracted from the data [5],[6] or by means of rule generation techniques. The point for rule generation techniques is to search for solutions in complex spaces, so that they demand expensive computational and time costs. The evolutionary computation, and particularly genetic programming, allows an outstanding exploration of solution spaces [7]. Genetic programming (GP) is a means of automatically generating computer programs by employing operations inspired by biological evolution [8]. First, the initial population is randomly generated, and then genetic operators, such as selection, crossover, mutation and replacement, are executed to breed a population of trial solutions that improves over time [9].

This paper employs a grammar-guided genetic programming (GGGP) [10] system, which is mainly based on the so-called grammar-based crossover and mutation operators, which most influence the evolution process. This crossover operator has been chosen because it strikes a good balance between search space exploration and exploitation capabilities and, therefore, enhances the GGGP system performance. The mutation operator that has been chosen does not generate illegal individuals, but individuals that match the syntactical constraints of the context-free grammar that defines the programs to be handled. The combined usage of these operators in the same GGGP system has been empirically demonstrated to provide a higher convergence speed and a lower likelihood of getting trapped in local optima than other related approaches.

The monitoring domain presented in this work concerns the railway domain. It is an especially critical domain in which is really important to assure the safety for every journey, for both passengers and cargo; which implies that all the components embedded into the train accomplish some reliability standards. In such domain, an exhaustive control of the life cycle parameters of the train components has to be carried out, guaranteeing correct operation working for all of them throughout their service life-time. This control can be carried out by employing a fuzzy knowledge-based system, composed by monitoring and diagnosing rules, extracted by means of the evolutionary system for the automatic generation of fuzzy rules employed in the testing procedures.

Section 2 in this paper describes diagnosis systems based on fuzzy logic. The monitored system belonging to the railway domain is explained in section 3. Section 4 details the automatic rule extraction process by means of genetic programming. Section 5 shows the results obtained from the testing procedure. Finally, conclusions are exposed in section 6.

2. FUZZY LOGIC BASED DIAGNOSIS SYSTEMS

Fuzzy logic principles, defined by Zadeh in the middle 60's, propose a solution to the modelling of vague or imprecise concepts managed by the human reasoning. The modelling of these concepts usually associates a quantitative value with a quantitative magnitude, which is close to the intuitive perception found in natural language. In this manner, fuzzy logic stands as a mathematical framework for studying vague concepts and phenomena. Applications of fuzzy logic can be found in a variety of domains: software sensors, alarming systems, process diagnostic, process control at different control levels, quality control, and production scheduling.

Concepts are modelled in fuzzy logic as fuzzy variables. The domain of every fuzzy variable is constituted by several fuzzy sets marked with fuzzy labels, also known as linguistic labels. A crisp value from a fuzzy variable is transformed into a membership degree to a fuzzy set through a membership function related to that fuzzy set. This is called the fuzzyfication process. Figure 1 shows the example of fuzzy definition of a variable called *temperature*. The variable is related to three fuzzy sets (low, medium and high) whose membership degrees are defined within the closed interval [0,1].

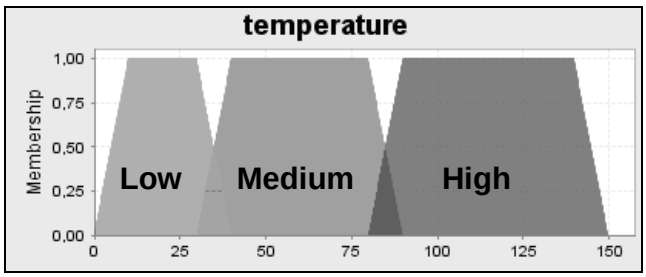

Figure 1: Example of fuzzy variable definition

The shape of the membership functions can generally be arbitrary. Typical are trapezoidal, triangular and Gaussian shapes. The usage of rectangular shapes allows the definition of disjoint linguistic values which implement the traditional bivalent logic, where the membership of a variable to a given set is an assertion that can only be false (0) or true (1).

Fuzzy knowledge-based systems are composed by a set of conditional rules in the form IF ... THEN. The inference mechanisms of fuzzy logic process these rules by following these steps:

- Premise Evaluation: The premise evaluation combines the membership degrees of the individual rule premise terms. This combination is made through the fuzzy logic operators: t-norm (T), t-conorm (S) and negation (N). These operators, inspired in the traditional logic operators AND, OR and NOT, respectively, are implemented in such a way that they are compatible with the quantitative membership degrees to fuzzy sets. E.g., the following expressions are valid definitions for the t-norm and the t-conorm, which are dual with regard to the negation N(x)=1-x:

$$\text{T (A,B): } \mu_{A\cap B} = \min(\mu_A, \mu_B) \text{ , S (A,B): } \mu_{A\cup B} = \max(\mu_A, \mu_B)$$

- Rule Activation: Application of the rule fulfilment to the rule consequences. Typical methods are a limiting or a scaling of the output membership functions.
- Consequent Accumulation: For every linguistic output variable the activated output membership function is combined. Maximum operation and membership function aggregations are the more common approaches. The result of the accumulation is then a linguistic output variable as a fuzzy set.
- Defuzzification: The output fuzzy set is converted to a crisp value by a process opposite to the fuzzyfication. Possible methods are the maximum, centre of gravity or mean of maximum.

The standard fuzzy system is typically reduced in fault diagnosis applications. The main reason for this is that the desired output of the diagnosis is a fault measure representing a gradual measure for the possibility of the corresponding fault, instead of an arbitrary value of a continuous variable. If the observed symptoms are far apart from the linguistically defined labels, this fault measure will be close to zero, whereas a perfect match will yield a fault measure of one.

3. CONDITION MONITORING SYSTEM

The monitored system based on intelligent diagnostics that has been used for the current study is related to one of the most critical components from the last generation, made by CAF [11] company: the self-propelled, dual voltage electric train units with a variable gauge system (ATPRD). As a safety measure, the ATPRD incorporates an ATMS (Acceleration and Temperature Monitoring System) equipment, developed by CAF; which allows to know the temperature and acceleration inside the train motion units, called bogies, at any time. The importance of these component measurements is critical, since the failures that can take place in the trains are mainly associated to anomalous behaviours inside the bogies.

There are several sensors monitoring the acceleration and temperature in the bogies, strategically replicated and placed over the train. Every 5 minutes during a train journey, sensors acquire readings of those parameters which are forwarded by means of a GSM connection with the train. Such information is registered and stored in a database to provide the needed data input to our approach. Sensors are distributed in 8 bogies per train, as it is showed in Figure 2. Each bogie has 32 sensors which can be divided in five groups: internal and external wheel groups with 16 temperature sensors installed on the wheels (4 per wheel), cylindrical and conic hollow shaft groups with 8 temperature sensors installed on the hollow shafts (HS hereafter), and finally, reduction gear group with 5 temperature sensors.

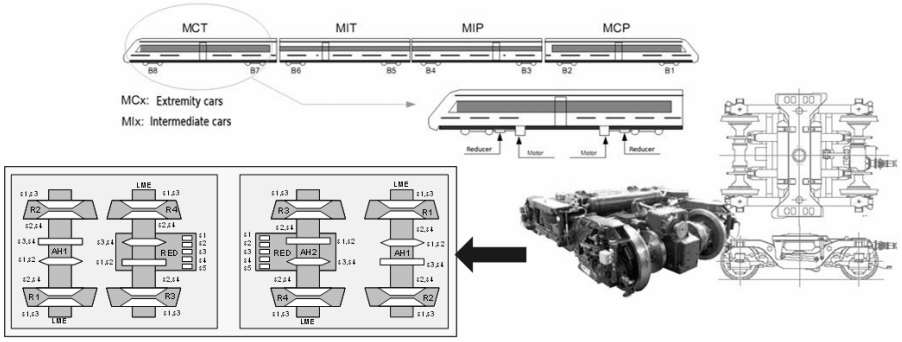

Figure 2: Bogies and sensors distribution in the ATPRD unit.

3.1.Fuzzy variables

In order to easily identify journey anomalies a set of quantitative variables has been defined to characterize the behaviour of each bogie during every journey. These derived parameters also eliminate the temporal dependency of collected sensor data; so, for each journey, each derived parameter has an associated value. Each of these parameters is related to a fuzzy domain composed by fuzzy sets. These fuzzy sets have been obtained as a result of the analysis of the histograms associated with the distributions of values taken by each derived parameter for the set of analyzed journeys. The histogram analysis procedure together with the knowledge gathered from the domain experts, allows the identification of interesting regions within the values domain in order to assign linguistic labels to each of the identified fuzzy sets.

The fuzzy derived parameters used in this study are the following:

- Correlation of a sensor with its pair: Pair sensors are sensors physically located very close, so that the correlation between their readings is used to validate its correct operation.
- Volatility of a sensor: this variable measures the variability of a certain sensor as the absolute differences mean.

- Sensors mean group.
- Mean square of sensors group: the mean of a group of sensors squared.
- Maximum absolute value of a subgroup of sensors: this derived parameter allows identifying atypical values measured by the sensors.
- Percentiles (10, 20, 80, and 90) of a group of sensors: P10 and P20 allow identifying low sensor values, whereas P80 and P90 are intended to identify high sensor values.

Considering the number of sensors and groups, the total number of fuzzy derived parameters obtained is 65. The next step in the diagnosis system design process is to codify into fuzzy rules the knowledge from the underlying thermodynamic model of the monitored system. In this case, an evolutionary system for the automatic generation of fuzzy knowledge-bases through genetic programming techniques has been used to accomplish this task.

4. EVOLUTIVE FUZZY KNOWLEDGE BASED GENERATION

Genetic Programming (GP) was developed by Koza (1992) in an attempt to make self-evolving computer programs. GP is an area located within the evolutionary computation that inspires in the evolution of species and natural selection theories to handle searching and optimization tasks.. The individuals employed in GP are programs or solutions whose codification has a tree-shaped structure. These individuals are stored in a population which is evolved by applying a set of genetic operators (selection, crossover, mutation and replacement) to it. The evolutionary process is performed until a satisfactory solution is reached. Each individual is evaluated through a fitness evaluation function. It is important for two individuals representing similar solutions to have similar codifications. This principle avoids a strictly random searching process. In addition to this, if the codification method and the selected crossover and mutation operators are carefully chosen, the searching process will preserve the best individuals from each generation as well it will tend to improve them. Figure 3 shows the topography of a search space from which an individual has been extracted. This individual codifies a derivation tree which represents the mathematical expression 3+4=7.

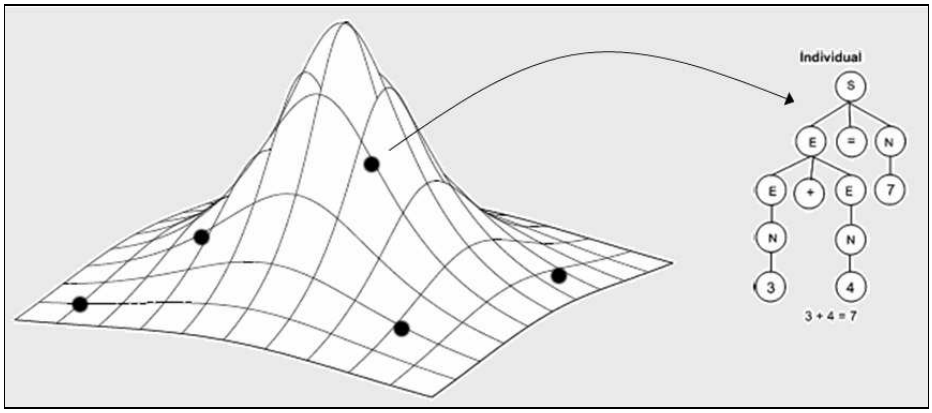

Figure 3: GP search space example. Each point represents a GP individual.

Grammar-Guided Genetic Programming (GGGP) is an extension of traditional GP systems [12], conceived to always generate valid individuals (points or possible solutions that belong to the search space). To do so, GGGP employs a context-free grammar (CFG) so that every individual in the population is a derivation tree that generates a sentence (solution) belonging to the language defined by

the CFG [13]. A context-free grammar G comprises a 4-tuple G = (Σ_N, Σ_T, S, P) such as $\Sigma_N \cap \Sigma_T = \emptyset$; where Σ_N is the alphabet of non-terminal symbols, Σ_T is the alphabet of terminal symbols, S is the axiom of the grammar and P is the set of production rules, written in BNF (Backus-Naur Form).

The method for initializing the population is very important in the convergence speed of GGGP [14]. The tree-generation algorithm that has been chosen in this study to initialize the GGGP system is called GBIM: grammar-based initialization method. GBIM is able to randomly generate valid individuals, not larger than a predefined size. The grammar-based crossover operator (GBC) is a general-purpose operator to solve problems with GGGP [10]. This operator has been chosen because it has three important and valuable features: it prevents code bloat, provides an adequate trade-off between search space exploration and exploitation capabilities and is able to improve convergence speed by taking advantage of the main feature of ambiguous grammars consisting on the existence of more than one derivation tree for a single sentence. All these characteristics together provide GBC with a high convergence speed and less probability of getting trapped in local optima, what leads to better solutions. GBC takes two parents (derivation trees) to produce, as a result, two new valid individuals (also derivation trees). Finally, the GGGP system employed in this study includes the so-called grammar-based mutation (GBM) operator [9], which works in a similar way to GBC. Given an individual (derivation tree) to be mutated, and having randomly chosen the mutation node, the operator substitutes the sub-tree whose root is the mutation node for any other that yields a valid derivation tree as a result.

5. RESULTS

In order to prove the evolutionary model used to automatically generate the fuzzy knowledge bases, a data set collected from 12 ATPRD units has been employed. These data cover eleven months (from January to October 2008), obtaining a total of 9.100 train bogies journeys. Each train bogie journey has associated 65 fuzzy derived parameter values.

Mainly, the grammar employed by the GGGP system includes both, fuzzy rules syntax definition and fuzzy variables and linguistic values declaration. Terminal and non-terminal symbols must also be declared. Table 1 shows the grammar associated to the monitored system fuzzy rules definition. Fitness function employed favors correct system diagnoses and penalizes the obtaining of complex knowledge bases. Fuzzy knowledge bases complexity depends on the number of terminal symbols involved. The next linear expression, referred to individual *i*, model these criteria:

$$F_i = k_1 TP + k_2 TN + k_3 FP + k_4 FN + k_5 NTS$$

where k_j are constants and TP, TN, FP, FN are the number of true positives, true negatives, false positives and false negatives respectively. NTS is the total number of non-terminal symbols. The best constant values combination obtained in the tests carried out was k_1=5, k_2=0.001,k_3=-0.01, k_4=-0.02 and k_5=-0.01.

Train bogies journeys where a problem was identified (typically sensor failures) were selected from the 9100 total train bogies journeys, in order to guide the GGGP algorithm to find the fuzzy rules set able to successfully diagnose these problems. Figure 4 shows one condition monitoring sensor failure example: The graph illustrates one sensors group temperature signals related to train speed (bottom signal). The irregular signal (upper signal) that is uncorrelated with the other ones is associated to a malfunction of the sensor that represents.

Table 1: Monitored System Fuzzy Logic grammar

P = { *S* ::= RULES_BLOCK ;
RULES_BLOCK ::= RULE | RULE - RULE_BLOCK ;
RULE ::= IF ANTECEDENT THEN CONSECUENT ;
ANTECEDENT ::= CONDITION | NOT CONDITION |
ANTECEDENT OR ANTECEDENT |
ANTECEDENT AND ANTECEDENT ;
CONSECUENT ::= DIAGNOSIS IS ACTIVATED
CONDITION ::=
CORRELATION1 IS CORRELATION_VALUES |
CORRELATION2 IS CORRELATION_VALUES |
...
VOLATILITY1 IS VOLATILITY_VALUES |
VOLATILITY2 IS VOLATILITY_VALUES |
...
MEAN_GROUP1 IS MEAN_GROUP_VALUES |
MEAN_GROUP2 IS MEAN_GROUP_VALUES |
...
M_SQUARE_GROUP1 IS MEAN_SQUARE_VALUES |
M_SQUARE_GROUP2 IS MEAN_SQUARE_VALUES |
...
MAX_ABS_GROUP1 IS MAX_ABS_VALUES |
MAX_ABS_GROUP2 IS MAX_ABS_VALUES |
...
P10_GROUP1 IS P10_GROUP_VALUES |
P10_GROUP2 IS P10_GROUP_VALUES |
...
P20_GROUP1 IS P20_GROUP_VALUES |
P20_GROUP2 IS P20_GROUP_VALUES |
...
P80_GROUP1 IS P80_GROUP_VALUES |
P80_GROUP2 IS P80_GROUP_VALUES |
...
P90_GROUP1 IS P90_GROUP_VALUES |
P90_GROUP2 IS P90_GROUP_VALUES | ... ;

DIAGNOSIS ::=
ANOMALOUS_EXTERNAL_CONDITION |
SENSOR_FAILURE | MECHANIC_FAILURE ; }

Σ_N = { S, RULES_BLOCK, RULE, ANTECEDENT, CONSECUENT, CONDITION, CORRELATION_VALUES, VOLATILITY_VALUES, MEAN_GROUP_VALUES, MEAN_SQUARE_VALUES, MAX_ABS_VALUES, P10_GROUP_VALUES, P80_GROUP_VALUES, P90_GROUP_VALUES, DIAGNOSIS }

Σ_T = { -, IF, THEN, NOT, AND, OR, IS, ACTIVATED, CORRELATION1, CORRELATION2, ..., VOLATILITY1, VOLATILITY2,..., MEAN_GROUP1, MEAN_GROUP2, ..., M_SQUARE_GROUP1, M_SQUARE_GROUP2, ..., MAX_ABS_GROUP1, MAX_ABS_GROUP2, ..., P10_GROUP1, P10_GROUP2, ..., P20_GROUP1, P20_GROUP2, ..., P80_GROUP1, P80_GROUP2,..., P90_GROUP1, P90_GROUP2, ..., VERY_LOW, LOW, MEDIUM, HIGH, VERY_HIGH }

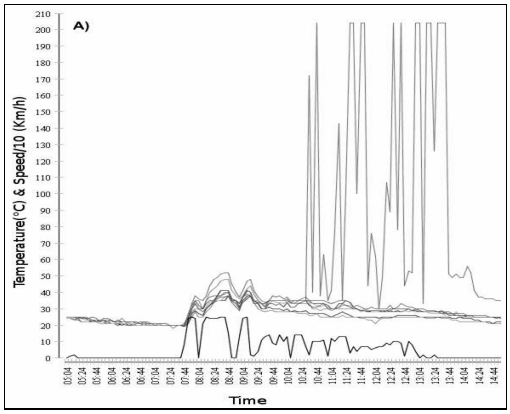

Figure 4: Condition monitoring sensor failure example.

The next fuzzy rules set is an example of solution found by the GGGP algorithm by using GBIM, GBC and GBM operators. This group of fuzzy rules detects the selected problematic journeys with no false positives and negatives.

IF CORRELATION1 IS low AND VOLATILITY1 IS NOT low THEN SENSOR_FAILURE IS activated ;
IF CORRELATION3 IS low AND VOLATILITY3 IS NOT low THEN SENSOR_FAILURE IS activated ;
IF VOLATILIY4 IS very_high AND MAX_ABS_GROUP4 IS very_high THEN SENSOR_FAILURE IS activated ;
IF MEAN_GROUP4 IS very_high THEN ANOMALOUS_EXTERNAL_CONDITION IS activated ;

As expected, it can be seen how situations where pair sensors correlations are low or anomalous high temperature peaks are detected are associated with sensor failures. A fuzzy rule modelling anomalous external condition can also be observed. It is based on detecting a high sensor group temperature mean.

6. CONCLUSIONES

Fuzzy Logic is a very suitable instrument to model knowledge concerning condition monitoring and diagnosis systems, since it allows knowledge to be modelled in an intuitive way. In those cases where knowledge cannot be directly modelled, due to either, non-existing previous expert knowledge or overly complex knowledge base management, evolutionary automatic fuzzy knowledge bases tools are highly recommended. In this work GGGP genetic programming algorithm has been employed to accomplish this task. Tests show that the presented model is valid when applying to a real world problem such as monitoring and intelligent diagnosing of the railway domain, in which by manually identifying interesting cases, the system is able to automatically generate rules that model such situations. This approach could be applied to other industrial sectors as search assistant tool to discover knowledge to model failures and interesting situations.

Acknowledgments. Present work has been financed by Fatronik-Tecnalia and by the Spanish Ministry of Science and Education under project no. DEP2005-00232-C03-03. The data used for the study has been provided by NEM Solutions and CAF.

REFERENCES

[1] Gonzalez, A.J. and Dankell D.D.: Engineering of knowledge-based systems. Prentice Hall (1993)
[2] Brachmand, R.J. and Levesque, H.J.: Knowledge Representation and Reasoning, MIT Press, Cambridge, MA (2003)
[3] Preece, A.D.: Validation of Knowledge-Based Systems: The State-of-the-Art in North America. J. Study of Artificial Intelligence Cognitive Science and Applied Epistomology. 11(4) (1994)
[4] Leiviskä,K.: Industrial Applications of Soft Computing. 2001.
[5] Babuska, R. and Verbruggen, H.B.: "A new identification method for linguistic fuzzy models."Proceedings FUZZ-IEEE/IFES'95, 1995, Yokohama, Japan, p. 905-912.
[6] Valente deOliveira, J. "Neuron inspired rules for fuzzy relational structures." Fuzzy Sets and Systems, 1993 57(1), p. 41-55.
[7] Couchet, J.; Font, J.M.; Manrique, D. Using Evolved Fuzzy Neural Networks for Injury Detection from Isokinetic Curves. International Conference on Artificial Intelligence, Dec. 9-11, 2008, Cambridge, UK.
[8] R. Koza, Genetically breeding populations of computer programs to solve problems in artificial intelligence, Tech. Rep. CS-TR-90-1314, Department of Computer Science, Standford University, (1990).
[9] W.B. Langdon, R. Poli, Foundations of Genetic Programming, Springer, Verlag, London, UK, 2001.
[10] Couchet, J., Manrique, D., Rios, J. and Rodríguez-Patón, A.: Crossover and mutation operators for grammar-guided genetic programming. Soft Comput.2007, 11(10): p. 943-955
[11] Construcciones y Auxiliar de Ferrocarriles, http://www.caf.net/caste/home/index.php
[12] Koza, J.R., Keane, M.A., Streeter, M.J. et al., Genetic Programming IV: Routine Human-Competitive Machine Intelligence, Kluwer Academic Publishers., Norwell, MA, 2005.
[13] Whigham, P.A., Grammatically-based genetic programming, in: J.P. Rosca, (Ed.), Proceedings of the Workshop on Genetic Programming: From Theory to Real-World Applications (Tahoe City, California, USA, 1995), p. 33–41
[14] Hao, H.T., Hoai, N.X., cKay, R.B.: Does this matter where to start in grammar guided genetic programming?, in: Proceedings of the second Pacific Asian Workshop in Genetic Programming (Cairns, Australia, 2004, Electronic).

The author has requested enhancement of the downloaded file. All in-text references underlined in blue are linked to publications on ResearchGate.