Journal of Intelligent & Fuzzy Systems xx (20xx) x–xx
DOI:10.3233/IFS-141418
IOS Press

Rule base simplification in fuzzy systems by aggregation of inconsistent rules

Alexander Gegov[a,*], Farzad Arabikhan[a] and David Sanders[b]
[a] *School of Computing, University of Portsmouth, Buckingham Building, Portsmouth, UK*
[b] *School of Engineering, University of Portsmouth, Anglesea Building, Portsmouth, UK*

Abstract. This paper proposes a rule base simplification method for fuzzy systems. The method is based on aggregation of rules with different linguistic values of the output for identical permutations of linguistic values of the inputs which are known as inconsistent rules. The simplification removes the redundancy in the fuzzy rule base by replacing each group of inconsistent rules with a single equivalent rule. The simulation results from a transportation demand management case study show that the aggregated fuzzy system with the consistent rule base approximates better the given data than the original fuzzy system with the inconsistent rule base. The main advantage of the proposed method over other methods is that it does not require any refinement of the rule base using additional data sets or expert knowledge. In this context, the method is quite suitable for applications where rule base refinement is unacceptable due to time constraints or impossible due to lack of additional data or knowledge.

Keywords: Fuzzy systems, complexity theory, data simplification, transportation demand management, control systems

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. 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 [2, 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 rule base reduction methods as they reduce the number of rules by reducing the number of inputs or the number of

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

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.

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, 20, 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.

Most of the above rule base reduction methods for fuzzy systems have 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.

This paper addresses the above two drawbacks of rule base reduction methods by proposing a novel rule base simplification method that is characterised by systematic nature and universal scope. Besides this, the method leads to solutions which approximate closely the data.

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 simplification method. Section 4 illustrates the application of this method to several examples with inconsistent rule bases. Section 5 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:

$$
\begin{aligned}
&If\ i_1\ is\ v_{11}\ and\ \ldots and\ i_m\ is\ v_{im1}\ then\ o_1\ is\ v_{o11}\\
&and \ldots and\ o_n\ is\ v_{on1}\\
&\qquad\qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&if\ i_1\ is\ v_{i1r}\ and\ \ldots and\ i_m\ is\ v_{imr}\ then\ o_1\ is\ v_{o1r}\\
&and \ldots and\ o_n\ is\ v_{onr} \qquad (1)
\end{aligned}
$$

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, $P = 1, \ldots m$ represents the p-th input, v_{ips}, $P = 1$,

$\ldots m, s = 1, \ldots r$ is the linguistic value of the p-th input in the s-th rule, $o_q, q = 1, \ldots n$ represents the q-th output and $v_{oqs}, q = 1, \ldots n, s = 1, \ldots r$ 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 [12, 14]. If this number is a constant, the maximum number of rules is given by:

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

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

Fuzzy rule bases have some important properties [1]. 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 propositions. These propositions 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 does not hold and vice versa.

Proposition 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.

Proposition 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.

Proposition 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.

Proposition 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 proposed rule base simplification approach in fuzzy systems is to remove the redundancy in the rule base that is caused by inconsistent rules, i.e. rules with different linguistic values of the output for identical permutations of linguistic values of the inputs rules. Inconsistent rules may be present in fuzzy systems irrespective of whether the rule base has been created using data sets or expert knowledge. In this case, the approach identifies all inconsistent rules and removes these rules from the rule base by aggregating them into a single equivalent rules. Therefore, this approach acts as an aggregator for inconsistent rules in the rule base that reduces the quantitative complexity in fuzzy systems. The approach is particularly useful in situations where rule base refinement is not suitable due to time constraints or impossible due to lack of additional data or knowledge.

In order to follow the proposed approach, 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 [22, 29]. 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. To facilitate the software implementation of the theoretical results, the notation used in the presentation of the proposed approach is similar to the one from the Matlab Fuzzy Logic Toolbox.

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

$$\begin{aligned}
&f_{ps} = 0,\ if\ x_{ps} \le a_{ps}\\
&f_{ps} = (x_{ps} - a_{ps})/(b_{ps} - a_{ps}),\ if\ a_{ps} \le x_{ps} \le b_{ps}\\
&f_{ps} = (c_{ps} - x_{ps})/(c_{ps} - b_{ps}),\ if\ b_{ps} \le x_{ps} \le c_{ps}\\
&f_{ps} = 0,\ if\ c_{ps} \le x_{ps}
\end{aligned} \tag{4}$$

where $x_{ps},\ P = 1, \ldots,\ m, s = 1, \ldots,\ r$ 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 Equation (4) and all subsequent equations.

The application sub-stage 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 in a definitive way the simultaneous effect of all inputs as opposed to rule bases with disjunctive terms that are more ambiguous and therefore not so common.

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

$$\begin{aligned} &g_1 = \min\,(f_{11}, \ldots, f_{m1}) \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad (5) \\ &g_r = \min\,(f_{1r}, \ldots, \ f_{mr}) \end{aligned}$$

where $f_{ps},\ P = 1, \ldots,\ m,\ s = 1, \ldots,\ r$ 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 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

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

where $f_{ksq}, k = 1, \ldots,\ t, s = 1, \ldots,\ r, q = 1, \ldots,\ n$ 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 Equation (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 Equation (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:

$$\begin{aligned} &f_{sq} = 0,\ if\ y_{sq} \le a_{sq} \\ &f_{sq} = (y_{sq} - a_{sq})/(b_{sq} - a_{sq}),\ if\ a_{sq} \le y_{sq} \le b_{sq} \\ &f_{sq} = g_s,\ if\ b_{sq} \le y_{sq} \le c_{sq} \\ &f_{sq} = (d_{sq} - y_{sq})/(c_{sq} - b_{sq}),\ if\ c_{sq} \le y_{sq} \le d_{sq} \\ &f_{sq} = 0,\ if\ d_{sq} \le y_{sq} \qquad (7) \end{aligned}$$

where $y_{sq},\ s = 1, \ldots, r,\ q = 1, \ldots, n$ 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 flexibly the individual effect from the most influential rule as opposed to conjunctive rule bases that are more restrictive and therefore not so common.

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 $F_{sq}, s = 1, \ldots r, q = 1, \ldots n$ 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:

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

where f_{kq}, $k = 1, \ldots, t$, $q = 1, \ldots, n$ 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. Equation (9) represents f_{ksq} and y_{ksq} from Equation (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 Equation (9) denote arithmetic multiplication and addition, respectively.

3. Rule base simplification method

The method introduced here removes the redundancy in an inconsistent 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 inconsistent rules and it is removed by aggregating the redundant subset of these rules with the aim of making the rule base consistent.

Aggregation of inconsistent rules in a fuzzy system is equivalent to representing a 'one-to-many' mapping as a 'one-to-one' mapping. A mathematical theorem for this representation is shown below. The proof of the theorem is shown further below and it is based on Boolean logic laws. In this proof, the operations of negation, conjunction, disjunction and implication are all defined in the context of classical binary logic.

Theorem 1: *A set of inconsistent disjunctive rules in the form*

$$\begin{gathered} If\ (A_{1s}\ and \ldots and\ A_{ms})\ then\ C_{q1} \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ If (A_{1s}\ and\ \ldots and\ A_{ms})\ then\ C_{qz} \end{gathered} \tag{10}$$

where $A_{ps} = (i_p \text{ is } v_{ip,s})$, $P = 1, \ldots, m$, $j = 1, \ldots, z$ and $C_{qz} = (o_q \text{ is } v_{oq,z})$, $q = 1, \ldots, n$ 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 Equation (1), s is a set label and z is the set cardinality, can be represented as a single rule in the form:

$$\begin{gathered} If (A_{1s}\ and \ldots and\ A_{ms}) \\ then\ (C_{q1}\ or \ldots or\ C_{qz}) \end{gathered} \tag{11}$$

Proof 1: Equation (10) represents a set of 'if-then' implications that can be rewritten as:

$$\begin{gathered} (A_{1s}\ and \ldots and\ A_{ms})\ imp\ C_{q1} \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ (A_{1s}\ and \ldots and\ A_{ms})\ imp\ C_{qz} \end{gathered} \tag{12}$$

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

The implications in Equation (12) are also disjunctive rules that can be rewritten as:

$$\begin{gathered} [(A_{1s}\ and \ldots and\ A_{ms})\ impC_{q1}]\ or \ldots or \\ [(A_{1s}\ and \ldots and\ A_{ms})\ imp\ C_{qz}] \end{gathered} \tag{13}$$

where all rules are disjuncted together in one rule.

Using implication related laws, Equation (13) can be rewritten as:

$$\begin{gathered} [not\ (A_{1s}\ and \ldots and\ A_{ms})\ or\ C_{q1}]\ or \ldots or \\ [not\ (A_{1s}\ and \ldots andA_{ms})\ or\ C_{qz}] \end{gathered} \tag{14}$$

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

Using commutative laws, Equation (14) can be rewritten as:

$$\begin{gathered} \{[not\ (A_{1s}\ and \ldots and\ A_{ms})]\ or \ldots or \\ [not\ (A_{1s}\ and \ldots and\ A_{ms})]\}\ or \\ (C_{q_1}\ or \ldots or\ C_{qz}) \end{gathered} \tag{15}$$

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

Using idempotent laws, Equation (15) can be rewritten as:

$$\begin{gathered} [not\ (A_{11}\ and \ldots andA_{ms})]\ or \\ (C_{q1}\ or \ldots or\ C_{qz}) \end{gathered} \tag{16}$$

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

Using again implication related laws, Equation (16) can be rewritten as:

$$(A_{11}\ and \ldots and\ A_{ms})\ imp$$
$$(C_{q1}\ or \ldots\ orC_{qz}) \tag{17}$$

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

Equation (17) represents an implication that can be rewritten as Equation (11) where the implication operator is replaced by an 'if-then' notation. So, this concludes the proof.

The 'one-to-many' mapping from Equation (10) is represented equivalently as a 'one-to-one' mapping from Equation (11). In this case, the z identical logical propositions (A_{1s} and ... and A_{ms}) ... (A_{1s} and ... and A_{ms}) in the 'if' part of the inconsistent set of rules in Equation (10) are represented by a single logical proposition (A_{1s} and ... and A_{ms}) in the 'if' part of a single equivalent rule in Equation (11).

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

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

Algorithm 1:

1. Put all inconsistent rules in disjoint sets whereby the rules in each set have the same permutation of linguistic values of inputs and different permutations of linguistic values of the inputs.
2. For each set of inconsistent rules, aggregate the rules into a single equivalent rule.
3. For each set of inconsistent rules, keep only the single equivalent rule.

Algorithm 1 guarantees that there are only consistent rules left in a fuzzy rule base after the completion of the simplification process. In this case, the number of consistent rules is equal to the number of inconsistent groups of rules plus the number of consistent rules. Therefore, the simplification process can be applied with a guaranteed success whereby the resulting simplified rule base is always consistent.

All steps in Algorithm 1 can be applied off-line. This is because the single equivalent rule can be found before the start of the fuzzification stage.

Algorithm 1 describes the aggregation process for inconsistent rules but it does not say when this process can be applied with full success, i.e. without any residual inconsistency being left. In other words, the question is when it would be possible to aggregate all inconsistent rules from each set into a single equivalent rule. This would be possible if the following three Conditions 1–3 are fulfilled with respect to the fuzzy membership functions for the output:

Condition 1: The number of these fuzzy membership functions is odd, i.e. there is a fuzzy membership function in the middle.

Condition 2: The fuzzy membership function in the middle is symmetrical, i.e. it has an axis of symmetry.

Condition 3: Each of the remaining fuzzy membership functions has a symmetrical image with respect to the axis of symmetry of another symmetrical fuzzy membership function.

Conditions 1–3 guarantee that the aggregation process will lead to a single equivalent rule for each set of inconsistent rules. In this case, the single equivalent rule for each set of inconsistent rules in the aggregated system would represent an approximation of the associated inconsistent rules from the same set in the original system. Although Conditions 1–3 may appear to be restrictive, they are actually not as most fuzzy systems meet these conditions anyway as part of the requirements for spreading the fuzzy membership functions for the output uniformly across its discrete variation range.

It should also be noted that Conditions 1–3 guarantee precise approximation of each set of inconsistent rules in the rule base with the associated single equivalent rule derived during the aggregation process. However, these conditions have mainly theoretical importance because precise approximation is possible under the assumption that the remaining rules from the rule base are ignored, as shown further in Section 4.

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 Equation (1) can be represented by the following n equivalent single-output fuzzy systems:

$$\begin{aligned}
&If\ i_1\ is\ v_{i11}\ and \ldots and\ i_m\ is\ v_{im1}\ then\ o_q\ is\ v_{q11} \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
&If\ i_1\ is\ v_{i1r}\ and \ldots\ and\ i_m\ is\ v_{imr}\ then\ o_q\ is\ v_{q1r} \\
&q = 1, \ldots, n \qquad (18)
\end{aligned}$$

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

4. Theoretical results

The rule base simplification method is applied here to a single-input-single-output example in which the rule base includes a single set of two inconsistent disjunctive rules. This example illustrates the rule base simplification method theoretically whereby the remaining rules from the rule base are ignored.

A fuzzy system has the following set of two inconsistent rules:

$$\begin{aligned}
&If\ i_1\ is\ P\ then\ o_1\ is\ S \\
&\qquad or \\
&If\ i_1\ is\ P\ then\ o_1\ is\ B \qquad (19)
\end{aligned}$$

where the simple linguistic terms *P*, *S* and *B* denote the linguistic values *positive*, *small* and *big*, respectively.

In accordance with Theorem 1, this system can be approximated precisely with the single equivalent rule:

$$If\ i_1\ is\ P\ then\ o_1\ is\ M \qquad (20)$$

whereby the linguistic term *M* (*medium*) for the output in this rule has replaced the terms *S* (*small*) and *B* (*big*) for the same output from the set of two inconsistent disjunctive rules. In this case, Algorithm 1 should be applied and Conditions 1–3 should hold.

For clarity, the fuzzy system from Equation (19) will be called 'original' whereas the fuzzy system from Equation (20) will be referred to as 'aggregated'. The difference between these two systems can be illustrated by the implication substage, the aggregation substage and the defuzzification stage. In this case, the fuzzification stage and the application substage for the two systems are the same due to the identical 'if' parts for the input, as shown by Equations (19, 20).

As the 'if' parts of the two rules in the original system are identical, the firing strength g_S for the first rule and the firing strength g_B for the second rule in this system are assumed to have been found to be equal to 0.66. Likewise, due to the identity between the 'if' part of the single rule in the aggregated system and the antecedent parts of the two rules in the original system, the firing strength g_M for this single rule must also have been found to be equal to 0.66.

At the implication substage, the fuzzy membership functions F_S and F_B for the output from the original system are obtained as:

$$\begin{aligned}
F_S = &\{0/0,\ 0.33/1, 0.66/2, 0.66/3, 0.66/4,\ 0.33/5, \\
&0/6,\ 0/7,\ 0/8,\ 0/9,\ 0/10,\ 0/11,\ 0/12\} \\
F_B = &\{0/0,\ 0/1,\ 0/2,\ 0/3,\ 0/4,\ 0/5,\ 0/6,\ 0.33/7, \\
&0.66/8,\ 0.66/9,\ 0.66/10,\ 0.33/11,\ 0/12\} \qquad (21)
\end{aligned}$$

where F_S and F_B represent the linguistic values *S* and *B*, respectively.

Due to the trapezoidal shape F_S and F_B, the associated fuzzy membership degrees f_S and f_B for any element *y* from the discrete variation range for the output will be mapped by:

$$\begin{aligned}
&f_s = 0,\ if\ y \le a_s \\
&f_s = (y - a_s)/(b_s - a_s),\ if\ a_s \le y \le b_s \\
&f_s = 0.66,\ if\ b_s \le y \le c_s \\
&f_s = (d_s - y)/(d_s - c_s),\ if\ c_s \le y \le d_s \\
&f_s = 0,\ if\ d_s \le y \qquad (22)
\end{aligned}$$

$$\begin{aligned}
&f_B = 0,\ if\ y \le a_B \\
&f_B = (y - a_B)/(b_B - a_B),\ if\ a_B \le y \le b_B \\
&f_B = 0.66,\ if\ b_B \le y \le c_B \\
&f_B = (d_B - y)/(d_B - c_B),\ if\ c_B \le y \le d_B \\
&f_B = 0,\ if\ d_B \le y \qquad (23)
\end{aligned}$$

where the parameters of the membership functions F_S and F_B are the following

$$\begin{aligned}
&a_s = 0,\ b_s = 2,\ c_s = 4,\ d_s = 6 \\
&a_b = 6,\ b_B = 8,\ c_B = 10,\ d_B = 12 \qquad (24)
\end{aligned}$$

At the aggregation substage, the aggregated fuzzy membership functions F_{SB} for the output from the original is obtained as follows:

$$\begin{aligned}
&F_{SB} = F_s\ or\ F_B = \{0/0,\ 0.33/1,\ 0.66/2, \\
&0.66/3,\ 0.66/4,\ 0.33/5,\ 0/6,\ 0.33/7,\ 0.66/8, \\
&0.66/9,\ 0.66/10,\ 0.33/11,\ 0/12\} \qquad (25)
\end{aligned}$$

At the defuzzification stage, the defuzzified value D_{SB} for the output from the original system is obtained as follows:

$$\begin{aligned} D_{SB} = {} & [(0.0) + (0.33.1) + (0.66.2) \\ & +(0.66.3) + (0.66.4) + (0.33.5) + 0.6 \\ & +(0.33.7) + (0.66.8) + (0.66.9) \\ & +(0.66.10) + (0.33.11) + (0.12)]/(0 \\ & +0.33 + 0.66 + 0.66 + 0.66 + 0.33 + 0 \\ & +0.33 + 0.66 + 0.66 + 0.66 + 0.33 + 0) \\ & = 32/5.33 = 6 \end{aligned} \tag{26}$$

At the implication substage, the fuzzy membership function F_M for the output from the aggregated system is obtained as:

$$\begin{aligned} F_M = {} & \{0/0,\ 0/1,\ 0/2,\ 0/3,\ 0.33/4,\ 0.66/5, \\ & 0.66/6,\ 0.66/7,\ 0.33/8,\ 0/9,\ 0/10,\ 0/11, \\ & 0/12\} \end{aligned} \tag{27}$$

where F_M represents the linguistic value M.

Due to the trapezoidal shape of F_M, the associated fuzzy membership degree f_M for any element y from the discrete variation range for the output will be mapped by:

$$\begin{aligned} & f_M = 0,\ if\ y \le a_M \\ & f_M = (y - a_M)/(b_M - a_M),\ if\ a_M \le y \le b_M \\ & f_M = 0.66,\ if\ b_M \le y \le c_M \\ & f_M = (d_M - y)/(d_M - c_M),\ if\ c_M \le y \le d_M \\ & f_M = 0,\ if\ d_M \le y \end{aligned} \tag{28}$$

where the parameters of the membership functions F_M and F_B are the following:

$$a_M = 3,\ b_M = 5,\ c_M = 7,\ d_M = 9 \tag{29}$$

At the aggregation substage, the aggregated fuzzy membership function for the output from the aggregated system is equal to F_M because there is only one rule in this system.

At the defuzzification stage, the defuzzified value D_M for the output from the aggregated system is obtained as follows:

$$\begin{aligned} D_M = {} & [(0.0) + (0.1) + (0.2) + (0.3) \\ & +(0.33.4) + (0.66.5) + (0.66.6) + (0.66.7) \\ & +(0.33.8) + (0.9) + (0.10) + (0.11) \\ & +(0.12)]/(0 + 0 + 0 + 0 + 0.33 \\ & +0.66 + 0.66 + 0.66 + 0.33 + 0 + 0 + 0 \\ & +0) = 16/2.66 = 6 \end{aligned} \tag{30}$$

It follows from Equations (26 and 30) that the defuzzified value D_{SB} for the output from the original system is equal to the defuzzified value D_M for the same output from the aggregated system. This shows that the two systems from Equations (19, 20) are equivalent in terms of their behaviour for the chosen crisp value of the input.

5. Simulation results

The rule base aggregation method is applied to a case study on transportation demand management where the main goal is to model preferences of employees to telecommuting. In this case, inconsistency is dealt with using the aggregation approach whereby the original and the aggregated model are compared to each other with regard to their ability to approximate the given data. In this comparison, Theorem 1 and Algorithm 1 are used for the derivation of the aggregated model under the assumption that Conditions 1–3 hold.

The data is based on a survey that has been obtained from seven government organisations in the central business district of Tehran - capital city of Iran. The inputs taken into account for determining preferences of employees are computer time usage, travel cost from home to work and work experience. The output from this process is the number of days on which each employee prefers to telecommute to work from home.

The three inputs and the output are presented by three linguistic terms each, as shown in Figs. 1–4. These terms belong to the set *low, medium, high* and they are represented by symmetric triangular fuzzy membership functions that cover uniformly the whole variation range for each of these four variables.

The rule bases for the aggregated and original model are presented in Tables 1 and 2, respectively. In this case, the original model has 36 rules whereas the aggregated model has only 27 rules. This is due to the fact each of the 9 pairs of inconsistent rules from the original

Table 1
Aggregated model

No	Computer usage	Travel cost	Work experience	Output
1	1	1	1	1
2	1	1	2	1
3	1	1	3	2
4	1	2	1	1
5	1	2	2	2
6	1	2	3	2
7	1	3	1	2
8	1	3	2	2
9	1	3	3	2
10	2	1	1	2
11	2	1	2	2
12	2	1	3	2
13	2	2	1	2
14	2	2	2	2
15	2	2	3	2
16	2	3	1	2
17	2	3	2	2
18	2	3	3	3
19	3	1	1	2
20	3	1	2	3
21	3	1	3	3
22	3	2	1	2
23	3	2	2	2
24	3	2	3	2
25	3	3	1	3
26	3	3	2	3
27	3	3	3	3

model has been replaced by a single equivalent rule in the aggregated model in accordance with Theorem 1, Algorithm 1 and Conditions 1–3.

The output surfaces for the aggregated and original model are plotted in Figs. 5–10, respectively. In this case, the output surface for each of the two models is plotted for three fixed values for the third input. It is

Table 2
Original model

No	Computer usage	Travel cost	Work experience	Output
1	1	1	1	1
2	1	1	2	1
3	1	1	3	1
4	1	1	3	3
5	1	2	1	1
6	1	2	2	2
7	1	2	3	1
8	1	2	3	3
9	1	3	1	1
10	1	3	1	3
11	1	3	2	2
12	1	3	3	2
13	2	1	1	1
14	2	1	1	3
15	2	1	2	2
16	2	1	3	1
17	2	1	3	3
18	2	2	1	1
19	2	2	1	3
20	2	2	2	2
21	2	2	3	1
22	2	2	3	3
23	2	3	1	2
24	2	3	2	2
25	2	3	3	3
26	3	1	1	1
27	3	1	1	3
28	3	1	2	3
29	3	1	3	3
30	3	2	1	1
31	3	2	1	3
32	3	2	2	2
33	3	2	3	2
34	3	3	1	3
35	3	3	2	3
36	3	3	3	3

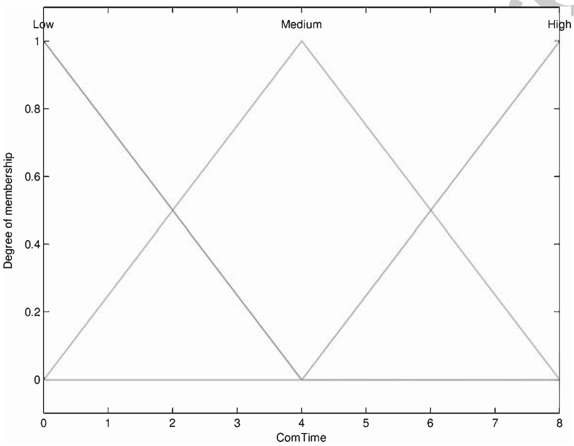

Fig. 1. Linguistic terms for first input (computer time usage).

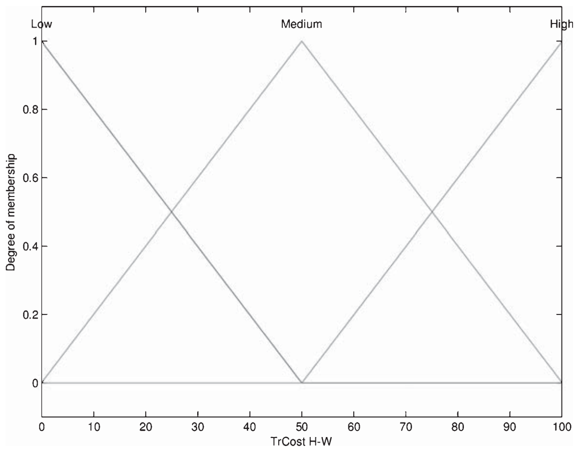

Fig. 2. Linguistic terms for second input (travel cost).

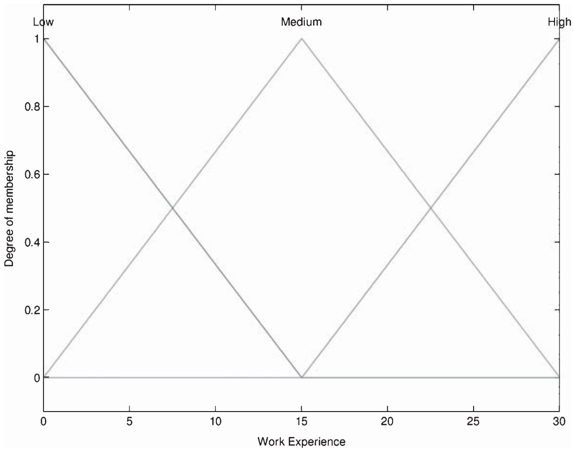

Fig. 3. Linguistic terms for third input (work experience).

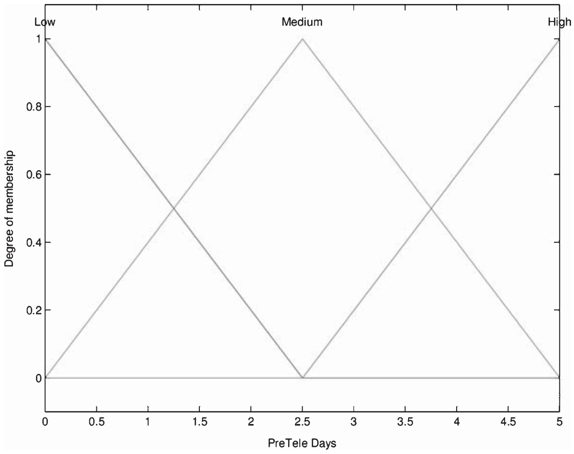

Fig. 4. Linguistic terms output (telecommuting preference).

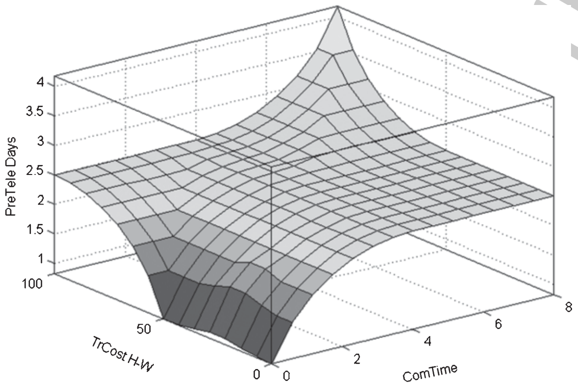

Fig. 5. Output surface for aggregated model with third input fixed to 0.

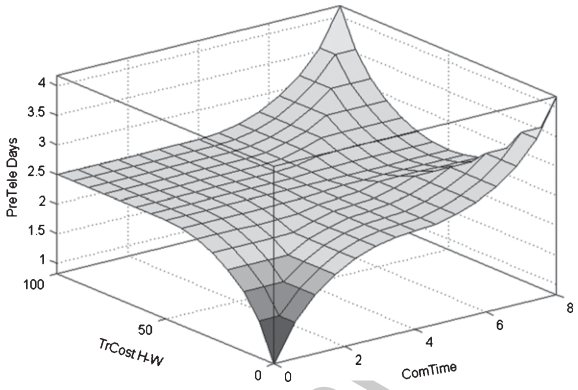

Fig. 6. Output surface for aggregated model with third input fixed to 15.

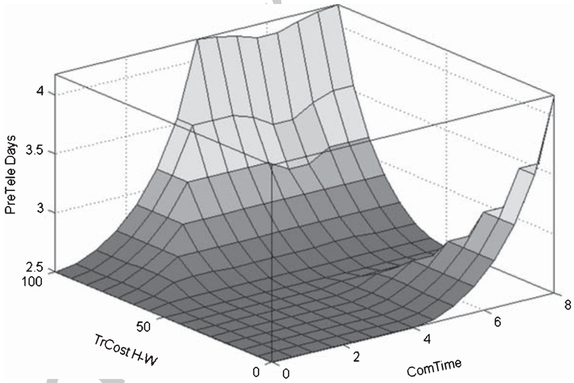

Fig. 7. Output surface for aggregated model with third input fixed to 30.

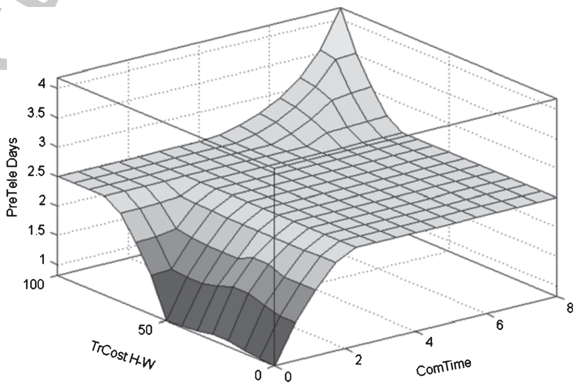

Fig. 8. Output surface for original model with third input fixed to 0.

obvious from these plots that the output surfaces of the two models are quite similar.

The output values for the aggregated and original model are plotted in Figs. 11 and 12, respectively. In

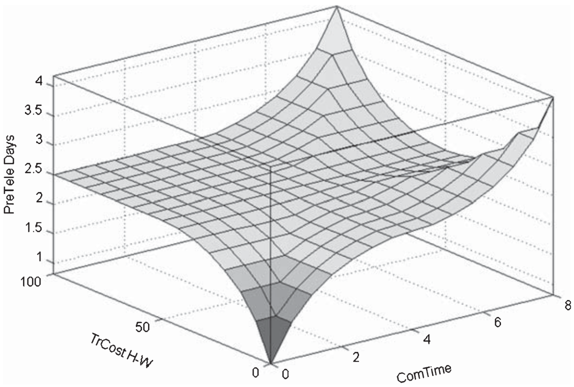

Fig. 9. Output surface for original model with third input fixed to 15.

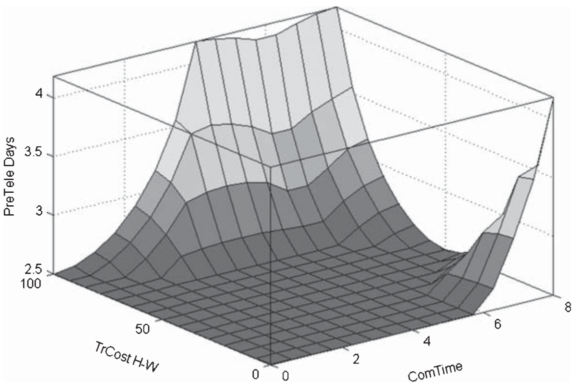

Fig. 10. Output surface for original model with third input fixed to 30.

this case, the output value for each of the two models is plotted alongside the data output, i.e. the observation value for each of the 245 individuals from the survey. It is also obvious from these plots that the output values of two models are quite similar. Finally, the aggregated and the original model are evaluated comparatively in terms of the Mean Absolute Deviation (MAD) in Table 3. The column labels in this table have the following meanings and notations: number of individual (Ind), output value for original model (Org), rounded value for original model (Rounded-Org), output value for aggregated model (Agg), rounded value for aggregated model (Rounded-Agg), output value from observations (Obs), MAD for original model (MAD Org) and MAD for aggregated model (MAD Agg). All fractional values of the outputs for the two models have been rounded to the nearest integer to make them more compatible with the integer format of the data output.

The last row in Table 3 shows the average MAD for the aggregated and original model taken across all 245 individuals from the survey. In this case, the aggregated model outperforms the original model in terms of accuracy although it has a substantially smaller number of rules. This implies that the removal of the inconsistent rules from the original model has led not only to improvement of efficiency but also to improvement of accuracy. In other words, the inconsistent rules represent redundancy in the rule base whose removal through aggregation leads to improvement in these two model performance indicators.

6. Conclusion

The proposed rule base aggregation method reduces the number of rules in a fuzzy system. This translates into a reduction of the associated computational complexity in terms of the overall amount of operations

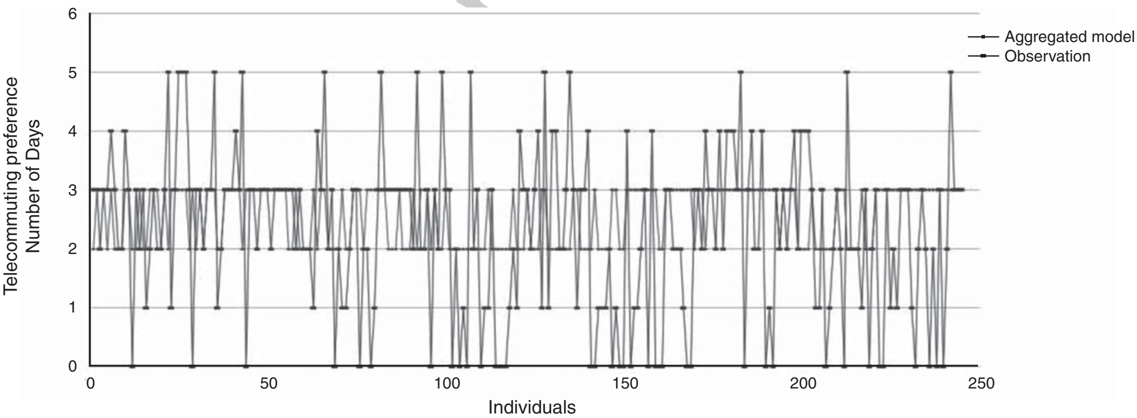

Fig. 11. Simulation results for aggregated model.

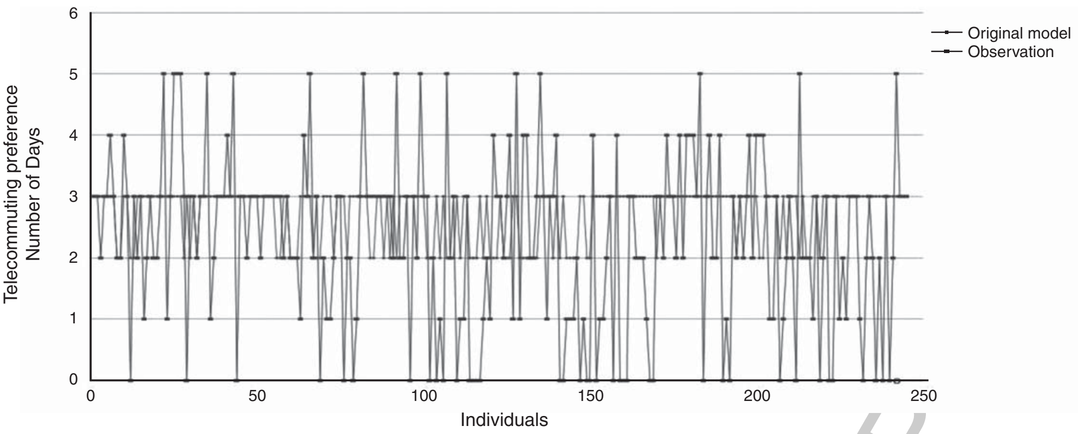

Fig. 12. Simulation results for original model.

Table 3
Comparative evaluation of both models

Ind	Org	Rounded Org	Agg	Rounded Agg	Obs	MAD Org	MAD Agg
1	2.5	3	2.4155	2	3	0	1
2	2.5	3	2.5315	3	3	0	0
3	2.5	3	2.526	3	2	1	1
4	2.5	3	2.5086	3	3	0	0
5	2.5	3	2.4155	2	3	0	1
6	2.5	3	2.5	3	4	1	1
7	2.5	3	2.4197	2	3	0	1
8	1.9888	2	1.8479	2	2	0	0
9	1.8329	2	1.7478	2	2	0	0
10	2.5	3	2.5856	3	4	1	1
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
235	3.2393	3	3.2522	3	2	1	1
236	2.5	3	2.6114	3	0	3	3
237	2.5	3	2.5908	3	2	1	1
238	2.5	3	2.5	3	0	3	3
239	2.5	3	2.5	3	3	0	0
240	2.5	3	2.6985	3	0	3	3
241	2.5	3	2.5999	3	2	1	1
242	3.0555	3	3.1394	3	5	2	2
243	2.5	3	2.529	3	3	0	0
244	2.5	3	2.7698	3	3	0	0
245	2.5	3	2.6392	3	3	0	0
			Average			1.114	1.110

during the stages of fuzzification, inference and defuzzification. Therefore, the method is suitable for time-critical applications in which rule base refinement is either unacceptable due to time constraints or impossible due to lack of additional date or knowledge. Besides this, the solution obtained by the proposed method outperforms the one obtained without using this method in terms of both efficiency and accuracy for the case study under consideration.

The proposed method can be used without modification 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.

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 whose rule base is updated before the start of the fuzzification stage. In this case, if the updated rule base is inconsistent, it can be made consistent by aggregation of the inconsistent rules.

Acknowledgments

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, *Complexity management in fuzzy systems*, Springer, Berlin, 2007.

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

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

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

[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* **2005** (150), 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), 217–223.

[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 Dissertation, 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 Dissertation*, 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.