Filtration of Non-Monotonic Rules for Fuzzy Rule Base Compression

Alexander Gegov
University of Portsmouth, School of Computing
Buckingham Building, Portsmouth PO1 3HE, United Kingdom

Neelamugilan Gobalakrishnan
University of Portsmouth, School of Computing
Buckingham Building, Portsmouth PO1 3HE, United Kingdom
E-mail: muges_3106@hotmail.com

David Sanders
University of Portsmouth, School of Engineering
Anglesea Building, Portsmouth PO1 3DJ, United Kingdom
Email: david.sanders@port.ac.uk

Abstract

This paper proposes a rule base compression method for fuzzy systems. The method is based on filtration of rules with identical linguistic values for the output that are known as non-monotonic rules. The filtration removes the redundant computations in the fuzzy inference with respect to the crisp values of the inputs to the fuzzy system. The method identifies the redundant rules after fuzzification and removes them while preserving the defuzzified output from the fuzzy system for each simulation cycle. In comparison to the known rule base reduction methods, this rule base compression method does not compromise the solution and has better efficiency in terms of on-line computations. The method processes the rule base for a fuzzy system during simulation cycles by contracting it to a rule base of a smaller size at the start of each inference stage and then expanding it to its original size before the next fuzzification stage.

Keywords: Fuzzy systems, Complexity theory, Simulation, Data compression.

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 (see Refs.1-4). This exponential function has been used as a main indicator for the quantitative complexity of the associated fuzzy system. However, this is a fairly rough indicator because the quantitative complexity depends on the overall amount of operations

during fuzzification, inference and defuzzification. For example, a 4-input fuzzy system with 2 linguistic values per input has the same number of 16 rules as a 2-input fuzzy system with 4 linguistic values per input but the amount of operations in the first system is about twice as big as the one in the second system due to the twice bigger number of inputs in the rules.

There has been a growing interest recently in complexity issues of fuzzy systems (see Refs.5-8). 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 rule base 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 is aimed at removing less significant or merging similar linguistic values (see Refs.9-10). 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.

The second group of methods is aimed at removing less significant or merging similar inputs (see Refs.11-12). 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 is based on singular value decomposition of the matrix representing the crisp values of the output from a fuzzy system (see Refs.13-14). 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 is based on conversion of the intersection rule configuration of a fuzzy system into a union rule configuration with a smaller number of rules (see Refs.15-16). 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 converts a fuzzy system into spatially decomposed subsystems as a result of which the overall number of rules is reduced (see Refs.17-22). 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 has 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 rearranges the inputs in a fuzzy system in a way that leads to the reduction of the number of rules (see Refs.23-28). 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. Most of the above rule base reduction methods for fuzzy systems have serious drawbacks such as empirical nature, limited scope and approximate solutions. 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, the approximate solutions obtained by the methods in all groups 1-6 compromise to some extent their performance.

This paper addresses the above three drawbacks of rule base reduction methods by proposing a novel rule base compression method that is characterised by systematic nature, universal scope and precise solutions. Besides this, the method uses a more accurate indicator

for the quantitative complexity of fuzzy systems in terms of overall amount of on-line operations as opposed to the fairly rough indicator in terms of number of rules that is used by existing methods.

The remaining part of this paper is structured as follows. Section 2 provides some theoretical preliminaries for fuzzy systems. Section 3 introduces the rule base compression method. Section 4 illustrates the application of this method for a service centre operation case study. Section 5 evaluates the performance of the method in a quantitative and comparative context. Section 6 summarises the main advantages of the method and highlights future research directions.

2. Theoretical Preliminaries

A fuzzy system can be represented by the following rule base

$$If\ i_1\ is\ v_{i11}\ and\ ...\ and\ i_m\ is\ v_{im1}$$

$$then\ o_1\ is\ v_{o11}\ and\ ...\ and\ o_n\ is\ v_{on1}$$

$$.............................$$

$$If\ i_1\ is\ v_{i1r}\ and\ ...\ and\ i_m\ is\ v_{imr}$$

$$then\ o_1\ is\ v_{o1r}\ and\ ...\ and\ o_n\ is\ v_{onr} \qquad (1)$$

where m is the number of inputs, n is the number of outputs and r is the number of rules. 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.

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 (see Refs.29-30). If this number is a constant, the maximum number of rules is given by

$$r = w^m \qquad (2)$$

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

$$r = w_1\ ...\ w_m \qquad (3)$$

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 (see Ref.31). 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.*

The aim of the rule base compression approach in fuzzy systems is to remove all redundant operations during fuzzification, inference and defuzzification whereby the associated redundancy is caused by non-monotonic rules, i.e. rules with identical permutations of linguistic values of outputs. Such rules are often present in large quantities in fuzzy systems irrespective of whether the rule base has been created using expert knowledge or available data. In this case, the approach has to identify all redundant non-monotonic rules with respect to the crisp non-fuzzified values of inputs and remove these rules from the rule base without affecting the crisp defuzzified values of outputs. Therefore, this approach acts as a filter for redundant non-monotonic rules in the rule base that reduces the quantitative

complexity in fuzzy systems without compromising the solution.

In order to identify the redundancy in a rule base for a fuzzy system, it is necessary to consider the stages of fuzzification, inference and defuzzification. This consideration is presented further below whereby the inference stage includes three substages - application, implication and aggregation (see Refs.32-33). The considerations are for single-output systems but they can be easily extended to multiple-output systems whereby each output is considered separately and in relation to the same set of inputs.

The fuzzification stage in a fuzzy system maps the crisp value of each input to the system to a fuzzy value by a fuzzy membership degree. This degree can be obtained from the fuzzy membership functions for the inputs to the fuzzy system. The considerations presented are based on normal triangular or trapezoidal fuzzy membership functions that have a maximum equal to 1 and are commonly used in fuzzy systems due to their simplicity.

In this case, the fuzzy membership degree f_{ps} for an input is derived by

$$f_{ps} = 0, \text{ if } x_{ps} \leq a_{ps}$$

$$f_{ps} = (x_{ps} - a_{ps}) / (b_{ps} - a_{ps}), \text{ if } a_{ps} \leq x_{ps} \leq b_{ps}$$

$$f_{ps} = (c_{ps} - x_{ps}) / (c_{ps} - b_{ps}), \text{ if } b_{ps} \leq x_{ps} \leq c_{ps}$$

$$f_{ps} = 0, \text{ if } c_{ps} \leq x_{ps} \qquad (4)$$

where x_{ps}, $p=1,..,m$, $s=1,..,r$ is the continuous crisp value of the p-th input in the s-th rule of the fuzzy system and a_{ps}, b_{ps}, c_{ps} are the parameters of the triangular fuzzy membership function used for fuzzification of this input. In particular, a_{ps} is the point at which the membership function becomes greater than 0, b_{ps} is the point at which the membership function reaches its maximum at 1 and c_{ps} is the point at which the membership function becomes equal to 0 again. The symbol '/' denotes arithmetic division in Eq. (4) and all subsequent equations.

The application substage in a fuzzy system maps the fuzzy membership degrees of the inputs in each rule to a firing strength for this rule. The considerations presented here are based on rule bases with conjunctive terms in the 'if' part. Such rule bases are commonly used in fuzzy systems due to their ability to represent the simultaneous effect of all inputs.

In this case, the firing strength g_s for a rule is derived by

$$g_1 = min (f_{11},...,f_{m1})$$

$$...........................$$

$$g_r = min (f_{1r},...,f_{mr}) \qquad (5)$$

where f_{ps}, $p=1,..,m$, $s=1,..,r$ is the fuzzy membership degree for the p-th input in the s-th rule of the fuzzy system.

The implication substage in a fuzzy system maps the firing strength for each rule to a fuzzy membership function for the output in this rule. The considerations presented here are based on horizontal truncation that cuts the normal fuzzy triangular membership function for the output in each rule to a subnormal fuzzy trapezoidal membership function whose maximum is equal to the firing strength for this rule. This type of truncation is commonly used in fuzzy systems due to its simplicity.

In this case, the fuzzy membership function F_{sq} for an output is defined by

$$F_{sq} = \{ f_{1sq} * y_{1sq},...,f_{tsq} * y_{tsq} \} \qquad (6)$$

where f_{ksq}, $k=1,..,t$, $s=1,..,r$, $q=1,..,n$ is the fuzzy membership degree for the k-th element from a discrete variation range for the q-th output in the s-th rule of the fuzzy system, y_{ksq} is the associated element from this range and t is the number of such elements. The symbol '*' in Eq. (6) denotes binary association, i.e. the fuzzy membership degree f_{ksq} is associated with the element y_{ksq} from the discrete variation range for this output.

As the subscript k for f_{ksq} and y_{ksq} in Eq. (6) is not required further, this subscript will be omitted for simplicity. Therefore, the element y_{sq} is mapped to its fuzzy membership degrees f_{sq} by

$$f_{sq} = 0, \text{ if } y_{sq} \leq a_{sq}$$

$$f_{sq} = (y_{sq} - a_{sq}) / (b_{sq} - a_{sq}), \text{ if } a_{sq} \leq y_{sq} \leq b_{sq}$$

$$f_{sq} = g_s, \text{ if } b_{sq} \leq y_{sq} \leq c_{sq}$$

$$f_{sq} = (d_{sq} - y_{sq}) / (c_{sq} - b_{sq}), \text{ if } c_{sq} \leq y_{sq} \leq d_{sq}$$

$$f_{sq} = 0, \text{ if } d_{sq} \leq y_{sq} \qquad (7)$$

where y_{sq}, $s=1,..,r$, $q=1,..,n$ is the discrete crisp value of the q-th output in the s-th rule of the fuzzy system and a_{sq}, b_{sq}, c_{sq}, d_{sq} are the parameters of the trapezoidal

fuzzy membership function for this output. This function is obtained during the implication substage from the initial triangular fuzzy membership function for the output. In particular, a_{sq} is the point at which the membership function becomes greater than 0, b_{sq} is the point at which the membership function becomes equal to its maximum g_s, c_{sq} is the point at which the membership function becomes less than its maximum at g_s and d_{ps} is the point at which the membership function becomes equal to 0 again.

The aggregation substage in a fuzzy system maps the fuzzy membership functions for all rules to an aggregated fuzzy membership function representing the overall output for the rules. The considerations presented here are based on disjunctive rule bases. Such rule bases are commonly used in fuzzy systems due to their ability to represent the effect from the most dominant rule.

In this case, the aggregated fuzzy membership function F_q for an output is derived by

$$F_q = F_{1q} \, or \ldots or \, F_{rq} \qquad (8)$$

where F_{sq}, $s=1,..,r$, $q=1,..,n$ is the fuzzy membership function for the q-th output in the s-th rule of the fuzzy system. The symbol 'or' denotes a union operation that is applied to the fuzzy membership functions for the output in all rules. This operation is applied to the fuzzy membership degrees for all the elements from the discrete variation range for this output.

The defuzzification stage in a fuzzy system maps the aggregated fuzzy membership function for an output to a crisp value from the discrete variation range for this output. As this value is of a continuous type, the associated discrete variation range is mapped to its continuous counterpart. The considerations presented assume that the defuzzified value of the output is the centre of gravity for the aggregated fuzzy membership function for this output. This defuzzification method commonly used in fuzzy systems due to its applicability for any shape of aggregated fuzzy membership function for the output.

In this case, the defuzzified value D_q for an output is derived by

$$D_q = (f_{1q} . \, y_{1q} + \ldots + f_{tq} . \, y_{tq} \,) / (f_{1q} + \ldots + f_{tq}) \quad (9)$$

where f_{kq}, $k=1,..,t$, $q=1,..,n$ is the aggregated fuzzy membership degree for the k-th element from the discrete variation range for the q-th output of the fuzzy system and y_{kq} is the associated element from this range.

Eq. (9) represents f_{ksq} and y_{ksq} from Eq. (6) without the rule index s as the defuzzification stage is independent of the rules. Obviously, D_q can take any values within the continuous counterpart for the discrete variation range for this output. The symbols '.' and '+' in Eq. (9) denote arithmetic multiplication and addition, respectively.

3. Rule Base Compression Method

The method introduced here removes dynamically the redundancy in a non-monotonic rule base of a fuzzy system during the fuzzification, inference and defuzzification stages for each simulation cycle. The redundancy is expressed by the presence of non-monotonic rules and it is removed by filtering the redundant subset of these rules with the aim of making the rule base monotonic.

Filtration of non-monotonic rules in a fuzzy system is equivalent to representing a 'many-to-one' mapping as a 'one-to-one' mapping. A mathematical theorem for this representation is shown below. The proof of the theorem is based on Boolean logic laws and it is also shown further below.

Theorem 1. *A set of non-monotonic disjunctive rules in the form*

$$If \, (A_{11} \, and \, \ldots \, and \, A_{m1}) \, then \, C_q$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$If \, (A_{1z} \, and \, \ldots \, and \, A_{mz}) \, then \, C_q \qquad (10)$$

where $A_{pj} = (i_p \, is \, v_{ip,j})$, $p=1,..,m$, $j=1,..,z$ and $C_q = (o_q \, is \, v_{oq})$, $q=1,..,n$ are logical propositions describing the terms for the p-th input in the j-th rule and the terms for the q-th output in accordance with Eq. (1), q is a set label and z is the set cardinality, can be represented as a single rule in the form

$$[(A_{11} \, and \, \ldots \, and \, A_{m1}) \, imp \, C_q] \, or \, \ldots$$
$$or \, [(A_{1z} \, and \, \ldots \, and \, A_{mz}) \, imp \, C_q] \, . \qquad (11)$$

Proof. Eq. (10) represents a set of 'if-then' implications that can be rewritten as

$$(A_{11} \, and \, \ldots \, and \, A_{m1}) \, imp \, C_q$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$(A_{1z} \, and \, \ldots \, and \, A_{mz}) \, imp \, C_q \qquad (12)$$

where the 'if-then' notations are replaced by 'implication' operators.

The implications in Eq. (12) are also disjunctive rules that can be rewritten as

$$[(A_{11} \text{ and } ... \text{ and } A_{m1}) \text{ imp } C_q] \text{ or } ...$$
$$\text{ or } [(A_{1z} \text{ and } ... \text{ and } A_{mz}) \text{ imp } C_q] \quad (13)$$

where all rules are disjuncted together in one rule.

Using implication related laws, Eq. (13) can be rewritten as

$$[\text{not } (A_{11} \text{ and } ... \text{ and } A_{m1}) \text{ or } C_q] \text{ or } ...$$
$$\text{ or } [\text{not } (A_{1z} \text{ and } ... \text{ and } A_{mz}) \text{ or } C_q] \quad (14)$$

where the 'implication' operators are replaced by 'negation' and 'disjunction' operators.

Using commutative laws, Eq. (14) can be rewritten as

$$\{[\text{not } (A_{11} \text{ and } ... \text{ and } A_{m1})] \text{ or } ...$$
$$\text{ or } [\text{not } (A_{1z} \text{ and } ... \text{ and } A_{mz})]\}$$
$$\text{ or } (C_q \text{ or } ... \text{ or } C_q) \quad (15)$$

where the terms for the inputs are grouped separately from the terms for the output.

Using idempotent laws, Eq. (15) can be rewritten as

$$\{[\text{not } (A_{11} \text{ and } ... \text{ and } A_{m1})] \text{ or } ...$$
$$\text{ or } [\text{not } (A_{1z} \text{ and } ... \text{ and } A_{mz})]\} \text{ or } C_q \quad (16)$$

where only one of the z identical terms for the output is preserved.

Using De Morgan laws, Eq. (16) can be rewritten as

$$\text{not } [(A_{11} \text{ and } ... \text{ and } A_{m1})] \text{ and } ...$$
$$\text{ and } (A_{1z} \text{ and } ... \text{ and } A_{mz})] \text{ or } C_q \quad (17)$$

where the z groups of terms for the inputs are conjuncted together and negated as a whole.

Using again implication related laws, Eq. (17) can be rewritten as

$$[(A_{11} \text{ and } ... \text{ and } A_{m1})] \text{ and } ...$$
$$\text{ and } (A_{1z} \text{ and } ... \text{ and } A_{mz})] \text{ imp } C_q \quad (18)$$

where the 'negation' and 'disjunction' operator are replaced by an 'implication' operator.

Eq. (18) represents an implication that can be rewritten as Eq. (11) where the implication operator is replaced by an 'if-then' notation. So, this concludes the proof.

The 'many-to-one' mapping from Eq. (10) is represented equivalently as a 'one-to-one' mapping from Eq. (11). In this case, the z logical propositions $(A_{11} \text{ and } ... \text{ and } A_{m1})$ ... $(A_{1z} \text{ and } ... \text{ and } A_{mz})$ in the 'if' part of the non-monotonic set of rules in Eq. (10) are represented by a single logical proposition $[(A_{11} \text{ and } ...$ and $A_{m1}) \text{ and } ... \text{ and } (A_{1z} \text{ and } ... \text{ and } A_{mz})]$ in the 'if' part of a single equivalent rule in Eq. (11).

Theorem 1 can be trivially extended to an arbitrary number of sets of non-monotonic rules where each of these sets can be represented by a separate single equivalent rule. In this way, the non-monotonic rule base of a fuzzy system can be converted to an equivalent monotonic rule base of a smaller size that contains only the single equivalent rules.

Theorem 1 describes the theoretical foundations of the rule base compression method. The practical implementation of this method is given by the algorithm below.

Algorithm 1.

(i) Put all non-monotonic rules in separate groups sorted in an increasing order with respect to the linguistic values of the output, whereby the rules in each group have the same linguistic values of the output and different permutations of linguistic values of the inputs.

(ii) For each group of rules, find a single equivalent rule whose effect on the defuzzified output is the same as the effect of all rules in this group.

(iii) For each group of rules, keep the single equivalent rule and remove all other rules.

Algorithm 1 guarantees that there is only monotonic rules left in a fuzzy rule base after the completion of the filtration process. In this case, the number of monotonic rules is equal to the number of non-monotonic groups of rules and the number of different of linguistic values of the output. Therefore, the filtration process can be applied with a guaranteed success whereby the resulting compressed rule base is always monotonic.

Step 1 in Algorithm 1 can be applied off-line but steps 2-3 can only be applied on-line. This is because the single equivalent rule is a dominant rule that can be

found only after the completion of the fuzzification stage and the application substage. This dominancy is expressed by the rule with the maximal firing strength for each group as a result of which the effect of all other rules from the group on the defuzzified output is completely neutralised. When there is more than one such dominant rule in a group, i.e. two or more rules with maximal firing strength for the group, any of these rules can be selected arbitrarily as a single equivalent rule.

As Algorithm 1 is applied right after the completion of the fuzzification stage and the application substage for each simulation cycle, the remaining substages of implication and aggregation as well as the defuzzification stage can make use of the compressed monotonic rule base. However, for the next simulation cycle, the original non-monotonic rule base must be used again in the fuzzification stage and the application substage.

Theorem 1 and Algorithm 1 are presented above for a single-output fuzzy system but they can be trivially extended to a multiple-output fuzzy system with an arbitrary number of outputs. In this case, the multiple-output fuzzy system from Eq. (1) can be represented by the following n equivalent single-output fuzzy systems

$$If\ i_1\ is\ v_{i11}\ and\ ...\ and\ i_m\ is\ v_{im1}\ then\ o_q\ is\ v_{q11}$$

$$...$$

$$If\ i_1\ is\ v_{i1r}\ and\ ...\ and\ i_m\ is\ v_{imr}\ then\ o_q\ is\ v_{q1r}$$

$$q=1,..,n \qquad (19)$$

whereby all considerations from the theorem and the algorithm can be applied repetitively to each of these systems.

4. Simulation Results

The rule base compression method is applied to a case study for service centre operation. A fuzzy system for service centre operation is described by the inputs i_1, i_2, i_3 and the output o_1 where i_1 is the *repair utilisation factor* (r), i_2 is the *number of servers* (s), i_3 is the *mean delay of service* (d) and o_1 is the *number of spare parts* (p). In this case, i_1 can take the three linguistic values *low* (L), *medium* (M) and *high* (H), i_2 can take the three linguistic values *small* (S), *medium* (M) and *large* (L), i_3 can take the three linguistic values *very short* (VS),

short (S) and *medium* (M), whereas o_1 can take the seven linguistic values *very small* (VS), *small* (S), *rather small* (RS), *medium* (M) *rather large* (RL), *large* (L) and *very large* (VL).

By making the substitutions L=1, M=2, H=3 for i_1, S=1, M=2, L=3 for i_2, VS=1, S=2, M=3 for i_3, as well as the substitutions VS=1, S=2, RS=3, M=4, RL=5, L=6, VL=7 for o_1, the rule base for this fuzzy system can be presented, as shown in Table 1. Then, by applying step 1 from Algorithm 1, this initial rule base can be rearranged into a sorted rule base by putting all non-monotonic rules in separate groups, as shown in Table 2. The empty rows in Tables 1-2 are used for the purpose of visual separation of rules from different groups.

Table 1. Initial rule base.

Rule number	First input	Second input	Third input	Output
1	1	1	1	1
2	1	1	2	1
3	1	1	3	1
4	1	2	1	1
5	1	2	2	1
6	1	2	3	1
7	1	3	1	2
8	1	3	2	2
9	1	3	3	1
10	2	1	1	2
11	2	1	2	1
12	2	1	3	1
13	2	2	1	3
14	2	2	2	2
15	2	2	3	1
16	2	3	1	4
17	2	3	2	3
18	2	3	3	2
19	3	1	1	7
20	3	1	2	6
21	3	1	3	4
22	3	2	1	4
23	3	2	2	4
24	3	2	3	2
25	3	3	1	5
26	3	3	2	4
27	3	3	3	3

Table 2. Sorted rule base.

Rule number	First input	Second input	Third input	Output
1	1	1	1	1
2	1	1	2	1
3	1	1	3	1
4	1	2	1	1
5	1	2	2	1
6	1	2	3	1
9	1	3	3	1
11	2	1	2	1
12	2	1	3	1
15	2	2	3	1
7	1	3	1	2
8	1	3	2	2
10	2	1	1	2
14	2	2	2	2
18	2	3	3	2
24	3	2	3	2
13	2	2	1	3
17	2	3	2	3
27	3	3	3	3
16	2	3	1	4
21	3	1	3	4
22	3	2	1	4
23	3	2	2	4
26	3	3	2	4
25	3	3	1	5
20	3	1	2	6
19	3	1	1	7

Further on, by applying steps 2-3 from Algorithm 1, the sorted rule base can be transformed into a compressed rule base, as shown in Table 3. This rule base contains only the single equivalent rule from each of the seven non-monotonic groups of rules from the sorted rule base. The determination of these single equivalent rules is described below.

Table 3. Compressed rule base.

Rule number	First input	Second input	Third input	Output
4	1	2	1	1
7	1	3	1	2
13	2	2	1	3
16	2	3	1	4
25	3	3	1	5
20	3	1	2	6
19	3	1	1	7

The following paragraphs consider in detail the fuzzification, inference and defuzzification stages for the initial rule base first and then for the compressed rule base. The aim of these considerations is to show the behavioural equivalence of the two rule bases, i.e. that the defuzzified output is the same for any crisp values of the inputs. In the current example, the three inputs and the output can take crisp values within the normalised discrete variation range [0, 1] where the particular values taken here are $r = 0.5$, $s = 0.65$ and $d = 0.2$.

At the fuzzification stage for the initial rule base, it is necessary to consider all possible linguistic values for each input. In this context, it should be specified how the fuzzy membership degree for a particular linguistic value of a given input can be obtained. This specification is based on the parameters of the fuzzy membership functions used for fuzzification of the inputs, as shown below.

The fuzzy membership degree f_r^L for the linguistic value *low* of the input *repair utilisation factor* can be obtained by the formula

$$f_r^L = 0, \; if \; r < a_r^L$$

$$f_r^L = 1, \; if \; a_r^L \le r \le b_r^L$$

$$f_r^L = 1, \; if \; b_r^L \le r \le c_r^L$$

$$f_r^L = (\, d_r^L - r\,)\,/\,(\, d_r^L - c_r^L\,), \; if \; c_r^L \le r \le d_r^L$$

$$f_r^L = 0, \; if \; d_r^L \le r \qquad (20)$$

where a_r^L, b_r^L, c_r^L, d_r^L are the parameters of the associated trapezoidal fuzzy membership function whose values are given by

$$a_r^L = 0.0, \; b_r^L = 0.0, \; c_r^L = 0.4, \; d_r^L = 0.6. \qquad (21)$$

Eq. (20) differs slightly from Eq. (7). In particular, the 'equality' part of the sign in the first line of Eq. (7) is removed form Eq. (20) to account for the vertical left shoulder of the membership function. Also, the arithmetic division in the second line of Eq. (7) is removed from Eq. (20) to avoid division by zero as a result of the equality of the parameters a_r^L and b_r^L.

The fuzzy membership degree f_r^M for the linguistic value *medium* of the input *repair utilisation factor* can be obtained by the formula

$$f_r^M = 0, \text{ if } r \leq a_r^M$$

$$f_r^M = (r - a_r^M) / (b_r^M - a_r^M), \text{ if } a_r^M \leq r \leq b_r^M$$

$$f_r^M = (c_r^M - r) / (c_r^M - b_r^M), \text{ if } b_r^M \leq r \leq c_r^M$$

$$f_r^M = 0, \text{ if } c_r^M \leq r \qquad (22)$$

where a_r^M, b_r^M, c_r^M are the parameters of the associated triangular fuzzy membership function whose values are given by

$$a_r^M = 0.4, \ b_r^M = 0.6, \ c_r^M = 0.8. \qquad (23)$$

The fuzzy membership degree f_r^H for the linguistic value *high* of the *input repair utilisation factor* can be obtained by the formula

$$f_r^H = 0, \text{ if } r \leq a_r^H$$

$$f_r^H = (r - a_r^H) / (b_r^H - a_r^H), \text{ if } a_r^H \leq r \leq b_r^H$$

$$f_r^H = 1, \text{ if } b_r^H \leq r \leq c_r^H$$

$$f_r^H = 1, \text{ if } c_r^H \leq r \leq d_r^H$$

$$f_r^H = 0, \text{ if } d_r^H < r \qquad (24)$$

where a_r^H, b_r^H, c_r^H, d_r^H are the parameters of the associated trapezoidal fuzzy membership function whose values are given by

$$a_r^H = 0.6, \ b_r^H = 0.8, \ c_r^H = 1.0, \ d_r^H = 1.0. \qquad (25)$$

Eq. (24) differs slightly from Eq. (7). In particular, the arithmetic division in the fourth line of Eq. (7) is removed from Eq. (24) to avoid division by zero as a result of the equality of the parameters c_r^H and d_r^H. Also, the 'equality' part of the sign in the fifth line of Eq. (7) is removed from Eq. (24) to account for the vertical right shoulder of the membership function.

The parameters of the fuzzy membership functions for the first input to the fuzzy system are summarised in Table 4.

The fuzzy membership degree f_s^S for the linguistic value *small* of the input *number of servers* can be obtained by the formula

Table 4. Membership function parameters for first input.

Linguistic value / Input	Repair utilisation factor
Low	[0.0 0.0 0.4 0.6]
Medium	[0.4 0.6 0.8]
High	[0.6 0.8 1.0 1.0]

$$f_s^S = 0, \text{ if } s < a_s^S$$

$$f_s^S = 1, \text{ if } a_s^S \leq s \leq b_s^S$$

$$f_s^S = 1, \text{ if } b_s^S \leq s \leq c_s^S$$

$$f_s^S = (d_s^S - s) / (d_s^S - c_s^S), \text{ if } c_s^S \leq s \leq d_s^S$$

$$f_s^S = 0, \text{ if } d_s^S \leq s \qquad (26)$$

where a_s^S, b_s^S, c_s^S, d_s^S are the parameters of the associated trapezoidal fuzzy membership function whose values are given by

$$a_s^S = 0.0, \ b_s^S = 0.0, \ c_s^S = 0.15, \ d_s^S = 0.35. \qquad (27)$$

Eq. (26) differs slightly from Eq. (7). In particular, the 'equality' part of the sign in the first line of Eq. (7) is removed form Eq. (26) to account for the vertical left shoulder of the membership function. Also, the arithmetic division in the second line of Eq. (7) is removed from Eq. (26) to avoid division by zero as a result of the equality of the parameters a_s^S and b_s^S.

The fuzzy membership degree f_s^M for the linguistic value *medium* of the input *number of servers* can be obtained by the formula

$$f_s^M = 0, \text{ if } s \leq a_s^M$$

$$f_s^M = (s - a_s^M) / (b_s^M - a_s^M), \text{ if } a_s^M \leq s \leq b_s^M$$

$$f_s^M = (c_s^M - s) / (c_s^M - b_s^M), \text{ if } b_s^M \leq s \leq c_s^M$$

$$f_s^M = 0, \text{ if } c_s^M \leq s \qquad (28)$$

where a_s^M, b_s^M, c_s^M are the parameters of the associated triangular fuzzy membership function whose values are given by

A.Gegov et al.

$$a_s^M = 0.3, \, b_s^M = 0.5, \, c_s^M = 0.7. \qquad (29)$$

The fuzzy membership degree f_s^H for the linguistic value *high* of the input *number of servers* can be obtained by the formula

$$f_s^H = 0, \, if \, s \leq a_s^H$$

$$f_s^H = (\, s - a_s^H \,) / (\, b_s^H - a_s^H \,), \, if \, a_s^H \leq s \leq b_s^H$$

$$f_s^H = 1, \, if \, b_s^H \leq s \leq c_s^H$$

$$f_s^H = 1, \, if \, c_s^H \leq s \leq d_s^H$$

$$f_s^H = 0, \, if \, d_s^H < s \qquad (30)$$

where a_s^H, b_s^H, c_s^H, d_s^H are the parameters of the associated trapezoidal fuzzy membership function whose values are given by

$$a_s^H = 0.6, \, b_s^H = 0.8, \, c_s^H = 1.0, \, d_s^H = 1.0. \qquad (31)$$

Eq. (30) differs slightly from Eq. (7). In particular, the arithmetic division in the fourth line of Eq. (7) is removed from Eq. (30) to avoid division by zero as a result of the equality of the parameters c_s^H and d_s^H. Also, the 'equality' part of the sign in the fifth line of Eq. (7) is removed from Eq. (30) to account for the vertical right shoulder of the membership function.

The parameters of the fuzzy membership functions for the second input to the fuzzy system are summarised in Table 5.

Table 5. Membership function parameters for second input.

Linguistic value / Input	Number of servers
Small	[0.0 0.0 0.15 0.35]
Medium	[0.3 0.5 0.7]
Large	[0.6 0.8 1.0 1.0]

The fuzzy membership degree f_d^{VS} for the linguistic value *very short* of the input *mean delay of service* can be obtained by the formula

$$f_d^{VS} = 0, \, if \, d < a_d^{VS}$$

$$f_d^{VS} = 1, \, if \, a_d^{VS} \leq d \leq b_d^{VS}$$

$$f_d^{VS} = 1, \, if \, b_d^{VS} \leq d \leq c_d^{VS}$$

$$f_d^{VS} = (\, d_d^{VS} - d \,) / (\, d_d^{VS} - c_d^{VS} \,), \, if \, c_d^{VS} \leq d \leq d_d^{VS}$$

$$f_d^{VS} = 0, \, if \, d_d^{VS} \leq d \qquad (32)$$

where a_d^{VS}, b_d^{VS}, c_d^{VS}, d_d^{VS} are the parameters of the associated trapezoidal fuzzy membership function whose values are given by

$$a_d^{VS} = 0.0, \, b_d^{VS} = 0.0, \, c_d^{VS} = 0.1, \, d_d^{VS} = 0.3. \qquad (33)$$

Eq. (26) differs slightly from Eq. (7). In particular, the 'equality' part of the sign in the first line of Eq. (7) is removed form Eq. (26) to account for the vertical left shoulder of the membership function. Also, the arithmetic division in the second line of Eq. (7) is removed from Eq. (26) to avoid division by zero as a result of the equality of the parameters a_s^S and b_s^S.

The fuzzy membership degree f_d^S for the linguistic value *short* of the input *mean delay of service* can be obtained by the formula

$$f_d^S = 0, \, if \, d \leq a_d^S$$

$$f_d^S = (\, d - a_d^S \,) / (\, b_d^S - a_d^S \,), \, if \, a_d^S \leq d \leq b_d^S$$

$$f_d^S = (\, c_d^S - d \,) / (\, c_d^S - b_d^S \,), \, if \, b_d^S \leq d \leq c_d^S$$

$$f_d^S = 0, \, if \, c_d^S \leq d \qquad (34)$$

where a_d^S, b_d^S, c_d^S are the parameters of the associated triangular fuzzy membership function whose values are given by

$$a_d^S = 0.1, \, b_d^S = 0.3, \, c_d^S = 0.5. \qquad (35)$$

The fuzzy membership degree f_d^M for the linguistic value *medium* of the input *mean delay of service* can be obtained by the formula

$$f_d^M = 0, \, if \, d \leq a_d^M$$

$$f_d^M = (\, d - a_d^M \,) / (\, b_d^M - a_d^M \,), \, if \, a_d^M \leq d \leq b_d^M$$

$$f_d^M = 1, \, if \, b_d^M \leq d \leq c_d^M$$

$$f_d^M = 1, \, if \, c_d^M \leq d \leq d_d^M$$

$$f_d^M = 0, \, if \, d_d^M < d \qquad (36)$$

where $a_d{}^M$, $b_d{}^M$, $c_d{}^M$, $d_d{}^M$ are the parameters of the associated trapezoidal fuzzy membership function whose values are given by

$$a_d{}^M = 0.4, b_d{}^M = 0.6, c_d{}^M = 1.0, d_d{}^M = 1.0. \quad (37)$$

Eq. (36) differs slightly from Eq. (7). In particular, the arithmetic division in the fourth line of Eq. (7) is removed from Eq. (36) to avoid division by zero as a result of the equality of the parameters $c_d{}^M$ and $d_d{}^M$. Also, the 'equality' part of the sign in the fifth line of Eq. (7) is removed from Eq. (36) to account for the vertical right shoulder of the membership function.

The parameters of the fuzzy membership functions for the third input to the fuzzy system are summarised in Table 6.

Table 6. Membership function parameters for third input.

Linguistic value / Input	Mean delay of service
Very short	[0.0 0.0 0.1 0.3]
Short	[0.1 0.3 0.5]
Medium	[0.4 0.6 1.0 1.0]

At the application substage of the inference stage for the initial rule base, it is necessary to find the firing strength for each rule. For this purpose, the fuzzy membership degrees for the inputs obtained during the fuzzification stage are processed, as shown by Eqs. (38)-(64).

$$g_1{}^{VS} = min \, (f_r{}^L, \, f_s{}^S, \, f_d{}^{VS}) =$$

$$min \, (0.5, \, 0, \, 0.25) = 0 \quad (38)$$

$$g_2{}^{VS} = min \, (f_r{}^L, \, f_s{}^S, \, f_d{}^S) =$$

$$min \, (0.5, \, 0, \, 0.75) = 0 \quad (39)$$

$$g_3{}^{VS} = min \, (f_r{}^L, \, f_s{}^S, \, f_d{}^M) =$$

$$min \, (0.5, \, 0, \, 0) = 0 \quad (40)$$

$$g_4{}^{VS} = min \, (f_r{}^L, \, f_s{}^M, \, f_d{}^{VS}) =$$

$$min \, (0.5, \, 0.25, \, 0.25) = 0.25 \quad (41)$$

$$g_5{}^{VS} = min \, (f_r{}^L, \, f_s{}^M, \, f_d{}^S) =$$

$$min \, (0.5, \, 0.25, \, 0.75) = 0.25 \quad (42)$$

$$g_6{}^{VS} = min \, (f_r{}^L, \, f_s{}^M, \, f_d{}^M) =$$

$$min \, (0.5, \, 0.25, \, 0) = 0 \quad (43)$$

$$g_7{}^{S} = min \, (f_r{}^L, \, f_s{}^L, \, f_d{}^{VS}) =$$

$$min \, (0.5, \, 0.25, \, 0.25) = 0.25 \quad (44)$$

$$g_8{}^{S} = min \, (f_r{}^L, \, f_s{}^L, \, f_d{}^S) =$$

$$min \, (0.5, \, 0.25, \, 0.75) = 0.25 \quad (45)$$

$$g_9{}^{VS} = min \, (f_r{}^L, \, f_s{}^L, \, f_d{}^M) =$$

$$min \, (0.5, \, 0.25, \, 0) = 0 \quad (46)$$

$$g_{10}{}^{S} = min \, (f_r{}^M, \, f_s{}^S, \, f_d{}^{VS}) =$$

$$min \, (0.5, \, 0, \, 0.25) = 0 \quad (47)$$

$$g_{11}{}^{VS} = min \, (f_r{}^M, \, f_s{}^S, \, f_d{}^S) =$$

$$min \, (0.5, \, 0, \, 0.75) = 0 \quad (48)$$

$$g_{12}{}^{VS} = min \, (f_r{}^M, \, f_s{}^S, \, f_d{}^M) =$$

$$min \, (0.5, \, 0, \, 0) = 0 \quad (49)$$

$$g_{13}{}^{RS} = min \, (f_r{}^M, \, f_s{}^M, \, f_d{}^{VS}) =$$

$$min \, (0.5, \, 0.25, \, 0.25) = 0.25 \quad (50)$$

$$g_{14}{}^{S} = min \, (f_r{}^M, \, f_s{}^M, \, f_d{}^S) =$$

$$min \, (0.5, \, 0.25, \, 0.75) = 0.25 \quad (51)$$

$$g_{15}{}^{VS} = min \, (f_r{}^M, \, f_s{}^M, \, f_d{}^M) =$$

$$min \, (0.5, \, 0.25, \, 0) = 0 \quad (52)$$

$$g_{16}{}^{M} = min \, (f_r{}^M, \, f_s{}^L, \, f_d{}^{VS}) =$$

$$min \, (0.5, \, 0.25, \, 0.25) = 0.25 \quad (53)$$

$$g_{17}{}^{RS} = min \, (f_r{}^M, \, f_s{}^L, \, f_d{}^S) =$$

$$min \, (0.5, \, 0.25, \, 0.75) = 0.25 \quad (54)$$

$$g_{18}{}^{S} = min \, (f_r{}^M, \, f_s{}^L, \, f_d{}^M) =$$

$$min\ (0.5,\ 0.25,\ 0) = 0 \tag{55}$$

$$g_{19}{}^{VL} = min\ (\ f_r{}^H,\ f_s{}^S,\ f_d{}^{VS}\ ) =$$

$$min\ (0,\ 0,\ 0.25) = 0 \tag{56}$$

$$g_{20}{}^{L} = min\ (\ f_r{}^H,\ f_s{}^S,\ f_d{}^{S}\ ) =$$

$$min\ (0,\ 0,\ 0.75) = 0 \tag{57}$$

$$g_{21}{}^{M} = min\ (\ f_r{}^H,\ f_s{}^S,\ f_d{}^{M}\ ) =$$

$$min\ (0,\ 0,\ 0) = 0 \tag{58}$$

$$g_{22}{}^{M} = min\ (\ f_r{}^H,\ f_s{}^M,\ f_d{}^{VS}\ ) =$$

$$min\ (0,\ 0.25,\ 0.25) = 0 \tag{59}$$

$$g_{23}{}^{M} = min\ (\ f_r{}^H,\ f_s{}^M,\ f_d{}^{S}\ ) =$$

$$min\ (0,\ 0.25,\ 0.75) = 0 \tag{60}$$

$$g_{24}{}^{S} = min\ (\ f_r{}^H,\ f_s{}^M,\ f_d{}^{M}\ ) =$$

$$min\ (0,\ 0.25,\ 0) = 0 \tag{61}$$

$$g_{25}{}^{RL} = min\ (\ f_r{}^H,\ f_s{}^L,\ f_d{}^{VS}\ ) =$$

$$min\ (0,\ 0.25,\ 0.25) = 0 \tag{62}$$

$$g_{26}{}^{M} = min\ (\ f_r{}^H,\ f_s{}^L,\ f_d{}^{S}\ ) =$$

$$min\ (0,\ 0.25,\ 0.75) = 0 \tag{63}$$

$$g_{27}{}^{RS} = min\ (\ f_r{}^H,\ f_s{}^L,\ f_d{}^{M}\ ) =$$

$$min\ (0,\ 0.25,\ 0) = 0. \tag{64}$$

After the application substage of the inference stage, it is necessary put all the rules from the initial rule base in non-monotonic groups in accordance with step 1 of Algorithm 1. Then, it is necessary to identify the single equivalent rule for each group and remove all other rules from the initial rule base in accordance with steps 2-3 of Algorithm 1, as shown below.

The firing strength for the rules in each group is given by Eqs. (65)-(71).

$$Group\ VS:\ g_1{}^{VS} = 0,\ g_2{}^{VS} = 0,\ g_3{}^{VS} = 0,$$

$$g_4{}^{VS} = 0.25,\ g_5{}^{VS} = 0.25,\ g_6{}^{VS} = 0,\ g_9{}^{VS} = 0,$$

$$g_{11}{}^{VS} = 0,\ g_{12}{}^{VS} = 0,\ g_{15}{}^{VS} = 0 \tag{65}$$

$$Group\ S:\ g_7{}^{S} = 0.25,\ g_8{}^{S} = 0.25,\ g_{10}{}^{S} = 0,$$

$$g_{14}{}^{S} = 0.25,\ g_{18}{}^{S} = 0,\ g_{24}{}^{S} = 0 \tag{66}$$

$$Group\ RS:\ g_{13}{}^{RS} = 0.25,$$

$$g_{17}{}^{RS} = 0.25,\ g_{27}{}^{RS} = 0 \tag{67}$$

$$Group\ M:\ g_{16}{}^{M} = 0.25,\ g_{21}{}^{M} = 0,$$

$$g_{22}{}^{M} = 0,\ g_{23}{}^{M} = 0.25,\ g_{26}{}^{M} = 0 \tag{68}$$

$$Group\ RL:\ g_{25}{}^{RL} = 0 \tag{69}$$

$$Group\ L:\ g_{20}{}^{L} = 0 \tag{70}$$

$$Group\ VL:\ g_{19}{}^{VL} = 0. \tag{71}$$

The single equivalent rules for the above groups are given by Eqs. (72)-(78).

$$Group\ VS:\ g_4{}^{VS} = 0.2 \tag{72}$$

$$Group\ S:\ g_7{}^{S} = 0.25 \tag{73}$$

$$Group\ RS:\ g_{13}{}^{RS} = 0.25 \tag{74}$$

$$Group\ M:\ g_{16}{}^{M} = 0.25 \tag{75}$$

$$Group\ RL:\ g_{25}{}^{RL} = 0 \tag{76}$$

$$Group\ L:\ g_{20}{}^{L} = 0 \tag{77}$$

$$Group\ VL:\ g_{19}{}^{VL} = 0. \tag{78}$$

Therefore, the compressed rule base will contain only the seven single equivalent rules presented above. In this case, the implication and aggregation substages of the inference stage will be quite simple, as shown by Eqs. (79)-(85) and Eq. (86), respectively. These substages take into account the parameters of the fuzzy membership functions for the output from the fuzzy system that are summarised in Table 7.

$$F4 = \{0.25/0,\ 0.25/0.1,\ 0.25/0.2,\ 0/0.3,\ 0/0.4,\ 0/0.5,$$
$$0/0.6,\ 0.7,\ 0/0.8,\ 0/0.9,\ 0/1\} \tag{79}$$

Table 7. Membership function parameters for output.

Linguistic value / Output	Number of spare parts
Very small	[0.0 0.0 0.1 0.3]
Small	[0.0 0.2 0.4]
Rather small	[0.25 0.35 0.45]
Medium	[0.3 0.5 0.7]
Rather large	[0.55 0.65 0.75]
Large	[0.6 0.8 1.0]
Very large	[0.7 0.9 1.0 1.0]

$$F7 = \{0/0, 0.25/0.1, 0.25/0.2, 0.25/0.3, 0/0.4, 0/0.5, \\ 0/0.6, 0/0.7, 0/0.8, 0/0.9, 0/1\} \quad (80)$$

$$F13 = \{0/0, 0/0.1, 0/0.2, 0.25/0.3, 0.25/0.4, 0/0.5, 0/0.6, \\ 0/0.7, 0/0.8, 0/0.9, 0/1\} \quad (81)$$

$$F16 = \{0/0, 0/0.1, 0/0.2, 0/0.3, 0.25/0.4, 0.25/0.5, \\ 0.25/0.6, 0/0.7, 0/0.8, 0/0.9, 0/1\} \quad (82)$$

$$F25 = \{0/0, 0/0.1, 0/0.2, 0/0.3, 0/0.4, 0/0.5, 0/0.6, 0/0.7, \\ 0/0.8, 0/0.9, 0/1\} \quad (83)$$

$$F20 = \{0/0, 0/0.1, 0/0.2, 0/0.3, 0/0.4, 0/0.5, 0/0.6, 0/0.7, \\ 0/0.8, 0/0.9, 0/1\} \quad (84)$$

$$F19 = \{0/0, 0/0.1, 0/0.2, 0/0.3, 0/0.4, 0/0.5, 0/0.6, 0/0.7, \\ 0/0.8, 0/0.9, 0/1\} \quad (85)$$

$$F = F4 \ or \ F7 \ or \ F13 \ orF16 \ or \ F25 \ or \ F20 \ or \ F19 =$$

$$\{0.25/0, 0.25/0.1, 0.25/0.2, 0.25/0.3, 0.25/0.4,$$

$$0.25/0.5, 0.25/0.6, 0/0.7, 0/0.8, 0/0.9, 0/1\}. \quad (86)$$

The aggregated fuzzy membership function *F* for the output of the compressed rule base is the same as the one for the output of the initial rule base. This is because the single equivalent rules for each non-monotonic group override the effect of the other rules during the implication and aggregation substages of the inference stage. Therefore, the seven rules of the compressed rule base will lead to the same defuzzified output as the twenty rules of the initial rule base.

Like the fuzzification stage and the application substage of the inference stage, the defuzzification stage for the compressed rule base is the same as the one for the initial rule base. However, the efficiency gained by the removed redundant operations during the implication and aggregation substages of the inference stage significantly outweighs the complexity added by

the selection process for single equivalent rules, as shown further below. For terminological clarity, the initial rule base and the compressed rule base will be associated with a conventional fuzzy system and a filtered fuzzy system, respectively.

The behavioural equivalence of the conventional fuzzy system and the filtered fuzzy system is illustrated analytically in Tables 8-10 for three different crisp values of the first input, i.e. 0, 0.5 and 1. These tables show the numerical crisp values of the output surfaces for the two systems, which are identical for each permutation of crisp values of the inputs. These numerical values are derived for 3x3 equally spaced points within the discrete variation ranges for the inputs in analogy with the rule bases for the two systems which have 3x3 possible permutations of linguistic values of the inputs.

Table 8. Values for 3x3 point output surfaces (first input=0).

Point number / Component	Second input	Third input	Conventional system output	Filtered system output
1	0.00	0.00	0.10	0.10
2	0.00	0.50	0.12	0.12
3	0.00	1.00	0.10	0.10
4	0.50	0.00	0.10	0.10
5	0.50	0.50	0.12	0.12
6	0.50	1.00	0.10	0.10
7	1.00	0.00	0.20	0.20
8	1.00	0.50	0.12	0.12
9	1.00	1.00	0.10	0.10

Table 9. Values for 3x3 point output surfaces (first input=0.5).

Point number / Component	Second input	Third input	Conventional system output	Filtered system output
1	0.00	0.00	0.17	0.17
2	0.00	0.50	0.12	0.12
3	0.00	1.00	0.12	0.12
4	0.50	0.00	0.20	0.20
5	0.50	0.50	0.12	0.12
6	0.50	1.00	0.12	0.12
7	1.00	0.00	0.35	0.35
8	1.00	0.50	0.17	0.17
9	1.00	1.00	0.17	0.17

Table 10. Values for 3x3 point output surfaces (first input=1).

Point number / Component	Second input	Third input	Conventional system output	Filtered system output
1	0.00	0.00	0.89	0.89
2	0.00	0.50	0.50	0.50
3	0.00	1.00	0.50	0.50
4	0.50	0.00	0.50	0.50
5	0.50	0.50	0.20	0.20
6	0.50	1.00	0.20	0.20
7	1.00	0.00	0.65	0.65
8	1.00	0.50	0.35	0.35
9	1.00	1.00	0.35	0.35

The dominant single equivalent rules for all permutations of crisp values of the inputs for the filtered fuzzy system are presented in Tables 11-13 for the above three crisp values of the first input. These tables show the corresponding rule numbers from the conventional fuzzy system in accordance with Table 1.

Table 11. Dominant rules for filtered system (first input=0).

Point number / Component	Second input	Third input	Dominant rule numbers
1	0.00	0.00	1, 7, 13, 16, 25, 20, 19
2	0.00	0.50	1, 7, 13, 16, 25, 20, 19
3	0.00	1.00	3, 7, 13, 16, 25, 20, 19
4	0.50	0.00	4, 7, 13, 16, 25, 20, 19
5	0.50	0.50	6, 7, 13, 16, 25, 20, 19
6	0.50	1.00	6, 7, 13, 16, 25, 20, 19
7	1.00	0.00	1, 7, 13, 16, 25, 20, 19
8	1.00	0.50	9, 7, 13, 16, 25, 20, 19
9	1.00	1.00	9, 7, 13, 16, 25, 20, 19

Table 12. Dominant rules for filtered system (first input=0.5).

Point number / Component	Second input	Third input	Dominant rule numbers
1	0.00	0.00	1, 10, 13, 16, 25, 20, 19
2	0.00	0.50	12, 7, 13, 16, 25, 20, 19
3	0.00	1.00	12, 7, 13, 16, 25, 20, 19
4	0.50	0.00	4, 7, 13, 16, 25, 20, 19
5	0.50	0.50	6, 7, 13, 16, 25, 20, 19
6	0.50	1.00	6, 7, 13, 16, 25, 20, 19
7	1.00	0.00	1, 7, 13, 16, 25, 20, 19
8	1.00	0.50	9, 18, 13, 16, 25, 20, 19
9	1.00	1.00	9, 18, 13, 16, 25, 20, 19

Table 13. Dominant rules for filtered system (first input=1).

Point number / Component	Second input	Third input	Dominant rule numbers
1	0.00	0.00	1, 7, 13, 16, 25, 20, 19
2	0.00	0.50	1, 7, 13, 21, 25, 20, 19
3	0.00	1.00	1, 7, 13, 21, 25, 20, 19
4	0.50	0.00	1, 7, 13, 22, 25, 20, 19
5	0.50	0.50	1, 24, 13, 16, 25, 20, 19
6	0.50	1.00	1, 24, 13, 16, 25, 20, 19
7	1.00	0.00	1, 7, 13, 16, 25, 20, 19
8	1.00	0.50	1, 7, 27, 16, 25, 20, 19
9	1.00	1.00	9, 7, 27, 16, 25, 20, 19

The behavioural equivalence of the conventional fuzzy system and the compressed fuzzy system is illustrated also graphically in Figs. 1-6 for the above three different crisp values of the first input. These figures show the output surfaces for the two systems, which are identical for each permutation of crisp values of the inputs. The underlying numerical crisp values for these output surfaces are derived for 30x30 equally spaced points within the discrete variation ranges for the inputs. In this case, the precise numerical values and dominant rules are not shown as in Tables 8-13 due to the high space requirements for representing such as a large number of points.
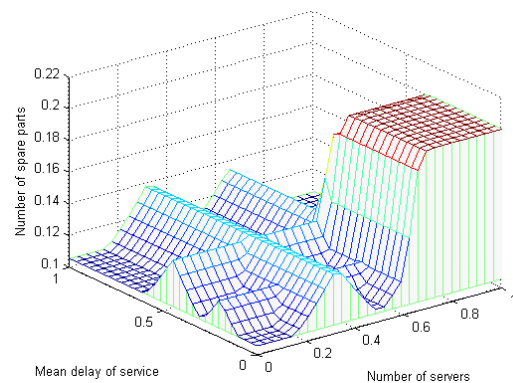

Fig.1. Output surface with 30x30 points for conventional system (first input = 0).
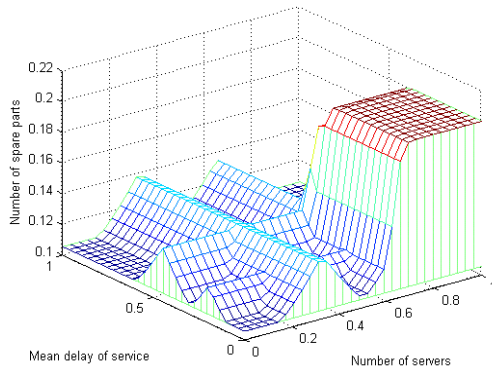
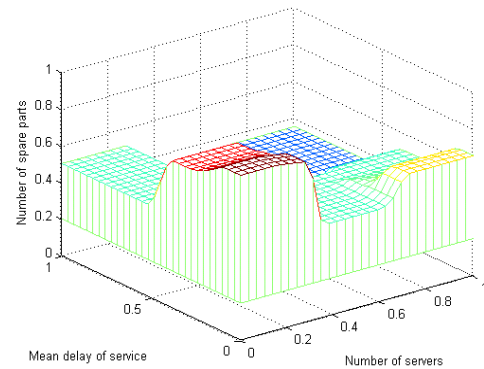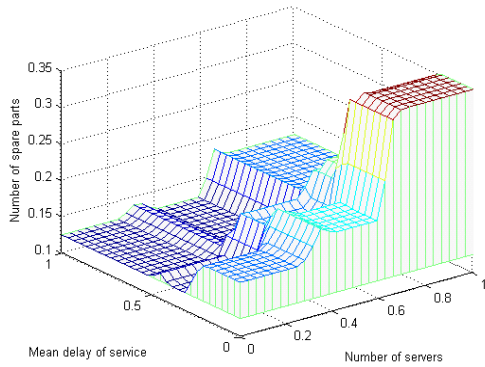Fig.2. Output surface with 30x30 points for filtered system (first input = 0).

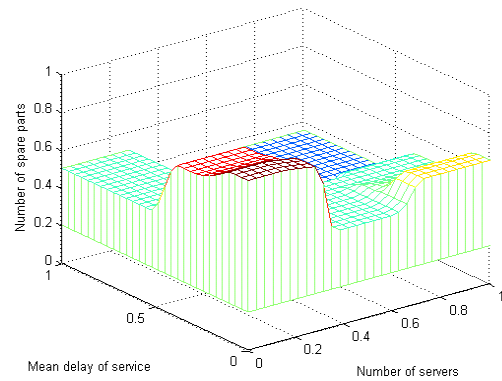Fig.3. Output surface with 30x30 points for conventional system (first input = 0.5).

Fig.4. Output surface with 30x30 points for filtered system (first input = 0.5).

Fig.5. Output surface with 30x30 points for conventional system (first input = 1).

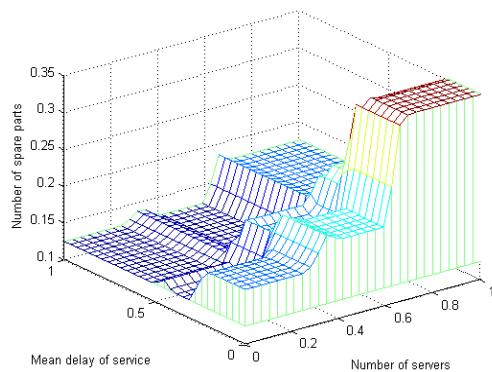Fig.6. Output surface with 30x30 points for filtered system (first input = 1).

5. Performance Evaluation

The proposed rule base compression method is evaluated comparatively in terms of its computational complexity, which reflects the level of quantitative complexity in general. In particular, a filtered fuzzy system that uses this rule base compression method is compared to a conventional fuzzy system that operates with all its rules and a hierarchical fuzzy system that uses the most popular rule base reduction method. The presented evaluation is two-fold – for the service centre operation case study and in a general context. The purpose of the evaluation is to demonstrate the efficiency of a filtered system in relation to a conventional system and a hierarchical system whereby unlike the hierarchical system whose accuracy is usually worse than the one of the conventional system, the

accuracy of the filtered system is always equal to the one of the conventional system.

The parameters used for evaluating the computational complexity of the three systems are: m – number of inputs, w – number of linguistic values per input, n – number of outputs, t – number of elements in the discrete variation range for the output, h – number of simulation cycles. For the case study where the number of linguistic values per inputs varies, the associated complexity evaluation formulas are modified accordingly to reflect this.

The amount of elementary operations (EO) for the separate stages and substages in a fuzzy system is determined by the overall number of arithmetic and logical operations such as addition, subtraction, multiplication, division and comparison. For simplicity, it is assumed that each of these operations is equal to one computational time unit. Therefore, each stage and substage in the fuzzy system is quantified by means of the overall number of these units.

The next step is to find the overall number of EO for the three systems under consideration, i.e. the conventional system (CS), the hierarchical system (HS) and the filtered system (FS). This is done by summing the number of operations for the stages of fuzzification (FU) and defuzzification (DE) as well the substages of application (AP), implication (IM) and aggregation (AG) of the inference stage for each of the three systems, as shown by Eqs. (87)-(89).

$$EO^{CS} =$$

$$EO_{FU}{}^{CS} + EO_{AP}{}^{CS} + EO_{IM}{}^{CS} + EO_{AG}{}^{CS} + EO_{DE}{}^{CS} =$$

$$(6 . m . w . n . h) + (w + m - 2) . w^{m-1} . n . h +$$

$$(7 . w^{m} . t . n . h) + (w^{m} - 1) . t . n . h +$$

$$(3 . t - 1) . n . h =$$

$$[(8 . t + 1) . w^{m} + (m - 2) . w^{m-1} +$$

$$6 . m . w + 2 . t - 1] . n . h \qquad (87)$$

$$EO^{HS} =$$

$$EO_{FU}{}^{HS} + EO_{AP}{}^{HS} + EO_{IM}{}^{HS} + EO_{AG}{}^{HS} + EO_{DE}{}^{HS} =$$

$$(m - 1) . (12 . w . n . h) + (m - 1) . w^{2} . n . h +$$

$$(m - 1) . (7 . w^{2} . t . n . h) +$$

$$(m - 1) . (w^{2} - 1) . t . n . h +$$

$$(m - 1) . (3 . t - 1) . n . h =$$

$$(m - 1) . [(8 . t + 1) . w^{2} + 12 . w +$$

$$2 . t - 1] . n . h \qquad (88)$$

$$EO^{FS} =$$

$$EO_{FU}{}^{FS} + EO_{AP}{}^{FS} + EO_{IM}{}^{FS} +$$

$$EO_{AG}{}^{FS} + EO_{DE}{}^{FS} + EO_{CO}{}^{FS} =$$

$$(6 . m . w . n . h) + (w + m - 2) . w^{m-1} . n . h +$$

$$(7 . w . t . n . h) + (w - 1) . t . n . h +$$

$$(3 . t - 1) . n . h + (w^{m} - w) . n . h =$$

$$[2 . w^{m} + (m - 2) . w^{m-1} + (6 . m - 1) . w +$$

$$(8 . w + 2) . t - 1] . n . h . \qquad (89)$$

Eqs. (87)-(89) are used for comparative evaluation of the computational complexity of the three fuzzy systems in relation to the case study. As the three inputs there can take 3 linguistic values each, the value of w is also set equal to 3. The overall number of EO for one simulation step is 2487 for the CS and 1716 for the HS whereas this number is 399 for the FS. This shows that the FS is more than 6 times more efficient than the CS and more than 4 times more efficient than the HS.

As Eqs. (87)-(89) represent general formulas, they can also be used in a wider context, i.e. outside the scope of specific case studies. By varying some of the parameters in these formulas, it would be possible to see the dependency between the computational complexity for each system and the values of its parameters.

As the increase of the parameters n and h would always lead to a similar linear increase of the computational complexity for the three systems, it would be reasonable to keep these parameters fixed. As far as the other parameters m, w and t are concerned, it

would be necessary to vary them because their increase would usually lead to a different exponential increase of the computational complexity of the three systems. Therefore, the parameters will have the following fixed values and discrete variation ranges

$$n = 1; h = 1; m = 2,3,4; w = 3,5,7,9,11;$$

$$t = 7,13,19,25,31. \tag{90}$$

In order to reduce the number of possible permutations of values for m, w and t, it would be reasonable to assume that the variation of the parameters w and t is fixed by the formula

$$t = 3 . w - 2. \tag{91}$$

Eqs. (90)-(91) define a fairly wide scope for evaluating the computational complexity of fuzzy systems. In particular, most fuzzy systems are initially considered for one simulation step of one output before more simulation steps of this output or simulations of other outputs are considered. Also, fuzzy systems are usually represented with up to several inputs because the number of rules for more inputs would be almost unmanageable. In addition, the inputs and outputs of fuzzy systems are often described by an odd number of linguistic values as this provides better coverage of the associated discrete variation ranges. And finally, the number of elements in the discrete variation ranges for an output is often between 2 and 3 times greater than the number of linguistic values for this output, whose number is often close or equal to the number of linguistic values that each input can take.

Table 14 presents the results from a general comparative evaluation of the computational complexity of the CS, the HS and the FS. This evaluation is made using Eqs. (87)-(89) and in accordance with the assumptions made for the values of all relevant parameters, as shown by Eqs. (90)-(91).

Table 14 shows that the FS is more efficient for all considered permutations of values for the relevant parameters. As in terms of accuracy the HS is only an approximation of the CS whereas the FS is equivalent to the CS, it is obvious that the FS outperforms significantly the CS and the HS. This superiority is valid for fuzzy systems whose number of rules is up to 14000 and above.

Table 14. Computational complexity of three fuzzy systems.

Number of rules / On-line operations	Conventional system	Hierarchical system	Filtered system
$3^2 = 9$	562	562	232
$3^3 = 27$	1,615	1,124	295
$3^4 = 81$	4,756	1,686	466
$5^2 = 25$	2,710	2,710	650
$5^3 = 125$	13,265	5,420	905
$5^4 = 625$	66,020	8,130	2,160
$7^2 = 49$	7,618	7,618	1,276
$7^3 = 343$	52,691	15,236	1,955
$7^4 = 2401$	368,244	22,854	6,750
$9^2 = 81$	16,438	16,438	2,110
$9^3 = 729$	146,821	32,876	3,541
$9^4 = 6,561$	1,320,484	49,314	16,636
$11^2 = 121$	30,322	30,322	3,152
$11^3 = 1,331$	331,779	60,644	5,759
$11^4 = 14,641$	3,648,596	90,966	34,986

6. Conclusion

The proposed rule base compression method for fuzzy systems reduces significantly the number of rules during the implication and aggregation substages of the inference stage. This translates into a substantial reduction of the associated computational complexity in terms of the overall amount of on-line operations. In addition, the solution is not compromised because the defuzzified output of the filtered system is the same as the output of the conventional system. Therefore, the method is suitable for both time-critical and safety-critical applications.

The proposed method can be extended easily for other types of fuzzification, inference and defuzzification. For example, instead of triangular membership functions for fuzzification, it is possible to use trapezoidal ones or others. Also, instead of truncation type of implication, it is possible to use scaling type or others. And finally, instead of centre of gravity type of defuzzification, it is possible to use weighted average type or others. In all these cases, only small adjustments have to be made to ensure that the defuzzified value of the output for the filtered system is the same as the one for the conventional system.

The proposed method is illustrated for single simulation cycles of single-output fuzzy systems but it can be easily extended to multiple simulation cycles and multiple-output fuzzy systems. In this case, all procedures presented can be applied in exactly the same way to each simulation cycle for each output. This

would lead only to a linear increase of the associated computational complexity, which would be proportional to the number of simulation cycles and the number of outputs.

The proposed method is illustrated for a two-input fuzzy system whereby each of the inputs can take up to a few linguistic values. However, it is also applicable to fuzzy systems with an arbitrary number of inputs that can take an arbitrary number of linguistic values. In this context, the method is evaluated for fuzzy systems with up to four inputs and up to eleven linguistic values per input, i.e. fuzzy systems with up to 14000 rules and above. The comparison shows the superiority of the method to two widely used other methods. In particular, the proposed method has the same accuracy as the method used in a conventional system but much better efficiency while at the same time it has fairly better accuracy and quite better efficiency than the method used in a hierarchical system.

The proposed method is illustrated for fuzzy systems with a single rule base but it can be also used for fuzzy systems with multiple rule bases such as fuzzy networks. In this case, the fuzzy network can be transformed into a linguistically equivalent single rule base system by means of rule base merging operations and the method can then be applied in exactly the same way to this single rule base system.

The proposed method is illustrated for non-evolving fuzzy systems. However, it can be also used for evolving fuzzy systems whereby the rule base can be updated at the start of each simulation cycle. In this case, the updated rule base can be processed in the same way as the initial rule base of the conventional system during the fuzzification stage and the application substage of the inference stage.

7. Acknowledgement

References

1. M. Guven and K. Passino, Avoiding exponential parameter growth in fuzzy systems, *IEEE T Fuzzy Syst* **9**(1) (2001) 194-199.
2. Y. Kim Y, S. Ahn and W. Kwon, Computational complexity of general fuzzy logic control and its simplification for a loop controller, *Fuzzy Set Syst* **111** (2000) 215-224.
3. B. Lazzerini and F. Marcelloni, Reducing computation overhead in MISO fuzzy systems, *Fuzzy Set Syst* **113** (2000) 485-496.
4. N. Pal, V. Eluri and G. Mandal, Fuzzy logic approaches to structure preserving dimensionality reduction, *IEEE T Fuzzy Syst* **10**(3) (2002) 277-286.
5. A. Gegov, *Distributed Fuzzy Control of Multivariable Systems* (Kluwer, Dordrecht, 1996).
6. M. Gupta, J. Kiszka and G. Trojan, Multivariable structure of fuzzy control systems, *IEEE T Syst Man Cyb* **16**(5) (1986) 638-655.
7. F. Wan, H. Shang, L. Wang L and Y. Sun, How to determine the minimum number of fuzzy rules to achieve given accuracy: a computational geometric approach to SISO case, *Fuzzy Set Syst* **150** (2005) 199-209.
8. N. Xiong and L. Litz, Reduction of fuzzy control rules by means of premise learning – method and case study, *Fuzzy Set Syst* **132** (2002) 217-231.
9. H. Roubos and M. Setnes, Compact and transparent fuzzy models and classifiers through iterative complexity reduction, *IEEE T Fuzzy Syst* **9**(4) (2001) 516-524.
10. M. Setnes, R. Babuska and H. Verbruggen, Rule-based modelling: precision and transparency, *IEEE T Syst Man Cyb* **28**(1) (1998) 165-169.
11. V. Lacrose, *Complexity Reduction of Fuzzy Controllers: Application to Multivariable Control* (PhD Thesis, Toulouse Laboratory for Systems Analysis and Architecture, 1997).
12. M. Jamshidi, *Large Scale Systems: Modelling, Control and Fuzzy Logic* (Prentice Hall, Upper Saddle River, 1997).
13. Y. Yam, P. Baranyi and C. Yang, Reduction of fuzzy rule base via singular value decomposition, *IEEE T Fuzzy Syst* **7**(2) (1999) 120-132.
14. C. Tao, Comments on reduction of fuzzy rule base via singular value decomposition, *IEEE T Fuzzy Syst* **9**(4) (2001) 675-676.
15. W. Combs and J. Andrews, Combinatorial rule explosion eliminated by a fuzzy rule configuration, *IEEE T Fuzzy Syst* **6**(1) (1998) 1-11.
16. J. Mendel and Q. Liang, Comments on combinatorial rule explosion eliminated by a fuzzy rule configuration, *IEEE T Fuzzy Syst* **7**(3) (1999) 369-371.
17. S. Chen, F. Yu and H. Chung, Decoupled fuzzy controller design with single-input fuzzy logic, *Fuzzy Set Syst* **129** (2002) 335-342.
18. A. Gegov and M. Frank, Hierarchical fuzzy control of multivariable systems, *Fuzzy Set Syst* **72** (1995) 299-310.
19. S. Mollov, *Fuzzy Control of Multiple-input Multiple-output Processes* (PhD Thesis, Delft University of Technology, 2002).
20. A. Gegov and M. Frank, Decomposition of multivariable systems for distributed fuzzy control, *Fuzzy Set Syst* **73** (1995) 329-340.

21. C. Xu, Linguistic decoupling control of fuzzy multivariable processes, *Fuzzy Set Syst* **44** (1991) 209-217.

22. C. Xu and Y. Lu, Decoupling in fuzzy systems: a cascade compensation approach, *Fuzzy Set Syst* **29** (1989) 177-185.

23. O. Huwendiek and W. Brockmann, Function approximation with decomposed fuzzy systems, *Fuzzy Set Syst* **101** (1999) 273-286.

24. M. Joo and J. Lee, Universal approximation by hierarchical fuzzy system with constraints on the fuzzy rule, *Fuzzy Set Syst* **130** (2002) 175-188.

25. M. Joo and J. Lee, A class of hierarchical fuzzy systems with constraints on the fuzzy rules, *IEEE T Fuzzy Syst* **13**(2) (2005) 194-203.

26. M. Lee, H. Chung and F. Yu, Modelling of hierarchical fuzzy systems, *Fuzzy Set Syst* **138** (2003) 343-361.

27. G. Raju, J. Zhou and R. Kisner, Hierarchical fuzzy control, Int J Control **54**(5) (1991) 1201-1216.

28. L. Wang, Analysis and design of hierarchical fuzzy systems, *IEEE T Fuzzy Syst* **7**(5) (1999) 617-624.

29. 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).

30. J. Yan, M. Ryan and J. Power, *Using Fuzzy Logic* (Prentice Hall, New York, 1994).

31. A. Gegov, *Complexity Management in Fuzzy Systems* (Springer, Berlin, 2007).

32. M. Negnevitsky, *Artificial Intelligence: A Guide to Intelligent Systems* (Pearson Education, Harlow, 2002).

33. T. Ross, *Fuzzy Logic with Engineering Applications* (Wiley, Chichester, 2004).