Chapter 2

Literature Survey

2.1 Survey Process

This chapter provides a comprehensive review of various task scheduling techniques, irrespective of their specific scheduling type. The objectives of this review are threefold: *i) to classify heuristic algorithms, ii) to classify meta-heuristic algorithms, and iii) to classify hybrid algorithms*. The formulation of research questions guided the organization of this review:

i *Research Question: What are the relevant approaches it follows?*
It helps to find out the approach any algorithm is following to reach its goal. It helps us to classify it according to the classification mentioned.

ii *Research Question: What is the main objective of the algorithm, for example, Cost, deadline, throughput, etc.*
It helps to focus on the main objective a researcher is working on and to classify them according to the objective it is working on.

iii *Research Question: Is it workflow or independent scheduling along with the new work it is providing us?*
This question helped us to identify the algorithms which are not tested on workflow or tested. Also, the tool used for simulation is also used as a parameter to see if the simulation results are there or only a theoretical presentation is there.

2.2 Systematic Review of Cloud Computing Task Scheduling Literature

The review follows a systematic approach to comprehensively cover various aspects of the field. The methodology applied in reviewing the diverse set of research papers can be outlined as follows:

i **Formulation of Research Questions:**

- A set of initial research questions was crafted based on the ongoing discussions and trends in contemporary research on cloud computing task scheduling.

ii **Refinement of Research Questions:**

- The initially formulated research questions underwent a refinement process to ensure clarity and presentability in an anonymous manner. This step aimed at framing questions that would guide a structured and systematic literature survey.

iii **Systematic Literature Survey:**

- Following the refined research questions, a systematic literature survey was conducted, covering a wide spectrum of relevant papers. This inclusive approach aimed to consider and review papers from various aspects of cloud computing task scheduling.

By adopting this methodology, the review aims to provide a comprehensive overview of the existing literature, offering insights into the diverse research directions, methodologies, and findings in the field of cloud computing task scheduling. The structured approach ensures the inclusion of a broad range of papers and facilitates a coherent presentation of the surveyed literature.

Papers taken from

We have thoroughly searched digital library of research papers in order to obtain a wide collection of research papers related to task scheduling in cloud computing. Following digital libraries are covered:

- IEEE Xplore
- Elsevier
- ACM Digital Library

- Wiley
- Google Scholar
- Springer

2.2.1 Query Formed

This section presents various queries that were executed in order to find relevant papers. The queries were designed in a sequential manner and executed accordingly. These included:

- Task Scheduling in Cloud Computing
- Task scheduling in cloud computing environment with cost and time
- Task scheduling in cloud computing environment with cost and time with deadline and qos
- Task scheduling in cloud computing environment with cost and time with deadline and qos simulation tool
- Task scheduling in cloud computing environment with cost and time with deadline and qos simulation tool workflow and CloudSim
- Task scheduling in cloud computing environment with cost and time with deadline and qos simulation tool workflow and CloudSim after 2010
- Task scheduling in cloud computing environment with cost and time with deadline and qos simulation tool workflow and CloudSim after 2010 with heuristics
- Task scheduling in cloud computing environment with cost and time with deadline and qos simulation tool workflow and CloudSim after 2010 with meta heuristics
- Task scheduling in cloud computing environment with cost and time with deadline and qos simulation tool workflow and CloudSim after 2010 with Hybrid Algorithm

2.2.2 Paper Selection Criteria

Huge amounts of papers were obtained from digital libraries based on the formulated queries. To meet the objectives of the review, following criterias were formulated:

- The publication period is considered from 2010 to 2018, encompassing the most recent publications for comprehensive coverage.

- Only papers written in English were included, as reputable publications typically use English as the language of communication.
- In addition to journals, conferences, workshops, or peer-reviewed journals were selected for review.
- For papers with extended versions, the latest and extended version was selected to evaluate the most recent work.
- Surveys, systematic literature surveys, and review papers were excluded to focus on original research contributions.
- To avoid redundancy, duplicate or slightly different papers were carefully examined and managed.
- Extended abstracts or short papers were not considered for inclusion in this review.
- Work in progress papers were excluded to ensure the inclusion of mature and finalized research.

Despite these inclusion and exclusion criteria, a considerable number of papers remained for review. A strategic review process was established in stages:

i *Stage 1:* Initially, all 1000 papers were checked based on their titles, leading to the elimination of approximately 599 papers due to similarities or unrelated titles.

ii *Stage 2:* Subsequently, from the remaining approximately 400 papers, the abstracts and conclusions were read, resulting in a further reduction to 108 papers.

iii *Stage 3:* Finally, these 108 papers were thoroughly read and evaluated based on various criteria, leading to a further decrease in the number of papers and resulting in the final selection of 51 papers for review in this paper.

These 51 papers are considered to encompass a diverse range of objectives and approaches. The subsequent organization of this paper is detailed in Section 2.3, providing a comprehensive review of the selected papers.

2.3 Reviewed Task Scheduling Papers

This section presents the various task scheduling papers reviewed. The main classification can be seen in Fig. 2.1. Table 2.3 presents a detailed survey of all the papers with objective and strategy used as discussed in above section.

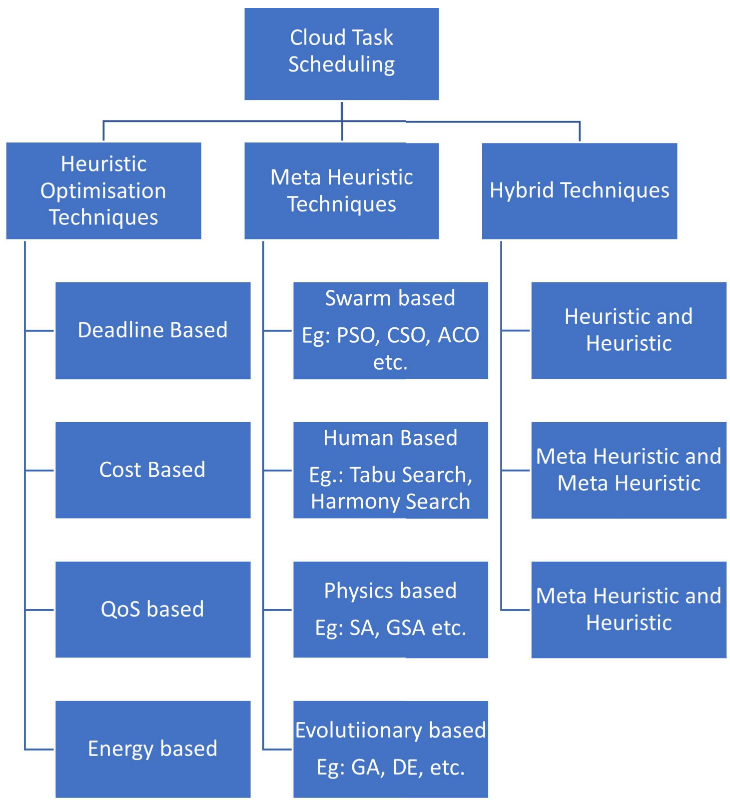

Fig. 2.1 Detailed classification of papers

No.	Title	Objective	Year	Strategy Used	Simulation Tool Used	Findings	Scheduling Type
1	Deadline-constrained workflow scheduling in software as a service Cloud[2]	minimize the total cost and to finish task before deadline	2011	QoS based algorithm based on Partial Critical Path(PCP)	Java based Simulator	Even in tight deadlines tasks are finished before deadline	Workflow
2	A Dynamic Optimization Algorithm for Task Scheduling in Cloud Environment[18]	Deadline and minimise the cost	2012	Grouping of tasks based on prioritization and then assigning	CloudSim	Improves cost and completion type of tasks as compared to Sequential allocation	Independent
3	Scheduling deadline constrained scientific workflows on dynamically provisioned cloud resource[7]	Deadline constrained with computation cost reduction in dynamically provisioned cloud environment	2017	PDC- parallelising of workflow levels to achieve desired deadline DCCP- determines the constrained critical path to co-locate tasks communicating on same instance	CloudSim	high success rates and throughput, while in most cases presenting the lowest overall pay-per-use cost DCCP mostly outperforms PDC	Workflow
4	A task scheduling algorithm based on QoS driven in Cloud Computing[81]	Load balancing with QoS	2013	Sorts tasks according to priority and then assigns according to the Minimum completion time	CloudSim	Achieves well load balancing even when VMs are increased	Independent

Continued on next page

Table 2.1 continued from previous page

No.	Title	Objective	Year	Strategy Used	Simulation Tool Used	Findings	Scheduling Type
5	Grouped tasks scheduling algorithm based on QoS in cloud computing network[5]	QoS	2016	Group tasks into various categories and then assign a category to execution by deciding the task according to minimum execution time	Java Based Simulator	Minimum execution time and latency is obtained along with QoS driven schedule	Independent
6	SABA: A security-aware and budget-aware workflow scheduling strategy in clouds[87]	Shorter makespan and security services	2015	Clustering and prioritization assigning a rank then assigning VMs to minimise cost which leads to overlapping of data movement and task exec during runtime	Java Based Simulator	Security aware and budget aware algorithm for communication intensive workflows	Workflow
7	HCOC: A cost optimization algorithm for workflow scheduling in Hybrid Cloud[14]	Cost Optimisation	2011	Which resources should be leased from the public cloud and merged to the private cloud to provide sufficient processing power to execute a workflow within a given execution time	Simulation environment was made with private and public infrastructure as hybrid infrastructure	It has the strong ability to reduce the execution cost in public cloud but can't deal with multiple workflows	Workflow
8	Multi-Objective Approach for Energy-Aware Workflow Scheduling in Cloud Computing Environments[84]	Minimise energy consumption with cost and deadline also	2013	Hybrid PSO having DVFS technique	Simulation for parallel and hybrid structure	Less makespan than HEFT algorithm	Workflow

Continued on next page

Table 2.1 continued from previous page

No.	Title	Objective	Year	Strategy Used	Simulation Tool Used	Findings	Scheduling Type
9	A three-dimensional virtual resource scheduling method for energy saving in cloud computing[92]	Reduce energy consumption on a power VM	2016	Scheduling is divided into three stages: virtual resource allocation, virtual resource scheduling and virtual resource optimization	CloudSim	improve the resources utilization and effectively reduce the energy consumption	Independent
10	EATS: Energy-Aware Tasks Scheduling in Cloud Computing Systems[30]	Increase application efficiency with minimum energy consumption	2016	distributes a Big data into N independent chunks and then the execution time is optimised	Oscilloscope	increase the speedup of an application by distributing its data and processing	Independent
11	Energy-aware scheduling of virtual machines in heterogeneous cloud computing systems[20]	Energy efficiency and resource utilisation	2017	a prediction model based on fractal mathematics and a scheduler on the basis of an improved ant colony algorithm	CloudSim	used real workload traces from the compute clusters of Google	Workflow
12	Dynamic energy-aware scheduling for parallel task-based application in cloud computing[32]	Energy-aware run-time scheduler for task-based applications	2018	apply heuristic rules for ranking according to SPT, LPT, LNS and LSTF and then apply heuristics to allocate the resources	Prototype of the cloud Infrastructure	a polynomial-time algorithm	Workflow

Continued on next page

Table 2.1 continued from previous page

No.	Title	Objective	Year	Strategy Used	Simulation Tool Used	Findings	Scheduling Type
13	Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm[72]	optimize task scheduling and resource allocation based on the proposed cost and time models	2013	IDEA combines the Taguchi method and a differential evolution algorithm	Not Specified	high effectiveness and easy optimization but only on 5 and 10 task problems	Tasks divided into subtasks and then further allocation
14	A Study of Task Scheduling Based On Differential Evolution Algorithm in Cloud Computing[82]	minimum completion time, maximum load balancing degree, and the minimum energy consumption	2014	adopted the adaptive zooming factor mutation strategy and adaptive crossover factor increasing strategy	CloudSim	Able to achieve better results for task completion time in cloud environment and not all	Independent
15	Independent Tasks Scheduling Based on Genetic Algorithm in Cloud Computing[90]	Minimise application execution time	2009	Capability Requirement vector and supply vector is used along with GA	Not tested on any cloud environment	designed to across different platforms where two tasks with distinct CRVS to be dispatched on two process units, one of which is 32-bit while the other is 64-bit	Independent

Continued on next page

Table 2.1 continued from previous page

No.	Title	Objective	Year	Strategy Used	Simulation Tool Used	Findings	Scheduling Type
16	Scheduling Using Improved Genetic Algorithm in Cloud Computing for Independent Tasks[36]	Minimise makespan	2012	Used Min-Min and Max-Min as initial population and then GA is applied	CloudSim	Used not for Cost but makespan	Independent
17	An Efficient Approach to Genetic Algorithm for Task Scheduling in Cloud Computing Environment[33]	Minimise execution cost and time	2012	Initialise the population with LCFP, SCFP and random, then apply GA	Eclipse with Java with JGAP(java Genetic algorithm package)	Used for single user but not applicable in runtime scheduling	Independent
18	HSGA: a hybrid heuristic algorithm for workflow scheduling in cloud systems[33]	To obtain the response Quickly	2013	Uses Best fit and round Robin as initial population along with prioritization and then applied GA	Not Specified	Failure frequency is same as other algorithm but speedup is achieved	Workflow
19	Genetic-Based Task Scheduling Algorithm in Cloud Computing Environment[19]	minimize the completion time and cost of tasks, and maximize resource utilization	2016	Used Tournament selection as a strategy for selecting individuals among the population.	CloudSim	Not Dynamic but achieves better speedup, cost efficiency, and resource utilisation than GA and RR	Independent

Continued on next page

Table 2.1 continued from previous page

No.	Title	Objective	Year	Strategy Used	Simulation Tool Used	Findings	Scheduling Type
20	A Tabu Search Based Heuristic for Optimized Joint Resource Allocation and Task Scheduling in Grid/Clouds	Minimise cost and improve resource utilisation[85]	2013	Perform FCFS, SJF, ESTF and SSF(Simpler Structure First) , evaluate them. Assign initial population according to best and then run Tabu search	Used for both Grid and Cloud Environment	Reduces blocking rate when initialised by SSF but not pre-emptive	Dependent and independent
21	Time-Constrained Workflow Scheduling In Cloud Environment Using Simulation Annealing Algorithm[31]	Minimise time	2013	Used basic SA for scheduling	No tool specified	Good convergence rate as compared to PSO but not evaluated on any environment	Workflow
22	Differential evolution-GSA based optimal task scheduling in cloud computing[66]	minimize finishing time	2016	Differential algorithm with Gravitational search algorithm is proposed	CloudSim	Better than basic DE, PSO and GSA	
23	Honey bee behaviour inspired load balancing of tasks in cloud computing environments[34]	Load balancing of tasks	2013	Grouping of VMs used as OVM, LVM and BVM and then trigger load balancing according to priority	CloudSim	Works well for pre-emptive independent tasks	Independent
24	Artificial Bee Colony for Workflow Scheduling[41]	Reduce the response time	2014	Applies ABC by firstly merging multiple workflows and initialising by SPT, LPT along with some random population	Borland C++ Builder with Gurobi Optimiser	Two ABC based algos are simulated and compared, one having single neighboring solution	Workflow

Continued on next page

Table 2.1 continued from previous page

No.	Title	Objective	Year	Strategy Used	Simulation Tool Used	Findings	Scheduling Type
25	Task scheduling in the Cloud Environments based on an Artificial Bee Colony Algorithm[58]	Reduce task execution time	2015	Non Pre-emptive tasks are scheduled according to Bee Algorithm	Matlab	Random costs are taken into consideration to compare	Workflow
26	Workflow Scheduling in Cloud Computing Environment Using Cat Swarm Optimization[12]	Reduce cost	2014	CSO is applied directly with zero modifications	Not specified	PSO and CSO are compared for 17 tasks	Workflow
27	A multi objective Cat Swarm Optimisation algorithm for Workflow Scheduling in Cloud Computing Environment[13]	Reduce cost, makespan and CPU idle time	2015	Applied CSO for multi-objective fitness function	Matlab	Not considered the running time of algorithm but gives better result than PSO	Workflow
28	A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments[61]	To reduce the computation and communication cost between tasks	2010	PSO is applied directly with some polling time in between iterations for two mappings to obtain	Jswarm Package	Found to be good for synthetic workflow but not evaluated on real time schedules	Workflow

Continued on next page

Table 2.1 continued from previous page

No.	Title	Objective	Year	Strategy Used	Simulation Tool Used	Findings	Scheduling Type
29	Improved PSO-based Task Scheduling Algorithm in Cloud Computing[88]	Reduce average running time and improve resource utilisation	2012	introduces simulated annealing, adds it into every iteration of PSO	CloudSim	Increases utilisation but doesn't take load balancing of tasks	Independent
30	Task scheduling using Modified PSO Algorithm in the Cloud Computing Environment[1]	Minimum completion time	2014	Initialise population by SJFP and then apply PSO	Not specified	Not tested on any cloud environment	Independent
31	Enhanced Particle Swarm Optimization For Task Scheduling In Cloud Computing Environments[9]	account reliability, execution time, transmission time, make span, round trip time, transmission cost and load balancing	2015	Run PSO, find wrongly allocated tasks or unallocated tasks, check for load and then assign again according to PSO	CloudSim	Improves reliability for a set of tasks	Independent

Continued on next page

Table 2.1 continued from previous page

No.	Title	Objective	Year	Strategy Used	Simulation Tool Used	Findings	Scheduling Type
32	Task Scheduling Using PSO Algorithm in Cloud Computing Environments[4]	Minimise makespan and increase resource utilisation	2015	amalgamation of the Dynamic PSO (DAPSO) algorithm and the Cuckoo search (CS) algorithm	CloudSim	Tested for upto 40 Cloudlets only but performs better than basic version	Independent
33	Cloud Task scheduling based on Load Balancing Ant Colony Optimization[39]	Load balancing and minimum makespan	2011	Rule of choosing next VM in ACO is changed	CloudSim	Applied on non-preemptive computational tasks	Independent
34	Ant Colony Optimization Based Service flow Scheduling with Various QoS requirements in Cloud Computing[43]	Monitor running state of Cloud Services	2011	Use QoS requirements firstly then apply ACO with 5models	Simple PC is used with Java	Four optimisation objectives are designed according to users demands	Workflow
35	Cloud Task Scheduling Based on Ant Colony Optimization[71]	Minimise makespan	2013	Pheromone updation is done according to task length submitted, exec. Time and transfer time	CloudSim	Compared with FCFS and RR for upto 1000 tasks	Independent
36	Scheduling Workflow in Cloud Computing Based on Ant Colony Optimization Algorithm[91]	Minimise the time to assign resources	2013	Two ways ant strategy as forward and backward is used. Master Node is there to determine next node to be allocated with tasks.	CloudSim	Complexity is O(k(m+n)) where K is the number of tasks ,O(m) forward ant and O(n) putting tasks	Workflow

Continued on next page

Table 2.1 continued from previous page

No.	Title	Objective	Year	Strategy Used	Simulation Tool Used	Findings	Scheduling Type
37	A Multi-Objective Optimization Scheduling Method Based on the Ant Colony Algorithm in Cloud Computing[93]	Minimise makespan and cost	2015	ACO is applied by modifying pheromone update according to start time and optimal path	CloudSim	Complexity is O(KN) where K Tasks and N Resources	Independent
38	Critical Path-Based Ant Colony Optimization for Scientific Workflow Scheduling in Cloud Computing Under Deadline Constraint[38]	Reduce cost and time	2018	Used ACO with Pareto distribution initialisation for under utilised VMs	Workflow Simulator	ACO takes more response time as compared to other	Workflow
39	Cost-Optimal Scheduling in Hybrid IaaS Clouds for Deadline Constrained Workloads[74]	Maximise utilization of datacenter and minimise the cost of running with QoS	2010	Introduced a binary integer program to obtain the objective	Language: AMPL and Programming solver: CPLEX	Synthetic data is used for Hybrid cloud	Independent

Continued on next page

Table 2.1 continued from previous page

No.	Title	Objective	Year	Strategy Used	Simulation Tool Used	Findings	Scheduling Type
40	Auto-Scaling to Minimize Cost and Meet Application Deadlines in Cloud Workflows[49]	Both performance requirements and minimise cost within deadline	2011	Calculate VMs needed for workload assign VMs according to EDF; If needed consolidate extra VMs	Cloud Workload	Unlimited budget is assumed and minimisation of cost is done	Workflow
41	Cost-Efficient Scheduling Heuristics for Deadline Constrained Workloads on Hybrid Clouds[75]	Minimise cost and deadline effective with QoS constraints	2011	Use exact wait time queue but user's don't provide exact time so estimated runtime is used	Java based discrete time simulator	The average error for runtime estimation is upto 50%	
42	Online cost-efficient scheduling of deadline-constrained workloads on hybrid clouds[76]	Minimise cost, deadline effective and data requirements	2013	Two scheduling types for public and hybrid cloud and decide if the requirements can be met	Java-based discrete time simulator	an EDF scheduling policy significantly increase robustness with respect to runtime estimation errors, at an additional cost in turnaround time	Workflow

Continued on next page

Table 2.1 continued from previous page

No.	Title	Objective	Year	Strategy Used	Simulation Tool Used	Findings	Scheduling Type
43	A security and cost aware scheduling algorithm for heterogeneous tasks of scientific workflow in clouds[40]	Dynamic resource scaling with security and cost constraint	2015	Based on PSO but considered security and cost as priority	CloudSim	An offline schedule is generated for a workflow with high time complexity	Workflow
44	Genetic Simulated Annealing Algorithm for Task Scheduling based on Cloud Computing Environment[23]	Smaller completion time, cost, maximise bandwidth, reliability	2010	genetic simulated annealing algorithm in cloud computing with QoS parameters	Not specified	Algorithm converges faster but not applied on benchmark workflows	Independent
45	Scheduling Workflow in Cloud Computing Based on Hybrid Particle Swarm Algorithm[83]	Minimum cost within a given time	2012	crossover and mutation of genetic algorithm is embedded into PSO, also hill climbing problem is introduced to remove local search problem	Java based Simulator	Improves global search optimisation as compared to basic PSO	Workflow
46	A task scheduling algorithm based on genetic algorithm and ant colony optimization in cloud computing[42]	Minimise task execution time and convergence rate	2014	Combined GA with ACO as initial pheromone value upto certain iteration	IM Simulator	Not much improvement than ACO and GA basic versions	Independent

Continued on next page

Table 2.1 continued from previous page

No.	Title	Objective	Year	Strategy Used	Simulation Tool Used	Findings	Scheduling Type
47	Workflow Scheduling Using Hybrid GA-PSO Algorithm in Cloud Computing[48]	Minimise makespan and cost with load balancing of tasks	2018	GA and PSO are run for half-half iterations	CloudSim	Compared for only montage workflow but is better than PSO and GA	Workflow
48	A parallel bi-objective hybrid meta-heuristic for energy-aware scheduling for cloud computing systems[55]	Minimise makespan with energy consumption	2011	Used dynamic voltage scaling for energy consumption with GA with parallel island model	ParadisEO (tools for the design of parallel meta-heuristics for multi-objective optimization)	Choose between three approaches ECS, hybrid and insular approach	Workflow
49	Self-Adaptive Learning PSO-Based Deadline Constrained Task Scheduling for Hybrid IaaS Cloud[94]	maximize the profit of IaaS provider while guaranteeing QoS	2014	four updating strategies are used to adaptively update the velocity of each particle to ensure diversity	Matlab	an integer programming model is established in a hybrid cloud environment.	Independent

Continued on next page

Table 2.1 continued from previous page

No.	Title	Objective	Year	Strategy Used	Simulation Tool Used	Findings	Scheduling Type
50	Deadline and Cost Based Ant Colony Optimization Algorithm for Scheduling Workflow Applications in Hybrid Cloud[68]	Minimise cost while meeting deadline in hybrid cloud	2014	Find schedule in private cloud first, if deadline not met then migrate VMs to public cloud	CloudSim	five Data Centres were interconnected and one of them was private cloud and others were public.	Workflow
51	Deadline constraint heuristic-based genetic algorithm for workflow scheduling in cloud[78]	Task scheduling and resource allocation which meets deadline too	2014	assign priority using bottom-level (b-level) and top-level (t-level) to increase the population diversity, these priorities are then used to create the initial population of HGAs.	Java based simulator	Synthetic workflows are used with different size and structures	Workflow

2.4 Heuristic Optimisation Techniques

This section provides detailed insights into papers that focus on heuristic techniques for optimization in task scheduling. Heuristic techniques are problem-specific algorithms applicable to particular problem domains. Therefore, the classification of these papers is organized as follows:

2.4.1 Deadline Based

The primary objective of the papers reviewed in this category is to efficiently assign tasks to Virtual Machines (VMs) before their specified deadlines. The focus in most of these papers extends beyond deadlines and incorporates the cost associated with running tasks on allocated processors. The following papers delve into various techniques to achieve this objective:

i **Paper [2]:** - *Algorithm Implemented:* Critical path-based algorithm. - *Simulation Environment:* Java Based Simulator. - *Advantage:* Tasks are scheduled before the specified deadline, even in scenarios with tighter deadlines.

ii **Paper [18]:** - *Algorithm Implemented:* Task grouping based on priority, followed by task assignment. - *Prioritization Criteria:* Set based on cost and deadlines. - *Comparison:* Algorithm compared with RoundRobin and FCFS algorithms.

iii **Paper [7]:** - *Algorithms Implemented:* PDC (Priority-Driven Scheduling with Cost-Time Trade-off) and DCCP (Deadline Constrained Critical Path). - *PDC Details:* Prioritization of tasks with parallelization to minimize dependencies, implemented at two levels (Bottom and Top). - *DCCP Details:* Backfilling strategy, with variations in ranks and scheduling modifications for improved efficiency.

2.4.2 QoS Based

In the realm of cloud computing, Quality of Service (QoS) plays a pivotal role in ensuring better reliability, throughput, and overall service quality. This category encompasses papers that primarily focus on evaluating and enhancing QoS provided by Cloud Service Providers:

i **Paper [81]:** - *Algorithm Design:* Prioritization based on User Type, Expected Priority of a Task, and Task Length in a normalized manner. - *Comparison:* Evaluation against Min-Min algorithm and Berger Model.

ii **Paper [5]:** - *Algorithm Design:* Similar prioritization basis as [81], but with adjusted priorities for enhanced performance. - *Comparison:* Demonstrated improvement over the algorithm presented in [81].

2.4.3 Cost Based

This category focuses on algorithms where the primary objective is minimizing cost in task scheduling. While many papers in this category combine cost with other objectives, the algorithm presented in [14] stands out for its specific emphasis on cost while considering the makespan of the schedule:

i **Paper [14]:** - *Algorithm Design:* Prioritizes cost as the first objective, with consideration for the makespan of the schedule.

2.4.4 Energy Based

This category highlights research endeavors aimed at achieving task schedules with the minimal energy consumption. Papers in this category may have energy as the primary objective, potentially alongside other goals. The techniques employed in [20, 30, 32, 84, 92] papers can vary, but the overarching objective is consistently focused on energy efficiency.

2.5 Meta-Heuristics Techniques

This category explores research employing meta-heuristic techniques for task scheduling. Meta-heuristic techniques are utilized to select algorithms that can offer good solutions, although they do not guarantee optimality. The strength of these techniques lies in their ability to converge and provide improved solutions.

2.5.1 Evolutionary Algorithms

This category delves into research employing evolutionary algorithms, which leverage biological evolution methods such as reproduction, mutation, combination, crossover, etc. These algorithms utilize a fitness function to assess the quality of solutions and evolve populations through these operators. Within this category, papers related to Genetic Algorithm (GA) and Differential Evolution (DE) are classified, as both utilize various operators to generate new populations.

i **Paper [72]:** - *Algorithm:* IDEA (Integrated Differential Evolution Algorithm) combining Taguchi method with DE. - *Features:* Well-balanced exploration and exploitation, uses Pareto-optimal front for makespan and cost solutions, employs Iterated Function Systems (IFS) with Taguchi method.

ii **Paper [82]:** - *Algorithm Features:* Adaptive zooming factor mutation strategy and adaptive crossover increasing strategy. - *Objectives:* Maximum load balancing and minimum completion time.

iii **Papers [90], [36], [33], [19], [29]:** - *Main Algorithm:* Genetic Algorithm (GA). - *Variations:* Changes in how the initial population is provided to the GA. - Paper [36]: Uses Min-Min for initial population. - Paper [33]: Uses SCFP and LCFP (Shortest Cloudlet to Fastest Processor, Longest Cloudlet to Fastest Processor) as initial population or best fit strategy. - Paper [19]: Implements a best fit strategy. - Paper [29]: Utilizes Tournament Selection strategy.

2.5.2 Human Based

This category explores algorithms inspired by human behavior, incorporating search strategies similar to how humans would approach problem-solving. Examples include Harmony Search, Tabu Search, among others. In [85], the Tabu Search algorithm is employed to discover optimal schedules. Tabu Search utilizes a local search approach based on human search methods, but with certain modifications to enhance performance. Notably, worsening moves are accepted only if no improved move is apparent, and prohibitions are introduced to restrict certain solutions.

2.5.3 Physics Based

Physics based category consists of those which have concepts based on Physics. Simulated annealing, Gravitational search algorithm(GSA) are some examples. In [31], used basic Simulated Annealing algorithm for scheduling the tasks and results show that it is better than PSO but the authors have not used any environment to evaluate the same. In next paper [66], Differential evolution algorithm is used alongwith the Gravitational search algorithm to map the tasks to the Virtual Machines(VMs) and results show that it is better than DE, PSO and GA.

2.5.4 Swarm Based

This category explores algorithms inspired by the collective behavior of swarms, such as groups of flying insects, birds, and other organisms. The behavior observed in these swarms serves as the foundation for optimization techniques applied in the Cloud Computing environment to obtain optimized schedules. The following papers, denoted as [34], [41], ..., [38], are all based on one or another swarm-based optimization algorithm.

i **Papers [34], [41], [58]:** - *Algorithm Basis:* Honey Bee Behavior. - *Features:* - Paper [34]: Load balancing for preemptive tasks. - Paper [41]: Merging workflows and applying Artificial Bee Colony (ABC) algorithm. - Paper [58]: ABC used to reduce execution time.

ii **Papers [12], [13]:** - *Algorithm Basis:* Cat Swarm Optimization (CSO). - *Features:* - Paper [12]: Basic CSO applied to reduce cost. - Paper [13]: Multi-objective fitness function for a workflow.

iii **Papers [1, 4, 9, 61, 88]:** - *Algorithm Basis:* Particle Swarm Optimization (PSO).

2.6 Hybrid Techniques

In this category, we have classified those papers which uses a combination of both the above categorised objectives and techniques. The classification can be well understood by the following classification.

2.6.1 Heuristic and Heuristic Techniques Combination

This category includes papers that address more than two objectives, combining heuristic techniques with different strategies. The details of each paper are presented below:

i **Paper [74]:** - *Objectives:* Utilization, cost, and QoS in a hybrid IaaS cloud. - *Techniques:* Binary Integer Program, AMPL, IBM Optimization Studio (CPLEX). - *Details:* Considers a hybrid cloud structure (both private and public cloud properties), introduces a Binary Integer Program, and utilizes AMPL with CPLEX as a programming solver for robustness.

ii **Paper [49]:** - *Objectives:* Performance requirements, deadline, and cost in cloud workflows. - *Techniques:* Auto-scaling, EDF (Earliest Deadline First) scheduling. - *Details:* Focuses on meeting performance requirements with deadline and cost considerations, assumes an unlimited budget, and applies workload assignment based on EDF followed by reconsideration and consolidation of overloaded VMs.

iii **Paper [75]:** - *Objectives:* Deadline, cost, and QoS constraints in hybrid clouds. - *Techniques:* Heuristics considering transfer and computational costs. - *Details:* Incorporates both public and private clouds, employs heuristics considering both transfer and computational costs, and addresses deadline, cost, and QoS constraints.

iv **Paper [76]:** - *Objectives:* Minimizing cost, meeting deadlines, and data requirements in hybrid clouds. - *Techniques:* Online scheduling, EDF scheduling strategy. - *Details:* Focuses on cost-efficient scheduling with deadline constraints, utilizes different scheduling types for public and hybrid clouds, employs an EDF scheduling strategy, and uses a Java-based simulator for evaluation.

2.6.2 Meta-Heuristic and Meta-Heuristic Techniques Combination

This category includes papers that employ a combination of two or more meta-heuristic algorithms to achieve their scheduling objectives. The details of each paper are outlined below:

i **Paper [23]:** - *Objectives:* Deadline, cost, bandwidth, and reliability. - *Algorithms:* Genetic Algorithm (GA) + Simulated Annealing (SA). - *Details:* Combines GA with SA for mapping tasks to VMs, faster convergence observed, applied on synthetic data.

ii **Paper [83]:** - *Objectives:* Minimizing cost within a given time for a workflow. - *Algorithms:* Genetic Algorithm (GA) components embedded in Particle Swarm Optimization (PSO) + Hill Climbing. - *Details:* Embeds GA components in PSO, incorporates Hill Climbing to address local search problems, improves global search optimization compared to basic PSO.

iii **Paper [42]:** - *Objectives:* Minimizing task execution time and convergence rate. - *Algorithms:* Genetic Algorithm (GA) + Ant Colony Optimization (ACO). - *Details:* Combines GA with ACO, uses ACO's pheromone value as the initial value for GA population, tasks are independent, simulated on IM simulator.

iv **Paper [48]:** - *Objectives:* Minimising Cost, time with load balancing. - *Algorithms:* Hybrid Genetic Algorithm (GA) + Particle Swarm Optimization (PSO). - *Details:* Divides total iterations into two halves, dedicates the first half to GA and the second half to PSO, GA uses tournament selection, PSO uses the final result of GA as the initial population.

2.6.3 Heuristic and Meta-Heuristic Techniques Combination

In cloud computing task scheduling, the combination of heuristic and meta-heuristic techniques has emerged as a powerful approach to address multiple optimization objectives. This section explores papers that integrate both heuristic and meta-heuristic algorithms to leverage their respective strengths in achieving efficient and effective task scheduling. The hybridization of these techniques aims to enhance the convergence and solution quality, providing a more robust and versatile approach to handle complex scheduling challenges. The following papers present innovative strategies that synergize heuristic and meta-heuristic algorithms, demonstrating their effectiveness in addressing various scheduling objectives.

i **Paper [40]:** - *Objectives:* Dynamic resource scaling, security, and cost-efficient scheduling. - *Algorithms:* Particle Swarm Optimization (PSO) based. - *Details:* Implements a PSO-based algorithm for offline scheduling with high time complexity, focuses on security and minimizing schedule cost, implemented in CloudSim.

ii **Paper [55]:** - *Objectives:* Minimizing makespan and energy consumption. - *Algorithms:* Parallel Island Model with Dynamic Voltage Scaling (DVS). - *Details:* Uses ParadiseEO simulation tool, introduces three approaches (ECS, hybrid, insular) and selects one for workflow scheduling.

iii **Paper [94]:** - *Objectives:* Maximizing IaaS provider profit while guaranteeing QoS. - *Algorithms:* Integer Programming Model in a hybrid cloud environment. - *Details:* Incorporates four updating strategies for diversity among schedules, uses velocity update policy, hybrid cloud environment, and integer programming model.

iv **Paper [68]:** - *Objectives:* Considering deadline and cost in a hybrid cloud environment. - *Algorithms:* Strategy involving migrating VMs to public cloud if private cloud deadline is not met. - *Details:* Utilizes a hybrid cloud with 5 datacenters (4 public, 1 private), migrates VMs to public cloud if private cloud deadline is not met.

v **Paper [78]:** - *Objectives:* Allocating resources meeting the deadline. - *Algorithms:* Hybrid Genetic Algorithm (GA) with deadline constraint heuristic. - *Details:* Assigns priority based on top and bottom levels to increase diversity in the initial population of Hybrid GAs, evaluated on synthetic workflows using a Java-based simulator.

2.7 State Of The Art

This section list out some new research papers in the area of cloud task scheduling.

2.7.1 Resource Allocation Algorithms

Distributed resources are an important aspect of cloud computing, and scheduling protocols play an important role in this area. In a paper, task allocation has been done considering service reliability in the grid computing environment based on social spider optimization technique. The authors also formulated formulae for transactions in the environment keeping in mind the resource availability[46].

In another paper, the cloudlets are assigned to different hosts according to the Shortest Deadline First combined with First Come First Serve where the Cloudlets are being submitted at different arrival times. The deadline provided by the user is used to determine whether the Cloudlet can finish execution within the prescribed time interval or not[37].

In the paper [51], authors have developed an energy-efficient and reliability aware workflow task scheduling in a cloud environment (EERS) algorithm using task rank calculation, task clustering and sub-target time distribution algorithms for reducing energy and provide maximum reliability. Shang[65] in his paper introduces a dynamic resource allocation algorithm for huge-scale scientific computations in cloud computing. It employs a directed acyclic graph to symbolize task dependencies, applies fuzzy clustering to group nodes based totally on diverse features, and optimizes subtask allocation across clusters. The algorithm complements useful resource allocation, load balancing, and decreases completion time and cost via feedback-driven resource reputation.

The paper[80] presents an optimal computing resource allocation algorithm for cloud computing using hybrid differential parallel scheduling. It constructs data and grid structures, employs sample clustering analysis to classify resource attributes, and uses singular price decomposition for allocation. The approach demonstrates high clustering performance, quick allocation with a 3.67 speedup, and reduced overhead, making it practical for efficient cloud resource allocation.

Resource allocation is crucial in cloud computing, aiming to optimize resource utilization, provider profit, and user satisfaction. This paper by Vakilifard et. al.[73] introduces a fuzzy logic-based algorithm enhanced with meta-heuristic methods, particularly particle swarm optimization (PSO), to fine-tune fuzzy inference system parameters. The fuzzy-PSO approach significantly improves resource efficiency by responding to more requests and increasing utilization compared to the fuzzy-only algorithm. The paper by Mahboubeh Afzali et. al.[3] proposes an optimized approach using the improved binary particle swarm optimization (IBPSO) algorithm, significantly reducing latency and improving load balancing compared to other methods. The IBPSO-based approach outperforms binary genetic, binary PSO, binary grey wolf, and ranked-based resource allocation methods by approximately 11%, 22%, 21%, and 22%, respectively, in terms of various performance metrics. Additionally, it reduces total latency by approximately 11%, 28%, 27%, and 25% compared to other methods and enhances runtime by approximately 45%, 9%, and 8% compared to binary genetic, binary PSO, and binary grey wolf-based resource allocation methods.

2.7.2 Multiobjective Scheduling

Multi-objective scheduling are popular as cloud services become more diverse and complex. The paper by Maha Zeedan et. al. [86] introduces the Enhanced Binary Artificial

Bee Colony based Pareto Front (EBABC-PF) approach for scheduling workflow applications in cloud computing, considering diverse Quality of Service (QoS) requirements. It minimizes makespan and processing costs while maximizing resource usage without violating SLAs. The method combines HEFT, GRASP, and modified BABC, outperforming HEFT, DHEFT, NSGA-II, and standard BABC in terms of makespan, processing cost, and resource utilization across various task sizes and benchmark workflows. In a paper titled, "Multi-Objective Load Balancing in Cloud Computing: A Meta-Heuristic Approach"[35], workload balancing challenges has been addressed in cloud computing, particularly in Infrastructure as a Service (IaaS). It introduces a novel multi-objective load balancing framework, considering factors like power usage, bandwidth, migration costs, memory usage, and various load balancing parameters. The proposed Mouse Customized Golden Eagle Optimization (MCGEO) model enhances convergence and achieved superior throughput in a cloud environment, demonstrating its effectiveness.

In another study[25], the authors introduce a hybrid multi-objective meta-heuristic optimization-based load balancing model for cloud computing. It assigns virtual machines to physical machines based on criteria like memory usage, migration costs, power usage, and load balancing settings. The approach utilizes a novel hybrid optimization technique called Dingo Customized Cat Mouse Optimization (DCCO), achieving significant improvements in server load compared to other methods. The DCCO model is evaluated across various performance metrics in a cloud environment.

2.7.3 Meta Heuristics Based Approaches

Many papers use different meta heuristics algorithms to solve their problems such as [46], [44], [47], [45] etc. In a paper titled, "Energy-aware workflow task scheduling in clouds with virtual machine consolidation using discrete water wave optimization"[52]. The article introduces EASVMC, an energy-efficient algorithm for scheduling scientific workflows in cloud computing with VM consolidation. EASVMC addresses energy consumption, resource utilization, and VM migrations by using a two-phase approach. In the first phase, tasks are mapped to VMs for minimized energy usage. The second phase employs Water Wave Optimization (WWO) to consolidate VMs on hosts based on CPU utilization, enhancing resource efficiency and lowering energy consumption. Experimental results using real-world scientific workloads demonstrate EASVMC's better performance over existing methods in achieving stated objectives. In a paper by Rambabu Medara et. al[50], the paper focuses on an energy-efficient algorithm for VM consolidation, utilizing live migrations to optimize server energy usage. The approach adapts the Water Wave Optimization (WWO) meta-heuristic algorithm to solve this NP-hard problem, aiming to minimize energy consumption and maximize server shutdowns effectively. Also, in another paper by same author[53], the article introduces an energy and cost-aware scheduling (ECWS) approach using a heterogeneous earliest finish time (HEFT) based heuristic. ECWS aims to reduce energy use, execution cost, and maximize resource utilization in the cloud data center. The algorithm's effectiveness is demonstrated through experiments using WorkflowSim, showcasing its achievements in energy conservation, resource utilization, and cost savings compared to existing algorithms.

2.7.4 Whale Optimization Based Scheduling

The Whale Optimization Algorithm (WOA), inspired by humpback whale social behavior, particularly their bubble-net hunting strategy. WOA's effectiveness is demonstrated through testing on 29 mathematical optimization problems and 6 structural design problems, showcasing its competitiveness against state-of-the-art meta-heuristic algorithms and conventional methods[56].

The whale optimization algorithm (WOA) has shown good results as an optimization method with many applications. In the context of cloud task scheduling, Cloud computing's dynamic nature poses challenges in efficient virtual machine allocation (VMA). The paper by Ankita Srivastava et. al. introduces a binary whale optimization approach to minimize resource wastage and active server count. Compared to established VMA algorithms, it achieved impressive results with resource wastage fitness at 15.68, minimum active servers at 216, and effective CPU and memory utilization of 88.31% and 88.79%, respectively[69]. In paper, "Whale Optimization-Based Task Offloading Technique in Integrated Cloud Fog Environment" [67], the authors introduced a whale optimization approach to make runtime offloading decisions, outperforming existing methods in simulations.

Virtualization is a fundamental technology in cloud computing, but it can pose security challenges. Another paper by Bhavana Gupta et. al.[27] presents a novel approach to detect attacks in virtualized cloud environments by extracting and analyzing network data. It uses Python's multi-process framework for efficient data processing and employs features obtained from mutual information. Detection is enhanced using an optimized Deep Belief Network (DBN) with a whale optimization algorithm. The proposed model's performance surpasses traditional methods like SVM, naïve Bayes, and k-nearest neighbors across multiple evaluation criteria.

2.7.5 Hybrid Scheduling Approaches

Hybrid scheduling methods utilize the potential of several different optimization techniques to improve scheduling performance. In the paper, "An Improved Hybrid Swarm Intelligence for Scheduling IoT Application Tasks in the Cloud"[8], the authors introduce MRFOSSA, a hybrid swarm intelligence method combining Manta ray foraging optimization (MRFO) and salp swarm algorithm (SSA) for IoT task scheduling. MRFOSSA enhances convergence and outperforms other metaheuristic techniques in terms of makespan time and cloud throughput, as demonstrated through experiments on various datasets. The paper "Cost optimized Hybrid Genetic-Gravitational Search Algorithm for load scheduling in Cloud Computing"[17] introduces the Hybrid Genetic-Gravitational Search Algorithm (HG-GSA) for load scheduling, aiming to reduce computation costs. HG-GSA, combining genetic and gravitational search techniques, outperforms existing methods like PSO, Cloudy-GSA, and LIGSA-C in reducing total computation costs, as demonstrated through convergence and statistical analysis in the CloudSim simulator.

Cloud computing technology has evolved significantly, highlighting the importance of scientific workflow scheduling, the paper "A hybrid genetic algorithm for scientific workflow scheduling in cloud environment"[10] presents a hybrid approach using genetic algorithms to optimize task allocation in cloud-based scientific workflows. It incorporates the Heterogeneous Earliest Finish Time (HEFT) heuristic for initial population generation and outperforms existing strategies, demonstrating efficiency and potential applicability in cloud workflow scheduling. Additionally, a GA-based module was integrated into the WorkflowSim framework based on CloudSim for practical implementation. The Table 2.2 gives a summary of the papers discussed above.

Category	Algorithm	Key Features
Resource Allocation	Social Spider, SDF+FCFS, EERS	- Service reliability consideration - Energy-efficient workflow task scheduling - Dynamic resource allocation for scientific computations - Optimal computing resource allocation using hybrid differential parallel scheduling
Whale Optimization-Based	Binary Whale, Whale for Task Offloading	- Minimizes resource wastage and active server count - Effective CPU and memory utilization - Optimized detection of attacks in cloud environments
Multiobjective Scheduling	EBABC-PF, MCGEO, DCCO	- Considers diverse QoS requirements - Enhances convergence and achieves superior throughput - Achieves significant improvements in server load - Utilizes a hybrid multi-objective optimization-based load balancing model

Continued on next page

Table 2.2 continued from previous page

Category	**Algorithm**	**Key Features**
Meta-heuristics-Based	Discrete Water Wave, EASVMC	- Energy-efficient VM consolidation using Water Wave Optimization - Energy and cost-aware scheduling using HEFT-based heuristic
Hybrid Scheduling Approaches	MRFOSSA, HG-GSA, Hybrid GA	- Hybrid swarm intelligence for IoT task scheduling - Hybrid genetic-gravitational search algorithm for cost-optimized load scheduling - Hybrid approach using genetic algorithms for scientific workflow scheduling

Table 2.2 Summary of state of the art

2.8 Research Gaps

After the state of the art survey, the research gaps that were identified are as follows:

i Despite the exploration of heuristic and meta heuristic optimization techniques for various QoS objectives such as deadline, cost, and load, there is a research gap in seamlessly integrating multiple QoS objectives.

ii The application of the Jaya algorithm is unexplored in the task scheduling challenge of cloud computing.

iii One can focus on developing holistic approaches that consider the interplay and trade-offs between diverse QoS metrics.

iv Many optimization techniques demonstrate effectiveness in small to medium-scale scenarios. However, their scalability to large-scale cloud environments remains a challenge.

v The consideration of user preferences in multi-objective optimization can also be considered.

vi Defining optimal methods for hybridizing different algorithms is a critical research gap.

vii Investigating how hybrid algorithms can dynamically adjust their strategies in response to variations in workload, resource availability, and environmental conditions is a significant research gap.

2.9 Research Objectives

Based on the above research gaps, following Research Objectives have been identified and addressed in this thesis:

i **Cost and Time-Based Scheduling Strategies:**

- *Objective:* Explore and develop cost and time-based task scheduling strategies using the Jaya algorithm, focusing on user-assigned weights for customizable solutions.
- *Approach:* Investigate the impact of different weight assignments on the Jaya algorithm's performance in minimizing makespan and execution cost. Provide user-centric scheduling solutions with adaptable weight considerations.

ii **Multi-objective Task Scheduling:**

- *Objective:* Develop and enhance multi-objective task scheduling algorithms that strike a balance between conflicting goals, such as cost minimization and time efficiency alongwith load balancing.
- *Approach:* Design algorithms that consider both dependent and independent tasks, optimizing makespan and execution cost simultaneously. Evaluate the performance against established algorithms through comprehensive simulations and experiments.

iii **Hybrid Approaches for Cloud Task Scheduling:**

- *Objective:* Investigate the potential of hybrid approaches for achieving superior task scheduling outcomes.
- *Approach:* Combine the strengths of different optimization techniques and assess the synergistic effects of hybrid approaches. Evaluate their performance in terms of convergence speed, solution quality, and adaptability.

iv **Performance Evaluation and Comparative Studies:**

- *Objective:* Conduct rigorous performance evaluations and comparative studies of the proposed algorithms against existing state-of-the-art scheduling approaches.
- *Approach:* Utilize benchmark functions, diverse workflows (Montage, CyberShake, Inspiral, SIPHT), and simulation tools (CloudSim, WorkflowSim) to assess the efficiency, scalability, and adaptability of the algorithms. Provide insights into their real-world applicability.