Fuzzy Model Optimization Using Genetic Algorithm for Aircraft Engine Diagnosis

LiJie Yu, Daniel J. Cleary, Mark D. Osborn, Vrinda Rajiv

General Electric Global Research Center,
Niskayuna, NY 13024
yuli@crd.ge.com

Abstract

An accurate and up-to-date diagnostic model is critical for economic aircraft engine operation. However, for many commercial airline fleets, monitoring and diagnosing engine faults is often left to human operators due to lack of effective modeling. Individuals must manually interpret engine performance parameters and this results in inconsistent evaluation and the potential for error. A recent work has been performed to capture the knowledge of diagnostic engineers and apply fuzzy logic to engine fault classification. The developed fuzzy model is capable of capturing various engine fault signatures based on expert experience, and it provides accurate diagnosis by assessing the similarity between those fault signatures and observed trends in sensor data. The major drawback with this approach is the manual knowledge extraction and model maintenance process. Manual model tuning is not only labor intensive, but it also brings another source of inconsistency to the overall system performance. In this paper, we will present a hybrid approach to augment the existing expert knowledge-based diagnostic model with automatic learning capability using a genetic algorithm. The proposed approach not only allows for automatic model tuning, but also enables the diagnostic model to adapt throughout the engine service life as operating conditions change.

Introduction

Background

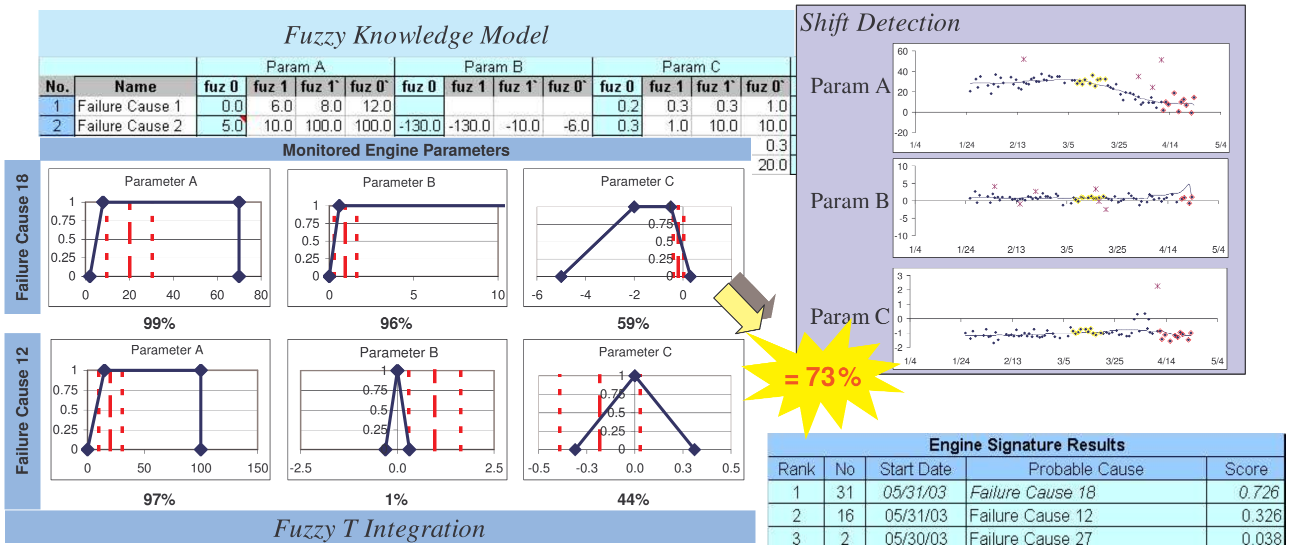
To ensure economical operation of aircraft engines, remote monitoring of these engines is a standard practice in the aviation industry. Sensor data are collected during various flight regimes, and transmitted to a ground-based system for anomaly detection and fault diagnostics. Here diagnostics refers to the process of classifying an engine fault into a set of non-overlapping and actionable categories such that it provides guidance for fleet operation decisions and maintenance work scope. Accurate and timely diagnosis is the key to quickly resolving a failure condition and preventing costly secondary damage and the possibility of an in-flight shutdown.

A previously developed approach (Yu, Cleary, and Cuddihy 2004) has been demonstrated that leverages engineering knowledge and fuzzy computing to achieve accurate engine failure root cause analysis. It extracts multi-parameter shifts from flight snapshot data, and maps these shifts against a fuzzy rule table that captures the engineering knowledge of failure signature patterns. The best-matched signature is chosen as the failure root cause.

Figure 1 shows the individual modules in this method and also provides an example of how each would function:

1. Shift Detection - removes outliers, detects and measures parameter shifts, and correlates multi-parameter shifts to decide on an event start date. The example shown in the upper right corner of Figure 1. Two data sets are selected (highlighted data points), before and after an event, respectively.
2. Fuzzy Knowledge Model - fuzzy representation of engineering expertise of failure root causes and symptoms. Two examples of root-cause signatures are shown in the upper left part of the figure, with three parameters defined in each rule. For each parameter, four numbers in the rule table specify the trapezoidal membership function for the parameter shift.
3. Fuzzy T Integration - assesses the likelihood of match between a particular engine sensor shift and an anomaly pattern for each parameter, and aggregates each likelihood estimate across multiple parameters to obtain a matching score for each fuzzy rule. This is illustrated in the lower left corner of Figure 1. Based on the confidence interval of a parameter shift, a matching parameter score is calculated for each fuzzy membership function, and then the parameter scores are aggregated to provide a score for each fuzzy rule.
4. Result - scores of each fuzzy rule are sorted in descending order, and the root cause with the highest score, 0.726 of failure cause 18 in the example, provides the most likely diagnosis and consequent recommendations for engine maintenance forces.

The Fuzzy Knowledge Model is created off-line by leveraging engineering knowledge and historical diagnostic experience. For on-line operation, engine flight data is downloaded and processed for trend analysis and shift detection, and the diagnostic result is obtained following the module procedure described above.



		Param A				Param B				Param C			
No.	Name	fuz 0	fuz 1	fuz 1`	fuz 0`	fuz 0	fuz 1	fuz 1`	fuz 0`	fuz 0	fuz 1	fuz 1`	fuz 0`
1	Failure Cause 1	0.0	6.0	8.0	12.0					0.2	0.3	0.3	1.0
2	Failure Cause 2	5.0	10.0	100.0	100.0	-130.0	-130.0	-10.0	-6.0	0.3	1.0	10.0	10.0
													0.3
													20.0

Engine Signature Results

Rank	No	Start Date	Probable Cause	Score
1	31	*05/31/03*	*Failure Cause 18*	*0.726*
2	16	05/31/03	Failure Cause 12	0.326
3	2	05/30/03	Failure Cause 27	0.038

Figure 1 Aircraft Engine Diagnostic: Root Cause Analyzer

One of the merits of this approach is that it separates the uncertainty introduced by sensor noise from the uncertainty of engineering knowledge, enabling clearer modeling of both. The variance introduced by sensor noise is handled with statistical processing and confidence testing, whereas the knowledge variance is captured using the fuzzy rule engine.

Motivation of Work

In this paper, we will present a machine learning method using genetic algorithms to optimize the fuzzy rule model mentioned above for aircraft engine diagnostics. Both the diagnostic procedures described in the above section and the learning method to be disclosed can be viewed as an integrated process to ensure accurate and up-to-date interpretation of engine health conditions. The reason that the fuzzy rule model is chosen as the optimizing target reflects some of the deficiencies of the existing approach. The diagnostic model explicitly captures the engineering knowledge about the known failure modes and associated failure signatures in a fuzzy rule table, which makes the rule well understood and easily adjustable. However, the major drawback of this approach is the semi-manual knowledge capturing and tuning process, in which diagnostic engineers provide rule definitions, which are then tested and manually adjusted to achieve better accuracy on test cases. This process is both tedious and labor intensive. For a small sized model, comprising 30 monitored parameters, 50 failure signatures, and using a trapezoid fuzzy membership function, which has four independent configuration variables, the total number of tunable continuous variables would be the product of all these factors, or 6000. This is well beyond a person's capability to tune the model, and it would be more so for larger and more complex models. More importantly, manual tuning is error prone, introduces unacceptable variance and is difficult to optimize.

Another major issue is model adaptation over a long period of time. An aircraft engine normally deteriorates at a certain rate that is reflected in parameter pattern changes, with a subsequent effect on the diagnostic model's sensitivity and accuracy. Given a fleet of similar engines to monitor it would make sense to adjust the diagnostic model configuration periodically to make sure its performance is within control limits. It is a desired feature that the fuzzy model would automatically and quickly adapt to the changing fleet characteristics to prevent diagnostic model obsolescence.

Given these concerns, it became necessary to leverage machine-learning techniques to automatically tune and adapt the fuzzy model configuration to optimize the overall system performance.

In the following sections, we will first discuss various learning approaches considered and the selection tradeoffs among them for the aircraft engine diagnostic application. Then our process and methodology details will be presented, followed by result discussion. Finally, conclusion and related future research direction will be discussed.

Learning Architectures

As artificial intelligence (AI) technology evolves to greater maturity in recent years, more interest has been focused on research and applications that leverage AI techniques to solve industrial problems (Cordon, et al., 1995; Karr and Freeman, 2001). Machine learning in particular, has been widely applied to system optimization, process control, and general classification problems. The idea is to rely on data driven computing methods, such as an artificial neural network (ANN), clustering algorithm or genetic algorithm (GA) to derive the best mapping from the data space to the feature space and then to the problem solution space.

To select a suitable architecture for the aircraft engine diagnostic model at hand, the first question is whether it shall be an entirely data driven model or more of a hybrid model with physics-based rules that are tuned by data. Approaches falling into the former category, to name a few, include ANN, multi-variable regression, and clustering methods. Exclusive data driven models are more straightforward to implement with minimum restrictions imposed by the existing model configurations. On the other hand, the capability of these models is limited by available data, and the end result is a black-box model that lacks visible knowledge representation.

For our purposes, we proposed using a hybrid method that combines the structure of the original diagnostic fuzzy model with data driven genetic algorithm parameter tuning. In this way, we were able to maintain a visible engineering-knowledge model, while leveraging AI techniques to improve model accuracy and isolation capability. This approach is superior when data is missing or of insufficient quantity to derive an solution with acceptable confidence.

Genetic Algorithm (GA) is a non-calculus based search algorithm, which utilizes natural selection and evolutionary theorems to select the best performer among multi-generation potential solutions (Goldberg, 1989). GA as a learning technique has been successfully used in fuzzy model tuning (Bonissone et al., 1995; Cordon et al., 2001; Ruan, 1997) for various industrial applications. It allows fast convergence to near optimum state in a vast search space with a small set of simple operators, namely mutation, crossover, and reproduction. Other ANN-based fuzzy rule tuners may also be promising, such as ANFIS (Jang, 1993) and NEFCLASS (Nauck and Kruse, 1997), even though they are out of the scope of the current study.

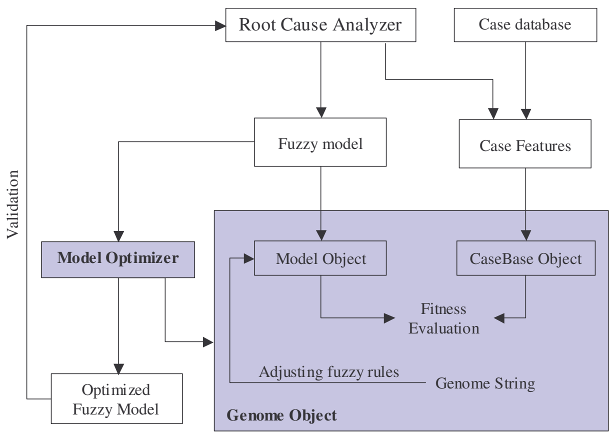


Figure 2 Fuzzy Model Tuning Process Using GA

Methodology

The approach we choose is based on the existing fuzzy diagnostic framework presented in an earlier work (Yu, Cleary and Cuddihy, 2004), and it applies GA to optimize the fuzzy diagnostic model to achieve better performance and adaptation capability. The model tuning and evaluation process is shown in Figure 2. The shaded areas in the chart indicate components directly used in the GA tuning process, whereas the non-shaded components are either within the existing diagnostic process or are input or output of the GA optimizer. The function of each component is described in the following:

Root Cause Analyzer

The Root Cause Analyzer (RCA) is the root cause diagnostic process to be optimized. As described in the introductory section, it includes a data analysis module for data feature extraction, a fuzzy knowledge module which captures engineering knowledge of failure signatures, and a fuzzy inference module that evaluates likelihood of a failure based on the similarity of data feature and fuzzy rules. To leverage the GA model tuning process, the fuzzy inference module is exported from RCA and loaded into the GA object for genome fitness evaluation.

Case Database

The optimization goal is to maximize accuracy for the diagnostic model. To assess the model accuracy, a case database is set up containing a set of historical cases. Each case record captures the data features of dimension n, and an associated event for the case at hand. Both positive and negative events are captured in the case base and they are used for model tuning and validation. Positive events refer to engine cases with known engine anomalies of specified types, whereas, negative events refer to engine cases that are “eventless” or engines that have not experienced any known anomalies. Positive events are used to derive failure signatures, whereas the negative ones are used to represent behaviors of normal engines thus helping to adjust fault detection sensitivities.

Data features of each case are extracted by the RCA based on a given engine serial number and an analysis date range (Yu, Cleary and Cuddihy, 2004). They provide a snapshot of the engine working condition under the assigned event, providing relevant information or signatures for fault detection and isolation. The characteristics of these data features may be quite diverse. For example, they may include continuous sensor data trend shifts, sensor slow drifting rates, indicators of maintenance actions or engine cycles. All features, including categorical information, are coded as real numbers.

As is the case in many data driven approaches, data quality is critical in capturing the essence of the knowledge. Extra effort is required to make sure only high quality cases are selected in order to reduce variation caused by inconsistent or misinterpreted data features and events. Another issue is the quantity of information. When there is insufficient information to represent the characteristic of a certain event, poor performance is expected. In such a situation, the advantage of a knowledge-guided hybrid model becomes most obvious since physical system knowledge could be used to close the gaps of missing or insufficient data.

Genome Object

This is the central component in the GA tuning process. The Genome object contains an internal representation of the fuzzy model and the case database, respectively. The Model object loads the fuzzy model exported by RCA, adjusting the fuzzy rule parameters based on the coded genome string. It then evaluates the fitness of the Genome object based on model evaluation performed on the cases loaded into the CaseBase object.

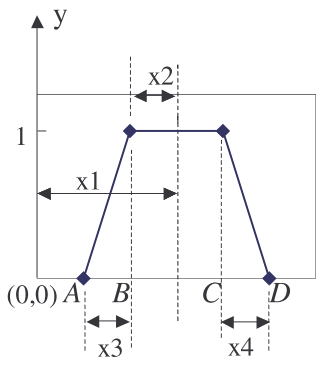


Figure 3 Rule Parameter Conversions

Here the fuzzy rule parameters to be adjusted are limited to the membership function, which is a trapezoid function for each parameter in each rule. Therefore, four real numbers are needed for each parameter, as shown in Figure 3. In this figure, the horizontal axis indicates the parameter value, and the vertical axis indicates the membership degree value that is in the range of [0,1]. The original definition of the membership function specifies two outer corner parameter values, *A* and *D*, where the membership degree is zero, and two inner corner values, *B* and *C*, where the membership degree is one. But there are dependencies among the four values, i.e.

$$A <= B <= C <= D \tag{1}$$

which makes these values unsuitable for GA coding. This is because the primitive GA operators, namely mutation, cross over, and reproduction, assume independency among each gene component. Thus, four independent variables are derived to address this issue,

$$\begin{aligned} x_1 &= (B+C)/2 \\ x_2 &= (C-B)/2 \\ x_3 &= B-A \\ x_4 &= D-C \end{aligned} \tag{2}$$

where x_1 is the mean of the two inner corners, x_2 is half span of the two inner corners, and x_3 and x_4 are the left and right flange horizontal length, respectively. These four variables are used in the genome string coding for each parameter membership functions.

As mentioned earlier, even for a relatively small model, the search space is extremely large due to the amount of tunable parameters as well as the continuous nature of the possible solution. To speed up convergence and reduce cycle time, a divide and conquer technique is used where a local optimization is performed for each failure mode one at a time, and the final optimized model is a concatenation of the best performers. Therefore, the coding string assigned is an array of real numbers representing the membership functions for a single failure mode. There may be multiple signatures or rules for a single failure mode, thus the length of the coding string, *L*, may vary, which is

$$L = \sum_{i=1}^{r} 4 * p_i \tag{3}$$

where P_i is the number of parameters in rule *i*, and *r* is the total number of rules for a failure mode, or event type.

Model Optimizer

The model optimizer configures, initiates, and controls the GA tuning process. First, a set of constraints is set up for the feasible range of each tunable fuzzy variable. For example, referring to Figure 3, the minimum values for x_2, x_3, and x_4 are zeros for each parameter. For the upper range of these variables and ranges for x_1, some heuristic constraints are used, such as, allowing a maximum of 50 percent variation from the original engineering rule guidelines.

Secondly, this module specifies GA parameters that control the population and reproduction process. An elitist strategy is applied that maintains the best performer at each generation. The number of generation is initially chosen as 100, which is adjustable when quick convergence is required. To prevent early convergence at local maxima and maintain candidate diversity, a relatively large population size, 300, is used for each generation. The reproduction control parameters, i.e., replacement rate, crossover rate, and mutation rate, are set at 0.3, 0.8, and 0.05, respectively.

The output of the optimization process is a fuzzy model with adjusted rule parameters. Since the internal model inference engine contained in the Genome object is only an approximation of the original RCA model, to validate the produced model, the model optimizer will load the tuned model back to RCA for an external evaluation.

Result

An initial set of tests has been performed on a GE CMF56-3 engine model. Seventy field-verified historical cases were selected from the case database for model optimization and evaluation. These cases represent nine different event types, which include one or more engine gas path, indication system, lubrication system, and vibration system failure modes. Only six event types, which have at least three representative cases, are used in the GA tuning. The RCA data analysis module extracted twenty or so features for these cases that are used by the fuzzy model for event isolation.

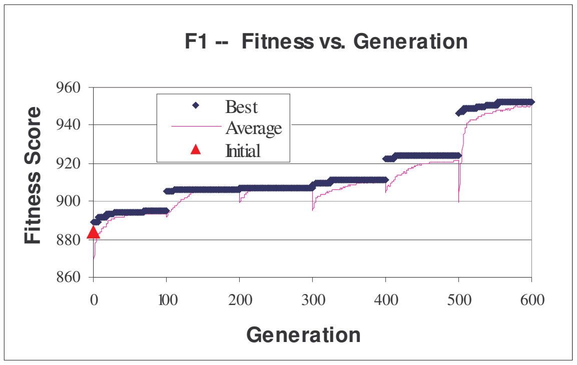
The tuning result is directly influenced by the optimization goals. There are two goals relevant to our work. One is to maximize the likelihood score of the target event, and the other goal is to minimize the score of the non-target, or “wrong” event, therefore maximizing the isolation margin. For comparison purposes, three different fitness functions are tested which are listed in Table 1. The first one relates to model accuracy assessment, where all diagnosis events are scored, then sorted by descending order, and the top ranked event is chosen as the failure diagnosis. The second one is designed to aggressively maximize the probability score of the targeted event, whereas the third one is to maximize the difference between the targeted and the highest wrong score.

For ease of comparison, 100 generations are run for each of the six failure modes, and the accumulative fitness scores are shown in Figure 4 for each of the three fitness function types. In each of the sub-figures, the dark symbol line is the best fitness obtained in a generation, the light solid line shows the average fitness among all the individuals within a generation, and the red triangle shows fitness of the initial, or non-optimized model.

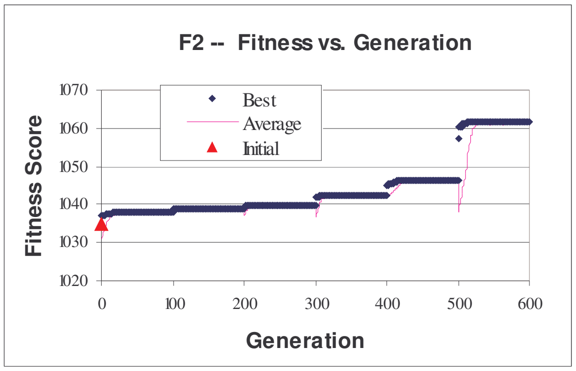
Optimization Goal	Type
Maximize ranking of the targeted event	F1
Maximize score of the targeted event	F2
Maximize score difference between the targeted and the first wrong event	F3

Table 1 Fitness Function Type Definition

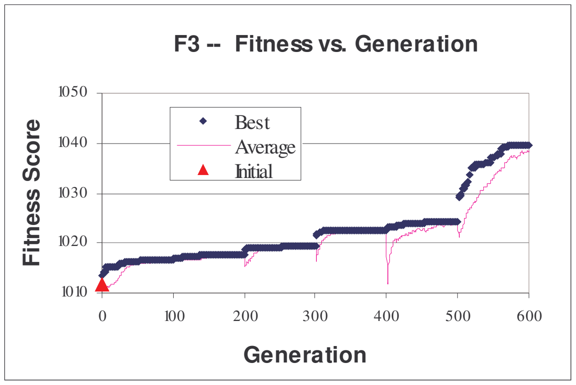
Due to unit difference among the different fitness types, absolute comparison of the fitness scores across the different objective functions is not meaningful. However, trend similarities among the three are very interesting. In Figure 4, all three tests achieve significant fitness improvement from the initial non-optimized model. The most improved failure mode is in the sixth one, whereas in some other failure modes negligible gain is made. It is also noted that the first two fitness types have somewhat quicker convergence than the third one. Discontinuity of trend is observed in most of the failure modes transition periods. That is, in the first generation of a new set of rules and constraints, the average fitness of the population drops, whereas the accumulative best fitness jumps higher. When the reproduction and evolution proceeds, the average fitness start to pick up and eventually converge to the best one.



(a)



(b)



(c)

Figure 4 Comparisons of fitness Scores

	Non-optimized	F1	F2	F3
Accuracy Rate (%)	82.9	85.7	88.6	87.1
$\sum_i p_t^i / n$	0.422	0.489	0.868	0.558
$\sum_i p_r^i / n$	0.273	0.247	0.729	0.314
$\sum_i p_t^i / n - \sum_i p_r^i / n$	0.149	0.242	0.139	0.244

Table 2 Comparison of results of non-optimized model with models tuned with different fitness functions.

All three optimized models are loaded back into RCA for testing. Several assessment metrics are set up as shown in the first column of Table 2. From the top row to bottom down, the metrics are the average accuracy rate, average likelihood score of the targeted failure mode, average score of the highest wrong diagnosis, and the difference between the last two. Table 2 shows the comparison of the testing result among the non-optimized and the three fitness designations. It is shown that F2 gives the highest accuracy rate, 88.6%, and the average probability score of the targeted failure mode is 0.868 given a 0 to 1 range. However, the isolation capability of F2 model is only 0.139, which is worse than the original model which is 0.149. F3 has slightly lower accuracy rate than F2, whereas the isolation capability is 0.244, which is significantly higher than both F2 and the original model. Given both accuracy and event isolation as the preferred model characteristics, the F3 might be the best optimization criteria of the three developed.

Conclusions and Future Work

In this paper, we have presented an integrated process that leverages GA learning in fuzzy diagnostic model tuning. The tuned model demonstrated performance gains in terms of both accuracy and event isolation capability. This automatic learning capability reduces the labor-intensive model maintenance work and enables model optimization and adaptation.

The diagnostic model structure is unaltered during the tuning process partly to reduce problem complexity, and partly to maintain the engineering knowledge. However by doing so, learning is limited to the existing rules, and this process is unable to derive new terms based on the experience data. This limitation would be addressed by future research. This will be particularly important for capturing signatures for unknown failure modes.

Another interesting effort would be to utilize cost functions in the optimization process. Currently, the accuracy and isolation metrics used in the objective functions assumes equal penalty for all classification defects. However, for a particular event, a false negative diagnosis may be much more expensive if it causes more serious consequences. Therefore, including tradeoff, or cost analysis, in the model fitness evaluation may help to fine tune the decision result.

References

Bonissone P.; Khedkar, P. S.; and Chen, Y. 1996. Genetic Algorithms for Automated Tuning of Fuzzy Controllers: a Transportation Application. In Proceedings of the Fifth IEEE International Conference on Fuzzy Systems, 674-680. New Orleans, LA.

Cordon, O.; Herrera, H.; and Lozano, M. 1995. A Classified Review on the Combination Fuzzy Logic-Genetic Algorithms Bibliography. Research Report DECSAI 95129, Department of Computer Science and AI, University de Granada, Granada, Spain.

Cordon, O.; Herrera, F.; Hoffmann, F.; and Magdalena, L. 2001. Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases. World Scientific.

Goldberg, D. E. 1989. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley.

Ruan, D. 1997. Intelligent Hybrid Systems: Fuzzy Logic, Neural Network. Kluwer Academic.

Jang, J.-S. R. 1993. ANFIS: Adaptive-Network-Based Fuzzy Inference System. IEEE Transactions on Systems, Man, and Cybernetics, 23(3): 665-684.

Nauck, D. and Kruse, R. 1997. A Neuro-Fuzzy Method to Learn Fuzzy Classification Rules from Data. Fuzzy Sets and Systems 89: 377-388.

Karr, C. L.; and Freeman, L. M. 2001. Industrial Applications of Genetic Algorithms. CRC Press.

Yu, L.; Cleary, D. J.; and Cuddihy, P. E. 2004. A Novel Approach to Aircraft Engine Anomaly Detection and Diagnostics. In Proceedings of the IEEE Aerospace Conference, 3468-3475. Big Sky, Montana.