The Journal of Supercomputing (2019) 75:4750–4810
https://doi.org/10.1007/s11227-019-02764-2

Toward energy-efficient cloud computing: a survey of dynamic power management and heuristics-based optimization techniques

Nagma Khattar[1] · Jagpreet Sidhu[2] · Jaiteg Singh[1]

Published online: 31 January 2019

Abstract
Cloud computing is the most prominent computing paradigm in the present era of information technology. However, data centers needed for hosting cloud services demand huge amount of electrical energy and release harmful gases to the atmosphere. To ensure a sustainable future, there is a need to focus on energy efficiency in cloud computing. Early literature pertaining to energy consumption in cloud computing is primarily focused on individual sub-domains like scheduling techniques, optimization, and green computing metrics. Research literature on cloud resource optimization is found to be the most discussed but less structured. This paper intends to provide a complete picture of energy efficiency in cloud computing. It also classifies heuristics-based optimization methods and the dynamic power management techniques. The survey shows the research trends based on regions, journals, conferences, etc., in the domain of energy efficiency in cloud computing. The study concludes with research issues and future research directions.

Keywords Energy-aware scheduling · Heuristics · Optimization · Cloud computing · Bibliographical analysis · Green cloud

1 Introduction

Energy consumption and greenhouse gas (GHG) emission by Information and Technology (IT) industry is increasing due to technological advances. It poses a severe threat to the environment. The threat is drastically increasing due to the increased use of computing in all aspects of life. According to Times report (2014), in the next

✉ Jagpreet Sidhu
jagpreet.pu@gmail.com

[1] Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India

[2] Department of Computer Science and Information Technology, Jaypee University of Information Technology, Waknaghat, Himachal Pradesh, India

half a decade, online users are expected to increase by 60% which will result in an increase in data consumption. It was predicted that data consumption would triple in 2012–2017, amounting to approximately 121 exabytes [1]. An IBM report "Ten key marketing trends in 2017" states that 2.5 exabytes of data are generated and consumed every day amounting to approximately 900 exabytes a year [2]. Internet users increased by 50% in 2018 [3]. As a consequence, there is an exponential increase in power requirement for data centers. This leads to severe environmental issues [4]. The demand for cooling infrastructure to control heat dissipation in data centers will also increase. Cooling infrastructure again would cause overhead by consuming more power and releasing GHGs. Maximum emission is due to electricity, air conditioning (AC), steam and gas supply [5]. According to a report of International Energy Outlook, during 2010–2040, the consumption of energy in the world will increase by 56%. Major consumers will be IT organizations [6]. The fact that the world will face an energy crisis is certain. So, it is a necessity to monitor energy consumption for a sustainable future. According to the World Wide Fund for Nature (WWF) report "A lack of access to energy is one of the main causes of poverty." The world needs a drastic reduction in carbon dioxide (CO_2) emissions within the next few years to avoid catastrophic climate [7].

Cloud Computing (CC) offers a promising solution for energy efficiency as it provides a virtualized environment which leads to multi-tenancy. Tarandeep et al. [6] state that "Cloud installations have higher server utilization levels and infrastructure efficiencies. Due to improvements in utilization levels achieved without compromising the desired performance, the role of CC in achieving energy efficiency has gained researchers' attention." The continuous acceptance of CC will reduce energy consumption in the data center by 31% during 2010–2020 [8]. Efforts are made by various organizations (Facebook, Google, and Amazon) to build energy-efficient data centers, and the research community has already taken it as a challenge [9].

Energy efficiency in CC has its own research problems and challenges as defined in Sect. 8. This article focuses on scheduling as it serves as a generic solution by effectively mapping tasks to efficient resources. Scheduling enhances other energy-efficient optimization solutions as discussed in Sect. 2.2. Several researchers have focused on energy efficiency, and a lot of literature is available. Further, most of the literature is published in conferences due to the dynamic nature of this domain [10]. As per our knowledge, a survey article has not been framed which addresses energy-efficient scheduling through heuristic-based optimization techniques and provides metrics on green cloud computing (GCC). Integration of sub-domains (scheduling techniques, optimization, and green computing (GC) metrics) is required to provide a complete overview of energy efficiency. Beloglazov et al. [11] presented a pioneering survey on energy-aware data centers and CC systems. It provides a taxonomy of hardware techniques, but it has a limited focus on scheduling techniques. Yu et al. [12] presented scheduling algorithms for grid computing specifically. Wu et al. [13] presented workflow scheduling for CC. The article mainly focused on scheduling techniques in contrast to energy awareness. Kaur and Chana [6] presented a survey on techniques for achieving energy efficiency in CC, but it did not provide a classification of heuristics-based optimization algorithms. Another Pioneering survey by Jing et al. [14] focused on processor server storage and cooling infrastructures. So, there is a need for a precise and concise survey to build a background for further research. The WWF report concludes on increasing the research on energy awareness and production of renewable energy. It indicates that the current expenditure on research and development on energy efficiency is €65 billion a year globally, out of about € 900 billion total expenditure across the globe on research and development in all segments. This expenditure must be increased twice in the next decade [7].

This survey is framed to provide a clear view of energy efficiency in CC through scheduling-based optimization. This survey makes it clear that scheduling- or heuristics-based optimization algorithms are a key solution for achieving energy efficiency in CC as discussed in Sect. 2.2. The survey tries to deliver a concise knowledge base about GCC using heuristics-based optimization techniques. The article discusses various concepts of energy efficiency and classifies the literature according to dynamic power management and heuristics-based optimization techniques.

The survey consists of nine sections. It is framed in a sequential manner covering all aspects related to energy efficiency using a problem-solving approach. Other sections are framed in the following manner: Sect. 2 describes the background on GC, CC, and their integration. It explains various energy-aware optimizations and concludes that scheduling is a holistic solution. Section 3 defines research questions and the adopted methodology for the survey. Section 4 presents a detailed overview of scheduling, energy efficiency, and their interrelation following a W4 approach. Section 5 gives details about the solution (optimization). Classifications of dynamic power management and heuristics-based optimization methods are presented. Section 6 describes metrics to measure the effectiveness of the solution. Section 7 provides global analysis to discover recent research trends. Section 8 provides research issues and a strategy for future research. Finally, conclusive remarks are provided in Sect. 9.

2 Background and motivation

Energy consumption in the data center has got ample consideration recently, but still many issues have not been addressed yet [15]. These issues are described in Sect. 8. A sustainable future needs energy-efficient techniques and reduction in GHG emission by cloud data centers.

2.1 Cloud computing and green computing

The National Institute of Standards and Technology (NIST) defines CC as “a model that facilitates expedient and dynamic access to a large pool of computing resources which can be shared, dynamically allocated, and discharged without much managerial involvement or service provider interaction” [15]. Virtualization allows several Virtual Machines’ (VMs) generation on a single physical machine [16].

GC includes planning, developing, consuming, and organizing of computing services in an environmentally friendly approach to promote sustainability [17, 18]. CC along with GC can prove to be a boon by employing energy-efficient computing practices at various service models. Service models are Infrastructure as a Service (IaaS), Software as a Service (SaaS), and Platform as a Service (PaaS) [19].

Figure 1 illustrates the GCC framework. Thus, application software, platform, tools, and infrastructure must be energy-efficient to promote GCC practices. Details regarding GCC techniques will be discussed in Sect. 2.2.

Cloud service providers (CSPs) deliver services to users by ensuring the security of data, quality of service (QoS), load balancing and traffic management as per the Service Level Agreement (SLA). Scheduling constitutes a generic solution for managing all these requirements [10]. Challenges in CC include (1) energy efficiency (2) QoS (3) SLA compliance (4) load balancing (5) security (6) traffic management and (7) cost-effectiveness.

These problems are addressed by using energy-efficient equipment, cost-effective provisioning, routing protocols, and encryption algorithms, etc. But scheduling can constitute a generic solution. Energy efficiency: Resources should be scheduled in such a manner that it minimizes carbon footprint by balancing the trade-off between overutilization and underutilization of resources.

Scheduling can be a solution in maintaining QoS-based application performance in the cloud. SLA compliance: CSPs have to complete the task on time and in the specified budget without affecting the reliability. So, optimized resource management requires scheduling based on makespan, deadline, and budget constraints. Load balancing requires mapping of VM resources to physical resources which involves scheduling algorithm. Security: Tasks are mapped to private or secure resources through scheduling. Traffic management: Data routing and forecasting techniques are in the solution domain, which involves scheduling for accessing cloud services efficiently. Cost-effectiveness: In this case, VM resources are scheduled in such a way that the total cost of deployment is reduced. Table 1 summarizes the literature to make it evident that scheduling is a generic solution for most of the issues in CC.

CC has transformed the way computing services are delivered. Although it brought a revolution in our lives, it cannot be considered a disruptive technology alone as addressed by [34]. Resources in data centers consume 3% of global energy

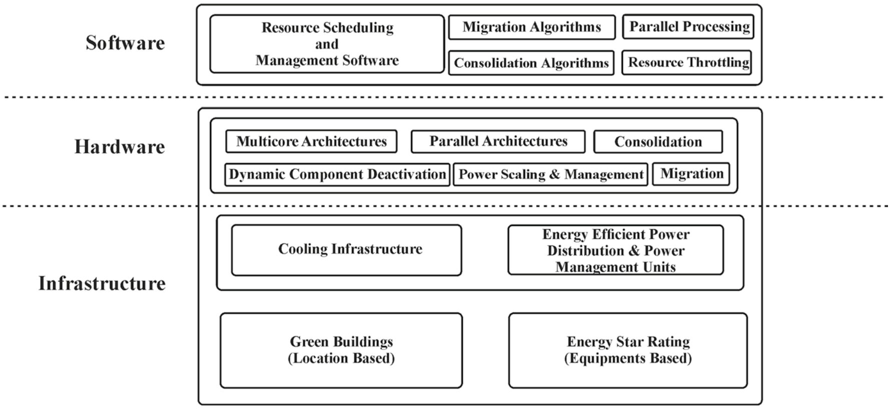
Software
Resource Scheduling and Management Software
Migration Algorithms
Parallel Processing
Consolidation Algorithms
Resource Throttling
Hardware
Multicore Architectures
Parallel Architectures
Consolidation
Dynamic Component Deactivation
Power Scaling & Management
Migration
Infrastructure
Cooling Infrastructure
Energy Efficient Power Distribution & Power Management Units
Green Buildings (Location Based)
Energy Star Rating (Equipments Based)

Fig. 1 Green cloud computing framework

Table 1 Summary of research issues in cloud computing

Problem	Focus	Objective	Techniques	Advantages	Disadvantages	Parameters
Energy efficiency [20, 21]	Job scheduling	Minimization of energy consumption, CO_2 emission	Resource provisioning and allocation algorithm, dynamic voltage–frequency scaling	Energy savings up to 30%	Did not consider idle servers' possible VM migration, hibernation without affecting response delay	Completion time, workflow size, resource utilization rate, QoS constraints
	Real-time tasks	Optimum CPU utilization and reducing energy consumption	Cooperative scheduling method	Energy savings up to 40%	No real cloud deployment	Turn-around time, CPU utilization
QoS [22, 23]	Resource scheduling	Maximizing reliability while scheduling	The online resource allocation algorithm	Component failures were considered and energy consumption, system unreliability reduction by 17% and 9%, respectively	Ignored degradation mode for components and overhead of exchanged messages	Reliability, energy consumption, total cost, QoS constraints
	Job scheduling	Guaranteeing QoS and optimization of power consumption and provider income	Two-level scheduling approach using acceptance and allocation policies	Evaluation of real workload traces, power consumption reduction without degrading service quality	Did not study algorithms for allocating resources for multiple classes of service	Total income, average income, degradation of power consumption, power consumption, performance ratio
SLA negotiation [24, 25]	Shared and heterogeneous cloud scheduling	Saving energy and also guaranteeing the QoS	Power-conserving scheduling method	Achieved 23% reduction in power consumption satisfying SLA, implementation on a real cloud platform	Did not consider disk usage and network communications, and future prediction of jobs was not there	Idle cloud energy consumption, the mean response time of a task
	Resource allocation and provisioning	Energy-efficient consolidation of VMs ensuring SLA	Modified best fit decreasing. Threshold-based dynamic consolidation	Reduced consumption of energy while maintaining SLA violation in the system to be not more than 1%	Focus on multi-core CPU architectures was missing, and memory, network interface was not considered	SLA violation, energy consumption, VM migration, CPU utilization

Table 1 (continued)

Problem	Focus	Objective	Techniques	Advantages	Disadvantages	Parameters
Load balancing [26–28]	Workload scheduling	Minimizing the cost of energy in data centers	Energy cost optimization-IDC algorithm, dynamic scheduling	Reduced energy cost for IDCs, guaranteed a service delay bound, and alleviated workload drop if the service delay bound was sufficiently large	Different geographical sites and different service delays were not considered	Energy cost, queuing delay bound, electricity price
	Thermal management by workload scheduling/balancing	Minimizing cooling infrastructure energy consumption and equipment damage risk because of overheating	Random and sequential load balancing algorithms combined the mechanical aspects and software-based scheduling	Jointly optimized the duty cycle of AC and speed of the fan to prevent imbalanced heating	Implementing proactive approach and analysis of improvement in the performance obtained was not done	Cooling energy, response time, temperature variation and equipment failure, risk
	Scheduling and balancing load for health care service	To improve network performance for telehealthcare service during load balancing	Scheduling	Performance improvement of the network provided best-streaming facilities	No research on whether the operational focus is really linking to the eye using a tracking device	Traffic received, queuing delay

Table 1 (continued)

Problem	Focus	Objective	Techniques	Advantages	Disadvantages	Parameters
Security [29–31]	Hybrid clouds	Securing and reducing inter-cloud communication	Tagged MapReduce	Presented modes for effective scheduling that also reduced usage of inter-cloud bandwidth	Small overheads were there	Inter-cloud communication, job elapsed time, computation outsourcing ratio
	Workflow partitioning	To partition applications across a set of clouds, while meeting security requirements	Multi-level security model	Reduced both security violations and execution costs	Not mentioned	Rules, transfers, cost
	Workflow scheduling	To preserve privacy in the scheduling of workflows considering deadlines and cost	Multi-terminal cut for privacy in hybrid clouds	Evaluated using real workflows, reduced the cost of executing workflows satisfying both the privacy and deadline constraints	Different limitations were assumed while scheduling	DAG's size, deadline, and private instance limitation, cost
Traffic management [32, 33]	Data center networks	To minimize the overall energy for data center traffic in time dimension deadline	Energy-efficient flow scheduling and routing algorithm	Reduce the total energy consumed overall energy and also decreased flow completion time on an average	Did not ensure that the network constrained flows. No real-world implementation	Energy, link utilization, switch utilization, deadline
	Circuit-switched tree (CST)	To power optimally configure and schedule well-nested communications on the CST	Power-aware dynamic reconfiguration (PADR)	The algorithm was power optimal, required only local information at processing elements (PEs), yet correctly established paths between communicating PEs	Did not develop computational algorithms for reconfigurable models and architectures	Only analysis has been made to check optimality

Table 1 (continued)

Problem	Focus	Objective	Techniques	Advantages	Disadvantages	Parameters
Cost-effectiveness [34, 35]	Scientific workflow	Minimize total cost	Scheduling and optimization algorithms: Particle swarm optimization (PSO), Earliest Finish Time (EFT)	Better results than another state-of-the art algorithms	Communication overhead and complex situations were not considered	Normalized total cost, makespan, total execution time
	Heterogeneous independent jobs	To offer a combined energy-aware scheduling and cost-effective framework in the presence of SLA without compromising QoS	Energy-aware task scheduling algorithm (EAMM) for heterogeneous clusters	48% reduction in energy consumption compared to the min–min algorithm	The model may not be efficient to apply to dependency preserving parallel applications	Normalized energy consumption, cost savings, energy savings, makespan extension, no. of processing elements, resource availability
Federation [36]	Federated clouds	To use facilities offered by cloud federations to schedule distributed calculations	Fair sharing algorithms	Efficient completion of short tasks is achieved using brokering algorithms	Not suitable for public clouds	Makespan, no. of cores, VM image, elapsed time

which is likely to get triple in the next decade. They are also responsible for 2% of GHG emission which will be 16% in the near future. It has been predicted in the year 2047 that there would be an enormous temperature on earth which leads to unsuitable living conditions [37]. So, CC may not be disruptive. But CC services availed in an energy-efficient manner can be disruptive. Major IT organizations have initiated efforts toward GCC. Apple as a leader has committed using renewable energy to power its iCloud. The organization deployed geothermal and solar energy to power its data centers. Google, Yahoo, and Facebook have started using renewable energy in their data centers as reported by Greenpeace [38].

2.2 Energy-efficient approaches for green cloud computing

There are three types of energy-efficient optimization techniques for cloud computing: (1) Infrastructure-based optimization which deals with infrastructural changes like making green buildings using energy-efficient equipment, air-conditioned racks, perforated tiles, floor raising, and other cooling equipment [39] for thermal management of data centers. Energy-efficient power distribution and energy star rating equipment [40] are infrastructure-based solutions, but are expensive to implement [41] and provide a limited reduction in energy consumption. Figure 2 represents infrastructure-based optimization in CC. (2) Hardware-based optimization: It includes employing multi-core architectures, voltage, frequency scaling, parallel architectures, energy-efficient hardware components, dynamic component deactivation, and consolidation. Angel et al. [42] were able to obtain significant energy consumption reduction using approximation algorithms on unrelated parallel machines. Dynamic voltage–frequency scaling (DVFS) practices are used on computing components for assisting the dynamic amendment of their performance uniformly to power consumption. Major techniques used are DVFS and DVFS with slack reclamation. DVFS and slack reclamation are described in detail in Sect. 5.1.

Figure 3 shows the architecture of a system using DVFS. A significant part of the research community is working on power management and scaling methods

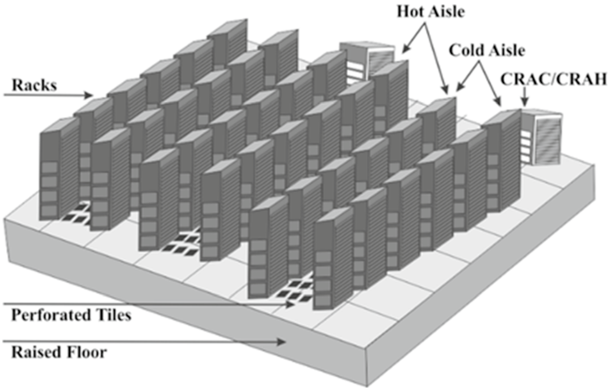

Fig. 2 Infrastructure-based optimization techniques

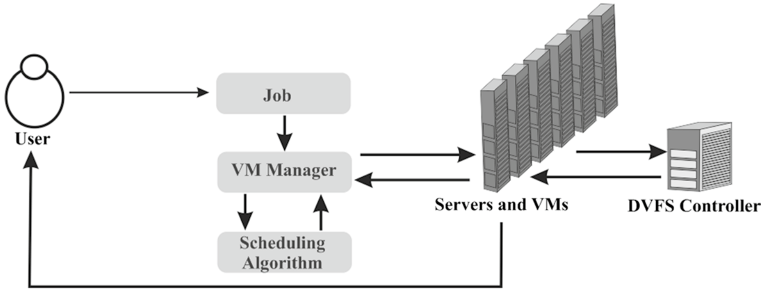

Fig. 3 Architecture of a system using DVFS

(Sect. 5.1). Liu [4] presents a renewable energy-based uninterrupted power supply scheme with an inbuilt central controller for power management. DVFS was used to tune the power demand of the server to get the efficient power point. Integer linear programming was also used. Results showed that the proposed framework improved backup energy capacity by 28%. Energy-efficient hardware components include efficient network and storage devices. A framework that implemented control strategies for the network and used network devices for local control mechanism was designed resulting in power savings [43]. Consolidation means to combine or integrate into one. Resources can be consolidated onto a lesser number of machines by halting underutilized machines to manage energy consumption. Consolidation can be performed at various levels: (a) VM consolidation, (b) server consolidation, and (c) task consolidation. Figure 4 represents the VM consolidation technique.

Consolidation results in optimization of hardware or infrastructure, but it involves the use of migration algorithms that come under software-based optimization. Further, hardware-based solutions are not sufficient unless resources are properly

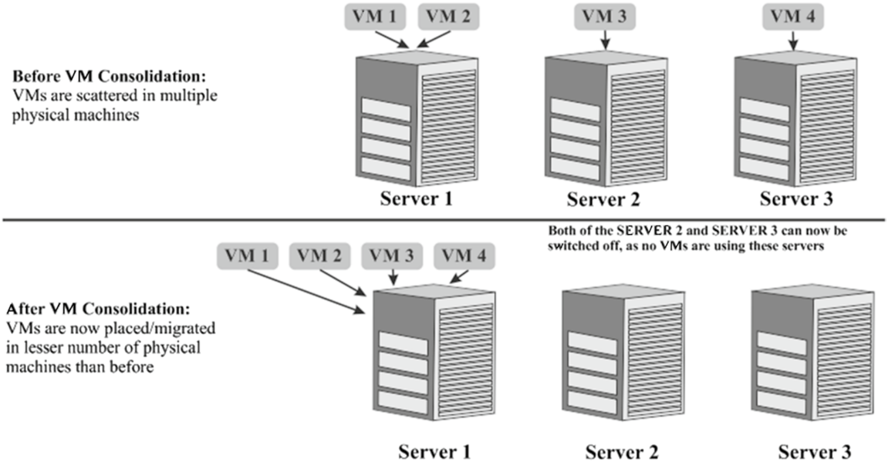
Before VM Consolidation:
VMs are scattered in multiple physical machines
VM 1
VM 2
VM 3
VM 4
Server 1
Server 2
Server 3
Both of the SERVER 2 and SERVER 3 can now be switched off, as no VMs are using these servers
After VM Consolidation:
VMs are now placed/migrated in lesser number of physical machines than before
VM 1
VM 2
VM 3
VM 4
Server 1
Server 2
Server 3

Fig. 4 VM consolidation technique

utilized. Utilization of resources depends upon scheduling. So, scheduling helps to optimize the infrastructure and hardware usage too. As per the literature, energy efficiency was not much achieved even after infrastructure or hardware changes due to poor design of software and programs. So, software-based optimizations are necessary. (3) Software-based optimizations include resource scheduling, allocation techniques, migration algorithms, throttling, and use of parallel programs. Scheduling is a holistic approach used everywhere. Being a non-polynomial (NP) Hard problem, scheduling can be done by various optimization techniques (as discussed in Sect. 5.2). The literature shows that scheduling algorithms have been widely used for minimizing energy consumption. Diaz et al. [44] utilized heuristics as a base of a scheduling algorithm to map the task on the heterogeneous system while reducing energy consumption. A model that used bi-level multi-objective programming for the locality and energy-aware multiple jobs scheduling was proposed, and numerical experiments indicated the effectiveness of algorithm [45] for reducing consumption of energy.

3 Research methodology

Table 2 illustrates the objectives and limitations of existing surveys. Surveys of highly reputed journals (SCI) with significant citations, most relevant subtopics are taken into consideration.

A total of 879 articles were excluded by reviewers on the basis of relevance, 671 were excluded on the basis of research contribution, and 256 were excluded on the basis of abstracts. Finally, 103 articles were left that clearly focused on the objectives of the survey. 245 references of publications published after the year 2010 were identified, and 167 were eliminated due to redundancy. Some not relevant articles included data networks (39), microprocessors (29), grid computing (10), and other irrelevant topics. The remaining 78 were inspected by reviewers, and 46 were included according to inclusion criteria (Fig. 5). The total articles left after this procedure were 156. The inclusion criterion was related to energy-aware scheduling in CC through optimization, measuring performance through GC metrics and GCC. Classification criteria (dynamic power management and heuristics- or scheduling-based optimization techniques) are finalized, and the literature is classified as discussed in Sect. 5. Table 3 illustrates the sets of keywords used in the title of surveyed articles, and these articles are classified in Sect. 5. Figure 6 shows the percentage of articles included in the survey according to keywords occurring in the title. Table 4 lists research questions and motivation.

4 Scheduling: a problem

Scheduling, in general, is a process of mapping of tasks to resources or target machines based on a criterion. Inefficient scheduling may result in performance degradation. According to the literature, scheduling is classified into three types: static, dynamic, and hybrid scheduling.

Table 2 Focus and limitations of existing surveys

Journal	References	Objective	Limitation
ACM CS	[6]	Hardware and software techniques of energy efficiency in CC	Detailed description about basic concepts, heuristic-based optimization techniques, and scheduling as a generic solution was not given
ACM CS	[46]	Focused on algorithms and criteria for scheduling workflows	Did not concentrate on energy efficiency
ACM CS	[47]	Resource management techniques and autonomic CC	Did not focus on energy efficiency
ACM CS	[48]	Hardware techniques	Focused on distributed systems and a detailed description of the classification of scheduling-based optimization techniques was missing
JSC	[49]	Concentrated on performance metrics	Did not focus on scheduling
JSC	[14]	Presented overall techniques for energy efficiency at different levels	Did not consider scheduling to be a generic solution
JSC	[13]	Considered scheduling techniques	Did not consider energy efficiency
FGCS	[50]	Presented challenges for workflow scheduling related to cost	Criterion considered was cost and not energy consumption
FGCS	[51]	Focused on the workflow scheduling problem	Presented a general picture related to scheduling and not specifically energy efficiency

ACM CS ACM computing surveys, *JSC* journal of supercomputing, *FGCS* future generation computer systems

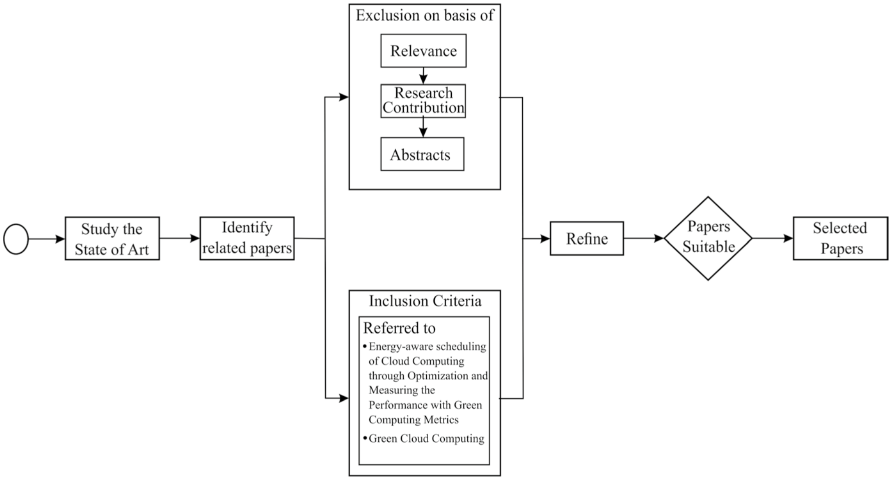

Fig. 5 Exclusion and inclusion criterion

Table 3 Sets of keywords in title of surveyed articles

S. no.	Name of set	Keywords
1	S_1	Energy
2	S_2	Energy, scheduling
3	S_3	Genetic/evolutionary approach
4	S_4	Genetic/evolutionary approach, energy
5	S_5	Genetic/evolutionary approach, energy, scheduling
6	S_6	Green cloud
7	S_7	Green cloud, scheduling
8	S_8	Optimization
9	S_9	Optimization, energy
10	S_10	Optimization, scheduling
11	S_11	Optimization, energy, scheduling
12	S_12	Heuristic
13	S_13	Heuristic, energy
14	S_14	Heuristic, energy, scheduling
15	S_15	Resource management
16	S_16	Miscellaneous
17	S_17	Scheduling

In *static scheduling*, the task execution environment and its characteristics are known in advance. Mapping of tasks to resources is determined before execution. At compilation time, information about cost and execution time is known. Jing Mei et al. developed an energy-aware scheduling algorithm by minimizing duplication and assuming task execution time, data size, and task dependencies are known before execution [52].

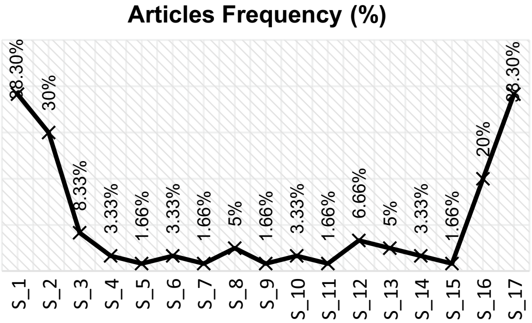

Fig. 6 Frequency of articles surveyed according to keywords in title

Table 4 Research questions and motivation

Research questions	Motivation
1. What are the basic concepts related to energy efficiency, scheduling, and heuristics-based optimization in CC and where should one start from in order to pursue research in this particular area?	There is a necessity for the article which provides an insight to energy-aware scheduling, all related terms, and state of the art
2. Why is energy efficiency important?	One needs to understand the role of energy in CC for optimum utilization of resources and environmental sustainability
3. Why focus on the consumption of energy in CC?	It is essential to know the harms CC is causing
4. What is the need for integration of energy efficiency and scheduling?	One should know the generic solution used in all energy efficiency techniques
5. How optimization can be a solution?	Decision problem employs the use of optimization algorithms
6. Which methods/techniques are used in energy-aware scheduling?	One should get to know the algorithms used in energy-aware scheduling
7. Which criteria are used for scheduling and what is their role?	There is a need to understand the importance of parameters
8. What are the metrics for GC?	There is a need to know the standards which can measure the effectiveness of data centers
9. What are the current trends of research in this area?	One should know the current state of research in this field and their limitations
10. What are gaps in previous studies?	It is essential to have knowledge about what is yet to be done
11. What are future directions for research/strategies?	There is a need to know the areas that need attention and future research plan

In *dynamic scheduling*, tasks are mapped to resources at runtime. Execution time and cost are available at runtime. A dynamic scheduling algorithm based on earliest deadline first (EDF) and power scaling method was used in hard real-time systems to reduce energy consumption [53]. Another dynamic scheduling algorithm used a multi-objective function to decrease the consumption of energy. The algorithm used resource allocation methods based on heuristics and was employed on parallel task-based applications. The results indicated significant energy savings for different scenarios. In directed acyclic graphs (DAG) types of embarrassingly parallel, matrix multiplication, and scatter–gather, the average energy savings were -22.44%, -33.17%, and -31.50%, respectively [54].

Hybrid scheduling is a combination of both dynamic and static scheduling strategies. Cost and execution time can be predicted at compilation time, but tasks can be assigned to resources only at runtime. Thus, it is statically planned, but dynamically scheduled. An energy-efficient algorithm which helped in achieving energy efficiency in smartphones used both dynamic and static scheduling practices. Results indicated significant improvements in energy savings [55].

Other than these types, the term “workflow scheduling” is commonly found in the literature. Workflow is a paradigm that represents various applications which are computationally complex. It is automation of procedure to process data by following certain rules. It represents various applications such as big data processing, scientific applications, web applications, data analysis [13, 19]. Many workflow applications are migrated to CC because of its ability to signify an extensive range of activities. Types of workflows are (1) abstract, (2) concrete, (3) business, and (4) scientific workflows. Abstract workflows provide tasks in an abstract form without describing specific resources and providing flexibility to users. It gives only service semantic information. Concrete workflows describe tasks for specific resources and give both service semantic and execution information [56]. Business workflows focus on control flows, and data are processed by machines. Scientific workflows are more abstract focusing on data flows based on data dependencies and processed by humans or machines [19]. Workflow is denoted by DAG where tasks are represented by vertices and edges depict dependencies. Through efficient workflow scheduling, optimal utilization of resources can be achieved.

Workflow scheduling can be separated into—(1) scheduling process, (2) scheduling tasks, and (3) scheduling resources [51]. Process scheduling focuses on scheduling criteria and techniques for schedule generation. Task scheduling constitutes mapping of tasks to target machines. Liu et al. presented Voltage Island Largest Capacity First (VILCF) algorithm to schedule periodic tasks on a multicore processor. The algorithm was based on DVFS and outperformed the existing algorithms for the multi-core voltage island [57]. Precedence constrained scheduling of parallel tasks on many-core processors was carried out by Keqin Li. Comparison with optimal algorithms proved its effectiveness [58]. Global task scheduling approaches used for prediction were discussed for mapping of tasks to their desired targets [21]. In resource scheduling, the execution model (public, private, and hybrid) and provisioning model are considered. Thread scheduling is necessary for the tiled multi-core environment to compensate for thread sensitivity toward shared resources [59]. Resource management was done through scheduling in the MapReduce model of cloud service (Cura) for providing cost-efficient services [60].

As per the literature, significant work is done on resource provisioning and task scheduling because these are the main steps to execute a workflow in CC [13].

4.1 W4 approach

W4 (What, Where, When, Why) approach is used to define all aspects related to energy-aware scheduling.

4.1.1 What is scheduling?: an NP-hard/complete problem

A problem is a decision problem if its solution is either "yes" or "no." Scheduling tasks on target systems is a decision problem [57]. Decision problems are of following types:

P class—A problem is said to be in P class if it is solvable in polynomial time means $O\ (n^m)$ in the worst case, where m is a constant. NP class—A problem is said to be NP if for instances where answers are yes can be tested in polynomial time. In this category of the problem, it is easy to check the correctness of the solution. NP-Hard class—There are some problems that cannot be solved directly; they are reduced to other problems. In addition, if the time taken for reduction is polynomial times, then they are reducible. If in NP class, all problems can be reduced in polynomial time to the other problem then they are NP-Hard. Also, for an NP-Hard problem, it is not compulsory to be in NP. In other words, if an algorithm takes huge time and is not feasible, then it is NP-Hard. NP-Complete class—Those problems which are NP-Hard and reside in NP Class too are NP-Complete [61].

There are different perspectives of researchers regarding scheduling. It is NP-Hard as mentioned in [12, 62]. However, scheduling is classified as an NP-Complete problem in [63, 64]. Therefore, this creates confusion. According to our view, the problem is NP-Hard if one can design a solution algorithm, but its execution takes huge time and is not feasible, e.g., if one maps a very large number of tasks say 5,000,000 tasks on computer systems in a certain short deadline, an algorithm can be designed, but it would not be feasible to schedule, i.e., NP-Hard. If one has to schedule 10 tasks, then this is possible, so it becomes NP-Complete.

4.1.2 Where energy-aware scheduling?

Energy-aware scheduling can be deployed with multi-core architectures, virtualization techniques, bio-inspired techniques, power-aware techniques, thermal-aware techniques. In all of these methods, scheduling algorithms are used as discussed in Sect. 2.2. Therefore, energy-aware scheduling becomes a holistic solution for achieving energy efficiency in CC.

4.1.3 When energy-aware scheduling?

Scheduling is based on criteria, which focus on one or more parameters [13]. These parameters can be objective or subjective. Objective parameters are measured directly, and output is numerically specified, e.g., time, cost, energy consumption, etc. Subjective parameters cannot be measured directly, e.g., fault tolerance can be measured through reliability. Energy-aware scheduling is done based on energy or power consumption criteria. Various parameters used for achieving energy efficiency are discussed in Sect. 5. Scheduling is classified as follows:

4.1.3.1 Best-effort scheduling (MIN–MAX) In this scheduling, only one objective is focused upon without considering other objectives such as QoS factors, e.g., considering time as the only constraint without focusing on cost/energy or only energy-aware scheduling without focusing on QoS (minimizing SLA violations). The current state of the art in achieving energy efficiency in CC has limited focus on QoS [24, 65, 66].

4.1.3.2 QoS-constrained workflow scheduling This type of scheduling is more used in actual applications. There is always a trade-off; if one tries to minimize certain factor, other automatically increases. Therefore, to handle the trade-off, it is done. The aim is to optimize one parameter while applying the constraint to another parameter. The goal is to generate a schedule in accordance with the preferable parameter meeting specified QoS constraints. Verma et al. presented a heuristic for scheduling workflow tasks having the budget and deadline constraints. The valuable trade-off was found between execution time and cost under these constraints. The simulation was performed with synthetic workflow applications to test the efficiency of presented heuristic. Results confirmed that offered heuristic decreased the cost keeping makespan as low as possible [67]. In another work on spectrum sensing, energy consumption was minimized by developing energy-efficient methods for body sensor networks while keeping satisfactory sensing quality [68]. In deadline-constrained workflow scheduling, one tries to minimize the cost while fulfilling timing constraints. Netjinda et al. focused on optimization of the cost of IaaS cloud services while executing scientific workflow within particular deadline constraints. Swarm optimization techniques were used. The results showed improvements in comparison with other algorithms by decreasing the total cost [34]. In budget-constrained workflow scheduling, execution of the workflow is completed while maintaining budget constraints. Kumar et al. proposed a scheme for SLA negotiation for budget, energy, and time. Authors made a strategy for making a cost-effective schedule without sacrificing performance. The simulation was performed for the evaluation of the proposed scheme which indicated it is worth [35].

4.1.3.3 Multi-criteria workflow scheduling In this type of scheduling, many parameters are considered simultaneously which conflict often. It could be QoS-constrained even. In [16], VM consolidation was performed using prediction algorithms. Novel multi-criteria techniques were employed for selection of overloaded hosts and appropriate VMs. Results showed 98.11% reduction in a metric composed of migrations, violations in SLA, and consumption of energy.

Aggregation approach uses a simple average of an objective function to select a final solution. Mukhopadhyay et al. [69] reported a work that performed the final selection of solution based on aggregation function and optimization of aggregated fitness function.

£-approach User does not always know to keep a certain constraint for one criterion. Therefore, mostly for solving a bi-criteria problem, this approach is used [70].

4.1.3.4 Pareto approach In this approach, solutions cannot be more optimized across any dimension without being deteriorated across another dimension at the same time [71]. Solution set obtained after using the Pareto approach is called Pareto optimal solution. In [72], two scheduling problems based on Pareto optimization were investigated. The maximum earliness cost was the objective in the first and in the second, maximum earliness cost was objective for one agent, and total earliness cost was objective for a second agent. As per authors, problems could be fixed in polynomial time by predicting Pareto optimal points.

4.1.4 Why energy-aware scheduling?

Energy consumed by IT equipment (in data centers) is increasing at an extremely high rate, and release of GHGs from them is making earth unfit to live. Energy-aware scheduling is ubiquitous and must be used for a sustainable future. Otherwise, it will have adverse effects on the environment. For optimal use of resources (in data centers) and to achieve QoS, energy-efficient scheduling plays a major role. Underutilized resources being idle consume power at leisure, and on the other side, overutilized resources degrade the performance.

5 Optimization: a solution

Scheduling is a decision problem as discussed in Sect. 4.1.1, so it cannot have a precise solution. Heuristics-based optimization techniques can assist in generating an optimal solution, where the objective is either to minimize or maximize a certain parameter. Parameters are finalized as per criteria which could be single or multi-objective.

To solve the energy-aware scheduling problem (optimization problem), various methods/techniques are available. Energy-aware scheduling focuses upon energy and/or power as a criterion for scheduling. According to Beloglazov and Buyya [11], the rate at which a system carries out work is power, whereas energy is total work done at a definite time interval as illustrated in Eq. (1).

$$\text{Energy} = \text{Work} = \text{Power} * \text{Time} \tag{1}$$

According to [11], a decrease in the consumption of power does not always decrease energy consumption. For example, if there is a decrease in power consumed (decreasing CPU's load), then the program may take longer time thus consuming more energy. Power is of two kinds—(1) static and (2) dynamic. Static power or leakage power is power consumed by the system not in functioning state. It depends upon low-level system design, i.e., transistors and processor technology. Therefore, it is difficult to reduce. Dynamic power depends upon usage scenarios, the voltage that is supplied and frequency of the clock. It can be reduced by reducing voltage and clock frequency. Power is a significant design constraint for computing systems.

Power efficient computing is the main focus during research. However, due to the close correlation, energy and power are used interchangeably in the literature. The survey classifies various methods for reducing power or energy consumption.

5.1 Dynamic power management methods for optimization

The methods for dynamic power management are shown in Fig. 7.

Dynamic performance scaling DVFS reduces power consumption by frequency scaling (up/down) according to the requirement. If a CPU runs at a lesser frequency, less voltage will be required and hence less power consumption. Therefore, voltage and frequency are balanced dynamically, resulting in decreased power consumption. It may take longer time if a processor is working at lower frequencies. Processor's governor monitors this process to regulate the performance. Generally, the processor starts at lower frequencies and steadily increases with workloads and so is energy consumption. A major challenge is to reduce power consumption through DVFS while handling deadline constraints. In [73], authors tried to decrease energy consumption by making CPU work at lower frequencies as long as task deadlines could be guaranteed.

Slack reclamation is a technique that can be used with DVFS in order to meet timing constraints for completion of tasks. In parallel processing, many tasks execute simultaneously. Further, if completion of a task depends upon two preceding tasks and these two tasks complete at dissimilar times, a task that completes earlier can manage addition runtime called as slack. This additional time can be used by DVFS for energy efficiency [74].

Dynamic Component Deactivation (DCD) It involves deactivating or shutting down components, which are at idle (not used) state. DCD has transition overhead, which is insignificant in the case of small problems. However, such transitions can cause performance degradation and delays drawing additional power in some cases. Therefore, a transition is only required when the idle timing period is sufficient to pay off for overhead during transitions. In real life scenarios, it is impossible to predict future workload. Therefore, an estimation of actual transition requires historical data or some system model [75, 76].

Table 5 compares power reduction techniques. Most of the work is based on benchmarks or synthetic datasets in contrast to real data. Most commonly used simulators are CloudSim, Sniper, McPat in contrast to testbed or the experimental environment. DVFS in combination with other techniques is used for reducing power consumption, and most results are objective in nature (numerically specified). Future work includes implementation of the real-time environment while scheduling more

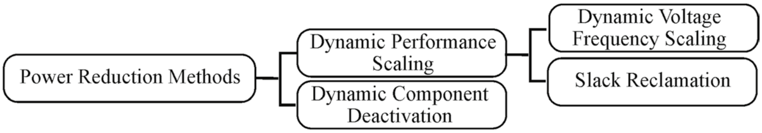
Power Reduction Methods
Dynamic Performance Scaling
Dynamic Component Deactivation
Dynamic Voltage Frequency Scaling
Slack Reclamation

Fig. 7 Methods for dynamic power management

Table 5 Comparison of power-aware scheduling techniques

References	Dataset	Tools	Techniques	Result and type	Limitations/future work
[63]	SPEC OMPM 2001 and NPB benchmark	Sniper	Global optimization, DVFS and cache-aware scheduler	Objective system performance improvement = 10%; Performance advantage = 4.2% on an average	Not mentioned
[77]	Randomly generated 100 jobs	Sniper, McPAT	Bin-based recipe packing heuristic, FF, FS, OTB	Objective complexity reduction = 84% as compared to RP	To develop generic framework to address platform heterogeneity
[78]	A benchmark program was used to generate a trace of dynamic power consumption	HotSpot PTScalar McPAT	Markovian analysis, an exact and approximate method for thermal analysis, DVFS, simulated annealing	Objective reduced power consumption by 15% for Tamb = 65 °C and for Tamb = 95 °C, it is more than 8%	Extending the work to some more general workloads
[79]	SPEC CPU 2000 benchmark suite	Xen hypervisor using Linux VMs for the × 86 architecture	DVFS during live migration	Objective power consumption was significantly reduced and shown with the help of graphs	The proposed scheme is not fit for suppression of controlled power consumption
[80]	Graph 500 Malstone benchmarks	Intel SCC many-core platform using Barrelfish multi-kernel OS	Dynamic power management through DVFS	Objective Extra energy saving = 24%; Energy–delay product reduction = 31.3%; Overhead reduction in execution time = 15.2% in comparison with latency un-aware approach	Ignored latency of frequency scaling, temperature effect on CPU chip's power
[55]	Used a 600 MHz Crusoe processor to obtain loads	Benchmarks including WDF, IIR, DPCM, 2D filter, and all-pole filter	DVS, EDTS, critical path based static scheduling	Objective Reduction in total energy consumption = 23.1% on average w.r.t. critical path scheduling method and 34.2% w.r.t parallelism-based scheduling approach	Not mentioned

Table 5 (continued)

References	Dataset	Tools	Techniques	Result and type	Limitations/future work
[73]	Tasks were submitted by cloud users with specified deadlines according to the SLA	CloudSim using CloudReport	EATS-FFD, EATS-WRR, DVFS, VM reuse	Objective Reduction in energy consumption of EATS-FFD = 20% compared with EFFD and of EATS-WRR = 12% compared with EWRR	Deployment on a real cloud platform
[81]	A number of tasks serve as an input value and is given	MATLAB	ACO, CS, VFS	Objective Increase in the efficiency and decrease in energy consumption is shown with graphs	To use the hybrid algorithm for more and more jobs of tasks
[20]	CPU power factors are derived from previous work	CloudSim	Heuristic, Backward and forward layer-based scheduling, DVFS, VM reuse	Objective Energy savings up to 30%	To run the experiments on Saluki Cloud managed by Eucalyptus with a few Beowulf clusters
[82]	Used real scientific workflow parameters to generate synthetic data (SIPHT, LIGO, and MONTAGE)	Cloud workflow simulator	DVFS, HEFT, slack reclamation	Objective Proposed algorithm achieved significant energy savings shown with graphs, sometimes by degrading execution time to reduce processors' idle time	To investigate the impact of different network characteristics, communication costs, job sensitivity on the algorithm's performance
[83]	DSPstone, Mediabench benchmark suites	SUIF, Wattch simulator	Power Gating DVS	Objective PACS can outperform by more than 33% and 41% in terms of energy–delay product and energy delay^2 product as compared to hard power gating	The work can also be applied to design a low CNC machine

Table 5 (continued)

References	Dataset	Tools	Techniques	Result and type	Limitations/future work
[84]	Real-world applications: Laplace equation solver, the LU decomposition and FFT	Made comparative analysis and simulator not mentioned	Heuristic task prioritization methods, HEFT, makespan-conservative energy reduction technique, DVS	Objective Compelling performance in reference to both completion time of application and consumption of energy. The comparison has been made, and numerical values have been given	Clock frequency transition overheads have been ignored
[85]	MATLAB and PJD script is used for trace file generation	Gem5 simulator	DPS, DPM using WCRQ, EDF	Objective Scheme performed on par with the solution that is optimal in the worst case and executed the real-time workload in an efficient way in other cases. Values are given in the form of tables	To deal with real-time applications having variable activation patterns
[87]	Used workloads used by others in their research work	Experimentally validated data center configuration and realistic parameter setting	DVFS	Objective Makespan improvement by 65% and dynamic energy by 20%	To include costs for cooling, i/o, communication-intensive applications, data transmission

SLA service level agreement, *FFT* fast Fourier transform, *PJD* periodic job releases with jitter and minimal release distances, *WCRQ* worst-case ready queue, *OS* operating system, *WDF* wave digital filter, *IIF* infinite impulse filter, *DPCM* differential pulse-code modulation device, *EDF* earliest deadline first, *DVFS* dynamic voltage–frequency scaling, *FF* first fit, *FS* frequency scaling, *OTB* online time balancing, *DVS* dynamic voltage scaling, *DPS* dynamic performance scaling, *RP* recipe packing, *PACS* power-aware code scheduling, *EADTS* energy-aware dynamic task scheduling, *EATS-FFD* energy-aware task scheduling-first fit decreasing, *VFS* voltage frequency scaling, *HEFT* heterogeneous earliest finish time, *CSN* computer numerically controlled, *EATS-WRR* energy-aware task scheduling-weighted round robin, *ACO* ant colony optimization, *CS* cuckoo search number of jobs, considering platform heterogeneity, network characteristics, communication costs, etc. The criteria or objective is to decrease power based on some parameters (objective or subjective). Various parameters used in power scaling are:

Sudhanshu et al. [63] developed a hierarchical framework for the management of power on many-core tiled processors. Weighted speed up, system throughput, cache-to-cache transfers were employed to improve system performance with DVFS and a special thread scheduler. In [77], parameters targeted were servers used, energy consumption, cumulative machine uptime (CMU) for energy savings while guaranteeing SLAs. A heuristic based on DVFS was proposed to combine different virtualized clusters on physical machines for batch-oriented cases. Morteza and Mehdi [78] provided an approach for analyzing and modeling of the real system serving stochastic workloads. Accuracy in terms of the average temperature of each core, thermal parameters including ambient temperature, convection resistance, convection capacitance, and others including relative error, and absolute error were focused upon to reduce power consumption. In [79], power consumption, performance overhead, execution time, CPU utilization were the parameters used. A scheme was devised which could estimate and regulate power consumption and its impact on performance to suit power capping schemes.

In [80], parameters were energy performance ratio, power, runtime performance, energy consumption, and energy delay product (EDP) index. The major thought was to examine latency characteristics. In [55], energy consumption per cycle, time and energy overhead parameters were focused upon. Energy-aware dynamic task scheduling (EDTS) algorithm was developed to test online communications between tasks and reduce the overall consumption of energy. The designed algorithm made use of static scheduling algorithm's results and blindly minimized consumption of energy. In [73], cloud servers' performance, VM overhead, resource, and CPU utilization were the parameters employed for improving performance in a cloud data center and also to make it energy efficient. Babukarthik et al. [81] focused on the number of processors, tasks, and speed of execution to minimize energy consumption while scheduling tasks. A hybrid algorithm was proposed with the advantage of ACO, cuckoo search, and voltage scaling. In [20], deadline, VM overhead, performance metrics, makespan, energy consumption, energy cost, CO_2 emission, provider's performance, the resource utilization rate were examined maintaining SLA and QoS constraints. The problem of scheduling scientific workflow applications in a time-dependent environment was addressed.

In [82], makespan, resource utilization, given deadline were the parameters to minimize the total energy consumption of scientific workflow. Tasks were allocated to heterogeneous machines having a deadline and different frequency capabilities. The proposed algorithm worked repeatedly for further scaling of frequency. In [83], performance, energy-related metrics were taken into consideration to reduce energy consumption. Authors presented a code scheduling approach that used DVS and power gating. The objective was to minimize consumption of power during application execution.

In [84], makespan, energy, schedule length ratio (SLR), and energy consumption ratio (ECR) were focused upon to address the problem of scheduling of precedence constrained parallel applications on multiprocessor computer systems. Algorithms with the incorporation of relative superiority metric (RS) and makespan-conservative energy reduction technique (MCER) greatly contributed to reducing energy consumption. The energy saving of energy-conscious scheduling (ECS) and EC$_{+\text{ idle}}$ was enabled by making use of the DVS technique. In [85], deadline constraints, execution time, and speed were the parameters used. An online scheme was presented to allocate speeds to hard real-time workloads on systems facing thermal problems. In [86], system performance, core utilization of active nodes, and node utilization were focused upon to target interference that applications experience at an inter-core granularity. Authors presented a model for improving system performance using slack-based scheduling. In [87], DVFS improved makespan by more than 65% and at the same time improved the dynamic energy by about 20%.

From the above-mentioned literature, it is found that the number of processors, servers, cores, resources, nodes, and CPU utilization are the most preferred parameters used in power scaling methods. It is considered to be the most favored technique for hardware-based optimization as the frequency of publications in this domain has increased in the last years (Sect. 7).

5.2 Heuristics-based optimization methods

Software-based optimizations include resource scheduling, allocation techniques, throttling, and use of parallel programs. Being an NP-Hard problem, scheduling is performed by using various software-based optimization techniques—heuristics and metaheuristics. Figure 8 shows the classification of optimization techniques used in energy-aware scheduling.

Heuristics are usually dependent on the problem type, whereas metaheuristics do not depend on the type of problem and can be applied to any problem [88]. Heuristics are generally based on the local search.

Metaheuristics are the advanced version of heuristics based on the generalized local search or global search and can be thought of guiding principle to design underlying heuristics [89].

Approximate Algorithms Heuristics have less time complexity than traditional methods. The aim of heuristic is to yield rapidly a solution that is decent enough for

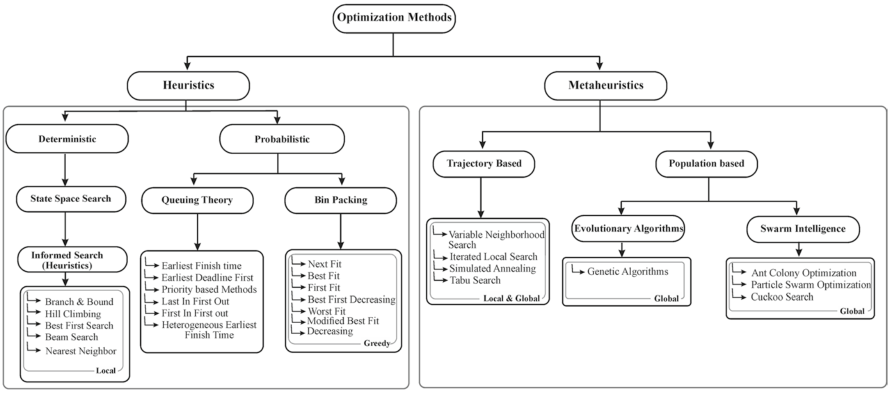

Fig. 8 Classification of heuristics-based optimization methods/techniques

resolving the problem. Heuristics are based on theories or experimental experiences, but approximation algorithms have a solid theoretical foundation.

Global search [90, 91] Heuristics are generally based on local search, whereas the metaheuristics do not get stuck in local optima as they use generalized local search or global search. In a global search, whole solution space is searched to find a minimum or maximum value. Generally, methods used in global search do not use greedy approach because entire search space needs to be examined and globally best is selected. They could utilize dynamic methods.

Local search Local search finds a minimum or a maximum solution in local space. Local search algorithms find an estimated optimal solution out of possible candidate solutions by moving from one solution to others within a time interval. These algorithms may use greedy approaches. Works on the local search are [90, 92, 93].

Greedy methods Greedy methods make the best choice at a particular instant of time. Thus, the local optimal solution is selected with an expectation that it would lead to a global best solution. At each step, greedy decisions are made to ensure the optimization of an objective function. The greedy algorithm cannot go backward to change the decision. [94–97] use greedy algorithms for scheduling.

Dynamic methods In dynamic programming, the problem is distributed into smaller problems, and each smaller problem is solved only once, resulting in a decrease in the number of computations. The solution is stored for next time usage. This method is advantageous when there is an exponential growth of repeating subproblems as a function of the size of the input. Authors in [98] used dynamic programming to find an approximate method for energy-efficient scheduling. Both dynamic and greedy approaches can be applied to the same problem; the difference is that the greedy approach does not reconsider its decision, whereas dynamic approach may keep on refining choices. Heuristics can be implemented independently or in combination with other optimization algorithms to give better efficiency.

Methods based on these techniques are described as follows:

A deterministic model produces a single outcome at every instant based on all given input values. Deterministic methods employ state space search algorithms. *Probabilistic methods* are based on randomness for the accomplishment of objectives.

State Space Search visits the entire space to reach a solution following certain rules. *Informed search* use heuristics (a function whose result indicates the next move). A branch and bound algorithm selects the optimum answer of an optimization problem. The entire space of the solution domain is examined for searching the best solution. The limits in objective function are merged with the last best solution. It continues to improve the solution once originated. Parts of solution space are found completely keeping the path with the lowest cost as a target [98, 99]. In the state space search, many states are traversed to reach a final state or goal state. Sequences of actions, which lead to the goal state, constitute the solution [100]. In hill climbing, the search continues in the direction which optimizes the cost using a greedy approach. There are many variants of hill climbing—the best neighbor, the first or nearest neighbor. The best-first search uses an evaluation function based on heuristic to explore the graph. The best-first search can be optimized to reduce the memory required and is called the beam search.

Table 6 compares the state space search (informed search) deterministic optimization methods used in energy-aware scheduling. Maximum datasets are either randomly generated. Mostly simulations are performed, and the results obtained are numerically specified (objective). Future scope includes increasing the complexity of networks and consideration of system components (disk, main memory, and communication networks) for energy efficiency. The objective is to decrease energy consumption based on some parameters (objective or subjective) that are as follows:

In [101] nominal execution time, nominal system utilization, system slackness, resource utilization were the parameters used. The focus was on the allocation of resources and robustness based on QoS constraints. Szynkiewicz et al. focused on traffic difference, power reduction, and QoS constraints. Design of a framework for centralized and hierarchal variants for a low energy-consuming network was made. Two control levels were implemented, network-based mechanism and local mechanism. Network-wide optimization problem was formulated in two ways, and an efficient algorithm was developed to solve it. Total system runtime, utility, QoS, average energy, and power consumption were the parameters used in [43]. A general adaptive task model was presented by utilizing existing ways of real-time adaptation for fault tolerance and graceful degradation [102]. Optimal solutions using heuristics were presented to get maximum advantages within the limited energy budget and a known time to recharge. A design space exploration (DSE) [103] method was developed to present architecture having multiple cores and optimal scheduling. Its efficiency was proved with large and hard graph problem. In [104], the technique used was Spreading Activation Partial Order Planner (SA-POP). Other techniques based on precedence constraints were applied to find harms and enhance ordering restrictions for autonomous coordination.

It is found that nominal execution time, total system runtime, utility, number of cores, number of buses, system utilization are the most preferred parameters in deterministic methods. As per the literature, deterministic algorithms are generally used with power scaling and genetic techniques.

Queuing Theory: It comes under the category of probabilistic methods. In the queuing theory [105], queues are presented and analyzed. Construction of models is done to estimate the length of queues and their waiting time. Queues are represented in mathematical equations for proving theorems known as Markov chains. Various scheduling policies can be used at queuing nodes and represented mathematically. First come first-serve (FCFS) [89] algorithm schedules processes by managing tasks or resources in order of their arrival times. It works on the principle of first in, first out (FIFO). The other algorithm is last in, first out (LIFO). It serves the task which has shorter waiting time first. Minimum completion time (MCT) and minimum execution time (MET) [106] are two heuristic algorithms. Min–Min and Max–Min are also two heuristics based on MCT and MET. Min–Min [106] picks the machine which gives MCT and assigns the smallest task to that machine. It increases the value of makespan, but does not consider the availability of resources while scheduling. Thus, completion and execution time come to be almost the same [106]. Other algorithms include priority-based scheduling using priority queues where jobs are

Table 6 Comparison of state space search (deterministic) technique

	Dataset	Tools	Techniques	Result and type	Limitations/future work
[101]	Generated data with sensors	A simulation environment was setup using designated hardware	Evolutionary algorithm, branch and bound, integer linear programming	Objective Improvement in system slackness as compared to genitor-based heuristic on an average by 4.65% and 6.46% incomplete and partial allocation scenario respectively	Not mentioned
[102]	Synthetic and real network topologies	Developed a green packet-level network simulator	DPM, DPS, branch and bound	Objective The results have been given objectively in the form of tables	The developed scheme is useful for small networks due to limitation inapplicability in medium and large networks because of their complexity
[43]	Real-time task sets were generated randomly for simulation	The parametric simulator was developed to model adaptive aspects of real-time systems	DP, DVS, branch and bound, state space search	Objective Achieved desired runtime with much-improved utility gain. Values have been given in tables	To incorporate system components, like a disk, main memory, and communication networks into the designed model
[103]	Two benchmarks: RGBOS and the second set included hard test examples	Not mentioned	GAA, best-first state space search algorithm using A-star, partitioning and pruning techniques	Objective Results show that a significant number of cores can be saved for providing reliability and fault tolerance	Not mentioned
[104]	Not mentioned	Not mentioned	SA-POP, RACE that employs state space search	Subjective The proposed scheme yields an architecture which will increase to allocation and large planning problems without being unaccomplished	The only possibility of scheduling and allocation of resources was considered by SA-POP at the system level, while the harder node-level resource and allocation optimization was considered by RACE

DPM dynamic power management, *DP* dynamic programming, *DVS* dynamic voltage scaling, *GAA* geometric algorithm, *DPS* dynamic performance scaling, *RACE* resource allocation and control engine, *SA-POP* spreading activation partial order planner, *SA-POP* spreading activation partial order planner, *RACE* resource allocation and control engine executed on the basis of size, time, etc. These include EDF, Shortest Job First (SJF), Earliest Finish Time (EFT), Heterogeneous Earliest Finish Time (HEFT), etc.

Table 7 compares queuing theory techniques based on dataset, tools, techniques, results, and future scope. Mostly synthetic datasets and real-world traces are used for evaluation. Techniques such as FCFS, HEFT, EDF are used, and most of the results are objective.

In [107], energy consumption, slack factor, good state probability, bad state probability, number of nodes were the parameters used for scheduling of periodic messages and to decrease overall consumption of energy in a wireless network. Thanavanich and Uthayopas [108] used a metric called SLR and ECR to balance energy consumption and makespan simultaneously. Two energy-efficient cloud-based scheduling approaches called Enhancing Heterogeneous Earliest Finish Time (EHEFT) and Enhancing Critical Path on a Processor (ECPOP) were proposed. They tried to achieve more energy reduction and satisfy performance constraints. The proposed approach used performance metric ratio of effectiveness (RE) to find a processor which is ineffective. In [82], makespan, utilization, user deadline were the parameters used to allocate tasks on heterogeneous machines with a deadline and diverse frequency capabilities. Authors relied upon the fact that even the minimum frequency may not always prove to be energy efficient.

In [74], makespan, energy, SLR, and ECR were the parameters used on multi-processor computer systems. Algorithms with RS and MCER significantly contributed to reducing energy consumption. In essence, an energy saving of ECS and $EC_{+\text{idle}}$ was enabled due to the exploitation of the DVS technique. In [85], deadline constraints, execution time, and speed were parameters used. An online scheme was presented which considered deadline constraints while providing speeds to systems processing real-time workloads having thermal issues. In [34], execution time, makespan, and total cost were used to optimize the cost of purchasing IaaS to achieve scientific workflow execution within specific deadlines. Authors used PSO along with VNS to optimize numerous factors such as a number of machines, price, scheduling to minimize total cost. Jingcao and Marculescu et al. [109] developed an effective scheme for energy-aware scheduling, which considered a delay in communication by parallel scheduling of transactions with computation for Network-on-Chip (NoC) architectures. They also handled execution time and cost. Zhang et al. [110] simulated the thermal-aware task scheduling algorithm and thermal-aware task scheduling algorithm-backfilled based on thermal information and resource information obtained in the Center for Computational Research (CCR) log files. Data center average, maximum temperature, job response time, impact on the environment, consumed power were measured.

It is clear that QoS parameters, benchmark programs, and real data are also focused upon while reducing energy consumption. EFT and other algorithms based on EFT are most widely used algorithms in queuing theory. Network characteristics, power consumption by hardware components, and cost of data transfer are not considered significant in queuing theory methods.

Bin packing is a probabilistic heuristic technique which aims to switch off idle servers by packing the hosts on available VMs [111]. It involves VM migrations on physical machines to fulfill requests by utilizing a minimum number of servers

[112]. It uses next-fit, first-fit and best-fit scheduling algorithms. The next fit algorithm utilizes the same bin, used during the last processed item. First fit is a modification of the next fit; it examines earlier bins and selects the best. Best fit selects bin with minimum space wastage and is considered to be finest. First fit decreasing (FFD) and best fit decreasing (BFD) are modifications of these algorithms that are based on sorting.

Table 8 compares bin packing-based energy-aware scheduling techniques. The data include real measurements, real-time traces, and synthetically generated datasets. Experiment setups include either hardware devices or simulators. Techniques include MBFD, minimum migrations (MM), best-fit heuristics.

Taheri et al. [113] tried to decrease the consumption of energy in CC data centers revising VM scheduling method while keeping QoS parameters as high as possible. In [114], an investigation was performed to find performance assurances that could be severely proven for heuristics (MM and MBFD) according to capacity and number of hosts to indicate the effectiveness of approximation. Viswanathan et al. [115] proposed a novel resource heuristic framework that used a best-fit heuristic for reallocation of unfinished tasks to alternate (backup) service providers. Zeng et al. [116] targeted energy-efficient scheduling of real-time periodic tasks. Some constraints like idle power, ineffective speed, and application-specific power characteristics, etc., were also associated with them. An adaptive minimal bound first-fit (AMBFF) algorithm was proposed for both dynamic-priority and fixed-priority multiprocessor.

In [117], the objective was to provide a computational cost and performance benefit analysis of schemes in terms of both feasibility and overall energy consumption. Experimental evaluation was performed to check the impact of partitioning heuristics, admission control algorithms, and speed assignment schemes by introducing a hybrid metric. Kandhalu et al. [118] studied energy-efficient scheduling of periodic real-time tasks with implicit deadlines on-chip multi-core processors using normalized power consumption to indicate its performance. In [77], parameters targeted were servers used, energy consumption, cumulative machine uptime (CMU) for energy savings while guaranteeing SLAs. In [73], cloud servers' performance, VM overhead, resource, and CPU utilization were used to improve performance and energy efficiency. In [119], Hancong et al. proposed a scheduling approach named pre-ant policy. It consisted of a model for prediction using mathematics and a scheduler using improved ant colony algorithm.

In bin packing techniques, power is not aggressively reduced without considering performance. Most used algorithms are best-fit heuristics and modified best, and worst fit algorithms.

Metaheuristics are heuristics about heuristics, and they provide better results by avoiding local optima. They can be based on local search, global search, and both. So, there is an overlapping classification as shown in Fig. 8. There are many classifications related to these heuristics/metaheuristics-based optimization techniques. Figure 8 lists only algorithms related to energy-aware scheduling in CC. They can be put under multiple classes like swarm intelligence in nature-inspired, or Tabu search can come under memory-based metaheuristics. Therefore, to explain all the possible permutations is beyond the scope. So, algorithms used in energy-aware scheduling are covered and classified in the best possible way.

Table 7 Comparison of techniques using queuing theory in energy-aware scheduling

References	Dataset	Tools	Techniques	Result and type	Limitations/future work
[107]	Not mentioned	Not mentioned	EDF, FIFO, DHSA, SHLA	Objective The proposed algorithm performed better than the SISA throughout the range. Values were compared in the form of graphs	To extend the work to exploit parallelism by non-interfering message transmissions in a multi-hop environment
[108]	Randomly generated synthetic task graphs	Not mentioned	EHEFT and ECPOP	Subjective EHEFT has greater average SLR than HEFT for computation intensive application	To focus on energy consumption ration (ECR)
[82]	Generation of synthetic data which was based on benchmarks SIPHT, LIGO, and MONTAGE	Cloud workflow simulator [26]	DVFS, HEFT, slack reclamation	Objective Graphs have been made to show energy savings, even at the cost of execution time to decrease processors' idle time	To examine the impact of the cost of communication, the sensitivity of jobs. Different network characteristics on the performance of the algorithm
[84]	Real-world applications: Laplace equation solver, the LU decomposition and FFT	Made comparative analysis and simulator not mentioned	Heuristic task prioritization methods, HEFT, makespan-conservative energy reduction technique, DVS	Objective Compelling algorithm performance in relation to both energy consumption and application completion time	Clock frequency transition overheads have been ignored

Table 7 (continued)

References	Dataset	Tools	Techniques	Result and type	Limitations/future work
[85]	MATLAB script was used to create trace files using the PJD model	Gem5 simulator	DPS, DPM using WCRQ, EDF	Objective Scheme performed on par with the solution that is optimal in the worst case and executed the real-time workload in an efficient way in other cases. Values are given in the form of tables	To expect application pattern variability of real-time systems
[109]	Two categories of random benchmarks were generated	Environment included MP3/H263 audio/video (A/V); C++	EAS algorithm, EFT, Heuristic (Greedy), slack allocation	Objective More energy consumption on an average by 55% and 39% for category I and II benchmarks compared with EAS, respectively	To consider environments with different network topologies or deterministic routing schemes
[34]	Four scientific workflows	Instances by Amazon EC2	PSO, EST, EFT	Objective Same results as other two algorithms and in some cases even better	To consider communication overhead to better simulate the real situation when time a task takes for execution is not known before

Table 7 (continued)

References	Dataset	Tools	Techniques	Result and type	Limitations/future work
[110]	CCR logs were submitted	Simulated a real data center environment	FCFS policy, TASA and TASA-B	Objective Average temperature decreased by 16.1 F and reduction in maximum temperature by 6.1 F by TASA. TASA-B can decrease the response time of a job by 13.9%, average temperature by 14.6 F. and increased response time overhead by 11%	To save task-temperature profiles, storage access and additional memory is required

FFT fast Fourier transform, *PJD* periodic job releases with jitter and minimal release distances, *CCR* center for computational research, *EDF* earliest deadline first, *FIFO* first in first out, *DHSA* direct hop scheduling algorithm, *SHLA* static hop length-based algorithm, *EAS* energy-aware scheduling, *EFT* earliest finish time, *PSO* particle swarm optimization, *EFT* earliest finish time, *EHEFT* enhancing heterogeneous earliest finish time, *ECPOP* enhancing critical path on processor, *FCFS* first come first served, *SISA* simple iterative scheduling algorithm, *SLR* schedule length ratio, *CPOP* critical path on a processor, *TASA* thermal-aware task scheduling algorithm, *TASA-B* thermal-aware task scheduling algorithm-backfilled, *HEFT* heterogeneous earliest finish time, *WCRQ* worst-case ready queue, *DPS* dynamic performance scaling, *DPM* dynamic performance management, *DVFS* dynamic voltage–frequency scaling

Trajectory-based methods use a single search strategy, and they focus on a single candidate solution. Trajectory-based algorithms that use both local search and global search are simulated annealing, Tabu search, iterated local search, variable neighborhood search, greedy randomized adaptive search procedure (GRASP). However, algorithms that use global search are generally *population-based metaheuristics* which use multiple candidates as search points, and characteristics of the population are used to guide the search. Examples are ant colony optimization (ACO), particle swarm optimization (PSO), and cuckoo Search.

Iterated local search is an improvement of local search in which search is repeated each time starting from a different state using certain criteria until the best solution is found. It helps to avoid getting stuck in local optima. Memory might be used to keep a record of previously visited states. Variable neighborhood search is based on dynamically changing neighbors. Random neighbors list can be made, but the certain sequence is to be followed. Iterated local search is applied and move to the next state is possible only if a better solution can be generated. Simulated annealing is inspired by metallurgical annealing. It is the type of trajectory-based metaheuristic that is used to generate global optimum in the large search space as per the objective function.

Tabu search uses improved local search and memory to avoid cycles. Memory records recently visited states and prevented moving toward them. The best state that has not been visited yet is chosen in each iteration until the algorithm is stopped at a terminating condition.

GRASP generates solution using dynamic constructive heuristic and randomization. The next state is chosen at random. It continues to improve the solution until the best is found.

Table 9 compares (metaheuristics) techniques based on trajectory methods. In [120], completion time, disc utilization, processor utilization, the power consumed were used for scheduling of VMs to physical machine considering energy efficiency, synchronizing utilization of the processor, disc, and cost of migration. In [121], the number of migrations, performance degradation due to migrations, SLA violations were the parameters considered to evaluate the proposed framework that consolidates VMs while taking care of QoS. Alkhashai et al. [122] considered makespan, utilization of resources and, cost for scheduling of tasks in a cloud environment. The proposed algorithm was able to reduce time, cost, and increase resource utilization. In [123], to maintain energy performance trade-off, a mechanism was introduced that focused on probability functions and a number of cycles generated to reduce carbon footprints.

It can be seen that CloudSim and MATLAB are the most widely used tools in the case of trajectory-based methods and results are objective in nature. However, real cloud implementation is lacking behind.

Population-based metaheuristics are further divided into evolutionary algorithms and algorithms based on swarm intelligence.

Evolutionary algorithm (EA) EA is a subgroup of evolutionary computation, which is based on metaheuristic technique. It uses various nature-inspired mechanisms such as reproduction, mutation, and crossover. Genetic algorithms [88, 124] are a most popular type of EA.

Table 8 Comparison of bin packing-based energy-aware scheduling techniques

References	Dataset	Tools	Techniques	Result and type	Limitations/future work
[113]	Synthetic dataset (randomly generated)	CloudSim	MBFD	Objective The presented approach achieved better results than other algorithms in terms of violations in SLA, the number of migrations having same CPU energy consumption	To make the value of maximum SLA violation that happens for a user to average SLA violation value while try to reduce both values
[114]	Synthetic dataset	Not mentioned	MM, MBFD	Objective Algorithms MM and MBFD give optimum results, however, not always when used in combination. Approximation results under suitable conditions	Practical implementation of suggestions given has not been done
[115]	Service requester submits an object's image which is to be recognized under deadline constraints	Mobile devices based on Linux and android having heterogeneous features	BF heuristic	Objective Significant impact on response time, relevance, quality of mobile applications	The designed framework is not better than RR under variable and highly stable operating conditions
[116]	Randomly generated	Synthetic multiprocessor with homogeneous architecture	AMBFF algorithm, dynamic-priority, and fixed-priority multiprocessor scheduling	Objective Average energy reduction of 16.1%, 17.2%, and 18.1% w.r.t. WFD method at MTU = 0.2 for XScale, PowerPC, and DSP models, respectively	Ignored constraints like idle power, discrete speed, inefficient speed, power characteristics of applications which are associated with parallel DVFS applications

Table 8 (continued)

References	Dataset	Tools	Techniques	Result and type	Limitations/future work
[117]	Generated tasks	Actual DVS architecture, namely Intel XScale	Partitioned scheduling, admission control algorithm, WF	Objective Sys-clock speed assignment scheme combined with admission control is best in both offline and online scenarios. WF is best in an offline scenario	Not mentioned
[118]	Actual processor measurements	A simulation environment was set up	WFD, task partition, RMS, SFAA	Objective A significant gain in energy (saves 55% more power) is provided by SFAA in comparison with WFD	To deal with cases when idle time cannot be used by DVFS to deal with battery powered system's energy consumption
[77]	Randomly generated 100 jobs	Sniper McPAT	Bin-based recipe packing heuristic, FF, FS, OTB	Objective Complexity reduction = 84% as compared to RP	To develop generic framework to address platform heterogeneity
[73]	Tasks along with deadlines were submitted by cloud users in accordance with SLA	CloudSim using CloudReport	EATS-FFD, EATS-WRR, DVFS, VM reuse	Objective Reduction in energy consumption of EATS-FFD = 20% compared with EFFD and of EATS-WRR = 12% compared with EWRR	Deployment in a real cloud environment

Table 8 (continued)

References	Dataset	Tools	Techniques	Result and type	Limitations/future work
[119]	Real workload Google traces	Developed a light and powerful simulator that references CloudSim	Pre-Ant Policy, fractal mathematics, ACO, FF, RR, MMP	Objective Pre-ant policy represents 76.17%, 75.28%, and 60.98% improvement in energy consumption and in average utilization of CPU by about 6%, 52% and 65% w.r.t. FF, RR, and MM, respectively	Restrictions were there on resource requirements of a task

MBFD modified best fit decreasing, *MM* minimization of migrations, *BF* best fit, *ACO* ant colony algorithm, *WFD* worst fit decreasing, *RMS* rate monotonic scheduling, *AMBFF* adaptive minimal bound first-fit, *SFAA* single-clock domain multiprocessor frequency assignment algorithm, *SLA* service level agreement, *FF* first fit, *RR* round robin, *MMP* minimum migration power, *WF* worst fit, *RR* round robin, *DVS* dynamic voltage scaling, *DVFS* dynamic voltage–frequency scaling

Table 10 compares techniques based on genetic algorithms. In [125], for efficient scheduling, Multi-objective genetic algorithm (MOGA) was proposed which considered both optimization and global makespan to reduce energy consumption. Kołodziej et al. [126] formulated task scheduling for CC as a bi-objective minimization problem having makespan and energy consumption as criteria and QoS constraints. Youness et al. [103] focused on average deviation from optimality, number of cores, execution time, and scalability. A design space exploration (DSE) methodology was proposed to create architectures with multiple cores and optimal scheduling. In [127], cost of storage, computation, and data transfer were considered. In [128], performance, resource utilization, number of servers were considered. To reduce the number of running servers and resource wastage, a hybrid algorithm was proposed. In [129], parameters that were focused upon included makespan, VM resource utilization, the degree of imbalance, performance.

Few works on evolutionary algorithms have actually focused on QoS parameters. Researchers employed various techniques based on genetic algorithms for saving energy, but results cannot be compared or correlated because of different constraints and scenarios.

Swarm intelligence [130] is based on mutual behavior of a population of agents occurring in nature. It motivates from the behavior of animals as a group, how they interact and communicate among themselves. Metaheuristics-based swarm intelligence algorithms for scheduling include (1) ACO [131] which is inspired by the behavior of ants to discover the shortest path to the source of food. (2) PSO [131] is motivated by the social behavior of particles. PSO generally combines local search and global search methods for resource allocation. Cuckoo search [132] is motivated by blood parasitism of cuckoo species. Table 11 compares swarm intelligence-based energy-aware scheduling techniques on the basis of dataset, tools, techniques, results. Synthetically generated dataset and real-world traces are used for evaluation. Techniques like PSO, ACO, cuckoo Search are employed, and most results are objective. The main aim is to save energy by using different parameters.

Babukarthik et al. [81] focused on quality of schedule, number of tasks, number of processors, and speed of execution to minimize energy consumption while scheduling tasks. In [133], makespan, cost, job rejection ratio, number of jobs meeting the deadline, and user satisfaction were considered. Authors designed and developed CLOUD Resource Broker (CLOUDRB) for effective management of resources in the cloud. Jeyarani et al. [134] proposed an adaptive power-aware virtual machine provisioner (APA-VMP) which drew minimum power without compromising performance. In [135], a strategy for scheduling and resource provisioning for different workflows on IaaS was presented. It optimized application execution cost maintaining deadline constraints. In [34], execution time, makespan, and the total cost were the parameters used to optimize the cost of purchasing IaaS to achieve scientific workflow execution within specific deadlines. In [119], Hancong et al. proposed a scheduling approach named pre-ant policy. Faragardi et al. [22] presented a scheme for the allocation of resources considering energy efficiency, reliability, timing constraints, memory limitation, etc. Reliability and quality of schedule were also considered.

Table 9 Comparison of trajectory-based energy-aware scheduling techniques

References	Dataset	Tools	Techniques	Result and type	Limitations/future work
[120]	Collected processor and disc utilization traces of real enterprise servers	MATLAB	SA INLP	Objective Savings in power by 24.9% and 1.2% less degradation in performance as compared to other traditional algorithms	Consideration of network optimization and network workloads in the proposed model
[121]	Used Planet Lab data	CloudSim	Markov chain model, Parallel SA	Objective Better performance by 10%, 6.9%, 12.8% in case of the low, moderate, and high load as compared to LR-MMT	Implementation in a real environment, considering geographically distributed data centers, energy consumed by switching networks
[122]	Random data (synthetically generated)	CloudSim	Tabu search Best fit PSO	Objective The proposed algorithm performed better by 18.71%, 16.45%, and 13.99% on average with respect to cost, resource utilization, and execution cost as compared to PSO	Improvement in PSO algorithm using the worst fit and considering independent tasks
[123]	Random data	MATLAB	SA	Found best possible solution values	Considering energy consumed by network devices

SA simulated annealing, *INLP* integer nonlinear programming, *PSO* particle swarm optimization, *LR-MMT* local regression-minimum migration time

Table 10 Comparison of genetic algorithm-based energy-aware scheduling techniques

References	Dataset	Tools	Techniques	Result and type	Limitations/future work
[125]	Real workload traces	GridSim	Multi-objective GA	Objective Energy consumption reduced by 10% with the use of multi-objective GA as compared to other techniques	To use high-level plans to estimate minimum computational resources for the future workload
[126]	Generated many jobs	MATLAB, CloudSim	DVS, ETU_GA, ETDF_GA	Objective Proposed schemes can efficiently balance energy and makespan in CC represented in the form of graphs	To formulate problem to address trade-off among system performance and QoS
[103]	Two benchmarks: RGBOS and the second set included hard test examples	Not mentioned	GAA, best-first, state space search algorithm using A-star, partitioning and pruning techniques	Objective Results show that a significant number of cores can be saved for other work (fault tolerance and reliability)	Not mentioned
[127]	Used the public data of Amazon	Data centers in Amazon region	TLGGA	Objective The proposed algorithm performed 49% better than Hadoop and 40% better than other current algorithms	The excellence of the algorithm comes at the sake of time
[128]	Made a dataset according to some parameters (number of VMs, servers)	Used C language and LibGA package	A bio-inspired technique based on ordering GA. Developed COFFGA and CONFGA	COFFGA is better than PP, FF, FFD by 4%, 34%, and 39%, respectively	Consideration of problems in allocating resources using vector bin packing workloads having multiple capacities

Table 10 (continued)

References	Dataset	Tools	Techniques	Result and type	Limitations/future work
[129]	MetaCentrum2 workload trace	Real data center (Czech National Grid Infrastructure)	RGA	Objective Outperforms current approach for application assignment in terms of energy savings by 48% and static application management by 10% for resource utilization efficiency	No investigation of patterns which make easy the matching process between VMs and applications

ETDF_GA energy consumption time double fitness genetic algorithm, *GAA* geometric algorithm, *CONFGA* combinatorial ordering next fit genetic algorithm, *TLGGA* two-level grouping genetic algorithm, *COFFGA* combinatorial ordering first-fit genetic algorithm, *GA* genetic algorithm, *DVS* dynamic voltage scaling, *ETU_GA* energy consumption time unify genetic algorithm, *ET_GA* energy time first genetic algorithm, *PP* permutation pack, *FF* first fit, *FFD* first fit decreasing, *RGA* repairing genetic algorithm

Table 11 Comparison of swarm intelligence-based energy-aware scheduling techniques

References	Dataset	Tools	Techniques	Result and type	Limitations/future work
[81]	Synthetic dataset	MATLAB, Xen cloud platform, Microsoft Window Azure	ACO, CS, VFS	Objective Minimized consumption of energy with the help of a hybrid algorithm. Values are given in the form of tables	To utilize the algorithm for more jobs
[133]	Real-time HPC applications	MATLAB, Eucalyptus-based cloud environments	Deadline-based Job Scheduling, PSO	Objective PSO performs 1.23 times good than ACO, 1.3 times good than GA and yields 1.77 times good performance than RBA in terms of the number of deadlines satisfying tasks	To integrate semantic resource invention and SLA for improving QoS and scheduling
[134]	Randomly generated dataset	CloudSim	SAPSO	Objective Graphs have been made to indicate significant power reduction in comparison with existing policies	Framework design for modern power conservative data center using neural networks and adaptive provisioning service
[135]	Montage, LIGO, and CyberShake workflows were used	Java simulator	ASFLA	Objective Reduction in OEC by 41% in comparison with PSO. Many performance benefits in comparison with SFLA	To evaluate approaches designed on real cloud systems
[34]	The experiments used four scientific workflows	Cloud instances by Amazon EC2	PSO, EST, EFT	Objective Same results as other two algorithms and in some cases even better	To consider the communication overhead and the scenario when the time of execution of tasks is not known in advance

Table 11 (continued)

References	Dataset	Tools	Techniques	Result and type	Limitations/future work
[123]	Real workload Google traces	Developed a light and powerful simulator that reference CloudSim	Pre-ant policy, fractal mathematics, ACO, FF, RR, MMP	Objective Pre-ant policy represents 76.17%, 75.28%, and 60.98% improvement in energy consumption and in average utilization of CPU by about 6, 52 and 65% w.r.t. FF, RR, and MM respectively	Restrictions were there on resource requirements of a task
[19]	Real-world data	Common hosts containing system was used	SI technique based on ICA, EDF, online scheduling algorithm	Objective 17% reduction in consumption of energy by ICA, non-trust worthiness by 9%. Reduction in energy consumption by 83% as compared to power unaware policy.	Ignored complex component modes for simplification

HPC high-performance computing, *VFS* voltage frequency scaling, *ACO* ant colony optimization, *CS* cuckoo search, *PSO* particle swarm optimization, *SAPSO* self-aware particle swarm optimization, *ASFLA* augmented shuffled frog leaping algorithm, *EFT* earliest finish time, *FF* first fit, *MMP* minimum migration power, *ICA* imperialist competitive algorithm, *RR* round robin

Most used algorithms are PSO and ACO which helped to satisfy the goal of reducing energy consumption.

Miscellaneous techniques Table 12 compares these techniques. Experimental setup includes real cloud deployments; CloudSim is the most used simulator. Most of the result analysis is objectively discussed. Future work includes considering more constraints, implementation in a real environment, and developing a generic framework for energy savings.

6 Performance metrics for green cloud computing

Performance metrics for the green data center are categorized as basic metrics and extended metrics [46]. The basic metrics are a measure of environmental friendliness of data centers. The extended metrics are functions of basic metrics that generate detailed view about data center [49].

6.1 Basic metrics

These are used to illustrate the efficiency of data centers in terms of environmental effect.

6.1.1 Greenhouse gas emission

GHGs are gases in the atmosphere that absorb and produce heat rays in the thermal infrared range. GHG is CO_2 which constitutes 9–26% of effect [141]. Power consumed in data centers is enormous, and GHGs are released during power generation which causes harsh effects on the environment. Thus, GHG emission should be measured to check how green a data center is.

6.1.2 Humidity

Moisture content in the air is called humidity [142]. Hardware failures are caused due to high humidity. The quantity of water in the air is measured by relative humidity. Relative humidity difference (RHD) (Eq. (2)) is the difference between the relative humidity of return air and air supply in the data center:

$$\text{RHD} = \text{Return air relative humidity} - \text{Supply air relative humidity} \tag{2}$$

6.1.3 Thermal metrics

They play a significant role in maintaining the data center's efficiency.

Table 12 Miscellaneous techniques

References	Dataset	Tools	Techniques	Result and type	Limitations/future work
[136]	Randomly generated	Real Cloud using Open Nebula	EMLS-ONC	Objective Reduction in energy consumed by 15.6% w.r.t. the OpenNebula scheduler and 13% more VMs were assigned	To deploy the algorithm in EGI grid infrastructure
[54]	Generated differently sized tasks	Private cloud infrastructure at the BSC	MHRA	Objective It allows to save an average of −22.44%, −33.17%, −27.08% and −31.50% for different scenarios	Calculation of the cost of the model
[137]	A synthetic and real dataset	Linux based server., C++, MATLAB	A scheme for remapping of tasks in an offline environment and mixed technique of integer and quadratic programming, energy-gradient-based heuristic	Objective Time for design space exploration was reduced by 99% having 5% variation in the ideal solution. Reduction in energy used in communication by 35% and overhead in migration by 20% for single- and double-fault scenarios. It gives 50% more throughput per unit energy than existing schemes	To consider the energy of task computation and to minimize the overhead of mapping storage

Table 12 (continued)

References	Dataset	Tools	Techniques	Result and type	Limitations/future work
[138]	Randomly generated workflows and real traces	Amazon AWS EC2 CloudSim	SOLID	Objective SOLID is better than SOLID-R EFT-MER-EL and EFT-MER-R by 12.30%, 60.88%, and 115.53%, respectively on the basis of normalized makespan. SOLID shows improvements up to 75.64% in terms of resource utilization	Investigation of robust approaches for management of resources and scheduling of tasks and to study the effect of timeliness on accuracy
[60]	Generated 50 jobs by Swim MapReduce workload generator	Simulator developed in Java	Techniques for online VM reconfiguring, Provisioning algorithms	Objective Cura model produces more than 80% decrease in the cost of infrastructure and 65% lower job response times	To integrate pricing model and resource management techniques
[16]	Real data using SPEC power benchmark and CoMon project	CloudSim	WMA, MTPVS policy	Objective 98.11% decrease in a unit which is composed of violations in SLA, consumption of energy and migrations performed	Real cloud implementation, considering heterogeneous environment for online VM placement
[139]	Synthetic dataset	Experimental set up was done	VMSA, integer linear programming model	Objective Solutions obtained by VMSA used 15.22% less physical resources	To study flexible resource provisioning

Table 12 (continued)

References	Dataset	Tools	Techniques	Result and type	Limitations/future work
[140]	Used Planet Lab and Guangdong key lab's statistical data	CloudSim	A redesigned energy-aware heuristic framework	Objective Reduction in consumed energy by 21–34%, violations in SLA by 84–92%, energy performance metric by 87–94%, and 63% reduction in time for execution	Real-world environment evaluation considering bandwidth, IO and storage
[44]	Generated and assigned tasks randomly	A comparison was made only	List scheduling heuristics, Min–Min algorithm	Objective Better for consistent instances	Implementation of presented heuristic in Green cloud simulator of energy-aware CC and to modify the heuristic to deal with thermal behavior

EC2 elastic compute cloud, *VMSA* performance-preserving VM splitting and assignment, *BSC* Barcelona computer centre, *AWS* Amazon web services, *EMLS-ONC* energy-aware multi-start local search algorithm for an OpenNebula-based cloud, *MHRA* multi-heuristic resource allocation algorithm, *WMA* window moving average policy, *MTPVS* multi-criteria TOPSIS with prediction VM selection, *EFT-MER* exploring laxity time multi encryption, *EFT-MER* exploring laxity time with random encryption strategy

6.1.3.1 Data center temperature To attain system reliability, an optimal temperature range is between 20 to 24 °C [143]. If data center temperature is higher than 30 °C [144], it is recommended that no costly IT equipment should be kept there.

British thermal unit (BTU): Solutions for cooling a server room or a data center are governed by British Thermal Unit (BTU) [145]. A BTU is the measure of energy that is needed for increasing temperature of a pound of water by 1 °F.

Airflow performance index: It indicates the efficiency of the data center from a thermal point of view.

Cooling system efficiency metric: These include airflow efficiency (AE), cooling system efficiency (CSE), cooling system sizing (CSS), and water economizer utilization (WEU). AE (Eq. (3)) indicates how efficiently air passes through a data center.

$$\text{AE} = 1000 \times \frac{\text{The overall power of fan}}{\text{Overall airflow of the fan}} \tag{3}$$

CSE (Eq. (4)) is a measure of total efficiency in terms of cooling equipment usage, power withdrawn for cooling. It is defined as:

$$\text{CSE} = \frac{\text{Average power used by cooling systems}}{\text{Average load to be cooled}} \tag{4}$$

CSS (Eq. (5)) is a ratio of installed chilling capacity to highest chilling load [146].

$$\text{CSS} = \frac{\text{Installed capacity for cooling}}{\text{Highest load for cooling}} \tag{5}$$

WEU (Eq. (6)) is a measure of savings in energy by using a water-sider economizer system.

$$\text{WEU} = \frac{\text{Water economizer hours}}{24 \times 365} \tag{6}$$

WEU provides information on energy savings by using a water-side economizer system.

6.1.4 Power/energy metrics

They include Data Center Infrastructure Efficiency (DCiE or DCE), Power Usage Effectiveness (PUE), Heating, Ventilation and Air Conditioning (HVAC) System Effectiveness, Space, Watts and Performance (SWaP), Data Center Energy Productivity (DCeP). DCiE is widely accepted by industry [147, 148]. DCiE is calculated as mentioned in Eq. (7).

$$\text{DCiE} = \frac{\text{Power consumed by IT devices}}{\text{Overall facility power}} \tag{7}$$

PUE [147] measures the energy consumed by IT and non-IT equipment (cooling devices). It is defined in Eq. (8).

$$\text{PUE} = \frac{1}{\text{DCiE}} = \frac{\text{Overall facility power}}{\text{Power consumed by IT devices}} \tag{8}$$

The HVAC System Effectiveness (Eq. (9)) is a ratio of energy consumed by IT devices to HVAC (electrical energy for cooling, movement of a fan) system energy.

$$\text{HVAC Effectiveness } = \frac{\text{Energy consumption by IT devices}}{\text{HVAC} + (\text{Steam} + \text{Chilled Water} + \text{Fuel}) \times 293}, \quad (9)$$

where IT is total electrical energy consumed annually by IT devices. HVAC, fuel, steam, chilled water is annual electrical energy required for HVAC, fuel, steam, chilled water, respectively.

SWaP (Eq. (10)) measures energy efficiency by considering space, energy, and performance together [149].

$$\text{SWaP } = \frac{\text{System Performance}}{\text{Rack Space} \times \text{Consumption of Power}} \quad (10)$$

where performance is measured by using benchmarks set by the industry. Space is a measure of the height of the server in rack units. Power is in watts, which is used by the system during benchmark runs.

The DCeP (Eq. (11)) [150, 151] measures useful work done in comparison with consumption of energy.

$$\text{DCeP } = \frac{\text{Useful work done}}{\text{Total energy consumption}} \quad (11)$$

6.2 Extended metrics

Extended metrics give detailed information about the data center and are categorized into multiple indicators and total cost of ownership.

6.2.1 Multiple indicators

Multiple indicators include data center indicators and data center sub-level indicators.

Data center indicators include server utilization, network utilization, storage utilization, and data center utilization.

Server usage measures actions of the processor in contrast to its maximum capability during uppermost frequency state. Network usage is the ratio of bandwidth consumed to bandwidth capacity in the data center. Storage usage is a percentage of storage consumed compared to total storage within the data center. Data center utilization indicates the amount of power consumed by IT equipment comparative to the real capability of the data center.

Data center sub-level indicators Power, cooling, airflow, weight, and area constitute sub-level indicators [147]. They help to measure various inefficiencies in data centers.

6.2.2 Total cost of ownership (TCO)

TCO signifies cost required to buy, construct, run, and maintain a data center [152]. TCO includes capital expenses which include initial investments, cost of cooling equipment, power consumption, space, and operational expenses, etc. So, the cost of the data center should be in terms of dollars per watt. Generally, power and cooling forms 80% of the capital cost and rest 20% is spent on construction of data center [153]. Operational cost includes monthly costs of running a data center [153], e.g., implementation techniques, climate costs, etc.

The studies that have measured the values of these metrics to find the efficiency of data centers are [37, 146].

7 Global analysis

The rapid development of CC has escalated the need of energy efficiency. According to the best of our knowledge, there is a scarcity of broad scientometric analysis (empirical study) which provides a view of the present status of research in this domain. There is a need to explore the trends of research in this evolving field. Scientometric analysis [154] has been conducted to find global research trends which can serve as a direction for further research activities, collaborative research, sharing of knowledge. Publications mentioned in Sect. 5 are included in the analysis to examine research in this field. This survey evaluates the publications according to global regions, journals, conferences, year of publishing, the research community, and fund provisioning. The discussion below is based on trends found from this analysis. Peer researchers may interpret the trends differently.

Figure 9 illustrates the frequency of publications in various conferences and journals on energy-aware scheduling in CC. Journal considered are "Journal of Supercomputing (JSC), Journal of Parallel and Distributed Computing, Future Generation Computer Systems (FGCS), IEEE Transactions on Parallel and Distributed Computing (ITPD) and IEEE Communication Surveys and Tutorials (ICST)." Trends indicate that research work in this domain is published equally in journals and conferences. The reason may be the evolving and dynamic nature of the topic itself [10]. It

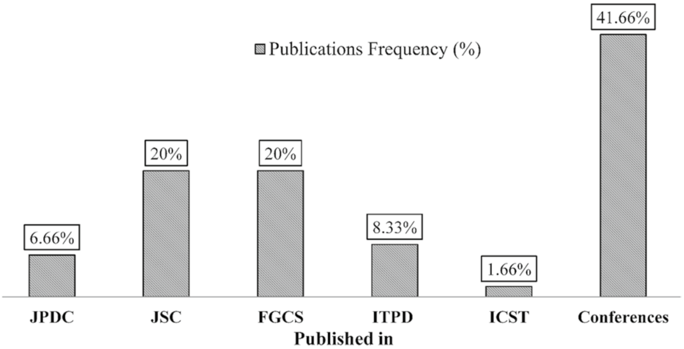

Fig. 9 Journal- and conference-wise publications in all sub-domains in energy-aware scheduling

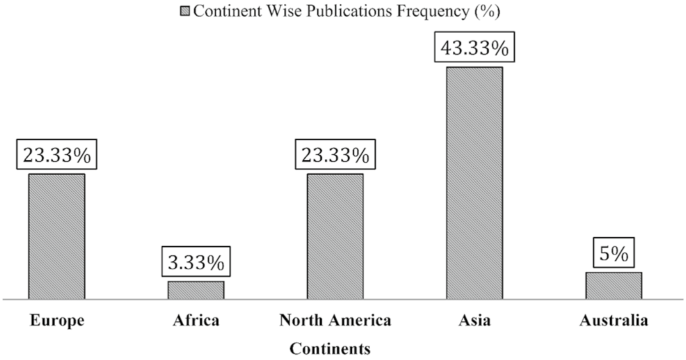

Fig. 10 Continent-wise publications in all sub-domains in energy-aware scheduling

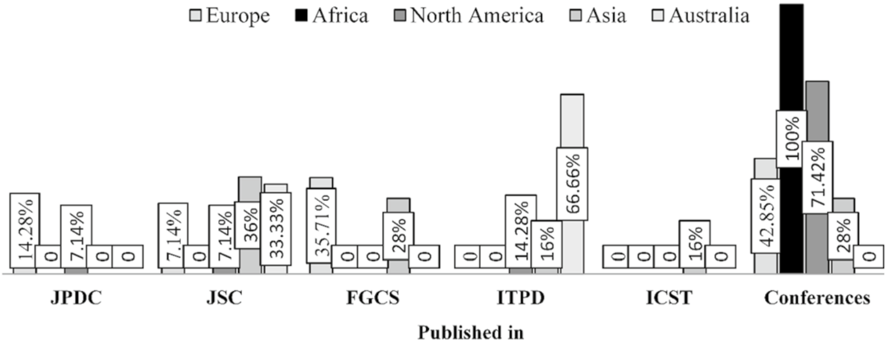

Fig. 11 Continent-wise publications according to journals and conferences

can also be analyzed that FGCS and JSC are most preferred journals in the research community. Figure 10 illustrates a geographical viewpoint of this research domain. It is clear that Asia, North America, and Europe are more active research continents, whereas others contribute least. Asian people contribute a major part (43.33%) in this research domain.

Figure 11 illustrates a more detailed viewpoint by providing insights on publication details. European researchers prefer to publish a major part of their research contribution in conferences and FGCS. North America publishes a major part of research contribution in conferences. Asia prefers to publish in FGCS, JSC, and conferences, whereas Australia and Africa have the least contribution. African researcher prefers to publish in conferences. Australian researcher prefers to publish in ITPD and JSC. Figure 12 illustrates the frequency of publications annually in this domain. The year 2016 depicts maximum frequency (21.66%) of publications. Coming years may have an increase in contribution or publications. Table 13 shows the comparative analysis of published and funded work in all sub-domains. DVFS has a maximum number of publications which account for 23.33% of total publications. Out of this 23.33% of total publications, 13.33% have got grants. In miscellaneous field, 13.33% of publications are funded out of

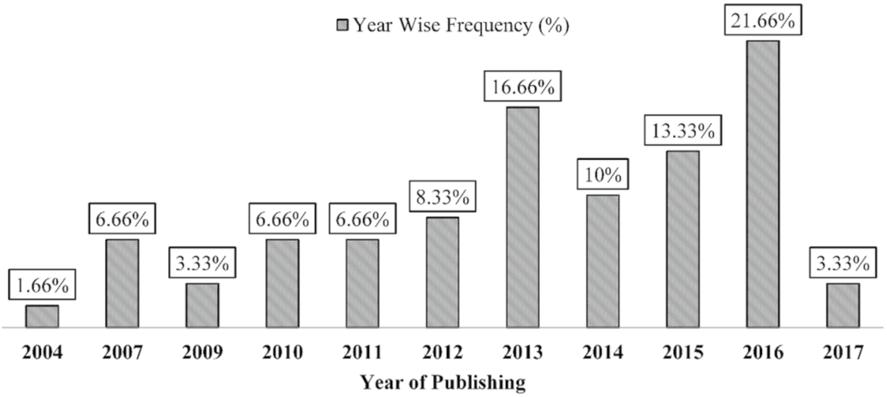

Fig. 12 Year-wise publications in all sub-domains in energy-aware scheduling

total 16.66%. Minimum funding is in a subdomain of trajectory-based methods in which only 1.6% works have got funding out of a total of 6.66%. Figure 13 illustrates the contribution in research by different groups in this domain. Academic

Table 13 Comparative analysis of published and funded

Sub-domains	Funded (%)	Total papers (%)
DVFS	13.33	23.33
State space search	5	8.33
Queuing theory	6.66	13.33
Bin packing	5	13.33
Trajectory-based	1.6	6.66
Genetic algorithms	8.33	10
Swarm intelligence	3.33	13.33
Miscellaneous	13.33	16.66

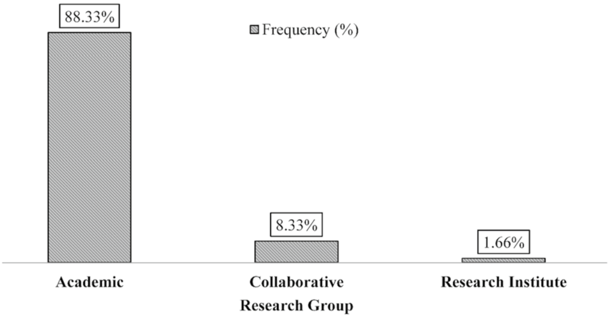

Fig. 13 Research contribution by different groups

research contribution is major in contrast to minor contribution from the collaborative effort of academia and industry. Research institutes of industry lack behind.

8 Research issues and a strategy for future research

Energy efficiency in CC is an active and vast research area. Literature indicates that researchers are working on a particular sub-part of the solution. The problem of energy consumption is addressed by using various mechanisms such as virtualization, multi-core architectures, parallel processing, power-aware methods, thermal-aware methods, bio-inspired methods. These mechanisms in turn use various optimization/scheduling algorithms like PSO, ACO, HEFT, FCFS, best fit, genetic algorithm, etc. Limited research is carried on investigating the tradeoff between energy efficiency and QoS compliance as per SLA. However, QoS is the foremost parameter of concern for cloud clients. To explore both performance and energy efficiency, the challenges include maintaining the reliability of a server because power cycling may reduce it, performing VMs consolidation without affecting QoS, accurate application performance management in a virtualized environment while maintaining SLA [65]. Major issues on energy optimization for CC are listed below:

- Much work is done on the development of energy-efficient framework using consolidation techniques, but that is not generic [77].
- How to balance energy efficiency and VM placement [66].
- Solution for VM performance degradation [121, 155].
- Implementation of optimization techniques for energy efficiency in the cloud environment taking into consideration a very high number of jobs [73, 81].
- Investigation of the impact of network characteristics, communication cost, overhead, system components like the disk, main memory while scheduling applications on cloud [82].
- The research in the area of harvesting renewable energy at different sites of a data center is at its initial stage. The challenge is to minimize nonrenewable energy usage, carbon cost and to investigate the effect of inter-region migration of VMs [156].
- Another issue is to select the VM that is to be migrated keeping in consideration the running application, SLA, data transfer, etc., and in some cases, multiple VMs are to be migrated. So, sharing of network resources effectively is a challenge [156].
- Performance management by synchronization with SLAs for the satisfaction of users [65].
- Minimizing energy consumption considering heterogeneous workloads and runtime migrations of VMs [156].
- Storing a large amount of data and its processing can lead to energy wastage. So, streamlining data storage, processing, energy consumption simultaneously is another issue [6].

A strategy for future research must address these issues. Work in harvesting renewable energy (data centers) is in its early stages. So a scheduling strategy employing heuristics-based optimization techniques that can migrate VMs to hosts of a green data center is needed. The solution should be generic and also take into account CPU utilization, RAM, network and storage devices. The new scheduling strategy should provide compliance to new paradigms of fog and edge computing by providing low-latency services to users so as to consider QoS factors (energy consumption and performance). Constraints associated with applications like speed, power in an idle state, power characteristics of an application [114] should be incorporated in the proposed strategy. The design strategy should also consider those applications in which execution time is not known beforehand [34].

9 Conclusion

CC has brought a revolution in today's world by changing the way of delivering computing services. Almost all the online users use it in a direct or an indirect way. However, CSPs and cloud users face a lot of challenges. Challenges are to provide (1) energy efficiency (2) QoS (3) SLA compliance (4) load balancing (5) security (6) traffic management, and (7) cost-effectiveness. Energy efficiency is a major challenge as data centers consume tremendous power and release GHGs, deteriorating the environment. This survey presents a taxonomy of energy-aware optimizations, dynamic power management methods. It is justified that heuristics-based optimization (scheduling) is a generic solution for achieving energy efficiency in CC.

This survey shows that (1) probabilistic algorithms specifically genetic algorithms (PSO, ACO) and bin packing-based algorithms (best fit, EFT, WFD) are the most extensively used techniques for reducing energy consumption. (2) DVFS is the commonly used method for power saving. (3) CloudSim is the widely adopted simulator for evaluation and validation in contrast to real data center implementation. (4) Benchmark programs and real-world traces are the commonly used sources of data. (5) Majorly used parameters are resource utilization, number of cores, node utilization, number of servers, and CPU utilization in contrast to QoS as per SLA compliance. (6) Widely focused GC metrics are GHG emission and PUE which tells the environmental friendliness of a data center.

Many factors such as network characteristics, communication cost, overhead, energy consumption by system components like the disk, main memory were ignored in past research studies. There is a need to design scheduling algorithms which consider the energy consumed by these factors and also work for heterogeneous workloads. Paper also classifies the results into (1) objective results (numerically specified) (2) subjective results (no discrete values). Most of the previous research contributions have objective results. The rapid increase in objective work in this domain indicates its dynamism. The paper concludes by a comprehensive scientometric analysis based on publications while considering bibliometric parameters for analysis. The outcome of this can serve as a direction for future research contribution. Future work will concentrate on the implementation of techniques for energy efficiency in CC and also on investigating the use of scheduling techniques in newly emerged fog and edge environment.

Funding This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

1. Walsh B (2014) Your data is dirty: the carbon price of cloud computing. TIME. http://time.com/46777/your-data-is-dirty-the-carbon-price-of-cloud-computing. Accessed 4 June 2017
2. Ten key marketing trends in 2017 and ideas for exceeding customers' expectations (2017). https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=WRL12345USEN. Accessed 9 Feb 2018
3. Worldwide internet user penetration from 2014 to 2021 (2017) eMarketer. https://www.statista.com/statistics/325706/global-internet-user-penetration. Accessed 9 Feb 2018
4. Liu L et al (2016) RE-UPS: an adaptive distributed energy storage system for dynamically managing solar energy in green datacenters. J Supercomput 72:295–316. https://doi.org/10.1007/s11227-015-1529-2
5. Energy Statistics of the European Union (2015) Concepts and definitions on all flows ("Aggregates") and products used in the energy statistics on quantities. http://ec.europa.eu/eurostat. Accessed 21 June 2017
6. Kaur T, Chana I (2015) Energy efficiency techniques in cloud computing: a survey and taxonomy. ACM Comput Surv 48:22. https://doi.org/10.1145/2742488
7. Living Planet Report (2014) Species and spaces, people and places. https://www.worldwildlife.org/pages/living-planet-report-2014. Accessed 21 June 2017
8. Cloud could cut energy data center consumption 31% by 2020 (2011). https://www.telecomengine.com/cloud-could-cut-energy-data-center-consumption-31-by-2020/9. Accessed 26 Nov 2018
9. Google, Facebook and Apple lead on green data centers (2014). https://www.theguardian.com/sustainable-business/greenpeace-report-google-facebook-apple-green-data-centers. Accessed 19 Feb 2018
10. Zhan Z, Liu X, Gong Y, Zhang J (2015) Cloud computing resource scheduling and a survey of its evolutionary approaches. ACM Comput Surv 47:63. https://doi.org/10.1145/2788397
11. Beloglazov A, Buyya R, Lee Y, Zomaya A (2011) A taxonomy and survey of energy-efficient data centers and cloud computing systems. Adv Comput 82:47–111. https://doi.org/10.1016/B978-0-12-385512-1.00003-7
12. Yu J, Buyya R, Ramamohanarao K (2008) Workflow scheduling algorithms for grid computing. In: Xhafa F, Abraham A (eds) Metaheuristics for scheduling in distributed computing environments. Studies in computational intelligence. Springer, Berlin, pp 173–214
13. Wu F, Wu Q, Tan Y (2015) Workflow scheduling in cloud: a survey. J Supercomput 71:3373–3418. https://doi.org/10.1007/s11227-015-1438-4
14. Yuan S, Shahzad J, Kun A, Yi S (2013) State-of-the-art research study for green cloud computing. J Supercomput 65:445–468. https://doi.org/10.1007/s11227-011-0722-1
15. Final Version of NIST Cloud Computing Definition Published (2011). https://www.nist.gov/news-events/news/2011/10/final-version-nist-cloud-computing-definition-published. Accessed 26 July 2017
16. Arianyan E, Taheri H, Sharifian S (2016) Novel heuristics for consolidation of virtual machines in cloud data centers using multi-criteria resource management solutions. J Supercomput 72:688–717. https://doi.org/10.1007/s11227-015-1603-9
17. Min-yi GUO (2010) Green computing: connotation and tendency. Comput Eng 36:1–7
18. Guo B, Shen Y, Shao Z (2009) The redefinition and some discussion of green computing. Chin J Comput 32:2311–2319
19. Liu J, Pacitti E, Valduriez P, Mattoso M (2015) A survey of data-intensive scientific workflow management. J Grid Comput 13:457–493. https://doi.org/10.1007/s10723-015-9329-8

20. Cao F, Zhu M (2013) Energy-aware workflow job scheduling for green clouds. In: IEEE International Conference on Green Computing and Communications IEEE and Internet of Things (iThings/CPSCom) and IEEE Cyber, Physical and Social Computing. IEEE, pp 232–239
21. Hosseinimotlagh S, Khunjush F (2014) A cooperative two-tier energy-aware scheduling for real-time tasks in computing clouds. In: 22nd Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP). IEEE, pp 178–182. https://doi.org/10.1109/PDP.2014.91
22. Faragardi R, Rajabi A, Shojaee R, Nolte T (2013) Towards energy-aware resource scheduling to maximize reliability in cloud computing systems. In: IEEE International Conference on Embedded and Ubiquitous Computing and High-Performance Computing and Communications. IEEE, pp 1469–1479. https://doi.org/10.1109/HPCC.and.EUC.2013.208
23. Tchernykh A et al (2014) Energy-aware online scheduling: ensuring quality of service for IaaS clouds. In: IEEE International Conference on High-Performance Computing and Simulation. IEEE, pp 911–918. https://doi.org/10.1109/HPCSim.2014.6903786
24. Vilaplana J et al (2015) An SLA and power-saving scheduling consolidation strategy for shared and heterogeneous clouds. J Supercomput 71:1817–1832. https://doi.org/10.1007/s11227-014-1351-2
25. Beloglazov A, Buyya R (2011) Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers. In: Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science. ACM, New York, pp 23–50. https://doi.org/10.1145/1890799.1890803
26. Luo J et al (2013) Temporal load balancing with service delay guarantee for energy cost optimization in internet data centers. IEEE Trans Parallel Distrib Syst 25:775–784. https://doi.org/10.1109/TPDS.2013.69
27. Lee K, Kulkarni I, Pompili D, Parashar M (2012) Proactive thermal management in green data centers. J Supercomput 60:165–195. https://doi.org/10.1007/s11227-010-0453-8
28. Van L et al (2016) An efficient Session_Weight load balancing and scheduling methodology for high-quality telehealthcare service based on WebRTC. J Supercomput 72:3909–3926. https://doi.org/10.1007/s11227-016-1636-8
29. Zhang C, Chang C, Yap RH (2014) Tagged-MapReduce: a general framework for secure computing with mixed-sensitivity data on hybrid clouds. In: 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. IEEE, pp 31–40. https://doi.org/10.1109/CCGrid.2014.96
30. Watson P (2012) A multi-level security model for partitioning workflows over federated clouds. J Cloud Comput Adv Syst Appl. https://doi.org/10.1186/2192-113X-1-15
31. Sharif S, Taheri J, Zomaya Y, Nepal S (2013) MPHC: preserving privacy for workflow execution in hybrid clouds. In: IEEE International Conference on Parallel and Distributed Computing, Applications and Technologies. IEEE, pp 272–280. https://doi.org/10.1109/PDCAT.2013.49
32. Xu G et al (2017) Bandwidth-aware energy efficient flow scheduling with SDN in data center networks. Future Gener Comput Syst 68:163–174. https://doi.org/10.1016/j.future.2016.08.024
33. El-Boghdadi M (2009) Power-aware routing for well-nested communications on the circuit switched tree. J Parallel Distrib Comput 69:135–142. https://doi.org/10.1016/j.jpdc.2008.09.003
34. Netjinda N, Sirinaovakul B, Achalakul T (2014) Cost optimal scheduling in IaaS for dependent workload with particle swarm optimization. J Supercomput 68:1579–1603. https://doi.org/10.1007/s11227-014-1126-9
35. Kumar N, Vidyarthi P (2017) An energy-aware cost-effective scheduling framework for heterogeneous cluster system. Future Gener Comput Syst 71:73–88. https://doi.org/10.1016/j.future.2017.01.015
36. Rubio-Montero J, Huedo E, Mayo-García R (2017) Scheduling multiple virtual environments in cloud federations for distributed calculations. Future Gener Comput Syst 74:90–103. https://doi.org/10.1016/j.future.2016.03.021
37. Avgerinou M, Bertoldi P, Castellazzi L (2017) Trends in data centre energy consumption under the European code of conduct for data centre energy efficiency. Energies. https://doi.org/10.3390/en10101470
38. Wilkins J (2017) How clean is the energy used by tech companies for cloud computing? Scientific American. https://www.scientificamerican.com/article/cloud-computings-substantial-footprint. Accessed 8 July 2017
39. Blazek M, Chong H, Loh W, Koomey G (2004) Data centers revisited: assessment of the energy impact of retrofits and technology trends in a high-density computing facility. J Infrastruct Syst 10:98–104

40. Beloglazov A, Abawajy J, Buyya R (2012) Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future Gener Comput Syst 28:755–768. https://doi.org/10.1016/j.future.2011.04.017
41. Moore D, Chase S, Ranganathan P, Sharma K (2005) Making scheduling "cool": temperature-aware workload placement in data centers. In: USENIX Annual Technical Conference, pp 61–75
42. Angel E, Bampis E, Kacem F (2012) Energy-aware scheduling for unrelated parallel machines. In: IEEE International Conference on Green Computing and Communications. IEEE, pp 533–540. https://doi.org/10.1109/GreenCom.2012.78
43. Niewiadomska-Szynkiewicz E et al (2014) Dynamic power management in energy-aware computer networks and data-intensive computing systems. Future Gener Comput Syst 37:284–296. https://doi.org/10.1016/j.future.2013.10.002
44. Diaz O et al (2011) Energy-aware fast scheduling heuristics in heterogeneous computing systems. In: IEEE International Conference on High-Performance Computing and Simulation. IEEE, pp 478–484. https://doi.org/10.1109/HPCSim.2011.5999863
45. Wang X, Wang Y, Cui Y (2014) A new multi-objective bi-level programming model for energy and locality-aware multi-job scheduling in cloud computing. Future Gener Comput Syst 36:91–100. https://doi.org/10.1016/j.future.2013.12.004
46. Benoit A, Çatalyürek V, Robert Y, Saule E (2013) A survey of pipelined workflow scheduling: models and algorithms. ACM Comput Surv 45:50. https://doi.org/10.1145/2501654.2501664
47. Singh S, Chana I (2016) QoS-aware autonomic resource management in cloud computing: a systematic review. ACM Comput Surv 48:42. https://doi.org/10.1145/2843889
48. Orgerie C, Assuncao D, Lefevre L (2014) A survey on techniques for improving the energy efficiency of large-scale distributed systems. ACM Comput Surv 46:47. https://doi.org/10.1145/2532637
49. Wang L, Khan U (2013) Review of performance metrics for green data centers: a taxonomy study. J Supercomput 63:639–656. https://doi.org/10.1007/s11227-011-0704-3
50. Alkhanak N, Lee P, Khan R (2015) Cost-aware challenges for workflow scheduling approaches in cloud computing environments: taxonomy and opportunities. Future Gener Comput Syst 50:3–21. https://doi.org/10.1016/j.future.2015.01.007
51. Smanchat S, Viriyapant K (2015) Taxonomies of workflow scheduling problem and techniques in the cloud. Future Gener Comput Syst 52:1–12. https://doi.org/10.1016/j.future.2015.04.019
52. Mei J, Li K (2012) Energy-aware scheduling algorithm with duplication on heterogeneous computing systems. In: ACM/IEEE 13th International Conference on Grid Computing. IEEE, pp 122–129. https://doi.org/10.1109/Grid.2012.32
53. Houben K, Halang A (2014) An energy-aware dynamic scheduling algorithm for hard real-time systems. In: 3rd Mediterranean IEEE Conference Embedded Computing. IEEE, pp 14–17
54. Juarez F, Ejarque J, Badia M (2018) Dynamic energy-aware scheduling for parallel task-based application in cloud computing. Future Gener Comput Syst 78:257–271. https://doi.org/10.1016/j.future.2016.06.029
55. Qiu M et al (2012) Towards power-efficient smartphones by energy-aware dynamic task scheduling. In: IEEE 9th International Conference on Embedded Software and Systems High-Performance Computing and Communication. IEEE, pp 1466–1472. https://doi.org/10.1109/HPCC.2012.214
56. Kyriazis D et al (2008) An innovative workflow mapping mechanism for grids in the frame of quality of service. Future Gener Comput Syst 24:498–511. https://doi.org/10.1016/j.future.2007.07.009
57. Liu J, Guo J (2016) Energy efficient scheduling of real-time tasks on multi-core processors with voltage islands. Future Gener Comput Syst 56:202–210. https://doi.org/10.1016/j.future.2015.06.003
58. Li K (2017) Scheduling parallel tasks with energy and time constraints on multiple many core processors in a cloud computing environment. Future Gener Comput Syst. https://doi.org/10.1016/j.future.2017.01.010 (In Press)
59. Jha S et al (2017) Shared resource aware scheduling on power-constrained tiled many-core processors. J Parallel Distrib Comput 100:30–41. https://doi.org/10.1016/j.jpdc.2016.10.001
60. Palanisamy B, Singh A, Liu L (2015) Cost-effective resource provisioning for MapReduce in a cloud. IEEE Trans Parallel Distrib Syst 26:265–1279. https://doi.org/10.1109/TPDS.2014.2320498
61. Aaronson S (2005) Guest column: NP-complete problems and physical reality. ACM Sigact News 36:30–52. https://doi.org/10.1145/1052796.1052804
62. Chawla Y, Bhonsle M (2012) A study on scheduling methods in cloud computing. Int J Emerg Trends Technol Comput Sci 1:12–17

63. Garey R, Johnson S (1990) Computers and intractability: a guide to the theory of NP-completeness. Freeman, New York
64. Tao F, Feng Y, Zhang L, Liao W (2014) CLPS-GA: a case library and Pareto solution-based hybrid genetic algorithm for energy-aware cloud service scheduling. Appl Soft Comput 19:264–279. https://doi.org/10.1016/j.asoc.2014.01.036
65. Buyya R, Beloglazov A, Abawajy J (2010) Energy-efficient management of data center resources for cloud computing: a vision, architectural elements, and open challenges. In: Proceedings of the 2010 International Conference on Parallel and Distributed Processing Techniques and Applications, pp 1–12
66. Zhou Z et al (2017) Minimizing SLA violation and power consumption in cloud data centers using adaptive energy-aware algorithms. Future Gener Comput Syst 86:836–850. https://doi.org/10.1016/j.future.2017.07.048
67. Verma A, Kaushal S (2015) Cost-time efficient scheduling plan for executing workflows in the cloud. J Grid Comput 13:495–506. https://doi.org/10.1007/s10723-015-9344-9
68. Ku L, Li W, Chen Y, Liu R (2016) Advances in energy harvesting communications: past, present, and future challenges. IEEE Commun Surv Tutor 18:1384–1412. https://doi.org/10.1109/comst.2015.2497324
69. Mukhopadhyay A, Maulik U, Bandyopadhyay S, Coello C (2014) A survey of multiobjective evolutionary algorithms for data mining: part I. IEEE Trans Evol Comput 18(1):4–19. https://doi.org/10.1109/TEVC.2013.2290086
70. Fard M, Prodan R, Barrionuevo D, Fahringer T (2012) A multi-objective approach for workflow scheduling in heterogeneous environments. In: Proceedings of the 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. IEEE, pp 300–309. https://doi.org/10.1109/CCGrid.2012.114
71. Kacem I, Hammadi S, Borne P (2002) Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic. Math Comput Simul 60:245–276. https://doi.org/10.1016/S0378-4754(02)00019-8
72. Wan L (2014) Pareto optimization for the two-agent scheduling problems with linear non-increasing deterioration. In: 10th International Conference Natural Computation. IEEE, pp 330–334
73. Alahmadi A et al (2015) An innovative energy-aware cloud task scheduling framework. In: 8th International IEEE Conference on Cloud Computing. IEEE, pp 493–500. https://doi.org/10.1109/CLOUD.2015.72
74. Lee C, Zomaya Y (2012) Energy efficient utilization of resources in cloud computing systems. J Supercomput 60:268–280. https://doi.org/10.1007/s11227-010-0421-3
75. Benini L, Bogliolo A, De-Micheli G (2000) A survey of design techniques for system-level dynamic power management. IEEE Trans Very Large Scale Integr (VLSI) Syst 8:299–316. https://doi.org/10.1109/92.845896
76. Albers S (2010) Energy-efficient algorithms. Commun ACM 53:86–96. https://doi.org/10.1145/1735223.1735245
77. Teng F et al (2017) Energy efficiency of VM consolidation in IaaS clouds. J Supercomput 73:782–809. https://doi.org/10.1007/s11227-016-1797-5
78. Mohaqeqi M, Kargahi M (2015) Thermal analysis of stochastic DVFS-enabled multicore real-time systems. J Supercomput 71:4594–4622. https://doi.org/10.1007/s11227-015-1562-1
79. Jeong J et al (2013) Analysis of virtual machine live-migration as a method for power-capping. J Supercomput 66:1629–1655. https://doi.org/10.1007/s11227-013-0956-1
80. Lai Z, Lam T, Wang L, Su J (2015) Latency-aware DVFS for efficient power state transitions on many-core architectures. J Supercomput 71:2720–2747. https://doi.org/10.1007/s11227-015-1415-y
81. Babukarthik G, Raju R, Dhavachelvan P (2012) Energy-aware scheduling using hybrid algorithm for cloud computing. In: 3rd International Conference on Computing Communication and Networking Technologies. IEEE, pp 1–6. https://doi.org/10.1109/ICCCNT.2012.6396014
82. Pietri I, Sakellariou R (2014) Energy-aware workflow scheduling using frequency scaling. In: 43rd IEEE International Conference on Parallel Processing Workshops. IEEE, pp 104–113. https://doi.org/10.1109/ICPPW.2014.26
83. Lee Y, Lin Y, Chang G (2014) Power-aware code scheduling assisted with power gating and DVS. Future Gener Comput Syst 34:66–75. https://doi.org/10.1016/j.future.2013.12.011

84. Lee C, Zomaya Y (2011) Energy conscious scheduling for distributed computing systems under different operating conditions. IEEE Trans Parallel Distrib Syst 22:1374–1381. https://doi.org/10.1109/TPDS.2010.208
85. Lampka K, Forsberg B, Spiliopoulos V (2016) Keep it cool and in time: with runtime monitoring to thermal-aware execution speeds for deadline constrained systems. J Parallel Distrib Comput 95:79–91. https://doi.org/10.1016/j.jpdc.2016.03.002
86. Dauwe D et al (2016) HPC node performance and energy modeling with the co-location of applications. J Supercomput 72:4771–4809. https://doi.org/10.1007/s11227-016-1783-y
87. Sun H, Stolf P, Pierson M (2017) Spatio-temporal thermal-aware scheduling for homogeneous high-performance computing data centers. Future Gener Comput Syst 71:157–170. https://doi.org/10.1016/j.future.2017.02.005
88. Suyyagh A, Tong G, Zilic Z (2016) Performance evaluation of meta-heuristics in energy-aware real-time scheduling problems. Jordan J Comput Inf Technol 2:68–85. https://doi.org/10.5455/jjcit.71-1450000176
89. Madni H, Latiff A, Abdullahi M, Usman J (2017) Performance comparison of heuristic algorithms for task scheduling in IaaS cloud computing environment. PLoS ONE. https://doi.org/10.1371/journal.pone.0176321
90. Winter A, Albonesi H (2008) Scheduling algorithms for unpredictably heterogeneous CMP architectures. In: IEEE International Conference on Dependable Systems and Networks with FTCS and DCC. IEEE, pp 42–51. https://doi.org/10.1109/DSN.2008.4630069
91. Agrawal P, Rao S (2014) Energy-aware scheduling of distributed systems. IEEE Trans Autom Sci Eng 11:1163–1175. https://doi.org/10.1109/TASE.2014.2308955
92. Guo L, Zhao S, Shen S, Jiang C (2012) Task scheduling optimization in cloud computing based on heuristic algorithm. J Netw 7:547–553. https://doi.org/10.4304/jnw.7.3.547-553
93. Verma A, Ahuja P, Neogi A (2008) pMapper: power and migration cost aware application placement in virtualized systems. In: Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware. IEEE, pp 243–264. https://doi.org/10.1007/978-3-540-89856-6_13
94. Gao Y, Wang Y, Gupta SK, Pedram M (2013) An energy and deadline aware resource provisioning, scheduling and optimization framework for cloud systems. In: IEEE International Conference on Hardware/Software Codesign and System Synthesis, IEEE, pp 1–10. https://doi.org/10.1109/CODES-ISSS.2013.6659018
95. Zhang S, Wang B, Zhao B, Tao J (2013) An energy-aware task scheduling algorithm for a heterogeneous data center. In: 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications. IEEE, pp 1471–1477. https://doi.org/10.1109/TrustCom.2013.178
96. Tchernykh A et al (2014) Energy-aware online scheduling: ensuring quality of service for IaaS clouds. In: IEEE International Conference on High-Performance Computing and Simulation (HPCS). IEEE, pp 911–918. https://doi.org/10.1109/HPCSim.2014.6903786
97. Li X et al (2017) Holistic virtual machine scheduling in cloud datacenters towards minimizing total energy. IEEE Trans Parallel Distrib Syst 29:1317–1331. https://doi.org/10.1109/TPDS.2017.2688445
98. Yang Y, Chen J, Kuo W, Thiele L (2009) An approximation scheme for energy-efficient scheduling of real-time tasks in heterogeneous multiprocessor systems. In: Proceedings of the Conference on Design, Automation and Test in Europe. IEEE, pp 694–699. https://doi.org/10.1109/DATE.2009.5090754
99. What is “branch and bound”. https://www.quora.com/What-is-branch-and-bound. Accessed 26 Aug 2017
100. State Space Search. https://www.computing.dcu.ie/~humphrys/Notes/AI/statespace.html. Accessed 26 Aug 2017
101. Shestak V et al (2008) A hybrid branch-and-bound and evolutionary approach for allocating strings of applications to heterogeneous distributed computing systems. J Parallel Distrib Comput 68:410–426. https://doi.org/10.1016/j.jpdc.2007.05.011
102. Pillai P, Huang H, Shin G (2003) Energy-aware quality of service adaptation. Technical report CSE-TR-479-03, University of Michigan
103. Youness H, Hassan M, Salem A (2010) A design space exploration methodology for allocating task precedence graphs to multi-core system architectures. In: IEEE International Conference on Microelectronics (ICM). IEEE, pp 260–263. https://doi.org/10.1109/icm.2010.5696133

104. Kinnebrew S et al (2007) A decision-theoretic planner with dynamic component reconfiguration for distributed real-time applications. In: 8th IEEE International Symposium on Autonomous Decentralized Systems. IEEE, pp 461–472. https://doi.org/10.1109/ISADS.2007.1
105. Zhang Q et al (2012) Dynamic energy-aware capacity provisioning for cloud computing environments. In: Proceedings of the 9th ACM International Conference on Autonomic Computing. ACM, New York, pp 145–154. https://doi.org/10.1145/2371536.2371562
106. Mathew T, Sekaran C, Jose J (2014) Study and analysis of various task scheduling algorithms in the cloud computing environment. In: IEEE International Conference on Advances in Computing, Communications and Informatics. IEEE, pp 658–664. https://doi.org/10.1109/ICACCI.2014.6968517
107. Kumar A, Manimaran G, Wang Z (2007) Energy-aware scheduling with deadline and reliability constraints in wireless networks. In: 4th International IEEE Conference on Broadband Communications, Networks and Systems. IEEE, pp 96–105. https://doi.org/10.1109/BROADNETS.2007.4550411
108. Thanavanich T, Uthayopas P (2013) Efficient energy aware task scheduling for parallel workflow tasks on hybrids cloud environment. In: International IEEE Conference on Computer Science and Engineering Conference. IEEE, pp 37–42. https://doi.org/10.1109/ICSEC.2013.6694749
109. Hu J, Marculescu R (2004) Energy-aware communication and task scheduling for network-on-chip architectures under real-time constraints. In: Proceedings of Design, Automation and Test in Europe Conference and Exhibition. IEEE, pp 234–239. https://doi.org/10.1109/DATE.2004.1268854
110. Zhang W et al (2016) Towards joint optimization over ICT and cooling systems in data centre: a survey. IEEE Commun Surv Tutor 18:1596–1616. https://doi.org/10.1109/COMST.2016.2545109
111. Gupta K, Katiyar V (2018) Energy-aware scheduling framework for resource allocation in a virtualized cloud data center. Int J Eng Technol 9:558–563. https://doi.org/10.21817/ijet/2017/v9i2/170902032
112. Ghribi C, Hadji M, Zeghlache D (2013) Energy efficient VM scheduling for cloud data centers: exact allocation and migration algorithms. In: 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. IEEE, pp 671–678. https://doi.org/10.1109/CCGrid.2013.89
113. Taheri M, Zamanifar K (2011) 2-phase optimization method for energy-aware scheduling of virtual machines in cloud data centers. In: IEEE International Conference on Internet Technology and Secured Transactions. IEEE, pp 525–530
114. Mann ZÁ (2015) Rigorous results on the effectiveness of some heuristics for the consolidation of virtual machines in a cloud data center. Future Gener Comput Syst 51:1–6. https://doi.org/10.1016/j.future.2015.04.004
115. Viswanathan H, Lee K, Rodero I, Pompili D (2015) Uncertainty-aware autonomic resource provisioning for mobile cloud computing. IEEE Trans Parallel Distrib Syst 26:2363–2372. https://doi.org/10.1109/TPDS.2014.2345057
116. Zeng G, Yokoyama T, Tomiyama H, Takada H (2009) Practical energy-aware scheduling for real-time multiprocessor systems. In: 15th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications. IEEE, pp 383–392. https://doi.org/10.1109/RTCSA.2009.47
117. AlEnawy A, Aydin H (2005) Energy-aware task allocation for rate monotonic scheduling. In: 11th IEEE Real-Time and Embedded Technology and Applications Symposium. IEEE, pp 213–223. https://doi.org/10.1109/RTAS.2005.20
118. Kandhalu A, Kim J, Lakshmanan K, Rajkumar R (2011) Energy-aware partitioned fixed-priority scheduling for chip multiprocessors. In: IEEE 17th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA). IEEE, pp 93–102. https://doi.org/10.1109/RTCSA.2011.75
119. Duan H, Chen C, Min G, Wu Y (2017) Energy-aware scheduling of virtual machines in heterogeneous cloud computing systems. Future Gener Comput Syst 74:142–150. https://doi.org/10.1016/j.future.2016.02.016
120. Sharifi M, Salimi H, Najafzadeh M (2012) Power-efficient distributed scheduling of virtual machines using workload-aware consolidation techniques. J Supercomput 61:46–66. https://doi.org/10.1007/s11227-011-0658-5
121. Rajabzadeh M, Haghighat T (2017) Energy-aware framework with Markov chain-based parallel simulated annealing algorithm for dynamic management of virtual machines in cloud data centers. J Supercomput 73:2001–2017. https://doi.org/10.1007/s11227-016-1900-y
122. Alkhashai M, Omara A (2016) An enhanced task scheduling algorithm on cloud computing environment. J Grid Distrib Comput 9:91–100

123. Ghosh A (2017) A well-organized energy efficient cloud data center using simulated annealing optimization technique. Int J Adv Res Comput Sci 8:974–977

124. Mezmaz M et al (2011) A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems. J Parallel Distrib Comput 71:1497–1508. https://doi.org/10.1016/j.jpdc.2011.04.007

125. Gabaldon E, Lerida L, Guirado F, Planes J (2017) Blacklist multi-objective genetic algorithm for energy saving in heterogeneous environments. J Supercomput 73:354–369. https://doi.org/10.1007/s11227-016-1866-9

126. Kolodziej J, Khan U, Xhafa F (2011) Genetic algorithms for energy-aware scheduling in computational grids. In: IEEE International Conference on P2P, Parallel, Grid, Cloud and Internet Computing. IEEE, pp 17–24. https://doi.org/10.1109/3PGCIC.2011.13

127. Zhang J et al (2016) Key-based data analytics across data centers considering bi-level resource provision in cloud computing. Future Gener Comput Syst 62:40–50. https://doi.org/10.1016/j.future.2016.03.008

128. Hallawi H, Mehnen J, He H (2017) Multi-capacity combinatorial ordering GA in application to cloud resources allocation and efficient virtual machines consolidation. Future Gener Comput Syst 69:1–10. https://doi.org/10.1016/j.future.2016.10.025

129. Vasudevan M et al (2017) Profile-based dynamic application assignment with a repairing genetic algorithm for greener data centers. J Supercomput 73:3977–3998. https://doi.org/10.1007/s11227-017-1995-9

130. Raju R, Amudhavel J, Kannan N, Monisha M (2014) A bio-inspired energy-aware multi objective chiropteran algorithm (EAMOCA) for hybrid cloud computing environment. In: IEEE International Conference on Green Computing Communication and Electrical Engineering. IEEE, pp 1–5. https://doi.org/10.1109/ICGCCEE.2014.6922463

131. Kalra M, Singh S (2015) A review of metaheuristic scheduling techniques in cloud computing. Egypt Inform J 16:275–295. https://doi.org/10.1016/j.eij.2015.07.001

132. Sivakumar Chitra, Madhusudhanan B (2016) Cloud workflow scheduling algorithms using cuckoo search (CS) with novel fitness function. Iioab J 7:261–268

133. Somasundaram S, Govindarajan K (2014) CLOUDRB: a framework for scheduling and managing high-performance computing (HPC) applications in science cloud. Future Gener Comput Syst 34:47–65. https://doi.org/10.1016/j.future.2013.12.024

134. Jeyarani R, Nagaveni N, Ram V (2012) Design and implementation of adaptive power-aware virtual machine provisioner (APA-VMP) using swarm intelligence. Future Gener Comput Syst 28:811–821. https://doi.org/10.1016/j.future.2011.06.002

135. Kaur P, Mehta S (2017) Resource provisioning and workflow scheduling in clouds using augmented shuffled frog leaping algorithm. J Parallel Distrib Comput 101:41–50. https://doi.org/10.1016/j.jpdc.2016.11.003

136. Kessaci Y, Melab N, Talbi G (2012). An energy-aware multi-start local search heuristic for scheduling VMs on the OpenNebula cloud distribution. In: IEEE International Conference on High-Performance Computing and Simulation (HPCS). IEEE, pp 112–118. https://doi.org/10.1109/HPCSim.2012.6266899

137. Das A, Kumar A, Veeravalli B (2014) Communication and migration energy aware task mapping for reliable multiprocessor systems. Future Gener Comput Syst 30:216–228. https://doi.org/10.1016/j.future.2013.06.016

138. Chen H et al (2017) Scheduling for workflows with security-sensitive intermediate data by selective tasks duplication in clouds. IEEE Trans Parallel Distrib Syst 28(2674):2688. https://doi.org/10.1109/TPDS.2017.2678507

139. Liu L et al (2016) VMSA: a performance preserving online VM splitting and placement algorithm in dynamic cloud environments. J Supercomput 72:3169–3193. https://doi.org/10.1007/s11227-015-1590-x

140. Cao Z, Dong S (2014) An energy-aware heuristic framework for virtual machine consolidation in Cloud computing. J Supercomput 69:429–451. https://doi.org/10.1007/s11227-014-1172-3

141. Kiehl T, Trenberth E (1997) Earth's annual global mean energy budget. Bull Am Meteor Soc 78:197–208. https://doi.org/10.1175/1520-0477(1997)078%3c0197:EAGMEB%3e2.0.CO;2

142. Fontecchio M (2007) Data center humidity levels source of debate. https://searchdatacenter.techtarget.com/news/1261265/Data-center-humidity-levels-source-of-debate. Accessed 20 Sept 2017

143. Rambo J, Joshi Y (2007) Modeling of data center airflow and heat transfer: state of the art and future trends. Distrib Parallel Databases 21:193–225. https://doi.org/10.1007/s10619-006-7007-3

144. Data Center Temperature (2009) 42U. http://www.42u.com/power/data-center-temperature.htm. Accessed 18 Sept 2017

145. Weaver T (2011) Cooling your datacenter/server room temperature control. http://www.bestpricecomputers.ltd.uk/servers/datacenter-cooling.htm. Accessed 2 Sept 2017

146. Mathew P, Ganguly S, Greenberg S, Sartor D (2009) Self-benchmarking guide for data centers: metrics, benchmarks, actions. Ernest Orlando Lawrence Berkeley National Laboratory, Berkeley

147. Belady C (2007) The green grid data center efficiency metrics: PUE and DCIE. https://www.premiersolutionsco.com/wp-content/uploads/TGG_Data_Center_Power_Efficiency_Metrics_PUE_and_DCiE.pdf. Accessed 17 Sept 2017

148. Verdun G et a. (2007) The green grid metrics: data center infrastructure efficiency (DCiE) detailed analysis. https://leonardo-energy.pl/wp-content/uploads/2017/08/greengridmetrics.pdf. Accessed 20 Sept 2017

149. GreenHill D (2011) SWaP (Space, Watts and Performance) metric. https://www.energystar.gov/ia/products/downloads/Greenhill_Pres.pdf. Accessed 23 Sept 2017

150. DCeP: Data Center Energy Productivity (2011) 42U. https://www.42u.com/measurement/dcep.htm. Accessed 24 Sept 2017

151. Haas J (2008) A framework for data center energy productivity. https://www.greenbiz.com/sites/default/files/document/GreenGrid-Framework-Data-Center-Energy-Productivity.pdf. Accessed 2 Sept 2017

152. Patterson K, Costello D, Grimm P, Loeffler M (2007) Data center TCO: a comparison of high-density and low-density spaces. Thermal challenges in next generation electronic systems. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.182.6237&rep=rep1&type=pdf. Accessed 22 Sept 2017

153. Barroso A, Clidaras J, Hölzle U (2013) The datacenter as a computer: an introduction to the design of warehouse-scale machines. Synth Lectures Comput Arch 8:1–154. https://doi.org/10.2200/S00516ED2V01Y201306CAC024

154. Heilig L, Voß S (2014) A scientometric analysis of cloud computing literature. IEEE Trans Cloud Comput 2:266–278. https://doi.org/10.1109/TCC.2014.2321168

155. Xu X et al (2017) A balanced virtual machine scheduling method for energy-performance trade-offs in cyber-physical cloud systems. Future Gener Comput Syst. https://doi.org/10.1016/j.future.2017.08.057

156. Khosravi A (2017) Energy and carbon-efficient resource management in geographically distributed cloud data centers. Dissertation, University of Melbourne

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.