IMPROVED SCIENTIFIC WORKFLOWS SCHEDULING WITH RANKING HEURISTIC AND HYBRID OPTIMIZATION FOR VM MIGRATION IN CLOUD ENVIRONMENT

Jaspreet Kaur[1], Dr R K Bathla[2]

[1, 2] CSE - Desh Bhagat University, INDIA

Abstract- The mechanism of cloud computing represents the latest technique that is growing faster bit by bit considering its convincing security and components. Cloud computing provides a way to get the data-based information from any place and at any point in time. This feature makes it well-known in such a way that it reduces the burden of customers. The sort of computing provides services in terms of phase, programming, and the framework of the system. On account of these components, the size of data over the cloud is extended and it impacts the cloud profitability. To conquer the problem, task arrangement and planning on data represents the best option.. In addition, the research also impacts the reliability of the parameter on various workflow processes. An intelligent optimization relying on distinctive convex methodology is performed along with PEFT-based positioning. The experimental analysis, GWO represents outcomes in terms of time and cost overall the cloud workflows and provides noteworthy outcomes. In order to optimize the VM i.e. both locally and globally, a hybrid optimization is proposed. The algorithm based on PEFT methodology is utilized for its primary usage and operation as a heuristic algorithm. Such type of algorithm lessens the optimization-based random initialization error. The proposed Grey Wolf Optimization and Flower Pollination Algorithm demonstrate noteworthy active outcomes than the Genetic Algorithm with Flower Pollination Algorithms. Further, this methodology involves comparative analysis of bio-inspired intelligence and swarm intelligence. The analysis comprises five kinds of logical work processes such as LIGO, GENOME, GENOME, CYBER SHAKE, and SIPHT. These procedures of workflow vary depending upon complexity and number of tasks. The proposed method utilizes two methods of optimization, one is based on GWO and FPA intelligence, and the second one is Bio-inspired GA. The experimental set up utilizes 2 to 20 number of workflow and VMs is done and it further involves the analysis of time and cost. Based on comparative analysis, FPA with PEFT ranking represent high time and cost, whereas GWO with FPA decreases time and cost on a significant basis. Here, the hybrid optimization methodology represents a mixture of GWO and FPA algorithm. Further, this work comprises of two types of parameters which are measured for time and cost schedules. Here, FPA is utilized in light of the fact that it is the local process of optimization algorithm and GWO represents global optimization which globally optimizes VM. Eventually, for correlation, an algorithm based on FPA_GA is utilized that additionally represents an algorithm based on swarm intelligence. In the working process, when the cost is expanded, the quantity of VMs is additionally expanded for a couple of quantities of workflow and tasks at the time of computing. The waiting time also gets minimized along with the reduced cost. Along these lines, the choice of optimization assumes an indispensable role to adjust the computing time and the VM and it requires optimization at both local and global level. For local and global optimization ACO and PSO are utilized yet they increase the period of time as a result of two kinds of optimization techniques discussed above. In the last chapter reliability by hybrid metaheuristic approach show the same improvement pattern like time of computation and cost.

***Keywords**: Grey Wolf Optimization, Flower Pollination Algorithm, Infrastructure as a Service, Virtual machines.*

I. INTRODUCTION

Cloud Computing, as demonstrated by the NIST (National Institute of Standards and Technology), is a method under which a prevalent pool of resources or assets (networks, servers, services, applications, and storage) could be easily retrieved on demand and it can be released (Mell and Grance, 2011). There seems to be a lot of discussion in the industrial and academic world about its description, future, and context of cloud computing (Almezeini and Hafez, 2018; Stergiou *et al.*, 2018). The development of cloud computing has resulted in many benefits for the implementation of scientific workflows. Workflows are widely utilized application models for computer sciences. It defines a series of calculations that allow data analysis in a distributed and systematic manner and has also been effectively used to produce meaningful technological innovations in multiple computational fields (Liu and Qiu, 2016). Cloud Infrastructure as a Service (IaaS) provides a simple flexible, scalable, and accessible infrastructure for the implementation of these applications. IaaS retailers offer an opportunity to install workflows at a cheap price and no need to own any infrastructure besides leasing virtualized virtual machines (VMs) or computing resources. This enables workflows to still be easily deployed and packaged and, more notably, allows workflow systems to obtain a virtually infinite VM pool that can be charged on a pay-per-use basis and can be dynamically obtained and discharged (Chen and Zhang, 2008). Throughout this way, the use of the resource workflow can indeed be adapted over time depending on the present application requirements. Scheduling algorithms seem to be crucial to exploiting such advantages and, in general, to efficiently automate the execution of scientific workflows in distributed environments. Such algorithms have become an important element of workflow management practices and are responsible for organizing a given task in a series of computing resources while preserving data dependence. Decisions taken by scheduling algorithms have been usually defined by a number of user-defined QoS needs. Their achievement in meeting the above QoS requirements depends upon the effective use of certain underlying resources and, as a consequence, decision-makers have to be aware of the different issues that exist from the features inherent mostly in the cloud service model. First, especially in comparison to several other distributed systems, like grids, the cloud provides more knowledge of the quantity and type of resources utilized (Selvarani and Sadhasivam, 2010). This abundance and flexibility of resources create the need for such a resource provisioning method that works in tandem with the algorithm of scheduling; a heuristic approach that determines the number and the type of VMs to be used and when and how to lease and discharge them. At the end of the day, algorithms have to be aware of the uncertainties and versatile nature of cloud systems. For example, VM de-provisioning and provisioning delays are highly unpredictable and variable, and resource performance variations such as VM CPUs, storage systems, and network links are observed. This rationality makes it tough for algorithms to give precise decisions for scheduling. -As a consequence, this dissertation raises the importance of effectively IaaS-based large-scale scheduling scientific workflows in cloud computing. It explores novel approaches for the process of scheduling and providing resources that resolve critical issues stemming from the unique traits of clouds. These are made possible by the development of a comprehensive categorization and a detailed survey relying on state-of-the-art algorithms. Besides, sequences of algorithms are suggested.

1.2.1 Cloud Deployment Models

Cloud computing provides on-demand computing resources and services through the Internet, such as computing servers, storage servers, hosting, etc. Several different cloud deployment models exist. The deployment of these models depends on the need of the organization; its storage size and who controls the infrastructure and its storage (Wang *et al.*, 2010; Bojanova and Samba, 2011). Following is the most popular deployment models: -

Private Cloud: For a single organization, the cloud infrastructure is operating. The cloud might be activated by the company or a third party, and it may be stored on-premises or in a datacenter of a 3rd party. Private clouds are usually more flexible than that of other cloud types since one client company dedicates and manages them. Within a current on-site datacenter, several private clouds are configured (Wang *et al.*, 2010; Bojanova and Samba, 2011).

Public Cloud: It provides a full virtualized environment that relies on full bandwidth internet connectivity to transmit data and resource usage. It provides a multitenant feature to share resources and also isolate the users from each other (Wang *et al.*, 2010; Bojanova and Samba, 2011).

Hybrid Cloud: A cloud provider that is a mixture of 2 or more of the operation models already described (public, private, VPC, or community). A typical example is a private cloud connecting more than one third-party public cloud service providers with other services such as email — all combined via the usage of a standard interface with cloud management & automation. A cloud management framework or cloud brokering program is needed to handle several service providers.

Virtual Private Cloud (VPC): A type of the hybrid cloud where one user is assigned to a segmented compartment in an otherwise distributed cloud network. VPC offerings introduce some of a large public cloud provider's better profitability although with a bit more customization, security, as well as VM differentiation, storage, and networking. VPC variants involve VMs and client resources which are controlled or unmonitored.

Community Cloud: A cloud infrastructure that offers common preferences or issues for a group of consumers or organizations. The program is run by one or more organizations, a single contractor, or a mixture of both. Organizations use this cloud software exchange tasks, policy criteria, compliance specifications, and policies. Throughout the user company, at mutual group sites, at a vendor, or a mixture of these, cloud applications may be housed on-premises. This cloud-based term is used in marketing to clarify the service's target consumers, although the actual cloud could be a VPC, personal, or hybrid cloud model technically.

1.2.2 Cloud Service Models

There are three of the cloud computing-based service delivery models that include Software as a Service (SaaS) where the applications or programming are accessible in terms of service progressively to numerous associations and end-clients; Platform as a Service (PaaS) where the customers mainly designed applications on a platform provided by the cloud. Infrastructure as a Service (IaaS) release model gives storage, processing, as well as the implementation of the application as on rent to the user by the cloud service provider. It helps in providing virtual assets, for example, virtual machine (VM) servers (Amazon EC2), stockpiling frameworks (Amazon S3), bandwidth, switches, routers, network, and other associated instruments that are important to fabricate an application domain (Sakr *et al.,* 2011; Sareen, 2013). The basic cloud service models are shown in Figure 1.1. Each class has an alternate reason that offers various offices to organizations and people. These classifications are also called the models of cloud administrations.

Software as a Service (SaaS): SaaS users reserve the service use that operates within the provider's Cloud network, such as SalesForce. Framework implementations are usually delivered to consumers via the Web and are completely managed by the cloud provider (Sakr *et al.,* 2011). For example, the organization providing these facilities is the supplier's responsibility to update and repair them. One big benefit of SaaS is that all customers operate a common programming type but that new functionality could be readily implemented by the manufacturer and are therefore available to the people (Wang *et al.,* 2010).

II. RELATED WORK

2.1 MODELS

2.1.1 Application Model

The algorithms used in this study share much of the characteristics of an application layout. Nonetheless, they vary in their capacity to plan one or more workflows (Adhikari *et al.*, 2019; Rodriguez and Buyya, 2017).

Workflows Multiplicity: Algorithms may be configured to plan a single workflow example, several examples of the same workflow, or several workflows. It defines three forms of planning procedure from a multiplicity viewpoint of the workflow.

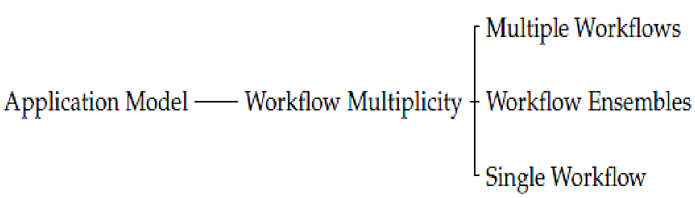

Figure 2.1. Application model

Single Workflow

Under this class, algorithms are programmed to automate a single workflow process.

• It is the standard model being used clusters as well as networks, as well as being the most popular configuration of cloud computing.

• It means that the planner handles process operation consecutively as well as autonomously.

• It focuses on cost optimization while satisfying a single consumer as well as single DAG QoS criteria.

Workflow Ensembles

Most research implementations consist of more than one instance of the workflow. Such interconnected workflows are called ensembles but are grouped together so a desirable performance is generated by their combined execution (Adhikari *et al.*, 2019; Rodriguez and Buyya, 2017).

- Scheduling algorithms in this group focuses on the implementation of ensemble workflow utilizing the available resources.
- Regulations will be mindful that the QoS specifications are meant for many workflows.
- The number of occasions is usually calculated in advance and should also be considered by the scheduling technique while organizing job execution.

Multiple Workflows

This group is close to the one collection of workflows but varies from it in that its planned workflows are not inherently linked to each other and the variations in form, scale, input details, and operations. (Adhikari et al., 2019; Xie *et al.,* 2017).

- The amount and form of workflows are not specified in progress as well as the development is thus treated as a complex mechanism wherein the schedule is constantly changing as well as workflows of different configurations are increased for implementation.
- Every workflow does have its individual QoS specifications.
- A variety of reports have shown the scheduling techniques for cloud system workflows as well as evolving patterns.

Adhikari *et al.* analyzed as well as categorized the models of different workflow scheduling approaches depending on their aims and implementation model, as well as addresses workflow scheduling in the light of these new cloud computing patterns (Adhikari *et al.*, 2019).

Rodriguez and Buyya discussed and also analyzed the current algorithms in terms of the scheduling frameworks they follow, and also the resource as well as the implementation model they find. A comprehensive taxonomy focused on cloud-specific functionality and the algorithms evaluated are categorized accordingly (Rodriguez and Buyya, 2017). Wang *et al.*

proposed four heuristic workflow task scheduling algorithms for WFaaS architecture. The researcher analyzed the algorithm usage in terms of cost as well as price/performance ratio differences through scientific investigations (Wang *et al.*, 2014).

2.1.2 Resource Model

Taxonomy is provided in this segment based on the calculations and predictions produced by algorithms centered on the resource model. This model is concerned with the number of VMs provided by IaaS, its pricing schemes; data transfer between VMs. Figure 2.2 demonstrates the features of the resource model.

➢ **VM Leasing Model**

Providers sell to a specific customer either a restricted or an infinite amount of VMs eligible for leasing. This comprises primarily of two types:

- **Limited:** Such algorithms presume that the suppliers have a limit on the amount of VMs that a customer can lease. In this way, the major issue of supplying resources somehow is simplified and seems to be comparable to scheduling with such restricted multiple processors.
- **Unlimited:** Algorithms believe they have links to a range of nearly infinite VMs. The amount of VMs the seller will lease is not limited.

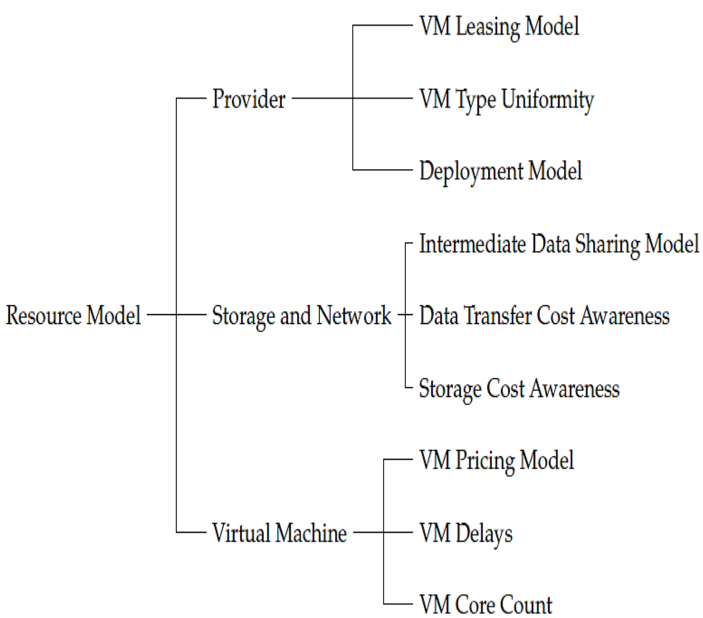

Figure 2.2. Resource Model

➢ **VM Type Uniformity**

Users can lease a single form of VMs or can use diverse VMs with specific configurations depending on their goals for scheduling (Xu *et al.*, 2016; Xu *et al.*, 2017). This consists important for two main reasons:

- **Single VM Type:** This is used to automate the scheduling cycle as well as the choice about which type on VM of using is taken without taking into account the workflow as well as features of the tasks. It causes negative effects on the algorithm's outputs as well as fails to fully exploit the varied nature of cloud resources.
- **Various VM Types:** It helps algorithms to use specific VM configurations or to plan programs with their specifications and features effectively.

Xu *et al.* suggested a comprehensive list of evaluating load balancing algorithms for VM positioning in data centers, as well as the algorithms evaluated are categorized by type.

➢ **Deployment Model:** Another manner in which algorithms are categorized is dependent on the amount of data centers as well as public cloud services they lease space. This generally consists of two forms following with two subgroups:

- **Single Provider:** It provides pay-per-use services and the expense of input and output data sets are assumed to be fixed depending on the planned workflow.
- **Different Providers:** This delivery model helps algorithms to assign activities on multiple cloud provider-owned infrastructures. A growing company has its individual product ranges, SLAs as well as pricing strategies. It is up to the scheduler to pick the one that better fits.
- **Single Data Center:** This delivery model is suitable for certain implementation situations because the amount of VMs needed to conduct the workflow is unlikely to surpass the capability of the data centers.
- **Multiple Data Centers:** For applications with globally diverse input data, this alternative is more appropriate. VMs could be defined in dissimilar data centers to reduce data relocate times depending on the position of the data.

➢ **Intermediate Data Sharing Model**

An ordinary method is to believe a peer-to-peer (P2P) model whereas another method is to utilize a shared global storage system as a repository for files. P2P algorithms presume that files are passed directly through the VM running the parent assigned to the VM running the child assignment. This ensures that activities are transmitted synchronously and therefore VMs should be kept going before all the descendant activities have provided the necessary data. As the lease time for VMs is extensive, this can lead to higher costs.

➢ **Data Transfer Cost Awareness**

IaaS companies provide varying payment structures for multiple forms of data transactions, based on how the data is transmitted within, inside, or beyond their facilities. Transfer inbound data is usually free along with all the analyzed algorithms, therefore, neglect this factor. Conversely, it is typically difficult to migrate data from the cloud service. Many other suppliers, including such Amazon S3, Google Cloud Storage as well as Rack Space Block Storage, may not charge for data relocate inside as well as outside the storage system as well as then this value could, therefore, be unnoticed by algorithms that utilize these amenities. Waibel *et al*. devise a device model utilizing several cloud-based platforms to achieve a robust and cost-effective storage operation. They formulate a national as well as global optimization difficulty within this system model, which finds historical access control information as well as predefined service quality standards to choose a cost-effective storage solution (Waibel et al., 2017).

➢ **Storage Cost Awareness**

Data storage is compensated according to the volume of data processed. Any companies are paid extra fees depending on the amount and form of storage facility operations (i.e., Receive, Place, Delete)[122]. This cost is only appropriate if cloud storage systems are being used is ignored in many models but in such cases, mostly due to the amount of data used as well as produced by workflow is constant as well as independent of the task development.

➢ **VM Pricing Model**

It defines four specific pricing models that are important and identified by the surveyed algorithms: static, interactive, time unit, including subscription dependent.

- The static price model is the standard cloud price model but is offered by many other givers. Google Compute Engine [123] is an instance of cloud examples under this pricing model.
- The competitive model of pricing includes consumers who purchase dynamically priced VMs by exchanges or agreements. Consumers request a VM in these auctions by exposing the most money they are ready to give for it, after which suppliers start deciding to admit or decline the request depending on the present market conditions.

Amazon EC2 Spot Instances are an illustration of VMs adopting the pricing pattern. Xu and Li performed an observational analysis of Amazon's spot price past, as well as consider that, unexpectedly, it is not possible that the spot price would be determined by consumer demand (Xu and Li, 2013). Javed et al. analyzed the suitability of competitive pricing for the cloud world as well as the number of clients and its suppliers. To do so, it evaluated many parameters such as the machine provided price or amount of active transactions (Javed *et al.,* 2013). It assumed that VMs are charged per unit of time in time-unit. Under this model, no resource wastage or additional costs occur in billing periods due to unused time units. The planning is also streamlined because no requirements utilize empty time slots of rented VMs because the expense of utilizing the machines is the same time at which they are required. Consequently, as pointed out by Arabnejad *et al.,* here the potential of new pricing models being introduced by givers or arising from established ones, for example, a community of users could rent a collection of VMs on a subscription-based basis, distribute it or price its usage on a time unit basis.

In the case of delivery, examples are set for extended spans of time, typically weekly or annually. Payment is usually provided in advance and is considerably cheaper as opposed to static prices. This pricing model for the cloud workflow scheduling issue means implementations have to utilize a finite number of VMs with corrected configurations to perform the tasks (Arabnejad *et al.,* 2019).

➢ **VM Delay**

Algorithms have to recognize VM provisioning delay while creating runtime calculations to make precise scheduling decisions. Sharma et al. developed a research project which provides an efficient lightweight mechanism for real-time service latency prophecy for optimal virtual machine allotment of resources in delay-sensitive cloud services (Sharma et al., 2015). The plan is to provide real- and precise-time delay & cloud resource situations through predicting latency in a short time. In the case of VM De-provisioning delay, the effect of delays in supplying VM is severely restricted to the cost of delivery.

➢ **VM Core Count**

It applies to how algorithms are conscious of multi-core VMs with various, overlapping activities to be performed on them. David has been classified as developing scheduling algorithms for asymmetric multicore processors. Several symbolic algorithms of such groups to provide a summary of asymmetric multicore machine scheduling algorithms (David, 2017). Xi *et al* have seen two big trends in complicated real-time systems growth. Firstly, several systems share computing platforms through virtualization technologies, rather than being installed independently on physically separated servers, to minimize costs and improve versatility. Second, multicore processors are becoming more and more used in real-time systems. Integrating real-time systems as virtual machines (VMs) at the top of traditional multicore architectures faces major new research difficulties in fulfilling multiple systems' real-time efficiency requirements (Xi et al., 2014). Sung et al. proposed a VM pre-provisioning scheme which reduces the delay by pre-procurement. The researchers gather a large-scale measurement trace of global users as well as incorporate them with Google's existing cloud track (Sung *et al.*, 2019). The extensive assessment indicates that the system introduced outperforms current systems in edge nodes.

III. THE PROPOSED METHOD

HGSA Algorithm [34]

HGSA algorithm is a meta-heuristic optimization algorithm which is used to get the optimal solution. It is basically based on the echolocation behaviour of the HGSAs with varying pulses rates of emissions and loudness. The working of this algorithm is depending on the velocity and position of HGSA which vary according to the frequency, wavelength and loudness. Following are the steps that are performed in the HGSA algorithms.

Initialization: Firstly generation counter t is set to be 1, p is the population of NP HGSAs which is initialize randomly. Each HGSA gives a potential solution of the given problem.

Here,

A: it defines the loudness

Q: is the frequency

V: are the initial velocities.

s: is the pulse rate

F: is the weight factor.

Step 1: Evaluate the quality f for each HGSA in P determined by f(x).

Step 2: while the termination criteria are not satisfied or t <MaxGeneration **do**

Sort the population of HGSAs P from the best to worst by order of quality f for each HGSA;

for i = 1: NP (all HGSAs) **do**

Select uniform randomly $s_1 \neq s_2 \neq s_3 \neq i$

r_4 = [NP * rand]

$v_i^t = v_i^{t-1} + (v_i^t - x_*) * Q$

$x_i^t = x_i^{t-1} + v_i^t$

if (rand > r**) then**

$x_u^t = x_* + \alpha \mathcal{E}^t$

else

$x_u^t = x_{r1}^t + F (x_{r2}^t - x_{r3}^t)$

end if

Evaluate the fitness for the offspring x_u^t , x_i^t ,x_{r4}^t

Select the offspring x_k^t with the best fitness among the off springs

x_u^t , x_i^t ,x_{r4}^t

if (rand < A) **then**

$x_{r4}^t = x_k^t$;

enf if

end for i

t = t+1 ;

Step 3:end while

Step 4: Post- processing the results and visualization;

End.

IV. RESULT ANALYSIS

4.1 Result Analysis

4.2.1 Result of HGSA and PSO_WCA Using SIPHT

Figures 4.1 - 4.2 show the behavior of SIPHT workflows in different number of workflows and Virtual machines which represent by ensemble size. In results, show the HGSA and hybridization of particle swarm optimization and WCA on total execution time, total execution cost and time delay.

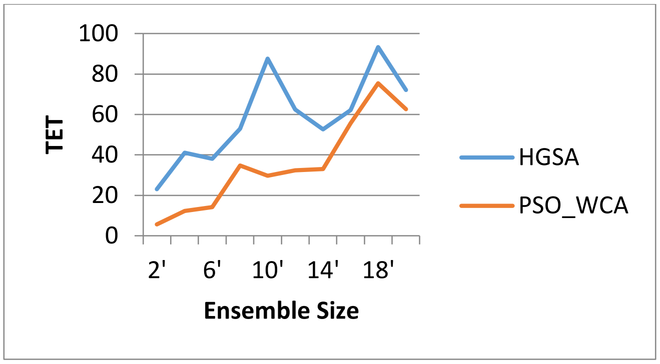

Figure 4.1 Comparison graphs of TET of HGSA and PSO_WCA using SIPHT

In figure 4.1 and 4.2 Line graph analysis The parameters of the PSO WCA total output and the expense performance are very good in cost and time because of the PSO WCA testing time, one by PSO, decides on two occasions when optimising or otherwise optimising WCA decision and the transfer of VM task depends on a Transient issue.

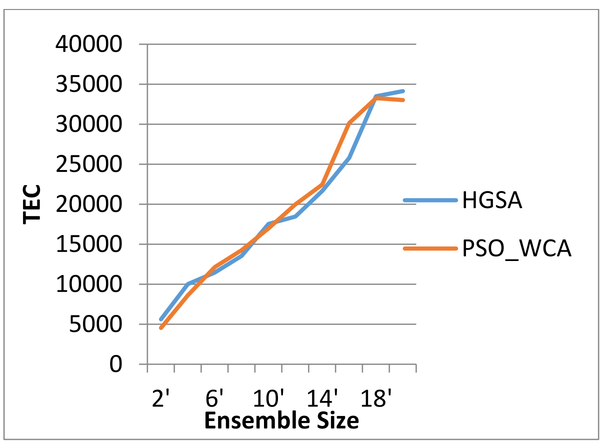

Figure 4.2 Comparison graph of TEC of HGSA and PSO_WCA using SIPHT

4.2.2 Results of HGSA and PSO_WCA Using MONTAGE

In Figure 4.3 show the behavior of MONTAGE workflows in different number of workflows and Virtual machines which represent by ensemble size. In results, show the HGSA and PSO_WCA on total execution time, total execution cost and time delay.

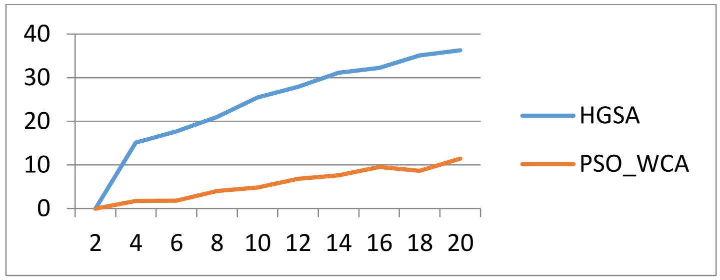

Figure 4.3 Comparison graph of TET of HGSA and PSO_WCA using MONTAGE

In figure 4.3 and 4.4 MONTAGE workflow analysis on different number of virtual machine. In analysis use two metrics first TET in figure 4.3 and TEC in figure 4.4. These analyses on HGSA and hybrid of two optimization PSO_WCA.

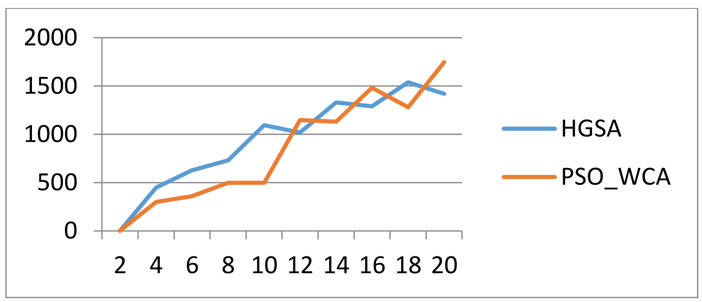

Figure 4.4 Comparison graph of TEC of HGSA and PSO_WCA using MONTAGEIn figure 4.4 Analysis The PSO WCA parameter performs well in both costs and periods, since the time of colonial searching ant is determined by adaptive pheromones and VM tasks migration, differs from the Transient problem but both in the genetic algorithm depends on the candidate solution. However, because of parsing the distribution it takes more time to map VM by work, the delay in genetic algorithm is better than PSO WCA.

4.3.3. Result of HGSA and PSO_WCA Using CYBERSHAKE

Figure 4.5 & 4.6 Should be present in various workflows and Virtual engines reflecting Ensemble Size, the conduct of

CYBERSHAKE and LIGO Workflows The findings demonstrate the optimisation of the Ant colony and genetic algorithms for overarching time , average cost and time delay.

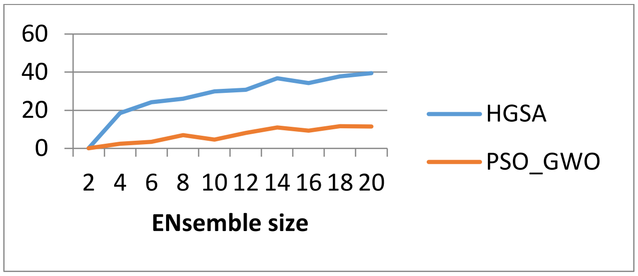

Figure 4.5 Comparison graph of TET of HGSA and PSO_WCA using CYBERSHAKE

In figure 4.5 The analyses of these PSO WCAs are well carried out in the cost and time parameter since the search time of particle swarm optimization depends on the Transient problem, and the migration of the VM mission is contingent upon the Transient problem but both in genetic algorithms focus on candidate solution.

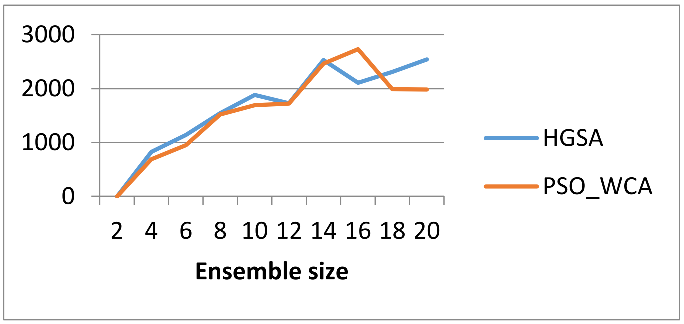

Figure 4.6 Comparison graph of TEC of HGSA and PSO_WCA using CYBERSHAKE

In figure 4.6 times delay of HGSA is better than ACO because of pare to distribution take more time for mapping of VM by task. It will effect on Total cost execution because pare to VM mapping but TET always significance improve.

TET LIGO and TEC in numerous devices and ensembles. LIGO in different sizes. In this text we use two to 20 sets and maximise the scale by genetic algorithm and the optimisation of the colony of ants. PSO WCA decreases the average TET and TEC of various workflows of experimental performance. We inferred that PSO WCA optimises and converges cloud workflow preparation.

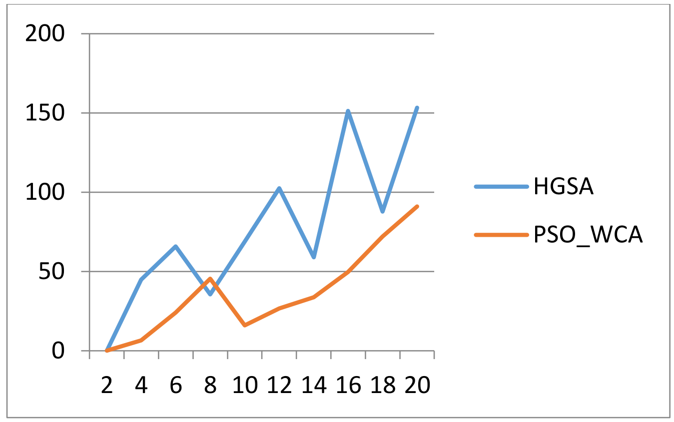

Figure 4.7 Comparison graph of Time delay of HGSA and PSO_WCA using LIGO

In figure 4.7 and 4.8 result analysis, Find out that PSO WCA time delay is more than in local simulation compared with HGSA. This helps you to run in real-time cloud environments with SLA if you want PSO WCA to reduce the delay in time. In order to find a solution for load balance and mission delays, this work can be generalised using the multiple goals algorithm.

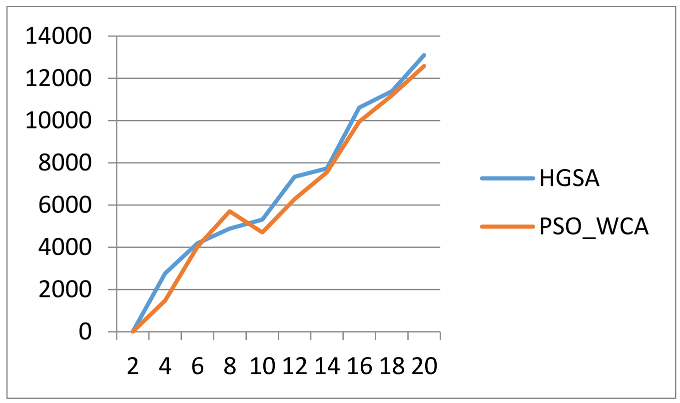

Figure 4.8 Comparison graph of Time delay of HGSA and PSO_WCA using LIGO

4.2.3 Result of HGSA and PSO_WCA Using Genome

Figure 4.9 Display GENOME activity in multiple workflows and Virtual machines reflecting an ensemble number. The findings demonstrate the optimization of the Ant colony and genetic algorithms for overarching time , average cost and time delay.

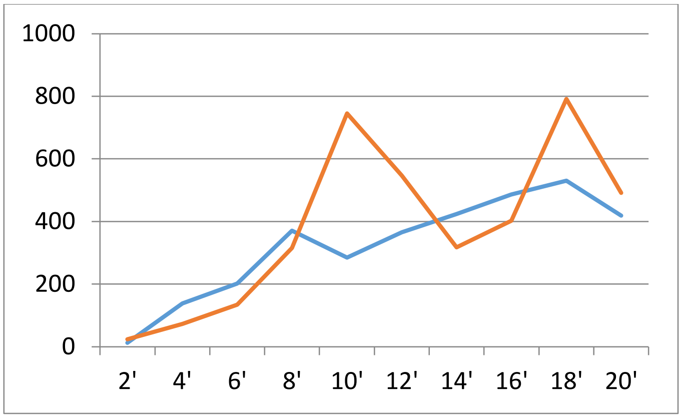

Figure 4.9 Comparison graph of TET of HGSA and PSO_WCA using GENOME

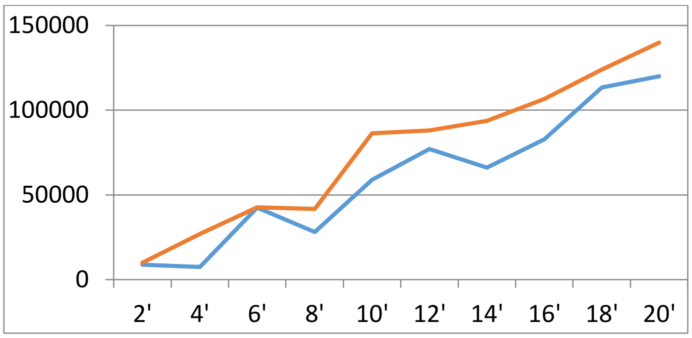

Figure 4.10 Comparison graph of TEC of HGSA and PSO_WCA using GENOME

In figure 4.9 and 4.10 Analysis These parameters of PSO WCA work well with the cost and time parameters because ant colony search time determining by adaptive pheromones and the VM mission migrations depends on the transient problem but all depend on candidate solution in genetic algorithms. However, since the par to distribution takes more time to map VM through tasking, HGSA's time delay is no better than PSO WCA. Total cost efficiency is influenced by VM-mapping, but TET is still essential.

IV CONCLUSION

We proposed in this work the planning framework for carrying out fair IaaS mists systems. The central problem in distributed computing, though at the same time decreasing, is the cost of execution. The use of the Hybrid PSO with WCA illuminates this dilemma. The experiments were orchestrated on Cloudsim by impersonating four unquestionably understood job types (Cybershake, Ligo, Genome, Montage), which indicates that our response has a more profitable general execution than other current calculations. A related investigation of TET and TEC parameters focused on Bio Motivated Streamlining (HGSA) and Particle Swarm Advancement (PSO) with Gray Wolf Optimization was discussed in the diagrams and tables described above. In the experiment, with the use of different types of conceptual work processes, we used work process preparation for cloud conditions. In our analysis, absolute cost and execution time are increased by streamlining even more depending on instant factors for improvement. We use Pareto appropriation in the suggested technique rather than discretionary instatement.

If unusual conveyances are used, it will take extra time to assemble and at some stage perform the assembly by emphasis, but retaining intermingling will establish the measurement and execution time along these lines that does not conform with the condition of time constraint. So, as characterised in this article, task presentation is an essential task. In these diagrams and tables, something else was listed is that PSO-WCA performs better in comparison with HGSA for reducing cost and time in view of the irregular hybrid. In view of the fact that PSO (molecule swarm improvement) implies substantial function in worldwide development and WCA update locally, the praiseworthy results are obtained and we have combined the two equations by extracting the best from them. In most of the job types, we can express a decreased cost-productive calendar with the suggested approach at that stage, even reducing the time wait.

V REFERENCES

[1] Ibtihal, M., & Hassan, N. "Homomorphic encryption as a service for outsourced images in mobile cloud computing environment." *Cryptography: Breakthroughs in Research and Practice*. IGI Global, 2020, pp. 316-330.

[2] Soe, T., Mon, S. S., & Thu, K. A. (2019). "Performance Analysis of Data Encryption Standard (DES)."

[3] Neshir, G., Rauber, A., & Atnafu, S. (2019). "Negation handling for Amharic sentiment classification." Available: http://ceur-ws.org/Vol-2540/FAIR2019_paper_6.pdf

[4] Ramachandran D, Parvathi R. "Enhanced Event Detection in Twitter Through Feature Analysis." *International Journal of Information Technology and Web Engineering (IJITWE)*, vol.14, no. 3, pp.1-15, Jul 2019.

[5] Back, B. H., & Ha, I. K. "Comparison of Sentiment Analysis from Large Twitter Datasets by Naïve Bayes and Natural Language Processing Methods." *Journal of information and communication convergence engineering*, vol. 17, no. 4, pp. 239-245, Dec 2019.

[6] Arunarani, A. R., D. Manjula, and Vijayan Sugumaran. "Task scheduling techniques in cloud computing: A literature survey." *Future Generation Computer Systems* 91, pp. 407-415, Feb 2019.

[7] Data Encryption Standard - Wikipedia. "Data Encryption Standard.", 2020. [Online]. Available at: https://en.wikipedia.org/wiki/Data_Encryption_Standard

[8] Alharbi, A. N., & Alnnamlah, H. " Classification of Customer Tweets Using Big Data Analytics." In *5th International Symposium on Data Mining Applications*, Springer, Cham, 2018, pp. 169-180.

[9] Koo, D., & Hur, J. "Privacy-preserving deduplication of encrypted data with dynamic ownership management in fog computing." *Future Generation Computer Systems*, vol. *78*, 2018, pp.739-752.

[10] Pai, S., Arathi, P. "A Survey on Various Encryption Algorithms. *International Journal of Computer Science and Technology*, vol. 9, no. 3, Jul 2018.

[11] Kathuria, A., & Sharma, A. "A Comparative Study of Various Approaches for Sentiment Analysis of Twitter Data.", 2018.

[12] Nayak, S. C., & Tripathy, C. "Deadline sensitive lease scheduling in cloud computing environment using AHP. " *Journal of King Saud University-Computer and Information Sciences*, vol. 30, no.2, pp. 152-163, Apr 2018.

[13] Kunal, S., Saha, A., Varma, A., & Tiwari, V. "Textual dissection of live Twitter reviews using naive Bayes." *Procedia computer science*, vol.132, 2018, pp.307-313.

[14] Cirqueira, D., Pinheiro, M. F., Jacob, A., Lobato, F., & Santana, Á. "A literature review in preprocessing for sentiment analysis for Brazilian Portuguese social media." In *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, 2018, pp. 746-749.

[15] Almezeini, N., & Hafez, A. "Review on scheduling in cloud computing. " *INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND NETWORK SECURITY*, vol. *18*, no. 2, pp. 108-111, Jan 2018.

[16] Stergiou, C., Psannis, K. E., Kim, B. G., & Gupta, B. "Secure integration of IoT and cloud computing." *Future Generation Computer Systems*, vol.*78*, 2018, pp. 964-975.

[17] Ghiassi, M., & Lee, S. "A domain transferable lexicon set for Twitter sentiment analysis using a supervised machine learning approach." *Expert Systems with Applications*, vol. 106, 2018, pp.197-216.

[18] Wu, T., Wen, S., Xiang, Y., & Zhou, W. (2018). "Twitter spam detection: Survey of new approaches and comparative study". *Computers & Security*, vol. 76, 2018, pp.265-284.