Survey on Task Scheduling Methods in Cloud RPS System

Henning Titi Ciptaningtyas
Information Technology Department,
Faculty of Electrical and Intelligent Information Technology (FTEIC),
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
henning@if.its.ac.id

Ary Mazharuddin Shiddiqi
Informatics Department,
Faculty of Electrical and Intelligent Information Technology (FTEIC),
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
ary.shiddiqi@if.its.ac.id

Diana Purwitasari
Informatics Department,
Faculty of Electrical and Intelligent Information Technology (FTEIC),
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
diana@if.its.ac.id

***Abstract*— The cloud platform is a popular technology because of its prominent feature, scalability, which allows users to easily adjust the usage of cloud resources. Cloud Service Provider (CSP) uses cloud provisioning methods to provide a sufficient amount of virtual resources the user needs on time, then uses a task scheduling algorithm to process the workload within the deadline or budget constraint. Ineffective scheduling may lead to underutilization (which wastage of cloud resources) or overutilization (which degrades cloud performance). This paper presents the optimization metric and comparison of task scheduling algorithms in the Cloud Resource Provisioning with Scheduling (Cloud RPS) system. We recommend using the hybrid algorithm to achieve the desired objective based on the algorithm that has been compared in this paper.**

Keywords—Task Scheduling, Cloud Computing, Hybrid algorithm, Cloud Provisioning.

I. INTRODUCTION

The cloud platform is a popular technology among businesses to host their applications and an attractive topic to be studied by researchers. Cloud Service Provider (CSP) that owns physical machines (PM) provides cloud services, cloud resources, and computing infrastructure. CSP used Virtual Machine (VM) to offer cloud services to cloud consumers [1]. Each VM has different physical resources, e.g., CPU, memory, bandwidth, and disk storage [2]. Cloud consumers can easily adjust the necessary resources on their application since cloud scalability is one of the most prominent cloud platform features. Hence, cloud consumer only needs to focus on their primary business process without considering the application's infrastructure and only pay for the cloud resources which has been used (pay-as-you-go policy) [3] [4].

The cloud consumer wants the application/tasks to run within an acceptable deadline and pay a reasonable price. At the same time, the CSP wants to maximize cloud resource utilization and profits [5]. Fulfilling those perspectives and ensuring cloud scalability features (tasks submitted to cloud platforms may vary from time to time) become an interesting topic for researchers: Cloud Resource Management.

Cloud resources should be appropriately managed by grouping them into Cloud Provisioning Paradigm and the Resource Scheduling paradigm.

The former paradigm is responsible for:

- Estimating upcoming workload/demand scale
- Allocating virtual resources to users (mapping workload to VM)
- Ensuring Service Level Agreement (SLA) negotiation and user needs are fulfilled based on QoS requirements.

The latter paradigm is responsible for:

- Ensuring QoS constraint fulfillment
- Selecting optimal VM to execute tasks [6].

Cloud Resource Provisioning with Scheduling (Cloud RPS) system ensures CSP uses an efficient provisioning technique, so the submitted workload can be analyzed and scheduled efficiently. After the cloud resource provisioning is complete, the task scheduling algorithm is executed to process the incoming workload within the deadline/budget constraints.

Task Scheduling is a technique to create a completion order of tasks or activities. Cloud resource is mapped to the appropriate task, then executed on the cloud at the scheduled time. Task Scheduling is NP (nondeterministic polynomial time) hard problem since it has a lot of solution spaces and needs some time to obtain the optimal solution [7]. The Cloud resource scheduler is called Cloud Broker. Task scheduling objectives are minimizing execution time and maximizing resource utilization. The scheduling algorithm needs to be efficient and able to balance the workload in the cloud system [8].

Based on the geographical location of cloud resources, task scheduling on cloud computing can be divided into two groups: Distributed Scheduling and Centralized Scheduling. In Distributed Scheduling, the task is assigned to different cloud resources located in different geographical places. In Centralised Scheduling, all cloud resources are in the same geographical location [9]. Distributed Scheduling will be discussed in this paper.

The main contribution of this paper is to present task scheduling algorithms for each task scheduling method (heuristic, metaheuristic, and hybrid) and its corresponding optimization metric. The algorithms presented in this paper review are selected from papers published on Sciencedirect and IEEExplore with the keyword "cloud task scheduling".

Other survey articles [5] [6] [10] discuss the metaheuristic task scheduling method in-depth, but the latest task algorithm is published in 2019, while in this paper the algorithms selected are published in 2019 – 2022 and cover all 3 task scheduling methods.

The remaining of this article discusses the optimization metric of task scheduling in Section 2, task scheduling taxonomy, algorithm, and its corresponding optimization metric in Section 3. Section 4 provides a discussion of the task scheduling method, and finally, Section 5 presents the concluding remarks.

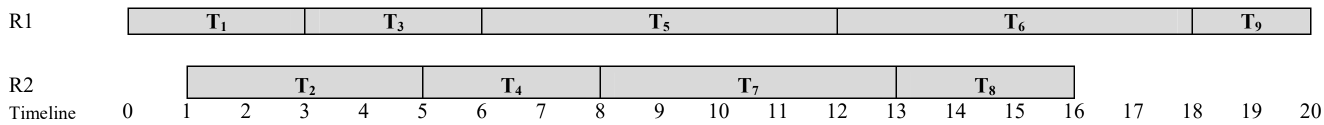

Fig. 1. Example of task scheduling on different cloud resources

TABLE I. OPTIMIZATION METRIC FOR CLOUD TASK SCHEDULING

Criteria (Abbreviation)	***Definition & Objective***	***Function & Example Based on Fig.1***
Makespan (Ms)	Finishing time of the last task. Goal: Minimize makespan. The faster the application execution time, the better.	$Makespan = max_{i \in tasks}\{F_i\}$ Eg: Makespan = 20 time units → last task (T_9) completion
Economic Cost (EC)	Total Consumer Cloud payments to Cloud Service Provider for cloud resource utilization Goal: Lowest payment for task completion	$Economic\ Cost = \sum_{i \in resources} \{C_i * T_i\}$ Eg: Economic Cost = 2,000*20 + 3,000*15 = 85,000
Flowtime (Ft)	Total time needed to finish all tasks. Goal: Minimize flowtime by executing tasks in ascending order to reduce the average response time of the schedule	$Flowtime = \sum_{i \in tasks} F_i$ Eg: Flowtime = Finishing time of ($T_1 + T_2 + \cdots + T_9$) = 3+4+6+7+12+18+12+15+20 =90 time units
Tardiness (Td)	Represent task execution delay (time elapsed between the deadline and task finishing time). Goal: Minimize tardiness, which is 0 time unit (task finishing time not missed the deadline)	$Tardiness_i = F_i - D_i$ Eg: If the deadline of $T_3$is 5, then Tardiness = 6 – 5 = 1 unit time
Waiting Time (WT)	The difference in start time execution and task submission time. Goal: Minimum waiting time, which is 0 time unit (the task is executed right after it is submitted)	$W_i = S_i - B_i$ Eg: If submission time of T_3 is 0, then Waiting Time for T_3 = 3 – 0 = 3 unit time
Turnaround Time (TaT)	Sum of waiting time and execution time for each task. It keeps track of time needed to finished the task since task submission. Goal: Turnaround time is equal to execution time	$Turnaround_i = W_i + E_i$ Eg: Turnaround for T_1 = 0 + 3 = 3 unit time
Fairness (F)	Every task should have equal share of resource utilization. Goal: No starved task	-
Resource Utilization (RU)	Keeping resource always occupied as busy as possible. Goal: Maximize resource utilization, so only limited number resource need to be rented	$Average\ Resource\ Utilization = \frac{\sum_{i=1}^{n} Resource\ i\ finishing\ time}{Makespan * n}$ Eg: Average Resource Utilization = (20+15) / (20*2) = 0.875
Throughput (Tg)	Total number of job completing execution per time unit. Goal: Maximize throughput based on single / multi criteria, such as cost execution time	-
Priority Constraint (PC)	Sequence of task based on task urgency to complete the earliest. It can be decided by: task deadline, task arrival time, or advance reservation. Goal: The highest priority task will be executed first	-
Dependency Constraint (DpC)	Sequence of tasks based on task dependency. Goal: In case of precedence order among tasks, all parents task should be finished first before the task is scheduled	-
Deadline Constraint (DlC)	Represent the time till when the task should be finished . Goal: Not exceeded the deadline	-
Budget Constraint (BC)	Represent total cost of task execution. Goal: Not exceeded the budget	-

II. OPTIMIZATION METRIC

There are two perspectives on cloud computing services, the cloud consumer perspective and the cloud service provider perspective. Cloud consumers submit the tasks to be processed on cloud resources and desired good application performance. Cloud consumers consider these criteria: makespan, economic cost, flowtime, tardiness, waiting time, turnaround time, and fairness. CSP provides cloud resources on a rental basis and desired efficient resource utilization. CSP considers these criteria: resource utilization, throughput, priority constraint, dependency constraint, deadline constraint, and budget constraint [10].

Fig. 1 shows an example of nine independent Tasks (T_1, T_2, … T_9) running on two different Cloud Resources (R_1 and R_2). Execution times for T_1, T_2, … T_9 are 3, 4, 3, 3, 6, 6, 5, 3, and 2 time unit respectively.

Table 1 shows criteria details and example from cloud customer and CSP point of view. F_i, T_i, D_i, S_i, B_i, W_i, E_i shows finishing time, utilization time, deadline time, start time execution, submission time, waiting time, and execution time of task i respectively. C_i is cost resource i per unit time, n is number of resource, and i is task number. Cost for using resource R_1 and R_2 are 2,000 per time unit and 3,000 per time unit respectively.

III. TASK SCHEDULING

Distributed task scheduling uses heuristic, hybrid, and metaheuristic methods to deal with the issue of cloud service scalability. Task scheduling taxonomy can be seen in Fig.2.

1) Heuristic

A heuristic algorithm is a procedure in mathematical programming to determine the solution to an optimization problem.

TABLE II. SUMMARY OF PAPERS PRESENTING HEURISTIC APPROACH FOR CLOUD TASK SCHEDULING

Algorithm	**Testing Env. / Tools**	**Strength**	**Weakness**	**Optimization Metric**
CHEFT Al Rahayfeh et al. [11]	Simulation / CloudSim	Load balancing using Dominant Sequence Clustering (DSC) and Weighted Least Connection (WLC)	Not support dynamic load balancing	Ms, Td, RU, Tg, PC
FDHEFT Zhou et al. [12]	Real world & synthetic workflow	Support dynamic load balancing using Fuzzy Dominance Sort Mechanism with HEFT	Not support static load balancing	Ms, EC, Tg
PMHEFT Sohani et al. [13]	Simulation / WorkflowSim	Lower power consumption, Support static and dynamic load balancing	Waiting time might be longer if a lot of emergency tasks pop out.	Ms, RU, PC
OCTHEFT Talouki et al. [14]	Molecular, LU-Like, FFT, Montage workflow	Significantly reduce makespan by using Optimistic Cost Table downward (OCTd) and upward (OCTu) to prioritize task	Not considering power consumption	Ms, Ft, WT, RU, PC
Max-Min on DAC Patnaik et al. [15]	Montage & Epigenomic Workflow	Allocate multiple tasks over multiple resources with heterogeneous capacity	Only on independent tasks, not yet considering dependent tasks	Ms, Ft

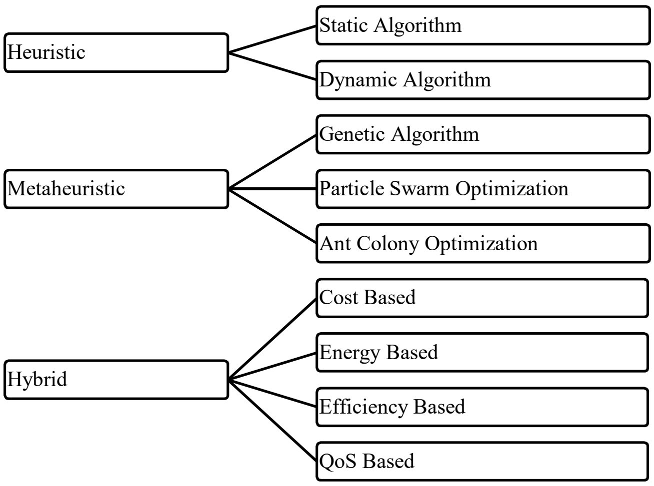

Fig. 2. Taxonomy of Task Scheduling on Cloud Computing

The heuristic algorithm is a problem dependent, hence it is better at finding local optimum but often fails in finding global optimum solution [16]. The heuristic can be categorized into the Greedy and the Local Search Method. The Greedy Method is suitable for cloud task scheduling since it could trade optimality, completeness, accuracy, or precision for speed [17]. A heuristic algorithm consists of two main categories: Static Algorithm and Dynamic Algorithm [9].

The Static Algorithm needs information about the incoming task and available VM resource. Its performance is better on minimum workload variation and minimum system behavior variation. Contrary to the Static Algorithm, the Dynamic algorithm does not need advanced information but requires continuous monitoring. The dynamic algorithm is more suitable for the cloud platform since cloud workload and system behavior frequently changes. It could solve uneven workload distribution by transferring the task from an overloaded VM to an underloaded VM. Heuristic cloud task scheduling methods can be seen in Table II.

HEFT (Heterogeneous Earliest Finish Time) is an example of a heuristic algorithm. HEFT consists of two stages: prioritizing tasks and assigning tasks to workers. The first stage will select the task with the highest upward rank value at each step using Critical-Path-on-a-Processor (CPOP) algorithm. The second stage will assign the critical tasks to the worker (processor), which minimizes the total execution time of the critical tasks [18].

TABLE III. SUMMARY OF PAPERS PRESENTING METAHEURISTIC APPROACH FOR CLOUD TASK SCHEDULING

Algorithm	**Testing Env. / Tools**	**Strength**	**Weakness**	**Optimization Metric**
NSGA III Imene et al. [19]	Simulation / CloudSim	Response time, cost, and power consumption is lower by selecting efficient resources	High runtime	EC, WT, RU
MoGA Emara et al. [20]	Simulation / CloudSim	Better makespan, scheduling length, throughput, resource utilization, and energy consumption.	Not considering task dependency and task priority	Ms, F, RU, Tg
MPSO-EECC Gorva et al. [21]	Simulation / CloudSim	Better load balancing using Modified Particle Swarm Optimization (MPSO) and has security feature using Enhanced Elliptic Curve Cryptography (EECC) algorithm	Not considering energy consumption	Ft, RU
ACO Liu [22]	Simulation / CloudSim	Shorter task completion time and cheaper execution cost on large-scale task scheduling.	Not considering energy consumption	EC, Ft, RU
IABS Yin et al. [23]	Real world (Amazon EC2) / Java	Optimize CSP profit and satisfy user deadline constraints on hybrid cloud (private & public cloud)	Not considering optimal resource utilization on private cloud	EC, DlC, BC
PSSA Zade et al. [24]	Simulation / CloudSim	Energy-aware and security-aware under high load conditions	Loose deadline constraint	Ms, Ft, F
ESFO Hojjat [25]	Simulation / CloudSim	Optimize task scheduling (makespan and energy consumption) in polynomial time	Not considering load balancing and cost	Ms, RU

TABLE IV. SUMMARY OF PAPERS PRESENTING HYBRID APPROACH FOR CLOUD TASK SCHEDULING

Algorithm	Testing Env. / Tools	Strength	Weakness	Optimization Metric
HAGA Ajmal et al. [26]	Simulation / CloudSim Plus	Reduce solution space by dividing task and VM into groups, so it can decrease execution time and data center cost	Not considering task dependency	Ms, Ft, TaT, BC
HCSO Bhagwan et al. [27]	Simulation / CloudSim	Workflow distributed on VM efficiently using HEFT, then using CSO to minimize makespan and cost	May get stuck on local minima and premature convergence	Ms, EC
EPETS Hussain et al. [28]	Real world / Python	1st stage: allocates tasks to the fastest VM, 2nd stage: reallocates to VM with less energy consumption but within the deadline to reduce energy consumption and improve performance	Not considering task dependency	Ft, PC, DlC
CR-PSO Dubey et al [29]	Simulation / CloudSim	1st stage: generate list solution with low time complexity using Chemical Reaction Optimization (CRO), 2nd stage: using PSO to reduce makespan, execution time, and cost.	Not considering energy consumption and load balancing	Ms, EC, RU, DlC
BWFSO Manikandan et al. [30]	Simulation / CloudSim	1st: VM clustering using Fuzzy C-Means, 2nd: Task scheduling using Black Widow Optimization, 3rd: Resource allocation using Fish Swarm Optimization to reduce energy, cost, and maximize resource utilization	Not considering task dependency	Ms, EC, RU
TSS Wang et al. [31]	Real world	The highest security level task is assigned to the local server. Earliest Deadline First (EDF) is used for task ordering, and Least Slack Time First (LSTF) used for task scheduling. After local server is fully occupied, then rent the best cost-performance VM on public cloud. Hence, it could guarantee security and good performance.	Not considering task dependency	Ms, EC, RU, DlC

HEFT later improved to be Multi-Object HEFT (MOHEFT) [32], Modified HEFT (MHEFT) [9]. Task scheduling using HEFT is implemented in Clustering HEFT (CHEFT) [11], Fuzzy Dominance HEFT (FDHEFT) [12], and Priority-based Modified HEFT (PMHEFT) [13]. Another heuristic algorithm example is using the Max-Min algorithm on Directed Acyclic Graph [15].

2) Metaheuristic

Metaheuristic algorithm is problem independent. Hence it can be used as a general problem-solving technique [16]. Metaheuristic cloud task scheduling methods can be seen in Table III.

Mapping task on unlimited cloud computing resources is an example of NP-Hard problems. No algorithms may produce optimal solutions within the polynomial time to solve it. The operating cost of scheduled solutions based on an exhaustive search is very high and not feasible. Metaheuristic algorithms effectively deal with the problem by providing near-optimal solutions within a reasonable time [10].

The Genetic Algorithm (GA) is based on natural selection of genetics "survival of the fittest". The genetics here represent the tasks, VMs, and servers. Task scheduling using GA is implemented in Non-dominated Sorting GA (NSGA-III) [19] and Multi-objective GA(MoGA) [20]. The particle swarm optimization (PSO) method is inspired by organism movement (particle) to move the swarm towards the best solution. It is implemented in Modified PSO [33]. Ant Colony Optimization (ACO) is inspired by feromon-based communication in an ant colony. It is implemented in the adaptive ACO [22].

Other emerging metaheuristic algorithms are the immune algorithm, Squirrel Search Algorithm, and SunFlower Optimization Algorithm.

The immune algorithm is based on antigen and antibody responses in nature. Antigen in the immune system represents optimization objective. Antibody represents a feasible solution. The affinity between both of them represents the quality of a feasible solution. Then, immune operations such as antibody clone, antibody mutation, and immune inhibition are applied to get the optimal solution. The immune algorithm is implemented in Immune Algorithm in Immune Algorithm-based Bag of Task Scheduling (IABS) [23].

The squirrel search algorithm is inspired by the dynamic behavior of southern flying squirrels searching for food resources by gliding among trees. The squirrel search algorithm is implemented in Parallel Squirrel Search Algorithm (PSSA) [24].

The sunflower algorithm is inspired by sunflower movement towards the sunlight to absorb solar radiation. It consists of two stages: pollination (cooperation to produce pollen gamete) and movement (to the light source or sun). The sunflower algorithm is implemented in Enhanced SunFlower Optimization Algorithm (ESFO) [25].

3) Hybrid

A hybrid algorithm is a combination of two or more task scheduling algorithms. It could combine two heuristic algorithms, two metaheuristic algorithms, or heuristic and metaheuristic algorithms. Hybrid algorithms focus on optimization to fulfill one objective: cost-based, energy-based, efficiency-based, and QoS-based [34]. Hybrid task scheduling methods can be seen in Table IV.

The cost-based algorithm is implemented in Hybrid Ant Genetic Algorithm (HAGA) [26] and HEFT and Cat Swarm Optimization (HCSO) [27]. The cat swarm optimization is inspired by the curiosity of the cats. It consists of two stages: seeking mode and tracking mode.

The energy-based algorithm is implemented in Energy and Performance Efficient Task Scheduling (EPETS) [28].

The efficiency-based algorithm implemented in Chemical reaction Partial Swarm Optimization (CR-PSO) [29] and Black Widow and Fish Swarm Optimization (BWFSO) [30]. The black widow is inspired by the mating behavior of the black widow which involved cannibalism. The fish swarm optimization is inspired by fish behavior to swarm in search of food.

QoS-based algorithm is implemented in the Task Scheduling method concerning Security (TSS) [31].

IV. Discussion

Task scheduling in a cloud platform is a complex problem that should meet both cloud consumer and cloud service provider expectations. Makespan, load balancing, and resource utilization costs are the most critical issues in task scheduling.

Heuristic algorithms solve task scheduling problems concerning the workflow of independent tasks. Their concern is mainly about minimizing makespan and flowtime, but they could not outperform metaheuristic algorithms in power consumption efficiency and cost of resource utilization.

The metaheuristic algorithm is not problem-dependent. Metaheuristic algorithms are basically inspired by intelligent biological processes (such as genetic crossover) and biological activities (such as swarm movement). Hence, it can efficiently explore the space search to find near-optimal solutions in a highly acceptable performance. The metaheuristic algorithm has better optimal use of cloud resources in load balancing, resource utilization, and power consumption. Emerging metaheuristic algorithms such as Sunflower Optimization or Squirrel Search Optimization have better performance in reducing energy consumption than previous metaheuristic algorithms.

Hybrid algorithms use two or more stages. Each stage has a different objective. The objective of the first stage is usually to narrow solution candidates based on low time complexity or low energy consumption. The second stage is to select near-optimal solutions based on performance (makespan, execution time, and cost). Swarm optimization is more suitable to be implemented in the second stage since it can find the global minima of a fitness function. In the hybrid cloud (cloud consumer uses private and public cloud), a hybrid task scheduling algorithm should also guarantee the security.

New algorithms are implemented in the real environment, case-based implementation environment, and simulated testing environment. 58% of the task scheduling algorithm studied in [6] is implemented using CloudSIM simulation toolkits, whereas in this paper 61% of them use CloudSIM. The benefits of using CloudSIM are the researcher needs less time and less deployment effort to set up a cloud task scheduling testing environment. It is also flexible enough to switch between space-shared or time-shared allocation and support large-scale cloud environments [35].

As observed in different algorithms listed in this paper, the objectives desired by Cloud Consumers and Cloud Service Providers cannot be achieved by a single heuristic or metaheuristic algorithm only. Using a hybrid method by combining two or more algorithms can overcome the shortcomings of existing algorithms and improve algorithm performance. But we must pay attention to the complexity of the hybrid method so that the deadline constraint can be met.

Hence, in future research, we recommend using Sunflower Optimization [25] or Squirrel Search Optimization [24] in the first stage of the task scheduling algorithm to minimize energy consumption. Then, Swarm Optimization (Particle Swarm Optimization / Cat Swarm Optimization / Fish Swarm Optimization) in the second stage is used to reduce makespan, execution time, cost, and maximize resource utilization on the given deadline. The proposed hybrid algorithm can be simulated using CloudSim first before being implemented in the real world.

V. CONCLUSION

This paper explores task scheduling algorithms based on the heuristic, metaheuristic, and hybrid methods and their corresponding optimization metrics. This paper reviews task scheduling algorithms from papers published on Sciencedirect and IEEExplore in 2019-2022. We found that generally, the hybrid method can improve task scheduling performance within given constraints.

Acknowledgment

The author would like to acknowledge the funding support provided by Institut Teknologi Sepuluh Nopember (ITS) under Institution Development Unit Research Grant Batch 2, Number 1879/PKS/ITS/2022.

References

[1] S. Bhagavathiperumal and M. Goyal, "Workload Analysis of Cloud Resources using Time Series and Machine Learning Prediction," in *2019 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, 2019.

[2] J. Zhang, H. Huang and X. Wang, "Resource provision algorithms in cloud computing: A survey," *Journal of Network and Computer Applications,* vol. 64, pp. 23-42, 2016.

[3] S. Goodarzy, M. Nazari, R. Han, E. Keller and E. Rozner, "Resource Management in Cloud Computing Using Machine Learning: A Survey," in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2020.

[4] M. Cheng, J. Li and S. Nazarian, "DRL-Cloud: Deep Reinforcement Learning-Based Resource Provisioning and Task Scheduling for Cloud Service Providers," *IEEE Xplore,* pp. 129-134, 2018.

[5] H. Singh, S. Tyagi, P. Kumar, S. S. Gill and R. Buyya, "Metaheuristic for Scheduling of Heterogeneus Tasks in Cloud Computing Environtments : Analysis, Performance Evaluation, and Future Directions," *Simulation Modelling Practice and Theory,* vol. 111, pp. 1-23, 2021.

[6] E. H. Houssein, A. G. Gad, Y. M. Wazery and P. N. Suganthan, "Task Scheduling in Cloud Computing Based on Meta-Heuristic : Review, Taxonomy, Open Challenges, and Future Trends," *Swarm and Evolutionary Computation,* vol. 62, pp. 1-41, 2021.

[7] Wolfram Research, Inc., "NP-Hard Problem," Wolfram Mathworld, 19 April 2022. [Online]. Available: https://mathworld.wolfram.com/NP-HardProblem.html.

[8] I. Ahmad, M. G. AlFailakawi, A. AlMutawa and L. Alsalman, "Container scheduling techniques: A Survey and assessment," *Journal*

of King Saud University –Computer and Information Sciences, pp. 1-14, 2021.

[9] K. Dubey, M. Kumar and S. C. Sharma, "Modified HEFT Algorithm for Task Scheduling in Cloud Environtment," *Procedia Computer Science,* vol. 125, pp. 725-732, 2018.

[10] M. Kalra and S. Singh, "A review of metaheuristic scheduling techniques in cloud computing," *Egyptian Informatics Journal,* vol. 16, pp. 275-295, 2015.

[11] A. Al-Rahayfeh, S. Atiewi, A. Abuhussein and M. Almiani, "Novel Approach to Task Scheduling and Load Balancing Using the Dominant Sequence Clustering and Mean Shift Clustering Algorithms," *Future Internet,* vol. 11, no. 5, pp. 1-15, 2019.

[12] X. Zhou, G. Zhang, J. Sun, J. Zhou, T. Wei and S. Hu, "Minimizing cost and makespan for workflow scheduling in cloud using fuzzy dominance sort based HEFT," *Future Generation Computer Systems,* vol. 93, pp. 278-289, 2019.

[13] M. Sohani and S. C. Jain, "A Predictive Priority-Based Dynamic Resource Provisioning Scheme With Load Balancing in Heterogeneous Cloud Computing," *IEEE access,* vol. 9, pp. 62653 - 62664, 2021.

[14] R. N. Talouki, M. H. Shirvani and H. Motameni, "A Heuristic-based Task Scheduling Algorithm for Scientific Workflows in Heterogeneous Cloud Computing Platforms," *Journal of King Saud University - Computer and Information Sciences,* pp. 1-12, 2021.

[15] H. K. Patnaik, M. R. Patra and R. Kumar, "A Workflow Based Approach for Task Scheduling in Cloud Environment," *Materials Today : Proceedings,* pp. 1-7, 2022.

[16] M. Kumar, S. C. Sharma, A. Goel and S. P. Singh, "A comprehensive survey for scheduling techniques in cloud computing," *Journal of Network and Computer Applications,* vol. 143, pp. 1-33, 2019.

[17] S. N. Sze and W. K. Tiong, "A Comparison between Heuristic and MetaHeuristic Methods for Solving the Multiple Traveling Salesman Problem," *World Academy of Science, Engineering and Technology,* vol. 25, pp. 300-303, 2007.

[18] H. Topcuoglu, S. Hariri and M.-Y. Wu, "Performance-effective and low-complexity task scheduling for heterogeneous computing," *IEEE Transactions on Parallel and Distributed Systems,* vol. 13, no. 3, pp. 260-274, March 2002.

[19] L. Imene, S. Sihem, K. Okba and B. Mohamed, "A Third Generation Genetic Algorithm NSGA-III for Task Scheduling in Cloud Computing," *Journal of King Saud University - Computer and Information Sciences,* pp. 1-15, 2022.

[20] F. A. Emara, A. A. Gad-Elrab, A. Sobhi and K. R. Raslan, "Genetic-Based Multi-objective Task Scheduling Algorithm in Cloud Computing Environment," *International Journal of Intelligent Engineering and Systems (INASS),* vol. 14, no. 5, pp. 571-582, 2021.

[21] S. K. Gorva and L. C. Anandachar, "Effective Load Balancing and Security in Cloud using Modified Particle Swarm Optimization Technique and Enhanced Elliptic Curve Cryptography Algorithm," *International Journal of Intelligent Engineering and Systems (INASS),* vol. 15, no. 2, pp. 190-199, 2021.

[22] H. Liu, "Research on cloud computing adaptive task scheduling based on ant colony algorithm," *Optik International Journal for Light and Electron Optics,* vol. 258, pp. 1-15, 2022.

[23] L. Yin, J. Zhou and J. Sun, "A Stochastic Algorithm for Scheduling Bag-of-Task Applications on Hybrid Clouds under Task Duration Variations," *The Journal of Systems and Software,* vol. 184, pp. 1-15, 2022.

[24] B. M. H. Zade, N. Mansouri and M. M. Javidi, "SAEA : A Security-aware and Energy-aware Task Scheduling Strategy by Parallel Squirrel Search Algorithm in Cloud Environtment," *Expert Systems with Applications,* vol. 176, pp. 1-30, 2021.

[25] H. Emami, "Cloud Task Scheduling using Enhanced Sunflower Optimization Algorithm," *The Korean Institute of Communications and Information Sciences,* vol. 8, pp. 97-100, 2022.

[26] M. S. Ajmal, Z. Iqbal, F. Z. Khan, M. Ahmad, I. Ahmad and B. B. Gupta, "Hybrid Ant Genetic Algorithm for Efficient Task Scheduling in Cloud Data Centers," *Computers and Electrical Engineering,* vol. 95, pp. 1-15, 2021.

[27] J. Bhagwan and S. Kumar, "An H-CSO Algorithm for Workflow Scheduling in Heterogeneous Cloud Environtment," *International Journal of Intelligent Engineering and Systems (INASS),* vol. 14, no. 5, pp. 422-434, 2021.

[28] M. Hussain, L.-F. Wei, A. Lakhan, S. Wali, S. Ali and A. Hussain, "Energy and Performance-Efficient Task Scheduling in Heterogeneous Virtualized Cloud Computing," *Sustainable Computing : Informatics and Systems,* vol. 30, pp. 1-12, 2021.

[29] K. Dubey and S. C. Sharma, "A Novel Multi-Objective CR-PSO Task Scheduling Algorithm with Deadline Constraint in Cloud Computing," *Sustainable Computing : Informatics and Systems,* vol. 32, pp. 1-20, 2021.

[30] N. Manikandan, P. Divya and S. Janani, "BWFSO: Hybrid Black-Widow and Fish Swarm Optimization Algorithm for Resource Allocation and Task Scheduling in Cloud Computing," *Materials Today,* pp. 1-6, 2022.

[31] B. Wang, C. Wang, W. Huang, Y. Song and X. Qin, "Security-aware Task Scheduling with Deadline Constraint on Heterogenous Hybrid Cloud," *Journal Parallel and Distributed Computing,* vol. 153, pp. 15-28, 2021.

[32] J. J. Durillo and R. Prodan, "Multi-objective workflow scheduling in Amazon EC2," *Cluster Comput,* vol. 17, pp. 169 - 189, 2014.

[33] G. Lindfield and J. Penny, "Particle Swarm Optimization Algorithms," in *Introduction to Nature-Inspired Optimization*, Academic Press, 2017, pp. 49-68.

[34] H. Naghavipour, T. K. Soon, M. Y. I. B. Idris, M. Namvar, R. B. Salleh and A. Gani, "Hybrid Metaheuristics for QoS-Aware Service Composition: A Systematic Mapping Study," *IEEEAccess,* vol. 10, pp. 12678-12701, 2022.

[35] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose and R. Buyya, "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms," *Software: Practice and experience,* vol. 41, no. 1, pp. 23-50, 2011.