Int. J. Fuzzy Syst. (2024) 26(1):368–389
https://doi.org/10.1007/s40815-023-01605-y

Enhanced Efficiency in Fog Computing: A Fuzzy Data-Driven Machine Selection Strategy

Hadi Zavieh[1] · Amir Javadpour[2,6] · Forough Ja'fari[3] · Arun Kumar Sangaiah[4,7] · Adam Słowik[5]

Received: 22 April 2023 / Revised: 12 August 2023 / Accepted: 30 August 2023 / Published online: 1 November 2023

Abstract With the rapid proliferation of IoT and Cloud networks and the corresponding number of devices, handling incoming requests has become a significant challenge. Task scheduling problems have emerged as a common concern, necessitating the exploration of new methods for request management. This paper proposes a novel approach called the Fuzzy Inverse Markov Data Envelopment Analysis Process (FIMDEAP). Our method combines the strengths of the Fuzzy Inverse Data Envelopment Analysis (FIDEA) and Fuzzy Markov Decision Process (FMDP) techniques to enable the efficient selection of physical and virtual machines while operating in a fuzzy mode. We represent data as triangular fuzzy numbers and employ the alpha-cut method to solve the proposed models. The paper provides a mathematical optimization model for the proposed method and presents a numerical example for illustration. Furthermore, we evaluate the performance of our method in a cloud environment through simulations. The results demonstrate that our approach outperforms existing methods, namely PSO + ACO and FBPSO + FBACO, in terms of key metrics, including energy consumption, execution cost, response time, gain of cost, and makespan.

Keywords Task scheduling · Markov decision process · Data envelopment analysis · Fuzzy numbers · Green computing

✉ Amir Javadpour
a.javadpour87@gmail.com

✉ Arun Kumar Sangaiah
aksangaiah@ieee.org

1 School of Economics and Statistics, Guangzhou University, Guangzhou 510006, China

2 Department of Computer Science and Technology (Cyberspace Security), Harbin Institute of Technology, Shenzhen, China

3 Department of Computer Engineering, Sharif University of Technology, Tehran, Iran

4 International Graduate School of AI, National Yunlin University of Science and Technology, Douliu, Taiwan

5 Department of Electronics and Computer Science, Koszalin University of Technology, Koszalin, Poland

6 ADiT-Lab, Electrotechnics and Telecommunications Department, Instituto Politécnico de Viana do Castelo, 4900–347 Porto, Portugal

7 Department of Electrical and Computer Engineering, Lebanese American University, Byblos, Lebanon

1 Introduction

Cloud is a large network that is used today in various places, including data centers of large organizations that store data. Given that in the cloud network there are many users such as mobile, computer, IoT, etc. that are connected, so there is a phenomenon called Task scheduling that must be considered, means that the data of which user and with what priorities should be processed first [1]. Task scheduling phenomenon and energy consumption are the most important challenges in heterogeneous systems and cloud computing. Recourse (requests) management and their proper allocation in order to run on machines, has a significant impact on the quality of system performance. When the huge volume of requests that enter the network are properly and accurately managed, it reduces energy consumption, execution time and operating costs, which ultimately leads to the establishment of green computing in the cloud network [2–4]. For this purpose, in the field of scheduling, researchers utilized different methods to optimize energy consumption. Given that we face an NP-hard problem in scheduling problems, a number of researchers used meta-heuristic methods to proposed some ranking models. The RR method assigned requests to machines randomly [2, 5], due to the weakness of this method, the possibility of not optimal use of machine capacity, ranking methods and meta-heuristic methods are presented. Dynamic voltage and frequency scaling (DVFS) algorithm allocates requests based on the characteristics of each machine and by giving weight to each component of the machine, including RAM, CPU and hard drive [2, 6, 7]. The most important meta-heuristic optimization algorithms for allocating requests to machines that consume less energy are Particle Swarm Optimization (PSO) and Ant colony optimization (ACO), which aim to select the shortest path for assigning requests to machines. Genetic Algorithm (GA) performs allocation based on the selection of the best chromosome obtained from a combination of several chromosomes [8, 9]. There are various meta-heuristic algorithms, but we will not discuss them since our work is based on only the above methods.

1.1 Problem Statement

In the Internet of Things (IoT) network, due to the large number of heterogeneous devices and sensors that are interconnected, we are faced with large amounts of data that require communication, coordination and interoperability of resources and devices, as well as in the Fog network, which is a cloud area with enormous sensors and connected devices, the need for task scheduling is felt, so choosing a machine that creates the shortest path for sending tasks to run on the network is essential [10, 11].

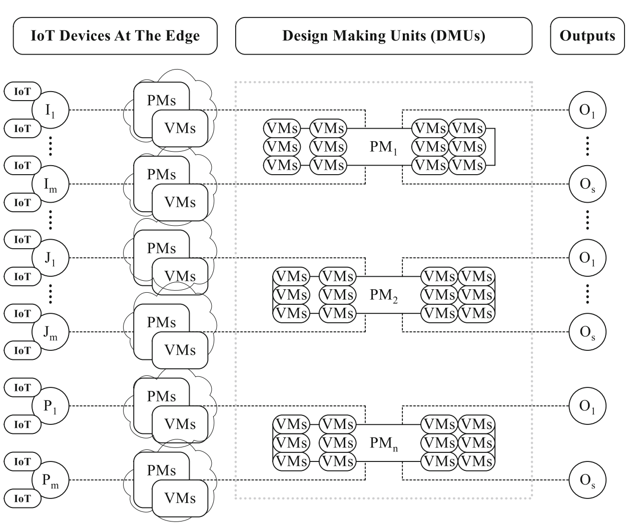


Fig. 1 Relationship between task scheduling and request assignment to virtual machines managed by efficient physical machines

Also, according to the Fig. 1, determining efficient physical machines using a mathematical optimization algorithm improves cloud network performance. As can be seen in Fig. 1, each physical machine generates outputs by receiving a series of inputs, and by using these inputs and outputs, the efficiency of each physical machine is obtained through a Data Envelopment Analysis model. The inputs of physical machines can be considered as one or more items including RAM, CPU, MIPS or HDD-Hard, and the output of the virtual machine optimization problem is assumed to be the result of task assignment [12, 13].

In short, the main problem of establishing Green Computing in the cloud network is to minimize energy and resource consumption, this process is done by observing the correct scheduling that results from the proper allocation of input requests to machines [14].

1.2 Motivation

Due to the problems in the cloud network that prevent the proper distribution of resources between users and cause users to have problems when using the network, providing solutions to the problems caused by the large cloud network and unlimited network-connected devices, as well as load balancing and resource depletion is very important. On the other hand, the problem of assigning the incoming requests to the network has motivated in-network research. Due to the increasing number of devices, especially in 5G environments, inter-vehicle networks, IoT networks and Fog environments and rapid growth of users, so these users and tasks that users perform on the network must be managed, so task scheduling problem has provided an essential motivation to start this research.

Finally, it is important that tasks are properly handled to store energy in the network, keep the network SLA normal, and maintain network load balanced. On the other hand, choosing an efficient physical machine in which a virtual machine is active in order to improve performance and increase system efficiency is considered as an optimization problem. Therefore, choosing machines that increase the efficiency and effectiveness of the system while considering the shortest path is important, especially when faced with inaccurate or fuzzy data.

We have special innovations in this field that use mathematical algorithms to solve existing problems. Our contribution is as follows:

- In order to select efficient physical machines, we used a mathematical optimization algorithm in which the efficiency of each machine is calculated with fuzzy inputs and outputs for machines.
- The FIDEA model is also proposed for the case where fuzzy inputs and outputs fluctuate, so that, it adjusts the

amount of input fluctuations caused by output fluctuations in such a way that the efficiency of the machines is maintained.

- For each of the physical machines, Ram and CPU are considered as inputs and virtual machines in fuzzy mode are considered as outputs, each of which can have tolerance, and with these inputs and outputs, the best physical and virtual machines are used in the Task Scheduling process.

1.3 Goal of This Research

In this paper, we propose a fuzzy Markov algorithm combined with a fuzzy Inverse Markov Data Envelopment Analysis Process (FIMDEAP) to perform task scheduling in the cloud network while maintaining machine performance. To evaluate the proposed algorithm, we used Service Level Agreement (SLA) which includes makespan and gain cost [7, 15] The main goal of this paper is to select efficient physical machines in order to increase system efficiency and also to allocate requests to virtual machines under the management of efficient physical machines with the shortest path in fuzzy environments. The proposed FIMDEAP model saves energy, establishes load balancing and improves SLA standards compared to other algorithms while maintaining machine performance.

In general, goals are divided into two categories:

- Main goals:

– Energy consumption reduction
– Response time
– Network Profitability
– Handle requests

- Sub-goals:

– Select efficient physical machines by presenting the FIDEA optimization model
– Select the appropriate virtual machines for scheduling with the fuzzy Markov method
– Energy and resource saving through choosing the right physical and virtual machines

1.4 Paper Structure

The structure of this paper is as follows: basic concepts and main challenges are discussed in Sect. 1. Section 2 deals with the research background and related work and their comparison and then details of the methods are given in Table 1. Our proposed method is provided in Sect. 3. Simulation results and the proof of the efficiency of the proposed method are shown in Sect. 4. Conclusions and future work of this research are presented in Sect. 5.

2 Related Work

In this section, we review task scheduling methods in the field of fuzzy markup, the details of which are as follows.

Authors in [7], processed data based on adaptability, data planning, and network coding to examine data encryption and decryption in IoT, then and to examine the data, they performed data dissemination (propagation) in IoT based on that. To do this, the authors proposed a hybrid Markov-Neural Networks (PECMAN) method for efficient scheduling (optimal time achievement), and reduction of packet sending losses. They used Markov's decision-making process in the Internet of Things to examine compatibility and dynamic innovative programming. Using the ADP method, they designed an AMDP with infinite phases to solve the propagation planning problem. Comparing EMUSA, Polynomial-time Optimal Storage Allocation (DSA) and Event-Aware Back Pressure Scheduling (EABS) with Contiki's proposed polymorphic erasure coding with Markov Design Adaptability and Neural Network (PECMAN) showed greater efficiency in terms of cost, time, and latency. As a result of the evaluation criteria, they compare the probability of reaching the maximum number of IoT devices, the number of IoT devices versus collisions, the number of IoT devices versus collisions with queue size, operational capacity, operational capacity with queue size, the number of gates versus failure possibility, propagation delay, propagation delay with queue size, energy consumption, as well as energy consumption with queue size and type.

In [16], authors considered Secure storage for data and used a heuristic fuzzy algorithm in the cloud for data allocation. Because of the excessive data entry in the cloud network, load balancing is disrupted, so to prevent it, a secure storage is placed in the network. When the sensors or tasks that move between the generated queue and the cloud space are too large, the network load and bandwidth will be disrupted, eventually causing the loss of Energy Consumption and Green Computing in the network. They considered the use of secure storage for data in the cloud infrastructure, used Secure storage to optimize resources, and then provided better energy to the network through this secure environment. In this paper, heuristic fuzzy algorithms are presented for securing allocation of storage in cloud environments for large networks with clustering of client systems using a fuzzy clustering mechanism. As scalable cloud storage is essential for sensitive data in industry and companies, the authors presented a heuristic fuzzy algorithm for secure cloud storage allocation in cloud

Table 1 Details of research background and related work

References	Short description	Criteria						Advantages	Disadvantages
		Cost	Overhead	Response time	Delay	Energy efficiency	QoS		
Rathanasalam and Kanagasabai [7]	PECMAN	*	*	*				Low execution time high speed convergence, low execution cost, low overhead, Energy Efficient, better adaptability	High complexity
Sivaram et al. [16]	Fuzzybased Heuristic					*		Enhance power consumption, Efficient load Balancing, efficient storage allocation	Without compared PSO
Kalnoor and Subrahmanyam [17]	MDP algorithm	*				*		Maintain low cost and efficient service time, resource and power optimization, exchange of data, object detection, coverage of sensing, and security. To achieve such design goals and to reduce energy consumption	High complexity and without considering Makespan
Huo et al. [18]	fuzzy control system for assembly line balancing							Increasing assembly performance using fuzzy control systems, higher production rates, lower work-in-progress, and improved quality while eliminating large amounts of production	Maintenance work is not discussed in this study
Li et al. [19]	RL, combined MILP			*			*	Minimize the makespan, high efficiency and high accuracy, better execution time	Without considering energy
Gao et al. [20]	MDPCO algorithm					*	*	Better proceeds on QOS criteria, Less Energy Efficient Cost (EEC), less offloading	There isn't good result for EEC, cost and reliability when the proposed method is compared with Chen and Zhang methods

Table 1 continued

References	Short description	Criteria						Advantages	Disadvantages
		Cost	Overhead	Response time	Delay	Energy efficiency	QoS		
He et al. [21]	AEOS, Generic MDP model, RL, combining HADRT with improved deep Q-learning				*			High accuracy and speed, Better performance on CPU time, good scalability, high total profit in a short time	In the complex situation with more constraint the model is not suitable, In some cases the total profit is less than others
Luo [22]	Deep reinforcement learning Algorithm							Minimizing the total tardiness	
Meena [23]	FSSOCO algorithm	*			*	*		Optimum multisite offloading, high speed, less total cost, less memory usage	High complexity and without considering Makespan, fuzzy Algorithm Considers only a phase of problem and detail of simulation is not considering in the real environment
Hamdy et al. [24]	ANN, firefly algorithm	*			*			Minimizing response time and cost of the service, efficient utilization, of cloud resources, service reliability and minimization of maintenance costs	Without considering energy
Xu et al. [25]	GEE Algorithm				*	*		Better energy-efficient with reliability	High complexity and without considering Makespan
Mei et al. [26]	Energy-aware Task Scheduling, DVFS-enabled Heterogeneous Clusters					*		Conclude the accurate amount of CPU Energy, 34% reduce Energy consumption	Minimize the makespan, high efficiency and high accuracy, better execution time
Peng et al. [27]	BHS algorithm					*		Save more energy, Reduce 56.12% of makespan	Minimize the makespan, high efficiency and high accuracy, better execution time
Mangalampalli et al. [28]	Whale Optimization Algorithm					*		Reduce energy consumption, reduce electricity consumption	High complexity

Table 1 continued

References	Short description	Criteria						Advantages	Disadvantages
		Cost	Overhead	Response time	Delay	Energy efficiency	QoS		
Ullah Tariq et al. [29]	Meta-heuristic ARSH-FATI algorithm, EECDF Algorithm					*		Energy efficient	High complexity and without considering Makespan

environments. The server generates a distributed public key using the RSA to ensure security and resolve memory recovery problems, and then server uses the K-nearest neighbors (KNN) method to search queries in order to store data in clusters by customers which is in fact a router for efficient storage allocation. The proposed method seeks to select the optimal cluster (cluster head among clusters) based on CPU Core, Speed and RAM Capacity. They claimed that, their method is a load balancing method in the cluster network, which increases efficiency by reducing energy consumption, in fact, it balances the load in the cluster network. Simulated results using Sheepdog and Apphorbor showed that the authors' method is better in terms of scheduling as well as load balancing than meta-heuristic techniques such as ACO and genetic algorithms, but their method was not compared to PSO.

In [17], authors examined Energy Efficient or green computing in the wireless sensor network. Due to the fact that in the cloud network, a series of inputs are considered sensors, so task scheduling for these sensors in the cloud network is essential in order to establish Energy Efficient to achieve green computing. This paper investigates the energy factor as one of the QOS factors, which reduces the energy consumption of sensor nodes using the MDP framework in the wireless sensor network (WSN). Solutions designed by MDP have been used for problems such as energy consumption, coverage area harvesting, resource optimization, solution tracking, power optimization, etc. A Semi-MDP based model is used to present several data aggregation and routing techniques. It has been investigated to obtain more and better solutions in the efficient sensor network by using the Markov process and the design of the MDP framework. The designed framework includes protocols and algorithms compatible with WSN.

Its objective is to assign requests to workstations, prioritizing requests, limiting workloads, and optimizing performance criteria. In [18], authors balanced the assembly line by considering the health of the machines and the real-time trigger settings for the first time. In fact, by improving the quality of decision making through using full information of the assembly process in real time, they determined the starting point for setting the initial assembly line. In the balancing process, they used a fuzzy control system to determine the rebalancing time and adjust the production rate of the workstations, which includes two types of fuzzy controllers to monitor and control the assembly line in real time. The first controller is used to support decision making for the rebalancing time of the assembly line, and the second controller adjusts production rate of each workstation. Also, when the machines are working, the health status of the machines is checked using a three-state Markov chain. Their method tries to react quickly to disturbances in the production line process. Their method intelligently controls the production line process and serves as a reference for researchers of Industry. Their proposed method, the assembly line with fuzzy control system (AS1), performed better than AS2, AS3, AS41 and AS42, so that compared to AS2 and AS3, it caused and compared to AS41 and AS42 that use a different control system, fuzzy control system AS1 performed better. In their work, the maintenance of machines, which is one of the influential factors in the production line process, has not been studied.

The authors [19] used NFV and SDN to optimize QOS and improved the evaluation criteria of QOS and delay of scheduling using the combination of Markov process and Reinforcement Learning (RL) method. They claimed that, NFV could be helpful in the field of IoT and QOS, and discussed the performance of virtualization models in modern networks such as 5 G and E2E. To reduce the task execution overhead, they used their hybrid method to predict subsequent tasks and used a combined MILP method to solve NP-hard problems and performed task scheduling using SDN/NFV. In fact, they investigated NFS problem to minimize the completion time of all services while meeting different End-to-End (E2E) latency requirements. In this regard, they presented a Mixed Integer Linear Programming (MILP) problem and used

Markov's decision as an efficient and accurate method to solve it. The authors also provided a Traffic Model for sending requests with a specific T between NFV components. By comparing their method with other heuristic ones in terms of Makespan and Running time criteria, they proved the superiority of their method which facilitates the guarantee of QOS in networks equipped with SDN/NFV. RL is one of the methods that can optimize scheduling in the cloud environment. This work has been done in the cloud to solve Np-hard problems, and one of its goals is to reduce traffic engineering, which optimizes network energy.

In [20], a Markov decision-making algorithm based on computation offloading is presented for the mobile network by considering time constraints in allocation process. They proposed a Markov Decision Process Algorithm (MDPCO) to minimize the energy consumption that may occur due to degradation of sent data to the cloud network remotely, and proposed an Offloading Algorithm based on Markov's decision making. By decomposing the main problem into two sub-problems, they optimized the local computation frequency sing convex optimization and the transmission power using the dual problem, and proposed efficient Offloading Algorithm MDPO for the decision-making problem. They modeled the whole loading process as MDP in terms of status, actions and benefits. They also proposed two improved algorithms to solve the MDP by introducing the state function and the Belman equation. They analyzed the time complexity of the MDPCO and MDPO algorithms. By comparing the proposed method with Guo, Chen and Zhang algorithms in terms of QOS, they proved the efficiency of their algorithm. In EEC, Cost and reliability criteria, MDPCO method performed better than Guo. As well, compared to the other two algorithms with different number of requests and weights in EEC criterion, the proposed method performed better. Also, the offloading rate is more reduced compared to other algorithms.

In the centralized model, we have a server such as a cloud network that is responsible for deciding on other nodes or network elements (client), but in the decentralized method, each element decides automatically. The authors in [30] applied Decentralized method to Fuzzy RL. To solve the unlimited Dec-POMDP, it was introduced an incremental fuzzy model to consider the cooperative mode, MAS. Additionally, a Multi-Agent algorithm was used where each node in a decentralized environment was assigned an agent that was defined as a queue pattern and a processing pattern. To approximate the optimal Q value for these belief vectors, an incremental clustering algorithm was used in conjunction with Q learning. The authors proposed the DIFRL algorithm as a decentralized controller to find a solution to evaluate large-scale Dec-POMDP criteria using a fuzzy rule for each factor. Finally, they expressed the scalability of their method in dealing with large scale problems in the real world as one of the advantages of the method. Experimental results showed this method performed better than compared ones in terms of execution time and execution accuracy and processing speed and occupied less memory. If the volume of tasks is small, the proposed method is more efficient than the centralized mode, so in the face of large tasks, the proposed method is not suitable, which is one of the disadvantages of this method.

In [21], they used the RL algorithm and Markov decision process to schedule Agile Earth Observation Satellites. An advanced heuristic algorithm (HADRT) and improved Q-learning algorithms were developed to solve the Markov decision-making process model for the scheduling problem (AEOS). In fact, they focused on the AEOS scheduling by proposing an effective RL approach. The authors proposed a new MDP framework for solving AEOS scheduling problems, which divides the problem into two sub-problems (1) work sequence problem and (2) scheduling problem that are processed in different parts of the MDP. They also developed an offline RL algorithm to guide the decision-making process, which includes a rule to avoid unlimited loops in each cycle in order to improve efficiency. They used a combination of deep Q-learning and an innovative innovative algorithm for scheduling AEOS, which solves the sequence and time problem in the AEOS scheduling problem, respectively. The proposed algorithm has higher accuracy and speed compared to other advanced RL algorithms. The comparison results of three methods HADRT, ALNS and RL showed that HADRT method with CPU time criterion performs better than other algorithms, while with HADRT algorithm, Total profit is less in most cases. In general, the proposed method works well for satellite scheduling problem with sufficient transmission and power sources, but is not suitable for more complex conditions with more constraints.

The authors in [22] proposed a Deep Q-Network (DQN) method for addressing DFJSP by inserting new tasks. In order to minimize input latency, the Dynamic Flexible Job shop scheduling problem (DFJSP) was modeled as a Markov decision-making process which determines the processing of operations sequentially in the next step and used the DRL method to solve the model. The paper considers general properties that the values [0,1] are extracted to represent a state at any point in time. Measures are designed to simultaneously determine which operations are processed in the next step and which devices are assigned to them. A DQN network is designed to measure the performance of each rule, which determines the most appropriate choice for dispatch in different decision-making situations. Numerical tests indicate the better performance and generality of QDN method and against other dispatching rules, this method has better performance in configurating trained and untrained productions.

In [23], the authors proposed a Fuzzy base optimization method to reduce Overloading on the Cloud Network. When a series of applications such as mobile, Fog Computing or IoT applications are integrated in the cloud, it causes a heavy load on the network, so we face an NP-hard problem for task scheduling. The authors presented a fuzzy hybrid algorithm with PSO, and obtained the initial population using PSO in order to perform task scheduling and applied a fuzzy pattern to it, which controls load and establishes overloading in the system. A fuzzy simplified swarm optimization (FSSOCO) algorithm based on cloud computing is proposed to achieve high processing speed, reduced latency, better computability, and better offloading (minimizing service costs and latency) with less memory consumption in Mobile Network Computing (MCC). They used fuzzy logic to identify powerful nodes and assign heavy calculations to those nodes. The proposed Fuzzy logic base method is used to reduce Complexity or Overhead, and the PSO is used to reduce weighted total cost, and Energy Consumption is controlled using Swarm Optimization.

The simulation results of the proposed FSSOCO method in comparison with other offloading techniques such as SPSO, MMRO and OMPSO confirm the better performance of the proposed method in terms of Energy consumption, Processing time and Weighted Total Cos, but Makespan that is a main criterion is not compared.

As a disadvantage, when PSO is used complexity is high and since the problem is NP-hard, it is not possible to control the load applied to the network, ie it cannot be detected that the resulting load includes which item for example Tasks Video, Noise or Communication or etc. As well, the fuzzy pattern considers a specific phase of the problem and does not take into account the details of the simulation and algorithm in a real environment, so it may not be appropriate to generate the initial population using PSO method in order to use the fuzzy method for task scheduling.

In [24], the authors presented a dynamic programming problem using cloud-optimized neural networks. This reference tries to take into account both the views of the user and the cloud service provider, which means that while minimizing response time and service costs, it focuses on optimal resource utilization, reliability, and minimizing maintenance costs. So, the authors presented an Artificial Neural Networks (ANN) based scheduling algorithm that is optimized using the firefly algorithm in the cloud environment. Experimental results showed that although this algorithm is simple, it increases the convergence speed and minimizes training error in different conditions.

In this paper, through ensuring system reliability in real-time embedded systems, energy-efficient planning problem is investigated. DVFS is efficient for optimizing energy consumption but it reduces system reliability, so the authors [25] proposed a greedy energy-efficient (GEE) algorithm to improve energy efficiency by considering reliable constraints. The GEE algorithm sets the task execution frequencies so that we have the least energy consumption in the execution of the tasks and this execution ends before the deadline. Since GEE reduces operating frequencies, a set of constraints has been analyzed and proposed to reduce prerequisite times during executions, which reduces system power consumption and execution time. To balance the execution frequencies, they proposed two energy-efficient algorithms that use processor. Given that the actual execution time of tasks may be less than the worst-case execution time in many cases, a dynamic global scheduling algorithm based on the real-time execution is proposed to maximize energy efficiency by taking into account system reliability. Experimental results indicated that their proposed methods have a significant improvement in energy efficiency and system reliability compared to other related methods.

In [26], Energy-aware request planning for constraints with deadlines in heterogeneous clusters equipped with DVFs is investigated. When dealing with very large data with high computational volume, energy storage and energy saving are of great importance. The authors examined energy savings in CPU hybrid clusters using a specific DVFs and aimed to minimize the total energy consumption for a batch of offline/online tasks or a sequence of tasks in real time by considering deadline for constraints. They provided a fast and accurate model for setting the right voltage/frequency for each task, packing a set of tasks on a number of servers to reduce static power consumption. Then, using a heuristic algorithm in both online and offline modes, they assigned several requests to a cluster that uses re-adjusting runtime of DVFs to create a balance between static and dynamic energy consumption. In fact, their model provides a non-linear relationship between the execution time of tasks and CPU speed for accelerated CPU applications, which results in achieving accurate CPU power consumption. Simulation showed that 34% of energy can be saved and the proposed method can be used for energy management in modern heterogeneous clusters.

Energy-Aware workflow scheduling in green cloud is investigate in [27] and a two-step algorithm called best heuristic scheduling (BHS) is proposed for DAG graph programming in cloud processors Resource allocation to requests is accomplished in the first step using four heuristic algorithms and the Grasshopper algorithm. According to the importance factor of the user or service provider, it determines the most appropriate way to accomplish tasks so as to achieve the optimal solution at the appropriate time. In the second step, the proposed BHS method minimizes energy consumption by considering the impact factor of the end user or service provider, start time, start-up and completion times of services and energy specifications of each virtual machine. In fact, the BHS method performs workflow scheduling and automatically generates the power consumption specifications by creating a methodology and selects the best sorting method based on a heuristic method and Grasshopper algorithm to increase the efficiency of task execution and then provides a model for energy consumption and duration of tasks. The simulation results in MATLAB emulator for 1000 requests using Dag graph showed that compared to BHS and multi-heuristic resource allocation algorithm (MHRA), the proposed BHS method saves 19.71% more energy and also 56.12% reduction for Makspan is observed in heterogeneous environments.

When assigning tasks to virtual machines in the Task Scheduling process, a large amount of carbon dioxide is released into the air, which is necessary to minimize it [28]. So, a whale optimization-based task scheduling algorithm is proposed for scheduling requests with prioritized performance on appropriate virtual machines in cloud network. It first determines the priorities of tasks and virtual machines to properly task assignment. then a multi-objective optimization model is developed to minimize energy consumption and power costs in the data center. Simulation results using the CloudSim simulator show that the proposed algorithm performed better than PSO and CS in terms of energy consumption and electricity cost and could minimize energy consumption. The energy consumptions in 100 requests were 10.75 and 9.87 for PSO and CS methods, respectively and 7.45 watts for the proposed method; in 500 requests were 13.46 and 12.38 for PSO and CS method respectively and 10.22 watts for the proposed method; and in 1000 requests were 17.65 and 16.86 for PSO and CS methods respectively and 11.56 watts for the proposed method.

In [29], inputs in the cloud network are considered as Internet of Things (IoT). They proposed an innovative rescheduling algorithm to convert in-period data dependencies to application-level periodic data dependencies to maximize parallelism and reduce rescheduling latency. They also introduced a new meta-heuristic algorithm called ARSH-FATI for task scheduling that considers processor power, the conflict between Network-on-Chip (NoC) links, and the communication between VFIs when mapping tasks. They also developed an Earliest Edge Consistent Deadline First (EECDF) Scheduling method that can be applied on both task nodes and communications. Their proposed method performs better in terms on energy management, so that, in 8 real criteria, the average energy efficiency was better than CA-TMESSearch and CA-TMES Quick by about 15% and 20% respectively. When ARSH-FATI is combined with the proposed method, energy saving increased to 35% and 40% respectively. Also compared to the R-DAG method, the proposed method achieved a 50% reduction in energy consumption. The authors [31] utilized fuzzy parameters when dealing with uncertainty conditions in the network and proposed the Fuzzy Logical Offloading method. Their approach utilized triangular fuzzy data in uncertain conditions and employed a ranking method to defuzzification such data. They used fuzzy numbers for data to reduce energy consumption. To achieve maximum machine processing within clusters with fuzzy data, they introduced an algorithm for learning and optimization. Their approach in the resource allocation process to different clusters adopted an Optimal Dag Task Offloading approach that utilized fuzzy logic for clustering. Through their conducted experiments, they demonstrated that their approach reduces resource utilization to improve energy consumption. simulation results by using triangular fuzzy numbers show that their proposed algorithm has better performance to complete tasks on time by considering Agreement Index and Robustness metrics. Their method tries to reduce energy consumption by completing tasks on time, but the algorithm did not consider other criteria such as cost, Makespan and response time. Also, they considered triangular fuzzy numbers in their scheduling process for inputs and then by using ranking methods, the fuzzy data changed to crisp data. Their method is not considered that if the fuzzy data fluctuates on the devices in the network, what will happen and we will solve this problem in our network. Authors [32] employed triangular fuzzy data for resource allocation and utilized the Cuckoo algorithm in the scheduling process to reduce energy consumption in the network. They used the Fuzzy Dematel method for scheduling, where the Dematel method is a pairwise comparison-based decision-making approach that experts use to create pairwise comparison matrices. Since the Dematel method struggles with uncertain conditions, the authors employed the Fuzzy Dematel algorithm for appropriate resource allocation in such circumstances. They also utilized triangular fuzzy data in the Fuzzy Dematel method when facing uncertainty conditions and employed ranking methods for defuzzification. They compared their proposed algorithm with ACO, CSA, and GA algorithms in terms of Energy Consumption, Service Level Agreement violation, Execution Time, Response Time, and Makespan metrics. They expressed that their simulation results, using CloudSim in four different scenarios, showed improved performance compared to other methods. Unlike [31], which used fuzzy data and then defuzzied them, their method utilizes a mathematical model called Dematel with fuzzy data and then fuzzified the fuzzy data to perform the scheduling process for energy consumption reduction, but also, they did not consider that what will be happen if the fuzzy data have fluctuated for machines on the network. Since problems with uncertainty data in Deep Leaning models cause low performance in classification tasks, the authors [33] utilized trapezoidal fuzzy data in the learning process to reduce energy consumption. They proposed an optimized fuzzy model called Optimization Fuzzy Deep Learning for data classification. Their proposed method combines DL techniques and fuzzy learning, where fuzzy values are represented as trapezoidal fuzzy numbers. They utilized NSGA II to optimize the membership function of trapezoidal fuzzy numbers considering multiple inputs and outputs in their optimization model. By presenting their proposed model, they reduced the impact of uncertainty by optimizing the membership function of trapezoidal fuzzy numbers and minimized the influence of noise resulting from such data using the optimized DL model. They demonstrated that the simulation results showed better performance of their proposed method compared to classifier methods and fuzzy DL models. Their method considered accuracy when they are facing with fuzzy data but the algorithm is not considered some criteria such as cost, makespan and execution time that influence on energy consumption in the network. Unlike [32] which used triangular fuzzy numbers, their method utilizes trapezoidal fuzzy number when they are facing with uncertain environments and then defuzzied the fuzzy numbers in their algorithm in the process of energy reduction. But still their algorithm cannot support the task scheduling process when they are facing fluctuating data, unlike our model. The authors [34] utilized trapezoidal fuzzy data and type-2 trapezoidal fuzzy data for modeling their proposed method to achieve energy reduction. They introduced a novel task scheduling approach in Industry 4.0 by considering trapezoidal fuzzy numbers with one core (triangular fuzzy numbers). They presented a multi-objective formula using trapezoidal fuzzy data for energy optimization in real-time embedded systems scheduling. For the first time in task scheduling parameters, they employed type-2 trapezoidal fuzzy data for modeling Interval Type-2 Fuzzy Sets and proposed an optimized approach based on the NSGA II evolutionary algorithms. They proposed two algorithms in the scheduling process and towards achieving an optimal energy consumption approach. The first algorithm creates a membership function of the type-2 trapezoidal fuzzy numbers as an interval from crisp input data, and the second algorithm aims to optimize energy consumption with NSGA-II using type-2 trapezoidal fuzzy numbers in the scheduling process. They categorized the input data into three groups to compare their proposed method: crisp data, trapezoidal fuzzy data, and type-2 trapezoidal fuzzy data. They demonstrated that the results obtained with type-2 fuzzy data were better than those obtained with other data types. They conclude crisp deadline, fuzzy deadline, and type-2 fuzzy deadline for satisfaction task scheduling and energy consumption. Still, they did not consider criteria such as cost, makespan execution time in the task scheduling process to achieve energy reduction. Unlike [31] and [31] which used triangular fuzzy numbers, and unlike [33], which utilizes trapezoidal fuzzy numbers they used crisp data, triangular fuzzy data, trapezoidal fuzzy data, and type-2 trapezoidal fuzzy data when they are facing uncertainty to achieve energy efficiency. But unlike our model, their model cannot support fluctuating data on the machines. The authors [35] employed fuzzy logic and fuzzy inference in neural networks to facilitate modeling under uncertainty. The data used in their proposed model are represented as fuzzy sets, indicating the degree of membership to specific sets. They introduced an approach for resource allocation and improved its performance to reduce energy consumption in the network. They also utilized the Ant Colony optimization method to enhance energy consumption. In their proposed approach, they employed virtual machine migration, which removes inefficient physical machines from the resource allocation cycle, resolves the resource granulation problem, and utilizes the full power of running servers. They compared their approach with reinforcement learning algorithms, multi-objective resource allocation, NSGA III, Whale Optimization and PSO. The results demonstrated that their proposed algorithm outperformed other methods. They considered delay criteria but they did not consider other criteria such as cost, Makespan and execution time for energy reduction in the network. Also, they considered fuzzy data in their scheduling process but they did not consider that if the fuzzy data are fluctuated on machines in the network, what will happen, unlike our method. The authors [36] utilized fuzzy logic to classify machine weights. They considered triangular fuzzy sets for the weights and employed the centroid method to defuzzification and make the weight data crisp. They proposed an algorithm for efficient task processing to energy reduction. To achieve proper scheduling while considering resource availability and task deadlines, they presented an integer programming problem that minimizes costs. To improve the learning process and select suitable machines, they utilized a fuzzy reinforcement learning algorithm that employs triangular fuzzy data. Their proposed algorithm combines fuzzy logic-based greedy heuristics with on-policy reinforcement learning, aiming to minimize energy consumption. Their proposed method shows an average improvement of 15.38 percent in energy consumption compared to other methods by considering the total task service time, and average response time criteria and cost, but they are not considered Makespan. Unlike other methods, their method utilizes triangular fuzzy number for both of the inputs and outputs when they are facing with uncertainty situation. But still their method cannot handle task scheduling process when the fuzzy inputs or output are fluctuated on machines unlike our method. The authors [37] utilized fuzzy sets when they are facing with uncertainty environment. They proposed a hybrid Fuzzy Interface System (FIS) with a genetic algorithm to reduce energy consumption. Since genetic algorithms are widely used in multi-objective optimization problems, the authors suggested a compatible genetic algorithm for proper scheduling of request execution to minimize the Makespan. Their proposed algorithm is a combined approach of partitioning and scheduling algorithms for suitable request execution between the fog and cloud layers under uncertainty conditions. Their proposed method showed a 48 percent improvement in performance compared to the EM-MOO method in the Makespan criterion and a 41 percent improvement compared to MAS-GA in the time criterion, but they did not consider cost. They used triangular fuzzy numbers in their task scheduling process and also their method cannot handle task scheduling process when the data are fluctuated on machines. The problem that has been overlooked in previous works is what happens if the input or output data is fuzzy and exhibits fluctuations. For example, if the output is fuzzy and fluctuates, to maintain the efficiency of the machines, to what extent should the inputs fluctuate in order to preserve machine efficiency and ensure uninterrupted scheduling in the network? Previous work has not addressed this issue. Therefore, in this paper, an optimization model is proposed for the first time in this regard, which determines the level of input fluctuations to maintain machine efficiency. Additionally, some related works in energy consumption that use crisp data for energy consumption are mentioned below. The advantage of our work compared to them is the ability to calculate energy consumption under uncertainty conditions and determine input fluctuations to preserve machine efficiency in case of tolerance in the outputs.

3 Proposed Method

We combined PSO and ACO methods with the fuzzy Bayesian method and used DVFS to establish a low-consumption cloud system. The present paper is based on this study in which we used the fuzzy Markov method for request assignment and then proposed a performance evaluation model to determine efficient physical machines in order to maintain the efficiency of machines with fuzzy data. Thus, when the proposed model is combined with the fuzzy Markov method, the shortest path for assigning requests to virtual machines under the management of physical machines is created while maintaining their efficiency. Finally, by choosing the right efficient physical and virtual machines, we can improve the evaluation criteria in the task scheduling process.

As shown in Fig. 2, the inputs of each machine are RAM and CPU with fuzzy data and the output of each physical machine is virtual machines with fuzzy data. In the proposed method, first the efficiency of decision units (physical machines) is obtained by considering the oscillating fuzzy values for inputs and outputs to maintain the efficiency of physical machines, and then the shortest path for sending requests is selected by fuzzy Markov method.

3.1 Data Envelopment Analysis

This non-parametric method was introduced by Charnes et al. to judge the performance of decision-making units (DMUs). [38] and named the CCR (Charnes, Cooper & Rohdes) model, the CCR model can be used for evaluating Physical and Virtual machines performance. Each machine uses some inputs to produce some outputs and the performance of machines conclude by the inputs and outputs. the multiplier form of the CCR model is introduced as follows:

$$\begin{aligned}
&\max\left(\sum_{j=1}^{n} u_r y_{rj}\right)\\
&S.t: \sum_{j=1}^{n} v_i x_{ij} = 1,\ i \in \{1,\ldots,m\},\\
&\qquad \sum_{j=1}^{n} u_r y_{rj} - \sum_{j=1}^{n} v_i x_{ij} \le 0,\ u_r, v_i \ge 0,\ r \in \{1,\ldots,s\}
\end{aligned} \tag{1}$$

where x_{ij} and y_{rj} are inputs and outputs of machines and u_r is the weight vector of inputs and v_i is the weight vector of outputs. Next, Banker et al. [39] introduced the BBC (Banker, Charnes & Cooper) model that is DEA model with a variable return to scale:

$$\begin{aligned}
&\max\left(\sum_{j=1}^{n} u_r y_{rj} - u_0\right)\\
&S.t: \sum_{j=1}^{n} v_i x_{ij} = 1,\\
&\qquad \sum_{j=1}^{n} u_r y_{rj} - \sum_{j=1}^{n} v_i x_{ij} - u_0 \le 0,\ u_r, v_i \ge 0,\\
&\qquad i \in \{1,\ldots,m\},\ r \in \{1,\ldots,s\},\ -\infty \le u_0 \le +\infty
\end{aligned} \tag{2}$$

In this article, we appraise the BCC model. The envelopment form of the BCC model is formulated as the following LP model

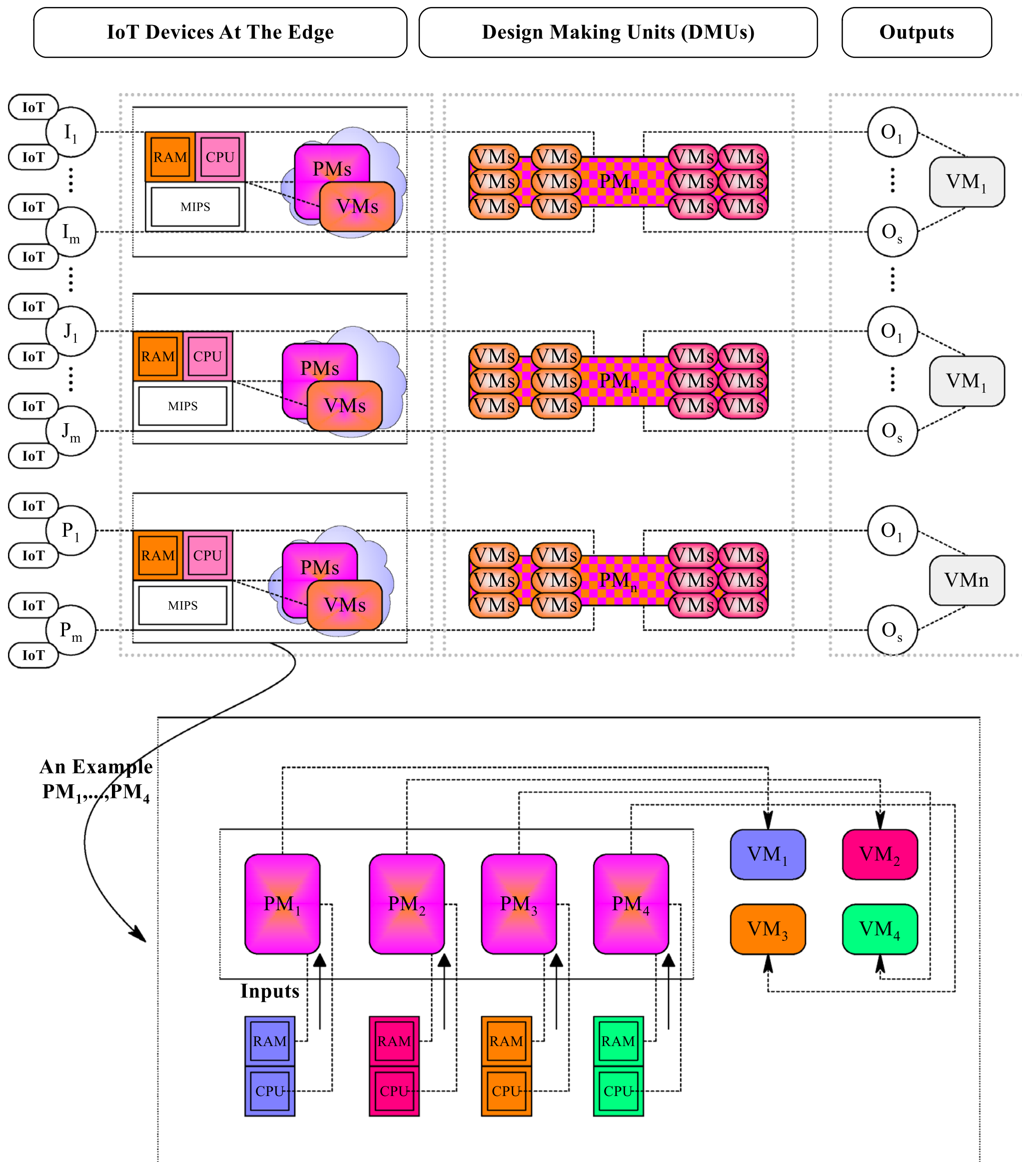


Fig. 2 Fuzzy inputs and outputs of physical machines in the cloud network

$$\begin{aligned} &\min(\theta_o) \\ &S.t: \sum_{j=1}^{n} \lambda_j x_{ij} \le \theta_o x_{io},\ i \in \{1,\ldots,m\} \\ &\quad \sum_{j=1}^{n} \lambda_j y_{rj} \ge y_{ro},\ r \in \{1,\ldots,s\}, \\ &\quad \sum_{j=1}^{n} \lambda_j = 1,\ \lambda_j \ge 0, \end{aligned} \tag{3}$$

where x ij and y_{rj} are inputs and outputs of machines.

3.2 Inverse DEA Model with the Variable Return to Scale (IBCC)

Lertworasirikul et. al. [40] proposed the Inverse BCC model (IBCC) and they have a solution approach to solve the IBCC model. They considered the output of DMU_o (machine o), y_o, change to $\beta_o = y_o + \Delta y_o$, $\Delta y_o \in R^S$, and they present the following model that gives to us the amount of input that is necessary to preserve the relative efficiency of DMU_o.

Since we have m inputs then we face a multi-goal problem, We dealt with the Dual form of the Inverse BCC model (DIBCC).

$$\begin{aligned}
&\min(x_{1o}+\Delta x_1, x_{2o}+\Delta x_2, \ldots, x_{mo}+\Delta x_m)\\
&S.t: \sum_{j=1}^{n} \lambda_j x_{ij} \le \theta_o^* \delta_i,\ i \in \{1,\ldots,m\},\\
&\quad \sum_{j=1}^{n} \lambda_j y_{rj} \ge \beta_{ro},\ r \in \{1,\ldots,s\}\\
&\quad \sum_{j=1}^{n} \lambda_j = 1,\ \lambda_j \ge 0,
\end{aligned} \tag{4}$$

where is the amount of optimal solution for the BCC model, necessary input that causes the relative performance for fluctuate inputs and is the PM is preserved.

3.3 Fuzzy Inverse Data Envelopment Analysis Method

In this section, we assume that the data are fuzzy numbers; it means that we consider a fuzzy number for inputs, outputs, inputs and outputs fluctuated. So, we should answer this question: in the IBCC model, if the outputs of the fluctuate problem are taking fuzzy values, then how many changes we should have to the inputs to maintain the relative performance of the machines? In this regard, a model with fuzzy data is presented:

$$\begin{aligned}
&\min(\tilde{\delta_1}, \tilde{\delta_2}, \ldots, \tilde{\delta_m})\\
&S.t: \sum_{j=1}^{n} \lambda_j \tilde{x_{ij}} \le \theta_o^* \tilde{\delta_i},\ i \in \{1,\ldots,m\},\\
&\quad \sum_{j=1}^{n} \lambda_j \tilde{y_{rj}} \ge \tilde{\beta_{ro}},\ r \in \{1,\ldots,s\},\\
&\quad \sum_{j=1}^{n} \lambda_j = 1, \lambda_j \ge 0
\end{aligned} \tag{5}$$

where and $\tilde{x}_{ij}$ are fuzzy inputs of physical machines, $\tilde{\delta_i}$ are fuzzy fluctuate inputs of physical machines, $\tilde{y_{rj}}$ are fuzzy outputs and $\tilde{\beta_{ro}}$ are fuzzy fluctuate outputs.

We will use an α-cut approach for solving Eq. 5. Now, suppose that inputs and outputs are a fuzzy triangular number in Eq. 5:

$$\tilde{x_{ij}} = (x_{ij}^L, x_{ij}^m, x_{ij}^U),\quad \tilde{y_{rj}} = (y_{rj}^L, y_{rj}^M, y_{rj}^U)$$
$$\tilde{\delta_i} = (\delta_i^L, \delta_i^M, \delta_i^U),\quad \tilde{\beta_{ro}} = (\beta_{ro}^L, \beta_{ro}^m, \beta_{ro}^U)$$

Now, we consider the α-cut of fuzzy triangular numbers for every constraint and objective function and replace that with Eq. 5:

$$\begin{aligned}
&\min_{k\in\{1,\ldots,m\}} \left(\left[\alpha\delta_k^m + (1-\alpha)\delta_k^L \alpha\delta_k^m + (1-\alpha)\delta_k^U\right]\right)\\
&S.t: \sum_{j=1}^{n} \lambda_j \left[\alpha x_{ij}^m + (1-\alpha)x_{ij}^L, \alpha x_{ij}^m + (1-\alpha)x_{ij}^U\right] \le A,\\
&\quad A = \theta_o^*\left[\alpha\delta_i^m + (1-\alpha)\delta_i^L, \alpha\delta_i^m + (1-\alpha)\delta_i^U\right],\\
&\quad \sum_{j=1}^{n} \lambda_j \left[\alpha y_{rj}^m + (1-\alpha)y_{rj}^L, \alpha y_{rj}^m + (1-\alpha)y_{rj}^U\right] \ge B,\\
&\quad B = \left[\alpha\beta_{ro}^m + (1-\alpha)\beta_{ro}^L, \alpha\beta_{ro}^m + (1-\alpha)\beta_{ro}^U\right],\\
&\quad \sum_{j=1}^{n} \lambda_j = 1,\ \lambda_j \ge 0
\end{aligned} \tag{6}$$

Equation 6 is a parametric problem with the parameter α. The choice of α varies depending on our manager's choice between zero and one. We will use the interval solution for solving Eq. 6, assuming that $\omega^* = [\omega_L^*, \omega_U^*]$. Hence, for ω_L^* we have:

$$\begin{aligned}
&\omega_L^* = \min(\delta_1^L, \delta_2^L, \ldots, \delta_m^L)\\
&S.t: \sum_{j=1}^{n} \lambda_j x_{ij}^L \le \theta_o^* \delta_i^L,\ i \in \{1,\ldots,m\},\\
&\quad \sum_{j=1}^{n} \lambda_j y_{rj}^L \ge \beta_{ro}^L,\ r \in \{1,\ldots,s\},\\
&\quad \sum_{j=1}^{n} \lambda_j = 1,\ \lambda_j \ge 0
\end{aligned} \tag{7}$$

and for ω_U^* we have:

$$\begin{aligned}
&\omega_U^* = \min(\delta_1^U, \delta_2^U, \ldots, \delta_m^U)\\
&S.t: \sum_{j=1}^{n} \lambda_j x_{ij}^U \le \theta_o^* \delta_i^U,\ i \in \{1,\ldots,m\},\\
&\quad \sum_{j=1}^{n} \lambda_j y_{rj}^U \ge \beta_{ro}^U,\ r \in \{1,\ldots,s\},\\
&\quad \sum_{j=1}^{n} \lambda_j = 1,\ \lambda_j \ge 0
\end{aligned} \tag{8}$$

The current models presented in Eqs. 7 and 8 consider a lower and an upper bound for Eq. 5, and an interval solution is given for the inverse DEA model in a fuzzy environment $\omega^* = [\omega_L^*, \omega_U^*]$. By solving these MOLP problems, the inputs are obtained to maintain the relative efficiency of the physical machines under evaluation.

In the solving process we used Eq. 9 for conclude θ_o^* that gives us the best condition for fuzzy DEA problems:

$$\min(\theta_o)$$
$$S.t: \sum_{j=1}^{n} \lambda_j x_{ij}^U \leq \theta_o^* x_{io}^L, i \in \{1, \ldots, m\},$$
$$\sum_{j=1}^{n} \lambda_j y_{rj}^L \geq y_{ro}^U,\ r \in \{1, \ldots, s\}, \tag{9}$$
$$\sum_{j=1}^{n} \lambda_j = 1,\ \lambda_j \geq 0$$

Where x_{ij}^U is upper bound of fuzzy inputs for all physical machines, x_{io}^L is lower bound of inputs for PM_o, y_{rj}^L is lower bound of fuzzy outputs for all physical machines, and y_{ro}^U is upper bound of fuzzy output for PM_o, that are given by α-cut approaches (Fig. 3).

Algorithm 1

```
Start
Perform fuzzy inputs and output population of PM
RÃM_i^α ← [RAM_i^l, RAM_i^u]
CP̃U_i^α ← [CPU_i^l, CPU_i^u]
VM̃_i^α ← [VM_i^l, VM_i^u]
Perform fuzzy fluctuate population of VM
ṽ_i^α ← [v_i^l, v_i^u]
for α ∈ {0.3, 0.5, 0.7} do
    if θ_o^* = 1 then
        PM_o is efficient
    else
        PM_o is non-efficient
Select efficient machines for different α_i
Calculate lower bound of ω_L^* by Equation 7
Calculate upper bound of ω_U^* by Equation 8
Conclude ω^*
[R_i^l, R_i^u] ← [RAM_i^l + Δx_1, RAM_i^u + Δx_2]
[C_i^l, C_i^u] ← [CPU_i^l + Δx_1, CPU_i^u + Δx_2]
Get fluctuate inputs
Preserve efficiency of PM_o
End
```

3.4 A Numerical Illustrative Example

In this subsection, we assume four physical machines each have two Ram and CPU inputs and one virtual machine output. The fuzzy values of each of the inputs and outputs are given in Table 2.

We would like to achieve the number of input changes by changing the output so that the efficiency of the machines is maintained. Firstly, we conclude θ_o^* for $o \in \{1, \ldots, 4\}$, the results of the α-cut method (the result is considered for different α) are shown in Table 3.

The results on Table 3 show that PM_1,PM_2, PM_4 are efficiency units and PM_3 is non-efficiency with all of the alphas.

We will use the weighting method to solve a multi-objective programming problem, and so we consider weight vector as $(w_1, w_2) = (1/2, 1/2)$. Suppose that the output of unites PM_1,PM_2, $PM_3$and PM_4 are change to (68, 73, 75), (82, 85, 89), (72, 75, 77) and (98, 100, 101), the result of the proposed method are presented in Table 4.

For example for an efficient machine PM_1 with $\alpha = 0.5$, the change rate of RAM and CPU must be [(0.527, 0.764), (0.964, 0.964)] until the efficiency of PM_1 is maintained.

In Sect. 3, we present a fuzzy optimization model for selecting efficient physical machines with respect to fuzzy inputs and outputs. In this model, when the fuzzy values of RAM and CPU fluctuate, the proposed model maintains the performance of physical machines by controlling the oscillations of virtual machines. Also, the fuzzy Markov method is used in the proposed algorithm for allocating input requests to virtual machines in the network and in the allocation process, the α-cut method is used for fuzzy data. The proposed method (FIMDEAP) with the proper choice of physical and virtual machines maintains the balance of load and resources in the scheduling process, which ultimately leads to the creation of green calculations in the network. The detail description about fuzzy (Basic fuzzy concepts) will present in Appendix.

4 Simulation and Results

The simulation of this research has been done in cloudsim, which includes physical machine, virtual machine and data centers having the ability to search virtual machines. A broker is defined in cloudsim, using it we can check the relationship between the generated queue and physical and virtual machines in the data center. To do so, we have used the proposed algorithm and it is related to the base paper including meta-heuristic and fuzzy ones. In addition, the intended data centers include virtual machines with different string and binary models, meaning that the requests that enter the network can be part of a large task or a small one.

The simulation results of the proposed FIMDEAP method are compared with PSO + ACO and FBPSO + FBACO methods [4] in terms of five criteria of Execution cost, Response time, Energy consumption, Gain of cost and Makespan.

After applying the benchmark data sets to the algorithm, a large load is imposed on the network, which we aim to

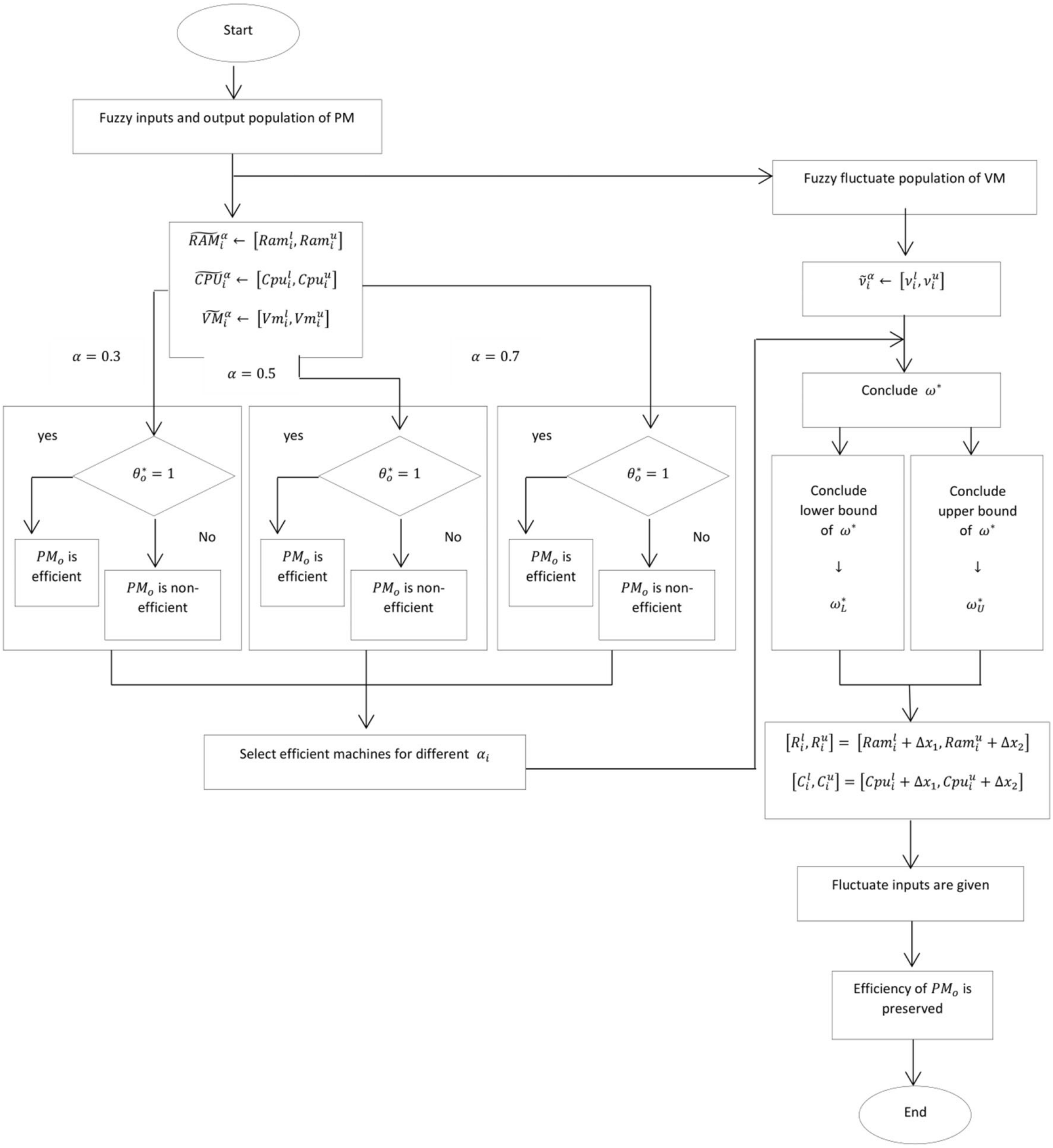


Fig. 3 Details of selecting physical and virtual machines in cloud environment

Table 2 Units with fuzzy inputs and output

PMs	RAM	CPU	VM
PM_1	(1,2,3)	(12,13,15)	(69,72,73)
PM_2	(8,10,13)	(21,22,24)	(81,85,88)
PM_3	(6,7,9)	(18,19,20)	(72,74,75)
PM_4	(15,17,18)	(33,34,36)	(97,99,100)

examine the execution cost. As can be seen in Fig. 4, by increasing the workload on tasks, the proposed method has a lower execution cost due to the appropriate resource allocation. In PSO and ACO, an initial population of the

Table 3 Units with fuzzy inputs and output

Units	Efficiency		
	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.7$
PM_1	$\theta_1^* = 1$	$\theta_1^* = 1$	$\theta_1^* = 1$
PM_2	$\theta_2^* = 1$	$\theta_2^* = 1$	$\theta_2^* = 1$
PM_3	$\theta_3^* = 0.995$	$\theta_3^* = 0.751$	$\theta_3^* = 0.750$
PM_4	$\theta_4^* = 1$	$\theta_4^* = 1$	$\theta_4^* = 1$

algorithmic environment is generated using mathematical equations and calculating the optimal P-value and G-value that are extracted from each section. This initial population

Table 4 Amount of input changes to preserve the relative efficiency of machines

Units	$\alpha = 0.3$				$\alpha = 0.5$				$\alpha = 0.7$			
	ω_L^*		ω_U^*		ω_L^*		ω_U^*		ω_L^*		ω_U^*	
	Δx_1	Δx_2	Δx_1	Δx_2	Δx_1	Δx_2	Δx_1	Δx_2	Δx_1	Δx_2	Δx_1	Δx_2
PM_1	4.398	5.278	5.628	5.233	0.527	0.764	0.964	0.964	0.215	0.308	0.822	0.86
PM_2	0.227	0.545	0.311	0.656	−0.645	0.927	0.231	0.462	−0.7	1	0.143	0.269
PM_3	0.15	0.246	0.552	0.636	−2.396	1.195	−1.675	2.138	−2.279	1.31	−3.213	−3.513
PM_4	0.325	0.779	0.444	0.937	0.527	0.764	0.462	0.923	0.538	0.769	0.478	0.896

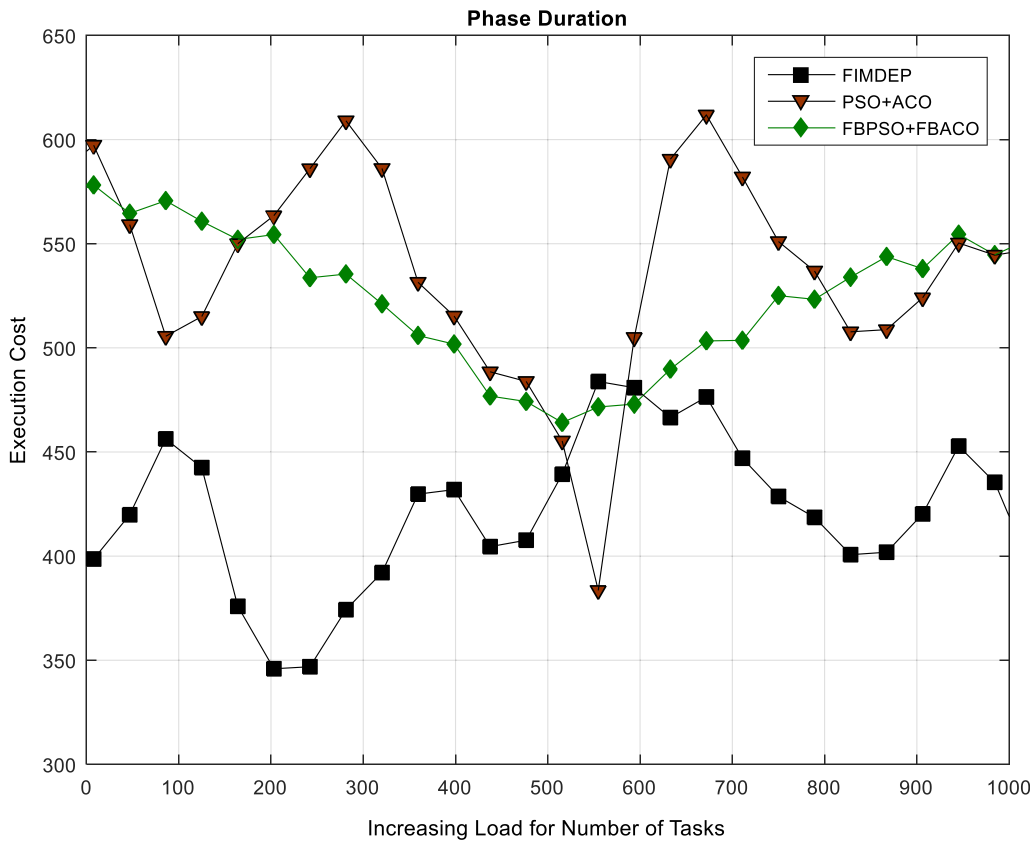


Fig. 4 Details of the cost of task execution against the increase in workload in benchmark 512

selects the optimal cases as a meta-heuristic method, but due to the fact that the initial population is generated randomly using the properties of the methods, only in some cases optimization is done. In PSO + ACO method, due to the fact that the initial population is generated randomly, the amount of execution cost tolerances increases in some cases. Also, the amount of operating cost of FBPSO + FBACO method is the same as that of PSO + ACO method, with the difference in each time gap, fuzzy mode is applied to the algorithm, which causes the extreme fluctuations caused by the randomness of the initial population to be controlled in the diagram of the operating cost (as can be seen in Fig. 4, when the number of requests is 290, the execution cost is very high, but when the number of requests is 570, the execution cost improves and again by increasing the number of requests to 640, PSO + ACO method imposes a high cost on the system but these fluctuations are modified in the fuzzy method). However, in the proposed FIMDEAP method, due to the fact that changes in fuzzy inputs has been done in such a way that the performance of physical machines is maintained against the fluctuations of virtual machines, so choosing an efficient physical machine to perform tasks has reduced execution costs. The simulation process is repeated 100 times and the results are the average of 100 simulations.

Response time in the process of assigning tasks to virtual machines under the management of physical machines in the data center includes waiting time in a queue, task

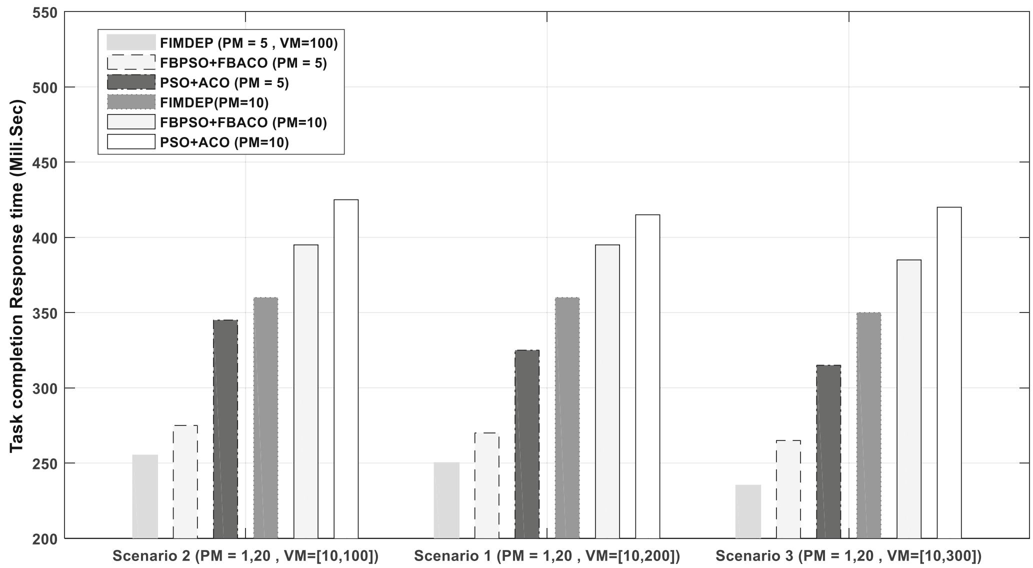


Fig. 5 Response time in different scenarios with the specified virtual machine and physical machine

processing time by the algorithm, time of assigning tasks to the data center, time of assigning tasks from data center to physical machines and finally the time required to allocate requests to virtual machines. The comparison of Response time between the proposed method and PSO + ACO and FBPSO + FBACO methods has been done in three different scenarios each with different modes for physical and virtual machines (for example, for Scenario 1, the number of physical machines is between 1 and 20 and the number of virtual machines is between 10 and 200, that can be randomly selected in these intervals). In each scenario, given that the number of physical machines is fixed, if we assign the requests from the queue to the virtual machines correctly, we can determine which one has the shortest response time. Both in the case of a small number and in the case of a large number of virtual machines, the presented fuzzy pattern has less execution time in some cases and needs more time in some cases, But in general, by calculating the average response time in three different scenarios, it is observed that the proposed method gives us less response time than other methods in the allocation process, which leads to impose lower costs on the system. As shown in Fig. 5, when the PSO + ACO algorithm is used, it generates a higher response time due to the random generation of its initial population. Also, the FBPSO + FBACO method, which uses both the meta-heuristic algorithm and the fuzzy pattern, causes a delay in the allocation process, but works better than the PSO + ACO method. Comparisons between methods with impact coefficients of 5 and 10 have been made for physical machines, and in both cases the response time value of the proposed method was lower than other methods.

Energy consumption refers to the energy consumed to allocate a task in the cloud network. Now we want to know how energy consumption changes as the number of tasks increases. By calculating the Energy Consumption criterion as shown in Fig. 6, through the time and increasing the number of tasks, the same results as the PSO + ACO and FBPSO + FBACO methods are obtained, so their diagram is the same but in some cases the proposed method performed better, and in other ones PSO + ACO and FBPSO + FBACO methods performed better. For the number of requests between 200 and 700, almost all three methods gave the same results, for the number between 700 and 1300, the FBPSO + FBACO method worked better, for the requests between 1300 and 2100, all methods gave the same results, and finally for the number of requests between 2100 and 2200, the proposed method performed better than other methods.

By investigating the details, we found that the proposed FIMDEAP method had a 2.12% improvement in performance compared to the FBACO + FBPSO method, and 3.31% improvement in Energy Consumption compared to the PSO + ACO method. This cost (energy consumed) includes all tasks including successfully executed, failed, or running in the system.

Considering that the different methods presented in this research have the same energy consumption and on the other hand, because a fixed benchmark has been applied to the methods, it is expected that the cost of the methods is their execution cost and the gain of cost is what we expected before. Comparing the proposed method with other methods in terms of Gain of Cost, it should be noted that assigning tasks to virtual machines in the proposed

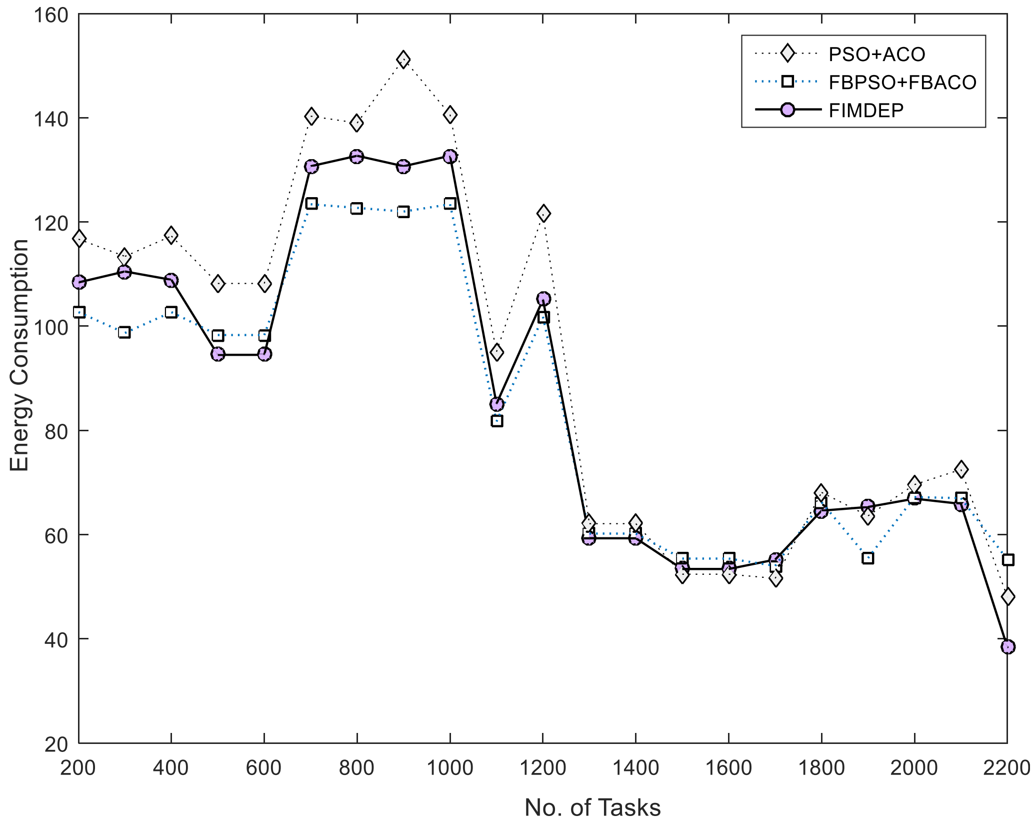


Fig. 6 Energy consumption of the network for the number of available tasks

method is less expensive than other methods. Due to the fact that in the proposed method, the assigned tasks are processed in physical machines whose efficiency is maintained and the fluctuations of virtual machines and fuzzy inputs are controlled to maintain performance, the proposed FIMDEAP method has a lower gain of cost than other methods. As can be seen in Fig. 7, because of producing different initial populations, the PSO + ACO method has a poorer performance. Also, the FBPSO, FBACO and FBPSO + FBACO methods have the same performance, but in the proposed method Gain of Cost improves by increasing physical machines.

One of the important criteria for evaluating the proposed method is Makespan, and if we have a proper makespan, we will have less delay and therefore a proper execution cost. Comparing the proposed method with other methods in Makespan criterion, it is observed that FIMDEAP method gives less Makespan. As well, when the number of requests increases, the proposed method has better performance compared to PSO + ACO and FBPSO + FBACO methods. Calculating the Makespan arithmetic mean for a number of different tasks indicated the better performance of the proposed method. As shown in Fig. 8, by increasing the number of tasks by 10 times, a lower average Makespan is observed for the proposed method. The values obtained for Makespan show that, as we expected, the proposed algorithm produced good values for Response times. To test the Makespan criterion, a set of 512 $\times$ 16 benchmark data sets was used.

5 Conclusion

In Cloud, Fog and IoT networks, due to the large number of heterogeneous devices and sensors that are interconnected, we are faced with large amounts of data, so resource allocation and request allocation with less energy consumption is a vital problem (Fig. 9). In this paper, we could improve the task scheduling process by introducing a new fuzzy pattern. In some cases, the proposed method performed better and in some other cases, compared methods performed better. But by investigating previous methods and the proposed method, it is clear that the proposed method has the appropriate efficiency and performance. we reached our goals that were optimizing energy, response time and Makespan and achieving green computing. The proposed FIMDEAP method, which was a combination of the of FIDEA and FMDP, improved the performance of the Task Scheduling process by selecting efficient physical machines and appropriately assigning requests to virtual

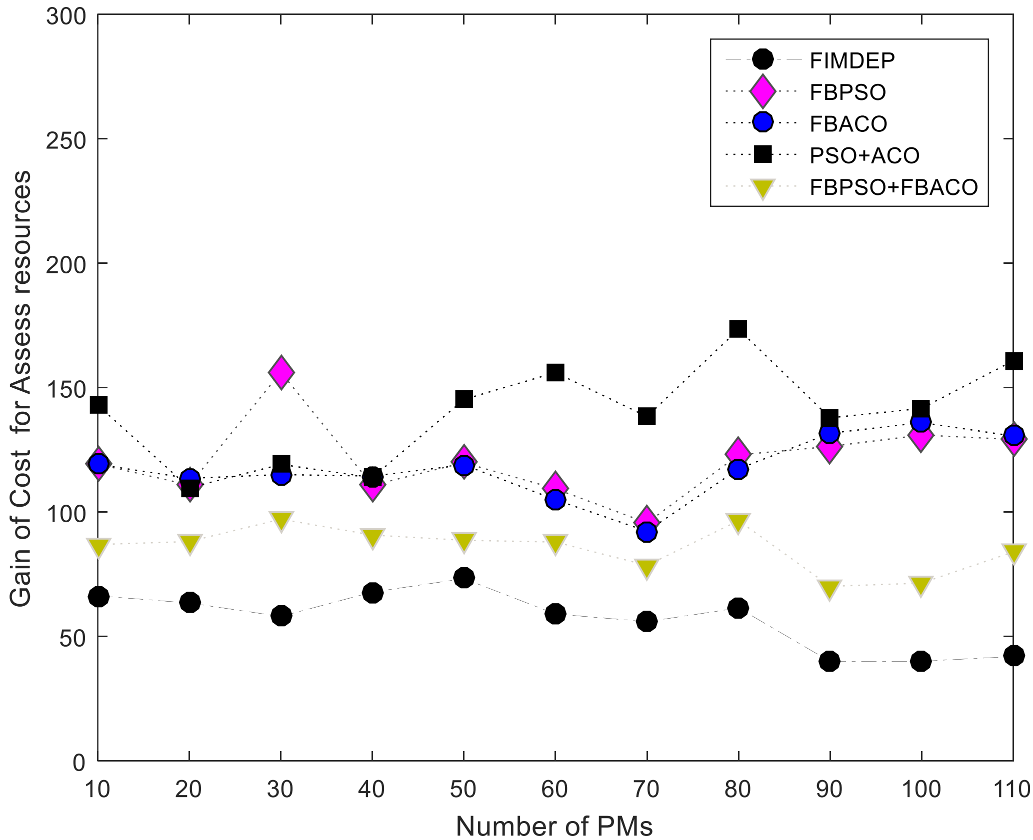


Fig. 7 Profitability of resources for virtual machines in the network

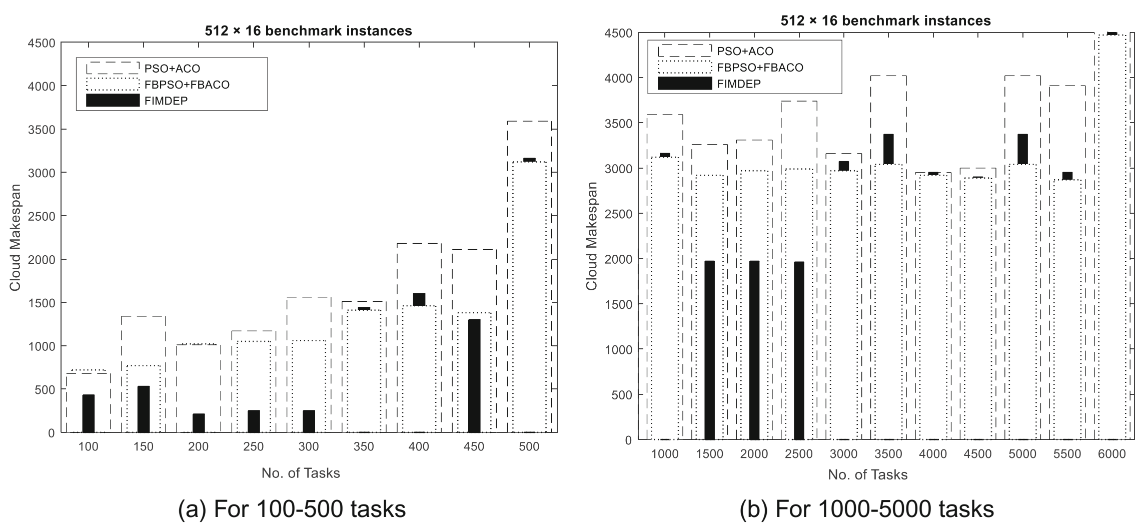


Fig. 8 Make span for different numbers of tasks

machines. We compared the proposed method with PSO + ACO and FBPSO + FBACO methods. Simulation results showed that the proposed method had improvements in Response time, Execution cost, Gain of cost, Makespan and Energy consumption. In this study, in addition to providing a mathematical optimization model, algorithms and flowcharts of the proposed method are also given. Given that the proposed model is fuzzy, we recommend

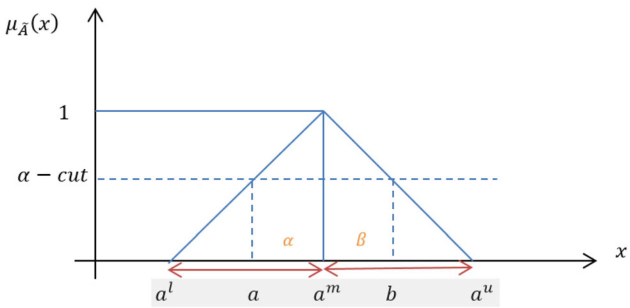


Fig. 9 The membership function and α-cut of a fuzzy number

researchers of this field to use other fuzzy methods and fuzzy ranking methods for machine selection process as future work. machine learning methods and request forecasting based methods are recommended for allocation process, as well as examining the overlap and conflict between requests and the queue generated to select virtual machines can be adopted for future work.

Funding Open access funding provided by FCT|FCCN (b-on).

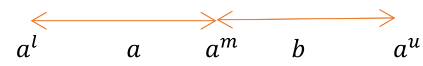
Appendix A

Basic Fuzzy Concepts

Definition 1 Let $\Re$ be the real line. A fuzzy set $\tilde{A}$ in $\Re$ is defined to be a set of ordered pairs $\tilde{A} = \{(x, \mu_{\tilde{A}}(x)), x \in \Re\}$, where $\mu_{\tilde{A}}(x)$ is called the membership function for the fuzzy set. The membership function maps each element of $\Re$ to a membership value between 0 and 1.

Definition 2 Assume $\tilde{A}$ is a fuzzy set and $\alpha \in (0, 1]$, then an α-cut of $\tilde{A}$ is defined as $\{x | x \in \Re, \mu_A(x)) \geq \alpha\}$, and we briefly denote it as $\tilde{A}_\alpha$.

Definition 3 The number $\tilde{A}$ is a triangular fuzzy number, if its membership function is as follows:



A triangular fuzzy number is shown as $\tilde{A} = (a^l, a^m, a^u)$, where a^m, a^l, a^u are respectively, the core, the lower, and the upper limits of support of the triangular fuzzy number $\tilde{A}$.

Definition 4 The α-cut of fuzzy triangular number $\tilde{A} = (a^l, a^m, a^u)$ is as follows:

$$\tilde{A}_\alpha = [\alpha a^m + (1 - \alpha)a^l, \ \alpha a^m + (1 - \alpha)a^u] \tag{10}$$

Data availability Data will be made available on request.

References

1. Javadpour, A., Wang, G.: cTMvSDN: improving resource management using combination of Markov-process and TDMA in software-defined networking. J. Supercomput. **78**, 3477–3499 (2022). https://doi.org/10.1007/s11227-021-03871-9
2. Zavieh, H., Javadpour, A., Li, Y., Ja'fari, F., Nasseri, S.H., Rostami, A.S.: Task processing optimization using cuckoo particle swarm (CPS) algorithm in cloud computing infrastructure. Clust. Comput. **26**(1), 745–769 (2023)
3. Juarez, F., Ejarque, J., Badia, R.M.: Dynamic energy-aware scheduling for parallel task-based application in cloud computing. Futur. Gener. Comput. Syst. **78**, 257–271 (2018)
4. Amulu, L.M., Ramraj, R.: Combinatorial meta-heuristics approaches for DVFS-enabled green clouds. J. Supercomput. **76**(8), 5825–5834 (2020)
5. Javadpour, A., Wang, G., Rezaei, S.: Resource management in a peer to peer cloud network for IoT. Wirel. Personal Commun. **115**, 2471–2488 (2020). https://doi.org/10.1007/s11277-020-07691-7
6. Javadpour, A., Wang, G., Rezaei, S., Chend, S.: Power Curtailment in Cloud Environment Utilising Load Balancing Machine Allocation. In: 2018 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), pp. 1364–1370 (2018)
7. Mirmohseni, S.M., Tang, C., Javadpour, A.: FPSO-GA: a fuzzy metaheuristic load balancing algorithm to reduce energy consumption in cloud networks. Wirel. Pers. Commun. **127**(4), 2799–2821 (2022)
8. Manasrah, A.M., Ba Ali, H.: Workflow scheduling using hybrid GA-PSO algorithm in cloud computing. Wirel. Commun. Mob. Comput. **2018**, 1934784 (2018)

9. Mirmohseni, S.M., Javadpour, A., Tang, C.: LBPSGORA: create load balancing with particle swarm genetic optimization algorithm to improve resource allocation and energy consumption in clouds networks. Math. Probl. Eng. **2021**, 15 (2021)
10. Javadpour, A., Nafei, A., Ja'fari, F., Pinto, P., Zhang, W., Sangaiah, A.: An intelligent energy-efficient approach for managing IoE tasks in cloud platforms. J. Ambient Intell. Hum. Comput. **14**, 3963–3979 (2023). https://doi.org/10.1007/s12652-022-04464-x
11. Javadpour, A., Sangaiah, A., Pinto, P., Ja'fari, F., Zhang, W., Abadi, A.M.H., Ahmadi, H.: An energy-optimized embedded load balancing using DVFS computing in cloud data centers. Comput. Commun. **197**, 255–266 (2023)
12. Javadpour, A., Ja'fari, F., Pinto, P., Zhang, W.: Mapping and embedding infrastructure resource management in software defined networks. Clust. Comput. **26**, 461–475 (2023). https://doi.org/10.1007/s10586-022-03789-8
13. Lakzaei, M., Sattari-Naeini, V., Sabbagh Molahosseini, A., Javadpour, A.: A joint computational and resource allocation model for fast parallel data processing in fog computing. J. Supercomput. **78**, 12662–12685 (2022). https://doi.org/10.1007/s11227-022-04374-x
14. Sangaiah, A., Javadpour, A., Pinto, P., Ja'fari, F., Zhang, W.: Improving quality of service in 5G resilient communication with the cellular structure of smartphones. ACM Trans. Sens. Netw. **18**, 9 (2022). https://doi.org/10.1145/3512890
15. Panda, S.K., Jana, P.K.: SLA-based task scheduling algorithms for heterogeneous multi-cloud environment. J. Supercomput. **73**(6), 2730–2762 (2017)
16. Sivaram, M., et al.: Secure storage allocation scheme using fuzzy based heuristic algorithm for cloud. J. Ambient Intell. Hum. Comput. **12**, 1–9 (2021)
17. Kalnoor, G., Subrahmanyam, G.: A review on applications of Markov decision process model and energy efficiency in wireless sensor networks. Procedia Comput. Sci. **167**, 2308–2317 (2020)
18. Huo, J., Zhang, J., Chan, F.T.S.: A fuzzy control system for assembly line balancing with a three-state degradation process in the era of Industry 4.0. Int. J. Prod. Res. **58**(23), 7112–7129 (2020)
19. Li, J., Shi, W., Zhang, N., Shen, X.: Delay-aware VNF scheduling: a reinforcement learning approach with variable action set. IEEE Trans. Cogn. Commun. Netw. **7**(1), 304–318 (2020)
20. Gao, Z., Hao, W., Zhang, R., Yang, S.: Markov decision process-based computation offloading algorithm and resource allocation in time constraint for mobile cloud computing. IET Commun. **14**(13), 2068–2078 (2020)
21. He, Y., Xing, L., Chen, Y., Pedrycz, W., Wang, L., Wu, G.: A generic Markov decision process model and reinforcement learning method for scheduling agile earth observation satellites. IEEE Trans. Syst. Man. Cybern. Syst. **99**, 1 (2020)
22. Luo, Sh.: Dynamic scheduling for flexible job shop with new job insertions by deep reinforcement learning. Appl. Soft Comput. J. **91**, 106208 (2020)
23. Meena, V., Gireesha, O., Krithivasan, K., Shankar Sriram, V.S.: Fuzzy simplified swarm optimization for multisite computational offloading in mobile cloud computing. J. Intell. Fuzzy Syst. **1**, 1–13 (2020)
24. Hamdy, N., Aboutabl, A.E., ElHaggar, N., Mostafa, M.S.M.: Dynamic task scheduling method in cloud computing environment using optimized neural network. Int. J. Futur. Comput. Commun. **6**(4), 143–147 (2017)
25. Xu, H., Li, R., Zeng, L., Li, K., Pan, C.: Energy-efficient scheduling with reliability guarantee in embedded real-time systems. Sustain. Comput. Informatics Syst. **18**, 137–148 (2018)
26. Mei, X., Wang, Q., Chu, X., Liu, H., Leung, Y.W., Li, Z.: Energy-aware task scheduling with deadline constraint in DVFS-enabled heterogeneous clusters. IEEE Trans. Parallel Distrib. Syst. **33**(12), 1–17 (2021)
27. Peng, Z., Barzegar, B., Yarahmadi, M., Motameni, H., Pirouzmand, P.: Energy-aware scheduling of workflow using a heuristic method on green cloud. Sci. Program. **2020**, 14 (2020)
28. Mangalampalli, S., Swain, S.K., Mangalampalli, V.K.: Prioritized energy efficient task scheduling algorithm in cloud computing using whale optimization algorithm. Wirel. Pers. Commun. **126**, 1–17 (2021)
29. Tariq, U.U., Ali, H., Liu, L., Hardy, J., Kazim, M., Ahmed, W.: Energy-aware scheduling of streaming applications on edge-devices in IoT-based healthcare. IEEE Trans. Green Commun. Netw. **5**(2), 803–815 (2021)
30. Hamzeloo, S., Jahromi, M.Z.: Decentralized incremental fuzzy reinforcement learning for multi-agent systems. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. **28**(01), 79–98 (2020)
31. Wu, C., Li, W., Wang, L., Zomaya, A.: An evolutionary fuzzy scheduler for multi-objective resource allocation in fog computing. Futur. Gen. Comput. Syst. **117**, 498–509 (2021)
32. Li, C., Zuo, X., Mohammed, A.: A new fuzzy-based method for energy-aware resource allocation in vehicular cloud computing using a nature-inspired algorithm. Sustain. Comput. **36**, 100806 (2022)
33. Yazdinejad, A., Dehghantanha, A., Parizi, R., Epiphaniou, G.: An optimized fuzzy deep learning model for data classification based on nsga-ii. Neurocomputing. **522**, 116–128 (2023)
34. Shukla, A., Nath, R., Muhuri, P., Lohani, Q.: Energy efficient multi-objective scheduling of tasks with interval type-2 fuzzy timing constraints in an Industry 4.0 ecosystem. Eng. Appl. Artif. Intell. **87**, 103257 (2020)
35. Sangaiah, A., Javadpour, A., Pinto, P., Rezaei, S., Zhang, W.: Enhanced resource allocation in distributed cloud using fuzzy meta-heuristics optimization. Comput. Commun. **209**, 14 (2023)
36. Vemireddy, S., Rout, R.: Fuzzy reinforcement learning for energy efficient task offloading in vehicular fog computing. Comput. Netw. **199**, 108463 (2021)
37. Mokni, M., Yassa, S., Hajlaoui, J., Omri, M., Chelouah, R.: Multi-objective fuzzy approach to scheduling and offloading workflow tasks in Fog-Cloud computing. Simul. Model. Pract. Theory. **123**, 102687 (2023)
38. Charnes, A., Cooper, W.W., Rhodes, E.: Measuring the efficiency of decision making units. Eur. J. Oper. Res. **2**(6), 429–444 (1978)
39. Banker, R.D., Charnes, A., Cooper, W.W.: Some models for estimating technical and scale inefficiencies in data envelopment analysis. Manag. Sci. **30**(9), 1078–1092 (1984)
40. Lertworasirikul, S., Charnsethikul, P., Fang, S.C.: Inverse data envelopment analysis model to preserve relative efficiency values: the case of variable returns to scale. Comput. Ind. Eng. **61**(4), 1017–1023 (2011)

Hadi Zavieh received the Ph.D. degree in applied mathematics from the Academy of Mathematics and Systems Science, Chinese Academy of Sciences, China, in 2017. He is currently a Researcher with the College of Mathematics and Information Science, Guangzhou University, China. His research interests include Cloud, secure multi-party computing, and outsourced computing. Moreover, he has reviewed several papers for prestigious journals such as Springer's Cluster Computing journals and Journal of Supercomputing.

Amir Javadpour obtained his M.Sc. degree in Medical Information Technology Engineering from University of Tehran, Iran, in 2014. From Guangzhou University, China, he received a Ph.D. in Computer Science/Mathematics/Cybersecurity. In addition, he has published papers with his colleagues in highly ranked journals and several ranked conferences in several topics including Cloud Computing, Software-Defined Networking (SDN), Big Data, Intrusion Detection Systems (IDS), and the Internet of Things (IoT), Moving Target Defence (MTD), Machine Learning (ML) and optimisation algorithms. Additionally, he reviewed papers for several reputable venues such as IEEE Transactions on Cloud Computing, IEEE Transactions on Network Science and Engineering, ACM Transactions on Internet Technology, the Journal of Supercomputing, several journals of Springer and Elsevier, etc. and also, he is the Technical Program Committee (TCP) Member of various conferences.

Forough Ja'fari is a Senior Researcher in cybersecurity and computer science. She received her Bachelor's degree from Sharif University of Technology and her Master's degree in Computer Network Engineering from Yazd University, Iran. She is a visiting scholar researcher at Guangzhou University, China. Cloud Computing, Software-Defined Networking (SDN), Intrusion Detection Systems (IDS), Internet of Things (IoT), Moving Target Defence (MTD), and Machine Learning are some of her research interests. She is currently a Guest Editor (GE) of Cluster Computing (CLUS) Journal, as well as a reviewer for several journals and conferences.

Arun Kumar Sangaiah received the masters of engineering degree from Anna University, Chennai, India, and the Ph.D. degree from VIT University, Vellore, India. He is currently a Professor with the School of Computing Science and Engineering, VIT University. He is holding a Visiting Professor position in various universities around the globe. His outstanding scientific production spans more than 300 contributions published in high standard ISI journals, such as IEEE Communication Magazine, IEEE Systems Journals, and IEEE Internet of Things (IoT). In addition, he has authored/edited eight books (Elsevier, Springer, etc.) and edited several special issues in reputed ISI journals, such as IEEE Communication Magazine, IEEE Transactions on Industrial Informatics, IEEE Internet of Things, ACM Transaction on Intelligent Systems and Technology, etc. He has also registered one Indian patent in the area of computational intelligence. His research interests include E-learning, machine learning, software engineering, computational intelligence. He is an Editorial Board Member and an Associate Editor for many reputed ISI journals. Furthermore, he is the recipient of many awards that include India-Top-10 Researcher award, Chinese Academy of Science-PIFI overseas visiting scientist award.

Adam Słowik (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in computer engineering and electronics, in 2001, and the Ph.D. degree in electronics with distinction from the Department of Electronics and Computer Science, Koszalin University of Technology, Koszalin, Poland, in 2007, respectively, and the Dr. habil. degree in computer science (intelligent systems) from the Department of Mechanical Engineering and Computer Science, Czestochowa University of Technology, Czestochowa, Poland, in 2013., Since October 2013, he has been an Associate Professor with the Department of Electronics and Computer Science, Koszalin University of Technology. He is an Author or Co-Author of more than 100 refereed articles in international journals, two books, and conference proceedings, including one invited talk. His research interests include soft computing, computational intelligence, and, particularly, bio-inspired optimization algorithms and their engineering applications., Dr. S^{3}owik is a Reviewer for many international scientific journals. He is an Associate Editor for the IEEE Transactions on Industrial Informatics. He is a Member of the program committees of several important international conferences in the area of artificial intelligence and evolutionary computation. He was a recipient of one Best Paper Award at the IEEE Conference on Human System Interaction in 2008.(Based on document published on 22 June 2021).