Escuela Politécnica Superior de Linares
Telecommunication Engineering Master

UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Linares

Trabajo Fin de Máster

ANALYSIS OF INTERPRETABILITY OF FUZZY SCHEDULERS FOR CLOUD COMPUTING

Student: **LAURA MORENO CHICHARRO**

Supervisors: Dr. José Enrique Muñoz Expósito
Dr. Sebastián García Galán

Department: Telecommunication Engineering

MARCH, 2020

CONTENTS

CONTENTS 1
INDEX OF FIGURES 3
INDEX OF TABLES 5
ABSTRACT 6
1. INTRODUCTION 7
2. OBJECTIVES 9
3. STATE OF ART 10
3.1 CLOUD COMPUTING 10
3.1.1 SERVICE MODELS 10
3.1.2 TYPES OF SYSTEMS 12
3.2 POWER MODEL 14
3.3 DYNAMIC VOLTAGE AND FREQUENCY SCALE (DVFS) 15
3.3.1 DVFS POWER 15
3.3.2 DYNAMIC GOVERNORS 17
3.3.3 STATIC GOVERNORS 17
3.4 FUZZY LOGIC 18
3.4.1 FRBS STRUCTURE 19
3.5 GENETIC ALGORITHMS 21
3.5.1 SEARCH FOR NEW POPULATION 22
3.6 INTERPRETABILITY 23
3.7 MULTI-OBJECTIVE OPTIMIZATION 27
4. IMPLEMENTATION 28
4.1 WORKFLOWSIM-DVFS 28
4.1.1 ENTITIES 31
4.1.2 TASKS PROCESSING 32
4.2 INTERPRETABILITY 34
4.2.1 ASSESSMENT INTERPRETABILITY 35
4.3 PITTSBURGH ALGORITHM 42
4.3.1 LEARNING PROCESS 42
5. SOFTWARE MODIFICATIONS 43
5.1 WFS-DVFS MODIFICATIONS 43
5.1.1 WORKFLOSIM-DVFS SCHEDULERS 44
5.1.2 REQUIREMENTS 52
5.2 GUAJE TOOL MODIFICATIONS 53
5.2.1 REQUIREMENTS 54
6. MATLAB INTEGRATION 55

6.1 ALGORITHM STAGES 57
7. SIMULATION ENVIRONMENT 62
7.1 SCENARIO 62
7.2 POWER OPTIMIZATION EXPERIMENT 66
7.2.1 INTERPRETABILITY ANALYSIS 68
7.3 TIME OPTIMIZATION EXPERIMENT 70
7.3.1 INTERPRETABILITY ANALYSIS 72
7.4 MULTI-OBJECTIVE OPTIMIZATION EXPERIMENT 74
7.4.1 INTERPRETABILITY ANALYSIS 76
8. COMPARISON SCHEDULING ALGORITHMS 77
8.1 POWER OPTIMIZATION EXPERIMENT 79
8.1.1 FIRST STAGE: TRAINING 79
8.1.2 SECOND STAGE: VALIDATION 80
8.1.3 ANALYSIS AND CONCLUSION 81
8.2 TIME OPTIMIZATION EXPERIMENT 81
8.2.1 FIRST STAGE: TRAINING 81
8.2.2 SECOND STAGE: VALIDATION 82
8.2.3 ANALYSIS AND CONCLUSION 83
8.3 MULTI-OBJECTIVE OPTIMIZATION EXPERIMENT 84
8.3.1 FIRST STAGE: TRAINING 84
8.3.2 SECOND STAGE: VALIDATION 85
8.3.3 ANALYSIS AND CONCLUSION 86
9. CONCLUSIONS 89
10. REFERENCES 91

INDEX OF FIGURES

Figure 1. RELEVANT ASPECT SERVICE MODELS 12
Figure 2. TYPES OF CLOUD SYSTEMS. 13
Figure 3. ANTECEDENTS MFS. 18
Figure 4. CONSEQUENT MFS. 19
Figure 5. FRBS STRUCTURE. 19
Figure 6. EXAMPLE SCALING FACTORS. 20
Figure 7. EXAMPLE RULE BASE. 20
Figure 8. GENETIC ALGORITHMS: CREATION OF NEW POPULATION. 23
Figure 9. LEARNING OVERVIEW. 27
Figure 10. MONTAGE WORKFLOW. 29
Figure 11. SIMULATOR INPUTS AND OUTPUT FILES. 31
Figure 12. WORKFLOWSIM OVERVIEW. THE AREA SURROUNDED BY RED LINES IS SUPPORTED BY CLOUDSIM. 32
Figure 13. EXPERT KNOWLEDGE AND EXPERIMENTAL DATA. 34
Figure 14. HIERARCHICAL FUZZY SYSTEM. 37
Figure 15. RB1 RULE BASE DIMENSION. 38
Figure 16. RB2 RULE BASE COMPLEXITY (LOW) 39
Figure 17. RB2 RULE BASE COMPLEXITY (HIGH) 39
Figure 18. RB3 RULE BASE INTERPRETABILITY. 40
Figure 19. RB4 INTERPRETABILITY INDEX. 41
Figure 20. MIPS ANTECEDENT MFS. 45
Figure 21. PES ANTECEDENT MFS. 46
Figure 22. BW ANTECEDENT MFS. 46
Figure 23. RAM ANTECEDENT MFS. 46
Figure 24. TRANSFERT ANTECEDENT MFS. 46
Figure 25. OUTPUT CONSEQUENT MFS. 47
Figure 26. MIPS ANTECEDENT MFS. Figure 27. POWER ANTECEDENT MFS. 48
Figure 28. LENGTH ANTECEDENT MFS. Figure 29. UTILIZATION ANTECEDENT MFS. 48
Figure 30. POWERMAX ANTECEDENT MFS. 49
Figure 31. SELECTION CONSEQUENT MFS. 49
Figure 32. MEMBERSHIP FUNCTIONS ANTECEDENTS AND CONSEQUENT. 52
Figure 33. MIPS VARIABLE EXAMPLE OF dvfs.fis FILE. 53
Figure 34. CONFIG FOLDER. 53
Figure 35. SCHEME IMPLEMENTATION (Using GUAJE). 56
Figure 36. PITTSBURGH EVOLUTIONARY LEARNING. 58
Figure 37. CROSSING. 60
Figure 38. MAIN STAGES OF PITTSBURGH ALGORITHM. 61
Figure 39. EXAMPLE IMAGE MONTAGE. 62
Figure 40 CONVERGENCE CURVE POWER OPTIMIZATION EXPERIMENT. 67
Figure 41. ZOOM CONVERGENCE CURVE POWER OPTIMIZATION EXPERIMENT. .67
Figure 42. RB1, RB2, RB3 AND RB4 FUZZY RULES. 68
Figure 43. HIERARCHICAL FUZZY SYSTEM WITH REAL VALUES KB23. 69
Figure 44. CONVERGENCE CURVE TIME OPTIMIZATION EXPERIMENT. 71
Figure 45. ZOOM CONVERGENCE CURVE TIME OPTIMIZATION EXPERIMENT. 71
Figure 46. HIERARCHICAL FUZZY SYSTEM WITH REAL VALUES KB. 73
Figure 47. CONVERGENCE CURVE POWER. MOA OPTIMIZATION EXPERIMENT. 75
Figure 48. ZOOM CONVERGENCE CURVE TIME. MOA OPTIMIZATION EXPERIMENT. 75

Figure 49. HIERARCHICAL FUZZY SYSTEM WITH REAL VALUES KB14.....77
Figure 50. % IMPROVEMENT VALIDATION STAGE MONTAGE WORKFLOWS. 87
Figure 51. % IMPROVEMENT VALIDATION STAGE INSPIRAL WORKFLOWS. 88

INDEX OF TABLES

Table 1. FREQUENCY AND MIPS MULTIPLIER VALUES.....17
Table 2. SYSTEM VARIABLES.....42
Table 3. UNIVERSE OF DISCOURSE VARIABLES TIME SCHEDULER. 47
Table 4. UNIVERSE OF DISCOURSE ANTECEDENTS MFs TIME SCHEDULER.....47
Table 5. UNIVERSE OF DISCOURSE CONSEQUENT MFs TIME SCHEDULER. 47
Table 6. DISCOURSE UNIVERSE OF VARIABLES POWER SCHEDULER. 49
Table 7.VALUES ANTECENDET MFS POWER 49
Table 8. VALUES CONSEQUENT MFS POWER SCHEDULER. 49
Table 9. VALUES ANTECEDENT MFS MOA SCHEDULER.....51
Table 10. VALUES CONSEQUENT MFS MOA SCHEDULER.....51
Table 11. DISCOURSE UNIVERSE OF VARIABLES MOA SCHEDULER.....51
Table 12. FIXED PARAMETERS POWER MODEL. 63
Table 13. PARAMETERS PHYSICAL MACHINES. 64
Table 14. EXAMPLE ESTIMATION MIPS VM.....65
Table 15. MIPS VMS. 65
Table 16. TRAINING STAGE POWER EXPERIMENT.....79
Table 17. TRAINING STAGE % POWER EXPERIMENT.....79
Table 18. VALIDATION STAGE POWER EXPERIMENT.....80
Table 19. VALIDATION STAGE % POWER EXPERIMENT. 80
Table 20. TRAINING STAGE TIME EXPERIMENT.....81
Table 21. TRAINING STAGE % TIME EXPERIMENT. 82
Table 22. VALIDATION STAGE TIME EXPERIMENT. 82
Table 23. VALIDATION STAGE % TIME EXPERIMENT. 83
Table 24. TRAINING STAGE MOA EXPERIMENT. 84
Table 25. TRAINING STAGE % MOA EXPERIMENT. 84
Table 26. VALIDATION STAGE MOA EXPERIMENT MONTAGE. 85
Table 27. VALIDATION STAGE MOA EXPERIMENT INSPIRAL. 85
Table 28. VALIDATION STAGE % MOA EXPERIMENT MONTAGE. 86
Table 29. VALIDATION STAGE % MOA EXPERIMENT INSPIRAL. 86

ABSTRACT

This work presents an analysis of the interpretability of fuzzy system for cloud computing. Interpretability study has been carried out in fuzzy systems to optimize power consumption and time in Datacenter. The proposal of an algorithm that is able to find a system that reduces the power consumed by these Datacenters when processing tasks and optimizing the execution time for these tasks, choosing the knowledge base of the most interpretable system. That is, among the knowledge bases of the system that reduce the power and time consumption, the one with the highest interpretability is chosen.

In addition, getting systems that are interpretable by the human. To achieve the interpretable systems that also optimize power and time, a simulator of real environments (WorkflowSim-DVFS), that is able to calculate power and time consumption in a Datacenter and the implementation of a hierarchical fuzzy system to calculate the interpretability index of the system obtained.

1. INTRODUCTION

Linguistic fuzzy based on rule systems allows to build a linguistic model which could become interpretable by human beings [28]. This linguistic fuzzy modelling has two requirements : interpretability and accuracy.

In recent years, machine learning systems are resented everywhere, with a great interest in interpretability. The interest of researches in interpretability has grown. Interpretable system that provide an explanation of their results. In spite of the interest generated, today it is not very clear what is and how interpretability should be assess. There are several taxonomies to perform an interpretation of interpretability. [4]

Interpretability is necessary because most machine learning models have been called ‘Black Boxes’. This means that although the results are good, the human cannot explain the logic behind these algorithms.

In this way, it is necessary to know why the model has obtained these results. This has to do with the impact that the model applied to real life would have, that is, depending on the objective of the algorithm. Different algorithms can be described, optimizing energy, but to consider that knowledge obtained from de data has been achieved, the human cognitive factor is necessary. Therefore, interpretability is very important if the algorithm is to be put into practice, since there are very good algorithms that, if they are not interpretable, may not be used. [5]

Some of the advantages that interpretability produces are to provide reliability to the results obtained, help in debugging, help to detect if it is necessary to collect new samples for learning process, help a person in decision making and greater safety and robustness in the final model.

Interpretability must be taken into account in many areas, from the creation of autonomous cars, predictive policing system and even security.

IN the development of this project, an analysis of the interpretability is going to be carried out in fuzzy systems for cloud computing.

Currently and for a few years, large companies are in a process of relocation and internationalization. The need for computing and data processing has been increasing faster than the computing capacity of personal computers.

Thanks to the emergence of the Internet, it has possible to create many paradigms whose purpose is to offer technology as a service, allowing decentralization. That is, the interconnection of several devices found in different places increases the amount of computing and storage resources distributed. This is how a new paradigm of computing and service provision called ‘Cloud Computing’ appeared.

Cloud Computing is a type of internet-based computing that offers different services. National Institute of Standards and Technology (NIST) defines this paradigm as ‘ Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." [1]

However, despite having numerous advantages of cloud computing (cost savings in infrastructure, greater efficiency, agility, scalability…) it also has some risks, such as increased energy consumption.

The Data Processing Centers (DPC) grow implies a greater number of servers with more resources, thus causing greater energy consumption.

In addition, increasing the number of servers and resources implies increasing Datacenter (DC) temperatures. Cooling systems are necessary to prevent high temperatures and thus provide greater system reliability. Reliability is greater because the higher the temperature, the error rates are also higher. These cooling systems also consume energy and emit CO2.

The proposed algorithm favors energy consumption, since it reduces the power consumption, improves processing time and, by reducing power, greater reliability is obtained in the system (for 10ºC the temperature increases, the failure rate doubles [2]).

Datacenter consume more than 416 terawatts of electricity per year. It is estimated to be 2% of the energy product worldwide. It is expected that by 2021, cloud computing will represent 95% of Datacenter traffic and will handle more than 94% of the workloads of high performance computing. [3]

Later, what is the interpretability will be explained in detail, the taxonomy used is and some examples. These parameters (power consumed, execution time and interpretability) imply the need to establish a multi-objective algorithm. This algorithm tries to optimize the power and the execution time, obtaining an interpretable system. Among all the systems that optimize power and time, the most interpretable will be chosen.

This multi-objective optimization obtains a balance between these two parameters, achieving values that satisfy both requirement and take into account the interpretability of the system. This optimization has been obtained through the Pareto front.

2. OBJECTIVES

The main objective of this work is the analysis of the interpretability of fuzzy rules based system. These systems, even when acquired automatically, may be interpreted by experts.

In addition, in the development of the project, secondary objective has been established as the fuzzy systems analysed optimize the power and time consumption in Datacenter. These systems will consist of knowledge bases whose main objective is to optimize the power consumed in the Datacenter and the execution time of the tasks in them, choosing the most interpretable.

For this, a multi-objective optimization will be used whose main objectives are the optimization of power and time. Because of this multi-objective, a knowledge base will be obtained from which its interpretability will be analysed.

To carry out the objectives of this master Thesis, the work is based on simulations. Experiments, that simulate a real scenario, will be carried out as if they were real Datacenter. If the results obtained in the simulations do not correspond to a real scenario, the analysis and conclusions would not make sense since the could not be applied to real life. Therefore, the proposal algorithm could not be developed in Datacenters already implemented.

Having these objectives clear, the first part of the work is to analyse what is the interpretability and how is its assessment. In addition to finding a good scenario for the simulation and the second part, analyse the interpretability of the systems obtained with this optimization.

The simulations will simulate a real scenario, but there will always be some difference with a real environment. In a real Datacenter, many parameters influence the simulations that cannot be taken into account. Because of this, simulators are increasingly getting better performance and are becoming more and more like reality.

To perform the simulations, a simulator already implemented, WorkflowSim-DVFS [6], with the ability to calculate the power consumption and time consumed in a Datacenter. In addition, a hierarchical fuzzy system with four linked knowledge bases are implemented.

Throughout this document, the changes made for these optimizations, the simulations performed and the analysis of these experiments will be specified.

3. STATE OF ART

In this section, some necessary concepts to understand the development of this work shall be explained briefly. It is necessary to know what is the interpretability, what is Cloud Computing, since the analysis is in fuzzy scheduler for Cloud Computing. It is also necessary to know the power model of the simulator used, as well as to know the methodology followed for the interpretability assessment.

3.1 CLOUD COMPUTING

As discussed in the Introduction, Cloud Computing is a type of internet-based computing that offers different services. It is a distributed processing paradigm leading the next generation computational platforms based on the externalization of computing needs offered as services. [6]

In addition, Cloud Computing is a technology that seeks to have all files and information on the Internet, without the need to have sufficient storage capacity on a physical machine, such as a computer.

Below, some of the advantages of this technology are presented:

- Security. The data is always safe.
- Low cost. Not having to invest in infrastructure or licenses, you only have to pay for products or applications per use. There are also free products.
- In the user's physical machines, it is not necessary to have large storage capacity.
- Real-time information
- Mobility. You only need an Internet connection to access the storage information
- Access to all cloud information [7] [45]

These advantages provide companies with greater flexibility in relation to their data and information, since they can access anywhere and at any time. It is essential for companies that have different locations around the world or in different work environments.

With minimal management, all software elements of cloud computing can be sized on demand, only with an Internet connection.

Datacenter is the server hardware for storing and accessing data through its local network. Typically, an internal IT department on the company maintains it.

This new technology is focused on business use. The usual scenario was the acquisition of new computers or servers along with their software licenses. Today, with the growth of cloud computing, companies have been acquiring a new way to consume resources. Companies instead of making an investment to acquire hardware and software, they pay a periodic fee based on the cloud services they consumed.

These service models are adapted to the needs and requirements that each company has and the users that hire them. The different types of service models are explained below.

3.1.1 SERVICE MODELS

There are different types of service adapted to the needs of the client. Based on these needs, providers offer Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS) and Hardware as a Service (HaaS). Figure 1 shows the relevant aspects of each of them.

These types are based on a layered architecture, being HaaS the bottom level and SaaS the upper level.

- **HaaS**. Hardware level.
  This level is managed by the Information Technology (IT) company that offer the cloud service and the physical resources as physical servers, switches and systems to provide power and cooling of the Datacenter are managed.
  It is a networked computing, where the computing power is leased to a central provider. Users rent products from a provider. This type can be remote or in-situ control, managing the maintenance and administration of hardware systems and helps users manage the hardware license requirements.
  The uses Internet Protocol (IP) connections to connect remotely. The user sends data to the provider and the hardware device sends the results. This prevents companies from having to buy hardware devices. [8]
  If the user wants to increase the service fee, the provider can replace the hardware with another one that has more capacity or more resources.
  One of the responsibilities of the company is to replace the hardware devices in case of failure. Thus, user do not have the need to invest in these devices every time they need an update on them.
  The Managed Service Provider (MSP) provides scalability, that is, if the client wants to increase the Datacenter's resources, he can install the necessary devices by increasing the fee. Instead, if the customer needs less, MSP would remove those hardware devices and the customer's fee would be reduced.[10]

- **IaaS**. Infrastructure level.
  IaaS contains the fundamental building blocks for IT in the cloud. It allows access to network connection features, equipment and data storage. Some of advantages are the highest level of flexibility and management control around its IT resources and try to maintain the same characteristics as the existing IT resources with which many IT departments and developers are familiar.
  In this layer, more privileges and control of the hardware devices are provided, with the characteristics request by the user, and with the management rights of the virtual machines (VMs). It is possible to decide how to manage host resources. That is, the host are divided into several VMs.
  These machines can configure and have the ability to choose the performance of each, managing resources dynamically. This implies that the user pays for the performance offered, without worrying about the hardware. [9][10]

- **PaaS**. Platform level.
  This layer is where more privileges are offered in the cloud and more flexibility than in the previous ones. In this way, eliminate the need for infrastructure management companies and thus focus on the implementation and administration of their application.
  The user manages the whole operating system, working as if it were a local server (in the cloud) but with certain responsibilities about the applications.
  Users do not have information or the privilege of changing cloud resources. It can be considered and advantage for users who do not know how to handle VMs.[9][10]

- **SaaS**. Application level.
  It is the final product that is provided to the user and that the provider executes and manages. Normally, when talking about software as a service, ir refers to the application of the end user.
  With SaaS, the user does not have to think about the service is maintained or how the infrastructure is managed. User just have to have the knowledge to use the software.
  Some of advantages of this service are: the user does not need to have specialized knowledge of the system, the responsibility of the service is of the company, offering the user application at any time and guaranteeing its operation. [9][10]

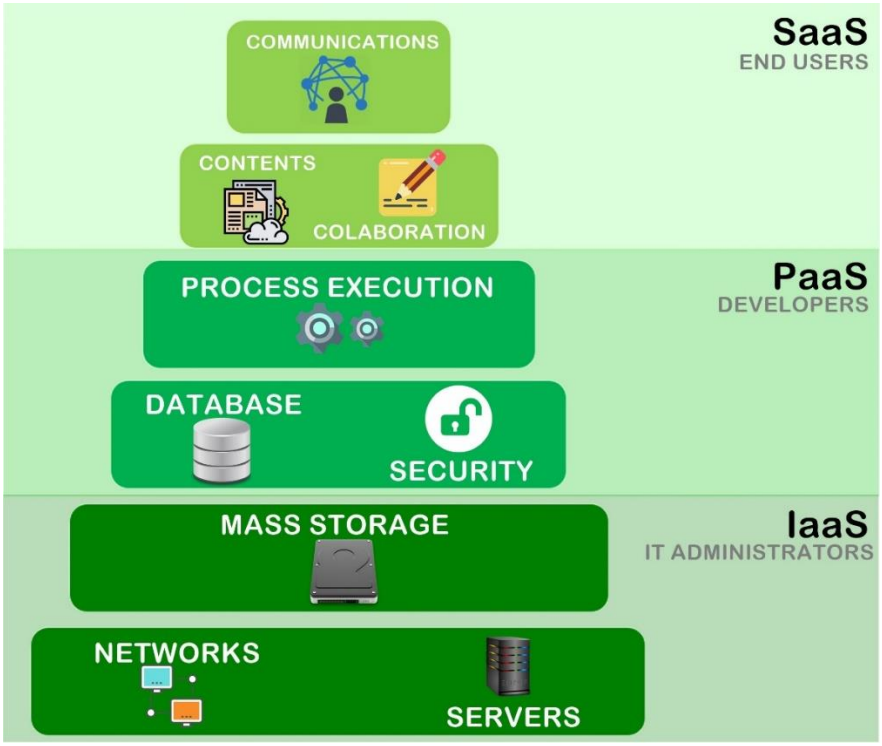

Figure 1. RELEVANT ASPECT SERVICE MODELS

3.1.2 TYPES OF SYSTEMS

There are different types of infrastructure depending on whether it is shared or not among more clients. [9]

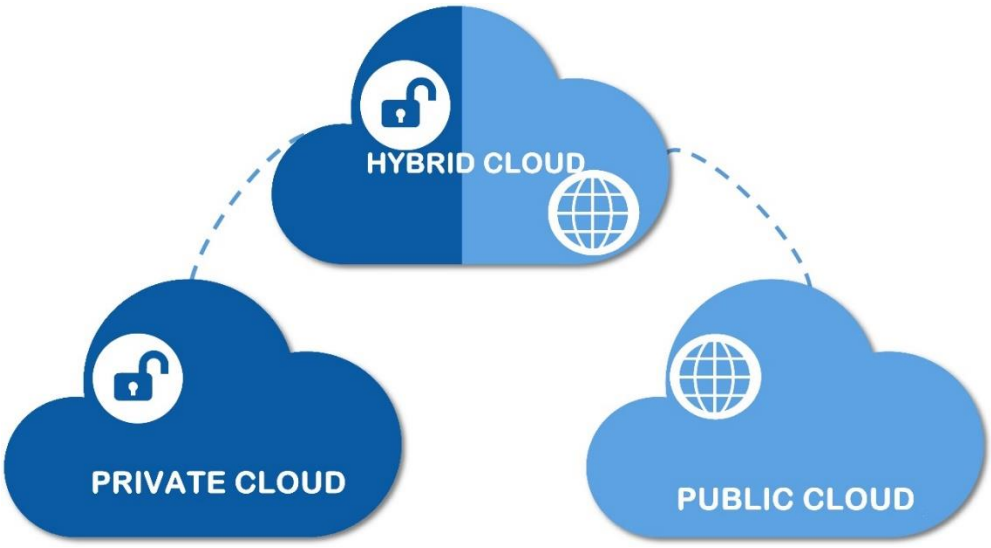

Figure 2. TYPES OF CLOUD SYSTEMS.

- **Public Cloud.**
  All parts of the application run in the cloud. There are applications that have been created directly in the cloud or that have been transferred from the existing infrastructure. This type offers the same IT infrastructure to multiple clients. It is usually accessed through the Internet or Virtual Private Networks (VPNs) access.
  The provider is in charge of the selection and technological evolution, as well as the management of the infrastructure.
  The disadvantage of this type of cloud is security, since it may not be enough for some users, which is why there is a private cloud. [10][11]
- **Private Cloud.**
  This type of cloud does not provide many of the benefits of the public cloud, but it is used to have dedicated resources and prevent other users from accessing the information. Only an organization has access to the resources used in the clout, it has an exclusive cloud environment.
  Some of its advantages are customized configurations (security, capacity, availability…), greater integration with own infrastructures services, interconnection capacity with similar technology infrastructures and virtualization advantages. [10][11]
- **Hybrid Cloud.**
  Combine resources from the private and public cloud. Some customers who have their own infrastructure want to take advantage of the services of an external provider.
  These clouds provide agility and cost reduction, although they lose control. In the hybrid cloud, there are two infrastructures parts of both networks, one in the public part and another in the private part.
  Thanks to this advantage, important data can be stored in the private part while other applications run in the public.
  They offer greater control and security than in the public cloud. The client pays for additional resources when he needs them, avoiding to comply with the Service Level Agreement (SLA) of the users and avoid breaking the conditions. [10][11]

3.2 POWER MODEL

The main objective of this project is to analyse the interpretability of fuzzy schedulers. In this case, the system will optimize two parameters, the power and time consumed in a Datacenter, always choosing the system that has the greatest interpretability.

Therefore, a power model that is able to measure the power consumed is necessary taking into account several parameters.

When a task is executed in a resource or host, it will be executed in a certain execution time and the host will consume a certain power. There is a relationship between both parameters that is given by the following equation:

$$E = P \cdot t \qquad (1)$$

This relationship is very important because taking into account two parameters (execution time and power consumption) the energy saved can be known.

When a machine takes less time to execute a task, it implies that it has a greater number of resources and therefore will consume more power. There must be a balance between the execution time and power consumed, since if the execution time is reduced but the power consumed is increased, the system it not meeting the main objectives. [10]

The execution time is calculated taking into account the length of tasks to be executed and the performance of the machine. Two parts, the dynamic and the static component represent the calculation of the energy.

$$E = E_d + E_s \qquad (2)$$

The static component depends on storage, I/O devices and the memory, and it is related with the dynamic component, while the dynamic component depends on the processor.

The dynamic component can be calculated with the following equation. It calculation is the product of squared processor voltage, the value of its capacitance, the processor frequency and a constant.

$$P_d = K_d C V^2 f \qquad (3)$$

If equation 1 and 3 are considered, then the value of dynamic energy can be estimated:

$$E_d = P_d \cdot t \qquad (4)$$

And the static component is considered as a portion of dynamic component, being Ks a constant.

$$E_s = K_s E_d \qquad (5)$$

With all the previous equation, it is possible to calculate the value of the total energy, this being the sum of the static and dynamic energy.

$$E = E_s + E_d = K_s E_d + E_d = (1 + K_s) E_d \quad \propto E_s(V, f) \qquad (6)$$

For this project, it must be taken into account that if the execution time of a task decreases, it implies that the machine's performance increases and therefore the power consumption is increased. [10]

3.3 DYNAMIC VOLTAGE AND FREQUENCY SCALE (DVFS)

DVFS is a strategy for reducing energy consumption, allowing the change of the performance of the machine dynamically, based on the workload. That is, it allows workflows to be processed considering the regulation of the frequency and voltage of the resources.

Today, there are CPUs that have this technique incorporated, being able to vary the voltage and frequency dynamically to adapt to the workflow. This means a reduction in power consumption, avoiding a delay in the execution of tasks.

Its main objective is to reduce the energy consumption and this saving depends on the governor selected. A governor is a core model that manages the operations points in both voltage and frequency based on an algorithm. [6]

There are five governors divided in two types: static and dynamic governors.

Dynamic governors are based on thresholds, that is, both voltage and frequency vary depending on the current used with respect to the threshold. The threshold are values set by each governor. Depending on the governor chosen, the frequency and voltage will vary in one way or another. However, static governors have a fixed voltage and frequency value. The different values of voltage and frequency depend on the processor and these values are displayed as multipliers.

3.3.1 DVFS POWER

The computing power of a CPU is given by the following expression: [10]

$$P(V, f) = aCV^2 f \qquad (7)$$

Where f is the frequency, V is the voltage and a and C are frequency multipliers. A CPU can work with different frequency values. Taking into account the relationship between the frequency and voltage, if the voltage is lower, then lower frequencies can be selected on CPU.

The multiplier values are specified as a percentage of the total processing capacity. As there is not explicit way to indicate the maximum frequency of the CPU, then MIPS are considered. This multiplier value is defined by the DVFS technique and by the governor used.

The higher the value of MIPS, it implies greater performance, which means more power consumed and, on the contrary, lowest execution time.

The simulator (WorkFlowSim-DVFS) works with two power consumed values for each multiplier, depending on whether the utilization is null or full (P_{idle} and P_{full}, respectively). To calculate the power consumed, the simulator uses these two power values for the current frequency value. Taking into account the utilization value, the power consumed can be estimated following the next equation:

$$P_{total}(\alpha, V, f) = P_{idle}(V, f) + [P_{full}(V, f) - P_{idle}(V, f)]\alpha \quad (8)$$

Where α is the CPU utilization rate, P_{idle} when the lowest performance (α = 0) and P_{full} when the highest performance (α = 1). The utilization value is also used in DVFS to check the thresholds of the governors. This is very important, since it is responsible for power optimization in terms of the CPU utilization value.

Both P_{idle} and P_{full} depend on the utilization of CPU, P_{idle} is when the utilization α) is 0 and P_{full} when the utilization is 1. If equation 7 is considered and if the voltage and frequency values is decreased, then the power consumption is reduced.

If the equation 8 is considered, the following equations are obtained for highest and lowest performance.

$$P(0, V, f) = P_{idle}(V, f) = aCV_{idle}^2 f_{idle} \quad (9)$$

$$P(1, V, f) = P_{full}(V, f) = aCV_{full}^2 f_{full} \quad (10)$$

With these equations in mind, in this power model, the power varies depending on the frequency. The frequency values are obtained from the processor frequency multiplier with the frequency of the motherboard's base.

The voltage is scaled with the frequency, and the frequency varies according to the multiplier. These multipliers depend on the processor CPU. The power model used allows measuring the P_{idle} and P_{full} values of the CPU that DVFS supports. Therefore, for different hosts, there are different frequency and voltage values based on the multipliers, and therefore different values for P_{idle} and P_{full}.

This behaviour resembles that of a real environment since it is possible to estimate the energy and power consumption considering the savings through the DVFS technique of the simulator.

Table 1 shows the multipliers values taking into account that the real processor used is CPU intel(R) Core (TM)2 Quad CPU Q6700 @2.670GHz with 4GB Ram.

MULTIPLIER (%)	59.925	69.93	79.89	89.89	100
FREQUENCY (GHz)	1.600	1.487	2.113	2.400	2.670
PERFORMANCE (MIPS)	898.875	1048.95	1198.35	1348.35	1500
NULL UTILIZATION P_{idle} (W)	82.75	82.85	85.95	83.10	83.25
FULL UTILIZATION P_{full} (W)	88.77	92.00	95.5	99.45	103.0

Table 1. FREQUENCY AND MIPS MULTIPLIER VALUES

The frequency values are obtained of the multiplication of the multiplier by the base frequency (2.670Ghz), being the possible discrete frequency of the processor. Since the value of MIPS is 1500, if the multipliers multiply this value, the performance of each one is obtained. Finally, the P_{idle} and P_{full} values are obtained from equations (9) and (10).

As equation (8) depends on the utilization value, the DVFS governors are the responsible for selecting the frequency according to their criteria, also selecting the performance and power values when the performance is the maximum and when it is the minimum (P_{full} and P_{idle}, respectively). Thus, governors compare the current utilization with the thresholds defined to know if the CPU needs higher performance or if the processor is idle and can save power.

In this work, the model power used is 'Power Model Analytical' that it will be explained in detail in section 7.1.

3.3.2 DYNAMIC GOVERNORS

Dynamic governors allow reducing energy consumption if the utilization value is below the down threshold or increase the performance if the utilization value is above the up threshold. There are two types of dynamic governors:

- **OnDemand.** This governor compares the CPU utilization values with the selected threshold value. In this simulator, the value is 95%. In the OnDemand governor, both thresholds have the same value. To prevent the variation from being constant, it has an iterator that counts and if the result is the same, then the multiplier decreases by one.
- **Conservative.** This governor considers two thresholds (up threshold and down threshold). The default values of this simulator are 80% for up threshold and 20% for down threshold.

3.3.3 STATIC GOVERNORS

Static governors use a fixed value for both voltage and frequency. There are three types of static governor.

- **Performance.** This one produces the lowest execution time but consumes more energy. It consists of fixing a voltage and frequency value to achieve maximum performance.

- **PowerSave.** This governor fixes the lowest voltage and frequency values, obtaining the lowest energy consumption but increasing the execution time. It is the opposite of Performance governor.

- **UserSpace.** The user program selects the frequency and voltage values.

3.4 FUZZY LOGIC

The fuzzy logic is used in Fuzzy Rule Based Systems (FRBS). The logic FRBS is an extension of the classical rule system determined by IF-THEN rules, where the antecedents and consequents are formed by fuzzy logic (FL). [12]

FRBS are expert system that including human knowledge in the form of fuzzy rules and fuzzy sets (linguistics variables and values), called membership functions (MFs). This system is comprised by two main components: Knowledge base (KB) and Inference System.

The knowledge base (KB) stores the available problem knowledge in the form of fuzzy rules. It is obtained from the expert knowledge or by machine learning methods. The design of KB is related with Data Base (DB) and Rule Base (RB).

- DB is defined by the variable universes of discourse, scaling factors or functions, number of linguistic terms or labels per variable and membership functions associated to the labels.
  The database is composed of the input variables (antecedents) and the output variables (consequents). Each variable (both input and output) is composed of several fuzzy sets or membership functions. An example can be seen in figure 3 and figure 4. In this example, the antecedents consider three MFs: low, average and high and the consequent consider five MFs: very low, low, average, high and very high. Figure 3 shows the membership functions of antecedent and figure 4 shows the MFs of consequent.
- RB is derived of fuzzy rule composition that includes all the rules that will be executed in the inference process. The rules are defined by IF-THEN form, where each antecedent has a value belonging to a MF and returns a value of MF of consequent.

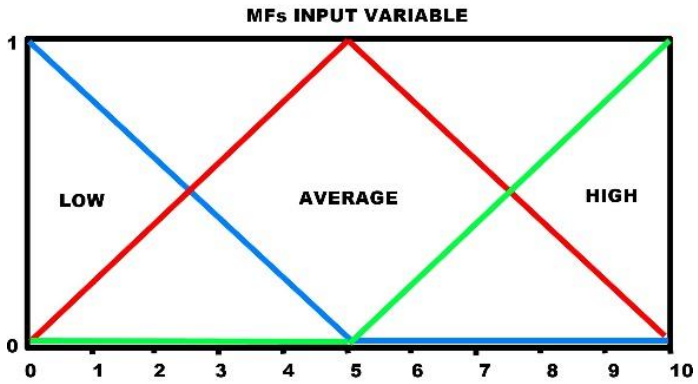

Figure 3. ANTECEDENTS MFS.

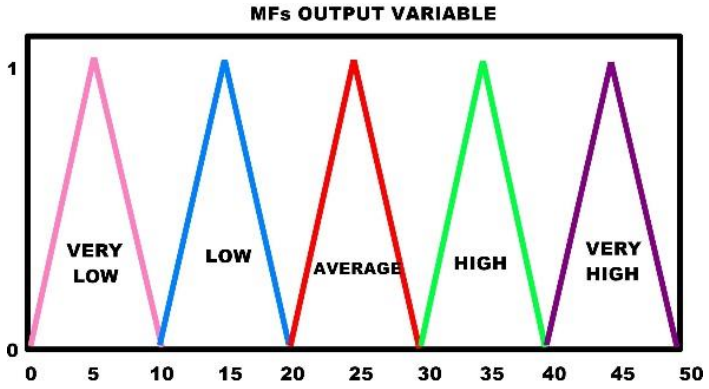

Figure 4. CONSEQUENT MFS.

The inference system applies a fuzzy reasoning method on the inputs and the KB rules to give a system output. It is configured by choosing the fuzzy operator for each component of the structure (more detail in FRBS STRUCTURE). [13]

The values of the system need to be normalized (range 0 to 1) to allow a single FRBS be able to work in different systems.

3.4.1 FRBS STRUCTURE

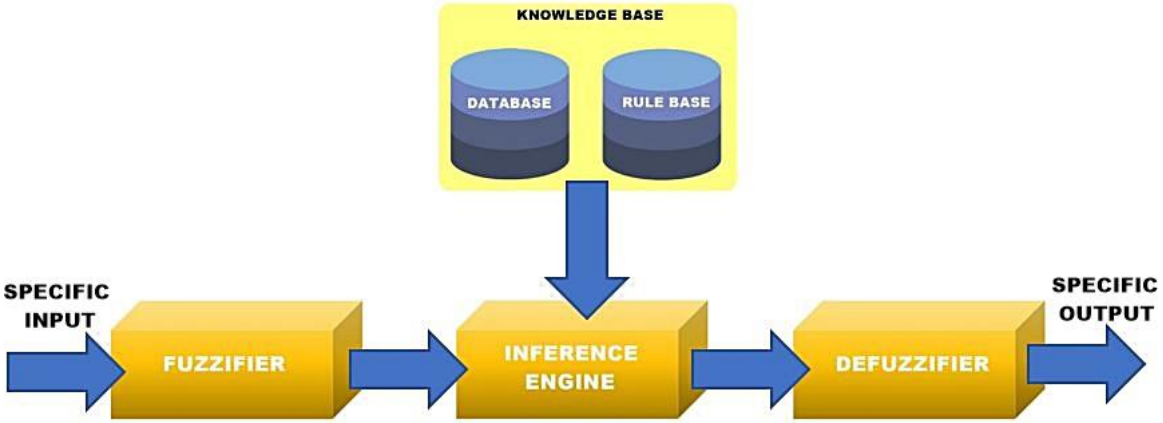

Figure 5. FRBS STRUCTURE.

Figure 5 shows the components of FRBS structure, that they will be explained below:

1. **Fuzzifier.** When measurements are made in any process, to later make a decision, these measures do not come in the form of a fuzzy set. Therefore, a module transforms those values of the FRBS input into fuzzy sets that can be treated by the inference engine. It convert the input value into a value of fuzzy set. Depending of the type of fuzzy system, fuzzifier converts each value into a fuzzy set with the degree of belonging equal to 1 and 0 to others values, match each value with the correspondent linguistic term or calculate the degree of belonging to each fuzzy set used that linguistic variable.
   [14]
2. **Knowledge base.** This module includes both the database and the rules base.
   The Database (DB) is used by the inferences engine and contains information about the shape of fuzzy sets. It contains the linguistic definition of the variables. Also, scaling factors can be defined to extend or reduced the universe of discourse (U).

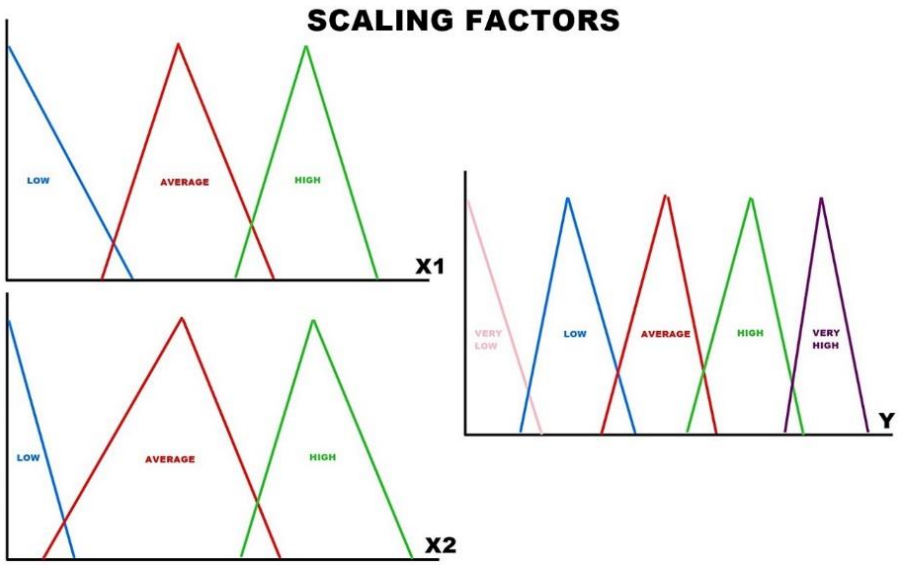

Figure 6. EXAMPLE SCALING FACTORS.

Where X1 and X2 are input variables and Y is output variable.

The Rule Base (RB) contains fuzzy control statements, using the information collected in the database. It contains the set of actions to be performed depending on the state. The following aspects must be taken into account: What control and status variables will be considered, the structure of the fuzzy rule and what see of rules will be used.

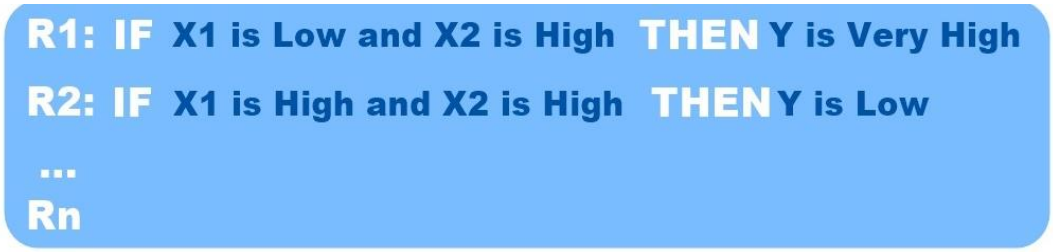

Figure 7. EXAMPLE RULE BASE.

In a FRBS there can be several rules, which are evaluated each time the system evaluates the antecedents to obtain the consequents. The rule system must cover most of the situations that can occur with the input parameters.

There are two types of rules: AND and OR. AND clauses set the consequent value as the minimum value of all the antecedents, and OR clauses set the consequent value as the maximum of the antecedent.

3. **Inference Engine.** The inference engine obtains the conclusions by applying classical logic to the rules provided by the knowledge base. It performs the reasoning process (fuzzy rules) to estimate the value of the output based on the input value. The inference engine evaluates the current values of each rule, receiving the grade of membership of each antecedent to their membership functions from the fuzzifier.
4. **Defuzzifier.** When the engine makes a fuzzy inference, it results in a fuzzy set, which is not applicable to any process or system to be controlled. Therefore, it is necessary a module that transforms that fuzzy set at the output of inference engine into a specific value at the input of the system to be controlled. [14]

The result of an FRBS system depends largely on the Rule Base. In most systems, it is difficult for the user to define these rules and cover all possibilities to obtain the best result. For this reason, the user uses the FRBS systems together with an algorithm that creates and optimizes these rules.

This optimization algorithm is Pittsburgh. There are several algorithms to optimize the fuzzy rules such as KASIA [37], MICHIGAN, PITTSBURGH, among others. In this work, the last one is the algorithm used.

In the case of Pittsburgh and Michigan, they are genetic algorithm and KASIA use Particle Swarm Optimization (PSO). The main objective is to create a population of individuals or particles and find the best solution. These algorithms have to make a balance between exploration and exploitation.

Exploration refers to the ability to show different parts of the search space in the population, while exploitation refers to the capacity of tuning and combination of the optimal solutions.

Exploration is useful to avoid focusing on local optimum while exploitation is used to obtain the global optimum once it has been sufficiently approximated.

In first stage, the algorithm must show great diversity, and in the end, that diversity must decrease to find the best solution.

For this reason, it is necessary to use one of these algorithms to obtain a more compact rule base than if it is created manually.

3.5 GENETIC ALGORITHMS

The Genetic Algorithms (GAs) are defined as search algorithms, based on natural evolution and genetics, which provide a robust search and optimization mechanism in complex system, offering a useful way for optimization problems that require efficiency and effectiveness. [15]

A GA has different parameter, such as the population size, the probability of selection, the probability of crossing and the probability of mutation. The choice of these parameters will determine, largely, that the algorithm finds an optimal solution and efficiently to the given problem.

For a GA obtain satisfactory performance, it is necessary to configure it, specifically, for each type of problem. There is no general rule by which appropriate parameters applicable to all types of problems can be found. However, in the most cases, the recommended values are used in the literature, and in others, their choice represents a problem of trial and error.

GAs simulate the Darwinian evolutionary process in artificial system, based on natural evolution system to solve problems. The evolution natural is the process through which organisms that best adapt to an environment displace the least adapted through favourable genetic changes in the population throughout the different generations.

Darwin said that if a species does not change, it would become incompatible with its environment because it tends to change over time. He highlighted the similarities between parents and children observed in nature and some characteristics of the species were hereditary. Thus, from generation to generation changes occur to make the new individual more able to survive.

The evolutionary process occur when an individual has the ability to reproduce, there is a population of individual that are able to reproduce and there is some difference between the individuals. The diversity of individuals in the population is because each individual has different chromosomes from the others and their behaviour is different in the environment. These changes of each individual will be reflected in their degree of survival, adaption and level of reproduction.

In most cases, the combination of the good characteristics of the parents can cause the offspring to be better adapted to the environment that the parents. In addition, mutation process causes alterations in the genetic information of individual, introducing new genetic variations and causing the offspring has different genetic material from the parents. In this way, the species evolve adapting better to the environment, but this adaptation is not only determined by their genetic characteristics, but also influence other factors such as learning acquired by the method of trial and error or acquired by imitation of the behaviour of parents. In this way, the population of the best-adapted individuals (offspring) replaces the population of worse adapted individuals (parents).

In conclusion, the population is a set of individuals, which their physical and behavioural differences affect the response of each in the environment. The population is evolving through natural selection introduced by random changes similar to biological evolution (genetic mutation and recombination) and finally, the individuals that survive are selected and which individuals are discarded.

3.5.1 SEARCH FOR NEW POPULATION

Each individual of the initial population is assigned a score, which indicates the degree of adequacy of said solution. Through a selection mechanism, individuals are chosen to reproduce. The greater the adaptation of an individual to the problem, the greater the probability it will be selected to reproduce, crossing its genetic material with another individual. On the contrary, the smaller the adaptation of an individual, the lower the probability of being selected for reproduction, and therefore, of being discarded.

As a result of the application of genetic operators of crossing and mutation, population evolves a new generation where the better adapted individual replace those worse adapted.

These genetic crossing and mutation operators have associated probabilities.

- The probability of crossover (Pc) is the probability of applying the crossing operator on the selected individuals as parents. If this operator does not apply, the offspring would be obtained by duplication of the parents.
- The probability of mutation is the probability of applying the mutation operator on each child individually, randomly altering each gene of the same.

Figure 8 shows the steps that the algorithm follows to generate the new population from the initial population.

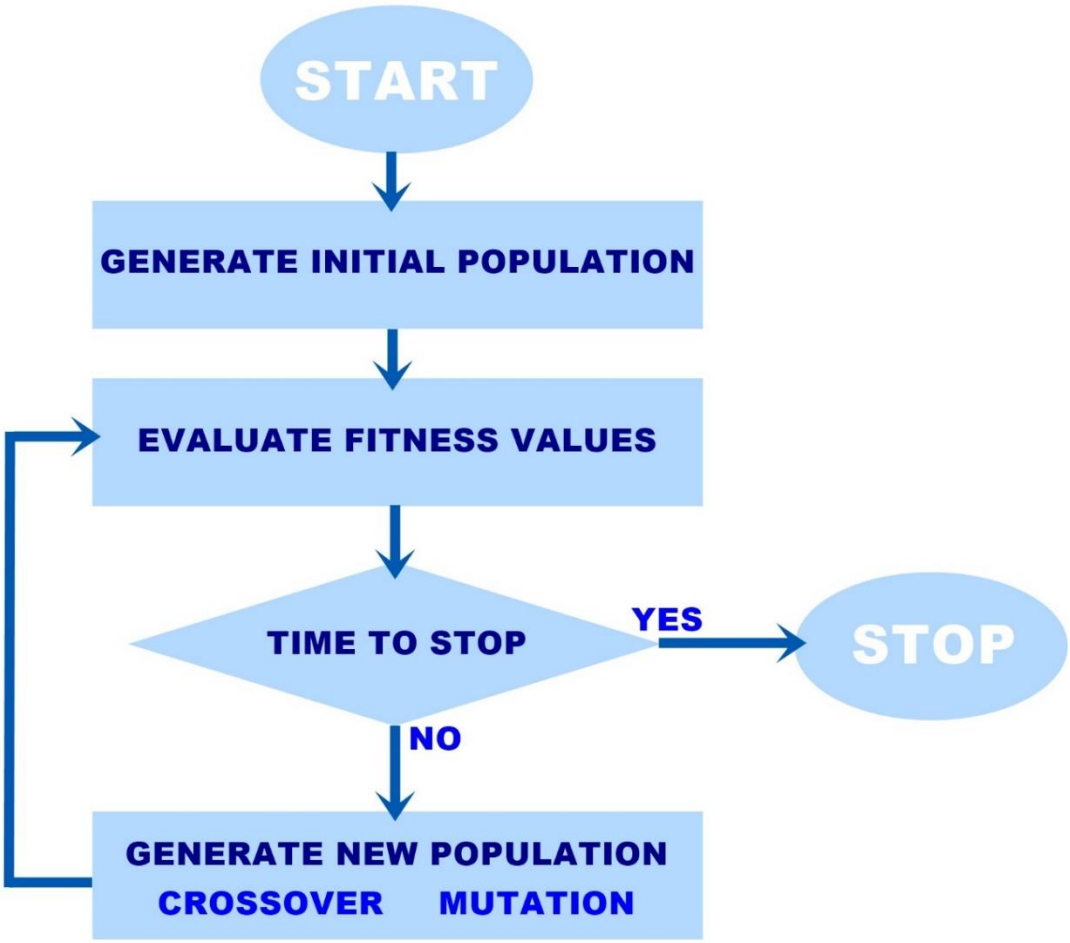

Figure 8. GENETIC ALGORITHMS: CREATION OF NEW POPULATION.

As shown in the scheme, the first step is to generate an initial population randomly and evaluate it according to the chosen fitness. This process is repeated until the number of iterations is finished (chosen by the user). If it is the last iteration, then the algorithm ends and the best solution is obtained. If it is not yet the last iteration, then a new population is obtained. This new population is generated by crossing the individuals of the initial population and a mutation process.

The idea is to generate a population that in addition to satisfying the objectives of the work (fitness) is the most interpretable system. That is, the genetic algorithm gets a population with a rule base with an interpretability index, in addition to getting one with the best position in terms of power and time, but the most interpretable of them.

As a genetic algorithm, 'Pittsburgh Approach' has been proposed for use in this work. Later, the process of generating new populations will be explained in detail.

3.6 INTERPRETABILITY

The main objective of this work is the analysis of the interpretability. There is not a formal definition of interpretability, but we can say that interpretability is the ability to interpret the significance of something. It is so difficult find a good definition because understanding is likely to be one of the most valuable human abilities, related to human intelligence and language processing capabilities. Thus, measuring the interpretability is a subjective task,

which depends of the preferences, knowledge and experience of the interpreter who assesses the system. [16][17][18]

In the case of this work, interpretability is an easily understood, explicated or accounted for a human being, and it is a system designed under fuzzy logic formalism.

Several authors say that there are necessary conditions for interpretable fuzzy partitions, because the fuzzy logic semantic favors the interpretability but does not guarantee it.

Some of them are:

- A justifiable number of fuzzy sets. Seven (plus or minus two) is a limit of information processing capability.
- Distinguishability. Each membership function represents a different linguist concept.
- Overlapping. All the fuzzy sets should significantly overlap with the aim of yielding gradual transitions between adjacent sets.
- Coverage. Each data point should belong significantly at least to one fuzzy set.
- Normalization. All the fuzzy sets should be normal.
- Use of convex fuzzy sets.

There are different methodologies to calculate the interpretability of a knowledge base. Several authors have been interested in the subject of interpretability developing different taxonomies.

The most of works considering complexity at rule base level, because it is the most extended way to work.

Guaje is a tool that ensures interpretable system if all the conditions enumerated above by means of imposing the use of linguistic variables, which are described by Strong Fuzzy Partitions (SFPs) and compute the interpretability index.

SFPs are partitions with membership functions of triangular, semi trapezoidal or trapezoidal shapes. Partitions satisfy the conditions in follow equations:

$$\forall x \, \epsilon \, U, \sum_{i=1}^{E} \mu A_i(x) = 1 \qquad (11)$$

$$\forall A_i \, \exists x, \; \mu A_i(x) = 1 \qquad (12)$$

Where U is the range, E is the number of term, $\sum_{i=1}^{E} \mu$ is the membership degree of x to the A_i fuzzy set. These partitions include only basic labels A_i, but in the rule description they can be combined using OR or NOT. Not composite labels by the equation:

$$NOT(A_i) \Rightarrow \begin{cases} A_1 \; OR \; \cdots \; OR \; A_{i-1} \; = \; Smaller \; than \; A_i \\ A_{i+1} \; OR \; \cdots \; OR \; A_m \; = \; Bigger \; than \; A_i \end{cases} \qquad (13)$$

And OR composite labels are defined as the convex hull of the combined terms. [16] [17][18]

Several authors have proposed different taxonomies as a way to analyse the interpretability aspect. There are two kinds to take into account to analyse the interpretability: Complexity and Semantics (Explained in detail, later). From both kinds of measures, linguistic fuzzy partition and rule base, and KB component are taken into account. The taxonomy can be divided by complexity versus interpretability and rule base versus fuzzy partition.

There are four derived groups from two above criteria:

- Q1- Complexity at the Fuzzy partition level
- Q2- Complexity at the Rule Base level
- Q3- Semantics at the Fuzzy partition level
- Q4- Semantics at the Rule Base level

Q1- Complexity at the Fuzzy partition level

In this section, we consider the number of variables and the number of MFs.

- The readability of the knowledge base is improved if the number of variables is smaller.
- It is necessary to have a good number of memberships function. It must not exceed of 9, because it is the number of entities a human being can handle.

Q2- Complexity at the Rule Base level

In this section, we consider the number of rules and the number of conditions.

- The minimum number of rules for which a system to be satisfactory. The smaller number of rules, higher the interpretability.
- The minimum number of conditions in the antecedent per rule for which a system to be satisfactory. The number must not exceed the limit of 7 more or minus 2.

One of propose to reduce the complexity is the selection of rules, considering three main objectives. To minimize the number of selected rules, to maximize the number of training patters and to minimize the total rule length.

The number of rules of the system depends proportionally on the number of variables and the memberships functions. Thus, Q1 is related to Q2.

Q3- Semantics at the Fuzzy partition level

Depending on the design process, an accurate system can be produced complex fuzzy partitions with the interpretability very low. To avoid this, some constraints are applied to the memberships function to improve these conditions.

Some of properties are:

- Coverage. Membership Function must cover the universe of discourse (U) and every data point must belong to an MF almost.
- Normalization. If there is a data point of U with value 1 in a MF, the MF is normal.
- Complementarity. The sum of all values inside of MFs of U should be near to one. Uniform distribution.
- Distinguishability. Each MF must have a semantic meaning and be easily differentiated from the rest of MFs of the corresponding variable.

Q4- Semantics at the Rule Base level

Assuming that Q3 is interpretable, semantics at the rule base level takes into account the consistency of rules, rules fired at the same time, the transparency of rule base.

- When we talk about consistency, we talk about the absence of contradictory rules in RB. The rules with similar premises should have similar consequents.
- Minimizing the number of rules firing that are activated for a given input because it is a promise to preserve the individual meaning of rules of RB.

Other authors such as Mencar [49] propose a taxonomy considering restrictions for the universe of discourse and fuzzy sets, for rules and for learning algorithms, among other parameters.

Alonso [48] proposes a taxonomy that takes into account two factors to assess interpretability. These two factors are the description, which the system is considered as a whole that describes the overall behaviour and global trend. Explication, each individual situation is analysed explaining specific behaviours. And the taxonomy only take into account the complexity quadrant. This methodology is used in this work (explained in more detail, later). To compute the interpretability following six criteria:

-Total number of rules (Q1)

-Total number of premises (Q2)

- Number of rules, which use one input

- Number of rules, which use two inputs

- Number of rules, which use three or more inputs

- Average the total number of label by input.

As previously mentioned, this work reduces the time and energy of DPC and make the system or model chosen interpretable by human. To achieve the success of this model, the combination of algorithms with the human experience and interpretation is necessary.

The data science work does not finish when the model is built, but it is an iterative process with feedback loops provided by experts, providing a robust result and with a degree of comprehension for the human.

Interpretability is necessary because most machine learning models have been called ‘Black Boxes’. This means that although the results are good, the human cannot explain the logic behind these algorithms.

In this way, it is necessary to know why the model has obtained these results. This has to do with the impact that the model applied to real life would have, that is, depending on the objective of the algorithm.

Some of the advantages that interpretability produces are to provide reliability to the results obtained, help in debugging, help to detect if it is necessary to collect new samples for learning process, help a person in decision making and greater safety and robustness in the final model.

The learning process could initially be considered from the collection raw data of the real world that are used for predictions and finally the process in which the human is able to understand the process.

The first phase is WORLD, it contains everything that can be observed and is of interest. Something wants to be learned from the world and interact with it. Data is captured from this phase.

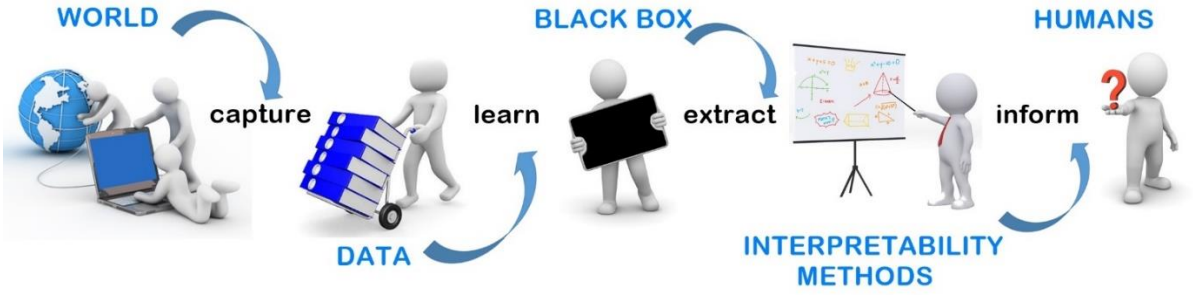

Figure 9. LEARNING OVERVIEW.

The second phase is the DATA. It is necessary to digitize the data captured from the world to be stored and processed by computers. This phase contains texts, images and anything.

By adjusting machine learning models of the previous layer, the BLACK BOX phase is obtained. These algorithms learn with real-world data to find structures or make predictions.

The next phase is INTERPRETBILITY METHODS. It is responsible for dealing with the opacity of Black Boxes, taking into account why some decisions were made or not for the results.

The last phase is the HUMAN. Humans are ultimately the consumers of Black Boxes explanations. Through all these layers, and if the system is interpretable, the human will be able to interpret the result obtained. [19]

3.7 MULTI-OBJECTIVE OPTIMIZATION

The main objective of this work is to analyse the interpretability of fuzzy systems that they reduces the power, time consumption of Data Processing Centers (DPC).

Previously, there has been a reduction in power but without reducing the processing time. This is because all fuzzy logic system and Pittsburgh algorithms are based on using energy as fitness only, regardless of execution time.

The execution time and power are related to the performance of the processor and the virtual machine where the tasks will be executed. If the performance is greater, then the power will be greater, but the execution time will be less.

This multi-objective optimization obtains a balance between these two parameters, achieving values that satisfy both requirements and also, choosing the most interpretable KB. This optimization has been obtained through the Pareto front.

Pareto front is a set of no dominated solutions, being chosen as optimal, if no objective can be improved without sacrificing at least one other objective. A solution x is referred to as dominated by another solution y if, and only if, y is equally good or better than x with respect to all objectives. [20]

The set that corresponds to Pareto set is composed of all Pareto optimal decision vectors and the plot of these vectors is called the Pareto front. [21]

4. IMPLEMENTATION

In order to analyse the interpretability of the systems that optimize the power and time of a Datacenter, it is necessary to use a tool called WorkFlowSim-DVFS. This simulator provides the value of the power and time consumed in a Datacenter, based on rule-based systems. This tool simulates a real scenario of a Datacenter in Cloud Computing.

Thanks to these simulations, knowledge bases will be obtained with a rule base that will be analysed and evaluated to obtain their interpretability index.

This is the main section where it will be explained how the assessment of the interpretability will be and also the changes made in the WorkflowSim-DVFS, Guaje tool to adapt these tools to the conditions of this work and the development of a hierarchical fuzzy system to assess the interpretability index. In addition, the integration of the two software with Matlab and how the experiments have been performed will be explained. That is, the general process of this work, from the modifications to the results of the experiments and the evaluation of the interpretability. The simulation environment will also be analysed, explaining the problems encountered and the reason for the final solution.

4.1 WORKFLOWSIM-DVFS

As previously mentioned, the interpretability will be analysed in fuzzy scheduler that optimize power and time consumed in Datacenter. WorkFlowSim is a free code simulator that combines two rule-based expert systems and the Dynamic Voltage and Frequency Scale (DVFS) algorithm to provide power-aware schedulers.

This simulator is the combination of WorkFlowSim[22] and CloudSim[46] simulators.

On the one hand, the CloudSim (developed by Weiwei Chen [46]) simulator provides an estimate of the execution time of the executed tasks, but does not implement any tool that provides the information about the energy consumption produced by the processing of the tasks. In addition, it is able to decide on its own the tasks that will be executed. The drawback is that it does not provide a real scenario and this is the reason why it cannot provide information on energy consumption.

Due to this need to know the power consumption, the extension of CloudSim with DVFS appeared. In this extension, the ability to estimate the consumed energy is added. To do this, when the Cloudlet execution is carried out, instead of calculating the execution time of this cloudlet, the process is repeated several times calculating the energy consumed. This process last until the current time reaches the end time of the cloudlet.

The end time is the sum of the current time of the simulator and the time estimated by the Datacenter to process the task. One the task loop is finished, the value of the power

consumer by each task and its execution time will be obtained. In this way, the system will obtain the value of the total energy consumed and the total execution time.

Although with this version of CoudSim it is already possible to estimate the energy consumed, it still does not allow the processing of tasks in a real scenario.

On the other hand, the WorkFlowSim (developed by Weiwei Chen [22]) simulator has been used as it allows to simulate traces of workflows of real Datacenters. One of the advantages of using workflows is the order of processing of each of the tasks, taking into account the restrictions that some tasks may have. That is, there are tasks that cannot be processed until other main tasks are processed, as would be done in a real scenario of a Datacenter.

It use Directed Acyclic Graph (DAG) in XML format (DAX). DAX is a description of an abstract workflow in XML format used in Pegasus as the primary input [23] and they are converted to an image format so that they can be understood more easily.

In this work, the example of DAG used has been 'Montage'. The figure 18 shows the workflow graph of Montage DAG. The Montage example is used to construct custom science-grade astronomical image mosaics on demand based on several existing images. The inputs to the workflow include the input images in standard FITS format taken from image archives such as 2MASS and a 'template header file' that specifies the mosaic to be constructed. [24]

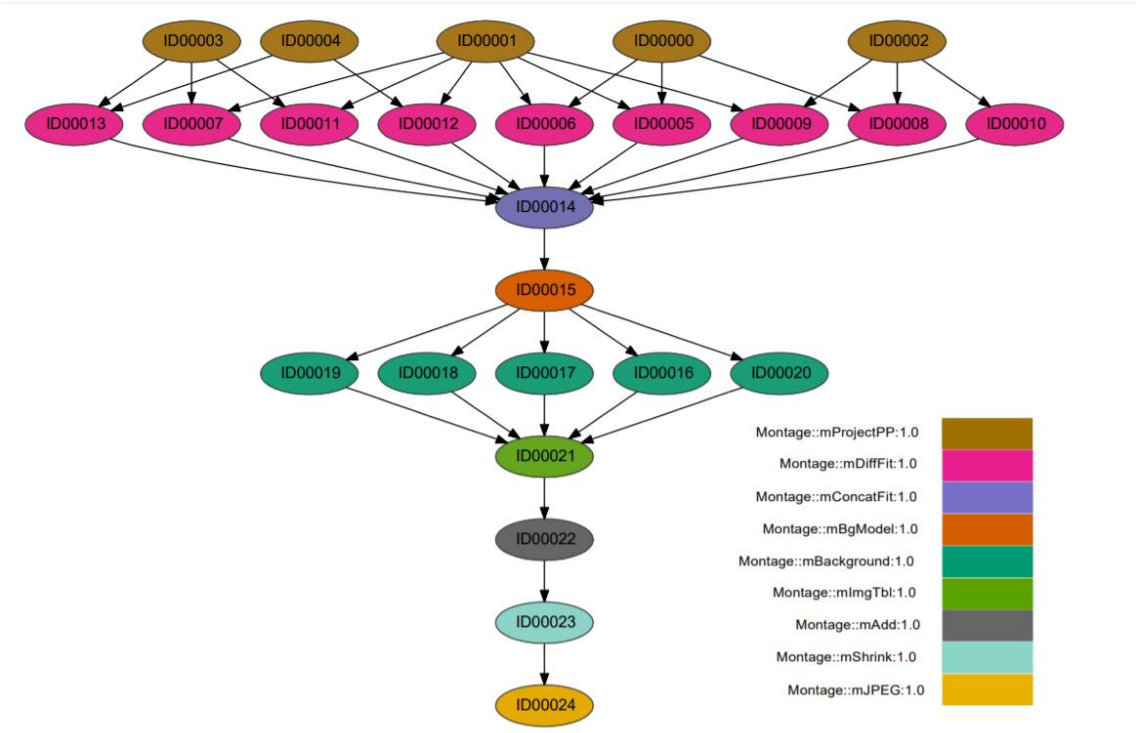

Figure 10. MONTAGE WORKFLOW.

(Font: [10])

Using workflows has numerous advantages. On the one hand, the order of processing tasks is established, including restrictions. That is, preventing some tasks from being executed before others tasks. This happens when traces of real tasks are simulated, which some depend on others tasks. Each task has a different size for input and outputs. To optimize cloud environments, these simulations are used based on real workflows.

This simulator, being based on the two previous simulators, allows creating the most real environment possible, since if the experiments are not performed on a real Datacenter scenario, it cannot be ensured that the algorithm works correctly in real life.

This simulator provides an estimate of energy consumed and performs real traces. In addition, it is designed to work in several real scenarios and to include a wide repertory of DVFS governors.

As mentioned above, one of the main objectives of this work is the saving of power consumed in Datacenters. For this, there are different techniques that reduce power consumption. The technique chosen in this simulator is the DVFS, already mentioned in the Introduction section.

DVFS is a strategy for reducing energy consumption, allowing the change of the performance of the machine dynamically, based on the workload. That is, it allows workflows to be processed considering the regulation of the frequency and voltage of the resources. It is the only simulator that executes real workflows taking into account the event of failures and different loads of work on the server, and makes an estimate of energy consumption. This technique allows a power increase when the performance is higher. To reduce the power consumption, it is necessary to reduce the performance; therefore, it is not advisable to set the power to a minimum. The frequency and voltage must be set to a minimum to comply with SLA. In this way, it is proposed to reduce the power consumed, since in the past, the servers were kept at maximum performance by consuming the maximum power.

Some of the other techniques to save energy could be to limit power. This technique is similar to DVFS but instead of adjusting the power values of each server, what it does it set the levels of the servers in a cluster.

Another technique is server virtualization that is, running several VMs on a single server. In this way, only one server would run and a single server consumes less power than a group of servers. In addition, the fact that you can run several VMs on one server, allows machines to migrate from one server to another, causing all VMs to be executed on the same server (provided that the sum of work can be executed on that server) and thus get rid of the static component of the power consumed from inactive servers.

Other techniques are the consolidation of workloads of all servers in the minimum number of them running in high performance and turning off those that are inactive or the possibility of adding a system that indicates the workload is also considered, thus avoiding the delay when turning on the servers.

The proposal simulator includes several types of DVFS governors, adjusting frequency levels considering to the workload to be executed.

In conclusion, the simulator used is a simulator with the ability to calculate total power consumption, use traces of real workflows and also incorporates the DVFS technique, which it will adjust the frequency and voltage of the CPU to minimize the energy consumed. This simulator will be integrated into Matlab as an executable (.jar), which will be explained in detail, later. Figure 11 shows the inputs and outputs of the simulator necessary for integration with Guaje program in Matlab.

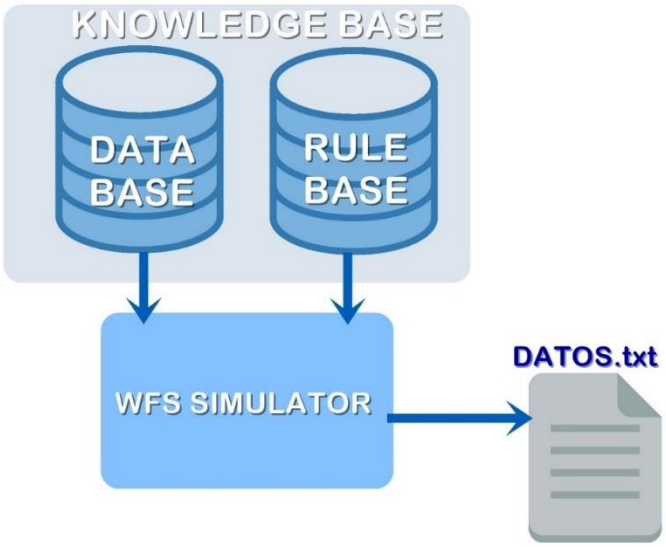

Figure 11. SIMULATOR INPUTS AND OUTPUT FILES.

In the figure 11, the input files are the database and the rules base, which together make up the knowledge base (KB). As will be explained later, the simulator includes FRBS system; therefore, this knowledge base is necessary to perform the actions of this simulator. 'DATOS.txt' will be the output file of the simulator. This file is composed of the necessary data that Guaje needs to calculate the interpretability index along with the KB (Explained in detail in section 5.2).

The simulator needs two rules bases, one contains the rules for VM Scheduler and one for the tasks of the scheduler. Each rule base will store different objects, since each one belongs to a different scheduler. (Explained in detail in section 5.1.1).

4.1.1 ENTITIES

The simulator has composed by several entities: Planner, Clustering Engine, Engine, Scheduler and Datacenter.

- **PLANNER.** It is responsible for initializing the system, converts the DAG file into xml (DAX file) and analyzes it to obtain the tasks. These tasks have been sent to Clustering Engine to convert them into jobs.
- **CLUSTERING ENGINE.** Group tasks into jobs to reduce overheads.
- **ENGINE.** The jobs are sent to the Engine where they are selected taking into account the workflow. That is, it ensures that the simulation follows the order established in the DAG file, avoiding the execution of secondary tasks before the main ones. In addition, Engine is responsible for sending the tasks that can be processed to the Scheduler.
- **SCHEDULER.** The Scheduler will receive the tasks from Engine and send them to the Datacenter for execution. This one decides on which VM each task will be executed.
- **DATACENTER.** Datacenter is a physical or virtual infrastructure composed of computer systems that can process, serve or store data. It provide data storage, backup service, data recovery and information management for companies. [25]

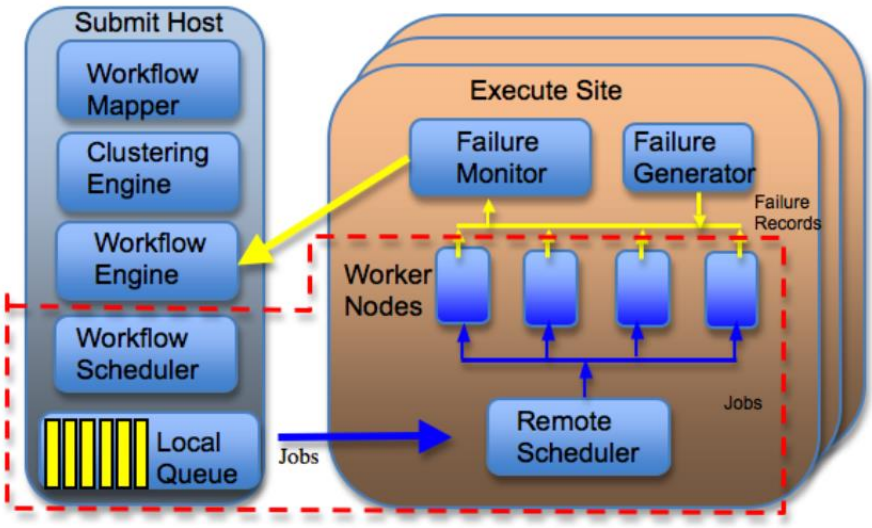

Figure 12. WORKFLOWSIM OVERVIEW. THE AREA SURROUNDED BY RED LINES IS SUPPORTED BY CLOUDSIM.

[Font: 10.1109 / eScience.2012.6404430]

One or more Datacenters can form a scenario. These Datacenters are composed of virtual machines that are not assigned tasks directly, but it is the Scheduler who decides what task is executed in each VM. Virtual machines are heterogeneous, that is, each one has a different performance. Once the tasks are executed on each machine, the results are sent to the user.

First, there are some physical machines where virtual machines will be stored. That is, the Scheduler will choose which physical machine has the necessary characteristics for the creation of the desired virtual machine. Thus, the execution time and energy consumption value will be different on several machines.

Once the virtual machines are created, the tasks are executed. At this stage, the Scheduler chooses in which VM the task are executed.

4.1.2 TASKS PROCESSING

'Selection' is the degree of selection each VM has to process the task. The high value of the output, the more appropriate will be the VM of being chosen. This process is repeated until all tasks are processed.

The process selects the VM that would obtain the lowest energy consumption when processing that task, based on the length of the task and the performance of the VM.

This occurs because virtual machines are not directly allocated, since in this way, many resources are wasted, consuming more energy and probably longest execution time.

4.1.2.1 FIRST STAGE: VM SCHEDULING

In this stage, physical machines that store virtual machines compose the Datacenter and these virtual machines are created.

Here, it is decided on which physical machine, each virtual machine is assigned, considering as metric the ratio watts/MIPS in the different hosts. This decision is made by Scheduler. The parameters used are the utilization, MIPS and the availability of machine.

In this way, VMs are allocated in Hosts, taking into account the host that need the less amount of watts/MIPS, optimizing the overall energy consumption of the system, as each VM will be consuming the minimum power as possible.

As this simulator uses DVFS, it ensures a reduction of energy in the use of the processors, maintaining a low utilization of the machines. Assuming this a saving of energy of the total of machines.

The user will request the assignment of a physical machine, which leads to the creation of a VM. The Scheduler will do this creation. The user in the request indicates the value of MIPS needed to measure the performance of the VM. The Scheduler receives the request and is responsible for assigning the VM to the corresponding physical machine taking into account the MIPS parameter.

Each machine has processing elements (PE) and the main objective is that there are no machines that have all their Pes busy when there are machines that haven them inactive. The higher number of Pes, the greater the energy consumed by the machine, since it implies a greater use of the CPU. If the CPU utilization exceeds the DVFS threshold, then this technique will increase the voltage and frequency of the CPU, increasing the energy consumption. [10]

The Scheduler can estimate which machine is the one with le lowest energy consumption by the POWERFUZZYSEQ algorithm. This algorithm looks for which machine is the one that can consume less energy for the MIPS required by the user for the VM.

- **FRBS PARAMETERS.** Here, four parameters are take into account: Utilization of the physical machine, power consumed in the host, total MIPS of the physical machine and MIPS requested by the VM. These values are normalized (range between 0 and 1) to be within the range of MFs.

4.1.2.2 SECOND STAGE: TASKS SCHEDULING

Here, the tasks that will be processed in the VMs are taken into account. That is, it is decided on which machine will execute the tasks. The simulator considers variables related with the power and the time.

For example, a variable is the length of the tasks. It is measured in Millions of Instructions (MI) and deadline define each task in seconds (s), so its performance is measured in MIPS.

The deadline of each tasks is the maximum time that can take for the execution of a task. The Quality of Service of the Service Level Agreement of the user fixes this value with the cloud provider.

The first thing that is checked is which VM can accomplish the deadline of the tasks and thus, discard VMs that cannot satisfy these conditions.

The simulator does not have the deadline parameter, but it must be taken into account since the energy depends on the time.

The scheduler is able to obtain a minimum value of performance and thus, know which virtual machine can execute the tasks and which not. Then, it will calculate which VM consumes less energy when executing the task and that will be the chosen VM.

In conclusion, the lower the value of MIPS, the lower the energy consumed but the longer the execution time.

However, in this work, it seeks to optimize the execution time and power consumption. Then, a FRBS are needed with related variables for each objective. Thus, depending on the fitness, FRBS will have some input variables or others, the consequent being the same (the decision to process the tasks, in question, in one VM or another.

In the next section (4.1.3 Workflowsim-DVFS Schedulers), FRBS will be explained in detail.

4.2 INTERPRETABILITY

For the development and analysis of this work, two methodologies have been used, one that implements Guaje and another one a fuzzy hierarchical system implemented.

Guaje is a java environment for generating understandable and accurate models. It is a free software tool with the aim of supporting the design of accurate and interpretable fuzzy system combining several pre-existing open source tools. This tool is used specially for interpretability issues. [16][17][18] [27].

Guaje is a very powerful tool, which among many other actions allows calculating and analysing the interpretability of a rule base. That is why it has been used in this work.

Interpretability is the main requirement and it is taken into consideration along the whole modelling process. Guaje combines several tools (KBCT, FisPro, Xfuzzy, ORE, WEKA and Matlab Fuzzy ToolBox) with the implementation of HILK (Highly Interpretable Linguistic Knowledge) fuzzy algorithm.

Guaje considers two sources of knowledge, experimental data and expert knowledge.

This tool allows, among others tasks, to extract knowledge about the specific problem to treat, induce partitions from data, integrate expert-based and data-based partitions, induced rules, integrate expert-defined and data-induced rules, check consistency and simplify the knowledge base (KB), optimize the KB, evaluate the KB quality according to accuracy and interpretability, and save and export the KB.

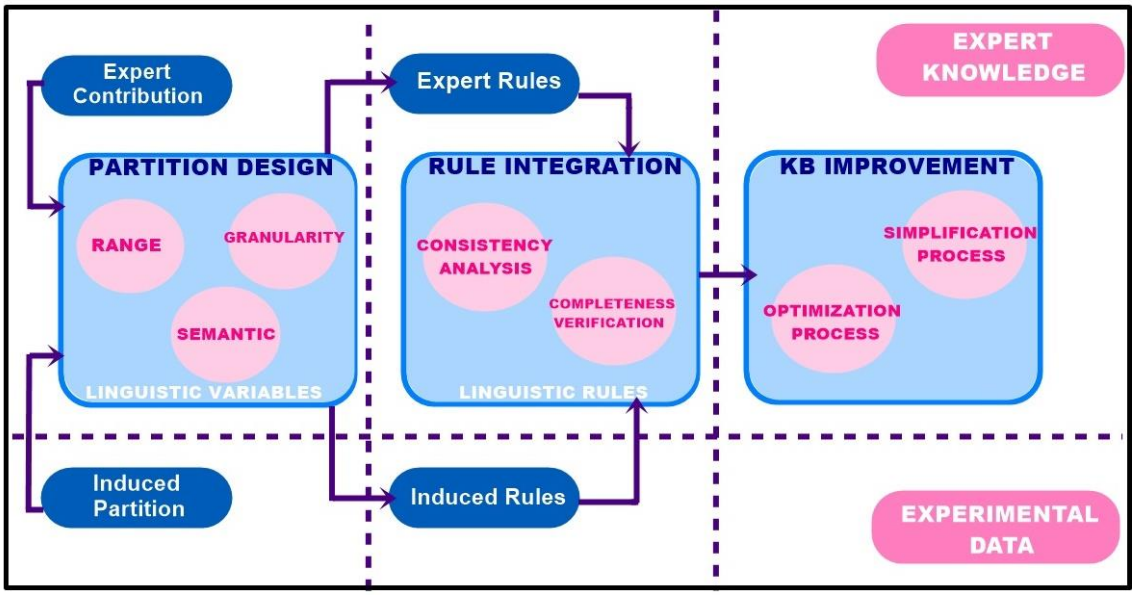

Figure 13. EXPERT KNOWLEDGE AND EXPERIMENTAL DATA.

In addition, Guaje works with expert knowledge and experimental data. [26].

It create knowledge bases and the system is composed by rules with premises and conclusion. For example, IF condition THEN conclusion. A fuzzy rule is considered compact when a premise is defined by a set the input.

The knowledge extraction process consists of the next steps:

1. Taking into account experimental data and expert knowledge, define the most predominant variables. With this information, it is possible to provide expert partitions and induced partitions.
2. Integrating both expert and induced partitions. First, partition design is created with all information about partition integration, then the rule definition. Guaje permits selecting from 2 to 9 terms for each input variable. Rule definition is a block to define rules of FRBS.
3. Describing the system behaviour. Rues are created by experimental data (induces rules) and by expert knowledge (expert rules). In our case, we only have expert rules provided by genetic algorithm.
4. Integrating both expert and induced rules.

This work is focused in the interpretability aspect.

4.2.1 ASSESSMENT INTERPRETABILITY

In this section, how to obtain the index of interpretability of a knowledge base (KB) according to the hierarchical methodology will be explained. [26]

Guaje [27], as discussed above, implements a methodology to evaluate the interpretability. This algorithm is a methodology for building interpretable fuzzy systems for classification problems and implement the four quadrants.

Base on the nature of the interpretability index and the elements of the fuzzy knowledge base, four derived groups of the above criteria (Complexity vs. Semantic and Fuzzy partitions vs. rule base) are obtained, as commented above.

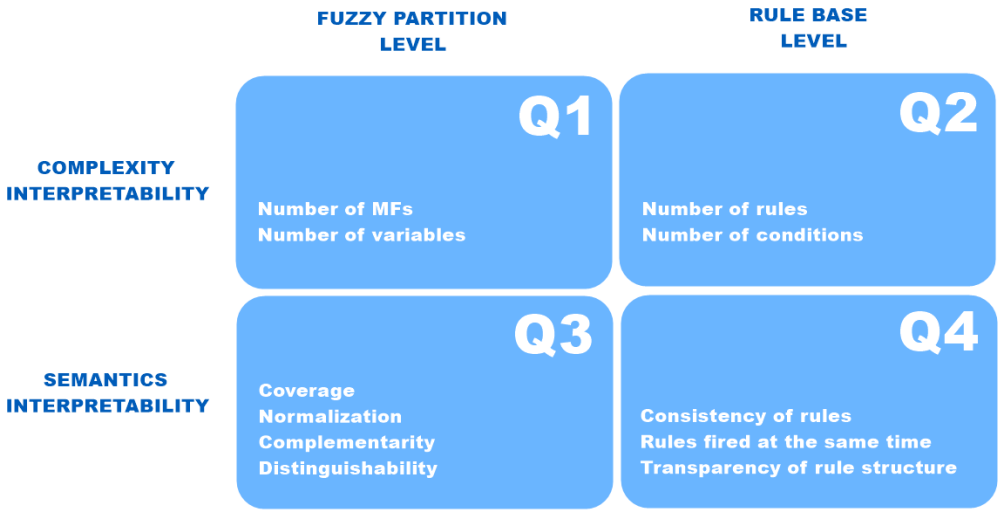

Figure 14. TAXONOMY TO ANALYZE THE INTERPRETABILITY OF LINGUISTIC FRBSS.

However, in the development of this project, the methodology proposed by Alonso [47] has been carried out.

In this methodology only the complexity of the system is taken into account. This complexity convers quadrant Q1 and Q2.

Q1- Complexity at the Fuzzy partition level

In this section, we consider the number of variables and the number of MFs.

- The readability of the knowledge base is improved if the number of variables is smaller.
- It is necessary to have a good number of memberships function. It must not exceed of 9, because it is the number of entities a human being can handle.

Q2- Complexity at the Rule Base level

In this section, we consider the number of rules and the number of conditions.

- The minimum number of rules for which a system to be satisfactory. The smaller number of rules, higher the interpretability.
- The minimum number of conditions in the antecedent per rule for which a system to be satisfactory. The number must not exceed the limit of 7 more or minus 2.

One of propose to reduce the complexity is the selection of rules, considering three main objectives. To minimize the number of selected rules, to maximize the number of training patters and to minimize the total rule length.

The number of rules of the system depends proportionally on the number of variables and the memberships functions. Thus, Q1 is related to Q2.

Guaje uses these four criteria to calculate the interpretability index. However, in this work, the methodology adopted to calculate the interpretability index will be based on the taxonomy proposed by Alonso, et all. [47]. In this taxonomy the total number of rules (Q1), total number of premises (Q2), the number of rules which use one input, which use two inputs and three or more inputs and the total number of labels defined per variable are taken into account. These criteria are taken as inputs of a fuzzy system.

Alonso [47] considering interpretability from two points of view: Description (System structure readability) and Explanation (System comprehension). [28]

Both cases seek contradictory goals. In first case, Description, the system has to be transparent to present the system as a whole describing its global behaviour and trend, and it prefers rules as compact as possible. In second case, Explanation, each individual situation is analysed explaining specific behaviours for specific events and it favors the use of complete rules.

In this work, we are going to focus in the first case (Description) to obtain the interpretability index. The description system present the system as a whole. First, we assume that the more compact KB, the simpler its understanding, so the interpretability is higher and we can assess the simplicity of the linguistic description.

To make an understandable and accurate system, interpretability works in two levels: rules and variable. We have taken into account the reasonable number of terms defined by each variable (7 more or minus 2 terms). As for the rule level, we must analyse the interpretability of each individual rule and the interpretability of a set of rules, because it is

possible that several rules can be activated at the same time. A rule base depends on the number of rules and the number of inputs used by rule. In conclusion, the interpretability is higher if the number of rules is smaller and if the number of inputs used by rule for a given number of rules is smaller.

The methodology followed can be seen in [29].

Quantifying a RB interpretability is not easy. However, the rule base is divided in three terms. Complexity of a classifier measured as the number of rules divided by the total number of premises, Partition index is computed as the inverse of number of level (minimum two in a partition) for each input variable, and the last one the coverage degree of the fuzzy partition, this value is equal to 1 for SFP.

For assessing interpretability, a hierarchical fuzzy system compose by the rule base dimension (RB1), the rule base complexity (RB2), the rule base interpretability (RB3) and finally, Interpretability Index (RB4) has been carried out.

This hierarchical fuzzy system is a proposal for measuring the interpretability of linguistic KBs. To understand this system, it is needed to explain in detail each internal rule base.

This system works with four rule bases linked:

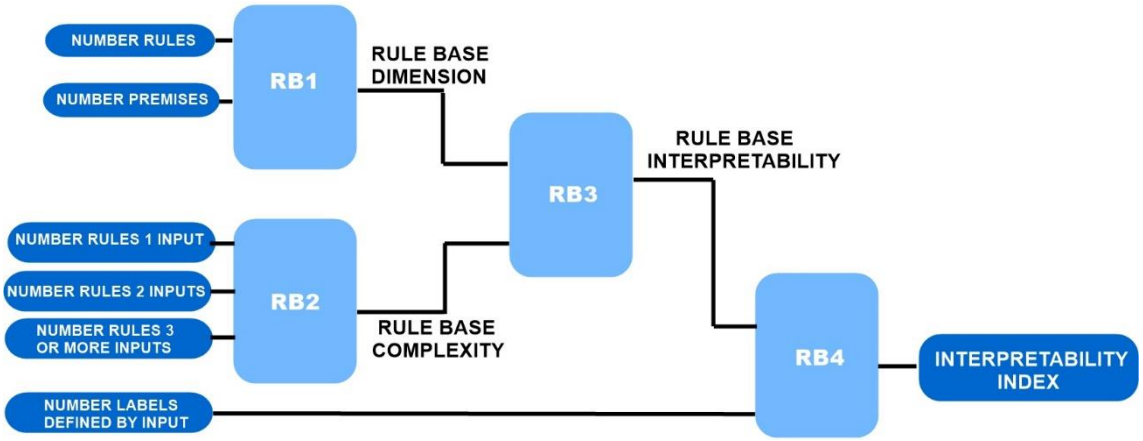

Figure 14. HIERARCHICAL FUZZY SYSTEM.

The hierarchical fuzzy system for assessing interpretability is represented in the figure 16, where you can see the following four rule bases with their inputs and outputs.

RB1- RULE BASE DIMENSION

This rule base take into account the total number of rules and the total number of premises to make an estimation of rule base dimension.

To make this estimation, RB1 implements several fuzzy rules.

If total number of premises is low, then rule base dimension mainly depends on total number of rules. If total number of premises is high, then whatever the total number of rules the system output is high.

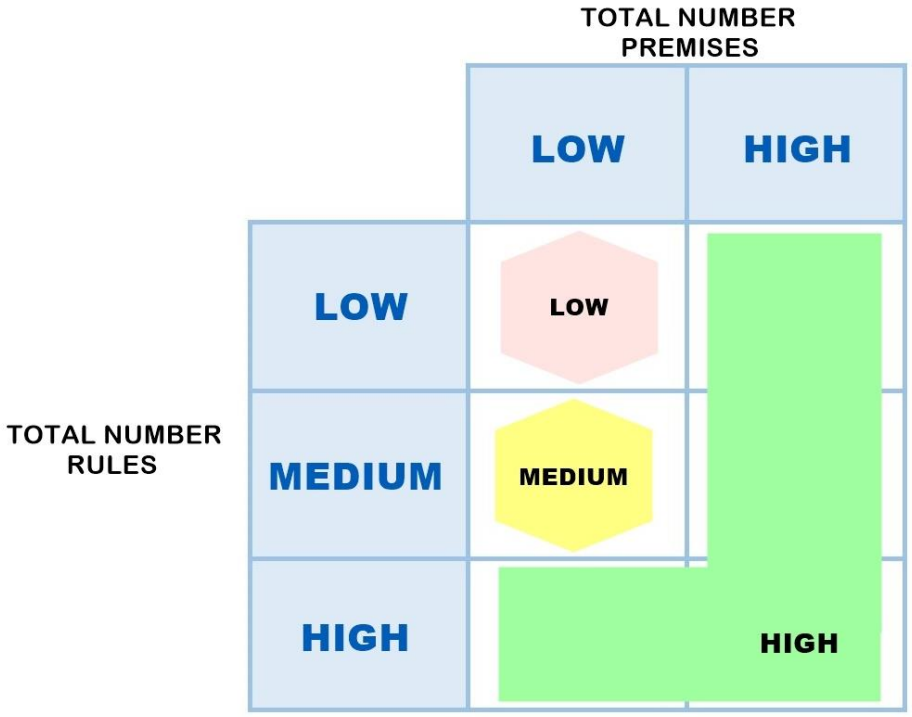

Figure 15. RB1 RULE BASE DIMENSION.

RB2- RULE BASE COMPLEXITY

At the same time RB1, RB2 evaluates the rule base complexity; take into account the number of inputs, which use by rule. RB2 makes an estimation about the number of rules, which use one input, number of rules, which use two inputs, number of rules, which use three or more inputs, and the total number of labels defined by input.

RB2 and RB1 use the same information but it is analysed with a different meaning. RB2 need to analyse separately each internal rule base to asses this system complexity.

The fuzzy system in RB2 is composed by several rules.

The larger number of rules, which use three or more inputs, the higher the rule base complexity. Two tables can explain this behaviour.

The first table deals with a small number of rules with three or more inputs, whereas the second table considers a large number of those rules.

In Figure 18, if there is a small number of rules with three or more inputs and whatever the value the number of rules, which use 1 input, and the value of number of rules which use 2 inputs is low, the rule base complexity is low.

However, in Figure 19, if the number of rules, which use 3 or more inputs, is high and whatever the value the number of rules, which use 2 inputs, and the value of number of rules which use 1 input is low, the rule base complexity is high.

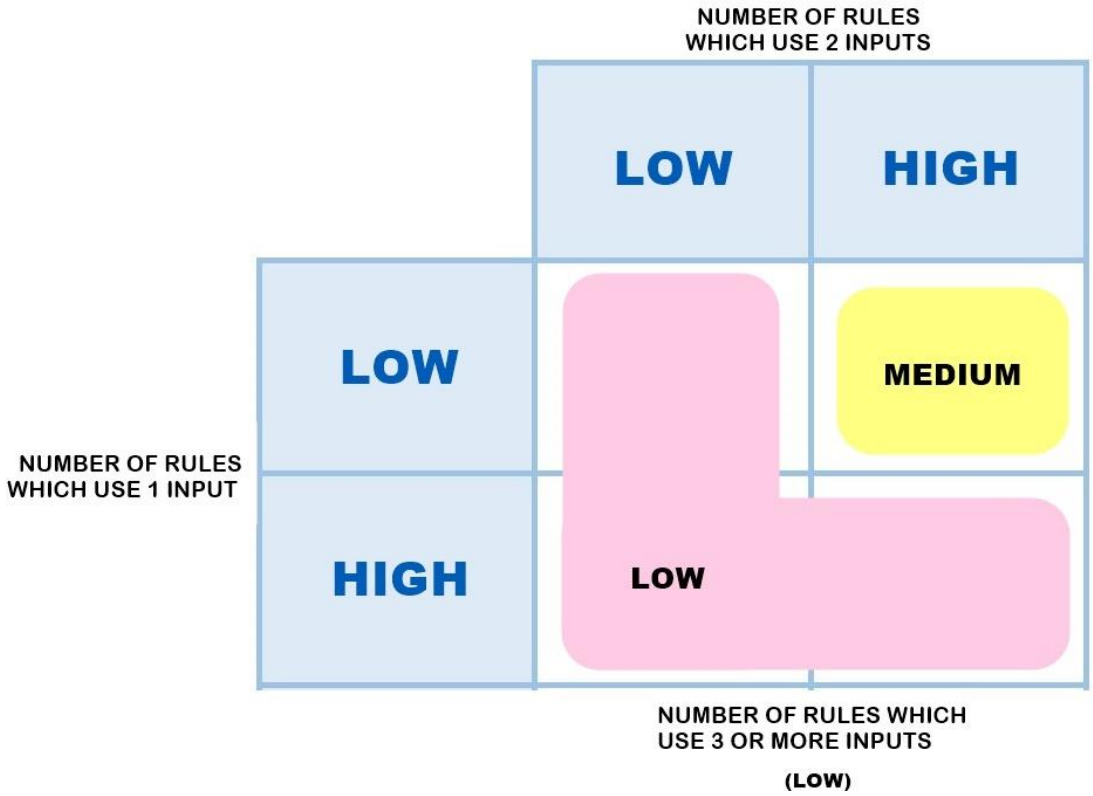

Figure 16. RB2 RULE BASE COMPLEXITY (LOW)

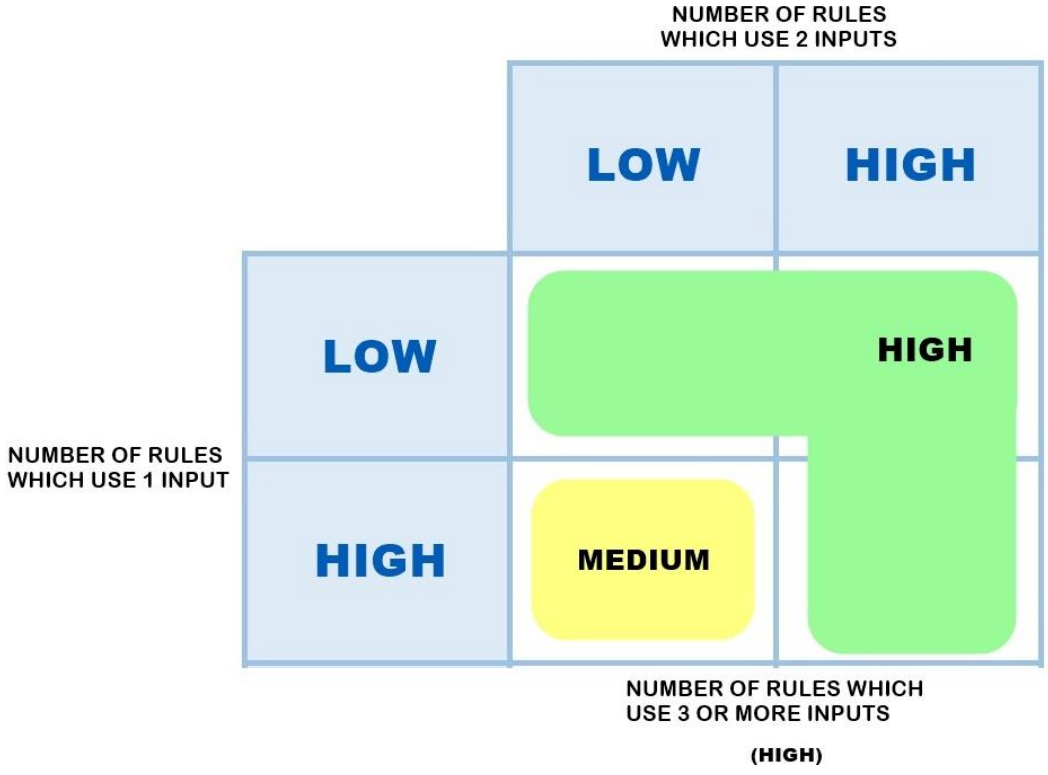

Figure 17. RB2 RULE BASE COMPLEXITY (HIGH)

RB3- RULE BASE INTERPRETABILITY

Rule base interpretability is a combination of rule base dimension and complexity (RB1 and RB2).

The larger the rule base dimension and/or complexity the smaller the rule base interpretability. So, in this case, mainly depends on the rule base dimension.

If rule base dimension is high and whatever the complexity of rule base, the interpretability is low or very low. Therefore, the best value of interpretability is obtain when the rule base dimension is low and rule base complexity is low or medium.

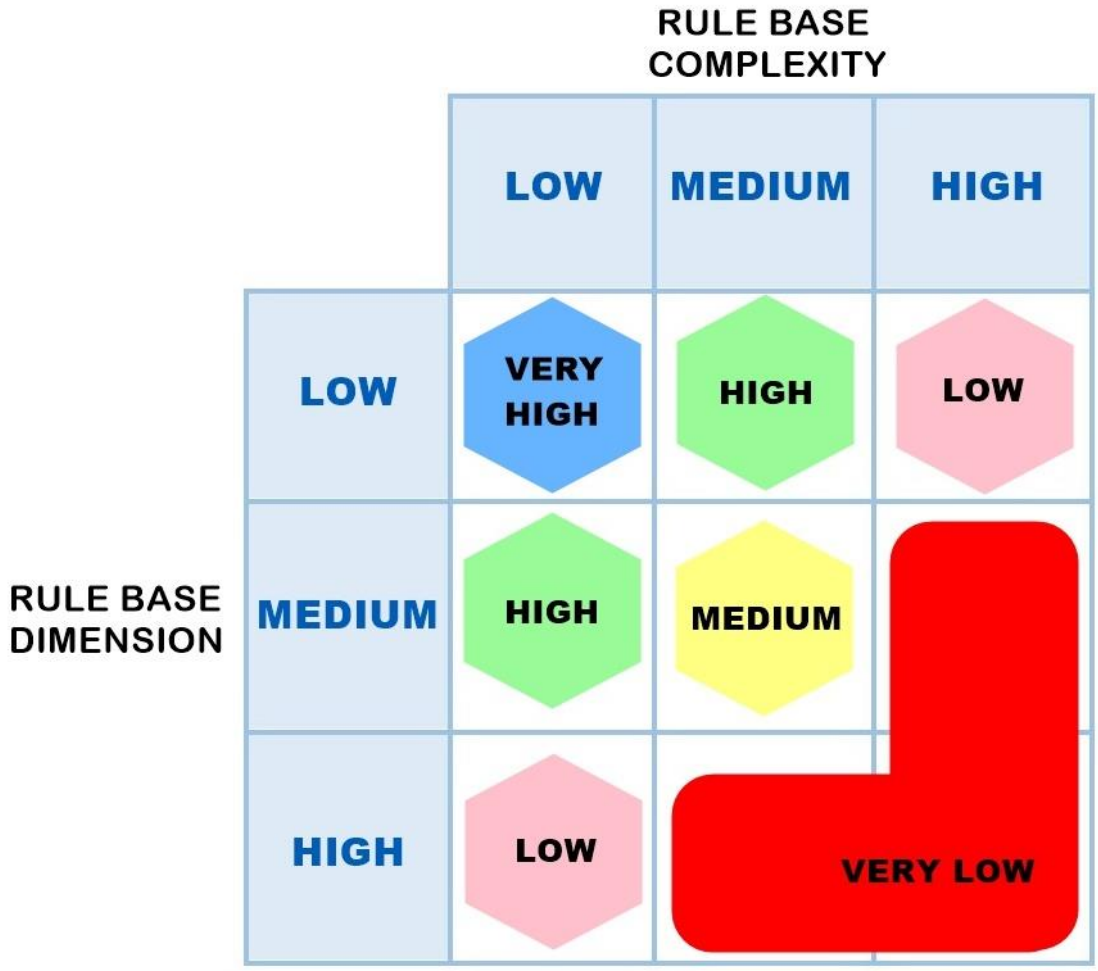

Figure 18. RB3 RULE BASE INTERPRETABILITY.

RB4- INTERPRETABILITY INDEX

This base rule integrates RB3 with an evaluation of the interpretability of the variables, considering the average number of labels defined by input and assuming that we are using SFPs.

If average number of labels defined by input is low, then the larger rule base interpretability the larger interpretability index. Nevertheless, if average number of labels defined by input is high, the interpretability index reproduces the rule base interpretability but proportionally reduced.

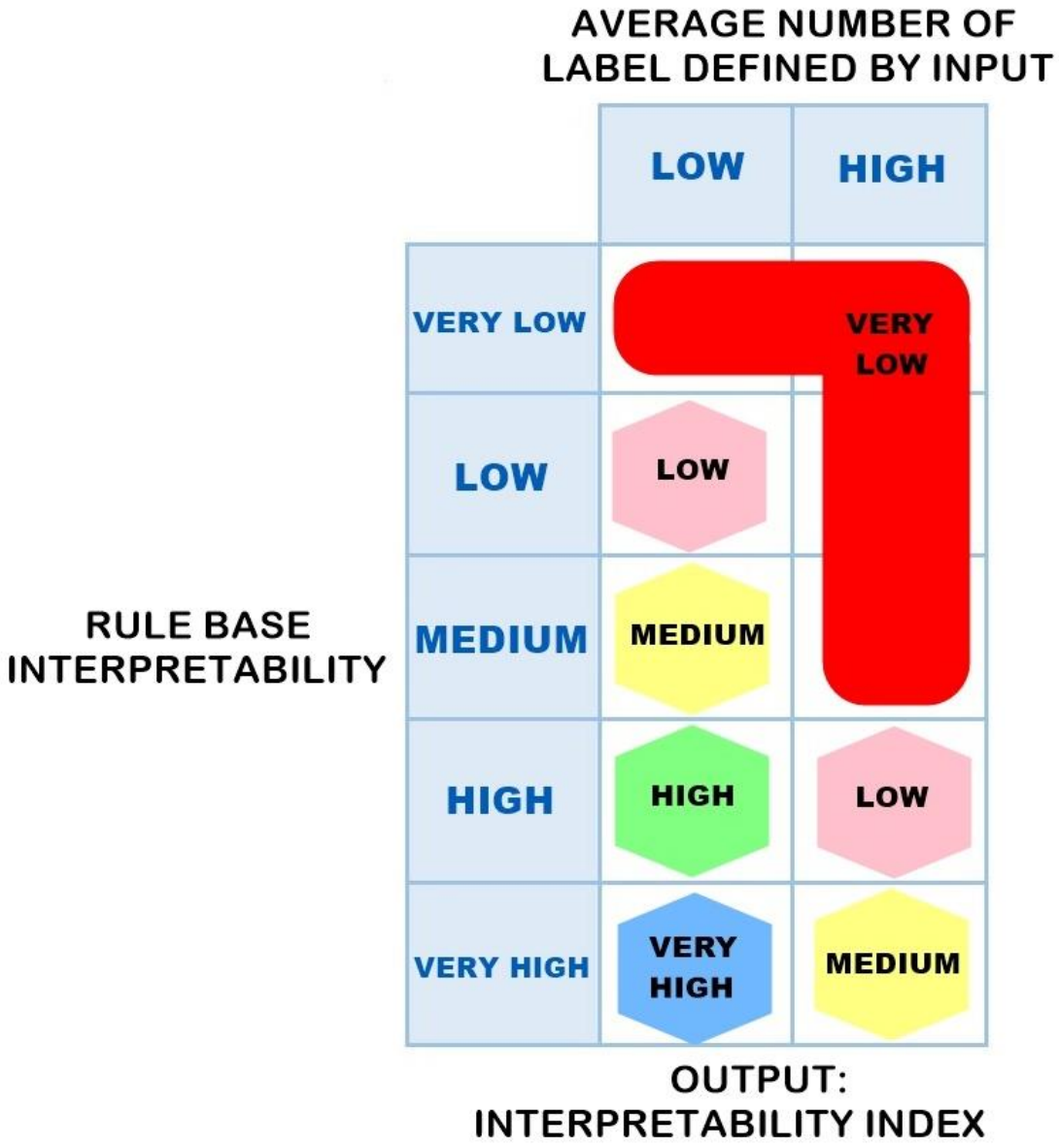

Figure 19. RB4 INTERPRETABILITY INDEX.

The following table shows the variables used as input of the hierarchical system and its universe of discourse and number of labels. [47]

VARIABLE	UNIVERSE	LABELS
TOTAL NUMBER RULES	[N Comp* N]	3
TOTAL NUMBER PREMISES	[N (2*Comp)* N]	2
RULE BASE DIMENSION	[0 1]	3
NUMBER RULES WHICH USE 1 INPUT	[N N*Comp]	2
NUMBER RULES WHICH USE 2 INPUTS	[0 $N*\frac{Comp}{2}$]	2
NUMBER RULES WHICH USE 3 OR MORE INPUTS	[0 $N*\frac{Comp}{4}$]	2
RULE BASE COMPLEXITY	[0 1]	3

RULE BASE INTERPRETABILITY	[0 1]	5
AVERAGE NUMBER LABEL BY INPUT	[2 9]	2
INTERPRETABILITY INDEX	[0 1]	5

Table 2. SYSTEM VARIABLES.

Being Comp the complexity of a classifier measured as the number of classes divided by the total number of premises and N the number of classes. [26]

4.3 PITTSBURGH ALGORITHM

To obtain the KB with the highest interpretability and to optimize the energy of a Datacenter, a genetic learning algorithm has been used. Pittsburgh is a genetic algorithm where each individual of population represents a complete knowledge entity (whole rule set). The chromosomes are knowledge bases and the evolution is developed through genetic operators that work with sets of fuzzy rules. Each time an individual is evaluated, the punctuation between the fuzzy rules of that KB is measured. In this way, the population can be evolve to obtain the kb with best punctuation. This model was proposed by Smith in 1980. [30]

One of the disadvantages of this algorithm is its difficulty in finding optimal solutions due to its wide search space. [31]

The representation of this algorithm is to regard individual rules as genes and population as strings of these genes. Thus, the cross between chromosomes will generate combinations of rules, while the mutation will generate new rules. [32]

We assume that all rules have a fixed-length and fixed-field format. This means that all the rules will the same number of antecedents and consequents. In this way, all the rules can be combines, since they have the same number of patterns. Thus, the rule will have as many elements as the number of variables in the system. Antecedents may have a non-zero value or if the value is equal to zero, that antecedent does not exist for that rule. Although, the rules have a fixed-length, the KBs can be different in terms of the number of rules.

Initially, the KBs had the same number of rules, in which the results have been very good. However, Smith worked in genetic algorithms to variable-length strings of rules. In addition, Smith complemented with a method for keeping down the size of the rule sets with a shorter string.

4.3.1 LEARNING PROCESS

This learning system works with a population of rule bases with a set of possible solutions, which each solution is a rule base. This initial population of solutions constitutes the first generation (parents). [15]

Each chromosome is a KB that is evaluated independently; for each rule base will be applied to the environment causing feedback that will reach the evaluation system that will analyse the behaviour of the rule bases. The score of KB is provided by the evaluation of the system; in this case, our score will be the power, the time and the interpretability index of each KB.

When all rule bases have been evaluated, a process of searching for new rules bases is carried out. This section generates new populations by applying a set of genetic or evolutionary operators.

(In the section 6, the algorithm's implementation will be explained in detail).

5. SOFTWARE MODIFICATIONS

In this section, the modifications that have been carried out in each of the software used will be explained, both the modifications in the WorkFlowSim-DVFS simulator and the Guaje tool (in the case of calculating the interpretability index).

Moreover, in section 8, how the genetic algorithm has been developed and how this software has been used, will be explained.

5.1 WFS-DVFS MODIFICATIONS

As previously mentioned, the simulator used WorkFlowSim-DVFS (WFS-DVFS) is the combination of two existing simulators. It extends CloudSim with DVFS and WorkFLowSim.

The main features of the simulator are:

- Be able to calculate the energy consumption, since if this estimation is not calculate, the proposal algorithm could not be assured that it can optimize power consumption and time in a real environment.
- On the other hand, the simulator is able to simulate real traces to look more like a real environment. Traces that could be executed in a real Datacenter.
- Being a free software, the simulator can be modified according to the user's preferences, being able to change the simulation scenario, add or modify schedulers, change scheduling algorithms or any modification that user want to make.

Each of these simulators has one of the necessary features for the simulator used. On the one hand, CloudSim with DVFS allows calculating and estimating the power consumption that a datacentre will have after executing traces composed of tasks. The problem is that the traces that it simulates are not real traces, that is, the power consumed of a real Datacenter could not be estimated, since its traces do not simulate a real environment.

On the other hand, WorkFlowSim is an extension of CloudSim, but in this case, it does allow simulating real workflow traces of Datacenter. These traces allow following an order in the tasks, that is to say, they allow the tasks dependent on others tasks not to be executed before the planned. This order simulates the processing of a real flow of tasks. To learn more about the combination of these two tools, see the article [6].

Being a free software, as indicated by one of the characteristics of the simulator, it is possible to make changes to the simulator and adapt it to the needs and objectives of this work.

First, modifications of this simulator have been carried out to obtain especially two schedulers. As mentioned previously, the simulator has two schedulers, one used to allocate the VMs in the physical machines, taking into account the performance and resources of each of them. On the other hand, a scheduler be able to choose what virtual machine will execute and process each task.

It is this last scheduler that has been modified in this project. In this case, two scheduler are needed, one for time optimization and other for power optimization. First, each scheduler will be analysed to optimize its objective independently. That is, the time scheduler will optimize only time, so its variables are related to time, and that of power scheduler, it only has variables related to power.

Finally, there will be a final scheduler that encompasses both, that is, this scheduler will have both the variables of the time scheduler and the power scheduler. It is called 'MOA scheduler'.

5.1.1 WORKFLOSIM-DVFS SCHEDULERS

The simulator has schedulers that implement FRBS systems. These are knowledge-based system that, based on input variables and fuzzy sets, provide an output variable, using rules that relate both variables.

FRBS systems are the local intermediaries within a Datacenter. As mentioned, it is composed of the VMs that will execute and process the workload for a specific application.

The workload is divided into jobs, and these jobs reach the scheduler, which is responsible for deciding on which machine the tasks will be processed and assigning the VM to the corresponding host.

This work has been developed with two schedulers, one implemented for the power consumed optimization and other for the execution time optimization.

Each of the schedulers have a different FRBS systems, each with its input variables and its output variables according to each objective. Both FRBS have the structure of Mandami fuzzy logic model. The structure of this model has a set of rules that combine the different inputs to provide an output. There rules have de following structure.

In the example of a car parking, the antecedents are the speed and the distance and the consequent is the acceleration. It is obviously to think that a rule example is 'IF **distance** is far AND **speed** is fast THEN **acceleration** is keep-speed'. This is the structure of Mamdami model.

For this project, there are two schedulers, one for each of the multi-objective parameters of this project, but that finally, it will be a single scheduler with the variables of both. A scheduler have a FRBS with variables related to time and another scheduler with variables related to power.

The description of the scheduler is given by FRBS system, consisting of the fuzzifier, the inference engine and the defuzzifier.

Initially, the fuzzifier is responsible for changing the values of each antecedent in fuzzy values. The antecedents are analysed and the objective is to determine how the inputs correspond to the fuzzy rules using membership functions. These MFs determine the membership of the fuzzy sets in the range [0,1]. In this way, a fuzzy set is associated with an antecedent or consequent.

The inference engine gets the fuzzy sets for the consequent taking into account the fuzzy rules and the antecedents. Finally, the defuzzifier returns a numerical value that describes the decision obtained from the process. First, the fuzzy sets of the consequent, which are converted to numerical values.

Next, the two schedulers will be explained in detail, indicating the input and output variables of the FRBS system, membership functions, the universe of discourse and all the information necessary to understand these schedulers. Finally, MOA scheduler will be explained.

5.1.1.1 TIME SCHEDULER

The objective of this scheduler is to find the best solution to optimize the execution time, that is, host the virtual machines on the hosts so that the execution time is the minimum.

For this, the FRBS needs antecedents and consequents to work. In this case, the input variables chosen for this FRBS are the following [33]:

- **MIPS.** Millions Instructions per Second being used in the virtual machine. Minimum performance that a VM has to have to execute the task. It is the ration between the length of task (measured in Millions of Instructions (MI) and the deadline of task (measured in seconds).
- **PEs.** Processing elements used in the virtual machine.
- **BW.** Bandwidth used by the virtual machine.
- **RAM.** Random Access Memory used in in the virtual machine.
- **TRANSFERT.** Transfer time for virtual machine.

In addition, there is an output variable **'OUTPUT'**. This consequent is the degree of selection each VM has to process the task. The high value of the output, the more appropriate will be the VM of being chosen.

Each antecedent considers three membership functions (MFs): low, average and high and the consequent considers five MFs: very low, low, average, high and very high.

The typical ways to define MFs are triangular, pyramidal and Gaussian functions among others. These experiments uses triangular functions for both the antecedents and the consequent. Figures 22-25 show the membership functions of antecedents and Figure 26 shows the membership functions of the consequent.

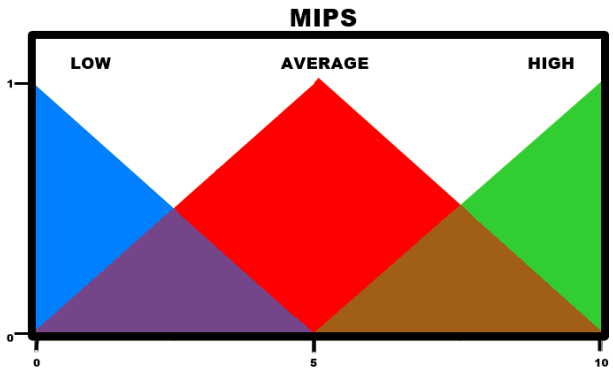

Figure 20. MIPS ANTECEDENT MFS.

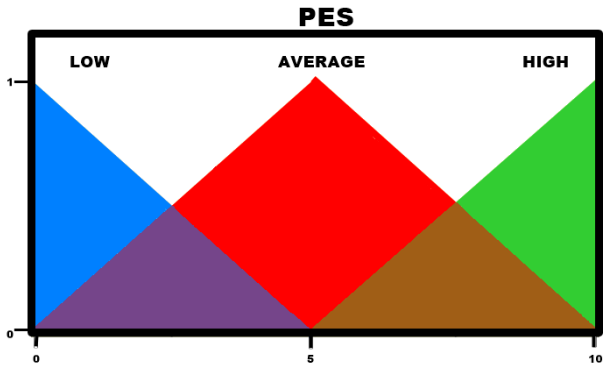

Figure 21. PES ANTECEDENT MFS.

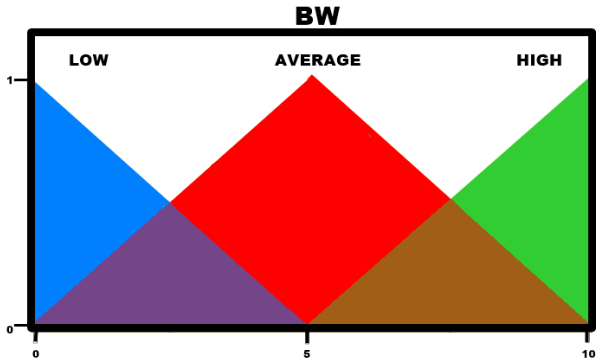

Figure 22. BW ANTECEDENT MFS.

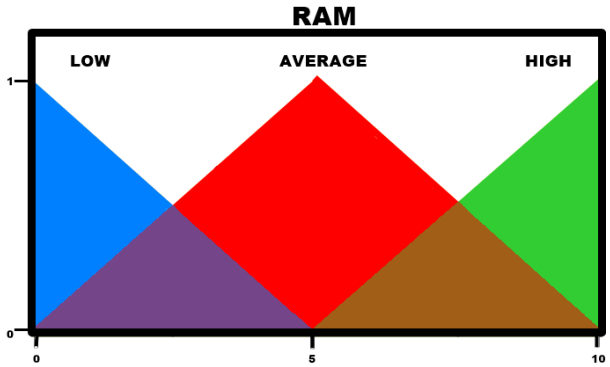

Figure 23. RAM ANTECEDENT MFS.

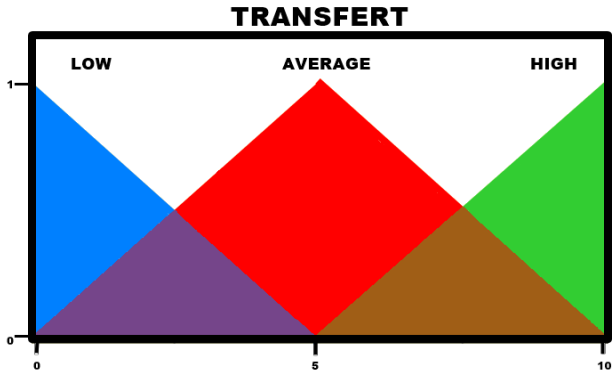

Figure 24. TRANSFERT ANTECEDENT MFS.

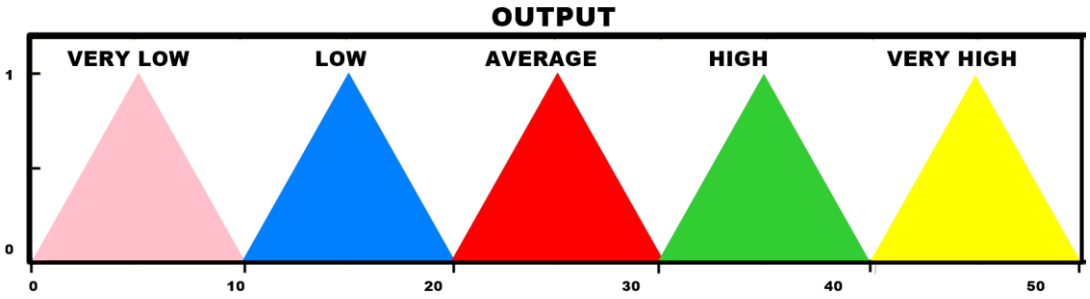

Figure 25. OUTPUT CONSEQUENT MFS.

Both the input variables and the output variable, each has a different universe of discourse. The following tables show the universe of discourse of the antecedents and consequent, and the membership function of each.

VARIABLE	UNIVERSE
MIPS [MIPS]	[0 10]
PEs	[0 10]
BW [Hz]	[0 10]
RAM [Bytes]	[0 10]
TRANSFERT [s]	[0 10]
OUTPUT	[0 50]

Table 3. UNIVERSE OF DISCOURSE VARIABLES TIME SCHEDULER.

	MFs		
VARIABLES	**LOW**	**AVERAGE**	**HIGH**
MIPS	[0 0 5]	[0 5 10]	[5 10 10]
PES	[0 0 5]	[0 5 10]	[5 10 10]
BW	[0 0 5]	[0 5 10]	[5 10 10]
RAM	[0 0 5]	[0 5 10]	[5 10 10]
TRANSFERT	[0 0 5]	[0 5 10]	[5 10 10]

Table 4. UNIVERSE OF DISCOURSE ANTECEDENTS MFs TIME SCHEDULER.

	VARIABLE
MFs	**OUTPUT**
VERY LOW	[0 5 10]
LOW	[10 15 20]
AVERAGE	[20 25 30]
HIGH	[30 35 40]
VERY HIGH	[40 45 50]

Table 5. UNIVERSE OF DISCOURSE CONSEQUENT MFs TIME SCHEDULER.

5.1.1.2 POWER SCHEDULER

As in the previous section, this new scheduler is to find the best solution to optimize the power consumed in a Datacenter. All hosts that form a Datacenter have to consume the minimum possible power.

For this, the FRBS needs antecedents and consequents to work. In this case, the input variables chosen for this FRBS are the following:

- **MIPS.** Millions Instructions per Second being used in the virtual machine. Minimum performance that a VM has to have to execute the task. It is the ration between the length of task (measured in Millions of Instructions (MI) and the deadline of task (measured in seconds). It is the performance of VM
- **Power.** Power is the power consumed for the host when a task is processed.
- **Length.** Task size measured in MI.
- **Utilization.** Utilization of CPU in percentage. Utilization created by all cloudlets running on the VM.
- **PowerMax.** Get the maximum power that can be consumed by the host.

In addition, there is an output variable **'selection'**. This consequent is the degree of selection each VM has to process the task. The high value of the output, the more appropriate will be the VM of being chosen.

Each antecedent considers three membership functions (MFs): low, average and high and the consequent considers five MFs: very low, low, average, high and very high.

The typical ways to define MFs are triangular, pyramidal and Gaussian functions among others. These experiments uses triangular functions for both the antecedents and the consequent. Figures 28-32 show the membership functions of antecedents and Figure 33 shows the membership functions of the consequent.

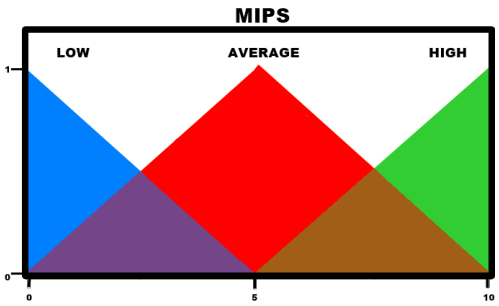

Figure 26. MIPS ANTECEDENT MFS.

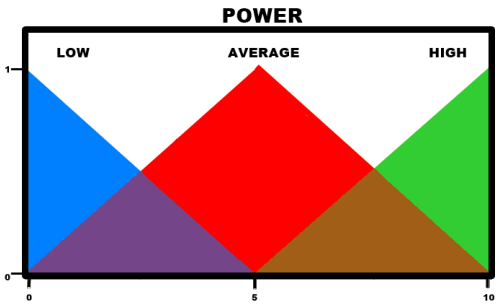

Figure 27. POWER ANTECEDENT MFS.

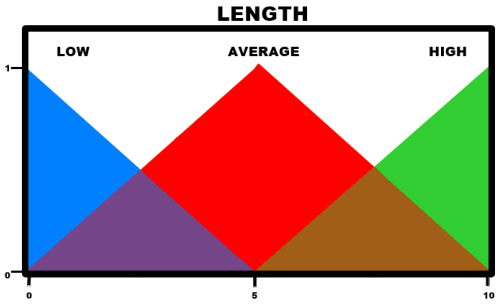

Figure 28. LENGTH ANTECEDENT MFS.

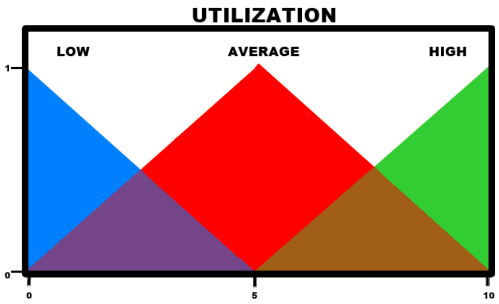

Figure 29. UTILIZATION ANTECEDENT MFS.

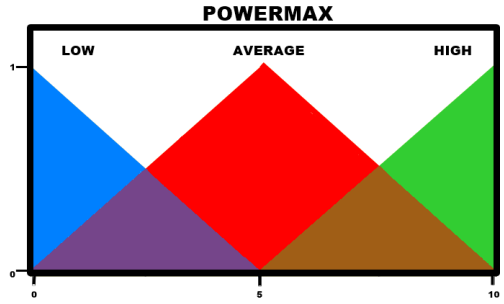

Figure 30. POWERMAX ANTECEDENT MFS.

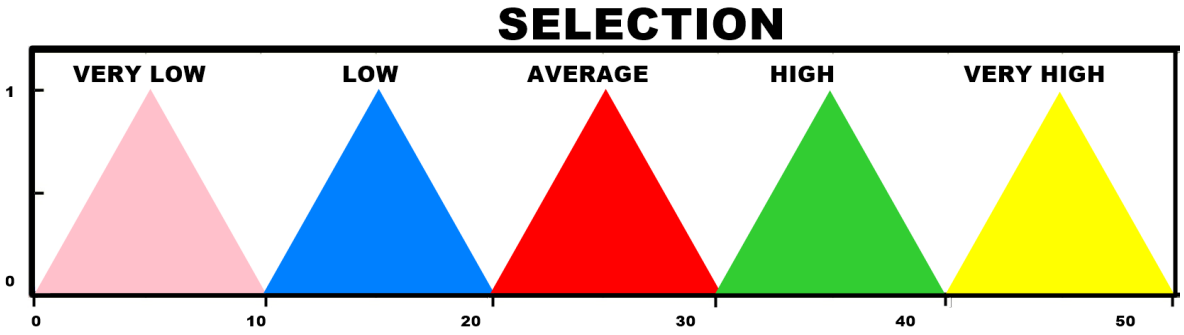

Figure 31. SELECTION CONSEQUENT MFS.

Both the input variables and the output variable, each has a different universe of discourse. The following tables show the universe of discourse of the antecedents and consequent, and the membership function of each.

Table 6 shows the discourse universe of each antecedent and consequent. Table 8 shows the values of antecedent MFs and Table 7 shows the values of consequent MFs.

VARIABLE	UNIVERSE
MIPS [MIPS]	[0 10]
POWER [W]	[0 10]
LENGTH [MI]	[0 10]
UTILIZATION [%]	[0 10]
POWERMAX [W]	[0 10]
SELECTION	[0 50]

Table 6. DISCOURSE UNIVERSE OF VARIABLES POWER SCHEDULER.

MEMBERSHIP FUNCTION	VALUES
LOW	[0 0 5]
AVERAGE	[0 5 10]
HIGH	[5 10 10]

Table 7.VALUES ANTECENDET MFS POWER SCHEDULER

MEMBERSHIP FUNCTION	VALUES
VERY LOW	[0 5 10]
LOW	[10 15 20]
AVERAGE	[20 25 30]
HIGH	[30 35 40]
VERY HIGH	[40 45 50]

Table 8. VALUES CONSEQUENT MFS POWER SCHEDULER.

The power model used determines the choice of these values. These values are normalized and multiplied by a factor with value 10.

With these two schedulers, the simulator would be prepared for multi-objective optimization. One these changes have been made in the simulator, two data files will be obtained. These files will be called 'DATOS.txt' and will contain the necessary data so that GUAJE tool, together with another file (KB), can calculate the system's interpretability index.

With this in mind and with the changes made, there are two different simulators, each with a scheduler and a data file related to the scheduler.

5.1.1.3 MOA SCHEDULER

As previously mentioned, this project has been included as a secondary objective the multi-objective optimization, taking into account the power and the time spent in a Datacenter.

In the previous sections, the two schedulers have been explained for each of the objectives independently. With this in mind, it is logical to think that the multi-objective scheduler must have variables that optimize power and time.

For this, the FRBS needs antecedents and consequents to work. In this case, the input variables are the input variables of the two previous schedulers, taking into account that both add up to 10 input variables but have one in common that are the MIPS. Thus, in this scheduler, it would have 9 input variables.

- **MIPS.** Millions Instructions per Second being used in the virtual machine. Minimum performance that a VM has to have to execute the task. It is the ration between the length of task (measured in Millions of Instructions (MI) and the deadline of task (measured in seconds). It is the performance of VM
- **Power.** Power is the power consumed for the host when a task is processed.
- **Length.** Task size measured in MI.
- **Utilization.** Utilization of CPU in percentage. Utilization created by all cloudlets running on the VM.
- **PowerMax.** Get the maximum power that can be consumed by the host.
- **PEs.** Processing elements used in the virtual machine.
- **BW.** Bandwidth used by the virtual machine.
- **RAM.** Random Access Memory used in the virtual machine.
- **TRANSFERT.** Transfer time for virtual machine.

In addition, there is an output variable **'selection'**. This consequent is the degree of selection each VM has to process the task. The high value of the output, the more appropriate will be the VM of being chosen.

As in the previous cases, each antecedent considers three membership functions (MFs): low, average and high and the consequent considers five MFs: very low, low, average, high and very high. The figure 34 shows the MFs of each input and output variable.

Both the input variables and the output variable, each has a different universe of discourse. The following tables show the universe of discourse of the antecedents and consequent, and the membership function of each.

Table 9 shows the discourse universe of each antecedent and consequent of this new scheduler. Table 11 shows the values of antecedent MFs and Table 10 shows the values of consequent MFs.

	MFs		
VARIABLES	**LOW**	**AVERAGE**	**HIGH**
MIPS	[0 0 5]	[0 5 10]	[5 10 10]
POWER	[0 0 5]	[0 5 10]	[5 10 10]
LENGTH	[0 0 5]	[0 5 10]	[5 10 10]
UTILIZATION	[0 0 5]	[0 5 10]	[5 10 10]
POWERMAX	[0 0 5]	[0 5 10]	[5 10 10]
PES	[0 0 5]	[0 5 10]	[5 10 10]
BW	[0 0 5]	[0 5 10]	[5 10 10]
RAM	[0 0 5]	[0 5 10]	[5 10 10]
TRANSFERT	[0 0 5]	[0 5 10]	[5 10 10]

Table 9. VALUES ANTECEDENT MFS MOA SCHEDULER.

VARIABLE	**UNIVERSE**
IMS [MIPS]	[0 10]
POWER [W]	[0 10]
LENGTH [MI]	[0 10]
UTILIZATION [%]	[0 10]
POWERMAX [W]	[0 10]
PEs	[0 10]
BW [Hz]	[0 10]
RAM [Bytes]	[0 10]
TRANSFERT [s]	[0 10]
OUTPUT	[0 50]

Table 11. DISCOURSE UNIVERSE OF VARIABLES MOA SCHEDULER.

	VARIABLE
MFs	**SELECTION**
VERY LOW	[0 5 10]
LOW	[10 15 20]
AVERAGE	[20 25 30]
HIGH	[30 35 40]
VERY HIGH	[40 45 50]

Table 10. VALUES CONSEQUENT MFS MOA SCHEDULER.

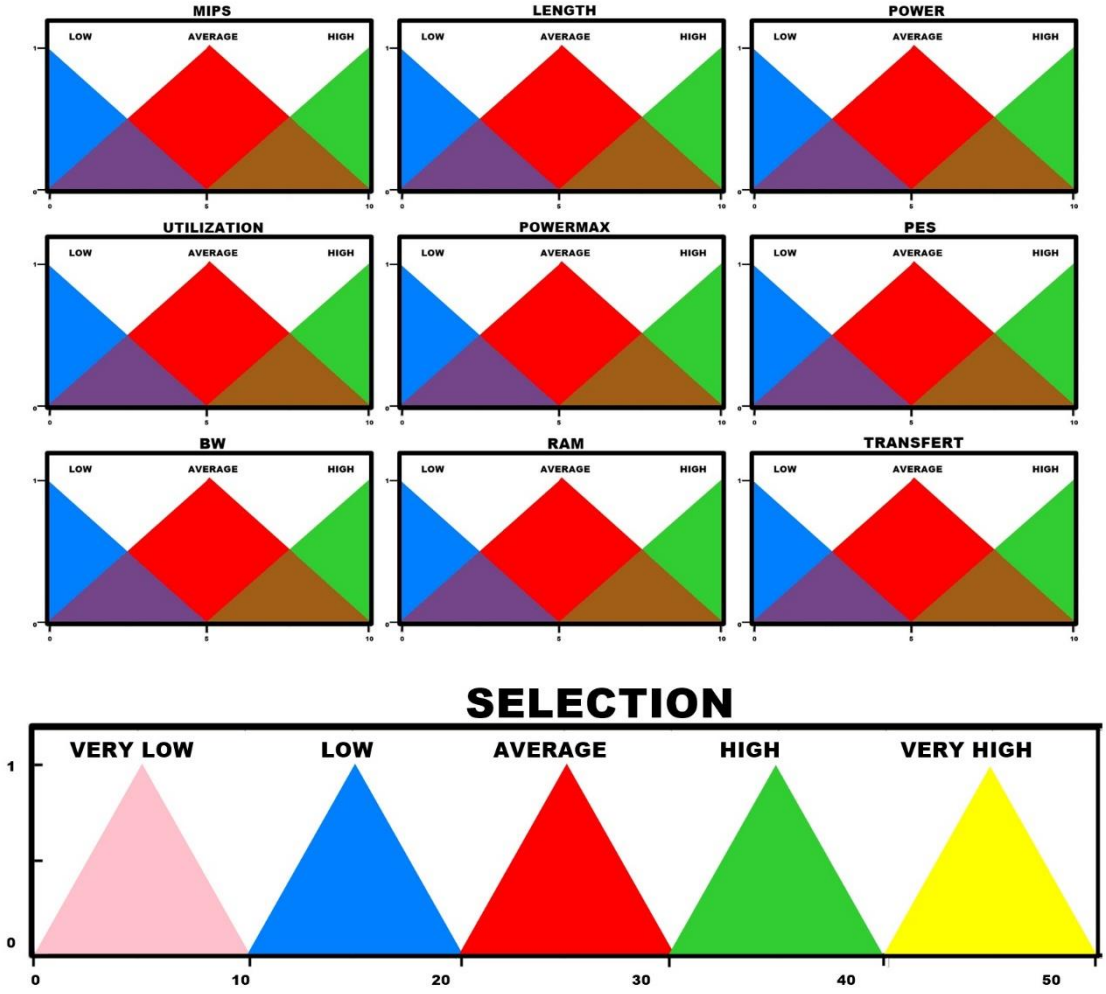

Figure 32. MEMBERSHIP FUNCTIONS ANTECEDENTS AND CONSEQUENT.

5.1.2 REQUIREMENTS

Each simulator will be exported in an executable file '.jar'. To run each executable file, a java virtual machine will be necessary.

In addition, as discussed in previous sections, there will have to be a folder in the directory called 'config'.

This folder will also have three subfolders, one where the input DAX files will be stored. Containing all traces with workflows for a simulation as a real as possible (In this case Montage_1000.xml). This folder will have the name of 'dax'.

Another folder ('fcl' where the .fcl files will be stored containing the database where the input and output variables are defined, and their fuzzy sets with the membership functions.

Finally, a third folder called 'rules'. There will be the rules file encoded in JSON.

An example of the folder 'Config' that should exist can be seen in figure 36.

There are two main files needed. These file are 'dvfs.fis' and fis2fcl.m.

- 'Dvfs.fis'. This file contains the name of input and output variables, as well as their fuzzy sets. A part of this file can be seen in figure 35.

```
[Input1]
Name='MIPS'
Range=[0 10]
NumMFs=3
MF1='low':'trimf',[0 0 5]
MF2='average':'trimf',[0 5 10]
MF3='high':'trimf',[5 10 10]
```

Figure 33. MIPS VARIABLE EXAMPLE OF dvfs.fis FILE.

- In addition to this file, a converter file is needed to translate the '.fis' file to a .fcl file. This class is 'fis2fcl.m'. It is a file, written in Matlab programming language, responsible for converting the fis file into an fcl file, as explained before; the simulator needs it, since it is the system database.

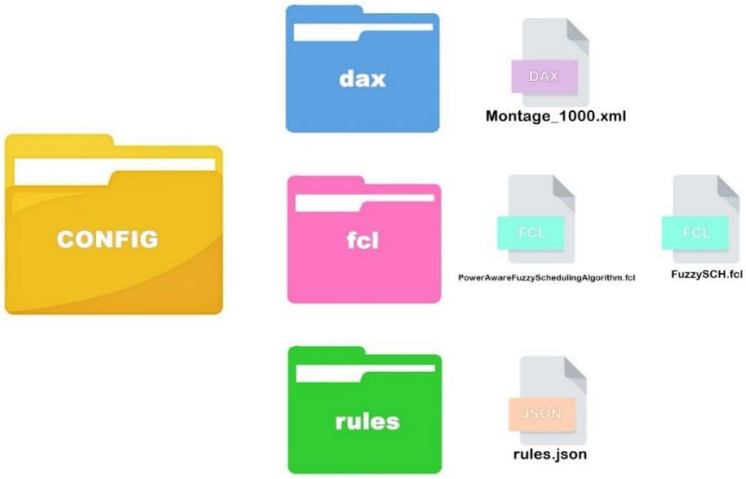

Figure 34. CONFIG FOLDER.

5.2 GUAJE TOOL MODIFICATIONS

Guaje is a java environment for generating understandable and accurate models. It is a free software tool and therefore allows that the source code can be modified and adapted to the user's needs. [16] [17] [18]

Guaje tool has a graphical user interface but in this work, we need an executable file (.jar) so that the simulations can be performed automatically, without the need of a user managing the graphical interface to perform the necessary actions to obtain the index of interpretability.

For this reason, changes have been made to the source code to select the necessary files without having to enter them manually.

For the functioning of GUAJE tool, two independent software of this tool are necessary: FISPRO-3.5 [34] and Graphviz 2.38 [35].

As for these software, if the user interface were used, GUAJE would ask that the path of the 'bin' folder of each of these programs be selected. This problem has been solved by

directly adding the path of the bin folder of each of them. By default, each computer installs Graphviz in 'ProgramFiles(86)' and FISPRO in 'ProgramFiles' folders.

This modification has been made in the main class of the program called 'MainKBCT.java'.

On the other hand, to obtain the interpretability index of the system, Guaje tool needs expert knowledge, provided by a knowledge base and experimental data, provided by a data file ('Datos.txt' file provided by the WFS-DVFS simulator). The modification is to make Guaje directly obtain these files (stored in the simulation directory folder).

These files are 'DATOS.txt' for the experimental data and 'DATOS.txt.kb.xml' for the knowledge base.

The methods responsible for performing these actions are jMenuDataOpen_actionPerformed() and jMenuExpertOpen_actionPerformed(), respectively. Both belonging to the JMainFrame.java class. In these methods, it is necessary to add the name of both files.

Finally, the last modification is in the method responsible of assessment the interpretability from the data and the knowledge base.

This method is AssessingInterpretability () of the JExpertFrame.java class. It is responsible for calculating the interpretability index of a KB.

Guaje's way of calculating this parameter is through a hierarchical FRBS system.

5.2.1 REQUIREMENTS

In order for Guaje to work correctly on any computer, it is necessary to have some programs installed on the computer, as previously mentioned.

- FISPRO 3.5 [34]
- Graphviz 2.38 [35]
- Java Virtual Machine (Recommended version jre1.8.0_181)

For these three programs, it is necessary to include the 'bin' folder in the environment variables of the computer.

In addition, as described in previous sections, for Guaje to calculate the interpretability index, it requires two types of files.

On is the knowledge base, which contains the input and output variables, the fuzzy sets and the membership functions of each variable and the rules that link the inputs to offer an output. Moreover, the other file is the data file, provided by executing the WFS-DVFS simulator. There files are 'DATOS.txt.kb.xml' and 'DATOS.txt', respectively.

There are two main files to create the DATOS.txt.kb.xml file. These file are 'dvfs.fis' and fis2kb.m.

- 'Dvfs.fis'. This file contains the name of input and output variables, as well as their fuzzy sets. As in WFS-DVFS simulator, this file serves as the base for creating the KB.
- In addition to this file, a converter file is needed to translate the '.fis' file to a file that Guaje understands. This class is 'fis2kb.m'. It is a file, written in Matlab

programming language, responsible for converting the fis file into an xml file, with which Guaje works.

6. MATLAB INTEGRATION

In this section, the integration with Matlab will take place. As previously mentioned, in fuzzy logic system, the rules can be created manually, but not all the scope that could cover a wide combination of the antecedents to produce outputs is covered. This is why the Pittsburgh Approach algorithm will be used.

This algorithm has been developed in Matlab language and it is composed of several classes; the main class called 'pit_moa.m' and other necessary classes that will be explained below.

Figure 36 shows the scheme of the implementation of this work.

The development of the implementation will be explained using the hierarchical system implemented in this work. To use Guaje tool, it would be done in the same way, only the requirements in section 5.2.1 should be taken into account.

In addition, it is necessary to use the WFS simulator. WFS simulator evaluates each KB, indicating the power consumption and the execution time it takes to process the input tasks for each KB and the hierarchical fuzzy system evaluates each KB indicating the interpretability index of each. WFS is developed in Java programming language and the hierarchical fuzzy system is a Matlab class.

Pittsburgh, as previous mentioned, is a genetic algorithm, based on population f individuals. Each individual of population represents a complete KB, whose chromosomes are knowledge bases and evolution is developed through genetic operators that work with fuzzy rule sets. This algorithm is based on FRBS.
It is necessary to design several parameters for its development, taking into account the scenario in which it will be developed (section 7). The defined parameters are the number of individuals of population, probability of selection, probability of mutation, number of iterations of the algorithm and maximum number of fuzzy rules that an individual can have.

It must be taken into account that the design of these parameters directly affects the results of the algorithm. For example, the high the number of rules, the smaller the system's interpretability. The main class is responsible for defining these parameters and to execute the necessary commands so that the WFS simulator and the fuzzy system can evaluate each individual (KB) of the population. In addition, here the process of crossing and mutation of the KB rules is carried out.

Each knowledge base has a variable number of rules, that is, the population will compose of different KBs and each will have a different number of rules.

The system has five input and one output variables, both to optimize time and power. Each one defined with its fuzzy sets and its membership functions.

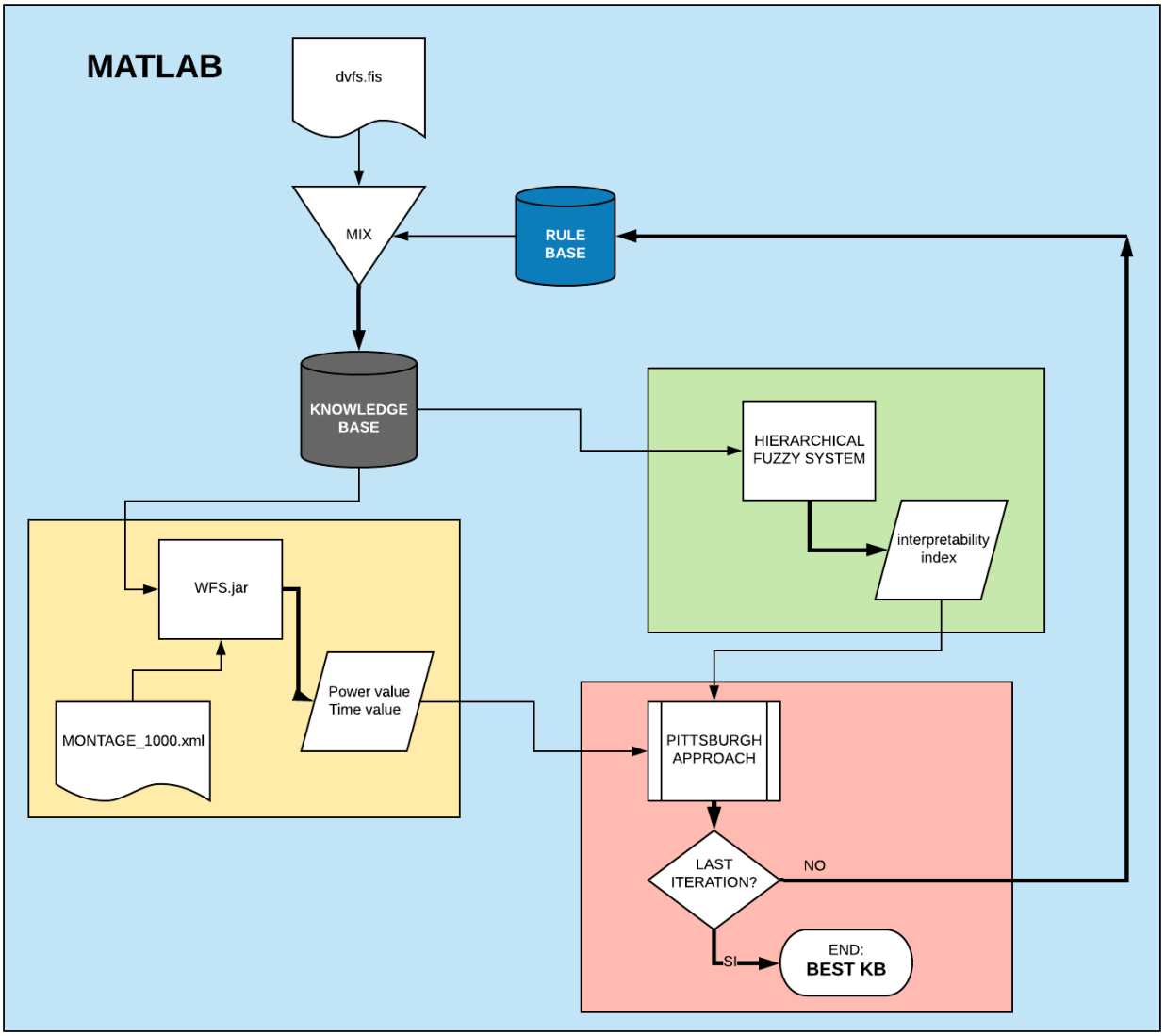

Figure 35. SCHEME IMPLEMENTATION (Using GUAJE).

The membership functions (MFs) of each variable are defined in the fis file. This file is called 'dvfs.fis' and contains the necessary information of the variables. Each input variable has three MFs (Low, average and high) and output variable has five MFs (very low, low, average, high and very high). In the following figures you can see how the MF of each variable.

The KB is composed by fuzzy rules, which will be generated randomly, taking into account the membership functions of each variable, as well as the operator (And or OR) and the weight.

In conclusion, there will be rules composed by twelve elements: nine input variables, one output variable, the weight, which will always be equal to 1 and the operator value that will be equal to 1 if it is AND clauses or equal to 2 if it is OR clauses.

It is necessary to encode these rules in two different ways, one to be recognized by the WFS-DVFS simulator and another for Guaje tool (if this were used).

For the WFS-DVFS simulator, the rule base will be encoded in JavaScript Object Notation (JSON). JSON is a simple text format for data exchange. The encoded rule base is passed by through an argument to the simulator jar file. The main class in the simulator must receive the arguments and use the fuzzy classes to decode and store the rules. For this encoding, 'fis2fcl.m' and 'rules2json.m' functions are required.

An example of this encoding (for time FRBS) would be as follows:

{"rules":[{"MIPS":"1", "PEs":"2", "BW":"1", "RAM":"2", "TRANSFERT":"-1", "OUTPUT":"4", "connection":"2"},{"MIPS":"3", " PEs ":"1", " BW ":"-2", " RAM ":"3", " TRANSFERT":"-":"1", " OUTPUT ":"3", "connection":"2"},{"MIPS":"1", " PEs ":"2", " BW ":"3", " RAM ":"2", " TRANSFERT":"-2", " OUTPUT ":"-1", "connection":"2"}]}

For the Guaje program, the rule base is necessary to create an xml file that will use Guaje to evaluate each KB indicating the interpretability index of each one. This xml file is called 'DATOS.txt.xml' and it is the KB that Guaje needs. Input and output variables, the fuzzy sets and the rule base compose this file.

An example of the coding of the rules of Guaje:

```
<Rules>
    <Rule1 Active="true" I1="1" I2="2" I3="1" I4="2" I5="-1" Nature="I" O1="4"/>
    <Rule2 Active="true" I1="3" I2="1" I3="-2" I4="3" I5="1" Nature="I" O1="3"/>
    <Rule3 Active="true" I1="1" I2="2" I3="3" I4="2" I5="-2" Nature="I" O1="-1"/>
  </Rules>
```

6.1 ALGORITHM STAGES

Pittsburgh algorithm performs an evolutionary learning of a knowledge base on a fuzzy controller. In this case, a set of predefined membership functions is predefined in the database. The evolutionary algorithm evolves the rule base, working with individuals that describe a whole rule base.

The fuzzy-genetic systems based on the Pittsburgh approach use complete knowledge bases or rule bases as individuals in the population. In this approach, the evolutionary system obtains new knowledge bases, which are evaluated by acting directly on the environment and analysing its evolution.

The following figure shows how the integration of the fuzzy control system with the system to be controlled and the evolutionary system is carried out.

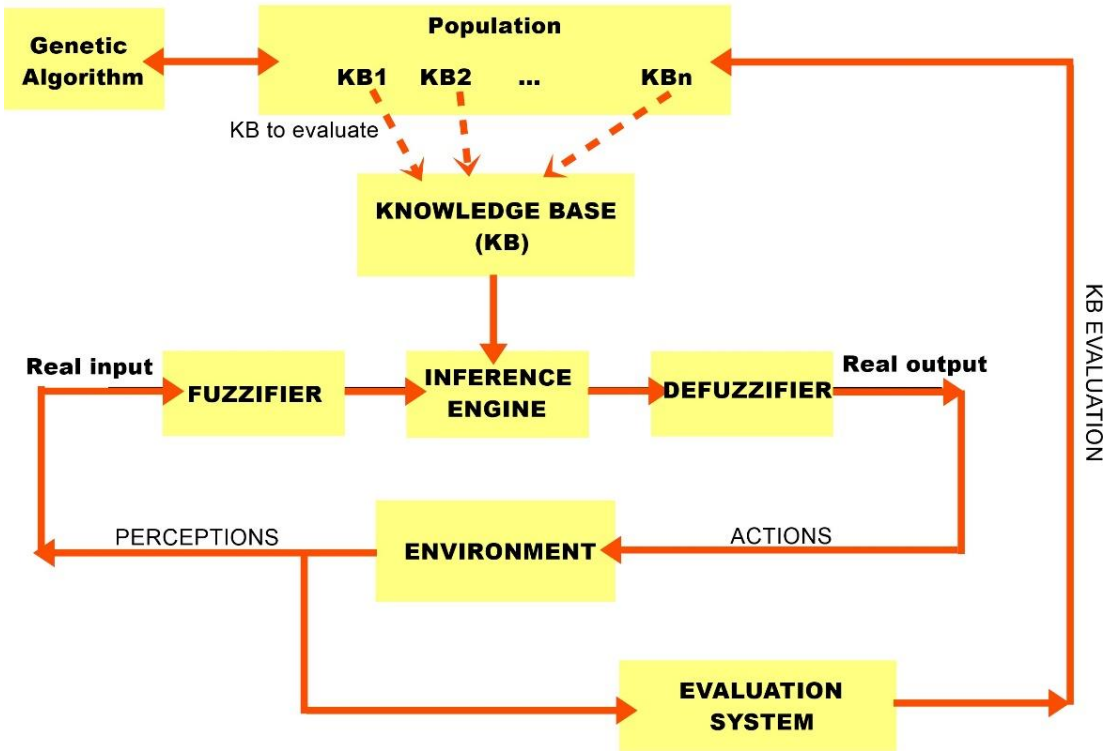

Figure 36. PITTSBURGH EVOLUTIONARY LEARNING.

The evolutionary schematic is compose of the following elements:

- Fuzzy system is composed of input and output values, the inference engine, the fuzzifier and defuzzifier and knowledge base. As explained in previous sections, the fuzzifier transforms the crisp value into fuzzy value belonging to a fuzzy set. The inference engine contains the KB with fuzzy sets, input and output variables and fuzzy rules, and decides which membership function the value belongs. Finally, the defuzzifier transforms the output fuzzy value into a crisp value.
- The fuzzy system acts on the environment through perceptions and actions (context and operation variables).
- The population of knowledge bases represents a set of control system solutions.
- Assessment system performs a degree of adaptation of KBs to the environment.
- Genetic system allows obtaining new KBs or rules bases from the initial population. The systems that are based on this approach centre the learning process on the rules bases, although it is also possible to incorporate a part of the database into the process (MFs). If the learning process acts on the two components of a KB, individuals must contain information on the rules bases and on membership functions. In turn, the learning process can be carried out in two phases: initially the base would be optimized of rules and, subsequently, the database. The system can act on the linguistic rules of knowledge, performing the antecedents crossing between rules or the mutation of a variable in the antecedent. [15]

The main stages of this algorithm are detailed below [36] and applying the hierarchical fuzzy system:

1) INIZIALIZE THE POPULATION. The population is initialized randomly. This initial population will have a number of knowledge bases and in turn, these will have a number of rules each one. It is the parents population. This population will have individuals and each individual will have a knowledge base compose of different

rules. These KBs come by minimum and maximum number of rules and a random component. This means that individuals have different number of rules in each KB. In addition, these KBs will be created taking into account the file 'dvfs.fis', where the input and output variables, the fuzzy sets and the membership functions will appear. Each rule base have eight parameters: five input variables, one output variable, the weight that will always be equal to 1 and the operator, 1 if it is AND clauses and 2 if it is OR clauses.

2) EVALUATE POPULATION. Each KB is evaluated by the WFS simulator jar file and the fuzzy system. The fitness of each one is obtained. The fitness are the power consumption, the time and the interpretability index. The first two obtained from WFS simulator and the last one from the hierarchical fuzzy system. So that the simulator can get the value of fitness, two files are necessary. On the one hand, the rules file encoded in JSON, called 'rules.json' and on the other hand, the .fcl database.
   Once the simulator is run, then create the 'DATOS.txt' file needed for GUAJE tool.
   Thus, one all the individuals of the population have been evaluated, there will be three vectors with the values of the fitness of each KB.
3) SORT BY FITNESS. In this step, each individual of the population will be sorted by their fitness. This sorting will be carried out through the concept of Pareto's Law, taking into account the power fitness and time fitness. Once the solutions that are in the Pareto front are obtained, these solutions are sorted according to the interpretability index of each KB, in descending order. In this way, the KBs will be sorted, so that the best ones belong to the Pareto front and have greater interpretability of their rule bases.
4) SELECT PARENTS FOR CROSSOVER. The parents are selected by the probability of selection. One, the population is sorted according to the interpretability index and taking into account the parental front solutions, then the parents are selected to be crossed to obtain the population of children. The idea is to keep the best parents and replace the worst by the population of the children.
5) CROSSING PARENTS→ GETS CHILDREN. The crossing is made between pairs of parents, that is, two KBs parents are combined to have two descendants with the combination of these parents. The 'n-point crossover' technique has been used for the crossing. This technique consists of choosing two crossing points randomly. The rule bases of individuals are fragmented by those crossing points. Children KBs are composed by alternating these fragments of the parents. The crossing operator explores the search space. Figure 39 shows this technique.
6) CHILDREN MUTATION. In addition to the crossing technique to obtain the children population, the mutation technique has been added. A child may have genetic material (some rule) that none of their parents have, that is, it is not inherited.
   There are several types of mutation:
   - Catastrophic: When the child is worse than the father, then the child is not viable.
   - Neutral: Altering the rule does not change its rule.
   - Advantageous: The child is better than the father.

   Mutation operators act on the genotype, that is, it acts on an antecedent of any of the rules in the rule base. Its randomness is essential; the value of a rule is altered with a probability of mutation (Pm). In this way, some gene of the individual of the new population is randomly altered.

   The mutation operator exploits the space that has already been discovered (Area close to parents).

7) JOIN CHILDREN WITH PARENTS. Many descendants may be worse than parents may, so the best parents are maintained through the probability of selection and the best children. In summary, the best parents according to the probability of selection and the remaining children compose the new population.
8) CHILDREN EVALUATION. The new population is evaluated and individuals are sorted again.

At the end of the cycle (From step 2 to 8), the final population is sorted, so at the end of the execution, the first position will contain the rule bases that achieves the lowest energy-time consumption and have greater interpretability.

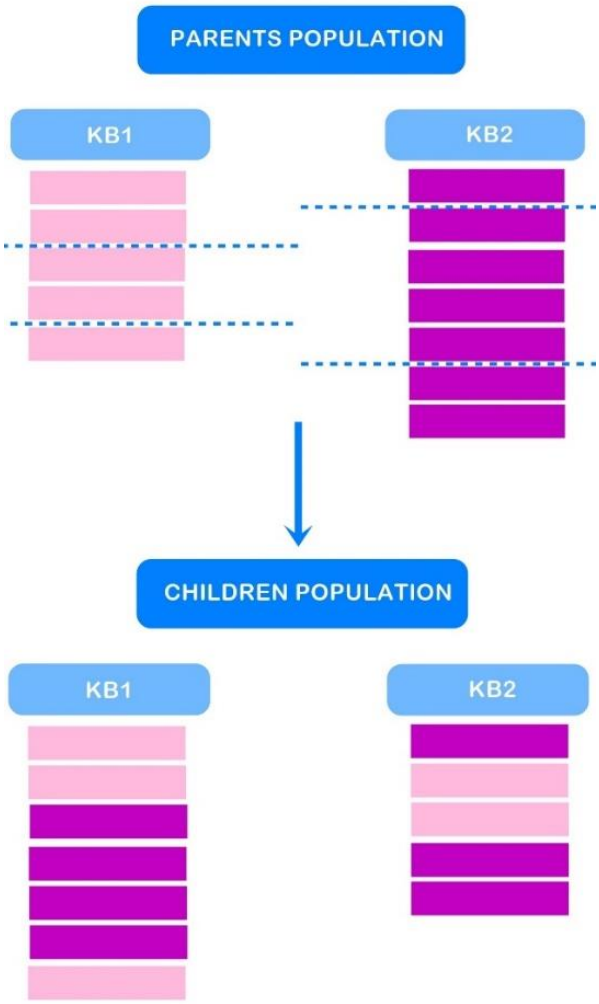

Figure 37. CROSSING.

In the following diagram, the steps of the proposed algorithm can be observed step by step.

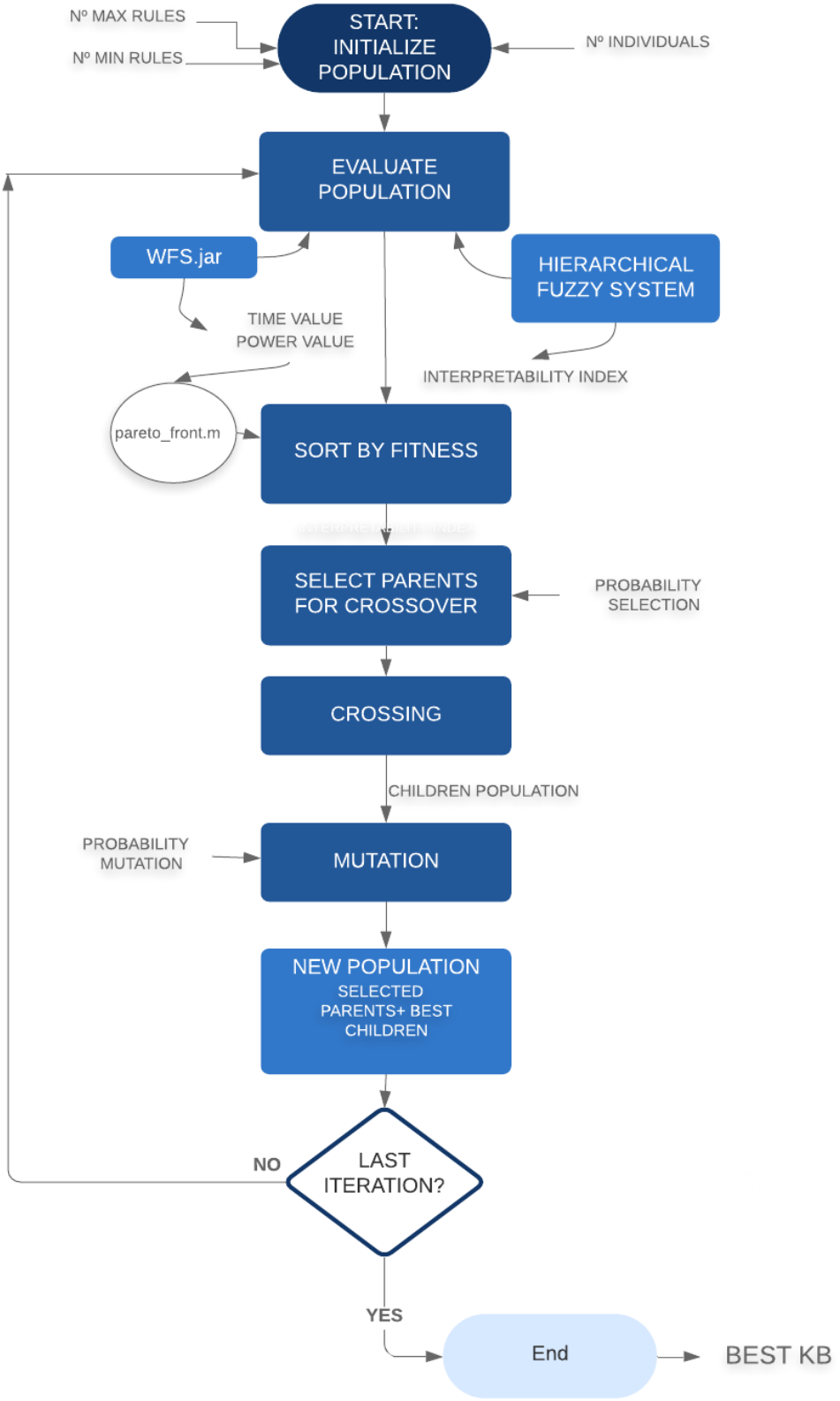

Figure 38. MAIN STAGES OF PITTSBURGH ALGORITHM.

7. SIMULATION ENVIRONMENT

This section will describe the scenario proposed for simulations. There are three different simulations, one optimizing power, another optimizing time and finally, a simulation optimizing the two objectives, time and power. The latter simulation also takes into account the interpretability of the fuzzy system.

All simulations will have the same scenario in terms of Datacenter resources. In addition, the interpretability of each knowledge base obtained in the simulations will be analysed.

7.1 SCENARIO

The work is to make use of a simulator that is able to optimize the power consumed and the time spent processing some tasks. Also, analyse the interpretability of the knowledge base obtained through a learning algorithm. Taking into account that in the multi-objective experiment, all the rules bases will be evaluated obtaining the interpretability index and the rule base that has a greater interpretability will be chosen depending on the power and time fitness.

In these simulations, WFS-DVFS is used. This simulator has the ability to adjust the CPU frequency to minimize total energy consumption. To do this, it uses several governors; in this case, the Powersave governor has been used. This governor selects the lowest frequency of the CPU and sets it to that value.

Another advantage of using this simulator is that it simulates real workflows and modern CPU already include the DVFS technique, the simulation results will be very similar to real ones in real Datacenters.

The real workflows used for these simulations have been Montage DAG with 1000 traces, created by Pegasus Workflow Management System.

Montage is an application that builds custom mosaics of astronomical images based on existing images. The inputs images are first reprojected to the coordinate space of the output mosaic, the reprojected images are then background rectified and finally coadded to create the final output mosaic. [38]

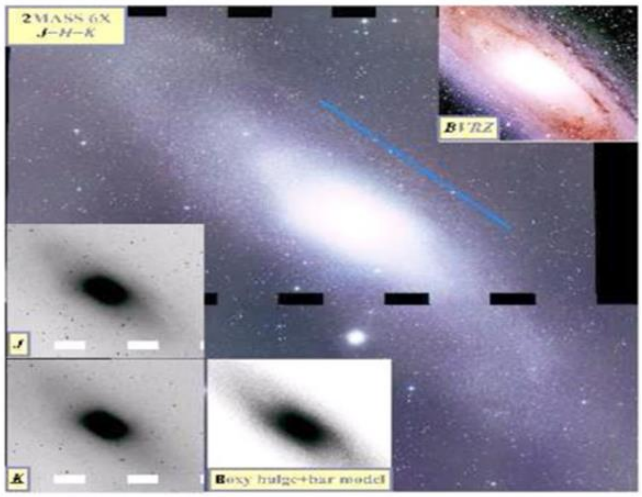

Figure 39. EXAMPLE IMAGE MONTAGE.

(Font: Beaton et al. Ap. J. Lett in press [39])

Montage offers scientific grade sky mosaics. Pegasus transform the Montage code of a single processor into a complex workflow and parallel computations to process images on a larger scale. These workflows have tens of thousands of executable tasks and process thousands of images.

Figure 41 represents a bar in the Spiral galaxy M31. [39]

The simulation scenario is a heterogeneous scenario. It has a Datacenter composed of different 20 physical machines (hosts). These hosts have different resources and performance, offering a real scenario.

In addition, as discussed in previous sections, there will be 20 virtual machines, of which each will be allocated on a host. That is, the scenario has 20 hosts, each with a virtual machine. Each host will store a virtual machine according to performance based on the simulator FRBS system (explained in section 4.1.2.1 'first stage: VM scheduling).

For each host to be different in terms of performance and utilization, the power model explained above is carried out.

This model power (PowerModelAnalytical) has several parameters to configure. In addition, three parameters are fixed for all hosts of the Datacenter. Table 12 shows these parameters.

- DYNAMIC CONSTANT is required to calculate the dynamic component of the power consumed by each host.
- STATIC COMPONENT is required to calculate the static component and total power.
- PERCENTAGES ARRAY composed of the different multipliers in percentages of the maximum values set. When the multiplier of the CPU is downs set from the maximum, the different values of frequency and voltage are obtained by applying these multipliers to the maximum value of frequency and voltage.

DYNAMIC CONSTANT	STATIC CONSTANT	PERCENTAGES ARRAY
0.5	0.7	[59.925, 69.93, 79.89, 89.89, 100.0]

Table 12. FIXED PARAMETERS POWER MODEL.

In table 13 (in green), the different input parameters are displayed.

- ID is the identifier of the host.
- MAXF is the maximum frequency of the processor of the host.
- MAXV is the maximum voltage of the processor of the host.
- C is the capacitance of the processor. Range from 5 GF to 30 GF.
- IPC is the instructions per cycle, it is used to get the value of MIPS performance. Range from 0.5 to 2.

ID	MAXF [GHz]	MAXV [V]	C [GF]	IPC	PERF (MIPS)	DYPOWER [W]	STPOWER [W]	TPOWER [W]
1	3.0	3.00	5.00	0.500	1500.0	67.50	157.50	225.00
2	2.9	2.95	6.25	0.575	1667.5	78.87	184.02	262.89
3	2.8	2.90	7.50	0.650	1820.0	88.31	206.05	294.35

4	2.7	2.85	8.75	0.725	1957.5	95.95	223.88	319.82
5	2.6	2.80	10.00	0.800	2080.0	101.92	237.81	339.73
6	2.5	2.75	11.25	0.875	2187.5	106.35	248.14	354.49
7	2.4	2.70	12.50	0.950	2280.0	109.35	255.15	364.50
8	2.3	2.65	13.75	1.025	2357.5	111.04	259.10	370.14
9	2.2	2.600	15.00	1.100	2420.0	111.54	260.26	371.80
10	2.1	2.550	16.25	1.175	2467.5	110.95	258.88	369.83
11	2.0	2.500	17.50	1.250	2500.0	109.38	255.21	364.58
12	1.9	2.450	18.75	1.325	2517.5	106.92	249.48	356.40
13	1.8	2.400	20.00	1.400	2520.0	103.68	241.92	345.60
14	1.7	2.350	21.25	1.475	2507.5	99.75	232.75	332.50
15	1.6	2.300	22.50	1.550	2480.0	95.22	222.18	317.40
16	1.5	2.250	23.75	1.625	2437.5	90.18	210.41	300.59
17	1.4	2.200	25.00	1.700	2380.0	84.70	197.63	282.33
18	1.3	2.150	26.25	1.775	2307.5	78.87	184.03	262.90
19	1.2	2.100	27.50	1.850	2220.0	72.77	169.79	242.55
20	1.1	2.050	28.75	1.925	2117.5	66.45	155.05	221.51

Table 13. PARAMETERS PHYSICAL MACHINES.

From these input parameters for each host, the power model used obtains the values for the output parameters, using the equations of section 3.2 and 3.3.3. The output values are the following:

- Frequency and voltage array, which contain the several frequencies and voltages corresponding at each multiplier.
- Dynamic power array, which contains the dynamic component of the power consumed by the host.
- Static power array, static component of the power that does not depend on the processor.
- Idle power and full power array, which contain the different values of the static power and de max power, respectively, depending on the multipliers.
- Maximum Power is the maximum power consumed by the host and the maximum performance that the host can achieve measured in MIPS.
- MIPS array, which contains the different values of the performance applying the multipliers to values of maximum performance.

Table 13 (in blue) shows the output values of maximum performance, the dynamic and static power and finally, the total power consumed.

With the las two tables, there is a wide range of different values for each host.

- PERF is the maximum performance of the processor measured in MIPS.
- DYPOWER is the dynamic component of the power.
- STPOWER is the static component of the power.
- TPOWER is the total power consumed by the host.

One the scenario is heterogeneous in terms of physical machines, the each of the 20 virtual machines is allocated on a host. That is, one VM per host.

Although VMs may have the required MIPS, a selection of 97.5% of the host's minimum MIPS value will be established for the DVFS technique.

If table 12 and table 13 are taken into account, for example for the first host (HOST ID 1), the following table shows the results to allocate the VM.

HOST ID	MAXMIPS HOST	1ºMULTIPLIER	MINIMIPS HOST	SELECTION PERCENTAGE	VM MIPS
1	1500	59.925%	898.87	97.5%	525.18

Table 14. EXAMPLE ESTIMATION MIPS VM.

The minimum MIPS value for the host 1 is 898.87 MIPS, obtained from multiplying the MIPS value of the host by the first multiplier.

If this minimum value of MIPS is multiplied by the selection percentage (97.5%) and the first multiplier, the MIPS of the VM is obtained, in this case 525.18 MIPS. In this way, if it is the value of VM1, this VM is allocated in HOST1.

Keep in mind that although the value of the performance of each VM is calculated based on the MIPS value of each host, the scheduler is responsible to assign each VM to each host.

The VM performance values are created to be similar to the host performance, since there is only one VM per host. Table 15 shows the MIPS for each VM.

VM ID	1	2	3	4	5	6	7	8	9	10
MIPS	**525**	**583**	**637**	**685**	**728**	**765**	**798**	**825**	**847**	**863**
VM ID	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
MIPS	875	881	882	877	868	853	833	807	777	741

Table 15. MIPS VMS.

One all virtual machines have been created, the next step is the decision of the scheduler to decide on which host each VM will be allocated, taking into account the FRBS system explained throughout this work.

Three different experiments have been performed, one to optimize the power, another to optimize the execution time (makespan) and another multi-objective experiment that takes into account the previous two. That is, it optimizes both time and power fitness.

For each experiment, the interpretability of the chosen knowledge bases will be analysed.

In all experiments, it has been used as an input workflow Montage_1000. The task scheduler is responsible for following the order established by the workflow as if it were a

real environment. The rule base has been obtained using the Pittsburgh approach algorithm in Matlab. These KBs will be generated randomly and through the learning system for a number of iterations, the best KB will be chosen.

7.2 POWER OPTIMIZATION EXPERIMENT

The FRBS system of this experiment is composed of five input variables and one output variable.

The input variables or antecedents are composed of three membership functions: low, average and high. The output variable or consequent has five membership functions: very low, low, average, high and very high.

The antecedents are the following:

- **MIPS.** Millions Instructions per Second being used in the virtual machine. Minimum performance that a VM has to have to execute the task. It is the ration between the length of task (measured in Millions of Instructions (MI) and the deadline of task (measured in seconds). It is the performance of VM
- **Power.** Power is the power consumed for the host when a task is processed.
- **Length.** Task size measured in MI.
- **Utilization.** Utilization of CPU in percentage. Utilization created by all cloudlets running on the VM.
- **PowerMax.** Get the maximum power that can be consumed by the host.

Moreover, the consequent is '**selection'**. This variable is the degree of selection de VM to process the task.

30 Experiments have been carried out with 700 iterations each. From each of these experiments the best knowledge base obtained according to fitness has been chosen. In this case, fitness is the power consumed. The population of this algorithm is composed by 20 KBs.

To know which of these experiments has the best knowledge base (based on the rules base); a convergence study has been carried out.

The convergence study consist of the arithmetic mean of all the power values obtained from each experiment in the 700 iterations and obtain the minimum value of power consumed, choosing the knowledge base that obtains that result. Since in this case, it is the parameter to optimize.

Thanks to this study, it is know if the number of iterations has been enough or not. The following figures show the convergence curve of these experiments and how starting from iteration 250-300 begins to converge. The x-axis shows the number iteration and the y-axis the power value in watts.

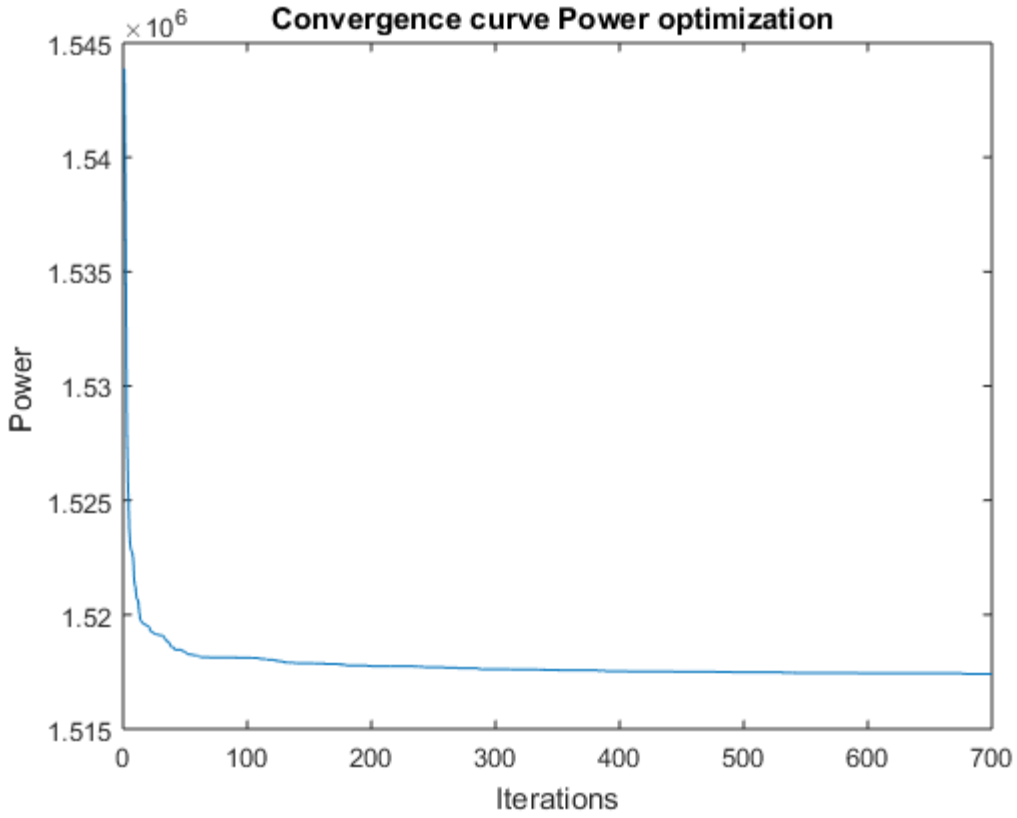

Figure 40 CONVERGENCE CURVE POWER OPTIMIZATION EXPERIMENT..

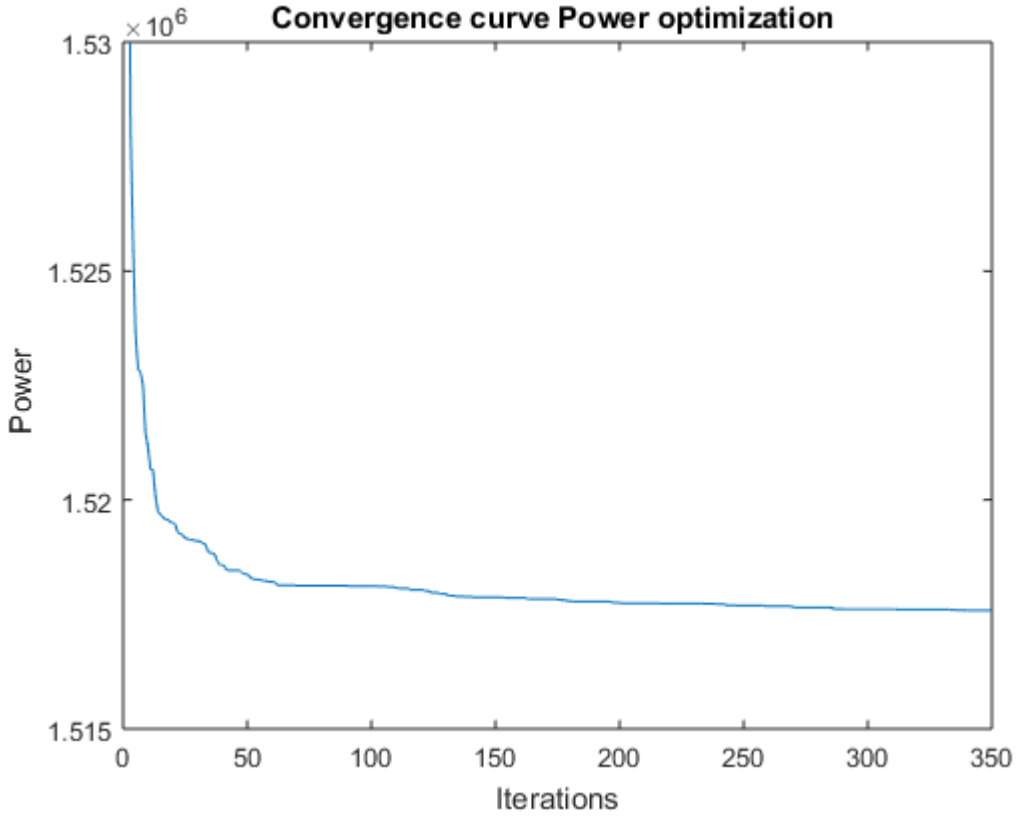

Figure 41. ZOOM CONVERGENCE CURVE POWER OPTIMIZATION EXPERIMENT.

The KB chosen is the number 23 formed by six rules. The rule base is as follows:

IF **'MIPS'** is high AND **'power'** is average AND **'length'** is high AND **'utilization'** is average AND **'powerMax'** is low THEN **'selection'** is NOT average.

IF **'MIPS'** is low AND **'power'** is high AND **'length'** is NOT high AND **'utilization'** is low AND **'powerMax'** is average THEN **'selection'** is high.

IF **'MIPS'** is average AND **'power'** is average AND **'length'** is low AND **'utilization'** is average AND **'powerMax'** is NOT average THEN **'selection'** is NOT very high.

IF **'MIPS'** is low AND **'power'** is high AND **'length'** is average AND **'utilization'** is average AND **'powerMax'** is average THEN **'selection'** is NOT low.

IF **'power'** is high AND **'length'** is average AND **'utilization'** is high AND **'powerMax'** is high THEN **'selection'** is NOT low.

IF **'MIPS'** is average AND **'power'** is low AND **'length'** is high AND **'utilization'** is NOT low AND **'powerMax'** is low THEN **'selection'** is low.

7.2.1 INTERPRETABILITY ANALYSIS

With this KB, an interpretability index of 0.4302 in the range [0,1] is obtained. Being 0 value for a non-interpretable system and 1 value for a very interpretable system.

Taking into account the index of interpretability obtained, we could ensure that the knowledge base is poorly interpretable.

For assessing interpretability, there is a hierarchical fuzzy system compose by the rule base dimension (RB1), the rule base complexity (RB2), the rule base interpretability (RB3) and finally, interpretability index (RB4). Figure 44 shows the fuzzy KB of the RBs.

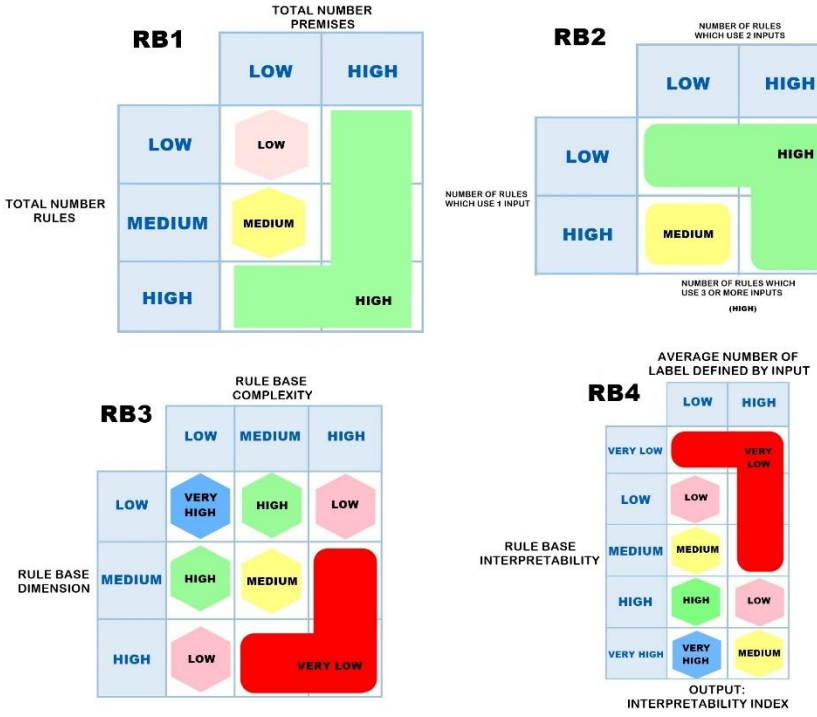

Figure 42. RB1, RB2, RB3 AND RB4 FUZZY RULES.

RB1 take into account the total number of rules and the total number of premises to make an estimation of rule base dimension. RB1 implements several fuzzy rules; if the number of premises is low (29 premises) and the number of rules is low, the output is low.

RB2 makes an estimation about the number of rules, which use one input, number of rules, which use two inputs, number of rules, which use three or more inputs, and the total number of labels defined by input.

If there are a large number of rules, which use three or more inputs, the higher the rule base complexity. In this case, all the rules use more than three inputs, so the output system is high.

RB3 is a combination of rule base dimension and complexity (RB1 and RB2). If the rule base dimension is low and the complexity of rule base is high, the rule base interpretability is low. In this case, like rules base dimension (RB1) is low and rule base complexity (RB2) is high, the rule base interpretability (RB3) is low.

RB4 integrates RB3 with an evaluation of the interpretability of the variables, considering the average number of labels defined by input.

If rule base interpretability (output RB3) is low and the average number of label defined by input is low (three labels per input), the interpretability index is low. In this case, as the output of RB3 is low and the average number label defined by input is low, then the interpretability index is low. As indicated above, the interpretability index has a value of 0.4302.

Figure 45 shows the values obtained from the different blocks of the hierarchical system and the result of the interpretability index for this KB.

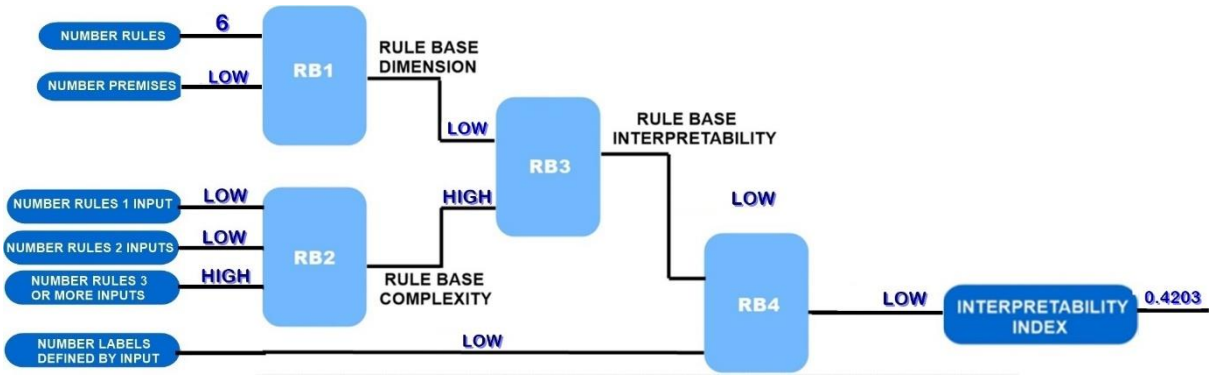

Figure 43. HIERARCHICAL FUZZY SYSTEM WITH REAL VALUES KB23.

7.3 TIME OPTIMIZATION EXPERIMENT

The FRBS system of this experiment is composed of five input variables and one output variable, as in the previous experiment.

The input variables or antecedents are composed of three membership functions: low, average and high. The output variable or consequent has five membership functions: very low, low, average, high and very high.

The antecedents are the following:

- **MIPS.** Millions Instructions per Second being used in the virtual machine. Minimum performance that a VM has to have to execute the task. It is the ration between the length of task (measured in Millions of Instructions (MI) and the deadline of task (measured in seconds).
- **PEs.** Processing elements used in the virtual machine.
- **BW.** Bandwidth used by the virtual machine.
- **RAM.** Random Access Memory used in in the virtual machine.
- **TRANSFERT.** Transfer time for virtual machine.

Moreover, the consequent is '**output'**. This variable is the degree of selection de VM to process the task.

32 Experiments have been carried out with 700 iterations each. From each of these experiments the best knowledge base obtained according to fitness has been chosen. In this case, fitness is the time consumed. The population of this algorithm is composed by 20 KBs.

To know which of these experiments has the best knowledge base (based on the rules base), a convergence study has been carried out.

The convergence study consist of the arithmetic mean of all the power values obtained from each experiment in the 700 iterations and obtain the minimum value of time, choosing the knowledge base that obtains that result. Since in this case, it is the parameter to optimize.

Thanks to this study, it is know if the number of iterations has been enough or not. The following figures show the convergence curve of these experiments and how starting from iteration 150-160 begins to converge. The x axis shows the number iterations and the y-axis the time value in seconds.

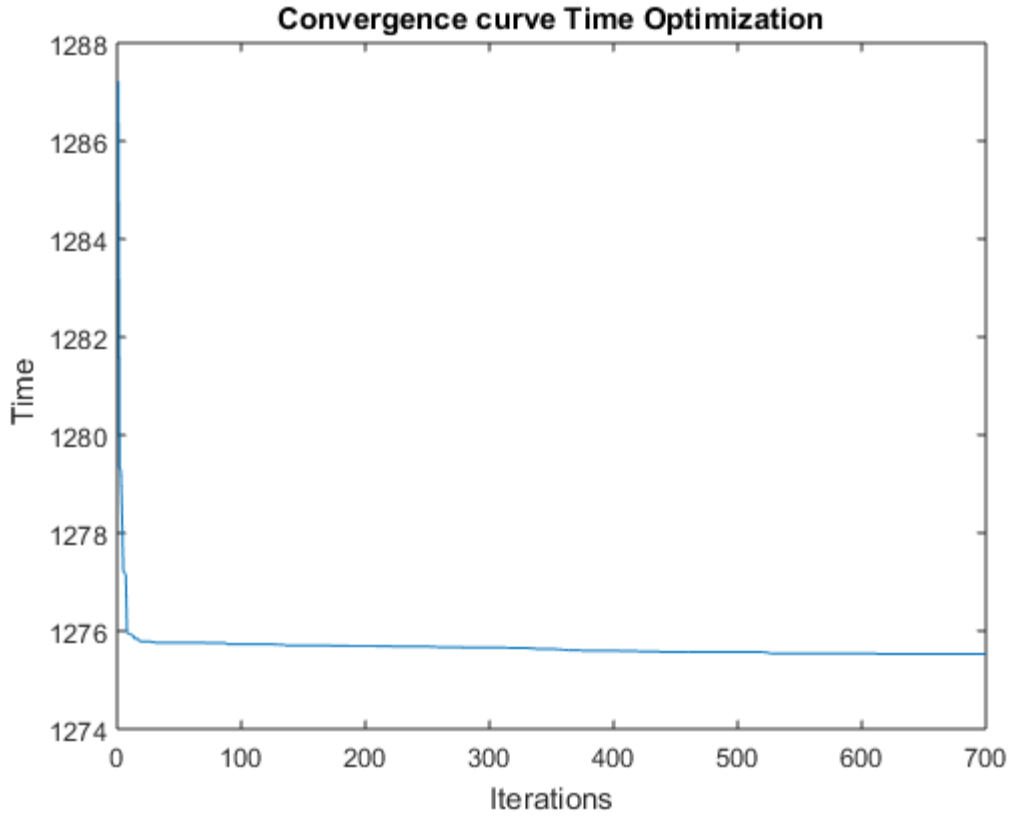

Figure 44. CONVERGENCE CURVE TIME OPTIMIZATION EXPERIMENT.

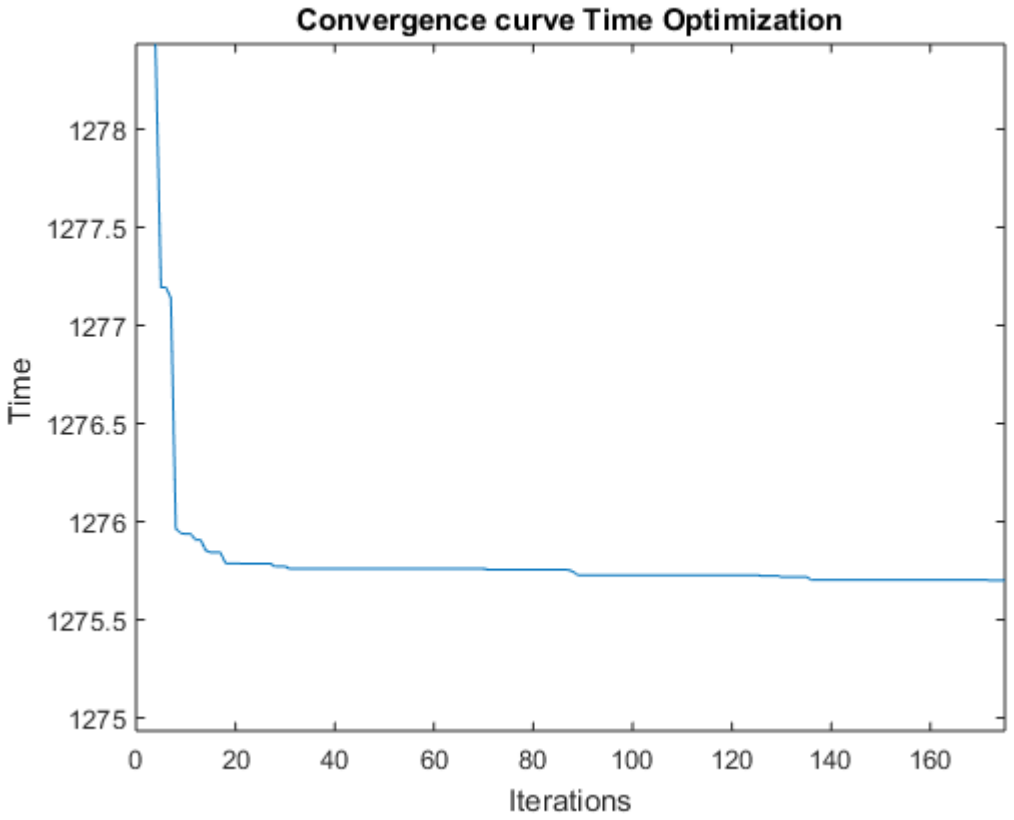

Figure 45. ZOOM CONVERGENCE CURVE TIME OPTIMIZATION EXPERIMENT.

The KB chosen is the number 26 composed of sixteen rules. The rule base is as follows:

IF '**MIPS'** is NOT high, '**PEs'** is average, **'BW'** is high, **'RAM'** is average, **'TRANSFERT'** is high THEN **'output'** is high.

IF **'MIPS'** is average, **'PEs'** is NOT average, **'BW'** is low, **'RAM'** is average, **'TRANSFERT'** is low THEN **'output'** is very low.

IF **'MIPS'** is low, **'PEs'** is average, **'BW'** is average, **'RAM'** is NOT low, **'TRANSFERT'** is low THEN **'output'** is NOT low.

IF **'MIPS'** is average, **'PEs'** is low, **'BW'** is average, **'RAM'** is low, **'TRANSFERT'** is high THEN **'output'** is low.

IF **'PEs'** is low, **'BW'** is high, **'RAM'** is average, **'TRANSFERT'** is average THEN **'output'** is low.

IF **'MIPS'** is NOT low, **'PEs'** is average, **'BW'** is low, **'RAM'** is high, **'TRANSFERT'** is high THEN **'output'** is NOT low.

IF **'MIPS'** is average, **'PEs'** is high, **'BW'** is average, **'RAM'** is low, **'TRANSFERT'** is low THEN **'output'** is low.

IF **'MIPS'** is high, **'PEs'** is low, **'BW'** is high, **'RAM'** is high, **'TRANSFERT'** is NOT low THEN **'output'** is very high.

IF **'MIPS'** is high, **'PEs'** is average, **'BW'** is high, **'RAM'** is average, **'TRANSFERT'** is NOT high THEN **'output'** is very high.

IF **'MIPS'** is high, **'PEs'** is average, **'BW'** is high, **'RAM'** is average, **'TRANSFERT'** is high THEN **'output'** is NOT low.

IF **'MIPS'** is average, **'BW'** is average, **'RAM'** is average, **'TRANSFERT'** is high THEN **'output'** is low.

IF **'MIPS'** is NOT low, **'PEs'** is low, **'BW'** is average, **'RAM'** is low, **'TRANSFERT'** is average THEN **'output'** is NOT average.

IF **'MIPS'** is high, **'PEs'** is high, **'BW'** is low, **'RAM'** is low THEN **'output'** is average.

IF **'MIPS'** is NOT low, **'PEs'** is low, **'BW'** is low, **'RAM'** is high, **'TRANSFERT'** is low THEN **'output'** is NOT low.

IF **'MIPS'** is NOT average, **'PEs'** is NOT high, **'BW'** is average, **'RAM'** is low, **'TRANSFERT'** is high THEN **'output'** is high.

IF **'MIPS'** is average, **'BW'** is low, **'RAM'** is NOT average, **'TRANSFERT'** is high THEN **'output'** is average.

7.3.1 INTERPRETABILITY ANALYSIS

With this KB, an interpretability index of 0.1626 in the range [0,1] is obtained. Being 0 value for a non-interpretable system and 1 value for a very interpretable system.

Taking into account the index of interpretability obtained, we could ensure that the knowledge base is not interpretable or very low interpretable.

For assessing interpretability, there is a hierarchical fuzzy system compose by the rule base dimension (RB1), the rule base complexity (RB2), the rule base interpretability (RB3) and finally, interpretability index (RB4).

RB1 take into account the total number of rules and the total number of premises to make an estimation of rule base dimension. RB1 implements several fuzzy rules; if the number of premises is high and the number of rules is high, the output is high.

RB2 makes an estimation about the number of rules, which use one input, number of rules, which use two inputs, number of rules, which use three or more inputs, and the total number of labels defined by input.

If there are a large number of rules, which use three or more inputs, the higher the rule base complexity. In this case, all the rules use more than three inputs, so the output system is high.

RB3 is a combination of rule base dimension and complexity (RB1 and RB2).

If the rule base dimension is high and, whatever the complexity of rule base, the rule base interpretability is low or very low. In this case, like rules base dimension (RB1) is high and rule base complexity (RB2) is high, the rule base interpretability (RB3) is very low.

RB4 integrates RB3 with an evaluation of the interpretability of the variables, considering the average number of labels defined by input.

If rule base interpretability (output RB3) is very low and whatever the average number of label defined by input (low or high), the interpretability index is very low. In this case, as the output of RB3 is very low and the average number label defined by input is low, then the interpretability index is very low. As indicated above, the interpretability index has a value of 0.1626.

Figure 48 shows the values obtained from the different blocks of the hierarchical system and the result of the interpretability index for this KB.

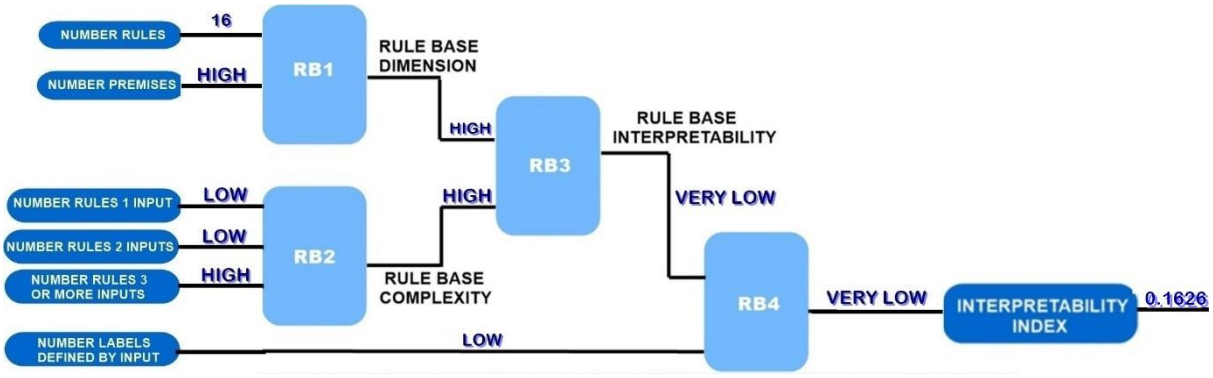

Figure 46. HIERARCHICAL FUZZY SYSTEM WITH REAL VALUES KB.

7.4 MULTI-OBJECTIVE OPTIMIZATION EXPERIMENT

The FRBS system of this experiment is composed of nine input variables and one output variable. This is because time and power are optimized, there must be variables related to time and power. This experiments is the fusion of the previous two.

The input variables or antecedents are composed of three membership functions: low, average and high. The output variable or consequent has five membership functions: very low, low, average, high and very high.

The antecedents are the following:

- **MIPS.** Millions Instructions per Second being used in the virtual machine. Minimum performance that a VM has to have to execute the task. It is the ration between the length of task (measured in Millions of Instructions (MI) and the deadline of task (measured in seconds). It is the performance of VM
- **Power.** Power is the power consumed for the host when a task is processed.
- **Length.** Task size measured in MI.
- **Utilization.** Utilization of CPU in percentage. Utilization created by all cloudlets running on the VM.
- **PowerMax.** Get the maximum power that can be consumed by the host.
- **PEs.** Processing elements used in the virtual machine.
- **BW.** Bandwidth used by the virtual machine.
- **RAM.** Random Access Memory used in in the virtual machine.
- **TRANSFERT.** Transfer time for virtual machine.

Moreover, the consequent is '**selection'**. This variable is the degree of selection de VM to process the task.

32 Experiments have been carried out with 700 iterations each. From each of these experiments the best knowledge base obtained according to fitness has been chosen. In this case, fitness is the time consumed. The population of this algorithm is composed by 20 KBs.

To know which of these experiments has the best knowledge base (based on the rules base), a convergence study has been carried out.

The convergence study consist of the arithmetic mean of all the power values obtained from each experiment in the 400 iterations and obtain the minimum value of time and power, choosing the knowledge base that obtains that result. 400 iterations have been taken into account because seeing the two previous experiments; the convergence was well before the 700 iterations. In the convergence figures, the chosen number of iteration is correct and does not fall short.

Thanks to this study, it is know if the number of iterations has been enough or not. The following figures show the convergence curve of these experiments and how starting from iteration 40 begins to converge. The x-axis shows the number iteration and the y-axis the power value in watts for figure 49 and the y-axis the time value in seconds for the figure 50.

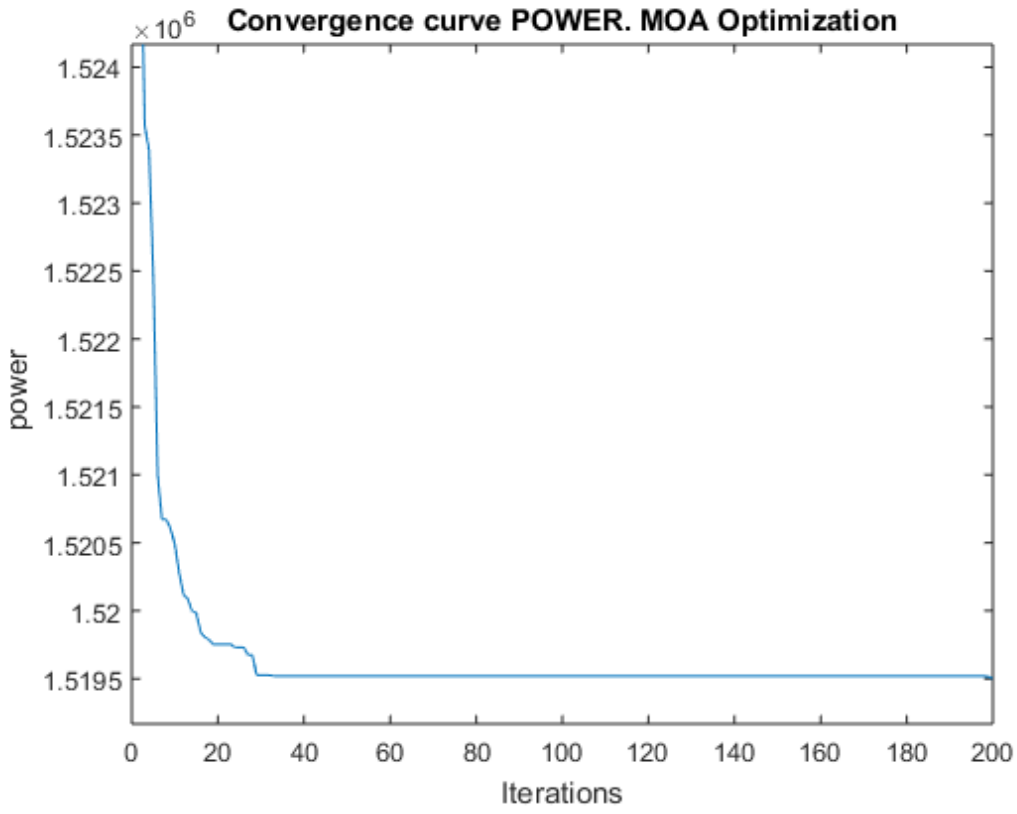

Figure 47. CONVERGENCE CURVE POWER. MOA OPTIMIZATION EXPERIMENT.

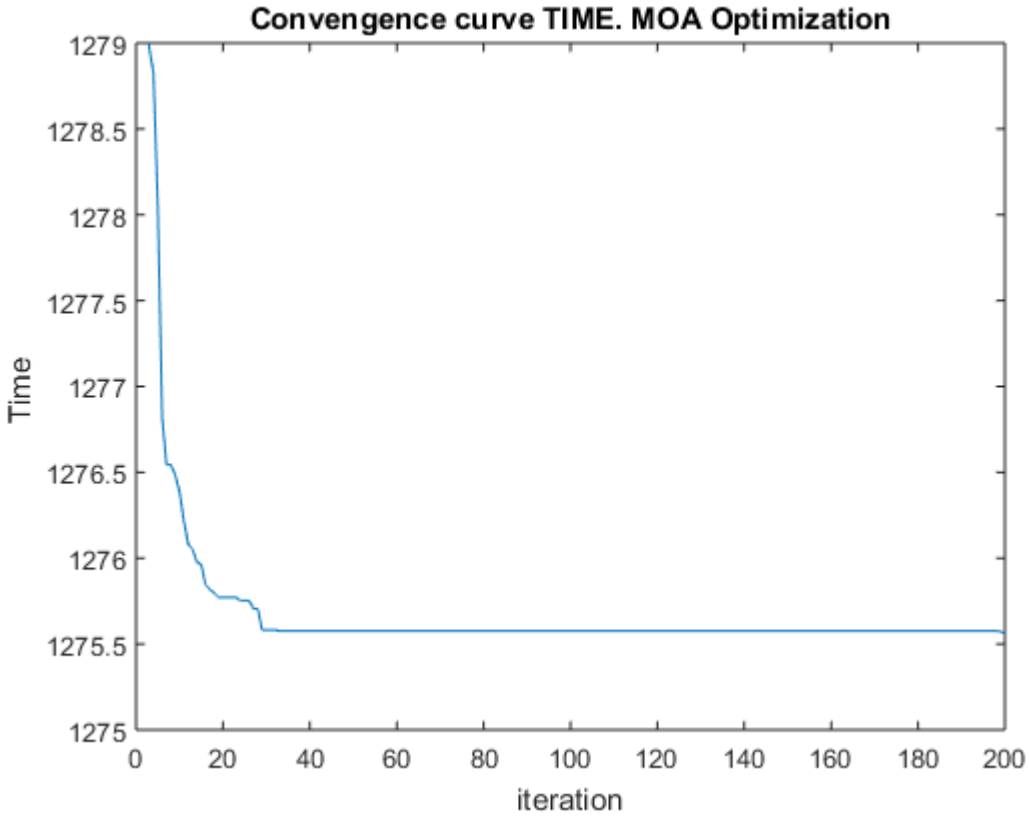

Figure 48. ZOOM CONVERGENCE CURVE TIME. MOA OPTIMIZATION EXPERIMENT.

The KB chosen is the number 14 composed of four rules. The rule base is as follows:

IF **'MIPS'** is average, '**power'** is NOT average, '**utilization'** is average, '**powerMax'** is low, **'PEs'** is NOT high, **'BW'** is NOT average, **'TRANSFERT'** is NOT average THEN '**selection'** is high.

IF '**length**' is NOT low, '**powerMax'** is NOT high, **'PEs'** is high, **'BW'** is NOT high, **'RAM'** is NOT low THEN '**selection'** is low.

IF **'MIPS'** is low, '**power'** is high, '**length'** is low, '**utilization'** is NOT average, '**powerMax'** is NOT average, **'PEs'** is low, **'RAM'** is NOT high, **'TRANSFERT'** is high THEN '**selection'** is NOT average.

IF **'MIPS'** is average, '**power'** is NOT average, '**utilization'** is low, '**powerMax'** is high, **'PEs'** is low, **'RAM'** is NOT average, **'TRANSFERT'** is NOT low THEN '**selection'** is NOT low.

7.4.1 INTERPRETABILITY ANALYSIS

With this KB, an interpretability index of 0.4692 in the range [0,1] is obtained. Being 0 value for a non-interpretable system and 1 value for a very interpretable system.

Taking into account the index of interpretability obtained, we could ensure that the knowledge base is poorly interpretable, although it is almost at average value.

For assessing interpretability, there is a hierarchical fuzzy system compose by the rule base dimension (RB1), the rule base complexity (RB2), the rule base interpretability (RB3) and finally, interpretability index (RB4).

RB1 take into account the total number of rules and the total number of premises to make an estimation of rule base dimension. RB1 implements several fuzzy rules; if the number of premises is low (27), whatever the number of rules (in this case low), the output is high.

RB2 makes an estimation about the number of rules, which use one input, number of rules, which use two inputs, number of rules, which use three or more inputs, and the total number of labels defined by input.

If there are a large number of rules, which use three or more inputs, the higher the rule base complexity. In this case, all the rules use more than three inputs, so the output system is high. Thus is because the number of rules, which use one input, is low and the number of rules, which use two inputs, is low, too.

RB3 is a combination of rule base dimension and complexity (RB1 and RB2).

If the rule base dimension is low and the complexity of rule base is high, the rule base interpretability is low. In this case, like rules base dimension (RB1) is low and rule base complexity (RB2) is high, the rule base interpretability (RB3) is low.

RB4 integrates RB3 with an evaluation of the interpretability of the variables, considering the average number of labels defined by input.

If rule base interpretability (output RB3) is low and the average number of label defined by input is low (three), the interpretability index is low. As indicated above, the interpretability index has a value of 0.4692.

Figure 51 shows the values obtained from the different blocks of the hierarchical system and the result of the interpretability index for this KB.

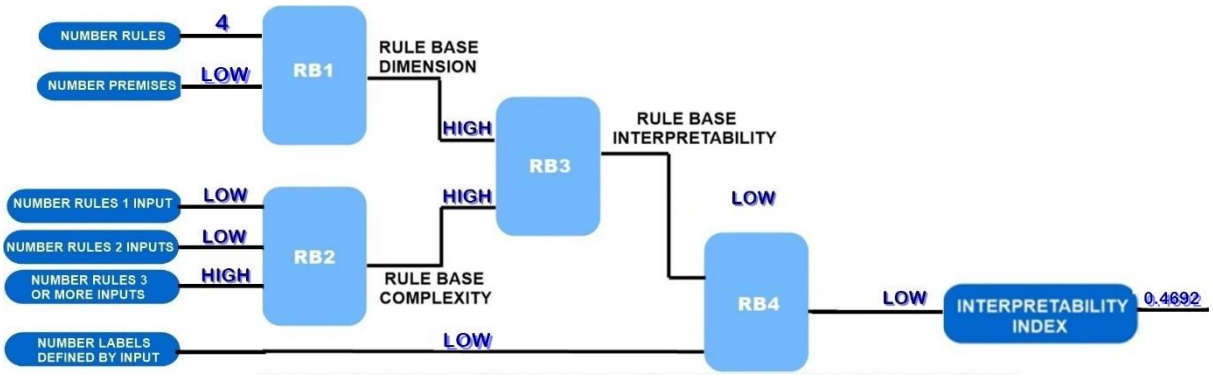

Figure 49. HIERARCHICAL FUZZY SYSTEM WITH REAL VALUES KB14.

8. COMPARISON SCHEDULING ALGORITHMS

In this section, the proposal algorithm will be compared with scheduling algorithms already known. Thus demonstrating that this algorithm improves the several techniques already implemented. The idea is that in addition to analysing the interpretability of scheduler that optimize energy, this scheduler improves the results of other scheduling strategies.

The algorithms to be compared are ROUND ROBIN, STATIC, POWERMIN, MAXMIN and MINMIN.

- **ROUND ROBIN**

Round Robin is a scheduling algorithm designed to select all the virtual machines of a group in an equitable way and in a rational order, usually starting with the first VM of the list until reaching the last one and, starting again from the VM.

For the equitable use of equipment resources, it is limiting each process to a small period (quantum), and then suspending this process to give opportunity to another process and so on.

The algorithm gives a maximum CPU usage time to each process, after this time; it is evicted and returned to the ready state. FIFO (first in, first out) plans the list of processes. [40] [41]

- **STATIC**

This algorithm does not do any planning. It is responsible for checking if a task or job has been sent to a virtual machine or not.

- **POWER MIN**

The POWER MIN algorithm is an adaptation of MINMIN algorithm that takes into account the power consumption. This algorithm selects the virtual machine that can achieve the least power consumption for the execution of a task. For this, an estimation is made previously of what VM would consume less power when executing that tasks, and thus choose the one with the lowest power consumption.

The power consumed in the current frequency and utilization levels and the time it would take to process the task are taken into account. The time being a ratio between the length of the task and the performance of virtual machine. [10]

- **MINMIN**

The MINMIN algorithm is a simple algorithm that manages resources efficiently. In this algorithm, the completion time of a task is calculated and initially, resources are assigned to those tasks that have a minimum execution time. It is responsible to order the workflow jobs in order of completion time. The job with minimum execution time is chosen and associated to the VM.

The main objective of this algorithm is to minimize the overall execution time by the prioritizing of the shortest jobs. [33]

This process is repeated until all unassigned workflows are assigned to the corresponding virtual machine. [42][43]

- **MAXMIN**

The MAXMIN algorithm is very similar to MINMIN algorithm. It selects the virtual machine that has a minimum completion time for all selected jobs, but unlike MINMIN algorithm, it selects the workflow with the maximum completion time and is assigned to this resource.

The main objective of this algorithm is to reduce the waiting time for large jobs [44].

A comparison of the proposed algorithm will be made with the scheduling strategies in each of experiments performed. Thus, an analysis of results can be done with these strategies.

As indicated in section 7, in the scenario that has been carried out for each of the experiments, the process of training of each experiment can be seen with a Montage input with workflows of 1000 jobs.

For the comparison, each scheduling algorithm will be carried out for this same input file, so that the results can be analysed under the same conditions.

In addition, a stage of validation of the results, where both the proposed algorithm and the different scheduling algorithm will be simulated with other input files than training. These input files will be Montage with workflows of 100 jobs and with a new input, Inspiral with workflows of 50 and 100 jobs.

8.1 POWER OPTIMIZATION EXPERIMENT

In this experiment to optimize power consumed, the proposal algorithm is called POWERFUZZYSEQ.

8.1.1 FIRST STAGE: TRAINING

	TRAINING STAGE	
MONTAGE	1000 JOBS	
ALGORITHM	POWER [W]	TIME [s]
POWERFUZZYSEQ	1.5168312e+06	1.273320e+03
POWER MIN	1.5197735e+06	1.275790e+03
MIN MIN	1.5255392e+06	1.280630e+03
MAX MIN	1.5226444e+06	1.278200e+03
ROUND ROBIN	1.8293725e+06	1.535680e+03
STATIC	2.7006717e+07	2.267104e+04

Table 16. TRAINING STAGE POWER EXPERIMENT.

The following table show the percentage of the proposal algorithm improvement compared to the other percentage scheduling strategies in the training stage.

	TRAINING STAGE IMPROVEMENT %
MONTAGE	1000 JOBS
ALGORITHM	
POWER MIN	0.1936 %
MIN MIN	0.5708 %
MAX MIN	0.3818 %
ROUND ROBIN	17.08 %
STATIC	94.38 %

Table 17. TRAINING STAGE % POWER EXPERIMENT.

8.1.2 SECOND STAGE: VALIDATION

	VALIDATION STAGE			
	MONTAGE 100 JOBS		INSPIRAL 100 JOBS	
ALGORITHM	POWER[W]	TIME[s]	POWER[W]	TIME
POWERFUZZYSEQ	1.74330000E+05	1.46340000E+02	2.32443490E+06	1.95127000E+03
POWER MIN	1.75294910E+05	1.47150000E+02	2.46020180E+06	2.06524000E+03
MIN MIN	1.75104310E+05	1.46990000E+02	2.68772920E+06	2.25624000E+03
MAX MIN	1.74901800E+05	1.46820000E+02	2.46505140E+06	2.06931000E+03
ROUND ROBIN	2.10162330E+05	1.76420000E+02	2.43104850E+06	2.04077000E+03
STATIC	2.56476270E+06	2.15302000E+03	4.79247830E+07	4.02309100E+04

Table 18. VALIDATION STAGE POWER EXPERIMENT.

The following table show the percentage of the proposal algorithm improvement compared to the other percentage scheduling strategies in the validation stage. The comparison will be for both Montage_100 and Inspiral_100.

	VALIDATION STAGE IMPROVEMENT %	
	MONTAGE 100 JOBS	INSPIRAL 100 JOBS
ALGORITHM	POWER	POWER
POWER MIN	0.5504 %	5.5185 %
MIN MIN	0.4422 %	13.5168 %
MAX MIN	0.3269 %	5.7044 %
ROUND ROBIN	17.0498 %	4.3855 %
STATIC	93.2029 %	95.1498

Table 19. VALIDATION STAGE % POWER EXPERIMENT.

8.1.3 ANALYSIS AND CONCLUSION

In tables 16, 17, the proposal algorithm improves all the other strategies. The table 16 shows the value of the power and time consumed, since it is a power optimization experiment, it is necessary to focus mainly at the power values, but it is observed that at lower power consumption, the time is also less.

These are very good results, but it was expected that in the training stage it would achieve better results than the other strategies, since it has been trained with this scheduling algorithm and with this input file.

The best results are obtained compared to STATIC algorithm, what could be expected since this strategy does not consider the state of the VMs to scheduler jobs and considers a static assignment during the whole process.

The worst results are obtained compared to POWERMIN algorithm, since this strategy consider the state of VM and the power consumed for each task.

In order to the validation stage, it can be seen that for both Montage 100 and Inspiral 100, the proposed algorithm improves power consumption with respect to the other algorithms. As in the training stage, the best results are obtained in comparison with the STATIC algorithm.

In the Montage 100 case, regardless of the STATIC algorithm, the best result are obtained in comparison with ROUNDROBIN algorithm, and the worst results with the MAXMIN algorithm. This can be explained because the MAXMIN algorithm tends to improve jobs that have large completion times, being an input file with fewer tasks, this favors this strategy.

In the Inspiral 100 case, it should be added that this input file has more complex tasks than Montage. The best results are obtained, regardless the STATIC algorithm, in comparison with MINMIN algorithm, and the worst results are obtained in comparison with ROUNDROBIN algorithm.

8.2 TIME OPTIMIZATION EXPERIMENT

In this experiment to optimize time consumed, the proposal algorithm is called FUZZY_SCH.

8.2.1 FIRST STAGE: TRAINING

	TRAINING STAGE	
MONTAGE	**1000 JOBS**	
ALGORITHM	**POWER** [W]	**TIME** [s]
FUZZY_SCH	1.5187610e+06	1.2749400e+03
POWER MIN	1.5197735e+06	1.2757900e+03
MIN MIN	1.5255392e+06	1.2806300e+03
MAX MIN	1.5226444e+06	1.2782000e+03
ROUND ROBIN	1.8293725e+06	1.5356800e+03
STATIC	2.7006717e+07	2.2671040e+04

Table 20. TRAINING STAGE TIME EXPERIMENT.

The following table show the percentage of the proposal algorithm improvement compared to the other percentage scheduling strategies in the training stage.

	TRAINING STAGE IMPROVEMENT %
MONTAGE ALGORITHM	1000 JOBS TIME
POWER MIN	0.0666 %
MIN MIN	0.4443 %
MAX MIN	0.2550 %
ROUND ROBIN	16.9788 %
STATIC	94.3763 %

Table 21. TRAINING STAGE % TIME EXPERIMENT.

8.2.2 SECOND STAGE: VALIDATION

	VALIDATION STAGE			
	MONTAGE 100 JOBS		INSPIRAL 100 JOBS	
ALGORITHM	POWER[W]	TIME[s]	POWER[W]	TIME[s]
FUZZY_SCH	1.7529491e+05	1.4615000e+02	2.4386221e+06	2.0439100e+03
POWER MIN	1.7594291e+05	1.4715000e+02	2.4602018e+06	2.0652400e+03
MIN MIN	1.7610431e+05	1.4799000e+02	2.6877292e+06	2.2562400e+03
MAX MIN	1.7590180e+05	1.4782000e+02	2.4650514e+06	2.0693100e+03
ROUND ROBIN	2.1016233e+05	1.7642000e+02	2.4510485e+06	2.0607700e+03
STATIC	2.5647627e+06	2.1530200e+03	4.7924783e+07	4.0230910e+04

Table 22. VALIDATION STAGE TIME EXPERIMENT.

The following table show the percentage of the proposal algorithm improvement compared to the other percentage scheduling strategies in the validation stage. The comparison will be for both Montage and Inspiral workflows.

	VALIDATION STAGE IMPROVEMENT %	
	MONTAGE 100 JOBS	INSPIRAL 100 JOBS
ALGORITHM	TIME	TIME
POWER MIN	0.6796 %	1.0328 %
MIN MIN	1.2433 %	9.4108 %
MAX MIN	1.1298 %	1.2275 %
ROUND ROBIN	17.1579 %	0.8181 %
STATIC	93.2119 %	94.9196 %

Table 23. VALIDATION STAGE % TIME EXPERIMENT.

8.2.3 ANALYSIS AND CONCLUSION

In tables 20 and 21, the proposal algorithm improves all the other strategies. The table 20 shows the value of the power and time consumed, since it is a time optimization experiment, it is necessary to focus mainly at the time values, but it is observed that at lower time consumption, the power is also less.

These are very good results, but it was expected that in the training stage it would achieve better results than the other strategies, since it has been trained with this scheduling algorithm and with this input file.

The best results are obtained compared to STATIC algorithm, what could be expected since this strategy does not consider the state of the VMs to scheduler jobs and considers a static assignment during the whole process.

The worst results are obtained compared to POWERMIN algorithm, since this strategy consider the state of VM and the power consumed for each task and the time spent in each process. There is very little improvement with respect to POWERMIN, but it is better.

In order to the validation stage, it can be seen that for both Montage 100 and Inspiral 100, the proposed algorithm improves power consumption with respect to the other algorithms. As in the training stage, the best results are obtained in comparison with the STATIC algorithm.

In the Montage 100 case, regardless of the STATIC algorithm, the best result are obtained in comparison with ROUND ROBIN algorithm, and the worst results with the POWERMIN algorithm.

In the Inspiral 100 case, it should be added that this input file has more complex tasks than Montage. The best results are obtained, regardless the STATIC algorithm, in comparison with MINMIN algorithm, and the worst results are obtained in comparison with ROUNDROBIN algorithm. With this workload, the results are better since the traces of tasks are more complex and this algorithm works better in scenarios that are more complex.

8.3 MULTI-OBJECTIVE OPTIMIZATION EXPERIMENT

In this experiment to optimize power consumed, the proposal algorithm is called POWERFUZZYSEQ.

The first stage, the training is executed with Montage 1000 jobs and the validation stage is executed with Montage 50 and 100 jobs and with other workflow like Inspiral 30 and 100 jobs.

8.3.1 FIRST STAGE: TRAINING

	TRAINING STAGE	
MONTAGE	**1000 JOBS**	
ALGORITHM	**POWER** [W]	**TIME** [s]
POWERFUZZYSEQ	1.5196934e+06	1.2757200e+03
POWER MIN	1.5197735e+06	1.2757900e+03
MIN MIN	1.5255392e+06	1.2806300e+03
MAX MIN	1.5226444e+06	1.2782000e+03
ROUND ROBIN	1.8293725e+06	1.5356800e+03
STATIC	2.7006717e+07	2.2671040e+04

Table 24. TRAINING STAGE MOA EXPERIMENT.

The following table show the percentage of the proposal algorithm improvement compared to the other percentage scheduling strategies in the training stage.

	TRAINING STAGE IMPROVEMENT %	
MONTAGE	**1000 JOBS**	
ALGORITHM	**POWER**	**TIME**
POWER MIN	0.0053 %	0.0055 %
MIN MIN	0.3832 %	0.3834 %
MAX MIN	0.1938 %	0.1940 %
ROUND ROBIN	16.9282 %	16.9280 %
STATIC	94.3729 %	94.3729 %

Table 25. TRAINING STAGE % MOA EXPERIMENT.

8.3.2 SECOND STAGE: VALIDATION

	VALIDATION STAGE			
	MONTAGE 100 JOBS		MONTAGE 50 JOBS	
ALGORITHM	POWER	TIME	POWER	TIME
POWERFUZZYSEQ	1.7409011e+05	1.46140e+02	1.1958e+05	1.0038e+02
POWER MIN	1.7529491e+05	1.47150e+02	1.1086e+05	9.3060e+01
MIN MIN	1.7510431e+05	1.46990e+02	1.1224e+05	9.4220e+01
MAX MIN	1.7490180e+05	1.46820e+02	1.1060e+05	9.2840e+01
ROUND ROBIN	2.1016233e+05	1.76420e+02	1.4602e+05	1.2258e+02
STATIC	2.5647627e+06	2.15302e+03	1.2115e+06	1.0170e+03

Table 26. VALIDATION STAGE MOA EXPERIMENT MONTAGE.

	VALIDATION STAGE			
	INSPIRAL 30 JOBS		INSPIRAL 100 JOBS	
ALGORITHM	POWER	TIME	POWER	TIME
POWERFUZZYSEQ	2.0081e+06	1.6857e+03	2.3738084e+06	1.99272e+03
POWER MIN	1.8134e+06	1.5222e+03	2.4602018e+06	2.06524e+03
MIN MIN	1.8497e+06	1.5528e+03	2.6877292e+06	2.25624e+03
MAX MIN	1.8208e+06	1.5285e+03	2.4650514e+06	2.06931e+03
ROUND ROBIN	3.0392e+06	2.5513e+03	2.4310485e+06	2.04077e+03
STATIC	1.5080e+07	1.2659e+04	4.7924783e+07	4.02309e+04

Table 27. VALIDATION STAGE MOA EXPERIMENT INSPIRAL.

The following table show the percentage of the proposal algorithm improvement compared to the other percentage scheduling strategies in the validation stage. The comparison will be for both Montage and Inspiral workflows.

	VALIDATION STAGE IMPROVEMENT %			
	MONTAGE 100 JOBS		MONTAGE 50 JOBS	
ALGORITHM	POWER	TIME	POWER	TIME
POWER MIN	0.6873 %	0.6864 %	-7.8658 %	-7.8659 %
MIN MIN	0.5792 %	0.5783 %	-6.5396 %	-6.5379 %
MAX MIN	0.4641 %	0.4632 %	-8.1193 %	-8.1215 %
ROUND ROBIN	17.1640 %	17.1636 %	18.1071 %	18.1106 %
STATIC	93.2122 %	93.2123 %	90.1296 %	90.1298 %

Table 28. VALIDATION STAGE % MOA EXPERIMENT MONTAGE.

	VALIDATION STAGE IMPROVEMENT %			
	INSPIRAL 30 JOBS		INSPIRAL 100 JOBS	
ALGORITHM	POWER	TIME	POWER	TIME
POWER MIN	-10.7367 %	-10.7410 %	3.5116 %	3.5116 %
MIN MIN	-8.5636 %	-8.5587 %	11.6798 %	11.6796 %
MAX MIN	-10.2867 %	-10.2846 %	3.7015 %	3.7012 %
ROUND ROBIN	33.9267 %	33.9278 %	2.3545 %	2.3545 %
STATIC	86.6837 %	86.6838 %	95.0468 %	95.0468 %

Table 29. VALIDATION STAGE % MOA EXPERIMENT INSPIRAL.

8.3.3 ANALYSIS AND CONCLUSION

In tables 24 and 25, the proposal algorithm improves all the other strategies. The table 24 shows the value of the power and time consumed.

These are very good results, but it was expected that in the training stage it would achieve better results than the other strategies, since it has been trained with this scheduling algorithm and with this input file.

In the training stage, he best results are obtained compared to STATIC algorithm, what could be expected since this strategy does not consider the state of the VMs to scheduler jobs and considers a static assignment during the whole process.

The worst results are obtained compared to POWERMIN algorithm, since this strategy consider the state of VM and the power consumed for each task.

In order to the validation stage, the algorithm has been validated with Montage and Inspiral workflows. Montage with 50 and 100 jobs and Inspiral con 30 and 100 jobs.

It can be seen that the proposal algorithm improves in complex situations, that is, when there are more tasks and they have more complexity to each other. Keep in mind that Inspiral traces are more complex than Montage traces.

That is why the best results are obtained with Montage 100 and Inspiral 100. And comparing these two, the best results are obtained with Inspiral 100, due to its complexity. The results are analysed with Inspiral 100, since a greater improvement is obtained both in time and power.

In the Montage 100 case, regardless of the STATIC algorithm, the best result are obtained in comparison with ROUNDROBIN algorithm, and the worst results with the MAXMIN algorithm. This can be explained because the MAXMIN algorithm tends to improve jobs that have large completion times, being an input file with fewer tasks, this favors this strategy.

In the Inspiral 100 case, it should be added that this input file has more complex tasks than Montage. The best results are obtained, regardless the STATIC algorithm, in comparison with MINMIN algorithm, and the worst results are obtained in comparison with ROUNDROBIN algorithm.

In terms of percentages, this algorithm improves a 95% of the static algorithm. However, the worst results are obtained with Round Robin algorithm; however, it implies a saving of 2.35%.

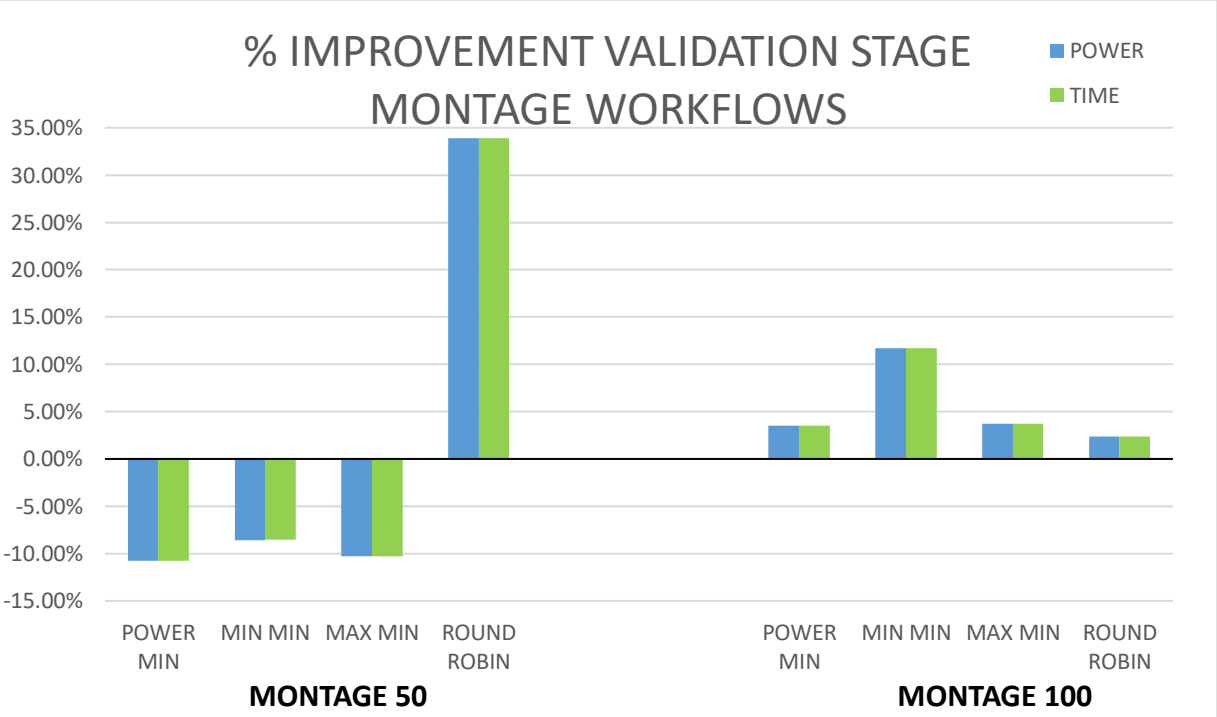

Figure 50. % IMPROVEMENT VALIDATION STAGE MONTAGE WORKFLOWS.

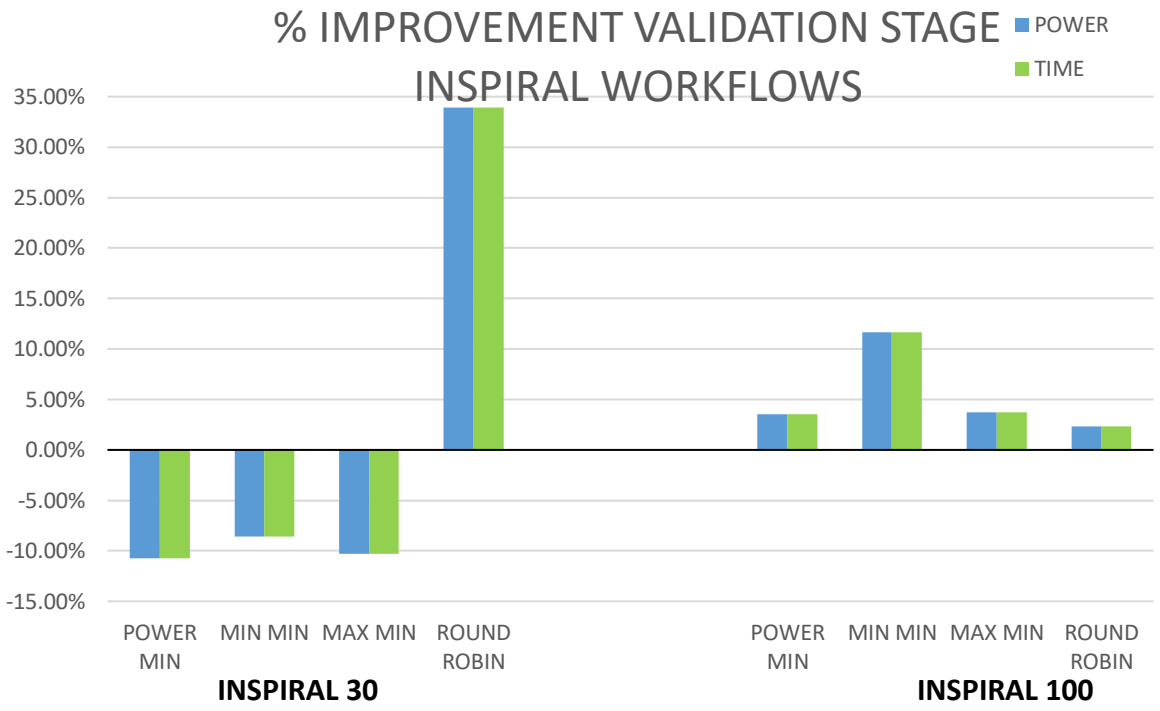

Figure 51. % IMPROVEMENT VALIDATION STAGE INSPIRAL WORKFLOWS.

9. CONCLUSIONS

This works presents an analysis of the interpretability the multi-objective scheduler based on FRBS system for Datacenter in cloud computing, whose main objective is to reduce power and time consumption and to assess the interpretability of the knowledge base obtained.

In addition, an interpretability study has been carried out. Understanding as interpretability the possibility of understanding the behaviour of a system by regarding its rule base. The main objective of interpretability is that the human can understand the system and interpret the result of it, but as the complexity increases, our ability to understand the behaviour decreases.

First, many works and authors have made a great effort to establish bases to construct interpretable fuzzy systems. That is why; this work has focused on the hierarchical fuzzy system methodology implemented, which takes into account only complexity (Q1 and Q2)[47].

An analysis of the different taxonomies proposed by other authors has been carried out to calculate the interpretability. As a Guaje that takes into account both the semantics and the complexity of the fuzzy system or the one used in this work focusing only on complexity. The idea of choosing Q1 and Q2 is because there are much used measures to quantify the complexity and are widely accepted. In addition, These measures are easy to use, for example, if a model has few conditions, then it will have fewer rules and fewer variables.

The way to evaluate and characterize the interpretability has been carried out through the point of view of the description. This point of view offers a transparency of the structure of the system in terms of number of variables (inputs, outputs), fuzzy sets, membership functions, rules, among others.

As a complexity, we can affirm that the more compact the knowledge base, the simpler its understanding will be and that the lower the complexity of KB, the greater the interpretability of the system. In conclusion, the interpretability is one of the most valued and valuable features of fuzzy system, although most fuzzy systems are not interpretable by ‘per se’.

Today, interpretability is an open topic, since the understanding of the systems is a subjective task that depends largely on the experience, preferences and knowledge of the person performing the evaluation, although many authors are beginning to focus and obtain results according to this term. It must be taken into account that the construction of interpretable system is a matter of careful design and not easy to implement.

In addition, a genetic algorithm (Pittsburgh) has been used to obtain the best KB. In this way, the approach is to minimize the complexity, for example fixing the maximum rules of the system. Moreover, a genetic algorithm combined with a fuzzy rule set reduction process that obtains a compact knowledge base with a reduced number of rules [28].

The value of the KB obtained for this multi-objective experiment has also been analysed. The value obtained is not a very high value, but taking into account the experiments performed it is of the highest values. Although this numerical value indicates the index interpretability of the methodology used, but today the issue of interpretability is not yet resolved, because it is a subjective task.

Interpretability assessment has been analysed using the hierarchical fuzzy system, previously explained, although there are other several methodologies.

On the other hand, after the execution of the experiments and be analysed, with this work it can be ensured that there is a saving of power and time in the Datacenter, this being a secondary objective of the work. In spite of the low index of interpretability obtained, the fulfilment of the main objective of this TFM, the analysis of the interpretability, can be ensured.

The performance of the scheduler used in WFS-DVFS simulator is based on the fuzzy knowledge of the VMs allocated on the Datacenter's hosts to decide which VM will execute each task for a time and power optimization.

Although these experiments have been simulations and the values will not correspond exactly to a real environment, the results will be very close to the real values.

This is thanks to the DVFS technique, since most processors have it integrated, and in addition to the workflows used. These workflows simulate real behaviours and real traces that could occur in a real environment, following the established order of each task. In addition, thanks to the characteristics of the WFS-DVFS simulator used and the simulation environment.

As can be seen in figure 50 and 51, in this last experiment whose objective is to optimize both energy and time, it is found that it improves the values of these two parameters compared to the other scheduling algorithm.

This scheduler works best with high workload, which is why with both Montage 100 and Inspiral 100, it is where improvements are obtained compared to all strategies. The relevant results is with Inspiral 100, since it is a different input file than training file. This means that the results are good, for example, if the results are compared with Round Robin (the worst case) an improvement of 2.35% is obtained.

Also, add that despite presenting only these three experiments, throughout this work, numerous experiments have been carried out that have led to failure. These three experiments are the result of the best scenario obtained, especially for the multi-objective experiment, which was the main objective of this project.

These previous experiments have been simulated with other FRBS system with other variables. In addition to trying other FRBS system, the scenario has also been modified. That is, modifying the number of resources and characteristics of the Datacenter. Thus achieving a simpler host scenario and virtual machines, hosting a virtual machine on each host of the Datacenter. This means that in a scenario where the workload is higher and the Datacenter has more resources, this algorithm will work better assuming an increase in the improvement of time and power.

As the experiments do not have obtained KB with a high interpretability, as futures lines of this work, it could be to build a fuzzy systems (maintaining multi-objective optimization to save energy) with high interpretability. It could eliminate fuzzy system components such as fuzzy inputs, rules and fuzzy sets whose reduction does not affect the accuracy of the system.

10. REFERENCES

- [1] Peter Mell,Timothy Grance,"The NIST Definition of Cloud Computing". NIST. National Institute of Standards and Technology, September 2011, NIST Special Publication 800-145.

- [2] P. Hale, "Acceleration and time to fail," Quality and Reliability Engineering International, vol. 2, no. 4, 1986.

- [3] Pablo Pérez Fernández. Los centros de datos están utilizando el dos por ciento de la energía mundial. 22 November 2018. https://tecnonucleous.com/2018/11/21/centros-de-datos-consumo-2-por-ciento-energia-mundial/

- [4] Doshi-Velez, Been Kim, "Towards A Rigorous Science of Interpretable Machine Learning". arXiv: 1702.08608v2 [stat.ML] ,2 Mar 2017.

- [5] Alfredo Vellido, José D. Martín-Guerrero, Paulo J.G. Lisboa. "Making machine learning models interpretable". European Symposium on Artificial Neuronal Networks, Computational Intelligence and Machine Learning, April 2012, v2, 163-172.

- [6] Iván Tomás Cotes-Ruiz, Rocío P.Prado, Sebastián García-Galán, José Enrique Muñoz-Expósito, Nicolás Ruiz-Reyes, "Dynamic Voltage Frequency Scaling Simulator for real Workflows Energy-Aware Management in Green Cloud Computing" PLOS ONE 12 (1), 2017, e0169803. doi:10.1371/journal.pone.0169803.

- [7] ¿Qué es el cloud computing? [https://debitoor.es/glosario/definicion-cloud-computing

- [8] Hardware as a Service (HaaS) https://www.techopedia.com/definition/13965/hardware-as-a-service-haas

- [9] Tipos de informática en la nube. https://aws.amazon.com/es/types-of-cloud-computing/

- [10] Iván Tomás Cotes Ruiz. "Energy optimization in cloud computing system" Master Thesis, 2016. http://tauja.ujaen.es/bitstream/10953.1/5204/1/Documentation.pdf

- [11] "Cloud computing para PyMES(y II): Modalidades https://www.nexica.com/es/blog/cloud-computing-para-pymes-y-ii-modalidades

- [12] Magdalena L. (2015) Fuzzy Rule-Based Systems. In: Kacprzyk J., Pedrycz W. (eds) Springer Handbook of Computational Intelligence. Springer Handbooks. Springer, Berlin, Heidelberg

- [13] F.Herrera. Genetic Fuzzy systems: State of the art and new trends.

- [14] Jorge Casillas, Oscar Cordón, María José del Jesús, Francisco Herrera. "Genetic tuning of Fuzzy Rule Deep Structures preserving interpretability and its interaction with fuzzy rule set reduction".
  Ieee transactions on Fuzzy Systems, VOL 13. NO.1. February, 2005.

- [15] José Ángel Fernández Prieto. "Optimización evolutiva de los parámetros de control de un algoritmo genético".
  Tesis Doctoral, 2009.

- [16] DP Pancho, JM Alonso y L. Magdalena, Quest for trade-trade-trade-off con el apoyo de Fingrams en la herramienta de modelado difuso GUAJE , International Journal of Computational Intelligence Systems, 6 (sup1): 46-60, 2013 (DOI: 10.1080 /18756891.2013.818189).

- [17] JM Alonso y L. Magdalena, Generando sistemas basados en reglas difusas comprensibles y precisas en un entorno java , Lecture Notes in Artificial Intelligence - 9th International Workshop on Fuzzy Logic and Applications, LNAI6857, 212-219, 2011 (DOI: 10.1007 / 978- 3-642-23713-3_27).

- [18] JM Alonso y L. Magdalena, HILK ++: An interpretability-guided fuzzy modeling methodology for learning readable and comprehensible fuzzy rule-based, Soft Computing, 15 (10): 1959-1980, 2011 (DOI: 10.1007 / s00500-010-0628 -5).

- [19] Christoph Molnar. Interpretable Machine Learning: A Guide for Making Black Box Models Explainable. 17 December 2019.

- [20] Janga Reddy Manne. "Swarm Intelligence for Multi-Objective Optimization in Engineering Design"
  Encyclopedia of Information Science and Technology, Fourth Edition. (DOI: 10.4018/978-1-5225-2255-3.ch022)

- [21]Mehdi Khrosrow-Pour, D.B.A, "Encyclopedia of Information Science and Technology"
  Third Edition, Vol.10. July,2014. (DOI: 10.4018/978-1-4666-5888-2).

- [22] Chen, W., & Deelman, E. (2012, October). Workflowsim: A toolkit for simulating scientific workflows in distributed environments. In E-Science (e-Science), 2012 IEEE 8th International Conference on (pp. 1-8). IEEE.

- [23] Chapter 4. Creating Workflows. Pegasus, Workflow generator.
  https://pegasus.isi.edu/documentation/creating_workflows.php

- [24] Gurmeet Singh, Karan Vahi, Arun Ramakrishnan, Gaurang Mehta, et all. "Optimizing Workflow Data Footprint". Vol 16. 2017.

- [25] https://www.datos101.com/blog/que-es-un-datacenter/

- [26] José M. Alonso, Luis Magdalena, Serge Guillaume. “HILK: A new Methodology for Designing Highly Interpretable Linguistic Knowledge Bases Using the Fuzzy Logic Formalism”.
  INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 23, 761–794 (2008). (DOI 10.1002/int.20288).

- [27] José M. Alonso, Luis Magdalena. “GUAJE - A JAVA ENVIRONMENT FOR GENERATING UNDERSTANDABLE AND ACCURATE MODELS”. 2010.

- [28] M.J. Gacto, R. Alcalá, F.Herrera.” Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures”.
  Inf. Sci.. 181. 4340-4360, 2011. (DOI: 10.1016/j.ins.2011.02.021).

- [29] José M. Alonso, Luis Magdalena. “HILK++: an interpretability-guided fuzzy modelling methodology for learning readable and comprehensible fuzzy rule-based classifiers”.
  Soft Comput (2011) 15:1959–1980 (DOI: 10.1007/s00500-010-0628-5).

- [30] S. F. Smith. A Learning System Based on Genetic Adaptive Algorithms. PhD thesis, University of Pittsburgh, 1980.

- [31] Francisco Herrera. “Ten Lectures on Genetic Fuzzy Systems”
  Scch. Software competence center hagenberg.

- [32] K. De Jong. “Learning with genetic algorithms: An overview”.
  Mach Learn 3, 121–138 (1988). (DOI: 10.1007/BF00113894)

- [33] Rocío Prado, Sebastián Galán, J.E. Expósito, L. R. López, et all. “Processing Astronomical Image Mosaic Workflows With An Expert Broker In Cloud Computing. Image Processing & Communications”.
  Image Processing & Communication, vol. 19, no. 4, pp.5-20. 2014 (DOI:10.1515/ipc-2015-0020).

- [34] Fispro 3.5 .
  https://www.fispro.org/en/install/install-on-windows/

- [35] Graphviz
  https://www.graphviz.org/

- [36] Fernando Berzal. “Algoritmos Genéticos”
  https://elvex.ugr.es/decsai/computational-intelligence/slides/G2%20Genetic%20Algorithms.pdf

- [37] R.P Prado, J.M. Expósito, A.J Yuste. “Knowledge acquisition in fuzzyrule-based systems with particle-swarm optimization”.
  IEEE Transactions on Fuzzy Systems, 18(6), 2010, 1083-1097.

- [38] G. Singh, K. Vahi, A.Ramakrishman, G. Mehta, et all. “Optimizing workflow Data Footprint”.
  Hindawi Publishing Corporation, vol 15, pages 20. (DOI: 10.1155/2007/701609).

- [39] Montage. https://pegasus.isi.edu/application-showcase/montage/

- [40] Laura Grajales Monsalve. “ALGORITMO DE PLANIFICACIÓN ROUND ROBIN”. 18 April 2016.

- [41] ROUND ROBIN SCHEDULING. https://es.wikipedia.org/wiki/Planificaci%C3%B3n_Round-robin

- [42] Soheil Anousha, Mahmood Ahmadi. “An Improved Min-Min Task Scheduling Algorithm in Grid Computing”. BOOK: Design, Analysis and Tools for Integrated Circuits and System. 2003. Pages 103-113. (DOI: 10.1007/978-3-642-38027-3_11).

- [43] S. Rehman, N. Javaid, S. Rasheed, K. Hassan, et all. “Min-Min Scheduling Algorithm for Efficient Resource Distribution Using Cloud and Fog in Smart Buildings.” 2019. Advances on Broadband and Wireless Computing, Communication and Applications. BWCCA 2018. Lecture Notes on Data Engineering and Communications Technologies, vol 25. Springer, Cham.

- [44] Neha Sharma, Sanjay Tyagi, Swati Atri. “A Comparative Analysis of Min-Min and Max-Min Algorithms based on the Makespan Parameter”. International Journal of Advanced Research in Computer Science. April 2017. Vol 8. ISSN No. 0976-5697.

- [45] Eva María Oviedo. “10 ventajas del ‘cloud computing’”, Octubre 2015. https://empresas.blogthinkbig.com/10-ventajas-del-cloud-computing/

- [46] Chen, W., & Deelman, E. (2012, October). Workflowsim: A toolkit for simulating scientific workflows in distributed environments. In E-Science (e-Science), 2012 IEEE 8th International Conference on (pp. 1-8). IEEE.

- [47] J.M. Alonso, L. Magdalena, S. Guillaume. “A hierarchical fuzzy system for assessing interpretability of linguistic knowledge bases in classification problems”, January, 2006.

- [48] J.M. Alonso, L. Magdalena, G.G. Rodríguez, Looking for a good fuzzy system interpretability index: an experimental approach, International Journal of Approximate Reasoning 51 (2009) 115–134.

- [49] C. Mencar, A. Fanelli, Interpretability constraints for fuzzy information granulation, Information Sciences 178 (2008) 4585–4618.