ENHANCING THE SCIENTIFIC WORKFLOW SCHEDULING BY ADAPTIVE APPROACHES WITH CONVEX OPTIMIZATION IN CLOUD ENVIRONMENT

A Thesis

Submitted in partial fulfillment of the requirements for the award of the degree of

DOCTOR OF PHILOSOPHY
in
Computer Science and Engineering

By

Ramandeep Sandhu

41700047

Supervised By

Dr. KamleshLakhwani

LOVELY PROFESSIONAL UNIVERSITY

Transforming Education Transforming India

LOVELY PROFESSIONAL UNIVERSITY
PUNJAB

2021

DECLARATION

This thesis is an account of research undertaken between August 2017 and April 2021 at the Department of Computer Science and Engineering, Lovely Professional University, Phagwara, India.

Except where acknowledged customarily, the material presented in this thesis is, to the best of my knowledge, original and has not been submitted in whole or part for a degree in any university.

Ramandeep Sandhu

School of Computer Science and Engineering,

Lovely Professional University,

Phagwara, Punjab-144411, India

CERTIFICATE

This is to certify that the declaration statement made by the student is correct to the best of my knowledge and belief. She has submitted the Ph.D. thesis **ENHANCING THE SCIENTIFIC WORKFLOW SCHEDULING BY ADAPTIVE APPROACHES WITH CONVEX OPTIMIZATION IN CLOUD ENVIRONMENT** under my guidance and supervision. The present work is the result of her original investigation, effort, and study. No part of the work has ever been submitted for any other degree at any University. The Ph.D. thesis is fit for the submission and fulfilment of the conditions for the award of Ph.D. degree in Computer Science and Engineering from Lovely Professional University, Phagwara.

Dr.KamleshLakhwani

School of Computer Science and Engineering,

Lovely Professional University,

Phagwara, Punjab-144411, India

ABSTRACT

In today's progressive scientific world, researchers increasingly go for workflow applications to automate their work. Workflow applications include a sequence of tasks that enables the analysis of data in a well-structured and distributed manner. These tasks are interdependent thus leading to a huge amount of data exchange between preceding and succeeding tasks during workflow execution. Many scientific research areas such as biological engineering, ocean sciences, earthquake science, and many other promising areas include processing of workflow applications having Terabytes or Petabytes of data. Processing and analyzing such data involves more complex computational Resources.

If we talk about cloud virtual systems, it has made scheduling fast yet it needs an efficient allocation of tasks to resources. Scheduling is a process that manages the execution of tasks on different resources which is stored virtually in a distributed manner. Task scheduling in a multiprocessor environment is an essential and computationally complex problem. A multiprocessor scheduling application can be modelled as a Directed Acyclic Graph (DAG) having an objective to find a schedule to map the tasks onto processors so that the completion time of this application can be minimized. The problems considered in this research are based on the scheduling of tasks in such a manner so that makespan, cost as well as response time can be minimized. Over the last few years, cloud computing has become very popular as its potentials significant cost reductions. It has taken a long stride towards success in providing maximum throughput as well as high qualitative services to its consumers. It is one of the latest and recently challenged areas in the modern era which provides easy access to information technology-based resources to varied customers, in the form of infrastructure, software, and platform level as per the end-users specific requirements. In a cloud environment, users need not buy the infrastructure for

various computing services but can access computers in any part of the world. Apart from this, a scheduling mechanism plays a crucial role in assigning the tasks to optimal resources to achieve the final goal of significantly reducing both the makespan and execution cost.

It is possible to impose two types of algorithms in a cloud environment which are Heuristic and Meta-Heuristic. HEFT which comes under the category of heuristic algorithms is applicable for the ranking of tasks. For optimal solutions in cloud systems then meta-heuristics algorithms like GA-PSO, hybrid optimization algorithms are more useful. To address the issue of better scheduling, this research provides a framework for scheduling and optimization of cloud systems in a cloud environment. This framework is using a better approach for scheduling a workflow in a cloud system. It works as elaborated below:

First, take input workflow tasks and rank the tasks based on parameters like makespan, cost, and deadline constraints. During the second stage, apply scheduling algorithm on ranked tasks and then schedule the tasks to cloud resources. The performance of the algorithm is analyzed by comparing it with other existing research work in literature.

The complete framework of the proposed study has been divided into two phases. Phase 1 is about ranking the input workflow tasks using the proposed method named as distributed-HEFT ranking method and phase 2 is about optimizing the scheduling of these ranked tasks on cloud resources and improving in performance parameters makespan, cost, energy consumption, and response time by using proposed optimization approach named TBW (Tabu Bayesian Whale) optimization technique. Simulation results depict the effectiveness of the proposed correlation-based Distributed-HEFT ranking method. It provides adequate results compared to other methods like HEFT and Fuzzy HEFT methods. In this research, our goal is to improve the utilization of cloud resources as well as enhancing the performance of the cloud system. The experiment results show that the proposed algorithm decreases the total execution time, total execution cost in comparison with HEFT, Fuzzy-HEFT, GA, PSO, GA-PSO, and Whale optimization. Besides, it improves energy consumption and response time.

ACKNOWLEDGEMENTS

I take this opportunity to express my sincere thanks to everyone who has helped me in various capacities to carry out this research and prepare the report.

I am delighted to thank our respected supervisor Dr.KamleshLakhwani who has offered tremendous support in the completion of this research. Their unparalleled knowledge, judgment, and motivation were together with their expertise.

I acknowledge the Department of Computer Science and Engineering, Lovely Professional University to provide me the appropriate resources to pursue the doctoral degree. I am grateful to the administration staff at the Centre for Research Degree Programmes for the numerous applications.

I would also thank my parents, husband, Mother-in-law, son, friends, and contemporaries for their co-operation and compliance. I cannot cherish a greater fortune other than having them in my life. Their care and love are indispensable for my achievements.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1. Introduction to Cloud Computing

Cloud Computing is a kind of technique that is widely preferred in various scientific applications of fields such as geography, astronomy, biology, etc. it provides a pool of virtualized resources that are readily available and can be configured according to the requirements of its users [1, 45]. Cloud environment provides a computing platform as Platform-as-a-Service (PaaS), even network means infrastructure on rent called Infrastructure-as-a-Service (IaaS) and apart from this, even an individual can get the favor of individual software on rent called SaaS (Software-as-a-Service) [2, 102, 103, 106]. So, it is highly appreciable to say that a cloud covers the demands of each type of consumer means from an individual to a business [7]. It is not wrong to say that a cloud is a hub that provides all services from one link over the internet [122]. The best about cloud is dynamic services using large scalable and virtualized resources over the Internet [17, 58]. It is like a user-friendly store that is available 24*7 for every consumer. Yet, the important focus is whether all services are up to mark. Apart from this, myriad issues are the challenges that cannot be ignored if one wants to boost up the performance of a system [3]. So, analysis based on resources utilization is highly demandable. Albeit, several factors after combinations make a system perfect like how much a cloud costs, how much time it is consuming, and what is the response time of each task execution. So, if a task is not executed within deadline constraints [4], it means the utilization is either improper or cloud is not managed sufficiently. [45, 46]

Cloud computing is one of the latest and recently challenged areas to provide inexpensive on-demand services with good quality of control to users through the internet. It doesn't matter when and where but it matters that every demand of customer must be successful without any kind of delay. It also provides dynamic services using large scalable and virtualized resources over the Internet. According to NIST i.e. U.S National Institute of Standards and Technology, Cloud Computing is described as a model and it enables suitable access to a pool of computing resources which may be networks, servers, storage, or applications [43]. It is a combination of a technology, platform that provides hosting and storage services via the internet.

Cloud computing indeed provides live access to all kinds of resources with minimum spent i.e. a shared pool of online available resources logically placed into the network, storage, etc. categories and these can be accessed as per need from an individual to a big organization [3]. Adding further, a cloud is an ocean of services as per the choice of users. Agility, flexibility, and most important cost and time are the features of cloud systems [44]. Productivity, as well as the performance of a cloud, has become a challenge for which better optimization from all angles is mandatory [4, 5, and 183]**.**

Over the last few years, cloud computing has become very popular as its potentials significant cost reductions. In an environment with cloud facilities, users need not to buy the infrastructure for their various computing services but can access computers in any region of the world. These cloud features support high scalability as well as multi-tenancy and also offer enhanced elasticity as compared to the earlier existing computing methodologies.

1.2 Essential Characteristics of Cloud Computing

Some of the unique characteristics of cloud computing have been illustrated in Figure 1.1. These characteristics provide quality-based services as described below [125].

Broad network access: Cloud computing resources are accessible over the network in a heterogeneous environment [25].

Resource pooling: The resources provided by cloud service providers are pooled to service more than one customer with the multi-tenant feature of cloud from the same physical and virtual resources [5]. The customer does not know the exact location of the provided resources.

Rapid elasticity: Resources are provisioned and released on-demand according to consumer's needs. It can be scaled inward and outward proportionate with the demand of the consumer.

Measured service: Resource usage is monitored and measured.

On-demand self-service: Various cloud users can provision cloud computing resources such as network access, storage, and others without requiring human interaction.

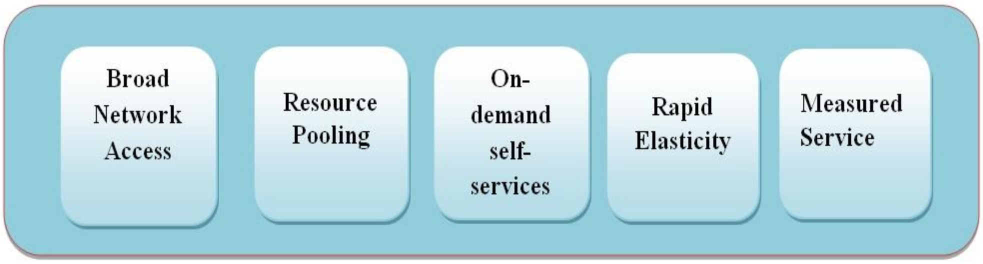


Figure 1.1: Unique characteristics of cloud computing

1.3 Types of Cloud Computing

There are total three types of cloud computing as listed below:

- Private Clouds
- Pubic Clouds
- Hybrid Clouds

Public Clouds

It is defined as a cloud system where the cloud environments are distributed among multiple tenants. Various examples of public clouds are Google cloud, Microsoft Azure, AWS (Amazon Web Services), Alibaba cloud and IBM cloud.

Private Clouds

This type of cloud is dedicated to individual end user. It provides isolated access to its users. For example, Attendance system of a company can have its own private cloud.

Hybrid Cloud

This type of cloud seems like single IT system created from several environments. It can be combination of private and public clouds like: private and public cloud combination, two or more public clouds, two or more private clouds. All hybrid clouds are also called multi clouds but vice versa are not true.

1.4 Service Models

As per the different types of services, the cloud architecture has three types of layers as depicted in figure 1.2. The name of different layers of the service model is as follows:-

- Software-as-a-Service (SaaS)
- Platform-as-a-Service (PaaS)
- Infrastructure-as-a-Service (IaaS)

Software-as-a-Service (SaaS)

SaaS is the top layer of cloud computing. SaaS is provided by Application Service Provider (ASP) with various software applications over the Internet. It makes the customer overcome the burden and cost of installation of this software and its licensed updating.

Platform-as-a-Service (PaaS)

The second layer of Cloud computing service model is known as PaaS. It provides a computing platform and allows users to develop their applications using the platform provided by cloud providers. It provides a setup to implement and test various applications of cloud.

Infrastructure as a Service (IaaS)

It is a layer of sharing various hardware resources for executing services using Virtualization technology and its main objective is to provide infrastructure on rent such as network, servers and physical data centres [4], CPU cycles, RAM, etc. as a service to the users. The users do not need to purchase physical hardware and physical data centres. Virtual infrastructure is available to the user and cost is paid as per its use. It is selected in the proposed study for the execution of workflow of scientific applications such as astronomy, bioinformatics, geophysics, etc. for this purpose; various scientific workflows MONTAGE, EPIGENOMICS, INSPIRAL,etc are accepted as input workflows [33, 34, 61] All three Services models of cloud are shown in figure 1.2.

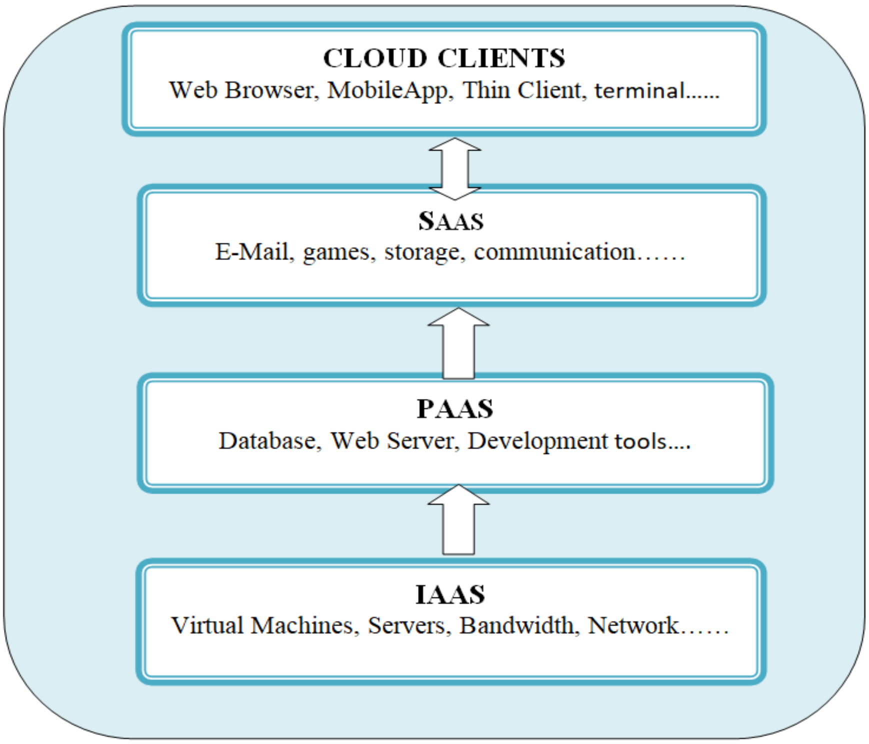


Figure 1.2: Cloud Service Model

1.5 Cloud Framework

Cloud environment is updating day by day to ease the life of their consumers. Yet, there are some major issues like total execution time, total execution cost, maximum resource utilization, energy consumption, charges as per consumption, recovery of lost data, portability of data and tasks, data transfer into virtual setups, ranking of tasks, security, reliability, Quality of Services, productivity, real-time monitoring, dependency of cloud consumers on cloud service providers, etcetera. All these issues are still challenges for cloud service providers [1, 126].

1.5.1 Challenges and Need of running workflow applications in Cloud Environment

Adopting and executing a workflow application in cloud environment is one of the major needs of various sectors. If we talk about the challenges faced then Zhao et al. (2011) identified few challenges listed below [65]:

- Large Scale Data Management
- Multi-Objective Service Composition
- Computing and Task Mapping Challenges

Large Scale Data Management

As workflow applications involve a huge number of tasks and for mapping these tasks on cloud resources, huge data transfer and its management is a serious issue in cloud.

Multi-Objective Service Composition

Cloud infrastructure is used by their various customers which not only depends on a single objective like time or cost or other QoS as a single parameter. So, finding an effective multi-objective solution as per customer requirements is a major challenge.

Computing and Task Mapping Challenges

No doubt, in cloud system the resources are unlimited which customers can use as per need and as per demand. Yet challenges like cloud resources characteristics, its requirement, virtualization, and others that need to be focused on during scheduling [84].

1.5.2 Need of Cloud Computing

Cloud Computing has various approaches which not only support handling a large amount of data remotely but also provide its protection as well as recovery if required. Scheduling is a process that arranges controls and also optimizes the workloads in a production process or manufacturing process. In cloud computing paradigm, the whole infrastructure is a service that is possible to be utilized by its various types of consumers. The requirement to use it is World Wide Web (WWW) facility. Cloud computing is indeed the next version of grid computing and it targets service-oriented facilities rather than application-oriented [101].

1.6 Importance of Scheduling in Cloud Computing

Various kinds of scheduling approaches are adopted by researchers in a cloud environment so that system throughput and load balance can be improved. Apart from this, it maximizes the utilization of resources, saves energy, reduces costs, and minimizes the total processing time [8].

Concept of Critical Path

It provides a graphical representation of any assignment and is a network logic diagram. It uses the concept of CPM. Till the end of the assignment, CPM calculates the longest path of planned tasks.

It is one of the list scheduling algorithms. A Critical path-based scheduling algorithm for workflow applications in cloud computing targeted the overall reduction in workflow cost by meeting deadline constraints [32, 131]. Figure 1.3 provides a representation of the critical path that helps in scheduling.

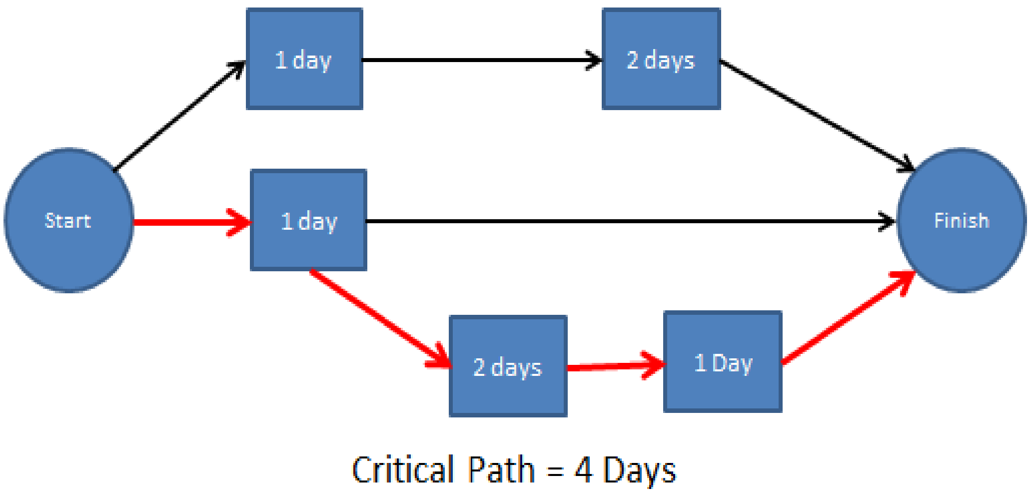


Figure 1.3: Representation of critical path

1.7 Workflow

A workflow is a combination of a series of activities that are required to complete a task. The tasks always have a dependency between them. A workflow includes a sequence of tasks that are joined with some control and data flow and the tasks of a workflow execute in some order. The concept of workflow has its uses as a business process tool that provides a smooth representation of various tasks of input activity. It helps for the automation as well as optimization of the processes of an organization. The workflow facilitates the execution of complex scientific applications that involves a vast amount of data. It has been expanded with the scientific community where uses of scientific workflows are highly appreciable for the support of large-scale and complex scientific systems. It takes scientific data in huge amount as input, and then manages, analyzes, simulates and optimizes it [79].

The main process in cloud computing is to meet large volumes of data and store data at a suitable data center during the execution of a workflow. The complexity of scientific workflows execution urges scientists to rely on Workflow Management Systems (WMS). The Workflow Management System is shown in figure 1.4.

The major components of Workflow Management Systems (WMS) are listed below.

Workflow Design: It includes the input workflow and its tasks. It comes under the application layer of the complete WMS.

Workflow Tasks: It describes data flow and also an underlying process that is part of a workflow.

Workflow Engine: It monitors the execution of Workflow Instances. Also, it provides the ability to start, pause and stop the executing workflow instances. It executes the jobs as per dependencies defined by the workflow. It manages jobs by tracking their status. It comes under the Service layer of the WMS.

Workflow Scheduling: It works on scheduling and management of data of input workflow. It is under the management layer of the WMS.

Storage Resources: The cloud environments now actually store the scheduled tasks on the cloud storage resources.

Workflow Client Applications: It is an interface to strengthen the client interface connection.

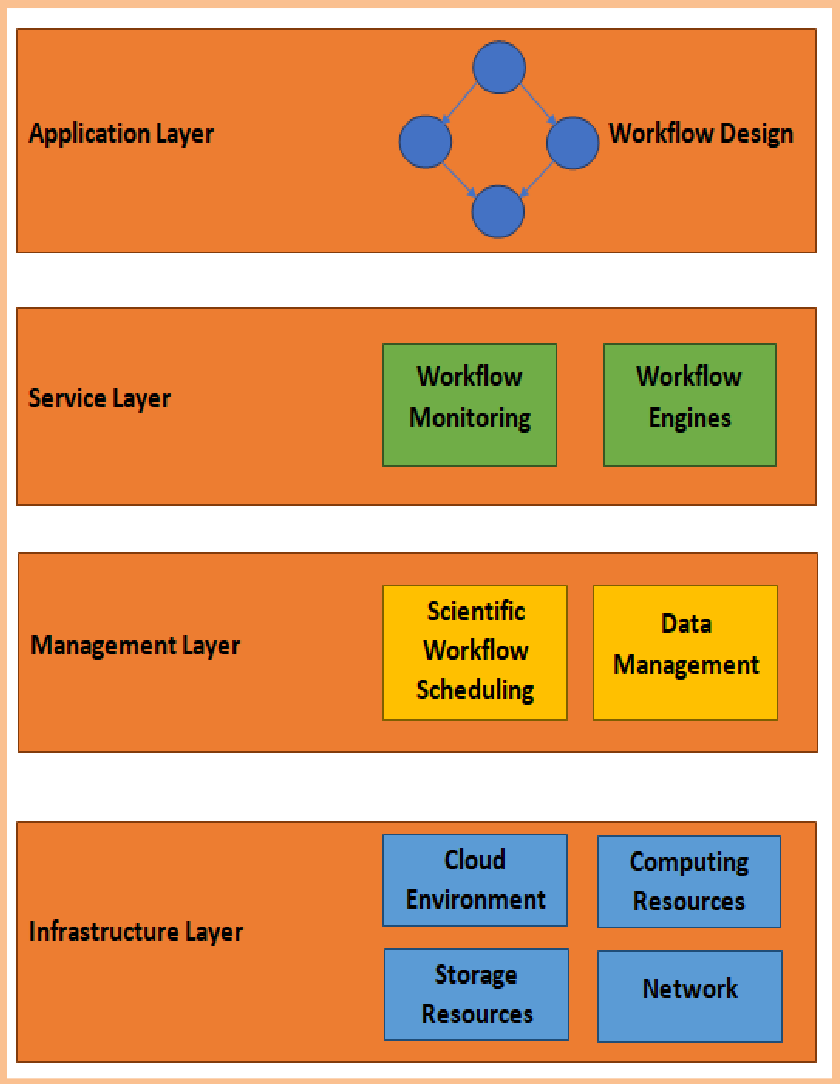


Figure 1.4: Workflow Management System

Administration and Monitoring: It is an interface that provides an observing framework and metric capabilities to encourage the administration of application situations for composite work processes.

Workflow Model: It is represented as Directed Acyclic Graph (DAG model). DAGs are commonly used by scientific applications as well as the research community. For example, workflow management systems such as Pegasus [33, 34], ASKALON [151], and DAGMap [152] support the execution of workflows modelled as DAGs.

Formally, a DAG representing a workflow application can be represented as W = (T, E).

It is a collection of the following:

T: A set of tasks represented as:

T = t1, t2, …..,tn and

E: A set of directed edges.

Figure 1.5 shows a DAG workflow model.

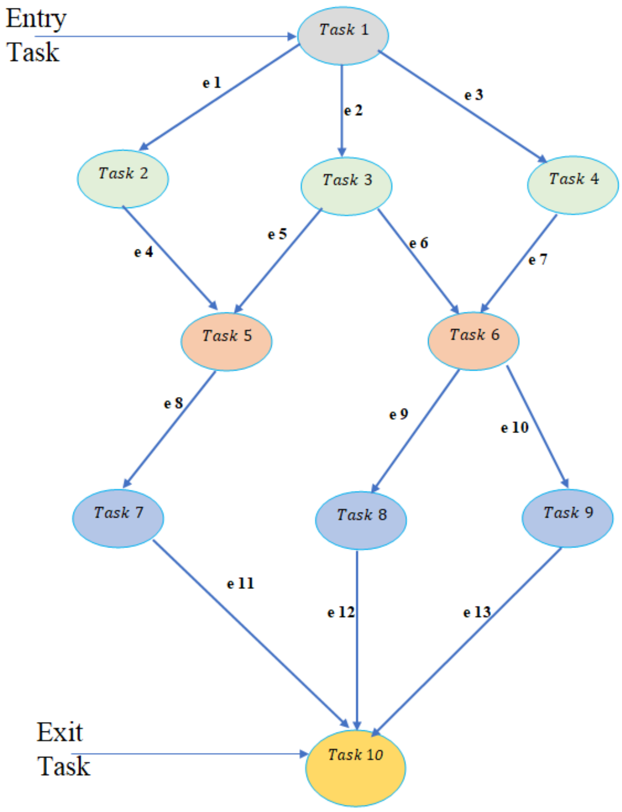


Figure 1.5 DAG Workflow Model

If a data dependency between task ti and task tj exists then an edge eij of the form (ti, tj) is created. There, ti is said to be the parent task of tj, and tj is said to be the child task of ti. Also, a child task cannot run until its entire parent tasks reach their completion, and data needed as its input is available in the resource computing it.The dependencies are usually dependent representations of the data that a task output is needed as input to another task. The execution of a secondary task cannot begin until the completion of the primary task.

1.8 Workflow Scheduling

A workflow is the computerization of a business process, in entire or part, and during its execution, tasks are passed from one participant to another for its action. The flow of the tasks of workflow is always according to some set of rules. Also, Tasks order, ranking, synchronization, etc. are important aspects. Scheduling is a method by which a task is assigned to an optimal VM to complete the work. Workflow scheduling is an order in which tasks should be completed. It is a process of mapping workflow tasks to various Virtual Machines (VMs). It means the allocation of tasks to VM is scheduling of any scientific workflow [69]. Figure 1.6 shows the concept of Workflow Tasks Scheduling to VMs.

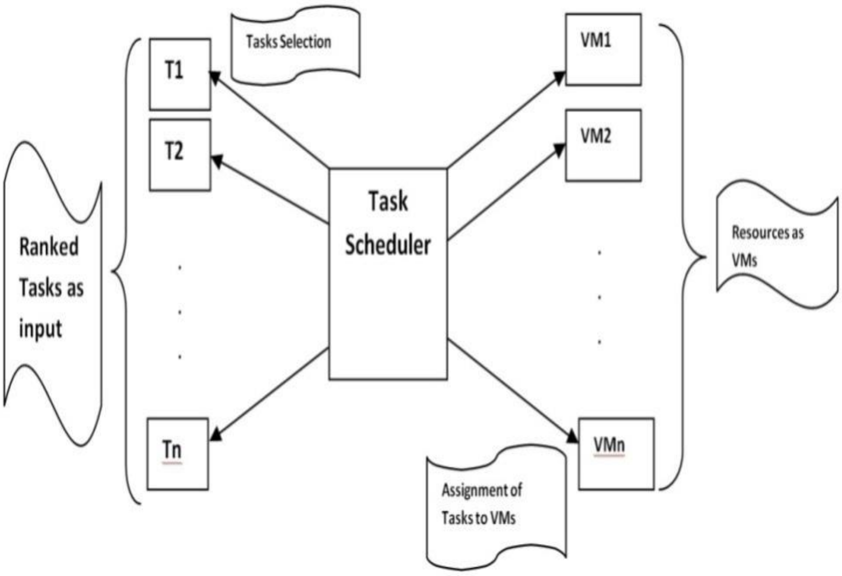


Figure 1.6: Concept of Workflow Tasks Scheduling to VMs

Also manages its execution as well as satisfying all dependencies, constraints, and objective functions. The main aim is to complete the workflow within its deadlines. No doubt, the energy problem in clouds is still a major concern. During a survey of Amazon, it is described that half of the budget goes into energy consumption for cooling of servers and power systems. It is not only the cost that goes for energy but a serious issue of global warming [35]. So, high attention to updated research is needed for saving energy and making the scheduling eco-friendly. One important category of workflow is scientific workflow.

1.9 Scientific Workflows

A scientific workflow is used to describe the dependencies between the tasks. It is depicted as a directed acyclic graph (DAG). DAG contains nodes and edges. Nodes are called tasks and the edges denote the task dependencies.

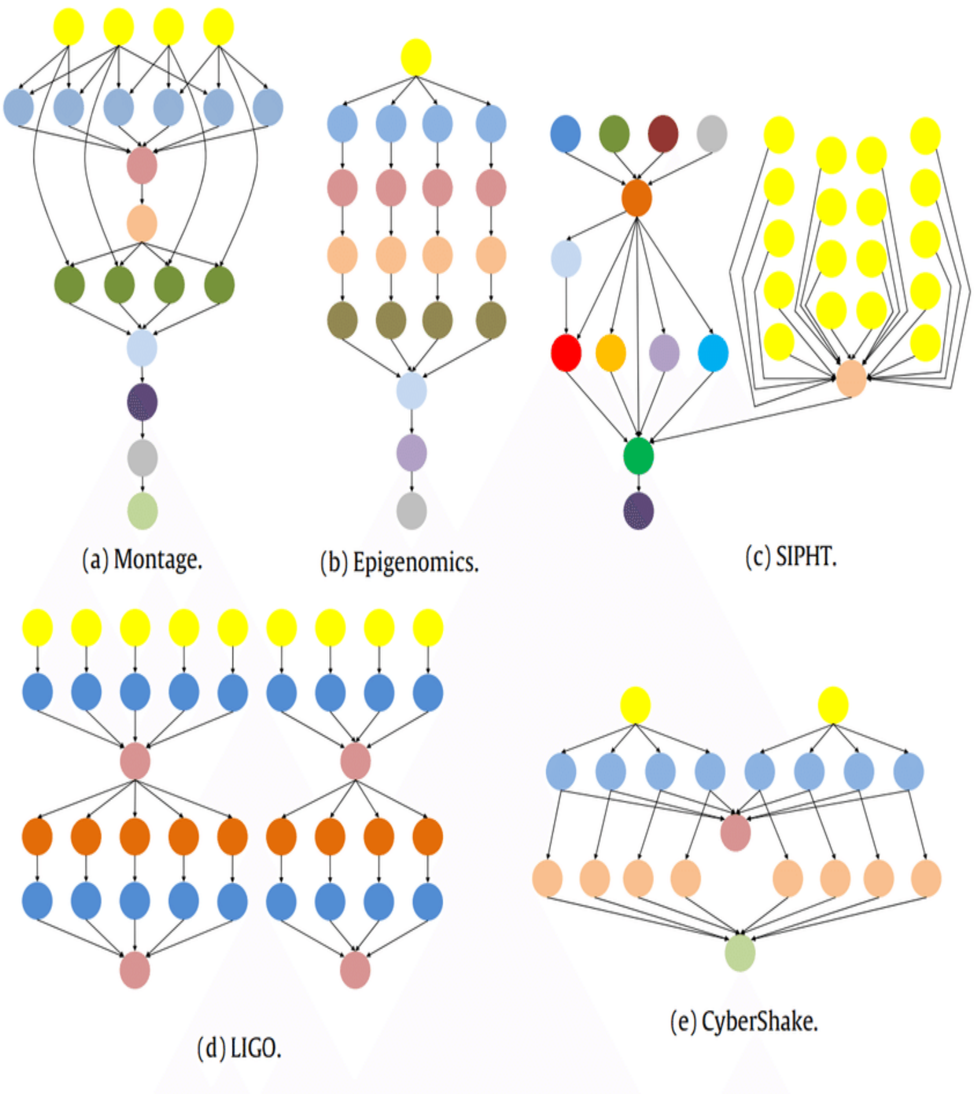


Figure 1.7: Five Realistic Scientific Workflows [9]

Scientific workflows can be simple or complex depending upon the number of tasks. Five types of scientific workflows are depicted in figure 1.7.

Different scientific workflows are used for various purposes as shown in table 1.1.

Table 1.1: Use of Scientific Workflows

Figure no.	Scientific Workflow Name	Application
1.7 (a)	MONTAGE	The Montage application created by NASA/IPAC is an open-source toolkit. It stitches together multiple input images to create custom mosaics of the sky. It is an astronomy-based application. It is used for creating custom mosaics of the sky based on a set of input images. It helps astronomers for producing a composite image of the sky which they are unable to produce with the help of their astronomical cameras [191].
1.7 (b)	EPIGENOMICS	USC EPIGENOME centre has created this scientific workflow. It is also a scientific data processing workflow. Various GENOME operations and their tasks are executed with it.
1.7 (c)	SIPHT	This scientific workflow Is linked with the projects of the f bioinformatics field. It makes bacterial replicons search strong in the NCBI database.
1.7 (d)	LIGO	From the data collected during the coalescing of compact binary systems, LIGO scientific workflows are used for analyzing and generating gravitational waveforms.
1.7 (e)	CYBERSHAKE	These scientific workflows are for sensing earthquake hazards in a region. The Southern Calfornia Earthquake Center uses it efficiently for its projects [30, 157]

1.10 A Stride towards Optimal Solution

In a Cloud System, there are myriad factors that are responsible for making a cloud system as most satisfactory. For this, task Ranking and task scheduling in an optimal way have become a vital need for making a cloud to meet all requirements.

Ranking in Cloud Environment

Ranking in cloud is possible in different situations like ranking of cloud service providers, ranking of cloud users, assigning rank values to different resources [10, 62, 108, 140, and 184]. Apart from this, ranking at the input, output as well as at intermediate stages is possible.

Assigning path to tasks based on deadline and budget

After all tasks are parsed properly, it is efficient to apply a ranking algorithm based on initialization [26].

1.11 Tasks Scheduling Algorithms in Cloud Computing

In the fast and demanding era of research, there are plenty of tasks that are running concurrently on the cloud and using the resources online. The scheduling of resources reduces the computation time and processing time of tasks [31].

Different types of algorithms and techniques are used for task scheduling in the cloud categorized in figure 1.8.

Task scheduling is an essential and most important part of a cloud computing environment. If tasks are scheduled in a good manner then it enhances the utilization of resources which automatically minimizes the response time [6, 39].

Scheduling targets the planning of various activities. This way, it helps to fulfil the user's demands. It maps various tasks to cloud resources for its smooth execution. Apart from this, it enforces priorities, gives preference to those which holding key resources, maximizes resource utilization, minimizes total execution time as well as total execution cost, maximizes throughput, avoids indefinite postponement, be predictable, and many more. [22, 86]

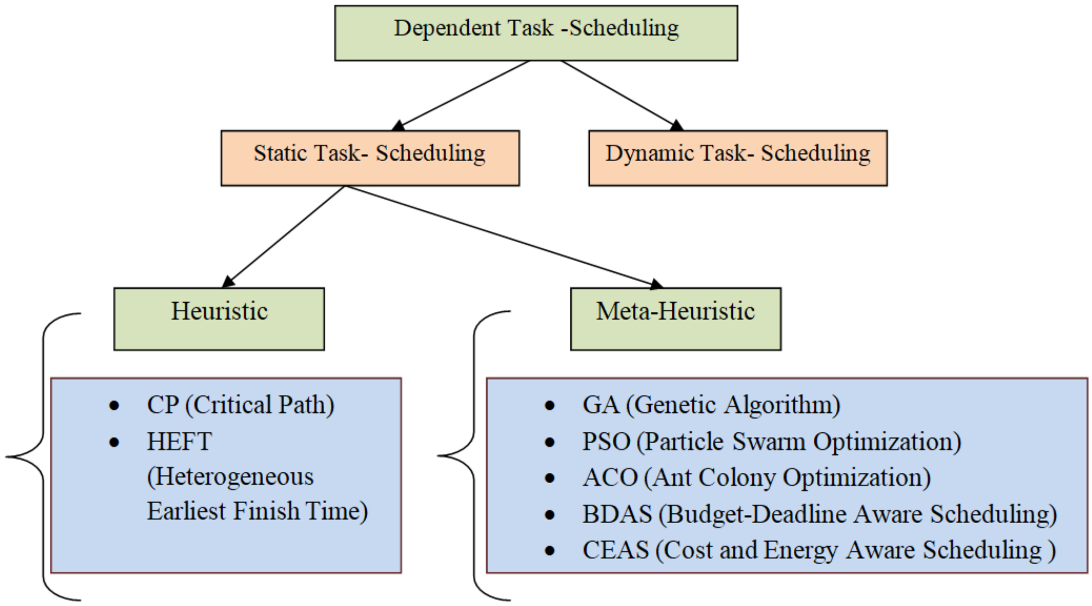


Figure 1.8 Dependent Task-Scheduling

1.12 Objectives of Scheduling in Cloud

The common objectives for workflow scheduling are described below [23]:

Budget: It is the cost that the consumer pays for taking various needed advantages of a cloud system.

Deadline: It is a significant QoS prerequisite. It means a specific time by which some work must be accomplished [39].

Reliability: To make a service faithful and trustworthy, reliable execution means satisfactory execution of an application. So, backup, dynamic replications can be applied in the scheduling [193].

Availability: By the correct workflow scheduling, assignments are executed quicker and the execution is ended rapidly. This improves the availability of cloud resources.

Minimize the makespan: It is also known as the total execution time of a system. It is the time at which the last work process task completes its execution.

Supporting Service Level Agreement: SLA is a record that has the different contemplations of the administration customers and suppliers. These incorporate the clarifications of QoS conveyance execution guarantees [121].

Security: Attackers may abuse some cloud highlights and segments to dispatch cloud explicit assaults. A protected scheduler creates a protected booking to moderate the impacts of the security assaults [4].

Load Balancing: For this situation, a scheduler upgrades the utilization of assets to stay away from the overburden of any cloud asset.

1.13 Problem definition

The mechanism for scheduling workflows in cloud computing involves coordinating task execution by mapping tasks and resources and preserving dependencies between tasks. Mapping must be done concerning the various QoS restrictions defined by the consumers (e.g. makespan, budget, deadline constraints, and energy consumption). In a cloud environment, several steps are used to execute a workflow. At first, the tasks are reserved with the available resources based on a minimum deadline. At the second level, the grouping of tasks is combined with the proposed scheduling algorithm concept to immediately achieve the QoS restrictions, such as makespan, cost, and performance gain.

1.14 Research Issues and Objectives

Research Issues

- It is identified that in several existing approaches, input tasks are randomly distributed to Virtual machines.
- In existing approaches, deadline constraint is dependent on workflow dependency.
- Due to random initialization of the optimization process, later the union of all tasks becomes time-consuming.
- During optimization, in many approaches, a single objective is taken into consideration, which sometimes conflicts with time and cost.

- The present study attempts to provide an optimal solution based on checking the status of VM utilization only.
- In existing approaches, optimization is either working on local VM or global data center but not both at the same time.

Various Objectives of this research are as below

- To design and develop a Task Ranking Algorithm based on tasks dependency and computation time.
- To propose an optimal scheduling algorithm based on a designed ranking algorithm on the scientific workflows with deadline constraints.
- To evaluate the effectiveness of the proposed technique using parameters Makespan, total execution cost, and response time.

1.15 Research Methodology/Contribution

In light of prevailing limitations and challenges in scheduling algorithms this research work attempts to offer the following contributions:

First of all, a detailed survey of different scheduling algorithms and their methodology adopted by researchers are carried out. Then distributed HEFT-based ranking method was designed for better ranking of input workflow tasks before scheduling them. During this ranking, heuristics budget, time, and deadline constraints are also considered.

Afterward, to check the performance of the distributed-HEFT ranking method, parameters TET (Total Execution Time) and TEC (Total Execution Time) are calculated by scheduling tasks [75]. Apart from this, a comparison has been done with existing task ranking methods named HEFT [159] and Fuzzy-HEFT [94] ranking methods.

For providing optimal results, an optimization algorithm named TBW (Tabu Bayesian Whale) is implemented. It used the optimization concepts of Tabu search [56], Bayesian optimization [160], and whale optimization [57, 60] `to make a better approach for better results in terms of mapping to cloud resources. It is implemented on various input scientific workflows and its performance is analyzed for improving resource utilization by minimizing makespan, cost, and response time. Also, the comparison of TBW has been done with existing optimization algorithms like GA-PSO, whale optimization, and more.

Various symbol used in this work are explained in table 1.2.

Table 1.2: Symbols used in this work

Abbreviation	**Definition**
TEC	Total Execution Cost
TET	Total Execution Time
VM	Virtual Machines
RT	Response Time
EC	Energy Consumption
HEFT	Heterogeneous Earliest Finish Time
T	Time taken by task in execution
D	Deadline of tasks
B	Budget
R	Rank of tasks
N	Number of VMs
W	Workflow
X	Total time consumed for workflow tasks extractions
Cp	Critical Path
$Task_{exit}$	Exit task of workflow
$task_p$	Parent task
T_t	Time taken by task in execution
T_R	Receiving or passing time of task
T_P	Processing Time of task
T_W	Waiting Time of Task
MF	Movement Factor
CF	Cost Factor
T_c	Tasks combinations
Th	Threshold
D	Distance between each whale
K	Workflow size

1.16 Thesis Organization

This thesis is organized as shown in Fig.1.9 and is structured as follows:

Chapter 2 presents various taxonomies related to the ranking of workflow tasks and various scientific workflow scheduling methods for minimizing the makespan, cost of execution, response time, and energy consumption by considering the deadline constraints.

Chapter 3 details the complete framework which is used to make cloud system better. It is all about phase 1 and phase 2 of the proposed framework.

Chapter 4 is about a proposed ranking method that is implemented to assign a rank to input workflow tasks before mapping them to cloud resources. Here, TET and TEC parameters are considered for analysis of the distributed-HEFT ranking method. Afterward, scheduling is done based on the score value of tasks.

Chapter 5 details TBW cloud optimization technique which has used tabu, Bayesian and whale optimization approaches for making cloud better. Parameters TEC, TET, response time, and energy consumption are considered for analysis of TBW optimization algorithm.

Chapter 6 concludes the thesis, summarizes its findings, and provides directions for future research.

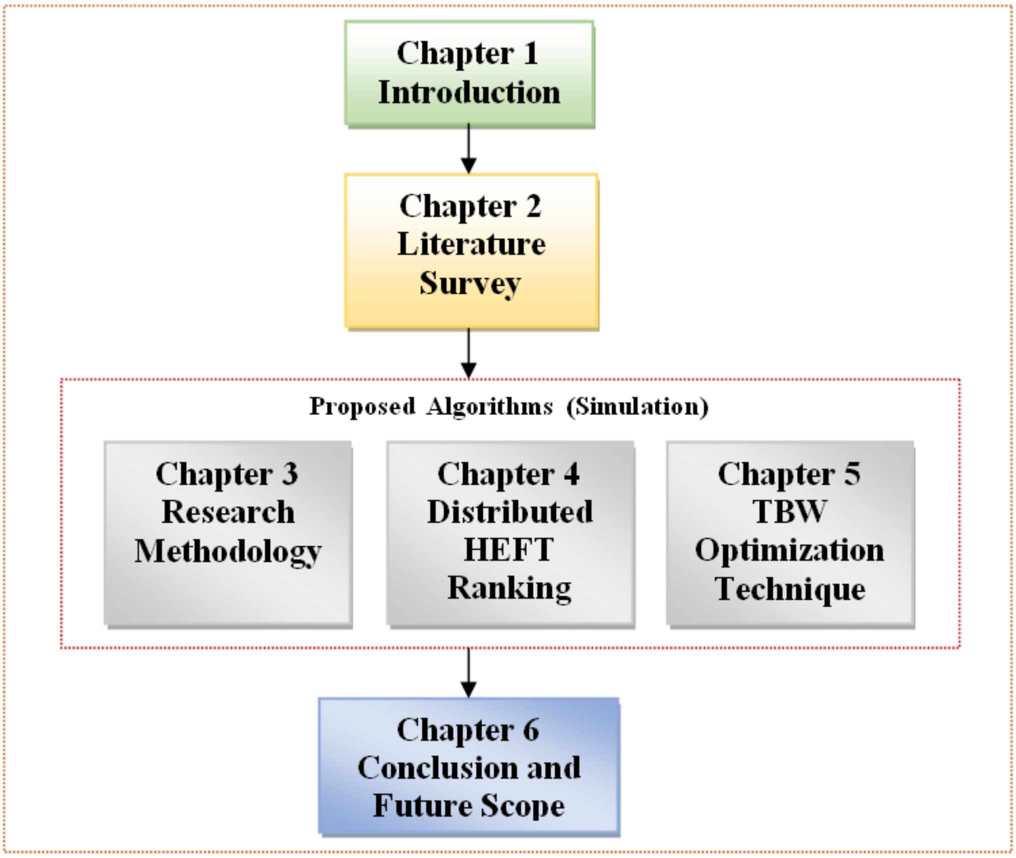


Figure 1.9 Organization of thesis

CHAPTER 2

LITERATURE REVIEW

2.1Introduction

In today's computational world, workflow systems need proper management. Cloud is a platform that supports workflows. Still, high performance and maximum utilization of resources is a challenge for all cloud systems [77]. From the previous research in this field, it is examined that effective scheduling is mandatory in all cloud systems to meet deadlines. In this chapter, challenges in scheduling, different algorithms used in scheduling, and their positive and negative impacts on services are discussed.

2.2 A Review of Challenges in Scientific Applications

Scientific applications play an important role in cloud computing. As it is a kind of sector that is using cloud computing at a fast rate. Still, several challenges are faced by researchers while using cloud facilities for various scientific applications. Table 2.1 lists some important challenges.

Table 2.1 Challenges Faced by Scientific Applications in Cloud Computing

Author's Name	**Year**	**Scientific Applications**	**Challenges**
Yong Zhao et al. [65]	2011	Amazon Map Reduce Workflows	Workflow scheduling, computation, and management.
Christian Vecchiola et al.	2009	FMRI	Prediction, scheduling, Pricing, and Computation Time.
[83] Jens Sonke et al.	2011	Workflow represented by DAG	Scheduling, Computation
[8]Suraj Pandey et al.	2010	Bio-informatics workflow	Scheduling, Cost of execution
[10] EwaDeelman et al.	2008	Montage 1 Degree	Computation Cost, Execution Time
[190] Scott Callaghan et al.	2010	CyberShake Overflow	Scaling up of resources, Execution time

2.3 A Survey on Various Scheduling Criteria

Based on the literature survey of scientific workflows and their scheduling algorithms [36, 72, 79, 92, 95, 154], it is examined that multi-objective-based scheduling is highly prominent for managing time and cost parameters at the most. After a detailed review of scheduling-based approaches by various researchers [89, 90, 100, 101, 102, and 104], we also elaborated our study targeting the scheduling criteria as shown in figure 2.1.

Detailed description of various parameters considered while scheduling is as below:

Time: Time or makespan is the primary objective of most scheduling techniques from the age of grid computing. As far as the cloud computing environment is concerned; the execution time plays a significant role since the cloud provider charges its customers based on time. In workflow application, the execution time is considered as the time taken to complete all tasks in a workflow. Hence reducing the time for executing workflow applications become a crucial factor.

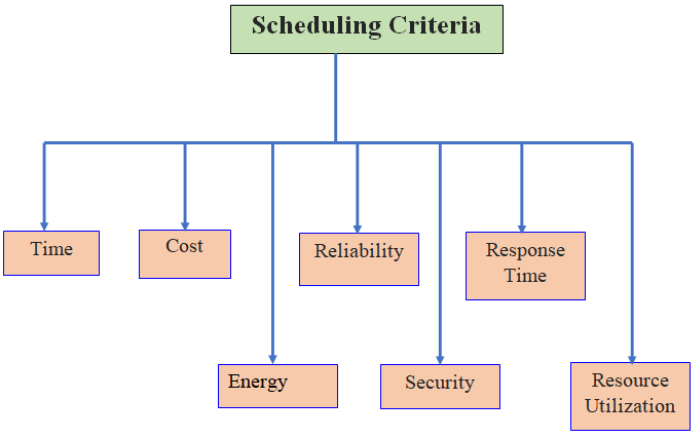


Figure 2.1: Scheduling Criteria

Cost: Generally, cloud providers charge customers for leasing the infrastructure which includes the resource usage cost, transferring cost, cloud storage cost, and others. As cost plays a dominant role in scheduling it is necessary to minimize these costs for the effective usage of a cloud platform. Thus an efficient scheduling algorithm that considers these costs during the resource provisioning is necessary for executing the workflow applications in the heterogeneous cloud environment [8, 12].

Energy: Due to the rising execution of workflows in various fields, the energy consumption of data centers has been gradually increased. Hence, energy conservation in cloud data centers has become a matter of concern. High energy consumption incurs high operational and maintenance costs.

If the load at a data center is low then also it becomes a high energy consumption center. Also if the resources are over-utilized at the servers then also it comes under the category of inefficient energy consumption cloud system. Thus an effective scheduling mechanism should be devised to address these issues which help to attain a green environment by reducing unnecessary power consumption.

Resource utilization: Resource utilization determines the efficient usage of resources. Leased Resources should be utilized efficiently to avoid unnecessary money expenditure as providers also charge for the unutilized slots. Improving resource utilization has considerable benefits for its various users in the form of cost and also for its providers in terms of profit and energy consumption. Hence improving resource utilization becomes a significant factor in scheduling. The thesis focuses on the significant scheduling criteria related to economic factors such as Time & Cost, and environmental factors such as consumption of energy and Utilization of the resources for computing Workflow applications in a cloud environment.

Security: Data privacy and security need to be addressed while adopting cloud computing as the workflows may contain confidential information which scientists may not wish to reveal. It can be further classified as single-level and multi-level. Single-level security specifies whether the data set needs security and in multi-level security, the security requirement can be specified at more levels.

2.4 A workflow Scheduling Objectives

Cloud Computing is a world where internet-based computing exists. So, the various kind of services in the form of storage, applications, or servers are provided to its user's computers through internet facility only [5]. For the success of the scientific world, the cloud has taken a well-built stride towards the facility of virtualization [50]. Huge developments have been provided for advancements in it. The primary objective of all the discussed algorithms in chapter 2 is to minimize the cost of execution. A majority of algorithms take other metrics, as outlined in Fig. 2.2. Few additionally deal with the workflow model’s safety and reliability).

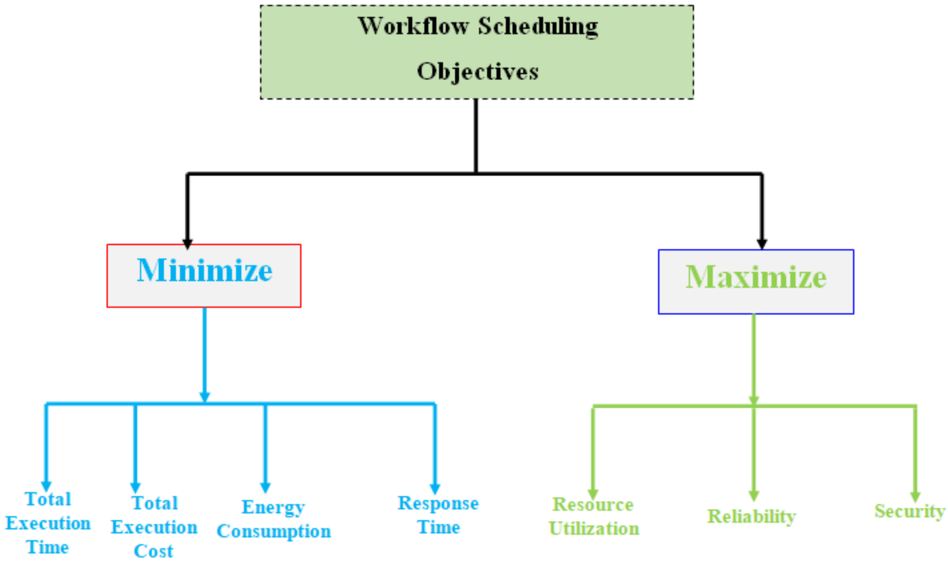


Figure 2.2: Objectives behind scheduling a Workflow

While Scheduling occurs, the important steps are about what to minimize and what to maximize in the whole process. As it is depicted in figure 2.2 that makespan which is TET or total execution time, TEC or total execution cost, total energy consumption, and response time should be minimized. On the other hand, utilization of cloud resources shall be as high as possible.

2.5 A Survey on Various Scheduling Approaches

Workflow Scheduling is a process in which tasks are mapped on available resources. Various scheduling approaches have been divided into 3 categories listed below:

- Heuristic Scheduling
- Mata-Heuristic Scheduling
- Hybrid Scheduling

Figure 2.3 as shown below describes the above-mentioned scheduling categories.

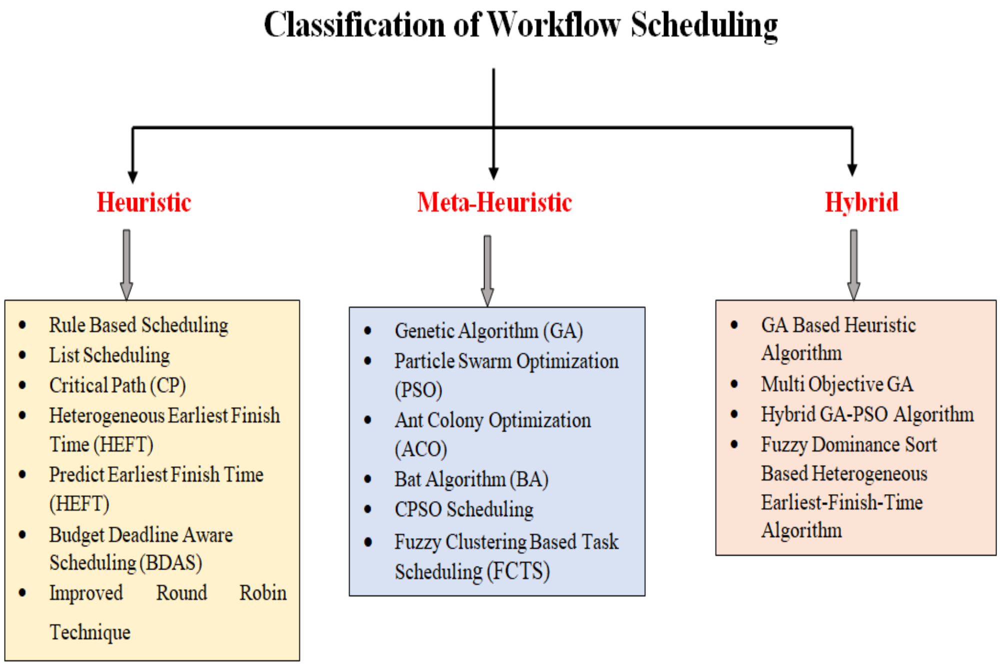


Figure 2.3: Classification of Workflow Scheduling Approaches

In heuristic scheduling, predictability is more. These are simple to use as compare to meta-heuristic and hybrid algorithms. Heuristic means the solution is reached by testing’s and trials. The chances of performance degradations are less in it.Meta-Heuristic algorithms work in a complex environment as compare to heuristic algorithms.

If we talk about work done for scheduling and task mapping to cloud instances, then we can start reviewing already work done based on a single objective or multi-objective scheduling algorithm. Also the study targeted recent approaches used by researchers while performing task mapping with QoS [186, 187 and 189].

In [73], various static and dynamic algorithms are defined in detail. Cost, budget, and deadline are QoS parameters and all should be satisfied. Direct dealing with tasks assignment to cloud resources cannot be considered fully optimized and no algorithm can indeed achieve all targets (QoS) achievement simultaneously. Workflow is one of the commonly used models for marking IaaS cloud-based science applications such as Amazon EC2 and other cloud providers [23, 121]. So, in the current study, it has been accepted that rather than directly schedule the tasks to VMs, one more stage preceding it must be added [40, 49].

ThiagoGenez, et al. [85] worked with the selection of CPU frequency configuration for resources carefully to reduce the total makespan. It is a combination of PSO and HEFT schedulers to make it better in case of time. A fitness function without any parameter value is imposed to measure the performance of various particles. Less is fitness value, better is a solution. So, particles of the swarm are moved towards less fitness value region the workflows with different sizes are accepted for simulation. Cybershake, SIPHT, and LIGO are the workflows in which experiments are performed.

The work in [16] reduced total execution time and cost. Researchers have implemented a gravitational workflow scheduling search algorithm in a cloud environment. Workflow enhancements minimize costs and makespan. Two algorithms for the workflow scheduling are hybridized GSA and HEFT. The performance assessment is conducted based on two metrics, which are the monetary cost ratio and the duration ratio of the schedule. The justification of the result is also tested by the ANOVA test and it shows that the proposed approach does better.The Rank hybrid scheduling algorithm first uses the HEFT ranking algorithm to compute the rank of each task [75, 144, and 149].

Description of various Scheduling Approaches is given below:

1. Rule-based Scheduling

The heuristic algorithms are useful in the situations for optimal solutions when meta-heuristic algorithms fail to discover the precise or optimal solution. First Come First Serve also known as FCFS, Max-min, and Min-min, Minimum Execution Time (MET) as well as Minimum Completion Time (MCT) are the heuristic algorithms. These algorithms are accepted to compare the performance and also for the analysis of task scheduling in cloud computing. These are achieved by accuracy, completeness, optimal transaction, or speed. It is considered a shortcut [128, 129].

2. List Based Scheduling

The steps involved in list scheduling are as follows.

- Assign the priorities to each task based on some criteria.
- Arrange the tasks according to their priorities in ascending order.

- Select the first task from the priority queue and allocate the selected resource from the resource pool to that task.

This notion of scheduling may be static or dynamic. If priorities of all tasks are assigned before the execution, it falls under the category of the static list. If each execution assigns priorities of the unscheduled tasks, it is called a dynamic list. Some list scheduling algorithms are Critical Path (CP), Improved Critical Route [162], Earliest Heterogeneous Finish Time (EHFT) [163], and Predict Earliest Finish Time (PEFT) [164].

3. Critical Path

A critical path-based scheduling approach meets the deadline by doing identification of critical path tasks and thus reduces the overall workflow costs [131]. To decrease the full running cost of the workflow application, workflow tasks were scheduled based on resources. The DCP algorithm to search the Task Path Critical Graph for a task at the Absolute-Earliest-Start-Time (ASET), to re-specify a task on each server, and to select a server that minimizes the finish time of the task in hand and the task that follows. The tasks are updated, prior activities are repeated and the targeted results are achieved when all tasks are scheduled. Because each task starts at a particular moment, the task will be re-scheduled, giving the DCP algorithm high time complexity [162].

4. HEFT Scheduling Approach

HEFT as well as Critical Path on Processor (CPOP) algorithms with a set amount of processors of distinct configurations has been already used by researchers in their research [22]. In HEFT, the task choice is based on a graded approach while sorting assignments assisting in minimizing their earliest completion time and generating feasible outcomes for DAG-associated issues. Budget Heterogeneous Earliest Finish Time (BHEFT) is the expansion of the HEFT algorithm [19, 56] which provides a Batch Data Communication (BDC) approach to verify whether or not a workflow application should be accepted. The suggested budget and deadline constraints are also discussed in detail [53, 161]. Researchers also suggested that HEFT be extended to the Budget and Deadline constrained Heterogeneous Earliest Finish Time (BDHEFT) method. In BDHEFT, both time and cost are taken into account under budgetary and QoS limitations. BDHEFT considers the scheduling at service and task levels. It was discovered that BDHEFT provided better results than BHEFT in terms of cost-effectiveness within budget constraints and deadlines. It selects input tasks randomly.

5. Predict Earliest Finish Time (PEFT) Scheduling Approach

It provides important improvements through the introduction of a look-ahead feature without raising the complexity of the time in the calculation of an optimistic price table (OPT). This result is optimistic, as the processor's availability is not considered in the calculation. This technique is based solely on an OCT used for processor classification and processor selection [97].

6. BDAS algorithm of Scheduling

A unique algorithm based on heuristics called Budget Deadline Aware Scheduling (BDAS) has been proposed in [37]. It schedules the workflows by considering budget and time limits for IaaS clouds. This proposal satisfied the budget as well as deadline constraints [59].

The authors used five different phases for achieving this as shown in Fig.2.4. In phase 1, they maximized the parallelism among the tasks using a partitioned method based on their dependencies. Since the role of cost is significant in a cloud environment, in phase 2, the user-defined budget was distributed at different levels. Another critical instance is meeting the deadline, so in phase 3, the user-defined deadline or workflow was distributed at different levels by fixing the same deadline for all the tasks at the same level. In phase 4, the independent tasks were chosen from the ready queue based on the priority for the parallel processing.

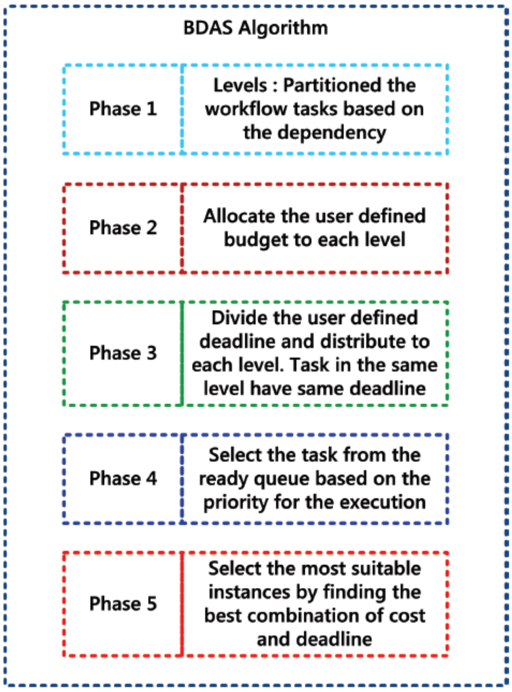


Figure 2.4: Phases of BDAS algorithm

The priority was fixed based on the earliest start time. Finally, in phase 5, a new trade-off was introduced between the cost of execution and time for finding the best combination of cost of execution and time for selecting the best instances. Although this method has low overheads, it is not preferable for dynamic scheduling.

7. Improved Round Robin Technique

A new approach that consolidates three of the current scheduling algorithms, namely FCFS, SJF, and RR, to produce a fresh approach that lowers the waiting time of a process has been provided [167]. Here, FCFS is First Come First Serve, SJF is Shortest Job First and RR is Round Robin algorithm.

The suggested algorithm decreases the number of context switches to provide an algorithm that is fair, efficient, and methodical. The algorithm is an expanded version of the Round Robin algorithm to determine and execute the time quota for each method with a level of priorities (low, medium, or high). The algorithm throughput is increased by running processes with less burst time by assigning the process to the CPU when it is ready but not pre-empting the running process.

8. GA Scheduling algorithm

One of the meta-heuristic approaches motivated by the evolutionary nature of natural selection, fitness function, and genetic operators is the genetic approach (GA) which helps to get the optimal solution [47, 70]. This approach is faster than other heuristic approaches [21]. The algorithm "GA based Budget constrained time reduction," is based on a GA that helps to reduce the failure rate in reliable workflow scheduling systems [107]. The main steps in the fundamental GA are creating the original population, selection to create new individuals, and assessment of fitness. An enhanced GA algorithm [111, 137, 145, and 146] using Max-Min and Min-Min to produce the original population of the two novel common heuristic approaches have been defined. In Max-Min, the task chosen for scheduling is structured on the grounds of the maximum moment and one of its assigned resources is assigned a greater time value. In the Min-Min algorithm, however, the priority is given to the minimum service time. After that, all other tasks will be updated in due course. Authors have discovered that enhanced GA is more efficient than GA.

9. PSO scheduling Approach

Particle swarm optimization also famous in the cloud research with the name PSO approach has been provided for optimizing the scheduling of workflows for reducing the makespan [168]. The tasks are allocated to the available resources based on a meta-heuristic approach by combining heuristic scheduling along with PSO. They proved that the running time of the suggested method is faster than GA for the given workflow. They showed that computation cost by PSO was better than the best resource allocation approach under deadline constraints. Since the convergence rate of PSO [20, 113, 114, 115] is very fast, the result may not be very accurate which leads to the tendency to get stuck when working towards a locally optimal solution because of a lack of local searchability. A novel scheduler concept called Particle Swarm Optimization with Dynamic and Static Topologies (PSO-DS) has been suggested that effectively optimizes the makespan i.e. time and execution cost [88]. A unique variant of PSO to address the problem of workflow schedule has been suggested called Particle Swarm Optimization iterative (PSOi). It contains approaches to enhance optimum alternatives by not sticking to local optimal conditions. It utilizes a new

”Inverse” method for moving particles to a different room. Furthermore, after each iteration, PSOi helps to update the particle’s place [171].

A PSO-based approach named SLPSO or Self-Learning Particle Swarm Optimizer is better than PSO. PSO already has a ”fast convergence” advantage. It involves distinct speed updating techniques, which assists in ensuring that the outcomes are not trapped in local optimum solution [172].

For runtime workflow scheduling, researchers considered PSO to discover a worldwide optimal solution for the cost of execution including computing and communication expenses [18, 170]. Two variables are used for determining the optimized alternatives: One is the particular scheduling technique, while the other is the PSO, which is used to achieve optimized mapping outcomes.

PSO (Particle Swarm Optimization) approach for WFMS worked based on utilization of CPU frequency. Researchers have used the concept of DAG (Direct Acyclic Graph) to represent global task scheduling which is workflow-based [26,133]. Both service consumers and service providers are vital phases for high profitability based on cost measurements. Also for minimizing overall executing cost and time, constraint scheduling strategies have been introduced [88, 120, and 166].

10. ACO Scheduling Approach

An optimization technique based on Ant Colony Optimization (ACO) has been used for identifying underutilized VMs by Pareto distribution. It describes VM migration-based optimization for the reduction of cost and makespan [134]. Also, improvements in task scheduling using ant colony optimization have been well discovered by focusing on various parameters like cost, makespan, load balancing [63, 81, 116, 117]. For heterogeneous distributed systems this model is presented [42]. The service level agreements are used for monitoring the service providers' quality of service (QoS). By using parameters cost, makespan, and resource utilization, the problem of workflow scheduling is solved. The ACO algorithms reduce the cost and makespanand enhance resource utilization.

11. Bat algorithm

To examine workflow scheduling problems with multi-objectives of workflows targeting a reduction in makespan and improvement in reliability has been proposed with the name Bat algorithm. It also considers the deadline constraints in the whole scheduling process [91, 174]. The complete working of the Bat algorithm is assisted by virtual bat echolocation that implies that a variety of alternatives are being understood. They compared the Bat algorithm to the Basic Random Evolutionary Algorithm (BREA) and showed that the suggested methodwas more efficient for budgetary items and other QoS restrictions [122, 125].

12. CPSO Scheduling

A novel approach to schedule the workflows to be performed in the IaaS platform was suggested in [174]. The approach produced a schedule of task-resource mapping. The Catfish particle swarm optimization (C-PSO) method was employed to choose the best schedule at the lowest makespan for handling and cost of execution. The suggested technique was then compared to conventional PSO performance.

13. Fuzzy Clustering based Task scheduling

Researchers have provided an approach based on fuzzy-based clustering of resources for workflow scheduling. The main target of this approach was a reduction in the makespan of workflow execution. In FBCRS, cloud computing takes into account a set of features that describe the synthetic efficiency of processing units in the scheme. The processing unit network is pre-treated by the fuzzy grouping technique for ensuring that the processing network is fairly split with these characteristics and the time impact of the prepared job in the critical path. Consequently, the expense for choosing which processor to perform the present task is greatly reduced [176].

14. Hybrid optimization algorithms

The meta-heuristic workflow scheduling algorithm has been recommended which worked significantly for the reduction of the cost and makespan [72]. It is a recombination of the famous HEFT, meta-heuristic, gravitational algorithm, and similar other heuristic search algorithms commonly used for workflow applications.

Also, a hybrid algorithm with the best features of the approaches based on heuristic techniques and a Genetic Algorithm (GA) has been proposed [13, 148, 150, 158, and 177]. They suggested Line-wise Earliest Finish Time (LEFT), a heuristic algorithm as an alternative to HEFT for original GA population generation. A new heuristic hybrid algorithm based on PSO and gravitational search algorithms was launched [178]. About the costs of processing and transferring, the suggested algorithm takes into account time limitations. Both consumers and utility suppliers can utilize the suggested workflow scheduling strategy.

15. Hybrid GA-PSO algorithm

Based on hybrid scheduling concepts, a meta-heuristic approach called Hybrid Genetic Algorithm and Particle Swarm Optimization (Hybrid GA-PSO) has been suggested in the research which effectively assigns the tasks to VMs [75]. The proposed method combines two heuristic optimization techniques and this approach aims to improve the QoS by minimizing makespan and cost by balanced allocation of dependent tasks to the resources in cloud computing. In this technique, during the first half of maximum iteration, GA is used for generating population and in the remaining half; PSO is applied over the generated population for getting accurate fitness function when compared to traditional GA and PSO approaches [119].

16. Fuzzy dominance sort based heterogeneous earliest-finish-time algorithm

Based on joint cost optimization in IaaS clouds, A heterogeneous FDHEFT algorithm is created in this system. It includes the fuzzy dominance sorting system coupled with the Heuristic HEFT scheduling list [94]. The effectiveness of the proposed system was demonstrated by extensive real-world and synthetic workflow experiments. With remarkably greater hypervolume, the proposed workflow schedule delivered considerably better cost-effective fronts and could operate quicker than existing approaches.

It is a superior updating of the algorithm named HEFT and it is known as FDHEFT [8]. It works in two phases. These steps are task prioritizing and process selection of cloud resources. The scheduling priorities of all tasks are allocated during the task-prioritizing process, and then the best option for each task in the scheduling list is decided in the cloud option selection process. It is the planning of a multi-objective workflow. It applied the sort method using fuzzy dominance to HEFT. It used fuzzy rules to compute the relative fitness of solutions. There is one issue, which has been overcome in the proposed research work. The issue with Fuzzy HEFT is its static threshold value.

17. Whale Optimization Approach

In WOA, the work is performed on three operators [55, 154]. Search for prey, and then trap the prey and then bubble-net foraging behave of whales is the main dedication of authors. This algorithm imitates the hunting behave of humpback whales. Unlike Grey Wolf Optimization, the hunting behavior of WOA is random. It identifies the best search agent to follow the prey and uses a spirally simulated and bubble-net attacking method of humpback whales. The biggest mammals considered in the world are whales. Humpback whales are a kind of whale, which grows to the length of 30 meters and weighs up to 200 tonnes. These humpback whales possess spindle cells [94] in its brain similar to humans. These spindle cells help to judge, socialize have, and manage emotions. Hence it is concluded whales can learn, think, judge, communicate, and be emotional, but at a very lower level of smartness, as compared with humans. Also, a comparison has been done in WOA on 26 mathematical benchmark functions. The optimization-based results have been compared with existing optimization algorithms like HGA, PSO, PSOPC, SOS, and HPSO for different design problems. WOA has provided better results. This bubble-net foraging method is a unique hunting method found with humpback whales.Apart from the above list, the following scheduling approaches are also useful in a cloud environment.

Ramandeep Sandhu et al. [46] have provided a list of elements to be imposed in a cloud for a high degree of satisfaction because it is true that satisfaction is directly propositional to QoS. These factors are Data sharing, Availability of Services, Audit, Backup, and recovery of data, Access privileges to users, Governance, Transparency, Investigation of data and stored data location.

AlexandruIosup et al. [31] have explained the differences between the actual scope field of cloud and the requirements of scientific applications. It evaluates the cloud and checks the capability of a cloud to run the applications efficiently. Based on the number of users, the evaluation is done. These users require scientific computing services followed by evaluating cloud services mostly used for scientific applications.

The hybrid Particle Swarm Optimization Algorithm proposed by Alkhanak et al. [14] has discussed different WFS cost methodologies. Mapping is also known as the task resource mapping concept is useful where PSO-based mapping emitted the concern of the cost of service providers. Even the order of tasks is an important consideration in all heuristic methods. More than it, performance-based meta-heuristics methods were implemented. SWFS cost optimization aspects are classified in private, public as well as in hybrid clouds. Several methods like critical path, heuristics, meta-heuristics, greedy, clustering, etc. Even cost parameters are categorized as monetary and temporal where six types of economic factors are added under monetary costs. Elasticity cost plays an important role in cloud systems. Temporal cost is estimated in the whole cloud system at three scheduling stages. These are pre, during, and post.

Ghose, Manojit, et al. [19] have given the energy resourceful arrangement approach as a scheduling method in a cloud environment. This algorithm considers the static and dynamic get-up-and-go consumption of the nodes and is divided into two parts non-divided allocation of VMs on a single host, and dividable allocation on multiple congregations. The performance of the proposed scheduling approaches is compared with existing policies and it presents an average energy reduction of 70%.

Li Liu, et al. [27] projected the genetic algorithm for workflow scheduling in cloud computing with deadline-constrained and have worked on four different types of workflows which are MONTAGE, INSPIRAL, EPIGENOMICS, and CYBERSHAKE. Both TET (Total Evaluation Time) and TEC (Total Evaluation Cost) were evaluated in the user's defined deadline constraints. A penalty function, as well as penalty regulation in CGA, is proposed which is CGA2 and it works without any parameter. Also, it has worked to overcome prematurity. Apart from this, focus on crossover and mutation probability was also a prior concern. Performance is evaluated by a task ranking system [64].

The researchers Durillo et al. [23] familiarized a multi-objective technique called MOHEFT (Multi-Objective Heterogeneous Earliest Finish Time). It is one of the renowned and best algorithms which have targeted Amazon EC2 for scheduling workflows. It is betterment in HEFT [13]. The section showing results and graphs in this study is impossible with only ranked tasks. So, scheduling of tasks to cloud resources using metaheuristic algorithms makes optimum use of cloud instances.

Following table 2.2 illustrates various findings in the research done for efficient use of the cloud.

Table 2.2: Survey of Various Scheduling Approaches in Cloud System

Authors	**Year of Publication**	**The main findings of the research done**
Alkhashai, et al. [6]	2016	• Worked for enhanced mapping of tasks on resources. • Has used Tabu Search (TS). • Reduced execution time, and cost. • Increased resource utilization. • Improved the local search
Sharma, Priya et al. [7]	2018	• Enhancement in energy consumption rate. • Made Bee Colony Optimization better using Tabu search. • A hybrid approach using tabu search is used. • It balances the load of virtual machines. • Worked for better utilization, high speed, and reduction in total time taken, more efficiency, saving energy consumption and makespan.

Bozorgi, Seyedet al. [190]	2019	• Exploration and exploitation abilities are balanced. • The performance of WOA is increased significantly.
M. Adhikari et al. [24]	2016	• Presented algorithm ESWA. • Worked upon maximizing resource utilization. • Executed the workflow within its deadline
J. Meenaet al. [29]	2018	• Worked upon heuristic and meta-heuristic algorithms. • A better hybrid approach using Two-phases for • application scheduling in a cloud computing environment to balance the workload on the available cloud resources. • ACO (Ant Colony Optimization) has been applied in phase 2.
Mohammed, Hardi et al. [182]	2019	• Covers the algorithmic backgrounds of WOA, its characteristics, limitations, modifications, hybridizations, and applications. • WOA is hybridized with BAT. The WOA-BAT algorithm is a better one and presented to obtain better results in fewer iterations compared to WOA. • Applying WOA-BAT for constrained optimization problems.
F. Yiqiuet al. [181]	2019	• The aim is to work on the total time required for scheduling the input tasks. Also targeted load balancing parameter, • It uses a CloudSim simulation-based

		platform for the tasks simulation process.
Sagarikaet al. [91]	2018	• Proposed BAT algorithm. help to handle the large size of data. • Comparison with particle swarm optimization algorithm and Cat swarm optimization algorithm
Vinothina, V. et al. [42]	2018	• ACO is also called Ant Colony Optimization algorithm for heterogeneous distributed systems. Worked on parameters cost, makespan, and resource utilization
Genes, Tet al. [85]	2015	• Reduced total makespan. • PSO and HEFT schedulers are used. • Worked on CYBERSHAKE, SIPHT, and LIGO
M. Manasrahet al. [75]	2018	• Hybrid methodology GA-PSO. • Reduction of execution time and cost. • GA-PSO is better than GA, PSO
Li, Zhongjinet al. [35]	2015	• Efficient deadline constrained workflow scheduling algorithm called CEAS (Cost and Energy Aware Scheduling). • Reduced cost and energy consumption. • Better than HEFT and MOHEFT
Arabnejad, Vahidet al. [37]	2018	• BDAS (Budget-Deadline Aware Scheduling) algorithm. • Work on five scientific workflows on IaaS.
Stromberget al. [180]	2019	• The hybridized whale optimization algorithm • WOA-AEFS is proposed and compared with the original WOA, CPSO, and PBACO metaheuristics
Sreenu et al.	2017	• Schedule the tasks to the virtual machines

[179]		while reducing the makespan and the cost.
Zhou, Xiumin et al. [94]	2019	• Minimized cost and makespan simultaneously for workflows. • Comparison has been done with several peer approaches like PSO, MOHEFT.
Mahendra et al. [188]	2020	• Comparison of various heuristic and meta heuristic algorithms has been performed. • Various parameters have been accepted for comparison like makespan, time, cost, load balancing.
K Pradeep et al. [189]	2021	• Used a hybrid approach named CWOA (Cuckoo Whale Optimization Algorithm) which is better as compare to existing scheduling approaches in terms of saving makespan, memory utilizations and energy consumption.

2.6 Summary

The various algorithms which are presented in the above literature survey are either linked with the task selection phase which is even objective 1 of our proposed work or are related with the VM allocation phase which is objective 2 of our study. Also, the target of researchers is to minimize various parameters like time, cost, energy consumption, response time, etc. If we consider the scheduling algorithms which are working for the task selection phase then the most common factors used for task selection are the earliest start time or finish time of the task, task runtime, critical path tasks, and priorities of the tasks or selecting the tasks based on its ranking.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

Cloud computing proposes a broad range of computations-based as well as resources based facilities for the execution of workflows for various applications. For the execution of a single workflow, many different resources are involved.

A dynamic environment is offered by a cloud computing system in which the status of resources changes frequently. These resources are used by various computing applications. With an increase in cloud services as well as its clients, the requests are also increasing. So, it is prior demand to handle these requests. It needs to efficiently schedule them for execution on different available resources [66, 67, 68, and 80].

Many factors are faced when cloud workflows are allocated and scheduled for execution [52]. A Workflow scheduling model helps to schedule jobs in such a way that all the jobs will get executed by taking minimal possible time, maintaining Quality of Service, and satisfying client's requirements [15].

DAG which is also known as Directed Acyclic Graph provides data dependency among the tasks [139]. The overall completion time of an application is known as the schedule length or makespan. So, the objective of workflow scheduling techniques is to minimize the makespan while mapping tasks to VMS. Consider a workflow as W

(T, E). Here W represents workflow name, T is used for tasks of workflow and E represents edges between two tasks of input workflow [149].

Set of tasks of workflow can be taken as:

$T = \{T_1, T_2, T_3, T_4, \ldots\ldots\ldots T_{n-1}, T_n\}$

Set of edges used to access dependencies among various tasks can be represented as:

$E = \{< T_1, T_2>, < T_2, T_3>, < T_3, T_4 >, \ldots\ldots\ldots, < T_{n-1}, T_n> \}$

Here, T_{n-1} is the parent task of T_n. Figure 3.1 represents a simple DAG with 16 tasks.

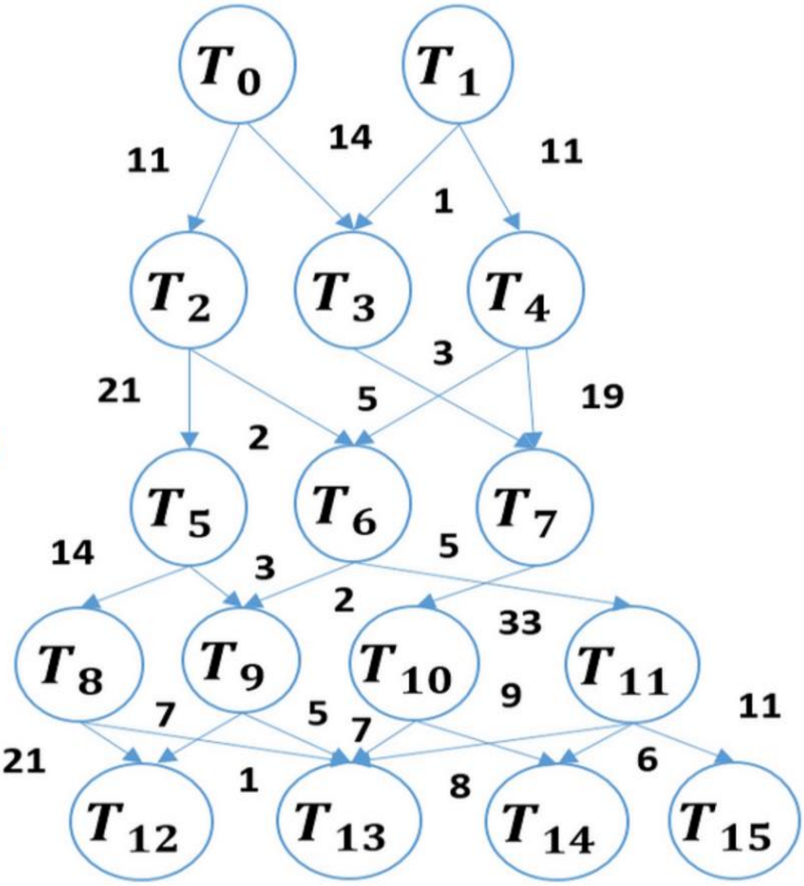


Figure 3.1: DAG representation

In a workflow, the root task or an entry task which is the first task of the workflow acts itself only as a parent but cannot be a child. Same like, the last task can only act as a child of some other task but cannot be a parent of any other task. In addition, the execution of a child task cannot initialize until all of its parent tasks are completed. A task is ready for execution if its entire parent tasks execution is finished.

The research aims at providing a better framework that will do scheduling in such a way that makespan, total execution cost, and response time of the system will be minimized. This system increases the performance efficiency as well.

3.2 Proposed Methodology

The proposed methodology is more effective in terms of workflow scheduling in cloud systems. This research works on 3 objectives. It targets effective task distribution on cloud resources. Also, it targets optimal scheduling for better performance. The scheduling and optimization algorithm consists of two phases as shown in figure 3.2.For the execution of phase 1 of the research, a collection of input workflow tasks is needed.For the execution of phase 2 of the research, the input required is ranked tasks that are mapped on cloud VMs successfully [48].This study is about making a cloud system more optimal. So, the framework has been developed in such a way so that various quality parameters can be achieved.

Total five workflows are accepted as input named MONTAGE, CYBERSHAKE, SIPHT, LIGO, and EPIGENOMICS. These are important scientific applications and uses datasets at a large scale. These datasets are from real experiments which are authentic [61]. The next step is Parsing. It is a step of analyzing input. In the proposed cloud workflow framework, parsing will occur at an initial stage. It will show tasks with dependencies. According to the critical path, all tasks of input workflow will be collected. After this, phase 1 of the system will start.

Phase 1: Task Ranking Phase

Phase 1 is to find out the rank value of each task of input workflow. In this phase, distributed HEFT task ranking algorithm has been proposed as shown in figure 2 for ranking various tasks of input workflow. Phases 1 is finding a better task ranking method and apply it to input workflow tasks to do ranking. It works on three heuristic parameters budget, time, and deadline. The distributed HEFT ranking method finds out the correlation between the above three parameters and then assigns the top rank to the task having a high distributed score among all tasks. Afterward, phase 2 of the hybrid approach works in which task scheduling on cloud VMs is performed.

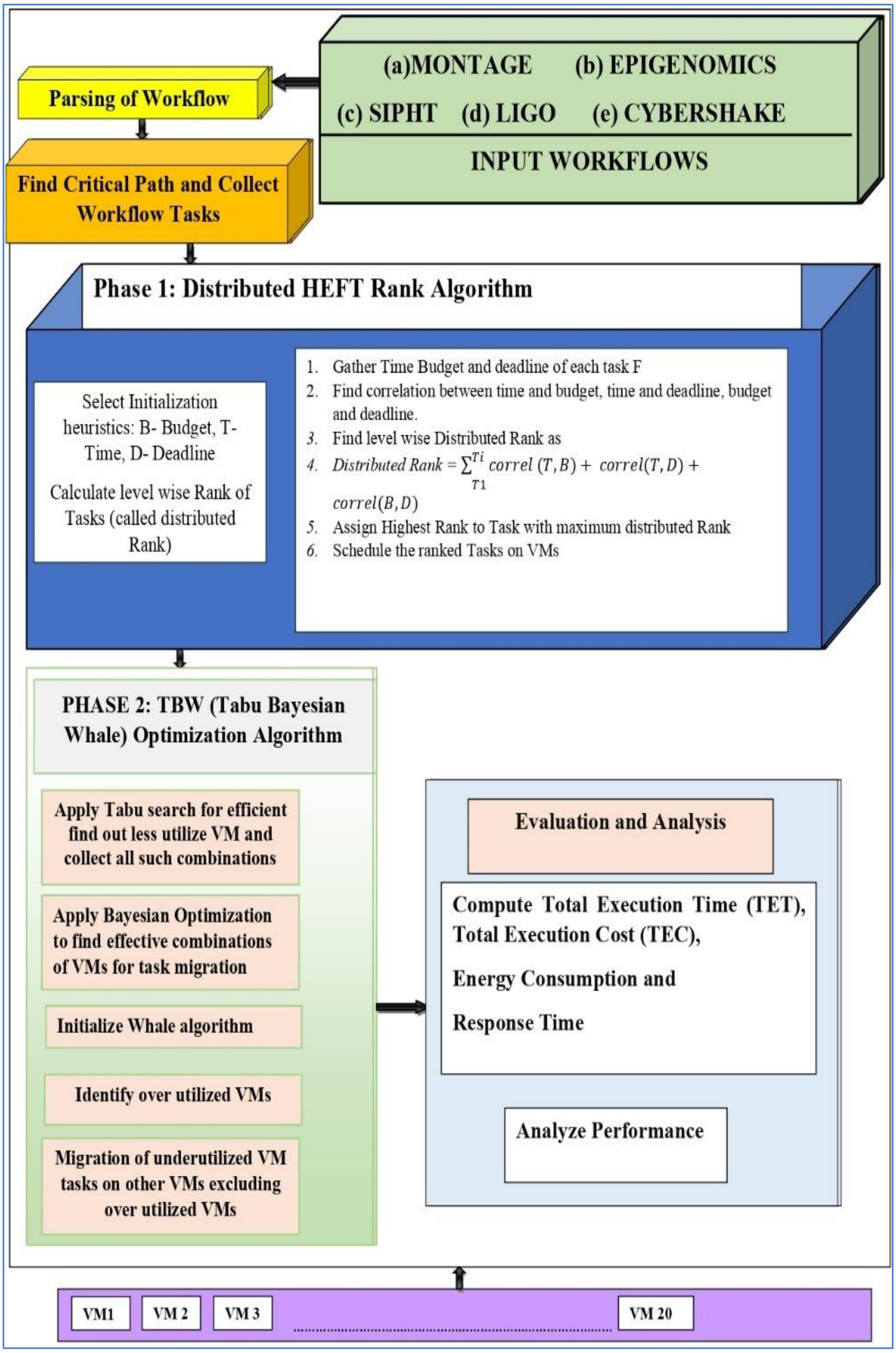


Figure 3.2 Proposed Scientific Workflow Scheduling Framework

This phase used the concept of correlation between input heuristics parameters time, budget, and deadline. Correlation can have a value in a range from -1 to +1. It is a relation between two parameters to express that how strong two parameters have a relation with each other. In the end, the task with high correlation is given maximum rank and the task with the least correlation is given minimum rank. Now, these ranked tasks are scheduled on cloud resources using the distributed-HEFT Rank method.

Phase 2: Cloud Optimization Phase

This phase is further using three optimization approaches to make the system more efficient.

First of all, it performs task migration in case underutilized machines are in the system which can support in saving energy consumption. During this process, an advanced optimization approach has been used which is based on Tabu search, Bayesian optimization, and whale optimization approach.

Tabu optimization helps to find out underutilized resources.

Afterward, the Bayesian optimization approach helps to provide a combination of VMs which are best suitable for tasks migration.

Whale optimization helps in the migration of tasks from underutilized machines to other ones but without an increase in time, cost, and response time[11, 78 and 192].

The performance evaluation and Analysis is also the main objective covered in this study. After performing VM migration in an optimized manner, Makespan, Cost, energy consumption, and response time of Workflow Execution are calculated [23, 25]. So, these four parameters will help to analyze the performance of the proposed framework. In the whole system, utilization of resources will be increased. The TBW approach has been proposed for mappings tasks to VMs in a cloud [34, 35, 45, 76, and 105].

In a Cloud System, there are myriad factors, which are responsible for making a cloud system as most satisfactory. For this, task Ranking and optimal task scheduling have become a vital need for making a cloud to meet all requirements.

3.3 Scope of the Research

The research aims at providing and designing a framework that will optimally perform scientific workflow scheduling to enhance the performance of the cloud system. The primary scope of the research is that rather than map the workflow tasks randomly on cloud systems, do apply an enhanced ranking methodology for ranking input tasks of the workflow. So mapping of tasks is performed after giving a rank to input tasks. Hence, the mapping can be kept optimally. Afterward, to enhance the overall performance of the system, migrations of tasks from less utilized resources to other resources have been done. During the whole process, makespan, cost, and response time have been minimized. This system increases the throughput and thereby increases performance efficiency.

3.4 Performance evaluation Parameters

In this research, the following parameters have been considered for simulation to evaluate the effectiveness of the proposed algorithms.For distributed HEFT rank algorithm, Total Execution Time (TET) and Total Execution Cost (TEC). To evaluate the effectiveness of the TBW (Tabu Bayesian Whale) optimization algorithm as used in phase 2 of the research, a total of four performance matrices are used in the simulation. These parameters are TET, TEC, energy consumption, and response time.

Total Execution Time

It is also known as makespan which is the total execution time of the workflow from start till its finish. It is inversely proportional to the performance of a scheduling algorithm.

Total Execution Cost

The cost consumed in running a workflow. It is inversely proportional to the performance of a scheduling algorithm.

Deadline

The maximum limit in which all tasks of a workflow need the execution.

3.5 System Development

The workflow applications are simulated using the CloudSim simulator toolkit which is easily extensible and it is an extension of the GridSim framework for simulation of resource provisioning and scheduling algorithms on cloud computing infrastructure.

Cloudsim

The proposed algorithm is simulated using a simulator named CloudSim. CloudSim is also known as cloud simulator is a simulation tool. It is based on Java language, especially for simulation of various applications as well as for the execution of cloud-based applications. This algorithm has been evaluated in an experimental cloud environment.CloudSim provides a virtualization engine to facilitate users with virtualization services. Figure 3.3 represents important features of cloudsim.

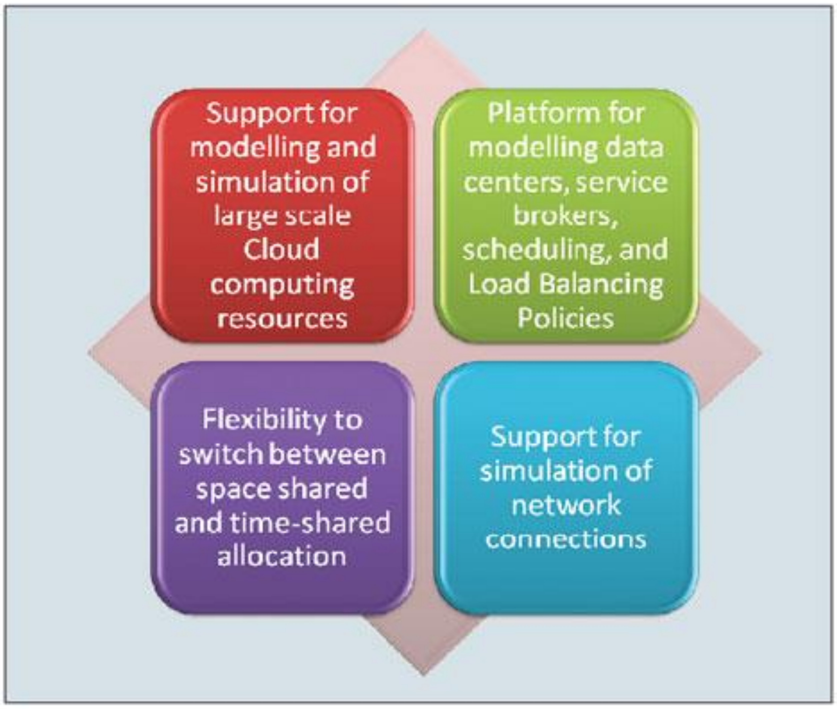


Figure 3.3: CloudSim Features

It can be defined as ‘run a model of an environment in a model of hardware’, with the abstraction of technology-specific details. CloudSim is such kind of platform which improves the cloud computing algorithms and techniques to run large data sets. The Cloudsim test bench executes one work at a time, according to the scheduled tasks which have an accurate estimation of the computation time and communication time of corresponds to workflow scheduling problems.It supports testing the performance. That is the main reason that cloud developer's faith in cloudsim. It can be used free of cost.It doesn’t run any actual software because it is a simulator.

3.6 Summary

This chapter is all about the complete framework of the proposed study. This framework includes the working of two algorithms, the first one is a ranking methodology used for providing rank to input tasks of the workflow and the second is mapping by migration of tasks from less utilized machines to other machines but without creating any queue on other VMs.

CHAPTER 4

WORKFLOW TASK RANKING ALGORITHM BASED ON DISTRIBUTED HEFT RANKING METHOD

4.1 Introduction

This chapter proposes an effective deadline constrained-based workflow Scheduling algorithm intending to minimize the time and cost of running workflow applications while reducing execution time in the cloud environment. The algorithm is evaluated using CloudSim with various well-known workflow applications of different sizes with various VMs [18, 123, and 130]. Result proves that the proposed algorithm has a better performance in terms of Total Execution Time (TET) and Total Execution Cost (TEC).

In past years of scientific discovery, the computational workflow continues to be popular among various disciplines of science, including Astronomy, Physics, Biology, Chemistry, and others. Cloud computing offers a wide computations and resources facility for the execution of workflows for various applications [132, 174].

It is highly acceptable that in the world of grid computing systems, cloud computing is an escalating trend. As it works without considering when and where, so its primary target is always to deliver services as per need. Delivering services is possible via Software as a Service, Platform as a Service, and Infrastructure as a Service. IaaS model is used by cloud environments when scientific workflows are in use [103, 106]. If we talk about scheduling of input workflow tasks, then the steps involved in list scheduling are:

1. Assign the priorities to each task based on some criteria. Criteria can be a high priority to shortest job, FCFS, earliest finish time, a task with high cost, a task with a maximum deadline, etc.
2. Arrange the tasks according to their priorities in an order.
3. Select the first task from the priority queue and allocate the selected resource

Above 3 steps are highly valuable as it works like better input for scheduling. Scheduling can be static or dynamic. If ranking is allocated before mapping for execution on cloud resources, then it comes under the category of static scheduling. But on the other side, while execution, if it is decided and priority is assigned then it is dynamic scheduling.

4.2 Importance of Ranking Workflow Tasks

As in this research, scientific workflows have been taken as input. Scientific workflows like GENOME, MONTAGE [16], and others include interconnected hundreds or more computational tasks. These tasks require large files of their data and their various instructions. Due to this reason, an optimal solution for mapping these tasks to cloud resources has become the main challenge. Therefore, to make this scheduling an optimal one, tasks, when received, are firstly ranked based on correlation factor, and then a further process of scheduling has started. The importance of the task ranking [187] phase is that it avoids the random distribution of tasks to VMs in the cloud system.

4.3 Background

4.3.1 HEFT Algorithm

The HEFT algorithm as described is a list heuristic-based algorithm. It works in two important phases. One is the assignment of priority to the phase of the task and another one is the Processor selection phase. The task prioritizing phase which is the first phase is used for calculating the ranks of all the tasks. The processor selection phase which is the second phase chooses the tasks in the order of their ranks or priorities and allocating tasks to their best-suited processor, which reduces the task's completion time.

Pseudo code for HEFT Algorithm

{

Step 1: Label all nodes with the computation cost.

Step 2: Label all edges with communication cost.

Step 3: Start from an exit node, traverse graph upward, and calculate $rank_u$ *of all tasks.*

Step 4: From highest to minimum $rank_u$ *values of tasks, sort the tasks in a scheduling list S.*

Step 5: While S is not empty

do

t ← remove the first task from S

r ← find a resource which can complete t as the earliest time

Schedule t to r.

end while

}

Phases of HEFT Algorithm

HEFT algorithm has two phases in which tasks are mapped on cloud resources. These phases are listed below:

- Task Prioritizing Phase
- Tasks assignment to machines Phase

Task Prioritizing Phase

Each task of the workflow is given a priority in this phase. It is assigned as an upward rank to each task. It is defined recursively as given in Eqn. 4.1.

$$\mathrm{rank_u}(\mathrm{n_i}) = \overline{\mathrm{w}}_\mathrm{i} + \max_{\mathrm{n_j} \in \mathrm{succ}(\mathrm{n_i})}\left(\overline{\mathrm{c}_{\mathrm{i,j}}} + \mathrm{rank_u}(\mathrm{n_j})\right) \quad (4.1)$$

In the above equation,n_i corresponds to the i^{th} task.

$\overline{w}_i$is an average computation cost of task i among all the processors.

$succ(n_i)$is the set of all jobs that immediately depend on task n_i.

$\overline{c_{i,j}}$is the average communication cost of the variable shift between jobs n_i and n_j between all pairs of machines.

Tasks assignment to machines Phase

It is the second phase of HEFT system and it targets tasks assignment to machines. It works after the execution of phase 1 as it takes prioritized tasks for scheduling to machines. It starts with the topmost priority.

That task is scheduled first on the machine which attains the highest priority and also for which all dependent tasks have finished. HEFT uses an insertion-based policy that fills sufficiently sized gaps between already scheduled tasks.

4.3.2 Fuzzy dominance sort based heterogeneous earliest-finish-time algorithm

Fuzzy logic is intended to solve problems in the similar way that human beings do: by taking into consideration all the available information and making the best achievable decision. It is one of the superior choices for several control problems [96].

Membership function terminology: There are various membership functions of fuzzy sets which are helpful to understand the importance of fuzzy rules and their implementation.

Universe of Discourse: It is the range of all possible values which are given as input to a fuzzy system.

Support: For a fuzzy set let named F, support is known as the crisp set of all points in the universe of discourse U. Its target is to attain membership function of F as non-zero. It is given in Eqn. 4.2.

Supp$A = \{x|\mu_A(x) > 0, \forall x \in X\}$ (4.2)

Core: For a fuzzy set let named F, the core is known as the crisp set of all points in the universe of discourse U. it is given in Eqn. 4.3. Its target is to attain membership function of F as 1.

Core $A = \{x|\mu_A(x) = 1, \forall x \in X\}$ (4.3)

Boundaries: For a fuzzy set let named F, boundaries are known as the crisp set of all points in the universe of discourse U as identified by Eqn. 4.4. Its target is to attain a membership function of F between 0 and 1.

Boundaries$A = \{x|0 < \mu_A(x) < 1, \forall x \in X\}\}$ (4.4)

Crossover point: In a fuzzy set let named F, a crossover point is the element in the universe of discourse U. Its target to provide its membership function is 0.5 i.e. $\mu(x) = 0.5$.

Height: In a fuzzy set, it is known as the biggest value of membership functions.

Normalized fuzzy set: It is a kind of fuzzy set of Height $(A) = 1$

Cardinality of the set: It is total number of elements in the set. In fuzzy sets, it is calculated as shown in Eqn. 4.5.

$$X: finite$$

$$|A| = \sum_{xeX} \mu_A(x) = \sum_{xeSupp(A)} \mu_A(x) \quad (4.5)$$

Relative cardinality: It is identified in Eqn. 4.6.

$$\|A\| = \frac{|A|}{|X|} \qquad (4.6)$$

Convex fuzzy set: Basically a fuzzy set is a class of membership mathematical objects which are continuous. A fuzzy set A *is* Convex, if for $\forall\lambda\epsilon[0,1]X\epsilon R$

$$\mu_A(\lambda x_1 + (1-\lambda)x_2) \geq min(\mu_A(x_1), (\mu_A(x_2)) \qquad (4.7)$$

Also the membership values are strictly monotonically increasing or strictly monotonically decreasing.

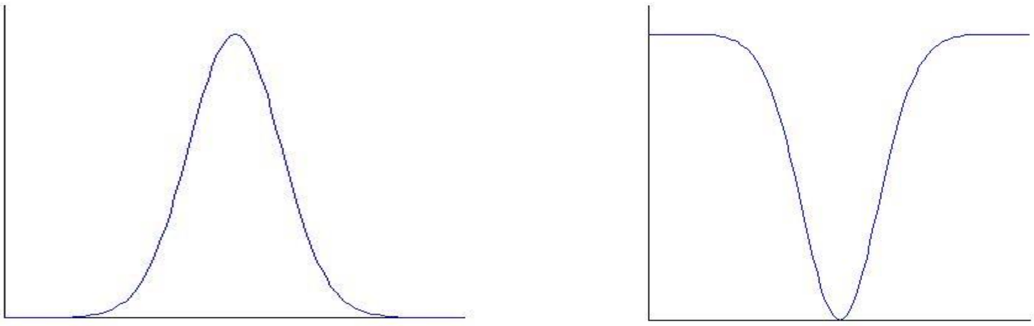


Figure 4.1: Convex Fuzzy Set

Type of membership functions

1. Its Numerical definition is its discrete membership functions as shown in Eqn. 4.8.

$$A = \sum_{x_i \in X} \mu_A(x_i)/x_i \qquad (4.8)$$

2. Function definition includes continuous membership functions as shown in Eqn.4.9.

$$A = \int_x \mu_A(x)/x \qquad (4.9)$$

It includes various functions as listed below:

- S function
- Z Function
- Pi function
- Triangular shape
- Trapezoid shape
- Bell shape.

S Function: Monotonically increasing membership function as shown with the Eqn. 4.10 and represented in figure number 4.2.

$$S\ (x;\alpha,\beta,\gamma) = \begin{cases} 0 & for\ x \le \alpha \\ 2\left(\dfrac{x-\alpha}{\gamma-\alpha}\right)^2 & for\ \alpha \le x \le \beta \\ 1-2\left(\dfrac{x-\alpha}{\gamma-\alpha}\right)^2 & for\ \beta \le x \le \gamma \\ 1 & for\ \gamma \le x \end{cases} \tag{4.10}$$

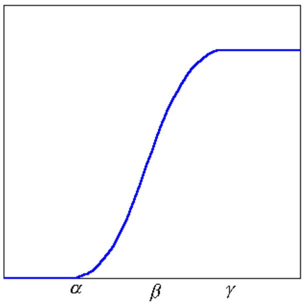


Figure 4.2: S Function in Fuzzy Logic

Z Function: It is a monotonically decreasing membership function. The equation of z function is as follows in Eqn. 4.11. Z function in fuzzy set is represented in figure 4.3.

$$\boxed{Z(x;\alpha,\beta,\gamma) = \begin{cases} 1 & for\ x \le \alpha \\ 1-2\left(\frac{x-\alpha}{\gamma-\alpha}\right)^2 & for\ \alpha \le x \le \beta \\ 2\left(\frac{x-\alpha}{\gamma-\alpha}\right)^2 & for\ \beta \le x \le \gamma \\ 0 & for\ \gamma \le x \end{cases}} \tag{4.11}$$

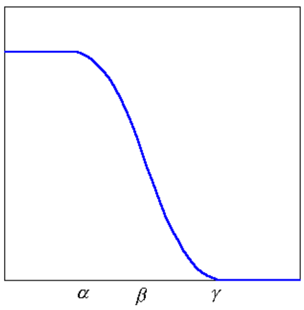


Figure 4.3: Z Function in Fuzzy Logic

Π Function: It is a combination of S function and Z function. The equation of Π function is as below in Eqn. 4.12 and figure 4.4.

$$\Pi(x,\beta,\gamma) = \begin{cases} S(x,\gamma-\beta,\gamma-\dfrac{\beta}{2},\gamma) & for\ x \le \gamma \\ 1-S(x,\gamma,\gamma+\dfrac{\beta}{2},\gamma+\beta) & for\ x \ge \gamma \end{cases} \tag{4.12}$$

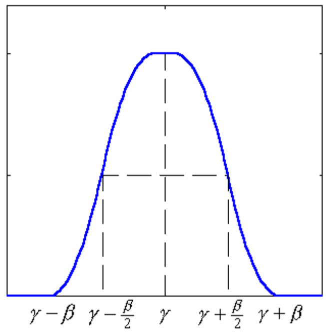


Figure 4.4: Π Function in Fuzzy Logic

Trapezoidal membership function: it is used to transform information in numerical form. Eqn. 4.13 and figure 4.5 defines trapezoidal membership function.

$$\mu_A(x) = \begin{cases} 0 & for\ x \le a_1 \\ \dfrac{x-a_1}{a-a_1} & for\ a_1 \le x \le a \\ 1 & for\ a \le x \le b \\ \dfrac{b_1-x}{b_1-b} & for\ b \le x \le b_1 \\ 0 & for\ b_1 \le x \end{cases} \tag{4.13}$$

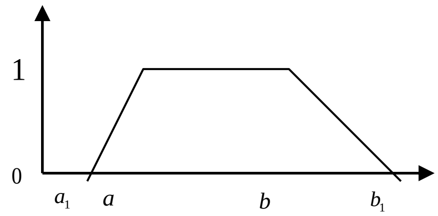


Figure 4.5: Trapezoidal membership function in Fuzzy Logic

Triangular membership function: It is as follows in Eqn. 4.14.Figure 4.6 shows its representation in fuzzy logic.

$$\mu_A(x) = \begin{cases} 0 & for\ x \le a_1 \\ \dfrac{x-a_1}{a-a_1} & for\ a_1 \le x \le a \\ \dfrac{b_1-x}{b_1-a} & for\ a \le x \le b_1 \\ 0 & for\ b_1 \le x \end{cases} \tag{4.14}$$

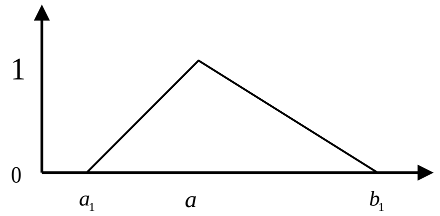


Figure 4.6: Triangular membership function in Fuzzy Logic

Bell-shaped membership function: It is as follows in Eqn. 4.15.Figure 4.7 shows its representation in fuzzy logic.

$$A(x) = ce^{-\frac{(x-a)^2}{b}} \tag{4.15}$$

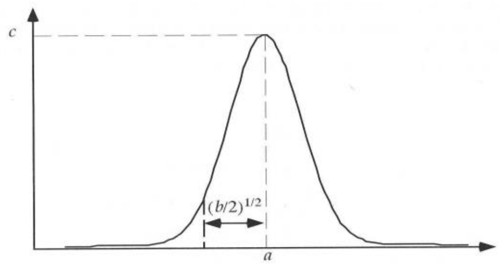


Figure 4.7: Bell-shaped membership function in Fuzzy Logic

Fuzzy logic handles the concept of partial truth- truth values between "completely true" and "completely false". It works on approximation rather than exact. Fuzzy logic is important because human sensing is also approximate in nature [143].

As fuzzy HEFT ranking method is an advance and modified approach of HEFT method. It applies fuzzy rules for ranking tasks. It provides better results in terms of total execution time and total execution cost. It provides rank to input workflow tasks based on the static threshold value. Figure 4.8 elaborates the ranking method of Fuzzy HEFT in cloud computing.

A heterogeneous FDHEFT algorithm is created, which includes the fuzzy dominance sorting system coupled with the Heuristic HEFT scheduling list [94]. The work searched joint cost optimization and proposed a novel workflow schedule. The effectiveness of the proposed system was demonstrated by extensive real-world and synthetic workflow experiments.

With remarkably greater hyper volume, the proposed workflow schedule delivered considerably better cost-effective fronts and could operate quicker than existing approaches.

A Task Ranking Algorithm based on tasks dependency and computation time has been designed and developed in this work. After parsing input workflow, ranking of all tasks is the next step in our study. Based on the distributed-HEFT score value, the level-wise rank value of input workflow tasks has been calculated before the actual scheduling process.

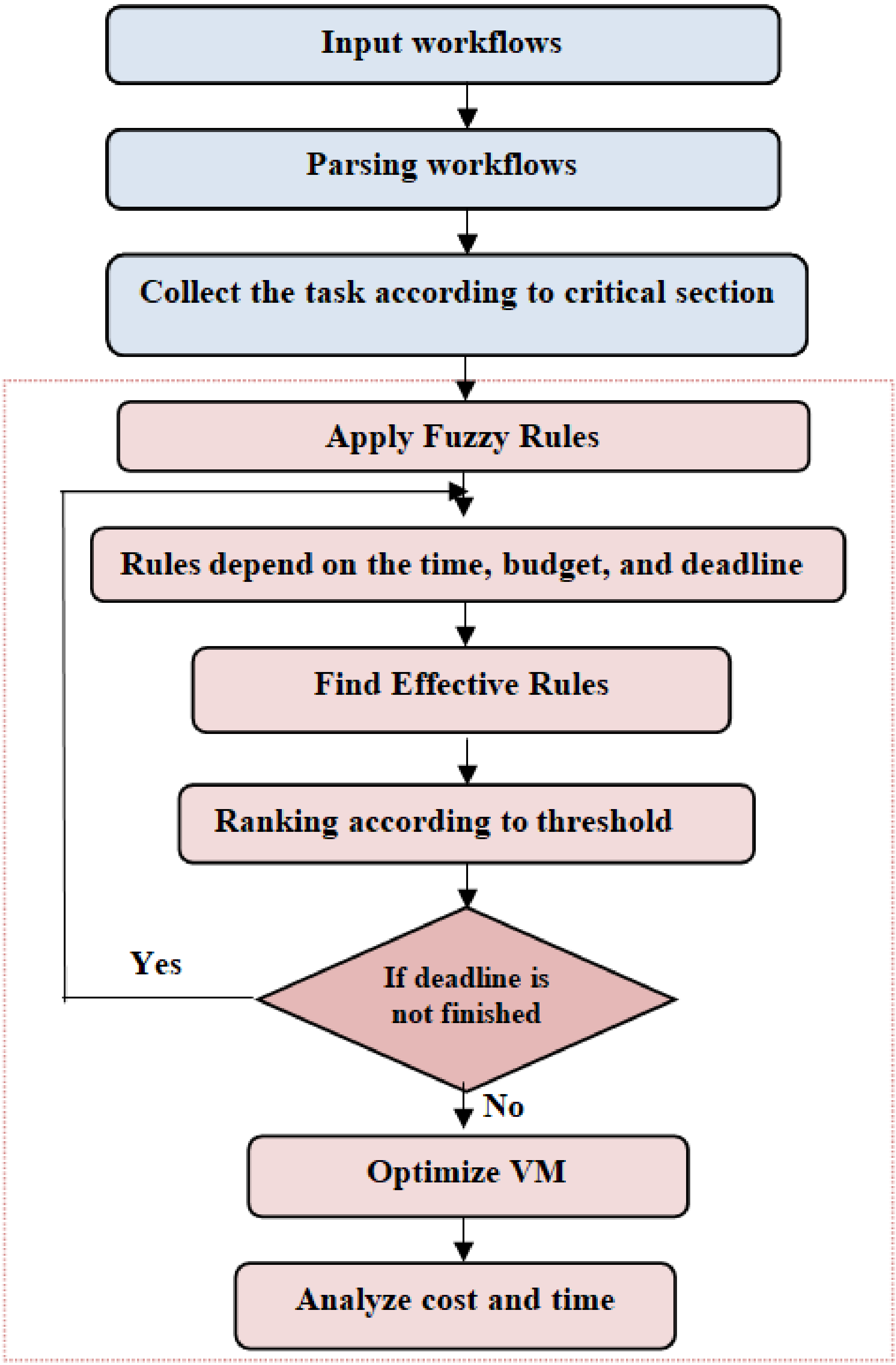


Figure 4.8: Fuzzy HEFT Task Ranking

4.4 Proposed Distributed HEFT Ranking Algorithm

The concern for ranking tasks firstly and then schedule the tasks are to save the total execution time of the system as well as its total execution cost. To check which ranking algorithm can effectively improve it, we accessed HEFT improvement as well as Fuzzy HEFT. Then we viewed the improvement given by distributed HEFT ranking system which gave better results in terms of TEC and TET.

Based on the concept behind Distributed HEFT ranking method, the implementation of the proposed algorithm has been explained in the following steps.

Select Initialization heuristics for performing task ranking algorithm:

Calculate level-wise Rank of Tasks (called distributed Rank) using the following steps.

1. Gather the Time, Budget, and deadline of each task.
2. Find correlation between time and budget, time and deadline, budget and deadline.
3. Find level-wise Distributed Rank as below in Eqn. 4.16.

$$\textit{Distributed Rank} = \sum_{T1}^{Ti} correl\,(T,B) + correl(T,D) + correl(B,D) \quad (4.16)$$

(Here, correl is correlation factor)

4. Assign Highest Rank to Task with maximum distributed Rank
5. Schedule the ranked Tasks on VMs in non-increasing order of their rank value.

Correlation can have a value in the range from -1 to +1. It is a relation between two parameters to express that how strong two parameters have a relation with each other. It works on three heuristic parameters budget, time, and deadline. The distributed HEFT ranking method finds out the correlation between the above three parameters and then assigns the top rank to the task having a high distributed score among all tasks. In the end, the task with high correlation is given maximum rank and the task with the least correlation is given minimum rank. Now, these ranked tasks are mapped to cloud resources for starting phase 2 of the framework.

By creating a correlation between time and deadline of input tasks, it has been evaluated that results are better than HEFT [8,9] method. In HEFT, the TET (Total execution Time) parameter of all input tasks is also calculated on cloudsim [38].

Pseudo Code for Distributed HEFT Ranking Method

Algorithm 1 Distributed HEFT Ranking Algorithm (W,D,B)

1. ***Begin***
2. *T = Time of each task execution, D= Deadline of tasks, B= Budget*
3. *Initialization Algorithm (W)* *//call Algorithm 2*
4. *R={} // R used for rank of tasks*
5. ***While (task$_i$> 0)***
6. *Compute distribution score of all tasks*
   Distribution score=$\sum_{i=0}^{N} \quad correl(T,B) + correl(T,D) + correl(B,D)$
   // D: deadline, B: Budget
7. *R= Distribution score (rank) //assign distribution score value to R*
8. *If optimize go to step 9 else go to step 5*
9. ***End whil**e*
10. *Schedule according to distribution score*
11. *Analysis of Scheduling*
12. *Analysis of the parameters total execution cost (TEC), total execution time (TET)*
13. ***End***

Algorithm 2 Algorithm for extracting input workflow tasks of W (W is the name of workflow)

1. ***Begin***
2. *Initialize number of resources and workflow*
3. *N = VM (number of resources)*
4. *W = Workflow*
5. *Parse input workflow*
6. *while (W)*
7. *start*
8. *Parse* W_i
9. *Extract tasks*
10. $T_i = critical\ path(W_i)$ *//call algorithm 3*
11. *End*
12. *X=* $\sum_{i=1}^{n} \quad T_i + W_i$
13. ***Return X***

14. End

Algorithm 3 Find Critical Path (W_i)

1. ***Begin***
2. *Cp=Ø, $task_i$ = $task_{exit}$,   // Here, Cp is Critical path*
3. ***While ($task_i$!= $task_{entry}$)***
   do
4. *Calculate critical parent $task_p$ of $task_i$*
5. $Criticalparent(taski) = \{taskp|maxtaskp\{finishtime+totaltime\}\}$
6. *Cp=Cp U $task_p$*
7. *$task_i$= $task_p$*
8. ***End While***
9. ***Return Cp***
10. ***End***

To start the ranking process of input workflow, algorithm 1 has been called. Here, T is workflow task execution time. D is deadline of tasks and B is budget of input workflow tasks.In statement 3, algorithm 2 for extraction of input workflow tasks has been called. Here, N is total VMs taken as input and W is workflow selected.

Statement 8 of the algorithm 2 is used for parsing step by step various tasks of workflow. Statement 10 is used to call algorithm 3 which is based on critical path selection of input workflow tasks. Here, $task_i$ is used to represent each task of the workflow iteratively. $Task_{entry}$ is used to represent first topmost task of the workflow. Then it calculated the distributed score value for each task which is based on correlation between time, budget, and deadlines. Variable R is used for this score value.

4.5 Experiment Results and Analysis

In Distributed HEFT ranking system. Based on TEC and TET, input scientific workflows are validated. The simulation experiments have been implemented on cloudSim simulator. To access comparative results, ranked tasks with HEFT, Fuzzy HEFT and distributed HEFT methodology have been scheduled on cloud resources [13].

Comparison of the proposed ranking method with methods already in use:

Proposed method- Distributed HEFT

The existing method for comparison- Fuzzy_HEFT and HEFT

The experimental results show positive results for the suggested method. Comparing the scheduled time, efficiency, and schedule length with other well-known task scheduling algorithms shows the success of the proposed heuristic. The comparison of the proposed task ranking method named Distributed HEFT ranking method has been done using various scientific workflows listed below:

- MONTAGE
- CYBERSHAKE
- LIGO
- GENOME
- SIPHT

In all cases, the comparison has been done on two parameters which are:

- Comparison based on TEC (Total Execution Cost)
- Comparison based on TET (Total Execution Time)

As shown in Table 4.1, the simulation-based outcome has been represented for SIPHT workflow where TET and TEC based comparison of Distributed HEFT task ranking method is done with existing methods.

Table 4.1: Comparison of the proposed Ranking method based on TET and TEC parameters using SIPHT workflow

Algorithm →	**HEFT**		**FUZZY_HEFT**		**Distributed HEFT**	
No of VMs ↓ / Parameter →	**TET**	**TEC**	**TET**	**TEC**	**TET**	**TEC**
2	24.06	5621.855	6.59	4551.063	5.39	4349.063
4	42.16	10010.35	13.26	8604.69	12.06	8402.69
6	39.05	13457.19	15.13	12126.43	13.93	11924.43
8	53.98	13524.39	35.83	14266.19	34.63	14064.19
10	88.61	17541.09	30.77	16959.17	29.57	16757.17
12	63.5	20457.6	33.34	19966.58	32.14	19764.58
14	53.64	21666.64	34.06	22433.37	32.86	22231.37
16	63.21	25786.99	56.59	30113.12	55.39	29911.12

18	94.3	33483.56	76.36	33231.33	75.16	33029.33
20	73.04	34113.99	63.55	33013.34	62.35	32811.34

It is clear from table 4.1 that TET as well as TEC values of Distributed HEFT for VMs from 2 to 20 is less than TET, TEC of Fuzzy HEFT and of HEFT algorithms. Figure 4.9 onwards in this chapter provides a graphical representation of cost or time parameters for different scientific workflows.

Colour indications in resultant graphs are as below:

- Blue color expressing HEFT algorithm,
- Orange color expressing Fuzzy HEFT algorithm and
- Green Color expressing Distributed HEFT

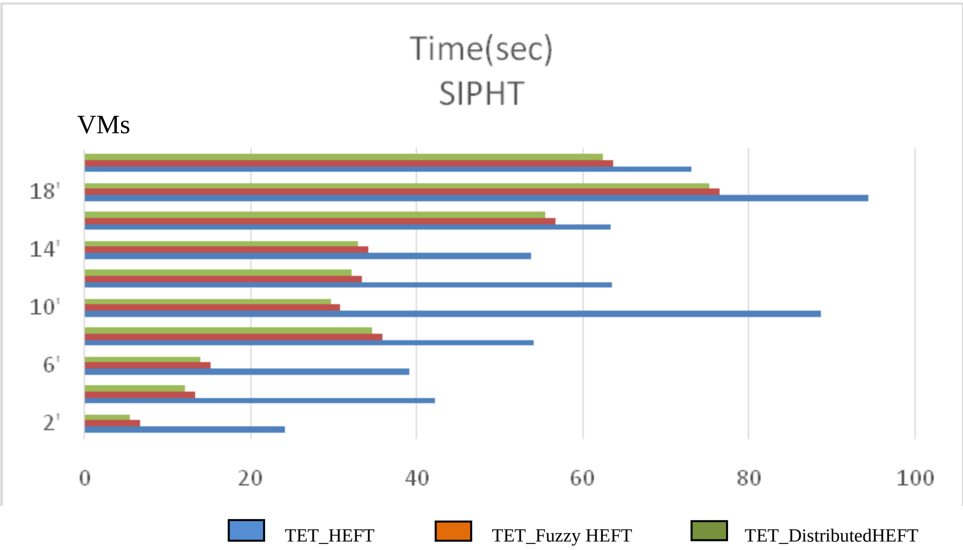


Figure 4.9: Comparison based on TET for SIPHT Workflow

In figure 4.9 as shown above, parameter Total Execution Time (TET) based comparison has been performed using data sets of SIPHT scientific workflow. The X-axis of the figure expresses the time in seconds and the y-axis expresses the total number of VMs used. It is clear that the time taken by Distributed HEFT algorithm to execute workflow tasks on 2 to 20 VMs is less than Fuzzy HEFT and HEFT algorithms.

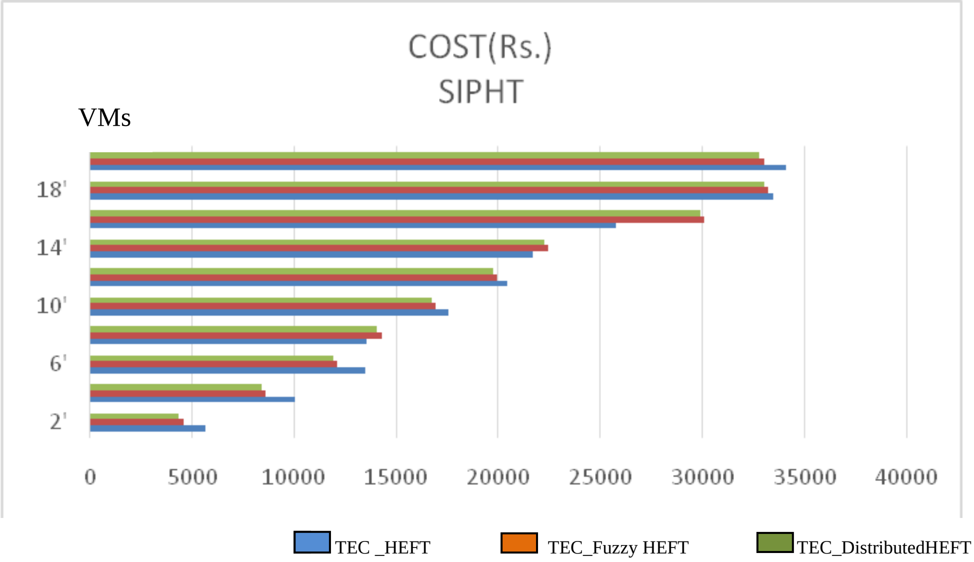


Figure 4.10.: Comparison based on TEC for SIPHT Workflow

As shown in figure 4.10, parameter Total Execution Cost (TEC) based comparison has been performed using data sets of SIPHT scientific workflow. The X-axis of the figure expressed cost in rupees and the y-axis expresses the total number of VMs used.

Table 4.2: Comparison of the proposed Ranking method based on TET and TEC parameters using MONTAGE workflow

Algorithm →	**HEFT**		**FUZZY_HEFT**		**Distributed HEFT**	
No of VMs↓ / Parameter →	**TET**	**TEC**	**TET**	**TEC**	**TET**	**TEC**
2	1.0	1.0	1.0	1.0	1.0	1.0
4	16.1	448.4	2.7	301.0	1.9	190.0
6	18.7	629.3	2.8	358.0	2.0	247.0
8	22.0	732.0	5.0	497.5	4.2	386.5
10	26.5	1094.2	5.8	497.5	5.0	386.5
12	28.9	1018.4	7.8	1148.3	7.0	1037.3
14	32.2	1330.8	8.6	1132.7	7.8	1021.7
16	33.2	1289.6	10.5	1481.9	9.7	1370.9
18	36.1	1539.4	9.7	1280.0	8.9	1169.0
20	37.3	1419.9	12.4	1746.8	11.6	1635.8

Same as the case of SIPHT, MONTAGE workflow has been accepted as input workflow. As shown in table 4.2, Distributed HEFT method has provided better results in the case of managing cost and time.

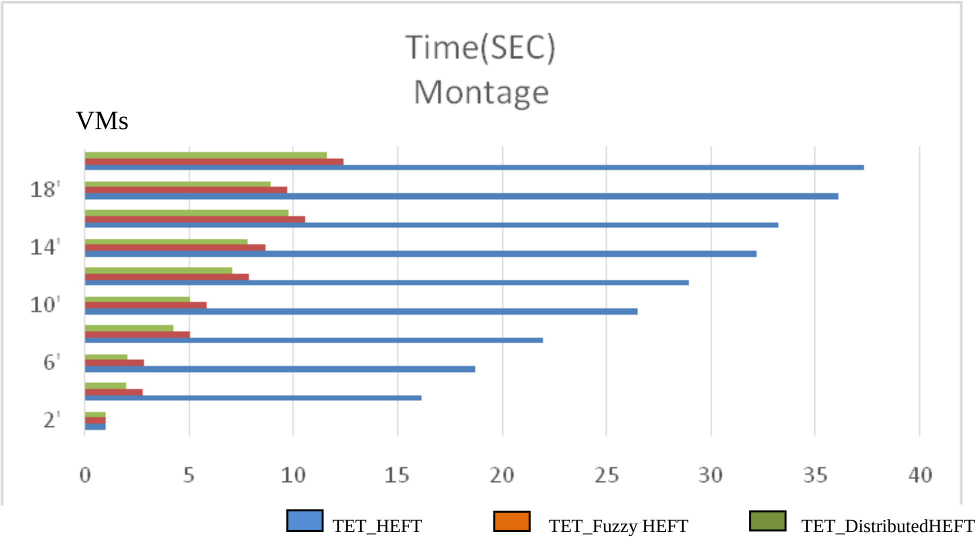


Figure 4.11: Comparison based on TET for MONTAGE Workflow

As shown in figure 4.11, parameter Total Execution Time (TET) based comparison has been performed using data sets of MONTAGE scientific workflow. The X-axis of the figure represents time in seconds and the y-axis expresses the total number of VMs used. Here, it is clear that time taken by HEFT algorithm to execute workflow tasks on VMs from 2 to 20 is high than Fuzzy HEFT and Distributed HEFT algorithm. Also distributed HEFT method is an improvement in time consumed during mapping tasks on cloud VMs.

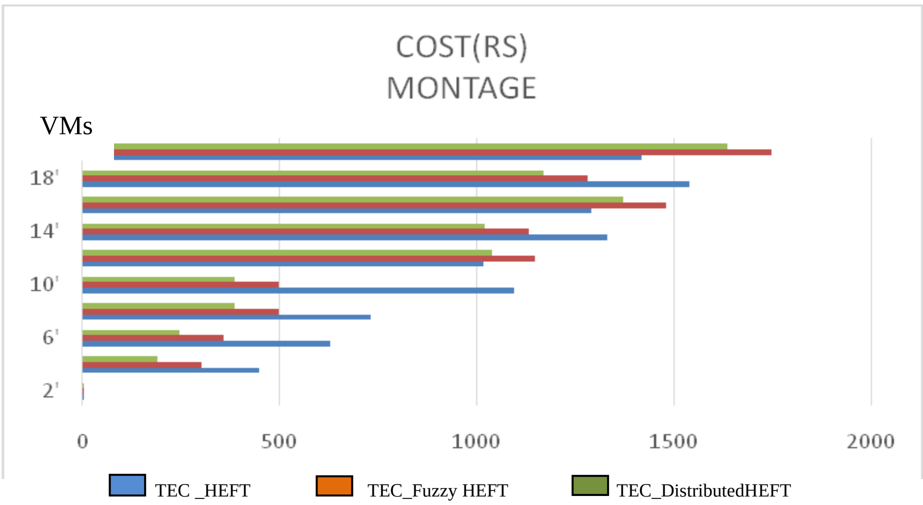


Figure 4.12: Comparison based on TEC for MONTAGE Workflow

As shown in figure 4.12, parameter Total Execution Cost (TEC) based comparison has been done using data sets of MONTAGE scientific workflow. The X-axis of the figure represents cost in rupees and the y-axis expresses the total number of VMs used tasks scheduling. Here, it is clear that cost consumed by HEFT algorithm to execute workflow tasks on VMs from 2 to 20 is high than Fuzzy HEFT and Distributed HEFT algorithm. Also distributed HEFT method is an improvement in cost consumed during mapping tasks on cloud VMs.

Table 4.3: Comparison of the proposed Ranking method based on TET and TEC parameters using CYBERSHAKE workflow

Algorithm →	**HEFT**		**FUZZY_HEFT**		**Distributed HEFT**	
No of VMs ↓ / Parameter →	**TET**	**TEC**	**TET**	**TEC**	**TET**	**TEC**
2	1.0	1.0	1.0	1.0	1.0	1.0
4	19.6	823.7	3.4	689.0	2.7	587.0
6	25.3	1138.0	4.4	952.0	3.7	850.0
8	27.1	1547.0	7.9	1519.4	7.2	1417.4
10	30.9	1884.2	5.5	1690.7	4.8	1588.7
12	31.8	1730.4	9.1	1725.1	8.4	1623.1
14	37.8	2531.3	12.0	2464.7	11.3	2362.7
16	35.3	2107.6	10.3	2732.9	9.6	2630.9
18	38.8	2312.3	12.6	1991.0	11.9	1889.0
20	40.5	2542.9	12.4	1983.9	11.7	1881.9

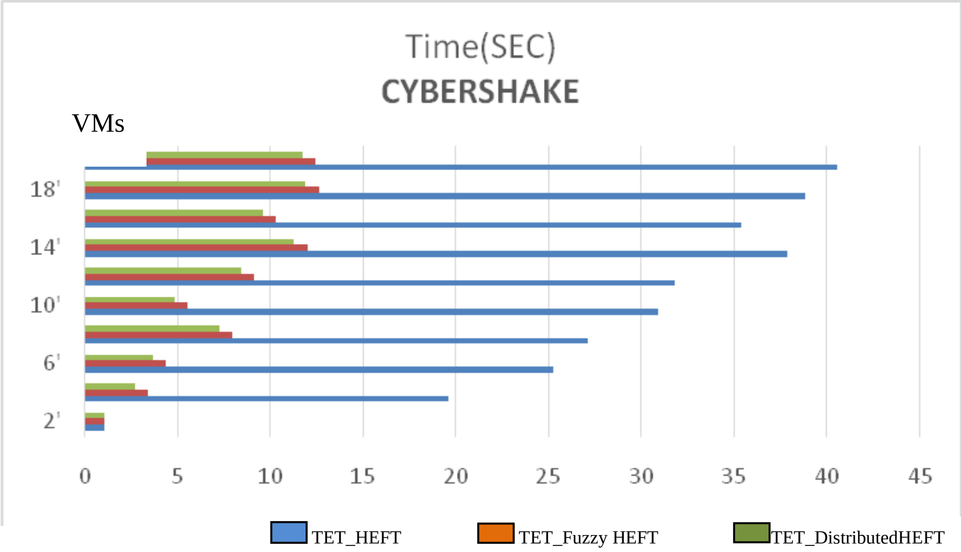


Figure 4.13: Comparison based on TET for CYBERSHAKE Workflow

As shown in figure 4.13, parameter Total Execution Time (TET) based comparison has been performed using data sets of CYBERSHAKE scientific workflow. The X-axis of the figure represents time in seconds and the y-axis expresses the total number of VMs used. Here, it is clear that time taken by HEFT algorithm to execute workflow tasks on VMs from 2 to 20 is high than Fuzzy HEFT and Distributed HEFT algorithm. Also distributed HEFT method is an improvement in time consumed during mapping tasks on cloud VMs.

Figure 4.14 depicts parameter Total Execution Cost (TEC) based comparison for CYBERSHAKE scientific workflow. The X-axis of the figure represents cost in rupees and the y-axis represents the total number of VMs used for tasks scheduling. Here, it is examined that the cost consumed by HEFT algorithm to execute workflow tasks on VMs from 2 to 20 is high than Fuzzy HEFT and Distributed HEFT algorithm. Also distributed HEFT method is an improvement in cost consumed during mapping tasks on cloud VMs.

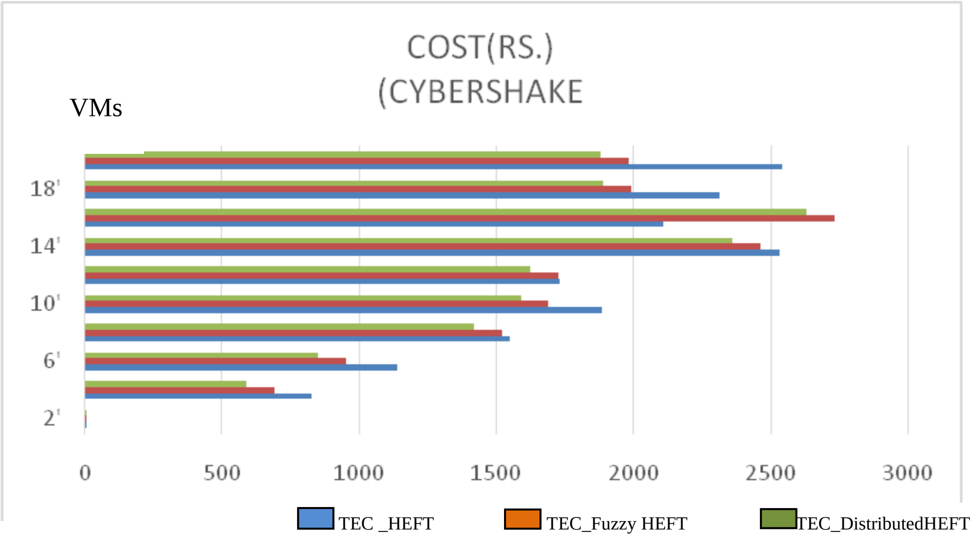


Figure 4.14: Comparison based on TEC for CYBERSHAKE Workflow

Table 4.4: Comparison of the proposed Ranking method based on TET and TEC parameters using LIGO workflow

Algorithm →	**HEFT**		**FUZZY_HEFT**		**Distributed HEFT**	
No of VMs↓ / Parameter →	**TET**	**TEC**	**TET**	**TEC**	**TET**	**TEC**
2	1.0	1.0	1.0	1.0	1.0	1.0
4	45.9	2763.6	7.6	1468.2	5.5	1380.2
6	66.9	4192.7	25.1	4023.4	23.0	3935.4
8	36.7	4884.2	46.3	5701.0	44.2	5613.0
10	69.8	5305.8	17.0	4709.4	14.9	4621.4
12	103.4	7332.1	27.7	6272.5	25.6	6184.5
14	60.0	7731.0	34.7	7551.5	32.6	7463.5
16	152.3	10609.9	50.6	9949.2	48.5	9861.2
18	88.8	11385.1	73.0	11189.0	70.9	11101.0
20	154.2	13095.9	91.9	12583.8	89.8	12495.8

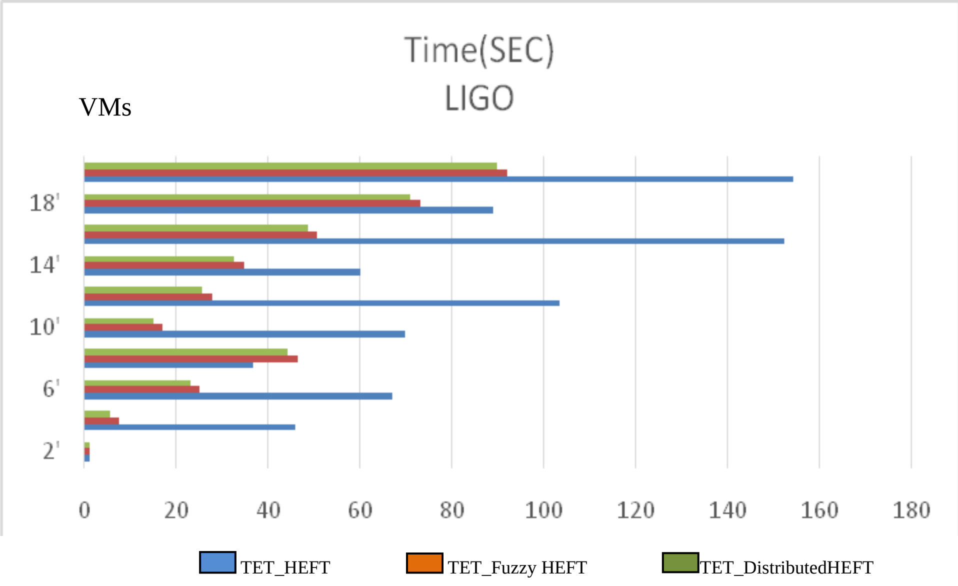


Figure 4.15: Comparison based on TET for LIGO Workflow

As shown in figure 4.15, parameter Total Execution Time (TET) based comparison has been performed using data sets of LIGO scientific workflow. The X-axis of the figure represents time in seconds and the y-axis expresses the total number of VMs used. Here, it is clear that time taken by HEFT algorithm to execute workflow tasks on VMs from 2 to 20 is high than Fuzzy HEFT and Distributed HEFT algorithm. Also distributed HEFT method is an improvement in time consumed during mapping tasks on cloud VMs.

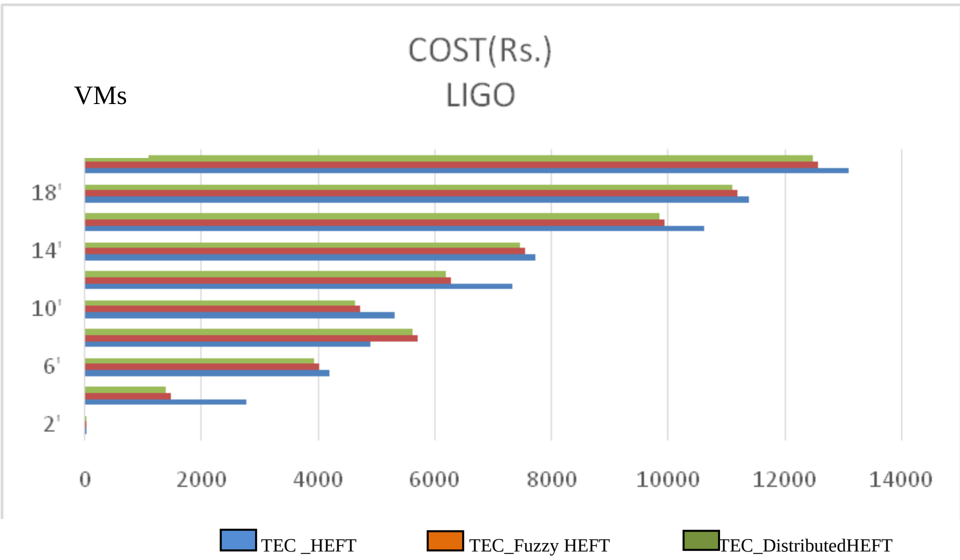


Figure 4.16: Comparison based on TEC for LIGO Workflow

Figure 4.16 depicts parameter Total Execution Cost (TEC) based comparison for LIGO scientific workflow. The X-axis of the figure represents cost in rupees and the y-axis represents the total number of VMs used for tasks scheduling. Here, it is examined that the cost consumed by HEFT algorithm to execute workflow tasks on VMs from 2 to 20 is high than Fuzzy HEFT and Distributed HEFT algorithm. Also distributed HEFT method is an improvement in cost consumed during mapping tasks on cloud VMs.

Table 4.5: Comparison of the proposed Ranking method based on TET and TEC parameters using GENOME workflow

Algorithm →	**HEFT**		**FUZZY_HEFT**		**Distributed HEFT**	
No of VMs ↓ /Parameter →	**TET**	**TEC**	**TET**	**TEC**	**TET**	**TEC**
2	12.2	8825.2	23.2	9915.5	22.0	9795.5
4	137.8	7549.6	73.2	27030.7	72.0	26910.7
6	201.6	42511.9	133.8	42569.7	132.6	42449.7
8	370.2	28155.5	315.7	41656.3	314.5	41536.3
10	284.5	58872.2	745.2	86234.9	744.0	86114.9
12	365.5	77034.6	546.8	88012.7	545.6	87892.7
14	423.8	66005.0	317.8	93654.4	316.6	93534.4
16	486.7	82720.6	402.0	106579.9	400.8	106459.9
18	530.1	113448.6	791.0	123916.8	789.8	123796.8
20	419.1	119935.5	491.2	139920.0	490.0	139800.0

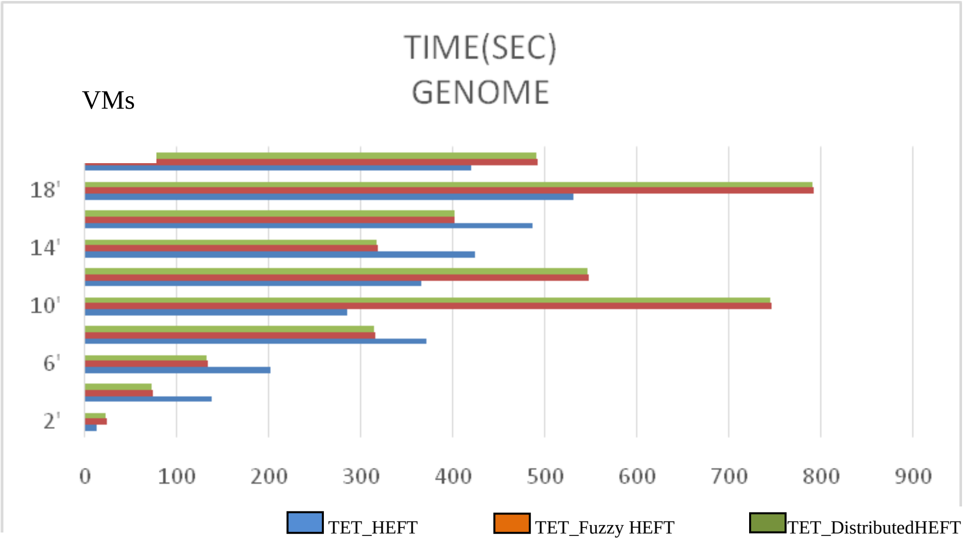


Figure 4.17: Comparison based on TET for GENOME Workflow

As shown in figure 4.17, parameter Total Execution Time (TET) based comparison has been performed using data sets of GENOME scientific workflow. The X-axis of the figure represents time in seconds and the y-axis expresses the total number of VMs used. Here, it is clear that time taken by HEFT algorithm to execute workflow tasks on VMs from 2 to 20 is high than Fuzzy HEFT and Distributed HEFT algorithm. Also Distributed HEFT method is an improvement in time consumed during mapping tasks on cloud VMs.

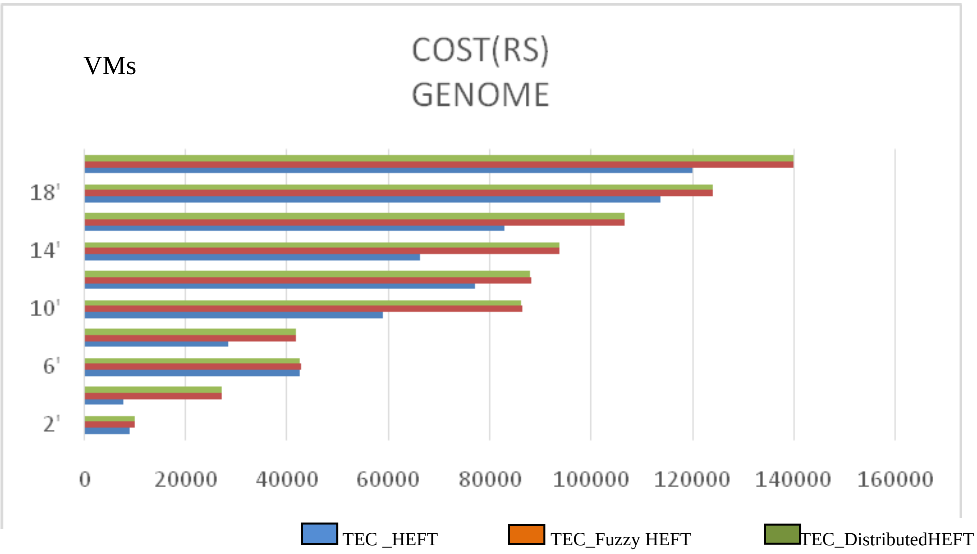


Figure 4.18: Comparison based on TEC for GENOME Workflow

Figure 4.18 depicts parameter Total Execution Cost (TEC) based comparison for GENOME scientific workflow. The X-axis of the figure represents cost in rupees and the y-axis represents the total number of VMs used for tasks scheduling. Here, it is examined that the cost consumed by HEFT algorithm to execute workflow tasks on VMs from 2 to 20 is high than Fuzzy HEFT and Distributed HEFT algorithm. Also distributed HEFT method is an improvement in cost consumed during mapping tasks on cloud VMs.

From experimental analysis, we have found that the distributed HEFT is more efficient as compared to Fuzzy HEFT and HEFT-based ranking techniques.

4.6 Summary

In this chapter, a ranking algorithm is proposed for the scientific workflows that are highly effective for ranking tasks of input workflows. The proposed ranking algorithm is advance to the existing fuzzy HEFT algorithm. It minimized total execution time which is represented with TET in the above graphs and total execution cost represented as TEC in the above results-based graphs. For getting these results, mapping of tasks to VM resources has been done. The proposed algorithm named distributed HEFT has provided a better ranking system and so has minimized TEC and TET while scheduling tasks to VMs. In the future, we aim at providing enhanced scheduling algorithm to provide better results.

CHAPTER 5

OPTIMIZING CLOUD USING TABU BAYESIAN WHALE OPTIMIZATION TECHNIQUE

5.1 Introduction

This chapter is all about improving the utilization of cloud resources as well as enhancing the performance of the cloud system. The experiment results illustrate a better outcome as the proposed algorithm decreases the total execution time, total execution cost in comparison with HEFT, Fuzzy-HEFT, GA, PSO, GA-PSO, and Whale optimization. Besides, it improves energy consumption and response time [173].

5.2 Need for Task Migration

Task Migration

It is the process of migrating executing tasks from one host processor to another. Selecting a resource to place tasks is also a challenge in a cloud task mapping system.

Several issues need to be focused on while performing task migration as it is a kind of process which provides several benefits to cloud providers.

Benefits of task Migration

In a cloud system, task migration provides various benefits. Some of the benefits are described below.

Load Balancing

In a system, if the load on various resources is balanced then automatically it will enhance the performance of a system.

Resource access

Indeed, all resources are not available at all times across the network. So for access to a device, sometimes task migration is required.

5.3 Proposed Algorithm

5.3.1 TBW Optimization in Cloud Environment

There is certainty to say that a cloud is an arrangement that allows any person to be its continuous consumer by providing maximum contentment to each being. Yet for optimizing it appropriately, it should be having maximum quality for all its services for both cloud providers as well as at the end of cloud users. Furthermore, rather than applying only workflow scheduling at run time, it is much supportive if the same is applied at its pre-stage. This process is explored in figure 5.1.

The explanation of the proposed work with the algorithm and flow chart of the methodology in detail is provided in this division. The concept of scheduling for optimization is implementation on the cloud. It is accepted on IaaS (Infrastructure-as-a-Service). In a further study, it is acknowledged that a total of 20 VMs are proposed to be created at a time on IaaS. It means tasks assignment to VMs should be in perfection.

The best part of this proposed stride is more towards cloud best utilization in terms of VMs. So, more and more the VMs are utilized in terms of resources, the best is cloud and highly recommendable by all.

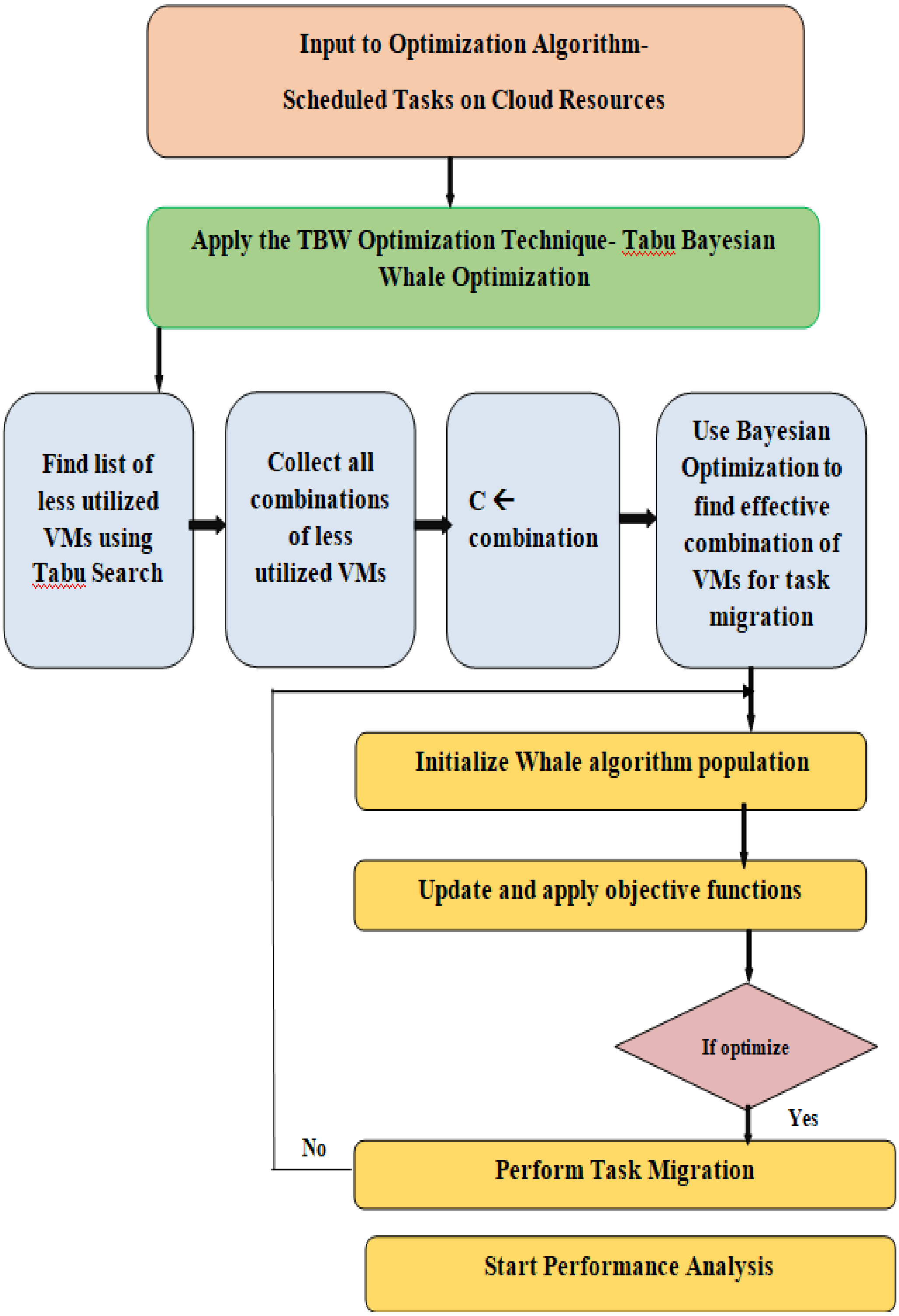


Figure 5.1: TBW Optimization Technique in a cloud environment

Methodology Steps

- Input the workflows
- Parse the tasks
- Ranking of tasks
- Provides the virtual machines according to ranking-based paths
- Initialize the optimization
- Update the status of the fitness function
- Check the output is optimized or not. If yes then analyze otherwise again initialize
- Analyze the total resource utilization

TBW algorithm is denoted as optimization algorithm as it works step by step to find an optimum solution based on objective functions.

These all steps are a part of the schedule which is a major measurement to manage a cloud suitably and resourcefully so that not any Virtual Machine should be free for a while even and not even any virtual machine should have tasks assignment in a queue. A scheduler decides which task/job should go to which machine. The whole process is executed on one server. Two main operations performed in the setup are: searching underutilized VMs and searching over-utilized VMs.

TBW algorithm is capable of avoiding local optima. So, it is suitable for most of the applications which occur practically. Apart from this, no alterations are performed in the algorithm for solving different constrained or unconstraint optimization problems.

The first type of force attracts the nodes towards these nodes, and the second type of force pulls them towards the neighbourhood nodes on the path. The movement of vertexes is determined by using Eqn. 5.1.:

$$\Delta Y_j = \alpha \sum w_{ij} \left(X_i - Y_j \right) + \beta K \left(Y_{j+1} - 2Y_j + Y_{j-1} \right) \tag{5.1}$$

Where X_i is x coordinate of the customer i and Y_j is y coordinate of node j, w_{ij} is the coefficient of the interaction between node i and node j. α and β are the constant weights.

$$w_{ij} = \frac{\phi\left(d_{X_iY_j}, K\right)}{\sum_k \phi\left(d_{X_iY_k}, K\right)}$$

Where $d_{X_iY_j}$ is the Euclidean distance between node i and node j and here the function ϕ is Gaussian function denoted as $\exp\left(-\frac{d^2}{2K^2}\right)$. Since w_{ij} is the derivation of the energy function E, it implies that this model would be gradient descent to reach the global optimization. Furthermore, the update function is shown in Eqn. 5.2:

$$E = -\alpha K \sum_i \ln \sum_j \phi\left(d_{X_iY_j}, K\right) + \sum_j \left|Y_{j+1} - Y_j\right|^2 \text{ (3)} \qquad (5.2)$$

This model analyzes the route formulation with a new point of view, such that it can provide us with a view that the 'virtual' force imposed on nodes playing an important role to formulate a route.

The shortest-path tree was firstly formulated to prepare the routes for the set covering model. 2-cycle elimination procedure was applied in shortest path tree algorithm to get the minimum marginal cost from depot to the node i defined as $H_i(q,t)$ shown in Eqn. 5.3.

$$H_d(0,0) = 0;$$

$$H_j(q,t) = \min \left\{ \begin{array}{l} \left[H_i\left(q', t'\right) + \bar{c}_{ij} \middle| j \neq p_i\left(q', t'\right), t' + t_{ij} \leq t, a_i \leq t' \leq b_i \ and \ q' + q_j \leq q \right], \\ \left[H_i'\left(q', t'\right) + \bar{c}_{ij} \middle| j = p_i\left(q', t'\right), t' + t_{ij} \leq t, a_i \leq t' \leq b_i \ and \ q' + q_j \leq q \right] \end{array} \right\}$$

(5.3)

Where a_i *and* b_i are the lower bound and upper bound of the time windows, q is the demand of all customers, t is the time to reach customer j, t' is the time to reach customer I, q' is the demand of all previously visited customers.

For all j, q, t such that $j \in N, q_j \leq q \leq Q \ and \ a_j \leq t \leq b_j$. $p_i(q,t)$ is the predecessor of node i associated with $H_i(q,t)$.

The original set covering model can be described using Eqn. 5.4 as below:

$$
S_M : Min \sum_{r \in R} c_r x_r
$$

$$
\left.\begin{aligned}
&\sum_{r=1}^{R_M} r_{ir} x_r \geq 1, \quad \forall i = 1,2,\dots,n && (1)\\
&\sum_{r \in R} x_r - X_d = 0 && (2)\\
&\sum_{r \in R} c_r x_r - X_c = 0 && (3)\\
&x_r \in \{0,1\} \quad \forall r = R && (4)
\end{aligned}\right\} \qquad (5.4)
$$

$$
X_d, X_c \geq 0, \quad Integer
$$

where X_d is the number of routes, X_c is the total travelled distance, r_{ir} = 1 if customer i will be visited by path r otherwise 0, c_r is the cost of route r, x_r is the binary integer value equal to 1 if path r is used and 0 otherwise, R is the feasible route set.

(1) There will be at least one route serving each customer. (2) (3) ensure that the number of customers and distance travelled are integer.

Then the dual can be available by imposing π_i, π_d, π_c multipliers on the constraints (1), (2), and (3), which turn out as expressed in Eqn. 5.5 below.

$$
\overline{c_r} = c_r - \sum_{i \in N \backslash (d)} \pi_i r_{ir} - \pi_d - \pi_c c_r \qquad (5.5)
$$

Since the route $(i_{0,} i_1, \cdots i_k, i_{k+1})$ cost was defined as $\sum_{k=0}^{K} c_{i_k i_{k+1}}$, then the reduced cost can be defined as shown in Eqn. 5.6.

$$
\overline{c_{ij}} = (1 - \pi_c) c_{ij} - \pi_i \qquad (5.6)
$$

It is mandatory that not any task should go in the waiting queue and not even any resource should go underutilized. It means, to increase resources utilization, it is a high requirement that scheduling should be pre-planned. The cloud promotion is dependent on valuable products supply as per demand and as per quick need. Every individual wants a facility that should be cost-effective as well as well planned. Apart from this, the selection of the right cloud is also a facility among consumers of cloud.

In today's advanced world of technology, cloud and its applications have provided various techniques. While the literature survey, many papers have been studied related to workflow mapping of tasks to the resources in cloud environment. It has been analyzed that it depends on the static configuration of a virtual machine, which is not a real condition. Rather than only work for managing cloud resources, it has been formulated that betterment in terms of cost and time saving must be important parameters. For all this system, if the distribution of tasks should not be randomly then improvement can be high. Random distribution of tasks has been done in existing works [26]. Also, optimize the task depends on a single objective, which is sometimes a conflict like time and cost. Furthermore, only workflow dependency is the main aspect on which deadlines are dependent. Also, in existing approaches, optimization use local VM or global Datacenter.

In the proposed ideology, 3 algorithms are used:-Use of Tabu Search, Use of Bayesian Optimization, and Use of Whale Optimization.

Then all 3 steps have been combined to make it TBW Scheduling Approach

- **Tabu Search**
    - Input to tabu search method is: ranked tasks mapped to VMs
    - Apply tabu search algorithm to find neighbors of current VM.
    - It works iteratively.
    - If a neighbor is less utilized than the current VM then add such neighbor in the tabu list.
    - The result after applying tabu search:
        - A final list of VMs which are not efficiently utilized.
    - After this, the research has targeted tasks of these VMs and migrated them on other VMs but without making Queue.
- **Use of Bayesian Optimization**
    - Works for choosing the best combination for mapping tasks of not properly utilized VMs on utilized VMs
    - Targets tasks of the VMs which are not efficiently utilized and Targets those VMs which are underutilized category
    - Provides all combinations of VMs where tasks can be shifted.
- **Whale optimization**
    - Takes input from Bayesian Optimization.
    - Based on its objective functions, chooses the best combination and shifts tasks on VMs efficiently without increasing TEC and TET.
    - Overall results: Better results than GA-PSO for scientific workflows

5.3.2 Process of TBW Algorithm

Figure 5.1 elaborates all steps of TBW algorithm. Tabu Search is an important Meta heuristic optimization algorithm. It is an effective local search approach. It is denoted as TS algorithm also. It is considered as a performance enhancement and optimization method as it uses a guided local search procedure and so rejects moves to those resources which are already visited.

It works iteratively and each next iteration results in a better solution. It is useful in several applications like scheduling, energy distribution, pattern classification, telecommunications, resource planning, waste management, biomedical analysis, and more. It provides enhanced performance by a better local search approach. Tabu optimization algorithm works on tasks already mapped to VMs. Then it targets next to the next neighbors of each VM. It works fast and is dynamic. Apart from this, it doesn't repeat already covered VMs. If a neighbor is less utilized than the current VM then adds such neighbor in the tabu list. The result after applying tabu search is an array named TL. TL is a tabu list which is the final list of VMs which are not efficiently utilized.

Now the next stage of the proposed algorithm is task migration from those VMs shared by the tabu optimization algorithm to other VMs but without increasing cost and time parameters. So, Bayesian Optimization is used here. It provides all combinations where such underutilized VMs tasks can be migrated without crossing deadlines and other constraints. It targets tasks of the VMs which are not efficiently utilized and Targets those VMs which are underutilized category. The stage ahead in this TBW approach is the whale optimization algorithm. Whale Optimizer is a smart intelligent scheduling algorithm that perfectly calculates the fitness value of the system. It works dynamically and reduces randomness. It mimics the hunting nature of humpback whales which are intelligent kinds of fish. The whales have cells as intelligent as can judge, feel and show emotions. They behave like human beings. They are mostly observed as living in groups. To perform optimization, these whales work as below:

Search for prey (Also called exploration phase):In this phase, the whales search for the prey randomly. These whales are smart to find out the location of the prey, so they encircle them. It assumes the targeted prey as the current best solution.

The bubble-net feeding hunting strategy is the basic inspiration of the whale optimization algorithm as shown in figure 5.2. This is a unique behaviour and it is examined only in humpback whales. It works as the best search agent to chase the prey. Also, it uses the concept of spiral attack for bubble-net attack.

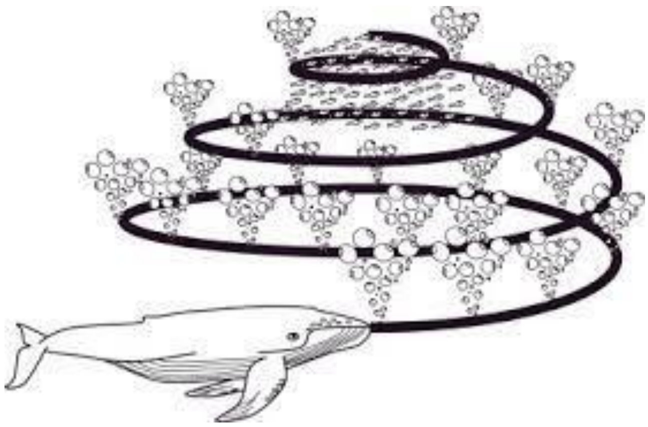


Figure 5.2: Humpback whales Bubble-net feeding behavior [185]

It has its objective functions. Both local and global optimization is possible in this TBW approach. The whale is a highly smart algorithm and it copies simple ideas from nature. It takes input from Bayesian Optimization. Then it works on its objective functions and chooses the best combination and shifts tasks on VMs efficiently. In the whole process, TEC and TET are calculated.

Input: Scientific Workflow and VMs= V1 to V20

Output: Analysis of scheduling and performance parameters.

The following equations 5.7 to 5.11 have been used to calculate TET and TEC in the whole process.

$$T_t = \sum T_R + \sum T_P + \sum T_W \qquad (5.7)$$

Total time for executing a workflow on cloud VM is calculated as addition of processing time, waiting time and its time until it is received completely. This calculation is shown in Eqn. 5.7.

Equation 5.8 expresses the actual cost of complete process as below:

Actual Cost = Total cost of under deadline processing +Deadline crossed Task Cost (5.8)

Here total cost itself is addition of movement factor (MF) and cost factor (CF) which can be directly calculated using equations 5.9 and 5.10 as below:

MF = Number of Migrations /used VMs (5.9)

CF = (Process Cost * task memory)/ involved VMs (5.10)

The Pseudo Code of Proposed TBW Optimization Algorithm has been provided as below:

Algorithm 4: TBW optimization algorithm

1. ***BEGIN***
2. *Take input as ranked scheduled Tasks on VMs*
3. ***VM_s= TABU (task, rank)*** *// call algorithm 5*
4. *T_c<- Task all combination*
5. ***X_{final}= Bayesopt(VM_s, T_c)*** *// call algorithm 6*
6. *For each $X_{final}(i)$*
7. ***Whale(i)<-X_{final}(i)*** *//whale optimization used*
8. *Update whale parameters a, A, C and L*
9. *Calculate distance Between each whale(i)*
10. *D= C. X_{final}–X_{rand}*
11. *If (A<1)*
12. *Update current whale position*
13. *X(t+1)=1/data centre {1/VM (Memory cost + processor cost + time delay)}*
14. *Else*
15. *X(t)<-threshold*
16. *End if*
17. *End for*
18. *Th<-threshold*
19. *According to Th, migrate task to VM*
20. *Analyze the parameters total execution cost (TEC) ,total execution time(TET), response time (RT) and energy consumption (EC).*
21. *END*

Algorithm 5: TABU (task, rank)

//Input to TABU: ranked tasks already Mapped on VMs

1. BEGIN
2. Set T=task with rank and Iteration
3. = N
4. m = 0
5. for it = 0 to N do
6. m = m+1
7. Execute searching neighbor (T, T_m) //Tasks are ranked
8. Execute searching VM based on tasks mapped on it
9. $if(it = N)$ then

   $VM_s \leftarrow Tabulist(optimize\ task\ sequence)$

10. else

    it =it+1 // next iteration and run whole process

11. Return VM_s
12. End if
13. End for
14. END

Algorithm 6: Bayesopt (VM_s,T_c)

1. *Begin*
2. *Initial combination* X_0<- T_c
3. ***Bound of length*** *Øε {*$\mathbf{X_L,X_u}$*}*
4. *Ø*$_p$ *ε u{*$\mathbf{X_L,X_u,P}$*}*
5. ***Work flow size:K***
6. ***For t=1 to*** *task*$_i$
   a. $X_{1:n}$*=argmax* $\alpha_t(X|P_r(X_{L-1},Ø_{1:n})$
   b. X_{final}*=Evaluate(*X_{L-1}, $X_{1:n}$*)*
7. *End for*
8. ***Return*** $\mathbf{X_{final}}$
9. *END*

Algorithm 4 named TBW optimization algorithm is used to start the process of TBW optimization on cloud environment. It works on already mapped tasks which are successfully ranked and then scheduled on different VMs in cloud system. Here to find out less utilized VMs, tabu search algorithm (algorithm 5) is called. It works in N iterations. N is total VMs in the system. Here T is input task of the workflow. It works in forward direction and provides optimize tabu list which is collection of less utilized VMs in the system. Variable VM_s provides the less utilize VMs list which has been used as input by Bayesian optimization algorithm in TBW optimization algorithm.

Step 5 of the algorithm 4 is used to call algorithm 6 which is Bayesian optimization process execution. It targets VMs which are underutilized (list provided by tabu search algorithm 5) and uses T_c*for* all tasks combinations. It works on selection of best combination of tasks among all tasks list provided by tabu search. X_0 *is used for input tasks combinations list.* $\boldsymbol{X_L}$ represents lower bounds and $\boldsymbol{X_U}$ represents upper bound limit. K represents size of input workflow. Variable X_{final}provides list of best combination of underutilized VMs tasks which can be migrated on other VMs.

Step 7 of algorithm 4 used whale optimization concept. Here, a, A, C and L are whale parameters used as coefficient vectors during tasks migration from underutilized VMs to other VMs but without making any queue. D is used to calculate distance between each whale during its execution. T is current iteration of whale process. X(t) is position vector of best solution obtained during the process. After statement 11, the search agent will move towards best search agent. Also, the complete process has analyzed the scheduling based on time, cost, response time, and energy consumption parameters.

5.4 Experimental Results and Analysis

The validation is performed via simulation-based experiments. The efficiency of TBW has been analyzed with the simulation-based results of scheduling different workflow tasks on various VMS using TBW algorithm. The whole agenda of the proposed study and its implementation is to reduce the total cost and time consumed in the whole process of mappings input tasks to VMs. Pegasus site has supported providing information about input scientific workflows. Using cloud simulator, TBW (Tabu-Bayesian-Whale) scheduling algorithm has provided us better results as compared to existing scheduling and optimization algorithms. As the proposed algorithm is named as TBW algorithm and a Comparison of TET and TEC parameters of TBW Algorithm with state-of-the-art existing algorithms has been performed.

Table 5.1 -5.5 summarizes the comparison of TBW with GA-PSO, PSO, GA, and WOA. The comparison is based on TET and TEC parameters. Table 5.1 as below provides the comparison of proposed approach named TBW algorithm with existing scheduling approaches PSO-GA, PSO, GA and Whale algorithm. Result provides better outcome of TBW approach as compare to existing approaches.

Table 5.1: Comparison of TET and TEC parameters of Tabu Bayesian Whale (TBW) Algorithm with existing algorithms (for LIGO workflow)

LIGO										
Number of VM	**PSO-GA**		**Tabu-Bayesian-Whale (TBW)**		**PSO**		**GA**		**WHALE**	
	TET (PSO-GA)	**TEC (PSO-GA)**	**TET(TBW)**	**TEC(TBW)**	**TET (PSO)**	**TEC (PSO)**	**TET(GA)**	**TEC(GA)**	**TET(WHALE)**	**TEC(WHALE)**
2	24.6	13.0	11.2	4.1	36.3	24.6	40.6	37.7	26.1	40.5
4	29.1	17.1	12.0	6.5	39.3	30.5	43.3	57.2	40.5	43.2
6	31.3	19.6	12.5	7.1	41.6	53.6	46.1	84.0	66.8	46.4
8	33.5	30.8	13.0	11.1	44.6	83.1	49.6	107.7	95.4	50.1
10	36.1	86.3	15.0	30.0	47.7	110.7	53.5	124.4	116.2	54.5
12	40.2	108.1	16.7	42.3	52.0	124.7	58.4	135.0	127.4	59.0
14	42.7	113.8	17.0	44.3	56.4	133.1	62.7	142.0	136.3	63.2
16	49.1	128.1	17.5	50.0	62.3	142.5	66.7	147.5	142.6	65.9
18	53.4	133.6	20.1	52.3	64.9	145.7	68.2	149.3	146.3	67.5
20	60.4	141.9	23.4	55.0	68.4	149.9	69.9	151.4	148.4	68.4

Here we have worked upon five different types of scientific workflows MONTAGE, CYBERSHAKE, LIGO, GENOME, and SIPHT. Figures 5.3-5.13 show the performance of the TBW algorithm with existing optimization algorithms in terms of time and cost consumption.

As per the data in table 5.1 we observe that the values of TET and TEC parameters for TBW approach are minimize for LIGO scientific workflow. Along the x-axis, the total VMs used parameter has been taken. Along the y-axis, in TEC graphs, cost (in rupees), and TET graphs, time (in ms) is taken. As shown in Table 5.1, the simulation-based outcome has been represented for LIGO workflow where TET and TEC based comparison of TBW method is done with existing methods.

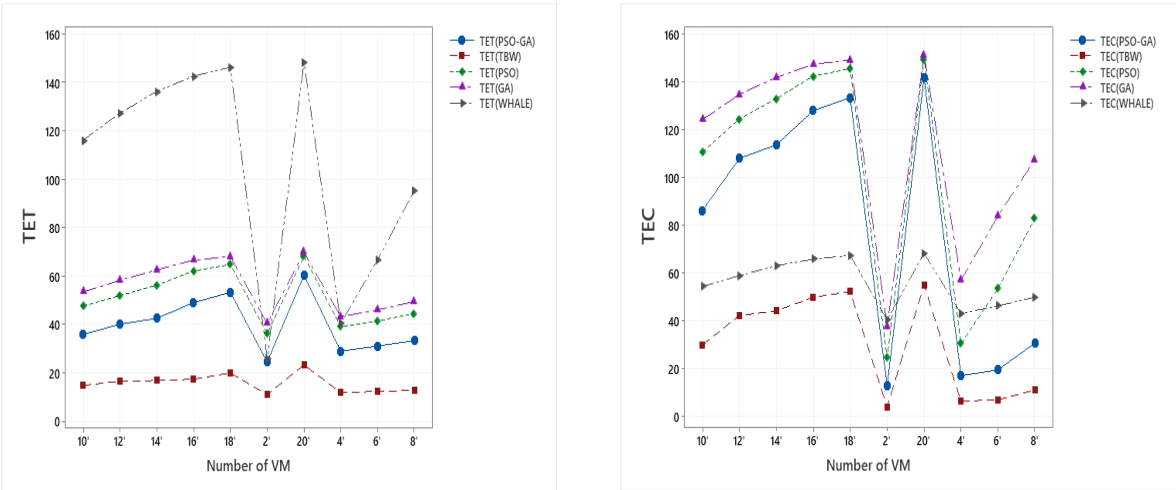


Figure 5.3: Simulation results of TET and TEC parameters of scheduling LIGO workflow for different optimization algorithms

In figure 5.3, TET refers to total execution time, TEC refers to total execution cost, PSO-GA is particle swarm optimization- genetic algorithm hybrid optimization algorithm, TBW is tabu Bayesian whale optimization algorithm, PSO is particle swarm optimization, GA is a genetic algorithm, and last but not the least is whale optimization.The X-axis of both left and right graphs of the figure refers to total VMs utilized in the system. A range of 2-20 VMS has been used. Here, LIGO workflow has been taken as input, and during optimization; TBW has given better results than others in the case of TET and TEC.

Table 5.2 as below provides the comparison of proposed approach named TBW algorithm with existing scheduling approaches PSO-GA, PSO, GA and Whale algorithm. Result provides better outcome of TBW approach as compare to existing approaches. The workflow named CYBERSHAKE has been selected as input workflow for it.

Table 5.2: Comparison of TET and TEC parameters of Tabu Bayesian Whale (TBW) Algorithm with existing algorithms (for CYBERSHAKE workflow)

CYBERSHAKE										
Number of VM	**PSO-GA**		**Tabu-Bayesian-Whale (TBW)**		**PSO**		**GA**		**WHALE**	
	TET(PSO-GA)	**TEC(PSO-GA)**	**TET(TBW)**	**TEC(TBW)**	**TET(PSO)**	**TEC(PSO)**	**TET(GA)**	**TEC(GA)**	**TET(WHALE)**	**TEC(WHALE)**
2	39.8	15.4	5.8	11.8	51.7	35.2	57.1	58.3	43.3	57.7
4	44.0	21.8	8.1	12.4	55.6	54.5	61.4	81.2	66.1	62.3
6	47.3	44.4	15.9	13.4	59.5	80.7	66.2	103.1	90.8	67.6
8	51.5	73.3	27.7	14.8	64.5	103.9	72.0	119.8	110.6	73.6
10	55.9	100.4	38.8	16.2	70.0	120.3	78.2	131.0	123.9	79.8
12	62.0	114.0	45.4	17.0	76.9	130.6	84.4	138.5	132.6	85.2
14	68.2	122.3	48.8	18.1	83.0	137.5	89.0	143.2	138.8	89.2
16	76.6	131.5	52.3	20.2	88.7	143.0	92.4	146.3	142.3	91.4
18	80.3	134.7	53.6	21.7	90.7	144.7	93.5	147.2	144.2	92.6
20	85.2	138.8	54.9	23.3	93.2	146.8	94.7	148.3	145.3	93.2

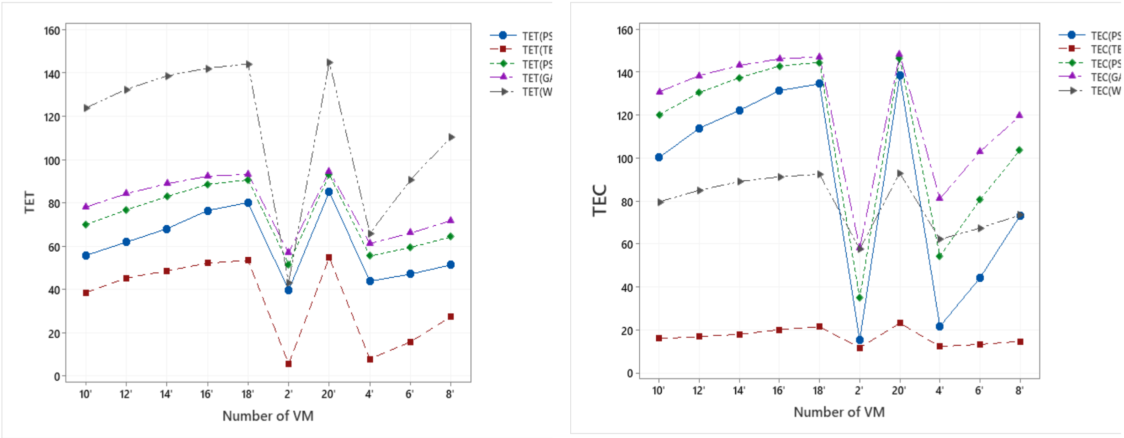


Figure 5.4: Simulation results of TET and TEC parameters of scheduling CYBERSHAKE workflow for different optimization algorithms

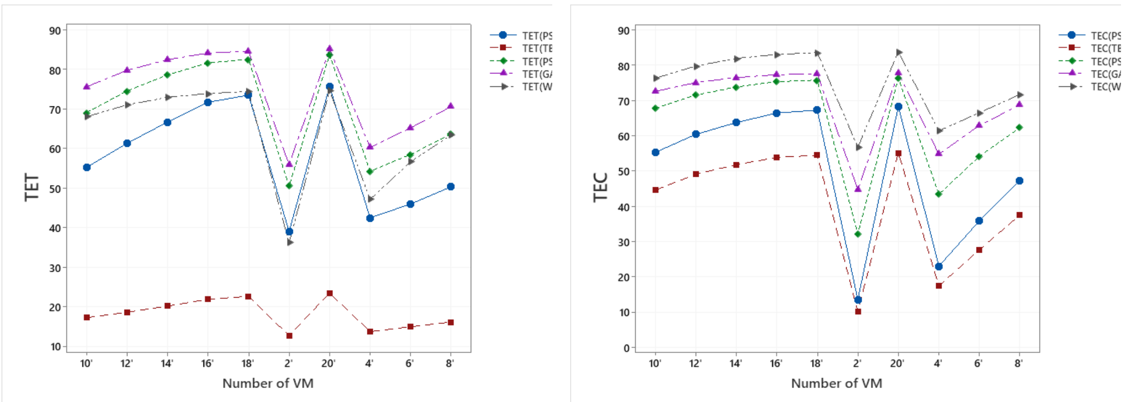


Figure 5.5: Simulation results of TET and TEC parameters of scheduling GENOME workflow for different optimization algorithms

Same as the case of LIGO, CYBERSHAKE workflow has been accepted as input workflow and TBW has given better results in the case of managing cost and time. Figure 5.4 shows the results based on total execution Time (TET) and total execution cost (TEC). Again results are better in the case of TBW algorithm. The X-axis of both left and right graphs of the figure refers to total VMs utilized in the system. A range of 2-20 VMs has been used.

Table 5.3 provides the comparison of proposed approach named TBW algorithm with existing scheduling approaches PSO-GA, PSO, GA and Whale algorithm. Result provides better outcome of TBW approach as compare to existing approaches. The workflow named GENOME has been selected as input workflow for it.

Table 5.3: Comparison of TET and TEC parameters of Tabu Bayesian Whale (TBW) Algorithm with existing algorithms (for GENOME workflow)

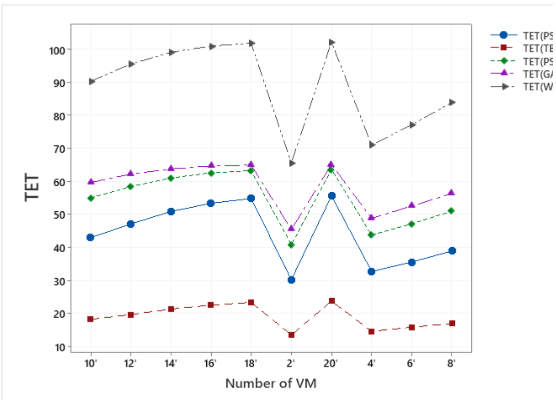
GENOME										
Number of VM	**PSO-GA**		**Tabu-Bayesian-Whale (TBW)**		**PSO**		**GA**		**WHALE**	
	TET(PSO-GA)	**TEC(PSO-GA)**	**TET(TBW)**	**TEC(TBW)**	**TET(PSO)**	**TEC(PSO)**	**TET(GA)**	**TEC(GA)**	**TET(WHALE)**	**TEC(WHALE)**
2	39.1	13.6	12.8	10.2	50.6	32.2	56.0	44.7	36.3	56.7
4	42.6	23.0	13.8	17.5	54.3	43.4	60.4	54.8	47.3	61.4
6	46.1	35.9	15.0	27.7	58.6	54.1	65.3	62.9	56.7	66.5
8	50.4	47.3	16.2	37.5	63.7	62.3	70.7	68.7	63.5	71.7
10	55.3	55.3	17.3	44.6	69.2	67.8	75.7	72.5	68.1	76.3
12	61.4	60.4	18.7	49.1	74.7	71.5	79.9	75.0	71.1	79.7
14	66.8	63.7	20.2	51.8	78.7	73.8	82.5	76.4	73.0	81.9
16	71.8	66.4	22.0	53.8	81.7	75.3	84.2	77.3	74.0	83.0
18	73.6	67.2	22.7	54.5	82.7	75.7	84.7	77.5	74.5	83.5
20	75.8	68.2	23.6	55.1	83.8	76.2	85.3	77.7	74.7	83.8

Same like above cases, tasks of GENOME workflow have been accepted as input and TBW has given better results in the case of managing cost and time as shown in figure 5.5. The X-axis of both left and right graphs of the figure refers to total VMs utilized in the system. A range of 2-20 VMS has been used.

Table 5.4 as below provides the comparison of proposed approach named TBW algorithm with existing scheduling approaches PSO-GA, PSO, GA and Whale algorithm. Result provides better outcome of TBW approach as compare to existing approaches. The workflow named SIPHT has been selected as input workflow for it.

Table 5.4: Comparison of TET and TEC parameters of Tabu Bayesian Whale (TBW) Algorithm with existing algorithms (for SIPHT workflow)

SIPHT										
Number of VM	**PSO-GA**		**Tabu-Bayesian-Whale (TBW)**		**PSO**		**GA**		**WHALE**	
	TET(PSO-GA)	**TEC(PSO-GA)**	**TET(TBW)**	**TEC(TBW)**	**TET(PSO)**	**TEC(PSO)**	**TET(GA)**	**TEC(GA)**	**TET(WHALE)**	**TEC(WHALE)**
2	30.3	52.1	13.6	15.9	41.0	64.6	45.5	71.4	65.6	45.7
4	32.9	56.5	14.7	25.8	43.9	69.6	48.9	77.3	71.1	49.3
6	35.7	61.4	16.0	37.0	47.3	75.5	52.7	83.7	77.3	53.0
8	39.1	67.1	17.1	46.5	51.1	82.1	56.4	90.2	84.0	56.6
10	43.1	74.0	18.3	53.0	55.1	88.9	59.8	95.9	90.4	59.5
12	47.2	81.2	19.8	57.0	58.6	94.9	62.3	100.3	95.6	61.6
14	51.0	87.7	21.4	59.7	61.1	99.3	63.9	103.0	99.2	62.8
16	53.5	91.9	22.7	61.2	62.7	102.0	64.8	104.5	101.1	63.4
18	54.9	94.4	23.5	62.0	63.3	103.1	65.0	104.9	101.9	63.6
20	55.7	95.8	23.9	62.3	63.7	103.8	65.2	105.3	102.3	63.7



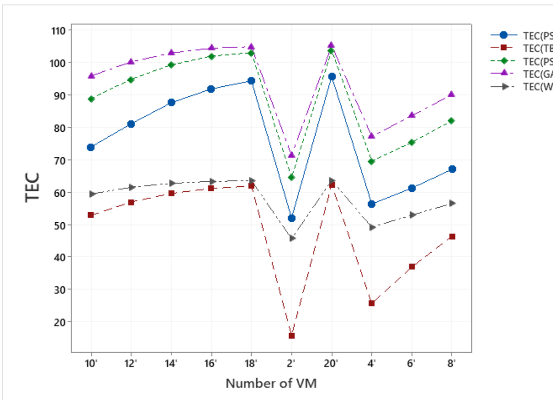


Figure 5.6: Simulation results of TET and TEC parameters of scheduling SIPHT workflow for the different optimization algorithm.

The above results in figure 5.6 are representing comparison of TBW algorithm with GA-PSO, Whale, GA, and PSO algorithms for SIPHT workflow. TBW has provided better results in terms of both execution time and cost consumption parameters. The X-axis of both left and right graphs of the figure refers to total VMs utilized in the system. A range of 2-20 VMS has been used.

Table 5.5 as below provides the comparison of proposed approach named TBW algorithm with existing scheduling approaches PSO-GA, PSO, GA and Whale algorithm. Result provides better outcome of TBW approach as compare to existing approaches. The workflow named MONTAGE has been selected as input workflow for it.

Table 5.5: Comparison of TET and TEC parameters of Tabu Bayesian Whale (TBW) Algorithm with existing algorithms (for MONTAGE workflow)

MONTAGE										
Number of VM	**PSO-GA**		**Tabu-Bayesian-Whale (TBW)**		**PSO**		**GA**		**WHALE**	
	TET(PSO-GA)	**TEC(PSO-GA)**	**TET(TBW)**	**TEC(TBW)**	**TET(PSO)**	**TEC(PSO)**	**TET(GA)**	**TEC(GA)**	**TET(WHALE)**	**TEC(WHALE)**
2	27.6	50.3	10.2	16.8	38.4	63.1	43.2	70.3	64.4	43.5
4	30.3	54.9	11.4	27.4	41.6	68.6	46.8	76.5	70.2	47.2
6	33.4	60.2	12.5	37.8	45.2	74.8	50.5	82.9	76.7	50.8
8	37.1	66.6	13.7	45.7	49.1	81.5	54.1	89.0	83.2	54.1
10	41.1	73.6	15.1	51.0	52.8	87.9	57.0	94.1	89.0	56.6
12	45.1	80.4	16.6	54.4	55.9	93.1	59.1	97.7	93.4	58.2
14	48.2	85.8	18.1	56.5	57.9	96.7	60.3	99.8	96.2	59.1
16	50.2	89.2	19.1	57.6	59.1	98.7	60.9	100.8	97.6	59.5
18	51.3	91.1	19.7	58.1	59.5	99.4	61.1	101.1	98.1	59.7
20	51.7	91.8	19.9	58.3	59.7	99.8	61.2	101.3	98.3	59.7

Since the main focus of our research is on four major parameters which are reduction in TET as well as TEC and minimizing the response time as well as energy consumption while mapping tasks on resources.

In figure 5.7, Montage workflow has been selected as input workflow and again results showing TBW is better than other algorithms.

Apart from this, if we consider the other parameters like response time (RT) and energy consumption (EC) then it can be analyzed from the following graphical representations that our proposed approach TBW has successfully provided an improvement over previous algorithms.

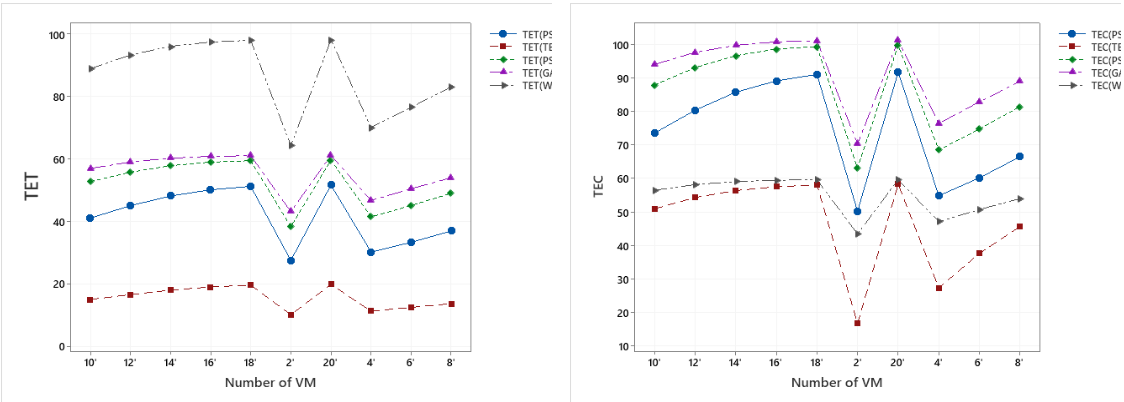


Figure 5.7: Simulation results of TET and TEC parameters of scheduling MONTAGE workflow for different optimization algorithms

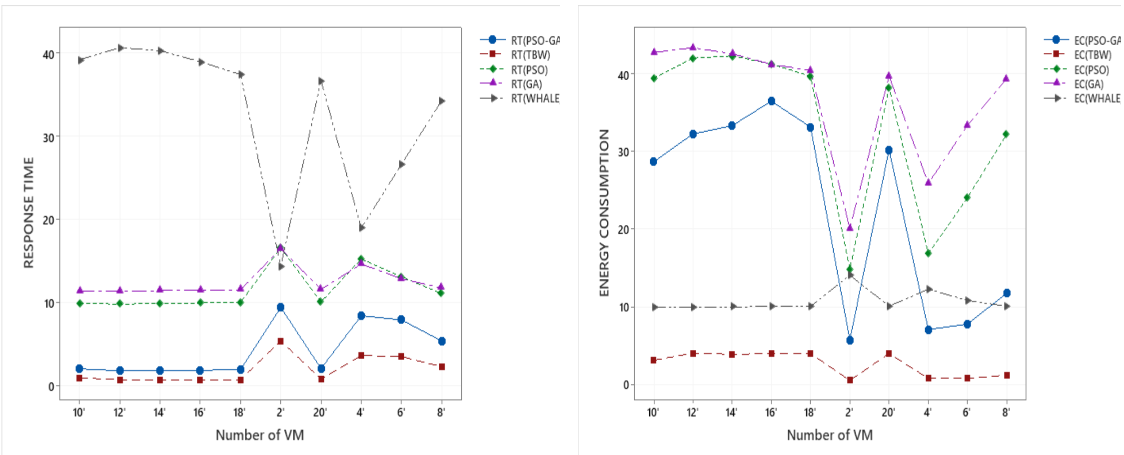


Figure 5.8: Simulation results of response time and energy consumption parameters of scheduling LIGO workflow for different optimization algorithms

Table 5.6: Comparison of Response Time and Energy Consumption of Tabu Bayesian Whale (TBW) Algorithm with existing algorithm (for LIGO workflow)

LIGO										
Number of VM	**PSO-GA**		**Tabu-Bayesian-Whale (TBW)**		**PSO**		**GA**		**WHALE**	
	RT(PSO-GA)	EC(PSO-GA)	RT(TBW)	EC(TBW)	RT(PSO)	EC(PSO)	RT(GA)	EC(GA)	RT(WHALE)	EC(WHALE)
2	9.5	5.8	5.4	0.6	16.6	14.9	16.5	20.1	14.4	14.1
4	8.5	7.1	3.7	0.8	15.3	16.9	14.7	25.9	19.0	12.3
6	8.0	7.8	3.6	0.8	13.2	24.1	12.9	33.4	26.7	10.9
8	5.4	11.8	2.3	1.2	11.1	32.3	11.8	39.4	34.4	10.1
10	2.1	28.7	1.0	3.1	9.9	39.5	11.4	42.8	39.3	9.9
12	1.9	32.3	0.8	4.1	9.9	42.1	11.4	43.4	40.7	10.0
14	1.9	33.4	0.8	3.9	9.9	42.4	11.5	42.6	40.3	10.0
16	1.9	36.5	0.7	4.0	10.0	41.3	11.6	41.3	39.0	10.1
18	2.0	33.2	0.8	4.0	10.1	39.7	11.6	40.5	37.5	10.1
20	2.1	30.3	0.9	4.0	10.1	38.3	11.6	39.8	36.8	10.1

Table 5.6 as shown above is showing the improvement in RT (response time) and EC (energy consumption) of the proposed method TBW which provided better results as compared to other existing techniques. Simulation results based on response time (RT) and energy consumption (EC) parameters have been represented in figures 5.8 to 5.12 for various scientific workflows.

The results in figure 5.8 are for LIGO workflow and a comparison of TBW algorithm has been done with GA-PSO, Whale, GA, and PSO algorithms. TBW has provided better results in terms of both parameters. The X-axis of both left and right graphs of the figure refers to total VMs utilized in the system. A range of 2-20 VMS has been used.Table 5.7 as below provides the response time and energy consumption based comparison of proposed approach named TBW algorithm with existing scheduling approaches PSO-GA, PSO, GA and Whale algorithm. Result provides better outcome of TBW approach as compare to existing approaches. SIPHT workflow has been selected as input workflow for it.

Table 5.7: Comparison of Response Time and Energy Consumption of Tabu Bayesian Whale (TBW) Algorithm with existing algorithm (for SIPHT workflow)

SIPHT Workflow										
Number of VM	**PSO-GA**		**Tabu-Bayesian-Whale (TBW)**		**PSO**		**GA**		**WHALE**	
	RT(PSO-GA)	**EC(PSO-GA)**	**RT(TBW)**	**EC(TBW)**	**RT(PSO)**	**EC(PSO)**	**RT(GA)**	**EC(GA)**	**RT(WHALE)**	**EC(WHALE)**
2	2.9	19.1	1.7	0.8	10.9	27.2	12.4	28.9	25.7	10.9
4	2.9	19.1	1.1	1.2	10.9	27.3	12.4	29.2	26.0	10.9
6	2.9	19.2	0.9	1.6	10.9	27.7	12.4	29.5	26.4	10.9
8	2.9	19.6	0.7	1.8	10.9	28.1	12.4	29.8	26.7	10.9
10	2.9	20.2	0.7	1.9	10.9	28.4	12.4	29.8	26.9	10.9
12	2.9	20.5	0.7	1.9	10.9	28.4	12.4	29.8	26.8	10.9
14	2.9	20.4	0.7	2.0	10.9	28.3	12.4	29.7	26.7	10.9
16	2.9	20.3	0.7	2.0	10.9	28.1	12.4	29.6	26.6	10.9
18	2.9	20.1	0.8	2.0	10.9	28.1	12.4	29.6	26.6	10.9
20	2.9	20.0	0.8	2.0	10.9	28.0	12.4	29.5	26.5	10.9

In figure 5.9, SIPHT scientific workflow has been taken and a comparison of TBW algorithm has been done with existing state of the art approaches named GA-PSO, Whale, GA, and PSO algorithms. TBW has provided better results in terms of both Response Time and Energy Consumption parameters. The X-axis of both left and right graphs of the figure refers to total VMs utilized in the system. A range of 2-20 VMS has been used.

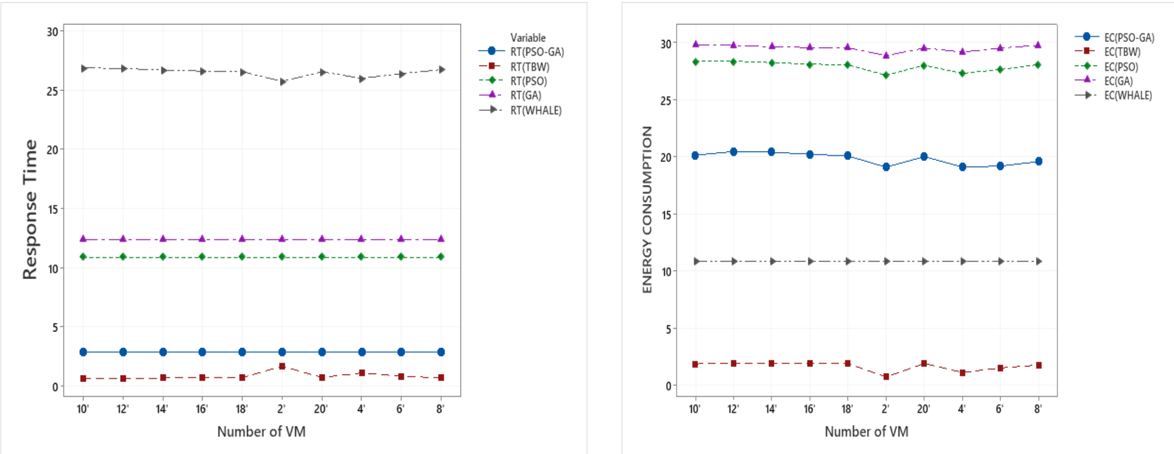


Figure 5.9: Simulation results of response time and energy consumption parameters of scheduling SIPHT workflow for different optimization algorithms

In table 5.8 as below, the response time and energy consumption based comparison of proposed approach named TBW algorithm with existing scheduling approaches PSO-GA, PSO, GA and Whale algorithm has been done. Result provides better outcome of TBW approach as compare to existing approaches. GENOME workflow has been selected as input workflow for it.

Table 5.8: Comparison of Response Time and Energy Consumption of Tabu Bayesian Whale (TBW) Algorithm with existing algorithm (for GENOME workflow)

GENOME Workflow										
Number of VM	**PSO-GA**		**Tabu-Bayesian-Whale (TBW)**		**PSO**		**GA**		**WHALE**	
	RT(PSO-GA)	**EC(PSO-GA)**	**RT(TBW)**	**EC(TBW)**	**RT(PSO)**	**EC(PSO)**	**RT(GA)**	**EC(GA)**	**RT(WHALE)**	**EC(WHALE)**
2	14.4	5.3	2.5	0.4	18.0	16.5	17.0	20.9	16.6	14.7
4	9.2	8.4	1.6	0.6	15.0	19.6	15.4	23.3	19.4	13.6
6	6.4	11.9	1.1	0.9	13.6	22.2	14.8	24.7	21.4	13.2
8	5.3	14.6	0.9	1.2	13.1	23.6	14.7	25.2	22.3	13.2
10	5.0	16.0	0.8	1.3	13.1	23.9	14.7	25.1	22.3	13.3
12	5.1	16.2	0.8	1.3	13.2	23.7	14.9	24.7	21.9	13.4
14	5.2	15.7	0.8	1.3	13.4	23.2	15.0	24.4	21.5	13.5
16	5.4	15.1	0.8	1.3	13.5	22.8	15.0	24.1	21.2	13.5
18	5.5	14.8	0.8	1.3	13.5	22.6	15.0	24.1	21.1	13.5
20	5.6	14.5	0.9	1.3	13.6	22.5	15.1	24.0	21.0	13.6

In figure 5.10, GENOME scientific workflow has been taken and a comparison of TBW algorithm has been done with existing state of the art approaches named GA-PSO, Whale, GA, and PSO algorithms. TBW has provided better results in terms of both Response Time and Energy Consumption parameters. The X-axis of both left and right graphs of the figure refers to total VMs utilized in the system. A range of 2-20 VMS has been used.

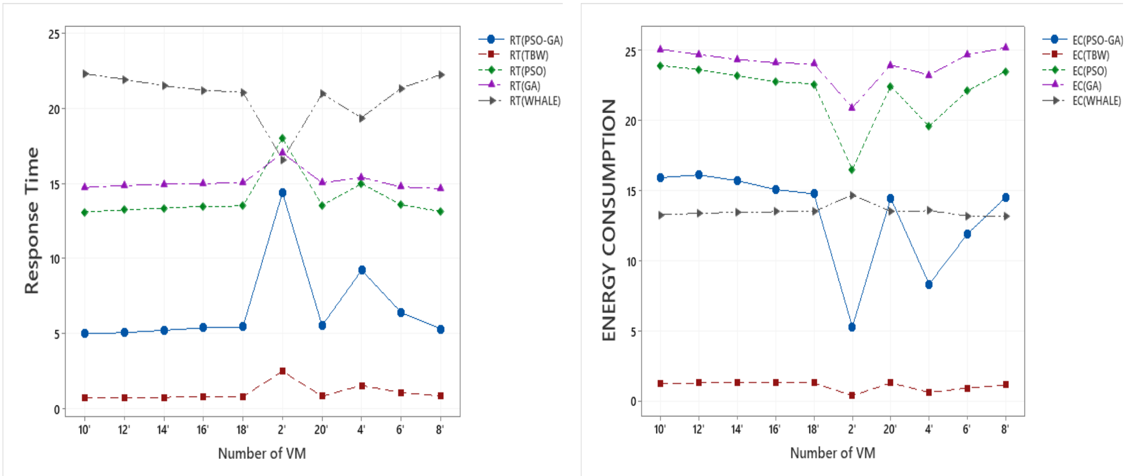


Figure 5.10: Simulation results of response time and energy consumption parameters of scheduling GENOME workflow for different optimization algorithms

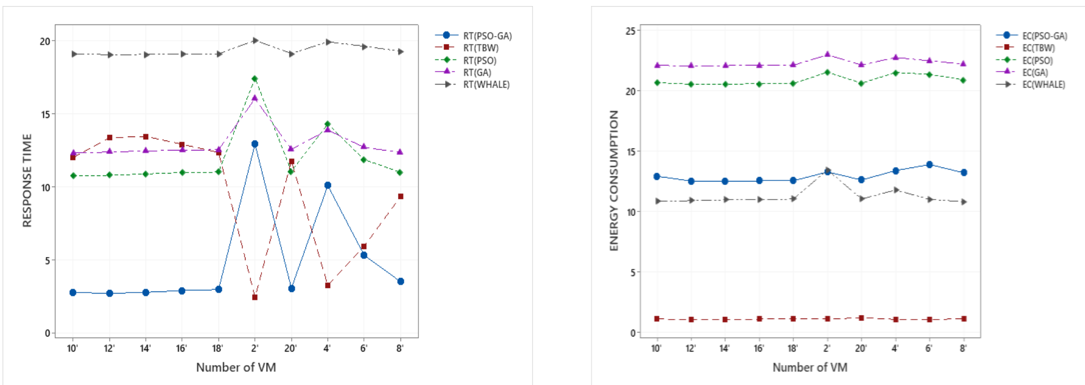


Figure 5.11: Simulation results of response time and energy consumption parameters of scheduling CYBERSHAKE workflow for different optimization algorithms

In table 5.9 as below, the response time and energy consumption based comparison of proposed approach named TBW algorithm with existing scheduling approaches PSO-GA, PSO, GA and Whale algorithm has been done. Result provides better outcome of TBW approach as compare to existing approaches. CYBERSHAKE workflow has been selected as input workflow for it.

Table 5.9: Comparison of Response Time and Energy Consumption of Tabu Bayesian Whale (TBW) Algorithm with existing algorithm (for CYBERSHAKE workflow)

CYBERSHAKE Workflow										
Number of VM	**PSO-GA**		**Tabu-Bayesian-Whale (TBW)**		**PSO**		**GA**		**WHALE**	
	RT(PSO-GA)	**EC(PSO-GA)**	**RT(TBW)**	**EC(TBW)**	**RT(PSO)**	**EC(PSO)**	**RT(GA)**	**EC(GA)**	**RT(WHALE)**	**EC(WHALE)**
2	12.9	13.3	2.4	1.1	17.5	21.5	16.0	23.0	20.0	13.5
4	10.1	13.4	3.3	1.1	14.3	21.5	13.9	22.8	19.9	11.8
6	5.3	13.9	5.9	1.1	11.9	21.4	12.7	22.5	19.6	11.0
8	3.5	13.2	9.3	1.1	11.0	20.9	12.4	22.2	19.3	10.8
10	2.8	12.9	12.0	1.2	10.8	20.7	12.3	22.1	19.1	10.9
12	2.7	12.6	13.4	1.1	10.8	20.6	12.4	22.1	19.1	10.9
14	2.8	12.5	13.5	1.1	10.9	20.6	12.5	22.1	19.1	11.0
16	2.9	12.6	12.9	1.1	11.0	20.6	12.5	22.1	19.1	11.0
18	3.0	12.6	12.4	1.2	11.0	20.6	12.5	22.1	19.1	11.1
20	3.1	12.6	11.8	1.3	11.1	20.6	12.6	22.1	19.1	11.1

Figure 5.11 is using the CYBERSHAKE workflow. Results are again improved for TBW approach.X-axis of both the left and the right graphs of the figure refer to total VMs utilized in the system. A range of 2-20 VMS has been used during simulation.

Table 5.10: Comparison of Response Time and Energy Consumption of Tabu Bayesian Whale (TBW) Algorithm with existing algorithm (for MONTAGE workflow)

MONTAGE Workflow										
Number of VM	**PSO-GA**		**Tabu-Bayesian-Whale (TBW)**		**PSO**		**GA**		**WHALE**	
	RT(PSO-GA)	**EC(PSO-GA)**	**RT(TBW)**	**EC(TBW)**	**RT(PSO)**	**EC(PSO)**	**RT(GA)**	**EC(GA)**	**RT(WHALE)**	**EC(WHALE)**
2	2.7	24.7	1.8	1.3	10.8	32.3	12.3	33.7	30.7	10.8
4	2.8	24.2	1.2	2.0	10.8	32.2	12.3	33.7	30.7	10.8
6	2.8	24.0	1.0	2.5	10.8	32.2	12.3	33.7	30.8	10.8
8	2.8	24.3	0.9	2.8	10.8	32.3	12.3	33.6	30.7	10.8
10	2.8	24.4	0.9	2.9	10.8	32.1	12.3	33.3	30.4	10.8
12	2.8	24.2	0.9	2.9	10.8	31.8	12.3	33.0	30.1	10.8
14	2.8	23.7	1.0	2.9	10.8	31.4	12.3	32.7	29.8	10.8
16	2.8	23.4	1.0	2.9	10.8	31.2	12.3	32.6	29.6	10.8
18	2.8	23.1	1.0	2.9	10.8	31.1	12.3	32.6	29.6	10.8
20	2.8	23.1	1.0	2.9	10.8	31.1	12.3	32.6	29.6	10.8

Table 5.10 provides the response time and energy consumption based comparison of proposed approach named TBW algorithm has been done with state-of-the-art existing scheduling approaches named PSO-GA, PSO, GA and Whale algorithm. Result provides better outcome of TBW approach as compare to existing approaches. The workflow selected as input is MONTAGE workflow.

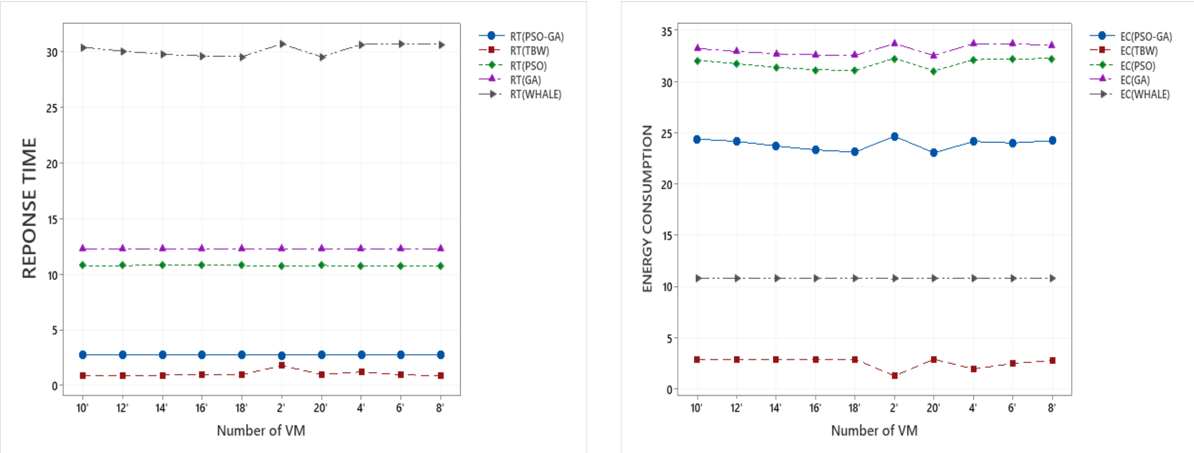


Figure 5.12: Simulation results of response time and energy consumption parameters of scheduling MONTAGE workflow for different optimization algorithms

Figure 5.12 is using MONTAGE workflow and providing better results again. The X-axis of both left and right graphs of the figure refers to total VMs utilized in the system. A range of 2-20 VMS has been used. It is observed that the energy consumption and response time show horizontal stability compared to an increase or decrease in VMs.

5.5 Summary

To manage the various challenges of scheduling in cloud computing environment, this chapter put across the workflow optimization algorithm based on Tabu Bayesian Whale optimization techniques. The objective is to formulate and propose the best fit schedule, which can efficiently facilitate the execution of workflow tasks in a minimum possible time, by utilizing the cloud computing resources to the maximum and also under the deadline constraints. Through the experiment outcome, it is confirmed that the proposed algorithm is the best fit which gives maximum real-time performance with further optimization abilities and thus proved to be an effective workflow scheduling algorithm in the cloud computing environment. The complex workflow scheduling challenge in cloud computing environments or infrastructure has shown the need for optimization.

We have incorporated TBW optimizer scheduler for optimizing the TET, TEC, Response time and Energy consumption parameter with GA-PSO, PSO, GA, and WOA methods integrated with scientific workflows MONTAGE, CYBERSHAKE, LIGO, GENOME, and SIPHT. The comparative data analytics shows that the TET and TEC parameters were most optimized with the WOA system compared to GA-PSO, PSO, and GA by almost 22% and 25% respectively. It was also observed that scientific workflows MONTAGE and GENOME give an optimized result compared to its counterpart LIGO, CYBERSHAKE, and SIPHT. The TET and TEC parameter increases exponentially as the number of VMs increases and the same trend can be seen in the response time and energy consumption. We were also able to define the optimal stability for the cloud system when were the TET, TEC, response time and energy consumption works with efficiency up to 95% for the range of 8 to 14 VMs. This analysis was calculated by analyzing different minima and maxima comparisons for TBW with GA-PSO, PSO, GA, and WOA.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

Cloud is one of the compassionate computing eras which offers services to make life more trustworthy, smart, and stress-free by a list of off-handed services which a person can utilize anytime anywhere with internet access. Yet, to make it best utilized from all aspects, it is required that analysis of whether all resources are perfectly supervised and utilized or not. If cloud consumption is imbalanced then such cloud use would be harmful to the environment also. It means, if under deadline constraints, a cloud is optimized and all resources are properly used then, it would increase the attraction of consumers and make it highly ecofriendly.

We have proposed a ranking algorithm for the scientific workflows that are highly effective for ranking tasks of input workflows. The proposed ranking algorithm is advance to the existing fuzzy HEFT algorithm. It minimized total execution time which is represented with TET in the above graphs and total execution cost represented as TEC in the above results-based graphs. For getting these results, mapping of tasks to VM resources has been done. The proposed algorithm named distributed HEFT has provided a better ranking system and so has minimized TEC and TET while scheduling tasks to VMs. In the future, we aim at providing enhanced scheduling algorithm to provide better results.

Cloud system has provided myriad features for better management and optimization. The proposed framework is an efficient framework for time, cost as well as energy saving. Apart from this, response time is also the main aspect of this study. By using a ranking method for input tasks, it is better to schedule tasks in an optimized matter rather than simply map the tasks on cloud VMs. Also, the use of tabu and whale optimization has enhanced the performance of our proposed research as underutilized resources are freed from the system within time and cost constraints. In our future study, we will implement the framework using advanced optimization techniques in a cloud environment so that better resource utilization with minimum time and cost can be provided. These all features provide users betterment so that optimized results would be fetched. This paper proposed an advanced whale optimization algorithm integrated with tabu search. In further work, results with TBW advance approach will be calculated and also will be compared with existing optimization approaches like GA, PSO, and WOA.The various algorithms which are presented in the above literature survey are either linked with the task selection phase which is even objective 1 of our proposed work or are related with the VM allocation phase which is objective 2 of our study. Also, the target of researchers is to minimize various parameters like time, cost, energy consumption, response time, etc.

If we consider the scheduling algorithms which are working for the task selection phase then the most common factors used are the earliest start time or the task finish time, task runtime, critical path tasks, and priorities of the tasks or selecting the tasks based on its ranking. Only providing task selection of input workflows is not sufficient for better cloud optimization. So, the resource selection phase is a succeeding and crucial phase of scheduling. The requirement of an optimal solution for scheduling tasks to the cloud resources for workflow execution is an important stage. For making resource selection most optimal, TET, TEC, energy consumption, response time, resource utilization, deadline constraints, budget are widely considered parameters. Migration is another procedure employed for energy reduction by context-switching and turning off the idle servers; also the resources that exhibit minimum power consumption for executing the workflow task are selected for the execution. Better resource optimization has become a challenge if need large-scale data center's management like cloud system. Also, scientific workflows need careful and managed scheduling in the cloud. So, an efficient optimization algorithm is required for achieving high-performance results. This study has simulated an advanced algorithm named TBW which is used for better optimization in cloud systems under deadline constraints. Also, time and cost parameters have been controlled during scheduling. In this research work, input workflow tasks are not randomly mapped to VMs but are firstly ranked using a distributed-HEFT method and then mapping has been done using TBW method. Our proposed algorithm is better than existing algorithms of optimization. Even comparisons based on TEC, TEC, response time have been done. The evaluated results have provided better results than GA, PSO, and hybrid approaches of optimization. The proposed system is more effective as it has used effective optimization in a better way.

6.2 Future Scope

As per the literature survey, still there is scope for further improvement in the present scheduling solutions for scientific workflow in cloud computing. In this field, the following aspects describe important directions shortly.

The proposed ranking approach can be implemented in a workflow engine and different methods can be used to perform the ranking of input tasks in a real environment.

In the future, the execution time of workflows can also be considered which can further increase the effectiveness of the strategy.

REFERENCES

[1] Dillon, Tharam., Elizabeth Chang, Chen Wu.: Cloud Computing: Issues and Challenges, 24th IEEE International Conference on Advanced Information Networking and Applications., 2010.

[2] Malawski, M., Juve, G., Deelman, E., Nabrzyski, J., 2015.Algorithm for cost-and deadline-constrained provisioning for scientific workflow ensembles in iaas clouds. In: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, IEEE Computer Society Press, pp. 1–18.

[3] J. D. Ullman, „NP-complete scheduling problems", 1975, In the Journal of Computer & System Sciences, vol. – 10, no.-3, pp 384-393.

[4] Hashizume, K., Rosado, D. G., Fernández-Medina, E., and Fernandez, E. B. 2013. An analysis of security issues for cloud computing. Journal of Internet and Applications, 4(1), 1-13.

[5] Pearson, S. 2013. Privacy, security and trust in cloud computing. In Privacy and Security for Cloud Computing (pp. 3-42). Springer London.

[6] Alkhashai, Hussin&Omara, Fatma.(2016). An Enhanced Task Scheduling Algorithm on Cloud Computing Environment. International Journal of Grid and Distributed Computing. 9. 91-100. 10.14257/ijgdc.2016.9.7.10

[7] Sharma, Priya&kaur, Kiranbir. (2018). Hybrid Artificial Bee Colony and Tabu Search Based Power Aware Scheduling for Cloud Computing. International Journal of Intelligent Systems and Applications. 10. 39-47. 10.5815/ijisa.2018.07.04.

[8] H. A. Chen, “On the design of task scheduling in the heterogeneous computing environments”, Proceedings of the 2005 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM. 2005) , pp. 396-399, 24-26 Aug. 2005.

[9] Al-Haboobi, Ali. (2017). Improving Max-Min scheduling Algorithm for Reducing the Makespan of Workflow Execution in the Cloud. International Journal of Computer Applications. 177. 975-8887. 10.5120/ijca2017915684.

[10] AlessioSignorini, “A Survey of Ranking Algorithms”, Department of Computer Science, University of Iowa, 2005.

[11] Ramandeep Sandhu and KamleshLakhwani (2021), Improved Scientific Workflow Scheduling Algorithm with distributed HEFT Ranking and TBW Scheduling Method, in International Conference on Wireless Sensor Networks, Ubiquitous Computing and Applications (ICWSNUCA 2021), ISSN- 2367-3370/ Vol. 244, issue no. 1.

[12] Andrei R. &Arjan J.C. van Gemund, “Fast and Effective Task Scheduling in Heterogeneous Systems”, IEEE Proceeding, 2000.

[13] Beasley, D., Bull, D.R. and Martin, R.R., “An Overview of Genetic Algorithms”, University Computing, 1993, Vol. 15, pp. 58-69, 170-181.

[14] Alkhanak, E., Lee, S., Rezaei, R., Parizi, R., 2018. Cost optimization approaches for scientific workflow scheduling in cloud and grid computing: A review, classifications, and open issues. Journal of Systems and Software, 113, pp. 1-26

[15] Bölöni, L., Damla, T., 2017. Value of information based scheduling of cloud computing resources. Future Generation Computer Systems ,71 , pp.212-220

[16] Deelman, E., Singh, G., Livny, M., Berriman, B., Good, J., (2008). The cost of doing science on the cloud: The Montage example. SC '08: Proceedings of the 2008 ACM/IEEE Conference on Supercomputing, Austin, TX, pp.1-12

[17] Y. Jadeja and K. Modi, "Cloud computing - concepts, architecture and challenges," 2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET), Kumaracoil, 2012, pp. 877-880.

[18] Vecchiola, Christian, Suraj Pandey, and RajkumarBuyya. "High-performance cloud computing: A view of scientific applications." Pervasive Systems, Algorithms, and Networks (ISPAN), 2009 10th International Symposium on. IEEE, 2009.

[19] Ghose, M., Verma, P., Karmakar, S., Sahu, A., (2017). Energy Efficient Scheduling of Scientific Workflows in Cloud Environment. 2017 IEEE 19th International Conference on High Performance Computing and Communications; IEEE 15th International Conference on Smart City; IEEE 3rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Bangkok, pp. 170-177.

[20] Goyal, M., Mehak, A., 2017. Optimize workflow scheduling using hybrid ant colony optimization (ACO) & particle swarm optimization (PSO) algorithm in cloud environment. Int. J. Adv. Res. Ideas Innov. Technol 3(2)

[21] Braun, T. D., Siegal, H. J., Beck, N., Boloni, L. L., Maheswaran, M., Reuther, A. I., Robertson, J. P., Theys, M. D., Bin Yao, Hensgen, D. and Freund, R. F. (1999), A comparison study of static mapping heuristics for a class of meta-tasks on heterogeneous computing systems, in ‘Proceedings. Eighth Heterogeneous Computing Workshop (HCW’99)’, pp. 15–29.

[22] Topcuoglu, H.|Hariri, S.|Wu, M.-Y.: Performance-Effective and Low Complexity Task Scheduling for Heterogeneous Computing. IEEE Transactions on Parallel and Distributed Systems, Vol. 13, 2002, No. 3, pp. 260{274, doi: 10.1109/71.993206.

[23] J.J. Durillo, R. Prodan, Multi-objective workflow scheduling in Amazon EC2, Clust. Comput.17 (2) (2014) 169–189.

[24] M. Adhikari and T. Amgoth, "Efficient algorithm for workflow scheduling in cloud computing environment," 2016 Ninth International

Conference on Contemporary Computing (IC3), 2016, pp. 1-6, doi: 10.1109/IC3.2016.7880222.

[25] Jiang, J., Yaping, L., GuoqiXie, L., Junfeng, Y., 2017. Time and energy optimization algorithms for the static scheduling of multiple workflows in heterogeneous computing system. Journal of Grid Computing , 15(4), pp. 435-456

[26] Kalka Dubey, Mohit Kumar, S.C. Sharma, “Modified HEFT Algorithm for Task Scheduling in Cloud Environment”, Procedia Computer Science, Volume 125, 2018, Pages 725-732, ISSN 1877-0509,

[27] Liu, L., Zhang, M., Buyya, R. and Fan, Q. (2016), “Deadline-constrained coevolutionary genetic algorithm for scientific workflow scheduling in cloud computing.” Concurrent Computant.:Pract. Exper., doi:10.1002/cpe.3942.

[28] Kumar, B., Mala, K., Poonam, S., (2017). Discrete binary cat swarm optimization for scheduling workflow applications in cloud systems. In Computational Intelligence & Communication Technology (CICT), 2017 3rd International Conference ,pp.1-6

[29] J. Meena, M. Kumar and M. Vardhan, "Cost Effective Genetic Algorithm for Workflow Scheduling in Cloud Under Deadline Constraint," in IEEE Access, vol. 4, pp. 5065-5082, 2016, doi: 10.1109/ACCESS.2016.2593903.

[30] R. Graves, T. H. Jordan, S. Callaghan, E. Deelman, E. Field, G. Juve, C. Kesselman, P. Maechling, G. Mehta, K. Milner et al., “Cybershake: A physics-based seismic hazard model for southern california,” Pure and Applied Geophysics, vol. 168, no. 3-4, pp. 367–381, 2011.)

[31] A. Iosup, S. Ostermann, M. N. Yigitbasi, R. Prodan, T. Fahringer and D. Epema, "Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing," in IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 6, pp. 931-945, June 2011.

[32] Callaghan, Scott, et al. "Scaling up workflow-based applications." Journal of Computer and System Sciences 76.6 (2010): 428-446.

[33] Deelman, E.; Vahi, K.; Juve, G.; Rynge, M.; Callaghan, S.; Maechling, P.J.;Wenger, K. Pegasus, a workflow management system for science automation. Future Gener. Comput. Syst. 2015, 46, 17–35.

[34] E. Deelman, et al., Pegasus: A framework for mapping complex scientific onto distributed systems, Sci. Program. 13 (3) (2005) 219–237.

[35] Li, Zhongjin&Ge, Jidong& Hu, Haiyang& Song, Wei & Hu, Hao&Luo, Bin. (2015). Cost and Energy Aware Scheduling Algorithm for Scientific Workflows with Deadline Constraint in Clouds. IEEE Transactions on Services Computing. 1-1. 10.1109/TSC.2015.2466545.

[36] Verma, A., Sakshi, K., 2010. A hybrid multi-objective Particle Swarm Optimization for scientific workflow scheduling. Parallel Computing 62 , pp.1-19

[37] Arabnejad, Vahid&Bubendorfer, Kris & Ng, Bryan. (2018). Budget and Deadline Aware e-Science Workflow Scheduling in Clouds. IEEE Transactions on Parallel and Distributed Systems. PP. 1-1. 10.1109/TPDS.2018.2849396.

[38] “USC Epigenome Center,” http://epigenome.usc.edu, accessed: October 2015.

[39] Xiaonian Wu, Mengqing Deng, Runlian Zhang, Bing Zeng, Shengyuan Zhou, "A Task Scheduling Algorithm based on QoS driven in Cloud Computing", Information Technology and Quantitative Management, 2013.

[40] Dr Ajay jangra, Tushar Saini,” Scheduling Optimization in Cloud Computing,” International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 4, April 2013.

[41] TAREGHİAN, Shahab, and Zarintaj BORNAEE. "A new approach for scheduling jobs in cloud computing environment." Cumhuriyet Science Journal 36.3 (2015): 2499-2506.

[42] Vinothina, V., Sridaran, R., 2018. An Approach for Workflow Scheduling in Cloud Using ACO. Big Data Analytics, Springer, Singapore, pp.525-531

[43] Buyya et al., "Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility," Future Generation Computer Systems, vol. 25, no. 6, pp. 599–616, 2009.

[44] M. Mao and M. Humphrey, "Auto-scaling to minimize cost and meet application deadlines in cloud workflows," SC '11: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, Seatle, WA, 2011, pp. 1-12.

[45] Zhang, Qi, Lu Cheng, and RaoufBoutaba. "Cloud computing: state-of-the-art and research challenges." Journal of internet services and applications 1.1 (2010): 7-18.

[46] Ramandeep Sandhu &Khushboo Khanna (2017). Satisfaction: A Scale to Fulfill Consumer's Expectation on Cloud Computing. IJRECE. VOl 5 Issue 3.

[47] Sung Ho Jang, Tae Young Kim, Jae Kwon Kim, Jong Sik Lee School, "The Study of Genetic Algorithm-based Task Scheduling for Cloud Computing", International Journal of Control and Automation Vol. 5, No. 4, December 2012.

[48] Ramandeep Sandhu and KamleshLakhwani (2020), Cloud System Enhancement using improved Workflow Task Ranking System, in Test Engineering and Management, ISSN: 0193-4120, Vol. 83, Page No. 10092 – 10101.

[49] MonirAbdullaha, Mohamed Othmanb, Cost-Based Multi QoS Job Scheduling using Divisible Load Theory in Cloud Computing, International Conference on Computational Science, ICCS 2013.

[50] Wang, Y., Jiajia, J., Yunni, X., Quanwang, W., Xin, L., Qingsheng, Z., (2018). 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, Springer, Cham, pp. 137-152

[51] Zhu, Z.; Zhang, G.; Li, M.; Liu, X.: Evolutionary multi-objective workflow scheduling in cloud. IEEE Trans. Parallel Distrib. Syst. 27, 1344–1357 (2016)

[52] Fard, H. M., Prodan, R., and Fahringer, T. 2013. A truthful dynamic workflow scheduling mechanism for commercial multicloud environments. Parallel and Distributed Systems, IEEE Transactions, 24(6), pp. 1203-1212.

[53] Singh, R. and S. Singh, Score based deadline constrained workflow scheduling algorithm for Cloud systems. International Journal on Cloud Computing: Services and Architecture (IJCCSA), 2013. 3(6): p. 31-41.

[54] Zhao, Y., et al. (2011). Opportunities and challenges in running scientific workflows on the cloud. Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), International Conference on. IEEE

[55] Alameer, Z., Elaziz, M., Ewees, A., Ye, H. and Jianhua, Z.,: Forecasting gold price fluctuations using improved multilayer perception neural network and whale optimization algorithm. Resources Policy, 61, pp.250-260 (2019)

[56] Priyasharma, Kiranbirkaur, "Hybrid Artificial Bee Colony and Tabu Search Based Power Aware Scheduling for Cloud Computing", International Journal of Intelligent Systems and applications(IJISA), Vol.10, No.7, pp.39-47, 2018. DOI: 10.5815/ijisa.2018.07.04

[57] Ahmed, M., Houssein, E., Hassanien, A., Taha, A. and Hassanien, E.,; Maximizing lifetime of large-scale wireless sensor networks using multi-objective whale optimization algorithm. Telecommunication Systems, 72(2), pp.243-259 (2019)

[58] Lu Huang, Hai-shan Chen and Ting-ting Hu, "Survey on Resource Allocation Policy and Job Scheduling Algorithms of Cloud Computing" ,Journal of Software, Vol. 8, No. 2, February 2013, pp. 480-487.

[59] Arabnejad, H.; Barbosa, J.G.; Prodan, R. Low-time complexity budget deadline constrained workflow scheduling on heterogeneous resources. Future Gener. Comput. Syst. 2016, 55, 29–40.

[60] Ben OualidMedani K, Sayah S, Bekrar A,. : Whale optimization algorithm based optimal reactive power dispatch: a case study of the Algerian power system. Electr Power Syst Res 163:696–705(2017)

[61] https://pegasus.isi.edu/workflow_gallery/gallery/

[62] Sandhu, R. and Lakhwani, K.,: Optimal Cloud System Enhancement using improved Workflow Task Ranking System. Test Engineering and Management, 83(May - June), pp.10092 – 10101(2020)

[63] Shaw, Subhadra Bose, and A. K. Singh, “A survey on scheduling and load balancing techniques in cloud computing environment,” Computer and Communication Technology (ICCCT), 2014 International Conference on. IEEE, pp: 87-95, 2014.

[64] Chirkin, M., Adam, B., Sergey, K., Marc, M., Mikhail, M., Alexander, V., Denis, A., 2017. Execution time estimation for workflow scheduling. Future Generation Computer Systems75 , pp.376-387

[65] Zhao, Yong, et al. "Opportunities and challenges in running scientific workflows on the cloud." Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2011 International Conference on. IEEE, 2011.

[66] Shi Z, Dongarra JJ (2006) Scheduling workflow applications on processors with different capabilities. Future GenerComputSyst 22(6):665–675

[67] A. GhorbanniaDelavar and Y. Aryan, “HSGA: A hybrid heuristic algorithm for workflow scheduling in cloud systems,” Cluster Computing, vol. 17, no. 1, pp. 129–137, 2014.

[68] Abrishami, S., M. Naghibzadeh, and D.H. Epema, Deadline-constrained workflow scheduling algorithms for Infrastructure as a Service Clouds. Future Generation Computer Systems, 2013. 29(1): p. 158-169.

[69] Zhang, J., Chen, X., Li, J., Li, X.: Task mapper and application-aware virtual machine scheduler oriented for parallel computing. J. Zhejiang Univ. Sci. C. 13, 155–177 (2012).

[70] Meena, J.; Kumar, M.; Vardhan, M.: Cost Effective Genetic Algorithm for Workflow Scheduling in Cloud Under Deadline Constraint. IEEE Access 4, 5065–5082 (2016)

[71] Verma, A. and Kaushal, S. (2015), ‘Cost-time efficient scheduling plan for executing workflows in the cloud’, Journal of Grid Computing 13(4), 495–506.

[72] Choudhary, A., et al 2018. A GSA based hybrid algorithm for bi-objective workflow scheduling in cloud computing. Future Generation Computer Systems, pp.1-10

[73] Kwok, Y.K., and Ahmad, I., 1996. Dynamic Critical-Path Scheduling: An Effective Technique for Allocating Task Graphs to Multiprocessors. IEEE Trans. Parallel and Distributed Systems, Vol. 7, No. 5, pp. 506-521.

[74] Shiroor, A., Springer, J., Hacker, T., Marshall, B., Brewer, J.: Scientific workflow management systems and workflow patterns. Int. J. Bus. Process Integr. Manag. 5, 63 (2010).

[75] M. Manasrah, Ahmad & Ba Ali, Hanan. (2018). Workflow Scheduling Using Hybrid GA-PSO Algorithm in Cloud Computing. Wireless Communications and Mobile Computing. 2018. 1-16. 10.1155/2018/1934784.

[76] S. Mirjalili and A. Lewis, “The whale optimization algorithm,” Advances in Engineering Software, vol. 95, pp. 51–67, 2016.

[77] M. A. Rodriguez and R. Buyya, “Deadline based Resource Provisioning and Scheduling Algorithm for Scientific Workflows on Clouds”. IEEE Transactions on Cloud Computing. pp- 1- 14, 2013.

[78] Ramandeep Sandhu and KamleshLakhwani (2021), Improving Cloud Environment with Deadline Constrained Based Scientific Workflow Scheduling, in International Conference on Communications and Cyber-Physical Engineering (ICCCE 2021), 978-981-16-7984-1/ Vol. 828.

[79] Barker and J. Van Hemert, “Scientific workflow: a survey and research directions,” in Parallel Processing and Applied Mathematics. Springer, 2008, pp. 746–753.

[80] E.-K. Byun, Y.-S. Kee, J.-S. Kim, and S. Maeng, “Cost optimized provisioning of elastic resources for application workflows,” Future Generation Computer Systems, vol. 27, no. 8, pp. 1011–1026, 2011.

[81] Karaboga, Dervis, and BahriyeBasturk, “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm,” Journal of global optimization, vol: 39, issue: 3, pp: 459-471, 2007.

[82] Al-maamari, Ali, and Fatma A. Omara, “Task scheduling using PSO algorithm in cloud computing environments,” International Journal of Grid and Distributed Computing, vol: 8, issue: 5, pp: 245-256, 2015.

[83] Vöckler, Jens-Sönke, Juve Gideon, DeelmanEwa, Rynge Mats and Berriman Bruce, "Experiences using cloud computing for a scientific workflow application." Proceedings of the 2nd international workshop on Scientific cloud computing. ACM, 2011.

[84] Rimal, Bhaskar Prasad, and Martin Maier. "Workflow scheduling in multi-tenant cloud computing environments." IEEE Transactions on Parallel and Distributed Systems 28.1 (2017): 290-304.

[85] Genez, T., Pietri, I., Sakellariou, R., Bittencourt, F., Madeira, E., 2015. A Particle Swarm Optimization Approach for Workflow Scheduling on Cloud Resources Priced by CPU Frequency. Data Science and Symptoms , pp.1-9

[86] S. Su, J. Li, Q. Huang, X. Huang, K. Shuang, J. Wang, Cost-efficient task scheduling for executing large program in the cloud, Parallel Comput. 39 (4) (2013) 177–188

[87] Y. Gil, E. Deelman, M. Ellisman, T. Fahringer, G. Fox, D. Gannon, C. Goble, M. Livny, L. Moreau, and J. Myers, “Examining the challenges of scientific workflows,” IEEE Computer, vol. 40, no. 12, pp. 26–34, 2007.

[88] Casas, I., Taheri, J., Ranjan, R. and Zomaya, A. Y. (2017), ‘Pso-ds: a scheduling engine for scientific workflow managers’, The Journal of Supercomputing 73(9), 3924–3947.

[89] Foster, I.|Kesselman, C.: The : Blueprint for a Future Computing Infrastructure. Morgan Kaufmann Publishers, USA, 1999.

[90] Chaptin, S.J., “Distributed and Multiprocessor Scheduling”, University of Minnesota, 2003.

[91] Sagnika, Santwana, SaurabhBilgaiyan, and Bhabani Shankar Prasad Mishra."Workflow Scheduling in Cloud Computing Environment Using Bat Algorithm." Proceedings of First International Conference on Smart System, Innovations and Computing.Springer, Singapore, 2018.

[92] Zhang, Longxin, Kenli Li, Changyun Li, and Keqin Li. "Bi-objective workflow scheduling of the energy consumption and reliability in heterogeneous computing systems." Information Sciences 379 (2017): 241-256.

[93] Barker and J. Van Hemert, “Scientific workflow: a survey and research directions,” in Parallel Processing and Applied Mathematics. Springer, 2008, pp. 746–753.

[94] Zhou, Xiumin& Zhang, Gongxuan& Sun, Jin & Zhou, Junlong& Wei, Tongquan& Hu, Shiyan. (2018). Minimizing Cost and Makespan for Workflow Scheduling in Cloud using Fuzzy Dominance Sort based HEFT. Future Generation Computer Systems.93. 10.1016/j.future.2018.10.046.

[95] Wang, Guan & Wang, Yuxin& Liu, Hui&Guo, He. (2016). HSIP: A Novel Task Scheduling Algorithm for Heterogeneous Computing. Scientific Programming. 2016. 1-11. 10.1155/2016/3676149.

[96] Abbas, M. and Yang, Z. (2010), Optimized-fuzzy-logic-based bit loading algorithms, IntechOpen.

[97] Arabnejad, H. and Barbosa, J. G. (2014), ‘List scheduling algorithm for heterogeneous systems by an optimistic cost table’, IEEE Transactions on Parallel and Distributed Systems 25(3), 682–694.

[98] Garey, M. R. and Johnson, D. S. (1990), Computers and Intractability; A Guide to the Theory of NP-Completeness, W. H. Freeman and Co.

[99] Gideon, J., Ann, C., Ewa, D., Shishir, B., Gaurang, M. and Karan, V. (2013), ‘Characterizing and profiling scientific workflows’, Future Generation Computer Systems 29(3), 682 – 692.

[100] Ghanbari, S. and Othman, M. (2012), ‘A priority based job scheduling algorithm in cloud computing’, Procedia Engineering 50, 778–785.

[101] Gupta, K. and Singh2, M. (2012), 'Heuristic based task scheduling in grid', International Journal of Engineering and Technology 4

[102] Hai, H. and Sakoda, S. (2009), 'Saas and integration best practices', Fujitsu Science Tech Journal 45(3), 257–264.

[103] Madni, S. H. H., Latiff, M. S. A., Abdullah, M., Abdulhamid, S. M. and Usman, M. J. (2017), 'Performance comparison of heuristic algorithms for task scheduling in iaas cloud computing environment', PLoS ONE 12(5).

[104] Mesbahi, M. R., Rahmani, A. M. and Hosseinzadeh, M. (2018), 'Reliability and high availability in cloud computing environments: a reference roadmap', Human-centric Computing and Information Sciences 8(1), 1–31.

[105] Rawat, P. S., Dimri, P. and Barthwal, V. (2016), 'A priority base task scheduling on virtual machines using workflowsim', Information Technology Journal 15, 39–45.

[106] Rodriguez, M. A. and Buyya, R. (2017), 'A taxonomy and survey on scheduling algorithms for scientific workflows in iaas cloud computing environments', Concurrency and Computation: Practice and Experience 29(8), 1–23.

[107] Singh, L. and Singh, S. (2014), A genetic algorithm for scheduling workflow applications in unreliable cloud environment, in 'Recent Trends in Computer Networks and Distributed Systems Security', Springer Berlin Heidelberg, pp. 139–150.

[108] Garg, S. K., Versteeg, S., and Buyya, R. 2013. A framework for ranking of cloud computing services. Future Generation Computer Systems, 29(4), 1012-1023.

[109] Kirkpatrick S. 1984. Optimization by simulated annealing: quantitative studies. J Stat Phys ;34:975–86.

[110] Mirjalili, S. (2015). The antlion optimizer. Advances in Engineering Software, 83, 80-98.

[111] E. Hou, N. Ansari, and H. Ren., “A genetic algorithm for multiprocessor Scheduling”, IEEE Transactions on Parallel and Distributed Systems, 5(2):113-120, Feb. 1994.

[112] Beloglazov, A., Abawajy, J., and Buyya, R. 2012. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future Generation Computer Systems, 28(5), 755-768.

[113] Hoos HH, Stützle T. 2004. Stochastic local search: foundations & applications. Elsevier.

[114] Eberchart, R., and Kennedy, J. 1995. A new optimizer using particle swarm theory, Proceedings of the International Symposium on Micro Machine and Human Science, pp. 39–43 9.

[115] Clerc, M., and Kennedy, J. 2002. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. Evolutionary Computation, IEEE Transactions, 6(1), 58-73.

[116] Zhulin Li, Z. L., Cuirong Wang, C. W., HaiyanLv, H. L., and Xin Song, X. S. (2014). Scheduling Tasks on Heterogeneous Multi-Core Processors Based on Modified Ant Colony Optimization. International Journal of Control and Automation, 7(9), 345-356.

[117] Li, K., Xu, G., Zhao, G., Dong, Y., and Wang, D. 2011. Cloud task scheduling based on load balancing ant colony optimization. In Chinagrid Conference (ChinaGrid), 2011 Sixth Annual, IEEE, pp. 3-9.

[118] Hu, X. X., and Zhou, X. W. 2014. Improved Ant Colony Algorithm on Scheduling Optimization of Cloud Computing Resources. In Applied Mechanics and Materials, Vol. 678, pp. 75-78.

[119] Gálvez, A., and Iglesias, A. 2013. A new iterative mutually coupled hybrid GA–PSO approach for curve fitting in manufacturing. Applied Soft Computing,13(3), 1491-1504.

[120] Chen, W. N., and Zhang, J. 2012. A set-based discrete PSO for cloud workflow scheduling with user-defined QoS constraints. In Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference, IEEE, pp. 773-778.

[121] Buyya, R., Garg, S. K., and Calheiros, R. N. 2011. SLA-oriented resource provisioning for cloud computing: Challenges, architecture, and solutions. In Cloud and Service Computing (CSC), 2011 International Conference, IEEE , pp. 1-10.

[122] Aron, R., and Chana, I. 2012. Formal QoS policy based Grid resource provisioning framework. Journal of Grid Computing, 10(2), 249-264.

[123] Zhao, Y., Li, Y., Tian, W., and Xue, R.., 2012. Scientific-Workflow-Management-as-a-Service in the Cloud. In Cloud and Green Computing (CGC), 2012 Second International Conference on (pp. 97-104).

[124] Ryan, M. D. 2013. Cloud computing security: The scientific challenge, and a survey of solutions. Journal of Systems and Software, 86(9), 2263-2268.

[125] L. Ramakrishnan& D. Gannon, 2008, „A Survey of Distributed Workflow Characteristics & Resource Requirements", Indiana University Technical Report TR671.

[126] L. Hongyou, W. Jiangyong, P. Jian, W. Junfeng& L. Tang, 2013 "Energy-aware scheduling scheme using workload-aware consolidation technique in cloud data centres," in China Communications, vol. 10, no. 12, pp. 114-124, Dec. 2013. doi: 10.1109/CC.2013.6723884.

[127] Yu, R. Buyya, 2008, „Workflow scheduling Algorithms for Grid computing", metaheuristics for scheduling in distributing computing environment, Springer, Berlin.

[128] O. M. Elzeki, M. Z. Reshad, M.A. Elsoud, 2012," Improved Max Min Algorithm in Cloud Computing "in International Journal of Computer Applications, vol. 50.

[129] Li, J, Ma, T, Tang, M., Shen, W, Jin Y,2017," Improved FIFO Scheduling Algorithm Based on Fuzzy Clustering in Cloud Computing" in journal of Information vol. 8,no.1,pp. 25.DOI: 10.3390/info8010025.

[130] Y. Zhao, X. Fei, I. Raicu& S. Lu, 2011, „Opportunities & Challenges in Running Scientific Workflows on the Cloud," 2011 International Conference on Cyber-Enabled Distributed Computing & Knowledge Discovery, Beijing, pp. 455-462.doi: 10.1109/CyberC.2011.80.

[131] Jailalita, Singh, S. and Dutta, M. (2016), Critical path based scheduling algorithm for workflow applications in cloud computing, in '2016 International Conference on Advances in Computing, Communication, Automation (ICACCA) (Spring)', pp. 1–6.

[132] V. Arabnejad& K. Bubendorfer, 2015, Cost Effective & Deadline Constrained Scientific Workflow Scheduling for Commercial Clouds,‟ 2015 IEEE 14th International Symposium on Network Computing & Applications, Cambridge, MA, pp.106-113.doi: 10.1109/NCA.2015.33.

[133] V. Choudhary, S. Kacker, T. Choudhury, V. Vashisht, 2012 „An approach to improve task scheduling in a decentralized cloud computing environment‟, in International Journal of Computing Technology & Applications vol. 3, no.1 2012, pp 312–316.

[134] Lal, A. and Rama Krishna, C. (2018), Critical path-based ant colony optimization for scientific workflow scheduling in cloud computing under deadline constraint, in 'Ambient Communications and Computer Systems', Springer Singapore, Singapore, pp. 447–461.

[135] A.S. Kaviani and Z.G. Vranesic, "On Scheduling in Multiprocessor Systems Using Fuzzy Logic", 2^{4th} International Symposium on Multiple-valued Logic, pp. 141-148, May 1994.

[136] Hamzeh m., Fakhraies.m., lucas C., "Soft real-time fuzzy task scheduling for multiprocessor systems", In Proceedings of World Academy of Science, Engineering and Technology, 2007.

[137] Harik GR, Lobo FG, and Goldberg DE, "The compact genetic algorithm", IEEE transactions on evolutionary computation, vol. 3, no. 4, November 1, 1999.

[138] HesamIzakian, Ajith Abraham, Vaclav Snasel, "Comparison of Heuristics for scheduling Independent Tasks on Heterogeneous Distributed Environments", Proceedings of the 2009 International Joint Conference on Computational Sciences and Optimization, Volume 01, Pages: 8-12, ISBN:978-0-7695-3605-7, IEEE Computer Society Washington, DC, USA, 2009.

[139] L. F. Bittencourt, R. Sakellariou, and E. R. M. Madeira, "DAG Scheduling Using a Look ahead Variant of the Heterogeneous Earliest Finish Time Algorithm", in Parallel, Distributed and Network- Based Processing (PDP), 2010 18th Euromicro International Conference on, pp. 27-34, 2010.

[140] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd, "The Pagerank citation algorithm: bringing order to the web", In Proceedings of the seventh conference on World WideWeb, Brisbane, Australia, April 1998.

[141] Min Chen, Simone A. Ludwig, "Fuzzy-guided Genetic Algorithm applied to the Web Service Selection Problem", proceedings of IEEE international conference of fuzzy systems, Brisbane, Australia, June 2012.

[142] M. Hellmann, "Fuzzy logic introduction", Laboratories Antennas Radar Telecom, 2001.

[143] Lee C. C., "Fuzzy logic in control systems: fuzzy logic controller-parts 1 and 2", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 20, No. 2, pp 404-435, 1990.

[144] N. Doulamis, E. Varvarigos, T. Varvarigou, "Fair Scheduling Algorithms in Grids", IEEE TPDS,2007.

[145] P.-C. Wang and W. Korfhage, "Process scheduling using genetic algorithms", In IEEE Symp. On Parallel and Dist. Proc., pages 638-641, Texas, USA, Oct. 1995.

[146] Pedro A. Diaz-Gomez and Dean F. Hougen, "Initial Population for Genetic Algorithms: A Metric Approach", International Conference, 2007.

[147] R. Sakellariou, H. Zhao, "A hybrid heuristic for dag scheduling on heterogeneous systems", in Proceedings of 13th Heterogeneous Computing Workshop, HCW2004, Santa Fe, NM, 2004.

[148] Yi-Hsuan Lee and Cheng Chen, "A Modified Genetic Algorithm for Task Scheduling in Multiprocessor Systems," Proc. of 6th International Conference Systems and Applications, pp. 382- 387, 1999.

[149] Y. Kwok and I. Ahmad, "Static Scheduling Algorithms for Allocating Directed Task Graphs to Multiprocessors", ACM Computing Surveys, vol. 31, no. 4, pp. 406-471, 1999.

[150] Tom V. Mathew, "Genetic algorithm", http://www.civil.iitb.ac.in/tvm/2701_dga/2701-ganotes/gadoc/gadoc.html.

[151] Wieczorek, M., Prodan, R. and Fahringer, T. (2005), 'Scheduling of scientific workflows in the askalon grid environment', SIGMOD Rec. 34(3), 56–62.

[152] Cao, H., Wu, X., Wu, S. and Shi, X. (2010), 'Dagmap: efficient and dependable scheduling of dag workflow job in grid', The Journal of Supercomputing 51(2), 201223.

[153] Arabnejad, H.; Barbosa, J.G.: A budget constrained scheduling algorithm for workflow applications. J. Grid Comput. 12, 665–679 (2014)

[154] G. N. Reddy, Kumar SP, "multi objective task scheduling algorithm for cloud computing using whale optimization technique," in International Conference on Next Generation Computing Technologies, Springer, Cham, Switzerland, 2017.

[155] "Gravitational Waves Detected 100 Years After Einstein's Prediction," https://www.ligo.caltech.edu/news/ligo20160211, accessed: October 2015.

[156] "LIGO Detection Portal," https://www.ligo.caltech.edu/detection, accessed: October 2015.

[157] "Southern California Earthquake Center," https://www.scec.org/, accessed: October 2015.

[158] Bittencourt, L., Roberto, E., Madeira, M., 2011. HCOC: a cost optimization algorithm for workflow scheduling in hybrid clouds. Journal of Internet Services and Applications, 2(3), pp.207-227.

[159] Huang, Kuo-Chan & Tsai, Ying-Lin & Liu, Hsiao-Ching. (2015). Task ranking and allocation in list-based workflow scheduling on parallel

computing platform. The Journal of Supercomputing. 71. 217-240. 10.1007/s11227-014-1294-7.

[160] Marrone, Stefano. (2015). Using Bayesian networks for highly available cloud-based web applications. Journal of Reliable Intelligent Environments. 1. 10.1007/s40860-015-0009-z.

[161] Zeng, L., Veeravalli, B. and Li, X. (2015), 'Saba: A security-aware and budget-aware workflow scheduling strategy in clouds', Journal of Parallel and Distributed Computing 75, 141–151.

[162] Yu-Kwong Kwok and Ahmad, I. (1996), 'Dynamic critical-path scheduling: an effective technique for allocating task graphs to multiprocessors', IEEE Transactions on Parallel and Distributed Systems 7(5), 506–521.

[163] Topcuoglu, H., Hariri, S. and Min-You Wu (2002), 'Performance-effective and low-complexity task scheduling for heterogeneous computing', IEEE Transactions on Parallel and Distributed Systems 13(3), 260–274.

[164] Arabnejad, H. and Barbosa, J. G. (2014), 'List scheduling algorithm for heterogeneous systems by an optimistic cost table', IEEE Transactions on Parallel and Distributed Systems 25(3), 682–694.

[165] FarisLlwaah, Nigel Thomas, JacekCa la,2015,"Improving MCT scheduling algorithm to reduce the makespan& cost of workflow execution in the cloud" in 31st UK Performance Engineering Workshop 17, pp. 1 – 8.

[166] Verma, A.; Kaushal, S.: Cost minimized PSO based workflow scheduling plan for cloud computing. Int. J. Inf. Technol. Comput. Sci. 8, 37–43 (2015)

[167] Parekh, H. B. and Chaudhari, S. (2016), Improved round robin cpu scheduling algorithm: Round robin, shortest job first and priority algorithm coupled to increase throughput and decrease waiting time and turnaround time, in '2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC)', pp. 184–187.

[168] Ramezani, F., Lu, J. and Hussain, F. (2013), Task scheduling optimization in cloud computing applying multi-objective particle swarm optimization, in 'Service-Oriented Computing', Springer Berlin Heidelberg, pp. 237–251.

[169] Casas, Israel, et al. "A balanced scheduler with data reuse and replication for scientific workflows in cloud computing systems." Future Generation Computer Systems 74 (2017): 168-178.

[170] Pandey, S., Wu, L., Guru, S. M. and Buyya, R. (2010), A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments, in '2010 24th IEEE International Conference on Advanced Information Networking and Applications', pp. 400–407.

[171] Thanh, T. P., The, L. N. and Doan, C. N. (2015), A novel workflow scheduling algorithm in cloud environment, in '2015 2nd National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS)', pp. 125–129.

[172] Zuo, X., Zhang, G. and Tan, W. (2014), 'Self-adaptive learning pso-based deadline constrained task scheduling for hybrid iaas cloud', IEEE Transactions on Automation Science and Engineering 11(2), 564–573.

[173] X. Xu, W. Dou, X. Zhang & J. Chen,2016, EnReal: An Energy-Aware Resource Allocation Method for Scientific Workflow Executions in Cloud Environment, in IEEE Transactions on Cloud Computing, vol. 4, no. 2, pp. 166-179, April-June 2016.doi: 0.1109/TCC.2015.2453966.

[174] Kaur, N. and Singh, S. (2016), 'A budget-constrained time and reliability optimization bat algorithm for scheduling workflow applications in clouds', Procedia Computer Science 98, 199 – 204.

[175] Nirmala, S. J. and Bhanu, S. M. S. (2016), 'Catfish-pso based scheduling of scientific workflows in iaas cloud', Computing 98(11), 1091–1109.

[176] Guo, W., Lin, B., Chen, G., Chen, Y. and Liang, F. (2018), 'Cost-driven scheduling for deadline-based workflow across multiple clouds',

IEEE Transactions on Network and Service Management 15(4), 1571–1585.

[177] Nasonov, D., Visheratin, A., Butakov, N., Shindyapina, N., Melnik, M. and Boukhanovsky, A. (2017), 'Hybrid evolutionary workflow scheduling algorithm for dynamic heterogeneous distributed computational environment', Journal of Applied Logic 24, 50 – 61.

[178] Mirzayi, S. and Rafe, V. (2015), 'A hybrid heuristic workflow scheduling algorithm for cloud computing environments', Journal of Experimental and Theoretical Artificial Intelligence 27(6), 721–735.

[179] Sreenu, Karnam&Sreelatha, Moturi. (2019). W-Scheduler: whale optimization for task scheduling in cloud computing. Cluster Computing. 22. 1-12. 10.1007/s10586-017-1055-5.

[180] Strumberger, Ivana &Bacanin, Nebojsa& Tuba, Milan & Tuba, Eva. (2019). Resource Scheduling in Cloud Computing Based on a Hybridized Whale Optimization Algorithm. Applied Sciences. 9. 4893. 10.3390/app9224893.

[181] F. Yiqiu, X. Xia and G. Junwei, "Cloud Computing Task Scheduling Algorithm Based On Improved Genetic Algorithm," 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 2019, pp. 852-856, doi: 10.1109/ITNEC.2019.8728996.

[182] Mohammed, Hardi& Umar, Shahla& Rashid, Tarik. (2019). A Systematic and Meta-Analysis Survey of Whale Optimization Algorithm. Computational Intelligence and Neuroscience. 2019. 25. 10.1155/2019/8718571.

[183] Christine Churchil, "Search Engine Algorithms and Research", Search Engine Watch, April 2005.

[184] DaliborFiala, François Rousselot, KarelJezek, "Ranking Algorithms for Web Sites", Finding Authoritative Academic Web Sites and Researchers.

[185] Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. Advances in engineering software, 95, 51-67.

[186] Z. He, J. Dong, Z. Li and W. Guo, "Research on Task Scheduling Strategy Optimization Based onACO in Cloud Computing Environment," 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), 2020, pp. 1615-1619, doi: 10.1109/ITOEC49072.2020.9141743.

[187] Ramamoorthy, S., G. Ravikumar, B. SaravanaBalaji, S. Balakrishnan, and K. Venkatachalam. "MCAMO: multi constraint aware multi-objective resource scheduling optimization technique for cloud infrastructure services." Journal of Ambient Intelligence and Humanized Computing 12, no. 6 (2021): 5909-5916.

[188] Mahendra Kumar Gourisaria, Palak Gupta, Harshvardhan G. M., S. S. Patra, P. M. Khilar (2020) “A Comparative Study of Various Task Scheduling Algorithms in Cloud Computing”, International Journal of Control and Automation, 13(4), pp. 1152-1169.

[189] K Pradeep, L Javid Ali, N Gobalakrishnan, C J Raman, N Manikandan, CWOA: Hybrid Approach for Task Scheduling in Cloud Environment, The Computer Journal, 2021;, bxab028.

[190] Bozorgi, Seyed&Yazdani, Samaneh. (2019). IWOA: An Improved whale optimization algorithm for optimization problems. Journal of Computational Design and Engineering. 6. 10.1016/j.jcde.2019.02.002.

[191] G. Berriman, A. Laity, J. Good, J. Jacob, D. Katz, E. Deelman, G. Singh, M. Su, and T. Prince, “Montage: The architecture and scientific applications of a national virtual observatory service for computing astronomical image mosaics,” in Proceedings of Earth Sciences Technology Conference, 2006.

[192] RamandeepSandhu and KamleshLakhwani (2021), Scientific Workflow Scheduling by Adaptive Approaches with Convex Optimization in Cloud Environment, in Design Engineering, ISSN: 0011-9342, Vol. 2021, Issue 07. 1686- 1712.

[193] Zhang, Longxin, Kenli Li, Changyun Li, and Keqin Li. "Bi-objective workflow scheduling of the energy consumption and reliability in heterogeneous computing systems." Information Sciences 379 (2017): 241-256.

LIST OF PUBLICATIONS

1. Ramandeep Sandhu and KamleshLakhwani (2021), Scientific Workflow Scheduling by Adaptive Approaches with Convex Optimization in Cloud Environment, in Design Engineering, ISSN: 0011-9342, Vol. 2021, Issue 07. 1686- 1712.

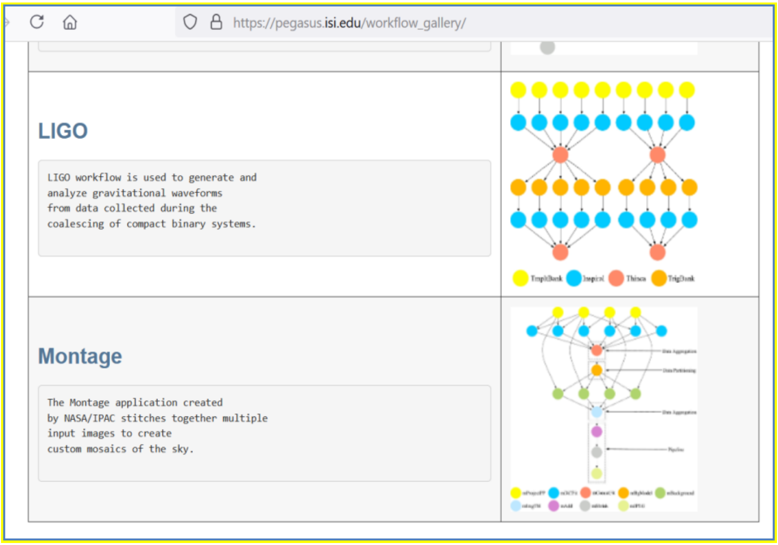
2. Ramandeep Sandhu and KamleshLakhwani (2021), Improved Scientific Workflow Scheduling Algorithm with distributed HEFT Ranking and TBW Scheduling Method, in International Conference on Wireless Sensor Networks, Ubiquitous Computing and Applications (ICWSNUCA 2021), ISSN- 2367-3370/ Vol. 244, issue no. 1.

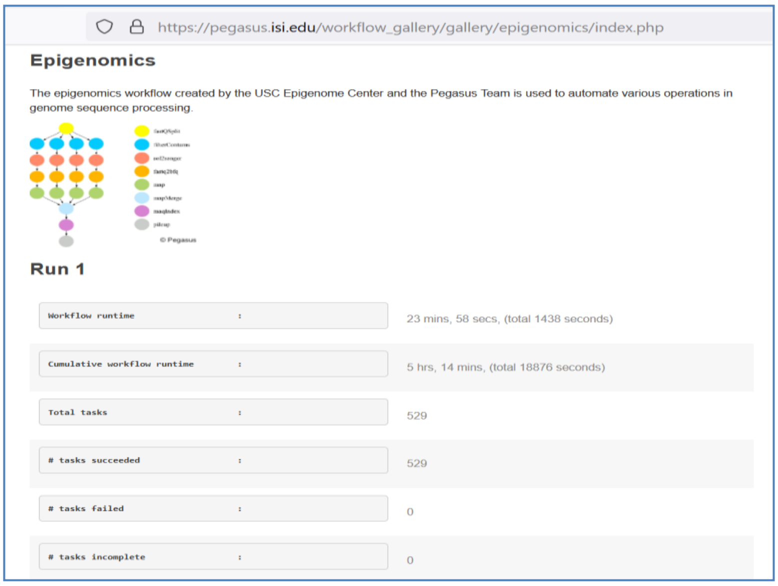
3. Ramandeep Sandhu and KamleshLakhwani (2021), Improving Cloud Environment with Deadline Constrained Based Scientific Workflow Scheduling, in International Conference on Communications and Cyber-Physical Engineering (ICCCE 2021), 978-981-16-7984-1/ Vol. 828.

4. Ramandeep Sandhu and KamleshLakhwani (2020), Cloud System Enhancement using improved Workflow Task Ranking System, in Test Engineering and Management, ISSN: 0193-4120, Vol. 83, Page No. 10092 – 10101.

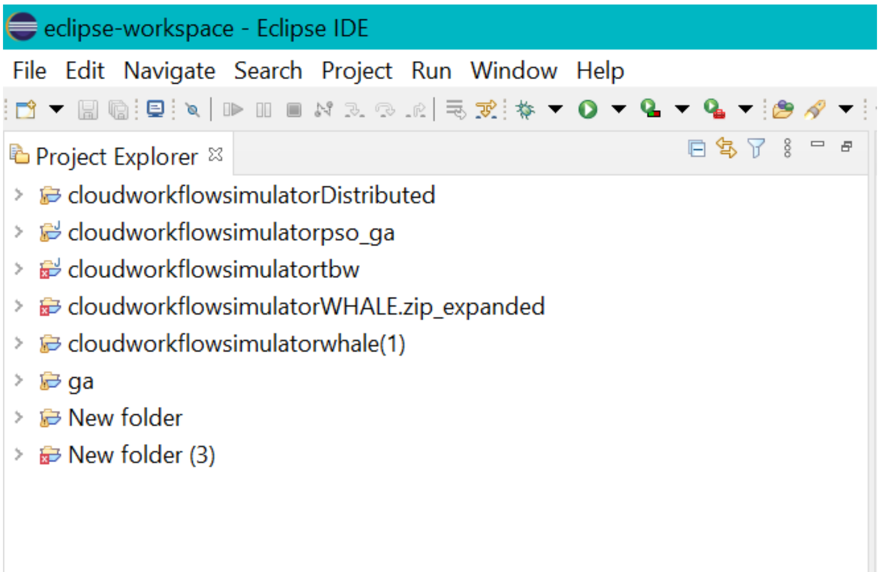
APPENDIX

Authentication of Input Data

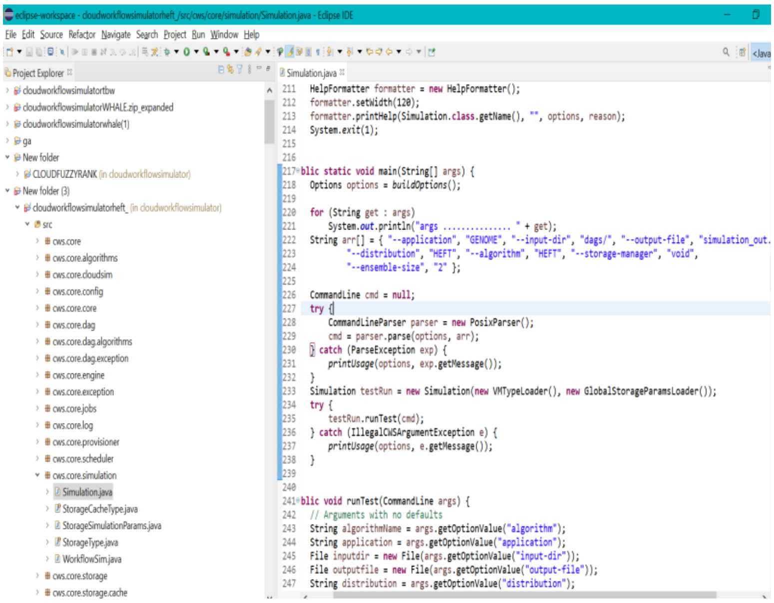




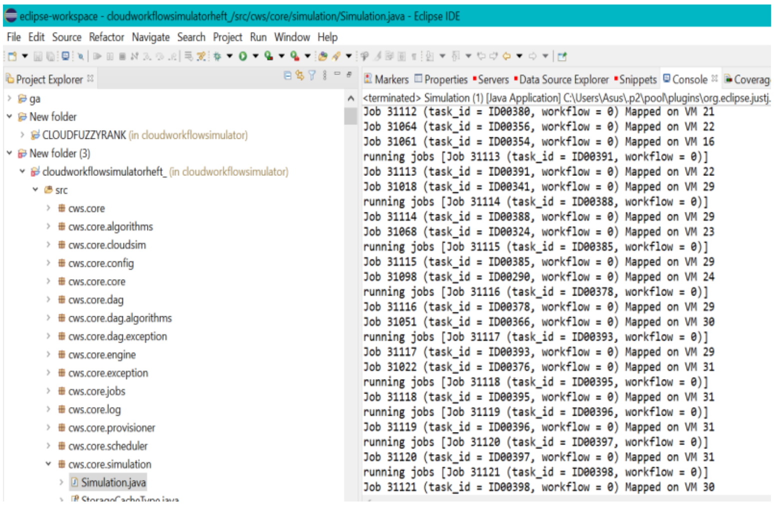
Various screenshots of Research Work Execution:



Comparison of Proposed Work with Existing Methodologies



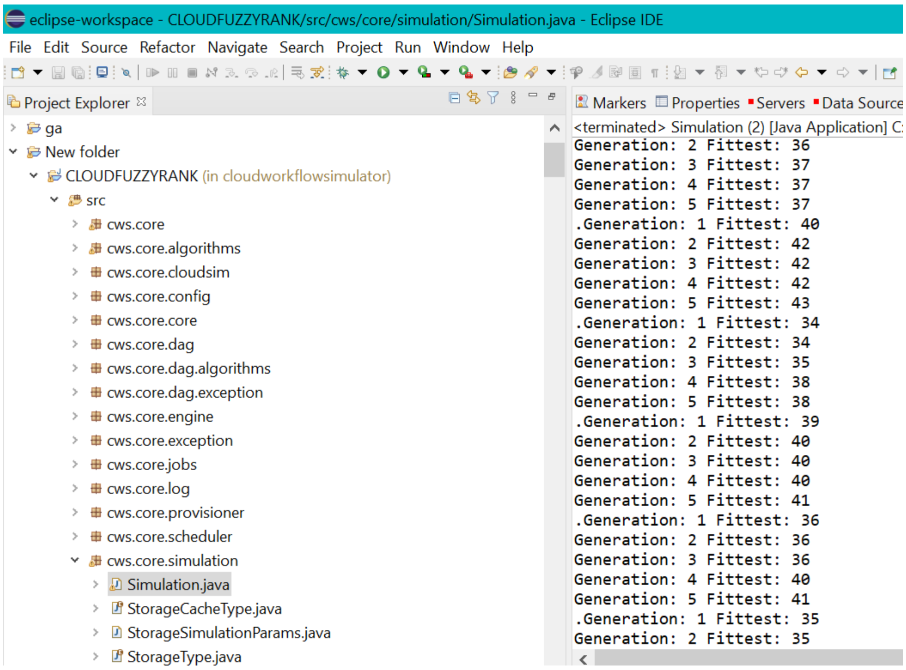
Above Screenshot is showing simulation code of HEFT algorithm when VM used are 2.



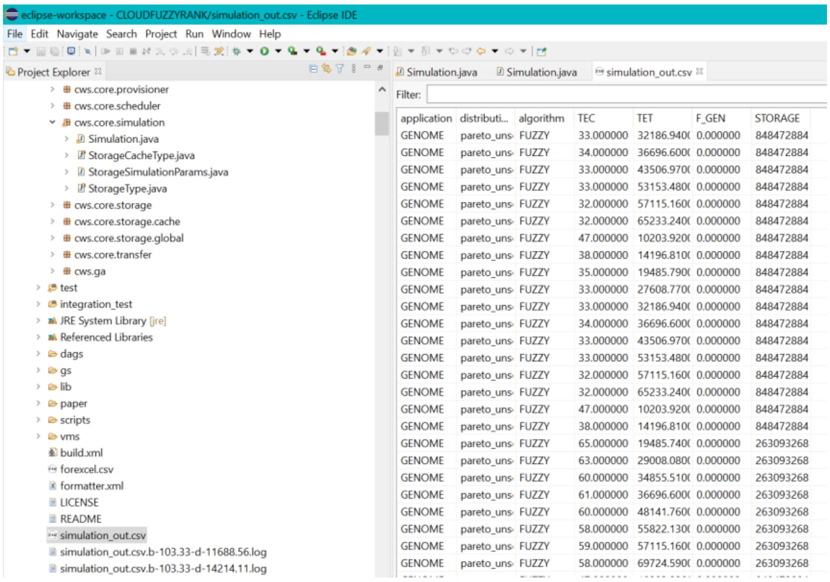
Output after execution of HEFT algorithm:

application	distribution	algorithm	cost	lastDagFinish	ResponseTime
GENOME	HEFT	GA	10	21	0.000043
GENOME	HEFT	GA	9	5	0.00002
GENOME	HEFT	GA	9	10	0.00002
GENOME	HEFT	GA	13	38	0.000015
GENOME	HEFT	GA	10	8	0.000017
GENOME	HEFT	GA	8	7	0.000019
GENOME	HEFT	GA	12	15	0.000013
GENOME	HEFT	GA	5	25	0.000073
GENOME	HEFT	GA	14	3	0.000006
GENOME	HEFT	GA	5	16	0.000028
GENOME	HEFT	GA	6	21	0.000021
GENOME	HEFT	GA	5	31	0.000025
GENOME	HEFT	GA	44	22812.59	0.00297
GENOME	HEFT	GA	44	22812.59	0.001549
GENOME	HEFT	GA	44	22812.59	0.000904
GENOME	HEFT	GA	44	22812.59	0.000548
GENOME	HEFT	GA	43	26836.65	0.000576
GENOME	HEFT	GA	42	30856.28	0.000611
GENOME	HEFT	GA	42	32543.89	0.000537
GENOME	HEFT	GA	42	39271.01	0.000729
GENOME	HEFT	GA	5	27	0.000015
GENOME	HEFT	GA	2	33	0.000038
GENOME	HEFT	GA	44	22812.59	0.000745
GENOME	HEFT	GA	44	22812.59	0.000556
GENOME	HEFT	GA	44	22812.59	0.00051
GENOME	HEFT	GA	44	22812.59	0.000458
GENOME	HEFT	GA	43	26836.65	0.000426
GENOME	HEFT	GA	42	30856.28	0.000542

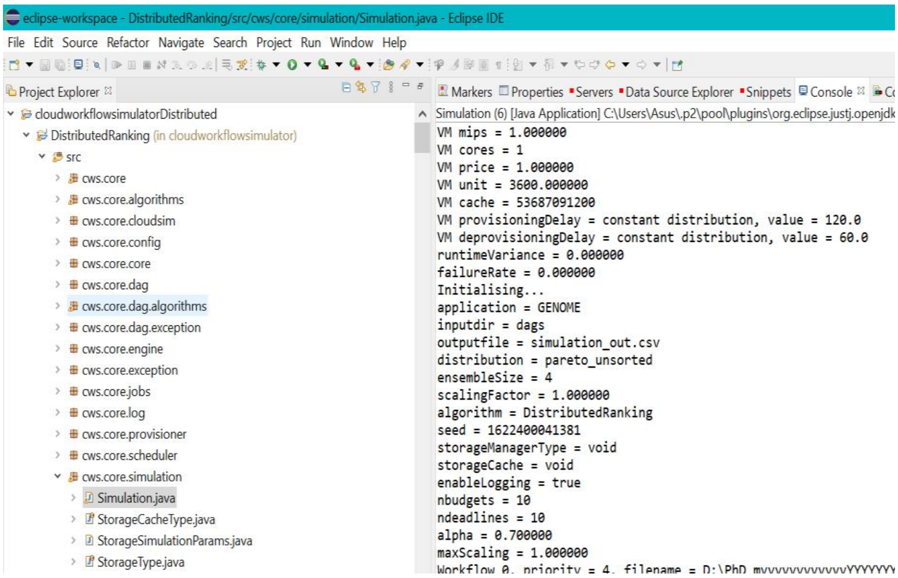
Below is screenshot of running Fuzzy Rank Algorithm:



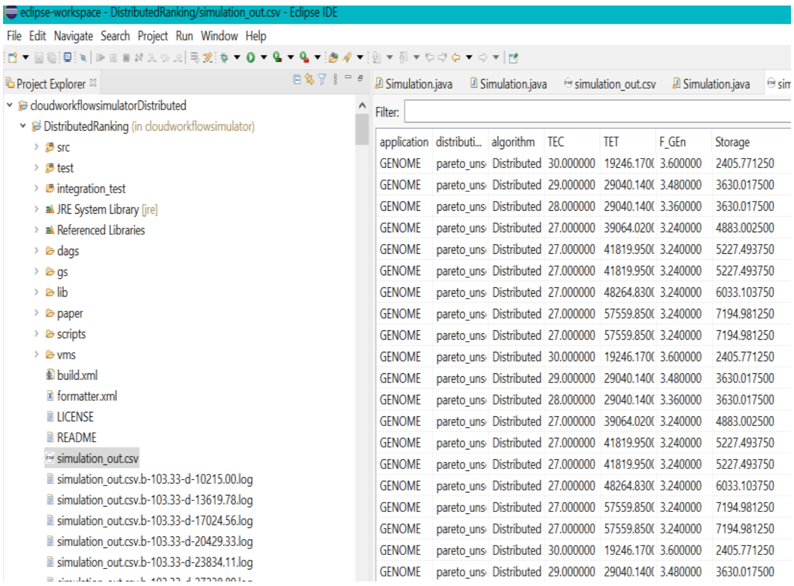
Output stored in simulation_out.csv file as below:



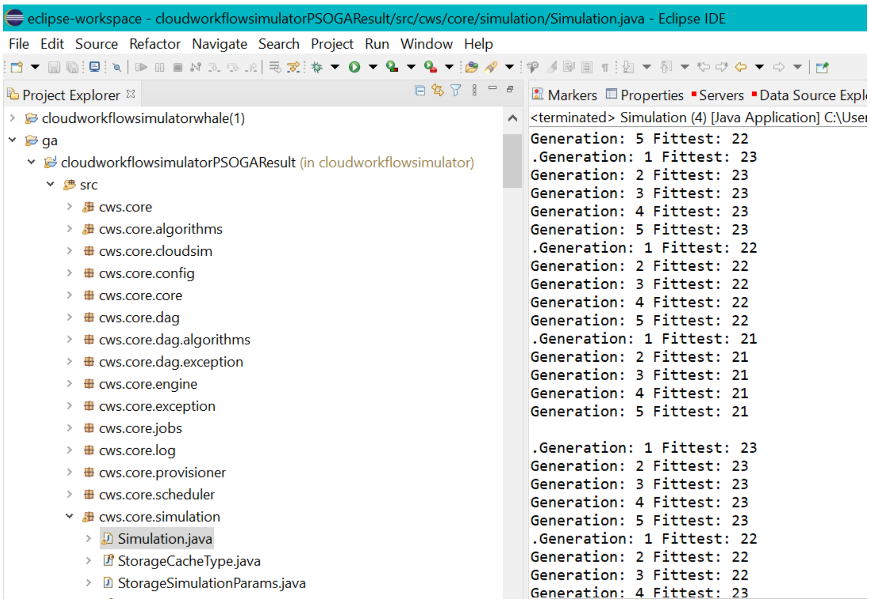
Distributed_HEFT Rank execution as below:

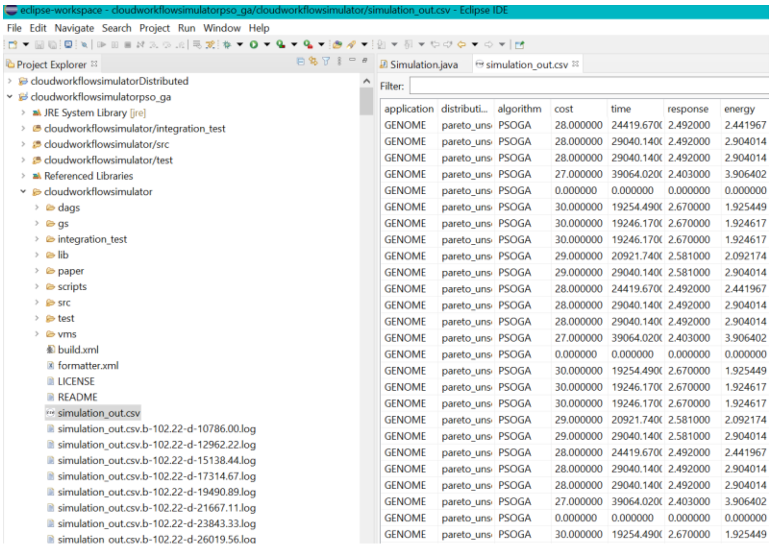


Output file:



Implementation of GA PSO





Implementation of Whale Optimization: