Fuzzy expert systems: Fuzzy inference

- Mamdani fuzzy inference
- Sugeno fuzzy inference
- Case study
- Summary

Fuzzy inference

In 1975, Professor **Ebrahim Mamdani** of London University built one of the first fuzzy systems to control a steam engine and boiler combination.

He applied a set of fuzzy rules supplied by experienced human operators.

Mamdani fuzzy inference

- The Mamdani-style fuzzy inference process is performed in four steps:

  - fuzzification of the input variables
  - rule evaluation
  - aggregation of the rule outputs
  - defuzzification

We examine a simple two-input one-output problem that includes three rules:

Rule: 1
IF *x* is *A*3
OR *y* is *B*1
THEN *z* is *C1*

Rule: 2
IF *x* is *A*2
AND *y* is *B*2
THEN *z* is *C*2

Rule: 3
IF *x* is *A*1
THEN *z* is *C*3

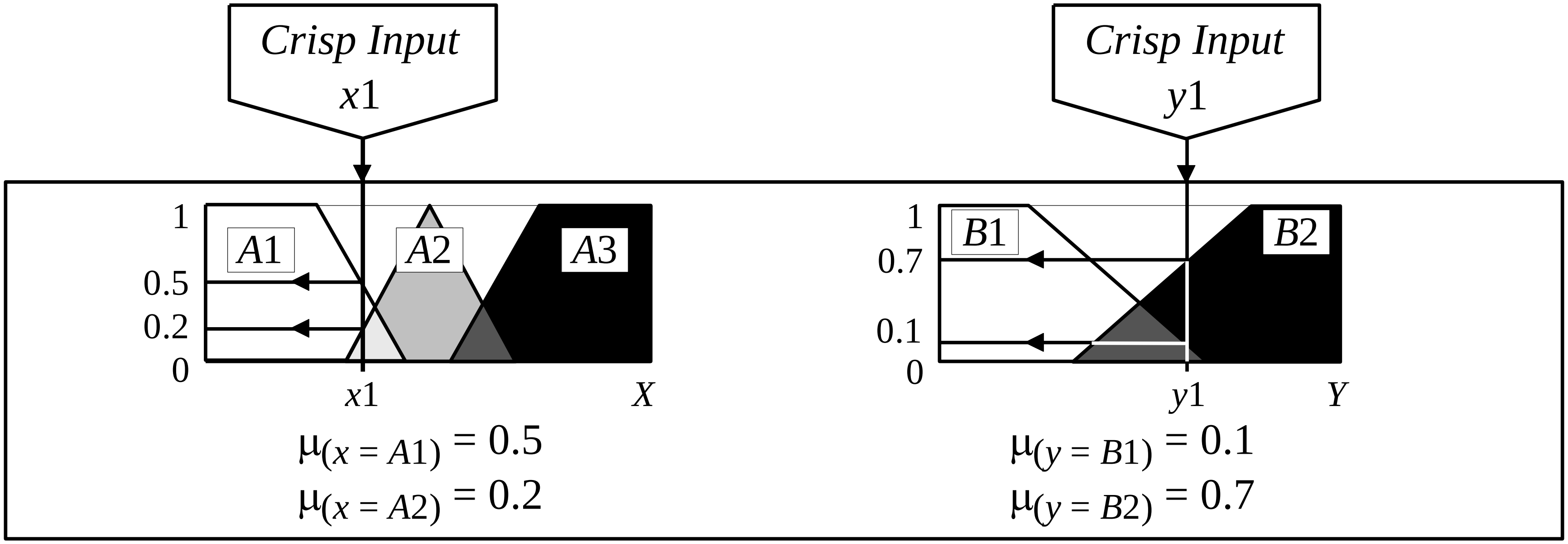
Rule: 1
IF *project_funding* is *adequate*
OR *project_staffing* is *small*
THEN *risk* is *low*

Rule: 2
IF *project_funding* is *marginal*
AND *project_staffing* is *large*
THEN *risk* is *normal*

Rule: 3
IF *project_funding* is *inadequate*
THEN *risk* is *high*

Step 1: Fuzzification

The first step is to take the crisp inputs, *x*1 and *y*1 (*project funding* and *project staffing*), and determine the degree to which these inputs belong to each of the appropriate fuzzy sets.



$\mu_{(x\,=\,A1)} = 0.5$
$\mu_{(x\,=\,A2)} = 0.2$

$\mu_{(y\,=\,B1)} = 0.1$
$\mu_{(y\,=\,B2)} = 0.7$

Step 2: Rule Evaluation

The second step is to take the fuzzified inputs, $\mu_{(x=A1)} = 0.5$, $\mu_{(x=A2)} = 0.2$, $\mu_{(y=B1)} = 0.1$ and $\mu_{(y=B2)} = 0.7$ and apply them to the antecedents of the fuzzy rules.

If a given fuzzy rule has multiple antecedents, the fuzzy operator (AND or OR) is used to obtain a single number that represents the result of the antecedent evaluation.

This number (the truth value) is then applied to the consequent membership function.

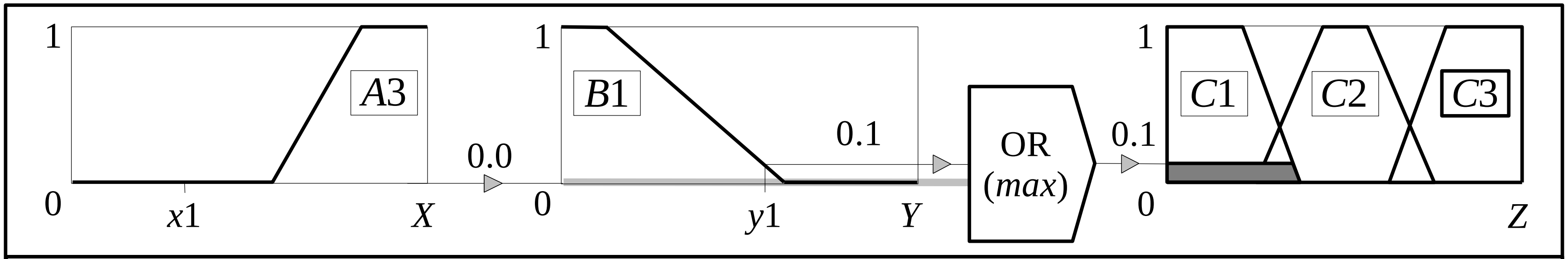
OR fuzzy operation. (Union)

$\mu_{A \cup B}(x) = max\ [\mu_A(x), \mu_B(x)]$

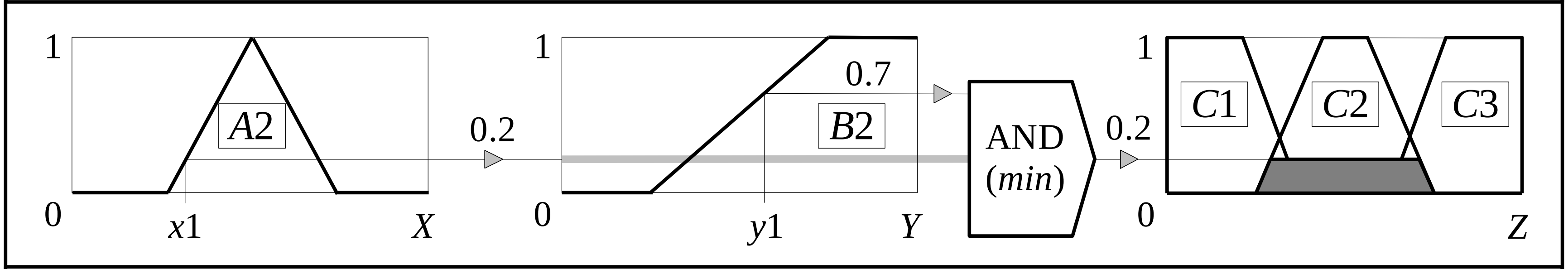
AND fuzzy operation. (Intersection)

$\mu_{A \cap B}(x) = min\ [\mu_A(x), \mu_B(x)]$

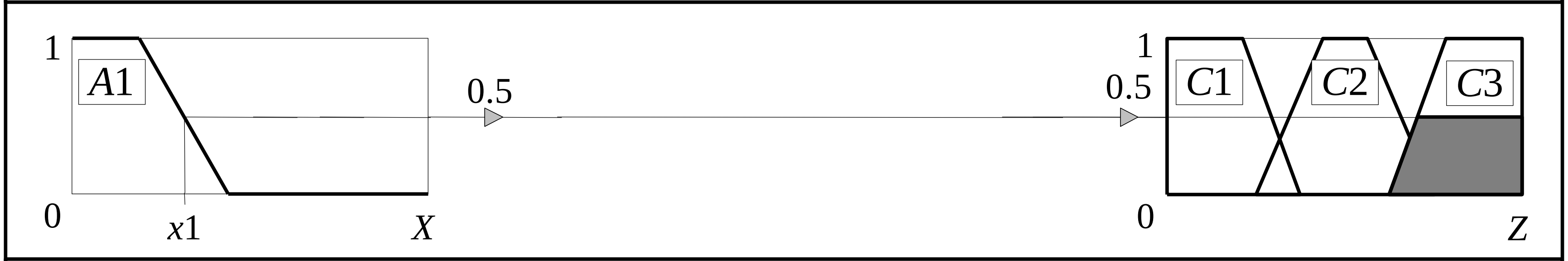
Mamdani-style rule evaluation



Rule 1: IF *x* is *A*3 (0.0) OR *y* is *B*1 (0.1) THEN *z* is *C*1 (0.1)



Rule 2: IF *x* is *A*2 (0.2) AND *y* is *B*2 (0.7) THEN *z* is *C*2 (0.2)



Rule 3: IF *x* is *A*1 (0.5) THEN *z* is *C*3 (0.5)

Now the result of the antecedent evaluation can be applied to the membership function of the consequent.

The most common method of correlating the rule consequent with the truth value of the rule antecedent is to cut the consequent membership function at the level of the antecedent truth.

This method is called clipping.

Since the top of the membership function is sliced, the clipped fuzzy set loses some information.

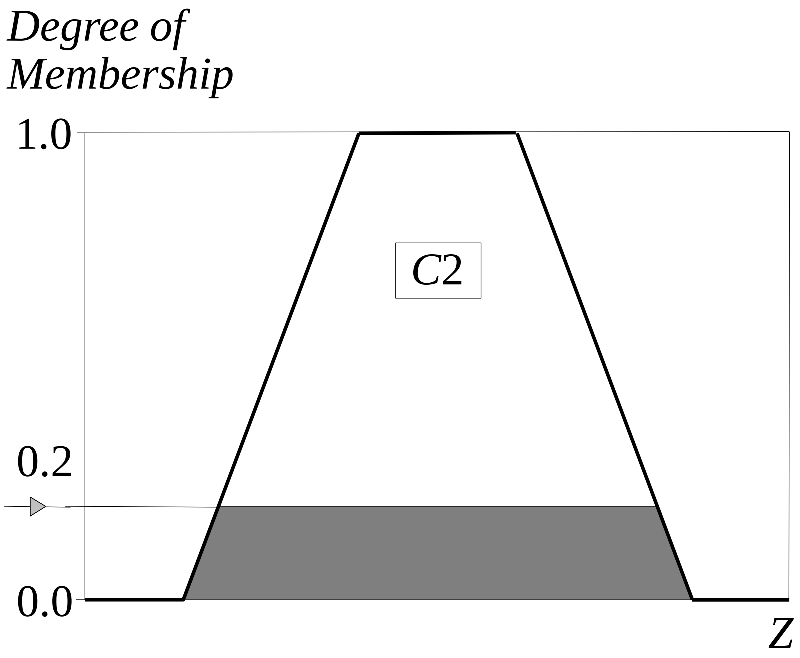
However, clipping is still often preferred because it involves less complex and faster mathematics, and generates an aggregated output surface that is easier to defuzzify.

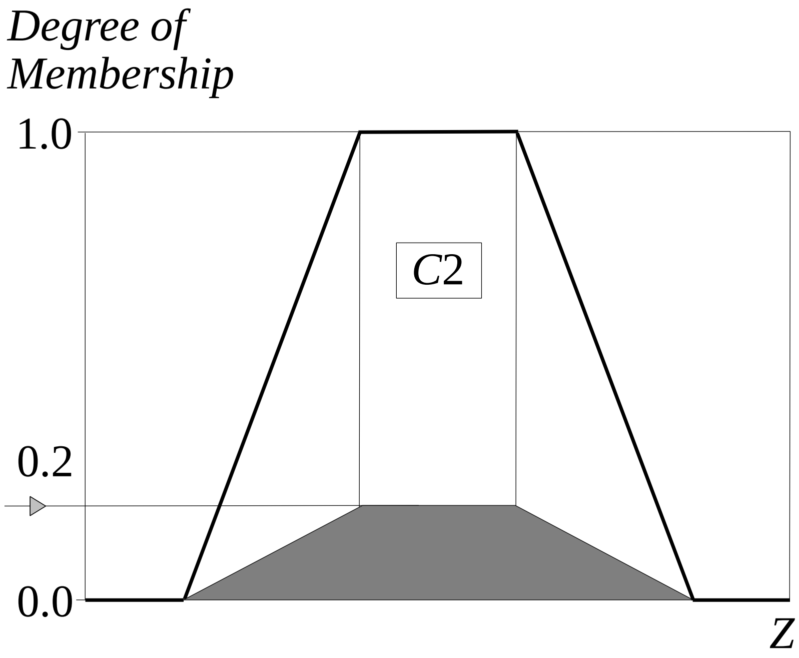
While clipping is a frequently used method, scaling offers a better approach for preserving the original shape of the fuzzy set.

The original membership function of the rule consequent is adjusted by multiplying all its membership degrees by the truth value of the rule antecedent.

This method, which generally loses less information, can be very useful in fuzzy expert systems.

Clipped and scaled membership functions





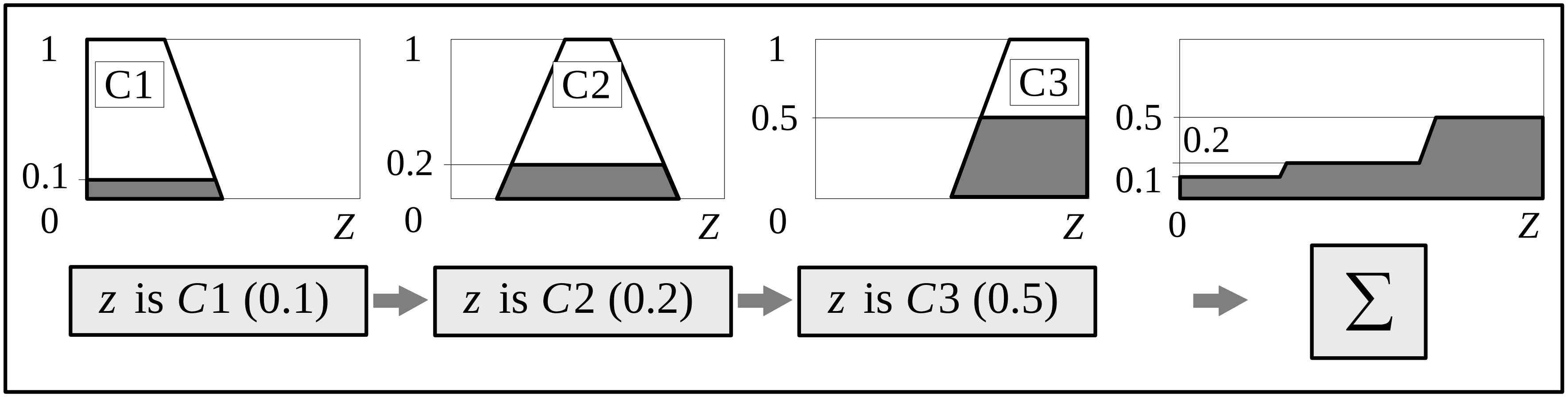
Step 3: Aggregation of the rule outputs

Aggregation is the process of unification of the outputs of all rules.

We take the membership functions of all rule consequents previously clipped or scaled and combine them into a single fuzzy set.

The input of the aggregation process is the list of clipped or scaled consequent membership functions, and the output is one fuzzy set for each output variable.

Aggregation of the rule outputs



Step 4: Defuzzification

The last step in the fuzzy inference process is defuzzification.

Fuzziness helps us to evaluate the rules, but the final output of a fuzzy system has to be a crisp number.

The input for the defuzzification process is the aggregate output fuzzy set and the output is a single number.

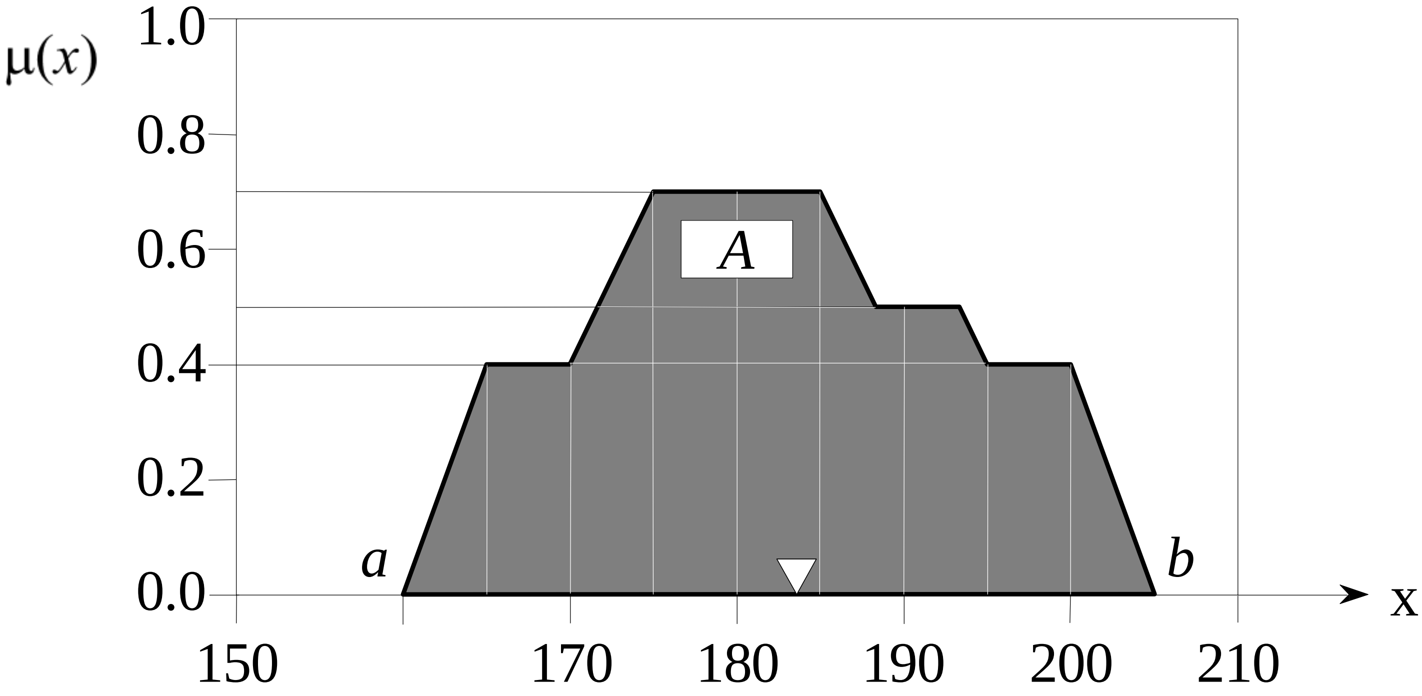
There are several defuzzification methods, but probably the most popular one is the centroid technique.

It finds the point where a vertical line would slice the aggregate set into two equal masses. Mathematically this centre of gravity (COG) can be expressed as:

$$COG = \frac{\int_a^b \mu_A(x)\, x\, dx}{\int_a^b \mu_A(x)\, dx}$$

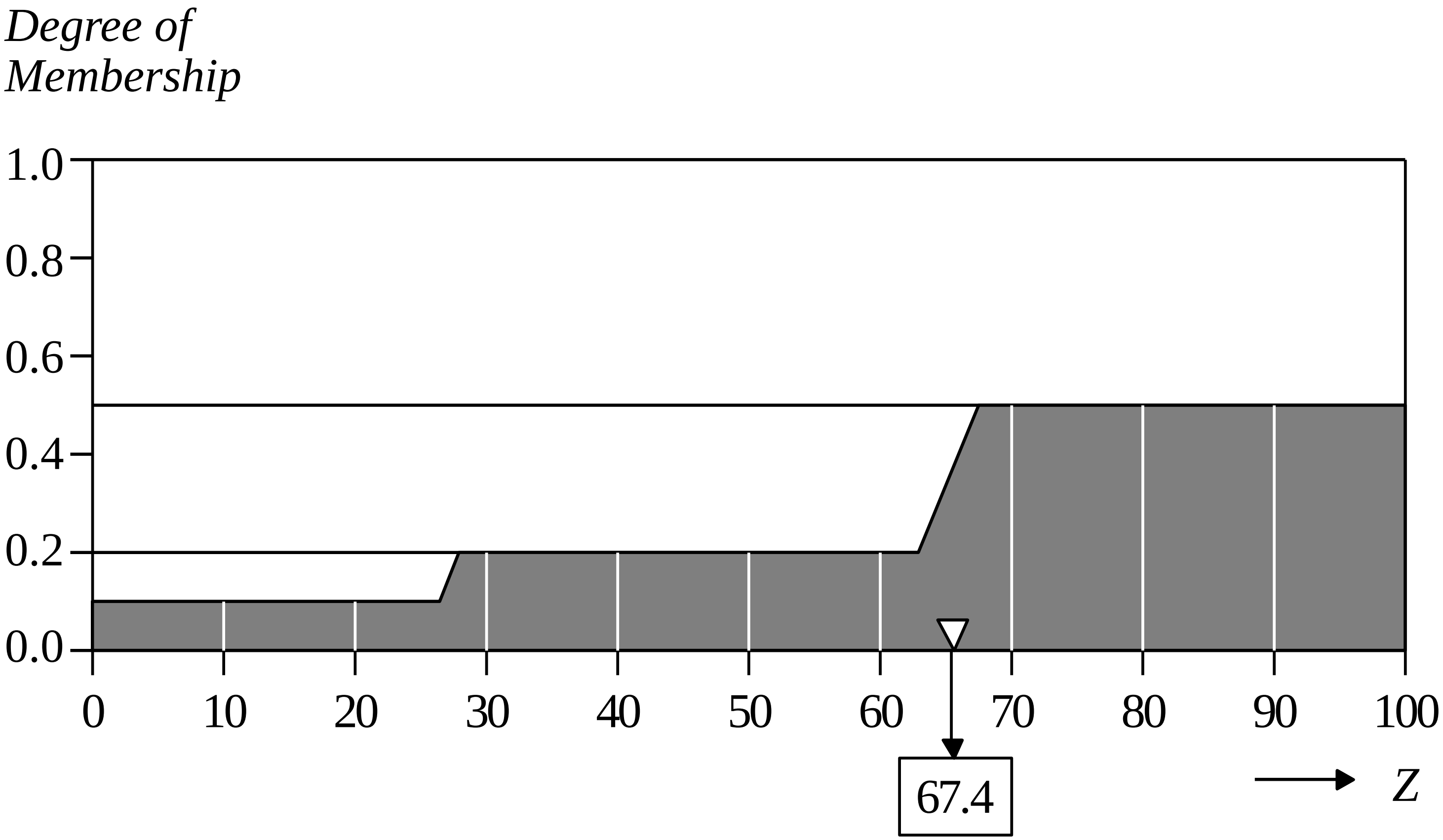
Centroid defuzzification method finds a point representing the centre of gravity of the fuzzy set, *A*, on the interval, *ab*.

A reasonable estimate can be obtained by calculating it over a sample of points.



Centre of gravity (COG):

$$COG = \frac{(0+10+20)\times\ 0.1+(30+40+50+60)\times\ 0.2+(70+80+90+100)\times\ 0.5}{0.1+0.1+0.1+0.2+0.2+0.2+0.2+0.5+0.5+0.5+0.5} = 67.4$$



Sugeno fuzzy inference

- Mamdani-style inference, as we have just seen, requires us to find the centroid of a two-dimensional shape by integrating across a continuously varying function. In general, this process is not computationally efficient.
- Michio Sugeno suggested to use a single spike, a *singleton*, as the membership function of the rule consequent. A singleton, or more precisely a fuzzy singleton, is a fuzzy set with a membership function that is unity at a single particular point on the universe of discourse and zero everywhere else.

Sugeno-style fuzzy inference is very similar to the Mamdani method.

Sugeno changed only a rule consequent.

Instead of a fuzzy set, he used a mathematical function of the input variable.

The format of the Sugeno-style fuzzy rule is

IF $\quad x$ is A

AND $\quad y$ is B

THEN $\quad z$ is $f\,(x, y)$

where x, y and z are linguistic variables; A and B are fuzzy sets on universe of discourses X and Y, respectively; and $f\,(x, y)$ is a mathematical function.

The most commonly used zero-order Sugeno fuzzy model applies fuzzy rules in the following form:

IF x is A

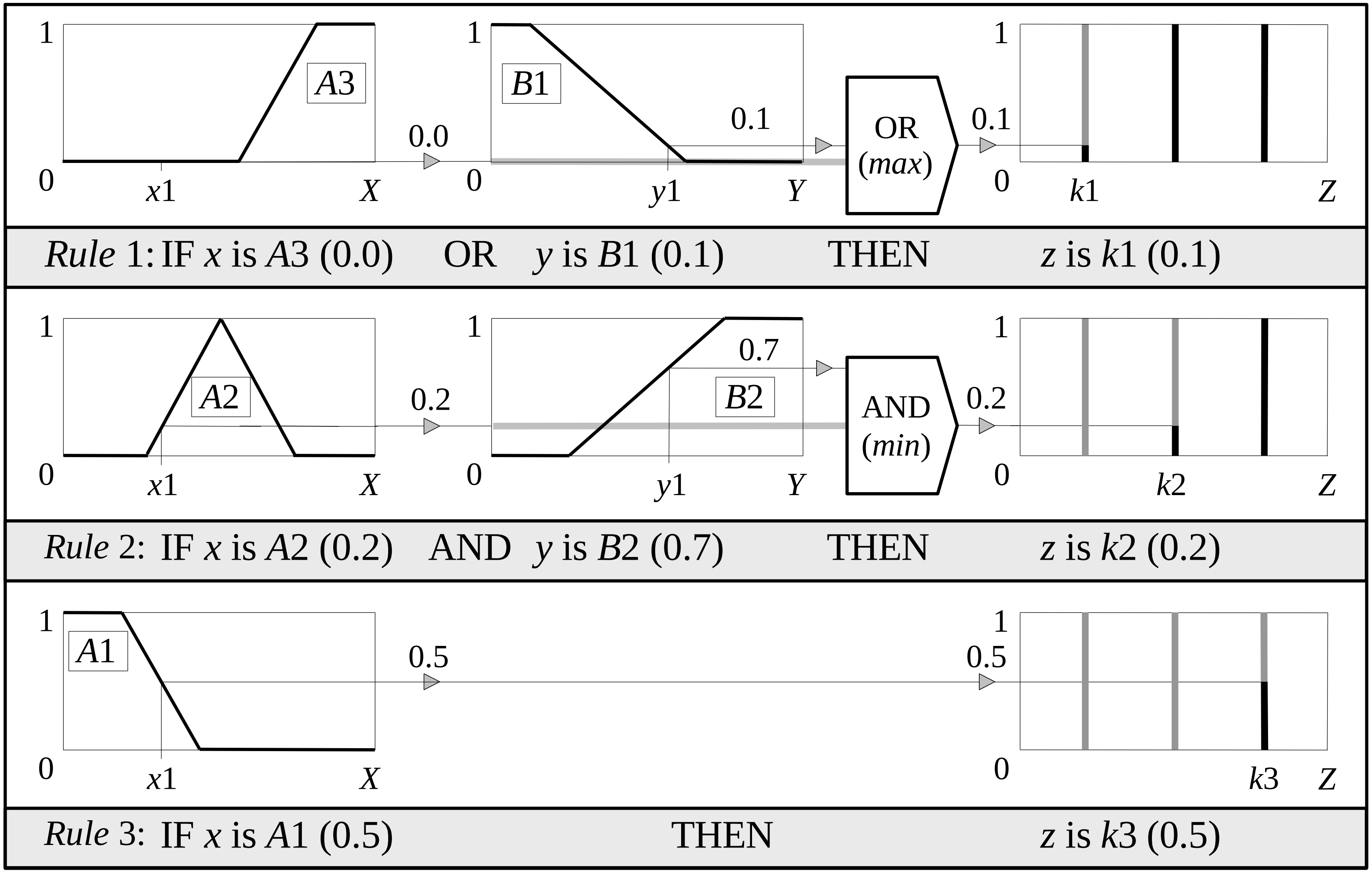
AND y is B

THEN z is k

where k is a constant.

In this case, the output of each fuzzy rule is constant. All consequent membership functions are represented by singleton spikes.

Sugeno-style rule evaluation

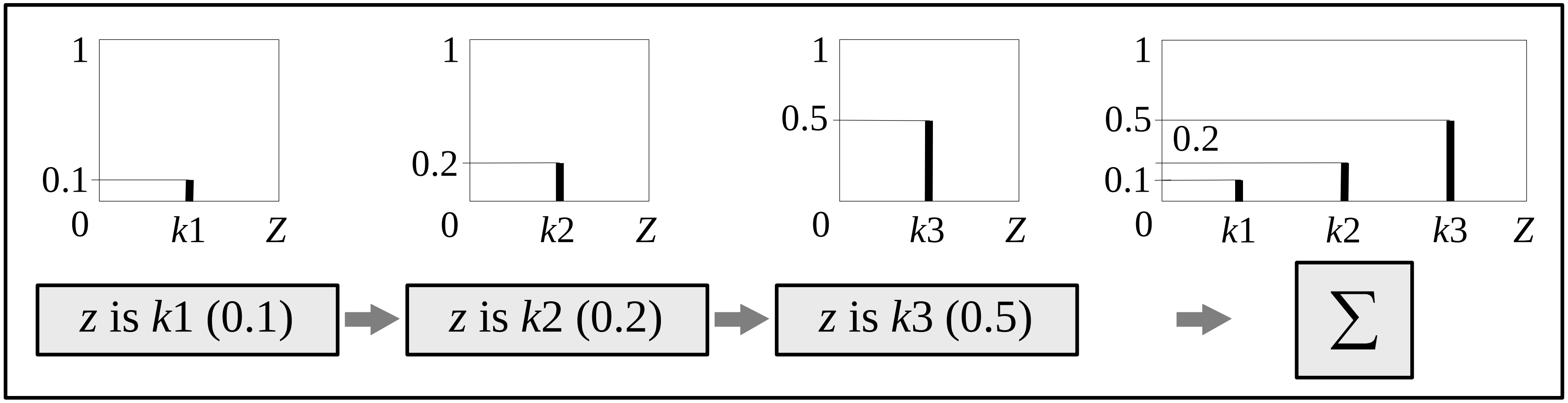


Rule 1: IF *x* is *A*3 (0.0) OR *y* is *B*1 (0.1) THEN *z* is *k*1 (0.1)

Rule 2: IF *x* is *A*2 (0.2) AND *y* is *B*2 (0.7) THEN *z* is *k*2 (0.2)

Rule 3: IF *x* is *A*1 (0.5) THEN *z* is *k*3 (0.5)

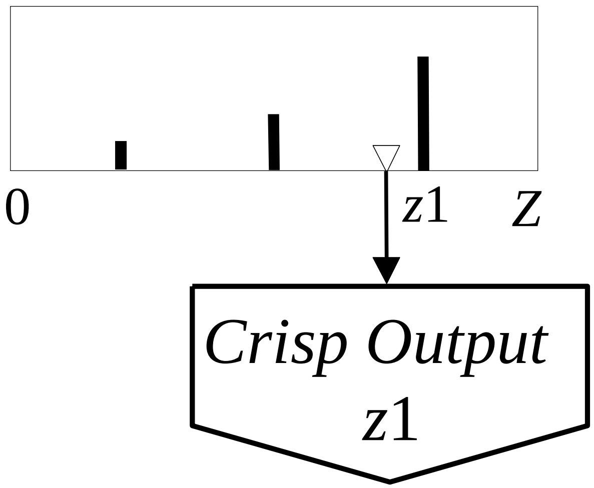
Sugeno-style aggregation of the rule outputs



Weighted average (WA):

$$WA = \frac{\mu(k1) \times k1 + \mu(k2) \times k2 + \mu(k3) \times k3}{\mu(k1) + \mu(k2) + \mu(k3)} = \frac{0.1 \times 20 + 0.2 \times 50 + 0.5 \times 80}{0.1 + 0.2 + 0.5} = 65$$

Sugeno-style defuzzification



Crisp Output
*z*1

How to make a decision on which method to apply – Mamdani or Sugeno?

- Mamdani method is widely accepted for capturing expert knowledge. It allows us to describe the expertise in more intuitive, more human-like manner. However, Mamdani-type fuzzy inference entails a substantial computational burden.
- On the other hand, Sugeno method is computationally effective and works well with optimisation and adaptive techniques, which makes it very attractive in control problems, particularly for dynamic nonlinear systems.

Building a **FUZZY EXPERT SYSTEM**: case study

- A service center keeps spare parts and repairs failed ones.
- A customer brings a failed item and receives a spare of the same type.
- Failed parts are repaired, placed on the shelf, and thus become spares.
- The objective here is to advise a manager of the service centre on certain decision policies to keep the customers satisfied.

Process of developing a fuzzy expert system

1. Specify the problem and define linguistic variables.
2. Determine fuzzy sets.
3. Elicit and construct fuzzy rules.
4. Encode the fuzzy sets, fuzzy rules and procedures to perform fuzzy inference into the expert system.
5. Evaluate and tune the system.

Step 1: Specify the problem and define linguistic variables

There are four main linguistic variables:

- average waiting time (mean delay) m
- repair utilisation factor of the service centre ρ
- number of servers s
- initial number of spare parts n

Linguistic variables and their ranges

Linguistic Variable: *Mean Delay, m*

Linguistic Value	Notation	Numerical Range (normalised)
Very Short	VS	[0, 0.3]
Short	S	[0.1, 0.5]
Medium	M	[0.4, 0.7]

Linguistic Variable: *Number of Servers, s*

Linguistic Value	Notation	Numerical Range (normalised)
Small	S	[0, 0.35]
Medium	M	[0.30, 0.70]
Large	L	[0.60, 1]

Linguistic Variable: *Repair Utilisation Factor*, ρ

Linguistic Value	Notation	Numerical Range
Low	L	[0, 0.6]
Medium	M	[0.4, 0.8]
High	H	[0.6, 1]

Linguistic Variable: *Number of Spares, n*

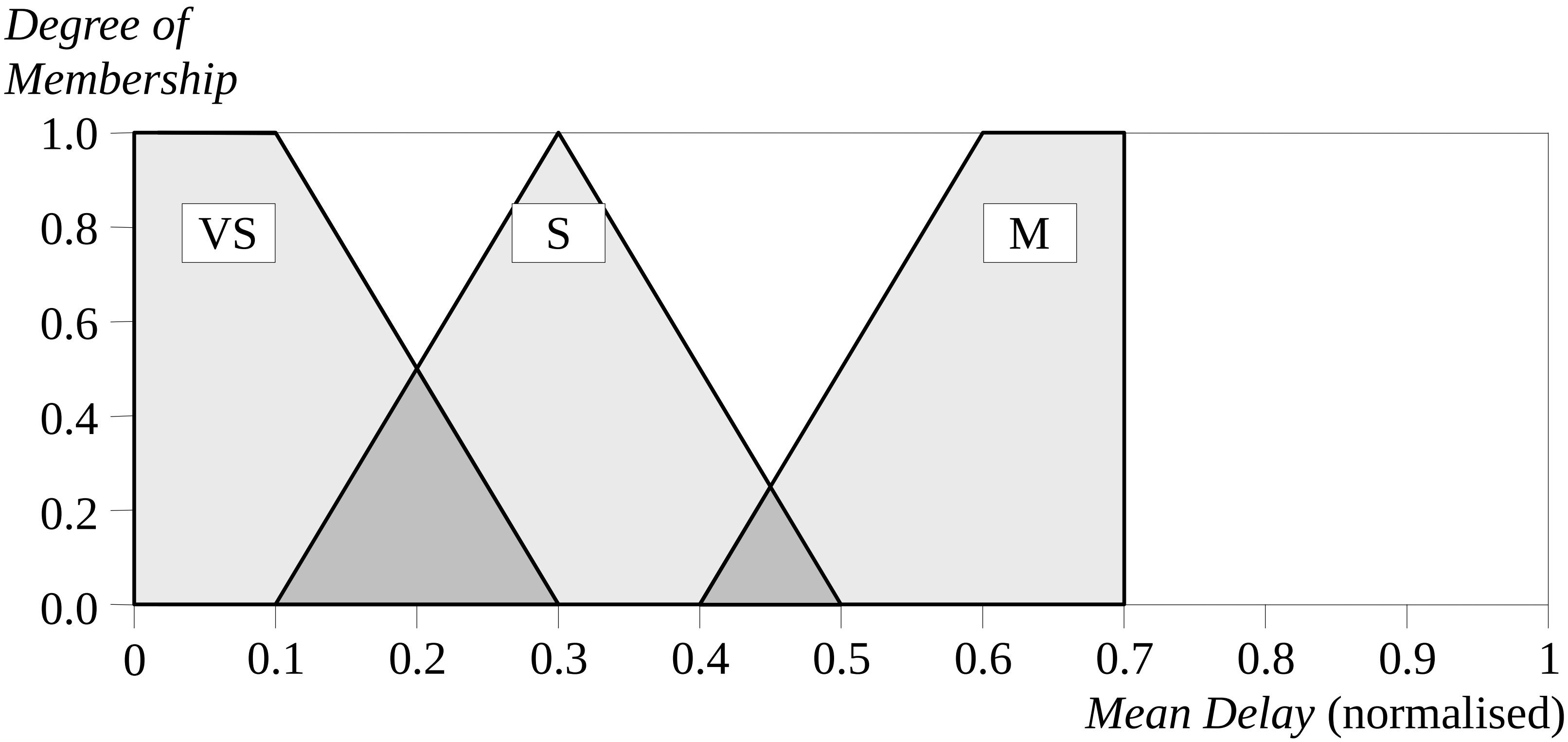
Linguistic Value	Notation	Numerical Range (normalised)
Very Small	VS	[0, 0.30]
Small Rather	S	[0, 0.40]
Small Medium	RS	[0.25, 0.45]
Rather Large	M	[0.30, 0.70]
Large	RL	[0.55, 0.75]
Very Large	L	[0.60, 1]
	VL	[0.70, 1]

Step 2: Determine fuzzy sets

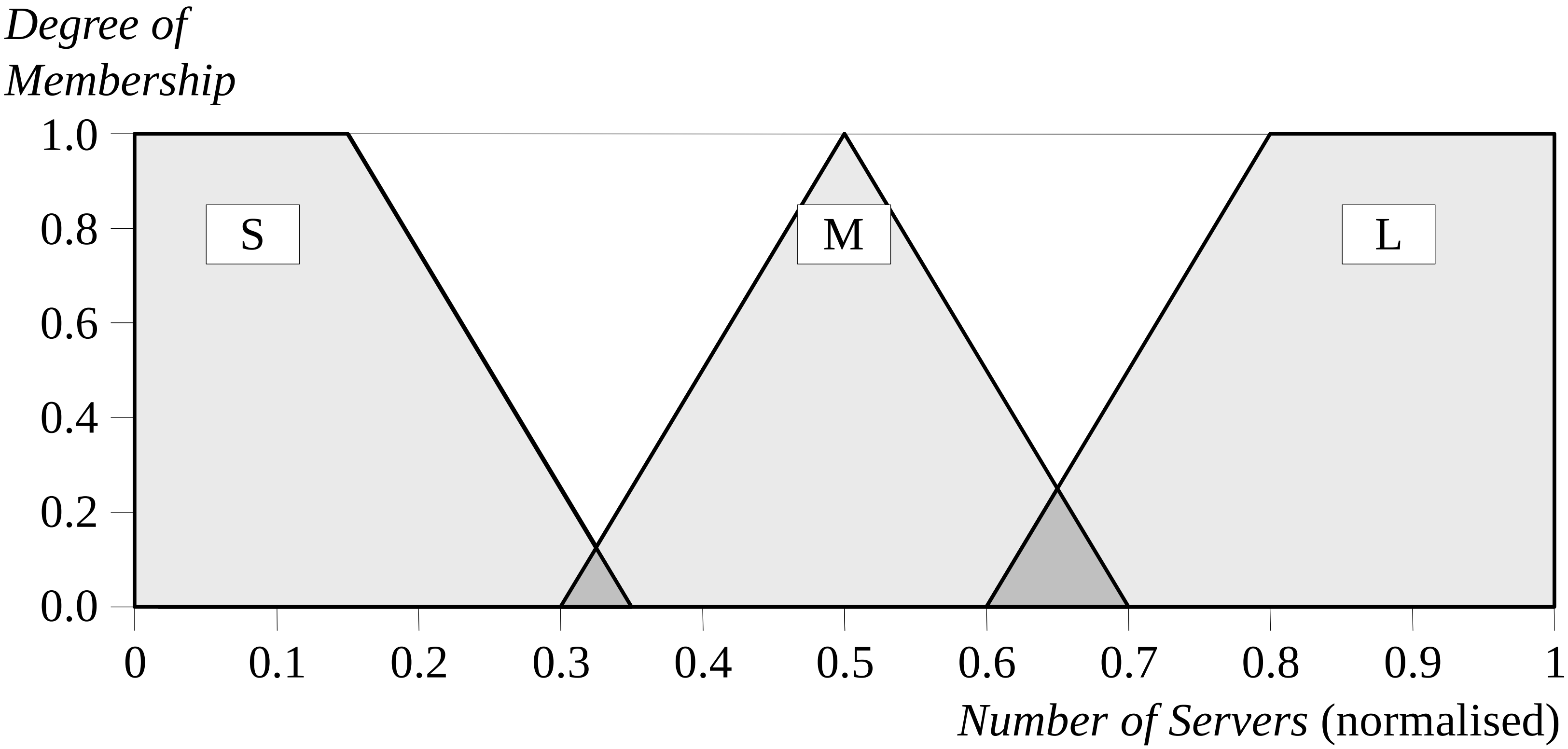
Fuzzy sets can have a variety of shapes.

However, a triangle or a trapezoid can often provide an adequate representation of the expert knowledge, and at the same time, significantly simplifies the process of computation.

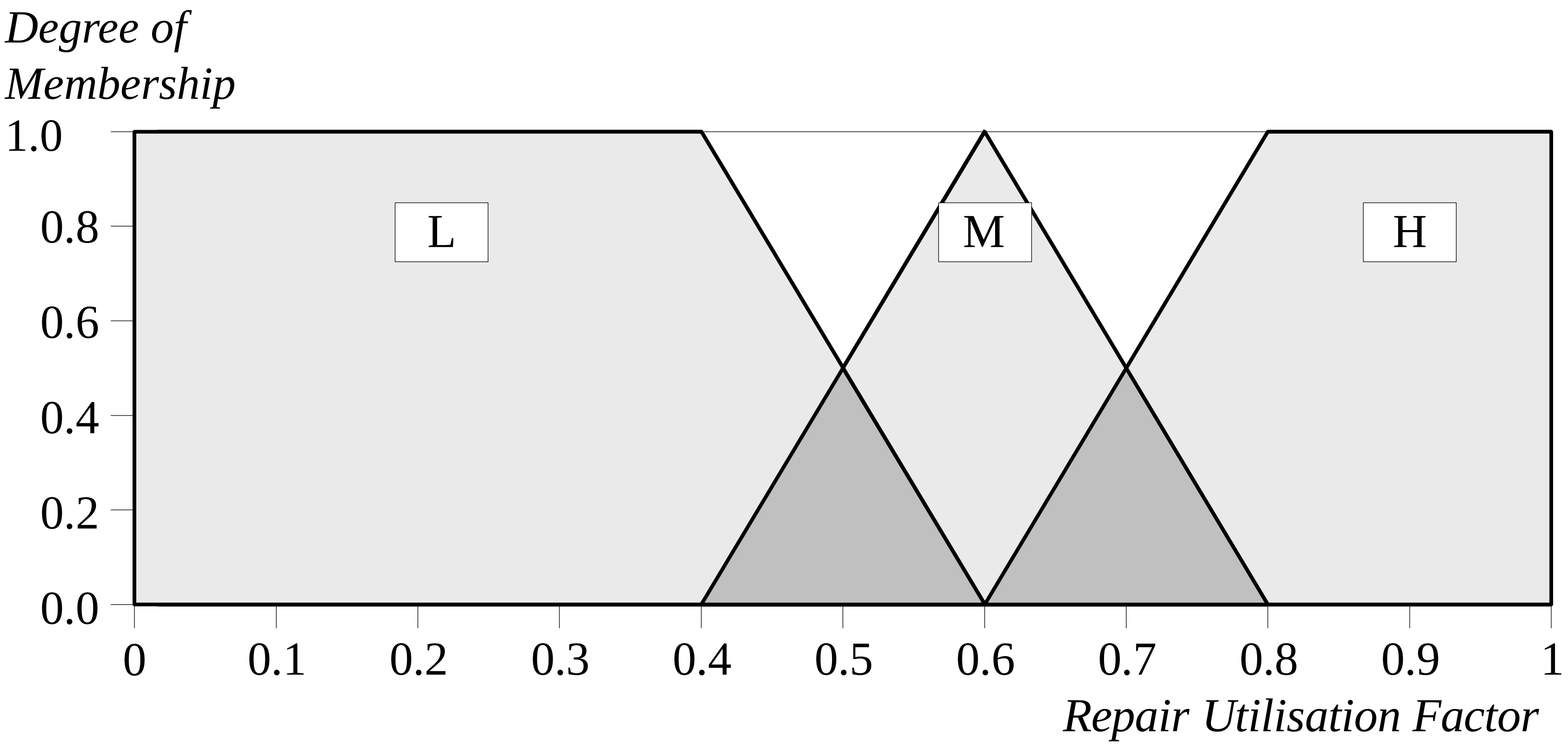
Fuzzy sets of *Mean Delay m*



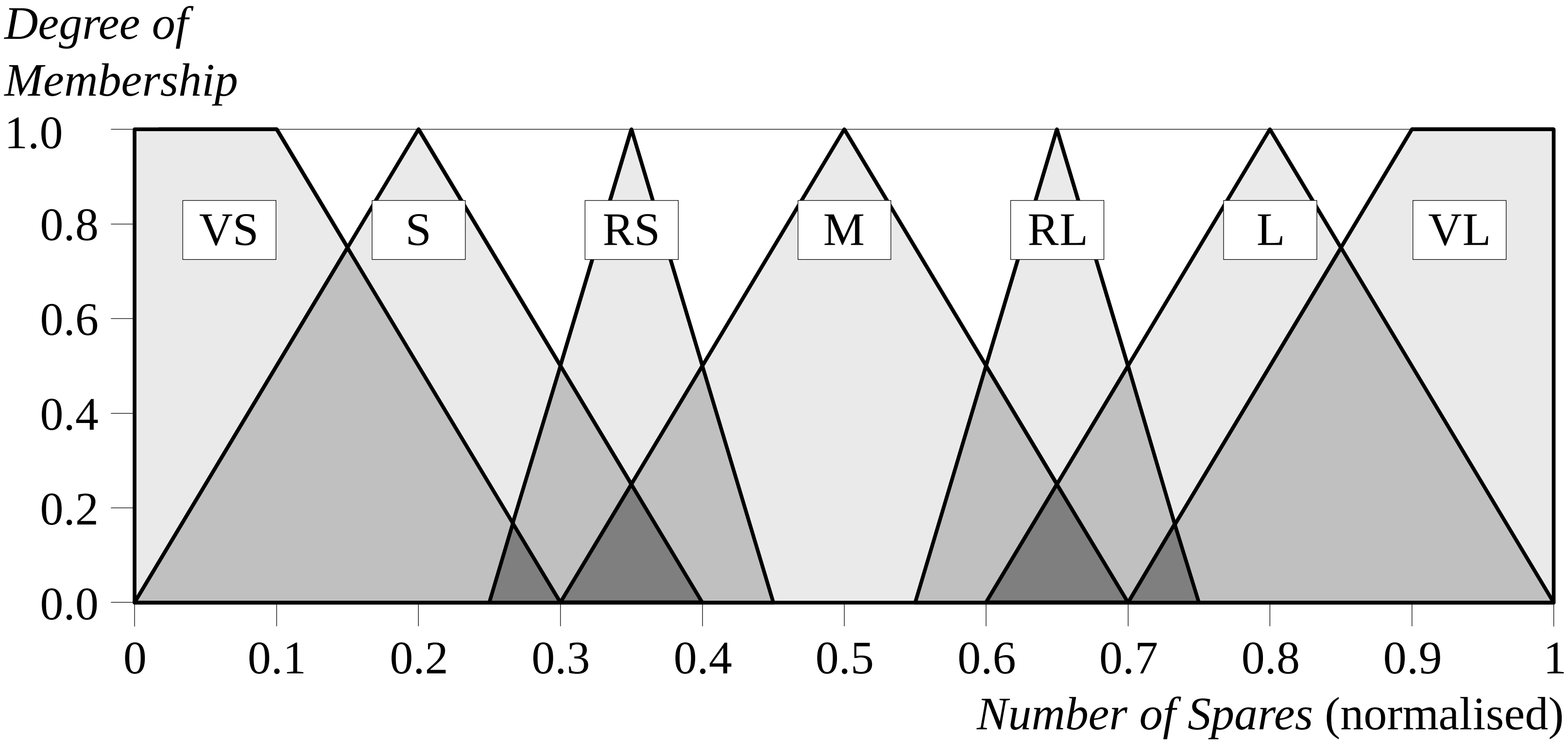
Fuzzy sets of *Number of Servers s*



Fuzzy sets of *Repair Utilisation Factor* ρ



Fuzzy sets of *Number of Spares n*



Step 3: Elicit and construct fuzzy rules

To accomplish this task, we might ask the expert to describe how the problem can be solved using the fuzzy linguistic variables defined previously.

Required knowledge also can be collected from other sources such as books, computer databases, flow diagrams and observed human behaviour.

Rule Base 1

1.If (utilisation_factor is L) then (number_of_spares is S)
2.If (utilisation_factor is M) then (number_of_spares is M)
3.If (utilisation_factor is H) then (number_of_spares is L)

4.If (mean_delay is VS) and (number_of_servers is S) then (number_of_spares is VL)
5.If (mean_delay is S) and (number_of_servers is S) then (number_of_spares is L)
6.If (mean_delay is M) and (number_of_servers is S) then (number_of_spares is M)

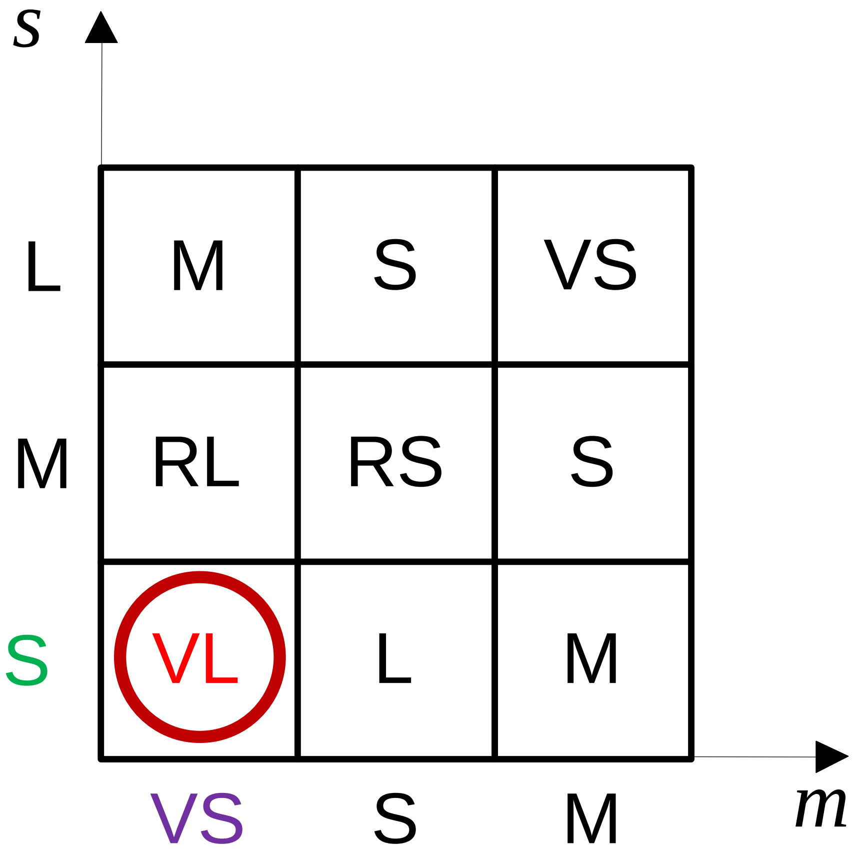
7.If (mean_delay is VS) and (number_of_servers is M) then (number_of_spares is RL)
8.If (mean_delay is S) and (number_of_servers is M) then (number_of_spares is RS)
9.If (mean_delay is M) and (number_of_servers is M) then (number_of_spares is S)

10.If (mean_delay is VS) and (number_of_servers is L) then (number_of_spares is M)

11.If (mean_delay is S) and (number_of_servers is L) then (number_of_spares is S)

12.If (mean_delay is M) and (number_of_servers is L) then (number_of_spares is VS)

The square fuzzy analytical method (FAM) representation



s				
L	M	S	VS	
M	RL	RS	S	
S	VL	L	M	
	VS	S	M	*m*

- average waiting time (mean delay) *m*
- repair utilisation factor of the service centre ρ
- number of servers *s*
- initial number of spare parts *n*

Example: Rule Base 1, Rule 4.

If (mean_delay is VS) and (number_of_servers is S)

then (number_of_spares is VL)

The rule table (Rule Base 2)

Rule	m	s	ρ	n	Rule	m	s	ρ	n	Rule	m	s	ρ	n
1	**VS**	**S**	**L**	**VS**	10	VS	S	M	S	19	VS	S	H	VL
2	S	S	L	VS	11	S	S	M	VS	20	S	S	H	L
3	M	S	L	VS	12	M	S	M	VS	21	M	S	H	M
4	VS	M	L	VS	13	VS	M	M	RS	22	VS	M	H	M
5	S	M	L	VS	14	S	M	M	S	23	S	M	H	M
6	M	M	L	VS	15	M	M	M	VS	24	M	M	H	S
7	VS	L	L	S	16	VS	L	M	M	25	VS	L	H	RL
8	S	L	L	S	17	S	L	M	RS	26	S	L	H	M
9	M	L	L	VS	18	M	L	M	S	27	M	L	H	RS

Rule 1: **If** (m is VS) and (s is S) and (ρis L) **Then** (n is VS)

where

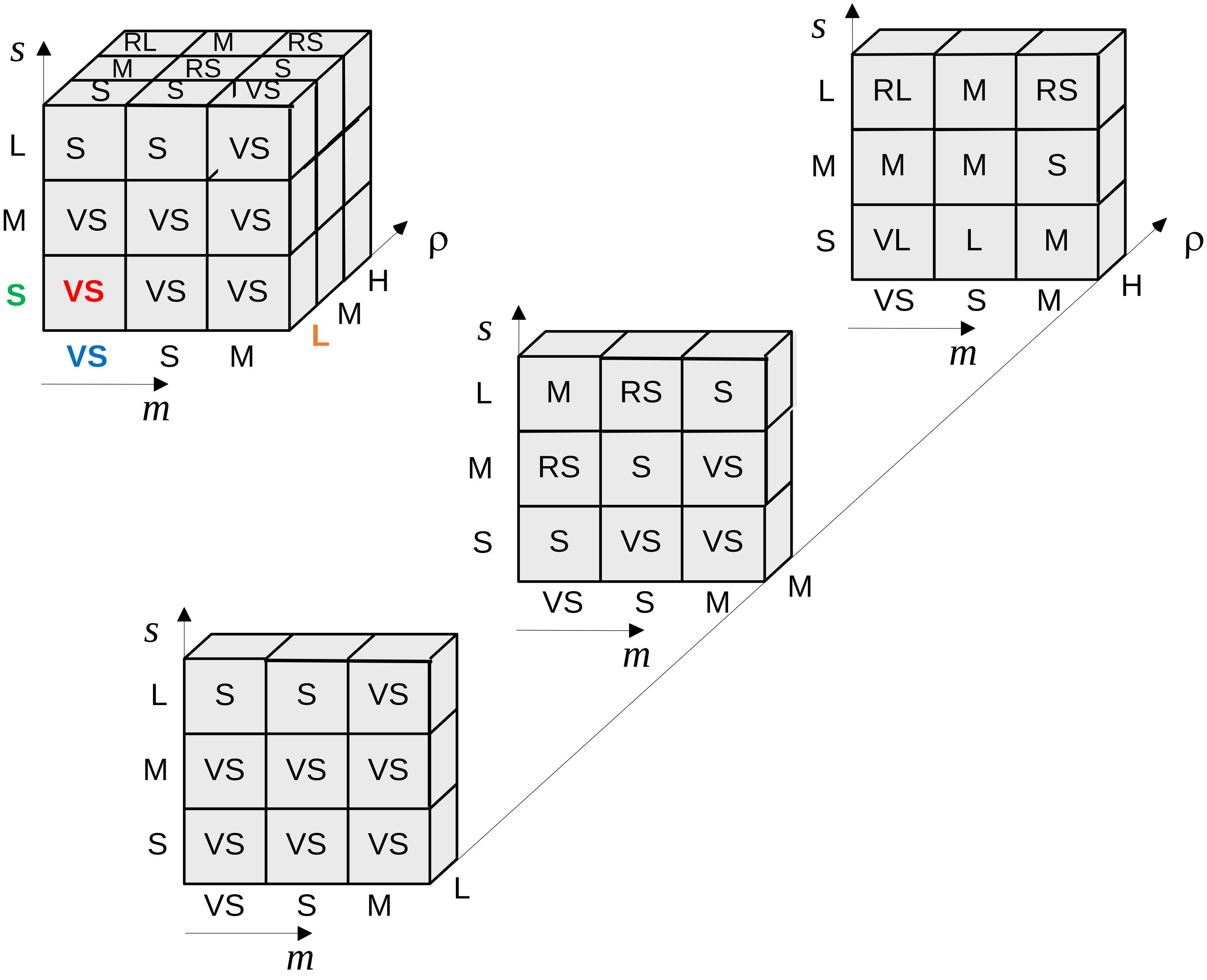
m is average waiting time (mean delay)

ρ is repair utilisation factor of the service centre

s is number of servers

n is initial number of spare parts

Cube FAM of Rule Base 2



L

	VS	S	M
L	S	S	VS
M	VS	VS	VS
S	VS	VS	VS

M

	VS	S	M
L	M	RS	S
M	RS	S	VS
S	S	VS	VS

H

	VS	S	M
L	RL	M	RS
M	M	M	S
S	VL	L	M

Rule 1:

If (m is **VS**)

and (s is **S**)

and (ρ is **L**)

Then (*n* is **VS**)

Step 4: Encode the fuzzy sets, fuzzy rules and procedures to perform fuzzy inference into the expert system

To accomplish this task, we may choose one of two options:

- ✓ to build our system using a programming language such as C/C++ or Pascal,

- ✓ or to apply a fuzzy logic development tool such as MATLAB Fuzzy Logic Toolbox or Fuzzy Knowledge Builder.

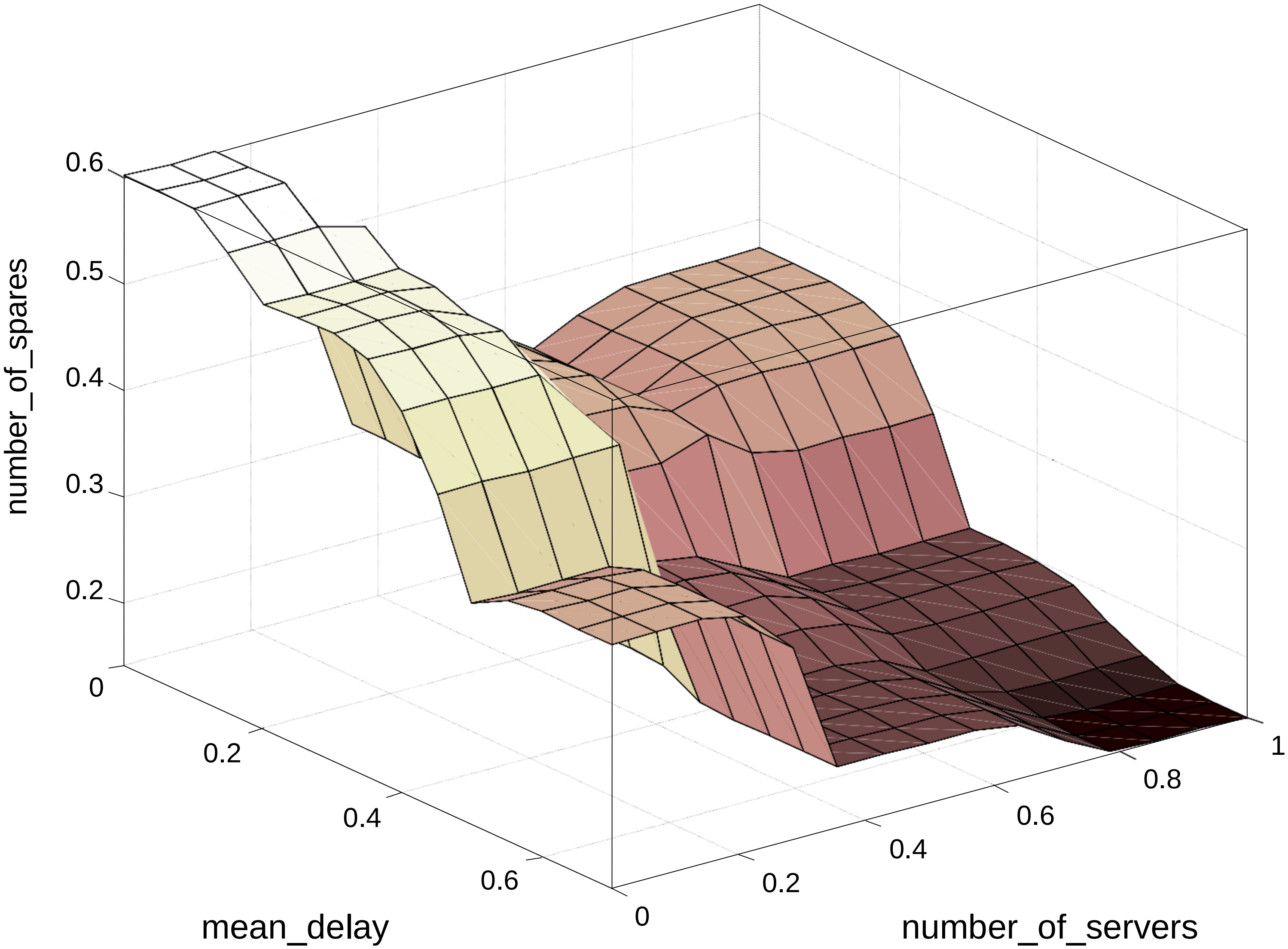
Step 5: Evaluate and tune the system

The last, and the most laborious, task is to evaluate and tune the system. We want to see whether our fuzzy system meets the requirements specified at the beginning.

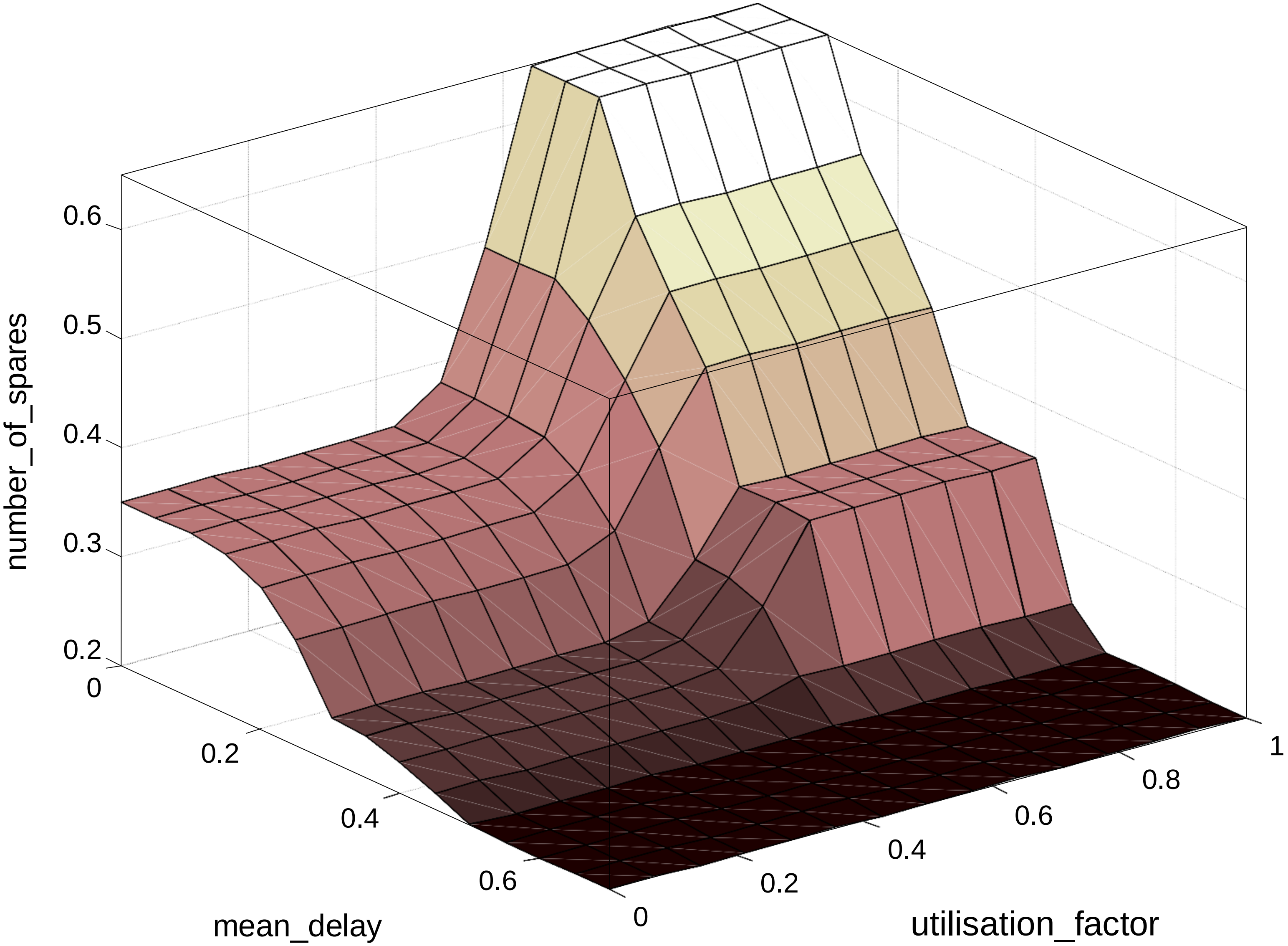
Several test situations depend on the mean delay, number of servers and repair utilisation factor.

The Fuzzy Logic Toolbox can generate surface to help us analyse the system's performance.

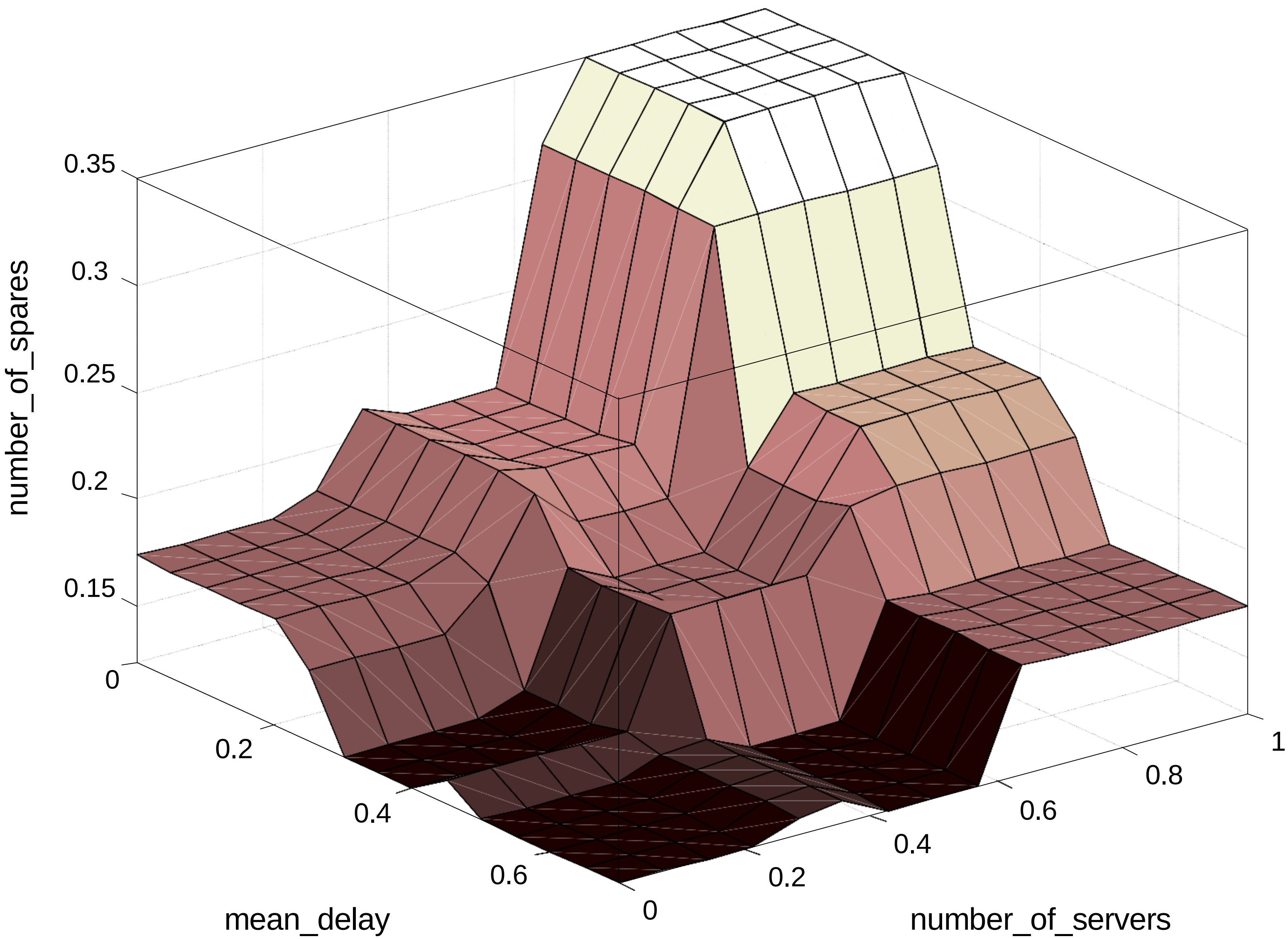
Three-dimensional plots for Rule Base 1



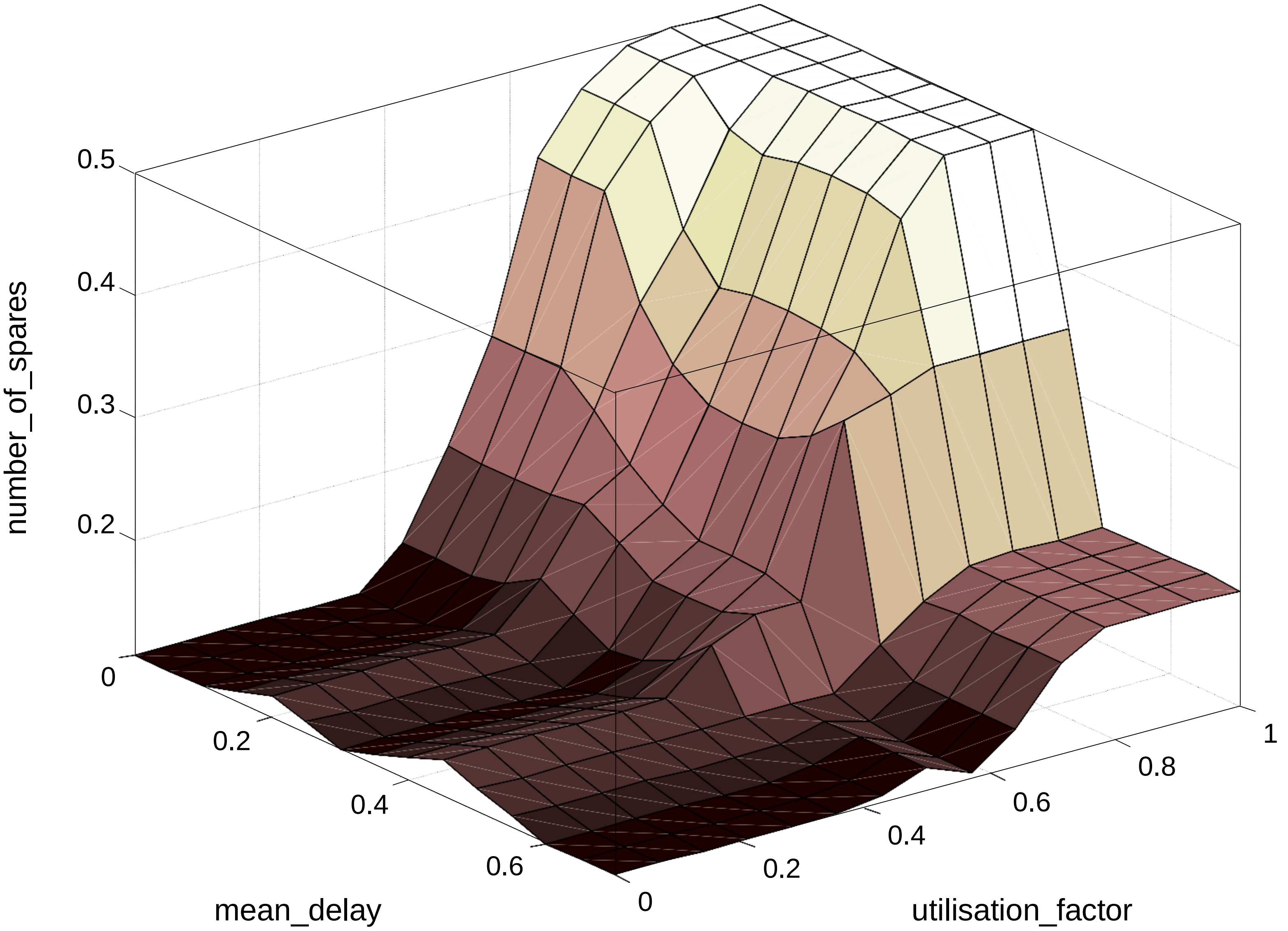
Three-dimensional plots for Rule Base 1



Three-dimensional plots for Rule Base 2



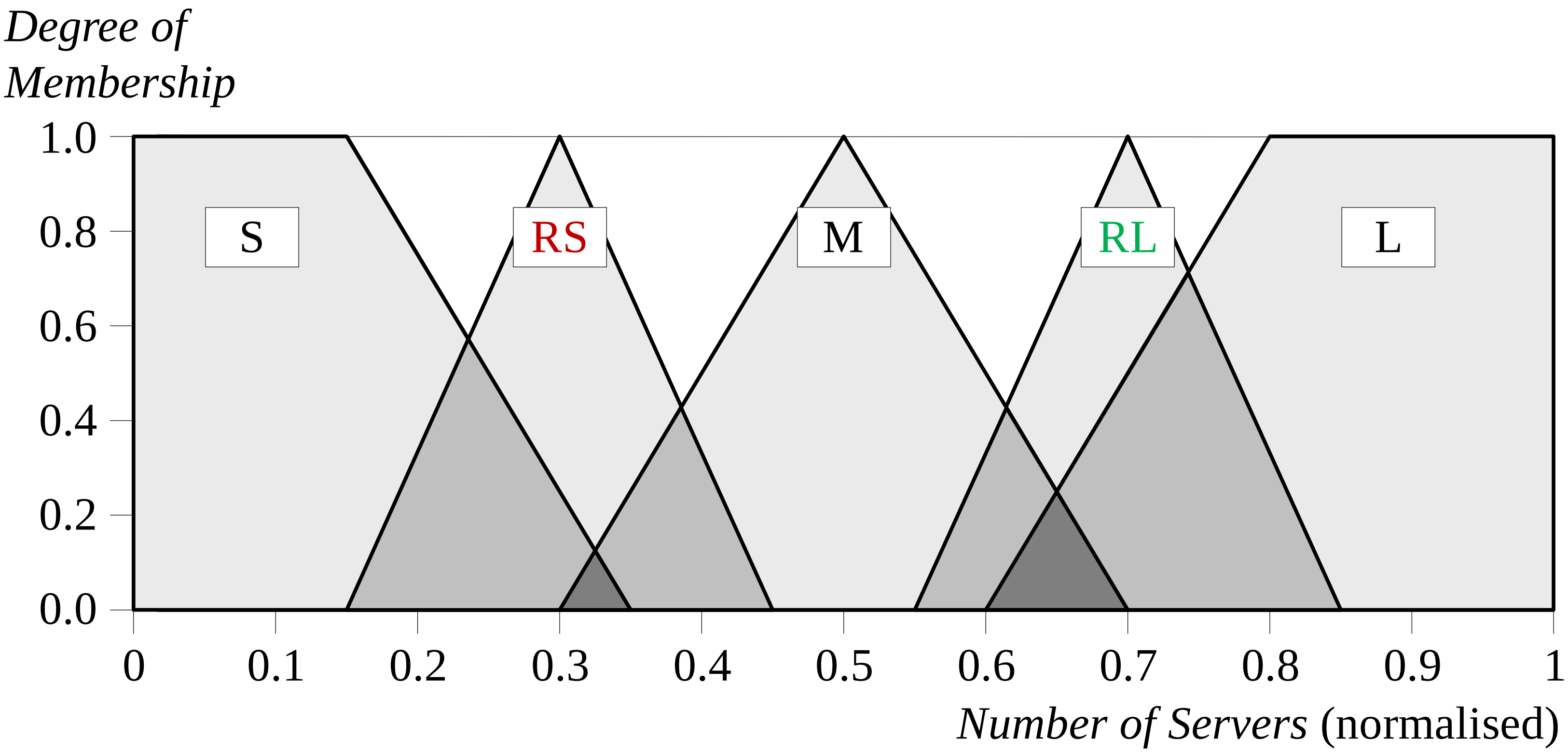
Three-dimensional plots for Rule Base 2



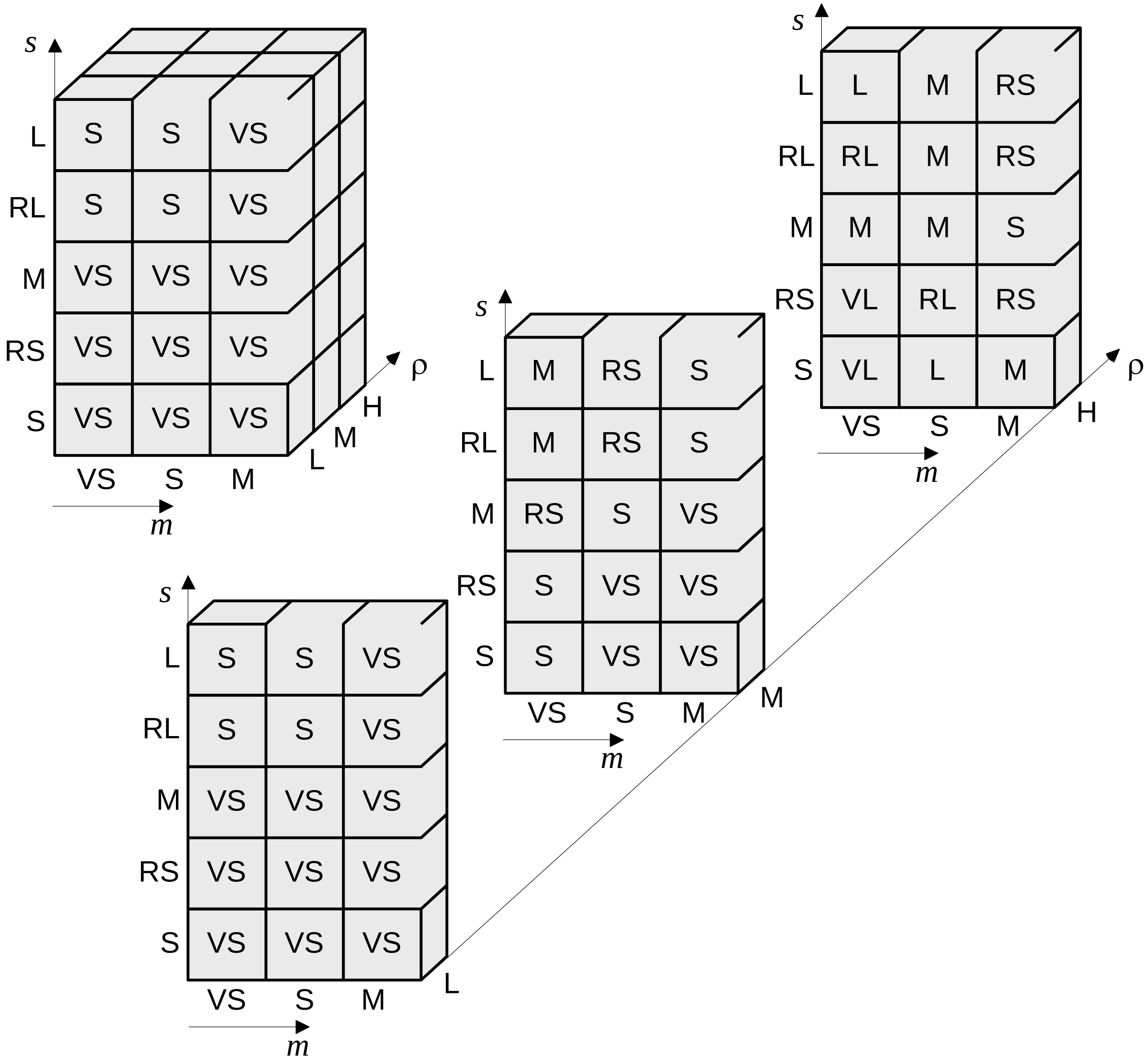
However, even now, the expert might not be satisfied with the system performance.

To improve the system performance, we may use additional sets – *Rather Small* and *Rather Large* – on the universe of discourse *Number of Servers*, and then extend the rule base.

Modified fuzzy sets of *Number of Servers s*



Cube FAM of Rule Base 3



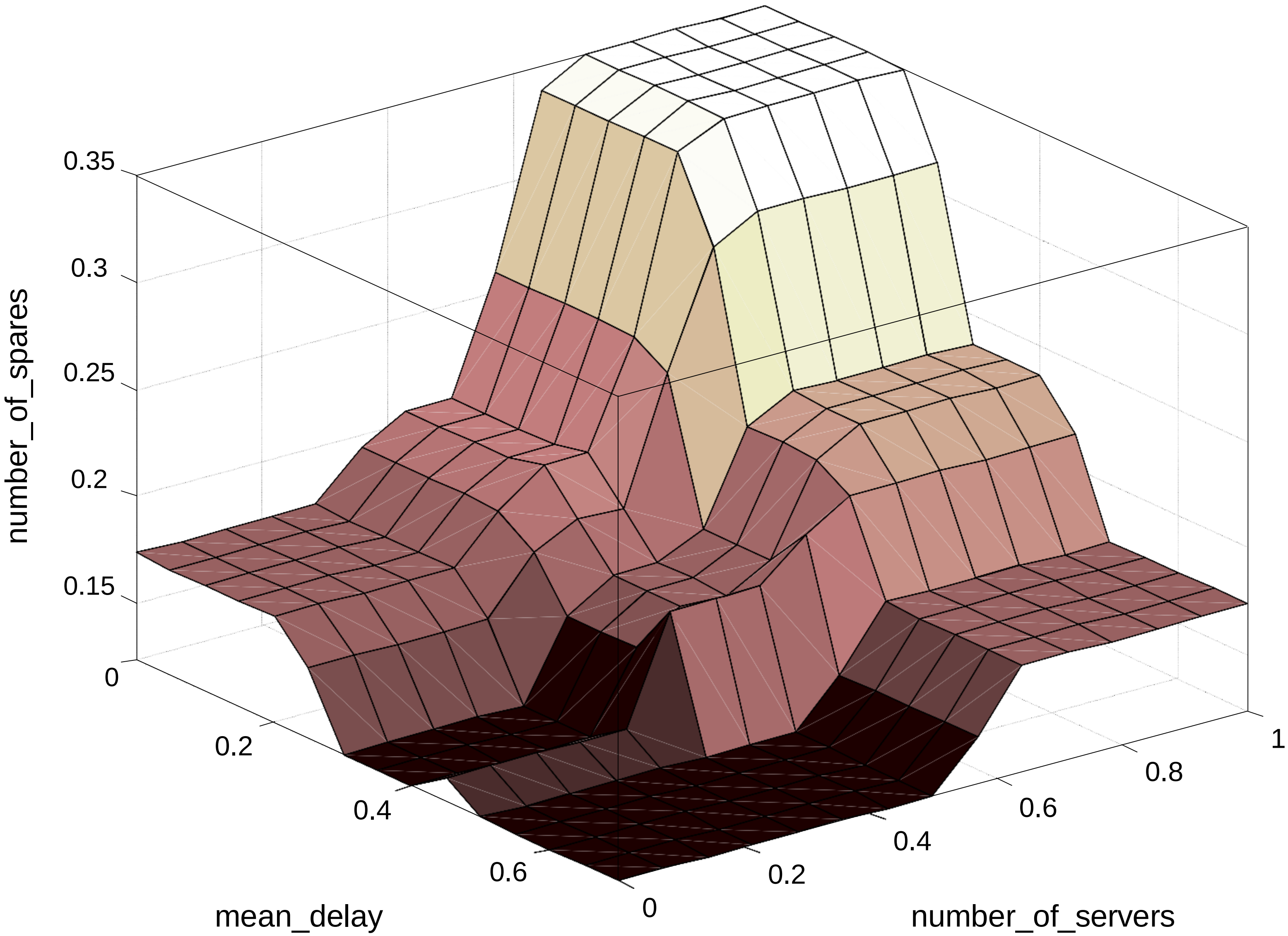
	VS	S	M
L	S	S	VS
RL	S	S	VS
M	VS	VS	VS
RS	VS	VS	VS
S	VS	VS	VS

	VS	S	M
L	S	S	VS
RL	S	S	VS
M	VS	VS	VS
RS	VS	VS	VS
S	VS	VS	VS

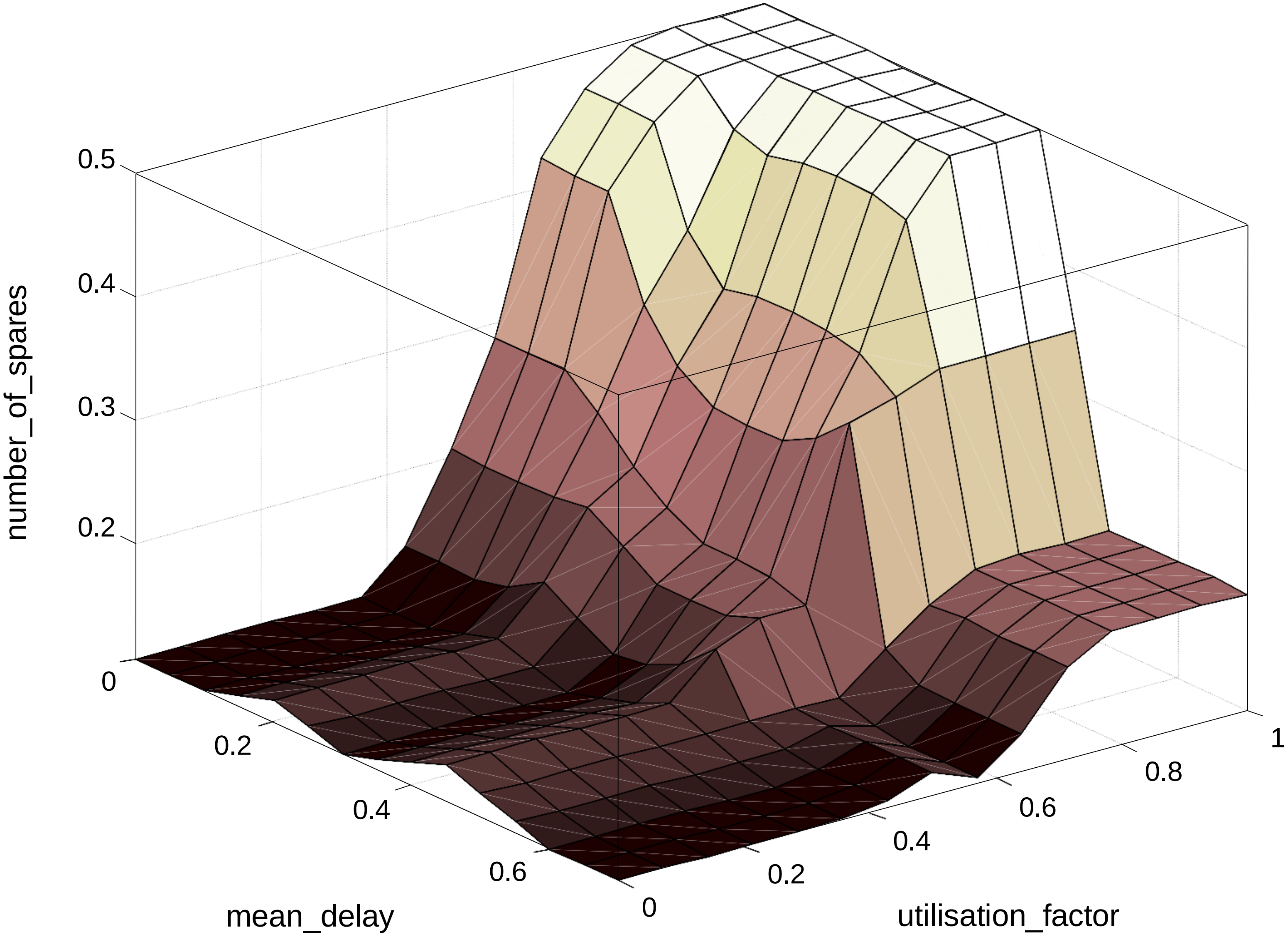
	VS	S	M
L	M	RS	S
RL	M	RS	S
M	RS	S	VS
RS	S	VS	VS
S	S	VS	VS

	VS	S	M
L	L	M	RS
RL	RL	M	RS
M	M	M	S
RS	VL	RL	RS
S	VL	L	M

Three-dimensional plots for Rule Base 3



Three-dimensional plots for Rule Base 3



Tuning fuzzy systems

1. Review model input and output variables, and if required redefine their ranges.
2. Review the fuzzy sets, and if required define additional sets on the universe of discourse. The use of wide fuzzy sets may cause the fuzzy system to perform roughly.
3. Provide sufficient overlap between neighbouring sets. It is suggested that triangle-to-triangle and trapezoid-to-triangle fuzzy sets should overlap between 25% to 50% of their bases.

4. Review the existing rules, and if required add new rules to the rule base.
5. Examine the rule base for opportunities to write hedge rules to capture the pathological behaviour of the system.
6. Adjust the rule execution weights. Most fuzzy logic tools allow control of the importance of rules by changing a weight multiplier.
7. Revise shapes of the fuzzy sets. In most cases, fuzzy systems are highly tolerant of a shape approximation.

Case: Fuzzy Control of a Real Time Inverted Pendulum System