Mathware & Soft Computing 14 (2007), 201-216

Advanced Inference in Fuzzy Systems by Rule Base Compression

A. Gegov[1] and N. Gobalakrishnan[2]
[1,2]University of Portsmouth, School of Computing, Buckingham Building, Portsmouth PO1 3HE, UK
Alexander.gegov@port.ac.uk [1], *muges_3106@hotmail.com* [2]

Abstract

This paper describes a method for rule base compression of fuzzy systems. The method compresses a fuzzy system with an arbitrarily large number of rules into a smaller fuzzy system by removing the redundancy in the fuzzy rule base. As a result of this compression, the number of on-line operations during the fuzzy inference process is significantly reduced without compromising the solution. This rule base compression method outperforms significantly other known methods for fuzzy rule base reduction.

Keywords: Fuzzy system, rule base reduction, rule base compression.

1 Introduction

Decision making processes are usually accompanied by uncertainty which is inherent to the environment in which the information is being gathered. Such uncertainty may seriously compromise the reliability of the information gathering process as well as the quality of any subsequent decisions made.

Fortunately, fuzzy systems are well suited for decision making tasks characterised by uncertainty. The latter can be taken into account by means of the approximate reasoning and logical inference capabilities of fuzzy systems. However, there is often a problem in this case caused by the large number of rules which depends on the number of inputs. This usually leads to a significant increase of the qualitative complexity in terms of poor transparency and unclear interpretation of the fuzzy rules as well as the quantitative complexity in terms of increased number of operations during the fuzzy inference process. This point is illustrated by Equations (1)-(2).

A fuzzy system is usually represented by if-then rules of the form

$$\text{If } i_1 \text{ is } v_{i1,1} \text{ and } \ldots \text{ and } i_m \text{ is } v_{im,1}$$
$$\text{then } o_1 \text{ is } v_{o1,1} \text{ and } \ldots \text{ and } o_n \text{ is } v_{on,1} \quad (1)$$

$$\text{If } i_1 \text{ is } v_{i1,r} \text{ and } \ldots \text{ and } i_m \text{ is } v_{im,r}$$
$$\text{then } o_1 \text{ is } v_{o1,r} \text{ and } \ldots \text{ and } o_n \text{ is } v_{on,r} \quad (2)$$

where m is the number of inputs, n is the number of outputs and r is the number of fuzzy rules in the system. In this case, i_p, $p=1,m$ represents the p-th input, $v_{ip,s}$ $p=1,m,$ $s=1,r$ 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 $v_{oq,s}$ $q=1,n,$ $s=1,r$ is the linguistic value of the q-th output in the s-th rule.

As shown in Figure 1, the number of rules in a fuzzy system r is an exponential function of the number of the inputs m and the number of linguistic values k that these inputs can take [3]. In most cases, this function is in the form of Equation (3).

$$r = k^m \quad (3)$$

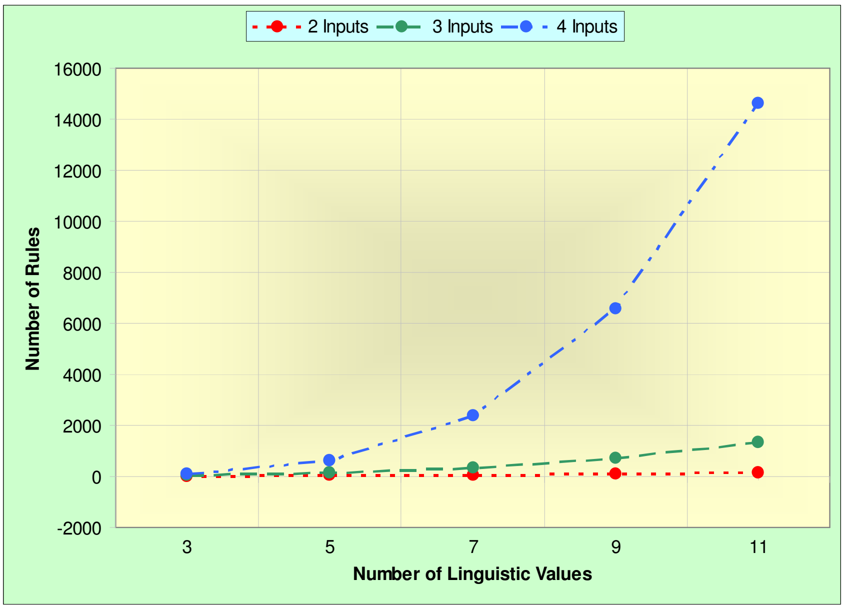

Figure 1: Number of rules for a fuzzy system with one, two and three inputs

Therefore, the question that arises here is how to use fuzzy systems for tackling uncertain information without making them fall into the trap of complexity and thus compromising their suitability for tackling the uncertainty in the first place. The assumption made in this case is that when it is impossible to improve the quality of information due to time or operation related constraints, it should still be possible to use this information in a reliable way by means of an enhanced decision making process which utilises the capabilities of fuzzy systems for dealing with uncertainty and simplifies their complexity at the same time.

2 Operation stages in fuzzy systems

Fuzzy systems map a given input to an output using the theory of fuzzy sets, as shown in Figure 2. The most commonly used fuzzy system is the Mamdani system, which is used in this paper. The mapping above consists of three major stages - fuzzification, inference and defuzzification. In the fuzzification stage, it has been decided to use the two most widely used types of membership functions - triangular and trapezoidal. The inference stage is divided into three substages - application, implication and aggregation. It has been decided to use the conjunctive method (MIN) in the application stage, the truncation method in the implication stage and disjunctive method (MAX) in the aggregation stage. In the defuzzification stage, it has been decided to apply the most widely used centroid method. For the software implementation of the method, it has been decided to use the MATLAB Fuzzy Logic Toolbox due to its wide applicability in both academia and industry.

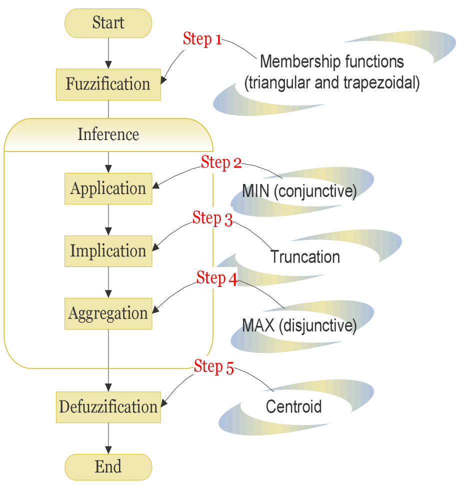

Figure 2: Mapping of inputs to outputs in a fuzzy system

3 Rule base compression method

A detailed algorithm for rule base compression is shown in Figure 3. The algorithm implements the rule base compression method, introduced in this paper. The method arranges monotonic rules in groups and finds the dominant rule in each group. Monotonic rules have the same linguistic value for the output and are very common in fuzzy systems. The dominant rule is the one with the highest firing strength whereby all other rules from the group do not have any impact on the output. This represents redundancy in the fuzzy system which is exploited by the rule base compression method.

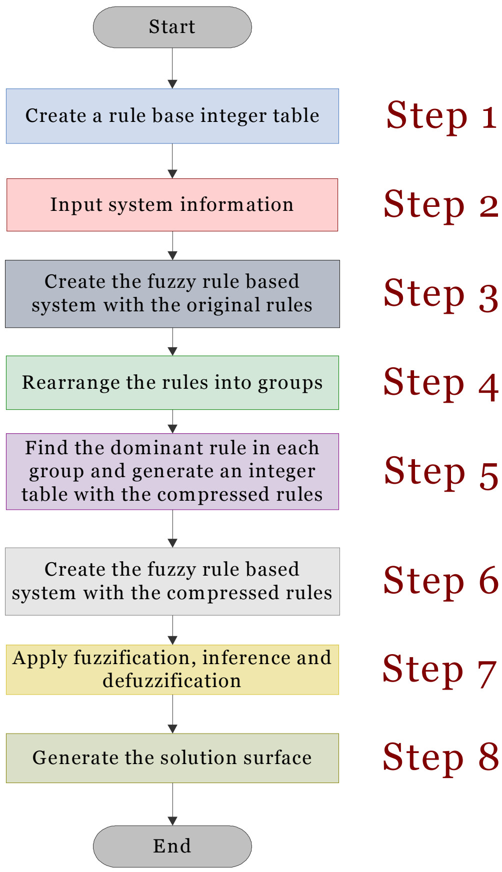

Figure 3: Algorithm for rule base compression

The algorithm in Figure 3 consists of eight major steps [4].

The first step is based on a dialogue with the user who is prompted to enter all the information about the rule base. The user is prompted to enter information such as the number of inputs and outputs, as well as the number of linguistic values for each input and output. The user also has to enter the output value for each possible combination of input linguistic values, which is displayed by the system. After the acquisition of all the necessary information, the algorithm creates an integer table with positive integer numbers.

The second step is again based on a dialogue with the user who is prompted to enter all the fuzzy inference system information such as variable names and membership function definitions.

In the third step, the software creates and saves the fuzzy system with the original rules.

The fourth step is to re-arrange the rules into groups. These groups are sorted in an increasing order with respect to the chosen output linguistic values. This aggregation process can be carried out entirely off-line.

The fifth step is to find the dominant rule for each group. This step can only be applied on-line, which is due to the fact that the dominant rules can be found only after the completion of the fuzzification and application stages.

In the sixth step, the software creates a fuzzy rule base system with the compressed rules and saves it.

Step seven is the system evaluation process which uses the file saved in step six. This step evaluates the output of a fuzzy rule based system for given inputs.

The final step eight is to generate the solution surface. This solution surface is created through a given number of points for the crisp input values and the defuzzified output values.

4 Software implementation

The MATLAB Fuzzy Logic Toolbox has been used for the software implementation of the rule base compression method. The software has been implemented with full functionality. The implemented software is discussed in more detail below, where two case studies are used for demonstrating the rule base compression method. In either case, the associated rule bases are represented for simplicity by integer tables whereby the linguistic values of inputs and outputs are replaced by integers.

The first case study is for a fuzzy system for aircraft landing control [5]. The system is described by the inputs i_1, i_2 and the output o_1 where i_1 is the relative height (h) of the aircraft in feet (ft), i_2 is the vertical velocity (v) of the aircraft in feet per second (ft/s) and o1 is the control effort (e) in libras (lb) that must be applied to the aircraft. In this case, i_1 can take the four linguistic values near zero

(NZ=1), small (S=2), medium (M=3) and large (L=4), whereas both i_2 and o_1 can take the five linguistic values down large (DL=1), down small (DS=2), zero (Z=3), up small (US=4) and up large (UL=5).

The integer tables for the original and the compressed fuzzy system for the crisp values 980 and -14.2 of the inputs are shown in Tables 1–2.

Table 1. Integer table of the original fuzzy system of aircraft landing control

Rule number	Linguistic value of i_1	Linguistic value of i_2	Linguistic value of o_1
10	2	5	1
14	3	4	1
15	3	5	1
18	4	3	1
19	4	4	1
20	4	5	1
4	1	4	2
5	1	5	2
9	2	4	2
13	3	3	2
17	4	2	2
3	1	3	3
8	2	3	3
12	3	2	3
16	4	1	3
7	2	2	4
11	3	1	4
1	1	1	5
2	1	2	5
6	2	1	5

Table 2. Integer table of the compressed fuzzy system for aircraft landing control

Rule number	Linguistic value of i_1	Linguistic value of i_2	Linguistic value of o_1
10	2	5	1
17	4	2	2
12	3	2	3
11	3	1	4
1	1	1	5

The second case is for a fuzzy system for the operation of a service centre for spare parts [6]. The system is described with 3 inputs i_1, i_2, i_3 and one output o_1. Whereby i_1 is the repair utilisation factor, i_2 is the number of servers, i_3 is the mean delay of service and o_1 is the number of spare parts [5]. In this case, i_1 can take the three linguistic values low (L=1) , medium (M=2) and high (H=3), i_2 can take the three linguistic values small (S=1), medium (M=2) and large (L=3), i_3 can take the three linguistic values very short (VS=1), short (S=2) and medium (M=3), whereas o_1 can take the seven linguistic values very small (VS=1), small (S=2), rather small (RS=3), medium (M=4) rather large (RL=5), large (L=6) and very large (VL=7).

The integer tables for the original and the compressed fuzzy system for the crisp values of the inputs 0.2, 0.3 and 0.5 are shown below in Tables 3-4.

Table 3. Integer table of the original fuzzy system for spare parts service

Rule number	Input 1 i_1	Input 2 i_2	Input 3 i_3	Output o_1
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 4. Integer table of the compressed fuzzy system for spare parts service

Rule number	Input 1 i_1	Input 2 i_2	Input 3 i_3	Output o_1
1 or 2 or 3 or 4 or 5 or 6 or 9 or 11 or 12 or 15	1 or 2	1 or 2 or 3	1 or 2 or 3	1
7 or 8 or 10 or 14 or 18 or 24	1 or 2 or 3	1 or 2 or 3	1 or 2 or 3	2
13 or 17 or 27	2 or 3	2 or 3	1 or 2 or 3	3
16 or 21 or 22 or 23 or 26	2 or 3	1 or 2 or 3	1 or 2 or 3	4
25	3	3	1	5
20	3	1	2	6
19	3	1	1	7

The output surfaces for the first case study are shown in Figures 4–5 and the output services for the second case study are shown in Figures 6–11.

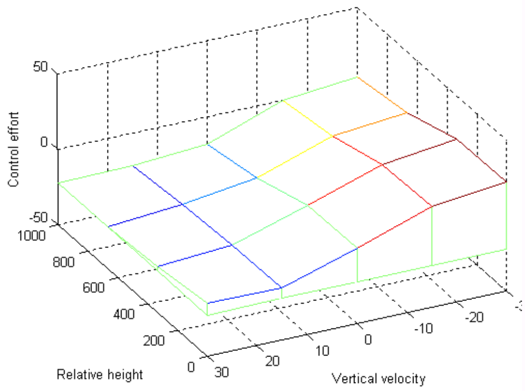

Figure 4: Output surface for the original system

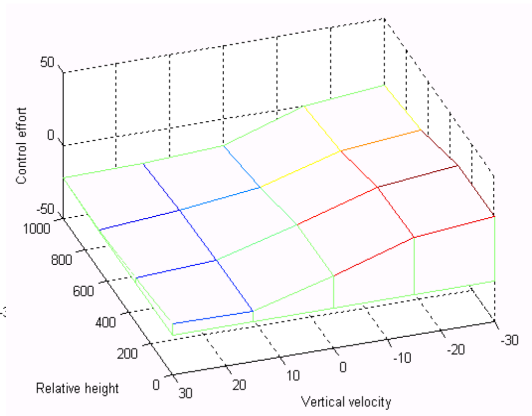

Figure 5: Output surface for the compressed system

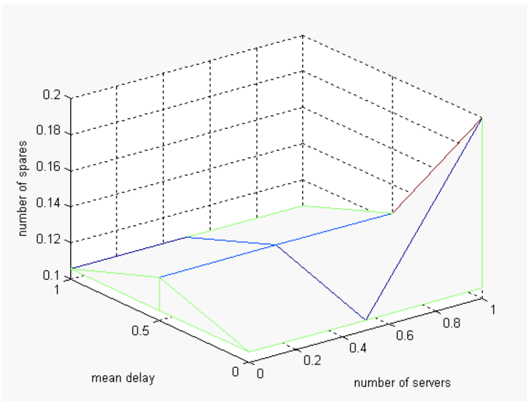

Figure 6: Output surface for the original system with input 1 = 0

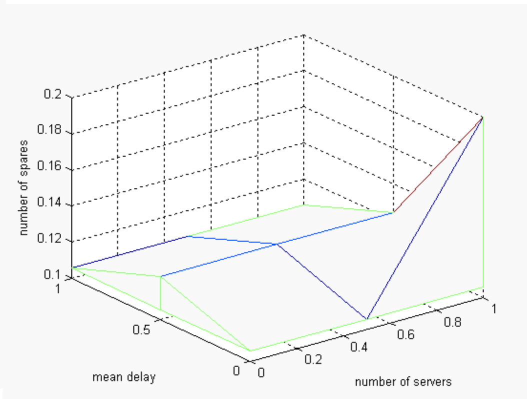

Figure 7: Output surface for the compressed system with input 1 = 0

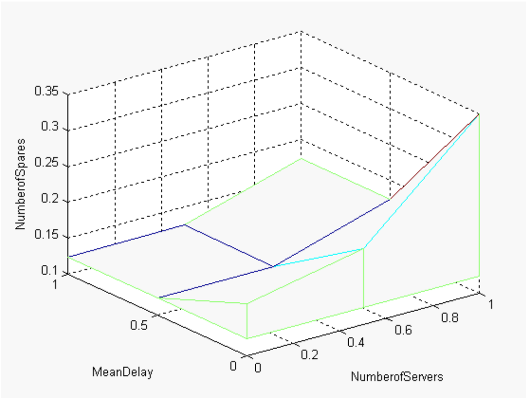

Figure 8: Output surface for the original system with input 1 = 0.5

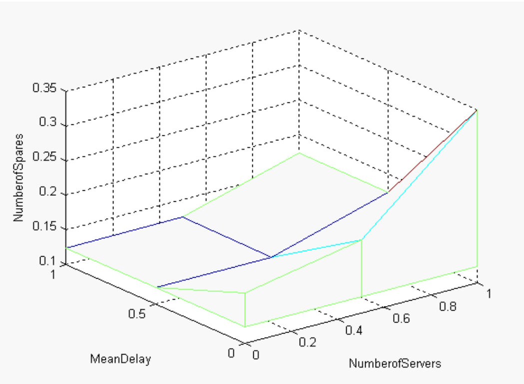

Figure 9: Output surface for the compressed system with input 1 = 0.5

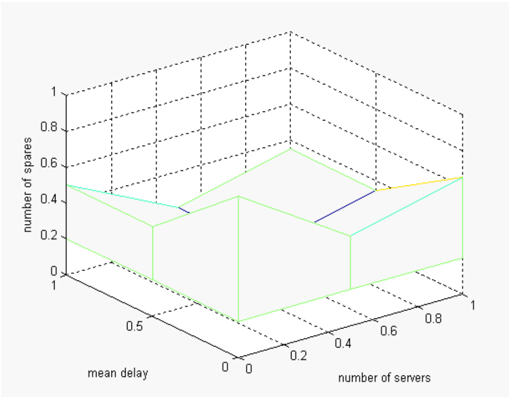

Figure 10: Output surface for the original system with input 1 = 1

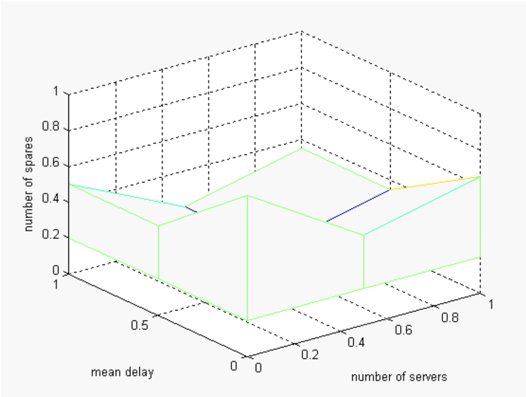

Figure 11: Output surface for the compressed system with input 1 = 1

As the above presented output surfaces are too rough, it has been decided to generate more detailed output surfaces with 10 times more data points. For the first case study, input 1 has been presented with 40 data points and input 2 has been presented with 50 data points. The associated output surfaces are shown in Figures 12–13. For the second case study, both input 2 and input 3 have been presented with

30 data points whereby input 1 has been fixed to 0, 0.5 and 1. The associated output surfaces are shown in Figures 14–19.

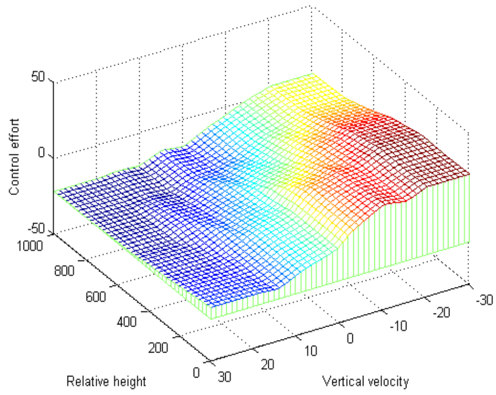

Figure 12: Output surface for the original system

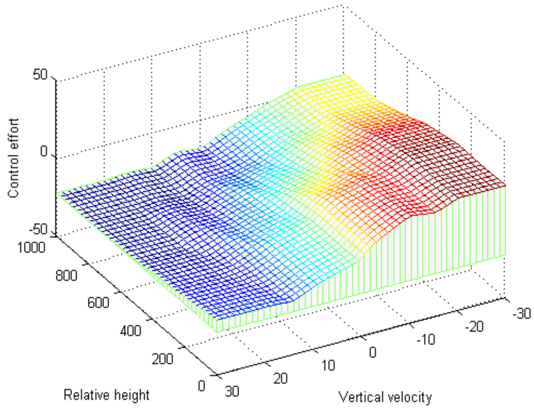

Figure 13: Output surface for the compressed system

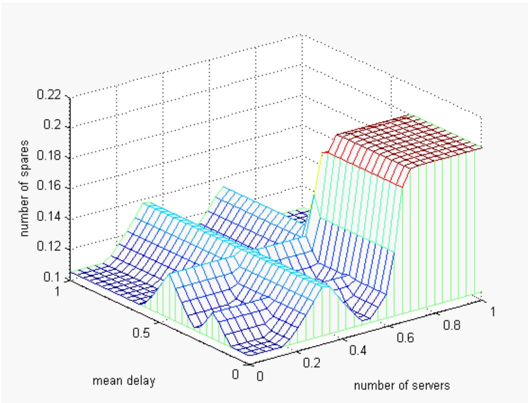

Figure 14: Output surface for the original system with input 1 = 0

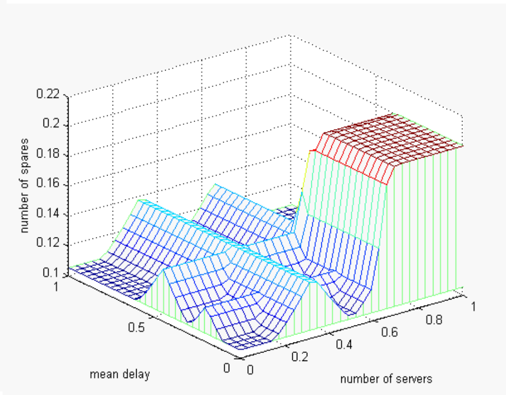

Figure 15: Output surface for the compressed system with input 1 = 0

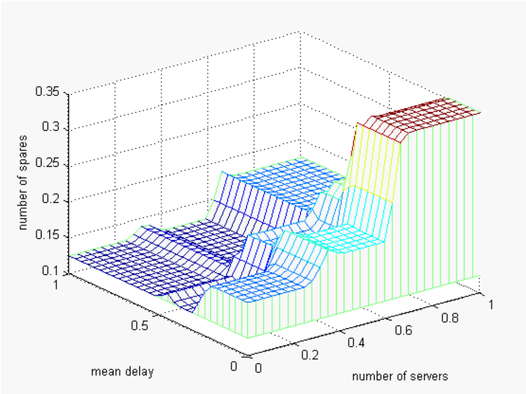

Figure 16: Output surface for the original system with input 1 = 0.5

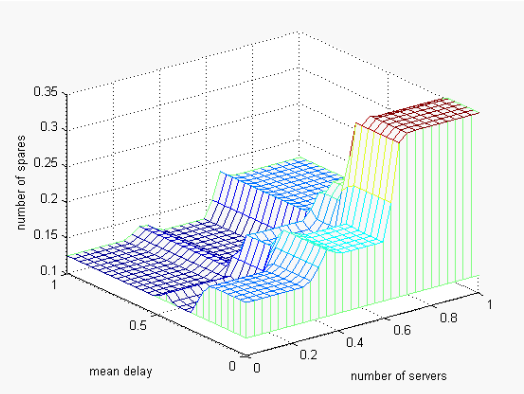

Figure 17: Output surface for the compressed system with input 1 = 0.5

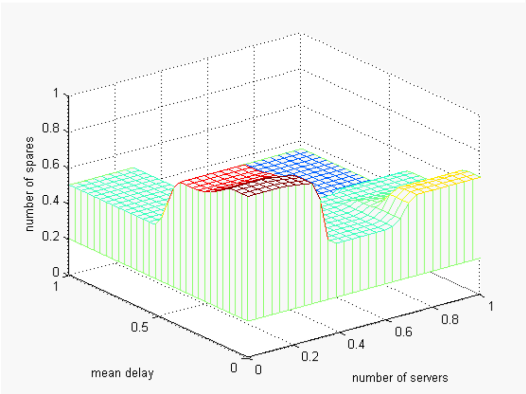

Figure 18: Output surface for the original system with input 1 = 1

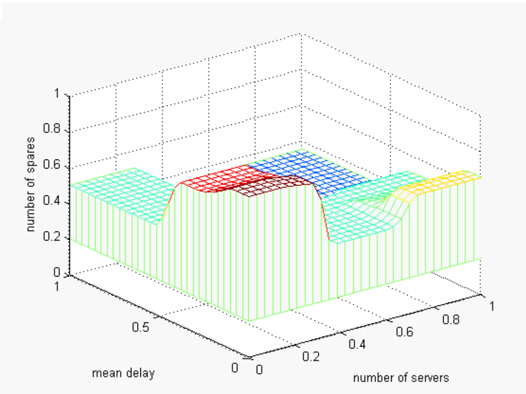

Figure 19: Output surface for the compressed system with input 1 = 1

5 Comparative evaluation

The evaluation approach used here is based on precise calculations and it is superior to the well known approximate BIG(O) approach. It has been decided to evaluate the developed rule base compression method with the hierarchical method, which is the most advanced method available for rule base reduction due to its systematic nature and wide applicability.

The fuzzy systems implementing these two methods are compared in terms of exact amount of on-line operations, which is determined by the overall number of elementary operations (EO) such as addition, subtraction, multiplication, division and comparison.

The quantitative complexity for both systems is calculated for each stage and substage such as fuzzification (EO_{FU}), inference that includes application (EO_{AP}), implication (EO_{IM}) and aggregation (EO_{AG}), and defuzzification (EO_{DE}). The compressed system has an additional stage of comparison (EO_{CO}) of the rule firing strengths whose purpose is to determine the dominant rules [1].

The quantitative complexity is calculated for the hierarchical and the compressed system, as shown by the Equations (4)-(5)

$$EO^{HS} = EO_{FU}{}^{HS} + EO_{AP}{}^{HS} + EO_{IM}{}^{HS} + EO_{AG}{}^{HS} + EO_{DE}{}^{HS} = (m - 1) . [(8 . t + 1) . w^2 + 12 . w + 2 . t - 1] . n . h \quad (4)$$

$$EO^{FS} = EO_{FU}{}^{FS} + EO_{AP}{}^{FS} + EO_{IM}{}^{FS} + EO_{AG}{}^{FS} + EO_{DE}{}^{FS} + EO_{CO}{}^{FS} = (m - 1) . [(8 . t + 1) . w^2 + 12 . w + 2 . t - 1] . n . h \quad (5)$$

where m is number of inputs, w is number of linguistic values per input, n is number of outputs, t is number of elements in the discrete universe discourse for the output and h is number of simulation cycles.

The results from the comparative evaluation of the quantitative complexity for the hierarchical and the compressed system are presented in Table 5 and Figure 20 whereby the hierarchical system implements the best available method of rule base reduction by decomposition into a multilayer hierarchical structure [2, 7, 8, 9, 10, 11, 12] and the compressed system implements the rule base compression method.

Table 5. Complexity of the hierarchical and the compressed system

Number of rules /fuzzy system	**Hierarchical system**	**Compressed system**
3^2 = 9	562	232
3^3 = 27	1,124	295
3^4 = 81	1,686	466
5^2 = 25	2,710	650

5^3 = 125	5,420	905
5^4 = 625	8,130	2,160
7^2 = 49	7,618	1,276
7^3 = 343	15,236	1,955
7^4 = 2,401	22,854	6,750
9^2 = 81	16,438	2,110
9^3 = 729	32,876	3,541
9^4 = 6,561	49,314	16,636
11^2 = 121	30,322	3,152
11^3 = 1,331	60,644	5,759
11^4 = 14,641	90,966	34,986

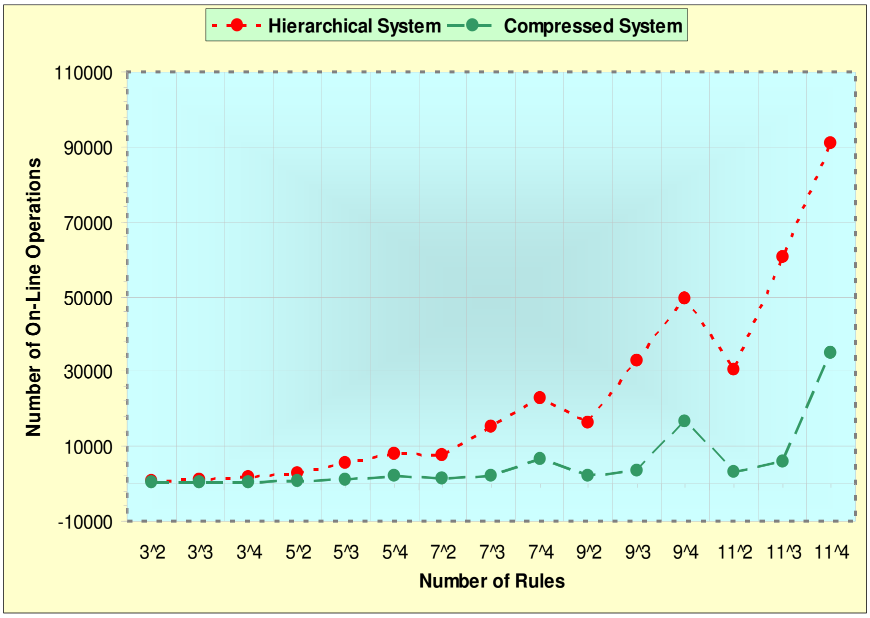

Figure 20: Complexity of the hierarchical and the compressed system

Table 5 and Figure 20 clearly show that the compressed system is superior to the hierarchical system for all considered permutations of linguistic values of inputs. These permutations have been chosen from the most commonly used applications of fuzzy systems.

6 Conclusion

The proposed method compresses a fuzzy system with an arbitrarily large number of rules into a smaller fuzzy system by removing the redundancy in the fuzzy rule base. As a result of this compression, the number of on-line operations is substantially reduced without compromising the solution. The method outperforms significantly all other known methods for fuzzy rule base reduction.

The method removes the redundant computations in the fuzzy inference stage with respect to the current crisp values of the inputs to the fuzzy system. This redundancy is caused by non-monotonic rules which have the same linguistic value for the output. Therefore, the rule base compression process has to identify the redundant rules after the fuzzification stage and remove them safely by preserving the defuzzified output from the fuzzy system. Once the defuzzification stage has finished, the original rule base of the fuzzy system is restored, the new crisp values of the inputs are measured at the beginning of the next fuzzification stage, and the rule base compression process is repeated. This approach is different from the known rule base reduction methods in that it reduces the complexity in fuzzy systems without compromising the solution by changing the structure of the fuzzy rule base, i.e. the latter contracts to a rule base of much smaller size before each inference stage and then expands to its original size before the beginning of the next fuzzification stage.

The method is a powerful tool for reducing the complexity in fuzzy systems. In particular, the removal of redundant non-monotonic rules leads to a significant reduction of the number of rules and the amount of operations during a standard simulation cycle of a fuzzy system which involves the stages of fuzzification, inference, defuzzification.

The method has been illustrated only for a single simulation cycle of single output fuzzy systems. However, it can be easily extended to multiple simulation cycles of multiple output systems. In this case, all procedures presented in this paper should be applied in exactly the same way to each simulation cycle of each output. This would obviously lead to a linear increase of the associated complexity, which would be proportional to the number of simulation cycles and the number of outputs.

The method facilitates the management of complexity in fuzzy systems. It allows the information contained in a non-monotonic rule base of a fuzzy system to be compressed in a non-lossy manner by removing the redundancy in the rule base. As a result this compression, the size of the large non-monotonic rule base is reduced significantly in each simulation cycle whereby the reduced monotonic rule base is equivalent to the large non-monotonic rule base in terms of its behaviour.

It has been shown in this paper that the rule base compression method can be applied successfully to a Mamdami type of fuzzy system irrespective of the number of inputs, outputs, membership functions, linguistic values and rules. This validation provides a solid basis for extending the method to a wider range of fuzzy systems.

References

[1] A.Gegov, *Complexity Management in Fuzzy Systems* (Springer, Berlin, 2007).

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

[3] M.Jamshidi, *Large Scale Systems: Modelling, Control and Fuzzy Logic* (Prentice-Hall, Englewood Cliffs, 1997).

[4] N.Gobalakrishnan, Software Tool for advanced Inference in Fuzzy system, BSc Final year Project (University of Portsmouth, 2006)

[5] T.Ross, *Fuzzy Logic with Engineering Applications* (John Wiley & Sons, Chichester, 2004).

[6] M.Negnevitsky, *Artificial Intelligence: A Guide to Intelligent Systems* (Pearson, Harlow, 2002).

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

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

[9] 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.

[10] 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.

[11] X.Zeng and J.Keane, Approximation capabilities of hierarchical fuzzy systems, *IEEE Transactions on Fuzzy Systems*, 13/5 (2005) 659-672.

[12] F.Chung and J.Duan, On multistage fuzzy neural network modelling, *IEEE Transactions on Fuzzy Systems*, 8/2 (2000) 125-142.