Fractional Knapsack Based Optimal Resource Allocation For Dynamic Cloud Environments

MSc Research Project
Masters in Cloud computing

Vyom Dubey
Student ID: 23185732

School of Computing
National College of Ireland

Supervisor: Sudarshan Deshmukh

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Vyom Dubey
Student ID:	23185732
Programme:	Masters in Cloud computing
Year:	2024
Module:	MSc Research Project
Supervisor:	Sudarshan Deshmukh
Submission Due Date:	12/12/2024
Project Title:	Fractional Knapsack Based Optimal Resource Allocation For Dynamic Cloud Environments
Word Count:	6956
Page Count:	19

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	12th December 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Fractional Knapsack Based Optimal Resource Allocation For Dynamic Cloud Environments

Vyom Dubey
23185732

Abstract

With the growing popularity of cloud applications and cloud services, the cost of cloud resources are increasing as well. It is due to the drastic increase in the use of cloud resources by companies, individuals and governments. This leads to inefficient allocation of cloud resources which result in computation power wastage and increased latency. Traditional resource allocation algorithms are more focused towards minimizing time for the allocation of resources rather than allocating the best resource for the available virtual machine which increase inefficiency and may lead to under-utilization or over-utilization of resources. The proposed approach will reduce the inefficiency in the allocation of resources and provide the best optimal allocation solution thus benefitting both the cloud service providers as well as the consumers. The proposed algorithm focuses on allocating the best resource to the virtual machine by finding the best host for a virtual machine and also tune according to the dynamic cloud environment requirements in real-time. Using this approach the cost of cloud resources will be reduced which is beneficiary for both cloud service providers as well as consumers.

1 Introduction

Cloud resources have became an integral part of today's fast moving modernized world. With the increasing number of applications and services being hosted on the cloud environment, the allocation of cloud resources is overwhelmed and overburdened. The traditional resources allocation algorithms are not quite efficient and tuned to such surge in the demand of cloud resources. Using these traditional allocation techniques result in inefficient utilization of resources and increased cost of cloud resources. According to a survey the average amount of resources being utilized at a cloud data center is 30% while the other 70% cloud resources remain idle. Singh et al. (2023) This huge amount of cloud resources being wasted and unutilized results in increased cost of resources. Such poor utilization of cloud resources drastically impact the cloud service providers as well as the consumers. There is need for an efficient allocation approach of cloud resources to reduce the cost of resources which is beneficial for both the cloud service platforms (AWS, Microsoft Azure, GCP) as well as the consumers of cloud resources.
The cloud data centers distribute the cloud resources among different virtual machines. The allocation of the cloud resources among virtual machines is done using scheduling

algorithms which are responsible for allocation resources top the tasks submitted by users. Walia and Kaur (2021) Traditional scheduling algorithms such as Round Robin, Ant Colony and Bin packing algorithms are not effective and efficient under dynamic heterogeneous cloud environments. Reddy et al. (2024) It is evident from the problem statement presented above that there is a need of an optimized efficient resources allocation algorithm which can provide optimal allocation of cloud resources under dynamic heterogeneous cloud environment. With optimal allocation of cloud resources, the cost of resources will be decreased drastically benefiting the cloud service providers and the clients/users.

Research Question. The proposed research statement motivates the following research questions.

RQ: *How well can the fractional knapsack scheduling algorithm using combinations of resource metrics can increase the cloud resources utilization and optimize the performance of allocation of cloud resources?*

Sub-RQ: *How well can the proposed fractional knapsack scheduling algorithm can improve the performance of allocation of cloud resources using priority of metrics under different dynamic scenarios and use cases?*

Sub-RQ: *To what extent can the proposed algorithm optimize the allocation of cloud resources in comparison with other known scheduling algorithms benefiting cloud service platforms and consumers?*

Taking into the account the research questions above, we have proposed the fractional knapsack scheduling algorithm which emphasis on increasing the resource utilization and provide optimal allocation of cloud resources. Also the proposed solution prioritize metrics based on different scenario (cpu-intensive tasks or storage-intensive tasks). The efficiency of the proposed solution is also compared and evaluated against other widely used scheduling algorithms for allocation of cloud resources.

Table 1: Objectives

Objectives	**Explanation**
Related Work Critical Review	Critically evaluating the gaps in the field and investigating the related work (2020-2024).
Setup Workspace	Integrating the workspace consisting of eclipse, cloudsim, and Microsoft Azure tools.
Fractional-Knapsack Scheduling Algorithm	Developing a virtual machine allocation algorithm using fractional knapsack technique.
Priority based allocation	Implementing a scheduling algorithm with dynamic priorities of metrics based on different scenario.
Comparison between Scheduling Algorithms	Comparing results between proposed and other widely used scheduling algorithms.

Table 1 discusses the objectives of the proposed research solution. The main objective of the proposed research solution is solving the problem of under-utilization or over-utilization of cloud resources by introducing a scheduling algorithm which provides optimal allocation of cloud resources. The traditional scheduling algorithms are not configured to handle modern era enormous traffic resulting in wastage of cloud resources. The proposed solution is aimed towards reducing the cloud resources wastage and increasing the utilization of resources which benefits both the cloud service platforms and the users. The proposed solution also fulfills the requirement of dynamic cloud environments by changing the priority of metrics of resources depending on the scenario or the use case. Also, the objective of the proposed solution is to compare the efficiency of the novel fractional knapsack resource scheduling algorithm to other scheduling algorithms to determine the effectiveness of the proposed algorithm under similar workloads.

The main contribution of the proposed research solution is the introduction of a novel resource scheduling algorithm which focuses on increasing the utilization of cloud resources and providing an optimal cloud resource allocation technique. This research solution also contribute towards developing a dynamic scheduling algorithm, providing an optimal allocation of resources under different workloads by prioritizing metrics according to respective scenario. Other contribution includes comparing and evaluating the performance and effectiveness of the proposed novel algorithm to other resource scheduling algorithms.

The research proposal follows the following structure. Next section (Section 2) proposes the related work investigation (2020 - 2024) with respect to the proposed research question. Section 3 describes the research tools, proposed solution approach an the methodology of the proposed solution. Section 4 consists of the design specification and presents the proposed Fractional-Knapsack resource scheduling algorithm. Section 5 narrates the proposed solution implementation. Moving forward, the Section 6 demonstrates the experiments conducted, discussion of the results and the evaluation of the proposed solution. Section 7 describes the conclusion of the proposed novel approach along with the future work.

2 Related Work

2.1 Introduction

This section outlines the previous related work done related to the optimization of allocation of cloud resources using novel approaches and methods. In order to develop an optimal scheduling algorithm we need to understand the gaps in the work done previously by evaluating the importance of scheduling algorithms in cloud computing, importance of optimal allocation of cloud resources using scheduling algorithms, impact of multi-tenant scheduling algorithms on the allocation of cloud resources and the impact of priority based scheduling algorithms in dynamic cloud environments.

2.2 Importance of scheduling algorithms in cloud computing

The research paper emphasizes the role of scheduling algorithm in the working of the cloud computing. The author differentiates between different scheduling algorithms available along with their advantages and disadvantages. This study also concluded that the scheduling algorithm will always remain a challenging part of cloud computing, specially when subjected to dynamic cloud environments. The author has done great work in emphasizing the impact of scheduling algorithms in cloud computing but failed to provide a practical evidence to support the argument. Wang and Wang (2022)

The author discusses the importance of scheduling algorithms in the efficient working of cloud environment. The author states that the scheduling algorithms are essential to prevent the underuse or overuse of the cloud resources to achieve optimal utilization of resources, improve quality of service (QoS) and to reduce makespan (total execution time to complete tasks). The author discusses the working of scheduling algorithms along with experiments conducted to provide evidence to the results but fails to compare different type of algorithms extensively and only compared the algorithms with same ideology. This prevents fairness in the evaluation and introduces limitations and more experiments using other types of scheduling algorithms are needed. George et al. (2021)

The paper highlights various benefits of efficient scheduling algorithm such as maximum resource utilization, improved scalability to different workloads and reduced makespan (total execution time). The author discusses the vital role of scheduling for efficient resource optimization, reduced cost efficiency and increased quality service but fails to conduct any experiment to establish the argument.Manchanda et al. (2023)

The author discusses the crucial role of scheduling in cloud computing which directly impacts the resource utilization, cost of resources and the total time taken to complete the tasks. The paper experiments various metaheuristic techniques with different hybrid algorithms. The author concludes that the combination of different algorithms and techniques can perform better than the traditional scheduling algorithms. The approach to combine more than one algorithm for scheduling seems to be a great approach whereas the overuse of complex metaheuristic algorithms can increase the complexity and the computational overhead of the scheduling algorithm resulting in increase in latency of cloud resources. Khare et al. (2023)

2.3 Impact of scheduling algorithms on the utilization of cloud resources

The author compares two approaches namely modified particle swamp optimization algorithm and genetic algorithm on the basis of energy consumption and resources utilization. The modified particle swamp optimization performs better than genetic algorithm by 48.39% in resources utilization resulting in much less energy consumption as well and reducing the cost incurred. The author establishes the fact that the cost of resources is directly related to the utilization of resources whereas used only complex algorithms which might induce computational overhead resulting in inefficient allocation of cloud resources. Supreeth et al. (2022)

The research paper discusses the essence of resource allocation and task scheduling being directly connected to the optimal optimization of resources and reduction in the cost of cloud resources. The author compares Deep Q-network(DQN) and Dynamic Heuristic Johnson Sequencing(DHJS) algorithms along with proposed scheduling algorithms. The experiments concluded that the proposed algorithm which is better than other compare algorithms in term of resource utilization and task scheduling excel other algorithms in term of performance and energy efficiency as well. Hemanth et al. (2024)

Presents a powerful reinforcement learning algorithm Deep Deterministic Policy Gradient(DDPG) aimed towards improving scheduling of resources and portrays the direct relationship between scheduling resources and the utilization of resources resulting in increase in the cost efficiency of cloud resources. Author compared the results with the traditional method of scheduling of resources but fails to specify the method used for comparison which questions the integrity of the experiments conducted. The experiments conducted are difficult to replicate and appeal to the authenticity of the research. Singh Panesar and Jain (2023)

The author establishes the connection between the energy consumption, cost of cloud resources and the utilization of resources. The author proposes minimum migration time (MMT) approach to increase the resources utilization and decrease the ideal time of resources to decrease the energy consumption and the cost of cloud resources. Though the research demonstrated useful correlations and outcomes but the experiments where not done on real-time data and the results might not be significant in real world applications. Bhandary et al. (2021)

The papers emphasizes on the cost per unit of cloud resources, the importance of job scheduling and the utilization of resources with the help of different metaheuristic approaches.The author compares different scheduling algorithm approaches but fails to provide evidence to the findings of the research paper using experiments. The research does not contribute much to the field due to the inability to provide evidence for the findings. Reddy and Reddy (2021)

In this research author compares various scheduling approaches in terms of makespan, average resource utilization, throughput and energy consumption. The author establishes that the cloud data centers can benefit by using optimized scheduling approach, smaller makespan and efficient resource utilization. The study also conducts experiments between different scheduling approaches and found that PSSLB and TASA have achieved highest resource utilization and minimum makespan and minimum energy cost. This study establishes the clear direct correlation between optimized scheduling algorithms and utilization of cloud resources resulting in minimum cost of resources. The study is important to the field of cloud computing due to the extensive experiments conducted and important findings and can be very beneficial in developing an optimized scheduling approach. Ibrahim et al. (2020)

2.4 Impact of priority based scheduling algorithms in dynamic cloud environments

The research proposes a dynamic priority based cloud task scheduling algorithm. Calculations are evaluated to establish the significance of priority based approach in resource scheduling but the author fails to conduct experiments providing results with the proposed approach and also no comparison of results with other scheduling algorithms is done. The calculations and evaluations of the proposed approach is important to the field but lacks support due to absence of real-time experiments. Bo et al. (2022)

The author proposed a dynamic multi-level task scheduling (DMLTS) priority based algorithm for allocation of cloud resources.The proposed algorithm is compared with other scheduling algorithms such as Multi-objective Optimization and Cat Swamp Optimization using experiments which presents that DMLTS is better in performance than other scheduling algorithms with better resource utilization. Karthik et al. (2023)

The proposal uses a M/M/n queuing model and a priority assignment algorithm for allocation resources to the virtual machines. The author evaluates the working of the proposed solution along with other known scheduling algorithm such as BATS, IDEA and BATS+BAR to determine the performance and efficiency of the proposed solution. Also three different scenarios with different priorities are demonstrated to understand better the use case of the priority based algorithm. The author evaluates the proposed algorithms and using experiments found that the proposed solution outperforms other known scheduling algorithms in terms of resource utilization and waiting time. The research provides an important value to the field stating the performance and efficiency of scheduling algorithm is improved with priority based algorithm using experiments whereas the experiments could have been more intensive with comparison between more types of scheduling algorithms providing useful insights into the experiments and evaluations. Lipsa et al. (2023)

The research proposes a priority based round-robin scheduling algorithm. The remaining execution time of the task is compared with the newly arrived high priority task to adjust to dynamic changes. The author performs experiments using variety of datasets and compared the proposed algorithm with Classical Priority induced and MARR scheduling algorithms. Through experiments conducted, the proposed solution performs better when compared to other algorithms in terms of resource utilization and execution time. The author emphasizes the idea of inducing priority based approach to increase the performance of scheduling algorithms and increase the resources utilization. The study provides a direct correlation between priority based approach and the performance of scheduling algorithms but fails to evaluate the performance of the proposed algorithm in comparison with other known scheduling algorithms. Niranjan and Thenmozhi (2023)

The proposal introduces a task scheduling algorithm using fuzzy logic based priority to improve the performance of the proposed scheduling algorithm. The author evaluates the decrease in time offset and increase in utilization of resources, increasing the efficiency of the proposed algorithm when priority based approach is induced. The proposed solution is great for establishing the significance of priority based approach in scheduling algorithms but lacks proper comparison with other scheduling algorithms questioning the significance of the results. Hai-ting and Lin (2020)

A novel priority based scheduling approach is proposed focused on optimization of execution time and the cost of resources. The proposed approach uses execution cost and execution time to prioritize the allocation of tasks. The author evaluates the performance of the proposed algorithm using experiments but fails to provide evidence to support the claim of efficiency of the proposed solution. Jagaty et al. (2020)

The research proposes a dynamic priority based technique to provide efficient scheduling of resources and minimize overall cost of resources. The author compares the proposed dynamic priority approach with priority list technique using factors such as costs, ramp rates and the overall cost of execution. The author tried to convey the importance of priority based approach in scheduling algorithms by comparing with other priority based approach and concludes that the proposed solution provides efficient scheduling approach whereas the experiments conducted are not clear and require more in-depth analysis. Faizah et al. (2024)

The author proposes a heuristic approach combined with multiple priority based techniques to provide an efficient scheduling solution. The proposed solution focuses on multiple scenario based scheduling allocation of resources and provides a balanced solution to maintaining the performance of the scheduling algorithm as well as reducing the cost of resources. The author conducted multiple different experiments based on real world applications but fails to compare the results with other known scheduling solutions. The proposal can play a vital role in the field of scheduling algorithm under dynamic cloud environments due to the flexibility of the solution and providing efficient results in multiple different scenarios but lacks the comparison with other known scheduling approaches. Song et al. (2022)

The research proposes Reliability-based Optimization approach along with priority based scheduling technique to achieve optimal allocation of resources and cost efficiency. The author uses priority based implementation to take care of utilization of resources and prevent resources from being either over-utilized or under-utilized. The author also emphasized on the importance of security in cloud computing by giving priority to critical tasks over other tasks while maintaining efficient resource utilization. The proposed research addresses a major concern with maintaining security in cloud computing field without compromising on the performance of the solution whereas no concrete evidence or evaluation of performance between other known approaches is done to prove the effectiveness of the proposed solution. Maruthi and Sivakumar (2023)

2.5 Conclusion of Related Work

As presented by the related works in the field of importance of scheduling algorithm sin cloud computing, impact of scheduling algorithms on the utilization of cloud resources and impact of priority based scheduling algorithms in dynamic cloud environments we can clearly identify gaps between the focus of increases the utilization of cloud resources to increase the performance of scheduling resources. Also, it can be seen that not much emphasis is done on the fact that inducing the priority based approach can increase both the utilization of resources and decrease the cost of cloud resources. Using these proposed solution we can confirm that there is a need of a solution focused towards maximizing the utilization of resources and also leveraging the priority based technique to enhance the performance of the scheduling algorithms in dynamic cloud environment.

3 Methodology

In the past various virtual machine placement algorithms have been proposed, specializing in different purposes and objectives. With these defined objectives of the respective algorithms, the tendency to be flexible and adapt in dynamic cloud environments is lacked. The proposed novel algorithm evaluates computation, storage, ram and types of virtual machines to optimize the allocation under different dynamic environments or scenarios. To implement the proposed approach we need to discuss the tools, techniques, metrics and methods used to achieve the proposed solution and the desired results.

3.1 Research Tools

To implement the proposed solution we have used varying tools for specific implementations, tasks and objectives.

Table 2: Research Tools

Tools	Purpose
Eclipse IDE	Integrated Development Environment for Java programming.
CloudSim	Framework for modeling and simulating cloud environments.
Microsoft Azure	Cloud computing platform for serverless computing service.
Java	High-level object-oriented programming language.

3.1.1 Eclipse IDE

Eclipse is an integrated development environment for java developers including rich java development tools (JDT) and easy architecture for third party functionality. We are using eclipse for implementing and debugging our simulation code and developing our novel resource alloation algorithm.

3.1.2 CloudSim

Cloudsim is an open-source framework used for simulation cloud computing infrastructure and services including virtual machines, data centers and resources for computing. Sundas and Panda (2020) Cloudsim is being used to simulate cloud data centers, with several virtual machines along with different cloudlets (tasks) to evaluate the perform of the proposed allocation algorithm. We are also comparing other known allocation algorithms with our proposed algorithm using cloudsim simulation code.

3.1.3 Microsoft Azure

Microsoft Azure is a cloud computing platform developed by Microsoft that develop, manage and access cloud applications and services for individuals, companies and governments.Samanta et al. (2024) We have used Microsoft Azure notebooks for plotting meaningful graphs and plots for better result evaluation and understanding.

3.1.4 Java

Java is an open source high level object oriented programming language used for developing and debugging applications and softwares. We have used java as programming language to develop and debug cloud environment simulation code and debug the virtual machine scheduling algorithms.

3.2 Knapsack Problem

Knapsack problem is a famous combinatorial problem that involves fitting a subset of items into a limited capacity for maximizing the total value of items. Fidanova (2020) We can use knapsack problem ideology in solving our problem of optimal resource allocation in cloud environments. If we replace items with virtual machines, total capacity as the capacity of hosts and the value as the metrics of virtual machines then it follows the famous knapsack problem. Using knapsack problem we can propose that we need to find optimal solution where subset of virtual machines are allocated to the fixed amount of hosts in cloud data centers for optimized working of cloud servers.
Using knapsack solution the best virtual machine is allocated to the host based on the capacity of the host and virtual machine which leads to another problem of under-utilization of resources. If the virtual machine does not perfectly fits the requirements of the hosts, then the remaining computational power goes waste and decreases the overall resource utilization. These resources computational power wastage can lead to inefficient and unreliable allocation.

3.3 Fractional Knapsack Problem

To overcome the inefficient allocation using knapsack solution we have used fractional knapsack solution. Fractional knapsack problem is an extension of knapsack problem where the items can be divided for maximizing the total value of the knapsack. Nacef et al. (2023) Using fractional knapsack solution we can divide the resources of the host to ensure better utilization by contributing to remaining resources. This feature also supports the dynamic nature of the cloud environments where the resources can adjust dynamically under different conditions. This helps in achieving maximum utilization of resources.

3.4 Priority Based Allocation

Dynamic cloud environments involves different scenarios and tasks at hand such as cpu-intensive tasks, storage-intensive tasks and energy-efficient tasks. With changing scenarios the allocation of the resources should also change to provide optimal solution with respect to the changing scenarios. To implement this priority based allocation of resources we have implemented weighted priority based value of resources using metrics (storage, bandwidth or mips) depending on the workload. This helps us in achieving true dynamic nature of the cloud environment and provide optimal best solution under varying circumstances.

4 Design Specification

In this section, we firstly describe the infrastructural architecture of the proposed solution presenting how the whole setup of the proposed solution is done. The infrastructural architecture of the proposed novel approach consists of the development, simulation and visualization of the solution. Then we propose the design architecture and the working of the proposed virtual machine allocation algorithm. The design architecture presents the implementation of the proposed solution. Finally the proposed novel Fractional Knapsack Resource Allocation algorithm is introduced which provides optimal allocation of cloud computing resources. The approach focuses on optimal utilization of resources while prioritizing metrics (bandwidth, storage or mips) for dynamic cloud environments.

4.1 Infrastructural Architecture

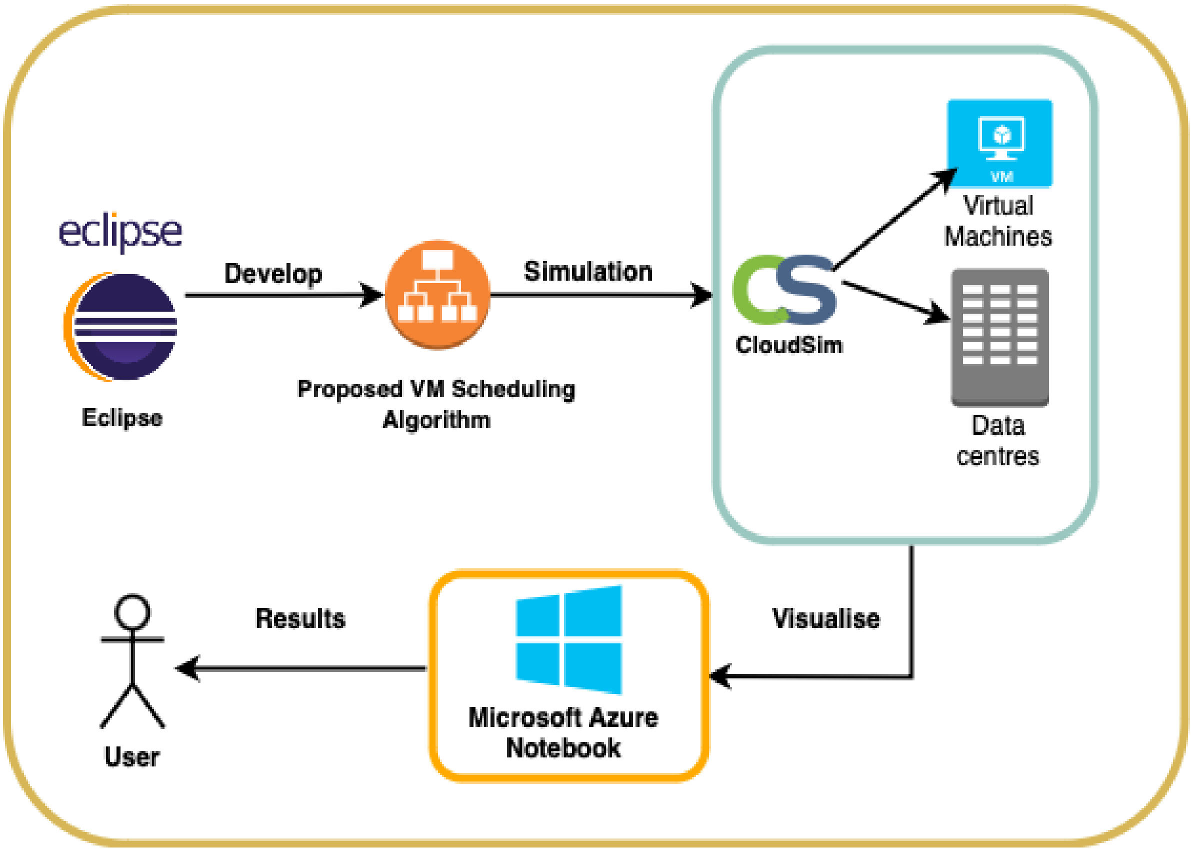

Figure 1: infrastructural Architecture of Proposed Solution

The figure 1 represents the infrastructural workflow of the proposed solution. The development and debugging of the proposed solution starts with eclipse integrated development environment (IDE) which is used to develop the proposed cloud resources scheduling algorithm solution. After the development of the proposed algorithm, the simulation code of cloud environment consisting of data centers, virtual machines and cloudlets is developed using Cloudsim framework. The proposed solution is evaluated and compared with other scheduling algorithms such as round-robin, ant colony and bin packing algorithms. After evaluations and comparison between proposed algorithm and other scheduling algorithms, the results are sent to visualize using Microsoft Azure Jupyter Notebooks.

4.2 Design Architecture

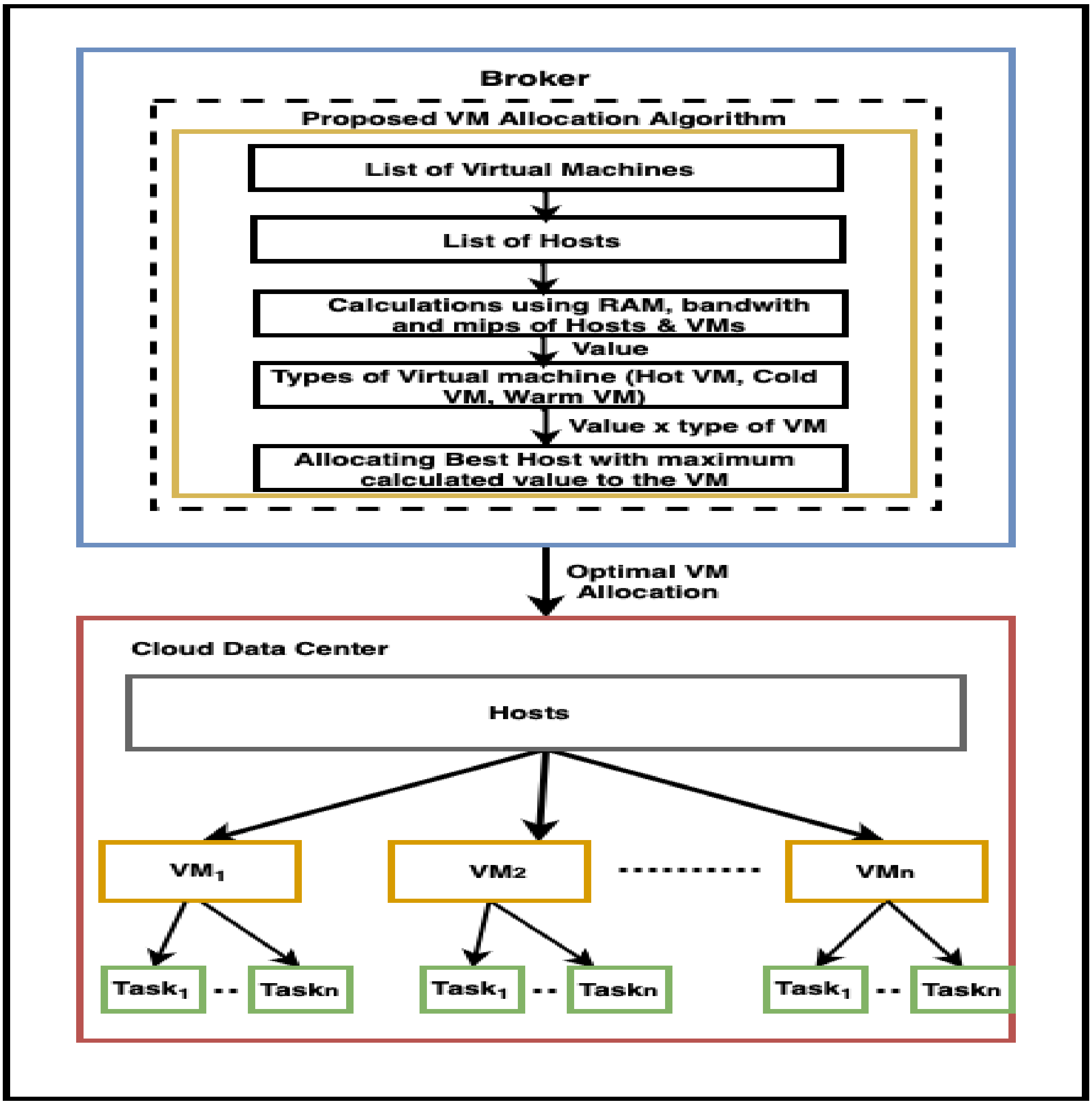

Figure 2: Design Architecture of Proposed Solution

The figure 2 presents the design specification of the proposed virtual machine allocation approach. Here, first the list of list of virtual machines and the list of hosts are passed along with their respective requirements. Then the best host for allocating each virtual machine is calculated based on the ram, bandwidth and the mips (million instructions per second) values of each virtual machine. After calculating the value of host per virtual machine, the value is evaluated with reference to the type of virtual machine (hot vm, cold vm and warm vm). Using all these calculations and metrics the best host for each virtual machine is allocated. Also, the metrics value can be changed based on the priority of the metrics for different scenarios (cpu-intensive tasks, high storage requirement or high bandwidth tasks) resulting in different allocation of resources. After the calculations of all the virtual machines present, the list of allocation to hosts is assigned to cloud data center and the tasks are allocated to the virtual machines. This helps us in providing optimal allocation of resources solution.

4.3 Fractional Knapsack based resource scheduling algorithm

The approach uses fractional knapsack greedy solution to allocate virtual machines to hosts in a cloud environment. The best host of specific virtual machines is being calculated based on the capacity and requirements of the virtual machine using metrics such as storage, mips (million instructions per second) and RAM. The best host for a virtual machine is calculated using a value-to-weight ratio for each VM and host as mentioned in the equation 1. The value of each VM to host is then sorted and the one with the highest value is allocated as the best host for that VM.

$$Value = \frac{VM_{RAM}}{Host_{RAM}} + \frac{VM_{mips}}{Host_{mips}} + \frac{VM_{BW}}{Host_{BW}} \tag{1}$$

After the allocation of a virtual machine to the host, the remaining resources of the host are sent back to the scheduler to be allocated to other Virtual machines. This helps to prevent the under-utilization and over-utilization of the resources.
In addition to the value-to-weight ratio, the weighted coefficient are introduced as well for prioritizing specific metrics (storage, bandwidth and mips) for specific scenarios such as cpu-intensive tasks or storage-intensive tasks. These weights of metrics are combined with the value-to-weight ratio of each VM to select the best host based on specific scenario as seen in the equation 2. Also the coefficient of the type of VM (hot VM, cold VM and warm VM) is multiplied by the value of each resource for VM allocation.

$$Value = RAM_{Weight} \times \frac{VM_{RAM}}{Host_{RAM}} + MIPS_{Weight} \times \frac{VM_{mips}}{Host_{mips}} + BW_{Weight} \times \frac{VM_{BW}}{Host_{BW}} \tag{2}$$

5 Implementation

The proposed novel algorithm is developed using Eclipse ide and Cloudsim framework. Using Cloudsim framework we were able to simulate dynamic cloud environment consisting of cloud data centers, hosts, virtual machines and cloudlets (tasks). We created Fractional Knapsack allocation based class for integrating with the simulation code of the cloud environment along with other resource allocation algorithms such as Round-Robin algorithm, Ant Colony Optimization algorithm and Bin-Packing allocation algorithm. These scheduling algorithms were integrated in the simulation code for comparing the performance between the proposed solution and the these algorithms. We also used priority based approach by introducing weights of each metric of resources (mips, ram and bandwidth) which can be tuned according to the specific scenario to produce more efficient result. We then visualized the obtained results of all the algorithms under different workloads and dynamic cloud environment using Microsoft Azure Notebooks by generating meaningful graphs and clear depictions of the results obtained.

```
For each VM in VM_List:
  bestHost ← null
  bestValue ← -1

  For each Host in Host_List:
    If Host has sufficient resources for VM:
      ramRatio ← VM.ram / Host.available_ram
      mipsRatio ← VM.mips / Host.available_mips
      bwRatio ← VM.bw / Host.available_bw

      valuePerResource ← VM.resource_coefficient *
                ((ramWeight * ramRatio) +
                 (mipsWeight * mipsRatio) +
                 (storageWeight * bwRatio))

      If valuePerResource > bestValue:
        bestValue ← valuePerResource
        bestHost ← Host

  If bestHost ≠ null:
    Allocate VM to bestHost
  Else:
    Log "VM could not be allocated"
```

Figure 3: Fractional Knapsack Resource Allocation Algorithm

Figure 5 shows the pseudo code of the proposed resource allocation algorithm where each virtual machine (VM) in VM_List selects best host among all the hosts present in the Host_List for each VM. This choice is facilitated using ramRatio, mipsRatio and bwRatio of each VM to every host capacity available which are the ratios of metrics of virtual machines to the metrics of host resources. Then these ramRatio, mipsRatio and bwRatio is multiplied by the corresponding weight coefficient which signifies the priority of each metric depending on the scenario. Also, the VM coefficient which defines which type of virtual machine (hot vm, cold vm and warm vm) with each corresponding value is multiplied to the final value of the resource. This value is then matched with corresponding values of other resources (hosts). The host with the maximum value (best host) among all hosts gets to allocate the virtual machine. If there is no host which fulfill the requirements of the VM then the VM is not allocated. This solution follows fractional knapsack approach where we are partially assign the resources to accommodate virtual machines for optimal resource allocation which helps in maximizing resource utilization in dynamic cloud environments.

6 Evaluation

To demonstrate the effectiveness and the efficiency of the proposed solution we conducted four experiments each with different number of cloudlets(tasks) while allocating five virtual machines and three hosts. Each of the virtual machine has varying requirements including one warm VM (mips weight = 0.5), one hot VM (mips weight = 1.0) and three cold VMs (mips weight = 0.1). We used Round-Robin, Ant Colony Optimization and Bin packing allocation algorithms for comparison with proposed solution. Also, we conducted an experiment to validate the performance of the proposed algorithm under different scenarios to measure the dynamic nature of the algorithm under different workload environments.

6.1 Experiment 1

For the first experiment we have used five number of cloudlets where cloudlets length were 100000, 105000, 110000, 115000 and 120000 million instructions respectively. Table 3 shows the average execution time, makespan and total execution time between different algorithms when subjected to these five cloudlets.

Table 3: Algorithm performance (5 cloudlets)

Algorithm	Average Execution Time(ms)	Makespan(ms)	Total Execution Time(ms)
Round Robin	86.16	120.09	430.83
Ant Colony	86.16	120.09	430.83
Bin Packing	530	550.1	2650
Fractional Knapsack	83.49	115.09	417.89

6.2 Experiment 2

In the second experiment we have used ten different cloudlets where cloudlets length were 100000, 105000, 110000, 115000, 120000, 125000, 130000, 135000, 140000 and 145000 million instructions respectively. Table 4 presents the results (average execution time, makespan and total execution time) when performed on round-robin, ant colony, bin-packing and proposed fractional knapsack algorithms using mentioned ten cloudlets.

Table 4: Algorithm performance (10 cloudlets)

Algorithm	Average Execution Time(ms)	Makespan(ms)	Total Execution Time(ms)
Round Robin	181.91	265.09	1819.16
Ant Colony	181.91	265.09	555.48
Bin Packing	1142.5	1225.1	11425
Fractional Knapsack	179.58	250.09	1795.84

6.3 Experiment 3

For the third experiment we have taken thirteen cloudlets but each with random different configurations to see the performance of the allocation algorithms under different workload conditions. Cloudlets with lengths 1000, 25000, 10000, 2500, 2000, 25000, 1000, 250000, 100000, 2500, 2000, 25000 and 25000 million instructions are used respectively. Table 5 presents the output when these thirteen tasks are sent to be fulfilled.

Table 5: Algorithm performance (13 cloudlets)

Algorithm	Average Execution Time(ms)	Makespan(ms)	Total Execution Time(ms)
Round Robin	43.24	287.82	562.12
Ant Colony	42.72	285.09	555.48
Bin Packing	129.83	471.08	1687.84
Fractional Knapsack	36.82	277.59	478.73

6.4 Experiment 4

In this experiment we evaluated the performance of the proposed fractional knapsack algorithm under different scenarios. We adjusted the priority metrics for cpu-intensive and storage-intensive scenario and evaluated the results. For cpu-intensive tasks the weights of metrics are adjusted as mips = 1.5, storage = 0.1 and RAM = 0.1 whereas for storage-intensive scenario we have assigned mips = 0.1, storage = 1.5 and RAM = 0.1 . Table 6 presents the results obtained under these scenarios.

Table 6: CPU-intensive vs Storage-intensive

Workload	mips	stoarge	ram	Makespan(ms)	Total Execution Time(ms)
Cpu-intensive	1.5	0.1	0.1	277.59	478.73
Storage-intensive	0.1	1.5	0.1	285.09	555.48

6.5 Discussion

The experiments are conducted using different cloudlets to better differentiate the performance of the algorithms under different workloads. Upon observation with respect to above performed experiments its evident that the proposed fractional knapsack algorithm is most efficient under different workloads when subject to a varying amount of cloudlets (tasks) when compared to other resource allocation algorithms as seen in the figure 4.

This is due to the partial allocation feature of fractional knapsack that helps the proposed algorithm to perform better under the dynamic conditions of cloud environments. Due to the partial allocation the resource wastage is minimized and the resource utilization is maximized, reducing the execution time when compared to other scheduling algorithms. This can also be seen in the figure 5, where the makespan of the proposed algorithm is minimal when compared to other algorithms. This establishes a direct proportionality between makespan (the maximum time taken by a task) and the execution time of scheduling algorithms. This helps us to understand the significance of optimal allocation of resources to virtual machines which directly impacts the performance of the scheduling algorithm. Scheduling algorithms such as round-robin and bin-packing does not perform well when put under varying workloads due to the inefficient way of allocation of resources whereas the ant colony optimization algorithm and proposed fractional knapsack algorithm are focused towards finding the best resource for each virtual machine and thus performs drastically better than other algorithms. The proposed scheduling algorithm also performs better than the ant colony due to its dynamic nature of allocation

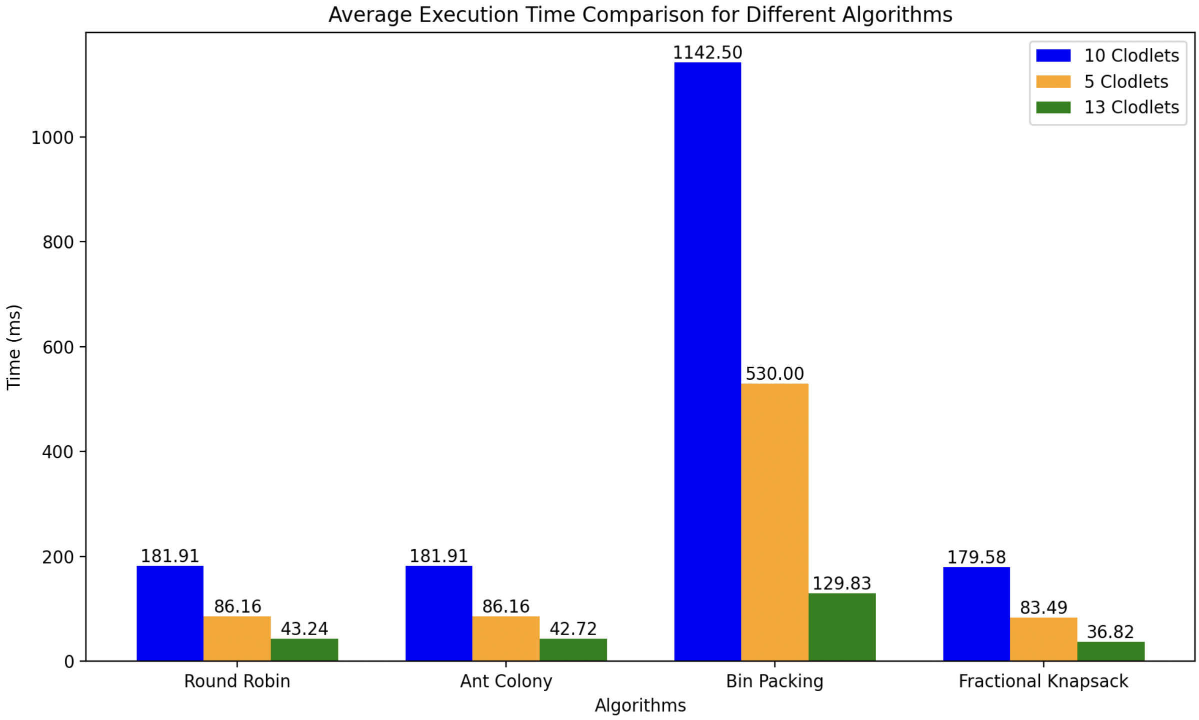

Figure 4: Execution Time vs Algorithms

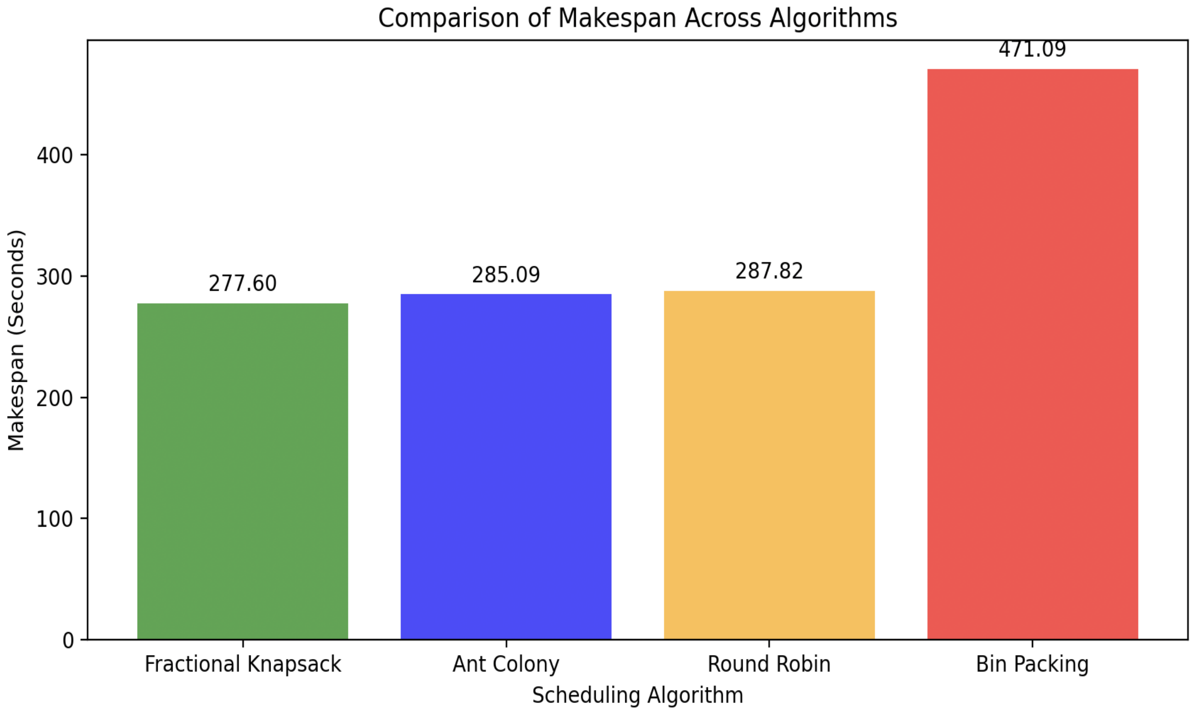

Figure 5: Makespan vs Algorithms

of resources and also due to the priority metric based allocation of resources which performs best under different scenarios. This nature of the proposed algorithm can be seen in the table 6 where by manipulating the weights of metrics (mips, storage, ram), the allocation of the resources can be varied depending on the requirements and workloads.

7 Conclusion

The proposed solution addresses a critical challenge of having scheduling algorithm with optimal allocation of cloud resources, maximum utilization fo resources and minimum cost of cloud resources which subjected under dynamic cloud environment. The proposed fractional knapsack approach helps in achieving optimal allocation of resources while maintaining maximum utilization whereas the priority based implementation was responsible for adjusting the metrics of resources such as to perform best under different dynamic cloud workloads. The proposed solution when compared with other widely known scheduling algorithms such as Round-Robin, Bin-Packing and Ant Colony Optimization, performed best n terms of execution time, makespan and scheduling of cloud resources. The proposed solution holds a significant importance in the field of scheduling cloud resources under dynamic cloud environments while integrating priority based approach for different workloads or scenarios. The proposed solution is limited to only three cloud resources metrics (mips, ram and bandwidth) for calculating the priority of cloud resources and scenarios affecting these metrics can only be tuned currently.

References

Bhandary, I., Atul, K., Athani, A., Patil, S. and Narayan, D. G. (2021). Energy-efficient vm scheduling in the cloud environment using reinforcement learning, *2021 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, pp. 46–51.

Bo, Y., Feng, L. and Xiaoyu, Z. (2022). Cloud computing task scheduling algorithm based on dynamic priority, *2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC)*, Vol. 6, pp. 1696–1700.

Faizah, F., Sudjoko, R. I., Syai'in, M. and Soeprijanto, A. (2024). Dynamic priority approach for unit commitment scheduling solution, *2024 International Electronics Symposium (IES)*, pp. 114–119.

Fidanova, S. (2020). Hybrid ant colony optimization algorithm for multiple knapsack problem, *2020 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, pp. 1–5.

George, N., Nandhakumar, K. and Vijayan, V. P. (2021). Survival study on resource utilization and task scheduling in cloud, *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pp. 1814–1819.

Hai-ting, D. and Lin, Z. (2020). Research on task scheduling algorithm based on fuzzy logic priority, *2020 IEEE International Conference on Information Technology,Big Data and Artificial Intelligence (ICIBA)*, Vol. 1, pp. 669–675.

Hemanth, S. V., Kirubha, D., Reddy, S. R., Chelladurai, T., Soundari, A. G. and Amirthayogam, G. (2024). Multi objective ant colony optimization technique for task scheduling in cloud computing, *2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, pp. 830–835.

Ibrahim, M., Nabi, S., Baz, A., Alhakami, H., Summair Raza, M., Hussain, A., Salah, K. and Djemame, K. (2020). An in-depth empirical investigation of state-of-the-art scheduling approaches for cloud computing, *IEEE Access* **8**: 128282–128294.

Jagaty, P., Sahoo, S., Patel, D. and Sahoo, B. (2020). Priority queue based real-time task scheduling in virtualized cloud environment, *2020 7th International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 862–866.

Karthik, G., Gupta, A., Rajeshgupta, S., Jha, A., Sivasangari, A. and Mishra, B. P. (2023). Efficient task scheduling in cloud environment based on dynamic priority and optimized technique, *2023 International Conference on Artificial Intelligence and Smart Communication (AISC)*, pp. 1–6.

Khare, M. D., Kukreti, R. and Jaison, F. (2023). A comprehensive review of scheduling and optimization paradigms in cloud systems, *2023 International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT)*, pp. 1220–1224.

Lipsa, S., Dash, R. K., Ivković, N. and Cengiz, K. (2023). Task scheduling in cloud computing: A priority-based heuristic approach, *IEEE Access* **11**: 27111–27126.

Manchanda, A., Kaur, A. and Kaur, A. (2023). Cloud computing: Resource management, categorization, scalability and taxonomy, *2023 2nd Edition of IEEE Delhi Section Flagship Conference (DELCON)*, pp. 1–5.

Maruthi, P. and Sivakumar, C. (2023). The smart performance analysis of reliability based optimization service for secured cloud servers using priority based scheduling, *2023 4th International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pp. 633–640.

Nacef, A., Mechta, D., Louail, L. and Benchikh, L. (2023). Efficient energy routing in smart grid networks using fractional knapsack and dijkstra algorithm, *2023 International Conference on Decision Aid Sciences and Applications (DASA)*, pp. 565–569.

Niranjan, P. and Thenmozhi, M. (2023). Priority-induced round-robin scheduling algorithm with dynamic time quantum for cloud computing environment, *2023 Second International Conference on Advances in Computational Intelligence and Communication (ICACIC)*, pp. 1–5.

Reddy, P. S., Trisha, K., Sree, N. H., Bhaskaran, S. and Sampath, N. (2024). Enhanced task scheduling in cloud computing: A comparative analysis and hybrid algorithm implementation using cloudsim, *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pp. 1–5.

Reddy, P. v. and Reddy, K. G. (2021). An analysis of a meta heuristic optimization algorithms for cloud computing, *2021 5th International Conference on Information Systems and Computer Networks (ISCON)*, pp. 1–6.

Samanta, D., Roy, S., Sarkar, P. and Misra, B. (2024). Extracting, transforming, and analyzing olympic data using azure services: A case study, *2024 International Conference on Advances in Computing Research on Science Engineering and Technology (ACROSET)*, pp. 1–6.

Singh Panesar, G. and Jain, N. (2023). Machine learning heuristics in cloud computing: An in-depth analysis of vm migration and job scheduling strategies, *2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS)*, pp. 1443–1448.

Singh, R., Singh, S. and Kaur, B. (2023). Energy-efficient vm allocation and migration approach using swarm intelligence algorithm in cloud computing environment, *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pp. 1–6.

Song, H., Jia, G. and Peng, W. (2022). Bi-objective reactive project scheduling problem under resource uncertainty and its heuristic solution based on priority rules, *IEEE Access* **10**: 52700–52711.

Sundas, A. and Panda, S. N. (2020). An introduction of cloudsim simulation tool for modelling and scheduling, *2020 International Conference on Emerging Smart Computing and Informatics (ESCI)*, pp. 263–268.

Supreeth, S., Patil, K., Patil, S. D. and Rohith, S. (2022). Comparative approach for vm scheduling using modified particle swarm optimization and genetic algorithm in cloud computing, *2022 IEEE International Conference on Data Science and Information System (ICDSIS)*, pp. 1–6.

Walia, N. K. and Kaur, N. (2021). Performance analysis of the task scheduling algorithms in the cloud computing environments, *2021 2nd International Conference on Intelligent Engineering and Management (ICIEM)*, pp. 108–113.

Wang, H. and Wang, H. (2022). Survey on task scheduling in cloud computing environment, *2022 7th International Conference on Intelligent Informatics and Biomedical Science (ICIIBMS)*, Vol. 7, pp. 286–291.