Enhanced Genetic Algorithm For Dynamic Dependent Workloads To Improve Load Balancing Efficiency in Cloud Computing

MSc Research Project
MSc Cloud Computing

Zain Ull Abbdin Mohammed
Student ID: x22142169

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Zain Ull Abbdin Mohammed
Student ID:	x22142169
Programme:	MSc Cloud Computing
Year:	2023
Module:	MSc Research Project
Supervisor:	Vikas Sahni
Submission Due Date:	14/12/2023
Project Title:	Enhanced Genetic Algorithm For Dynamic Dependent Workloads To Improve Load Balancing Efficiency in Cloud Computing
Word Count:	4556
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:	14th December 2023

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):	

Enhanced Genetic Algorithm For Dynamic Dependent Workloads To Improve Load Balancing Efficiency in Cloud Computing

Zain Ull Abbdin Mohammed
x22142169

Abstract

In the recent years, the potential of cloud computing (CC) has attracted a lot of attention to improve the scalability of cloud Data Centers (DC). Effective load balancing (LB) solutions are required to ensure the best possible distribution of workload. LB is a major issue for distributed computing systems, particularly in cloud environments with several customers. To enhance LB in cloud computing, numerous tactics, methods, and techniques have been developed over time. These methods concentrate on lowering execution time, cutting down on energy usage, maximizing resource utilization, and accelerating task scheduling among a group of virtual machines (VMs). However, these traditional methods often fail to consider the dynamic and interdependent workload, which may result in overloading-related problems.

An algorithm that can improve load balancing efficiency in cloud computing and dynamically manage dependent workloads is necessary to address this challenge. This research proposes an Enhanced Genetic Algorithm (EGA), which is based on natural process. It is a technique where two fittest parents (or virtual machines) are selected from the initial population (Datacenter, Virtual Machine, Cloudlets) and mutated to create an offspring (a new virtual machine) to manage the dynamic load. The outputs like total energy utilized, resource utilization, and execution time were captured in several iterations and the same was compared with Particle Swarm Optimization (a load balancing algorithm). The average results show that EGA outperforms PSO and is about 77% more efficient.

1 Introduction

In cloud computing, the purpose of load balancing is to optimally distribute compuing workloads across several servers to better the system performance and resource utilization Mishra and Mishra (2015). Advanced load balancing strategies are required because dynamic and dependent workloads are becoming more common in cloud environments. In an attempt to improve the load balancing efficiency, this study suggests an Enhanced Genetic Algorithm (EGA) for dynamic and dependent workloads in cloud computing environment.

The Enhanced Genetic Algorithm (EGA), which is derived from the Genetic Algorithm (GA), is a versatile and efficient approach to handle Load Balancing criterion.

Salvi (2022). It is made to properly distribute workloads among cloud resources, improving system efficiency and guaranteeing optimal use. To cater to the varying demands of cloud computing, the study attempts to adapt and deploy the EGA specifically for dynamic and dependent workloads. This research is a continuation of future work mentioned in the research Salvi (2022) done by Rohit Rajesh Salvi.

1.1 Motivation

Advanced load balancing mechanisms are required for dynamic and dependent workloads in cloud environments, and the EGA offers an adaptable and practical solution to these problems Salvi (2022). Because of the growing reliance on cloud services and the requirement for effective resource use, the research of the Enhanced Genetic Algorithm (EGA) is essential. There is a lot of potential for cloud computing in many different areas, but efficient load balancing is crucial to guaranteeing the best performance and resource distribution Shafiq et al. (2022).

Load balancing becomes increasingly important with the expansion of cloud data centres and the development in complicated computational jobs. The EGA, a more advanced evolutionary algorithm, gives prospective answers for how to improve load distribution and system performance Wang et al. (2014). To achieve optimal resource usage and satisfy the various requirements of cloud computing environments, it is important to research and modify the EGA to precisely cater to dynamic and dependent workloads.

1.2 Research Question

Can Enhanced Genetic Algorithm perform better than existing load balancing techniques to manage dynamic workloads in cloud computing environment?

1.3 Research Background and Objectives

A key component of cloud computing is load balancing, which guarantees optimum system performance and resource efficiency. Workloads are dynamic and frequently interdependent in cloud systems, making optimal task distribution difficult Joshi and Kumari (2016). Traditional load balancing methods have difficulty properly adjusting to these dynamic and variable workloads. Genetic Algorithms (GAs) provide an evolutionary computation approach that mimics natural selection processes. These algorithms have shown promise in finding optimal solutions for various optimization problems Makasarwala and Hazari (2016). However, applying a standard GA directly to load balancing in the cloud might not fully address the intricacies of dynamic and dependent workloads. To address this, an enhanced Genetic Algorithm (EGA) tailored for dynamic and dependent workloads in cloud computing has been proposed. EGA aims to distribute workloads effectively across virtual machines (VMs), preventing overload and optimizing resource usage. By considering the dynamic nature of workloads and their dependencies, this algorithm seeks to significantly improve load balancing efficiency in cloud systems.

2 Related Work

Users of the cloud receive scalable and effective services without having to provision or manage physical infrastructure. Daily growth in the cloud industry brings with it new challenges Raza et al. (2015). Cloud load balancing is one such challenge. To address these issues, a substantial amount of current research has been carried out, and it makes several approaches, algorithms, frameworks, etc. This section provides a comprehensive overview of research on resource allocation, job scheduling, virtual machine allocation, and load balancing with genetic algorithms.

2.1 Allocation of VMs in Cloud Computing

A detailed study on virtual machine placement (VMP) in cloud infrastructure management is presented in Lopez-Pires and Baran (2015). One of the difficult challenges in the cloud-infrastructure management is VMP according to experts. The paper discusses a variety of VMP-related subjects, including cloud service marketplaces, Quality of Service (QoS), Service Level Agreements (SLA), and, energy efficiency. Finding research possibilities in this important area is the goal, with a focus on how to choose the best cloud providers for hosting virtual machines and how to allocate resources optimally for performance and cost-effectiveness.

Enhancing resource utilization efficiency, lowering user expenses, and cutting down on computation time are the main objectives of research conducted by Shi and Lin (2022). In order to achieve distribution stability, the authors provide a brand-new multi-objective optimization method for dynamic resource allocation across many virtual machines. The optimization methodology is focused on maximizing resource use to provide consumers with savings in terms of costs and computation time.

The difficulties with virtual machine (VM) allocation in cloud computing settings are discussed in study Ezugwu et al. (2013), with a focus on private cloud architecture. Pertaining to the cloud environment, this study emphasizes the significance of VM allocation in an attempt to effectively use and manage resource and increase overall cloud system performance. The analysis and solution of the allocation problem reveals the complexity and problems involved in effectively distributing virtual machines across cloud infrastructures.

Task allocation method for Virtual Machines (VMs) in cloud computing has recently advanced. It draws attention to the difficulties cloud data centers have when allocating resources and applications according to demand. In order to maximize resource use and improve cloud data center performance, the study Ullah et al. (2022) underlines the significance of appropriate VM allocation. It looks at advancements from 2015 to 2021 and examines how VMs might be used to efficiently distribute resources within a data center.

In the study Hashemi et al. (2021), a multiple objectives approach to a virtual machine (VM) allocation in cloud computing environments is covered. Finding appropriate locations for virtual machines (VMs) on physical machines (PMs) to accomplish predetermined goals with an emphasis on lowering energy consumption. The goal of the study is to ensure that programs running on virtual machines execute promptly while optimizing resource distribution while taking energy efficiency into account. The suggested approach aims to achieve a balance between energy-saving cloud infrastructure and effective VM placement.

2.2 Task Scheduling in Cloud Computing

The fundamental idea of task scheduling in cloud computing is examined in this paper Arunarani et al. (2019). Optimizing task scheduling happens to be the main goal to judicially reduce time loss and increase performance. Numerous studies have looked into different work scheduling techniques to improve performance. The study emphasizes how important efficient scheduling algorithms are and how they affect cloud computing. These techniques are designed to ensure effective utilization of virtual machines and minimize makespan by allocating jobs to available resources as efficiently as possible. The paper focuses on the significance of assessing and choosing suitable scheduling strategies to ensure effective work allocation.

The paper Mahmood et al. (2021) offers a summary and analysis of the various task scheduling algorithms used in cloud computing. It explores the many algorithms used in the cloud computing environment, concentrating on how adaptable, practical, and appropriate they are for job scheduling. The review explains the difficulties these algorithms have encountered and how researchers have solved them.

Task scheduling in cloud context is covered in the paper Hai et al. (2023), with an emphasis on security and optimization. It discusses the difficulties cloud platforms have in effectively managing workflow submissions and highlights the need of task scheduling optimization. The study suggests several HEFT algorithm iterations modified to yield better outcomes, notably in the context of rank generation. With the goal of minimizing service response times, cutting costs, and improving overall quality of service (QoS), optimization is recognized as a crucial component. The study also recognizes the need of security while scheduling tasks in cloud systems.

2.3 Resource Allocation in Cloud Computing Using Optimization Methodologies

The technique adopted to assign resources to customers based on their adaptable needs in cloud computing is highlighted in the paper Manzoor et al. (2020). In order for cloud services to successfully and efficiently meet a range of demands, resource allocation is a critical component. The document describes methods and strategies used for resource allocation, allowing cloud service providers to minimize resource consumption and improve the quality of service.

The paper Kumain (2020) presents a summary of the various resource allocation optimization approaches utilized in cloud computing systems. It investigates strategies for boosting resource use and effectiveness in cloud computing settings. The authors examine the state of the art in resource allocation and optimization research, highlighting several strategies used to optimize resource allocation in cloud systems. The offered search results do not include a description of the strategies' specifics, but the paper appears to be a useful source for learning how optimization is essential to efficiently manage resources in cloud environments.

2.4 Genetic Algorithm for Load Balancing

The paper Naz et al. (2023) offers a technique to virtual machine (VM) allocation in cloud computing that uses genetic algorithms. By effectively distributing VMs to servers, it focuses on optimizing resource use with the goal of minimizing resource waste and maximizing performance. In order to establish the best allocation strategy, the Genetic Algorithm is used. It takes into account the workload, server capacity, and VM requirements. The goal is to efficiently distribute VMs among servers, improving system performance and resource utilization as a whole.

In order to optimize load distribution in cloud computing environments, the study Lagwal and Bhardwaj (2017) suggests a load balancing technique using a Genetic Algorithm (GA). The GA-based strategy uses intelligent sorting of virtual machines (VMs) depending on how long they take to execute to effectively distribute workloads to VMs. Load balancing is accomplished by allocating smaller jobs to VMs with the longest execution times, eventually enhancing the speed and efficiency of the cloud architecture. For an optimal resource usage and improve overall system efficiency, the study underlines the significance of effectively balancing the load in cloud computing.

The simulation study that the paper Greene (2001) presents focuses on dynamic load balancing in heterogeneously distributed systems. In order to maximize load distribution, the paper presents a Genetic Algorithm (GA) with multiple criteria. In a dynamic distributed system, this GA-based method seeks to efficiently divide the computational burden among several nodes. To achieve efficient load distribution, the method uses a nonnegative-valued function that has been normalized. The paper makes contributions to understanding how to deal with dynamic load-balancing difficulties, particularly in systems with fluctuating and changing workloads and computing demands.

The Balancer Genetic Algorithm (BGA) is a unique method presented in paper Gulbaz et al. (2021) to handle the task scheduling problem associated with cloud computing. A key component of cloud computing systems, work scheduling is handled well by the BGA. In order to decrease resource idle time and increase overall efficiency, the research suggests a method for allocating tasks to resources in the most effective way possible. The goal of this article is to improve task scheduling in cloud environments through the BGA, making it possible to use computational resources more efficiently.

2.5 Using Improved Genetic Algorithm For Load Balancing

In his research, Rohit Rajesh Salvi applies improved genetic algorithm (EGA) to increase load balancing efficiency especially in cloud computing environments. The aim here is distributing workloads among virtual machines (VMs) while considering the current load on VM clusters. The EGA evaluates workload and performance characteristics to distribute jobs to VMs, preventing VM overload. The report acknowledges the need for a reliable load balancing system. It is possible to establish effective load distribution within the cloud infrastructure using the suggested EGA-based techniqueSalvi (2022).

Table 1: Literature Review Summary

Literature Review	Description of Concept	Relevance to Research
Allocation of VMs in Cloud Computing	Shi and Lin (2022), Lopez-Pires and Baran (2015) Optimize dynamic resource allocation, VM placement issues, and cloud provider selection for performance and cost effectiveness.	Importance of choosing the right cloud provider and efficient resource allocation for performance and cost savings.
Allocation of VMs in Cloud Computing	Ezugwu et al. (2013), Ullah et al. (2022) Importance and impact of VM allocation on cloud data center performance.	Highlights difficulties and complexity in distributing VMs, essential for maximizing resource utilization and enhancing cloud system performance.
Allocation of VMs in Cloud Computing	Hashemi et al. (2021) Proposes a multi-objective method for VM allocation ensuring fast program execution and energy economy.	Optimization of program execution and energy efficiency crucial in cloud settings, balancing energy-efficient architecture and appropriate VM placement.
Task Scheduling in Cloud Computing	Arunarani et al. (2019), Mahmood et al. (2021) Implement efficient scheduling algorithms and discusses various task scheduling methods.	Efficient workload allocation depends on scheduling methods, impacting VM usage and minimizing task completion time.
Task Scheduling in Cloud Computing	Hai et al. (2023) Explores task scheduling optimization focusing on QoS improvements using a modified HEFT algorithm.	Acknowledges the need for better scheduling strategies for improved service quality.
Resource-Allocation in the Cloud Computing	Manzoor et al. (2020), Kumain (2020) Explain resource distribution techniques and overview optimization techniques for cloud system resource allocation.	Necessity of optimization techniques for successful cloud service delivery and resource management in cloud settings.
Genetic Algorithm For Load Balancing	Naz et al. (2023), Lagwal and Bhardwaj (2017) Suggest using Genetic Algorithm for load distribution.	Incorporating genetic algorithms in load balancing optimizes resource use and task distribution.
Improved Genetic Algorithm For Load Balancing	Salvi (2022) Proposes an Enhanced Genetic Algorithm (EGA) increasing load balancing efficiency in cloud computing, avoiding virtual machine overload.	Enhances load balancing efficiency in cloud computing.

2.6 Literature Review Summary

The Table 1 summarizes the literature review. The research Salvi (2022) shows efficiency of EGA in static workload conditions. However, this research in an attempt to evaluate load balancing in dynamic workload setup using EGA.

3 Methodology

This study uses an Enhanced Genetic Algorithm (EGA) in a systematic approach to improve load balancing efficiency in cloud computing. To evaluate the efficiency this research compares the performance of EGA with PSO (Particle Swarm Optimization). The PSO algorithm is a popular population-based optimization method that takes inspiration from the typical behaviors of bird flocking. Marini and Walczak (2015).

3.1 Experimental Setup

In order to mimic dynamically dependent workloads, the study makes use of CloudSim, a simulation tool for modeling and simulating cloud computing infrastructure. In this CloudSim configuration, a Java application is created to run the EGA and PSO algorithms.

3.2 Data Collection

3.2.1 Workload Simulation:

CloudSim generates a variety of dynamically dependant workload situations to simulate real-world cloud computing environments. These include a range of resource needs and computing requirements.EGA framework that uses an enhanced mutation method within the traditional Genetic Algorithm technique to intelligently handle dynamic workload fluctuations is used to simulate the dynamic workload.

- **Genetic Algorithm:** It is a technique for solving and optimizing problems that mimics natural selection. It provides a population of possible solutions recorded as chromosomes, which are often shown as binary code or strings of integers Lambora et al. (2019). These solutions evolve via several iterations through the process of selection, mutation and, crossover. Selection involves choosing the fittest individuals based on their fitness, in order for the fittest individuals to go on to the next generation. In order to produce new solutions and mimic reproduction, crossover combines genetic information from selected individuals. To ensure diversity, mutation introduces random alterations in some individuals. The approach is helpful in a variety of disciplines since it keeps going through this cycle until it converges towards an ideal or nearly optimum solution. The Figure 1 below depicts the flowchart of a Genetic Algorithm.

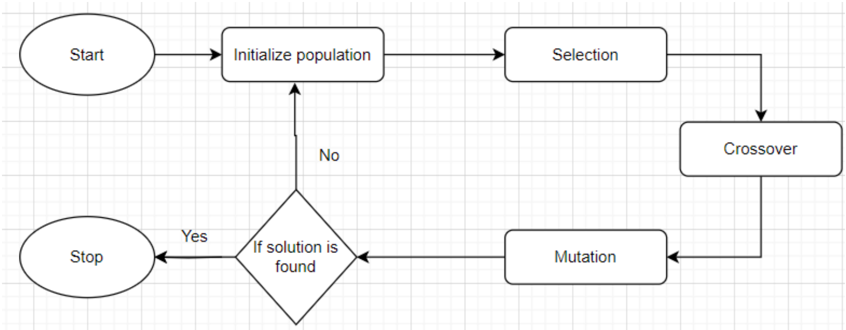

Figure 1: Genetic Algorithm Flowchart

- **Initialization** An initial population typically consists of a set of chromosomes, each of which has a group of genes that might be used to solve a particular problem. These chromosomes and their genes serve as the population's blueprints for a variety of solutions. Genetic algorithms mimic the natural evolutionary process by simulating the evolution of these chromosomes and genes through iterations (Source: [1]).

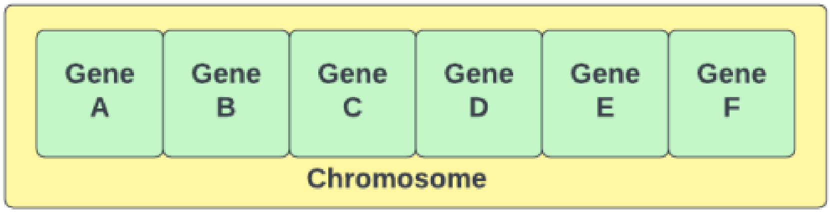

Figure 2: Initial Population

- **Selection:** Higher fitness levels, as determined by a fitness function, increases the probability that an individual will be chosen for reproduction. This is similar to natural selection's "survival of the fittest" theory, enabling better solutions to pass on to the next generation.

- **Crossover:** The GA uses a crossover operation after fitness evaluation and selection, in which selected parents are chosen to produce new offspring. Genetic information is exchanged between parents via methods such as single-point or two-point crossover, leading to the creation of new solutions (Source: [2]).

[1] Figure 2 `https://norma.ncirl.ie/6488/1/rohitrajeshsalvi.pdf`
[2] Figure 3 `https://norma.ncirl.ie/6488/1/rohitrajeshsalvi.pdf`

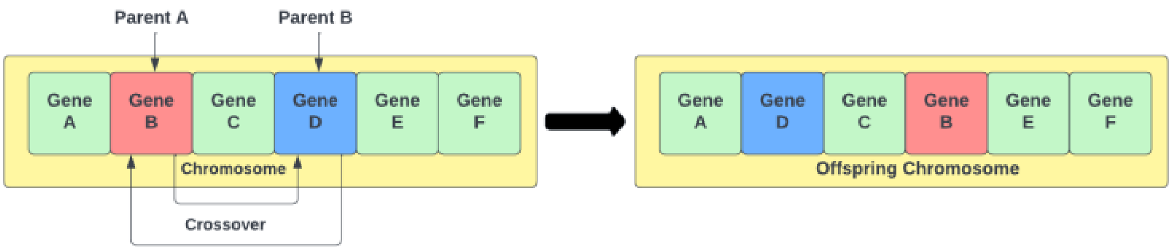

Figure 3: Crossover

- **Mutation:** Random changes are introduced at gene level in this phase to explore new solutions and maintain diversity. The above steps go through a number of iterations until a solution to a problem is achieved (Source: [3]).

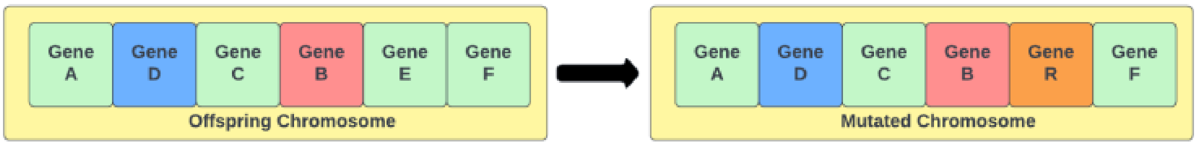

Figure 4: Mutation

3.2.2 Execution of Algorithms

The EGA and PSO algorithms are applied in the simulated set up and parameters like computation time, resource allocation and energy consumption are captured.

3.3 Data Analysis:

The analysis of data is done by taking an average of outputs captured through several iterations of both the algorithms. The average is taken because the workload is dynamic in nature and each run will result in a different output. Evaluation will be based on the time taken for the workload to get processed, how many resources were utilized, and the energy consumed during each iteration.

4 Design Specification

The approach of the traditional GA will be adopted by the Enhanced Genetic Algorithm (EGA), which will use an enhanced mutation strategy. The Datacenter (DC), Virtual Machines (VMs), and cloudlets parameters are established during initial phase. The starting population for EGA is comprised of these parameters. The fitness function then assesses this initial population to determine the ideal fitness value and the two fittest VMs, parent A and parent B. Next, we switch these two fittest VMs using a single-point crossover to create an offspring. These offspring then go through a mutation phase. Unlike the traditional GA, which randomly modifies one VM parameter, in this case, the VM

[3]Figure 4 `https://norma.ncirl.ie/6488/1/rohitrajeshsalvi.pdf`

is completely replaced with another VM that has the appropriate specifications in order to prevent any malfunctions brought on by incorrect parameter modifications. A cloud infrastructure made up of DCs, VMs, and cloudlets is created by the resultant modified population. The cloudlets are processed within the virtual machines (VMs) built in the DC during a simulation, and the simulation produces the total time taken to process each task or cloudlets.

4.1 Enhanced Genetic Algorithm

The Enhanced Genetic Algorithm (EGA) resembles to a smart approach to problem-solving that draws inspiration from natural processes. It is an improvement on a standard procedure known as the Genetic Algorithm (GA). EGA is better because it selects the finest solutions more cleverly, combines them effectively, and modifies them in a smart way. This aids in its ability to solve complex puzzles, particularly those involving dynamic and related tasks in cloud computing.

4.1.1 Initialization

The initial population is formed during EGA's initial phase, where the population of attainable solutions (also called as chromosomes) is created. Every solution, which is usually recorded as a string of values (genes), indicates a potential solution to the problem. In this study, improving LB efficiency of dynamic workload is the problem and the individuals that would help us discover a solution to the problem are DC, VMs and Cloudlets. The Figure 5 represents the initial population where a group of VMs form a chromosome and each VM is a gene(Source: [4]).

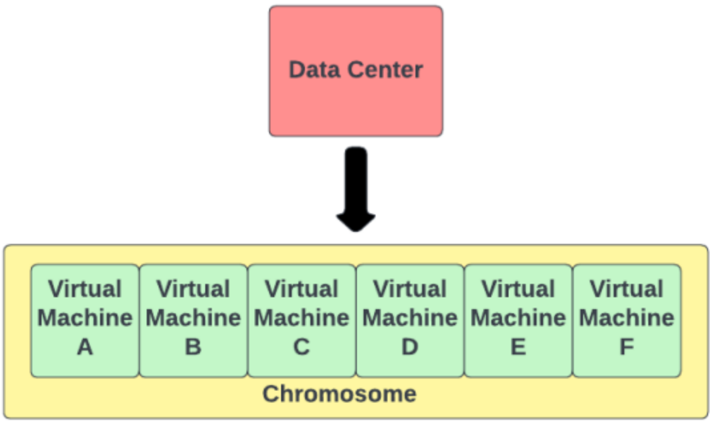

Figure 5: Initialization in EGA

4.1.2 Selection

Fitness Evaluation : Analysis of each VM's fitness within the population happens in this step. The fitness function evaluates a solution according to predetermined standards. Better solutions are indicated by higher fitness values.

[4]Figure 5 `https://norma.ncirl.ie/6488/1/rohitrajeshsalvi.pdf`

Selection Process : Chooses VMs from the population to become parents. Fitter VMs are more likely to be chosen since they pass on their genetic information to the next generation.

4.1.3 Crossover

This step involves pairing the selected VMs (parents) and performing a crossover operation. In order to make new offspring (VM in our case), this involves the exchange of genetic information between parents Figure 6. Multiple techniques exist to exchange genetic information, such as single-point crossover, two-point crossover, and uniform crossover. Each technique manipulates the genetic information differently (Source: [5]).

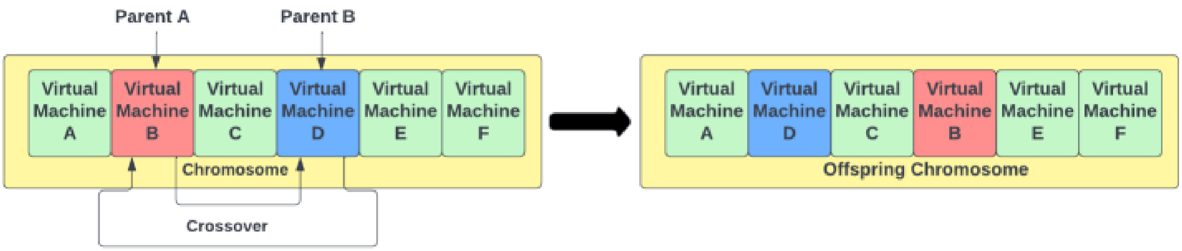

Figure 6: Crossover

4.1.4 Mutation

Offspring VM generated in the previous step is mutated to introduce diversity in the population, meaning a new VM is replaced in chromosome. Figure 7. This VM will be a better fit to handle dynamic workload more efficiently (Source: [6]).

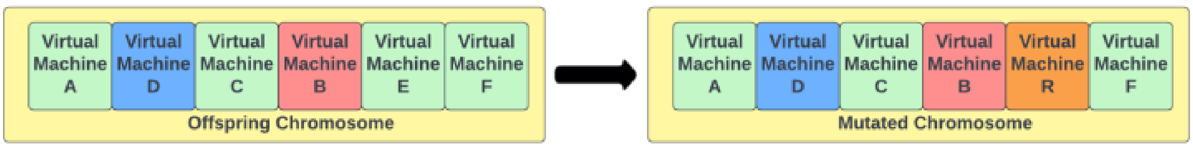

Figure 7: Mutation

4.2 Pseudocode

The algorithm's pseudocode is as follows:

Step 1: Details and parameters are defined for DC, VMs and Cloudlets.
Step 2: Setup the Initial Population

- Create Datacenter, Cloudlets, VMs randomly.

Step 3: Using the Fitness Function, find the fitness value.

[5]Figure 6 `https://norma.ncirl.ie/6488/1/rohitrajeshsalvi.pdf`
[6]Figure 7 `https://norma.ncirl.ie/6488/1/rohitrajeshsalvi.pdf`

- Initialize threshold = 800000
- Iterate through VMs:
- Iterate through Cloudlets:
- Calculate temp = Cloudlet length / VM's MIPS
- Calculate current = current + temp
- If current < threshold, update threshold, and store parent_a and parent_b
- Store the best_fit_parent_a and best_fit_parent_b VMs based on the fitness function.

Step 4: Crossover Machines

- Swap the VMs parent_a with parent_b.

Step 5: Mutation

- Mutate a random VM by changing its specifications to a new configuration.

Step 6: Submit and run Datacenter, Cloudlets, and Mutated VMs in Cloudsim.
Step 7: Capture results

5 Implementation

The research uses Cloudsim (Source: [7]) as a simulation tool to carry out the experiments pertaining to dynamic load balancing in cloud computing environments. Enhanced Genetic Algorithm (EGA) was proposed to optimize the efficiency for effective load balancing in a dynamic workload setup. Since EGA is based on a concept of natural process, it requires initial population, the initial population in this study are cloudlets (tasks), VMs, and Datacenter. Figure 8 depicts the initial population for cloudsim. The cloudlets in the figure represent the varying workload on VMs inside a datacenter.

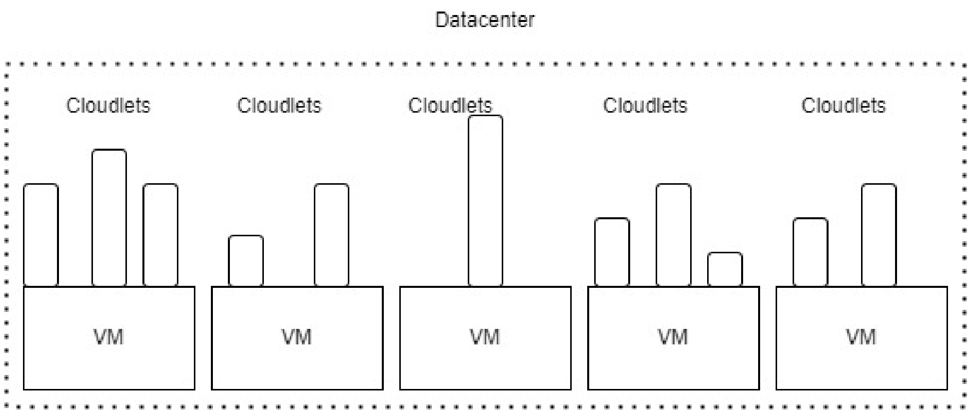

Figure 8: Initial population for Cloudsim

[7]Cloudsim URL `https://github.com/Cloudslab/cloudsim/releases/tag/cloudsim-3.0.3`

The code to implement this simulation was written in Java (JDK 17.0.5) programming language, run on an Eclipse IDE (4.27.0 version) environment. Since the workload on the VMs is dynamic in the simulation, two load balancing algorithms EGA and PSO were implemented and average of ten iterations was used to measure and compare the efficiency of the algorithms to handle load balancing in a dynamic environment. The parameters that were captured in the outputs are energy consumed, resources utilized, and the time taken to execute the cloudlets.

5.1 CloudSim

CloudSim is one of the widely used framework for modeling and simulating cloud computing services and infrastructures. It gives a platform to model multiple aspects of cloud environments and assess various architectures, rules, and algorithms in diverse contexts Goyal et al. (2012). The tool makes it possible to model hosts, virtual machines, cloud data centers, and other components of a cloud-based environment. This research uses cloudsim to simulate Datacenter, Virtual Machines, and Cloudlets to implement EGA and PSO algorithm to compare the efficiency of dynamic workload in a cloud environment.

5.1.1 Datacenter

An essential component of cloud computing, a datacenter is the infrastructure on which cloud resources (such as VMs, hosts, storage, and so on) are located. It is in charge of providing and overseeing these resources. Datacenter in this simulation initializes properties including the operating system, architecture, time zone, cost parameters, and so on. It sets the properties of the physical machines that will host the virtual machines (VMs) and generates hosts and PEs within the datacenter. Table 2 lists all the parameters that were defined in the simulation.

Table 2: Datacenter Parameters

Parameter	**Value**
Operating system	Linux
Cost per Bandwidth	0.2
RAM	16384
Cost per Storage	0.2
VMM	Xen
Cost per Memory	0.1
Timezone	10
Storage	100000
Architecture	x86
Cost	5

5.1.2 Virtual Machines

Virtual machines, or VMs, are the instances of virtual computers that are assigned to users to execute their tasks or applications. They contain the allotted memory, storage, computing power, and other resources Goldberg (1974). This simulation produces a certain number of virtual machines (VMs) dynamically with different specifications, including size, RAM, number of PEs, MIPS (Million Instructions Per Second), bandwidth, and VM scheduling strategy in order to process and execute cloudlets.

5.1.3 Cloudlets

Individual jobs or tasks that users submit to be completed in a cloud environment are represented by cloudlets. The simulation produces a certain number of cloudlets dynamically using different characteristics, like the length of execution time, the file size, the output size, and the quantity of Processing Elements (PEs) to schedule and execute them.

6 Evaluation

The research compares two algorithms (Enhanced Genetic Algorithm and Particle Swarm Optimization) to see which one is more efficient for load balancing in a dynamic workload environment. Since the outputs are dynamic, the average of 10 iterations of each of the algorithm is considered in this research to compare and justify which one is more efficient. Three parameters on which the comparison was done are resource utilization, the time of execution, and the energy consumed. Below are the results of each iteration. These results represent the total execution time, total number of resources used and total energy consumed per simulation.

Runs	Enhanced Genetic Algorithm (EGA)			Particle Swarm Optimization (PSO)		
	EGA Execution Time	EGA Resource Utilization	EGA Energy Consumption	PSO Execution Time	PSO Resource Utilization	PSO Energy Consumption
1	4.553	103.172	10.317	39.029	2188.707	87.548
2	5.982	120.923	12.092	15.981	253.81	10.152
3	8.648	210.216	21.021	14	273.153	10.926
4	4.968	92.515	9.251	22.417	620.051	24.802
5	8.059	209.797	20.979	46.25	2134.483	85.379
6	8.756	281.518	28.151	58.992	3474.215	138.968
7	7.738	183.776	18.377	7.292	52.453	2.098
8	5.138	83.125	8.312	43.077	2416.561	96.662
9	5.045	161.714	16.171	16.349	265.671	10.626
10	5.188	139.725	13.972	7.865	61.082	2.443

Figure 9: EGA and PSO Iteration Results

6.1 Execution Time

Execution time is the total amount of time the datacenter needs to process all of the tasks or cloudlets assigned to virtual machines. Figure 10 clearly depicts the time taken by the EGA is less compared to PSO. The reason is because the newly mutated VM generated by EGA executes cloudlets faster than the VM generated by PSO.

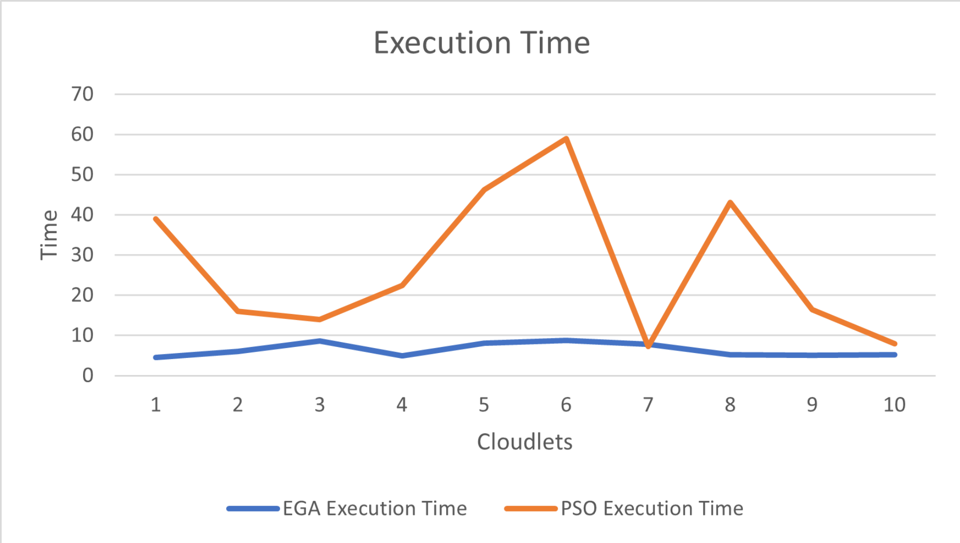

Figure 10: Comparision of execution time

6.2 Resource Utilization

It is the number of resources used to execute the tasks or cloudlets. The calculation of this can be achieved by multiplying the overall execution time of cloudlets with the CPU usage time. As shown in graph Figure 11, it can be said that EGA requires less resources compared to PSO.

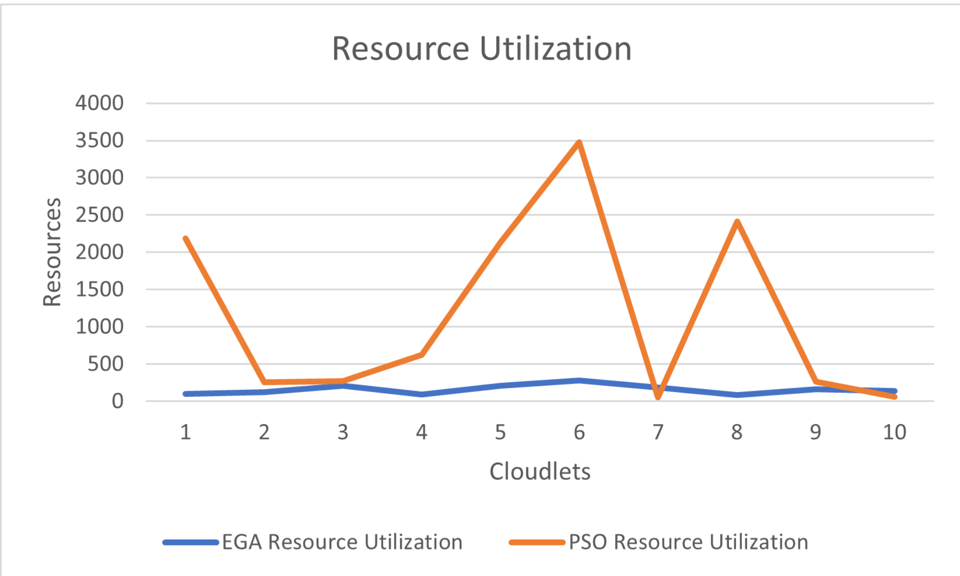

Figure 11: Comparison of resource utilization

6.3 Energy Consumption

Energy consumption is the energy used to to execute overall tasks or cloudlets. Resource allocation over time can be used as a formula to calculate energy consumption. This simulation uses joules metric to calculate the energy. As shown in graph figurename 12, EGA uses significantly less energy when compared to PSO to execute the cloudlets.Therefore it can be said that EGA is more energy efficient than PSO.

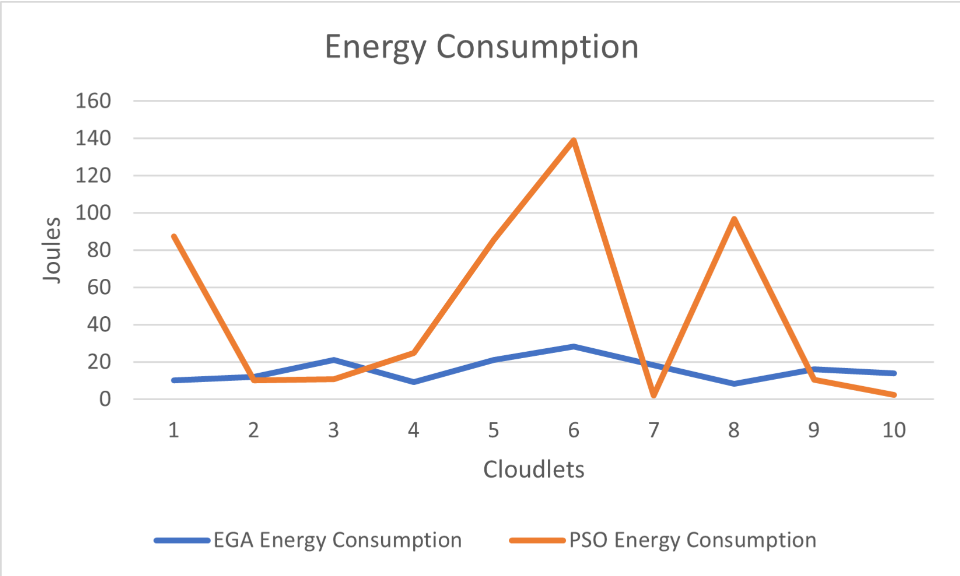

Figure 12: Comparison of energy consumption

6.4 Discussion

As observed in Figure 10, Figure 11, and Figure 12 the graphs show difference in the stability between PSO and EGA. There are sharp peaks and downfall in case of PSO, where as EGA is more stable. This is because of the dissimilarities between their operations. In PSO, particles are constantly moving in order to explore and exploit the solution space differently in each iteration. This behaviour along with dynamic workload in the simulation leads to peaks and sharp fluctuations in each run. Where as in EGA the probabilistic nature and straight forward approach to explore the solution space leads to more stable performance across the different iterations, because it relies on genetic operations like crossover and mutation which are probabilistic in nature.

The average result of iterations performed to compare the EGA and PSO clearly shows that EGA is definitely more efficient when compared to PSO for load balancing in cloud computing environments. The calculation of average based on the data in Figure 9 shows that EGA is 77% more efficient that PSO. This is because EGA uses less resources to execute the tasks which in turn reduces the execution time and energy consumption. This was possible because of the newly mutated VM in EGA that gets generated, it assigns a VM with a configuration that can handle load efficiently. The performance can further be improved in future by fine tuning parameters such as mutation rate or cross over probability. These parameters play a crucial role in shaping the algorithms behaviour and performance. Strategically adjusting these parameters can significantly optimize algorithm's ability to choose a VM that is more efficient, which in turn improves the load balancing.

7 Conclusion and Future Work

In conclusion, the goal of the research was to find out if the Enhanced Genetic Algorithm can optimize the load balancing with dynamic workloads in cloud computing environments. A simulation setup was created to implement this technique and assess the performance with another popular load balancing technique called PSO. Outputs like energy consumption, resource utilization, and execution time were captured through several iterations and the results showed that EGA is more efficient than PSO. Since EGA technique works in several iterations until a solution to a problem is found, it can offer better scalability and flexibility to handle varying workloads, which in turn will improve service quality and will result in smoother operations and better user experience. However, the short come of this research is that it relies on a simulation, the performance can differ when it is implemented in the real world. Also, it is compared with just one algorithm, comparing it with wider range of algorithms will provide more understanding.

There is a lot of scope to improve and add on, like developing a hybrid algorithm, where EGA can be combined with machine learning techniques or other algorithms which can potentially improve the results. Also privacy and security concerns associated with load balancing could be addressed in future.

References

Arunarani, A., Manjula, D. and Sugumaran, V. (2019). Task scheduling techniques in cloud computing: A literature survey, *Future Generation Computer Systems* **91**: 407–415.
URL: *https://www.sciencedirect.com/science/article/pii/S0167739X17321519*

Ezugwu, A., Buhari, S. and Junaidu, S. (2013). Virtual machine allocation in cloud computing environment, *International Journal of Cloud Applications and Computing (IJCAC)* **3**: 47–60.

Goldberg, R. P. (1974). Survey of virtual machine research, *Computer* **7**(6): 34–45.

Goyal, T., Singh, A. and Agrawal, A. (2012). Cloudsim: simulator for cloud computing infrastructure and modeling, *Procedia Engineering* **38**: 3566–3572. INTERNATIONAL CONFERENCE ON MODELLING OPTIMIZATION AND COMPUTING.
URL: *https://www.sciencedirect.com/science/article/pii/S1877705812023259*

Greene, W. (2001). Dynamic load-balancing via a genetic algorithm, pp. 121 – 128.

Gulbaz, R., Siddiqui, A. B., Anjum, N., Alotaibi, A. A., Althobaiti, T. and Ramzan, N. (2021). Balancer genetic algorithm—a novel task scheduling optimization approach in cloud computing, *Applied Sciences* **11**(14).
URL: *https://www.mdpi.com/2076-3417/11/14/6244*

Hai, T., Zhou, J., Jawawi, D., Wang, D., Oduah, U., Biamba, C. and Jain, S. K. (2023). Task scheduling in cloud environment: Optimization, security prioritization, and processor selection schemes, *Journal of Cloud Computing* **12**(1): 15.
URL: *https://doi.org/10.1186/s13677-022-00374-7*

Hashemi, M., Javaheri, D., Sabbagh, P., Arandian, B. and Abnoosian, K. (2021). A multi-objective method for virtual machines allocation in cloud data centres using an improved grey wolf optimization algorithm, *IET Communications* **15**(18): 2342–2353.

Joshi, S. and Kumari, U. (2016). Load balancing in cloud computing: Challenges issues, *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, pp. 120–125.

Kumain, K. (2020). Optimization techniques for resource allocation in cloud computing systems, *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* **11**(3): 1966–1974.

Lagwal, M. and Bhardwaj, N. (2017). Load balancing in cloud computing using genetic algorithm, pp. 560–565.

Lambora, A., Gupta, K. and Chopra, K. (2019). Genetic algorithm- a literature review, *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pp. 380–384.

Lopez-Pires, F. and Baran, B. (2015). Virtual machine placement literature review, *arXiv preprint arXiv:1506.01509* .

Mahmood, I., M.Sadeeq, M., Zeebaree, S., Shukur, H., Jacksi, K., Yasin, H., Radie, A. and Najat, Z. (2021). Task scheduling algorithms in cloud computing: A review, *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* **12**: 1041–1053.

Makasarwala, H. A. and Hazari, P. (2016). Using genetic algorithm for load balancing in cloud computing, *2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pp. 1–6.

Manzoor, M. F., Abid, A., Farooq, S., Ahmed, N. and Farooq, U. (2020). Resource allocation techniques in cloud computing: A review and future directions, *Elektronika ir Elektrotechnika* **26**: 40–51.

Marini, F. and Walczak, B. (2015). Particle swarm optimization (pso). a tutorial, *Chemometrics and Intelligent Laboratory Systems* **149**: 153–165.
URL: *https://www.sciencedirect.com/science/article/pii/S0169743915002117*

Mishra, N. K. and Mishra, N. (2015). Load balancing techniques: need, objectives and major challenges in cloud computing-a systematic review, *International Journal of Computer Applications* **131**(18): 0975–8887.

Naz, I., Naaz, S., Agarwal, P., Alankar, B., Siddiqui, F. and Ali, J. (2023). A genetic algorithm-based virtual machine allocation policy for load balancing using actual asymmetric workload traces, *Symmetry* **15**(5).
URL: *https://www.mdpi.com/2073-8994/15/5/1025*

Raza, M. H., Adenola, A. F., Nafarieh, A. and Robertson, W. (2015). The slow adoption of cloud computing and it workforce, *Procedia Computer Science* **52**: 1114–1119. The 6th International Conference on Ambient Systems, Networks and Technologies (ANT-2015), the 5th International Conference on Sustainable Energy Information Technology (SEIT-2015).
URL: *https://www.sciencedirect.com/science/article/pii/S187705091500928X*

Salvi, R. R. (2022). *Optimizing the load balancing efficiency using enhanced genetic algorithm in cloud computing*, PhD thesis, Dublin, National College of Ireland.

Shafiq, D. A., Jhanjhi, N. and Abdullah, A. (2022). Load balancing techniques in cloud computing environment: A review, *Journal of King Saud University - Computer and Information Sciences* **34**(7): 3910–3933.
URL: *https://www.sciencedirect.com/science/article/pii/S131915782100046X*

Shi, F. and Lin, J. (2022). Virtual machine resource allocation optimization in cloud computing based on multiobjective genetic algorithm, *Computational Intelligence and Neuroscience* **2022**: 7873131.

Ullah, A., Nawi, N. M. and Ouhame, S. (2022). Recent advancement in vm task allocation system for cloud computing: Review from 2015 to 2021, *Artificial Intelligence Review* **55**(3): 2529–2573.
URL: *https://doi.org/10.1007/s10462-021-10071-7*

Wang, T., Liu, Z., Chen, Y., Xu, Y. and Dai, X. (2014). Load balancing task scheduling based on genetic algorithm in cloud computing, pp. 146–152.