

Heuristic and Meta-Heuristic Optimization Models for Task Scheduling in Cloud-Fog Systems: A Review

Mohammed Najm Abdulredha *, Bara'a A. Attea, Adnan Jumaa Jabir

Department of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq

Correspondence

* Mohammed Najm Abdulredha
Department of Computer Science, College of Science,
University of Baghdad, Baghdad, Iraq
Email: mohammed.najm.422@gmail.com

Abstract

Nowadays, cloud computing has attracted the attention of large companies due to its high potential, flexibility, and profitability in providing multi-sources of hardware and software to serve the connected users. Given the scale of modern data centers and the dynamic nature of their resource provisioning, we need effective scheduling techniques to manage these resources while satisfying both the cloud providers and cloud users goals. Task scheduling in cloud computing is considered as NP-hard problem which cannot be easily solved by classical optimization methods. Thus, both heuristic and meta-heuristic techniques have been utilized to provide optimal or near-optimal solutions within an acceptable time frame for such problems. In this article, a summary of heuristic and meta-heuristic methods for solving the task scheduling optimization in cloud-fog systems is presented. The cost and time aware scheduling methods for both bag of tasks and workflow tasks are reviewed, discussed, and analyzed thoroughly to provide a clear vision for the readers in order to select the proper methods which fulfill their needs.

KEYWORDS: Bag of Task, Cloud computing, Makespan, Meta-heuristic, Task scheduling, Workflow.

I. INTRODUCTION

Cloud computing refers to the computer resources and systems available on-demand over the network to provide several integrated computing services [1]. Cloud computing offers three basic types of service models, namely Infrastructure as a Service (IaaS), Software as a Service (SaaS), and Platform as a Service (PaaS) [2]. Cloud computing can be implemented as a layered architecture and comes in four main development models: Public, Private, Hybrid, and Community clouds [3][4].

Cloud computing provides high flexibility and speed in expanding or reducing the infrastructure, anytime, anywhere computing, efficient allocation and distribution of resources to keep capital spending from wasting unused resources, and pay as use charging strategy. However, the salient downside of the cloud system is the high transmission delay as the data destined to cloud servers should traverse a long-distance, unstable, and usually congested internet connection [5]. As a result, CISCO announced the Fog computing which brings the computation capabilities to the user vicinity. The fog nodes, which are characterized by a low computing and storing capabilities in comparison with the cloud high-performance server, are placed at the edge network, such that the lightweight tasks can be processed locally without the

burden of communicating with the far cloud nodes [5]. The use of fog nodes reduces both the communication latency and processing cost which in turn satisfied the user demands by reducing both the processing cost and transmission latency. However, the combination between cloud and fogs encounters various issues to meet the user demands as they deliver dissimilar services, e.g., expensive and low latency cloud services versus cheap and high latency fog services.

In cloud computing, various cloud users require different services depending on their changing needs, thus the most important challenge is how the user tasks are scheduled efficiently to satisfy different criteria for both parties, named cloud provider and cloud users. The cloud providers aim to increase their revenue without affecting the cloud system performance; this can be achieved by resource utilization, load balancing, reducing the energy consumption, and increasing the system throughput [6]. Contrarily, the cloud users look for a low cost, low latency services which are not exceeding their tasks budget and deadline limits. In general, the key issues in cloud environments have been categorized into seven major categories: resource management, load balancing, privacy and security, transition to the cloud, availability and scalability, energy efficiency, interoperability, and compatibility [6]. Scheduling in a cloud



This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2020 The Authors. Published by Iraqi Journal for Electrical and Electronic Engineering by College of Engineering, University of Basrah.

computing environment means that many tasks can be performed on the pool of available computing resources in the most optimal way. This process relies on several improvement criteria such as reliability, outreach, load balancing, implementation cost, budget, and resource utilization [7]. In task scheduling process, the tasks are delivered from the users to the cloud scheduling software, and then the cloud scheduler explores the state of the resources from the cloud information service, as follows: the Datacenter Broker (DB) monitors all resources presented in the cloud system and collects the current status of all available resources and all remaining resources that might be available in the cloud system. The cloud scheduler then determines the target resource should be chosen and allocated to the task based on the exploration stage information received [8]. The task scheduling can be mainly categorized into static and dynamic fashions. In the former, all the tasks specifications are known in advance which allows scheduling all tasks before executing any task, while in the dynamic scheduling, such tasks information is not available which makes a cloud provider can't make an efficient plan before execution.

On the other hand, the tasks may appear in different models depending on the task interrelationship. One kind of task is the bag of tasks (BoT), where the tasks can be processed in parallel in a single processing level with zero dependencies among each other. Adversely, the workflow tasks can have a tight interrelationship, such that the execution of one subtask may be highly dependent on one or more subtasks in previous levels.

Task scheduling has been proven as a typical NP-hard optimization problem [9] which cannot be efficiently solved by conventional methods. Therefore, recent studies have intensified the research around the use of heuristic and metaheuristic algorithms for task scheduling optimization due to their effectiveness in solving problems of high complexity and large sizes [10]. For example, several researches formulated the BoT scheduling as a single objective problem to tackle the problem of cost [11] and makespan [12] optimization with some constraints like execution deadline and task budget. Alternatively, due to the high complexity of workflow tasks, several multi-objective meta-heuristic methods have been utilized to find the near-optimal scheduling solution by considering multiple conflicted optimization objectives, named cost and makespan [13].

In this work, the related heuristic and meta-heuristic task scheduling methods for the cloud-fog environment are reviewed, categorized, and analyzed thoroughly to give the readers a clear vision about the task scheduling algorithms to choose the proper optimization method for their problems. Specifically, this work concentrates on the heuristics and meta-heuristics task scheduling methods considering both the cost and makespan objectives.

The remainder of this paper is organized into five sections. Preliminary concepts are briefly presented in section 2. Bag-of-Tasks scheduling is discussed in section 3. The workflow scheduling is presented in section 4. Finally, a conclusion and a potential for future work are summarized in section 5.

II. PRELIMINARY CONCEPTS

This section presents the preliminary concepts which represent the base of our review. It states the task scheduling problem formulation, the performance metrics, and the concepts of the heuristic and meta-heuristic algorithms.

A. Task Scheduling Problem and Optimization Metrics

Task scheduling means the best use of the available resources. The high performance of the cloud computing environment depends greatly on the efficiency of resource scheduling. Most current scheduling algorithms take into account various parameters such as time (i.e. makespan) and cost [14]. However, there are other essential parameters, such as reliability and availability [15] to be considered.

The goal of task scheduling in the Cloud system aims at the benefit of service providers or users. On the service providers side, they concern about some criteria of resource utilization, load balancing, throughput, and energy efficiency. On the other hand, the purpose of the users is to minimize cost makespan based on some constraints like deadline and budget.

The task scheduling problem can be expressed as a three tuple system of (T, P, M) , where T represents the set of tasks, P is the set of processing nodes, and M is the optimal mapping between tasks and processing units. The mapping should satisfy the best execution time and cost.

The Cloud-Fog computing infrastructure consists of Cloud nodes and Fog nodes, which have the same attributes such as CPU rate, CPU usage fee, bandwidth usage fee, and memory usage fee. However, Cloud nodes are typically more powerful than the Fog nodes, but the cost of using them is greater. The set of P processors including nC Cloud nodes and nF Fog nodes in the system ($\mathcal{N} = nC \cup nF$) is expressed as:

$$P = \{P_1, P_2, P_3, \dots, P_N\} \quad (1)$$

Makespan and monetary cost metrics are the main optimization metrics used for scheduling both BoT and workflow. In the BoT scheduling problem, the goal is to find an optimal mapping between independent tasks and corresponding computing cloud-fog nodes. Let $T = \{T_1, T_2, T_3, \dots, T_m\}$ is a set of m independent tasks which have various characteristics. Each task T_k ($T_k \in T$) is assigned to the processor P_i ($P_i \in P$), which is represented as T_k^i , where each processing node can be utilized to process a set of one or more tasks:

$$P_i T = \{T_x^i, T_y^i, \dots, T_z^i\} \quad (2)$$

For a set of $P_i T$ tasks, the execution time (ET) that node P_i needs to complete all assigned tasks is:

$$ET(P_i) = \sum_{T_k^i \in P_i T} ET(T_k^i) \quad (3)$$

Where $ET(T_k^i)$ is the execution time of the task T_k on the processing node P_i .

Makespan (M) is the total time for the system to complete all tasks, defined from the time when the request is received to the time that the last task is completed, or the time when the last machine finishes. It can be expressed as:

$$M = \text{Max}_{1 \leq i \leq m} [ET(P_i)] \quad (4)$$

When one task is executed in the Cloud–Fog system, a monetary amount must be paid for processing cost, memory usage cost, bandwidth usage cost, and so on. The total cost for all tasks to be executed in the Cloud–Fog system is calculated as follows:

$$TotalCost = \sum_{P_i \in P} \sum_{T_k^i \in P_i T} Cost(T_k^i) \quad (5)$$

Where $Cost(T_k^i)$ is the cost required for executing the task T_k in the processing node P_i .

On the other hand, the workflow tasks are commonly represented as a Direct Acyclic Graph (DAG), where the graph nodes represent the tasks, and the task inter-dependencies are represented by the graph edges. Let $W = (T, D)$ is a DAG, where $T = \{T_0, T_1, \dots, T_m\}$ is the set of tasks and $D = \{(T_i, T_j) \mid T_i, T_j \in T\}$ is the set of data or control dependencies. In our proposed system, the graph nodes represent the execution time of the tasks, while the edges represent the size of data that is transferred between nodes. The data size transferred from T_i to T_j is denoted as $data(T_i, T_j)$ [13]. Also, according to the nature of workflow tasks, they are characterized by the task predecessors, entry task, and exit task. In this work, we assume that all workflow tasks have single entry task, multiple predecessors, and single exit task. The predecessors of a task T_i are defined as:

$$pred(T_i) = \{T_j \mid (T_j, T_i) \in D\} \quad (6)$$

The entry task and exit task should satisfy the following rules:

$$pred(T_{entry}) = \emptyset \quad (7)$$

$$\nexists T_i \in T : T_{exit} \in pred(T_i) \quad (8)$$

The node communication bandwidth is normally proportional to the processing power, the higher computation power, the higher node bandwidth, and vice versa. Let $bw(p_i)$ and $bw(q_i)$ denote the bandwidth of nodes p_i and q_i which are allocated to execute tasks T_i and T_j , respectively, then the communication time between task T_i and T_j can be derived as [13]:

$$T_{com}(T_i, T_j) = \begin{cases} \frac{data(T_i, T_j)}{\min\{bw(p_i), bw(q_i)\}}, & p \neq q \\ 0, & p = q \end{cases} \quad (9)$$

Let $FT(T_i)$ denotes the finish time of the task T_i , then the makespan can be expressed as [13]:

$$M = Max(FT(T_{exit})) \quad (10)$$

In workflow, due to the interdependency between tasks, the task cannot start until all of its predecessor tasks are finished. Thus, the task starting time (ST) is calculated as the sum of the finish time of the task predecessors, the communication time between its predecessors tasks, and the node available time. These relations can be expressed as [13]:

$$ST(T_{entry}) = 0 \quad (11)$$

$$ST(T_i) = \max \left\{ avail(p(T_i)), \max_{T_j \in pred(T_i)} \left(FT(T_j) + T_{com}(T_j, T_i) \right) \right\} \quad (12)$$

$$FT(T_i) = ST(T_i) + ET(T_i) \quad (13)$$

B. Meta-Heuristics

Complexities such as non-convexity, nonlinearities, discontinuities, various disciplines, and vast dimensionality are found in most real-world search and optimization

problems, a mixture of which makes conventional known algorithms either inefficient, impractical, or inapplicable [16]. These problems are known as NP-hard problem which cannot be solved easily by classical exact methods without incurring a very long execution time [17]. For NP-hard problems, as the size of the instance increases, the time needed to solve a given problem instance increases in a super polynomial fashion, rendering the implementation of precise methods impractical [18]. Thus, search and optimization algorithms are typically generated using some heuristics to solve certain problems. However, heuristic algorithms lack the solid mathematical foundations and are designed for specific problems. On the other hand, Metaheuristics are generalized methods that describe algorithmic frameworks to build a series of techniques capable of discovering approximate solutions required for optimization problems in a large search space efficiently [19]. More explicitly, metaheuristics reflect 'higher level,' heuristic-based, soft computing algorithms that can be targeted at some different optimization. Meta-heuristics are soft-computing strategies that are programmed to accept the presence of incorrect solutions and enhance them to higher performing solutions that fulfill the criteria of real-world problem instances. Besides, to compute correct optimization solutions, metaheuristics can accommodate imprecision, ambiguity, and/or approximation errors in the problem data, as well as intelligently treat partial knowledge.

The basic purpose of the metaheuristic search is to obtain a kind of balance for solution diversification and intensification. For any good search or optimization process, these two principles are potential. Diversification refers to the technique's potential to achieve a successful search space exploration pattern to reduce trapping in a local minima. Adversely, intensification focuses on exploit the current solution to improve its quality.

Metaheuristics are not primarily based on addressing problems of any kind, but they suggest basic solutions that are extremely relevant to a broad range of topics. These basic methods are typically based on the interpretation, among many others, of natural or physical phenomena, such as the behavior of flocks of birds and insects and natural evolution. Several proposals for metaheuristic approaches have been introduced, including well-known classical techniques such as particle swarm optimization (PSO), genetic algorithms (GA), tabu search (TS), ant colony optimization (ACO), simulated annealing (SA), memetic algorithm (MA), and bat algorithm (BA), are used to solve the task scheduling issue. Numerous research results [17], [20], [21] have been made to ensure that meta-heuristic scheduling algorithms can provide better optimal scheduling than traditional scheduling algorithms. Metaheuristics when defined algorithmically [22], are iterative methods that plan to intelligently apply a group of heuristic components over a provisional solution or a group of provisional solutions in each stage. Without loss of generality, the framework for a meta-heuristic algorithm is shown in Algorithm 1 [23].

Algorithm 1: A Generic Metaheuristic Framework

-
- 1:** Set iteration counter $t = 0$
 - 2:** Initialize N solution S_i randomly
 - 3:** Evaluate each member of S_t
 - 4:** Mark the best solution of S_t as x_t^*
 - 5:** **repeat**
 - 6:** Choose μ solutions (set P_t) from S_i using a selection plan (SP)
 - 7:** Generate λ new solutions (set C_t) from P_t using a generation plan (GP)
 - 8:** Choose p solution (set R_t) from P_t using replacement plan (RP)
 - 9:** Update S_t by replacing R_t using p solutions from a pool of any combination of at most three sets P_t , C_t and R_t using an update plan (UP)
 - 10:** Evaluate each member of S_t
 - 11:** Identify the best solution of S_t and update x_t^*
 - 12:** $t \leftarrow t + 1$
 - 13:** **until** (a termination plan (TP) is satisfied)
 - 14:** Declare the near-optimal solution as x_t^*
-

III. BAG-OF-TASKS SCHEDULING

The BoT applications are typical embarrassingly parallel types of applications consisting of many independent tasks, which can be processed in parallel without synchronization or communication. The BoT applications are widely spread in computer imaging, computational biology, parameter sweeps, fractal calculations, data mining, and Monte Carlo simulations [24]. Several researches have tackled the problem of finding the best scheduling solution for BoT tasks according to various optimization criteria and constraints. In literature, various works have been conducted to optimize the total scheduling cost for the BoT tasks concerning different constraints.

A cost-effective scheduling scheme was proposed by Den Borsches et al [11] to optimally allocate resources in private and multiple public clouds environments. They proposed a heuristic to find an efficient solution for the cost required for running as many applications as possible within deadline constraints.

Another study was proposed by Rodrigo et al [25], named Resource Management System (RMS) that manages many local resources in public Clouds. RMS provides a dynamic resource scheduling policy for deadline constrained applications to reduce cost and meet deadlines in public clouds. Also, they presented a new mechanism for billing by sharing the cost resource allocation among users.

Zhang et al [26] proposed a greedy heuristic (GH) approach to minimize the total cost for the due-date-constrained Bag-of-Tasks Scheduling Problem (BTSP) in hybrid clouds. It encompasses two phases, task ordering and task scheduling. The task ordering is accomplished by an Earlier Latest Start Time First method (ELSTF), while the task scheduling is achieved by a Task Dispatching method

(TD). The GH method outperformed the RoundRobin method in terms of total cost minimization.

Abdi et al. [27] formulated the problem of resource allocation for BoT in a federation of clouds as an integer linear programming problem. They proposed a greedy randomized adaptive search procedure (GRASP) model to minimize the total cost including the fees required for running VMs and data transfer while considering both execution deadline and resource constraints in the clouds.

A bio-inspired modified particle swarm optimization (MPSO) algorithm was proposed by Domanal et al [28] to find the minimum execution cost in a heterogeneous infrastructures of the cloud, where the backpropagation artificial neural network (ANN) was used to forecast the spot instances' future values using the past data of a particular region. Their proposed scheduling algorithm outperformed various state-of-the-art algorithms like Round Robin, First Come First Serve, Ant Colony Optimization, and Genetic Algorithm.

Zhang et al [29] considered the due date constraints while reducing the total cost for BoT the hybrid clouds. They proposed three task rescheduling heuristics by considering the data transmission cost and developed two effective acceleration methods to further enhance the efficiency.

Additionally, there have been several works focused on reducing the task completion time. To ensure a good trade-off between availability and Makespans, a hybrid Heuristic-Ant Colony Optimization (H2ACO) was developed by Tong et al. [30] for multiclass tasks scheduling on heterogeneous distributed systems with availability constraint.

Abdi et al [12] proposed a modified particle swarm (MPSO) scheme to optimize the task scheduling in terms of the completion time. MPSO has shown better performance in comparison with PSO and GA algorithms in the cloud environment.

A low complexity algorithm was proposed by Zeng et al [31], named, fog computing supported software-defined embedded system (FC-SDES) to jointly optimize the task scheduling, task image placement, and transmission time metrics to minimize the overall completion time. The minimization problem was converted into a mixed-integer nonlinear programming problem which was then solved by a low-complexity three-stage algorithm, where the first two stages dealt with the I/O and computation time while the third stage incorporate the transmission time.

Zhang et al [32] proposed an effective heuristic (EH) to find the minimum Makespan according to the budget in hybrid clouds. To accomplish the required task execution cost minimization, EH divided the optimization problem into task sequencing and scheduling phases. The longest Task First method was applied to generate a task sequence followed by the task assignment method to schedule all tasks in the obtained sequence. EH scheme outperformed the round-robin in terms of finding the minimum Makespan within the task budget.

Abdi et al [33] developed a mathematical programming model for resource allocation problems in federated clouds. The problem was formulated as a binary linear programming (BLP) to minimize the total execution cost by considering the deadline and resource constraints. The CPLEX solver,

which is a high-performance solver for linear programming, was used to solve the problem with the results showing that the cloud federations benefit customers, compared with a single cloud provider.

A bee's life algorithm (BLA) was proposed by Bitam et al [34] for task scheduling problems to efficiently utilize resources in the fog network by optimizing both the execution time and memory usage. Xuan et al [35] proposed a new scheduling algorithm, named sorting allocation-pulling (SAP) scheduling algorithm, which first allocates the tasks in available time segments on proper processors and then pulls them to a suitable available time segment for the sake of minimizing the makespan.

On the other hand, several kinds of literature have studied the BoT scheduling problem by considering both makespan and cost criteria. Thai et al [36] investigated the trade-off between performance and cost when executing Bag-of-Distributed-Tasks (BoDT) on the cloud and proposed a heuristic algorithm to find a proper execution plan based on a given budget constraint.

An online algorithm based on the Lyapunov optimization technique was proposed by Nan et al [37] to online adjust the tradeoff between average response time and average cost. Their proposed method outperformed both Fog-First and Cloud-First algorithms which were implemented as the performance benchmarks in their experiments.

Sindhu et al [38] proposed a novel deadline constrained bi-objective genetic algorithm-based scheduler (DBOGA) to schedule a BoT application in a cloud environment to find an optimal trade-off between execution time and execution cost. DBOGA has defined a new fitness based on the deflection of the individual fitness from the mean of lower and upper bounds, which contributed to finding the optimal solution quickly. DBOGA outperformed both the standard genetic algorithm (SGA) and BaTS in terms of execution cost and time.

Recently, Nguyen et al [39] presented a genetic algorithm for optimization of job scheduling in a Cloud-Fog environment to reduce the makespan and operating cost. Their proposed method was compared with the Bee Life Algorithm (BLA), Modified Particle Swarm Optimization (MPSO) and Round Robin (RR) approaches in terms of makespan and execution cost. However, the main downsides of their scheme are the high complexity of chromosome representation, and both the conflicting objectives (cost and makespan) are modeled as a single objective function. Table 1 summarizes the scheduling schemes in terms of methodology, environment, task type, optimization objectives, and constraints in BoT.

IV. WORKFLOW SCHEDULING

Workflow is a common model used to represent scientific computing, including complex simulation and precise analysis of massive data [43][44][13]. For example, CyberShake is used to depict the earthquake threats in an area with the help of the Probabilistic Seismic Hazard Analysis method. Epigenomics interprets the processing of genetic data to implement the different genome sequencing operations. SIPHT implements the bioinformatics problems.

LIGO is used for the identification of gravitational waves and to examine the data attained from compact binary systems. Montage in the aerospace field which synthesizes a series of FITS (Flexible Image Transport System) images into outer space nebula images. Grid workflow management systems, like Pegasus [45] and ASKALON [46], were designed to support executing workflows on Cloud platforms.

TABLE 1.
A Comparison of the existing BoT scheduling schemes.

Reference	Year	Algorithm	Environment	Optimization Criteria	Constraints
Den Borsches et al [11]	2011	heuristics for cost-efficient scheduling	Hybrid Clouds	cost	Deadline
Rodrigo et al [33]	2012	RMS	Cloud	cost	Deadline
Tong et al. [30]	2014	H2ACO	Hetero system	availability and Makespans	NA
Abdi et al [12]	2014	MPSO	Cloud	Makespan	NA
Zeng et al [31]	2016	FC-SDES	Fog	Makespan	NA
Nan et al [37]	2016	unit-slot optimization	Cloud fog	time and cost	NA
Zhang et al [26]	2016	BTSP	hybrid cloud	Cost	NA
Abdi et al [25]	2017	BLP	hybrid cloud	Cost	Deadline
Thai et al [36]	2017	Hybrid Scheduling Approach	Cloud	Cost	deadline
Zhang et al [32]	2017	BTSP	hybrid clouds	Makespan	budget
Bitam et al [34]	2018	BLA	Fog	execution time and memory	NA
Abdi et al [27]	2018	GRASP-FC, IP-FC	Cloud	Cost	deadline
Sindhu et al [38]	2018	DBOGA	Cloud	Cost and Makespan	deadline
Domanal et al [28]	2018	bio-inspired algorithm	Cloud	Cost	NA
Nguyen et al [39]	2019	TCaS	Cloud fog	Makespan and operating Costs	NA
Xuan et al [35]	2019	SAP	Heterogeneous Distributed System	Makespan	NA
Sun et al [40]	2019	CSIA	Public Cloud	Makespan	Budget
Zhang et al [29]	2019	TRH1-TRH3	Hybrid Clouds	Cost	Due date
Parlavantzas et al [41]	2020	DiFFuSE	Cloud	Cost	NA
Tychalas et al [42]	2020	Simulation	Cloud-fog	Cost	NA

A workflow is usually formulated as Directed Acyclic Graphs (DAGs), where nodes represent the tasks and edges between the nodes represent the data or control dependencies between the tasks. In the literature, there have been various studies to find an optimal scheduling solution for the workflow take to minimize both task execution cost and

Makespan subject to various constraints. Figure 1 depicts five different Structures of the real-world workflows.

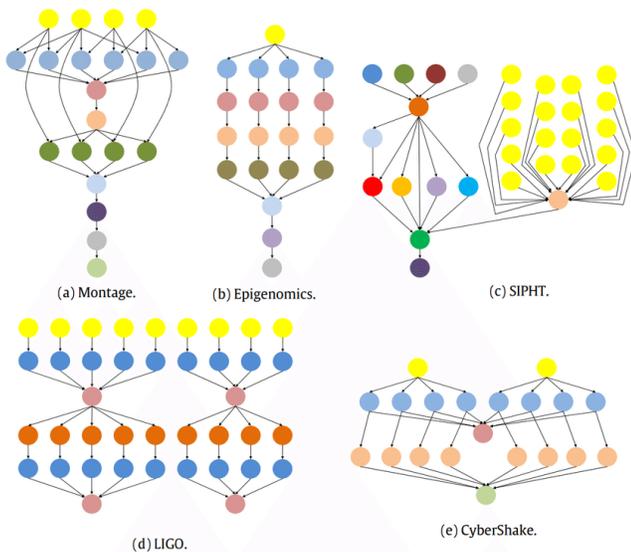


Figure 1: Structures of five real-world workflows.

Several methods have formulated the optimization problem as a single- or multi-objective optimization problem using some heuristics algorithms to find the optimal solution. To make a tradeoff between using the expensive high-speed public cloud and the free of charge private resources, Bittencourt et al [47] proposed a Hybrid Cloud Optimized Cost (HCOC) resource scheduling mechanism to solve the problem of determining the amount of resources required to execute workflows within budget and execution time. HCOC used Dynamic Deployment Virtual Resource (DDVR) method to improve resource research by finding adequate resources based on QoS requirements.

Szabo and Kröger [48] examined a task allocation problem in cloud computing using evolutionary genetic algorithms to minimize workflow completion time and cost, while considering the communication overhead and transfer bandwidth between tasks. They proposed a single and multiobjective evolutionary approach with a mutation and crossover operator designed to guide the search for the optimal solution. Their experimental results showed the superiority of the multiobjective scheme over the single objective in terms of finding the best solution.

Durillo et al [49] proposed a multi-objective Heterogeneous Earliest Finish Time (MOHFET) method to schedule workflows over Amazon EC2. The authors considered minimizing cost and time as two conflicting objectives and created the Pareto optimal schedules from which the user can select the best solution manually. However, as an enumeration-based approach, MOHFET incurs a very poor efficiency when dealing with large-scale workflows.

Xu et al [50] proposed a multi-objective heuristic algorithm, named, Min-min based time and cost tradeoff (MTCT) to optimize workflow scheduling subject to time and cost metrics with fault-tolerant considerations. Four different real-world scientific workflow scenarios were used,

and the outcomes indicated that fault recovery has considerable influence on the two performance criteria.

Casas et al [51] suggested a Balanced and file Reuse–Replication Scheduling (BaRRS) algorithm to optimize the cost and makespan metrics for workflow scheduling problems. The algorithm also computes the optimal number of VMs for a specific application, but it incurs a high computational overhead.

Zhou et al [52] proposed a fuzzy dominance sort based heterogeneous earliest finish-time (FDHEFT) algorithm to minimize both cost and makespan of workflows deployed on IaaS clouds. However, these approaches are significantly restricted by prior expert’s knowledge from a static global point of view, which cannot appropriately describe the dynamic process of workflow scheduling.

Zhou et al [53] proposed a multi-objective scheduling method for hybrid clouds intending to reduce Makespan and Cost. In their method, both single-objective (Cost reduction) and Pareto based multi-objective task scheduling (Cost and Makespan reduction) were presented. However, they have used a small number of tasks (100 Task) and the performance of their proposed method has not been compared against any state of the art methods.

Another study was proposed by Wu et al [54], named, multiobjective evolutionary list scheduling (MOELS) algorithm to minimize both execution makespan and cost in the cloud. In MOELS, the classical list scheduling technique is embedded in the evolutionary algorithm to improve the search efficiency. MOELS outperformed both multiobjective heterogeneous earliest finish time (MOHEFT) and evolutionary multiobjective scheduling for cloud (EMS-C) in terms of cost and makespan minimization.

A new two-tier bipartite graph with a fuzzy clustering task allocation (2tBiFTA) method was proposed by Gad-Elrab et al. [55] to jointly optimize the makespan and monetary cost for workflow tasks in the cloud-fog environment while considering the task delay constraints. The most optimal node for each set of tasks is determined in the first tier, followed by allocating the most optimal quota. It outperformed the greedy for cost (GFC), Hybrid Heterogeneous Earliest Finish Time (HEFT), CCSH, and CMaS methods in terms of timespan and cost.

On the other hand, there has been many research work utilized the meta-heuristic evolutionary algorithms to solve the scheduling problem. Zhu et al [13] proposed a novel multi-objective scheduling for cloud (EMS-C) approach based on the well-known non-dominated sorting genetic algorithm (NSGA-II) framework to optimize the execution time and cost. However, they considered only homogenous VMs with invariant and constant performance.

Choudhary et al. [56] proposed a Hybrid Gravitational Search Algorithm (HGSA) for bi-objective scheduling workflows to reduce the makespan as well as the cost of execution. The simulation results show the effectiveness of the HGSA over standard Gravitational Search Algorithm (GSA), Hybrid Genetic Algorithm (HGA), and the Heterogeneous Earliest Finish Time (HEFT).

To utilize the benefit of both Tabu search and Harmony search, Alazzam et al [57] proposed a hybrid Tabu–Harmony task scheduling algorithm in cloud computing to maximize

throughput and to minimize Makespan and Total Cost. Although their hybrid method outperformed Tabu search, Harmony search, and round-robin in terms of makespan and cost; however, it has not been assessed against the state of the art methods.

An improved particle swarm optimization (IPSO) was proposed by Xu et al [58] to find the optimal scheduling solution for workflow tasks by considering both makespan and cost in the cloud-fog environment. The inertia weight of PSO was formulated as a nonlinear function to balance the search capability of particles. The IPSO outperformed the original PSO while shown a similar performance in terms of economic cost.

An ant colony optimization (ACO) was proposed by Lal et al [59] as a meta-heuristics method to minimize the total execution cost and total execution time. Various workflow data sets were used for ACO evaluation and the obtained results proved the efficiency of ACO in comparison with GA. However, ACO has shown a longer response time.

Xiao et al [60] provided a toolkit for assessing and managing resources and tasks using various algorithms such as (GA, PSO, RR, FCFS, MaxMin, and MinMin) and they focused on three objectives (Time, Cost, and energy). However, their work has restricted the traditional algorithms without showing the data sets used in their experiments.

Abdullahi et al [61] proposed a chaotic symbiotic organism search (CMSOS) algorithm to solve the multiobjective large-scale task scheduling optimization problems on the IaaS cloud computing environment. The initial population is generated by the chaotic optimization strategy to ensure the diversity among organisms for global convergence. The CMSOS algorithm ensured optimal trade-offs between execution time and cost in comparison with the state of the art methods.

To maximize the resource reliability for workflow task scheduling in the cloud, and Intelligent Water Drop and Genetic (IWD-GA) was proposed by Kalra et al. [62]. It aims to minimize both Makespan and execution Cost while put enough consideration to maximize the resource reliability under budget and deadline constraints. Four well known scientific workflows were used to assess the performance IWD-GA using hypervolume and coverage ratio metrics. IWD-GA outperformed both NSGA-II and hybrid PSO (HPSO) algorithms in terms of accuracy (coverage ratio) and more diversity (hypervolume).

To prove the scalability of cloud infrastructure against other distributed architectures, Hafsi et al. [63] proposed the Nondominated Sorting Genetic Algorithm version 2 (NSGA-II) in hybrid large scale distributed systems to optimize cost and makespan for workflow tasks. The adopted NSGA-II considered the dynamic provision for resources and the results shown the high scalability of cloud in comparison with other distributed systems.

Xie et al. [64] proposed a novel directional and non-local-convergent particle swarm optimization (DNCPSO) in a cloud edge environment to find a tradeoff between makespan and cost for workflow tasks executed in the cloud-edge environment. DNCPSO adopted the standard PSO algorithm by employing nonlinear inertia weight with selection and mutation to reduce the local optima problem of the PSO

algorithm. DNCPSO has shown a better performance than the traditional and improved PSO in reducing the makespan and economic cost. Table 2 summarizes the scheduling schemes in terms of methodology, environment, task type, optimization objectives, and constraints in Workflow.

TABLE 2.
A Comparison of the existing workflow scheduling schemes.

Reference	Year	Algorithm	Environment	Optimization Criteria	Constraints
Bittencourt et al [47]	2011	HCOG	public cloud	Cost	deadline
Szabo and Kroeger [41]	2012	single and multi-objective evolutionary algorithms	public clouds.	Cost and Makespan	NA
Durillo et al [48]	2013	MOHFET	commercial Amazon EC2 Cloud	Cost and Makespan	NA
Zhu et al [13]	2015	EMO Frameworks	Cloud	Cost and Makespan	NA
Xu et al [50]	2016	MTCT	Cloud	Cost and Makespan	NA
Casas et al [51]	2017	BaRRS	Cloud	Cost and Makespan	NA
Choudhary et al. [56]	2018	HGSA	Cloud	Cost and Makespan	NA
Mouradian et al [42]	2018	IBM CPLEX	Cloud-fog	Cost and Makespan	NA
Xu et al [58]	2018	IPSO	Cloud-fog	Cost and Makespan	NA
Lal et al [59]	2018	ACO	Cloud	Cost and Makespan	deadline
Zhou et al [44]	2019	FDHEFT	IaaS clouds	Cost and Makespan	NA
Xiao et al [52]	2019	FogWorkflowSim	Cloud Fog	Time, Cost, and energy	NA
Zhou et al [53]	2019	DCOH	hybrid clouds	Cost and Makespan	deadline
Alazzam et al [49]	2019	Tabu and Harmony	Cloud	Throughput, Makespan and Total Cost	NA
Abdullahi et al [57]	2019	CMSOS	Cloud	Cost and Makespan	NA
Wu et al [54]	2019	MOELS	Cloud	Cost and Makespan	deadline
Kalra et al [62]	2019	IWD-GA	Cloud	Cost and Makespan	deadline and budget
Mouradian et al [63]	2019	TSCP	Cloud-fog	Cost and Makespan	NA
Xie et al [64]	2019	DNCPSO	cloud-edge	Cost and Makespan	NA
Afrin et al [65]	2019	NSGA-II	Cloud	Cost and Makespan	NA
Gad-Elrab et al [55]	2020	2tBiFTA	Cloud-fog	Cost and Makespan	NA
Rafieyan et al [66]	2020	BWM and VIKOR	Cloud	Cost and Makespan	NA

V. DISCUSSIONS

The aforementioned scheduling schemes (BoT and workflow) are either dedicated to working on a single

domain, i.e. cloud or fog domain, they assumed homogenous nodes with similar characteristics and unlimited capabilities or they study different optimization objectives separately. Also, the current research work has concentrated on addressing the scheduling of a single kind of tasks separately, either BoT or workflow. In addition, most of the research work formulates the BoT scheduling as a single objective optimization problem in the cloud-fog environment. Moreover, although some research work used multiple optimization criteria for the BoT scheduling, none of them have formulated the problem as a multiobjective optimization problem, instead, they aggregated all the objectives in a single fitness function.

According to the literatures, the meta-heuristics algorithms can efficiently cope with the problem of task scheduling than the heuristic algorithms. On the other hand, formulating the problem in a multiobjective optimization fashion has shown better performance than the single objective formulations. However, the existing metaheuristic scheduling algorithms present just higher-level picture that either directly use the canonical formulations or adopt different variations, though the main focus is on repairing infeasible solutions. Thus, it might be crucial to produce problem specific heuristic operators to guide the search for optimal solution. This mainly implies exploiting the most problem specific knowledge in the best possible way, picking from Algorithm 1 the right algorithmic components, and cross-fertilizing them in the most appropriate way. In fact one can develop either an implicit or explicit combination of a metaheuristic algorithm and the problem specific heuristic operator. For example, developing a problem specific generational plan operator (*GP*) or update plan operator (*UP*) would finally produce an implicit integration for both the heuristic and metaheuristic frameworks. On the other hand, an explicit collaboration between the heuristic and metaheuristic frameworks could be developed while delivering a good set of initial solutions or improving the final set of solutions.

VI. CONCLUSIONS

This paper has introduced a systematic literature review for the former and the state of the art strategies for task scheduling in the cloud-fog environment by considering the makespan and cost as optimization criteria. According to the analysis of the scheduling techniques used for optimizing BoT and workflow task types, the task scheduling problem has been formulated as a single or multi-objective optimization problem, both heuristic and meta-heuristic methods have been used to tackle such problem. The literature gives evidence that the meta-heuristic methods have the potential to find the optimal solution for the task scheduling optimization problem. In addition, it is more efficient to formulate the task scheduling problem as a multiobjective optimization problem. However, less attention has been paid to produce a problem-specific heuristic operator that can be used to improve the search for the optimal solution. As a future step, a deep analysis on the effect of the heuristic problem specific operator on the meta-heuristic optimization techniques performance.

CONFLICT OF INTEREST

The authors have no conflict of relevant interest to this article.

REFERENCES

- [1] F. F. Moghaddam, M. Ahmadi, S. Sarvari, M. Eslami, and A. Golkar, "Cloud computing challenges and opportunities: A survey," in *2015 1st International Conference on Telematics and Future Generation Networks (TAFGEN)*, 2015, pp. 34–38.
- [2] A. A. Laghari, H. He, I. A. Halepoto, M. S. Memon, and S. Parveen, "Analysis of quality of experience frameworks for cloud computing," *IJCSNS*, vol. 17, no. 12, p. 228, 2017.
- [3] V. Kumar, A. A. Laghari, S. Karim, M. Shakir, and A. A. Brohi, "Comparison of fog computing & cloud computing," *Int. J. Math. Sci. Comput.*, vol. 5, no. 1, pp. 31–41, 2019.
- [4] A. R. DAR, D. Ravindran, and S. Islam, "Fog-based Spider Web Algorithm to Overcome Latency in Cloud Computing," *Iraqi J. Sci.*, pp. 1781–1790, 2020.
- [5] X. Tan and B. Ai, "The issues of cloud computing security in high-speed railway," in *Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology*, 2011, vol. 8, pp. 4358–4363.
- [6] R. A. Al-Arasi and A. Saif, "Task scheduling in cloud computing based on metaheuristic techniques: A review paper.," *EAI Endorsed Trans. Cloud Syst.*, vol. 6, no. 17, p. e4, 2020.
- [7] N. Ranaldo and E. Zimeo, "Time and cost-driven scheduling of data parallel tasks in grid workflows," *IEEE Syst. J.*, vol. 3, no. 1, pp. 104–120, 2009.
- [8] X.-Q. Pham and E.-N. Huh, "Towards task scheduling in a cloud-fog computing system," in *2016 18th Asia-Pacific network operations and management symposium (APNOMS)*, 2016, pp. 1–4.
- [9] D. Rahbari and M. Nickray, "Scheduling of fog networks with optimized knapsack by symbiotic organisms search," in *2017 21st Conference of Open Innovations Association (FRUCT)*, 2017, pp. 278–283.
- [10] D. Vasiljević, "Comparison of optimization algorithms," in *Classical and Evolutionary Algorithms in the Optimization of Optical Systems*, Springer, 2002, pp. 83–88.
- [11] N. Soltani, B. Soleimani, and B. Barekatin, "Heuristic algorithms for task scheduling in cloud computing: a survey," *Int. J. Comput. Netw. Inf. Secur.*, vol. 11, no. 8, p. 16, 2017.
- [12] R. Van den Bossche, K. Vanmechelen, and J. Broeckhove, "Cost-efficient scheduling heuristics for deadline constrained workloads on hybrid clouds," in *2011 IEEE third international conference on cloud computing technology and science*, 2011, pp. 320–327.
- [13] S. Abdi, S. A. Motamedi, and S. Sharifian, "Task scheduling using modified PSO algorithm in cloud computing environment," in *International conference on machine learning, electrical and mechanical engineering*, 2014, pp. 8–9.

- [14] Z. Zhu, G. Zhang, M. Li, and X. Liu, "Evolutionary multi-objective workflow scheduling in cloud," *IEEE Trans. parallel Distrib. Syst.*, vol. 27, no. 5, pp. 1344–1357, 2015.
- [15] S. Kaur, P. Bagga, R. Hans, and H. Kaur, "Quality of Service (QoS) aware workflow scheduling (WFS) in cloud computing: A systematic review," *Arab. J. Sci. Eng.*, vol. 44, no. 4, pp. 2867–2897, 2019.
- [16] S. S. K. Kumar and P. Balasubramanie, "Dynamic scheduling for cloud reliability using transportation problem," *J. Comput. Sci.*, vol. 8, no. 10, p. 1615, 2012.
- [17] S. Bandaru and K. Deb, "Metaheuristic techniques," in *Decision Sciences*, CRC Press, 2016, pp. 693–750.
- [18] M. A. Tawfeek, A. El-Sisi, A. E. Keshk, and F. A. Torkey, "Cloud task scheduling based on ant colony optimization," in *2013 8th international conference on computer engineering & systems (ICCES)*, 2013, pp. 64–69.
- [19] A. S. Khalil and R. D. Al-Dabbagh, "A Comparative Study on Meta-Heuristic Algorithms For Solving the RNP Problem," *Iraqi J. Sci.*, pp. 1639–1648, 2019.
- [20] S. Nesmachnow, "An overview of metaheuristics: accurate and efficient methods for optimisation," *Int. J. Metaheuristics*, vol. 3, no. 4, pp. 320–347, 2014.
- [21] D. B. LD and P. V. Krishna, "Honey bee behavior inspired load balancing of tasks in cloud computing environments," *Appl. Soft Comput.*, vol. 13, no. 5, pp. 2292–2303, 2013.
- [22] C.-W. Tsai and J. J. P. C. Rodrigues, "Metaheuristic scheduling for cloud: A survey," *IEEE Syst. J.*, vol. 8, no. 1, pp. 279–291, 2013.
- [23] T. G. Crainic and M. Toulouse, "Parallel strategies for meta-heuristics," in *Handbook of metaheuristics*, Springer, 2003, pp. 475–513.
- [24] Y. Zhang and J. Sun, "Novel efficient particle swarm optimization algorithms for solving QoS-demanded bag-of-tasks scheduling problems with profit maximization on hybrid clouds," *Concurr. Comput. Pract. Exp.*, vol. 29, no. 21, p. e4249, 2017.
- [25] R. N. Calheiros and R. Buyya, "Cost-effective provisioning and scheduling of deadline-constrained applications in hybrid clouds," in *International Conference on Web Information Systems Engineering*, 2012, pp. 171–184.
- [26] Y. Zhang, J. Sun, and J. Zhu, "An effective heuristic for due-date-constrained bag-of-tasks scheduling problem for total cost minimization on hybrid clouds," in *2016 International Conference on Progress in Informatics and Computing (PIC)*, 2016, pp. 479–486.
- [27] S. Abdi, L. PourKarimi, M. Ahmadi, and F. Zargari, "Cost minimization for bag-of-tasks workflows in a federation of clouds," *J. Supercomput.*, vol. 74, no. 6, pp. 2801–2822, 2018.
- [28] S. G. Domanal and G. R. M. Reddy, "An efficient cost optimized scheduling for spot instances in heterogeneous cloud environment," *Futur. Gener. Comput. Syst.*, vol. 84, pp. 11–21, 2018.
- [29] Y. Zhang, J. Zhou, and J. Sun, "Scheduling bag-of-tasks applications on hybrid clouds under due date constraints," *J. Syst. Archit.*, vol. 101, p. 101654, 2019.
- [30] Z. Tong, K. Li, Z. Xiao, and X. Qin, "H2ACO: An optimization approach to scheduling tasks with availability constraint in heterogeneous systems," *J. Internet Technol.*, vol. 15, no. 1, pp. 115–124, 2014.
- [31] D. Zeng, L. Gu, S. Guo, Z. Cheng, and S. Yu, "Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system," *IEEE Trans. Comput.*, vol. 65, no. 12, pp. 3702–3712, 2016.
- [32] Y. Zhang, J. Sun, and Z. Wu, "An heuristic for bag-of-tasks scheduling problems with resource demands and budget constraints to minimize makespan on hybrid clouds," in *2017 fifth international conference on advanced cloud and big data (CBD)*, 2017, pp. 39–44.
- [33] S. Abdi, L. PourKarimi, M. Ahmadi, and F. Zargari, "Cost minimization for deadline-constrained bag-of-tasks applications in federated hybrid clouds," *Futur. Gener. Comput. Syst.*, vol. 71, pp. 113–128, 2017.
- [34] S. Bitam, S. Zeadally, and A. Mellouk, "Fog computing job scheduling optimization based on bees swarm," *Enterp. Inf. Syst.*, vol. 12, no. 4, pp. 373–397, 2018.
- [35] H. Xuan, S. Wei, Y. Li, and H. Guo, "Off-line time aware scheduling of bag-of-tasks on heterogeneous distributed system," *IEEE Access*, vol. 7, pp. 104777–104788, 2019.
- [36] L. Thai, B. Varghese, and A. Barker, "Executing bag of distributed tasks on the cloud: Investigating the trade-offs between performance and cost," in *2014 IEEE 6th International Conference on Cloud Computing Technology and Science*, 2014, pp. 400–407.
- [37] Y. Nan, W. Li, W. Bao, F. C. Delicato, P. F. Pires, and A. Y. Zomaya, "Cost-effective processing for delay-sensitive applications in cloud of things systems," in *2016 IEEE 15th international symposium on network computing and applications (NCA)*, 2016, pp. 162–169.
- [38] S. Sindhu and S. Mukherjee, "An evolutionary approach to schedule deadline constrained bag of tasks in a cloud," *Int. J. Bio-Inspired Comput.*, vol. 11, no. 4, pp. 229–238, 2018.
- [39] B. M. Nguyen, H. Thi Thanh Binh, and B. Do Son, "Evolutionary algorithms to optimize task scheduling problem for the IoT based bag-of-tasks application in cloud-fog computing environment," *Appl. Sci.*, vol. 9, no. 9, p. 1730, 2019.
- [40] P. Sun, Z. Cai, and D. Liu, "Budget Constraint Bag-of-Task Based Workflow Scheduling in Public Clouds," in *CCF Conference on Computer Supported Cooperative Work and Social Computing*, 2019, pp. 243–260.
- [41] N. Parlavantzas *et al.*, "A service-based framework for building and executing epidemic simulation applications in the cloud," *Concurr. Comput. Pract. Exp.*, vol. 32, no. 5, p. e5554, 2020.
- [42] D. Tychalas and H. Karatza, "A scheduling algorithm for a fog computing system with bag-of-tasks jobs: Simulation and performance evaluation," *Simul. Model. Pract. Theory*, vol. 98, p. 101982, 2020.
- [43] P. Lu, G. Zhang, Z. Zhu, X. Zhou, J. Sun, and J. Zhou, "A review of cost and makespan-aware workflow scheduling in clouds," *J. Circuits, Syst. Comput.*, vol. 28, no. 06, p. 1930006, 2019.

- [44] E. Deelman *et al.*, “Pegasus: A framework for mapping complex scientific workflows onto distributed systems,” *Sci. Program.*, vol. 13, no. 3, pp. 219–237, 2005.
- [45] T. Fahringer *et al.*, “ASKALON: A grid application development and computing environment,” in *The 6th IEEE/ACM International Workshop on Grid Computing, 2005.*, 2005, pp. 10–pp.
- [46] L. F. Bittencourt and E. R. M. Madeira, “HCOO: a cost optimization algorithm for workflow scheduling in hybrid clouds,” *J. Internet Serv. Appl.*, vol. 2, no. 3, pp. 207–227, 2011.
- [47] C. Szabo and T. Kroeger, “Evolving multi-objective strategies for task allocation of scientific workflows on public clouds,” in *2012 IEEE Congress on Evolutionary Computation*, 2012, pp. 1–8.
- [48] J. J. Durillo and R. Prodan, “Multi-objective workflow scheduling in Amazon EC2,” *Cluster Comput.*, vol. 17, no. 2, pp. 169–189, 2014.
- [49] H. Xu, B. Yang, W. Qi, and E. Ahene, “A Multi-objective Optimization Approach to Workflow Scheduling in Clouds Considering Fault Recovery,” *KSII Trans. Internet Inf. Syst.*, vol. 10, no. 3, 2016.
- [50] I. Casas, J. Taheri, R. Ranjan, L. Wang, and A. Y. Zomaya, “A balanced scheduler with data reuse and replication for scientific workflows in cloud computing systems,” *Futur. Gener. Comput. Syst.*, vol. 74, pp. 168–178, 2017.
- [51] 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,” *Futur. Gener. Comput. Syst.*, vol. 93, pp. 278–289, 2019.
- [52] J. Zhou, T. Wang, P. Cong, P. Lu, T. Wei, and M. Chen, “Cost and makespan-aware workflow scheduling in hybrid clouds,” *J. Syst. Archit.*, vol. 100, p. 101631, 2019.
- [53] Q. Wu, M. Zhou, Q. Zhu, Y. Xia, and J. Wen, “Moels: Multiobjective evolutionary list scheduling for cloud workflows,” *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 1, pp. 166–176, 2019.
- [54] A. A. A. Gad-Elrab and A. Y. Noaman, “A two-tier bipartite graph task allocation approach based on fuzzy clustering in cloud–fog environment,” *Futur. Gener. Comput. Syst.*, vol. 103, pp. 79–90, 2020.
- [55] A. Choudhary, I. Gupta, V. Singh, and P. K. Jana, “A GSA based hybrid algorithm for bi-objective workflow scheduling in cloud computing,” *Futur. Gener. Comput. Syst.*, vol. 83, pp. 14–26, 2018.
- [56] H. Alazzam, E. Alhenawi, and R. Al-Sayyed, “A hybrid job scheduling algorithm based on Tabu and Harmony search algorithms,” *J. Supercomput.*, vol. 75, no. 12, pp. 7994–8011, 2019.
- [57] R. Xu *et al.*, “Improved particle swarm optimization based workflow scheduling in cloud-fog environment,” in *International Conference on Business Process Management*, 2018, pp. 337–347.
- [58] A. Lal and C. R. Krishna, “Critical path-based ant colony optimization for scientific workflow scheduling in cloud computing under deadline constraint,” in *Ambient Communications and Computer Systems*, Springer, 2018, pp. 447–461.
- [59] X. Liu *et al.*, “FogWorkflowSim: an automated simulation toolkit for workflow performance evaluation in fog computing,” in *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2019, pp. 1114–1117.
- [60] M. Abdullahi, M. A. Ngadi, S. I. Dishing, and B. I. Ahmad, “An efficient symbiotic organisms search algorithm with chaotic optimization strategy for multi-objective task scheduling problems in cloud computing environment,” *J. Netw. Comput. Appl.*, vol. 133, pp. 60–74, 2019.
- [61] M. Kalra and S. Singh, “Multi-criteria workflow scheduling on clouds under deadline and budget constraints,” *Concurr. Comput. Pract. Exp.*, vol. 31, no. 17, p. e5193, 2019.
- [62] C. Mouradian, S. Kianpisheh, M. Abu-Lebdeh, F. Ebrahimnezhad, N. T. Jahromi, and R. H. Glitho, “Application component placement in NFV-based hybrid cloud/fog systems with mobile fog nodes,” *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1130–1143, 2019.
- [63] Y. Xie *et al.*, “A novel directional and non-local-convergent particle swarm optimization based workflow scheduling in cloud–edge environment,” *Futur. Gener. Comput. Syst.*, vol. 97, pp. 361–378, 2019.
- [64] C. Mouradian, S. Kianpisheh, and R. H. Glitho, “Application component placement in NFV-based hybrid cloud/fog systems,” in *2018 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*, 2018, pp. 25–30.
- [65] M. Afrin, J. Jin, A. Rahman, Y.-C. Tian, and A. Kulkarni, “Multi-objective resource allocation for Edge Cloud based robotic workflow in smart factory,” *Futur. Gener. Comput. Syst.*, vol. 97, pp. 119–130, 2019.
- [66] E. Rafieyan, R. Khorsand, and M. Ramezanzpour, “An adaptive scheduling approach based on integrated best-worst and VIKOR for cloud computing,” *Comput. Ind. Eng.*, vol. 140, p. 106272, 2020.