SLUJST
Sule Lamido University Journal of Science & Technology Vol. 8 No. 2 [Dec, 2024], pp. 69-82
https://doi.org/10.56471/
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

Multi-Objective Scheduling Of Workflow Applications in Cloud Environment Using a Fair Look Ahead Particle Swarm Optimization

Amina Sabiu Sullubawa[1], Faruku Umar Ambursa [2]*

1. *Dept. of Computer Science, Bayero University Kano, Nigeria,* ameenasabiu@gmail.com
2. *Dept. of Information Technology, Bayero University Kano, Nigeria,* fuambursa.it@buk.edu.ng

*Corresponding author

Abstract

Background: *The emergence of cloud computing as the latest paradigm in distributed systems offers the opportunity to run scientific applications in form of a workflow. Workflow scheduling plays a vital role in providing different QoS satisfaction to users. However, most existing workflow scheduling approaches consider few QoS objectives. Recently, an efficient workflow scheduling algorithm was proposed based on a novel Look-ahead PSO and min-max technique (LAPSO) to schedule workflows with six QoS objectives that include; time, cost, reliability, availability, security, and reputation. The algorithm aimed at reducing QoS violation and improving QoS satisfaction of user applications. However, LAPSO schedules a single workflow application at a time, based on first-in first-out policy, which renders the algorithm incapable of providing fair handling of QoS demands of multiple competing workflow applications.* ***Aim:*** *In this work, we extend the capability of LAPSO to fairly handle multiple workflows at a time.* ***Method:*** *Specifically, we propose a fair selection algorithm based on particle swarm optimization and a look ahead heuristic (FLAPSO) that uses QoS demands of applications to prioritize and fairly distributes compute resources among participating workflows.* ***Results:*** *Simulation results showed that the proposed method outperformed the benchmark work in terms of unfairness, in all the considered test cases. It achieved about 35% and 57% decrease in unfairness in the tight and moderate cases, respectively.* ***Conclusions:*** *The proposed FLAPSO, when simulated under different services settings, outperformed the benchmark LAPSO in all the settings. Further experiments can be performed with different scale of service settings.*

Keywords: *Scheduling, Multiple workflows, Fairness, Multi-objective scheduling, cloud computing.*

1. Introduction

Distributed computing systems like clouds continue to evolve to support different kinds of scientific applications, particularly scientific workflows, with dependable, reliable persistent and cheap access to geographically-distributed computational capacities (Wang, Jiang, Xia, Wu, & Luo, 2018). People are now moving away from traditional computing to cloud as it proves higher reliability, fault tolerance, broad network access and on-demand usage (Parveen & Kumar, 2014). Cloud computing is defined as a model that deals with computations, software, data access, and storage as well as services that may not require the end-user's knowledge of the physical location and the form of the system that is providing the services (Jadeja & Modi., 2012). The cloud computing services are; software as-a-Service (SaaS) it illuminates the need to install and run application on the user's PC examples are Google Docs and Salesforces. Platform as a Service (PaaS) provides a computing platform using the cloud infrastructure. It has all the application typically needed by the customers deployed on it, hence the customers need not buy and install the software and the hardware required for it, Google App Engine and Microsoft Azure4 are the famous examples. Infrastructure-as-a-Service (IaaS) provides the required infrastructure in form of a service, examples are Amazon Web Services EC2 and S3 (Jadeja& Modi., 2012; Puthal, Sahoo, Mishra, & Swain, 2015). The customers need not buy the required servers, network resources or the data center. The cloud offers four delivery models to its client namely private cloud, public cloud, community cloud and hybrid cloud (Jadeja&

Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

Modi., 2012; Puthal, Sahoo, Mishra, & Swain, 2015; Nazir, 2012). Cloud computing emerges the best and most promising solution for providing a flexible on-demand computing infrastructure over the Internet for large scale scientific workflow and workflow application based (Wang et al., 2018).

Generally, a workflow is a flow of complex tasks that are bonded together through dependencies. Workflow applications are usually represented as a DAG (directed acyclic graph). A DAG (D = T, E) is a graph having dependencies between the tasks, in which no child task can be executed until all its parent tasks have completed their execution successfully (Simranjit Kaur et al., 2018). Workflow scheduling is an essential part of cloud computing and based on different criteria its decides the cost, execution time and performance. Workflow scheduling is an NP-complete optimization problem (Wang et al., 2018; Simranjit Kaur et al., 2018; Hamid &Arabnejad,2014). For such Problems, no known algorithms are able to generate the optimal solution within a polynomial time. The problem can be described as Let T be a finite set of task Ti (I <=i<=n) and S be a finite set of schedules Si (I <=i<=n), for each containing the entire task from set T. The best workflow Si is chosen based on the constraint or objectives. The objectives have been specified by the users as per their requirement for workflow scheduling.

Numerous works have been proposed for handling the problem of scheduling a single workflow application on multiple resources (Asghari Alaie, Hosseini Shirvani & Rahmani, 2023; Perotin, Kandaswamy, Sun, & Raghavan. 2024; Huixian & Xuewen, 2022; Wang et al., 2018; Topcuoglu& Hariri, 2002; Tao, Hui-youChang, YangYi, Chun-qinGu, & Wen-jieLi, 2011). However, many works consider single objective (Topcuoglu& Hariri ,2002; LF & Madeira, 2015). Most of the researchers (Hamid & Arabnejad, 2014; Arun &Ravichandran, 2015; Zhenzhen Xu, 2017; Hilman, Rodriguez, & Buyya, 2015; Xu, Yang, Qi, & Ahene, 2016;Li, Chen, Zhan, Du, & Zhang, 2015; Liu, Pacitti, Valduriez, Oliveira, & Mattoso, 2016; Zhaomeng & Zhu, 2015) considered two objectives, while (Zeedan, Attiya, & El-Fishawy, 2023;Wang et al., 2018;Mainak & Santanu, 2017; Yuxin &Wang, 2017; Ramirez-Alcaraz, 2012; b, Caob, c, Khand, & Hwange, 2013; Yassa, Chelouah, Kadima, & Granado, 2013;Garg & Singh, 2013) considered three objectives. Only a few works considered multiple objectives (Faruku, Rohaya, Azizol, & Shamala, 2017; Q. Tao et al., 2011) where 6 objectives are considered namely time, cost, reliability, availability, security, and reputation.

Schedulers that assign resources widely to single workflow are not work preserving as they may be forced to leave gaps in their schedule because of the priority constraint in the workflows and this will cause wastage of resources and higher cost to the users. Various scheduling methods have been reported on how to solve the problem of scheduling a single or multiple workflow application in a cloud environment. However, despite the significant contribution, several challenges are left unsolved.

Many multi-workflow scheduling algorithms exists in the literature (Hanoosh et. al.,2024). However, these algorithms considered few QoS objectives. On the other hand, the current multi-objective scheduling approaches mostly schedule single workflow at a time. Recently, LAPSO was proposed to schedule single workflow while considering six QoS objectives (Ambursa et al., 2017). However, LAPSO selects workflows based on First In First Out policy. In addition, the algorithm assumes the same level of QoS requirements for all user workflows. According to the policy the levels of the QoS requirements of all the workflows are either in a moderate or strict range. It does not take into consideration the differences among workflow jobs in terms of degrees of QoS requirements. Therefore, in performing scheduling of workflows the algorithm selects the candidate workflows sequentially based on first in first out policy (FIFO). In reality, however, Web services environment consists of users with differing degrees of QoS requirements for their workflows. Specifically, some users have strict QoS requirements and therefore demand high quality services, whilst the requirements of others could belong to moderate range, demanding services with moderate quality. Therefore, by LAPSO policy, user jobs with strict QoS requirements might end up being scheduled to services that will eventually result to high violation of their QoS requirements.

The remainder of this paper is organized as follows Section 2 discusses related works, Sect. 3 highlights the description of the system model, Sect. 5 presents a description of the scheduling problem and the proposed

SLU JST
Sule Lamido University Journal of Science & Technology Vol. 8 No. 2 [Dec, 2024], pp. 69-82
https://doi.org/10.56471/
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

scheduling models. The PSO-based scheduling optimization with constraint handling and look-ahead task remapping heuristic is presented in Sect. 6, Sect 7 presents the performance results. Finally, Sect. 8 concludes the paper and suggests future work.

2. Related Works

In this section, the analyses of the related research works are presented, its features and the underlying approaches are described. Their strengths and weakness are also highlighted. It also contains a table that summarizes the comparison of various schemes in terms of multi QoS scheduling optimization and application model.

A multi-objective optimization multi workflow scheduling approach based on dynamic game-theoretic model aiming at reducing workflow make-spans, reducing total cost and maximizing system fairness in terms of workload distribution among heterogeneous cloud virtual machines (VMs) was presented by (Wang et al., 2018). In this study, the multi-stage dynamic game theory was applied to deal with the conflicts and competition among multiple optimization objectives for the multi workflow scheduling problem. The optimization objectives can be seen as players in the multi-stage dynamic game model, and the players are usually assumed to be fully rational. The game equilibrium solutions can be obtained as the optimal results. However, they considered limited QoS objectives.

A rotary chaotic particle swarm optimization (RCPSO) was presented by (Q. Tao et al., 2011) to solve the trustworthy scheduling of a grid workflow. Likewise, a trustworthy scheduling model, in which multi- QoS parameters are considered including time, cost, reliability, availability, security, and reputation was also presented. However, RCPSO did not guide the PSO towards a good region, in situation of tight constraints when a few feasible solutions are present in search space. The algorithm also involves slow optimization methods.

The work conducted by Ambursa et al.,2017 to address the issues of RCPSO algorithm is to propose a constraint aware multi QoS workflow scheduling strategy based on particle swarm optimization and a proposed look ahead heuristic (LAPSO) to improve performance in moderate and tight situations. The algorithm selects the best scheduling solutions based on the proposed constraint-handling strategy. It hybridizes PSO with a novel look-ahead mechanism based on a min-max heuristic, which deterministically improves the quality of the best solutions. This algorithm was made up of two steps of the multi-objective formulation; and solution evaluation and selection steps. The Multi-QoS scheduling model is made up of six QoS objectives, including; time, cost, reliability, availability, security, and reputation. In solution and evaluation steps, two models are used cumulative QoS violation rate which is defined as the sum of violation degree of user defined constraints (Ambursa et al., 2017).

The lower the value of V_Ø the better, since low value means the solution is approaching feasible region. The solution is feasible if it reaches 0% CVR.

The supremacy of LAPSO over the other previous works (Ambursa et al., 2017), such as it considers six QoS objectives, it significantly reduces the CPU time by about 75% compared to RCPSO algorithm. It also ensures satisfaction of end users (0% violation) QoS constraints in both tight and moderate situations. Generally, the results reveal the robustness and efficiency of the method in that it is capable of satisfying various QoS requirement settings of users. These attribute makes it superior and closer to this work. However, the algorithm schedules single workflow. It also assumes the same level of QoS requirements for all user workflows. According to the LAPSO policy, the levels of the QoS requirements of all the workflows are either in a moderate or strict range. It does not take into consideration the differences among workflow jobs in terms of degrees of QoS requirements. Therefore, in performing scheduling of workflows the algorithm selects the candidate workflows sequentially based on first

SLUJST
Sule Lamido University Journal of Science & Technology Vol. 8 No. 2 [Dec, 2024], pp. 69-82
https://doi.org/10.56471/
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

in first out policy (FIFO). In reality, however, web service environment consists of users with differing degrees of QoS requirements for their workflows. Specifically, some users have strict QoS requirements and therefore demand high quality services, whilst the requirements of others could belong to moderate range, demanding services with moderate quality. Therefore, by LAPSO policy, user jobs with strict QoS requirements might end up being scheduled to services that will eventually result to high violation of their QoS requirements. Recently, LAPSO was improved in terms of its solution enhancement at the end of every iteration (Ambursa, 2023). However, the proposed improvement is still based on sequential workflow scheduling.

On the other hand, in (Amin et al., 2018) the authors proposed an approach for scheduling multiple workflows that takes fairness into account with three different QoS objectives and four workflow selection policies based on unfairness metric. The selection policies include: ordered workflow, round-robin, weighted round-robin and direct heuristic. In ordered policy, a unique number is assigned to all workflows, in every iteration the unfinished workflow with lowest number is selected. Round robin, the scheduler maintains a queue of unfinished workflows iterating over and over until all workflows are selected. In weighed round robin a workflow with largest gap ratio is selected for scheduling. Lastly, direct heuristic policy selects workflow that minimizes the value of unfairness metric iterating over the workflows until all workflows are selected for scheduling. The best workflow selection policy to reduce unfairness and ensure fair schedule is direct policy (Amin et al.,2018). This makes it nearer and relevant to this work. However, this approach takes into account limited QoS objectives. Additionally, the unfairness is expressed as a distribution of gain in saving and speedup among the participating workflows. In our work, we measure unfairness from the view point of distribution of requirements violation among the competing workflows.

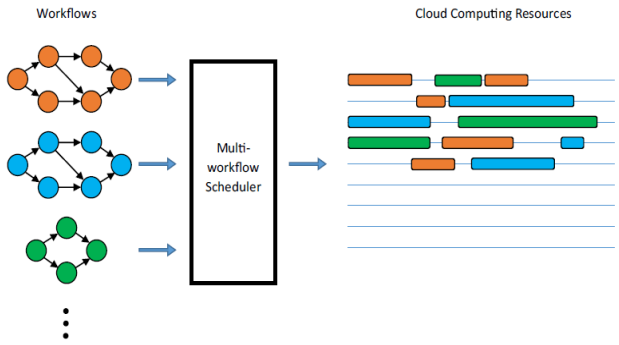

Figure 1: System model with multi-workflow scheduler (Amin et al.,2018)

Other multi-workflow scheduling algorithms have been developed recently (Chakravarthi, Neelakantan, Shyamala, & Vaidehi, 2022; Sun, Huang, Li, Gu, Xie, & Qian, 2023; Li, Xu, Chen, Huang, Xia, & Chai, 2023; Li, Tian, Xu, Abreu, Huang, Chai, & Xia, 2024). However, few objectives were still considered in the models.

The contribution of this work are; an enhanced LAPSO (FLAPSO) capable of scheduling multiple workflows is proposed. The algorithm determines levels of QoS requirements of users by categorizing the candidate jobs based on the QoS ranges defined by LAPSO. It then order the workflows based on degree of QoS demand. Therefore, in each iteration, workflow with highest QoS requirements is scheduled first. The idea is to ensure that users with high QoS demand are serviced by appropriate services, thereby minimizing unfairness through ensuring equal level of violation of QoS agreement among users.

Sule Lamido University Journal of Science & Technology Vol. 8 No. 2 [Dec, 2024], pp. 69-82
https://doi.org/10.56471/
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

3. Materials and methodologies

This section gives the description of the proposed method and its stages in detail.

2.1. System Model Description

In this work, we consider workflow applications (multiple workflows) that can be represented as a directed acyclic graph with node representing the computational task of the workflow and edges represent the data dependencies (data transfer) and control dependencies (order of execution) between the nodes of the applications. Figure 2 shows the system model with multi- workflow scheduler scheduling multiple workflows. Scheduling multiple workflows involve selecting and allocating resources to workflows while satisfying different QoS objectives specify by the user. A QoS requirement set by the user is called hard constraints while the metric to be minimized for each workflow is described as its soft constraints. In our work, the emphasis is on developing and integrating an efficient fairness mechanism for scheduling multiple workflow into LAPSO so that it can schedule multiple workflow at a time.

2.2. Previous Scheme Description

Look-ahead PSO (LAPSO) is a light weight constraint-aware multi QoS workflow scheduling algorithm based on particle swarm optimization with min-max strategy. The min-max approach is used to deterministically improve best PSO solution in terms of two models QoS violation and QoS satisfaction models. Cumulative QoS satisfaction rate is define as the weighted sum of the values of the six QoS metrics (Ambursa et al., 2017; Ambursa 2023). The metrics are described with Ø representing a workflow schedule. The algorithm was made up of two steps of multi-objective formulation; solution evaluation and selection(CSS) steps. The Multi-QoS scheduling model is made up of six QoS constraint time, cost, reliability, availability, security, and reputations. In solution and evaluation steps, a Constraints –aware solution evaluation and selection algorithm (CSS) was proposed evaluation algorithm to reduces violation of user's constraints by using BTS method (binary tournament selection) based on two models QoS satisfaction model and cumulative QoS violation rate (Ambursa et al., 2017). The operation of LAPSO algorithm is shown in Algorithm 1. The lower the value of V_Ø the better, since low value means the solution is approaching feasible region. The solution is feasible if it reaches 0% CVR.

The major advantages of LAPSO over the other previous works are that: it considers six QoS objectives, it reduces the number of PSO iterations and also reduces CPU time, it ensures satisfaction of end users (0% violation) QoS constraints. However, the algorithm schedules single workflow at a time, it also follows a sequential order of scheduling based on First In First Out (FIFO) policy and lacks a fairness selection mechanism for scheduling multiple workflows. In cloud you can have multiple workflows waiting in a queue and many users may encounter delays or even more costly schedule plans, this will cause unfairness to the priority task. The reader is referred to the reference (Ambursa et al., 2017) for details of Algorithm 1 (LAPSO). An improvement of LAPSO is also reported in (Ambursa, 2023).

Sule Lamido University Journal of Science & Technology Vol. 8 No. 2 [Dec, 2024], pp. 69-82
https://doi.org/10.56471/
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

Algorithm 1: Look Ahead PSO (LAPSO)

foreach particle
  initialize x_{ij}^t and v_{ij}^t
  compute $f(x_i^t)$
  initialize $pBest_i^t$ with x_i^t
compute $gBest_j^t$
while stopping condition not satisfied **do** //next iteration
  foreach particle
    update v_{ij}^{t+1}
    update x_{ij}^{t+1}
    compute $f(x_i^t)$
    update $pBest_i^{t+1}$
  update $gBest_j^{t+1}$
  //decode PSO's previous and current global best particles to schedules
  $\phi_{best}^t = gBest_j^t$
  $\phi_{best}^{t+1} = gBest_j^{t+1}$
  invoke LATR Algorithm to improve the schedule ϕ_{best}^{t+1}
return ϕ_{best}^+

2.3. Proposed Fair Selection Algorithm (FLAPSO)

The algorithm determines levels of QoS requirements of users by categorizing the candidate jobs based on the QoS ranges defined by LAPSO. It then orders the workflows based on degree of QoS demand using the formula in Equation 1. The model is adopted from (Ambursa et al., 2017). Therefore, in each iteration, a workflow with highest QoS requirements is scheduled first.

Algorithm 2: Proposed Fair Look-Ahead PSO Algorithm**.**

Input: QoS requirement, Workflows W
Output: Q_W
for-each w_i
  Compute priority $\boldsymbol{QD}_{w_i}$ using equation 2
  Order a W based on $\boldsymbol{QD}_{w_i}$
    for each iteration
      select w_i with highest $\boldsymbol{QD}_{w_i}$
      Invoke LAPSO algorithm to schedule w_i
  End
End
return Q_W

The idea is to ensure that users with high QoS demand are serviced by appropriate services, it also lets short jobs complete within a reasonable time while not starving long jobs thereby minimizing unfairness through ensuring equal level of violation of QoS requirement among users. The operation of the proposed FLAPSO algorithm is presented in Algorithm 2.

SLU JST
Sule Lamido University Journal of Science & Technology Vol. 8 No. 2 [Dec, 2024], pp. 69-82
https://doi.org/10.56471/
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

$$QD_{\emptyset} = w1\left(1-\frac{M_{\emptyset}}{D}\right)+w2\left(1-\frac{C_{\emptyset}}{B}\right)+w3\left(\frac{R_{\emptyset}}{R}-1\right)+w4\left(\frac{A_{\emptyset}}{A}-1\right)+w5\left(\frac{Sec_{\emptyset}}{S}-1\right)+w6\left(\frac{Rep_{\emptyset}}{R}-1\right) \quad (1)$$

Where:
$\boldsymbol{M}_{\emptyset}$= Minimum execution time
D= Deadline
$\boldsymbol{C}_{\emptyset}$= minimum cost
B= Budget
$\boldsymbol{R}_{\emptyset}$= maximum Reliability
R= Reliability
$A_{\emptyset}$= Maximum Availability
A= Availability
$Sec_{\emptyset}$= Maximum security
S= security
$Rep_{\emptyset}$= Maximum reputation
R= Reputation

In order, to achieve fairness the method in (Amin et al., 2018) is adopted and modified to carry out the task. The model proposed by (Amin et al., 2018) is shown in Equation 2 below;

$$U = \frac{\sum_{w\epsilon}W_B|S_w-\bar{A}|+\sum_{w\epsilon}W_D|E_w-\bar{A}|}{|W|} \quad (2)$$

Where W_B= budget workflow

W_B = Deadline workflow

$\bar{A}$ = average resource sharing

S_w = speedup of deadline

E_w = partial savings of budget

However, in the above unfairness model only two QoS objectives were taking into account: time and cost. And additionally, the unfairness is expressions of distribution of gain in saving and speedup among the participating workflows. In our work slightly, we measure unfairness from the view point of distribution of requirements violation among the competing workflows. Our scheme uses two scheduling models: violation model and unfairness model. The violation model handles the hard requirement of users, while satisfaction models handle the user soft requirement. These two models are adopted from the work by (Ambursa et al.,2017). Meanwhile the unfairness is to be defined relative to requirements violation.

First we determine the violation model, let σ and Ω respectively denotes a QoS dimension set of all QoS dimensions and $\mathcal{W}\sigma$ a QoS satisfaction rate in dimension σ. $\mathcal{W}\sigma$ represents the percentage satisfaction of constraints in dimension σ. let$\delta_{\pi\sigma}$ represent a violation rate for a constraint in dimensionσ. The term$\delta_{\pi\sigma}$is depicted in Equation 3 below:

$$\delta_{\pi\sigma} = \begin{cases} \frac{\pi\sigma-\mathcal{W}\sigma}{\gamma_{\pi\sigma}}, if\ \mathcal{W}\sigma < (\text{f or } \sigma_{max}) \\ \frac{\mathcal{W}\sigma-\pi\sigma}{\gamma_{\pi\sigma}}, if\ \mathcal{W}\sigma > \pi\sigma\ (\text{f or } \sigma_{min} \\ 0, otherwise \end{cases} \quad (3)$$

where, $\pi\sigma$ denote the constraint in the QoS dimension σ and $\gamma\pi\sigma$ is the largest violation of $\pi\sigma$ in the entire solution set. A solution is feasible if it ensures QoS satisfaction set by the users. Therefore a feasible solution has

Sule Lamido University Journal of Science & Technology Vol. 8 No. 2 [Dec, 2024], pp. 69-82
https://doi.org/10.56471/
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

zero violation for all the $\delta_{\pi\sigma}$. A workflow scheduling solution is infeasible if it has value of $\delta_{\pi\sigma}$ greater than zero. The model for the hard constraints is denoted in Equation (4):

Minimize $V_{\emptyset} = \sum_{\sigma\Omega} \Omega_{\pi\sigma}$ (4)

$V_{\emptyset} = \delta_{\pi dln} + \delta_{\pi bgt+}\delta_{\pi rel} + \delta_{\pi avl} + \delta_{\pi sec} + \delta_{\pi rep}$

s.t $V_{\emptyset}$= 0.

Where $V_{\emptyset}$ denotes the cumulative Qos violation rate (CVR) of schedule Ø , and $\delta_{\pi dln,}\delta_{\pi bgt,}\delta_{\pi rel}, \delta_{\pi avl}, \delta_{\pi sec}, \delta_{\pi rep}$ represents violations of deadline, budget, reliability, availability, security and reputation constraints respectively. The lower the value of $V_{\emptyset}$ the better, since low value means the solution is approaching feasible region. Therefore, a solution is feasible if it achieves 0% violation.

To model unfairness, we first give the following definition:

Table 1: Simulation Test Cases

Workflows	Number of Task	QoS Requirements (D, B, R, A, S, Rp)		
5 Task Application	5	Tight case	Moderate case	Loose case
10 Task Application	10	8.0, 23.0, 0.85, 5.0, 5.0	20.0, 50.0, 0.8, 0.8, 5.0,5.0	20.0, 50.0, 0.7,0.7, 5.0,4.0
Montage	20	20.0, 100.0, 0.8, 0.8, 4.5, 4.3	35.0,150.0, 0.8,0.8, 5.0, 5.0	35.0, 150.0, 0.7, 0.7, 5.0, 4.0
e-Protein Application	15	14.0, 80.0, 0.85, 0.85, 5.0, 5.0	30.0, 120, 0.8, 0.8, 5.0, 5.0	30.0, 120.0, 0.7, 0.7, 5.0, 4.0
Cybershake Application	20	28.0, 120.0, 0.85, 0.85, 5.0, 4.3	40.0,150.0, 0.8, 0.8, ,5.0, 4.3	40.0,150.0, 0.7, 0.7,5.0, 4.0
e-Business Application	36	35.0, 180.0, 0.8, 0.8, 4.5, 4.3	50.0, 200.0, 0.8, 0.8, 4.5, 4.3	50.0, 200.0, 0.7, 0.7, 5.0, 4.0

A fair solution is the one that ensures equal distribution of net violation among the participating workflows. Based on this, and a decision averaging theory we define unfairness as follows: we first determine the mean violation of all the multi workflow scheduling solution using the formula in Equation 5. Then, unfairness of a solution is defined as the mean of the absolute value of the difference between the individual workflow violation and the mean violation of all the workflows. The formula for unfairness of a solution is depicted in Equation 6. It is the proposed fair selection mechanism for scheduling multiple workflows application.

$$\bar{\omega} = \frac{\sum_{w\epsilon N} V_{\emptyset}}{N} \quad (5)$$

$$U = \frac{\sum_{w\epsilon N} |V_{\emptyset} - \bar{\omega}|}{N} \quad (6)$$

Where
N is the number of workflows
$\bar{\omega}$ is the mean violation of all multi workflows
$V_{\emptyset}$ is the cumulative QoS violation model

Sule Lamido University Journal of Science & Technology Vol. 8 No. 2 [Dec, 2024], pp. 69-82
https://doi.org/10.56471/
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

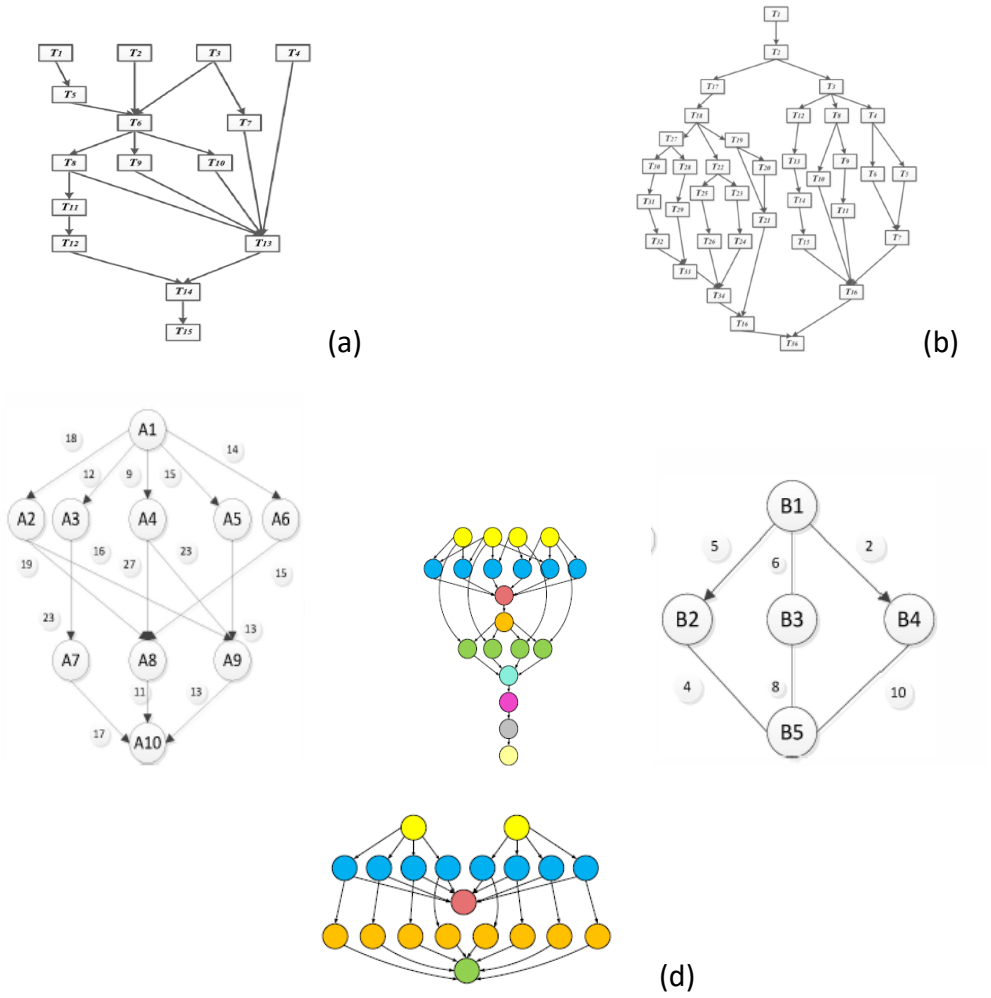

Figure 2: Six real world workflow applications

4. Performance Evaluation

The performance of the proposed algorithm is evaluated and compared with the bench mark work which is based on fairness that is minimizing the unfairness. To test the performance of the proposed fair LAPSO (FLAPSO) our scheme uses three models namely violation handler, user satisfaction handler adopted from (Ambursa et al., 2017) and unfairness handler, details of these models have been described in chapter three .The two approaches were tested using the scientific and business workflow applications of different domain such as, e-protein, e-business montage, cyber shake, epigenomics, simple DAGs with 15, 36, 20, 20, 10 and 5 tasks respectively as depicted in figure 2. To every individual tasks in the workflow, depending on the test case; a moderate scale was randomly assigned in the range 30-60 through 60 – 120 range of set of service instances as shown in Table 1. The values for the six QoS parameters of each service were generated randomly (Ambursa et al., 2017; Ambursa 2023). They maintain the rule that the service cost for the same task is dependent on the values of other parameters. The algorithm was assessed in 3 different test instances. The test instances reveal a range of different scheduling situations with different level of complication from low to moderate and tight.

SLU JST
Sule Lamido University Journal of Science & Technology Vol. 8 No. 2 [Dec, 2024], pp. 69-82
https://doi.org/10.56471/
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

4.1 Experiments Environment

This section presents the experimental environment for the proposed method such as computer resources, network topologies and the experimental setup. The resources are used to test the performance of the proposed algorithm. The experimentation study was conducted by means of simulation using NetBeans Java Programming Environments. The simulation was performed on a computer with a Celeron (R)Dual- Core CPU (1.8GHz, 2 Cores) and 2GB RAM. The operating System was MS Windows 10. The excel file was used to depict the result.

The performance metrics considered in this work are; unfairness (minimizing unfairness) it is depicted in Equation 6 and CVR (violation rate) depicted in Equation 4 that evaluate the six objectives namely cost, time, reliability, availability, security and reputation) as considered in the previous work. Details of these metrics have been described above.

4.2 Experimental Scenarios

The algorithms were tested using a e-Protein workflow application, e-Business workflow application as used in the benchmark work and four more additional scientific workflow of different domain astronomy (Montage workflow), earthquake science (Cyber Shake workflow) used in (Amin et al., 2018), and 10 tasks DAG and 5 tasks DAG used in (Yuxin et al.,2017). They are depicted in Figure 2 below.

5. Results and Discussion

This section presents the results of the proposed multi objective multi-workflow scheduling algorithm. The proposed scheme is developed, tested and compared with benchmark LAPSO. It also presents the QoS demand of each workflow, their priority and the degree of violation of each workflow. This subsection presents the overall results of the experiments. The proposed method and the adopted method have been developed and executed in the same settings to prove the effectiveness of our proposed scheme. The results of the proposed method against the LAPSO algorithm in terms of unfairness metric under different scale of service instances is presented in Table 2 through 4.

Experiment 1: Tight Case

In the tight case scenario as shown in Table 2, it can be seen that the proposed algorithm outperforms the benchmark in terms of spreading the QoS violation of the applications, where none of the applications' QoS violation goes up to 10%. In contrast, the benchmark recorded a QoS violation of e-Protein with more than 13%. This enables the proposed method achieved lower unfairness of only about 1.9%, where the benchmark recorded about 2.9%, thereby achieving about 35% decrease in unfairness.

Table 2: QoS violation and unfairness in different cases

QoS Violation (%)							
		Tight case		Moderate case		Loose Case	
S/N	Workflow Type	FLAPSO	LAPSO	FLAPSO	LAPSO	FLAPSO	LAPSO
1	Five-task	5.93	5.93	1.86	1.86	0.00	0.00
2	Ten-task	7.35	7.43	1.86	0.00	0.00	0.00
3	Montage	9.55	9.18	1.86	**3.31**	0.00	0.00
4	e-Protein	7.60	**13.72**	0.00	0.00	0.00	0.00
5	Cybershake	1.36	1.17	1.86	0.48	0.00	0.00
6	e-Business	5.85	7.80	**2.67**	**3.18**	0.00	**1.10**
	Unfairness	**1.89**	**2.69**	**0.56**	**1.31**	**0.00**	**0.31**

Sule Lamido University Journal of Science & Technology Vol. 8 No. 2 [Dec, 2024], pp. 69-82
https://doi.org/10.56471/
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

Experiment 2: Moderate Case

In the moderate situation, it can be observed, in Table 2, that the proposed algorithm also outperforms the benchmark in terms of spreading the QoS violation of the applications, where the proposed tries to provide equal distribution of the QoS violation with the five-task, ten-task, montage and cybershake applications each obtaining 1.86%. In the case of the benchmark, 2 applications suffer relatively high violation, which increased relative unfairness of the algorithm compared to the benchmark. The proposed method outperforms the compared approach by more than 57%.

Experiment 3: Loose Case

In the loose case depicted in Table 2, the proposed approach clearly achieves 0.00% violation across all the applications, which earns it 0.00 unfairness, where the compared benchmark still records violation and hence unfair.

Experiment 4: Unfairness with Different scale of services

From the simulation results, Figure 3 depicts the performance comparison of the proposed method against the LAPSO algorithm in terms of unfairness against different scale of service instances. It can be seen that under moderate situation you will realize that the unfairness is moderate as the number of services are being reduced, in the case of LAPSO it increases because it has no mechanism to select which workflow should be allocated to which service in such a way that good services are being allocated to workflows with high QoS demand.

6. Discussion

Overall, results of the proposed FLAPSO shows that unfairness is minimized even in tight situation because it has a mechanism to select an appropriate workflow to schedule in such a way that workflows that have critical requirement are scheduled first, followed by workflows with lower QoS demand. This makes it possible for FLAPSO to minimize the violation requirement of high priority demands at difficult situation, therefore the unfairness become minimal.

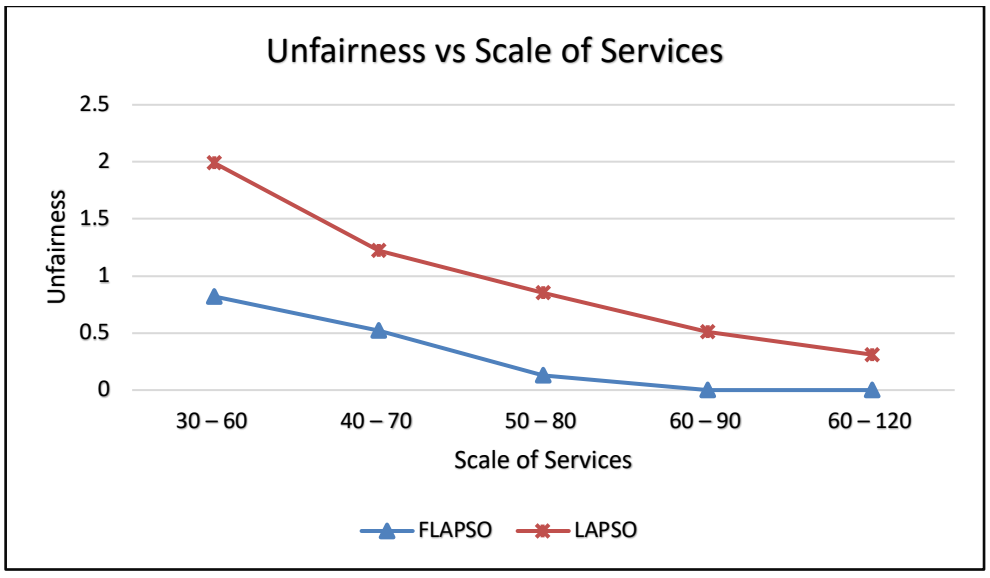

Figure 3. Unfairness metric against different scale of service instances.

Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

6. Conclusions

This paper proposes an enhanced efficient fairness mechanism FLAPSO (fairness LAPSO) for scheduling multiple workflows and ensuring fairness among the participating workflows under different scale of services setting. The algorithm determines levels of QoS requirements of users by categorizing the candidate jobs based on the QoS ranges defined by LAPSO. It then order the workflows based on degree of QoS demand. Therefore, in each iteration, workflow with highest QoS requirements is scheduled first. The idea is to ensure that users with high QoS demand are serviced by appropriate services, thereby minimizing unfairness through ensuring equal level of violation of QoS agreement among users. Results of the simulation experiments illustrated that the proposed FLAPSO significantly outperformed the benchmark LAPSO. The proposed FLAPSO when simulated under different services settings outperformed the benchmark LAPSO in all the settings, further experiments can be performed with different scale of service settings.

References

Ambursa, F. U. (2023). Improving Service Selection for Cloud Service Brokerage using Hybrid Method. Journal of Science, Technology and Education 11(2), June, 2023. http://www.atbuftejoste.net/index.php/joste/article/view/1821

Ambursa, F. U., Rohaya, L., Azizol, A., & Shamala, S. (2016). A particle swarm optimization and min–max-based workflow scheduling algorithm with QoS satisfaction for service-oriented grids. Springer Science+Business Media New York 2016. https://link.springer.com/article/10.1007/s11227-016-1901-x

Amin, R., Mahmoud, N., & J., E. D. (2018). Fair multiple-workflow scheduling with different quality-of-service goals. Springer Science+Business Media, LLC, part of Springer Nature 2018. https://link.springer.com/article/10.1007/s11227-018-2604-2

Arabnejad, H., & Barbosa, J. G. (2016). Multi-QoS constrained and Profit-aware scheduling approach for concurrent workflows on heterogeneous systems. Future Generation Computer Systems, 211-221. https://www.sciencedirect.com/science/article/pii/S0167739X16303740

Asghari Alaie, Y., Hosseini Shirvani, M., & Rahmani, A. M. (2023). A hybrid bi-objective scheduling algorithm for execution of scientific workflows on cloud platforms with execution time and reliability approach. *The Journal of Supercomputing*, *79*(2), 1451-1503. https://doi.org/10.1007/s11227-022-04703-0

Chakravarthi, K. K., Neelakantan, P., Shyamala, L., & Vaidehi, V. (2022). Reliable budget aware workflow scheduling strategy on multi-cloud environment. *Cluster Computing*, *25*(2), 1189-1205. https://doi.org/10.1007/s10586-021-03432-y.

Garg, R., & Singh, A. K. (2013). Multi-objective workflow grid scheduling using ε-fuzzy dominance sort based discrete particle swarm optimization. Springer Science+Business Media New York 2013. https://link.springer.com/article/10.1007/s11227-013-1059-8.

Hamid, A., & G., B. J. (2014). A Budget Constrained Scheduling Algorithm for Workflow Applications. Springer Science+Business Media Dordrecht 2014. https://link.springer.com/article/10.1007/s10723-014-9294-7

Hanoosh, Z. (2024). A Survey on Multiple Workflow Scheduling Algorithms in Cloud Environment. Al-Furat Journal of Innovations in Electronics and Computer. Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE), Vol. 3, No.4a, April 2024. https://doi.org/10.46649/fjiece.v3.1.4a.13.4.2024

Hilman, M. H., Rodriguez, M. A., & Buyya, R. (2015). Resource-sharing Policy in Multi-tenant Scientific Workflow as a Service Platform. JOURNAL OF LATEX CLASS FILES, VOL. 14, NO. 8,. https://arxiv.org/abs/1903.01113

Hirales-Carbajal, A., Tchernykh, A., Yahyapour, R., González-García, J. L., Röblitz, T., & Ramírez-Alcaraz, J. M. (2012). Multiple workflow scheduling strategies with user run time estimates on a grid. *Journal of Grid Computing*, *10*325-346. https://doi.org/10.1007/s10723-012-9215-6

Huixian, Q., & Xuewen, X. (2022).Multi-objective workflow scheduling based on genetic algorithm in cloud environment. Springer Nature 2022 Latex Template.https://orcid.org/0000-0002 4938-1479

Jadeja, Y., & Modi, K. (2012). Cloud computing concepts, architecture and challenges. 2012 Intenational Conference on Computing, Electronics and Electrical Technologies [ICCEET]. https://ieeexplore.ieee.org/abstract/document/6203873

Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

Kalra, M., & Singh, S. (2019). Multi-criteria workflow scheduling on clouds under deadline and budget constraints. Concurrency and Computation: Practice and Experience, 31(17), e5193.. https://doi.org/10.1002/cpe.5193

Kumar, P., & Rai, A. K. (2014). An Overview and Survey of Various Cloud Simulation Tools. Journal of Global Research in Computer Science, Volume 5, No. 1, January 2014, 24-26. https://www.rroij.com/open-access/an-overview-and-survey-of-various-cloud-simulation-tools-24-26.pdf

LF, B., & Madeira, E. (2015). HCOC: a cost optimization algorithm for workflow scheduling in hybrid cloud. J Internet Serv Appl 2(3):207–227. https://link.springer.com/article/10.1007/s13174-011-0032-0

Li, H., Xu, G., Chen, B., Huang, S., Xia, Y., & Chai, S. (2023). Dual-mutation mechanism-driven snake optimizer for scheduling multiple budget constrained workflows in the cloud. Applied Soft Computing, 149, 110966. https://doi.org/10.1016/j.asoc.2023.110966

Li, H.-H., Chen, Z.-G., Zhan, Z.-H., Du, K.-J., & Zhang, J. (2015). Renumber Coevolutionary Multiswarm Particle Swarm Optimization for Multi-objective Workflow Scheduling. Engineering Research Center of Supercomputing Engineering Software, Ministry of Education, P.R. China. https://dl.acm.org/doi/abs/10.1145/2739482.2764632

Li, H., Tian, L., Xu, G., Abreu, J. R. C., Huang, S., Chai, S., & Xia, Y. (2024). Co-evolutionary and Elite learning-based bi-objective Poor and Rich Optimization algorithm for scheduling multiple workflows in the cloud. *Future Generation Computer Systems*, *152*, 99-111. https://doi.org/10.1016/j.future.2023.10.015

Liu, J., Pacitti, E., Valduriez, P., Oliveira, D. D., & Mattoso, M. (2016). Multi-Objective Scheduling of Scientific Workflows in Multisites Clouds. Future Generation Computer Systems, Elsevier, 2016. https://www.sciencedirect.com/science/article/pii/S0167739X16300917

Mohammadzadeh, A., & Masdari, M. (2023). Scientific workflow scheduling in multi-cloud computing using a hybrid multi-objective optimization algorithm. *Journal of Ambient Intelligence and Humanized Computing*, *14*(4), 3509-3529.

Nazir, M. (2012). Cloud computing: overview & current research challenges. IOSR journal of computer engineering, 8(1), 14-22. 10.6084/M9.FIGSHARE.1145884.

Perotin, L., Kandaswamy, S., Sun, H., & Raghavan, P. (2024). Multi-resource scheduling of moldable workflows. Journal of Parallel and Distributed Computing, 184, 104792. https://doi.org/10.1016/j.jpdc.2023.104792

Puthal, D., Sahoo, B. P., Mishra, S., & Swain, S. (2015). Cloud Computing Features, Issues and Challenges: A Big Picture. 2015 International Conference on Computational Intelligence & Networks (CINE 2015). DOI: 10.1109/CINE.2015.31

Simranjit Kaur, P. B. (2018). Quality of Service (QoS) AwareWorkflow Scheduling (WFS) in Cloud. Arabian Journal for Science and Engineering.

Sun, Z., Huang, H., Li, Z., Gu, C., Xie, R., & Qian, B. (2023). Efficient, economical and energy-saving multi-workflow scheduling in hybrid cloud. *Expert Systems with Applications*, *228*, 120401.

Sun, Z., Mei, Y., Zhang, F., Huang, H., Gu, C., & Zhang, M. (2024). Multi-Tree Genetic Programming Hyper-Heuristic for Dynamic Flexible Workflow Scheduling in Multi-Clouds. *IEEE Transactions on Services Computing*. DOI: 10.1109/TSC.2024.3394691

Tao, Q., Chang, H. Y., Yi, Y., Gu, C. Q., & Li, W. J. (2011). A rotary chaotic PSO algorithm for trustworthy scheduling of a grid workflow. *Computers & operations research*, *38*(5), 824-836. https://doi.org/10.1016/j.cor.2010.09.012

Wang, D., Tan, D., & Liu, L. (2018). Particle swarm optimization algorithm: an overview. *Soft computing*, *22*(2), 387-408. https://doi.org/10.1007/s00500-016-2474-6

Wang, Y., Jiang, J., Xia, Y., Wu, Q., Luo, X., & Zhu, Q. (2018, June). A multi-stage dynamic game-theoretic approach for multi-workflow scheduling on heterogeneous virtual machines from multiple infrastructure-as-a-service clouds. In *International conference on services computing* (pp. 137-152). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-94376-3_9

Xu, H., Yang, B., Qi, W., & Ahene, E. (2016). A Multi-objective Optimization Approach to Workflow Scheduling in Clouds Considering Fault Recovery. KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS VOL. 10, NO. 3, Mar. 2016. 10.3837/tiis.2016.03.002

Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

Yassa, S., Chelouah, R., Kadima, H., & Granado, B. (2013). Multi-Objective Approach for Energy-Aware Workflow Scheduling in Cloud Computing Environments. The ScientificWorld Journal Volume 2013, Article ID 350934, 13 pages. https://www.hindawi.com/journals/tswj/2013/350934/

Yuxin Wang, S. C. (2017). Fairness Scheduling with Dynamic Priority for Multi Workflow on Heterogeneous. 2017 the 2nd IEEE International Conference on Cloud Computing and Big Data Analysis (pp. 404-409). IEEE. https://ieeexplore.ieee.org/abstract/document/7951947

Zeedan, M., Attiya, G., & El-Fishawy, N. (2023). Enhanced hybrid multi-objective workflow scheduling approach based artificial bee colony in cloud computing. Computing, 105(1), 217-247. https://doi.org/10.1007/s00607-022-01116-y

Zhaomeng Zhu, G. Z. (2015). Evolutionary Multi-Objective Workflow Scheduling in Cloud. TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS. https://ieeexplore.ieee.org/abstract/document/7127017

Zhenzhen, X., Xin, C., Xiujuan, X., Xiaowei, Z., Dai, J. J., & Mingfei. (2017). A look-ahead algorithm for online multiple workflow scheduling problem. Concurrent Engineering: Research. https://journals.sagepub.com/doi/abs/10.1177/1063293X17728763

Zhu, Z., Zhang, G., Li, M., & Liu, X. (2015). Evolutionary Multi-Objective Workflow Scheduling in Cloud. TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS. https://ieeexplore.ieee.org/abstract/document/7127017

Xia, Y., Zhan, Y., Dai, L., & Chen, Y. (2023). A cost and makespan aware scheduling algorithm for dynamic multi-workflow in cloud environment. *The Journal of Supercomputing*, *79*(2), 1814-1833. https://doi.org/10.1007/s11227-022-04681-3