Multi-Objective Optimization of Virtual Machine Migration among Cloud Data Centers

Francisco Javier Maldonado Carrascosa[1*†], Doraid Seddiki[1], Antonio Jiménez Sánchez[1], Sebastián García Galán[1†], Manuel Valverde Ibáñez[1], Adam Marchewka[2]

[1]Telecommunications Engineering Department, University of Jaen, Avenida de la Universidad, S/N, Linares, 23700, Jaen, Spain.
[2]Faculty of Telecommunications, University of Science and Technology of Bydgoszcz, Profesora Sylwestra Kaliskiego, 7, Bydgoszcz, 85-796, Poland.

*Corresponding author(s). E-mail(s): fjmaldon@ujaen.es;
Contributing authors: ds000025@red.ujaen.es; anjimene@ujaen.es; sgalan@ujaen.es; mvalver@ujaen.es; adam.marchewka@pbs.edu.pl;

[†]This work has been supported by the research project P18-RT-4046 and the NextGenerationEU recovery plan, funded by the Andalusian Government and the European Union, respectively.

Abstract

Workload migration among cloud data centers is currently an evolving task that requires substantial advancements. The incorporation of fuzzy systems holds potential for enhancing performance and efficiency within cloud computing. This study addresses a multi-objective problem wherein the goal is to maximize the interpretability and the percentage of renewable energy consumed by a fuzzy meta-scheduler system in cloud scenarios. To accomplish this objective, the present research proposes a novel approach utilizing a multi-objective Knowledge Acquisition with a Swarm Intelligence Approach algorithm. Additionally, it takes advantage of a framework built on CloudSim, which includes virtual machine migration capabilities based on an expert system. Furthermore, a hierarchical fuzzy system is employed to assess rule base interpretability, along with another multi-objective algorithm, named Non-dominated Sorting Genetic Algorithm II. The framework and hierarchical system are employed to perform various simulation results concerning renewable energy and interpretability, while the algorithms aim to enhance the system's performance and interpretability. Empirical results demonstrate that it is possible to improve the performance of cloud data centers while improving the interpretability of the corresponding fuzzy rule-based system. The proposed multi-objective algorithm shows comparable or superior performance to the genetic algorithm across diverse scenarios. The simulation results indicate that improvements in cloud data center performance can be achieved while enhancing system interpretability. The average improvement in the interpretability index ranges from 0.6% to 6%, with a corresponding increase in renewable energy utilization ranging from 5% to 6%.

Keywords: Explainable artificial intelligence, Cloud computing, Virtual machine migration, Energy sustainability, Multi-objective optimization.

1 Introduction

Over the course of time, Fuzzy Rule-Based Systems (FRBS) have emerged as significant tools in the field of Fuzzy Logic (FL) applications. FRBS utilize fuzzy sets and FL to represent knowledge pertaining to specific problems (Kacprzyk and Pedrycz 2015). These systems, also known as expert systems, incorporate expert knowledge into their Knowledge Bases (KBs) (Cordon et al 2001). The success of FRBS relies heavily on the quality of their acquired knowledge or KB (Seddiki et al 2022). However, it is important to note that these systems lack the inherent capacity for autonomous generation and improvement, and are therefore inadequate to address broader and more complex problem domains. Due to this fact, FRBS have required the adoption of automated approaches to knowledge generation. One such approach involves employing an optimization method, thus enabling the FRBS to become a self-learning system capable of generating rules to increase its KB (Prado et al 2010). There are several strategies to apply this learning system, such as Genetic Fuzzy Systems (GFS) (Prado et al 2011, Jin 2023) and Swarm Fuzzy Systems (SFS) (Prado et al 2010, Rawat et al 2023) . A GFS encloses a Fuzzy System (FS) enhanced or extended with a learning process based on evolutionary algorithms (Cordon et al 2001). Typically, this learning process implies the generation of new rules for the FRBS, often accomplished through the use of Genetic Algorithms (GAs) such as Pittsburgh (Smith 1980) or Michigan (Booker et al 1989). In the context of multi-objective optimization, the Non-dominated Sorting Genetic Algorithm II (NSGA-II) (Deb et al 2002) is frequently used. On the other hand, an SFS refers to an FS that integrates rule learning techniques inspired by the behavior of natural groups, where each individual or particle possesses certain attributes and collaborates to reach a solution (García-Galán et al 2014). Among the SFS learning techniques , the "Knowledge Acquisition with a Swarm-Intelligence Approach" (KASIA) algorithm (Prado et al 2010) deserves mention for its better performance compared to GFS in previous work. However, this automated approach introduces new challenges, as FSs alone lack interpretability, and the utilization of automated methods may lead to complex solutions that impede comprehension of the decision-making process, thereby reducing system interpretability.

Interpretability, as defined in the context of this paper, refers to the examination of comprehending the decisions made by systems and the design principles employed to ensure that these decisions are easily understandable or interpretable (Rudner and Toner 2021). Within the realm of FS, interpretability is considered a distinctive and highly valued capability of FRBS, as it enables the analysis of system behavior in a way that humans can readily understand (Alonso et al 2011, Alonso and Magdalena 2011). Furthermore, interpretability is considered an essential requirement in applications involving extensive interactions with humans (Alonso and Magdalena 2011). Nevertheless, understanding the details of FRBS behavior presents a challenging task that requires not only understanding the description of the system, but also an in-depth comprehension of the inference process. Moreover, when evaluating FRBSs, it is essential to take into account characteristics that go beyond interpretability , such as accuracy, since a system with a high degree of inaccuracy is considered ineffective. In particular, these two characteristics often have conflicting objectives, since achieving a high level of accuracy usually implies the inclusion of a considerable number of rules, while interpretability strives to minimize rule complexity in terms of number of rules (Alonso and Magdalena 2011). It should be noted that FRBS are used in various research fields, such as traffic flow modeling (Singh et al 2023) and scheduling (Prado et al 2009). Consequently, these systems can improve the scheduling process in cloud computing data centers. Presently, there is a significant increase in data generation, leading to a corresponding rise in the number of data centers. It is estimated that Cloud Data Centers (CDC) currently consume approximately 3% of the world's electricity, with projections indicating a potential increase to 4.5% in the coming years (Kirvan 2022). As a result, many of these centers are adopting renewable energy sources to cover part of their energy needs. This use of renewable energy allows Modular Data Centers (MDC) to dynamically distribute workloads among geographically dispersed nodes based on the availability of renewable energy, a practice commonly referred to as "follow the renewable" (Shuja et al 2016). Nonetheless, despite the potential advantages, workload migration remains an underdeveloped task that requires extensive research and development efforts.

In recent years, cloud computing has become indispensable across various fields due to its scalability, flexibility, and cost-effectiveness in providing computing resources and services, as noted in this document. However, the increasing demand for cloud services raises concerns about their environmental impact and sustainability. The substantial energy consumption of CDCs, primarily reliant on non-renewable sources, poses challenges in mitigating carbon emissions and reducing fossil fuel dependence (Zheng et al 2020). To tackle this, there's a focus on integrating renewable energies like solar and wind into cloud systems to significantly decrease their carbon footprint (Abbasi-khazaei and Rezvani 2022). This shift aligns with global efforts toward cleaner energy practices. Yet, balancing the improvement of renewable energy usage with maintaining interpretability in cloud systems, essential for transparency and efficiency in decision-making, remains a significant challenge (Lu et al 2021). Optimizing cloud systems involves intricate trade-offs between objectives such as energy efficiency, cost reduction, and meeting user demands. Methodologies and algorithms are crucial for simultaneously optimizing these objectives while ensuring clear decision-making processes. By enhancing cloud performance in renewable energy usage while upholding interpretability, a more sustainable and transparent cloud ecosystem can emerge. This advancement not only showcases providers' environmental responsibility but also enhances competitiveness in an increasingly environmentally conscious market. Improved interpretability empowers users and stakeholders to make informed decisions, understand system behavior, and evaluate environmental impacts (Zulueta et al 2016). In light of these concepts, multi-objective optimization algorithms present an opportunity to enhance cloud computing data center performance while enhancing interpretability. The current study leverages NSGA-II, MO-KASIA, a CloudSim-based framework for VM migrations, and Hierarchical Fuzzy System (HFS) for interpretability assessment (Alonso et al 2006). This framework integrates expert systems, dynamic workload migration techniques, and "follow the renewable" policies to facilitate VM migration to renewable energy-equipped data centers. Metrics like renewable energy consumption and interpretability index aid in ranking solutions into different Pareto fronts, refining rule bases within algorithm populations for improved performance. This research introduces a multi-objective approach and methodology to optimize cloud performance in renewable energy use while maintaining interpretability, aiming for greener and more sustainable cloud systems without sacrificing transparency. Its outcomes contribute to ecological sustainability and overall cloud computing efficiency, benefiting the environment, businesses, and society.

The subsequent sections of this paper are structured as follows: Section 2 presents an overview of previous research related to the topics addressed in this paper. Section 3 elucidates the theoretical framework and methodology employed in this research. It provides a detailed examination of the model used and analyzes its implications for the study's objectives. The Solution Methods section delves into the approaches and methodologies utilized to address the research and obtain meaningful results. Section 5, Results and Discussion, offers a comprehensive analysis of the findings obtained from the research. Finally, Section 6 offers concluding remarks summarizing the key findings of the study and outlines possible future avenues for research and development.

2 Literature Review

The continuous operation of CDCs entails significant energy consumption, and the number of CDCs is steadily increasing since prominent service providers such as Microsoft, Amazon, and Google employ CDCs (Shuja et al 2016) equipped with substantial storage and processing capabilities. Additionally, the seamless availability of services requires replication and transmission of information. Consequently, service providers employ innovative technologies to improve the energy efficiency of CDCs. One such technique involves strategically distribute MDCs that rely partially on renewable energy sources to meet their energy needs. This approach allows the use of workload migration techniques, facilitating the transfer and execution of workloads between data centers based on the availability of renewable energy (Seddiki et al 2022). However, there is still much work to be done in this area.

2.1 VM Migration

Virtualization and Virtual Machine (VM) migration play a key role in improving resource consumption efficiency. VM migration involves the transfer of VMs from one resource to another, with two distinct approaches: non-live migration and live migration (Chaudhary et al 2023, Imran et al 2022). The existing literature covers numerous migration techniques. For example, Singh et al (2023) presented a bio-inspired VM placement considering VM mapping aiming a sustainable cloud resource management. Similarly, Sansanwal and Jain (2022), announced an improved load balancing approach used to identify migratable resources and their destinations in a cloud environment. It should be noted that none of these techniques explicitly address green computing. In contrast, Ahmad et al (2023) introduced a VM migration approach that supports green computing and dynamically allocates resources in order to decrease carbon emissions.

A limitation of many VM migration techniques is their primary focus on reducing power consumption without fully considering other crucial factors such as quality of service (QoS) (Singh et al 2023). In this context, the work presented by Li et al (2022) emphasizes the efficient management of VM placement while respecting QoS constraints. Furthermore, Dynamic Voltage Frequency Scaling (DVFS) techniques, commonly employed to optimize energy consumption, are proposed by Javadpour et al (2023) for load balancing. These techniques incorporate DVFS and consolidation policies to maintain QoS and system performance while minimizing the need for migrations.

Recent studies in the field of VM migration have explored the integration of artificial intelligence (AI) techniques. For example, Belgacem et al (2023) used machine learning to reduce the number of migrations and energy consumption, thereby improving VM migration processes and selection. Mandal et al (2023) defined a framework that uses AI, including FL and reinforcement learning, to address high energy consumption in data centers and prevent Service Level Agreement (SLA) violations. Furthermore, a CloudSim-based framework that incorporates an expert system and "follow the renewable" policies for VM migration and cloud scheduling is proposed by Seddiki et al (2022).

2.2 Interpretability

Regarding interpretability, several important works in key areas where FRBSs have made substantial contributions are worth mentioning (Varshney and Torra 2023, Herrera and Lozano 1996, Gegov and Frank 1995, Jang 1991, Robles et al 2009). In (Mariotti et al 2023), carefully designed constraining neural additive models are used to generate Pareto-optimal solutions that preserve semantic interpretability. In (Razak et al 2023), an architecture is employed for HFS, adjusting parameters to improve accuracy and interpretability. Regarding Neuro-Fuzzy Systems (NFS), Bai et al (2024) combined these systems with fuzzy classifiers using convolutional neural networks to improve the interpretability of results. In the realm of Big Data, Aghaeipoor et al (2022) introduced an FRBS aimed at reducing the number of rules and rule metrics.

2.3 Multi-Objective Optimization

Finally, given the large number of challenges and factors involved in cloud computing, multi-objective optimizations have been increasingly employed in this domain. Cui et al (2023) formulated task placement as a multi-objective optimization problem with the goal of optimizing makespan and resource utilization. The results show that the hybrid fuzzy-hitchcock bird algorithm outperforms the other multi-objective approaches in terms of performance metrics. Similarly, Deepika and Dhanya (2023) employed a multi-objective Particle Swarm Optimization (PSO) algorithm for dynamic VM deployment, considering factors such as resource utilization and power consumption. Additionally, Martínez et al (2023) addressed the evaluation of multi-objective optimization algorithms in the context of real-world applications. Table 2 provides a comprehensive analysis of other multi-objective methods, offering a more detailed comparison with the proposed method.

Table 1 Literature state of the art review.

Work	Objectives	Method	Key Contribution	Results
Seddiki et al 2022	Optimize the use of renewable energy in cloud data centers	Knowledge Acquisition FRBS	Increase the MDCs sustainability by using follow the renewable VM migration policies	63-64% of renewable energy usage across 4 data centers
Singh et al 2023	VM placement optimizing resource utilization, energy consumption and carbon emission	Bio-inspired virtual machine placement FP-NSO	Combination of FPO and NSGA-II for VM placement	Significant reduction in PW, CE and improvement in RU
Sansanwal and Jain 2022	Load balancing and time optimization in cloud environment	Inquisitive Genetic Grey Wolf Optimization	Review and compare load balancing techniques with proposed one	Better load balancing and efficacy than others
Ahmad et al 2023	Use green computing resources to reduce carbon emissions	Qualitative method for collecting resources and data	Using the computer resource in an eco-friendly while decreasing the harmful environmental impact	The number of resources employed for green computing can be beneficial for lowering E-waste
Li et al 2022	QoS, migration cost and load balancing optimization	Multi-objective VM dynamic scheduling	VM scheduling in big cloud data centers	The proposed technique achieves better results than comparison ones
Javadpour et al 2023	Energy usage optimization through DVFS computing	Task Prioritization Algorithm	The task prioritization algorithm in DVFS cloud environment	Ensure resource efficiencies, improve response times and reduce energy consumption
Belgacem et al 2023	Improve VM migration performance in cloud computing	VM migration machine learning model	Solution-based supervised machine learning model to enhance VM migration in terms of processes/selection	Significant gains compared to other algorithms in terms of performance
Mandal et al 2023	Energy efficiency and SLA violation optimization	MECpVmS: SLA-aware energy-efficient VM selection policy	Effective trade-off between energy consumption and SLA violation	Overall improvement in energy efficiency and consumption and SLA violation
Mariotti et al 2023	Explainability and interpretability of black box models	Constrainable Neural Additive Models (CNAM)	Explaining CNAM models' interpretability	Balance between performance and interpretability in classification tasks
Razak et al 2023	Decompose fuzzy systems into HFS systems	Straight-forward method: knowledge extraction and translation	Decomposition of FLS into HFS	More interpretability since HFS are clearer and simpler
Bai et al 2024	Optimize interpretability and performance	Data-knowledge-driven IT2FNN	Introduce association rule mining techniques into the model learning	Proposed method could achieve more robust performance facing label noise data
Aghaeipoor et al 2022	Increase explainability and interpretability	Interpretable fuzzy classifier for Big Data (IFC-BD)	Rule learning enhancement for oftaining more interpretable rules	IFC-BD could achieve a straightforward yet accuracy fuzzy classifier
Cui et al 2023	Execution time, execution cost and load balancing improvement	Multi-Objective Task Scheduling Optimization based on Evolutionary Multi-Factor Algorithm	The proposed algorithm considering time, cost and load balancing	The proposed algorithm outperformed the comparison techniques in solving the model
Deepika and Dhanya 2023	Resource usage, power utilization and carbon footprint minimization	Multi-Objective PSO	Predictive framework for VM placement considering multi-resource usage	MO-PSO outperformed conventional baseline approaches
Martínez et al 2023	Optimize different objectives in real world applications	NSGA-II, SPEA2, GDE3, MOEA/D, VaEA, NSGA-III, SMS-EMOA, iSMS-EMOA, LIBEA	Comparative study of multi-objective evolutionary algorithms in real-world applications	Better performance in different terms for distinct algorithms

Table 2 State of the art of multi-objective approaches in Cloud environments.

Work	Objectives	Algorithm	Key Contribution	Results
Li et al 2022.	Maintain an appropriate level of QoS while minimizing the migration costs and load balancing in big cloud data centers.	QMOD VM scheduling: QoS-aware and multi-objective dynamic VM scheduling employing GAs.	Solve dynamic allocation problem of VMs on limited physical infrastructures in big cloud data centers.	The proposed method has low migration cost compared to other multi-objective approaches and simulations show good performance.
Cui et al 2023.	Decrease energy consumption while reducing network traffic and power consumption of the network components in edge-cloud data centers.	Seagull VM placement optimization algorithm.	Multi-objective Seagull algorithm for VM placement and energy consumption reduction.	The proposed methodology reduces the network traffic by 70% and electricity usage by 80% without compromising the QoS metrics.
Deepika and Dhanya 2023.	Reduce resource usage while minimizing power utilization and restricting carbon footprint in cloud data centers.	Multi-Input Random Forest Regressor (MO-RFR) mixed with Multi-Objective Particle Swarm Optimization (MO-PSO).	The proposed framework concurrently foresees the multi-resource usage of VMs.	The proposed model outperforms the conventional baseline approaches in terms of power utilization, resource wastage and carbon footprint optimization.
Martínez et al 2023.	Minimization of makespan, cost of performing tasks, energy consumption, and increase throughput in cloud computing data centers.	Hybrid between the Antlion Optimization (ALO) algorithm and the Grasshopper Optimization Algorithm (GOA).	Hybrid binary Greedy multi-objective ALO-GOA (HGALO-GOA) approach based in Chaos Theory for workflow scheduling.	The model results show that the proposed algorithm create a right balance between several criteria compared to other models.
Proposed.	Maximization of RB interpretability and renewable energy usage while minimizing power consumption in cloud data centers.	Multi-Objective Knowledge Acquisition with a Swarm Intelligence Approach (MO-KASIA).	A KASIA multi-objective model for interpretability and renewable energy usage using virtual machine migration between modular data centers.	The proposed model obtains better performance than NSGA-II in terms of interpretability and energy consumption.

2.4 Research Gaps

Despite significant advancements in renewable energy integration, there remains a noticeable research gap regarding the incorporation of a multi-objective framework for optimizing the percentage improvement in renewable energy adoption. Existing literature often overlooks the complexities of simultaneously optimizing multiple objectives, such as cost-effectiveness, environmental impact, and system reliability. Moreover, the interpretability of the models used in such optimization processes is frequently neglected, limiting the practical application of findings. By failing to address these aspects in the state-of-the-art literature, there is a missed opportunity to develop comprehensive solutions that balance conflicting objectives while providing actionable insights for decision-makers in the renewable energy sector. Therefore, future research efforts should focus on integrating multi-objective optimization techniques with interpretable models to effectively address this gap and contribute to the advancement of sustainable energy systems. In this context, the proposed approach MO-KASIA covers the gaps found in the existing literature, highlighting the renewable energy consumption and interpretability optimization.

3 Model Description and Assumptions

The objective of this section is to elucidate the fundamental concepts essential for understanding this work. It has been structured into three distinct parts: the initial part includes a exhaustive discussion on the employed simulator; the subsequent part delves into the use of the HFS to compute the RB interpretability index; and the final section explores multi-objective optimization techniques and the applied algorithms, providing a detailed description of the NSGA-II implementation and introducing the novel proposed algorithm, MO-KASIA.

3.1 Sustainable VM Migration Cloud Simulator

Deploying data centers requires substantial capital investment and optimizing load migration is a complex task. Hence, robust data center simulation tools are crucial for accurately model these infrastructures and services. CloudSim, designed specifically for this purpose, initially only supported VM migrations within a data center and not between data centers. In (Seddiki et al 2022), a new framework built on CloudSim was proposed, which introduces VM migration capabilities. This framework incorporates FRBS as core algorithm for the meta-scheduler within the CloudSim simulator. The FRBS guides decision-making processes to optimize renewable energy consumption by selecting the most suitable machine and performing VM migrations.

The FRBS used in the meta-scheduler is based on the Mamdani-FRBS paradigm, which uses real inputs, outputs, and linguistic rules, commonly known as Mamdani rules (Mamdani 1977). It adheres to the typical structure of Fuzzy Logic Systems (FLSs) (Cordon et al 2001). The system uses five input variables with three fuzzy sets of Gaussian shape each, and one output variable with five Gaussian-shaped fuzzy sets. The fuzzy sets for the input variables are classified as 'low, medium, and high,' while the output fuzzy sets are labeled 'very unsuitable, unsuitable, suitable, very suitable, and extremely suitable'. The system characteristics are summarized in Table 3 and the fuzzy sets for the inputs and output are visualized in Fig. 1, where the FRBS meta-scheduling system is depicted. Using Gaussian membership functions is relevant due to their smoothness, flexibility, and intuitive way of modeling uncertainty and gradual membership transitions.

The RB of the FRBS consists of seven rules. It was defined in a previous work (Seddiki et al 2022) and is derived from the knowledge and experience of the authors, ensuring a high degree of comprehensibility and interpretability. This RB acts as a "doping" individual or particle within the algorithm population. Table 4 illustrates that

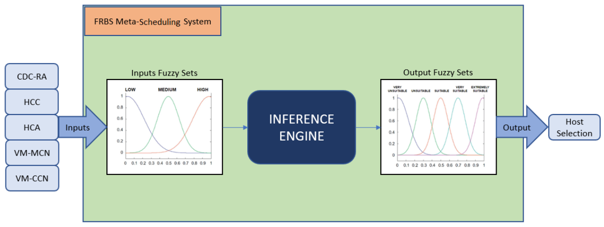


Fig. 1 FRBS meta-scheduler with gaussian fuzzy input and output membership functions.

rules 1, 2, 3, and 4 exclusively use two input variables, while the remaining rules use only one input variable.

3.2 Hierarchical Fuzzy System for interpretability

In contemporary times, FSs have attracted significant attention and emerged as a crucial area of study, mainly due to their ability to adapt and address problems in inaccurate or uncertain environments. This field of research has its roots Zadeh's seminal work (Zadeh 1965). FSs have found application in various domains, covering various fields. Examples include scheduling in Cloud computing simulators (Seddiki et al 2022), speech and music discrimination in audio (Exposito et al 2007), scheduling for Multi-Agent Systems for object tracking (Barbosa et al 2023), creation of depth maps (Bobyr et al 2023), and mark predictions for construction projects (Zaqout et al 2022).

Regardless of the specific application area, interpretability plays a vital role in FRBS. In recent years, interpretability has gained substantial importance in the field of AI, resulting in significant research efforts dedicated to its study. However, unlike accuracy, interpretability is difficult to define or measure (Lesot and Marsala 2021). It depends on multiple factors such as the structure of the model, the number of rules, the number of linguistic terms and the number of input variables (Gacto et al 2011). Some of these factors are closely related; for example, an increase in the number of input variables leads to an exponential rise in the required number of rules (Zhou et al 2009).

Regarding tools to measure interpretability, it is worth mentioning the HFS shown in Fig. 2, as

Table 3 Cloud system description characteristics.

Variable	Description
Cloud Data Center Renewable Availability (CDC-RA)	Renewable energy supplied to the CDC
Host Computational Capacity (HCC)	Maximum computational capacity in MIPS
Host Computational Availability (HCA)	Remaining computational capacity of the host in MIPS after holding other VMs
VM Maximum Computational Needs (VM-MCN)	Maximum needs of the VM in MIPS
VM Current Computational Needs (VM-CCN)	Remaining computational needs in MIPS for the VM

Table 4 Expert system's dopped rule base.

Rule	Dopped RB
1	**IF** HCC is high **AND** VM-MCN is low **THEN** output is extremely suitable
2	**IF** HCC is low **AND** VM-MCN is high **THEN** output is very unsuitable
3	**IF** HCA is high **AND** VM-CCN is low **THEN** output is extremely suitable
4	**IF** HCA is low **AND** VM-CCN is high **THEN** output is very unsuitable
5	**IF** CDC-RA is low **THEN** output is very unsuitable
6	**IF** CDC-RA is medium **THEN** output is suitable
7	**IF** CDC-RA is high **THEN** output is extremely suitable

detailed by (Alonso et al 2006). The HFS consists of four FRBS, each equipped with its own set of inputs, outputs, and KB. The RBs for each FRBS are presented in Tables 5, 6, 7 and 8, representing system's dimension, complexity, interpretability and interpretability index, respectively. The outcome of the HFS, called the Interpretability Index, provides a quantifiable measure to evaluate the degree of interpretability exhibited by a given RB.

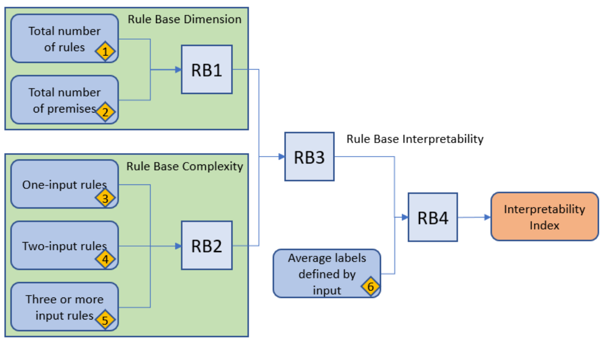


Fig. 2 Obtaining the interpretability index through an HFS.

Table 5 HFS dimension rule base (RB1).

Rule	Total number of rules	Total number of premises	RB1 output: RB Dimension
1	Low	Low	Low
2	Medium	Low	Medium
3	High	Low	High
4	Low	High	High
5	Medium	High	High
6	High	High	High

Table 6 HFS complexity rule base (RB2).

Rule	One input rules	Two input rules	Three or more input rules	RB2 output: RB Complexity
1	Low	Low	Low	Low
2	High	Low	Low	Low
3	Low	High	Low	Medium
4	High	High	Low	Low
5	Low	Low	High	High
6	High	Low	High	Medium
7	Low	High	High	High
8	High	High	High	High

4 Solution Methods

4.1 NSGA-II Approach

NSGA-II is a well-known algorithm widely used to solve multi-objective optimization problems. In this algorithm, individuals within the population "compete" with each other through evaluation and selection processes, following an elitist principle that favors the best individuals. The concepts of Pareto optimality and Crowding distance are used to maintain solution diversity. Genetic operators, such as crossover and mutation, are used in NSGA-II to create offspring and evolve the population. Fig. 3 graphically illustrates the NSGA-II procedure, and its detailed steps are presented in the Algorithm 1 below:

Table 7 HFS intrepretability rule base (RB3).

Rule	RB Dimension	RB Complexity	RB3 output: RB Interpretability
1	Low	Low	Very High
2	Medium	Low	High
3	High	Low	Low
4	Low	Medium	High
5	Medium	Medium	Medium
6	High	Medium	Very Low
7	Low	High	Low
8	Medium	High	Very Low
9	High	High	Very Low

Table 8 HFS interpretability index rule base (RB4).

Rule	RB Interpretability	Average number of labels defined by input	RB4 output: Interpretability Index
1	Very Low	Low	Very Low
2	Low	Low	Low
3	Medium	Low	Medium
4	High	Low	High
5	Very high	Low	Very high
6	Very Low	High	Very Low
7	Low	High	Very Low
8	Medium	High	Very Low
9	High	High	Low
10	Very high	High	Medium

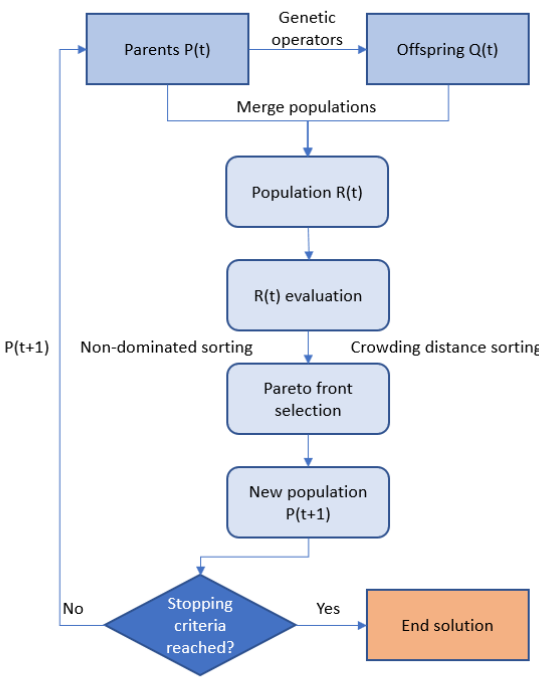


Fig. 3 NSGA-II general process.

Algorithm 1 NSGA-II Procedure.

1: Create a population $\boldsymbol{P}$ of size N.
2: Create an offspring $\boldsymbol{Q}$ of size N applying genetic operators to $\boldsymbol{P}$.
3: Merge the 2 groups into a population $\boldsymbol{R}$ of size $2N$.
4: Evaluate the population $\boldsymbol{R}$ and create a vector of solutions.
5: Classify the individuals of $\boldsymbol{R}$ on different Pareto fronts (Non-dominated sorting).
6: Calculate the crowding distance of all individuals in a non-dominated set $\boldsymbol{L}$ with the following algorithm:
7: Crowding-distance-assigment($\boldsymbol{L}$)
8: $l = |\boldsymbol{L}|$ % Take the number of solutions in $\boldsymbol{L}$
9: for each i,set $L[i]_{distance} = 0$ % Initialize distance of each individual
10: for each objective m
11: $\boldsymbol{L} = sort(\boldsymbol{L}, m)$ % Sort using each objective value
12: $L[1]_{distance} = L[l]_{distance} = inf$ % Set infinite distance to boundary points
13: for $i = 2$ to $(l-1)$ % Calculate crowding distance for the rest
14: $L[i]_{distance} = L[i]_{distance} + (L[i+1] \cdot m - L[i-1] \cdot m)/(f_m^{max} - f_m^{min})$
15: Perform the selection of individuals based on the Pareto front to which they belong and their crowding distance. The individuals that belongs to the lower Pareto front and have a higher crowding distance are chosen to create the population $\boldsymbol{P}$ of the next iteration, ***P(t+1)***.
16: Check whether the completion criteria were fulfilled. If not fulfilled, go back to step 2 considering that individuals already evaluated do not need to be evaluated again.

The NSGA-II is a multi-objective optimization algorithm. It starts by creating a population of solutions and then generates offspring by applying genetic operators. These solutions are merged into a single population. Then, the population is evaluated, and non-dominated sorting is performed to classify individuals into different Pareto fronts based on their dominance relationship. The crowding distance of individuals in each non-dominated set is calculated to maintain diversity. Finally, individuals are selected based on their Pareto front and crowding distance to form the next population. This process continues until completion criteria are met.

4.2 MO-KASIA Approach

In contrast, PSO has shown superior convergence speed and scalability compared to GAs (Shi and Eberhart 1999). Furthermore, these algorithms usually have simpler implementations since they do not require genetic operators, have fewer fixed parameters, and use memory within the algorithm's particles. Consequently, the use of a PSO-based algorithm for multi-objective optimization has advantages by reducing the complexity of these algorithms for such problem domains. Motivated by these considerations, an approach called

KASIA (Prado et al 2010) was introduced, offering capabilities to address multi-objective problems.

KASIA is a PSO-based algorithm that emulates the behavior of a swarm, with a set of RBs acting as particles. Each particle is characterized by a position matrix $\boldsymbol{P}$ and a velocity matrix $\boldsymbol{V}$. The position matrix, $\boldsymbol{P}$, represents the current position or solution of a particle, while the velocity matrix, $\boldsymbol{V}$, captures the inertia of the particles for its progress. Additionally, each particle stores its best solution or position reached during the execution of the algorithm (***Pbest***), and the global best position reached by any particle (***Gbest***) is also recorded. These values allow to calculate the velocity matrix for the next iteration using the following expression:

$$\begin{aligned} V(t+1) =&\omega V(t) + (d1r1)(Pbest(t) - P(t)) \\ &+ (d2r2)(Gbest(t) - P(t)) \end{aligned} \tag{1}$$

Here, t represents the iteration number, ***V(t+1)*** denotes the velocity matrix for the next iteration, ω is the iteration-dependent inertia weight, ***V(t)*** represents the current velocity value at iteration t, $d1$ and $d2$ are constant swarm coefficients, $r1$ and $r2$ are random factors within the $[0, 1]$ interval, and ***P(t)***, ***Pbest(t)***, and ***Gbest(t)*** denote the current position matrix, best position of the particle, and the best position reached by any particle in the swarm, respectively. Once ***V(t+1)*** is calculated, the particle's position is updated as follows:

$$P(t+1) = P(t) + V(t+1) \tag{2}$$

The general process of KASIA is presented graphically in Fig. 4 as follows:

In its original state, KASIA was designed to optimize a single variable. Consequently, to address multi-objective problems, it was necessary to adapt the algorithm by incorporating the concept of Pareto optimization. The multi-objective KASIA (MO-KASIA) procedure is very similar to its predecessor, but in this case, different solution vectors are used for each objective function to be optimized. The Pareto method is then used to identify non-dominated solutions, categorizing each particle into various Pareto fronts.

In the multi-objective variant of KASIA, the ***Gbest*** (global best position) represents a set of values corresponding to the objective functions

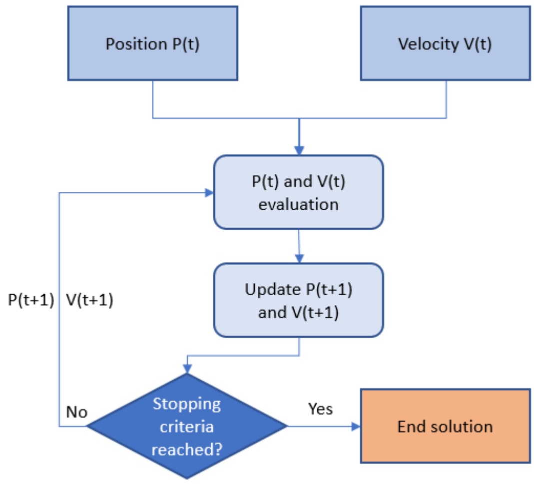


Fig. 4 KASIA general process.

that are optimized. To update ***Gbest***, the results of the individuals belonging to the best Pareto front are compared with the current ***Gbest*** values. If all variables demonstrate improvement, Gbest is updated accordingly. On the other hand, ***Pbest*** values (best positions of individual particles) are updated every time a particle improves the objective functions, regardless of the Pareto front to which it belongs. An overview of the MO-KASIA algorithm can be seen in Fig 5.

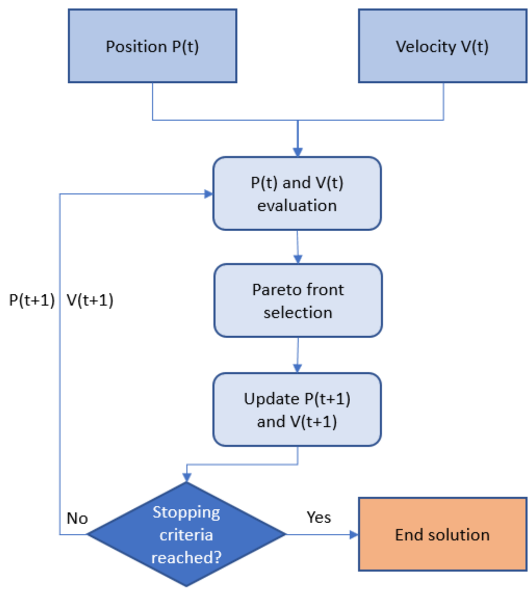


Fig. 5 MO-KASIA general process.

Algorithm 2 MO-KASIA Procedure.

Swarm initialization: $N_{particles}$, N_{rules}, N_{iter}, inertial weight ω, c_1 and c_2 factors.
Random setting of RB-Swarm position $\boldsymbol{P}$.
Random setting of velocity $\boldsymbol{V}$.
Apply $\boldsymbol{P}$ and $\boldsymbol{V}$ constraints.
Initialize ***Gbest*** and ***Pbest***.
while N_{iter} **do**
    while $N_{particles}$ **do**
        Update $\boldsymbol{P}$.
        Apply constraints to $\boldsymbol{P}$.
        Evaluate particle.
        Particles++.
    end while
    Classify each solution in different Pareto Fronts
    Select the best particles of the Pareto Fronts.
    Update ***Gbest***.
    while $N_{particles}$ **do**
        Update ***Pbest***.
        Update $\boldsymbol{V}$.
        Apply constraints to $\boldsymbol{V}$.
        Particles++.
    end while
    iter++.
end while
Return: ***Gbest***.

The provided Algorithm 2 outlines the multi-objective optimization approach called MO-KASIA for evaluating solutions with respect to interpretability and the percentage of renewable energy usage. It begins by initializing parameters and generating a swarm of particles with random positions and velocities while applying constraints. The algorithm then iteratively evaluates each particle's fitness, classifies solutions into Pareto Fronts, selects the best particles, and updates the global best solution. Particle velocities and positions bests are updated, and the process repeats for a specified number of iterations. Ultimately, it returns the global best solution, representing an optimal trade-off between interpretability and renewable energy usage, making it a versatile optimization tool for complex decision-making scenarios.

5 Results and Discussion

The evaluation of cloud computing and its performance in real-world deployments is crucial due to the increasing dependence on cloud services and the associated challenges. However, conducting large-scale experiments in real-world environments is often impractical or expensive. To overcome this limitation, the concept of using simulators that can mimic cloud computing scenarios and address societal problems has emerged.

In this context, the CloudSim simulator presented by Seddiki et al (2022) is used to evaluate and analyze cloud computing systems. The chosen CloudSim simulator incorporates advanced features that enhance its capabilities. Two notable features are virtual machine migration and renewable energy awareness. Virtual machine migration enables the dynamic movement of virtual machines across physical hosts, enabling load balancing, fault tolerance, and resource optimization. This feature allows to evaluate the impact of migration algorithms and policies on the performance and efficiency of cloud systems. Additionally, the simulator integrates green computing policies,reflecting the growing emphasis on sustainable and environmentally friendly computing. By incorporating renewable energy sources, such as solar power, into the simulation, the impact of green energy utilization on the overall system performance can be evaluated. This feature contributes to the evaluation of energy-efficient strategies and their potential benefits for cloud computing.

Moreover, the scientific workflows used in this simulator are not abstract workflows but rather real data obtained from Planet Lab, which is a global research network that allows access to a wide range of real-world distributed computing resources. Using current real-world scientific data from Planet Lab, the simulator enables the evaluation and analysis of cloud computing systems using current workload traces.

Using a CloudSim simulator with advanced features, virtual machine migration, renewable energy usage, and real-wolrd scientific data from Planet Lab, researchers and practitioners can gain insights into the performance, efficiency, and sustainability of cloud computing systems. This approach allows for the exploration of various scenarios, the evaluation of different algorithms and policies, and the development of optimization strategies to improve the overall effectiveness of cloud deployments.

5.1 Simulation Settings

The algorithms' performance evaluations has been carried in three different cloud scenarios, each representing a specific infrastructure size, ranging up to 800 hosts, which corresponds to the maximum capacity supported by the CloudSim simulator. In each scenario, the number of hosts and VMs is randomly distributed across four different CDCs. It should be noted that each CDC is characterized by a different percentage of available renewable energy, specifically 0%, 33%, 66%, and 100%, respectively, which dynamically changes every hour during the duration of this study. Moreover, each datacenter consists of 265/530/800 physical machines or hosts, 350/695/1,052 VMs and 1,500/5,000/10,000 cloudlets to represent the different cloud simulation scenarios. The specific details of these scenarios and their corresponding configurations are presented in Table 9 and 11 for reference and analysis.

Table 11 Cloud simulation scenarios.

Components	Scenario1	Scenario2	Scenario3
Hosts	265	530	800
VM	350	695	1052
Cloudlets	1500	5000	10000

By utilizing this specific cloud simulation configuration, researchers and practitioners can analyze and optimize various aspects of the cloud environment, can explore resource allocation strategies, evaluate workload management techniques, and assess system performance based on real-world values and characteristics. This configuration provides a realistic simulation environment for studying and improving cloud computing systems replicating the configuration set of this paper.

The algorithms used in the study were configured with consistent parameter settings across all scenarios, with the exception of the maximum number of iterations, which varied between 300 and 350 iterations. This approach aimed to achieve a normal distribution of samples and account for potential variations in the algorithm's performance. The NSGA-II algorithm featured a mutation probability that gradually decreased from 10% in the initial iteration to 0 in the final iteration, while the MO-KASIA algorithm used an inertia weight parameter (ω) that started at 1 and decreased exponentially to 0 over the iterations. For detailed information of the specific parameter values used in NSGA-II and MO-KASIA algorithms, see Table 10. This table provides a complete summary of the parameter settings for each algorithm throughout the iterations.

5.2 Simulation Results

5.2.1 Experimental results: Scenario 1

To provide the results show in Fig. 6, the NSGA-II and MO-KASIA algorithms were run within the simulation scenario 1 as described above. Fig. 6b illustrates the percentage of renewable energy consumed, while Fig. 6a displays the evolution of the interpretability index. A 95% confidence interval is considered in all experiments to take into account

Table 9 Cloud infrastructure configuration.

Host	Variable	Value	VMs	Variable	Value	Cloudlet	Variable	Value
	PEs	2		PEs	1		PEs	1
	RAM	4,096 MB		RAM	613/870/1,740 MB		Length	36,000,000
	BW	1,000,000 MB		BW	100,000 MB			
	MIPS	2,660		MIPS	500/1,000/2,000/2,500			
	Storage	1,000,000 MB		Size	2,500 MB			

Table 10 Parameter configuration for rule discovery in MO-KASIA and NSGA-II.

MO-KASIA		NSGA-II	
Variable	**Value**	**Variable**	**Value**
Number of rules	7	Number of rules	7
Number of antecedents	5	Number of antecedents	5
Number of consequents	1	Number of consequents	1
Number of particles	50	Number of individuals	50
Maximum iteration	[300, 350]	Maximum iteration	[300, 350]
Initial inertial weight (ω_0)	1	Crossover rate	0.8
Competitive factor ($c1$)	2	Initial mutation rate	0.1
Cooperative factor ($c2$)	2	Selection rate	0.8
		Replacement rate	0.8

statistical variations. After the analysis, it is evident that both algorithms present improvements in terms of the percentage of renewable energy consumed and the interpretability index. The red line in the figures represents the values obtained with the MO-KASIA algorithm throughout the iterations, with the orange area indicating the 95% confidence interval for this algorithm. Conversely, the blue line represents the values obtained with the NSGA-II algorithm, with the light blue area representing the confidence interval for NSGA-II.

Regarding the percentage of renewable energy consumed, it falls within the range between 63.8% and 64.8%. Regarding the interpretability index, the obtained values range between 0.6 and 0.65. It should be noted that for the renewable energy consumption, it is evident that the maximum values obtained differ between the two algorithms, and their confidence intervals do not overlap. This indicates that MO-KASIA achieves superior results regarding this variable. However, for the interpretability index, the confidence interval for NSGA-II is not represented for the last iterations due to the consistent value obtained across all experiments. Nevertheless, MO-KASIA obtains values slightly lower and very close to those obtained with the NSGA-II algorithm, resulting in overlapping confidence intervals. Based on these observations, it can be concluded that MO-KASIA outperforms NSGA-II in the first scenario. This conclusion follows from the fact that MO-KASIA achieves a higher percentage of renewable energy consumption while maintaining a comparable interpretability index.

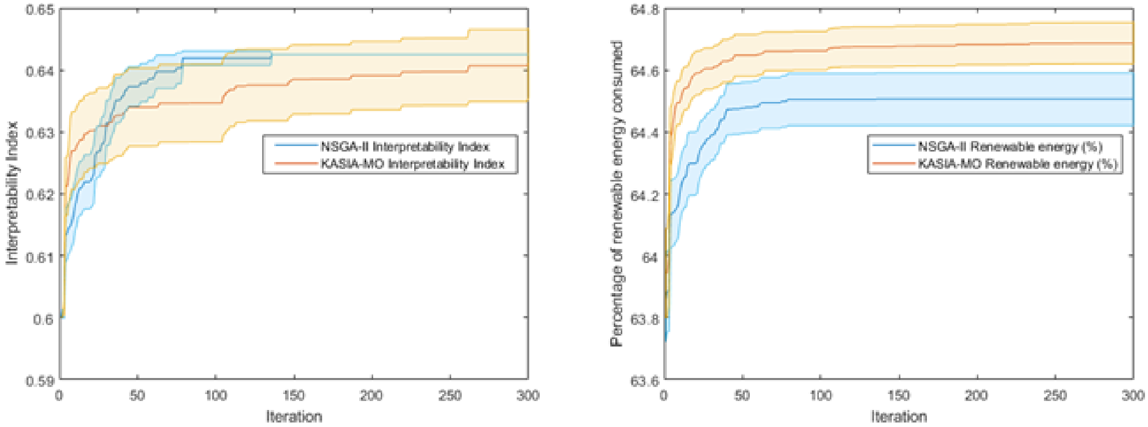
Table 12 presents the final RBs obtained by the NSGA-II algorithm and the MO-KASIA algorithm. Upon analysis, it is evident that both algorithms have successfully improved the interpretability of the RBs compared to the doping RB shown in Table 4. These RBs make use of all the antecedent variables, and the presence of the term "not" in a variable indicates a negative value for the antecedent. Greater interpretability is achieved by refining the rules and their composition.

For the NSGA-II algorithm, the best RB obtained from the 30 experiments reaches an interpretability index of 0.645. This improvement is mainly attributed to the inclusion of 7 rules, where 5 rules consist of a single antecedent variable, 1 rule contains 2 antecedent variables, and 1 rule comprises 3 antecedent variables. In the case of the best RB obtained by the MO-KASIA algorithm across the 30 experiments, the interpretability index reaches 0.672. This improvement is achieved through the creation of rules that consist of a single antecedent variable.

Furthermore, it is important to note that the total energy consumption for the dopant RB, the best RB obtained by NSGA-II, and the best RB obtained by MO-KASIA are 8.101 kWh, 8.1 kWh, and 8.102 kWh, respectively. This indicates that the use of renewable energy has increased instead of consuming a greater amount of total energy. In summary, both NSGA-II and MO-KASIA algorithms have successfully enhanced the interpretability of the RBs when compared to the initial dopant RB. The inclusion of more concise and meaningful rules has led to improvements in the interpretability index while maintaining a focus on the utilization of renewable energy resources.

5.2.2 Experimental results: Scenario 2

In simulation scenario 2, the evaluation is conducted using the average results of 30 experiments and a 95% confidence interval, similar to the approach in the first scenario. Once again, both algorithms demonstrate improvements in the percentage of renewable energy consumed and the interpretability index. The analysis reveals that the percentage of renewable energy consumed is in the range of 64.4% to 65% for both the NSGA-II and MO-KASIA algorithms. Likewise, the interpretability index obtained ranges between 0.6 and 0.65. When examining the consumption of renewable energy, it is clear from Fig. 7 that the results achieved by MO-KASIA exceed those of NSGA-II. The confidence intervals of the two algorithms do not overlap, indicating the superiority of MO-KASIA in terms of this variable. Regarding the interpretability index, although the confidence intervals overlap, the results obtained with MO-KASIA are consistently superior than those obtained with NSGA-II. This observation further supports the idea that the performance of MO-KASIA is superior to that of NSGA-II in the second scenario. In summary, for simulation scenario 2, both NSGA-II and MO-KASIA algorithms show improvements in the percentage of



a b

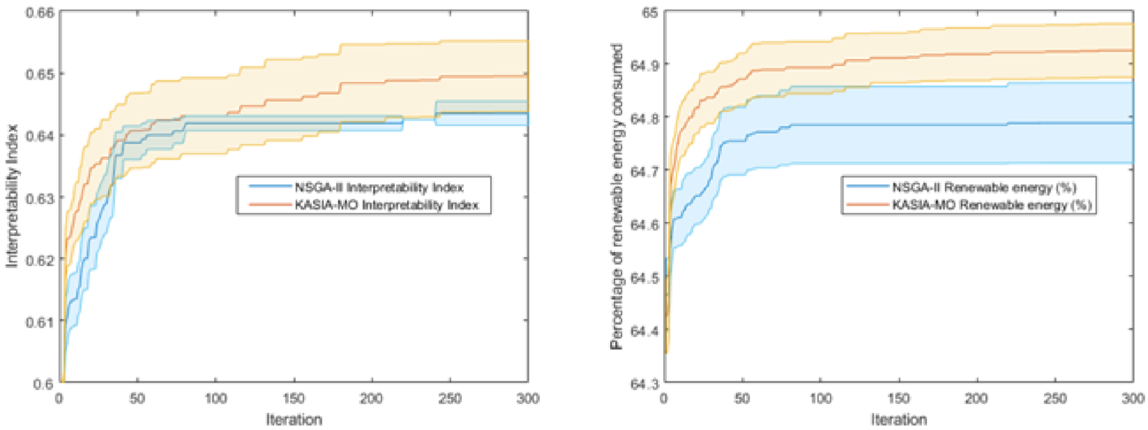
Fig. 6 Comparison of interpretability index (a) and renewable energy (b) values obtained by the algorithms in Scenario 1.

Table 12 Best NSGA-II and MO-KASIA RBs for Scenario 1.

Rule	NSGA-II RB	MO-KASIA RB
1	**IF** CDC-RA is high **AND** VM-CCN is medium **THEN** output is unsuitable	**IF** HCA is medium **THEN** output is extremely suitable
2	**IF** CDC-RA is high **THEN** output is unsuitable	**IF** HCC is not low **THEN** output is suitable
3	**IF** CDC-RA is low **THEN** output is very unsuitable	**IF** CDC-RA is not low **THEN** output is suitable
4	**IF** VM-MCN is low **THEN** output is extremely suitable	**IF** VM-MCN is low **THEN** output is very suitable
5	**IF** HCC is not high **AND** VM-MCN is medium **AND** VM-CCN is not medium **THEN** output is unsuitable	**IF** HCC is low **THEN** output is very unsuitable
6	**IF** HCA is low **THEN** output is very suitable	**IF** VM-CCN is high **THEN** output is very suitable
7	**IF** HCA is high **THEN** output is extremely suitable	**IF** VM-CCN is low **THEN** output is very unsuitable

renewable energy consumed and the interpretability index. However, MO-KASIA consistently outperforms NSGA-II in terms of both variables.

Table 13 displays the final RBs obtained by the NSGA-II algorithm and the MO-KASIA algorithm. It should be noted that, in this case, both RBs have an identical interpretability index of



a b

Fig. 7 Comparison of interpretability index (a) and renewable energy (b) values obtained by the algorithms in Scenario 2.

Table 13 Best NSGA-II and MO-KASIA RBs for Scenario 2.

Rule	NSGA-II RB	MO-KASIA RB
1	**IF** CDC-RA is not high **THEN** output is very unsuitable	**IF** CDC-RA is high **THEN** output is extremely suitable
2	**IF** CDC-RA is medium **THEN** output is unsuitable	**IF** CDC-RA is medium **THEN** output is suitable
3	**IF** CDC-RA is not medium **THEN** output is very unsuitable	**IF** CDC-RA is high **THEN** output is unsuitable
4	**IF** VM-MCN is low **THEN** output is extremely suitable	**IF** VM-MCN is low **THEN** output is extremely suitable
5	**IF** HCC is not medium **THEN** output is very unsuitable	**IF** HCC is low **THEN** output is suitable
6	**IF** HCA is low **THEN** output is very suitable	**IF** HCA is low **THEN** output is extremely suitable
7	**IF** VM-CCN is not low **THEN** output is suitable	**IF** VM-CCN is medium **THEN** output is very unsuitable

0.672, which is achieved by composing 7 rules consisting of a single antecedent each. The structure of the RBs is also similar, with variations limited to antecedent values. Additionally, in terms of energy consumption, the total energy consumption for the doping RB, the best RB obtained by NSGA-II, and the best RB obtained by MO-KASIA are 15.255 kWh, 15.260 kWh, and 15.246 kWh, respectively. It is important to note that MO-KASIA's RB demonstrates improvements in both total energy consumption and the utilization of renewable energy resources. On the other hand, the RB obtained by NSGA-II has increased the use of renewable energy, but has slightly increased the total energy consumption. In summary, the final RBs obtained by NSGA-II and MO-KASIA algorithms for this scenario have an identical interpretability index while increasing the use of renewable energy. Both RBs exhibit a similar structure with slight variations in the antecedent values.

5.2.3 Experimental results: Scenario 3

For this specific scenario, the NSGA-II and MO-KASIA algorithms were run for 350 iterations, as additional improvements were observed beyond the 300 iteration mark. In terms of performance, MO-KASIA outperforms NSGA-II in the percentage of renewable energy consumed. However, MO-KASIA obtains lower values for the interpretability index compared to NSGA-II, although the confidence intervals for interpretability overlap. The interpretability index achieved ranges between 0.6 and 0.65, while the percentage of renewable energy used is between 66.1% and 66.6%. To visually represent the results obtained for the algorithms in scenario 3, Fig. 8 is provided. This figure graphically illustrates the performance of the algorithms and provides further insights into the obtained results. In summary, in scenario 3, MO-KASIA demonstrates superior performance in terms of the percentage of renewable energy consumed. However, NSGA-II achieves slightly better results for the interpretability index, although the confidence intervals for interpretability overlap.

The best RBs obtained by the algorithms are presented in Table 14. In this case, the doping RB, the best RB obtained by NSGA-II, and the best RB obtained by MO-KASIA have very similar total energy consumption values: 22.6 kWh, 22.602 kWh, and 22.598 kWh, respectively. These results indicate that the use of renewable energy has increased instead of consuming a greater amount of total energy. Regarding interpretability, the best RB obtained by NSGA-II demonstrates higher interpretability. Each rule in this RB is composed of a single antecedent variable and its corresponding consequent. On the other hand, the best RB obtained by MO-KASIA consists of 5 rules with one antecedent variable, 1 rule with 2 antecedent variables and 1 rule with 3 antecedent variables. In summary, the best RBs obtained by NSGA-II and MO-KASIA algorithms for this scenario exhibit similar total energy consumption values, indicating an increased utilization of renewable energy. However, the interpretability differs between the two algorithms, with NSGA-II achieving higher interpretability by using rules composed of a single antecedent variable.

5.3 Time and Resources

When evaluating techniques in cloud computing, the computational cost plays a critical role. To assess the efficiency of MO-KASIA, it is important to reduce its computational cost. Therefore, the computational cost of MO-KASIA and NSGA-II systems was evaluated in terms of time. The elapsed time for the algorithm's computing delay in the proposed scenarios was tested on a computer with an Intel i7 processor and 8 cores, and the results are presented in Table 15. The MO-KASIA algorithm demonstrates superior

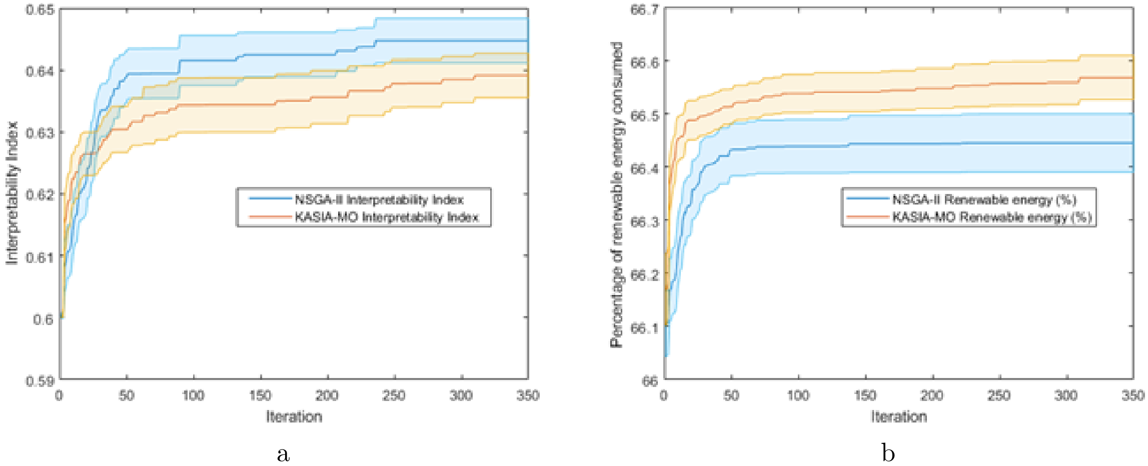


Fig. 8 Comparison of interpretability index (a) and renewable energy (b) values obtained by the algorithms in Scenario 3.

Table 14 Best NSGA-II and MO-KASIA RBs for Scenario 3.

Rule	NSGA-II RB	MO-KASIA RB
1	**IF** CDC-RA is high **THEN** output is suitable	**IF** CDC-RA is medium **THEN** output is extremely suitable
2	**IF** VM-CCN is medium **THEN** output is extremely suitable	**IF** CDC-RA is low **THEN** output is very suitable
3	**IF** HCA is low **THEN** output is extremely suitable	**IF** HCC is low **AND** HCA is low **AND** VM-CCN is low **THEN** output is unsuitable
4	**IF** HCC is high **THEN** output is extremely suitable	**IF** VM-MCN is low **THEN** output is extremely suitable
5	**IF** VM-MCN is high **THEN** output is very unsuitable	**IF** VM-MCN is medium **THEN** output is very suitable
6	**IF** HCA is low **THEN** output is extremely suitable	**IF** HCA is low **AND** VM-CCN is medium **THEN** output is extremely suitable
7	**IF** HCC is not medium **THEN** output is very unsuitable	**IF** HCA is medium **THEN** output is very unsuitable

Table 15 Overall execution time of simulations.

Workflow \Strategy	MO-KASIA	NSGA-II
Scenario 1 (s)	7.2654e+4	7.4751e+4
Scenario 2 (s)	7.4920e+5	7.6395e+5
Scenario 3 (s)	7.7567e+5	7.9125e+5

time computational cost performance compared to NSGA-II in the simulated scenarios. Furthermore, the number of RB evaluations serves as a reliable measure for evaluating the computational cost of the proposed systems. MO-KASIA requires 1,350,000 evaluations, while NSGA-II requires 1,080,000 evaluations. The computational cost formulation in terms of the number of evaluations for the three systems is as follows:

$$N_{MO-KASIA} = 50\ par. \cdot 300\ ite. \cdot 90\ sim. = 1,350,000\ RB\ evaluations \quad (3)$$

$$N_{NSGA-II} = 0.8 \cdot 50\ ind. \cdot 300\ ite. \cdot 90\ sim. = 1,080,000\ RB\ evaluations \quad (4)$$

The MO-KASIA system's lower computational cost, coupled with its superior performance regarding makespan and renewable energy consumed, suggests that it is the preferred system over NSGA-II.

5.4 Managerial Insights

For further improvements, the implementation of Kubernetes into the scheduling process would enhance efficiency. Adding managerial insights to the paper can be immensely beneficial for providing practical implications and guidance for decision-makers. This section can offer concise summaries of key findings, recommendations for managerial actions, and potential business impacts of implementing container migration through Kubernetes in cloud-based environments. It could address aspects such as cost-effectiveness, scalability, efficiency gains, and potential challenges in adoption.

Regarding the applicability and limitations of the proposed method in cloud-based environments like the Internet of Things (IoT), it's essential to consider several factors. Kubernetes-based container migration can efficiently scale to accommodate varying workloads, which is crucial in IoT environments with dynamic demands (Samir and Dagenborg 2023). By dynamically managing resources, Kubernetes can optimize resource utilization in IoT deployments, ensuring efficient operation. Additionally, Kubernetes' flexibility allows for deploying diverse workloads, accommodating the diverse requirements of IoT applications. However, there are limitations to consider. IoT devices often have limited computational resources, which may pose challenges for running Kubernetes clusters. Container migration in Kubernetes involves communication between nodes, which can introduce network overhead, potentially impacting IoT devices with limited bandwidth. Moreover, ensuring security in IoT environments is paramount, and Kubernetes must be configured and managed securely to prevent vulnerabilities that could compromise IoT devices and data.

Regarding additional layers or components to the method, it depends on specific requirements and objectives (Chrobak et al 2023). However, potential considerations could include integrating edge computing capabilities to enhance latency-sensitive IoT applications by processing data closer to the source. Implementing additional security measures, such as encryption and access control, can further strengthen the security posture of Kubernetes deployments in IoT environments. Additionally, developing resource management strategies tailored to IoT devices' constraints and characteristics can optimize performance and mitigate resource-related challenges.

In conclusion, while Kubernetes-based container migration offers promising benefits for cloud-based IoT environments, careful consideration of its applicability, limitations, and potential enhancements is essential for successful implementation.

6 Conclusions and Future Recommendations

This study introduces a novel multi-objective approach, the MO-KASIA algorithm, along with a methodology to optimize performance and interpretability in cloud computing systems. The research takes advantage of the CloudSim framework, specifically utilizing the "follow the renewable" strategy to generate simulation results related to renewable energy usage, as well as the HFS to calculate the interpretability index. The proposed methodology is applicable to a wide range of problems and algorithms, offering flexibility and adaptability. The simulation results obtained support the idea that improvements in cloud data center performance can be achieved while enhancing the interpretability of the system. In scenario 1, the average improvement in the interpretability index is 1%, accompanied by a 6% increase in the utilization of renewable energy. In the second scenario, there is a 0.6% improvement in the interpretability index and a 5% increase in renewable energy usage. Finally, in the last scenario, the interpretability index improves by 6%, coinciding with a 6% rise in the consumption of renewable energy.

In the future, the techniques employed in this study will be extended to optimize multiple objective functions and explore different ones altogether. Objective functions such as makespan, renewable energy consumption, and interpretability may be considered for inclusion. For example, a focus could be placed on reducing total energy consumption and makespan while maximizing the utilization of renewable energy and interpretability. Furthermore, Kubernetes container migration would be a proper update for cloud environment.

In summary, this work introduces the MO-KASIA algorithm, showcasing a novel approach for multi-objective optimization and a methodology for enhancing performance and interpretability in cloud computing systems. The study demonstrates improvements in interpretability and renewable energy utilization across various scenarios. Future directions include the expansion of these techniques to incorporate more objective functions and the exploration of additional optimization possibilities.

References

[1] Abbasi-khazaei T, Rezvani MH (2022) Energy-aware and carbon-efficient vm placement optimization in cloud datacenters using evolutionary computing methods. Soft Computing 26(18):9287–9322. https://doi.org/10.1007/s00500-022-07245-y

[2] Aghaeipoor F, Javidi MM, Fernández A (2022) IFC-BD: An Interpretable Fuzzy Classifier for Boosting Explainable Artificial Intelligence in Big Data. IEEE Transactions on Fuzzy Systems 30(3):830–840. https://doi.org/10.1109/TFUZZ.2021.3049911

[3] Ahmad S, Mishra S, Sharma V (2023) Green Computing for Sustainable Future Technologies and Its Applications. In: Contemporary Studies of Risks in Emerging Technology, Part A. Emerald Publishing Limited, p 241–256

[4] Alonso J, Magdalena L (2011) HILK++: an interpretability-guided fuzzy modeling methodology for learning readable and comprehensible fuzzy rule-based classifiers. Soft Computing 15:1959–1980. https://doi.org/10.1007/s00500-010-0628-5

[5] Alonso J, Guillaume S, Magdalena L (2006) A hierarchical fuzzy system for assessing interpretability of linguistic knowledge bases in classification problems. In: Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU), pp 348–355

[6] Alonso JM, Magdalena L (2011) Special issue on interpretable fuzzy systems. Information Sciences 181(20):4331–4339. https://doi.org/10.1016/j.ins.2011.07.001

[7] Alonso JM, Cordón O, Quirin A, et al (2011) Analyzing interpretability of fuzzy rule-based systems by means of fuzzy inference-grams. In: World Conference on Soft Computing (WConSC 2011)

[8] Bai K, Zhang W, Wen S, et al (2024) A data-knowledge-driven interval type-2 fuzzy neural network with interpretability and self-adaptive structure. Information Sciences 660:120133

[9] Barbosa G, Dantas M, Tiago de Oliveira Filho A, et al (2023) A fuzzy scheduler for MAS applied to object tracking. Engineering Applications of Artificial Intelligence 119:105796. https://doi.org/10.1016/j.engappai.2022.105796

[10] Belgacem A, Mahmoudi S, Ferrag MA (2023) A Machine Learning Model for Improving Virtual Machine Migration in Cloud Computing. Journal of Supercomputing 79(9):9486–9508. https://doi.org/10.1007/s11227-022-05031-z

[11] Bobyr M, Arkhipov A, Emelyanov S, et al (2023) A method for creating a depth map based on a three-level fuzzy model. Engineering Applications of Artificial Intelligence 117:105629. https://doi.org/10.1016/j.engappai.2022.105629

[12] Booker L, Goldberg D, Holland J (1989) Classifier systems and genetic algorithms. Artificial Intelligence 40(1):235–282. https://doi.org/10.1016/0004-3702(89)90050-7

[13] Chaudhary P, Kumar V, Joshi M, et al (2023) Live Virtual Machine Migration Techniques in Cloud Computing: A Review. In: 2023 IEEE World Conference on Applied Intelligence and Computing (AIC), IEEE, pp 446–451, https://doi.org/10.1109/AIC57670.2023.10263955

[14] Chrobak R, Galán SG, Expósito EM, et al (2023) Color tracking application using ai-based docker container scheduling in fog computing. In: International Conference on Computer Recognition Systems, Springer, pp 169–183, https://doi.org/10.1007/978-3-031-41630-9_17

[15] Cordon O, Herrera F, Hoffmann F, et al (2001) Genetic Fuzzy Systems. WORLD SCIENTIFIC, Spain, https://doi.org/10.1142/4177

[16] Cui Z, Zhao T, Wu L, et al (2023) Multi-Objective Cloud Task Scheduling optimization based on Evolutionary Multi-factor Algorithm. IEEE Transactions on Cloud Computing https://doi.org/10.1109/TCC.2023.3315014

[17] Deb K, Pratap A, Agarwal S, et al (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2):182–197. https://doi.org/10.1109/4235.996017

[18] Deepika T, Dhanya NM (2023) Multi-objective Prediction-Based Optimization of Power Consumption for Cloud Data Centers. Arabian Journal for Science and Engineering 48:1173–1191. https://doi.org/10.1007/s13369-022-06694-9

[19] Exposito JM, Galan SG, Reyes NR, et al (2007) Audio coding improvement using evolutionary speech/music discrimination. In: 2007 IEEE International Fuzzy Systems Conference, pp 1–6, https://doi.org/10.1109/FUZZY.2007.4295472

[20] Gacto M, Alcalá R, Herrera F (2011) Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures. Information Sciences 181(20):4340–4360. https://doi.org/10.1016/j.ins.2011.02.021, special Issue on Interpretable Fuzzy Systems

[21] García-Galán S, Prado RP, Expósito JEM (2014) Swarm Fuzzy Systems: Knowledge Acquisition in Fuzzy Systems and Its Applications in Grid Computing. IEEE Transactions on Knowledge and Data Engineering 26(7):1791–1804. https://doi.org/10.1109/TKDE.2013.118

[22] Gegov AE, Frank PM (1995) Decomposition of multivariable systems for distributed fuzzy control. Fuzzy Sets and Systems 73(3):329–340. https://doi.org/10.1016/0165-0114(94)00303-O

[23] Herrera F, Lozano M (1996) Adaptation of Genetic Algorithm. Parameters Based on Fuzzy Logic Controllers. In: Genetic Algorithms and Soft Computing, URL https://api.semanticscholar.org/CorpusID:18275513

[24] Imran M, Ibrahim M, Din MSU, et al (2022) Live virtual machine migration: A survey, research challenges, and future directions. Computers and Electrical Engineering 103:108297. https://doi.org/10.1016/j.compeleceng.2022.108297

[25] Jang JSR (1991) Fuzzy modeling using generalized neural networks and kalman filter algorithm. In: Proceedings of the Ninth National Conference on Artificial Intelligence - Volume 2. AAAI Press, p 762–767, https://doi.org/10.5555/1865756.1865795

[26] Javadpour A, Sangaiah AK, Pinto P, et al (2023) An Energy-optimized Embedded load balancing using DVFS computing in Cloud Data centers. Computer Communications 197:255–266. https://doi.org/10.1016/j.comcom.2022.10.019

[27] Jin Q (2023) Genetic algorithm and support vector machine application in english text classification for intelligent teaching. Soft Computing pp 1–12. https://doi.org/10.1007/s00500-023-09084-x

[28] Kacprzyk J, Pedrycz W (2015) Springer Handbook of Computational Intelligence. Springer, Berlin, https://doi.org/10.1007/978-3-662-43505-2

[29] Kirvan P (2022) How much energy do data centers consume? https://www.techtarget.com/searchdatacenter/tip/How-much-energy-do-data-centers-consume

[30] Lesot MJ, Marsala C (2021) Fuzzy Approaches for Soft Computing and Approximate Reasoning: Theories and Applications. Dedicated to Bernadette Bouchon-Meunier. Springer, https://doi.org/10.1007/978-3-030-54341-9

[31] Li J, Zhang R, Zheng Y (2022) QoS-aware and multi-objective virtual machine dynamic scheduling for big data centers in clouds. Soft Computing 26:10239–10252. https://doi.org/

10.1007/s00500-022-07327-x

[32] Lu Y, Murzakhanov I, Chatzivasileiadis S (2021) Neural network interpretability for forecasting of aggregated renewable generation. In: 2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), pp 282–288, https://doi.org/10.1109/SmartGridComm51999.2021.9631993

[33] Mamdani EH (1977) Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis. IEEE Transactions on Computers C-26(12):1182–1191. https://doi.org/10.1109/TC.1977.1674779

[34] Mandal R, Mondal MK, Banerjee S, et al (2023) MECpVmS: an SLA aware energy-efficient virtual machine selection policy for green cloud computing. Cluster Computing 26(1):651–665. https://doi.org/10.1007/s10586-022-03684-2

[35] Mariotti E, Alonso Moral JM, Gatt A (2023) Exploring the balance between interpretability and performance with carefully designed constrainable Neural Additive Models. Information Fusion 99:101882. https://doi.org/10.1016/j.inffus.2023.101882

[36] Martínez SZ, García-Nájera A, Menchaca-Méndez A (2023) Engineering applications of multi-objective evolutionary algorithms: A test suite of box-constrained real-world problems. Engineering Applications of Artificial Intelligence 123:106192. https://doi.org/10.1016/j.engappai.2023.106192

[37] Prado RP, Galán SG, Yuste AJ, et al (2009) Evolutionary fuzzy scheduler for grid computing. In: Cabestany J, Sandoval F, Prieto A, et al (eds) Bio-Inspired Systems: Computational and Ambient Intelligence. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 286–293, https://doi.org/10.1007/978-3-642-02478-8_36

[38] Prado RP, Galán SG, Expósito JEM, et al (2010) Knowledge Acquisition in Fuzzy-Rule-Based Systems With Particle-Swarm Optimization. IEEE Transactions on Fuzzy Systems 18(6):1083–1097. https://doi.org/10.1109/TFUZZ.2010.2062525

[39] Prado RP, Galán S, Expósito JE, et al (2011) Genetic Fuzzy Rule-Based Scheduling System for Grid Computing in Virtual Organizations. Soft Computing 15(7):1255–1271. https://doi.org/10.1007/s00500-010-0660-5

[40] Rawat P, Kumar P, Chauhan S (2023) Fuzzy logic and particle swarm optimization-based clustering protocol in wireless sensor network. Soft Computing 27(9):5177–5193. https://doi.org/10.1007/s00500-023-07833-6

[41] Razak TR, Kamis NH, Anuar NH, et al (2023) Decomposing Conventional Fuzzy Logic Systems to Hierarchical Fuzzy Systems. In: 2023 IEEE International Conference on Fuzzy Systems (FUZZ), IEEE, pp 1–7, https://doi.org/10.1109/FUZZ52849.2023.10309727

[42] Robles I, Alcalá R, Benítez JM, et al (2009) Evolutionary parallel and gradually distributed lateral tuning of fuzzy rule-based systems. Evolutionary Intelligence 2(5). https://doi.org/10.1007/s12065-009-0025-0

[43] Rudner TGJ, Toner H (2021) Key Concepts in AI Safety: Interpretability in Machine Learning. Center for Security and Emerging Technology https://doi.org/10.51593/20190042

[44] Samir A, Dagenborg H (2023) Adaptive controller to identify misconfigurations and optimize the performance of kubernetes clusters and iot edge devices. In: European Conference on Service-Oriented and Cloud Computing, Springer, pp 170–187, https://doi.org/10.1007/978-3-031-46235-1_11

[45] Sansanwal S, Jain N (2022) An Improved Approach for Load Balancing among Virtual Machines in Cloud Environment. Procedia Computer Science 215:556–566. https://doi.org/10.1016/j.procs.2022.12.058, 4th International Conference on Innovative Data Communication Technology and Application

[46] Seddiki D, García Galán S, Muñoz Expósito JE, et al (2022) Sustainable expert virtual machine migration in dynamic clouds. Computers and Electrical Engineering 102. https://doi.org/10.1016/j.compeleceng.2022.108257

[47] Shi Y, Eberhart R (1999) Empirical study of particle swarm optimization. In: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), pp 1945–1950 Vol. 3, https://doi.org/10.1109/CEC.1999.785511

[48] Shuja J, Gani A, Shamshirband S, et al (2016) Sustainable Cloud Data Centers: A survey of enabling techniques and technologies. Renewable and Sustainable Energy Reviews 62:195–214. https://doi.org/10.1016/j.rser.2016.04.034

[49] Singh AK, Swain SR, Saxena D, et al (2023) A Bio-Inspired Virtual Machine Placement Toward Sustainable Cloud Resource Management. IEEE Systems Journal 17(3):3894–3905. https://doi.org/10.1109/JSYST.2023.3248118

[50] Singh VP, Sharma K, Chakraborty D (2023) Solving capacitated vehicle routing problem with demands as fuzzy random variable. Soft Computing pp 1–21. https://doi.org/10.1007/s00500-023-08888-1

[51] Smith SF (1980) A Learning System Based on Genetic Adaptive Algorithms. PhD thesis, University of Pittsburgh, USA, https://doi.org/10.5555/909835

[52] Varshney A, Torra V (2023) Literature review of the recent trends and applications in various fuzzy rule-based systems. International Journal of Fuzzy Systems 25:2163–2186. https://doi.org/10.1007/s40815-023-01534-w

[53] Zadeh L (1965) Fuzzy sets. Information and Control 8(3):338–353. https://doi.org/10.1016/S0019-9958(65)90241-X

[54] Zaqout IS, Islam MS, Hadidi LA, et al (2022) Modeling bidding decisions and bid markup size for construction projects: A fuzzy approach. Engineering Applications of Artificial Intelligence 113:104982. https://doi.org/10.1016/j.engappai.2022.104982

[55] Zheng J, Chien AA, Suh S (2020) Mitigating Curtailment and Carbon Emissions through Load Migration between Data Centers. Joule 4(10):2208–2222. https://doi.org/10.1016/j.joule.2020.08.001

[56] Zhou SM, Garibaldi JM, John RI, et al (2009) On Constructing Parsimonious Type-2 Fuzzy Logic Systems via Influential Rule Selection. IEEE Transactions on Fuzzy Systems 17(3):654–667. https://doi.org/10.1109/TFUZZ.2008.928597

[57] Zulueta Y, Rodríguez D, Bello R, et al (2016) A linguistic fusion approach for heterogeneous Environmental Impact Significance Assessment. Applied Mathematical Modelling 40(2):1402–1417. https://doi.org/10.1016/j.apm.2015.07.016

Compliance and Ethical Standards

Funding

This work has been supported by the research project P18-RT-4046 and the NextGenerationEU recovery plan. Author S. Galan has received support from Andalusia Government and author F. Maldonado has recieved support from the European Union.

Conflict of Interest

Author Francisco Javier Maldonado Carrascosa declares that he has no conflict of interest. Author Doraid Seddiki declares that he has no conflict of interest. Author Antonio Jiménez Sánchez declares that he has no conflict of interest. Author Sebastián García Galán declares that he has no conflict of interest. Author Manuel Valverde Ibáñez declares that he has no conflict of interest. Author Adam Marchewka declares that he has no conflict of interest.

Ethical Approval

This article does not contain any studies with human participants or animals performed by any of the authors.

Statements and Declarations

Competing Interests

The authors have no relevant financial or non-financial interests to disclose.

Author Contributions

All authors contributed equally to this work.

Data Availability

The datasets generated during and/or analysed during the current study are available in the Multi-Objective GitHub repository, https://github.com/FranMaldonado7/Multi-Objective.