Journal of Intelligent & Fuzzy Systems 26 (2014) 451–464
DOI:10.3233/IFS-130857
IOS Press

Complexity management methodology for fuzzy systems with feedback rule bases

Alexander Gegov[a,*], David Sanders[b] and Boriana Vatchova[c]
[a]*School of Computing, University of Portsmouth, Buckingham Building, Portsmouth, UK*
[b]*School of Engineering, University of Portsmouth, Angleasea Building, Portsmouth, UK*
[c]*Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, Sofia, Bulgaria*

Abstract. This paper proposes a complexity management methodology for fuzzy systems with feedback rule bases. The methodology is based on formal methods for presentation, manipulation and transformation of fuzzy rule bases. First, Boolean matrices are used for formal presentation of rule bases. Then, binary merging operations are used for formal manipulation of rule bases. Finally, repetitive merging operations are used for formal transformation of rule bases. The formal methods facilitate the understanding and modelling of fuzzy systems in terms of interacting subsystems. In particular, the methods reduce the qualitative complexity in fuzzy systems by improving the transparency of the rule bases.

Keywords: Fuzzy systems, complexity management, formal methods, rule bases, fuzzy networks

1. Introduction

Fuzzy systems are usually good at capturing the qualitative complexity of a wide range of problems by means of their linguistic modeling and approximate reasoning capabilities. However, this comes at a price because the associated operations during fuzzification, inference and defuzzification increase the quantitative complexity of the solution to these problems. This price gets even higher as the amount of fuzzy operations increases as a result of the increased number of rules in the fuzzy system.

The number of rules in a fuzzy system is often an exponential function of the number of inputs to the system and the number of linguistic values that these inputs can take [5, 17, 24, 32]. This exponential function has been used as a main indicator for the quantitative complexity of the associated fuzzy system.

There has been a growing interest recently in complexity issues of fuzzy systems [1, 9, 16, 25]. This is due to the fact that fuzzy systems are already more widely used in large-scale applications where their quantitative complexity becomes more obvious. In particular, many methods have been developed for reducing this quantitative complexity. These are known as complexity reduction methods as they reduce the number of rules by reducing the number of inputs or the number of linguistic values that these inputs can take. The main objective in this case is to suppress the associated exponential function. These methods are classified into six groups and discussed below.

The first group of methods are aimed at removing less significant or merging similar linguistic values [11, 23]. From these two strands, the one based on removal of linguistic values is more straightforward but it involves a higher risk as a result of the removal of the associated fuzzy set. On the other hand, the strand based on merging of linguistic values is more difficult for application due to the necessity to define a new fuzzy set for each of the merged linguistic values.

[*] Corresponding author. Dr Alexander Gegov, School of Computing, University of Portsmouth, Buckingham Building, Portsmouth PO1 3HE, UK. Tel.: +44 23 9284 6381; E-mail: alexander.gegov@port.ac.uk.

The second group of methods are aimed at removing less significant or merging similar inputs [18, 30]. From these two strands, the one based on removal of inputs is more straightforward but it involves a higher risk as a result of the removal of the associated physical variable. On the other hand, the strand based on merging of inputs is more difficult for application due to the necessity to justify physically the merging of the associated variables.

The third group of methods are based on singular value decomposition of the matrix representing the crisp values of the output from a fuzzy system [6, 33]. As a result of this decomposition, the number of linguistic values for the inputs to the system is reduced. Although this group of methods can be quite effective in reducing the number of rules in a fuzzy system, they are applicable mainly for systems with two inputs. In the case of more inputs, the singular value decomposition process becomes quite complex as the dimension of the space in which the associated matrix is defined increases significantly.

The fourth group of methods are based on conversion of the intersection rule configuration of a fuzzy system into a union rule configuration with a smaller number of rules [13, 31]. This group of methods can be quite effective in reducing the number of rules in a fuzzy system but they can only be applied to a special class of problems called ‘additively separable’. For problems that don’t belong to this class, the conversion of the intersection rule configuration into a union rule configuration is not possible.

The fifth group of methods convert a fuzzy system into spatially decomposed subsystems as a result of which the overall number of rules is reduced [3, 4, 7, 8, 27, 28]. In this case, the interactions among the subsystems are partially compensated and the resulting decomposed system has a decoupled structure. Although this group of methods have been widely used recently, the success of their application depends on the strength of interactions among the subsystems and the level of their compensation.

The sixth group of methods rearrange the inputs in a fuzzy system in a way that leads to the reduction of the number of rules [10, 15, 19–21, 26]. In this case, the fuzzy system is decomposed into a multilayer hierarchical structure such that each layer has only two inputs and one output. Although these methods have become quite popular recently, they don’t offer clear interpretation of the intermediate variables between the first and the last layer. Besides this, only two inputs are taken into account in each layer while all other inputs are ignored.

The above complexity reduction methodology for fuzzy systems has serious drawbacks such as empirical nature and limited scope. The empirical nature of the methods in groups 1–2 and 5–6 assumes the use of a ‘trial and error’ approach that can be unreliable. Besides this, the limited scope of the methods in groups 3–4 makes them inapplicable to a wide range of fuzzy systems. And finally, all methods in groups 1–6 do not reduce the qualitative complexity in terms of the opaqueness of rules.

This paper addresses the above three drawbacks of the current complexity reduction methodology for fuzzy systems by proposing a novel complexity management methodology. The main advantages of this novel methodology are its systematic nature and universal scope as well as its focus on reducing qualitative complexity in fuzzy systems.

The underlying philosophy of this novel methodology deals with complexity related issues in fuzzy systems from a wider perspective. This perspective takes into account mainly factors that affect the qualitative complexity of the fuzzy system, e.g. the way in which the rule bases are handled. For this reason, the more general term ‘complexity management’ is used here instead of the fairly specific term ‘complexity reduction’.

The remaining part of this paper is structured as follows. Section 2 introduces some theoretical preliminaries for fuzzy systems. Sections 3–5 describe the complexity management methodology for fuzzy systems in terms of formal methods for presentation, manipulation and transformation of rule bases. Section 6 illustrates this methodology for a feedback fuzzy rule based network. Section 7 summarises the main advantages of the methodology and highlights future research directions.

2. Theoretical preliminaries

A fuzzy system can be represented by the following rule base

$$
\begin{aligned}
&\text{If } i_1 \text{ is } v_{i11} \text{ and } \ldots \text{ and } i_m \text{ is } v_{im1} \text{ then } o_1 \text{ is } v_{o11} \text{ and} \\
&\ldots \text{ and } o_n \text{ is } v_{on1} \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
&\text{If } i_1 \text{ is } v_{i1r} \text{ and } \ldots \text{ and } i_m \text{ is } v_{imr} \text{ then } o_1 \text{ is } v_{o1r} \text{ and} \\
&\ldots \text{ and } o_n \text{ is } v_{onr}
\end{aligned}
\qquad (1)
$$

where m is the number of inputs, n is the number of outputs and r is the number of rules [12, 14]. In this case, i_p, p = 1,..,m represents the p-th input, v_{ips}, p = 1,..,m, s = 1,..,r is the linguistic value of the p-th input in the s-th rule, o_q, q = 1,..,n represents the q-th output and v_{oqs}, q = 1,..,n, s = 1,..,r is the linguistic value of the q-th output in the s-th rule.

A fuzzy system operates in three main stages - fuzzification, inference and defuzzification. The inference stage includes three substages - application, implication and aggregation [22, 29]. In multiple-output fuzzy systems, each output is considered separately and in relation to the same set of inputs. Therefore, the three main stages above are applied repetitively for each output.

The maximum number of rules r in a fuzzy system is an exponential function of the number of inputs m and the number of linguistic values w that each input can take. If this number is a constant, the maximum number of rules is given by

$$r = w^m \tag{2}$$

where v is the number of linguistic values per input.

However, if the number of linguistic values that each input can take is not a constant, the maximum number of rules in a fuzzy system is given by

$$r = w_1 \ldots w_m \tag{3}$$

where w_p, p = 1,..,m is the number of linguistic values that the p-th input can take.

Fuzzy rule bases have some important properties [2]. These properties describe the extent to which the permutations of linguistic values of inputs and outputs are present in the rule base. The properties also describe the type of mapping in the rule base between permutations of linguistic values of inputs in the ‘if’ part and permutations of linguistic values of outputs in the ‘then’ part. Four basic properties of fuzzy rule bases are introduced below by definitions. These definitions make use of logical equivalence, i.e. a property is present when the corresponding condition holds and vice versa. This logical equivalence also implies that a property is absent when the corresponding condition doesn’t hold and vice versa.

Definition 1. A fuzzy rule base is complete if and only if all possible permutations of linguistic values of inputs are present in the ‘if’ part of the rule base.

Definition 2. A fuzzy rule base is exhaustive if and only if all possible permutations of linguistic values of outputs are present in the ‘then’ part of the rule base.

Definition 3. A fuzzy rule base is consistent if and only if every present permutation of linguistic values of inputs is mapped to only one permutation of linguistic values of outputs.

Definition 4. A fuzzy rule base is monotonic if and only if every present permutation of linguistic values of outputs is mapped from only one permutation of linguistic values of inputs.

It is desirable for a fuzzy rule base to be complete, i.e. with all possible permutations of linguistic values of inputs present, although that may not always be the case. As far as the permutations of linguistic values of the outputs are concerned, it is fairly common for some of them to be missing and therefore a fuzzy rule base is likely to be non-exhaustive. Ideally, a fuzzy rule base must be consistent, i.e. with each present permutation of linguistic values of inputs mapped to only one permutation of linguistic values of outputs. And finally, it is quite common for some permutations of linguistic values of outputs to be mapped from more than one permutation of linguistic values of inputs and therefore a fuzzy rule base is likely to be non-monotonic.

The aim of the complexity management methodology for fuzzy systems is to provide formal methods for presentation, manipulation and transformation of rule bases. These methods facilitate the understanding and modelling of fuzzy systems in terms of interacting subsystems. In particular, the methods reduce the qualitative complexity in fuzzy systems by improving the transparency of the rule bases.

3. Formal presentation of fuzzy systems

Fuzzy systems can be presented formally by Boolean matrices. The latter have been studied thoroughly by mathematicians and applied successfully by engineers in many areas. Some basic definitions for Boolean matrices are given below.

Definition 5. An m × n Boolean matrix is a matrix with m rows and n columns whose elements can take only the values 0 and 1.

Definition 6. An m × n null Boolean matrix is a matrix with m rows and n columns all of whose elements are equal to 0.

Definition 7. An m × n universal Boolean matrix is a matrix with m rows and n columns all of whose elements are equal to 1.

Definition 8. A Boolean matrix is square if and only if the number of its rows is equal to the number of its columns.

Definition 9. A Boolean matrix is homogenous if and only if its row and column labels are of the same type.

Definition 10. An element in a Boolean square matrix is on-diagonal if and only if its row and column index are the same.

Definition 11. An element in a Boolean square matrix is off-diagonal if and only if its row and column index are different.

Definition 12. An identity Boolean matrix is a square homogenous Boolean matrix all of whose on-diagonal and off-diagonal elements are equal 1 and 0, respectively.

The basic operations that can be applied to elements of Boolean matrices are 'addition' and 'multiplication'. They are both binary operations as they can only be applied to two operands. In the case of more than two elements, each of the two operations can be applied in a sequential manner, i.e. only two elements are considered at each step and the result from the current step becomes an operand in the next step. The 'addition' operation has the effect of taking the 'maximum' of the elements whereas the 'multiplication' operation has the effect of taking the 'minimum' of the elements. Both operations are commutative, i.e. the result is not affected if the positions of the two elements are swapped.

In terms of the values of the first and the second element, there are four different permutations for the 'addition' operation which are described by the following equations:

$$1 + 1 = \max (1, 1) = 1 \tag{4}$$

$$0 + 1 = \max (0, 1) = 1 \tag{5}$$

$$1 + 0 = \max (1, 0) = 1 \tag{6}$$

$$0 + 0 = \max (0, 0) = 0 \tag{7}$$

Similarly, there are four different permutations for the 'multiplication' operation which are described by the following equations:

$$1 \,.\, 1 = \min (1, 1) = 1 \tag{8}$$

$$0 \,.\, 1 = \min (0, 1) = 0 \tag{9}$$

$$1 \,.\, 0 = \min (1, 0) = 0 \tag{10}$$

$$0 \,.\, 0 = \min (0, 0) = 0 \tag{11}$$

Boolean matrices are multiplied in almost the same way as conventional matrices, i.e. matrices whose elements can take any values. Each element in a Boolean matrix product A*B can be obtained by multiplying each row from the first matrix A with its counterpart column from the second matrix B. In this case, the row index of an element A*B is the same as the index of the corresponding row from the matrix A whereas the column index of an element in A*B is the same as the index of the corresponding column from the matrix B.

The multiplication compatibility rule for Boolean matrices is the same as the rule for conventional matrices, i.e. the number of columns in the first matrix must be equal to the number of rows in the second matrix. The only difference is that instead of applying the arithmetic 'addition' and 'multiplication' operations on elements of the matrices, we apply the 'maximum' and 'minimum' operations, respectively. The formal presentation of a fuzzy rule base by a Boolean matrix is described by Algorithm 1 and illustrated by Examples 1–3.

Algorithm 1:

1. Sort all possible permutations of linguistic values of inputs from the rule base in an ascending order.
2. Sort all possible permutations of linguistic values of outputs from the rule base in an ascending order.
3. Label the rows of the Boolean matrix with the sorted permutations of linguistic values of inputs.
4. Label the columns of the Boolean matrix with the sorted permutations of linguistic values of outputs.
5. Go through all the elements of the Boolean matrix and set each element equal to 1 or 0 using steps 6 and 7.

6. If an element of the Boolean matrix reflects an existing mapping from an input onto an output permutation, set it equal to 1.
7. If an element of the Boolean matrix reflects a non-existing mapping from an input onto an output permutation, set it equal to 0.

Example 1. A single-input-single-output fuzzy system is considered whereby the input i and the output o can take the linguistic values S (small), M (medium) and B (big). This system is described by the following rule base:

$$\begin{aligned}&\text{If } i \text{ is S then } o \text{ is B}\\&\text{If } i \text{ is M then } o \text{ is M}\\&\text{If } i \text{ is B then } o \text{ is S}\end{aligned}\tag{12}$$

The linguistic values S, M, and B can be substituted by the integers 1, 2 and 3, respectively. In this case, the rule base can be presented by the following Boolean matrix:

$$\begin{array}{c|ccc} i/o & 1 & 2 & 3\\ 1 & 0 & 0 & 1\\ 2 & 0 & 1 & 0\\ 3 & 1 & 0 & 0\end{array}\tag{13}$$

Example 2. A two-input-two-output fuzzy system is considered whereby the inputs i_1, i_2 and the outputs o_1, o_2 can take the linguistic values S (small), M (medium) and B (big). This system is described by the following rule base:

$$\begin{aligned}&\text{If } i_1 \text{ is S and } i_2 \text{ is S then } o_1 \text{ is B and } o_2 \text{ is B}\\&\text{If } i_1 \text{ is S and } i_2 \text{ is M then } o_1 \text{ is B and } o_2 \text{ is M}\\&\text{If } i_1 \text{ is S and } i_2 \text{ is B then } o_1 \text{ is B and } o_2 \text{ is S}\\&\text{If } i_1 \text{ is M and } i_2 \text{ is S then } o_1 \text{ is M and } o_2 \text{ is B}\\&\text{If } i_1 \text{ is M and } i_2 \text{ is M then } o_1 \text{ is M and } o_2 \text{ is M}\\&\text{If } i_1 \text{ is M and } i_2 \text{ is B then } o_1 \text{ is M and } o_2 \text{ is S}\\&\text{If } i_1 \text{ is B and } i_2 \text{ is S then } o_1 \text{ is S and } o_2 \text{ is B}\\&\text{If } i_1 \text{ is B and } i_2 \text{ is M then } o_1 \text{ is S and } o_2 \text{ is M}\\&\text{If } i_1 \text{ is B and } i_2 \text{ is B then } o_1 \text{ is S and } o_2 \text{ is S}\end{aligned}\tag{14}$$

The linguistic values S, M, and B can be substituted by the integers 1, 2 and 3, respectively. In this case, the rule base can be presented by the following Boolean matrix:

$$\begin{array}{c|ccccccccc} i_1, i_2 / o_1, o_2 & 11 & 12 & 13 & 21 & 22 & 23 & 31 & 32 & 33\\ 11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\\ 12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0\\ 13 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0\\ 21 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0\\ 22 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\ 23 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0\\ 31 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\ 32 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 33 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\end{array}\tag{15}$$

Example 3. A three-input-three-output fuzzy system is considered whereby the inputs i_1, i_2, i_3 and the outputs o_1, o_2, o_3 can take the linguistic values S (small) and B (big). This system is described by the following rule base:

$$\begin{aligned}&\text{If } i_1 \text{ is S and } i_2 \text{ is S and } i_3 \text{ is S}\\&\quad\text{then } o_1 \text{ is B and } o_2 \text{ is B and } o_3 \text{ is B}\\&\text{If } i_1 \text{ is S and } i_2 \text{ is S and } i_3 \text{ is B}\\&\quad\text{then } o_1 \text{ is B and } o_2 \text{ is B and } o_3 \text{ is S}\\&\text{If } i_1 \text{ is S and } i_2 \text{ is B and } i_3 \text{ is S}\\&\quad\text{then } o_1 \text{ is B and } o_2 \text{ is S and } o_3 \text{ is B}\\&\text{If } i_1 \text{ is S and } i_2 \text{ is B and } i_3 \text{ is B}\\&\quad\text{then } o_1 \text{ is B and } o_2 \text{ is S and } o_3 \text{ is S}\\&\text{If } i_1 \text{ is B and } i_2 \text{ is S and } i_3 \text{ is S}\\&\quad\text{then } o_1 \text{ is S and } o_2 \text{ is B and } o_3 \text{ is B}\\&\text{If } i_1 \text{ is B and } i_2 \text{ is S and } i_3 \text{ is B}\\&\quad\text{then } o_1 \text{ is S and } o_2 \text{ is B and } o_3 \text{ is S}\\&\text{If } i_1 \text{ is B and } i_2 \text{ is B and } i_3 \text{ is S}\\&\quad\text{then } o_1 \text{ is S and } o_2 \text{ is S and } o_3 \text{ is B}\\&\text{If } i_1 \text{ is B and } i_2 \text{ is B and } i_3 \text{ is B}\\&\quad\text{then } o_1 \text{ is S and } o_2 \text{ is S and } o_3 \text{ is S}\end{aligned}\tag{16}$$

The linguistic values S and B can be substituted by the integers 1 and 2, respectively. In this case, the rule base can be presented by the following Boolean matrix:

$$\begin{array}{c|cccccccc}
i_1, i_2, i_3/o_1, o_2, o_3 & 111 & 112 & 121 & 122 & 211 & 212 & 221 & 222 \\
111 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
112 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
121 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
122 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
211 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
212 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
221 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
222 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array} \tag{17}$$

It is obvious from the examples above that the Boolean matrix presentation of a fuzzy rule base facilitates the definition of the properties of the rule base. These properties can be implied directly from some properties of the Boolean matrix such as the number of non-zero elements in its rows and columns. The following definitions show how this works.

Definition 13. A fuzzy rule base is complete if and only if each row in its Boolean matrix contains at least one non-zero element.

Definition 14. A fuzzy rule base is exhaustive if and only if each column in its Boolean matrix contains at least one non-zero element.

Definition 15. A fuzzy rule base is consistent if and only if each row in its Boolean matrix contains not more than one non-zero element.

Definition 16. A fuzzy rule base is monotonic if and only if each column in its Boolean matrix contains not more than one non-zero element.

4. Formal manipulation of fuzzy systems

Fuzzy systems can be manipulated formally using Boolean matrices. In this case, pairs of individual rule bases in a multiple rule based fuzzy system can be merged either horizontally, vertically or with respect to common inputs. The specific type of manipulation is chosen on the basis of the location of the corresponding rule bases with respect to each other. The remaining part of this section describes in detail three different operations for formal manipulation of fuzzy rule bases.

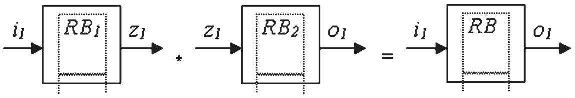

Fig. 1. Horizontal merging of rule bases RB_1 and RB_2 into rule base RB.

4.1. Horizontal merging operation

The process of merging two fuzzy rule bases in sequence into a single fuzzy rule base is called 'horizontal merging' and it is shown in Fig. 1.

The horizontal merging operation is based on horizontal composition of rule bases which is identical to a Boolean matrix multiplication. This operation is binary in that it can be applied to only two operands at a time. The operands in this case are the Boolean matrices representing the operand rule bases. The result from the application of this operation is a single Boolean matrix representing the product rule base.

The application of the horizontal composition operation to Boolean matrices is described by Algorithm 2 and illustrated by Example 4.

Algorithm 2:

1. Label the rows of the product matrix with the row labels from the first operand matrix.
2. Label the columns of the product matrix with the column labels from the second operand matrix.
3. Set each element of the product matrix equal to 1 or 0 by mapping it from the corresponding row in the first operand matrix and the corresponding column in the second operand matrix, as described in step 4.
4. Find the product matrix by multiplying the operand matrices using the operations for 'addition' and 'multiplication' of elements, as defined by Equations (4)–(11).

Example 4. The operand rule bases RB_1 and RB_2 are presented by the following Boolean matrices:

$$RB_1: \quad \begin{array}{c|ccc}
i_1/z_1 & 1 & 2 & 3 \\
1 & 0 & 1 & 0 \\
2 & 1 & 0 & 0 \\
3 & 0 & 0 & 1
\end{array} \tag{18}$$

$$RB_2: \quad \begin{array}{cccc} z_1/o_1 & 1 & 2 & 3 \\ 1 & 1 & 0 & 0 \\ 2 & 0 & 0 & 1 \\ 3 & 0 & 1 & 0 \end{array} \qquad (19)$$

The horizontal merging of RB_1 and RB_2 into a product rule base RB will be denoted by $RB_1{*}RB_2 = RB$ where RB will be presented by the following Boolean matrix:

$$RB: \quad \begin{array}{cccc} i_1/o_1 & 1 & 2 & 3 \\ 1 & 0 & 0 & 1 \\ 2 & 1 & 0 & 0 \\ 3 & 0 & 1 & 0 \end{array} \qquad (20)$$

4.2. *Vertical merging operation*

The process of merging two fuzzy rule bases in parallel into a single fuzzy rule base is called 'vertical merging' and it is shown in Fig. 2.

The vertical merging operation is based on vertical composition of rule bases which is identical to a Kroneker matrix product. This operation is binary in that it can be applied to only two operands at a time.

The operands in this case are the Boolean matrices representing the operand rule bases. The result from the application of this operation is a single Boolean matrix representing the product rule base.

The application of the vertical composition operation to Boolean matrices is described by Algorithm 3 and illustrated by Example 5.

Algorithm 3:

1. Construct all possible permutations of row labels from the operand matrices and sort them.
2. Construct all possible permutations of column labels from the operand matrices and sort them.
3. Label the rows of the product matrix with the sorted permutations of row labels from the operand matrices.
4. Label the columns of the product matrix with the sorted permutations of column labels from the operand matrices.
5. Go through all the elements of the operand matrices and set each element of the product matrix equal to 1 or 0, as described in steps 6 and 7.
6. If an element of the product matrix is mapped from a pair of non-zero elements in the product matrices, set this element equal to 1 in accordance with Equations (8)–(11).
7. If an element of the product matrix is mapped from a pair of elements in the product matrices at least one of which is zero, set this element equal to 0 in accordance with Equations (8)–(11).

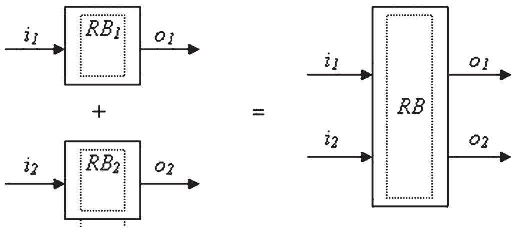

Fig. 2. Vertical merging of rule bases RB_1 and RB_2 into rule base RB.

Example 5. The operand rule bases RB_1 and RB_2 are presented by the following Boolean matrices:

$$RB_1: \quad \begin{array}{cccc} i_1/o_1 & 1 & 2 & 3 \\ 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 0 & 0 & 1 \end{array} \qquad (21)$$

$$RB_2: \quad \begin{array}{cccc} i_2/o_2 & 1 & 2 & 3 \\ 1 & 1 & 0 & 0 \\ 2 & 0 & 0 & 1 \\ 3 & 0 & 1 & 0 \end{array} \qquad (22)$$

The vertical merging of RB_1 and RB_2 into a product rule base RB will be denoted by $RB_1 + RB_2 = RB$ where RB will be presented by the following Boolean matrix:

$$RB: \quad \begin{array}{cccccccccc} i_1, i_2\,/o_1, o_2 & 11 & 12 & 13 & 21 & 22 & 23 & 31 & 32 & 33 \\ 11 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 12 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 13 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 21 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 22 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 23 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 31 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 32 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 33 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array} \qquad (23)$$

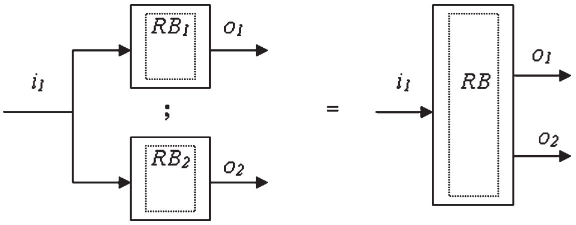

Fig. 3. Output merging of rule bases RB_1 and RB_2 into rule base RB.

4.3. Output merging operation

The process of merging two fuzzy rule bases with common inputs into a single fuzzy rule base is called 'output merging' and it is shown in Fig. 3.

The output merging operation is a special case of the vertical merging operation. This operation is based on output composition of rule bases which is identical to a modified Kroneker matrix product. This operation is binary in that it can be applied to only two operands at a time. The operands in this case are the Boolean matrices representing the operand rule bases. The result from the application of this operation is a single Boolean matrix representing the product rule base.

The application of the output composition operation to Boolean matrices is described by Algorithm 4 and illustrated by Example 6.

Algorithm 4:

1. Label the rows of the product matrix with the common row labels of the two operand matrices.
2. Label the columns of the product matrix with the sorted permutations of the column labels of the two operand matrices.
3. Go through all the elements of the operand matrices and set each element of the product matrix equal to 1 or 0, as described in steps 4 and 5.
4. If an element of the product matrix is mapped from two non-zero elements in the operand matrices, set this element equal to 1 in accordance with Equations (8)–(11).
5. If an element of the product matrix is mapped from two elements in the operand matrices such that at least one of them is zero, set this element equal to 0 in accordance with Equations (8)–(11).

Example 6. The operand rule bases RB1 and RB2 are presented by the following Boolean matrices:

$$RB_1: \quad \begin{array}{cccc} i_1/o_1 & 1 & 2 & 3 \\ 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 0 & 0 & 1 \end{array} \qquad (24)$$

$$RB_1: \quad \begin{array}{cccc} i_2/o_2 & 1 & 2 & 3 \\ 1 & 1 & 0 & 0 \\ 2 & 0 & 0 & 1 \\ 3 & 0 & 1 & 0 \end{array} \qquad (25)$$

The output merging of RB_1 and RB_2 into a product rule base RB will be denoted by $RB_1;RB_2 = RB$ where RB will be presented by the following Boolean matrix:

$$RB: \quad \begin{array}{cccccccccc} i_1/o_1,\, o_2 & 11 & 12 & 13 & 21 & 22 & 23 & 31 & 32 & 33 \\ 11 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 12 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 13 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array} \qquad (26)$$

5. Formal transformation of fuzzy systems

Fuzzy systems can be transformed formally using Boolean matrices. In this case, three or more individual rule bases in a multiple rule based fuzzy system can be merged either horizontally, vertically or with respect to common inputs. The specific type of manipulation is chosen on the basis of the location of the corresponding rule bases with respect to each other. The remaining part of this section describes in detail three different methods for formal transformation of fuzzy rule bases.

As all three merging operations are associative, it is possible to change the order of operations on three or more operand rule bases in the case of repetitive merging. Therefore, the changing of the order of operations for any three or more operand rule bases will not affect the product rule base.

5.1. Horizontal merging property

The associativity property of the horizontal merging operation is illustrated by Example 7.

Example 7. The rule bases in sequence RB_1, RB_2 and RB_3 are presented by the Boolean matrices in Equations (27)–(29).

$$\mathrm{RB}_1: \quad \begin{array}{ccc} i_1/z_{12} & 1 & 2 \\ 1 & 1 & 0 \\ 2 & 1 & 0 \end{array} \tag{27}$$

$$\mathrm{RB}_2: \quad \begin{array}{ccc} z_{12}/z_{23} & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{array} \tag{28}$$

$$\mathrm{RB}_3: \quad \begin{array}{ccc} z_{23}/o_3 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 0 & 1 \end{array} \tag{29}$$

The associativity of horizontal merging of rule bases RB_1, RB_2 and RB_3 is described by Fig. 4 and Equation (30).

$$\begin{aligned} &\text{If } \mathrm{RB}_{(1*2)*3} = (\mathrm{RB}_1 * \mathrm{RB}_2) * \mathrm{RB}_3 \text{ and } \mathrm{RB}_{1*(2*3)} \\ &\qquad = \mathrm{RB}_1 * (\mathrm{RB}_2 * \mathrm{RB}_3) \\ &\text{then } \mathrm{RB}_{(1*2)*3} = \mathrm{RB}_{1*(2*3)} \end{aligned} \tag{30}$$

In this case, the horizontal merging of rule bases RB_1, RB_2 and RB_3 from left to right will give the same result as their horizontal merging from right to left and the product rule base will be presented by the following Boolean matrix:

$$\mathrm{RB}_{(1*2)*3} = \mathrm{RB}_{1*(2*3)}: \quad \begin{array}{ccc} i_1/o_3 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 0 & 1 \end{array} \tag{31}$$

5.2. *Vertical merging property*

The associativity property of the vertical merging operation is illustrated by Example 8.

Example 8. The rule bases in parallel RB_1, RB_2 and RB_3 are presented by the Boolean matrices in Equations (32)–(34).

$$\mathrm{RB}_1: \quad \begin{array}{ccc} i_1/o_1 & 1 & 2 \\ 1 & 1 & 0 \\ 2 & 1 & 0 \end{array} \tag{32}$$

$$\mathrm{RB}_2: \quad \begin{array}{ccc} i_1/o_2 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{array} \tag{33}$$

$$\mathrm{RB}_3: \quad \begin{array}{ccc} i_3/o_3 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 0 & 1 \end{array} \tag{34}$$

The associativity of vertical merging of rule bases RB_1, RB_2 and RB_3 is described by Fig. 5 and Equation (35).

$$\begin{aligned} &\text{If } \mathrm{RB}_{(1+2)+3} = (\mathrm{RB}_1 + \mathrm{RB}_2) + \mathrm{RB}_3 \text{ and } \mathrm{RB}_{1+(2+3)} \\ &\qquad = \mathrm{RB}_1 + (\mathrm{RB}_2 + \mathrm{RB}_3) \quad \text{then} \\ &\mathrm{RB}_{(1+2)+3} = \mathrm{RB}_{1+(2+3)} \end{aligned} \tag{35}$$

In this case, the vertical merging of rule bases RB_1, RB_2 and RB_3 from top to bottom will give the same result as their vertical merging from bottom to top and the product rule base will be presented by the following Boolean matrix:

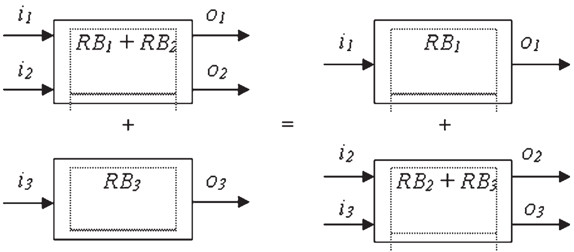

Fig. 5. Associativity of vertical merging of rule bases RB_1, RB_2 and RB_3.

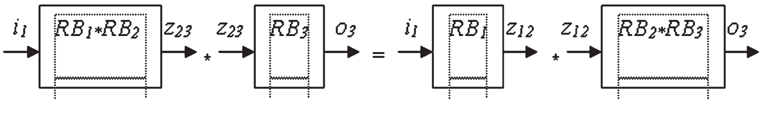

Fig. 4. Associativity of horizontal merging of rule bases RB_1, RB_2 and RB_3.

$$
\mathrm{RB}_{(1+2)+3} = \mathrm{RB}_{1+(2+3)}:
$$

$$
\begin{array}{ccccccccc}
\mathrm{i}_1,\mathrm{i}_2,\mathrm{i}_3/\mathrm{o}_1,\mathrm{o}_2,\mathrm{o}_3 & 111 & 112 & 121 & 122 & 211 & 212 & 221 & 222 \\
111 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
112 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
121 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
122 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
211 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
212 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
221 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
222 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
\qquad (36)
$$

5.3. Output merging property

The associativity property of the output merging operation is illustrated by Example 9.

Example 9. The rule bases in parallel RB$_1$, RB$_2$ and RB$_3$ are presented by the Boolean matrices in Equations (37)–(39).

$$
\mathrm{RB}_1: \quad
\begin{array}{ccc}
\mathrm{i}/\mathrm{o}_1 & 1 & 2 \\
1 & 1 & 0 \\
2 & 1 & 0
\end{array}
\qquad (37)
$$

$$
\mathrm{RB}_2: \quad
\begin{array}{ccc}
\mathrm{i}/\mathrm{o}_2 & 1 & 2 \\
1 & 0 & 1 \\
2 & 1 & 0
\end{array}
\qquad (38)
$$

$$
\mathrm{RB}_3: \quad
\begin{array}{ccc}
\mathrm{i}/\mathrm{o}_3 & 1 & 2 \\
1 & 0 & 1 \\
2 & 0 & 1
\end{array}
\qquad (39)
$$

The associativity of output merging of rule bases RB$_1$, RB$_2$ and RB$_3$ is described by Fig. 6 and Equation (40).

$$
\begin{aligned}
&\text{If } \mathrm{RB}_{(1;2);3} = (\mathrm{RB}_1 \,;\, \mathrm{RB}_2)\,;\, \mathrm{RB}_3 \text{ and } \mathrm{RB}_{1;(2;3)} \\
&\qquad = \mathrm{RB}_1 \,;\, (\mathrm{RB}_2 \,;\, \mathrm{RB}_3) \\
&\text{then } \mathrm{RB}_{(1;2);3} = \mathrm{RB}_{1;(2;3)}
\end{aligned}
\qquad (40)
$$

In this case, the output merging of rule bases RB$_1$, RB$_2$ and RB$_3$ from top to bottom will give the same result as their output merging from bottom to top and the product rule base will be presented by the following Boolean matrix:

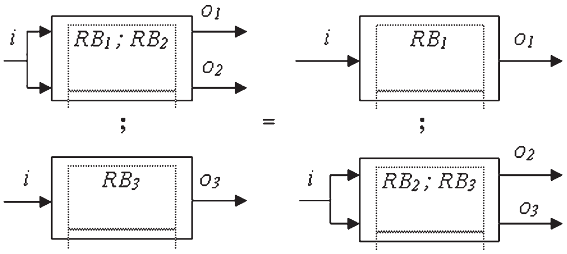

Fig. 6. Associativity of output merging of rule bases RB$_1$, RB$_2$ and RB$_3$.

$$
\mathrm{RB}_{(1;2);3} = \mathrm{RB}_{1;(2;3)}:
$$

$$
\begin{array}{ccccccccc}
\mathrm{i}/\mathrm{o}_1,\, \mathrm{o}_2,\, \mathrm{o}_3 & 111 & 112 & 121 & 122 & 211 & 212 & 221 & 222 \\
11 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
12 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
\qquad (41)
$$

6. Application to feedback fuzzy systems

The proposed complexity management methodology is applied to the modelling of a module management system with two rule bases and it is evaluated comparatively in terms of model transparency. In this case, RB is a feedforward rule base and RBF is a feedback rule base. The delivery aspects of module management are dealt with only in feedforward fashion whereas the assessment aspects are also included in the feedback loop. The pedagogic justification for this approach is based on the assumption that the assessment level for the current assessment period should reflect the assessment outcome from the previous assessment period such that students are not advantaged or disadvantaged in some way.

The above module management system can be presented as an initial fuzzy network, as shown in Fig. 7. The inputs and the outputs for this system are given below whereby the associated linguistic values are listed in brackets:

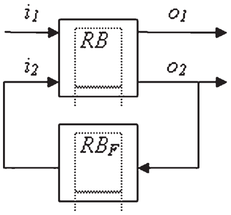

Fig. 7. Initial fuzzy network.

i_1 – module delivery level (low, medium, high)
i_2 – module assessment level (low, medium, high)
o_1 – module delivery outcome (poor, good, excellent)
o_2 – module assessment outcome (poor, good, excellent)

The locations of the feedforward rule bases in this initial fuzzy network are presented by a grid structure with horizontal levels and vertical levels as follows:

level/layer	layer 1
level 1	RB, i_1, i_2, o_1, o_2

(42)

The connections between all the bases in the initial fuzzy network are presented by a similar grid structure as follows:

level/layer	layer 1
level 1	$F(o_2) = i_2$

(43)

The initial fuzzy network is with local feedback because the associated feedback loop embraces only one rule base. In this case, the linguistic values of the output from the feedforward rule base RB are mapped to corresponding linguistic values of the input to the same rule base by the feedback rule base RB_F. In other words, the output from RB is an input to RB_F and the input to RB is an output from RB_F.

By introducing a second layer with two levels such that the first level is occupied by an identity rule base RB_I mapping the output o_1 to itself and the second level is occupied by the feedback rule base RB_F, the initial fuzzy network with non-identity feedback is transformed into a final fuzzy network with identity feedback and three rule bases - RB, RB_I and RB_F. In this case, RB_I and RB_F are standing in parallel in layer 2 whereas RB and RB_F are standing in sequence in level 2.

The locations of the rule bases in this final fuzzy network are presented by a grid structure with horizontal levels and vertical levels as follows:

level/layer	layer 1	layer 2
level 1		RB_I, i_I, o_I
level 2	RB, i_1, i_2, o_1, o_2	RB_F, i_F, o_F

(44)

The connections between the rule bases in the final fuzzy network are presented by a similar grid structure as follows:

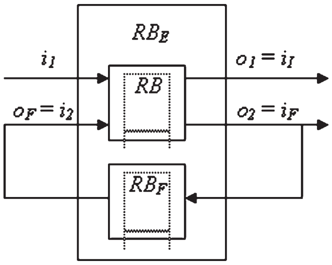

Fig. 8. Equivalent fuzzy system.

level/layer	layer 1	layer 2
level 1		o_I
level 2	$o_1 = i_I$ $o_2 = i_F$	$o_F = i_2$

(45)

The transformation of the initial into a final fuzzy network leads to the appearance of RB_I and RB_F in the feedforward part of the network as well as the appearance the four new connection variables i_I, o_I, i_F and o_F. In this context, Equation (44) shows that the final fuzzy network has two inputs (i_1 and i_2) and two outputs (o_I and o_F). In addition, Equation (45) shows that the output o1 from RB is the same as the input i_I to RB_I, the output o_2 from RB is the same as the input i_F to RB_F and the output o_F from RB_F is the same as the input i_2 to RB.

The final fuzzy network can be further transformed into an equivalent fuzzy system. The latter is shown in Fig. 8 and its rule base RB_E is derived from Equation (46).

$$RB_E = RB * (RB_I + RB_F) \tag{46}$$

The locations of the rule bases in this equivalent fuzzy system are presented by a grid structure with horizontal levels and vertical levels as follows:

level/layer	layer 1
level 1	RB_E, i_1, i_2, o_I, o_F

(47)

The connections between the rule bases in the equivalent fuzzy system are presented by a similar grid structure as follows:

level/layer	layer 1
level 1	o_I $o_F = i_1$

(48)

The rule base RB_E of the equivalent fuzzy system must satisfy the constraints imposed by the identity feedback whereby the linguistic values of the output o_F are fed back unchanged into the input i_2 for each of the fuzzy rules. As each of the inputs and outputs for this fuzzy system can take three linguistic values, the rule base RB_E will be as follows:

$$RB_E:\quad \begin{array}{cccccccccc} i_1, i_2/o_I, o_F & 11 & 12 & 13 & 21 & 22 & 23 & 31 & 32 & 33 \\ 11 & ? & 0 & 0 & ? & 0 & 0 & ? & 0 & 0 \\ 12 & 0 & ? & 0 & 0 & ? & 0 & 0 & ? & 0 \\ 13 & 0 & 0 & ? & 0 & 0 & ? & 0 & 0 & ? \\ 21 & ? & 0 & 0 & ? & 0 & 0 & ? & 0 & 0 \\ 22 & 0 & ? & 0 & 0 & ? & 0 & 0 & ? & 0 \\ 23 & 0 & 0 & ? & 0 & 0 & ? & 0 & 0 & ? \\ 31 & ? & 0 & 0 & ? & 0 & 0 & ? & 0 & 0 \\ 32 & 0 & ? & 0 & 0 & ? & 0 & 0 & ? & 0 \\ 33 & 0 & 0 & ? & 0 & 0 & ? & 0 & 0 & ? \end{array} \qquad (49)$$

The questions marks in the Boolean matrix from Equation (49) stand for elements that may be non-zero, i.e. they represent rules that may be present in the rule base RB_E. If at least one question mark in each row of this Boolean matrix stands for a non-zero element, then RB_E is complete. If only one question mark in each row of this Boolean matrix stands for a non-zero element, then RB_E is both complete and consistent. It is obvious from Equation (49) that the linguistic values of the output o_F are the same as the linguistic values of the input i_2 for each fuzzy rule that is present in the rule base RB_E.

Equation (46) can be simplified by decomposing the rule base of the equivalent fuzzy system RB_E in two parts - RB_{E1} and RB_{E2}, as shown by Equations (50)–(51). In this case, the rule bases RB_1 and RB_2 are such that $RB_1;RB_2 = RB$, i.e. their output merging results into the rule base RB_E.

$$RB_1 * RB_I = RB_{E1} \qquad (50)$$

$$RB_2 * RB_F = RB_{E2} \qquad (51)$$

It is obvious from Equations (50)–(51) that the feedback rule base RB_F appears only in the second equation. In this case, RB_{E1} can be derived by horizontal merging of RB_1 and RB_I, whereas RB_{E2} can be derived by horizontal merging of RB_2 and RB_F. Also, RB_E can be derived by output merging of RB_{E1} and RB_{E2}. Furthermore, RB_{E1} is the same as RB_1 while RB_{E2} will be as follows:

$$RB_{E2}:\quad \begin{array}{cccc} i_1, i_2/o_F & 11 & 12 & 13 \\ 11 & ? & 0 & 0 \\ 12 & 0 & ? & 0 \\ 13 & 0 & 0 & ? \\ 21 & ? & 0 & 0 \\ 22 & 0 & ? & 0 \\ 23 & 0 & 0 & ? \\ 31 & ? & 0 & 0 \\ 32 & 0 & ? & 0 \\ 33 & 0 & 0 & ? \end{array} \qquad (52)$$

It is obvious from Equation (52) that the linguistic values of the output o_F are the same as the linguistic values of the input i_2 for each fuzzy rule that is present in the rule base RB_{E2}.

The proposed complexity management methodology is evaluated comparatively in terms of model transparency for the initial fuzzy network and the equivalent fuzzy system. These two models are also compared to a standard fuzzy system model that has only one rule base in the feedforward loop with two inputs and two outputs.

The model transparency index used is given by Equation (53)

$$(s + z)/(m + n) \qquad (53)$$

where s is the number of subsystems, z is the number of connections, m is the number of inputs and n is the number of outputs. Equation (53) implies that the model transparency increases with the increase of the number of subsystems and the connections among them as well as with the decrease of the number of inputs and outputs.

Equation (53) is used for the derivation of the model transparency indexes for a standard fuzzy system, the initial fuzzy network model and the equivalent fuzzy system model. In this case, the indexes for the three models are calculated by Equations (54)–(56) as follows:

$$(1 + 0)/(2 + 2) = 1/4 = 0.25 \qquad (54)$$

$$(2 + 2)/(2 + 2) = 4/4 = 1.00 \qquad (55)$$

$$(3 + 3)/(2 + 2) = 6/4 = 1.50 \qquad (56)$$

Equations (54)–(56) show that the initial fuzzy network and the equivalent fuzzy system are between 4 and 6 times superior to the standard fuzzy system in terms of modelling transparency and ability to reflect qualitative complexity.

The proposed complexity management methodology can be extended to an arbitrarily complex fuzzy system with m inputs and n outputs. In this case, the rule base RBE of the equivalent fuzzy system can be presented by Equation (57):

$$
\begin{array}{lll}
\mathrm{RB_E}: & & \\
\mathrm{i}_1, \mathrm{i}_m/\mathrm{o}_1, \mathrm{on} & [\mathrm{v}_1(\mathrm{o}_1), \ldots, \mathrm{v}_1(\mathrm{o}_n)].. & [\mathrm{v}_k(\mathrm{o}_1), \ldots, \mathrm{v}_k(\mathrm{o}_n)] \\
[\mathrm{v}_1(\mathrm{i}_1), \ldots, \mathrm{v}_1(\mathrm{i}_m)] & \mathrm{r_{bE}}[\mathrm{v}_1(\mathrm{i}), \mathrm{v}_1(\mathrm{o})] \ldots & \mathrm{r_{bE}}[\mathrm{v}_1(\mathrm{i}), \mathrm{v}_k(\mathrm{o})] \\
\ldots\ldots\ldots\ldots\ldots\ldots.. & & \ldots\ldots\ldots\ldots\ldots\ldots \\
[\mathrm{v}_k(\mathrm{i}_1), \ldots, \mathrm{v}_k(\mathrm{i}_m)] & \mathrm{r_{bE}}[\mathrm{v}_k(\mathrm{i}), \mathrm{v}_1(\mathrm{o})] \ldots.. & \mathrm{r_{bE}}[\mathrm{v}_k(\mathrm{i}), \mathrm{v}_k(\mathrm{o})]
\end{array}
\tag{57}
$$

The notations used in Equation (57) have the following meaning: $\mathrm{i}_1, \ldots, \mathrm{i}_m$ are the inputs to the system, $\mathrm{o}_1, \ldots, \mathrm{o}_n$ are the outputs from the system, $\mathrm{v}_1, \ldots, \mathrm{v}_k$ are the linguistic values for each input and output and $\mathrm{r_{bE}}$ are the elements of the Boolean matrix that represent the rule base $\mathrm{RB_E}$ of the equivalent fuzzy system. For simplicity of notations, all inputs and outputs are assumed to take k linguistic values but Equation (57) can be extended easily to systems in which the number of linguistic values for individual inputs and outputs varies. For the same reason, all elements $\mathrm{rb_E}$ are presented in an aggregated form whereby the associated detailed form is given by Equations (58)–(61).

$$\mathrm{rb_E}[\mathrm{v}_1(\mathrm{i}), \mathrm{v}_1(\mathrm{o})] = \mathrm{rb_E}\{\, [\mathrm{v}_1(\mathrm{i}_1), \ldots, \mathrm{v}_1(\mathrm{i}_m)], [\mathrm{v}_1(\mathrm{o}_1), \ldots, \mathrm{v}_1(\mathrm{o}_n)]\} \tag{58}$$

$$\mathrm{rb_E}[\mathrm{v}_1(\mathrm{i}), \mathrm{v}_k(\mathrm{o})] = \mathrm{rb_E}\{\, [\mathrm{v}_1(\mathrm{i}_1), \ldots, \mathrm{v}_1(\mathrm{i}_m)], [\mathrm{v}_k(\mathrm{o}_1), \ldots, \mathrm{v}_k(\mathrm{o}_n)]\} \tag{59}$$

$$\mathrm{rb_E}[\mathrm{v}_k(\mathrm{i}), \mathrm{v}_1(\mathrm{o})] = \mathrm{rb_E}\{\, [\mathrm{v}_k(\mathrm{i}_1), \ldots, \mathrm{v}_k(\mathrm{i}_m)], [\mathrm{v}_1(\mathrm{o}_1), \ldots, \mathrm{v}_1(\mathrm{o}_n)]\} \tag{60}$$

$$\mathrm{rb_E}[\mathrm{v}_k(\mathrm{i}), \mathrm{v}_k(\mathrm{o})] = \mathrm{rb_E}\{\, [\mathrm{v}_k(\mathrm{i}_1), \ldots, \mathrm{v}_k(\mathrm{i}_m)], [\mathrm{v}_k(\mathrm{o}_1), \ldots, \mathrm{v}_k(\mathrm{o}_n)]\} \tag{61}$$

In this case, if an output o_s, $s = 1, \ldots, n$, is fed back unchanged as an input i_p, $p = 1, \ldots, m$, the corresponding element $\mathrm{rb_E}[\mathrm{v}(\mathrm{i}_1), \ldots, \mathrm{v}(\mathrm{i}_p), \ldots, \mathrm{v}(\mathrm{i}_m)], [\mathrm{v}(\mathrm{o}_1), \ldots, \mathrm{v}(\mathrm{o}_s), \ldots, \mathrm{v}(\mathrm{o}_n)]$ from the Boolean matrix for the rule base $\mathrm{RB_E}$ of the equivalent fuzzy system may exist, i.e. there may be a rule associated with it whereby $\mathrm{rb_E} = ?$. Therefore, the linguistic value for the output o_s is identical with the linguistic value for the input i_p, i.e. $\mathrm{v}(\mathrm{i}_p) = \mathrm{v}(\mathrm{o}_s)$. In the above general notations, the subscripts for the linguistic values of inputs and outputs have been left out as they do not contain any additional information.

7. Conclusion

The proposed complexity management methodology for feedback fuzzy rule based systems improves the transparency of the models used. This allows the structure of a fairly complex process in terms of interacting sub-processes to be reflected explicitly in the model. As a result, any complex process can be modelled by a fuzzy network in a more transparent way than by a fuzzy system due to the better visibility inside the process. This also leads to better understanding of the modelled process.

The proposed methodology is based on formal methods for presentation, manipulation and transformation of fuzzy rule bases. These methods make us of Boolean matrices are used for formal presentation of rule bases, binary merging operations for formal manipulation of rule bases and repetitive merging operations formal transformation of rule bases.

The proposed methodology is illustrated for feedback fuzzy networks with a fairly small number of sub-networks, connections, inputs and outputs. However, it can be easily extended to feedback fuzzy networks with an arbitrarily large number of sub-networks, connections, inputs and outputs. In this case, all binary merging operations presented can be applied repetitively in a flexible way using of the associativity property. This would lead only to a linear increase of the associated quantitative complexity.

Acknowledgment

The first author would like to thank the Faculty of Technology at the University of Portsmouth for the granted research sabbatical that made possible the writing of this paper.

References

[1] A. Gegov, *Distributed fuzzy control of multivariable systems*, Kluwer, Dordrecht, 1996.

[2] A. Gegov, *Complexity management in fuzzy systems*, Springer, Berlin, 2007.

[3] A. Gegov and M. Frank, Hierarchical fuzzy control of multivariable systems, *Fuzzy Sets and Systems* **72** (1995), 299–310.

[4] A. Gegov and M. Frank, Decomposition of multivariable systems for distributed fuzzy control, *Fuzzy Sets and Systems* **73** (1995), 329–340.

[5] B. Lazzerini and F. Marcelloni, Reducing computation overhead in MISO fuzzy systems, *Fuzzy Sets and Systems* **113** (2000), 485–496.

[6] C. Tao, Comments on "reduction of fuzzy rule base via singular value decomposition", *IEEE Transactions on Fuzzy Systems* **9**(4) (2001), 675–676.

[7] C. Xu, Linguistic decoupling control of fuzzy multivariable processes, *Fuzzy Sets and Systems* **44** (1991), 209–217.

[8] C. Xu and Y. Lu, Decoupling in fuzzy systems: A cascade compensation approach, *Fuzzy Sets and Systems* **29** (1989), 177–185.

[9] F. Wan, H. Shang, L. Wang and Y. Sun, How to determine the minimum number of fuzzy rules to achieve given accuracy: A computational geometric approach to SISO case, *Fuzzy Sets and Systems* **150** (2005), 199–209.

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

[11] H. Roubos and M. Setnes, Compact and transparent fuzzy models and classifiers through iterative complexity reduction, *IEEE Transactions on Fuzzy Systems* **9**(4) (2001), 516–524.

[12] J. Jang, C. Sun and E. Mizutani, *Neuro-fuzzy and soft computing: A computational approach to learning and machine intelligence*, Prentice Hall, Upper Saddle River, 1997.

[13] J. Mendel and Q. Liang, Comments on "combinatorial rule explosion eliminated by a fuzzy rule configuration", *IEEE Transactions on Fuzzy Systems* **7**(3) (1999), 369–371.

[14] J. Yan, M. Ryan and J. Power, *Using fuzzy logic*, Prentice Hall, New York, 1994.

[15] L. Wang, Analysis and design of hierarchical fuzzy systems, *IEEE Transactions on Fuzzy Systems* **7**(5) (1999), 617–624.

[16] M. Gupta, J. Kiszka and G. Trojan, Multivariable structure of fuzzy control systems, *IEEE Transactions on Systems, Man and Cybernetics* **16**(5) (1986), 638–655.

[17] M. Guven and K. Passino, Avoiding exponential parameter growth in fuzzy systems, *IEEE Transactions on Fuzzy Systems* **9**(1) (2001), 194–199.

[18] M. Jamshidi, *Large scale systems: Modelling, control and fuzzy logic*, Prentice Hall, Upper Saddle River, 1997.

[19] M. Joo and J. Lee, Universal approximation by hierarchical fuzzy system with constraints on the fuzzy rule, *Fuzzy Sets and Systems* **130** (2002), 175–188.

[20] M. Joo and J. Lee, A class of hierarchical fuzzy systems with constraints on the fuzzy rules, *IEEE Transactions on Fuzzy Systems* **13**(2) (2005), 194–203.

[21] M. Lee, H. Chung and F. Yu, Modelling of hierarchical fuzzy systems, *Fuzzy Sets and Systems* **138** (2003), 343–361.

[22] M. Negnevitsky, *Artificial intelligence: A guide to intelligent systems*, Pearson Education, Harlow, 2002.

[23] M. Setnes, R. Babuska and H. Verbruggen, Rule-based modelling: Precision and transparency, *IEEE Transactions on Systems, Man and Cybernetics* **28**(1) (1998), 165–169.

[24] N. Pal, V. Eluri and G. Mandal, Fuzzy logic approaches to structure preserving dimensionality reduction, *IEEE Transactions on Fuzzy Systems* **10**(3) (2002), 277–286.

[25] N. Xiong and L. Litz, Reduction of fuzzy control rules by means of premise learning – method and case study, *Fuzzy Sets and Systems* **132** (2002), 217–231.

[26] O. Huwendiek and W. Brockmann, Function approximation with decomposed fuzzy systems, *Fuzzy Sets and Systems* **101** (1999), 273–286.

[27] S. Chen, F. Yu and H. Chung, Decoupled fuzzy controller design with single-input fuzzy logic, *Fuzzy Sets and Systems* **129** (2002), 335–342.

[28] S. Mollov, Fuzzy Control of multiple-input multiple-output processes, PhD thesis, Delft University of Technology, 2002.

[29] T. Ross, *Fuzzy logic with engineering applications*, Wiley, Chichester, 2004.

[30] V. Lacrose, Complexity reduction of fuzzy controllers: Application to multivariable control, PhD thesis, Toulouse Laboratory for Systems Analysis and Architecture, 1997.

[31] W. Combs and J. Andrews, Combinatorial rule explosion eliminated by a fuzzy rule configuration, *IEEE Transactions on Fuzzy Systems* **6**(1) (1998), 1–11.

[32] Y. Kim, S. Ahn and W. Kwon, Computational complexity of general fuzzy logic control and its simplification for a loop controller, *Fuzzy Sets and Systems* **111** (2000), 215–224.

[33] Y. Yam, P. Baranyi and C. Yang, Reduction of fuzzy rule base via singular value decomposition, *IEEE Transactions on Fuzzy Systems* **7**(2) (1999), 120–132.