People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

Mustapha Stambouli University

Mascara

Faculty of Exact Sciences

Computer Science Department

DOCTORAL THESIS

Option: Networks and Distributed Systems

Entitled:

Energy-Constrained Resource Allocation and Scheduling in Cloud Computing

Presented by: Yamina Mehor

On: 14/07/2025

Before the Committee:

President:	Hamza Teggar	MCA	University of Mascara
Examiner:	Mahmoud Zennaki	MCA	University of Usto, Oran
Examiner:	Meriem Meddeber	MCA	University of Mascara
Examiner:	Sabrina Abid	MCA	University of Mascara
Supervisor:	Mohammed Rebbah	Pr.	University of Mascara

Academic Year: 2024-2025

Dedication

To my very dear parents.

To my brothers.

To everyone who encouraged me.

ABSTRACT

In virtualized cloud computing systems, energy reduction is a major concern since it can provide many major benefits, such as reducing operating costs, increasing system efficiency, and protecting the environment. Typically, customers submit their applications with millions of tasks executed in cloud data centers by thousands of high-performance servers installed. The cloud offers a variety of services through virtual machines (VMs). These latter usually consume a large amount of energy. Such energy consumption increases the cost of electricity and has a negative environmental effect. To maintain a better performance of the services offered by data centers and a reasonable energy consumption. A detailed study of the behavior of these systems is essential for the design of efficient optimization solutions to reduce energy consumption. This thesis work focuses on the development of a task scheduling model in order to minimize the energy consumption of data center resources while meeting customer requirements for quality of services.

Keywords: *Cloud Computing*, *Data center*, *Task scheduling*, *VM allocation*, *Energy consumption*.

Résumé

Dans les systèmes de Cloud Computing virtualisés, la réduction d'énergie est une préoccupation majeure car elle peut offrir de nombreux avantages majeurs, tels que la réduction des coûts de fonctionnement, l'augmentation de l'efficacité du système et la protection de l'environnement. Généralement, les clients soumettent leurs applications avec des millions de tâches exécutées dans les centres de données Cloud par des milliers de serveurs hautes performances installés. Le Cloud offre une variété de services via des machines virtuelles (MV). Ces derniers consomment généralement une grande quantité d'énergie. Une telle consommation d'énergie augmente le coût de l'électricité et a un effet environnemental négatif. Pour maintenir une bonne performance des services offerts par des centres de données, et une consommation énergétique raisonnable, une étude détaillée du comportement de ces systèmes est essentielle pour la conception des solutions d'optimisation efficaces permettant de réduire la consommation énergétique. Ce travail thèse s'intéressera au développement d'un modèle d'ordonnancement des tâches dans le but de minimiser la consommation énergétique des ressources des centres de données tout en répondant aux exigences des clients pour la qualité de services.

Mots-clés : Cloud Computing, Centre de données, Ordonnancement des tâches, Allocation de machines virtuelles, Consommation d'énergie.

ملخص

في أنظمة الحوسبة السحابية الافتراضية، يعد تقليل استهلاك الطاقة من القضايا الرئيسية لأنه يمكن أن يوفر العديد من الفوائد المهمة، مثل تقليل تكاليف التشغيل، وزيادة كفاءة النظام، وحماية البيئة. بشكل عام، يقوم العملاء بتقديم تطبيقاتهم التي تحتوي على ملايين المهام التي تُنفذ في مراكز البيانات السحابية بواسطة آلاف الخوادم عالية الأداء المثبتة.

والتي تستهلك عادةً كميات كبيرة من الطاقة تقدم الحوسبة السحابية مجموعة متنوعة من الخدمات عبر الآلات الافتراضية.

يزيد هذا الاستهلاك من تكلفة الكهرباء وله تأثير بيئي سلبي.

للحفاظ على أداء جيد للخدمات المقدمة من قبل مراكز البيانات واستهلاك طاقة معقول، فإن دراسة تفصيلية لسلوك هذه الأنظمة أمر ضروري لتصميم حلول فعالة لتحسين الأداء وتقليل استهلاك الطاقة.

ستركز هذه الأطروحة على تطوير نموذج لجدولة المهام بهدف تقليل استهلاك الطاقة لموارد مراكز البيانات مع تلبية متطلبات العملاء من حيث جودة الخدمات.

الكلمات المفتاحية: الحوسبة السحابية، مركز البيانات، جدولة المهام، تخصيص الآلات الافتراضية، استهلاك الطاقة

ACKNOWLEDGEMENT

First of all, I would like to thank Allah for giving me patience and strength to realize this work.

I sincerely thank my supervisor Pr. Mohammed REBBAH for his continuous guidance, patience and motivation throughout my PhD studies. I greatly appreciate his support and guidance.

I wish to express my profound gratitude to Pr. Omar SMAIL for having given me the opportunity to work with them.

My profound appreciation to the president of the jury: Dr. Hamza TEGGAR and to the examiners: Dr. Mahmoud ZENNAKI, Dr. Meriem MEDDEBER and Dr. Sabrina ABID for the valuable criticisms and suggestions.

Finally, I would like to thank my full family and all my friends for the continuous support and encouragement.

CONTENTS

List of Figures

List of Tables

CHAPTER 1

Introduction

Cloud computing has rapidly emerged as a successful paradigm for providing IT infrastructure, resources, and services on a pay-per-use basis over the past few years. The wide use of cloud and virtualization technologies has resulted in the establishment of large-scale data centers that offer cloud services. This evolution intensifies energy consumption, that in turn causes the costs of data center ownership and increases the carbon footprint. For these reasons, the significance of energy efficiency in data centers and Cloud is on the rise. The importance of minimizing energy consumption in Clouds is underscored by the fact that electricity consumption is expected to increase by 76% from 2007 to 2030 (Liu et al. (2020)), with data centers contributing a significant portion of this increase. The average data center is estimated to consume as much energy as 25,000 households, as per the Gartner report (Ghribi (2014)). Additionally, the McKinsey report states that ”the total estimated energy bill for data centers in 2010 is 11.5 billion, and energy costs in a typical data center double every five years”. Energy-efficient data center solutions have emerged as one of the most significant challenges in response to the substantial amount of energy required to operate data centers and the electronic detritus they generate. Idle electricity is a critical contributor to energy inefficiency in data centers, as it is squandered when resources are not in use. Furthermore, the issue of low resource utilization is compounded by the fact that servers are perpetually powered on, even when they are not in use, and they continue to utilize up to 70% of their peak power. In order to resolve these issues, it is required to eliminate power waste, enhance efficiency, and modify the allocation of resources. This thesis concentrates on the development of energy-efficient task scheduling and resource allo-

cation solutions at various Cloud levels. In addition to these obstacles, the solutions that are provided must be scalable in multiple dimensions and cloud providers must also cope with the increasingly complex requirements of their users. Users must deploy their own applications with the topology they select, and they must maintain control over both infrastructure and programs. Consequently, requested services are more sophisticated and comprehensive. The traditional three-layer paradigm is evolving, and the convergence of IaaS and PaaS is regarded as a natural evolutionary progression in cloud computing. Cloud resource allocation solutions must be sufficiently adaptable to accommodate the changing cloud landscape and the needs of users. It is required that we thoroughly examine this critical aspect of cloud levels in this thesis, as it is crucial to our research. The issue of task scheduling and resource allocation in the Cloud is extremely difficult to resolve while maximizing energy efficiency and adhering to the mentioned dimensions. This study addresses the issue in this thesis by examining its various aspects and levels in order to offer a comprehensive and generic approach, in addition to a specific solution.

1.1 Contributions

Based on the objectives defined previously, the main contributions of this thesis are outlined:

We have achieved a survey of the state of the art on energy efficient task scheduling and resource allocation in cloud environments.

1. An Energy-Aware Scheduling Model (EASM) for task scheduling in cloud computing. The objective of the proposed model is to reduce the energy consumption, execution time, and SLA violation. EASM works in two phases, i.e., pre-processing and optimization with Adaptive Genetic Algorithm. In the first phase, tasks are allocated in VMs. In the next phase, GA is used to optimize scheduling and find better solutions.

2. A Threshold Q-learning VM Migration (TQVM), a unique artificial intelligence VM migration technique is introduced in this study. Two thresholds can be established by the suggested algorithm. The migration virtual machines (VMs) as efficiently as pos-

sible while using the least amount of energy and maintaining the necessary level of service quality.

3. A Novel Task Scheduling and VM Placement (TSVMP) Optimizing energy consumption and task scheduling using an modified genetic algorithm and resource allocation using double threshold Q-learning VM migration.

- The integration of genetic algorithms and deep learning for task scheduling and VM allocation.
- The use of thresholds to balance the load and turn off underutilized servers.
- An adaptive learning to different loads which implies minimization of energy consumption.

1.2 Thesis Organization

This thesis is structured around six chapters. In the following, the first chapter provides a short description of the subjects treated by the subsequent chapters:

Chapter 2 - Cloud computing and energy consumption

This chapter delves into the fundamental principles of cloud computing and virtualization. The issue of energy efficiency in Cloud data centers, the primary causes of energy waste, energy measurement and modeling in Cloud environments, and presenting various power-saving techniques.

Chapter 3 - Background and State of the Art

This chapter described the main research efforts in the area of energy efficient Cloud tasks scheduling and resource allocation.

Chapter 4 - Energy-aware scheduling of tasks in cloud computing

This chapter presents an Energy-Aware Scheduling Model (EASM) for task scheduling in cloud computing. The objective of the proposed model is to reduce the energy consumption, execution time, and SLA violation. EASM works in two phases, i.e., pre-processing and optimization with Adaptive Genetic Algorithm. In the first phase, tasks with longer execution times are allocated in VMs with high processing capabili-

ties. In the next phase, Genetic Algorithm is used to optimize scheduling and find better solutions.

Chapter 5 - Energy-efficient resource management in cloud computing

The objective of this chapter is to migrate virtual machines (VMs) as efficiently as possible while using the least amount of energy and maintaining the necessary level of service quality. A Threshold Q-learning VM Migration (TQVM), a unique artificial intelligence VM migration technique is exposed in this chapter. Two thresholds can be established by the suggested algorithm.

Chapter 6 - Energy-aware task scheduling and resource allocation in cloud computing

It reveals a novel Task Scheduling and VM Placement (TSVMP) in cloud computing are proposed in this chapter. The objective is to optimize energy consumption and task scheduling using an modified genetic algorithm and resource allocation using double threshold Q-learning VM migration.

The integration of genetic algorithms and deep learning for task scheduling and VM allocation.

The utilization of thresholds to balance the load and turn off underutilized servers.

An adaptive learning to different loads which implies minimization of energy consumption.

Chapter 7 - General Conclusion

In conclusion, this section summarizes our primary contributions and addresses future work directions, challenges, and perspectives.

CHAPTER 2

Cloud computing and energy consumption

2.1 Introduction

The significance of energy efficiency in cloud data centers is on the rise (Katal et al. (2022)). The issue of power consumption has become increasingly vital due to the widespread use and expanding scope of these units. Identifying the underlying causes and conducting a comprehensive analysis of the issue is crucial prior to commencing the resolution process. This chapter delves into the fundamental principles of cloud computing and virtualization, that operates as its enabling technology. We delve deeper into the issue of energy efficiency in Cloud data centers by examining the primary causes of energy waste, introducing energy measurement and modeling in Cloud environments, and presenting various power-saving techniques. Ultimately, we emphasize the thesis's orientation and focus.

2.2 Cloud Computing and Energy Efficiency

2.2.1 Cloud definitions

The term "Cloud" has become one of the most frequently used terms in the IT industry since 2007 (Marston et al. (2010)). There is no precise definition of cloud computing, despite the efforts of numerous researchers to define it from various application perspectives. We have selected three definitions that are frequently cited, as follows:(Teng (2011))

• (Foster et al. (2008)): "A large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted virtualized, dynamically-scalable,

managed computing power, storage, platforms, and services are delivered on demand to external customers over Internet." As an academic representative, Foster focuses on several technical features that differentiate cloud computing from other distributed computing paradigms. For example, computing entities are virtualized and delivered as services, and these services are dynamically driven by economies of scale.

• (Plummer and Cearley (2008)): "A style of computing where scalable and elastic IT capabilities are provided as a service to multiple external customers using Internet technologies." Garter is an IT consulting company, so it examines qualities of cloud clouding mostly from the point of view of industry. Functional characteristics are emphasized in this definition, such as whether cloud computing is scalable, elastic, service offering or Internet based.

• (Mell and Grance (2010)): "Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." Compared with other two definitions. The U.S. National Institute of Standards and Technology provides a relatively more objective and specific definition, that not only defines cloud concept overall, but also specifies essential characteristics of cloud computing and delivery and deployment models.

2.2.2 Deployment models

The manner in which clouds are deployed is contingent upon the purview of their utilization. There are four principal cloud deployment models.(Patel and Kansara (2021))

2.2.2.1 Public cloud

Public cloud is the conventional cloud computing paradigm, in which a service provider provides the general public with access to resources, including storage and applications, via the Internet. Service providers establish fees on a fine-grained utility computing basis. IBM's Blue Cloud, Sun Cloud, Google AppEngine and Windows Azure Services Platform are all examples of public clouds (Mboula (2021)).

2.2.2.2 Private cloud

Private cloud appears to be more akin to a marketing concept than the conventional mainstream definition. The architecture of the proprietary computing system is described, this offers services to a restricted number of individuals on internal networks. The private cloud is the preferred choice for organizations that require precise data management. This permits them to enjoy all the scalability, metering, and agility benefits of a public cloud without sacrificing control, security, or recurring costs to a service provider. Private cloud deployments are generated by both eBay and HP CloudStart (Choukairy (2018)).

2.2.2.3 Hybrid cloud

Hybrid cloud is a common practice among most IT vendors, as it employs a hybrid of public cloud, private cloud, and local infrastructures. Hybrid strategy involves the allocation of duties in accordance with operational, compliance, and cost factors. In order to provide services to the business, major vendors such as Oracle, VMware, HP, and IBM develop suitable strategies for utilizing a hybrid environment. It is possible for users to deploy an application that is hosted on a hybrid infrastructure, which consists of some nodes that are operating on actual physical hardware and others that are running on cloud server instances (Teng (2011)).

2.2.2.4 Community cloud

Community cloud overlaps with Grids to a certain extent. It is stated that cloud infrastructure is shared among multiple organizations within a private community. Mission, security requirements, policy, and compliance considerations are typically shared among the organizations. A cross-boundary structure can be established by aggregating community cloud with public cloud (Ghribi (2014)).

2.2.3 Cloud service

Cloud computing capability is provisioned as services, essentially in three tiers: infrastructure, platform and software as the underlying delivery mechanism.(Armbrust et al. (2009)).

2.2.3.1 Infrastructure as a Service

Infrastructure as a Service (IaaS) offers consumers processing, storage, networks, and other fundamental computing resources. The infrastructure is capable of dynamically scaling up and down, allowing IaaS users to deploy arbitrary applications, software, and operating systems. The IaaS user transmits programs and associated data, while the vendor's computer performs computation processing and returns the outcome. In order to satisfy user needs, the infrastructure is scalable, manageable, flexible, and virtualized. Examples of IaaS include Amazon EC2 (*Amazon Web Services (Amazon EC2).* (2025)), VPC(*Amazon Web Services (Amazon VPC)* (2025)), IBM Blue Cloud(*IBM Corporation.* (2025)), Eucalyptus(*Eucalyptus Systems Inc.* (2025)), Joyent(*Joyent Inc.* (2025)), and Rackspace Cloud(*Rackspace US, Inc.* (2025)).

2.2.3.2 Platform as a Service

Platform as a Service (PaaS) provides a comprehensive, integrated environment for the development, testing, deployment, and hosting of applications that have been developed by customers or acquired by them. In exchange for the inherent scalability of an application, developers typically concede to certain limitations on the type of software that can be written. The underlying infrastructure is not managed by PaaS customers, as is the case with SaaS users. However, they have control over the deployed applications and their hosting environment configurations. The primary objective of PaaS offerings is to simplify the development of applications and the administration of associated issues. Some are designed to offer a comprehensive development environment, while others are limited to hosting-level services, including on-demand scalability and security. Google App Engine(*Google App Engine* (2025)), Windows Azure(*Windows Azure (Microsoft*

Azure) (2025)), Engine Yard(*Google App Engine* (2025)), Force.com(*Salesforce Platform* (2025)), Heroku(*Heroku – Cloud Platform for Apps.* (2025)), and MTurk(*Amazon Mechanical Turk (MTurk)* (2025)) are among the most common examples of PaaS.

2.2.3.3 Software as a Service

Software as a Service (SaaS) is a software delivery paradigm that enables users to access applications through a straightforward interface, such as a web browser, over the Internet. The consumers are not concerned with the fundamental cloud infrastructure, that includes the network, servers, operating systems, storage, and platform. Moreover, this paradigm eliminates the necessity of installing and running the application on local devices. Salesforce.com, that distributes business software on a subscription basis rather than on a traditional on-premise basis, is the entity that popularized the term ”SaaS.” One of the most well-known is the solution for its Customer Relationship Management (CRM). SaaS has now become a prevalent delivery paradigm for the majority of business applications, such as accountancy, collaboration, and administration. The family of SaaS-based services is enhanced by applications such as social media, office software, and online games. For example, netSuite(*Oracle NetSuite.* (2025)), Google Docs(*Google Docs* (2025)), Microsoft online(*Microsoft Online* (2025)), web Mail(*Gmail* (2025);*Outlook* (2025);*Yahoo* (2025)), Facebook(*Meta Platforms,* (2025)), and MMOG Games(*World of Warcraft* (2025);*Final Fantasy XIV* (2025)).

2.2.4 Virtualization and Cloud Computing

Cloud computing is primarily enabled by virtualization technology. It is founded on the abstraction of physical resources, that permits the multiplexing of multiple virtual resources on a single physical resource. Virtualization is employed to facilitate the coexistence of heterogeneous services on the same physical infrastructure, as well as to provide isolation, flexibility, higher resource utilization, simple resource management, and resource elasticity.(Ghribi (2014))

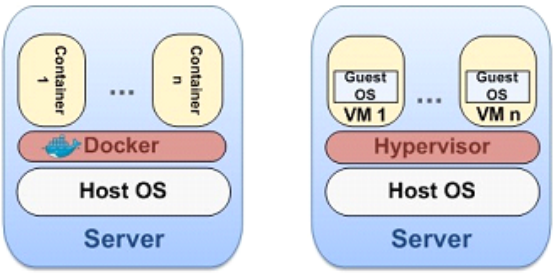

Fig. 2.1 Container based virtualization vs hypervisor based virtualization

2.2.4.1 Virtualization Forms

Virtualization incorporates a variety of technologies. Server virtualization, storage virtualization, and network virtualization comprise the primary categories of virtualization. The concept of physical resource abstraction and partitioning is the foundation of all of these kinds. The major focus of this thesis is server virtualization, that is the most prevalent resource abstraction technique in cloud computing. This type of virtualization allows the operation of multiple isolated virtual servers on a single server and can be implemented in a variety of ways. The implementation strategies encompass full virtualization, para-virtualization, and OS-level virtualization. Para-virtualization and full virtualization use a hypervisor to share the underlying hardware. However, they differ in the manner in which the host and guest operating systems are modified to support virtualization and in their interactions with one another. Operating system level virtualization does not employ a hypervisor, in contrast to full virtualization and para-virtualization. Therefore, server virtualization can be categorized into two primary categories: hypervisor-based virtualization and OS or container-based virtualization, depending on the method by which virtualization is accomplished. The following section provides additional information regarding this classification.

2.2.4.2 Server virtualization categories

In Cloud computing, there are two primary methods for virtualizing resources: container-based virtualization and hosted virtualization, that utilizes a hypervisor.

Hypervisor based virtualization: The conventional method of virtualization in the Cloud is hypervisor-based virtualization. The physical server resources are managed by a software layer known as a hypervisor, that is the foundation of this technology. KVM(KVM (2025)), VMWare(VMware (2025)), Microsoft Hyper-V(Microsoft (2025)), Xen(Project (2025)), and Virtual Box(VirtualBox (2025)) are all examples of hypervisors. On the same physical host, guests are referred to as virtual machines (VMs) and operate under a variety of operating systems, including Linux and Windows. Despite the introduction of an additional software layer by this form of virtualization, it facilitates the consolidation of resources into virtualized servers(Srikantaiah et al. (2008)) and provides a live migration feature(Travostino et al. (2006)) that allows VMs to be transferred to other servers without the need to close them down.

Container based virtualization: Container-based virtualization is a more lightweight alternative to hypervisors (Soltesz et al. (2007))(Xavier et al. (2013)). It is technology that operates at the operating system level and enables the operation of multiple isolated virtual environments on the same host. In contrast to traditional virtual machines (VMs), containers are built on shared operating systems and utilize a single operating system (the host's OS). The distinction between the two types of virtualization is illustrated in Figure 2.1. Docker(Docker (2025)), Linux containers (LXC)(Containers (2025)), Solaris Containers(Oracle (2025)), Virtuozzo Containers(Parallels (2025)), and OpenVZ(OpenVZ (2025)) are among the container-based solutions. Hypervisor-based virtualization is more suitable for situations where security and flexibility are necessary, as well as when heterogeneous operating systems are required. When performance is necessary, container-based virtualization is a practical solution. It offers superior manageability with near-native performance, as well as a significantly higher consolidation ratio and the most efficient resource utilization, as it supports a large number of instances on a single host. This solution offers portability, transport, and process-level isolation across hosts, in addition to being lightweight. Despite their distinctions, hypervisors and container-based virtualization are not mutually exclusive; rather, they are increasingly employed in conjunction. The deployment of complex services that

combine both applications and underlying infrastructures over hybrid IaaS/PaaS cloud providers is facilitated by the use of both container-based virtualization and hypervisors, that are commonly used to build lightweight PaaS environments and IaaS Cloud services, respectively. Certain solutions, such as Proxmox (Proxmox (2025)), provide both technologies on a single physical server.

2.2.5 Quality of Service (QoS)

In cloud computing environments, the Quality of Service (QoS) is typically denoted by high-level parameters (Mboula (2021)). A Service Level Agreement (SLA) is a contract between a cloud user and its cloud service provider that specifies the values that must be satisfied for the various parameters. There are four categories into which these various QoS parameters within a cloud can be classified(Guérout (2014)): dependability, performance, security and data, and cost. We will only discuss the numerous existing parameters in our work. An exhaustive enumeration can be found in (Guérout (2014)).

2.2.5.1 Performance and dependability category

The Performance category includes two key metrics: Execution Time and Response Time. Execution time depends on the capacity of the virtual machine and the complexity of the request, particularly in terms of the number of instructions to be executed. Response time refers to the interval between the submission of a user request and the reception of the corresponding response from the service. It represents the time needed to make a service available and usable for the user, serving as a measure to evaluate the service's efficiency.

The Dependability category focuses on Reliability, which can be defined in different ways. According to (Endo et al. (2017)), reliability is the ability of a component to perform its required functions within a defined time frame and under specified operational conditions. Zhang and Chakrabarty (2003) describe system reliability as the probability of successfully completing a task without errors. Additionally, (Garg and Mittal (2019)) highlight that the reliability of computational nodes is particularly crucial for computation-intensive applications.

2.2.5.2 Cost category

The cloud provider establishes the service cost in relation to the user's selection of service and the service's duration. A common invoicing model for the leasing of virtual machines is hourly-based. In other words, each partial hour consumed will be adjusted up to a full hour. For example, 1 hour and 1 minute (61 minutes) will be regarded as 2 hours (120 minutes) of utilization. The service cost will be determined by multiplying the total number of hours by the unit hour cost of the VM.

2.2.5.3 Energy consumption

The energy required to operate an equipment over a specified time period is denoted by the kilowatt-hour (kWh) energy cost. It is determined by the capacity (in watts) and duration of use (in hours) of the apparatus (physical machine, for instance). A composite unit of energy, the kilowatt-hour (kWh) is equivalent to one kilowatt (kW) of electricity that is sustained for one hour.

2.2.6 Data center architecture

For a sake of establishing the context of the energy consumption issue in a Cloud, we will examine the operation of a data center that is responsible for the processing of Cloud services and identify the equipment that consumes the most energy. A data center is defined by a collection of components and systems, including the power supply system, the refrigeration system, the air distribution system, and the servers, as illustrated in Figure 2.2. Each of these systems will be introduced in this section.(Choukairy (2018))

2.2.6.1 Power supply

A PSU (Power Supply Unit) is the device that provides power to each server. This device is subsequently powered by a secure electrical network that originates from the power supply unit, that comprises two devices: the UPS (Uninterrupted Power Supply) and a PDU (Power Delivery Unit). A series of devices are established to guarantee a

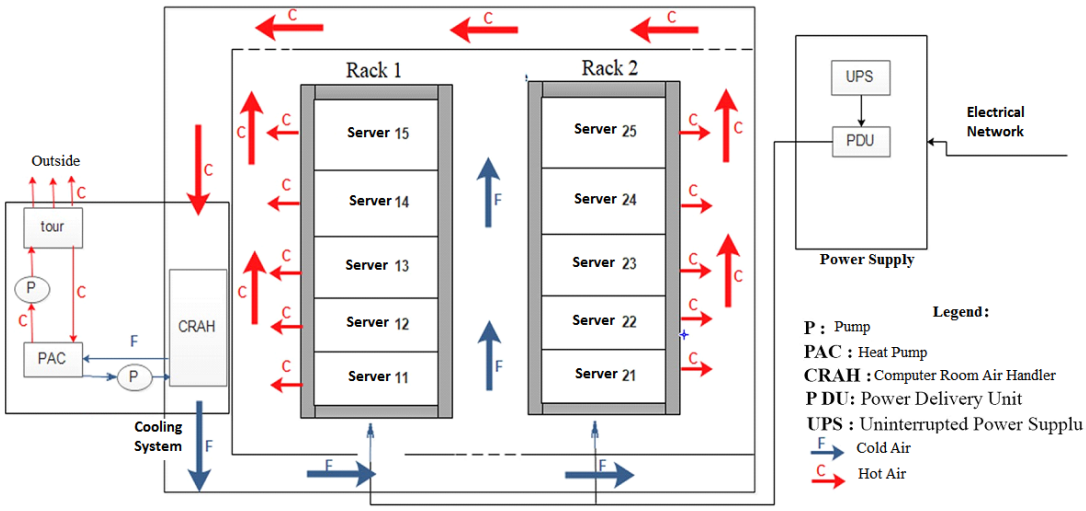

Fig. 2.2 General architecture of a data center.(Choukairy (2018))

stable and permanent power supply, that must be greater than 99% (Relaza (2016)), in order to assure the availability of the data centers. The electrical energy from the main network is initially directed to the UPS to supply power to the various PSUs. In the event of a primary power failure, the latter reserves a substantial quantity of electrical energy in batteries and subsequently restores it. The servers must be powered by the rectified and transformed alternating current that is produced at the device's output. The PDU is responsible for distributing the load among the numerous units.

2.2.6.2 The cooling system

To ensure the reliability of servers, it is crucial to regulate the heat they emit by maintaining suitable temperature and humidity levels. The Joule effect performs, in fact, convert nearly all of the energy consumed by the apparatus into heat. In order to achieve this aim, the data center's temperature is maintained by a refrigeration system. This chilling can be accomplished through the use of either air or water. In general, cooling techniques are solutions that rely on the direct use of the external environment (air or water) in its ambient conditions to achieve chilling, either in full or in part. Figure 2.2 (Beloglazov et al. (2011)) illustrates a cold air conditioning system. It is responsible for perpetually capturing the heated air generated by the IT equipment, conditioning it to the desired frigid temperature, and then blowing it into the room. The cooling system is typically comprised of circulation compressors, a CRAH (Computer Room Air

Handling) unit, and a heat pump (PAC). A condenser is located within the PAC and is chilled by a water conditioning structure. The CRAH is equipped with a fan that guarantees the circulation of both supply and return air. Lastly, the water in the primary and secondary circuits of the system is circulated by circulation pumps, that are denoted by the letter P in Figure 2.2.

2.2.6.3 Air distribution system

The air distribution system (ADS) employs a method that is designed to direct the frigid air generated by the CRAH unit to the computer servers, while simultaneously extracting the heated air that is rejected by the servers to condition it in the cabinet (rack) (Erden et al. (2016)). In the majority of current data centers, the air travels at a high velocity and exhibits turbulent behavior. The major obstacle of the distribution system is to regulate the amount of mingling between the frigid air that is intended to chill the servers and the heated air that must be evacuated. Therefore, the ADS is of paramount significance to the data center's effective operation, despite the fact that it is not composed of physical elements and, as a result, does not assimilate energy. For instance, in Figure 2.2, the distribution system is symbolized by arrows with the letter F to represent frigid air and arrows with the letter C to represent heated air, symbolizing the air fluxes in the data center. Hot air is captured at the ceiling level, while cold air is released at the floor level in close proximity to the IT (Information Technology) equipment. Hot Aisle/Cold Aisle is the term used to describe this configuration.

2.2.6.4 Computer servers

The operating mode of all ancillary installations is determined by these components. They are arranged in racks (or computer cabinets) of five floors, as illustrated in Figure 2.2. The CPU, memory and disk are also the primary components of the server that consume the most energy, as demonstrated in Figure 2.2. The CPU is the element that consumes the most energy among these components (Djouhra (2016)). The IT portion is comprised of all of these components and is responsible for the processing of the tasks that are received by the servers. The PSU (Power Supply Unit) is the source of

power for the IT component.

2.2.7 Energy Efficiency in Cloud Data Centers

Cloud computing consumes a large amount of energy, which is a major concern. This high energy use mainly comes from servers, storage, networking devices, and cooling systems. It is often caused by poor resource usage, idle servers, and unbalanced task distribution. To reduce this, solutions like virtualization, dynamic resource allocation, and smart scheduling are used.

2.2.7.1 Issue of High Energy Consumption in the Cloud

Energy consumption is a vital concern in cloud computing environments, as indicated in the introductory section of this chapter. Moore's law (Moore (1998)) explains that the capacity of data centers has been expanded by the efficient design of the system and the increasing density of the component(s). As a result, the efficacy per watt ratio has been consistently enhanced; however, the total power consumed by computer systems has not significantly decreased. The energy consumption of data centers alone will rise from 200 TWh in 2016 to 2967 TWh in 2030 (Katal et al. (2022)). Consequently, it is imperative to identify the primary causes of the issue of cloud power and energy consumption.(Mboula (2021))

2.2.7.2 Sources of excessive energy consumption

There is no doubt that energy consumption is also influenced by power delivery infrastructure and refrigeration equipment, as they are perpetually supplying power to equipment. Nevertheless, the inefficient allocation of server resources is the main cause of half of the data center's energy waste (Koot and Wijnhoven (2021)). Efficiency can be managed at various levels of a computing system (Piraghaj et al. (2017)) (refer to Figure 2.3). While it is challenging to obtain precise information regarding the utilization of cloud resources by cloud providers. Various studies have demonstrated the general trend of cloud resource utilization. Cloud infrastructures are actually underutilized. Approximately 52% of cloud resources are classified as highly underutilized

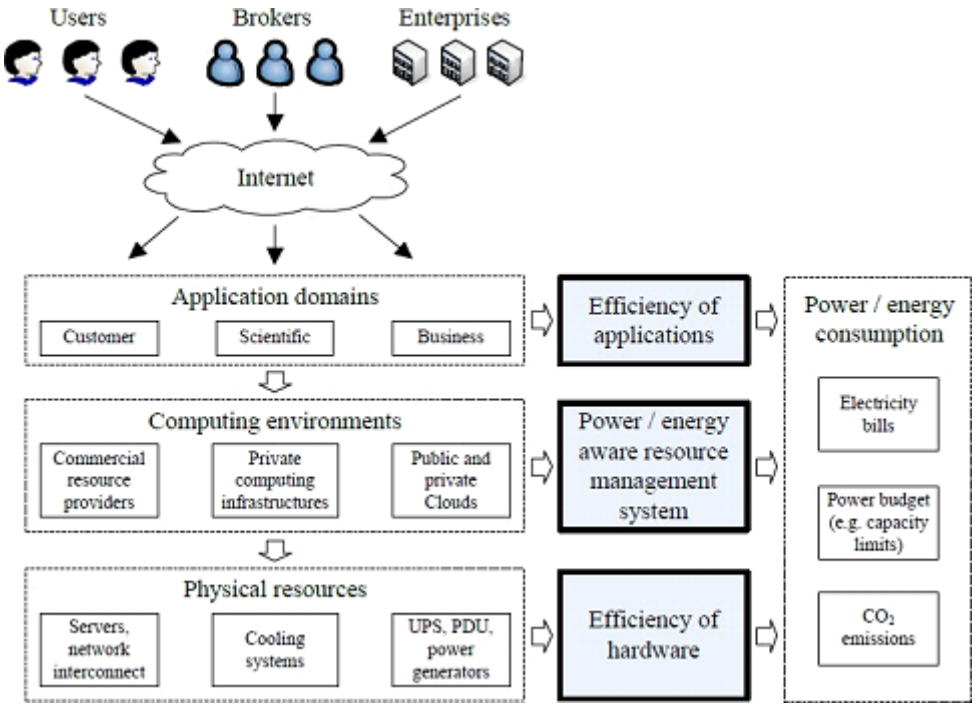

Fig. 2.3 Energy consumption incurred at divers levels in computing systems.(Mboula (2021))

(Khan et al. (2020)), with a significant number of them remaining inactive or having been used sparingly. The cluster is only 20% – 40% utilized, according to researchers who analyze Google traces (Khan et al. (2020)). Resource management techniques, that are designed to increase the rate of resource utilization, can result in substantial energy savings in data centers, despite the significant under-utilization of resources(Khan et al. (2020)). One of the most frequently proposed techniques for increasing the rate while reducing energy consumption is the consolidation of virtual machines.

2.2.7.3 Energy consumption reduction approaches

From Figure 2.3, it is evident that there are three primary levels from which the energy consumption of the system can be reduced through effective management: the physical machine level, virtual machine level, and application level. Although the PaaS user is accountable for the application level, the provider is responsible for the other two.

Among the existing energy reduction approaches we have the following:

Switching idle servers off:

This method involves the shutdown of servers that are not in useless. It has the potential to substantially reduce server consumption by ensuring that servers are powered down, thereby achieving near-zero energy consumption. Nevertheless, prior studies that implemented this methodology encountered challenges in ensuring service-level agreement as a result of the absence of a dependable instrument for forecasting future demand to facilitate the decision-making process for turning off/on (Duy et al. (2010)).

VMs/workload consolidation:

The technique of energy-efficient dynamic VM consolidation permits cloud environments by the ability to migrate VMs at runtime from one physical host to another. The first technique can be utilized to disable the second technique, that increases the load of one host at the expense of another. (Beloglazov (2013)) have conducted a comprehensive investigation of that technique.

The Dynamic Voltage and Frequency Scaling (DVFS):

This method involves the dynamic adjustment of the frequency of the CPUs in tangible devices in accordance with their utilization rates. In order to reduce power consumption, the intended is to reduce the supply voltage of the CPU, that in turn reduces the clock frequency (Herbert and Marculescu (2007)).

2.2.7.4 Potential power consuming units in cloud data centers

It is crucial to investigate the power transfer in typical data centers and comprehend the distribution of power in order to enhance energy efficiency in the Cloud. In reality, the IT tasks are receiving over half of the electrical power (refer to Figure 2.4). Servers account for 80% of the total IT demand and 40% of the total data center power consumption, as indicated in the Environmental Protection Agency's Report to Congress on Server and Data Center Energy (Environmental Protection Agency (EPA) (2007)). The remaining power is consumed by other devices, including distribution wiring, air conditioners, compressors, illumination, and transformers.

The power consumption of refrigeration equipment is decisive; however, it is di-

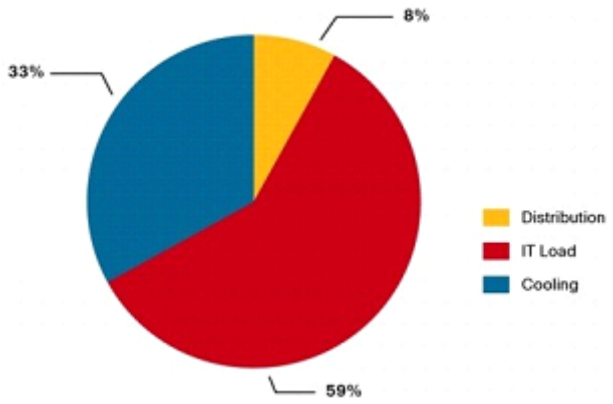

Fig. 2.4 Typical power draw in a data center. (Ghribi (2014))

rectly proportional to the power consumption of IT. The power consumption of cooling can be reduced by utilizing technologies such as free cooling, that are employed by large corporations (e.g., Google, Facebook, eBay). These methods reduce the temperature of the air in data centers by employing naturally chilly air or water, as opposed to mechanical refrigeration. Consequently, the electrical power required for refrigeration has been significantly reduced. Zero refrigeration is feasible in numerous climates, that can result in savings of up to 100%. (Ghribi (2014))

2.2.7.5 Major causes of energy waste

Servers are the primary power consumers in cloud data centers, as previously stated. The primary causes of this substantial consumption are as follows:

Low server utilization:

The quantity of servers is increasing in tandem with the expansion of data centers. The majority of servers in data centers are underutilized. The Natural Resources Defense Council (NRDC) report ((NRDC))(Natural Resources Defense Council (NRDC) (2014)) indicates that the average server utilization remained constant between 12% and 18% from 2006 to 2012, while servers consumed between 60% and 90% of peak power. By consolidating virtual servers on a smaller number of hosts. It is possible to operate the same applications with significantly reduced power consumption. The number of servers required and the overall energy consumption will be significantly reduced by

increasing server usage.

Idle power waste:

Data center servers are idle and do not perform any valuable tasks for approximately 85-95% of the time(Natural Resources Defense Council (NRDC) (2014)). Even when not in use, a dormant server consumes approximately 70% of its maximal power (Naone (2009)). This energy inefficiency is primarily caused by the waste of inactive power. Therefore, it is feasible to disable inactive servers in data centers in order to mitigate energy consumption.

Lack of a standardized metric of server energy efficiency:

In order to guarantee energy efficiency optimizations, it is crucial to employ an energy efficiency metric to arrange servers based on their energy efficiency. This metric enables scheduling algorithms to make judgments and select the most efficient resources to optimize energy efficiency. Despite the emergence of a few metrics that concentrate on IT efficiency in recent years (Naone (2009)), they do not offer a straightforward benchmark that can be used to drive the optimization of energy efficiency ((NRDC)).

Energy efficient solutions are still not widely adopted:

According to the NRDC report((NRDC)), numerous large-scale cloud farms demonstrate exceptional energy efficiency; however, they account for less than 5% of the energy consumed by global data centers. The average efficiency of the remaining 95% of small, medium, corporate, and multi-tenant operations is significantly lower. Therefore, it is imperative that energy efficiency best practices are more widely implemented, particularly in the case of small and medium-sized data centers, that are notoriously inefficient and utilize approximately half the power of all data centers.

2.2.8 Power measurement and modeling in Cloud

It is crucial to recognize the relationship between power and energy and to present their units of measurement prior to engaging in power and energy measurement and model-

ing. Power consumption is the rate at which a machine can operate and can be calculated by multiplying voltage and current, whereas electrical energy is the total quantity of power consumed over a specific period. The watt (W) is the standard metric unit of electricity, while the watt-hour (Wh) is the unit of energy. The definitions of power and energy are illustrated in Eq. 2.1 and 2.2, where P represents power consumption, I is current, V represents voltage, E represents energy and T represents a time interval and is expressed as:

$$P = IV \tag{2.1}$$

$$E = PT \tag{2.2}$$

In order to quantify the consumption of power and energy in the cloud, we distinguish between power and energy estimation models and measurement techniques. The initial one employs immediate monitoring tools to directly measure actual power consumption. Power metering models estimate the power consumption of servers and VMs by utilizing metrics provided by the operating system or the hardware.

2.2.8.1 Power and energy estimation models

Models that estimate the power and energy consumption, as well as the power cost of virtual machine migration, are becoming increasingly appealing for power metering, since the majority of servers in contemporary data centers lack power measurement devices and VM power cannot be measured by sensors. Data center energy efficiency metrics are introduced in this section, that also provides a general overview of power estimation models and tools in the Cloud.

Power and energy modeling for servers:

In the literature, power consumption models for servers have been extensively investigated (Basmadjian et al. (2011)) and range from complex to uncomplicated. CPU-based linear models are a straightforward and lightweight method for estimating the power consumption of servers, as the CPU is the primary energy consumer and the

relationship between power and CPU utilization is linear (Kansal et al. (2010)). Simple usage-based power models for servers are proposed in (Economou et al. (2006)), (Heath et al. (2005)), (Raghavendra et al. (2008)), and (Beloglazov and Buyya (2010)). The power models they present are based on the assumption that the CPU is the sole factor and present an approximation for total power in relation to CPU utilization (u). This study investigates the utilization of central processing units (CPUs) to ascertain the amount of electricity that tangible devices utilize. About 70% of the power of a physically active machine is used when it is inactive. So, the power consumption (u) as CPU utilization is defined as in Eq. 2.3.(Saadi and El (2020))

$$P(u_i) = P_{max}(0,7 + 0,3u_i) \tag{2.3}$$

Another equation (Equ.2.4) can be used to estimate the power consumption of a server based on its utilization level. This more flexible model takes into account both the minimum and maximum power of the server, thus enabling a more realistic energy representation: (Goyal et al. (2021))

$$P(u_i) = P_{min} + u_i(P_{max} - P_{min}) \tag{2.4}$$

where u_i represents the current CPU utilization, P_{min} is the idle power and P_{max} represents the maximum power of a physical system that is operating at 100% CPU utilization.

CPU utilization is defined as a function of time $u(t)$, since it changes over time.As a result, Eq. 2.5 establishes a physical machine's (PM_i) total energy consumption:

$$E_i = \int P(u(t))\,\mathrm{d}t \tag{2.5}$$

2.3 Conclusion

The concepts of cloud computing and virtualization were introduced in this chapter, and the issue of energy efficiency in the cloud was examined. The major causes of energy

waste in Cloud data centers were discussed, as well as, the methodologies for energy measurement and modeling. Additionally, the power-saving techniques employed in Cloud data centers were described. A discussion of the orientation and focus of this thesis has also concluded this chapter. The subsequent chapter delves deeper into the issue of task scheduling in the cloud. We offer state-of-the-art solutions and background information for energy-efficient task scheduling. Subsequently, we deliberate on the challenges and issues that are pertinent.

CHAPTER 3

Background and State of the Art

3.1 Introduction

Many research studies have focused on the energy problem. These works are different disciplines. Their main objective is to reduce energy consumption. This chapter presents our study of the works and provide a synthesis of the work related to our main objective in this thesis. Existing state-of-the-art approaches and models must be examined and analyzed in order to offer effective solutions, approach the problem from many perspectives, and manage its limitations. The state of the art is presented in this chapter and a given summary of the current state of the art in this area at various levels and dimensions is provided.

3.2 Categories of solutions

In this review, the various task scheduling and resource allocation algorithms found in the literature are categorized into four types: Threshold-Based Scheduling, meta-heuristic algorithms, hybrid meta-heuristic and Machine learning based algorithms (Khan et al. (2023)).

1. Threshold based algorithms are simple but can be rigid, relying on static limits for power and workload management.

2. Meta-heuristic algorithms constitute high level strategies, independent of any specific problem, making them applicable to a wide range of problems. Particle Swarm Optimization and Genetic Algorithm are two prominent meta-heuristic algorithms used across many disciplines.

3. Hybrid meta-heuristic algorithms utilizes more than one meta-heuristic algorithm to schedule tasks and allocate resources on the cloud. Hybridizing two meta-heuristic algorithms are intended to alleviate potential weaknesses in a specific meta-heuristic algorithm.

4. Machine learning based algorithms: provide predictive capabilities and adapt to dynamic workloads, making them suitable for minimizing real-time energy consumption.

3.2.1 Threshold-Based Scheduling

Threshold-based algorithms rely on predefined values (e.g., CPU usage or energy) to guide task allocation or migration. They are simple and fast, making them suitable for real-time systems, though they lack adaptability. Several studies have proposed enhancements to improve their performance.

Maurya and Sinha (2013) introduce a load balancing strategy that is both energy-conscious and power-aware. In addition, it is based on the adaptive migration of virtual machines (VMs). This strategy will be implemented for virtual machines on the cloud, with a focus on the establishment of both higher and lower thresholds for the migration of virtual machines to the servers. Authors also take into account RAM and bandwidth in order to optimize performance and balance loads. In the event that the load exceeds or falls below the predetermined upper and lower thresholds, the virtual machines will be migrated accordingly, thereby increasing the cloud data center's resource utilization and decreasing their energy consumption. To decrease the number of migrations, authors implement a minimum migration time policy. This policy is capable of reducing the number of migrations and the energy consumption of virtual machine migration, as well as achieving load balancing and meeting service level agreement (SLA) requirements.

Li et al. (2019) implement a tradeoff strategy that enables the attainment of optimal energy consumption with a delay threshold. Initially, the function of the delay threshold in the reduction of delay is discussed. Then, the queue theory to analyze the energy

Table 3.1 Taxonomy of algorithms based on thresholds

Author and year	Technique	Performance metric
Maurya and Sinha (2013)	Thresholds VM migration	Energy consumption migration number SLA violation.
Malik et al. (2021)	Task classification Thresholds PSO	Energy consumption Resource usage
Saadi and El (2020)	Thresholds VM consolidation	Energy consumption
Hijji et al. (2022)	DVFS Thresholds	Energy consumption SLA violation
Adhikari and Patil (2013)	Thresholds VM consolidation	Energy consumption Resource usage SLA violation
Semmoud et al. (2020)	Thresholds	Response time Migration cost
Awasthi et al. (2022)	Task classification Thresholds PSO ESCEL	Energy consumption Resource usage
Shally et al. (2020)	Thresholds VM consolidation	Energy consumption
Karim et al. (2024)	Modified ABC Thresholds	Energy consumption Resource usage
Singh and Kumar (2022)	Thresholds MAMFO DT-ESAR	Energy consumption CPU and Memory usage

consumption and latency of the cloud server layer, fog node layer and mobile terminal layer is used. The energy optimization problem is resolved through the application of nonlinear programming, that determines the optimal workload for every layer. In order to mitigate energy consumption, authors develop a cloud-fog cooperation scheduling algorithm.

Malik et al. (2021) investigate the issue of energy consumption and the efficient utilization of resources in virtualized cloud data centers. Task classification and thresholds are the foundations of the proposed algorithm, that is designed to optimize resource utilization and scheduling. In the initial phase, workflow tasks are preprocessed to prevent bottlenecks by separating tasks with lengthy execution periods and more dependencies into distinct queues. Tasks are categorized according to the intensity of the necessary resources in the subsequent steps. Ultimately, the optimal schedules are determined through the application of Particle Swarm Optimization (PSO).

In (Saadi and El (2020)), the authors suggest an Energy-Efficient Strategy (EES) for consolidating virtual machines in a cloud environment. The objective is to reduce energy consumption while simultaneously completing a greater number of tasks with the maximum possible throughput. The performance-to-power ratio is employed in their proposal seeking upper thresholds for over detection. In addition, EES establishes lower thresholds by taking into account the overall data center workload utilization, that can decrease the frequency of virtual machine migrations..

The paper of Hijji et al. (2022) aims to address the challenges associated with energy conservation in gaming data centers by utilizing dynamic voltage and frequency scaling techniques. Additionally, to assess the dynamic voltage and frequency scaling techniques in comparison to static threshold detection and non-power-aware techniques. The results will assist service suppliers in overcoming the quality of service and experience limitations by adhering to the Service Level Agreements.

DT-PALB (Double Threshold Energy Aware Load Balancing) is an algorithm that maintains the state of all compute nodes and determines the number of compute nodes that should be operational based on utilization percentages (Adhikari and Patil (2013)).

An Adaptive Threshold-Based Approach. Researchers suggest a new method for dynamic consolidation of virtual machines (VMs) that is based on adaptive utilization thresholds. This method guarantees a high level of compliance with Service Level Agreements (SLAs).

The authors Semmoud et al. (2020) suggest a novel distributed load balancing algorithm that is predicated on an adaptive starvation threshold. It endeavors to maintain the stability of the system, minimize the response time of the cloud, maximize the utilization rate of the servers, decrease the overall migration cost, and balance the load between the servers. based Load Balancing (STLB) algorithm which is a distributed load balancing algorithm. Unlike many methods that execute the load balancing algorithm even if all the nodes are busy, STLB does not start until at least one VM is close to starvation which drastically reduces the number of migrations. To reduce the complexity of the proposed algorithm and the number of exchanged messages, only the direct neighbors were considered for information gathering. Indeed, using an extended node's neighborhood may lead to an additional overhead cost due to the management of non direct neighbors that could participate in the load balancing process. The global workload will be asynchronously and iteratively propagated between direct neighbors until reaching the global equilibrium in the system.

The authors of (Awasthi et al. (2022)) discuss the issue of energy consumption and the efficient utilization of resources in virtualized cloud data centers. Task classification and thresholds are the foundations of the proposed algorithm, that is designed to optimize resource utilization and scheduling. In the initial phase, workflow tasks are preprocessed to prevent bottlenecks by separating tasks with lengthy execution periods and more dependencies into distinct queues. This paper suggests an algorithm that would facilitate the efficient allocation of server resources. In order to optimize resources for optimal performance, this study implemented PSO and ESCEL (Equally Spread Current burden Execution) to balance the load assigned to the servers.

Using dynamic thresholds, a novel approach has been suggested by Shally et al. (2020). Thresholds are employed to consolidate virtual machines (VM) on physical machines (PM). In an effort to mitigate the energy consumption of the physical devices in the data centers, a dynamic threshold selection method is implemented. The upper and lower thresholds are dynamically established in accordance with the CPU utilization pattern that has been observed.

The authors Karim et al. (2024), suggest a novel, efficient algorithm for the deployment of virtual machines in a cloud computing environment. This method utilizes a modified artificial bee colony optimization algorithm to identify underutilized physical machines by analyzing energy consumption and resource allocation charts. An adaptive threshold method is subsequently suggested to identify underutilized physical host devices by selecting appropriate threshold levels for energy consumption.
Singh and Kumar (2022) introduce the energy-efficient multi-objective adaptive Manta ray foraging optimization (MAMFO) as a method for optimized workflow planning. It also optimizes multi-objective factors, including CPU and memory utilization and energy consumption. Dynamic Threshold with Enhanced Search and Rescue (DT-ESAR) is made available for the virtual machine consolidation System. The dynamic threshold identifies the hosts that are normalized, overutilized, and underutilized. The threshold number is used by ESAR to migrate the virtual machines from one host to another. The framework that has been suggested enhances energy efficiency and reduces the duration of the process flow.

3.2.2 Meta-heuristics

This category comprises studies based on meta-heuristic algorithms. It showcases scheduling techniques that utilize a single meta-heuristic algorithm in addition to other strategies.

Table 3.2 Taxonomy of meta-heuristic algorithms

Author and year	Technique	Performance metric
Choudhary and Perinpanayagam (2022)	PSO	Energy consumption Execution time
Pradhan et al. (2022)	DRL PPSO	Execution time
Saif et al. (2023)	MOP NPSO MLLF	Energy consumption Delay
Alsaidy et al. (2020)	LJFP MCT PSO	Convergence speed Performance
Ibrahim et al. (2018)	ILP GA	Energy consumption
Pirozmand et al. (2021)	GA ECS	Energy consumption Time
Imene et al. (2022)	GA	Energy consumption cost Execution time
Hoseiny et al. (2021)	PGA	Energy consumption Execution time

3.2.2.1 Particle Swarm Optimization(PSO)

A novel approach based on multi-objective optimization is utilized with CloudSim as the underlying simulator in order to evaluate the virtual machine allocation performance. In (Choudhary and Perinpanayagam (2022)), authors determine the energy consumption, CPU utilization, and number of executed instructions in each scheduling interval for complex virtual machine scheduling solutions to improve the energy efficiency and reduce the execution time. Based on the results, multi-objective PSO (particle swarm optimization) optimization can achieve better and more efficient effects for different parameters than multi-objective GA (genetic algorithm) optimization can.

Pradhan et al. (2022) describe a parallel computing scheduling algorithm known as the Deep Reinforcement Learning with Parallel PSO (DRLPPSO) algorithm. This algorithm uses both the DRL learning algorithm and the Parallel PSO algorithm. Authors use the DRL learning technique to train their neural network to receive the greatest reward. Using Parallel Particle Swarm Optimization (PPSO), the overall processing time of all incoming load is decreased. This scheduling approach is intended to enhance different load balancing parameters in a shorter time period than other common current scheduling algorithms in a cloud environment.

Saif et al. (2023) introduce a novel Multiple-objective Problems (MOP) approach, the Non-dominated Particle Swarm Optimization (NPSO) algorithm, for workload distribution in cloud-fog computing. The mathematical framework used in the study to describe energy consumption and delay functions is queue theory. To tackle the delay optimization problem, authors propose the Modified Least Laxity First (MLLF) method to minimize the delay threshold and an external archive to save the non-dominated solution from the Pareto optimum solutions.

In (Alsaidy et al. (2020)), heuristic techniques are employed to supplement the PSO algorithm for task scheduling. The PSO particles are heuristically initialized using the longest job to fastest processor (LJFP) and Minimum Completion Time (MCT) algorithms. Starting the search process using LJFP and MCT-based methods can considerably improve convergence speed and performance. It should be noted that the suggested

heuristic initialization of the PSO population generates initial particles that all begin the search process from the same beginning point.

3.2.2.2 Genetic Algorithm (GA)

Ibrahim et al. (2018) focus on the development of a dynamic task scheduling algorithm by proposing an Integer Linear Programming (ILP) model that reduces energy consumption in a Cloud data center. In order to accommodate the dynamic nature of the cloud environment and provide a scheduling solution that is virtually optimal in terms of energy consumption, an adaptive genetic algorithm (GA) is recommended. The proposed adaptive GA is validated in this environment by conducting a series of performance and quality evaluation studies and simulating the Cloud infrastructure.

In (Pirozmand et al. (2021)), a two-step approach known as the Genetic approach and Energy-Conscious Scheduling Heuristic (GAECS) is introduced with the goal of saving time and energy. In the GAECS method, authors employed the Genetic method to generate optimum schedules and three ranking algorithms to generate main chromosomes. The authors also employed the ECS algorithm, an energy-aware approach, to improve resource allocation to processors. The GAECS algorithm generates the first three primary chromosomes using three prioritization algorithms, then passes the primary chromosomes to the GA, who completes the primary population using the Genetic Algorithm. Then, using the prescribed crossover and mutation operators, improved chromosomes are chosen, and lastly, the optimal chromosomes are chosen in terms of time and energy.

In (Imene et al. (2022)), a third-generation Multi-objective optimization method known as Non-dominated Sorting Genetic Algorithm (NSGA-III) is used for the first time in their knowledge to schedule a set of user tasks on a set of available virtual machines (VMs) in the cloud based on a new Multi-objective adaptation function to minimize the runtime (TE), power consumption (CE), and cost.

Hoseiny et al. (2021) examine task scheduling in fog-cloud computing systems with diverse computational nodes. Authors classified jobs based on their characteristics, such as deadlines and instructions, to identify a suitable setting for each. authors introduced

PGA, a priority-aware evolutionary algorithm that optimizes computing time, energy usage, and task completion rate.

3.2.2.3 Simulated Annealing Algorithm (SA)

Feng et al. (2021) examine a global-energy-consumption virtual machine placement (VMP) model that considers the utilization of both IT and non-IT resources. The server, virtual machine, and network consumption model are all factors that authors take into account when evaluating IT resources. The cooling system and the heat recirculation paradigm of data centers are considered for non-IT resources. In order to resolve the NP-hard VMP problem, they implement a two-step algorithm, simulated annealing and greedy algorithm (SAG). The Simulated Annealing algorithm (SA algorithm) is the foundation of the initial phase, that is designed to reduce the energy consumption of the server and cooling system. The Greedy algorithm is employed in the second phase to reduce the energy consumption of the network.

3.2.2.4 Water Wave Optimization (WWO)

An Energy-Aware algorithm for workflow Scheduling in cloud computing with Virtual Machines Consolidation (EASVMC) is proposed (Medara et al. (2021b)). To address the multi-objectives of energy consumption, resource utilization, and virtual machine migrations, the proposed EASVMC approach is modeled. Task scheduling and virtual machine consolidation (VMC) are the two phases of the EASVMC algorithm. The task with the maximum execution length is assigned to the virtual machine that will execute it with the least amount of energy during the initial phase. A prominent NP-hard problem is the virtual machine consolidation, that is included in the second phase. The physical hosts are categorized into normal load, under-loaded, and overloaded hosts during the VMC phase, as determined by their CPU utilization. Double threshold values are employed for this purpose. Virtual machines from under-loaded and overcrowded hosts are transferred to hosts that are normal loaded. The authors employed the Water Wave Optimization (WWO) algorithm, a meta-heuristic approach inspired by nature, during the VMC phase. This algorithm determines an appropriate migration strategy to

reduce energy consumption by optimizing resource utilization and turning off idle hosts after migrating their virtual machines to an appropriate target host.

The proposed Energy-Deadline Aware Task Scheduling using the Water Wave Optimization (EDATSWWO) in (Medishetti et al. (2025)) optimizes task scheduling in multi-cloud systems by reducing energy usage, execution time, and meeting deadlines. The method utilizes Water Waves Optimization to balance execution time, energy efficiency, and task priority restrictions.

Rambabu Medara (2023) divide servers into three categories: underloaded, overloaded, and typically loaded depending on CPU load. The system distributes a few virtual machines (VMs) from overloaded machines to usually loaded ones for load balancing. It also moves all VMs from underloaded machines to normally loaded servers to eliminate idle servers. The modified water wave optimization (MWWO) technique is used to create a migration strategy that reduces server overload and maximizes resource efficiency. Overloaded hosts spend more energy over time compared to typical hosts.

3.2.2.5 Salp Swarm algorithms based Integration

Gharehpasha et al. (2020) introduce a novel approach to the optimal deployment of virtual machines by combining the Sine-Cosine and Salp Swarm algorithms as discrete multi-objective and chaotic functions. The initial objective of the proposed approach is to reduce the quantity of physically active devices in cloud data centers, thereby reducing the amount of electricity consumed. The second objective is to strategically position virtual machines on actual equipment in cloud data centers in order to reduce resource waste and manage it. The third objective is to minimize the Service Level Agreement among the active physical computers in cloud data centers. The migration of virtual machines onto actual equipment is prevented from growing by employing the proposed methodology. In the final analysis, the results of the suggested algorithm are compared to those of the First Fit, Modified Best Fit Decreasing, and Virtual Machine Placement Ant Colony System.

Parthiban et al. (2022) present a unique energy-efficient VMP approach for CDCs based on the Disordered Salp Swarm Optimization Algorithm (EAVMP-CSSA). The

EAVMP-CSSA approach aims to minimize CDC energy consumption by reducing the number of active servers hosting virtual machines. The recommended EAVMP-CSSA technique balances active server resources (such as CPU, RAM, and bandwidth) to reduce waste and improve efficiency. The CSSA combines chaotic maps with the Salp Swarm Optimization Algorithm (SSA) to increase performance and minimize computing expenses.

Alresheedi et al. (2019) introduce a multiobjective optimization (MOP) technique that combines salp swarm and sine-cosine algorithms (MOSSASCA) to find the best solution for virtual machine placement (VMP). The proposed MOSSASCA aims to increase mean time before host shutdown (MTBHS), reduce power consumption, and minimize service level agreement violations (SLAVs). The suggested method enhances the salp swarm and sine-cosine algorithms with a MOP strategy. The SCA improves standard SSA performance by adopting a local search method to minimize Entrapment in local optimum solutions and speed up convergence.

3.2.2.6 Ant Colony Optimization based Integration

In order to optimize resource allocation in cloud networks and minimize energy consumption.Sangaiah et al. (2023) aime to develop an intelligent method for dynamic resource allocation that utilizes Takagi–Sugeno–Kang (TSK) neural fuzzy systems and ant colony optimization (ACO) techniques. It utilizes a drop-down window to track CPU usage in order to predict future demands. ACO can decrease its energy consumption by optimizing the migration of virtual machines.

Lilhore et al. (2025) present a novel hybrid optimisation approach that combines water wave optimization (WWO) and ant colony optimization (ACO) to successfully address these problems. ACO specializes at conducting successful local searches, resulting in efficient and high-quality solutions. WWO specializes in globally exploration, providing comprehensive coverage of the solution space. These strategies use their unique advantages to improve reaction times, resource efficiency, and reduce operating costs.

Xing et al. (2021) demonstrate an energy- and traffic-aware ant colony optimization (ETA-ACO) technique. Three new approaches are introduced to improve the performance of ETA-ACO: energy- and bandwidth-aware PM selection, traffic-based VM ordering, and direct information exchange. The first strategy involves two phases for selecting a PM to host a VM. The first phase preserves PMs with low power usage. In the second stage, the one with lowest bandwidth resource use is selected to host the VM. In the second approach, ETA-ACO places VMs by traffic demand. The third strategy creates new solutions by distributing the components of the best solution through a group of produced solutions.

3.2.3 Hybrid meta-heuristics

A hybrid meta-heuristic utilizes more than one meta-heuristic algorithm to schedule tasks and allocate resource on the cloud computing.

3.2.3.1 Genetic Algorithm based Integration

Shishido et al. (2018) investigate the impact of both Particle Swarm Optimization (PSO) and Genetic-based algorithms (GA) on workflow scheduling optimization efforts. The metaheuristics' efficacy is evaluated using a workflow scheduling algorithm that is both cost-effective and secure.

In the virtual machine migration problem, a hybrid optimization algorithm is employed to introduce a novel approach to enhance the energy consumption and execution time of virtual machines. Aron and Abraham (2022) propose a method that is based on the genetic algorithm (GA) and particle swarm optimization (PSO) algorithm, as this issue is one of the popular NP-hard problems. The hybrid algorithm employs a GA to overcome the constraints of PSO algorithms, including weak convergence, stymie in global optima, and artificial intelligence.

A novel secure and multiobjective virtual machine placement (SM-VMP) framework is proposed by Saxena et al. (2021) that has an efficient virtual machine migration. By minimizing intercommunication delays, the proposed framework guarantees an energy-efficient allocation of physical resources among virtual machines. Thereby

Table 3.3 Taxonomy of hybrid meta-heuristics (1)

Author and year	Technique	Performance metric
Feng et al. (2021)	SA Greedy	Energy consumption of servers, cooling and network
Medara and Singh (2021)	Thresholds WWO	Energy consumption Resource usage
Medishetti et al. (2025)	WWO	Energy consumption Execution Time Deadlines
Rambabu Medara (2023)	Modified WWO	Energy consumption Resource usage
Parthiban et al. (2022)	Chaotic Maps SSA	Energy consumption Resource usage Low reject rate
Gharehpasha et al. (2020)	SCA SSA	Energy consumption Resource usage SLA violation
Alresheedi et al. (2019)	SSA SCA	Energy consumption Time SLA violation
Sangaiah et al. (2023)	TSK ACO	Energy consumption Minimization of VM migration
Lilhore et al. (2025)	WWO ACO	Response time Resource usage Costs
Xing et al. (2021)	ACO	Energy consumption Resource usage
Shishido et al. (2018)	PSO GA	Costs Security
Aron and Abraham (2022)	PSO GA	Energy consumption Execution time
Saxena et al. (2021)	WOA GA	Energy consumption Delay Security

Table 3.4 Taxonomy of hybrid meta-heuristics (2)

Author and year	Technique	Performance metric
Goyal et al. (2021)	PSO CSO BAT CSA WOA	Energy consumption Resource usage
Mirmohseni et al. (2021)	PSO GA	Energy consumption Resource allocation Execution cost Makespan
Fu et al. (2021)	PSO Phagocytos GA	Makespan Resource usage Response Time QoS / SLA Deadline
Al-Wesabi et al. (2022)	GTOA RSO	Resource allocation

promoting the secure and timely execution of user applications. The proposed Whale Optimization Genetic Algorithm (WOGA) is used to implement the VMP, that is influenced by nondominated sorting-based genetic algorithms and whale evolutionary optimization.

3.2.3.2 Particle Swarm Optimization based Integration

In order to minimize the energy consumption in the cloud environment, Goyal et al. (2021) implement a variety of optimization algorithms, including particle swarm optimization (PSO), cat swarm optimization (CSO), BAT, cuckoo search algorithm (CSA) optimization algorithm, and whale optimization algorithm (WOA). These algorithms are employed to balance the load, improve energy efficiency, and optimize resource scheduling.

In (Mirmohseni et al. (2021)), combining the results of the particle swarm genetic optimization (PSGO) algorithm and using a combination of the advantages of these two algorithms resulted in improved results and the development of a suitable solution

for load balancing operation, because in the proposed approach (LBPSGORA), instead of randomly assigning the initial population in the genetic algorithm, the best result is obtained by putting the initial population.

Authors in (Fu et al. (2021)) investigated the cloud scheduling tasks process and suggested a particle swarm optimization genetic hybrid method based on phagocytosis (PSO_PGA). The particle swarm is separated into subpopulations utilizing phagocytosis and genetic crossover mutation in order to expand the search range for solutions. The subpopulations are then incorporated, ensuring particle diversity and lowering the risk of the algorithm falling into the local optimum solution. Finally, the feedback mechanism is employed to transmit back the particle's flight experience as well as the companion's flight experience to the next generation particle population, ensuring that the particle population is always moving in the direction of an ideal solution.

3.2.3.3 Rat swarm optimizer algorithm based Integration

Al-Wesabi et al. (2022) introduce novel hybrid metaheuristics for the allocation of energy efficiency resources (HMEERA) in the CC environment. The feature extraction process is initially conducted by the proposed model in accordance with the task demands of numerous clients, and the feature reduction process is conducted using principal component analysis (PCA). The HMEERA technique then employs the integrated features to ensure the most efficient allocation of resources. The HMEERA model is a hybrid of the Group Teaching Optimization Algorithm (GTOA) and the rat swarm optimizer (RSO) algorithm, that is referred to as GTOA-RSO. This algorithm is designed to optimize resource allocation. The optimization of resource allocation among virtual machines in cloud data centers is facilitated by the incorporation of RSO and GTOA algorithms.

3.2.4 Machine learning based algorithms

Machine learning-based algorithms enable systems to make intelligent decisions using historical or real-time data. In the context of task scheduling and resource management, these techniques learn to predict workloads, estimate energy consumption, or adapt

allocation strategies. Their adaptability to complex and dynamic environments makes them a promising alternative to traditional approaches.

A Q-learning based Energy-Efficient Cloud computing (QEEC) is proposed in (Ding et al. (2020)). The framework is based on Q-learning. There are two phases of the QEEC. In the initial phase, the M/M/S queuing model is implemented using a centralized task dispatcher. This model assigns the user requests that arrive to each server in a cloud structure. A Q-learning-based scheduler on each server prioritizes all requests by task laxity and task life time in the second phase. It then uses a continuously-updating policy to assign tasks to virtual machines, applying incentives to reward the assignments that can minimize task response time and maximize each server's CPU utilization.

A Prediction-enabled feedback Control with Reinforcement learning based resource Allocation (PCRA) method is proposed by Chen et al. (2020). Initially, a novel Q-value prediction model is developed to forecast the values of management operations (by Q-values) at various system states. By incorporating the Q-learning algorithm, the model employs multiple prediction learners to generate precise Q-value predictions. Subsequently, the objective resource allocation plans can be identified through the implementation of a novel feedback-control-based decision-making algorithm.

A novel artificial intelligence algorithm, deep Q-learning task scheduling (DQTS), is proposed in (Tong et al. (2019)). This algorithm incorporates the benefits of a deep neural network and the Q-learning algorithm. The objective of this novel methodology is to resolve the issue of managing directed acyclic graph (DAG) duties within a cloud computing environment. The fundamental model learning of this approach is primarily inspired by the popular deep Q-learning (DQL) method, which is used in task scheduling.

In (Cheng et al. (2018)), a novel Deep Reinforcement Learning (DRL)-based Resource Provisioning (RP)and Task Scheduling (TS) system, DRL-Cloud, is introduced to reduce energy costs for large-scale Cloud Service Providers (CSPs) with a large number of servers that receive an immense number of user requests per day. A two-stage RP-TS processor that is based on deep Q-learning is intended to autonomously produce the most optimal long-term decisions by learning from the evolving environment,

Table 3.5 Taxonomy of machine learning based algorithms (1)

Author and year	Technique	Performance metric
Ding et al. (2020)	M/M/S queuing model Q-learning	Energy consumption Response time
Chen et al. (2020)	Q-learning Feed Back-control	Resource usage
Tong et al. (2019)	DQL DNN	Makespan
Cheng et al. (2018)	DQL	Energy consumption Low reject rate Execution time
Qiu (2017)	DRL LSTM	Energy consumption QOS
Belgacem et al. (2023)	ML	Energy consumption Migration Number Network overhead
Mahilraj et al. (2023)	LSTM NBW	Energy consumption Resource usage Makespan Completion time
Choppara and Mangalampalli (2024)	DQN	Energy consumption Makespan Fault tolerance
Wei et al. (2022)	Q-learning	Energy consumption Energy stability
Wang et al. (2023)	ACA	Energy consumption Carbon emissions Waiting time Cooling
Tong et al. (2021)	DDQN	Energy consumption Makespan SLA violation
Panwar et al. (2024)	ML	Energy consumption SLA violation
Ounifi et al. (2022)	MLP DNN LSTM	Energy consumption

Table 3.6 Taxonomy of machine learning based algorithms (2)

Author and year	Technique	Performance metric
Uma et al. (2022)	DRQL	Energy consumption Cost Response time Resource usage
Liang et al. (2021)	K-means KNN	Energy consumption
Zhang et al. (2017)	DQL SAE DVFS	Energy consumption

including realistic electric prices and user request patterns. The proposed DRL-Cloud accomplishes a remarkable high energy cost efficiency, low reject rate, and low duration with rapid convergence by utilizing training techniques such as target network, experience replay, and exploration and exploitation.

Liu et al. (2017) suggest a new hierarchical framework for addressing the general resource allocation and power management issue in cloud computing systems. A global tier is included in the proposed hierarchical framework to allocate virtual machine resources to the servers, while a local tier is used to manage the capacity of local servers in a distributed approach. The global tier problem is resolved by implementing the emergent deep reinforcement learning (DRL) technique, that is capable of addressing complex control problems with a large state space. Additionally, the convergence speed is expedited by the implementation of a novel weight sharing structure and an autoencoder to manage the high-dimensional state space. Conversely, the local tier of distributed server power managements employs a model-free RL-based power manager and an LSTM-based workload predictor that operate in a distributed manner.

The virtual machines migration issue is addressed in (Belgacem et al. (2023)) by employing a machine learning model to decrease the number of virtual machines migration and energy consumption. The Virtual Machine migration based machine Learning Model algorithm (VMLM) that has been suggested is designed to enhance the migration and selection processes of virtual machines. The VMLM (Virtual Machine Local

Migration) algorithm aims to optimize energy consumption in cloud data centers by intelligently performing local migrations of virtual machines. It monitors resource usage, particularly the processor, to identify overloaded or underutilized servers. The VMs are then selected and migrated to target hosts capable of meeting their needs without disrupting the overall system. Unlike traditional methods, VMLM avoids excessive migrations and prioritizes the overall balance of resources, which helps reduce the number of active servers and, consequently, energy consumption.

Mahilraj et al. (2023) suggest a machine learning technique known as short-term or Long-Term Memory (LSTM) for the efficient scheduling of power tasks in order to address the increasing energy and carbon emissions. The scheduling strategy that is most effective takes into account the standardization process and the completion time or exclusive utilization of a resource task. The Novel Black Window (NBW) is employed to enhance the efficacy of LTSM and reduce its weight.

In (Choppara and Mangalampalli (2024)), an advanced fog-cloud integration approach is proposed that employs a deep reinforcement learning-based task scheduler, DRLMOTS (Deep Reinforcement Learning based Multi Objective Task Scheduler in Cloud Fog Environment). This innovative scheduler dynamically allocates computation to either fog nodes or cloud resources by intelligently evaluating task characteristics, including length and processing capacity. Authors formulated the machine learning based Deep Reinforcement Learning (DRL) technique to schedule the tasks in fog layers to minimize makespan, fault tolerance, consumption of energy. Therefore, they have used the DRLMOTS scheduler, which fed priorities to task manager and generates schedules by considering priorities while minimizing the metrics like makespan, fault tolerance, energy consumption.

An energy-saving scheduling strategy based on Q-learning is proposed by Wei et al. (2022). The agent can reduce the energy consumption during task execution by ensuring that the remaining energy of the system is sufficient to maintain the normal execution of the scheduling task and by arranging the task scheduling sequence reasonably manner, based on the real-time required for tasks.

Wang et al. (2023) introduce Eco-friendly Reinforcement Learning in Federated

Cloud (ERLFC), a framework that employs reinforcement learning to schedule tasks in a federated cloud environment. The objective of ERLFC is to intelligently evaluate the condition of each data center and effectively optimize the differences in energy and carbon emission ratios among geographically dispersed cloud data centers in the federated cloud. Authors construct ERLFC using the Actor-Critic algorithm (ACA), that determines the most suitable data center to assign a task based on a variety of factors, including the energy consumption, cooling method, waiting time of the task, energy type, emission ratio, and total energy consumption of the current cloud data center, as well as the details of the next task.

A multi-agent deep reinforcement learning approach for cloud workflow scheduling cost and makespan optimization is introduced in (Tong et al. (2021)). This approach is based on deep Q-learning. The research examines multi-agent cooperation as a Markov game with a connected equilibrium in order to prevent the makespan and cost agents from unilaterally deviating from the joint distribution. The on-demand access to resources globally that cloud computing provides is a result of its accelerated growth, that also has a substantial carbon impact and high power consumption.

For the sake of mitigating environmental impact, reducing operational costs, and guaranteeing sustainable growth, it is imperative to optimize their energy utilization, as they utilize substantial quantities of energy. To address this issue, researchers have investigated effective energy-saving strategies that employ machine learning techniques (Panwar et al. (2024)). By analyzing data, identifying patterns, and optimizing resource utilization, Machine Learning methods have the potential to significantly improve energy efficiency in Cloud data centers (CDCs). The primary objectives are to optimize energy utilization and acquire resources by predicting CPU usage, identifying overloads, estimating under-loads, selecting, migrating, and relocating virtual machines.

In order to forecast Power Usage Effectiveness (PUE) values, researchers introduce three machine learning models: Multilayer Perceptron (MLP), Resilient Backpropagation-based Deep Neural Network (DNN), and Attention-based Long Short-Term Memory (LSTM) (Ounifi et al. (2022)). DC energy efficiency can be measured and optimized through the use of Power Usage Effectiveness (PUE). Consequently, it is difficult to

make precise predictions regarding Power Usage Effectiveness (PUE).

(Uma et al. (2022)) introduce a new artificial algorithm, Deep Reinforcement Q-learning (DRQL), for resource scheduling. The purpose of this novel methodology is to address the issue of managing energy consumption in a cloud computing environment.

The virtual machine and physical machine models are methodically analyzed by Liang et al. (2021). Simultaneously, the K-means clustering algorithm for unsupervised learning and the KNN classification algorithm for supervised learning are improved to establish a dynamic hybrid resource deployment rule. Subsequently, a dynamic hybrid machine learning (EHML)-based energy-aware resource deployment algorithm for cloud data centers is proposed in accordance with the theory of machine learning. The energy consumption is reduced by this algorithm, that increases the average utilization of physical machines.

DQL-EES is a scheduling scheme for periodic tasks in real-time systems that is energy-efficient and is based on a deep Q-learning model (Zhang et al. (2017)). The feature is the integration of a Stacked Auto-Encoder (SAE) into the deep q-learning model to supplant the Q-function in the process of learning the Q-value of each DVFS technology for any system state.

3.3 Conclusion

The state of the art presented in this chapter clearly shows the interest given to the problem of energy consumption. Several techniques have been proposed to reduce this consumption. The following chapters present our contributions according to the issues raised in this thesis. The thesis position is also discussed in this chapter in relation to previous research. Developing models and algorithms for resource allocation in cloud data centers while increasing energy efficiency is the primary focus of this thesis. Our contributions to the research direction are detailed in the following chapters.

CHAPTER 4

Energy-aware scheduling of tasks in cloud computing

4.1 Introduction

Cloud computing has emerged as a key paradigm in the world of computing. It contributes to the increasing expectations for availability and flexibility. Users of the Internet and computers are becoming more interested in the services proposed by the cloud computing providers due to its impressive growth in recent years.(Goyal et al. (2021)) Energy consumption is a crucial topic in cloud computing that has become a significant issue. It requires appropriate solutions and several data centers contain servers, cooling systems, switching and network components that make up the cloud computing infrastructure.(Mboula (2021)) The energy consumed by data centers has increased due to the rising demand for cloud infrastructure that has become a serious problem. Higher expenses of profit and CO2 emissions result from the excessive energy used. Therefore, efficient solutions are required to reduce the negative effects on the environment and cloud provider profit. Every year, energy cost rises and several studies have examined how much energy is needed by data centers and individual servers (Choppara and Mangalampalli (2024)). Numerous studies have been launched on the subject of energy and power in computing systems. The creation of virtual machine (VMs) within a physical server is made possible by virtualization technology that also enables to utilize resources more efficiently while using less hardware (Medara and Singh (2021)). Task scheduling and energy efficiency are two key obstacles in resource allocation (Hassan et al. (2020)). This chapter presents an Energy-Aware Scheduling Model (EASM) for task scheduling in cloud computing. The objective of the proposed model is to re-

duce the energy consumption, execution time, and SLA violation. EASM works in two phases, i.e., pre-processing and optimization with Adaptive Genetic Algorithm. In the first phase, tasks with longer execution times are allocated in VMs with high processing capabilities(Malik et al. (2021)). In the next phase, GA is used to optimize scheduling and find better solutions. In the popular meta-heuristic method known as the genetic algorithm, populations of potential candidate solutions, known as individuals, are developed over many generations to find the best solution for a specific problem. With the contribution of various genetic operations, the optimization begins with random individuals and eventually reaches the global optimum (Nahhas et al. (2021)). The simulations' results confirm that the suggested approach is more robust and efficient in terms of energy usage, execution time, and SLA violations.

4.2 Related Work

For load balancing, energy efficiency, and better resource scheduling, an effective cloud environment, (Feng et al. (2021)) using a variety of optimization algorithms, including the Whale Optimization Algorithm (WOA), Cat Swarm Pptimization (CSO), Cuckoo Search Algorithm (CSA), BAT, and Particle Swarm Optimization (PSO) is created. The suggested work employs a cost-effective solution to the load balancing and resource scheduling issues.

Ibrahim et al. (2018) have selected two advanced scheduling algorithms to examine the outcomes in a same cloud computing environment and examine the approaches that maximize energy and cost in a cloud computing environment. The main objective of the Energy-Efficient Strategy (EES) is to spread out the maximum load over the fewest possible virtual machines. By assigning appropriate resources to the required tasks, Cost-based Scheduling using Genetic Algorithm minimizes execution time that decreases user costs. The results are then studied and compared to other scheduling algorithms, such as Round-Robin (RR) and First-come- First-served (FCFS).

Kakkottakath Valappil Thekkepuryil et al. (2021) present an Integer Linear Programming (ILP) model for cloud computing energy optimization and an Adaptive Ge-

netic Algorithm (GA) for dynamic work scheduling in the cloud data center. In order to account for the dynamic nature of the cloud environment and to offer a near-optimal scheduling solution that reduces energy usage, an Adaptive Genetic Algorithm (GA) is developed. By allocating incoming tasks to resources in a way that both user needs and the energy consumption of cloud data centers are fulfilled. This study attempts to establish a model and an algorithm for reducing the energy consumption in a cloud computing infrastructure. It concentrates on a single Cloud data center as its environment settings.

Medara et al. (2021), authors suggest to use an energy-aware workflow scheduling technique for cloud computing with VM consolidation. The suggested EASVMC technique is designed to achieve many objectives including resource usage, VM migrations, and energy consumption. Task scheduling and VM consolidation are the two stages of the EASVMC algorithm's operation (VMC). The virtual machine that will consume the least amount of energy during the first phase is assigned to the task with the longest possible execution time. The second phase includes a well-known NP-hard issue, namely VM consolidation. Based on CPU utilization, the VMC phase divides the physical hosts into hosts with a regular load, under-loaded hosts, and overloaded hosts. Therefore, double threshold values are employed. Migration of virtual machines from overloaded and underloaded hosts to normally loaded hosts. Authors used the Water Wave Optimization (WWO) algorithm, a nature-inspired meta-heuristic approach, for the VMC phase. This algorithm finds an appropriate migration plan to reduce energy consumption by increasing overall resource utilization and switching off idle hosts after migrating their VMs to an appropriate target host. They evaluated the effectiveness of this algorithm in comparison to three well-known methods: HEFT, EES, and PESVMC. The simulation results demonstrated that the EASVMC algorithms surpass the other three techniques in terms of overall performance.

To overcome the drawbacks of task consolidation and scheduling, (Panda and Jana (2019)) proposed an energy-efficient task scheduling algorithm (ETSA). The proposed algorithm uses a normalization process to determine when to schedule tasks while taking into consideration their completion times and resource use overall. The energy

efficient task-scheduling algorithm that is presented for reducing energy consumption and execution time is the foundation of this study. For heterogeneous cloud computing systems, the authors created an online energy-efficient work scheduling system. The proposed system can be used for cloud, application, energy, and scheduling models. In order to decide on scheduling, the method computes the completion time and overall resource usage of a job on the resources.

Choudhary and Perinpanayagam (2022) suggested a new approach based on multi-objective optimization. For complicated VM scheduling solutions, they calculate the amount of energy used, the CPU usage, and the number of instructions performed in each scheduling period. Multi-objective PSO (particle swarm optimization) optimization can lead to better and more efficient results for various parameters than multi-objective GA (genetic algorithm) optimization in terms of energy efficiency and execution time reduction.

Badr et al. (2022) focused on the issue of power consumption and proposes a powerful method called Task Consolidation based Power Minimization (TCPM). It effectively allocates jobs to the cloud environment's available resources in order to reduce power consumption. The best-fit approach is employed to achieve the optimum resource usage and prevent energy waste in the proposed TCPM algorithm that improves and incorporates various advantages of the current algorithms. The results of the proposed TCPM algorithm are compared with FCFS, WWO, and MCT algorithms using the CloudSim toolkit.

Malik et al. (2021) suggested a method for effective scheduling and improved resource usage based on task categorization and thresholds. Workflow tasks are preprocessed in the first stage to prevent bottlenecks by separating tasks with high dependencies and lengthy execution durations. The following phase is classifying tasks according to the intensity of the resources needed. To choose the optimum schedules, Particle Swarm Optimization (PSO) is employed. To verify the suggested approach, experiments were done. Comparative results from benchmark datasets are given. The findings demonstrate how the suggested algorithm performs better than the other algorithms in terms of energy usage, execution time, and load balancing.

Gharehpasha et al. (2021) developed a novel method for optimum placement of virtual machines utilizing a combination of the Sine-Cosine and Salp Swarm algorithms as discrete multi-objective and chaotic functions. The initial objective of the suggested method was to decrease the amount of electricity used in cloud data centers by reducing the quantity of physically active devices. The second objective was to decrease resource waste and control it by strategically placing virtual machines on actual equipment in cloud data centers. The third goal was to keep Service Level Agreement amongst the active physical computers in cloud data centers to a minimum. By using the suggested approach, the migration of virtual machines onto real equipment is prevented from growing. In the end, the suggested algorithm's results were compared with the results of First Fit, Modified Best Fit Decreasing, and Virtual Machine Placement Ant Colony System.

In (Garg et al. (2021)), In order to schedule the workflow tasks to the VMs and dynamically deploy/undeploy the VMs in accordance with the workflow task's needs, an energy and resource efficient workflow scheduling algorithm (ERES) is presented. To determine the energy consumption of the servers, an energy model is offered. It uses a double threshold strategy to determine if the server is overloaded, underloaded, or operating normally. Live VM migration is used to balance the load on the overloaded/underloaded servers. Live VM migration strategy is used. Extensive simulation tests are run to evaluate the efficacy of the suggested approach. On the basis of resource utiliztaion, energy efficiency, and task execution time, the suggested approach is compared to the PESVMC (power efficient scheduling and VM consolidation) algorithm. Additionally, the results are validated in a genuine cloud environment.

Shishido et al. (2018) investigated the effectiveness of using meta-heuristic techniques for scheduling cloud processes. The purpose of this study was to evaluate the effects of GA and PSO augmentation on workflow scheduling optimization. To assess the competency of the meta-heuristic technique, a cost-aware workflow scheduling issue was used. PSO, GA, and Multi Population GA meta-heuristics were also used in the experiments. The evaluation of meta-heuristic algorithms was based on the objectives of cost minimization and time for response. These algorithms produced more effective

schedules that reduce costs in a reasonable amount of time.

The proposal of (Alasady et al. (2023)) presents a multi-objective optimization method for cloudlet computing that makes use of the non-dominated sorting idea. The objectives taken into consideration include delay, user energy consumption, cloudlet energy consumption, and cost, which are determined by the number of cloudlets. Non-dominated sorting genetic algorithms (NSGA-III and NSGA-II) are employed to be compared to this proposed work.

In (Gourisaria et al. (2021)), authors offer a task scheduling heuristic for heterogeneous cloud systems that saves energy. It performs by choosing the best physical host with virtual machines while taking into account the utilization of any incoming tasks on that specific virtual machine. They demonstrate the superiority of the proposed heuristic in energy-efficient task scheduling in heterogeneous cloud settings by comparing its energy efficiency with other previous methods, including ECTC, MaxUtil, Random, and FCFS, on both synthetic and benchmark datasets.

The authors in (Vijaya and Srinivasan (2024)) provide a new hybrid method for effective virtual machine placement that combines the Sine Cosine Algorithm (SCA) with the Ant Colony Optimization (ACO) algorithm. The results obtained by the ACO algorithm have been examined using SCA, an advancing search method that makes use of the Sine and Cosine functions in the engineering domain.The ACO method has been utilized to exploit the search space's solutions for effective virtual machine placement, hence facilitating power management and reducing resource wastage.

The table 4.1 illustrates a summary that compares the reviewed approaches in terms of key performance metrics, such as energy efficiency, execution time, and SLA violation rates.

4.3 The proposed model

In this section, discussion of the system model and energy model is followed by the details of each phase.

Year	Approche	Energy consumption	Execution time	SLA violation
2021	Feng, H. et al.	Yes	Yes	No
2018	Ibrahim, H. et al.	Yes	Yes	No
2021	Thekkepuryil, J.K.V. et al.	Yes	Yes	Yes
2021	Medara, R. et al.	Yes	Yes	No
2019	Panda, S.K.	Yes	Yes	No
2022	Rajkumar Choudhary et al.	Yes	Yes	No
2022	Shaimaa Badr et al.	Yes	Yes	Yes
2021	Nimra Malik et al.	Yes	Yes	No
2021	Sasan Gharehpasha et al.	Yes	No	Yes
2021	Neha Garg et al.	Yes	Yes	Yes
2018	Shishido, H., Y. et al.	No	Yes	Yes
2023	Ali Salah Alasady et al.	Yes	No	Yes
2021	Mahendra Kumar Gourisaria et al.	Yes	Yes	No
2024	C.Vijaya et al.	Yes	Yes	Yes

Table 4.1 Summary table

4.3.1 System Model

Scheduling is the process of allocating a number of tasks to a number of resources (virtual machines). In the cloud data centers, there are two levels of scheduling: (i) series of rules for deploying VMs at the server level and (ii) rules for assigning tasks to VMs. The main focus of our contribution is VM-level task scheduling techniques. The scheduling approach is a strategy for selecting which resources to use to execute tasks in order to shorten execution times and conserve energy.

Consider the Cloud Data Center (CDC) consists of N physical machines (PM). It can be represented in Eq. 4.1:

$$CDC = \{PM_1, PM_2, ..., PM_N\} \tag{4.1}$$

Where $(i = 1, \ldots, N)$ denotes the PMs presented in the CDC. The features of PM_i are defined in Eq.4.2:

$$PM_i = \{C_i, Size\_PM_i, RAM\_PM_i, Bandwidth\_PM_i, \#Core\_PM_i\} \tag{4.2}$$

Consider the physical machine consists of M virtual machines (VMs) It can be

Symbol	Description
CDC	Cloud Data Center.
PM_i	Physical machine. i = (1,. . . ,N)
N	Number of physical machines in the cloud environment
$Size\_PM_i$	The size of PM_i.
RAM_PM_i	The RAM of PM_i.
$Bandwidth\_PM_i$	The Bandwidth of PM_i.
$\#Core\_PM_i$	Number of cores in PM_i.
VM_{ij}	Virtual machine. j=(1,. . . ,M)
M	The number of virtual machines in the cloud environment.
$Size\_VM_{ij}$	The size of VM_{ij}.
RAM_VM_{ij}	The RAM of VM_{ij}.
$Bandwidth\_VM_{ij}$	The Bandwidth of VM_{ij}.
$\#Core\_VM_{ij}$	Number of cores in VM_{ij}.
$Tasks_k$	The tasks submitted by the users in DCD. k=(1,. . . ,Ntsk)
Ntsk	The number of tasks submitted in the cloud environment, where Ntsk = L+P.
$Tasks\_D_{kd}$	Set of tasks with deadline constraints. Kd=(1,. . . ,P)
P	The number of deadlined tasks submitted in the cloud environment.
$Tasks\_O_{ko}$	Set of tasks without deadline constraints. Ko=(1,. . . ,L)
L	The number of no_deadlined tasks submitted in the cloud environment.
Td_{kd}	Deadlined Task.
TO_{ko}	No-deadlined task.
$Length_{kd}$	The length of deadlined task.
$FileSize_{kd}$	The size of deadlined task.
$Deadline_{kd}$	Time till which the tasks should be finished.
$Length_{ko}$	The length of no-deadlined task.
$FileSize_{ko}$	The size of no-deadlined task.
C_i	The total processing capacity of PM_i.
c_j	The processing capacity of VM_{ij}.
u_i	The current CPU utilization of PM_i.
P_i	The power of PM_i.
P_{max}	The maximum power of a physical machine.
E_i	The total energy consumption PM_i.
ECT_k	The required execution time of task on VM_{ij}.
$\%SLAviolation_i$	The percentage of tasks that have exceeded their deadlines in PM_i.

Table 4.2 Symbols used in the proposed method.

represented as in Eq.4.3:

$$PM_{ij} = \{VM_{i1}, VM_{i2}, ..., VM_{iM}\} \tag{4.3}$$

Where $(j = 1, \ldots, M)$ is the number of virtual machines obtained from PM_i. The features of VM are defined in Eq.4.4:

$$VM_{ij} = \{c_{ij}, Size\_VM_{ij}, RAM\_VM_{ij}, Bandwidth\_vm_{ij}, \#Core_{ij}\} \tag{4.4}$$

The tasks submitted by the users can be represented as in Eq.4.5:

$$Tasks = Tasks\_D \cup Tasks\_O \tag{4.5}$$

Where Tasks_D is set of tasks submitted by users with the consideration of deadline constraints. Tasks_O is set of tasks submitted by users without the consideration of deadline constraints.

$$Tasks\_D_{kd} = \{TD_1, TD_2, ..., TD_P\} \tag{4.6}$$

where P is the number of tasks submitted with the consideration of deadline constraints.

$$Tasks\_O_{ko} = \{TO_1, TO_2, ..., TO_L\} \tag{4.7}$$

where L is the number of tasks submitted without consideration of deadline constraints. The features of Tasks_D and Tasks_O are defined in Eq.4.8 and 4.9:

$$TD_{kd} = \{length_{kd}, FileSize_{kd}, Deadline_{kd}\} \tag{4.8}$$

$$TO_{ko} = \{length_{ko}, FileSize_{ko}\} \tag{4.9}$$

4.3.2 Energy Model

The processing capacity c_{ij} of a resource VM_{ij} is computed with the MIPS of each VM. The capacity of M VMs is calculated with Eq.4.10.

$$C_i = \sum_{j=1}^{M} c_{ij} \tag{4.10}$$

In cloud computing, resource use has a major effect on how much energy is used. The utilization can be calculated with Eq.4.11.

$$u_i = \frac{\sum_{j=1}^{M} c_{ij}}{C_i} \tag{4.11}$$

Where M is the number of VMs running on PM_i, and c_{ij} refers to the computing allocated to VM_{ij}. C_i is the total processing capacity of the PM_{ij}. This research examines CPU use that determines how much electricity physical devices consume. About 70% of the power of a physically active machine is used when it is inactive. So, using Eq. 4.12, the power consumption (u) as CPU utilization is defined as:

$$P(u)_i = P_{max}(0,7 + 0,3u_i) \tag{4.12}$$

where u_i is the current CPU usage and P_{max} is the maximum power of a physical system operating at 100% CPU utilization. CPU usage is defined as a function u(t) of time since it varies over time. As a result, Eq.4.13 establishes a physical machine's (PM_i) total energy consumption:

$$E_i = \int P(u(t))\,\mathrm{d}t \tag{4.13}$$

4.3.3 Scheduling Model

The main objective of the suggested approach is to decrease the amount of energy used, the execution time, and SLA violations of the cloud resources while taking diverse users priorities into account and optimizing the energy and execution time under the deadlines constraints. We propose a tasks scheduling model (**?**) in cloud computing that treats two sets of tasks. The first set of tasks takes priority since the users require the deadline unlike the second set of tasks. The respect of deadline of first set will involve more energy consumption compared to the energy consumption of the second

set. Two possible scenarios are distinguished, one for deadlined tasks and the other for no-deadlined ones. In these two cases, two phases are applied. In the first phase, thresholds are used for the tasks length. Tasks with longer execution times are allocated in VMs with high processing capabilities. Once the energy consumption reaches a threshold. The second phase will be launch. The genetic algorithm is a global scheduler that allocates incoming cloud tasks to suitable VMs. These two phases are used to reduce the execution time applying to decrease energy consumption as resources are utilized efficiently.

4.3.3.1 Task allocation phase

In the first phase, a new method that is proposed and intended to dynamically prioritize the tasks and schedule them to the best suitable selected resource. The tasks in Cloud Computing require to be executed by the available resources to achieve minimal total time for completion. The expected completion time for the task is defined in Eq.4.14:

$$ECT_k = \frac{Length_k}{c_{ij}}; k = 1, 2, ..., Ntsk; \\ i = 1, 2, 3, ..., N; j = 1, 2, 3, ..., M \quad (4.14)$$

Where ECT_k is the time needed for the kth task to execute on the Mth virtual machine and Nth physical machine, where N is the number of PMs and M is the number of VMs and $Ntsk$ is the number of tasks.

$Length_k$ is the length of a task in Million Instruction (MI) and c_{ij} is the VM_{ij} speed Million Instructions Per Second (MIPS).

Allocation of tasks with deadline constraint

$$Tasks\_D_{kd} = \{TD_1, TD_2, ..., TD_P\} \quad (4.15)$$

Where P is the number of tasks submitted with the consideration of deadline constraints. Tasks_D is the first set of tasks requiring top priority processing compared to

the second set of tasks to avoid SLA violation. The execution time ECT of each task should be less or equal then the deadline.

$$(ECT_{kd} \leq Deadline_{kd}) AND (Min(ECT_{kd})) \tag{4.16}$$

Task allocation depends on the length of each task and respecting the deadline constraint where the proposed algorithm sets a threshold for the length of the tasks and the threshold values are applied. They are intended to prioritize tasks during execution. To decrease the total execution time, the lengthier tasks need to be processed first. High processing capacity virtual machines are assigned to these tasks.

Allocation of tasks without deadline constraint

$$Tasks\_O_{ko} = \{TO_1, TO_2, ..., TO_L\} \tag{4.17}$$

where L is the number of tasks submitted without consideration of deadline constraints. Tasks_O is the second set of tasks requiring second priority processing.

$$Min(ECT_{ko}) \tag{4.18}$$

Task allocation depends on the length of each task where the proposed algorithm sets a threshold for the length of the tasks and the threshold values are used to prioritize tasks during execution. As a result, each task's priority is established according to its duration. Tasks with longer length need to be processed with priority and VMs with high processing capabilities are allocated to these tasks.

4.3.3.2 Task scheduling phase

In the second phase, the proposed algorithm uses two modified Genetic Algorithm (GA) to optimize scheduling and find better solutions for the two sets of tasks (with deadline and without deadline).

Encoding

The choice is to use a direct representation; Table 4.3 illustrates the encoding represen-

tation. In the suggested example, there are two major information. The tasks that are scheduled and the number of the VM instance to which it is assigned are shown in table 4.3.

Table 4.3 Encoding

Task ID	T1	T2	...	Tn-1	Tn
VM ID	VM1	VM3	...	VM2	VM4

Initial Population

The creation of an initial population of T-size solution candidates for evolution is the first stage in the optimization utilizing genetic algorithms process. T-size refers to the population size. Every population set has numerous chromosomes containing genes corresponding to various tasks planned on distinct virtual machines. The chromosome of the initial population is presented in table 4.4.

Table 4.4 Initial Population

Task ID	T1	T2	T3	T4	T5
VM ID	VM1	VM3	VM2	VM2	VM3

Fitness function:

Which chromosomes to pass on to the following generation depends critically on their level of fitness.

Fitness Function for set of tasks with deadline constraint:

The fitness value of deadlined tasks is defined in Eq.4.19:

$$\begin{cases} \text{Fitness Function 01} = \alpha \sum_{i=1}^{N} E_i + \beta \sum_{i=1}^{N} \%\text{SLA violation}_i, \\ \alpha + \beta = 1 \end{cases} \tag{4.19}$$

Where E_i refers to the energy consummated by the PM_i. $\%SLA\_violation$ refers to the percentage of tasks that have exceeded their deadlines in PM_i.

Fitness Function for set of tasks without deadline constraint is defined in Eq.4.20:

$$fitnessfunction02 = \sum_{i=1}^{N} E_i \tag{4.20}$$

where E_i refers to the energy consummated by the PM_i. Therefore, the population's best and worst chromosomes have the lowest and highest fitness rates, respectively.

Selection operation

Populations are ranked according to their fitness levels, choosing the best elite chromosomes of a predefined size, and passing them on to the following generation.

Crossover

In GAs, the crossover operator is crucial for changing the population chromosomes. The crossover operator improve population evolution in GAs. The operator joins several chromosomes to form a new generation of chromosomes. While certain characteristics are inherited from both parents, others are inherited from one parent only. The individuals from the previous stage are used in this study. They go through a process called crossover where genes are exchanged at random crossing points. As seen in tables 4.5,4.6,4.7 and 4.8, Chosen individuals will produce two offspring following a crossover.

Table 4.5 Parent 1

Task ID	**T1**	T2	**T3**	T4	**T5**
VM ID	**VM1**	VM3	**VM2**	VM2	**VM3**

Table 4.6 Parent 2

Task ID	**T1**	T2	**T3**	T4	**T5**
VM ID	**VM3**	VM2	**VM1**	VM3	**VM2**

Table 4.7 Offspring 1

Task ID	**T1**	T2	**T3**	T4	**T5**
VM ID	**VM3**	VM3	**VM1**	VM2	**VM2**

Table 4.8 Offspring 2

Task ID	**T1**	T2	**T3**	T4	**T5**
VM ID	**VM1**	VM2	**VM2**	VM3	**VM3**

Mutation

By changing chromosomes, mutations are used to maintain population variety. To create variation in the population, many chromosomes are mutated by the mutation operator after being combined using the combination operator. As indicated in tables 4.9 and 4.10, a random VM has been provided to a task from the list of tasks at random.

Table 4.9 Before Mutation

Task ID	T1	T2	**T3**	T4	T5
VM ID	VM1	VM2	**VM2**	VM3	VM3

Table 4.10 After Mutation

Task ID	T1	T2	**T3**	T4	T5
VM ID	VM1	VM2	**VM3**	VM3	VM3

Termination conditions

For each objective function, the individual of each generation is compared to the previous best fitness value. If the new individual outperforms the old one, the best value is updated. The suggested method ends when all of the chromosomes, or solutions, converge to the same degree of fit. However,there are no further improvements to the fitness value.

4.4 Experimental evaluation

The experiments conducted to evaluate the suggested energy-aware scheduler are presented in this section. The evaluation configuration, comprising the cloud infrastructure, the scheduler algorithm, and the machine utilized to perform the scheduling, is discussed in the initial part of this section. The results of the scheduling for the various

situations are shown in the second section. A series of experiments are completed to assess the effectiveness of the scheduling algorithm after analyzing the impact of different factors and algorithms on the execution time and energy consumed.

The research presents a unique method of work scheduling in cloud settings that is solely evaluated with the CloudSim simulator and makes use of modified genetic algorithms. This approach outperforms conventional heuristic techniques in terms of minimizing SLA violations, reducing execution time, and optimizing consumption of energy. In practical terms, the approach can save energy costs and increase operational efficiency for cloud data centers.However, it still has to be verified in real-world scenarios. Before a large-scale implantation, a phased approach that begins with controlled test settings is advisable. Cloud service providers may manage varying workloads with more efficiency and dependability by using this strategy.

4.4.1 Simulation experiments

In order to evaluate the proposed model, the proposed solutions has been implemented using the CloudSim simulator.

4.4.1.1 Cloud infrastructure

In this simulation experiments, one data center was created and contained a number of PMs. A variety of VMs types are created in this simulation environment. The specific parameters are listed in Table 4.11.

In this research, several important factors inform the choice of simulation settings. Firstly, representativeness is essential; the simulations are pertinent since the characteristics selected match common data center schemes based on previous studies. Second, to represent a wide range of system behaviors and situations, different Physical Machines (PMs), Virtual Machines (VMs), and tasks are chosen. Thirdly, the use of standard setups ensures repeatability, making it easier to compare the results with previous studies and increasing their validity.

We treat two sets of tasks. The first one takes priority since the users require the deadline unlike the second one. The respect of the deadline will involve more energy

Entity Type	Parameters	Values
Data Center	Number of Data Center	1
PM	Number of PM	50
	C (MIPS)	4000-8000
VM	Number of VM	10-60
	C(mips)	1000-4000
Tasks	Number of Tasks	10-10000
	Length (MI)	10000-30000

Table 4.11 The Resources Parameters.

consumption compared to the energy consumption by the second set.

4.4.1.2 Scheduler configuration

After the submission of the tasks by the users, this study seeks to allocate the tasks with deadlines to the first. Then, we allocate the tasks without deadline. In this phase, we allocate each task to the fastest VM. Finally, we control the energy levels based on an energy threshold.

Threshold for energy consumption: Once the energy consumption reaches this threshold, we will launch the scheduling phase based on Genetic Algorithm.

Genetic Algorithm: Initialization and looping algorithms were divided into two categories. The optimal solution was identified by evaluating the fitness values after a random viable solution had been created during the initialization procedure. Subsequently, the looping segments confirmed if a certain terminal condition was satisfied. The mutation, crossover, and selection processes were used in order throughout the continuous loop. In the end, the process of iteration produced the optimum solution.

A sensitivity study is conducted to evaluate the model's resilience under various conditions. This included analyzing workload intensity variations to comprehend how varying demand levels affected system performance. Also, modifications were examined to the VM and PM setups to determine how resource allocation influenced results. Additionally, a variety of resource management strategies were evaluated, focusing on how different work allocation techniques affected system performance. This thorough examination strengthened the model's validity and resilience, guaranteeing its depend-

ability in a variety of situations.

4.4.1.3 Experimental Results

The algorithms were evaluated in terms of execution time, energy consumption, and SLA violation. To confirm the effectiveness of the approaches over the ones already in use, an extensive statistical studies were conducted, including comparison tests. A statistically meaningful improvement is shown from the results.

In the simulation experiments, we compare the proposal with:

Naïve Genetic Algorithm (NGA): in this experiment, we allocate user tasks by First Come First Serve (FCFS) technique and we use GA after reaching the energy threshold without difference between deadline and no-deadline tasks.

Round-Robin: we allocate deadline tasks by Round-Robin technique and the no-deadlined tasks with the FCFS technique. The proposal treats two priorities of two types of tasks that are deadlined and no-deadline tasks. The deadline tasks take the first priority to involve violations. In these two cases of tasks, two thresholds are proposed, one for tasks length and the other for energy consumed. Tasks with length longer than first threshold are allocated in VMs with high processing capabilities. Once the energy consumption reaches the second threshold, we will launch the genetic algorithm.

Execution time

First, we evaluated the performance of our algorithm by varying the number of tasks from 50 to 600.(As shown in Table 4.11)

Experiment 1: We changed the number of tasks as indicated in Figure 4.1 and measured the performance efficiency using a fixed number of virtual machines (VMs) of 30.

Experiment 2:

As seen in Figure 4.2, we set a limit of 30 tasks and a range of 10 to 50 virtual machines. Compared to the suggested approach, NGA and RR take longer in both scenarios to accomplish a task.

Experiment 3: In the third scenario, the number of VMs and tasks are not fixed (as shown in Figure 4.3).

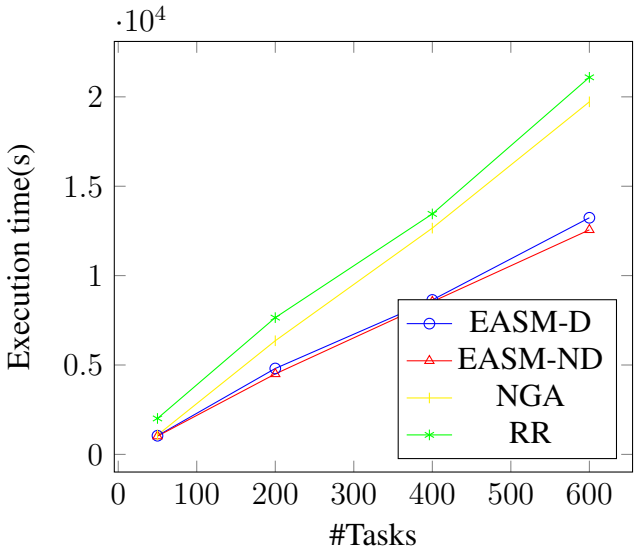

Fig. 4.1 Execution time(s) of different numbers of deadlined and no-deadlined tasks.

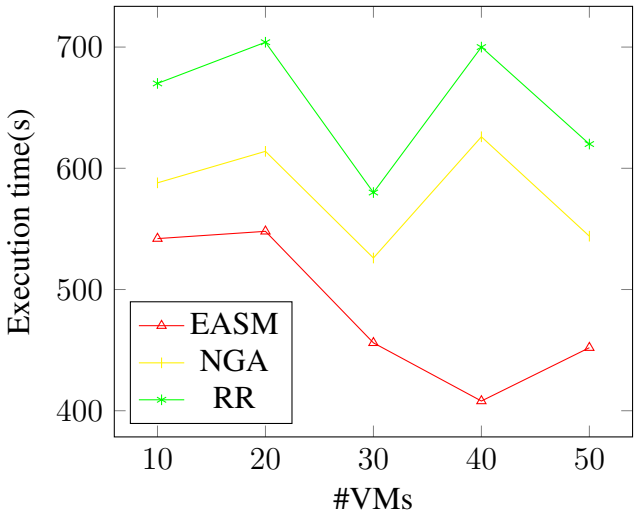

Fig. 4.2 The Execution time(s) of different numbers of VMs.

Figure 4.1, 4.2, and 4.3 show the comparative analysis of the execution time of set of algorithms. The three figures present the evaluation results of EASM vs. NGA and RR. Figure 4.1 presents the execution time regarding tasks number. Figure 4.2 presents the execution time regarding VMs number and Figure 4.3 presents the execution time in different experimentations. As shown, EASM outperforms NGA and RR. This is due to using the proposed algorithm that decreases the execution time. The execution time of EASM is satisfying when compared with the NGA and RR since it is based on task classification and thresholds. The model has the potential to improve the execution time speed and optimization efficiency of the EASM. Even when the number of tasks rises,

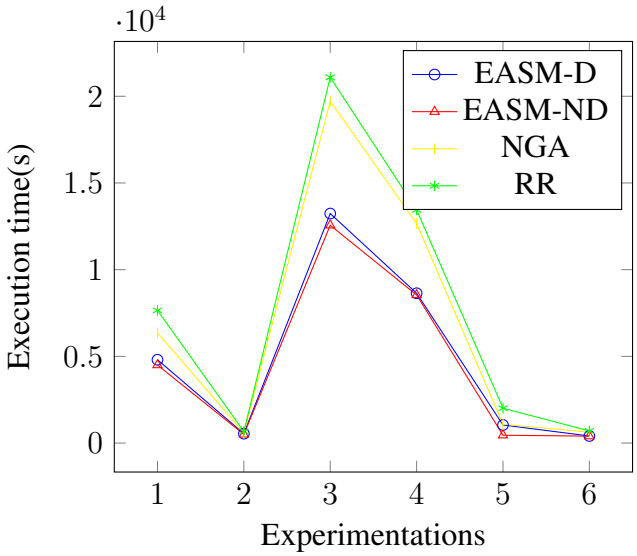

Fig. 4.3 The Execution time(s) of different experimentation.

EASM has strong capacity to assess the outcomes attained, identify the greatest fitness value, and make the best decision. The respect of the deadline of the deadlined-tasks involves more execution time compared to the execution time of the no-deadlined ones.

Energy consumption

Now, we evaluated the performance EASM for energy consumption by varying the number of the tasks from 50 to 600.

Experiment 4: The fourth scenario of the experiments evaluated the energy consumption by the fixed number of VMs at 30 and a changed number of tasks as shown in Figure 4.4.

Experiment 5:

As shown in Figure 4.5, the tasks in this scenario are fixed at 30 and the range of virtual machines (VMs) is 10 to 50, increasing by 10.

Figure 4.4 and 4.5 show the energy consumption comparison between deadlined and no-deadlined tasks of the proposed EASM, NGA and RR. The energy consumed by the EASM is considerably less than the NGA and RR. It is evident that there is a significant difference among the compared algorithms and EASM consumes less energy in different tasks. The respect of the deadline of the first set involves more energy

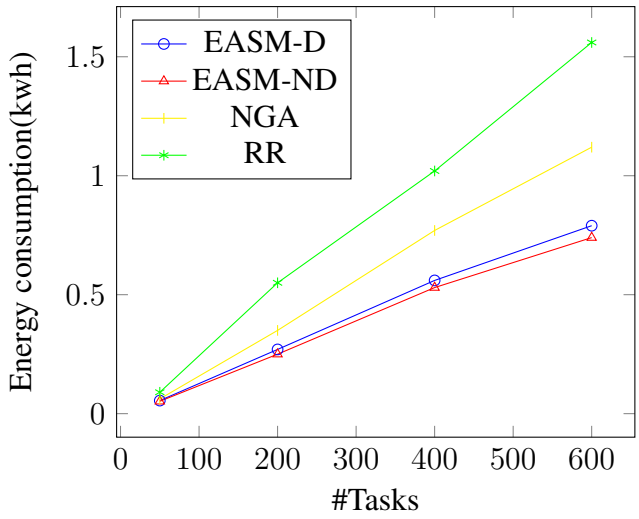

Fig. 4.4 The energy consumption (Kwh) of different numbers of tasks.

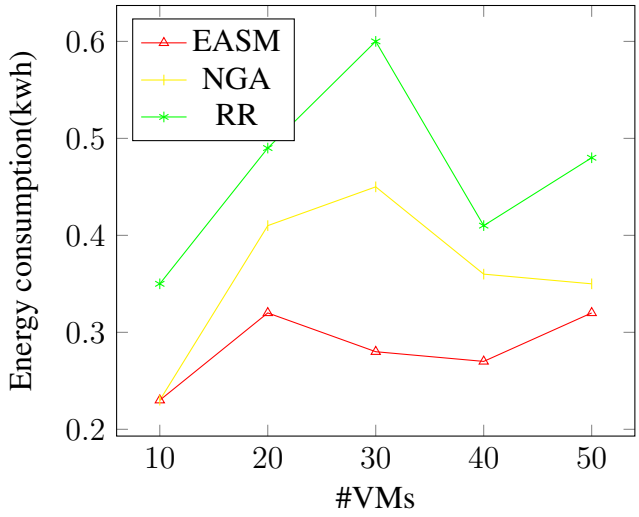

Fig. 4.5 The energy consumption (Kwh) of different numbers of VMs.

consumption compared to the energy consumption of the second set of tasks.

Experiment 6: Energy consumption with the changing of VMs and Tasks.

In this scenario, the number of VMs and tasks are not fixed as shown in Figure 4.6.

Figure 4.6 presents the energy consumption in different experimentation. Our EASM outperforms NGA and RR. This is due to using the proposed algorithm that decreases the consumed energy. The energy consumption of the proposed algorithm is better when compared to the NGA and RR.

Experiment 7: The average SLA violation rate for all methods are shown in Figure 4.4.1.3. In comparison to different approaches, EASM produced the lowest rate of SLA violations. The obtained results confirm the effectiveness of the model in minimizing

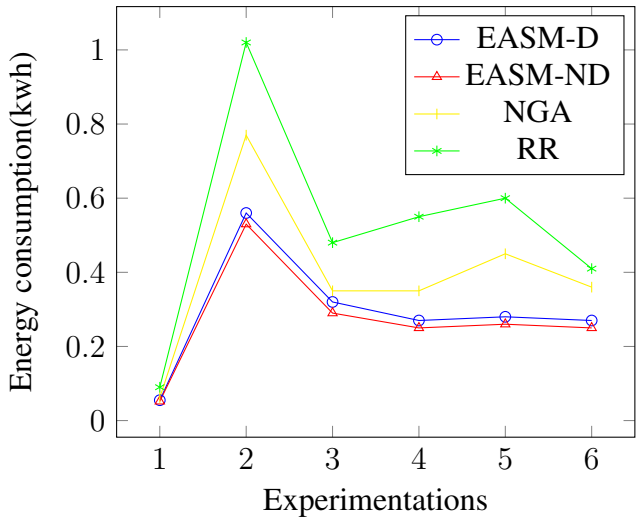

Fig. 4.6 The energy consumption (Kwh) of different experimentation.

SLA violations, due to using the task classification and thresholds.

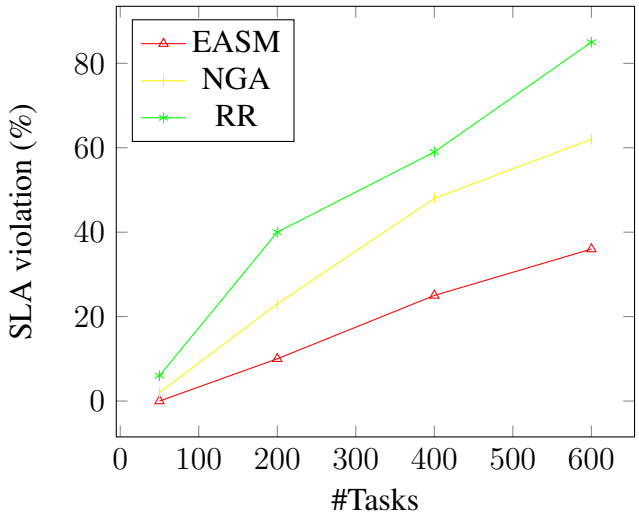

Fig. 4.7 Average SLA violation of different tasks number.

The performance of EASM can be explained by the classification and priority mechanisms and algorithm searches for an optimal solution more quickly. This algorithm considers not only processing time and energy consumption, but also resource utilization and the number of resources that can effectively complete the user's task.

This study has resulted in several implications for cloud computing settings. Firstly, cloud service providers may be able to significantly minimize their operational costs as a result of the increased energy efficiency as well as shorter execution times. This would increase the financial viability of their offerings. Second, this method guaran-

tees improved service quality by lowering Service Level Agreement (SLA) violation rates, that greatly raises customer satisfaction and trust in cloud services. Additionally, maintaining optimal performance in diverse settings and guaranteeing scalability and flexibility in the face of changing demands depend largely on the model's capacity to adjust to dynamic workload fluctuations in cloud data centers.

Although results are encouraging, this method has encountered several limitations. A significant constraint pertains to the results' generalization, as the experiments were carried out inside a simulated setting. In real-world, cloud settings must verify their validity so the findings are considered valuable. The scalability at large scale is another drawback. While being built for dynamic contexts, the model's effectiveness at extremely high scales still has to be carefully assessed to ensure it can manage complex and large-scale cloud infrastructures.

4.5 Conclusion

Due to the size of cloud data centers, there is a significant energy consumption and longer task execution times. As a result, users must regularly transmit data and the system uses virtual machine scaling to improve the efficiency of system resources. The main purpose of this work is to schedule effectively work into the available cloud environment resources, minimal energy consumption, execution time, and SLA violation. Task categorization, thresholds, and queuing are the foundation of the proposed work. Tasks are gathered into queues in the first phase based on how long they will take to complete. Then, GA is applied to find better solutions to improve scheduling. The suggested model had been validated and the comparative experimental findings were presented in terms of execution time, energy efficiency, and SLA violation. The results demonstrated that for all parameters, the suggested algorithm surpassed the other approaches.

CHAPTER 5

Energy-efficient resource management in cloud computing

5.1 Introduction

In cloud computing systems, managing resources efficiently is one of the main challenges. The data centers that provide cloud services are using more energy as the demand for these services increase (Kumar and Singh (2019)). The exponential growth in the needs for data processing, storage, and transmission results in enormous energy consumption and environmental impacts. Optimizing resource usage and reducing energy consumption in cloud computing environments requires the development of efficient techniques and algorithms (Al-Wesabi et al. (2022)). Although cloud computing has many advantages, including cost-effectiveness, scalability, and flexibility, it also has limitations with relation to energy use. To power its computers, cooling systems, and networking equipment, data centers need a lot of energy (R et al. (2022)). This energy consumption contributes to greenhouse gas emissions and degradation of the environment in addition to increase operating costs. Researchers and practitioners have concentrated on energy-efficient resource management in cloud computing systems in order to solve these issues. The objective is to migrate virtual machines as efficiently as possible while using the least amount of energy and maintaining the necessary level of service quality (Chhabra and Singh (2021)). A Threshold Q-learning VM Migration (TQVM), an artificial intelligence VM migration technique is introduced in this study. Two thresholds can be established by the suggested algorithm. Several experiments are conducted to evaluate the efficacy of the current approach, and the findings show that the TQVM algorithm may significantly lower energy usage.

5.2 Related Work

Several studies that use various types of strategies that include energy efficiency are discussed in this section. In (Wang et al. (2019)), a deep Q learning-based multi-agent deep reinforcement learning approach for cloud workflow scheduling cost and makespan optimization is introduced. The research analyzes multi-agent cooperation as a Markov game with a connected equilibrium to avoid motivating the makespan and cost agents to unilaterally deviate from the joint distribution. Li et al. (2021) introduced a weighted double deep Q network-based reinforcement learning method for cost and makespan optimal process scheduling in cloud environments. There are two stages to the scheduling process. In the first, a task is chosen from all of the available tasks. The changeable length of the input state is handled effectively by a pointer network. The second step involves choosing a virtual machine (VM) to carry out the chosen job. At every stage of the scheduling process, a different sub agent with a different incentive is utilized for every goal. Q learning was utilized by Qin et al. (2019) to minimize process executions' makespan and energy usage while staying within a budgetary restriction. Within a financial restriction, (Qin et al. (2019) seek to reduce workflows' makespan and energy usage. Workflow tasks are arranged according to a priority value that is determined by taking communication dependencies and task execution time into account. The Q learning method is then used to plan the sorted jobs. Each time step's VM usage is taken into account by the agent environment as the current state, and an action is equivalent to choosing a VM to carry out a job. To restrict the number of activities that may be taken at each time step, a budget restriction is placed on the action space. The agent receives a multi-vector reward, where one vector represents the ratio of the task's actual and quickest completion times, and the other represents the ratio of the task's actual and least energy usage. There is a weight selection issue since the reward has two vectors, This study picks the least scalarized Q value in a greedy manner after secularizing the Q values of state-action pairings using the Chebyshev scalarization function. If the relevant solution isn't dominated by any other solutions at the conclusion of each episode, it is added to the Pareto set; otherwise, all solu-

tions that are dominated by it are eliminated. Particle Swarm Optimization (PSO), a two-phase, energy-efficient load balancing method that makes use of virtual machine migration, was proposed by Masoudi et al. (2021). The author was able to reduce the cloud data center's initial energy usage by shutting off a large number of PMs. The author used the PSO to establish load balancing in the second phase. They also took into account the Dynamic Voltage Frequency Scaling (DVFS) technique in their proposed methodology. Their testing research indicates that after switching from the PSO algorithm to their suggested method, the cloud data center's energy usage reduced by about 10%. Li et al. (2015) proposed a Modified Particle Swarm Optimization (MPSO) based cloud data center energy-efficient virtual machine (VM) migration and consolidation approach. In their proposed work, the cloud data center's maximum number of virtual machines (VMs) was reduced to less PMs, and the VMs were relocated according to double threshold values. Their strategy reduced energy use and prevented the cloud data center's SLA violation. The energy efficient task scheduling algorithm (ETSA) is the foundation of the contribution (Panda and Jana (2018)), which aims to minimize makespan and energy usage. An online energy-efficient job scheduling system for heterogeneous cloud computing systems was created by the authors. They included the energy, cloud, application, and scheduling models into the suggested method. The foundation of this study is a combination of the TOPSIS approach for a more effective combination of the two techniques and the usage of DNN for regression. Another important key concept of the work given in this study is that the data center's energy usage may be reduced by choosing the physical machine first, followed by the right virtual machine. The four components of the VM consolidation approach are host overloading detection, host underloading detection, VM selection, and VMP (Beloglazov and Buyya (2011)). Furthermore, they provide the popular Power Aware Best-Fit Decreasing algorithm (PABFD). To manage cloud resources, the authors of (Nikzad et al. (2022)) proposed a method for multiobjective virtual machine allocation in a dynamic cloud setting. In their suggested solution, the author took into account eight criteria to lower energy usage and SLA breaches. They also used heuristics and metaheuristic algorithms to solve the same multiobjective issue. Comparing their suggested research

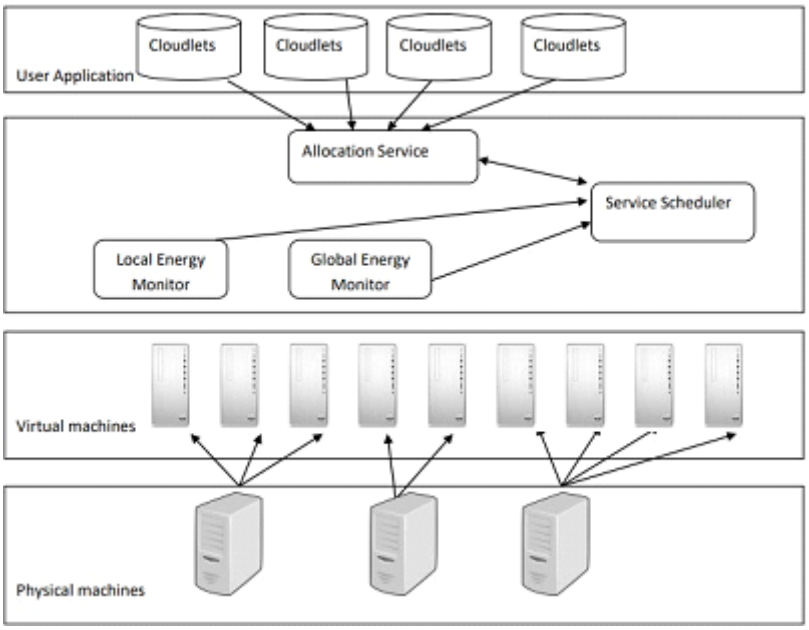

Fig. 5.1 Architecture of TQVM.

to current algorithms, they were able to obtain an energy reduction of up to 12.5%. They also decreased the amount of SLA violations and virtual machine migrations at the cloud data center.

5.3 Proposed model

In this section, we discuss the system architecture and energy model (Mehor Yamina and Omar (2025*b*)) followed by the details of the suggested model.

5.3.1 System architecture

The objective of this research is to decrease cloud data centers' energy use. Since the CPU is the primary resource in this activity, the procedure assigns VMs to PMs and schedules tasks to VMs. Figure 5.1 illustrates the different processes of this proposal which are defined as:

Data center: it is an IT entity composed of a set of physical machines.

$$CDC = \{PM_1, PM_2, ..., PM_N\} \tag{5.1}$$

Physical Machines:

$$PM = \{PM_1, PM_2, ..., PM_n\} \tag{5.2}$$

where n is the total number of PMs.

Virtual Machines: the virtual machine consists of creating more execution environments on a single physical machine. It provides each user with a service according to demand. – A set of VMs

$$VM = \{VM_1, VM_2, ..., VM_m\} \tag{5.3}$$

where m is the total number of VMs.

Cloudlets: They represent cloud application services. – A set of cloudlets

$$T = \{T_1, T_2, ..., T_k\} \tag{5.4}$$

where k is the total number of cloudlets.

Allocation Service: This component is responsible for dynamically allocating virtual machines (VMs) to available physical resources in the cloud. It also ensures the distribution of cloudlets (user tasks) to associated VMs, ensuring that execution is optimized based on host capacity and cloudlet requirements.

Service scheduler: Acts as a strategic layer responsible for temporally scheduling VM allocation. It decides when and on which physical host each VM (and therefore the cloudlets that will run on it) should be launched. This scheduling takes into account resource availability, expected performance, and overall energy consumption.

Local Energy Monitor: Each physical server is monitored locally using an energy monitor. This monitor measures the energy consumption related to VM and cloudlet execution in real time, providing crucial data for local allocation decisions.

Global Energy Monitor: This component provides a consolidated view of the energy consumption of the entire cloud infrastructure. It enables dynamic adjustment of VM

and cloudlet allocation across the data center, to optimize overall energy efficiency and meet sustainability constraints.

5.3.2 Energy Model

This research examines CPU use that determines how much electricity physical devices consume. About 70% of the power of a physically active machine is used when it is inactive. So, using Eq. 6.1, the power consumption (u) as CPU utilization is defined as:

$$P(u)_i = P_{max}(0,7 + 0,3u_i) \tag{5.5}$$

where u_i is the current CPU usage and P_{max} is the maximum power of a physical system operating at 100% CPU utilization. CPU usage is defined as a function u(t) of time since it varies over time. As a result, Eq. 5.6 establishes a physical machine's (PM_i) total energy consumption:

$$E_i = \int P(u(t))\,\mathrm{d}t \tag{5.6}$$

5.3.3 Allocation Model

Reducing energy consumption is the main objective of the proposed strategy. This research proposes a VM migration model in cloud computing that proposes two thresholds, local and global threshold. The proposed model uses a reinforcement learning approach to minimise the energy consumption and SLA violation.

5.3.3.1 Local threshold

We first calculated the energy consumption of PMs as shown below in Eq. 5.6 and also declared a local threshold to perform the migration. The physical machine is considered to be underutilized when the energy consumption is below this value so all VMs are migrated to other physical machines. Sort the PM list in the decreasing order of its VM energy consumption and compare the current PM energy consumed value to the local

threshold value of that PM. If the energy consumed of the PM is less than the lower threshold value then add all the VM of the PM to the migration list and remove all the VM from the PM and switch it off.

5.3.3.2 Global threshold

In this step we calculate the sum of the energy consumption of all PMs and also declare a global threshold to perform the migration. The summation of the energy consumed by all PMs is compared with the global threshold. If the summation is greater than global threshold value a migration must be established.

5.3.3.3 Q-learning algorithm

In this research, we provide a new approach based on the Q-learning algorithm for lowering energy consumption in cloud computing infrastructures (Wei et al. (2022)). Through dynamic resource adaptation, the primary objective is to optimize overall energy usage. The system model shows a grid of physical machines, with each cell representing a physical machine. Each physical machine can be: balanced, over, or underutilized. A physical machine utilization criteria has been established to determine whether physical machines are over-utilized and underutilized. VMs are automatically divided across the physical machines to get the best possible resource usage. The Q-learning algorithm learns and finds the best actions to achieve this balance while using the least amount of energy possible at each stage. The optimal VM reallocation technique is gradually discovered using the Q-learning algorithm.

Q-learning is one of the Reinforcement Learning (RL) algorithms(Chen et al. (2020)). RL is one of the machine learning methods that allows agents to learn in their environment and action by changing their state to receive rewards or penalties based on the feedback obtained from the environment. The hand purpose of RL is to learn the agent through trial and error between the agent and the environment. The agent is able to receive the environment situation through a state and choose an action that affects the environment to obtain the best reward and learn through past mistakes.

The Q-learning algorithm process is shown in Figure 5.2. Given that the set of states is in the environment, each state has a set of actions.

$$S = \{s_1, s_2, s_3, ..., s_n\} \tag{5.7}$$

$$A = \{a_1, a_2, a_3, ..., a_m\} \tag{5.8}$$

Agent selects action A in state S to pass to the next state s_{t+1}, S through the transition process and receives a reward r_{t+1} from the environment. To process the tasks, it is necessary to select the appropriate action to maximize the Q-value of each state, which is the primary objective of finding the optimal policy in cloud computing. The Q-value function depends on the selection of action in the state. Given the agent in a state and selecting an action, the Q-value function is expected to move to the best state and gain to maximize the total expected reward in the environment.

The Q-value derives from creating a Q-table that stores all possible states, Q-values, and appropriate actions. The Q-learning algorithm attempts to establish the optimal state from their experience. Q-value can be computed using Eq.5.9.

Eq. 5.9 drives the learning process.

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max Q(s',a') - Q(s,a) \right] \tag{5.9}$$

Where:

s is the current state of the physical machine (balanced, over-utilized, or underutilized),

a is the action taken (redistribute VMs to another physical machine or turn off a physical machine),

α is the learning rate,

r is the reward received (negative when energy consumption is high, positive when energy is optimized),

γ is the discount factor (importance of future rewards),

s' is the new state after the action,

$\max Q(s',a')$ represents the estimated value of the best possible action from the new

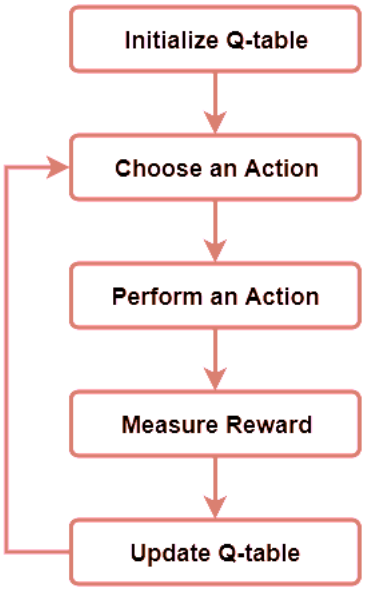

Fig. 5.2 Q-learning for Energy-aware VM allocation.(Kruekaew and Kimpan (2022))

state.

In this research, a Q-learning algorithm for VM migration is proposed as a way to reduce energy usage in a cloud setting. Begin by setting up a grid that presents the resources, with the percentage of resource utilization in each cell. The energy is computed using Eq. 5.6. Using the Q-learning algorithm, an agent is trained to learn about the grid and choose the best course of action to lower energy use.(As shown in Figure 5.2)

Servers that are below an inferior utilization threshold are turned off, and VMs have been allocated to identify over and underutilized physical machines in order to balance the load. This approach aims to increase system efficiency by reducing energy consumption and guaranteeing efficient resource allocation.

Case Study: Energy-aware Q-learning in a 3x3 Server Grid

We consider a cloud infrastructure composed of nine physical machines organized in a 3x3 grid. Each cell represents a physical machine capable of hosting multiple virtual machines (VMs). The objective is to reduce overall energy consumption by dynamically balancing the load using the Q-learning algorithm.

To illustrate the proposed optimization process, Table 5.1 presents an example of the initial state of servers in a cloud infrastructure. Each server has a certain CPU

utilization rate, which allows us to determine whether it is over-utilized, underutilized, or balanced.

After applying the Q-learning algorithm, the virtual machines (VMs) are automatically redistributed to maximize resource utilization while minimizing energy consumption. Table 5.2 shows the state of the servers after this optimization. We observe that the CPU loads have been balanced between the remaining servers, and that servers PM3 and PM6, which were initially underutilized, have been shut down to save energy.

It is important to note that, in this example, the total sum of CPU utilization rates remains constant before and after optimization, which respects the principle of workload conservation.

Assumptions

Each physical machine can be:

Over-utilized ($CPUutilization > 70\%$)

Underutilized ($CPUutilization < 30\%$)

Balanced (between 30% and 70%)

Underutilized physical machines can be shut down if their VMs are moved.

Energy consumption is calculated based on Eq. 5.6

How Q-learning Works

State: The current load distribution across the physical machines.

Action: Move one or more VMs from one physical machine to another.

Reward: Low total energy consumption after the action.

Optimal policy: Progressively learned by Q-learning to always choose the action that minimizes energy.

Table 5.1 Example of initial physical machine situation

PM	CPU Utilization (%)	State
PM1	80%	over-utilized
PM2	60%	balanced
PM3	20%	Underutilized
PM4	75%	over utilized
PM5	50%	balanced
PM6	25%	Underutilized
PM7	85%	over-utilized
PM8	55%	balanced
PM9	30%	balanced

Table 5.2 Corrected physical machine status after optimization by Q-learning

PM	CPU Utilization (%)	Status
PM1	70%	Balanced
PM2	65%	Balanced
PM3	0%	Powered off
PM4	70%	Balanced
PM5	70%	Balanced
PM6	0%	Powered off
PM7	65%	Balanced
PM8	70%	Balanced
PM9	70%	Balanced

5.4 Experimental evaluation

This section presents the experiments carried out to evaluate our proposal. The first part of the section describes the cloud infrastructure and the second part presents the scheduler configuration.

5.4.1 Cloud infrastructure

Several PMs were set up in a data center that was established for our simulation research. This simulation environment creates a variety of virtual machines. The key consideration when choosing simulation settings is whether the chosen features align with frequent data center architectures as determined by previous studies.

5.4.2 Scheduler configuration

To test the effectiveness of the proposed approach, many experiments are conducted on physical machine grids of different sizes, started with random utilization values. The Q-learning algorithm's parameters were set at a learning rate α of 0.1 and a discount factor γ of 0.99, with a total of 1000 learning episodes. Two thresholds can be established by the suggested algorithm: local and global thresholds. These thresholds, representing the percentage of CPU utilization of a physical machine. Energy consumption was measured using an equation. The results show a considerable decrease in energy use. Grid state analysis additionally showed significant improvements in physical machine load balancing, indicating that the recommended approach not only optimizes energy usage but also improves resource utilization in a cloud computing setting.

5.5 Results and Discussion

The suggested VM migration algorithm significantly increases energy efficiency, according to the results.

5.5.1 Baseline Results

The energy consumption, SLA violations, and number of VM migrations for NPA, DVFS (Zhou et al. (2017)), MAD, THR, (Beloglazov and Buyya (2011)), MBFD, PEBFD, PEFFD and MFPED, PABFD, BFTHR (Moges and Abebe (2019)), GA (Nahhas et al. (2021)), and TQVM are shown in the following figures.

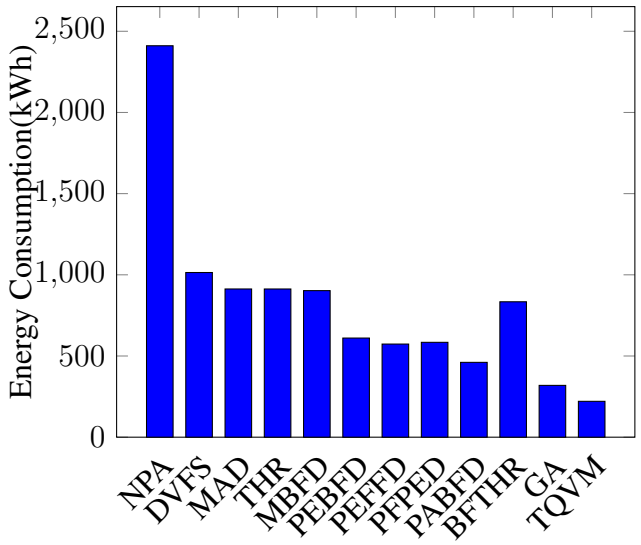

Fig. 5.3 Average energy consumption in the datacenter

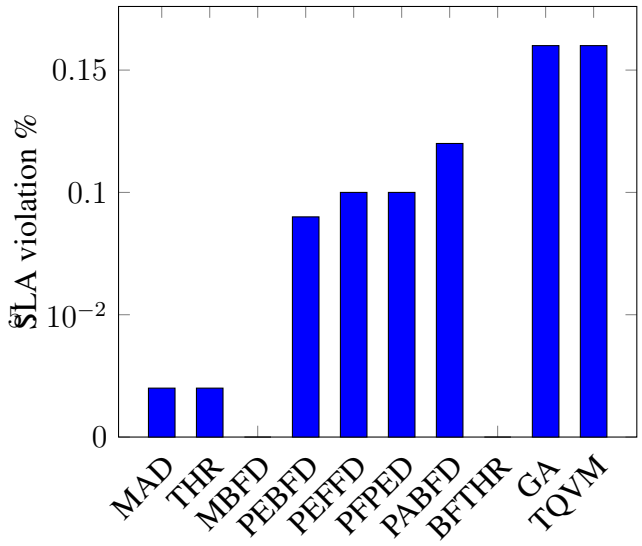

Fig. 5.4 Average SLA violations in the datacenter

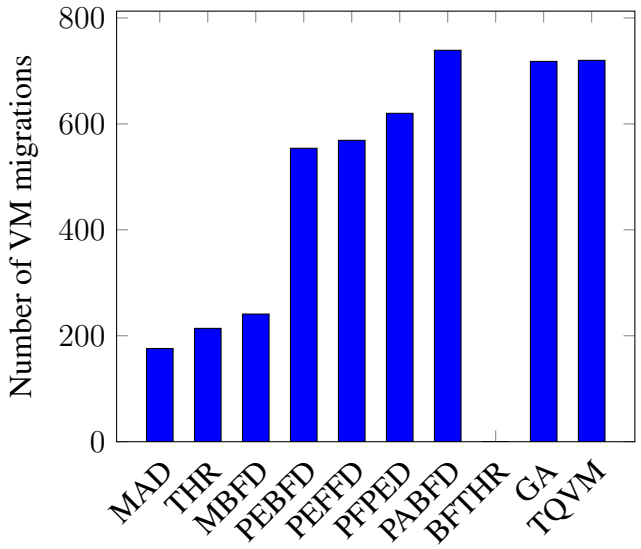

Fig. 5.5 Average number of migrations of virtual machines

Our research shows that the number of migrated virtual machines has an important impact on lowering energy use. The main objective is to compare the performance of the suggested algorithm with the heuristics described in (Beloglazov and Buyya (2011)), (Moges and Abebe (2019)), (Nahhas et al. (2021)), and (Zhou et al. (2017)). As shown in Figure 5.3, our method has demonstrated a significant reduction in energy usage. The NPA uses 2410.8 KWh. In order to lower its energy usage, DVFS uses 1014.21 KWh. The effectiveness of DVFS is superior to that of NPA. However, neither NPA nor DVFS entail VM migrations; instead, we employ the notation ”—-” to indicate the number of VM migrations and nonexistent SLA violation. The suggested strategy outperforms the other strategies, including PEBFD, PEFFD, and MFPED, that were presented in (Moges and Abebe (2019)).

However, as Figure 5.3 illustrates, these methods use a lot of energy. Among heuristic-based algorithms, the PABFD algorithm, which uses less energy consumption, has a SLA violation of 0.12%, while our approach displays a violation of 0.16%. The BFTHR did not perform any virtual machine migrations. As shown in Figure 5.3, the genetic algorithm started 15 fewer migrations than the PABFD method and 163, 148, and 97 more than the PEBFD, PEFFD, and MFPED algorithms, respectively.

There are no SLA violations seen by the Best-Fit based heuristic algorithms, such as MBFD and BFTHR as shown in Figure 5.4.

As shown in Figure 5.5, the number of VM migrations also decreased from 739 in PABFD to 720 in TQVM. When compared to Cloudsim's fundamental algorithms, the suggested method is effective.

The results highlight the trade-offs between energy efficiency, SLA compliance, and VM migration frequency among different scheduling algorithms in cloud computing. TQVM emerge as the most effective solution for reducing energy consumption, making them ideal for cloud environments where minimizing power usage is a top priority.

A balanced approach should be considered, integrating hybrid strategies that optimize energy consumption while minimizing SLA violations and controlling migration overhead.

5.5.2 Impact of Local, Global Thresholds and Q-Learning Parameters

Figure 5.6 illustrates the comparison of energy consumption as a function of the applied threshold, for the two optimization phases: local and global. We observe that the global phase systematically leads to slightly lower energy consumption than that obtained with the local phase, particularly in the intermediate threshold range (30% to 70%), considered the optimal equilibrium zone.

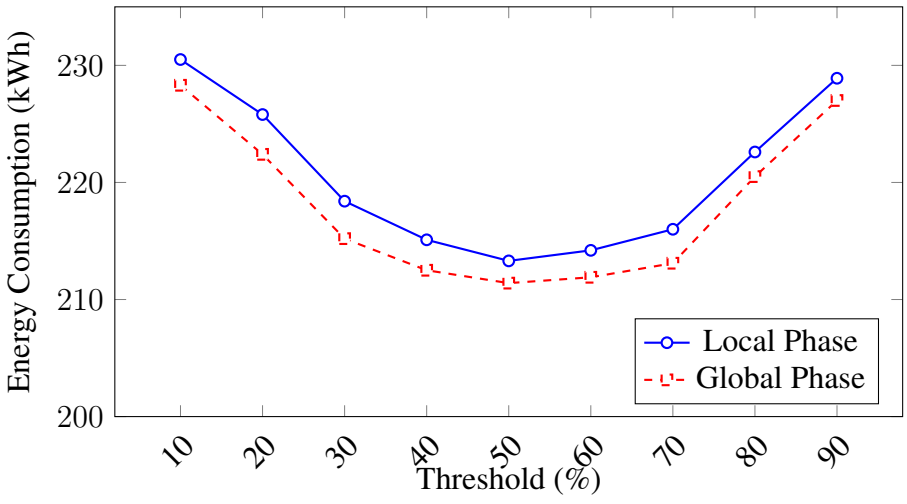

Fig. 5.6 Comparison of energy consumption under local and global thresholds

This improvement is explained by the consolidated view of the global state of the servers provided by the global phase, allowing for better task consolidation and more

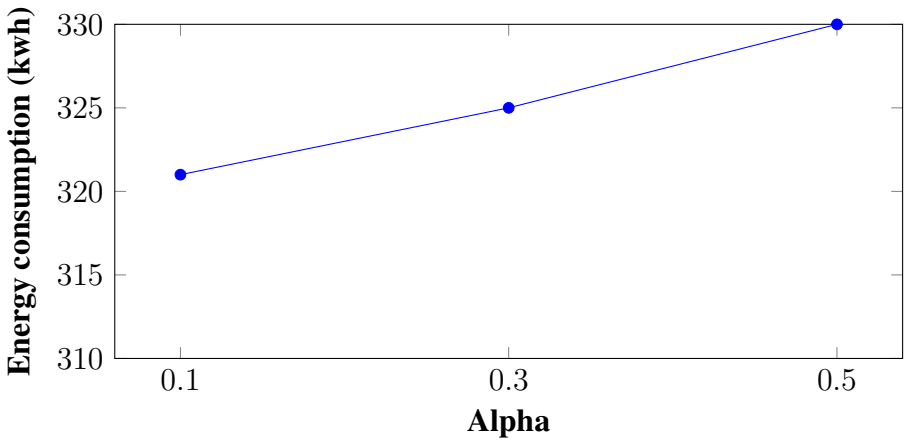

Fig. 5.7 Impact of Alpha on Energy Consumption

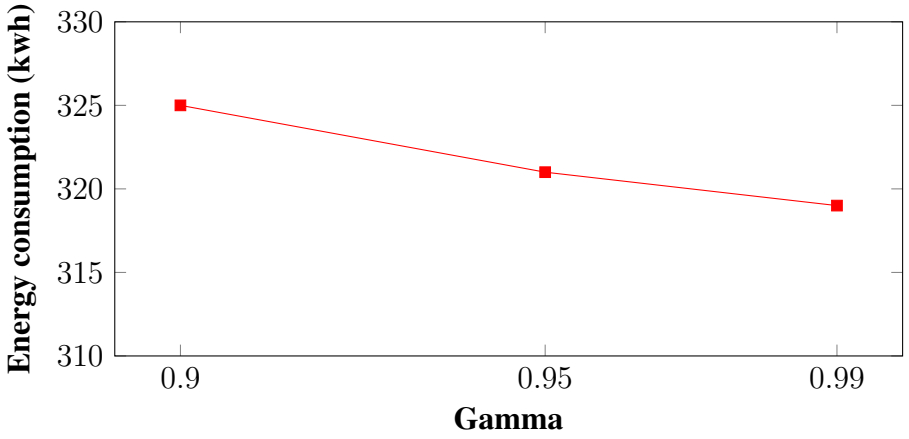

Fig. 5.8 Impact of Gamma on Energy Consumption

efficient shutdown of underutilized resources. Thus, the sequential combination of the two optimization phases achieves better overall energy performance.

Figure 5.7 and 5.8 show how Q-learning parameters, namely Alpha (learning rate) and Gamma (discount factor), affect energy usage, measured in kWh. Energy consumption tends to grow as Alpha increases, as Figure 5.7 illustrates. This might be because of more extensive investigation and hence higher resource demand. However, Figure 5.8 shows that energy usage decreases as the Gamma value increases, indicating that a more thorough evaluation of future benefits enables a more effective use of resources over the long term. These findings demonstrate how crucial it is to appropriately adjust these parameters in order to get the best possible balance between learning performance and energy efficiency in a cloud computing setting.

5.6 Conclusion

This research offers a novel artificial intelligence technique (TQVM) to solve the high energy consumption and resource usage problem in cloud computing. According to experimental data, TQVM minimizes energy consumption and resource usage. This suggests that TQVM may be applied to cloud computing work allocation, especially in complex scenarios.

CHAPTER 6

Energy-aware task scheduling and resource allocation in cloud computing

6.1 Introduction

Distributed computing is a rapidly developing area that includes cloud computing (Thekkepuryil et al. (2021)). A large amount of energy usage is attributed to cloud data center servers (Peng et al. (2022)). Virtualization technology increases resource usage by putting several virtual machines (VM) on a physical host (PM) and decreases the amount of hardware components in use, improving data center energy efficiency. The goals of the cloud provider and the users must be maximized by scheduling user tasks onto virtual machines (VMs) and carefully placing these VMs on physical hosts (PMs). Service provisioning in a cloud data center may be done at two levels: task scheduling is the first level, where each task of user is mapped into the appropriate virtual machine and the allocation of the virtual machines is the second level.

The objective is to optimize energy consumption and task scheduling using an modified genetic algorithm and resource allocation using double threshold Q-learning VM migration (Mehor Yamina and Omar (2025*a*)). Task Scheduling and VM Placement TSVMP differs from existing approaches by:

- The integration of genetic algorithms and Reinforcement learning algorithm for task scheduling and VM allocation.
- The use of thresholds to balance the load and turn off underutilized servers.
- An adaptive learning to different loads which implies minimization of energy

consumption.

6.2 Related work

The scheduling and resource allocation issues have been addressed by several researchers. In (Kruekaew and Kimpan (2022)), a Q-learning algorithm-based multi-objective task scheduling optimization based on the Artificial Bee Colony Algorithm (ABC) is suggested.

The Sine Cosine Algorithm (SCA) and the Ant Colony Optimization (ACO) algorithms are combined by the researchers in (Vijaya and Srinivasan (2024)) to compose a novel hybrid approach for efficient VM deployment.

Machine Learning methods have the potential to significantly improve energy efficiency in Cloud data centers (CDCs) by Panwar et al. (2024). The primary objectives are to optimize energy utilization and acquire resources by predicting CPU usage, identifying overloads, estimating under-loads, selecting, migrating, and relocating virtual machines.

6.3 The proposed model

6.3.1 System and energy model

Consider the Data Center (DC) consists of m physical machines (PM).

$P_i = \{P_1, P_2, ..., P_m\}$, where (i = 1,...,m).

Consider the physical machine consists of n virtual machines (VM)

$V_j = \{V_1, V_2, ...V_n\}$, where (j = 1,...,n).

and $T_k = \{T_1, T_2, ..., T_l\}$ denote a set of tasks where (k = 1,...,l).

In the present research, we focus on the relationship between server power usage and CPU use. Eq. 6.1: defines the power consumption P(u) as CPU usage.

$$P(u)_i = P_{max}(0,7 + 0,3u_i) \tag{6.1}$$

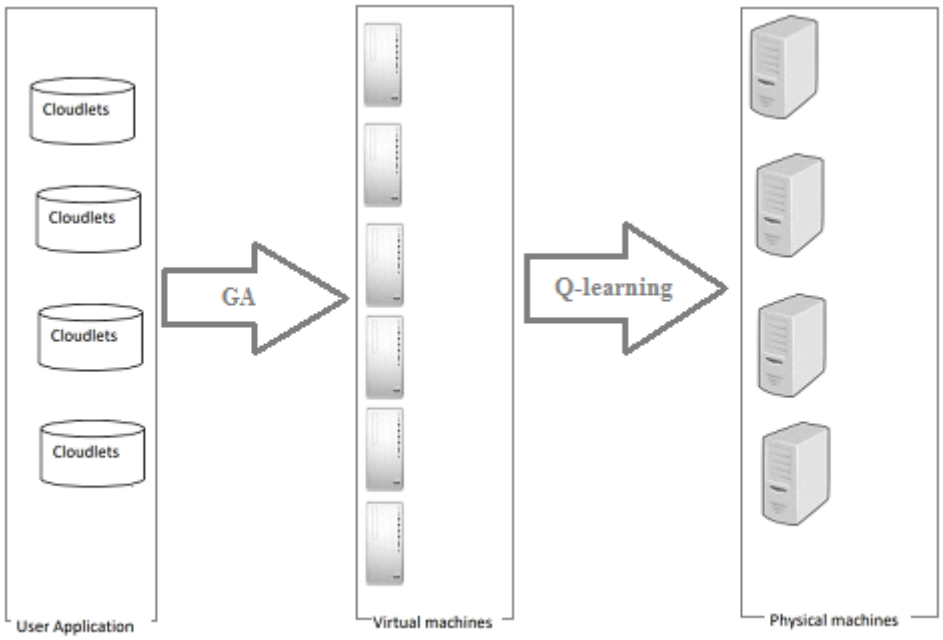

Fig. 6.1 Architecture of TSVMP

where P_{max} represents the maximum power of a server and u represents the current CPU utilization. Eq. 6.2 measures the total energy consumption of a physical machine:

$$E_i = \int P(u(t))\,\mathrm{d}t \tag{6.2}$$

6.3.2 Optimization model

The proposed optimization model aims to improve resource utilization and reduce energy consumption. It consists of two main parts: task scheduling and virtual machine allocation.(As shown in Figure 6.1)

6.3.2.1 Task scheduling

At this phase, a cloud computing scheduling model that suggests two stages has been presented. In the first stage, the tasks in cloud computing must be completed in the shortest amount of time possible using the resources that are available. Virtual machines (VMs) with high processing power are assigned to tasks with longer length to minimize the overall execution time (Malik et al. (2021)).

In the next phase, A modified Genetic Algorithm is used to optimize task scheduling:

Initial population: Tasks are assigned to virtual machines randomly.

Crossover: The best individuals are selected, and a single-point crossover is applied to generate new solutions.

Mutation: A random mutation is performed based on a predefined probability. The process repeats until the optimal solution is found.

6.3.2.2 VM migration

At this stage, a VM migration model in cloud computing is proposed, that proposes two thresholds, local and global threshold.The local threshold is defined as 30% and indicates underutilization of a PM; the VMs in that PM must be migrated to shut it down. The global threshold is defined as 70% and identifies overload, requiring the migration of certain VMs for performance and load balancing. These threshold values are defined to reduce energy consumption by previous studies.

Q-learning algorithm: The Q-learning algorithm learns and finds the best actions to achieve this balance while using the least amount of energy possible at each stage. The optimal VM reallocation technique is gradually discovered using the Q-learning algorithm. Eq. 6.3 drives the learning process:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max Q(s',a') - Q(s,a) \right] \tag{6.3}$$

The Q-learning algorithm optimizes energy consumption by learning to shut down underutilized servers and redistribute VMs to balance the load. The agent adjusts its actions based on a reward r and estimates the best future decision using $Q(s',a')$(As shown in Figure 6.2).

6.4 Experimental evaluation

The CloudSim simulator has been used to implement the suggested solutions. 560 physical machines were used in a single data center that was set up for the simulation tests.

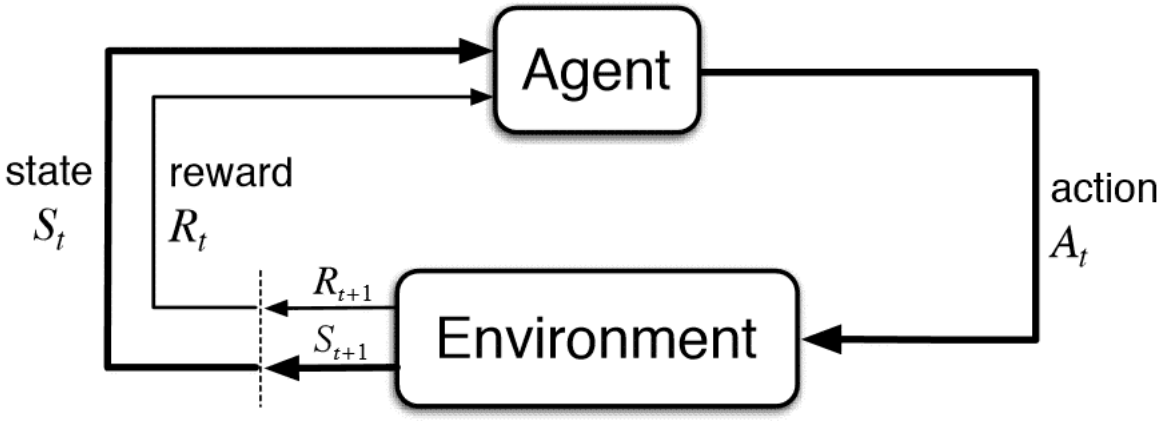

Fig. 6.2 Q-learning for Energy-aware VM allocation.

Table 6.1 Simulation Parameters

Algorithm	Parameters	Values
Modified Genetic Algorithm	Population size	10
	Elite count	2
	Tournament size	3
	Crossover rate	0.5
	Mutation rate	0.2
Q-learning Algorithm	α	0.1
	γ	0.9
	learning episodes	1000

The algorithms' resource utilization, energy usage, and SLA violations had been evaluated. In the simulation experiments, the solution approach is evaluated using heuristic algorithms. Such algorithms include: Modified Best-Fit Decreasing (MBFD), Power Efficient Best-Fit Decreasing (PEBFD),Power Efficient First-Fit Decreasing (PEFFD), Medium-Fit Power Efficient Decreasing (MFPED),Power Aware Best-Fit Decreasing (PABFD),Best-Fit Static Threshold (BFTHR) and Genetic Algorithm (GA) (Moges and Abebe (2019))(Nahhas et al. (2021)). Table 6.1 represents simulation parameters of the proposal: The evaluation of each method in optimizing cloud computing resources is the objective. Figure 6.3 represents the average energy consumption in a data center. TSVMP is efficient and outperforms the comparison heuristics, while MBFD consumes the most energy.

Figure 6.4 represents the average number of migrations of virtual machines. TSVMP reports a high number of migrations which optimizes resource usage. Turning off un-

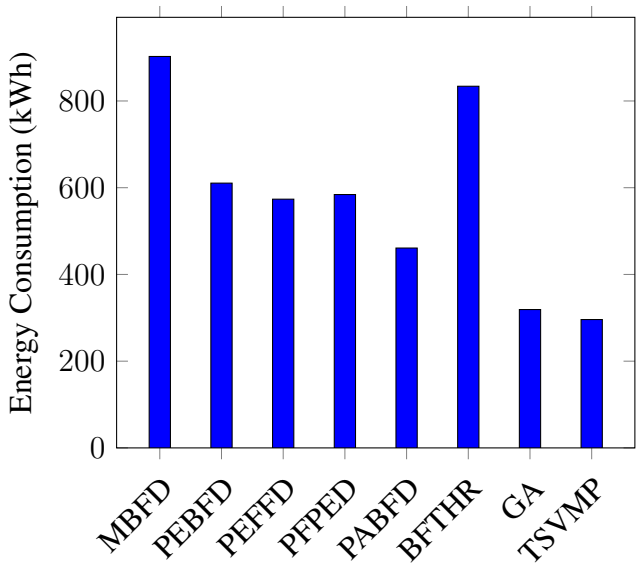

Fig. 6.3 Average energy consumption in the data center

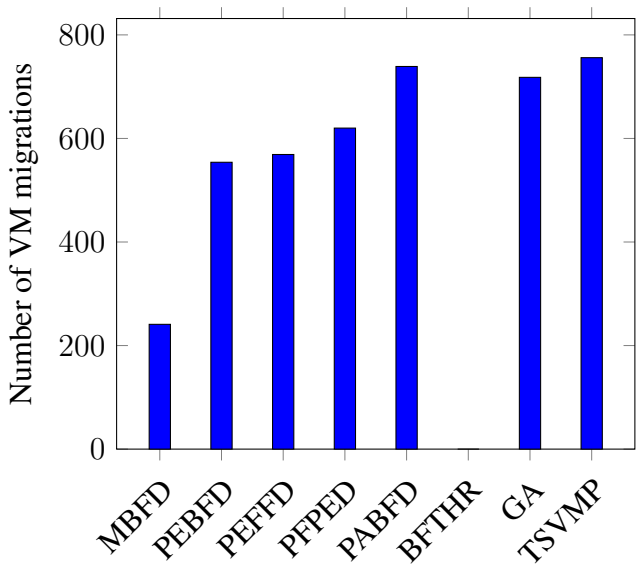

Fig. 6.4 Average number of migrations of virtual machines

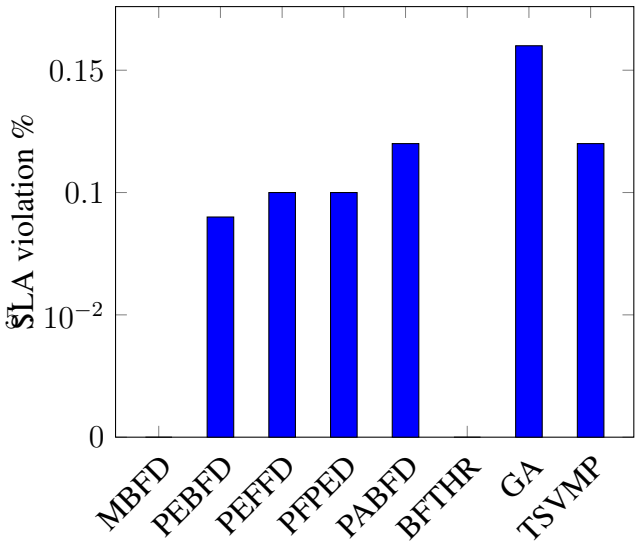

Fig. 6.5 Average SLA violations in the datacenter

derused servers and avoiding server overload increases the migrations number, but the energy usage of a server turned on unnecessarily is much greater than that of the migrations. We obtain an overall energy gain even if it implies a temporary cost to the migrations. BFTHR shows no migrations.

Figure 6.5 represents the average SLA violations in the data center. TSVMP represents a good balance between energy efficiency and quality of service. MBFD guarantees a perfect SLA but consumes a lot of energy. The highest SLA is for GA.

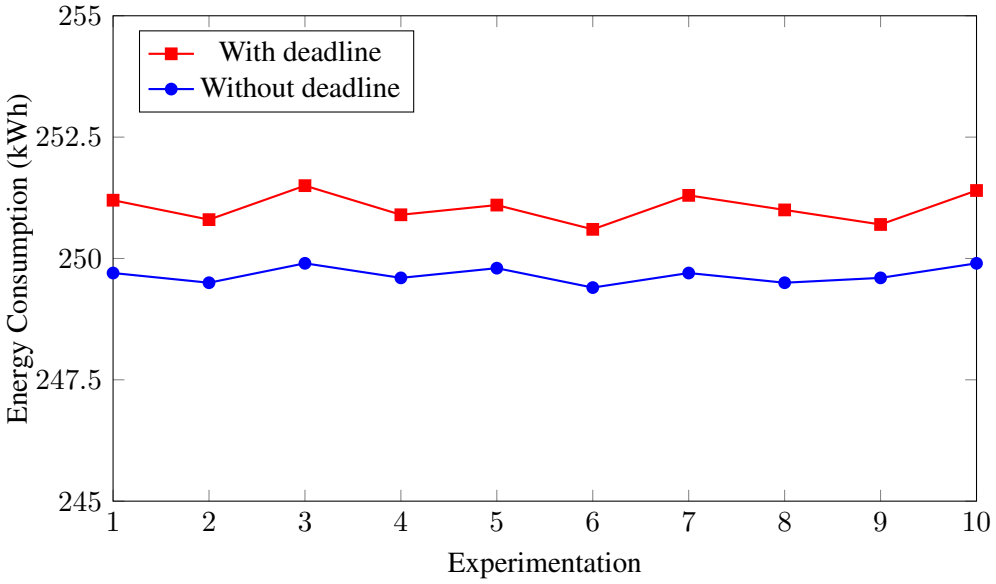

Fig. 6.6 Impact of Task Deadlines on Energy Consumption

The Figure 6.6 compares the energy consumption of two scheduling strategies over 10 experiments:

Every experiment uses a little more energy when there are deadlines (around 251 kWh).

Without deadlines, usage stays somewhat lower at about 249.6 kWh.

The minor but persistent difference indicates that there is a slight increase in energy use when deadlines are added.

6.5 Conclusion

This research effectively schedules tasks and manages resources in cloud computing settings. To schedule tasks to virtual machines and assign virtual machines to physical machines, TSVMP is applied. By using less energy and resources, the suggested task performs better than the current methods.

CHAPTER 7

Conclusions and Future Research Directions

This thesis concentrates on the development and design of models and algorithms for energy-efficient task scheduling and resource allocation. In this concluding chapter, we present our findings regarding the work presented in this thesis and suggest potential future directions for its expansion. The initial section provides a concise summary of the primary contributions and draws conclusions. Following this, we present prospective directions for future investigations in the second section.

7.1 Conclusions and Discussion

Energy efficiency is becoming increasingly significant for cloud data centers. The significant issue of power consumption is growing as a result of their wide availability and increasing scope. The primary goal of this work is to create and improve models and algorithms for the efficient allocation of resources, while taking into account various aspects of the issue. The resource provisioning plan, the dynamicity of the solution, the type of virtualization, and the Cloud service model are the four primary dimensions. The issue of resource allocation in the Cloud is extremely difficult to resolve while maximizing energy efficiency and following to the discussed dimensions. This thesis addresses the issue in its entirety, incorporating its numerous aspects and levels. Our objective is to offer a comprehensive and generic solution, in addition to a specific solution.

The concepts of cloud computing and virtualization, which serve as its facilitating technology, are introduced in chapter 2. We conduct additional research on the energy issue.

By examining the primary causes of energy waste, introducing energy measurement and modeling in Cloud environments, and presenting various power-saving techniques, efficiency in Cloud data centers can be improved.
Chapter 3 offers a thorough analysis of the present state of the art in the allocation of energy-efficient resources in cloud environments. The issue of energy-efficient resource allocation in Cloud data centers is further described upon, and an overview of the current state of energy-efficient Cloud resource allocation at various levels and dimensions is provided. We have made an effort to acquire a more thorough comprehension of the issue, situate the thesis in relation to existing research, and identify the primary challenges and issues through this survey. This thesis has made the following significant contributions:

1. In Chapter 4, we propose the use of an Energy-Aware Scheduling Model (EASM) for task scheduling in cloud computing. The objective of the proposed model is to reduce energy consumption, execution time, and SLA violation.

2. Chapter 5 introduces a threshold Q-learning VM migration (TQVM), a unique artificial intelligence VM migration technique.

3. In Chapter 6, we optimize task scheduling and virtual machine allocation by integrating genetic algorithms and deep learning. The implementation of thresholds to distribute the workload and terminate servers that are underutilized. An adaptive learning process that involves the reduction of energy consumption in response to changing demands.

7.2 Future Research Directions

This thesis does not address specific issues related to the energy-efficient task scheduling and resource allocation problem in Cloud environments; however, these limitations will be addressed in future research. The subsequent are potential prospective directions of this research:

- The optimization of supplementary metrics, such as throughput and latency

- Investigating hybrid strategies that incorporate additional optimization techniques
- Conducting experiments on real cloud infrastructure.

References

Adhikari, J. and Patil, S. (2013), 'Double threshold energy aware load balancing in cloud computing', *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)* p. 1–6.
URL: *https://doi.org/10.1109/icccnt.2013.6726664*

Al-Wesabi, F. N., Obayya, M., Hamza, M. A., Alzahrani, J. S., Gupta, D. and Kumar, S. (2022), 'Energy aware resource optimization using unified metaheuristic optimization algorithm allocation for cloud computing environment', *Sustainable Computing Informatics and Systems* **35**, 100686.
URL: *https://doi.org/10.1016/j.suscom.2022.100686*

Alasady, A. S., Awadh, W. A. and Hashim, M. S. (2023), 'Non-dominated sorting genetic optimization-based fog cloudlet computing for wireless metropolitan area networks', *Informatica (Slovenia)* **47**, 1–8.

Alresheedi, S. S., Lu, S., Elaziz, M. A. and Ewees, A. A. (2019), 'Improved multiobjective salp swarm optimization for virtual machine placement in cloud computing', *Human-centric Computing and Information Sciences* **9**(1).
URL: *https://doi.org/10.1186/s13673-019-0174-9*

Alsaidy, S. A., Abbood, A. D. and Sahib, M. A. (2020), 'Heuristic initialization of pso task scheduling algorithm in cloud computing', *Journal of King Saud University - Computer and Information Sciences* **34**(6), 2370–2382.
URL: *https://doi.org/10.1016/j.jksuci.2020.11.002*

Amazon Mechanical Turk (MTurk) (2025).
URL: *https://www.mturk.com/*

Amazon Web Services (Amazon EC2). (2025).
URL: *https://aws.amazon.com/ec2/*

Amazon Web Services (Amazon VPC) (2025).
URL: *https://aws.amazon.com/vpc/*

Armbrust, M., Fox, A. and Griffith, R. (2009), Above the clouds: A berkeley view of cloud computing, Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley.

Aron, R. and Abraham, A. (2022), 'Resource scheduling methods for cloud computing environment: The role of meta-heuristics and artificial intelligence', *Engineering Applications of Artificial Intelligence* **116**, 105345.
URL: *https://doi.org/10.1016/j.engappai.2022.105345*

Awasthi, T., Rattan, D., Singh, P. and Markandeshwar, R. M. (2022), "to develop an improve energy efficient algorithm for load balancing in cloud computing", Technical report.
URL: *https://www.researchgate.net/publication/368247037*

Badr, S., El Mahalawy, A., Attiya, G. and Nasr, A. A. (2022), 'Task consolidation based power consumption minimization in cloud computing environment', *Multimedia Tools and Applications* .

Basmadjian, R., Ali, N., Niedermeier, F., de Meer, H. and Giuliani, G. (2011), A methodology to predict the power consumption of servers in data centres, *in* 'Proceedings of the 2nd International Conference on Energy-Efficient Computing and Networking (e-Energy '11)', ACM, New York, NY, USA, pp. 1–10.

Belgacem, A., Mahmoudi, S. and Ferrag, M. A. (2023), 'A machine learning model for improving virtual machine migration in cloud computing', *The Journal of Supercomputing* **79**(9), 9486–9508.
URL: *https://doi.org/10.1007/s11227-022-05031-z*

Beloglazov, A. (2013), Energy-efficient Management of Virtual Machines in Data Centers for Cloud Computing, PhD thesis, The University of Melbourne.

Beloglazov, A. and Buyya, R. (2010), Energy efficient allocation of virtual machines in cloud data centers, *in* 'Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing', IEEE, pp. 577–578.

Beloglazov, A. and Buyya, R. (2011), 'Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers', *Concurrency and Computation Practice and Experience* **24**(13), 1397–1420.
URL: *https://doi.org/10.1002/cpe.1867*

Beloglazov, A., Buyya, R., Lee, Y. C. and Zomaya, A. (2011), A taxonomy and survey of energy-efficient data centers and cloud computing systems, *in* 'Advances in Computers', Vol. 82, Elsevier, pp. 47–111.

Chen, X., Zhu, F., Chen, Z., Min, G., Zheng, X. and Rong, C. (2020), 'Resource allocation for cloud-based software services using prediction-enabled feedback control with reinforcement learning', *IEEE Transactions on Cloud Computing* **10**(2), 1117–1129.
URL: *https://doi.org/10.1109/tcc.2020.2992537*

Cheng, M., Li, J. and Nazarian, S. (2018), 'Drl-cloud: Deep reinforcement learning-based resource provisioning and task scheduling for cloud service providers', *2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC)* p. 129–134.
URL: *https://doi.org/10.1109/aspdac.2018.8297294*

Chhabra, S. and Singh, A. K. (2021), 'Dynamic resource allocation method for load balance scheduling over cloud data center networks', *Journal of Web Engineering* .
URL: *https://doi.org/10.13052/jwe1540-9589.2083*

Choppara, P. and Mangalampalli, S. (2024), 'An efficient deep reinforcement learning based task scheduler in cloud-fog environment', *Cluster Computing* **28**(1).
URL: *https://doi.org/10.1007/s10586-024-04712-z*

Choudhary, R. and Perinpanayagam, S. (2022), 'Applications of Virtual Machine Using Multi-Objective Optimization Scheduling Algorithm for Improving CPU Utilization and Energy Efficiency in Cloud Computing'.

Choukairy, F. E. (2018), Optimisation de la consommation d'énergie dans un environnement Cloud, PhD thesis, University of Laval, Québec, Canada. thesis.

Containers, L. (2025), 'Linux containers official website',
urlhttp://www.linuxcontainers.org/.

Ding, D., Fan, X., Zhao, Y., Kang, K., Yin, Q. and Zeng, J. (2020), 'Q-learning based dynamic task scheduling for energy-efficient cloud computing', *Future Generation Computer Systems* **108**, 361–371.
URL: *https://doi.org/10.1016/j.future.2020.02.018*

Djouhra, D. (2016), Optimisation des Performances des Data Centers des Clouds sous Contrainte d'Optimisation d'Energie, PhD thesis, Université d'Oran.

Docker (2025), 'Docker official website',
urlhttps://www.docker.com/.

Duy, T. V. T., Sato, Y. and Inoguchi, Y. (2010), Performance evaluation of a green scheduling algorithm for energy savings in cloud computing, *in* '2010 IEEE International Symposium on Parallel Distributed Processing, Workshops and PhD Forum (IPDPSW)', IEEE, pp. 1–8.

Economou, D., Rivoire, S. and Kozyrakis, C. (2006), Full-system power analysis and modeling for server environments, *in* 'Workshop on Modeling, Benchmarking and Simulation (MOBS)'.

Environmental Protection Agency (EPA) (2007), Report to congress on server and data center energy efficiency, Technical report, Environmental Protection Agency.
URL: *www.energystar.gov/ia/partners/prod*$_d evelopment/downloads/EPA_Datacenter_Report_Congr$

Erden, H. S., Yildirim, M. T., Koz, M. and Khalifa, H. E. (2016), Energy assessment of crah bypass for enclosed aisle data centers, *in* 'Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm)', IEEE, pp. 433–439.

Eucalyptus Systems Inc. (2025).
URL: *https://www.eucalyptus.cloud/*

Feng, H., Deng, Y. and Li, J. (2021), 'A global-energy-aware virtual machine placement strategy for cloud data centers', *Journal of Systems Architecture* **116**(July 2020), 102048.
URL: *https://doi.org/10.1016/j.sysarc.2021.102048*

Final Fantasy XIV (2025).
URL: *https://na.finalfantasyxiv.com/*

Foster, I., Zhao, Y., Raicu, I. and Lu, S. (2008), Cloud computing and grid computing 360-degree compared, *in* 'Proceedings of Grid Computing Environments Workshop', pp. 1–10.

Fu, X., Sun, Y., Wang, H. and Li, H. (2021), 'Task scheduling of cloud computing based on hybrid particle swarm algorithm and genetic algorithm', *Cluster Computing* **26**(5), 2479–2488.
URL: *https://doi.org/10.1007/s10586-020-03221-z*

Garg, N., Singh, D. and Goraya, M. S. (2021), *Energy and resource efficient workflow scheduling in a virtualized cloud environment*, Vol. 24, Springer US.
URL: *https://doi.org/10.1007/s10586-020-03149-4*

Gharehpasha, S., Masdari, M. and Jafarian, A. (2020), 'Power efficient virtual machine placement in cloud data centers with a discrete and chaotic hybrid optimization algorithm', *Cluster Computing* **24**(2), 1293–1315.
URL: *https://doi.org/10.1007/s10586-020-03187-y*

Gharehpasha, S., Masdari, M. and Jafarian, A. (2021), 'Power efficient virtual machine placement in cloud data centers with a discrete and chaotic hybrid optimization algo-

rithm', *Cluster Computing* **24**(2), 1293–1315.
URL: *https://doi.org/10.1007/s10586-020-03187-y*

Ghribi, C. (2014), Energy efficient resource allocation in cloud computing environments, PhD thesis, Institut National des Télécommunications, Paris, France. doctoral thesis.
URL: *https://theses.hal.science/tel-01149701v1*

Gmail (2025).
URL: *https://mail.google.com/*

Google App Engine (2025).
URL: *https://cloud.google.com/appengine*

Google Docs (2025).
URL: *https://docs.google.com/*

Gourisaria, M. K., Khilar, P. M. and Patra, S. S. (2021), 'Esps: Energy saving power spectrum-aware scheduling to leverage differences in power ratings of physical hosts in datacenters', *Informatica (Slovenia)* **45**, 63–75.

Goyal, S., Bhushan, S., Kumar, Y., Rana, A. U. H. S., Bhutta, M. R., Ijaz, M. F. and Son, Y. (2021), 'An optimized framework for energy-resource allocation in a cloud environment based on the whale optimization algorithm', *Sensors* **21**(5), 1583.
URL: *https://doi.org/10.3390/s21051583*

Guérout, T. (2014), Ordonnancement sous contraintes de qualité de service dans les clouds, PhD thesis, Université (à préciser si disponible).

Hassan, H. A., Salem, S. A. and Saad, E. M. (2020), 'A smart energy and reliability aware scheduling algorithm for workflow execution in DVFS-enabled cloud environment', *Future Generation Computer Systems* **112**, 431–448.
URL: *https://doi.org/10.1016/j.future.2020.05.040*

Heath, T., Diniz, B., Carrera, E. V., Wagner Meira, J. and Bianchini, R. (2005), Energy conservation in heterogeneous server clusters, *in* 'Proceedings of the Tenth ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP '05)', ACM, New York, NY, USA, pp. 186–195.

Herbert, S. and Marculescu, D. (2007), Analysis of dynamic voltage/frequency scaling in chip-multiprocessors, *in* 'Proceedings of the 2007 International Symposium on Low Power Electronics and Design (ISLPED'07)', IEEE, pp. 38–43.

Heroku – Cloud Platform for Apps. (2025).
URL: *https://www.heroku.com/*

Hijji, M., Ahmad, B., Alam, G., Alwakeel, A., Alwakeel, M., Alharbi, L. A., Aljarf, A. and Khan, M. U. (2022), 'Cloud servers: resource optimization using different energy saving techniques', *Sensors* **22**(21), 8384.
URL: *https://doi.org/10.3390/s22218384*

Hoseiny, F., Azizi, S., Shojafar, M., Ahmadiazar, F. and Tafazolli, R. (2021), 'Pga: a priority-aware genetic algorithm for task scheduling in heterogeneous fog-cloud computing', *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* p. 1–6.
URL: *https://doi.org/10.1109/infocomwkshps51825.2021.9484436*

IBM Corporation. (2025).
URL: *https://www.ibm.com/cloud/*

Ibrahim, H., Aburukba, R. O. and El-Fakih, K. (2018), 'An integer linear programming model and adaptive genetic algorithm approach to minimize energy consumption of cloud computing data centers', *Computers Electrical Engineering* **67**, 551–565.
URL: *https://doi.org/10.1016/j.compeleceng.2018.02.028*

Imene, L., Sihem, S., Okba, K. and Mohamed, B. (2022), 'A third generation genetic algorithm nsgaiii for task scheduling in cloud computing', *Journal of King Saud Uni-*

versity - Computer and Information Sciences **34**(9), 7515–7529.
URL: *https://doi.org/10.1016/j.jksuci.2022.03.017*

Joyent Inc. (2025).
URL: *https://www.joyent.com/*

Kakkottakath Valappil Thekkepuryil, J., Suseelan, D. P. and Keerikkattil, P. M. (2021), 'An effective meta-heuristic based multi-objective hybrid optimization method for workflow scheduling in cloud computing environment', *Cluster Computing* **24**(3), 2367–2384.
URL: *https://doi.org/10.1007/s10586-021-03269-5*

Kansal, A., Zhao, F., Liu, J., Kothari, N. and Bhattacharya, A. A. (2010), Virtual machine power metering and provisioning, *in* 'Proceedings of the 1st ACM Symposium on Cloud Computing (SoCC '10)', ACM, New York, NY, USA, pp. 39–50.

Karim, F. K., Sivakumar, N. R., Alshetewi, S., Ibrahim, A. Z. and Venkatesan, G. (2024), 'An adaptive threshold-based modified artificial bee colony optimization technique for virtual machine placement in cloud datacenters', *IEEE Access* **12**, 94296–94309.
URL: *https://doi.org/10.1109/access.2024.3420173*

Katal, A., Dahiya, S. and Choudhury, T. (2022), 'Energy efficiency in cloud computing data centers: a survey on software technologies', *Cluster Computing* **26**(3), 1845–1875.
URL: *https://doi.org/10.1007/s10586-022-03713-0*

Khan, A. A., Zakarya, M., Rahman, I. U., Khan, R. and Buyya, R. (2020), 'Heporcloud: An energy and performance efficient resource orchestrator for hybrid heterogeneous cloud computing environments', *Journal of Network and Computer Applications* **173**, 102869.
URL: *https://doi.org/10.1016/j.jnca.2020.102869*

Khan, Z. A., Abdul Aziz, I., Osman, N. A. and Ullah, I. (2023), 'A review on task

scheduling techniques in cloud and fog computing: Taxonomy, tools, open issues, challenges, and future directions', *IEEE Access* **11**, 143417–143445.

Koot, M. and Wijnhoven, F. (2021), 'Usage impact on data center electricity needs: A system dynamic forecasting model', *Applied Energy* **291**, 116798.
URL: *https://doi.org/10.1016/j.apenergy.2021.116798*

Kruekaew, B. and Kimpan, W. (2022), 'Multi-objective task scheduling optimization for load balancing in cloud computing environment using hybrid artificial bee colony algorithm with reinforcement learning', *IEEE Access* **10**, 17803–17818.
URL: *https://doi.org/10.1109/access.2022.3149955*

Kumar, J. and Singh, A. K. (2019), 'Cloud datacenter workload estimation using error preventive time series forecasting models', *Cluster Computing* **23**(2), 1363–1379.
URL: *https://doi.org/10.1007/s10586-019-03003-2*

KVM (2025), 'Kvm official website',
urlhttp://www.linux-kvm.org/.

Li, G., Yan, J., Chen, L., Wu, J., Lin, Q. and Zhang, Y. (2019), 'Energy consumption optimization with a delay threshold in cloud-fog cooperation computing', *IEEE Access* **7**, 159688–159697.
URL: *https://doi.org/10.1109/access.2019.2950443*

Li, H., Huang, J., Wang, B. and Fan, Y. (2021), 'Weighted double deep q-network based reinforcement learning for bi-objective multi-workflow scheduling in the cloud', *Cluster Computing* **25**(2), 751–768.
URL: *https://doi.org/10.1007/s10586-021-03454-6*

Li, H., Zhu, G., Cui, C., Tang, H., Dou, Y. and He, C. (2015), 'Energy-efficient migration and consolidation algorithm of virtual machines in data centers for cloud computing', *Computing* **98**(3), 303–317.
URL: *https://doi.org/10.1007/s00607-015-0467-4*

Liang, B., Wu, D., Wu, P. and Su, Y. (2021), 'An energy-aware resource deployment algorithm for cloud data centers based on dynamic hybrid machine learning', *Knowledge-Based Systems* **222**, 107020.
URL: *https://doi.org/10.1016/j.knosys.2021.107020*

Lilhore, U. K., Simaiya, S., Prajapati, Y. N., Rai, A. K., Ghith, E. S., Tlija, M., Lamoudan, T. and Abdelhamid, A. A. (2025), 'A multi-objective approach to load balancing in cloud environments integrating aco and wwo techniques', *Scientific Reports* **15**(1).
URL: *https://doi.org/10.1038/s41598-025-96364-1*

Liu, N., Li, Z., Xu, Z., Xu, J., Lin, S., Qiu, Q., Tang, J. and Wang, Y. (2017), 'A hierarchical framework of cloud resource allocation and power management using deep reinforcement learning'.
URL: *http://arxiv.org/abs/1703.04221*

Liu, Y., Wei, X., Xiao, J., Liu, Z., Xu, Y. and Tian, Y. (2020), 'Energy consumption and emission mitigation prediction based on data center traffic and pue for global data centers', *Global Energy Interconnection* **3**(3), 272–282.
URL: *https://doi.org/10.1016/j.gloei.2020.07.008*

Mahilraj, J., Sivaram, P., Lokesh, N. and Sharma, B. (2023), 'An optimised energy efficient task scheduling algorithm based on deep learning technique for energy consumption', *2021 5th International Conference on Information Systems and Computer Networks (ISCON)* p. 1–7.
URL: *https://doi.org/10.1109/iscon57294.2023.10112019*

Malik, N., Sardaraz, M., Tahir, M., Shah, B., Ali, G. and Moreira, F. (2021), 'Energy-efficient load balancing algorithm for workflow scheduling in cloud data centers using queuing and thresholds', *Applied Sciences* **11**(13), 5849.
URL: *https://doi.org/10.3390/app11135849*

Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J. and Ghalsasi, A. (2010), 'Cloud computing — the business perspective', *Decision Support Systems* **51**(1), 176–189.
URL: *https://doi.org/10.1016/j.dss.2010.12.006*

Masoudi, J., Barzegar, B. and Motameni, H. (2021), 'Energy-aware virtual machine allocation in dvfs-enabled cloud data centers', *IEEE Access* **10**, 3617–3630.
URL: *https://doi.org/10.1109/access.2021.3136827*

Maurya, K. and Sinha, R. (2013), International journal of computer science and mobile computing energy conscious dynamic provisioning of virtual machines using adaptive migration thresholds in cloud data center, Technical report.
URL: *www.ijcsmc.com*

Mboula, J. E. N. (2021), Energy-efficient Workflow Scheduling with Budget and Deadline constraints in a Cloud Datacenter, PhD thesis, Faculty of Science, University of Ngaoundere, Cameroon. doctoral thesis.
URL: *https://theses.hal.science/tel-03590132v1*

Medara, R. and Singh, R. S. (2021), 'Energy Efficient and Reliability Aware Workflow Task Scheduling in Cloud Environment', *Wireless Personal Communications* **119**(2), 1301–1320.
URL: *https://doi.org/10.1007/s11277-021-08263-z*

Medara, R., Singh, R. S. and Amit (2021), 'Energy-aware workflow task scheduling in clouds with virtual machine consolidation using discrete water wave optimization', *Simulation Modelling Practice and Theory* **110**(December 2020), 102323.
URL: *https://doi.org/10.1016/j.simpat.2021.102323*

Medishetti, S. K., Kurupati, R., Donthi, R. K. and Karri, G. R. (2025), 'Energy and deadline aware scheduling in multi cloud environment using water wave optimization algorithm', *International Journal of Intelligent Systems and Applications* **17**(3), 48–64.
URL: *https://doi.org/10.5815/ijisa.2025.03.04*

Mehor Yamina, R. M. and Omar, S. (2025*a*), Energy-aware task scheduling and resource allocation in cloud computing, *in* 'The First International Conference on Artificial Intelligence, Smart Technologies and Communications (AISTC'2025)', Hassiba Benbouali University of Chlef, Chlef, Algeria. Oral Presentation, April 14–15.

Mehor Yamina, R. M. and Omar, S. (2025*b*), Energy-efficient resource management in cloud computing, *in* 'The First International Conference on Artificial Intelligence and Sustainable Development (ICAISD'25)', Ahmed Zabana University, Relizane, Algeria. Oral and Online Presentation, April 12–13.

Mell, P. and Grance, T. (2010), The nist definition of cloud computing (draft), Technical Report 7, National Institute of Standards and Technology.

Meta Platforms, (2025).
URL: *https://www.facebook.com/*

Microsoft (2025), 'Microsoft hyper-v official website',
urlhttp://www.microsoft.com/Hyper-V. Accessed September 20, 2014.

Microsoft Online (2025).
URL: *https://www.microsoft.com/microsoft-365*

Mirmohseni, S. M., Javadpour, A. and Tang, C. (2021), 'Lbpsgora: Create load balancing with particle swarm genetic optimization algorithm to improve resource allocation and energy consumption in clouds networks', *Mathematical Problems in Engineering* **2021**, 1–15.
URL: *https://doi.org/10.1155/2021/5575129*

Moges, F. F. and Abebe, S. L. (2019), 'Energy-aware vm placement algorithms for the openstack neat consolidation framework', *Journal of Cloud Computing Advances Systems and Applications* **8**(1).
URL: *https://doi.org/10.1186/s13677-019-0126-y*

Moore, G. E. (1998), 'Cramming more components onto integrated circuits', *Proceedings of the IEEE* **86**(1), 82–85.

Nahhas, A., Cheyyanda, J. T. and Turowski, K. (2021), 'An adaptive scheduling framework for the dynamic virtual machines placement to reduce energy consumption in cloud data centers', *Proceedings of the Annual Hawaii International Conference on System Sciences* **2020-January**(January), 878–887.

Naone, E. (2009), 'Conjuring clouds', *Technology Review* **112**(4), 54–56.

Natural Resources Defense Council (NRDC) (2014), Scaling up energy efficiency across the data center industry: Evaluating key drivers and barriers, Technical report.
URL: *http://www.nrdc.org/energy/files/data-center-efficiency-assessment-IP.pdf*

Nikzad, B., Barzegar, B. and Motameni, H. (2022), 'Sla-aware and energy-efficient virtual machine placement and consolidation in heterogeneous dvfs enabled cloud datacenter', *IEEE Access* **10**, 81787–81804.
URL: *https://doi.org/10.1109/access.2022.3196240*

(NRDC) (2025), 'Natural resources defense council'.
URL: *http://www.nrdc.org/energy*

OpenVZ (2025), 'Openvz official website',
urlhttp://www.openvz.org/.

Oracle (2025), 'Solaris containers official website',
urlhttp://www.oracle.com/technetwork/server-storage/solaris/containers-169727.html.

Oracle NetSuite. (2025).
URL: *https://www.netsuite.com/*

Ounifi, H.-A., Gherbi, A. and Kara, N. (2022), 'Deep machine learning-based power usage effectiveness prediction for sustainable cloud infrastructures', *Sustainable Energy Technologies and Assessments* **52**, 101967.
URL: *https://doi.org/10.1016/j.seta.2022.101967*

Outlook (2025).
URL: *https://outlook.live.com/*

Panda, S. K. and Jana, P. K. (2018), 'An energy-efficient task scheduling algorithm for heterogeneous cloud computing systems', *Cluster Computing* **22**(2), 509–527.
URL: *https://doi.org/10.1007/s10586-018-2858-8*

Panda, S. K. and Jana, P. K. (2019), 'An energy-efficient task scheduling algorithm for heterogeneous cloud computing systems', *Cluster Computing* **22**(2), 509–527.
URL: *https://doi.org/10.1007/s10586-018-2858-8*

Panwar, S. S., Rauthan, M., Barthwal, V., Mehra, N. and Semwal, A. (2024), 'Machine learning approaches for efficient energy utilization in cloud data centers', *Procedia Computer Science* **235**, 1782–1792.
URL: *https://doi.org/10.1016/j.procs.2024.04.169*

Parallels (2025), 'Virtuozzo containers official website',
urlhttp://sp.parallels.com/fr/products/pvc/.

Parthiban, S., Harshavardhan, A., Neelakandan, S., Prashanthi, V., Alolo, A.-R. A. A. and Velmurugan, S. (2022), 'Chaotic salp swarm optimization-based energy-aware vmp technique for cloud data centers', *Computational Intelligence and Neuroscience* **2022**, 1–9.
URL: *https://doi.org/10.1155/2022/4343476*

Patel, H. B. and Kansara, N. (2021), 'Cloud computing deployment models: A comparative study', *International Journal of Innovative Research in Computer Science Technology* **9**(2), 45–50.
URL: *https://doi.org/10.21276/ijircst.2021.9.2.8*

Peng, Z., Pirozmand, P., Motevalli, M. and Esmaeili, A. (2022), 'Genetic Algorithm-Based Task Scheduling in Cloud Computing Using MapReduce Framework', *Mathematical Problems in Engineering* **2022**.

Piraghaj, S. F., Dastjerdi, A. V., Calheiros, R. N. and Buyya, R. (2017), A survey and taxonomy of energy efficient resource management techniques in platform as a service cloud, *in* 'Handbook of Research on End-to-End Cloud Computing Architecture Design', IGI Global, pp. 410–454.

Pirozmand, P., Hosseinabadi, A. A. R., Farrokhzad, M., Sadeghilalimi, M., Mirkamali, S. and Slowik, A. (2021), 'Multi-objective hybrid genetic algorithm for

task scheduling problem in cloud computing', *Neural Computing and Applications* **33**(19), 13075–13088.
URL: *https://doi.org/10.1007/s00521-021-06002-w*

Plummer, D. M. S. D. C. and Cearley, D. W. (2008), Cloud computing confusion leads to opportunity, Technical report, Gartner Research.

Pradhan, A., Bisoy, S. K., Kautish, S., Jasser, M. B. and Mohamed, A. W. (2022), 'Intelligent decision-making of load balancing using deep reinforcement learning and parallel pso in cloud environment', *IEEE Access* **10**, 76939–76952.
URL: *https://doi.org/10.1109/access.2022.3192628*

Project, X. (2025), 'Xen official website',
urlhttp://www.xenproject.org/.

Proxmox (2025), 'Proxmox official website',
urlhttp://www.proxmox.com/.

Qin, Y., Wang, H., Yi, S., Li, X. and Zhai, L. (2019), 'An energy-aware scheduling algorithm for budget-constrained scientific workflows based on multi-objective reinforcement learning', *The Journal of Supercomputing* **76**(1), 455–480.
URL: *https://doi.org/10.1007/s11227-019-03033-y*

Qiu, N. L. Z. L. J. X. Z. X. S. L. Q. (2017), 'A hierarchical framework of cloud resource allocation and power management using deep reinforcement learning', *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)* **4**(1), 132–141.
URL: *10.1109/ICDCS.2017.123*

R, D., J, U. U., Sharma, T., Singh, P., R, K., Selvan, S. and Krah, D. (2022), 'Energy-efficient resource allocation and migration in private cloud data centre', *Wireless Communications and Mobile Computing* **2022**, 1–13.
URL: *https://doi.org/10.1155/2022/3174716*

Rackspace US, Inc. (2025).
URL: *https://www.rackspace.com/cloud*

Raghavendra, R., Ranganathan, P., Talwar, V., Wang, Z. and Zhu, X. (2008), 'No "power" struggles: Coordinated multi-level power management for the data center', *SIGARCH Comput. Archit. News* **36**(1), 48–59.

Rambabu Medara, R. S. S. (2023), 'Dynamic virtual machine consolidation in a cloud data center using modifed water wave optimization'.

Relaza, T. J. R. (2016), Sécurité et Disponibilité des Données Stockées dans les Nuages, PhD thesis, Université Paul Sabatier-Toulouse III.

Saadi, Y. and El, S. (2020), 'Energy-efficient strategy for virtual machine consolidation in cloud environment', *Soft Computing* **2**.
URL: *https://doi.org/10.1007/s00500-020-04839-2*

Saif, F. A., Latip, R., Hanapi, Z. M., Alrshah, M. A. and Kamarudin, S. (2023), 'Workload allocation toward energy consumption-delay trade-off in cloud-fog computing using multi-objective npso algorithm', *IEEE Access* **11**, 45393–45404.
URL: *https://doi.org/10.1109/access.2023.3266822*

Salesforce Platform (2025).
URL: *https://www.salesforce.com*

Sangaiah, A. K., Javadpour, A., Pinto, P., Rezaei, S. and Zhang, W. (2023), 'Enhanced resource allocation in distributed cloud using fuzzy meta-heuristics optimization', *Computer Communications* **209**, 14–25.
URL: *https://doi.org/10.1016/j.comcom.2023.06.018*

Saxena, D., Gupta, I., Kumar, J., Singh, A. K. and Wen, X. (2021), 'A secure and multiobjective virtual machine placement framework for cloud data center', *IEEE Systems Journal* **16**(2), 3163–3174.
URL: *https://doi.org/10.1109/jsyst.2021.3092521*

Semmoud, A., Hakem, M., Benmammar, B. and Charr, J. (2020), 'Load balancing in cloud computing environments based on adaptive starvation threshold', *Concurrency*

and Computation Practice and Experience **32**(11).
URL: *https://doi.org/10.1002/cpe.5652*

Shally, N., Sharma, S. K. and Kumar, S. (2020), 'A dynamic threshold based energy efficient method for cloud datacenters', *International Journal of Software Innovation* **8**(2), 54–67.
URL: *https://doi.org/10.4018/ijsi.2020040104*

Shishido, H. Y., Estrella, J. C., Toledo, C. F. M. and Arantes, M. S. (2018), 'Genetic-based algorithms applied to a workflow scheduling algorithm with security and deadline constraints in clouds', *Computers and Electrical Engineering* **69**, 378–394.
URL: *https://doi.org/10.1016/j.compeleceng.2017.12.004*

Singh, S. and Kumar, R. (2022), 'Energy efficient optimization with threshold based workflow scheduling and virtual machine consolidation in cloud environment', *Wireless Personal Communications* **128**(4), 2419–2440.
URL: *https://doi.org/10.1007/s11277-022-10049-w*

Soltesz, S., Pötzl, H., Fiuczynski, M. E., Bavier, A. and Peterson, L. (2007), Container-based operating system virtualization: A scalable, high-performance alternative to hypervisors, *in* 'Proceedings of EuroSys 2007', pp. 275–288.

Srikantaiah, S., Kansal, A. and Zhao, F. (2008), Energy aware consolidation for cloud computing, *in* 'Proceedings of the 2008 Conference on Power Aware Computing and Systems (HotPower'08)', USENIX Association, Berkeley, CA, USA, pp. 10–10.

Teng, F. (2011), Ressource allocation and schelduling models for cloud computing, PhD thesis, Ecole Centrale Paris, Paris, France. doctoral thesis.
URL: *https://theses.hal.science/tel-00659303v1*

Thekkepuryil, J. K. V., Suseelan, D. P. and Keerikkattil, P. M. (2021), 'An effective meta-heuristic based multi-objective hybrid optimization method for workflow scheduling in cloud computing environment', *Cluster Computing* **24**(3), 2367–2384.
URL: *https://doi.org/10.1007/s10586-021-03269-5*

Tong, Z., Chen, H., Deng, X., Li, K. and Li, K. (2019), 'A scheduling scheme in the cloud computing environment using deep q-learning', *Information Sciences* **512**, 1170–1191.
URL: *https://doi.org/10.1016/j.ins.2019.10.035*

Tong, Z., Ye, F., Liu, B., Cai, J. and Mei, J. (2021), 'Ddqn-ts: A novel bi-objective intelligent scheduling algorithm in the cloud environment', *Neurocomputing* **455**, 419–430.
URL: *https://doi.org/10.1016/j.neucom.2021.05.070*

Travostino, F., Daspit, P., Gommans, L., Jog, C., de Laat, C., Mambretti, J., Monga, I., van Oudenaarde, B., Raghunath, S. and Wang, P. Y. (2006), 'Seamless live migration of virtual machines over the man/wan', *Future Generation Computer Systems* **22**(8), 901–907.

Uma, J., Vivekanandan, P. and Shankar, S. (2022), 'Optimized intellectual resource scheduling using deep reinforcement q-learning in cloud computing', *Transactions on Emerging Telecommunications Technologies* **33**(5).
URL: *https://doi.org/10.1002/ett.4463*

Vijaya, C. and Srinivasan, P. (2024), 'Multi-objective meta-heuristic technique for energy efficient virtual machine placement in cloud computing data centers', *Informatica (Slovenia)* **48**, 1–18.

VirtualBox (2025), 'Virtualbox official website',
urlhttp://www.virtualbox.org/.

VMware (2025), 'Vmware official website',
urlhttp://www.vmware.com/.

Wang, Y., Liu, H., Zheng, W., Xia, Y., Li, Y., Chen, P., Guo, K. and Xie, H. (2019), 'Multi-objective workflow scheduling with deep-q-network-based multi-agent reinforcement learning', *IEEE Access* **7**, 39974–39982.
URL: *https://doi.org/10.1109/access.2019.2902846*

Wang, Z., Chen, S., Bai, L., Gao, J., Tao, J., Bond, R. R. and Mulvenna, M. D. (2023), 'Reinforcement learning based task scheduling for environmentally sustainable federated cloud computing', *Journal of Cloud Computing Advances Systems and Applications* **12**(1).
URL: *https://doi.org/10.1186/s13677-023-00553-0*

Wei, J., Miao, X., Xu, K., Liu, R. and Ge, Y. (2022), 'An energy-efficient scheduling based on q-learning for energy harvesting embedded system', *2021 8th International Conference on Dependable Systems and Their Applications (DSA)* p. 503–509.
URL: *https://doi.org/10.1109/dsa56465.2022.00073*

Windows Azure (Microsoft Azure) (2025).
URL: *https://azure.microsoft.com/*

World of Warcraft (2025).
URL: *https://worldofwarcraft.blizzard.com/*

Xavier, M. G., Neves, M. V., Rossi, F. D., Ferreto, T. C., Lange, T. and Rose, C. A. F. D. (2013), Performance evaluation of container-based virtualization for high performance computing environments, *in* 'Proceedings of the 2013 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing', IEEE Computer Society, Washington, DC, USA, pp. 233–240.

Xing, H., Zhu, J., Qu, R., Dai, P., Luo, S. and Iqbal, M. A. (2021), 'An aco for energy-efficient and traffic-aware virtual machine placement in cloud computing', *Swarm and Evolutionary Computation* **68**, 101012.
URL: *https://doi.org/10.1016/j.swevo.2021.101012*

Yahoo (2025).
URL: *https://mail.yahoo.com/*

Zhang, Q., Lin, M., Yang, L. T., Chen, Z. and Li, P. (2017), 'Energy-efficient scheduling for real-time systems based on deep q-learning model', *IEEE Transactions on Sustain-*

able Computing **4**(1), 132–141.

URL: *https://doi.org/10.1109/tsusc.2017.2743704*

Zhou, Z., Abawajy, J., Chowdhury, M., Hu, Z., Li, K., Cheng, H., Alelaiwi, A. A. and Li, F. (2017), 'Minimizing sla violation and power consumption in cloud data centers using adaptive energy-aware algorithms', *Future Generation Computer Systems* **86**, 836–850.

URL: *https://doi.org/10.1016/j.future.2017.07.048*

LIST OF PUBLICATIONS

International Journal

1. Mehor, Y., Rebbah, M., and Smail, O. (2024). Energy-Aware Scheduling of Tasks in Cloud Computing. Informatica (Slovenia), 48(16), 125–136.

International Conferences

1. Mehor, Y., Rebbah, M., and Smail, O. (2025). Energy-efficient resource management in cloud computing. The First International Conference on Artificial Intelligence and Sustainable Development (ICAISD'25)* (Oral and Online Presentation, April 12–13). Ahmed Zabana University of Relizane, Algeria.

2. Mehor, Y., Rebbah, M., and Smail, O. (2025). Energy-aware task scheduling and resource allocation in cloud computing. The First International Conference on Artificial Intelligence, Smart Technologies and Communications (AISTC'2025)* (Oral Presentation, April 14–15). Hassiba Benbouali University of Chlef, Algeria.