Applied Energy 372 (2024) 123798

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apenergy

Game theory-based virtual machine migration for energy sustainability in cloud data centers

Francisco Javier Maldonado-Carrascosa [a,*], Sebastián García-Galán [a], Manuel Valverde-Ibáñez [b], Tomasz Marciniak [c], Małgorzata Szczerska [d], Nicolás Ruiz-Reyes [a]

[a] Telecommunications Engineering Department, Linares Higher Polytechnic School, Jaén University, Avenida de la Universidad S/N, 23700 Linares, Jaén, Spain
[b] Electrical Engineering Department, Linares Higher Polytechnic School, Jaén University, Avenida de la Universidad S/N, 23700 Linares, Jaén, Spain
[c] Institute of Telecommunications and Computer Sciences, Bydgoszcz University of Science and Technology, Profesora Sylwestra Kaliskiego 7, 85-796 Bydgoszcz, Poland
[d] Metrology and Optoelectronics Department, Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, Gabriela Narutowicza 11/12, 80-233 Gdańsk, Poland

ARTICLE INFO

Keywords:
Cloud computing
Virtual machine migration
Scheduling
Follow the renewable
Game theory

ABSTRACT

As the demand for cloud computing services increases, optimizing resource allocation and energy consumption has become a key factor in achieving sustainability in cloud environments. This paper presents a novel approach to address these challenges through an optimized virtual machine (VM) migration strategy that employs a game-theoretic approach based on particle swarm optimization (PSO) (PSO-GTA). The proposed approach leverages the collaborative and competitive dynamics of Game Theory to minimize energy consumption while using renewable energy. In this context, the game is represented by the swarm, where each player, embodied by particles, carries both competitive and cooperative elements essential to shape the collective behavior of the swarm. PSO is integrated to refine migration decisions, improving global convergence and optimizing the allocation of VMs to hosts. Through extensive simulations and performance evaluations, the proposed approach demonstrates significant improvements in resource utilization and energy efficiency, promoting sustainability in cloud computing environments. This research contributes to the development of environmentally friendly cloud computing systems, thus ensuring the delivery of energy-efficient cloud computing. The results demonstrate that the proposed approach outperforms fuzzy and genetic methods in terms of renewable energy usage. The PSO-GTA algorithm consistently outperforms Q-Learning, Pittsburgh and KASIA across three simulation scenarios with varying cloudlet dynamics, showcasing its efficiency and adaptability, and yielding improvements ranging from 0.68% to 5.32% over baseline results in nine simulations.

1. Introduction

The rapid proliferation of cloud computing services in recent years has ushered in an era of unprecedented digital transformation and data-driven innovation. This remarkable evolution of information technology has triggered a paradigm shift, fundamentally altering the way businesses and institutions operate in the digital age. The appeal of cloud computing lies in its ability to offer scalable, on-demand access to computational resources, thereby erasing the traditional hardware boundaries and geographical limitations. Cloud technology has quickly become the backbone of contemporary organizations, providing them with the agility and flexibility needed to navigate the complexities of today's digital landscape. However, this increase in cloud adoption, while empowering, has revealed a critical challenge that cannot be ignored: optimizing resource allocation and power consumption in cloud environments [1]. As the reach of the cloud increases, so does its environmental footprint, making it crucial to strike a balance between growing demand for computing resources and ecological responsibility. The very scalability that makes the cloud so attractive also poses a dilemma, as uncontrolled growth can lead to inefficiencies, wasted resources and increased energy consumption. Therefore, achieving sustainability in cloud computing requires innovative strategies that not only address the growing appetite for computing power, but also embrace ecological awareness as an integral part of cloud operations. This convergence of technology and sustainability calls for inventive solutions that optimize resource allocation, reduce energy consumption

* Corresponding author.
E-mail addresses: fjmaldon@ujaen.es (F.J. Maldonado-Carrascosa), sgalan@ujaen.es (S. García-Galán), mvalver@ujaen.es (M. Valverde-Ibáñez), tomasz.marciniak@pbs.edu.pl (T. Marciniak), malszcze@pg.edu.pl (M. Szczerska), nicolas@ujaen.es (N. Ruiz-Reyes).

https://doi.org/10.1016/j.apenergy.2024.123798
Received 30 January 2024; Received in revised form 27 May 2024; Accepted 23 June 2024
Available online 3 July 2024

and mitigate the environmental impact of cloud computing, resulting in a more responsible and environmentally friendly digital era [2].

Consequently, sustainability has emerged as a key concern in this context, transcending mere industry trends and becoming a cornerstone of responsible technological progress. The exponential growth of cloud data centers (CDCs), which form the backbone of the increasingly digital world, has triggered the environmental footprint associated with their operation. The ever-expanding network of servers, cooling systems, and energy-intensive hardware components has highlighted the pressing need to mitigate the ecological impact of these massive data centers. The imperative to act is increasingly evident, driven both by environmental awareness and regulatory measures aimed at curbing emissions and energy consumption. The environmental repercussions of unsustainable cloud infrastructure are now palpable, extending beyond the confines of data centers to affect the planet at large. Finding innovative approaches that can reconcile the relentless demand for computing power with responsible energy use and resource allocation has never been more critical [3]. Balancing the equation between technological advancement and environmental responsibility has thus become a primary objective in the field of cloud computing. In this dynamic landscape, fostering sustainability in CDCs is not only a noble aspiration but an urgent necessity. It requires a paradigm shift in the way cloud computing solutions are conceived and deployed, going beyond improving efficiency to address fundamental issues of resource utilization, energy efficiency and carbon emissions. This is where technology, environmental stewardship, and sustainability converge to chart a new course for the digital future. Holistic rethinking of cloud infrastructure design, power supply and operational strategies is needed to steer cloud computing towards a more sustainable, responsible and environmentally friendly future. In this context, innovative solutions like the one discussed in this paper, which leverage cutting-edge technologies such as PSO and Game Theory [4], play a pivotal role in addressing these imperatives and shaping the sustainable cloud ecosystem.

This paper presents a solution to address these formidable challenges: a novel approach centered on optimized VM Migration. In the quest for sustainability, VM migration is a key tool that enables the dynamic reallocation of virtualized workloads to better utilize resources and optimize power consumption [5,6]. However, the power of VM migration is greatly enhanced with the integration of PSO-based Game Theory Approach (PSO-GTA). By harnessing the cooperative and competitive dynamics intrinsic to Game Theory [7,8], this approach seeks to maximize renewable energy usage while minimizing energy consumption. In this intricate interaction, the “game” takes the form of a swarm, while the “players” are embodied as particles. Each particle represents a potential VM migration decision and carries both competitive and cooperative attributes that depict the collective behavior of the swarm. These elements, guided by game theory principles, organize the VM migration scheduling. Furthermore, to refine migration decisions, PSO is seamlessly integrated since it augments the global convergence, ensuring that VMs are optimally allocated to hosts, thereby achieving a balance between computational efficiency and energy conservation [9].

This research substantiates the viability of the PSO-GTA through rigorous simulations and performance evaluations in a CloudSim simulation environment that enables VM migration between datacenters. The results are striking, showcasing a substantial enhancement in resource utilization and energy efficiency compared to the results obtained in [5], showcased as the baseline. This synergy between sustainability and cloud computing promises to significantly reduce the carbon footprint of CDCs while rationalizing operational costs and strengthening the uninterrupted delivery of cloud services [10]. In summary, this research heralds a promising frontier for green cloud computing systems. By combining the aforementioned concepts, it paves the way for a more responsible and efficient cloud ecosystem, underscoring the potential of the PSO-GTA in optimizing VM migrations for a sustainable cloud computing future.

In the realm of computational intelligence, genetic algorithms and fuzzy systems stand out as established methodologies that have significantly contributed to problem-solving in various domains. Genetic algorithms, inspired by the principles of natural selection and genetics, employ a population of potential solutions to evolve and adapt over successive generations, ultimately converging towards optimal or near-optimal solutions for complex problems [11,12]. On the other hand, fuzzy systems, rooted in fuzzy logic, provide a framework for handling uncertainty and imprecision by allowing variables to take on degrees of membership in linguistic sets [13]. These two paradigms have proven effective in addressing diverse challenges. To further bolster the findings, a comparative analysis is conducted against established methodologies such as the Pittsburgh genetic algorithm, Knowledge Acquisition with a Swarm Intelligence Approach (KASIA) and a Q-Learning algorithm [14]. An important feature of the first two algorithms is the application of knowledge acquisition, which gives value to the comparison. The inclusion of these comparisons aims to provide a comprehensive understanding of the PSO-GTA’s performance and its standing within the broader landscape of optimization techniques in cloud computing.

In practical terms, optimizing sustainability in cloud infrastructures involves a multi-faceted approach aimed at reducing environmental impact while maintaining operational efficiency. By harnessing a higher percentage of renewable energy sources, such as solar or wind, CDCs can significantly reduce their reliance on conventional energy sources, thereby reducing carbon emissions and the overall environmental footprint. This optimization goes beyond energy procurement and encompasses various aspects of data center management, such as resource allocation and workload scheduling. For example, the implementation of intelligent workload management strategies can dynamically allocate computing resources based on the availability of renewable energy, thereby maximizing the use of green energy and minimizing dependence on fossil fuels during peak demand periods [15,16]. In this context, this paper leveraged this idea and implemented it in a simulation scenario. Overall, the pursuit of sustainability optimization in cloud infrastructures is critical to mitigating the environmental footprint of data centers and fostering a greener and more resilient digital ecosystem.

The subsequent sections of this paper unfold a comprehensive exploration of various facets within the realm of cloud computing, employing game theory as a guiding framework. Initially, the background and related works are meticulously reviewed to establish a foundational understanding of the subject matter. Following this, the material and methods section is presented, intricately integrating game theory concepts into the analysis, alongside delineating computing algorithms chosen for comparison. Subsequently, the experimental results are elucidated, providing empirical insights into the efficacy of the proposed methodologies. Finally, culminating the discourse, the conclusion encapsulates the key findings, implications, and potential avenues for future research.

2. Background and related works

Cloud computing has emerged as a transformative force in the information technology landscape, revolutionizing the way organizations and individuals access and use computational resources, data storage and software services. At its core, cloud computing offers a revolutionary on-demand model, providing access to a shared set of configurable resources over the Internet [17]. This paradigm shift is based on a set of essential characteristics: on-demand self-service, broad network access, resource pooling, rapid elasticity, measured service, etc. Together, these characteristics define the dynamic and user-centric nature of cloud computing [18]. The cloud has seen rapid growth and transformation in recent years, becoming a central force in modern businesses and institutions. This fast rise can be attributed to various factors such as scalability and flexibility [19], cost-Efficiency [20], global reach of cloud providers [21], and more. Furthermore, cloud computing drives innovation by offering state-of-the-art technologies such as machine learning, data analytics and IoT solutions, empowering organizations to innovate and remain competitive in the ever-evolving digital landscape [22].

The central role of cloud computing in modern companies and institutions is undeniable. It supports critical functions such as data storage, processing and service delivery, making it an indispensable tool for organizations seeking to navigate the complexities of the digital age. However, alongside the immense benefits, cloud computing faces a number of challenges. These challenges are significant and multifaceted. Resource allocation is a complex task that involves efficiently allocating computing resources to meet fluctuating demands while minimizing waste. Inefficient allocation can result in increased operational costs and suboptimal performance [23]. Energy consumption is another pressing concern. The energy demands of CDCs, which house the hardware that powers cloud services, are increasing. This surge in energy consumption affects not only operational costs but also has far-reaching ecological implications [24]. Scalability, while a hallmark of cloud computing, can pose its own set of challenges. Balancing scalability with resource allocation and energy efficiency is a challenge. As CDCs expand to meet increasing service demands, managing this growth efficiently becomes crucial to ensure sustainability and operational efficiency. As these challenges continue to shape the cloud computing landscape, the need for innovative and greener solutions has become increasingly apparent. Achieving sustainability in cloud computing is a top priority, where technology and sustainability intersect to create a more responsible and resource-efficient digital future. Addressing these challenges is crucial to ensure the sustainable growth of cloud computing while minimizing its environmental impact [25].

Studies in the field of cloud computing and sustainability have been instrumental in shedding light on the challenges and opportunities associated with making cloud services more environmentally friendly [26]. This area of research has delved into the complexities of resource optimization and energy efficiency within data centers, emphasizing the importance of adopting green technologies and sustainable practices to reduce the environmental impact of cloud operations. By exploring innovative techniques and technologies that promote sustainability, researchers have sought to harmonize the ever-expanding cloud infrastructure with ecological responsibility. This involves not only minimizing energy consumption, but also incorporating renewable energy sources to create eco-friendly data centers [5].

The VM migration literature offers valuable insights into the advantages and complexities of this practice. Researchers have rigorously examined various approaches to VM migration, shedding light on the strengths and limitations of different strategies. Some methods focus on minimizing downtime and ensuring smooth migrations [27], while others prioritize resource allocation and load balancing [28]. Furthermore, the impact of VM migration on resource allocation and energy consumption has been a subject of intense research [5]. It has been demonstrated that well-scheduled VM migrations can lead to significant improvements in resource utilization and energy efficiency, a vital consideration for both cloud service providers and customers. The complex interplay between the benefits and challenges of VM migration forms the basis of innovative solutions to optimize cloud operations. Scheduling techniques within cloud computing have been explored in depth to improve resource utilization and energy efficiency. Numerous studies have meticulously compared various scheduling algorithms and strategies to evaluate their effectiveness in optimizing resource allocation and workload management [29]. The diversity of these approaches reflects the multifaceted nature of the cloud environment, where dynamic workloads, multi-tenancy and variable resource demands present unique scheduling challenges. Researchers have also identified the associated challenges and complexities in scheduling for the cloud, including issues related to task placement, load balancing and response time guarantees. This ongoing research underscores the need for innovative scheduling methods that can adapt to the changing cloud computing landscape.

Applications of game theory in cloud computing have demonstrated the relevance of the field in modeling interactions within cloud environments [37]. These innovative approaches have unveiled the potential for leveraging game theory to address resource allocation and energy optimization, taking advantage of the competitive and cooperative dynamics inherent to cloud operations [38,39]. By treating cloud elements as players in a strategic game, these approaches seek to strike a balance between the needs of VMs and the available resources within the cloud infrastructure. Each player, embodied by the particles of a swarm, carries both competitive and cooperative attributes that shape the collective behavior of the swarm, optimizing the allocation of VMs to hosts. Game theory presents an exciting avenue for managing the intricate dynamics of cloud computing, promoting efficiency and sustainability.

Table 1 offers a thorough comparison of works related to Adaptive Neuro-Fuzzy Inference System (ANFIS), presenting specific objectives, algorithms, key contributions, and results. Each row represents a distinct work, including proposed approaches. In the first row, the focus is on enhancing cloud security through VM migration, employing a Game Theoretic approach for optimal migration strategies and system security improvement [30]. The second row addresses dynamic job scheduling and resource efficiency using an Adaptive Chaotic Sparrow Search Algorithm Optimization and coalitional game, resulting in notable reductions in latency, processing time, workload imbalance, and energy consumption [31]. The third row handles computationally intensive tasks with limited resources, employing a two-stage computing offloading algorithm based on game theory, showcasing improvements in energy consumption [32]. Smys et al. (2020) optimize edge computing platforms using Cooperative and Non-Cooperative Gaming-based PSO, surpassing other algorithms in time-related variables [33]. The fifth row focuses on optimal edge server placement with a Multi-objective Whale Optimization Algorithm and game theory, leading to reductions in network latency and improved server load balance [34]. The sixth row optimizes energy-efficient sensor networks through Ant Colony Optimization with Game Theory Clustering, achieving reduced energy consumption and improved data transmission [35]. The seventh row addresses cloud task scheduling using a Red Fox Optimizer with fuzzy and game theory, showing improved fitness values compared to other algorithms [36]. Finally, the last row, corresponding to the proposed PSO-GTA approach, aims to optimize VM migration for cloud sustainability, emphasizing renewable energy.

While existing studies have made significant contributions to these fields, substantial gaps remain in the application of game theory to virtual machine migration using inter-data center scheduling. Specifically, existing research has not fully leveraged game theory to optimize the efficiency of migration processes, in the context of improving the utilization of energy and time resources, and enhancing energy sustainability in an intercloud environment. This research seeks to address these gaps and extend the current state of knowledge in a dynamic and evolving landscape by applying game theory to achieve more efficient and sustainable data center operations. The subsequent sections of this paper will delve deeper into the novel approach of integrating PSO-GTA for optimized VM migration. In the upcoming sections, this document meticulously outlines the material and methods employed in the research, elucidating the intricate concepts of game theory and Nash equilibrium. The proposed comparing algorithms are also expounded upon, shedding light on their theoretical foundations and operational mechanisms. Subsequently, the fourth section delves into the experimental results, meticulously showcasing the superior performance of the algorithm rooted in game theory.

Table 1
Comparison of Game Theory state of the art with proposed work.

Work	Objective	Used algorithm	Key contribution	Results
[30].	VM migration for cloud environment security.	Game Theoretic approach.	Attacker and defender game for VM migration to increase Cloud security.	Optimal migration strategies in order to ensure the security of the system.
[31].	Dynamic job scheduling and resource efficiency.	Adaptive chaotic sparrow search algorithm optimization and coalitional game.	Efficient utilization of resources through a multi-objective search algorithm and innovative coalitional game-theoretic approach.	The algorithm reduces the latency overhead in 9%, processing time in 14%, workload imbalance in 15% and energy consumption in 19%.
[32].	Handle computationally intensive tasks with limited resources.	Two-stage computing offloading algorithm based on game theory.	A multi-device partial offload energy consumption model for cloud–edge collaboration scenarios alongside a game theory-based method to achieve Nash equilibrium state of the Cloud.	The simulation results show that under the time-delay constraint, this method has an average improvement of 32% compared with the traditional method.
[33].	Time performance evaluation in edge computing.	Cooperative and Non-Cooperative Gaming-based PSO.	Utilization of game theory method for optimizing edge platform in terms of computation time and waiting time.	The proposed method achieved better results than GA, NGSA and GSA in terms of time variables.
[34].	Optimal edge server placement.	Multi-objective whale optimization algorithm and game theory.	Using WOA algorithm, neural networks and game theory to optimize the divided resource deployments.	The proposed method reduced the network average access latency by 33.5% and also improved the load balance on servers by 28.2%, compared to similar algorithms.
[35].	Energy efficient sensor networks.	Ant colony optimization with game theory clustering.	Exploring cluster-based networks to optimize power usage and maintain energy balance.	The proposed approach effectively reduces network sensor energy consumption, improves sensor node data transmission, minimizes end-to-end latency, reduces packet loss, and maximizes cluster formation.
[36].	Task scheduling in cloud environment.	Red fox optimizer with fuzzy and game theory.	Use the proposed method to find optimal mapping between task and resources.	The experimental results indicated that the proposed algorithm could improve the fitness value by 24.18%, 12.81%, 15.65%, 33.49%, and 14.73% compared with the BA, GWO, PSO, ALO, and RFO, respectively.
Proposed.	Percentage of renewable energy.	Game theory-based optimization.	VM migration optimization for cloud sustainability using game theory-based optimization.	The proposed algorithm consistently surpasses Pittsburgh and KASIA in nine simulation scenarios featuring diverse cloudlet dynamics, achieving improvements ranging from 0.68% to 5.32% over baseline results.

3. Material and methods

In the intersection of game theory and cloud computing lies a realm of fascinating challenges and opportunities. Game theory, a discipline originally developed to analyze strategic interactions among rational decision-makers, finds a new domain in the landscape of cloud computing. Here, the convergence of multiple entities, such as DCs, hosts, VMs and cloudlets, creates a dynamic environment ripe for strategic decision-making. At its core, the game problem formulation in this context revolves around optimizing resource allocation, task scheduling, and network management to maximize utility and efficiency. Cloud infrastructure serves as the battleground, where players seek to deploy strategies that yield favorable outcomes. Payoff functions and strategies become the linchpins of decision-making, determining the success or failure of various players in this complex ecosystem. Meanwhile, computational algorithms emerge as the engines driving strategic behavior and adaptation. From the classic Pittsburgh approach to the more recent KASIA, PSO-GTA, and Q-Learning methods, these algorithms harness the power of computation to navigate the intricate landscape of cloud-based interactions, paving the way for optimized resource utilization and enhanced performance. Fig. 1 shows the logical relations between the contents of this section.

3.1. Game theory and cloud computing

In the changing landscape of cloud computing systems, it is imperative to design robust strategies for effective resource and energy management. This involves formulating better selection, scheduling and allocation policies. By establishing a foundation that optimally balances these elements, the cloud infrastructure can operate more efficiently. To calibrate and refine system performance, it is essential to extract key data points. These encompass critical parameters such as runtime, renewable energy usage and computational and resource cost complexities. A thorough assessment of these factors lays the foundation for future optimization efforts. One avenue for improving system efficiency and resource allocation is to apply an optimization algorithm based in game theory. In this framework, datacenters, hosts, VMs, and cloudlets are treated as players, each with a set of possible actions. Strategies for these entities are meticulously identified, focusing on increasing the players' profits through iterative payoffs.

The iterative nature of the game is marked by continuous indication of payoff/performance metrics. These metrics, derived from the characteristics of datacenters, hosts, VMs, and cloudlets, provide crucial information on the evolving dynamics of the system. The calculation of a payoff matrix, based on the analysis of these metrics, serves as a fundamental tool for discerning strategic perspectives. The ultimate goal is to determine whether the game converges to an equilibrium state, be it a Nash equilibrium or a Pareto front (in this paper, Nash equilibrium is considered). If such equilibrium is achieved, the optimal strategy is realized. However, in cases where equilibrium remains elusive, the game moves smoothly to the improvement phase. This iterative process of refinement ensures a dynamic and adaptive approach to the changing challenges of cloud computing.

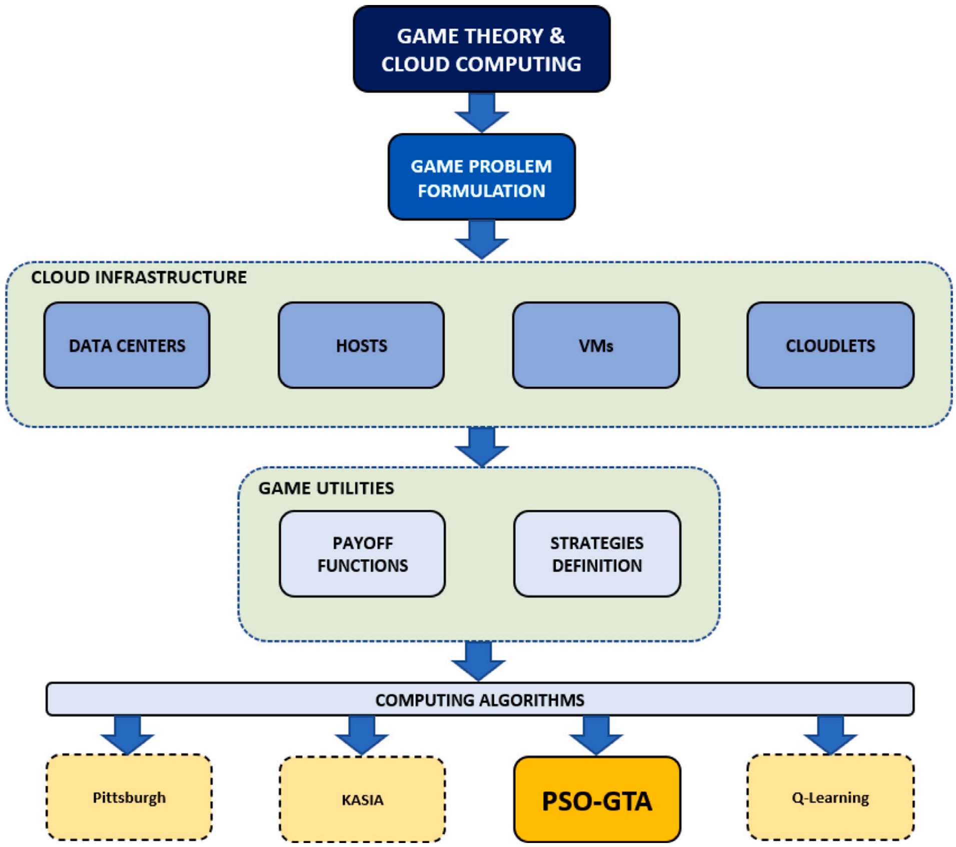

Fig. 1. Logical relationships flow chart.

Table 2
Parallelism between swarm optimizer and game.

Swarm optimizer	Game
Particle/Individual in the swarm	Players in the game
Search space	Strategies space
Objective functions	Payoff functions
Best position in the swarm	Stable strategy

Table 3
Current characteristics from the system.

Feature	Metric
Host	Computational Capacity (MIPS)
Host	Renewable Energy Efficiency (% Renewable energy)
Host	Computational Availability (MIPS)
VM	Computational Maximum Needs (MIPS)
VM	Computational Actual Needs (MIPS)

Before solving the equilibrium, it can be beneficial to leverage optimization algorithms, such as PSO. This proactive approach allows for a comprehensive exploration of the solution space, aiding in the identification of potential Nash equilibrium. Table 2 illustrates the analogous structure shared between PSO and the conventional game theory, showcasing their potential synergy. When using the optimization algorithm and reaching the stopping criteria, the post-processing of the results can be done, which should provide a set of strategies that represent the Nash Equilibrium.

The measured metrics encompass various facets of the system's current characteristics. These metrics are outlined in Table 3, which provides a comprehensive overview of the key features and corresponding metrics. The features include the host, with metrics such as Computational Capacity (MIPS), Renewable Energy Efficiency (% Renewable energy), and Computational Availability (MIPS). Additionally, the VM is assessed through metrics such as Computational Maximum Needs (MIPS) and Computational Actual Needs (MIPS). This structured approach to measurement ensures a thorough evaluation of the system's performance and resource utilization, offering valuable insights into its computational capacity, energy efficiency, and availability.

3.2. Game problem formulation

To delve into the intricacies of a game within the context of cloud computing, an illustrative example is presented, offering a simplified depiction to elucidate the mathematical concepts that underpin this dynamic framework. A scenario where multiple entities, or player contenders, interact within a cloud computing environment is considered. These entities include datacenters, hosts, VMs and cloudlets. The interaction between these players involves strategic decision-making processes aimed at optimizing their individual gains with the payoff.

Datacenters: the subscript i denotes the distinct datacenters participating in the game, ranging from DC_1 to DC_4. Each datacenter is a strategic player in the evolving landscape.

$$DC = DC_i;\ i = 1,\dots,4 \tag{1}$$

Hosts: hosts are characterized by a dual subscript, i and j, representing the datacenter index and the specific host within the datacenter, respectively. This multidimensional representation captures the diverse hosting configurations across the cloud infrastructure.

$$H = H_{i,j};\ i = 1,\dots,4;\ j = 1,\dots,m \tag{2}$$

Virtual Machines: VMs, denoted by VM_i, where i ranges from 1 to k, symbolize the virtualized computing instances. Each VM contributes to the strategic decisions within the game.

$$VM = VM_i;\ i = 1,\dots,k \tag{3}$$

Cloudlets: cloudlets, represented by C_i, are entities that generate computational tasks within the cloud environment. The subscript i signifies the distinct cloudlets participating in the strategic interactions.

$$C = C_i;\ i = 1, \dots, z \tag{4}$$

This breakdown establishes the foundational players in the game, laying the groundwork for a comprehensive understanding of the mathematical formulations and subsequent strategic dynamics. In the following sections, we delve into the mathematical concepts and strategies employed by these player contenders in pursuit of optimizing their respective objectives within the evolving cloud computing system. In the intricate interaction of a game in cloud computing, each player pursues distinct objectives, striving to maximize utility or minimize costs. The possible payoff functions (utilities) of the main participants in the game are explained below. The ultimate goal is to reduce resource use and maximize the use of renewable energy over total energy.

Datacenters: Datacenters aim to maximize revenue by strategically allocating resources (R_i) to Hosts and setting computational costs (Y_i), in terms of MIPS. The summation reflects the cumulative revenue generated by these strategic decisions.

$$Revenue(DC_i) = \sum (R_i \cdot Y_i) \tag{5}$$

Hosts: Hosts seek to maximize profit through the strategic allocation of resources (X_i) (MIPS and renewable energy usage) to VMs and setting computational costs (Y_i). The cost function (cost(X_i)) factors into the profit calculation.

$$Profit(H_i) = X_i \cdot Y_i - cost(X_i) \tag{6}$$

Virtual Machines: VMs strive to minimize execution time (T(VM_i)) by considering the ratio of resource requirements (RR_i) to the allocated resources (X_i) (MIPS and renewable energy usage). This reflects the efficiency of VM utilization.

$$T(VM_i) = \frac{RR_i}{X_i} \tag{7}$$

$$Cost(VM_i) = RR_i \cdot Y_i \tag{8}$$

To define a task C, the data D_n, total computation τ_n and maximum tolerable latency T_n required by the task is considered. In the field of cloud computing, task definition relies on sophisticated computational models that take into account key parameters of the data associated with the task, taking into account factors such as type, volume and dependencies. The total computation aspect assesses the algorithmic complexity, determining the number of operations involved in task execution and influencing resource allocation decisions. Additionally, these models factor in the critical element of maximum tolerable latency, setting constraints on task completion time for applications that require real-time processing. Together, these computational models form the basis for efficient task scheduling within the cloud infrastructure, guiding resource allocation, scheduling policies, and system optimization to satisfy various computational requirements.

$$C = \{D_n, \tau_n, T_n\} \tag{9}$$

In addition to the intricate computational models that govern task definition in cloud computing, the strategy set (S) further shapes the decision-making landscape within the system. This set, denoted as S, is carefully defined to encompass a range of strategic choices available to entities involved in the cloud environment. Each strategy within S represents a distinct course of action that actors, such as datacenters, hosts, VMs, and cloudlets, can employ to achieve their specific objectives.

The composition of S is derived from a meticulous analysis of the system dynamics, taking into account the diverse interactions and dependencies between entities. Strategies within S may include resource allocation policies, energy utilization approaches, scheduling algorithms, and cost optimization tactics tailored to the unique roles of each player. As the cloud computing landscape evolves, the strategic set S serves as a dynamic framework, adapting to new challenges and technological advances. The definition and evolution of S contribute significantly to the adaptability and efficiency of the cloud computing system, fostering a versatile environment where entities can make informed and strategic decisions to optimize their respective goals.

$$S = \{s_n / s_n \in \{s_n^l, s_n^m\}, n \in H, m \in DC\} \tag{10}$$

$$\begin{cases} \text{if } s_n^l = 1 \text{ then migrate VM to host} \\ \text{if } s_n^m = 1 \text{ then choose another host inside the DC} \end{cases} \tag{11}$$

Then, some computations are performed following the definition of the strategy set S and the intricate computational models governing task execution within the cloud computing system, a series of computations are undertaken. These operations are fundamental in translating strategic decisions into actionable outcomes, ensuring the efficient allocation of resources, and evaluating the overall performance of the system.

Computational time t_n^l: where f_n^l is the CPU computing power. measures the duration taken to execute computational tasks within the system. It encompasses the total time from task initiation to completion, providing insights into the speed and efficiency of task execution. Strategies within the set S are scrutinized for their impact on minimizing computational time, optimizing resource allocation, and enhancing overall system efficiency.

$$t_n^l = \frac{\tau_n}{f_n^l} \tag{12}$$

Energy consumption e_n^l: where σ_n^l is the energy consumption factor per CPU. is a critical metric that quantifies the amount of energy utilized during task execution. This metric is closely tied to the sustainability and environmental impact of the cloud infrastructure. The set S aims to incorporate strategies that minimize energy consumption, potentially leveraging renewable energy sources and energy-efficient resource allocation policies.

$$e_n^l = \tau_n \cdot \sigma_n^l \tag{13}$$

Cost of computational task E_n^l. reflects the financial expenditure associated with executing tasks within the cloud system. It considers factors such as resource costs, computational expenses, and any additional overhead incurred during task execution. Strategies within S are assessed based on their impact on cost reduction and financial efficiency.

$$E_n^l = \beta_n^t \cdot t_n^l + \beta_n^e \cdot e_n^l \tag{14}$$

$$\begin{cases} \beta_n^t, \beta_n^e \text{ are the weight factors of delay and energy} \\ 0 \leqslant \beta_n^t, \beta_n^e \leqslant 1 \\ \beta_n^t + \beta_n^e = 1 \end{cases} \tag{15}$$

Makespan μ. is a critical metric representing the total time taken to complete a set of tasks within the system. It is a holistic measure of system efficiency, encompassing the initiation, execution, and completion phases of tasks. The set S is designed to optimize makespan by incorporating strategies that enhance resource utilization, minimize idle time, and streamline task scheduling.:

$$\mu = \sum t_n^i;\ i = 1, \dots, z. \tag{16}$$

Finally, compute a utility from the system for optimization purposes (u_n^l), and obtain the optimal computational resource allocation/scheduling (f_n^{m*}). The utility in this paper is the percentage of renewable energy usage in the cloud environment. As computations unfold within the cloud computing environment, these calculations provide a quantitative framework for evaluating the success of implemented strategies. The iterative nature of the system allows for continuous refinement, adapting strategies within S to achieve optimal performance in terms of resource management and energy consumption.

3.3. Computing algorithms

Algorithm 1 Game Algorithm Pseudo-Code

Initialize DC, H, VM and C.
Optimization process
Output: Strategy S^*
while $S^*(t) \neq S^*(t-1)$ **do**
  $S^*(t) = S^*(t-1),\ set\ n = 1$
  while $n \leqslant N$ **do**
    Compute utility u_n^l.
    Compute optimal allocation/scheduling f_n^{m*}.
    Compute the best response $\Delta_n(t)$.
    $n++$.
    for $n = 1 : N$ **do**
      if n **then** wins the t-th iteration
        Update $s_n(t)$
      else
        $s_n(t) = s_n(t-1)$
      end if
    end for
  $t++$
  end while
end while
Output: Optimal computation resource allocation F^* and optimal strategy S^*.

The algorithm at play 1 in this cloud computing framework orchestrates a dynamic and strategic interaction among entities. It begins by defining a comprehensive set of strategies, denoted as S, which encapsulates diverse decision-making approaches available to the entities. The algorithm then performs a series of computations, implementing these strategies to optimize resource allocation and energy consumption. These computations involve translating strategic decisions into tangible outcomes, evaluating the system performance based on quantitative metrics. The iterative nature of the algorithm ensures its adaptability, allowing the system to refine and optimize strategies within S to meet evolving technological challenges and advances. Through this systematic approach, the algorithm seeks to create an efficient and responsive cloud computing environment that balances the diverse objectives of its constituent entities while meeting the quantitative parameters for optimal performance.

The Nash equilibrium, a fundamental concept in game theory, materializes when every one of the n players involved finds themselves in a strategic stance where altering their individual strategies provides no advantageous outcome. At this juncture, the system reaches a state of stability, similar to an equilibrium, where each player's chosen strategy is optimal given the strategies of the others. In this equilibrium, players have thoroughly considered the potential adjustments in their strategies, yet no unilateral deviation promises a superior result. The decision-making nexus of each player is saturated with a sense of strategic contentment, as any deviation would only lead to a less favorable position in the overall game. The Nash equilibrium encapsulates a fascinating interplay of rationality and mutual anticipation among the players. It serves as a beacon, indicating a convergence point where the complex dynamics of strategic decision-making harmonize, resulting in a state where no individual has the motivation to unilaterally alter their chosen course of action.

In the KASIA framework 2 proposed by Garcia et al. [13], fuzzy rule bases (RBs) are viewed as autonomous entities subject to assessment and refinement through Swarm Intelligence (SI) techniques. Within this context, the methodology leverages the PSO algorithm for the acquisition of fuzzy RBs. The system harnesses the collective dynamics observed in swarms to drive its operation. Initially, a population of RBs is generated, each represented as a collection of particles denoted by matrices P, where each row corresponds to an individual fuzzy rule. These particles encompass antecedents, consequent, and connectives following Mamdani coding. Through particle initialization, velocities (V) are assigned, enabling iterative rule modifications. Subsequently, the algorithm iterates to determine optimal positions for individual particles and the entire swarm collectively. This approach essentially evaluates the performance of RBs, adhering to specified constraints on algorithm variables, as detailed by Garcia et al. [13].

Algorithm 2 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
  Update ***Gbest***.
  while $N_{particles}$ **do**
    Update ***Pbest***.
    Update $\boldsymbol{V}$.
    Apply constraints to $\boldsymbol{V}$.
    Particles++.
  end while
  iter++.
end while
Return: ***Gbest***.

$$P_i = \begin{bmatrix} a^i_{1,1} & a^i_{1,2} & \cdots & a^i_{1,n} & b^i_1 & c^i_1 \\ a^i_{2,1} & a^i_{2,2} & \cdots & a^i_{2,n} & b^i_2 & c^i_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ a^i_{m,1} & a^i_{m,2} & \cdots & a^i_{m,n} & b^i_m & c^i_m \end{bmatrix} \tag{17}$$

$$V_i = \begin{bmatrix} v^i_{1,1} & v^i_{1,2} & \cdots & v^i_{1,n} & v^i_{1,n+1} & v^i_{1,n+2} \\ v^i_{2,1} & v^i_{2,2} & \cdots & v^i_{2,n} & v^i_{2,n+1} & v^i_{2,n+2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ v^i_{m,1} & v^i_{m,2} & \cdots & v^i_{m,n} & v^i_{m,n+1} & v^i_{m,n+2} \end{bmatrix} \tag{18}$$

$$V(t+1) = \omega \otimes V(t) \oplus c_1 r_1 \otimes (P^B(t) - P(t)) \oplus c_2 r_2 \otimes (G^B(t) - P(t)) \tag{19}$$

$$P(t+1) = P(t) \oplus V(t+1) \tag{20}$$

Within the particle P_i, denoting the RB matrix, the elements $ra_{m,n}$ signify the encoded fuzzy input value corresponding to the antecedent n of rule m. Simultaneously, $rc_{m,1}$ stands for the encoded fuzzy input value related to the consequent of rule m, while $ro_{m,1}$ represents the encoded fuzzy input value pertaining to the logical operator of rule m. The structure of RB rules mirrors that of Table 5. Each row in the particle matrix comprises seven elements, encapsulating the details of a fuzzy rule. The initial five elements encode values for considered input variables, namely CDC efficiency, host's computation capacity, computational availability, VMs computational maximum needs, and actual needs. The specific values of these entries are subject to variation based on the granularity of the universe of discourse. The constraints of the algorithms are presented below:

$$ra_{m,n} \in [-MF_a, +MF_a] \tag{21}$$

where MF_a is the number of antecedent membership functions.

The sixth element represents the encoded value for the rule consequent, and it will vary in the following interval:

$$rc_{m,n} \in [-MF_c, +MF_c] \tag{22}$$

where MF_c is the number of consequent membership functions.

The seventh and last element represent the logical operator for the rule, and its value is either 1 for AND or 2 for OR:

$$ro_{m,n} \in [1, 2] \tag{23}$$

Although the initial generated particles could belong to the allowed boundaries of the search space, the next matrix updates could eventually lead to an incoherent matrix whose elements reach values outside the permitted limits of the search space. To avoid these situations, the former values are constrained to ensure the coherence of the RB matrices during the execution of the algorithm. The constraints are described by the following equations:

$$ra_{m,n} = \begin{cases} -fa(MF_a) & if \quad ra_{m,n} < -fa(MF_a) \\ +fa(MF_a) & if \quad ra_{m,n} > +fa(MF_a) \end{cases} \tag{24}$$

$$rc_{m,n} = \begin{cases} -fc(MF_c) & if \quad rc_{m,n} < -fc(MF_c) \\ +fc(MF_c) & if \quad rc_{m,n} > +fc(MF_c) \end{cases} \tag{25}$$

$$ro_{m,n} = \begin{cases} 1 & if \quad ro_{m,n} < 1 \\ 2 & if \quad ro_{m,n} > 2 \end{cases} \tag{26}$$

where $fa(MF_a)$ represents the coding value of the membership function of the antecedent and $fc(MF_c)$ represents the coding value of the membership function of the consequent. Finally, any value outside the range for the value of the operator $ro_{m,n}$ will fit within the limits established by (23).

The Pittsburgh genetic strategy 3, as outlined by Smith et al. [40, 41], amalgamates evolutionary concepts, fuzzy reasoning, and a population-centric framework to tackle intricate challenges such as task scheduling in cloud computing. Here, the genetic system operates by treating entire RBs as chromosomes, where each genetic entity embodies a complete RB. Through the application of genetic operators, populations of RBs evolve over successive iterations. At the conclusion of the optimization process, the most optimal RB is singled out for integration into the fuzzy system. In every generation, all RBs within the population undergo assessment within the operational domain of the fuzzy system. This assessment is facilitated by the evaluation system, which gauges the effectiveness of each RB based on its fitness within the optimized environment. Subsequently, RBs are ranked according to their performance, with higher-ranked entities undergoing crossover to generate new RBs within the RB discovery system. Moreover, these top-performing RBs are susceptible to mutations. Concurrently, the least effective RBs in the generation are replaced by the newly evolved counterparts, with the replacement rate dictating the extent of this substitution. Through this iterative process, RBs demonstrating enhanced fitness criteria across generations are iteratively refined, culminating in the selection of the most optimal RB to act as the cornerstone of the fuzzy system in the final generation.

Algorithm 3 Pittsburgh algorithm.

```
Initialization: N_Population, N_Rules, N_Iter, Crossover_Rate,
  Mutation_Rate_Init, Selection_Rate, Replacement_Rate.
Random setting of RB Population P.
Initialize G^B.
while N_Iter do
   Update Mutation_Prob with Mutation_Prob = Mutation_Rate_Init * exp(-5t / N_iter)
   while N_Particles do
      Evaluate fitness (makespan).
      Particles++.
   end while
   Update G^B.
   while N_Particles·Replacement_Rate do
      Generate Q offspring:
      Apply crossover to P→Q.
      Apply mutation to Q→Q.
      Apply constraints to Q.
   end while
   Update part of P with Q.
   iter++.
end while
Return: G^B
```

Q-learning, a foundational algorithm in reinforcement learning, balances exploration and exploitation to learn optimal strategies without requiring a model of the environment. Based on the Bellman equation, it iteratively updates estimates of state–action values, converging towards the optimal policy. As a model-free and off-policy learning algorithm, Q-learning is robust to unknown dynamics and exploration strategies. However, its convergence and stability can be affected by factors like large state spaces and non-stationary environments. To address this, Q-learning often employs function approximation techniques, such as neural networks, enabling scalability to complex problems but introducing challenges like convergence guarantees and overfitting.

Algorithm 4 Q-Learning Algorithm

```
Initialization: Q-table with random values, α (learning rate), γ (discount
  factor), N_Episodes (number of episodes).
for episode = 1 to N_Episodes do
   Reset environment to initial state.
   s ← current state
   while episode not done do
      Choose action a using policy (e.g., ε-greedy).
      Take action a, observe reward r and new state s'.
      Q(s,a) ← Q(s,a) + α (r + γ max_a Q(s',a) − Q(s,a))
      s ← s'
   end while
end for
Return: Q-table
```

The Bellman equation in Q-learning describes how to update the Q-value of a state–action pair based on the immediate reward received, the maximum Q-value of the next state, and discounting future rewards. It is represented as:

$$Q(s,a) = (1-\alpha)\cdot Q(s,a) + \alpha \cdot \left(r + \gamma \cdot \max_{a'} Q(s',a')\right) \tag{27}$$

where $Q(s,a)$ is the Q-value of state s and action a, α is the learning rate, r is the immediate reward, γ is the discount factor, s' is the next state, and $\max_{a'} Q(s',a')$ represents the maximum Q-value for the next state s' over all possible actions a'.

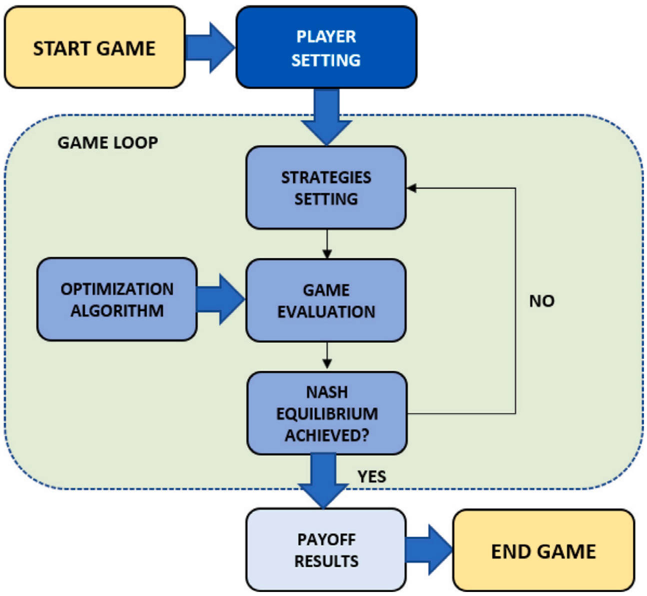

Fig. 2. Game procedure flow chart.

The Q-Learning algorithm begins by initializing a Q-table with random values, alongside setting parameters like the learning rate (α), discount factor (γ), and the number of episodes (N_{Episodes}). Through an episodic loop, the algorithm iterates over a specified number of episodes, resetting the environment to its initial state at the start of each episode. Within each episode, the agent interacts with the environment by selecting actions based on a policy, commonly an ϵ-greedy strategy. After taking an action, the agent observes the resulting reward and the new state. The Q-value for the current state–action pair is updated using the observed reward, the maximum Q-value for the next state, and the defined learning rate and discount factor. This process continues until the episode is completed. Finally, the trained Q-table is returned, representing the learned values for each state–action pair.

3.4. Particle swarm optimization based game theory approach

The game theory procedure depicted in Fig. 2 employs a dynamic and iterative approach to strategic decision-making. Beginning with the "Start Game" phase, subsequent stages orchestrate interactions among players. The "Set Players" and "Set Strategies" segments establish fundamental parameters, empowering participants to define roles and potential actions. Advancing to "Evaluate Game using Optimization Algorithm" introduces computational intricacy, systematically scrutinizing chosen strategies against predetermined criteria. At the point of "Nash Equilibrium Achieved?", equilibrium signifies stability in strategic dynamics, marking a pivotal point of balance. The "Change Strategies" segment embodies player adaptability, facilitating strategy refinement in the absence of Nash equilibrium, fostering learning and evolution. Following equilibrium attainment or strategy adjustments, the "Payoff Results" phase crystallizes outcomes, delineating rewards or losses linked to chosen strategies. This phase offers a concrete gauge of success or failure, underscoring the competitive and results-driven essence of strategic interactions. The "End Game" segment signifies the culmination of the process, denoting resolution in strategic dynamics and finalization of outcomes. This comprehensive perspective underscores the nuanced interplay of decision-making, optimization, and adaptation within the game theory framework.

In the intersection of optimization algorithms and game theory, this study introduces a novel approach that integrates PSO into the strategic evaluation of complex games. The proposed algorithm 5 conceptualizes particles as dynamic players within the game, utilizing positions represented by fuzzy matrices to capture the nuanced strategies inherent in strategic decision-making. The choice of PSO stems from its inherent capability to traverse and optimize diverse solution spaces, making it particularly apt for addressing the intricacies associated with strategic interactions. In the context of our metaphorical game, the primary objective is to unveil the Nash equilibrium, a state where no player can unilaterally deviate from their strategy to achieve a more favorable outcome. The algorithm orchestrates an iterative refinement process, guiding the fuzzy matrices towards optimal configurations that emulate the evolving strategies players might adopt. The dynamic interplay between PSO and game theory becomes apparent as particles adapt their positions within the PSO algorithm, mimicking the strategic maneuvers of players within the game. This research elucidates the algorithm's capacity to navigate through a multitude of potential strategies, converging towards solutions that encapsulate the essence of Nash equilibrium. By exploring this synergy between PSO and game theory, the study not only contributes to the understanding of strategic interactions but also presents a nuanced perspective on how optimization algorithms, particularly PSO, can serve as potent tools in unraveling the complexities inherent in strategic decision-making. The findings open avenues for further exploration in the domain of algorithmic game theory, shedding light on the potential applications and implications of this novel approach.

The PSO-GTA Algorithm integrates Particle Swarm Optimization (PSO) and Game Theory to optimize resource allocation and energy management in cloud computing environments. Beginning with parameter initialization, the algorithm employs particles to represent potential solutions and iteratively refines them through the PSO process. Fitness evaluation considers both renewable energy utility and the attainment of Nash equilibrium, ensuring a balance between energy efficiency and strategic stability. Particle positions are updated based on personal and global best solutions, guided by PSO equations. By leveraging PSO's exploration capabilities and strategic insights from game theory, the algorithm converges to optimal resource allocations that maximize efficiency while maintaining strategic equilibrium. This combined approach offers a robust and sustainable solution for cloud resource management, facilitating improved performance and energy savings.

Algorithm 5 Modified PSO-GTA Algorithm

```
Initialize DC, H, VM, C, renewable energy parameters, and PSO parameters.
Initialize particle positions S* and velocities.
Initialize global best position S_global.
PSO process
while stopping criterion not met do
    for each particle n do
        Compute utility u^l_n considering renewable energy.
        Evaluate fitness based on renewable energy utility and Nash equilibrium.
        if current position is better than personal best then
            Update personal best: S*_personal ← S*(t)
        end if
        if current position is better than global best then
            Update global best: S_global ← S*(t)
        end if
        Update particle velocity and position using PSO equations.
    end for
end while
Output: Optimal computation resource allocation F* and optimal strategy S* in Nash equilibrium.
```

The PSO-GTA Algorithm is a sophisticated optimization technique that integrates the strengths of both PSO and Game Theory to address the complexities of resource allocation in dynamic environments, with

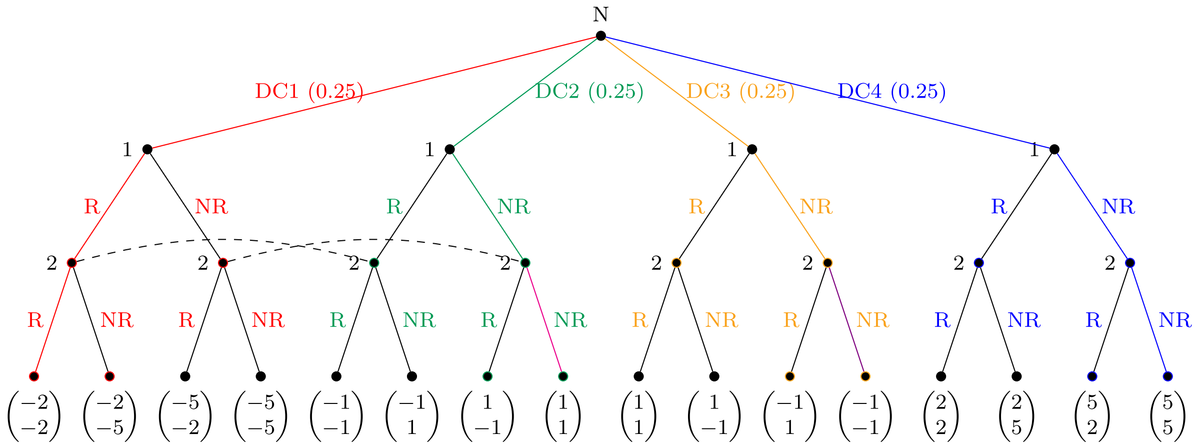

Fig. 3. Induced normal form tree.

a particular focus on scenarios incorporating renewable energy sources. It commences by initializing critical parameters essential for the optimization process, including demand, capacity, virtual machines, and pertinent parameters specific to both the PSO algorithm and renewable energy considerations. Subsequently, particle positions and velocities are established, along with the initialization of the global best position to guide the optimization process. Throughout the iterative PSO process, each particle evaluates its utility, incorporating the dynamic nature of renewable energy, and assesses its fitness with respect to both renewable energy utility and Nash equilibrium attainment. Notably, the algorithm continually refines its solutions by updating personal and global best positions whenever a superior solution is encountered. This dynamic adaptation, coupled with the iterative refinement of particle velocities and positions based on PSO equations, ultimately converges to yield optimal resource allocation (F^*) and strategic decisions (S^*) representative of Nash equilibrium. By seamlessly integrating optimization and game-theoretic principles, PSO-GTA offers a powerful tool for navigating the intricacies of decision-making in complex, evolving environments.

The Nash equilibrium is obtained in the following way. Let us assume that $\bar{s}_i$ is a strictly dominant strategy for player i ($\bar{s}_i \in S_i$) that is not played in some Nash equilibrium $N = (s_1, \dots, s_i', \dots, s_n)$ because $s_i' \neq \bar{s}_i$ is played there instead.

From the definition of a Nash equilibrium it is found that s_i' is a best response to s_{-i}. The definition of a Best response of the player i to the strategy profile s_{-i} states that $s_i' \in S_i$ is a mixed strategy such that:

$$\forall s_i \in S_i \, : \, u_i(s_i', s_{-i}) \geq u_i(s_i, s_{-i}) \tag{28}$$

In particular the following also holds for the strictly dominant strategy since $\bar{s}_i \in S_i$:

$$u_i(s_i', s_{-i}) \geq u_i(\bar{s}_i, s_{-i}) \tag{29}$$

Because $\bar{s}_i$ is strictly dominant the following holds:

$$\forall s_i \in S_i / \{\bar{s}_i\} \, : \, \forall s_{-i} \in S_{-i} \, : \, u_i(\bar{s}_i, s_{-i}) > u_i(s_i, s_{-i}) \tag{30}$$

And in particular since $\bar{s}_i \neq s_i' \Rightarrow s_i' \in S_i / \{\bar{s}_i\}$:

$$u_i(\bar{s}_i, s_{-i}) > u_i(s_i', s_{-i}) \tag{31}$$

If Eqs. (29) and (31) are combined:

$$u_i(\bar{s}_i, s_{-i}) > u_i(s_i', s_{-i}) \geq u_i(\bar{s}_i, s_{-i}) \tag{32}$$

This inequality clearly cannot hold under the given assumptions. Indeed, if instead $s_i' = \bar{s}_i$ Eq. (31) cannot be deducted (since $\bar{s}_i$ is exempted from S_i) and in that case it is trivial to see that $u_i(s_i', s_{-i}) \geq u_i(\bar{s}_i, s_{-i})$ does hold. By contradiction, it is proved that if a player has a strictly dominant strategy, $\bar{s}_i$ it will always be played in any Nash equilibrium.

Table 4
Example of game considering only the two first Data centers.

	$R_1 R$	$R_1 NR$	$NR_1 R$	$NR_1 NR$
$R_2 R$	(−1.5, −1.5)	(−3, −1.5)	(−1.5, −0.5)	**(−1.5, −2)**
$R_2 NR$	(−3, −1.5)	(−3, −3)	(−3, −0.5)	(−3, −2)
$NR_2 R$	(−0.5, −1.5)	(−3, −0.5)	(−0.5, −0.5)	−(0.5, −2)
$NR_2 NR$	(−2, −1.5)	(−2, −3)	(−2, −0.5)	(−2, −2)

In a game where the player i has N information sets indexed $n = 1, \dots, N$ and M_n possible actions at information set n, a good study is how many pure strategies does the player i have. From the definition of pure strategies for games, it is known that a pure strategy is composed of the N choices made at each information set. Since an information set n has M_n options, all options have to be multiplied by each other to find the amount of all possible pure strategies. This gives

$$\prod_{n=1}^{N} M_n$$

as solution.

The game tree presented in Fig. 3 represents a sequential decision-making process with different decision criteria and associated probabilities. Players make choices, leading to different outcomes with corresponding payoffs. The dashed and bend lines represent additional relationships and dependencies between certain decisions in a cooperative meaning.

Players 1 and 2 must decide whether to choose a host from a datacenter or another when processing tasks. They know that there is a 25% chance of scheduling. Each player's payoff is −5 if they do not invest in renewable energy and the energy demand is high, −2 if they invest in renewable energy and the demand is high, −1 if they invest in renewable energy and the demand is low, and 1 if they do not invest in renewable energy and the demand is low. Player 1 is informed about the energy demand before making a decision. Player 2, however, is not aware of the demand but can observe Player 1's choice before making their own decision. An example of this two-player game can be observed in Table 4, where the payoff is represented

4. Experimental results

4.1. Simulation framework

The parameters employed in the fuzzy RBs for knowledge acquisition are delineated in Table 5, which presents key features essential for describing the cloud system. These variables encompass crucial aspects such as Renewable Availability (CDC-RA), denoting the renewable energy supplied to the Cloud Data Center (CDC), Host Computational

Table 5
Features for cloud system description.

Variable	Description
Renewable Availability (CDC-RA)	Renewable energy supplied to the CDC
Host Computational Capacity (HCC)	Maximum computational capacity (CC) in MIPS
Host Computational Availability (HCA)	Remaining CC 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 6
KASIA parameter configuration.

Simulations	Particles	Iterations	Initial weight	Final weight	c_1	c_2
30	64	50	0.9	0.2	2	2

Capacity (HCC) representing the maximum computational capacity in MIPS, Host Computational Availability (HCA) indicating the remaining computational capacity of the host in MIPS after accommodating other VMs, VM Maximum Computational Needs (VM-MCN) representing the maximum computational requirements of the VM in MIPS, and VM Current Computational Needs (VM-CCN) characterizing the remaining computational needs in MIPS for the VM. This comprehensive set of parameters forms the foundation for the fuzzy RBs, facilitating a nuanced understanding and representation of the intricate dynamics within the cloud system for effective knowledge acquisition. For knowledge acquisition, fuzzy RBs utilizing Gaussian membership functions are employed, adhering to the structure outlined below in Eq. (33).

$$\mu_i^{X_m} = \frac{1}{\sqrt{2\pi\sigma_i^{2X_m}}} exp\left(\frac{-(z-\tau_i^{X_m})^2}{2\sigma_i^{2X_m}}\right), \; \{z \in \mathbb{R}, \; z \leq 1\} \tag{33}$$

where $\tau_i^{X_m}$ and $\sigma_i^{X_m}$ denote the mean and the standard deviation, respectively. Here, z denotes the independent variable describing the feature, and m corresponds to the current feature. The selection of Gaussian functions is driven by their advantageous property of having an extended area of influence, covering the entire universe of discourse. This feature allows the system to make valuable contributions across a wide range of system conditions. In a fuzzy system, each system variable value is associated with a membership degree related to a fuzzy set, assigned with normalized values ranging from 0 (complete exclusion of the variable) to 1 (full membership of the variable). Intermediate values indicate partial membership to the set, enabling precise and flexible reasoning within the fuzzy framework.

Tables 6–9 outline the parameter configurations for various optimization algorithms, offering a comprehensive view of their settings for experimentation and analysis. The first table presents details such as the number of simulations, particles, and iterations, along with initial and final weight values, and coefficients c_1 and c_2 used in the algorithm. Similarly, the next table delineates simulation count, particle count, iteration count, crossover rate, initial mutation rate, selection rate, and replacement rate specific to the Pittsburgh optimization method. Furthermore, the following table specifies simulation count, player count, iteration count, learning rate (α), discount factor (γ), and reward value (r) utilized in the Q-Learning algorithm. Lastly, PSO-GTA table succinctly lists simulation count, player count, and iteration count for the proposed algorithm. Together, these tables provide essential insights into the setup of each algorithm, facilitating comparative analysis and aiding researchers in understanding the impact of different parameter choices on algorithm performance.

4.2. Simulation scenarios

The simulation scenarios utilize an improved CloudSim simulator, as introduced in [5], facilitating VM migration across four datacenters, as illustrated in Fig. 4. These datacenters feature two types of hosts: HP ProLiant ML110 G4 (Xeon 3040, dual-core, 1.8 GHz, 4 GB RAM, and 1 Gbps of Bandwidth), and HP ProLiant ML110 G5 (Xeon 3075, dual-core, 2.6 GHz, 4 GB RAM, and 1 Gbps of Bandwidth).

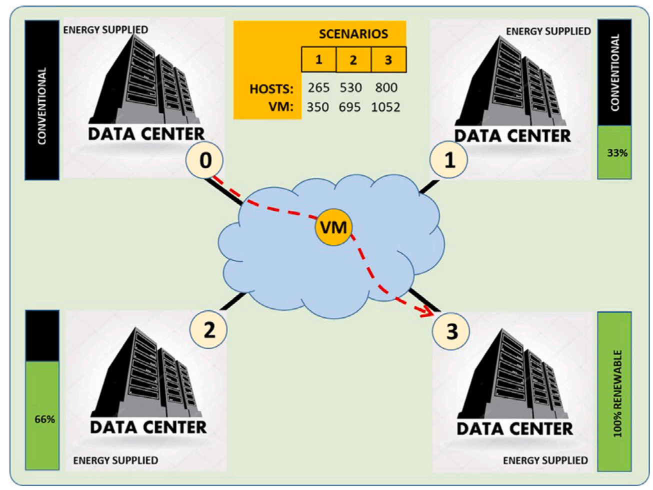

Fig. 4. General structure of meta-scheduler.

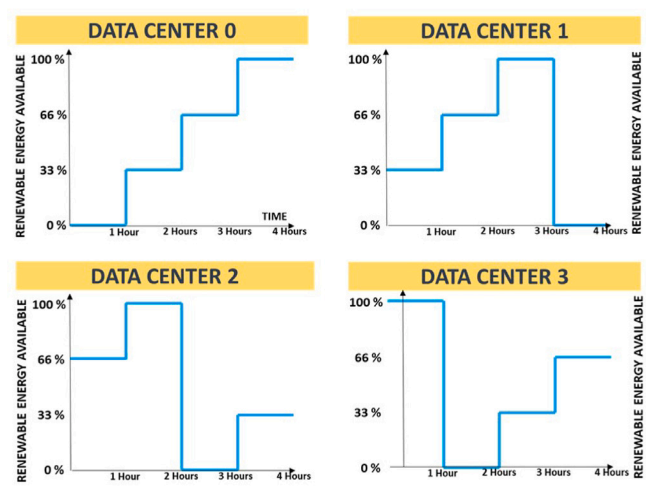

Fig. 5. Dynamic availability of renewable energy in data centers.

Realistic and representative workload scenarios are ensured by conducting simulations with traces from the Standard Performance Evaluation Corporation (SPEC) [42], including real traces from PlanetLab. Within the datacenters, the distribution of renewable energy fluctuates across four modes: 0%, 33%, 66%, and 100% of available renewable energy within each datacenter, as depicted in Fig. 5. This variation is observed over a 4-hour simulation time-lapse, allowing an exploration of the impact of different renewable energy levels on the cloud infrastructure's performance.

The simulation encompasses four datacenters, each representing distinct geographic regions, providing insight into distributed cloud infrastructure behavior, including considerations of data locality and network latency. Each datacenter is composed of a variable number

Table 7
Pittsburgh parameter configuration.

Simulations	Particles	Iterations	Crossover rate	Initial mutation rate	Selection rate	Replacement rate
30	64	50	0.8	0.1	0.8	0.8

Table 8
Q-Learning parameter configuration.

Simulations	Players	Iterations	Learning rate α	Discount factor γ	Reward r
30	64	50	0.5	0.9	0.5

Table 9
PSO-GTA parameter configuration.

Simulations	Players	Iterations
30	64	50

Table 10
Host configuration.

Variable	Value
PEs	2
RAM (MB)	4096
BW (MB)	1,000,000
MIPS	2660
Storage (MB)	1,000,000

Table 11
VM configuration.

Variable	Value
PEs	1
RAM (MB)	613/870/1740
BW (MB)	1000
MIPS	500/1000/2000/2500
Size (MB)	2500

Table 12
Cloudlet configuration.

Variable	Value
PEs	1
Length	36,000,000

of hosts – 265, 530, or 800 physical machines – each with specific characteristics detailed in Table 10, such as processing elements (PEs), RAM, bandwidth (BW), MIPS, and storage capacity.

The distribution of VMs spans the four datacenters, totaling 350, 695, or 1052 VMs. Each VM is distinguished by various features, including Processing Elements (PEs), RAM, Bandwidth (BW), MIPS, and size, as outlined in Table 11. The cloud simulation configuration incorporates a workload represented by cloudlets, which can vary significantly depending on the tasks within the workload. Each cloudlet is defined by its PEs and length, as detailed in Table 12.

During each execution of the fitness function, a simulation is carried out based on one of the sub-scenarios outlined in Table 13. This simulation captures all interactions between hosts and VMs, including migrations across the four CDCs. The results obtained encompass the total energy consumed over the four-hour simulation period, the renewable energy consumed for the current Knowledge Base (KB), the number of migrations executed, and the percentage of renewable energy over the total energy consumed. Subsequently, in subsequent simulations, the best KB is determined based on the percentage of renewable energy achieved among all possible solutions (particles/wolves). The KB demonstrating the highest percentage of renewable energy is deemed the most favorable and selected as the best solution.

The experiments were conducted based on three distinct scenarios, as described in [5] and summarized in Table 13, each characterized by different sizes: small, medium, and large. In Scenario 1, the simulation included 256 hosts, 350 VMs, and 500, 1500, or 3000 cloudlets.

Table 13
Scenarios of simulation based on Hosts, VMs and Cloudlets.

Scenario	Hosts	VMs	Cloudlets
1	265	350	500/1500/3000
2	530	695	1000/2000/5000
3	800	1052	1500/5000/10,000

Scenario 2 comprised 530 hosts, 695 VMs, and 1000, 2000, or 5000 cloudlets. Lastly, Scenario 3 involved 800 hosts, 1052 VMs, and 1500, 5000, or 10,000 cloudlets. Within each scenario, three sub-scenarios were initiated, corresponding to small, medium, and large simulations, with each sub-scenario representing varying numbers of cloudlets as shown in Table 13, thus resulting in nine simulation scenarios. The hosts were evenly distributed across four distant data centers, ensuring an equitable representation of the cloud infrastructure. These diverse scenarios and simulation variations allowed for a comprehensive evaluation of the system's performance under varying workload sizes and infrastructure capacities.

4.3. Simulation results

The comprehensive examination of convergence behavior across three distinct simulation scenarios, each featuring a variable number of cloudlets, provides valuable insights into the performance of the proposed PSO-GTA approach in comparison to the Q-Learning, Pittsburgh and KASIA algorithms. In the inaugural scenario, the PSO-GTA algorithm not only converges in the first place but also achieves superior results compared to Q-Learning, Pittsburgh and KASIA, both of which exhibit comparable performance. This initial success underscores the efficiency of the PSO-GTA approach in managing cloudlet dynamics and optimizing system outcomes. Building on this promising start, the second scenario amplifies the effectiveness of the PSO-GTA algorithm by yielding even better results. The convergence behavior of Q-Learning, Pittsburgh and KASIA remains similar in this scenario, further accentuating the unique advantages offered by the proposed approach. The nuanced adaptability of PSO-GTA to varying simulation conditions becomes evident, positioning it as a robust and versatile solution for cloudlet-based systems. A noteworthy observation emerges from the broader analysis of six simulations, where the PSO-GTA approach consistently outperforms the baseline results established by [5]. The improvements range from 3.68% to 5.32%, showcasing the reliability and efficacy of the PSO-GTA algorithm across diverse scenarios. This suggests that the proposed approach not only excels in convergence but also yields tangible benefits in optimizing cloudlet interactions, thereby enhancing the overall performance of cloud computing systems. In the final scenario, the PSO-GTA algorithm continues its streak of success by demonstrating optimal convergence behavior over Q-Learning, Pittsburgh and KASIA. Although the performance advantage over the baseline results is incremental, this consistency reinforces the reliability of the PSO-GTA approach in real-world cloudlet environments. The cumulative findings from these scenarios collectively emphasize the superiority of the proposed algorithm, making a compelling case for its adoption in optimizing cloudlet-based systems for enhanced efficiency and performance (see Figs. 6–8)

The results' comparison, as illustrated in Table 14, provides a detailed analysis of the percentage of renewable energy used across various simulation scenarios. The table delineates the performance metrics for Baseline results, KASIA, Pittsburgh, Q-Learning, and the proposed PSO-GTA approach. In each scenario characterized by different numbers of hosts, VMs, and cloudlets, the PSO-GTA approach consistently

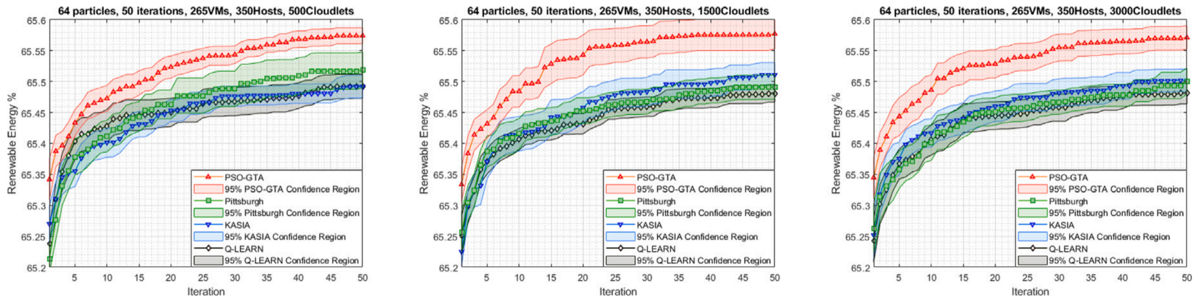

Fig. 6. Convergence behavior in first scenario.

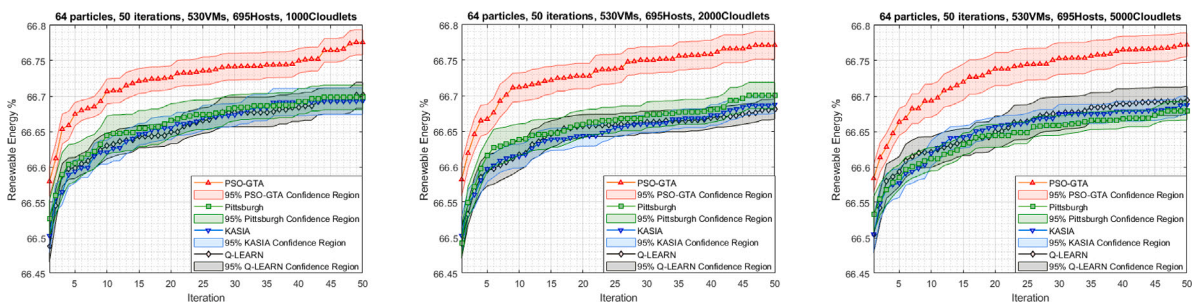

Fig. 7. Convergence behavior in second scenario.

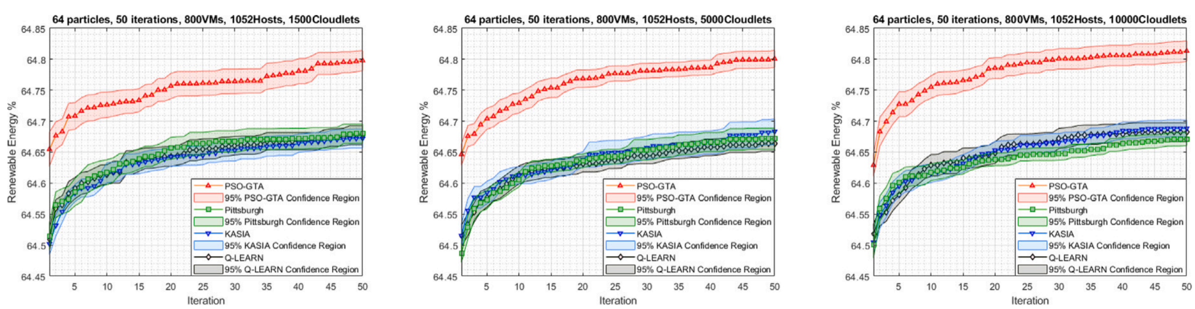

Fig. 8. Convergence behavior in third scenario.

Table 14
Results comparison between the Baseline results, KASIA, Pittsburgh, Q-Learning and PSO-GTA: Percentage of renewable energy used.

Hosts	VMs	Cloudlets	[5] Results	KASIA	Imp.	Pittsburgh	Imp.	Q-Learning	Imp.	PSO-GTA	Imp.
256	350	500	63.31 ± 0.13	65.49 ± 0.02	3.45%	65.52 ± 0.03	3.49%	65.49 ± 0.02	3.44%	65.57 ± 0.01	3.58%
256	350	1500	62.75 ± 0.11	65.51 ± 0.02	4.40%	65.49 ± 0.02	4.37%	65.48 ± 0.01	4.35%	65.58 ± 0.02	4.51%
256	350	3000	62.68 ± 0.15	65.50 ± 0.02	4.50%	65.50 ± 0.02	4.50%	65.48 ± 0.02	4.47%	65.57 ± 0.02	4.61%
530	695	1000	63.71 ± 0.07	66.69 ± 0.02	4.68%	66.70 ± 0.02	4.69%	66.70 ± 0.02	4.69%	66.78 ± 0.02	4.81%
530	695	2000	63.40 ± 0.08	66.69 ± 0.01	5.19%	66.70 ± 0.02	5.21%	66.68 ± 0.01	5.17%	66.77 ± 0.02	5.32%
530	695	5000	63.65 ± 0.09	66.69 ± 0.01	4.77%	66.68 ± 0.01	4.76%	66.69 ± 0.02	4.78%	66.77 ± 0.02	4.91%
800	1052	1500	64.36 ± 0.05	64.67 ± 0.02	0.49%	64.68 ± 0.02	0.50%	64.68 ± 0.02	0.49%	64.80 ± 0.02	0.68%
800	1052	5000	63.85 ± 0.07	64.68 ± 0.02	1.30%	64.67 ± 0.02	1.29%	64.66 ± 0.01	1.27%	64.80 ± 0.01	1.49%
800	1052	10,000	64.28 ± 0.09	64.69 ± 0.02	0.64%	64.67 ± 0.01	0.61%	64.68 ± 0.02	0.63%	64.81 ± 0.02	0.83%

Table 15
Results comparison between KASIA, Pittsburgh, Q-Learning and PSO-GTA: Total energy consumed in kWh.

Scenarios			KASIA	Pittsburgh	Q-Learning	PSO-GTA
Hosts	VMs	Cloudlets	Total (kWh)	Total (kWh)	Total (kWh)	Total (kWh)
256	350	500	8.09	8.12	8.14	8.09
256	350	1500	8.07	8.09	8.04	8.04
256	350	3000	8.11	8.13	8.11	8.08
530	695	1000	15.14	15.18	15.23	15.10
530	695	2000	15.17	15.27	15.44	15.10
530	695	5000	15.24	15.17	15.15	15.38
800	1052	1500	22.72	22.62	22.66	22.38
800	1052	5000	22.31	22.45	22.51	22.51
800	1052	10,000	22.64	22.61	22.56	22.53

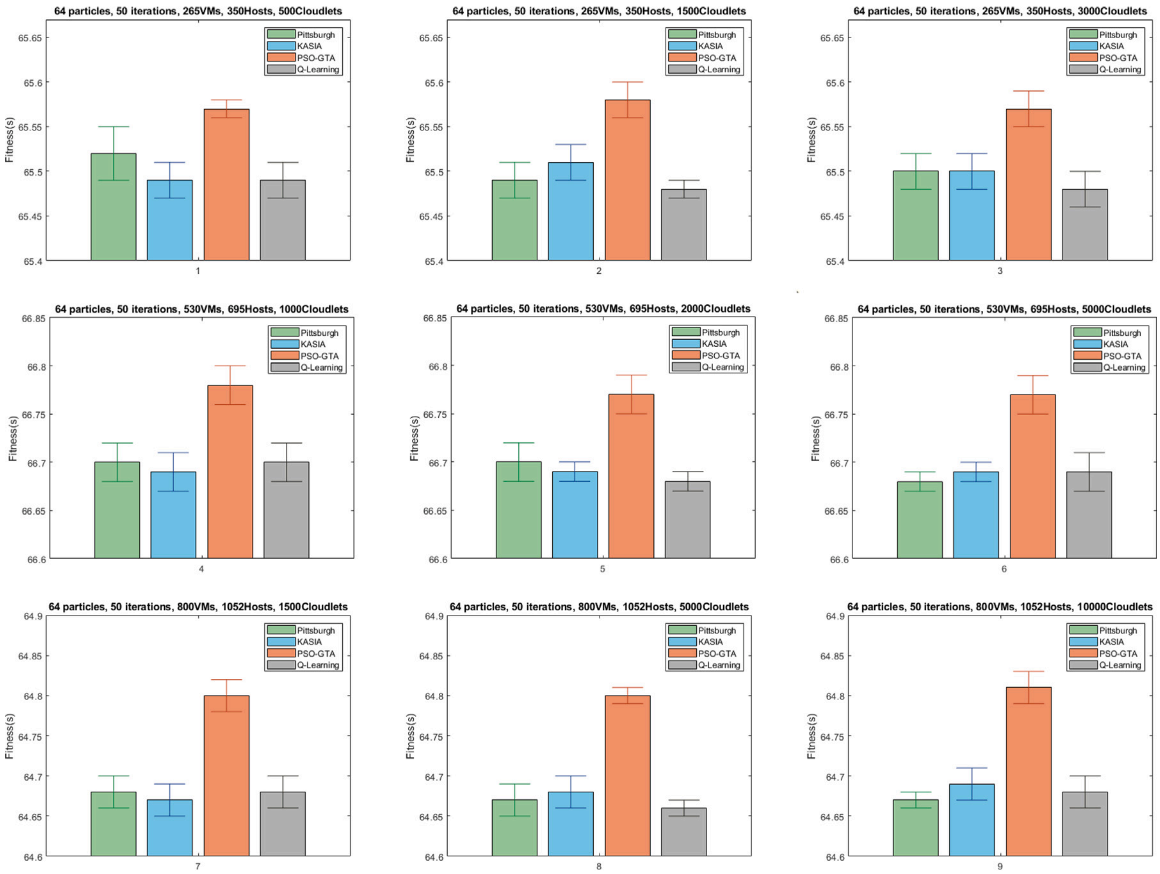

Fig. 9. Optimized values across all scenarios.

outperforms both the baseline and the comparative algorithms, showcasing its efficacy in optimizing the utilization of renewable energy resources.

Table 15 presents a comprehensive comparison of results among KASIA, Pittsburgh, Q-Learning, and PSO-GTA, regarding the total energy consumption in kWh achieved by the best iteration in the simulations for each scenario. The data is structured to show the performance metrics for various configurations of hosts, VMs, and cloudlets. Each algorithm's effectiveness is evaluated based on the total energy consumption. Furthermore, across the different scenarios, the table reveals subtle variations in energy consumption, highlighting the potential efficiency gains or losses achieved by each algorithm. This detailed analysis facilitates informed decision-making in selecting the most suitable optimization approach for cloud infrastructure, considering energy efficiency.

Furthermore, Fig. 9 visually represents the optimal values attained by each algorithm in the various simulation scenarios. The plotted graph distinctly shows the superior performance of the PSO-GTA approach, consistently achieving higher percentages of renewable energy utilization compared to KASIA, Pittsburgh, and the baseline results. This graphical representation reinforces the findings from the tabulated results, highlighting the PSO-GTA algorithm's ability to converge optimally and make efficient use of renewable energy resources across diverse cloud computing scenarios. The incremental improvements achieved by the PSO-GTA approach underscore its potential for enhancing sustainability and efficiency in cloudlet-based systems.

5. Conclusions and future actions

In conclusion, the PSO-GTA approach, utilized for optimizing renewable energy resources within cloud computing environments, has demonstrated superior performance when compared to traditional methods such as Q-Learning, KASIA and Pittsburgh. The infusion of game theory principles into the PSO algorithm has greatly facilitated more efficient decision-making processes, resulting in heightened energy utilization and improved cost-effectiveness. The comparative analysis showcased that the PSO-GTA approach not only outperformed existing methodologies in optimizing renewable energy resources but also exhibited robust adaptability to the dynamic and uncertain nature of cloud computing environments. The synergistic integration of game theory and PSO has provided a more comprehensive and flexible framework for addressing challenges associated with renewable energy optimization in cloud computing.

One promising avenue for future research involves extending the PSO-GTA approach to incorporate multi-objective optimization. This entails simultaneously optimizing multiple conflicting objectives, such as minimizing energy costs, maximizing resource utilization, and minimizing environmental impact. Introducing a multi-objective perspective will offer a more holistic and nuanced solution, allowing decision-makers to explore a diverse set of trade-offs and find Pareto-optimal solutions that effectively balance competing objectives. Another area for future work revolves around refining the PSO-GTA approach to incorporate advanced container migration strategies. As cloud environments increasingly rely on containerization for resource allocation and scalability, optimizing the migration of containers to leverage renewable energy sources becomes crucial. Investigating and implementing intelligent container migration algorithms within the PSO-GTA framework can lead to more efficient resource utilization, reduced energy consumption, and improved overall system performance. This research direction aligns with the evolving trends in cloud computing and sustainable practices, ensuring the adaptability of the PSO-GTA approach to cutting-edge technologies.

Glossary

CDC Cloud Data Center
PSO Particle Swarm Optimization
VM Virtual Machine
MIPS Million Instructions Per Second
DC Data Center
H Host
C Cloudlet/Task
R Resources
Y Computational Cost
X Profit
RR Resource Requirements
T Execution Time
D_n Data
τ_n Total Computation
S/s Strategy
t_n Computational Time
f_n Computing Power
e_n Energy Consumption
σ_n Energy Consumption Factor
E_n Cost of Computational Task
β_n^t Delay Weight Factor
β_n^e Energy Weight Factor
μ Makespan
u_n Utility
F Resource Scheduling
P Position/Strategy
V Velocity
ra Encoded Antecedent Fuzzy Input
rc Encoded Consequent Fuzzy Input
ro Encoded Logical Operator Fuzzy Input
MF_a Antecedent Membership Function
MF_c Consequent Membership Function
RB Rule Base
KB Knowledge Base

CRediT authorship contribution statement

Francisco Javier Maldonado-Carrascosa: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Sebastián García-Galán:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Resources, Project administration, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Manuel Valverde-Ibáñez:** Writing – review & editing, Visualization, Supervision, Investigation, Conceptualization. **Tomasz Marciniak:** Writing – review & editing, Visualization, Supervision, Investigation, Conceptualization. **Małgorzata Szczerska:** Writing – review & editing, Visualization, Supervision, Investigation, Conceptualization. **Nicolás Ruiz-Reyes:** Writing – review & editing, Visualization, Supervision, Resources, Investigation, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Sebastian Garcia Galan reports financial support was provided by Government of Andalusia. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work has been supported by the Research Project P18-RT-4046, funded by the Andalusian Government in Spain.

Funding

This work was supported by the Government of Andalusia [P18-RT-4046].

References

[1] Lei X, Shen Z, Štreimikienė D, Baležentis T, Wang G, Mu Y. Digitalization and sustainable development: Evidence from OECD countries. Appl Energy 2024;357:122480. http://dx.doi.org/10.1016/j.apenergy.2023.122480.

[2] Rosero D, Díaz N, Trujillo C. Cloud and machine learning experiments applied to the energy management in a microgrid cluster. Appl Energy 2021;304:117770. http://dx.doi.org/10.1016/j.apenergy.2021.117770.

[3] Katal A, Dahiya S, Choudhury T. Energy efficiency in cloud computing data centers: a survey on software technologies. Cluster Comput 2023;26(3):1845–75. http://dx.doi.org/10.1007/s10586-022-03713-0.

[4] Li J, Ai Q, Yin S, Hao R. An aggregator-oriented hierarchical market mechanism for multi-type ancillary service provision based on the two-loop Stackelberg game. Appl Energy 2022;323:119644. http://dx.doi.org/10.1016/j.apenergy.2022.119644.

[5] Seddiki D, Galán SG, Expósito JEM, Ibañez MV, Marciniak T, Rocío J, de Prado P. Sustainable expert virtual machine migration in dynamic clouds. Comput Electr Eng 2022;102:108257. http://dx.doi.org/10.1016/j.compeleceng.2022.108257.

[6] Mandal R, Mondal MK, Banerjee S, Srivastava G, Alnumay W, Ghosh U, Biswas U. MECpVmS: an SLA aware energy-efficient virtual machine selection policy for green cloud computing. Cluster Comput 2023;26(1):651–65. http://dx.doi.org/10.1007/s10586-022-03684-2.

[7] Serna M. An Introduction to Cooperative Game Theory. Lecture Notes for AGT-MIRI, 2016.

[8] Wang S, Hu Z, Deng Y, Hu L. Game-theory-based task offloading and resource scheduling in cloud-edge collaborative systems. Appl Sci 2022;12(12):6154. http://dx.doi.org/10.3390/app12126154.

[9] Ajmera K, Tewari TK. SR-PSO: server residual efficiency-aware particle swarm optimization for dynamic virtual machine scheduling. J Supercomput 2023;1–37. http://dx.doi.org/10.1007/s11227-023-05270-8.

[10] Yenugula M, Sahoo S, Goswami S. Cloud computing for sustainable development: An analysis of environmental, economic and social benefits. J Future Sustain 2024;4(1):59–66. http://dx.doi.org/10.5267/j.jfs.2024.1.005.

[11] Prado R, García-Galán S, Yuste AJ, Expósito JM. Genetic fuzzy rule-based scheduling system for grid computing in virtual organizations. Soft Comput 2011;15:1255–71. http://dx.doi.org/10.1007/s00500-010-0660-5.

[12] Santiago AS, Yuste A, Exposito JM, Galan SG, Marin JM, Bruque S. A dynamic-balanced scheduler for genetic algorithms for grid computing. WSEAS Trans Comput 2009;8(1):11–20.

[13] García-Galán S, Prado RP, Expósito JEM. Swarm fuzzy systems: knowledge acquisition in fuzzy systems and its applications in grid computing. IEEE Trans Knowl Data Eng 2013;26(7):1791–804. http://dx.doi.org/10.1109/TKDE.2013.118.

[14] Tran CH, Bui TK, Pham TV. Virtual machine migration policy for multi-tier application in cloud computing based on Q-learning algorithm. Computing 2022;104(6):1285–306. http://dx.doi.org/10.1007/s00607-021-01047-0.

[15] Buyya R, Ilager S, Arroba P. Energy-efficiency and sustainability in new generation cloud computing: A vision and directions for integrated management of data centre resources and workloads. Softw - Pract Exp 2024;54(1):24–38. http://dx.doi.org/10.1002/spe.3248.

[16] Zhao M, Wang X, Mo J. Workload and energy management of geo-distributed datacenters considering demand response programs. Sustain Energy Technol Assess 2023;55:102851. http://dx.doi.org/10.1016/j.seta.2022.102851.

[17] Jeyaraj R, Balasubramaniam A, MA AK, Guizani N, Paul A. Resource management in cloud and cloud-influenced technologies for internet of things applications. ACM Comput Surv 2023;55(12):1–37. http://dx.doi.org/10.1145/3571729.

[18] Vellela SS, Reddy BV, Chaitanya KK, Rao MV. An integrated approach to improve E-healthcare system using dynamic cloud computing platform. In: 2023 5th international conference on smart systems and inventive technology. ICSSIT, IEEE; 2023, p. 776–82. http://dx.doi.org/10.1109/ICSSIT55814.2023.10060945.

[19] Ghandour O, El Kafhali S, Hanini M. Computing resources scalability performance analysis in cloud computing data center. J Grid Comput 2023;21(4):61. http://dx.doi.org/10.1007/s10723-023-09696-5.

[20] Kumar MS, Karri GR. Eeoa: cost and energy efficient task scheduling in a cloud-fog framework. Sensors 2023;23(5):2445. http://dx.doi.org/10.3390/s23052445.

[21] Taraghi B, Hellwagner H, Timmerer C. LLL-CAdViSE: Live low-latency cloud-based adaptive video streaming evaluation framework. IEEE Access 2023;11:25723–34. http://dx.doi.org/10.1109/ACCESS.2023.3257099.

[22] Yu X, Hu Y, Zhou D, Wang Q, Sang X, Huang K. Carbon emission reduction analysis for cloud computing industry: Can carbon emissions trading and technology innovation help? Energy Econ 2023;106804. http://dx.doi.org/10.1016/j.eneco.2023.106804.

[23] Wu Y, Cai C, Bi X, Xia J, Gao C, Tang Y, Lai S. Intelligent resource allocation scheme for cloud-edge-end framework aided multi-source data stream. EURASIP J Adv Signal Process 2023;2023(1):56. http://dx.doi.org/10.1186/s13634-023-01018-x.

[24] Chang W, Dong W, Yang Q. Day-ahead bidding strategy of cloud energy storage serving multiple heterogeneous microgrids in the electricity market. Appl Energy 2023;336:120827. http://dx.doi.org/10.1016/j.apenergy.2023.120827.

[25] Park J, Han K, Lee B. Green cloud? An empirical analysis of cloud computing and energy efficiency. Manage Sci 2023;69(3):1639–64. http://dx.doi.org/10.1287/mnsc.2022.4442.

[26] Ahmad S, Mishra S, Sharma V. Green computing for sustainable future technologies and its applications. In: Contemporary studies of risks in emerging technology, part a. Emerald Publishing Limited; 2023, p. 241–56. http://dx.doi.org/10.1108/978-1-80455-562-020231016.

[27] Alyas T, Ghazal TM, Alfurhood BS, Ahmad M, Thawabeh OA, Alissa K, Abbas Q. Performance framework for virtual machine migration in cloud computing. Comput Mater Contin 2023;74(3).

[28] Ullah A, Nawi NM. An improved in tasks allocation system for virtual machines in cloud computing using HBAC algorithm. J Ambient Intell Humaniz Comput 2023;14(4):3713–26. http://dx.doi.org/10.1007/s12652-021-03496-z.

[29] Prity FS, Gazi MH, Uddin KA. A review of task scheduling in cloud computing based on nature-inspired optimization algorithm. Clust Comput 2023;26(5):3037–67. http://dx.doi.org/10.1007/s10586-023-04090-y.

[30] El Mir I, Kandoussi EM, Hanini M, Haqiq A, Kim DS. A game theoretic approach based virtual machine migration for cloud environment security. Int J Commun Netw Inf Secur 2017;9(3):345–57. http://dx.doi.org/10.17762/ijcnis.v9i3.2579.

[31] Khaleel MI. Region-aware dynamic job scheduling and resource efficiency for load balancing based on adaptive chaotic sparrow search optimization and coalitional game in cloud computing environments. J Netw Comput Appl 2024;221:103788. http://dx.doi.org/10.1016/j.jnca.2023.103788.

[32] Xu F, Xie Y, Sun Y, Qin Z, Li G, Zhang Z. Two-stage computing offloading algorithm in cloud-edge collaborative scenarios based on game theory. Comput Electr Eng 2022;97:107624. http://dx.doi.org/10.1016/j.compeleceng.2021.107624.

[33] Smys DS, Ranganathan DG. Performance evaluation of game theory based efficient task scheduling for edge computing. J IoT Soc Mob Anal Cloud 2020;2(1):50–61. http://dx.doi.org/10.36548/jismac.2020.1.005.

[34] Asghari A, Azgomi H, Darvishmofarahi Z. Multi-objective edge server placement using the whale optimization algorithm and game theory. Soft Comput 2023;1–15. http://dx.doi.org/10.1007/s00500-023-07995-3.

[35] Gunigari H, Chitra S. Energy efficient networks using ant colony optimization with game theory clustering. Intell Autom Soft Comput 2023;35(3). http://dx.doi.org/10.32604/iasc.2023.029155.

[36] Zade BMH, Mansouri N. Improved red fox optimizer with fuzzy theory and game theory for task scheduling in cloud environment. J Comput Sci 2022;63:101805. http://dx.doi.org/10.1016/j.jocs.2022.101805.

[37] Zeng B, Zhou Y, Xu X, Cai D. Bi-level planning approach for incorporating the demand-side flexibility of cloud data centers under electricity-carbon markets. Appl Energy 2024;357:122406. http://dx.doi.org/10.1016/j.apenergy.2023.122406.

[38] Godhrawala H, Sridaran R. A dynamic Stackelberg game based multi-objective approach for effective resource allocation in cloud computing. Int J Inf Technol 2023;15(2):803–18. http://dx.doi.org/10.1007/s41870-022-00926-9.

[39] Saeed MA, Eladl AA, Alhasnawi BN, Motahhir S, Nayyar A, Shah MA, Sedhom BE. Energy management system in smart buildings based coalition game theory with fog platform and smart meter infrastructure. Sci Rep 2023;13(1):2023. http://dx.doi.org/10.1038/s41598-023-29209-4.

[40] Smith SF. A learning system based on genetic adaptive algorithms. University of Pittsburgh; 1980.

[41] Prado R, Galán SG, Yuste AJ, Expósito JM, Santiago AS, Bruque S. Evolutionary fuzzy scheduler for grid computing. In: Bio-inspired systems: computational and ambient intelligence: 10th international work-conference on artificial neural networks, IWANN 2009, Salamanca, Spain, June 10-12, 2009. proceedings, part i 10. Springer; 2009, p. 286–93. http://dx.doi.org/10.1007/978-3-642-02478-8_36.

[42] Standard Performance Evaluation Corporation (SPEC), http://www.spec.org/index.html. [Accessed 26 July 2023].

Francisco Javier Maldonado Carrascosa (M.Sc.) received his M.Sc. degree in Physics and Mathematics from Granada University. Currently, he is a Ph.D. student at the University of Jaén. His current research interests include artificial intelligence, cloud computing scheduling and game theory.

Sebastián García Galán (Ph.D.) is a full professor at the Telecommunication Engineering Department of the University of Jaén. His research interests include artificial intelligence, IoT technologies, biomedical signals and cloud computing. He is the author of more than 100 research publications.

Manuel Valverde Ibáñez (Ph.D.) is an associate professor in the Electrical Engineering Department of the University of Jaén, Spain. His research interests include renewable energy power systems, power quality, modeling and control of power converters and smart grids. He is the author of almost 30 research publications.

Tomasz Marciniak (Ph.D.) is an associate professor in the Telecommunication, Computer Science and Electrical Engineering Department of the Bydgoszcz University of Science and Technology. His research interests include networking, artificial intelligence, IoT technologies, and bioinformatics. He is the author of more than 60 research publications.

Małgorzata Szczerska (Ph.D.) is a full professor at the Metrology and Optoelectronics Department of the Gdańsk University of Technology. Her research interests include biophotonics, spectroscopy and fiber-optic sensor. She is the author of more than 100 research publications.

Nicolás Ruiz Reyes (Ph.D.) is a full professor at the Telecommunication Engineering Department of the University of Jaén. His research interests include artificial intelligence, sound and biomedical signals and optimization. He is the author of more than 100 research publications.