

# Evolving Fuzzy Rules for Pattern Classification

Hugh Mallinson and Peter Bentley

*Department of Computer Science, University College London, Gower Street,  
London WC1E 6BT, UK. P.Bentley@cs.ucl.ac.uk, H.Mallinson@cs.ucl.ac.uk*

## **Abstract**

This paper describes the use of a Hybrid Fuzzy-Genetic Programming system to discover patterns in large databases. It does this by evolving a series of variable-length fuzzy rules which generalise from a training set of labelled classes. Numerous novel techniques, including the use of genotypes in Genetic Programming, two new genetic crossover operators, and the processes of Modal Evolution, Modal Re-evolution and Nested Evolutionary Search are described. Experimental results show that the system is able to classify data from the Wisconsin Breast Cancer database correctly 95% of the time.

## **1. Introduction**

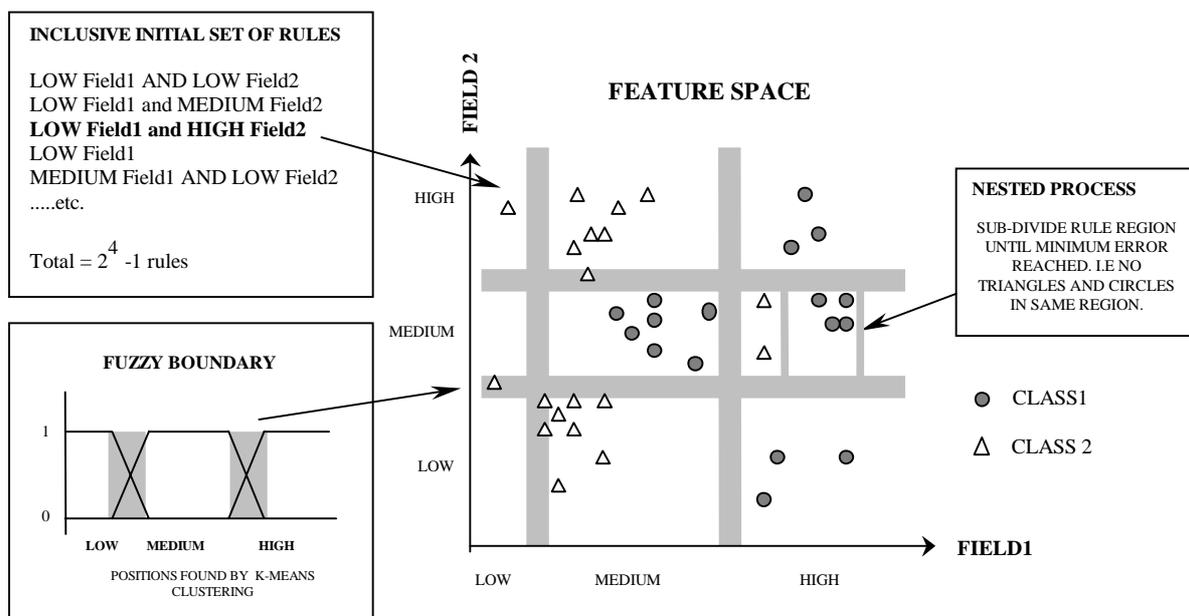
Pattern classification systems represent an attempt, given a subset of labelled data of identify underlying structures in the whole database. This is achieved by isolating important fields and choosing appropriate groupings that will carve the problem feature space into homogenous regions. The compromise sought is between sufficient coarseness such that some shared trait is found (and the search process is manageable), and a sensitivity to the non-linearities inherent to discrete and continuous data with ranges that offer no easy means of normalisation. Moreover, it is characteristic of these problems that phenomena to be characterised occur very rarely in the population.

This paper describes the use of an evolutionary-fuzzy system to discover patterns in *large* databases. Genetic Programming is used to evolve a series of variable-length fuzzy rules which characterise the differences between classes of data held in a database. The system is being developed with the specific aim of insurance-fraud detection – which involves the challenging task of classifying data into the categories: "safe" and "suspicious".

## **2. Background**

### ***2.1 Fuzzy Rules***

Fuzzy rules offer an appealing method for individuating classes of phenomena described in data sets (Eberhart et al., 1996). They are easy to understand, verify and extend. This makes them attractive for use in domains where experts exist who can seed the systems with a number of effective rules from the outset. Results on standard data sets have been comparable with Neural Net approaches (Chiu, 1997). Typically, clustering is carried out in the N-dimensional space defined by all records, using Kohonen unsupervised learning algorithms and back propagation methods (Chung & Lee, 1985). Initial rules are then associated with each cluster centre (Ross, 1995).



**Figure 1.** Subdividing the problem space using fuzzy cells

Other approaches include the use of GP to combine and refine class partitions (Marmelstein & Lamont, 1998). This avoids problems caused by discrete or small-ranged values that cannot be easily normalised. Alternatively, Abe and Lan (1993) present an algorithm in which rules are extracted from nested fuzzy cells. The process terminates when the ‘fuzzy grid’ contains only homogenous cells.

We offer fig.1 as a summary of the problem addressed by these systems (after Ishibuchi et al. 1998). It illustrates a two-class classification problem. The gridded box in the centre shows the two-dimensional feature space defined by field1 and field2. The grey lines subdivide the feature space into fuzzy cells, where each cell corresponds to a fuzzy rule, shown in the top left box of figure 1. A point falling on a grey line (fuzzy boundary) has partial membership in the neighbouring cells. The position and size of each fuzzy boundary is defined by the clustering algorithm and fuzzy membership functions, as shown in the bottom left box in fig. 1. The nested process mentioned above is illustrated by the further subdivisions of (IS\_HIGH Field1 AND IS\_MEDIUM Field2). This corresponds to a notion of ‘fine-tuning’. The work described in this paper also uses such a ‘nested’ algorithm, revolving rules on shrinking regions of the problem space in order to minimise the misclassification of vectors.

## 2.2 Genetic Search

Genetic search can be applied to fuzzy pattern recognition to optimise the position and shape of membership functions (González et al.,1994) or as an aid at the clustering stage (Zimmerman, 1994). In addition, the rule searching process can itself be accomplished by different uses of a GA. One can either evolve populations of rule sets (the *Pittsburgh approach*, (Smith, 1980)) or just individual rules (the *Michigan approach* (Booker et al., 1989)). There follows a brief comparison of these two techniques.

### *The Pittsburgh Approach*

This method involves the evolution of rule *sets* (Smith, 1980). Ishibuchi et al. (1998) describe one recent implementation: an inclusive set of rules is generated (see top left of fig.1), and the GA is used to find an optimal subset of these rules. Each individual is a binary

string of length equal to the size of the inclusive rule set. A one in the  $n^{\text{th}}$  place on the chromosome means that the  $n^{\text{th}}$  rule is present in that rule set, a zero that it is absent.

An advantage of this approach is that one can produce a non-dominated set of solutions; i.e., the best classification for a rule set of size  $n$ ,  $n+1$ ,  $n+2$ ... and so on. The two-objective GA maximises the number of training examples correctly classified and minimises the size of the rule set. Once a rule set that is comprehensive has been created, this approach is appealing. However with large feature spaces, the creation of a reasonable rule set is the problem. For example, in a ten field problem with five membership functions,  $6 * 10^7$  rules would be necessary. Clearly, this would be unmanageable.

### *The Michigan Approach*

Systems following the Michigan approach (also known as evolutionary classifiers), take a single rule as an evolutionary individual (Booker et al., 1989; Valenzuela-Rendon, 1991). The work described in this paper takes this approach. It is suitable for larger problems which suffer from the 'combinatorial explosion' (as the number of fields/dimensions increases the search space of rules grows exponentially).

In contrast to the Pittsburgh method, the Michigan approach uses the entire population to model the rule base, where every individual in the population is a single rule. More complex credit-apportionment algorithms (such as bucket-brigade) are required to determine the fitness of each rule. To ensure diversity is maintained, *crowding* is commonly employed, where rules are updated one at a time in the population. Further details can be found in Goldberg (1989).

In this paper we describe an alternative approach which removes the need for such credit-apportionment and crowding. By allowing the evolutionary classifier to converge on a single rule, but performing multiple runs, a rule set is built up, one rule at a time. In addition, this work adds an extra evolutionary operator that allows new individuals to be created by appending one fit rule to another. Returning to figure 1, the rule (IS\_HIGH Field1) would have reasonable fitness (it contains a number of circles, the class we want to characterise), as would (HIGH Field2). Our algorithm could combine them and find the improved rule: (IS\_HIGH Field1 AND IS\_HIGH Field2). Information about the position and shape of the input member functions can also be contained on the chromosome so better division of the feature space can be evolved.

Naturally it is possible, once a number of rules have been evolved using the Michigan approach, to attempt to evolve the best compact subset of them using a Pittsburgh method. The remaining parts of the paper detail our implementation of a Michigan-style classifier system.

### **3. The Evolutionary-Fuzzy System**

The system comprises two main elements: a Genetic Programming (GP) search algorithm (Koza, 1992) and a fuzzy expert system (Belman et al., 1966). Figure 2 provides an overview of the system.

As shown in fig. 2, the system first clusters the data into three groups (one column at a time) using a K-means algorithm. The extreme points in each cluster are taken as the 'feet' of the membership functions: *LOW*, *MEDIUM*, and *HIGH* for each data column. Standard trapezoidal membership functions (see figure 1, bottom left) provide the fuzzifier with "degrees of membership" for every data item, resulting in a new database of fuzzified data. The GP engine is then seeded with random genotypes (coded rules) and evolution is initiated.

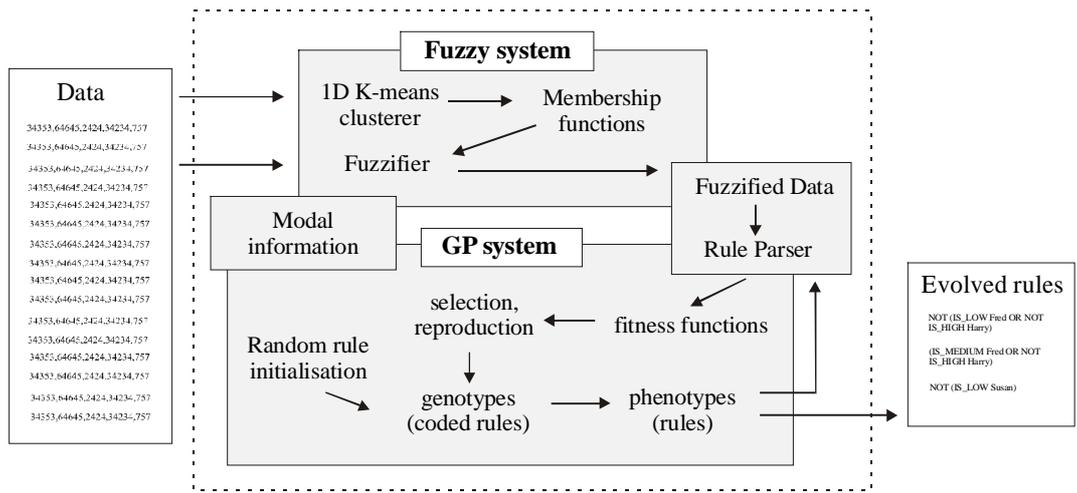


Figure 2 Block diagram of the Evolutionary-fuzzy system.

### 3.1 Genotypes and Phenotypes

Genotypes consist of variable sized trees, where each node consists of a binary number and a flag defining whether the node is binary, unary or a leaf, see figure 3.

Genotypes are mapped onto phenotypes to obtain fuzzy rules, e.g. the genotype shown in fig. 3 maps onto the phenotype:

"(IS\_MEDIUM (Height OR IS\_LOW Age) AND IS\_MEDIUM Age)".

The fuzzy grammar imposes certain constraints upon allowable solutions. For example, the arguments to 'IS\_LOW', 'IS\_MEDIUM' or 'IS\_HIGH' must always consist of fuzzy vectors:  $\langle Low_{membership}, Medium_{membership}, High_{membership} \rangle$ . The arguments to 'AND', 'OR' and 'NOT' functions must always be single-valued results obtained from the application of one or more of the functions.

As is clear from the example phenotype given above, evolved rules do not always satisfy the constraints imposed by fuzzy grammars. However, rather than impose these damaging constraints on evolution, such grammatically incorrect rules are corrected by the rule parser.

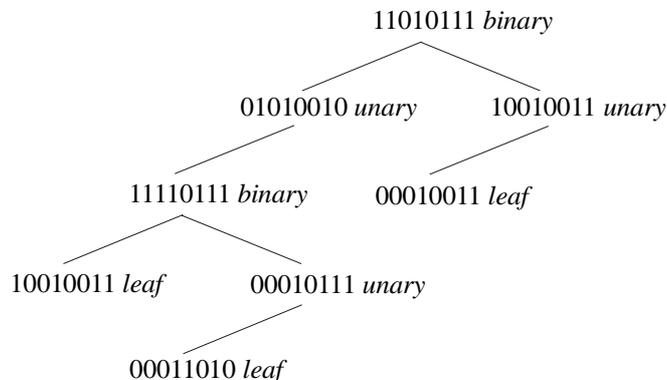


Figure 3 An example genotype used by the system.

Functions requiring a fuzzy vector, but receiving a single value do nothing. Functions requiring a single value, but receiving a fuzzy vector, apply 'IS\_HIGH' by default in order to generate the single value. Consequently, when interpreted by the fuzzy rule parser, the rule in the example equates to:

"((IS\_HIGH Height OR IS\_LOW Age) AND IS\_MEDIUM Age)".

### **3.2 Rule Evaluation**

Every evolved rule is evaluated by being applied to the fuzzified data, resulting in scores for every fuzzified data item. (The rule parser takes the minimum of conjuncts for "AND", the maximum for "OR"). This list of scores is then assessed by two fitness functions, based on:

- the degree to which the rule provides a high final score for at least one "suspicious" data item.
- the difference between the average of the best *nichesize* scores for "suspicious" items and the average of all of the scores for the "safe" data items.

To prevent the occurrence of bloat during evolution, a third fitness function penalises overlong genotypes.

### **3.3 Rule Generation**

Using these three fitness values for each rule, the GP system then employs the SWGR multiobjective optimisation technique (Bentley & Wakefield, 1997) to determine how many offspring each pair of rules should have. Child rules are generated using one of two forms of crossover. The first type of crossover emulates the single-point crossover of genetic algorithms by finding two random points in the parent genotypes that resemble each other, and splicing the genotypes at that point. By ensuring that the same type of nodes, in approximately the same places, are crossed over, and that the binary numbers within the nodes are also crossed, an effective exploration of the search space is provided without excessive disruption. The second type of crossover generates child rules by combining two parent rules together using an 'AND' operator. This corresponds to the insight that each rule can partially yet independently describe the same data class. For such cases, the conjunction provides a fuller, fitter description. Mutation is also occasionally applied, to modify randomly the binary numbers in each node by a single bit.

The GP system employs population overlapping, where the worst  $n\%$  of the population are replaced by the new offspring generated from the best  $m\%$ . Typically values of  $n = 80$  and  $m = 40$  seem to provide good results. The population size was normally between 100 and 200 individuals.

### **3.4 Modal Evolution**

Our real-world problem spaces are multi-modal. Each evolutionary run, usually no more than 30 generations, of the GP system evolves a rule describing one pattern (or mode). Characterised data (recorded in the modal database) are then removed from the problem space and the system restarts and evolves new rules for those remaining. The parameter *nichesize* specifies the number of "suspicious" data items sought to be classified in each run. This enables monitoring of over-fitting; the fitness is correlated with the number of class members classified by a rule as well as the number mis-classified.

### 3.5 Modal Re-evolution

In addition to the process of Modal Evolution, described above, the system re-examines each mode already classified by a rule; it attempts to improve the rule by ignoring all data except that characterised (and mis-characterised) by the rule already. This provides a shrinking environment, with the associated gains: a 'purer' gene pool of solutions for each archetype is facilitated, and an accelerated search.

### 3.6 Nested Evolutionary Search

After shrinking the environment (reducing the number of claims against which a rule is tested) the system can recluster and carry out a finer search. The process can be repeated until all subspaces are homogenous, i.e. each populated solely by one class. This is similar to the process described by Chiu (1997).

Once the system has completed the processes of Modal Evolution, Modal Re-evolution and Nested Evolutionary Search, the data will be classified and execution terminates. The resulting fuzzy rules within the evolved rule-set are pruned (to remove any duplicates or any rules with excessively low fitness scores) and connected using 'OR' operators. The rule set is then applied to the data one final time, to obtain the accuracy of the overall classification.

## 4. Results

The system has been applied to numerous different data sets, including hand-designed test data, the standard "Breast Cancer" data set from the UCI data repository, and insurance data provided by Lloyds/TSB. This paper presents the results obtained from the "Breast Cancer" data set.

### 4.1 Results For Wisconsin Breast Cancer Database

The Wisconsin Breast Cancer database contains information about clinical cases from University of Wisconsin Hospitals, reported by Dr. William Wolberg. As shown in table 1, there are ten attributes for each data item, and each may be classified as benign or malignant. The pattern classifier must predict the class of new cases, given the ten attribute values.

New samples arrive periodically as Dr. Wolberg reports his clinical cases. The database therefore reflects this chronological grouping of the data, and is constantly increasing in size and difficulty for pattern-recognition. Table 2 shows the results obtained from two earlier pattern recognition experiments, and the results obtained using the fuzzy-GP system introduced in this paper. Preliminary results are promising. Training the system on 50% of the "Breast Cancer" data set and testing on the other half, a success rate of 95% was attained. This compares well with the other reported results in the literature of: 93.5% & 95.9% (Wolberg & Mangasarian, 1990) or 93.7% & 92.2% (Zhang, 1992).

Table 3 shows an example rule-set evolved by the fuzzy-GP system for this data.

**Table 1** The attributes within each breast cancer data item

<i>Attribute</i>	<i>Domain</i>	<i>Attribute</i>	<i>Domain</i>
1. Sample code number	id number	7. Bare Nuclei	1 - 10
2. Clump Thickness	1 - 10	8. Bland Chromatin	1 - 10
3. Uniformity of Cell Size	1 - 10	9. Normal Nucleoli	1 - 10
4. Uniformity of Cell Shape	1 - 10	10. Mitoses	1 - 10
5. Marginal Adhesion	1 - 10	11. Class:	(2 for benign, 4 for malignant)
6. Single Epithelial Cell Size	1 - 10		
Missing attribute values: 16 (Given random values here) Class distribution: Benign: 458 (65.5%), Malignant: 241 (34.5%)			

**Table 2** Comparison of classification results

<i>Name</i>	<i>Method</i>	<i>Size Of Set</i>	<i>Best Result</i>	<i>Detail</i>
Wolberg,W.H., Mangasarian,O.L. (1990).	Pattern separation for medical diagnosis applied to breast cytology	369	95.9%	Three pairs of parallel hyperplanes were found to be consistent with 67% of data
Zhang,J. (1992).	Applied 4 Instance-based Learning Algorithms	369	93.7%	Trained on 200 instances, tested on the other 169
<b>Bentley, P. Mallinson,H. (1998)</b>	<b>Evolving Fuzzy Rules</b>	<b>699</b>	<b>95%</b>	<b>Trained on half of Data Set and tested on remainder, (Data: -229 Benign, 120 Malignant) In six runs average 6 misclassifications 6/120 = 5% error.</b>

**Table 3** Example of Evolved Rule Set.

<i>Final Rule Set</i>
(IS_LOW Bare AND IS_LOW Clump)
(IS_LOW Bare AND IS_LOW Uniformity1)
(IS_LOW Bare AND IS_LOW Single)
(IS_LOW Uniformity2 AND IS_LOW Clump)
(IS_LOW Uniformity2 AND NOT IS_HIGH Bland)
(IS_LOW Bland AND IS_LOW Uniformity1)
(IS_LOW Uniformity1 AND IS_LOW Clump)
(IS_MEDIUM Clump AND IS_MEDIUM Bare)
(NOT Marginal AND NOT Bland)
(IS_LOW Normal AND IS_LOW Clump)

(Rules form a disjunction)

## 5. Conclusions

This paper has described the use of an evolutionary-fuzzy system to evolve sets of variable-length fuzzy rules which describe the difference between two classes of data held in a single database. Numerous novel techniques, including the use of genotypes in Genetic Programming, two new genetic crossover operators, and the processes of Modal Evolution, Modal Re-evolution and Nested Evolutionary Search were described. The results of experiments indicate that this approach to pattern recognition is showing significant success.

## 6. Further Work

This work is on-going. Potential avenues for future investigation include:

- Comparison of clustering techniques for the initial positions and shapes of the membership functions; fuzzy k-means, mountain methods, and Kohonen networks for example.
- Evolution of membership functions positions. Presently we are only able to fine-tune the rules by a process of nested subdivision.
- Comparison of t-norms on the Insurance Claim set. Presently we use max-min.

## Acknowledgements

Data was provided by the Data Set Repository at the Information and Computer Science Dept., University of California, Irvine. The breast cancer database originated from University of Wisconsin Hospitals, Madison, assembled by Dr. William H. Wolberg: <ftp://ftp.ics.uci.edu/pub/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wisconsin.names>

Insurance data has been kindly provided by Lloyds/TSB. This project is being performed in collaboration with Searchspace Ltd and Lloyds/TSB, and is funded by the EPSRC, ref: GR/L3/708.

## References

- [1] S. Abe, and M. S. Lan. A Classifier using Fuzzy Rules Extracted Directly from Numerical Data. *Proc. of 2<sup>nd</sup> IEEE International Conference on Fuzzy Systems*. San Francisco, Calif., (1993) 1191-1198.
- [2] R. Belman, R. Kalaba, and L. Zadeh, Abstraction and pattern classification, *Journal of Math.Anal. Appl.*, vol. 13 (1966) 1-7.
- [3] P. J. Bentley, & J. P. Wakefield. Finding Acceptable Solutions in the Pareto-Optimal Range using Multiobjective Genetic Algorithms. Chawdhry, P.K., Roy, R., & Pant, R.K. (eds) *Soft Computing in Engineering Design and Manufacturing*. Springer Verlag London Limited, Part 5, (1997) 231-240.
- [4] L. B. Booker, D. E. Goodberg, and J. H. Holland. Classifier Systems and Genetic Algorithms, *Artificial Intelligence* vol. 40. No. 1-3 () 235 - 282.
- [5] S. Chiu. Extracting Fuzzy rules from Data for Function Approximation and Pattern Classification. In *Fuzzy Information Engineering*. Ed. Dubois,Prade, Yager Wiley (1997).
- [6] F. L. Chung and T. Lee. A Fuzzy k-nearest Neighbour Algorithm. *IEEE Transactions on System, Man, and Cybernetics*, Vol. 15 (1985) 580 –585.
- [7] R. Eberhart, P. Simpson, R. Dobbins. *Computational Intelligence PC Tools*. Academic Press, London, 1996.
- [8] D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.
- [9] A. González, R. Pérez, and J. L. Verdegay. Learning the structure of a fuzzy rule: A genetic approach. *Fuzzy Systems and Artificial Intelligence* 3(1) (1994) 57-70.
- [10] H. Ishibuchi, T. Murata, T. Nakashima. Genetic- Algorithm-Based Approaches to Classification Problems', in *Fuzzy Evolutionary Computation* Witold Pedrycz (Ed) (1997) 127-153.
- [11] J. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.
- [12] R. E. Marmelstein and G. B. Lamont. Evolving Compact Decision Rule Sets. In Koza, John R. (ed.). *Late Breaking Papers at the Genetic Programming 1997 Conference*, University of Wisconsin, July 22-25, 1998. Madison, WI: Omni Press (1998) 144-150.
- [13] R. J. Ross. *Fuzzy Logic with Engineering Applications*. McGraw-Hill, 1995.
- [14] S. F. Smith. *A learning system based on genetic algorithms*, Ph. D. Dissertation, Univeristy of Pittsburgh, Pittsburgh, PA, 1980.
- [15] M. Valenzuela-Rendon, The fuzzy classifieer system: A classifier system for continuoiuusly varying variables, *Proc. Of 4<sup>th</sup> Cinternational Conf. On Genetic Algorithms*, University of California, San Diego, CA, July 13-16 (1991) 346-353.
- [16] W. H. Wolberg & O. L. Mangasarian. Multisurface method of pattern separation for medical [fig.1]nosis applied to breast cytology. In *Proc. of the National Academy of Sciences* (1990) 9193--9196.
- [17] J. Zhang. Selecting typical instances in instance-based learning. In *Proceedings of the Ninth International Machine Learning Conference*. Aberdeen, Scotland: Morgan Kaufmann. (1992) 470--479
- [18] H. J. Zimmerman, 'Hybrid Approaches for Fuzzy Data Analysis and Configuration Using Genetic Algorithms and Evolutionary Methods', in *Computational Intelligence Imitating Life*, Zurada, J. M., Marks, R.J., Robinson, C. J. (Eds) (1994) 364-370.