SMART GRID: BIO-INSPIRED ALGORITHMS ENERGY DISTRIBUTIONS FOR DATA CENTRES

BY

WOO YU HANG

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

JUNE 2025

COPYRIGHT STATEMENT

ACKNOWLEDGEMENTS

I would like to express my heartfelt gratitude to my supervisor, Dr. Aun Yichiet, for his valuable guidance, support, and encouragement throughout this research project on power management methods based on bio-inspired algorithms. His insights and advice have been instrumental in helping me complete this work. I sincerely appreciate his support and guidance.

I would also like to extend my deepest thanks to my family for their unwavering support, love, and encouragement during this journey. Their belief in me is my constant source of strength and motivation. I am grateful to have them by my side throughout all the challenges.

ABSTRACT

The growing demand for data centre services has led to significant increases in data centre power consumption, highlighting the need for efficient power management strategies to ensure sustainable and energy-efficient operations. Virtualisation technology enables multiple virtual machines (VMs) to run on a single physical server, improving resource sharing and utilisation. However, it also introduces challenges in optimising VM placement and migration to minimise power consumption while maintaining performance. This project proposes and evaluates three bio-inspired and evolutionary algorithms for VM allocation and migration: Ant Colony Optimisation (ACO), Particle Swarm Optimisation (PSO), and a Modified Genetic Algorithm (MGA). These algorithms aim to reduce power consumption, improve resource utilisation, and enhance overall data centre efficiency. The system is implemented and simulated using the CloudSim Plus framework under both homogeneous and heterogeneous data centre environments. Four different workload scenarios were tested, and the performance of the three algorithms was compared against the data centre's baseline VM allocation policy. Each scenario was executed 30 times to ensure the reliability and consistency of results. Simulation results demonstrate that all three proposed algorithms consistently achieved lower total power consumption across all servers compared to the baseline policy. These findings highlight the potential of bio-inspired VM allocation and migration strategies for improving energy efficiency and resource optimisation in modern data centres.

Area of Study (Minimum 1 and Maximum 2): Cloud Computing, Power Management

Keywords (Minimum 5 and Maximum 10): Data Centre, Cloud Computing, Virtual Machine Placement, Virtual Machine Migration, Ant Colony Optimisation (ACO), Particle Swarm Optimisation (PSO), Modified Genetic Algorithm (MGA), Bio-inspired Algorithms, Power Consumption, Resource Utilisation

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

α	alpha (Importance factor of pheromone trail in ACO)
β	beta (Importance factor of heuristic information in ACO)
ρ	Pheromone evaporation rate
τ	Pheromone trail/level
η	Heuristic information
k	Power consumption fraction
q_0	Exploitation parameter
M	Migration plan
c_1	Cognitive acceleration coefficient
c_2	Social acceleration coefficient
r_1, r_2	Random scalars in the range [0, 1]
w	Inertia weight (Particle Swarm Optimisation)
$W1$	Weight assigned to the number of underutilized hosts in the fitness function. (Modified Genetic Algorithm)
$W2$	Weight assigned to the number of overutilized hosts in the fitness function. (Modified Genetic Algorithm)
ε	Utilisation threshold adjustment factor

LIST OF ABBREVIATIONS

ACO	Ant Colony Optimisation
ACS	Ant Colony System
AI	Artificial Intelligence
AMD	Average Median Deviation (AMD)
CRAC	Computer Room Air Conditioning
CSA	Cuckoo Search Algorithm
CPU	Central Processing Unit
DVFS	Dynamic Voltage Frequency Scaling
ELM	Extreme Learning Machine
ELM_MPACS	Multi-population Ant Colony System Algorithm with the Extreme Learning Machine (ELM) prediction
ESWCT	Energy aware Scheduling using the Workload-aware Consolidation Technique
FCFS	First-Come-First-Served
FFA-SA	Fuzzy Hybrid Firefly Algorithm based on Simulated Annealing
FFD	First-fit Decreasing
FPO	Flower Pollination Optimisation
GA	Genetic Algorithm
GB	Gigabyte
GHG	Greenhouse gas
HGA	Hybrid Genetic Algorithm
IEA	International Energy Agency
IDE	Integrated Development Environment
LACE	Locust-inspired scheduling Algorithm to reduce energy consumption in Cloud datacenters
LLM	Large Language Model
Mbps	Megabits Per Second
MIPS	Million Instructions Per Second
MGA	Modified Genetic Algorithm
MM	Minimisation Algorithm

MW	Megawatt
NSGA-II	Nondominated Sorting technique-based Genetic Algorithm
OEM	Order Exchange and Migration
OH_BAC	Osmotic Hybrid Artificial Bee and Ant Colony Optimization Load Balancing Algorithm
PM	Physical Machine
PSO	Particle Swarm Optimisation
QoS	Quality of Service
RAM	Random Access Memory
RGGA	Reordering Grouping Genetic Algorithm
SLA	Service Level Agreement
SSD	Solid State Drive
stdev	Standard Deviation
TB	Terabyte
ThrMu	Threshold with Minimum Utilization policy
VM	Virtual Machine
VMP	Virtual Machine Placement

Chapter 1
Introduction

This chapter provides an overview of the project, including its background, motivation, and objectives. It is structured as follows: Section 1.1 presents the project background, Section 1.2 describes the problem statement and motivation, Section 1.3 outlines the research objectives, Section 1.4 defines the project scope and directions, and Section 1.5 highlights the key contributions of this project.

1.1 Project Background

Data centres are specialized facilities designed to house computer systems and associated components such as telecommunications and storage systems. They provide the essential infrastructure required for storing, managing and processing vast amounts of data, which is essential for businesses, government agencies and cloud service providers. To ensure uninterrupted and optimal operations, data centres are equipped with sophisticated cooling systems, redundant power supplies and physical security measures such as biometric access controls and video surveillance [1]. These facilities can vary in size, from small server rooms to extensive complexes.

Data centres play a crucial role in supporting various online services, including cloud computing, big data analytics, and the hosting of websites and applications. Since the 2020s, data centres have rapidly evolved to meet the demands of modern businesses. With the rise of big data, large language models, GPT and the Internet of Things (IoT), larger, more efficient and scalable data centres are needed to handle real-time data processing demands [1]. Data centres are required to offer high speed connectivity, greater storage capacity, and improved computational power to support these innovations. However, this rapid expansion of data centres has also led to higher energy consumption.

Data centre energy consumption is a significant concern due to the high-power demands of its cooling systems, servers, and other infrastructure to keep operations running smoothly. Electricity consumption in data centres primarily stems from two processes, namely the computing process which represents about 40% of the total

electricity demand, and the cooling process which accounts for another 40% to maintain stable processing efficiency [2]. The remaining 20% of electricity is used by other related IT equipment [2]. As data centres evolve in size and capacity to handle more data, their energy consumption has also increased, making them one of largest consumers of electricity globally. A report by the International Energy Agency (IEA) revealed that data centres, cryptocurrencies and artificial intelligence (AI) consumed 460TWh of electricity worldwide in the year 2022, which is almost 2% of total global electricity demand [2]. Moreover, it is estimated that electricity demand from data centres could double in many countries by 2026. This includes the United States, which represents 33% of global data centres, and China, which accounts for 10% [2]. Figure 1-1 below shows the trend of global electricity demands from data centres, AI and cryptocurrencies from 2019 to 2026.

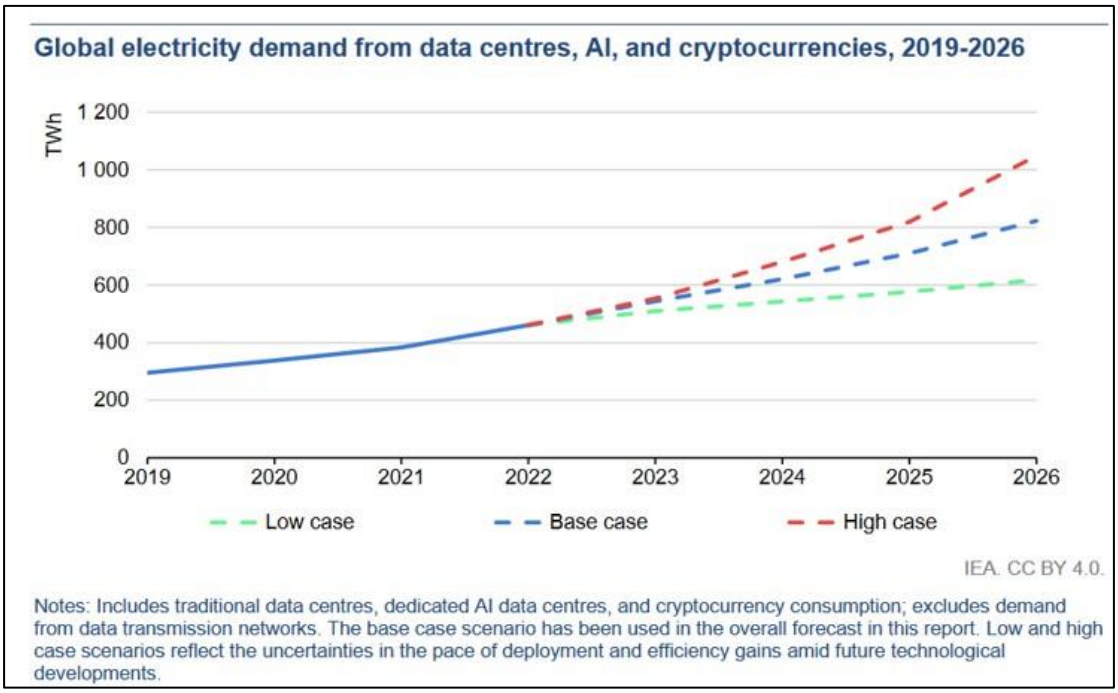

Figure 1-1: Global electricity demands from data centres, AI and cryptocurrencies from 2019 to 2026 [2].

The same situation also applies to ASEAN countries, including Malaysia, where a considerable number of data centres are either under construction or planned, with additional new deployments anticipated in the coming years as shown in Figure 1-2 below. The fast-growing data centre market in Southeast Asia not only brings

opportunities such as increased employment and economic growth, but also various challenges related to energy consumption, operational efficiency, and sustainability. Tackling these challenges will require creating advanced power management solutions and embracing more sustainable technologies to ensure that the growth of data centres is both economically beneficial and environmentally sustainable.

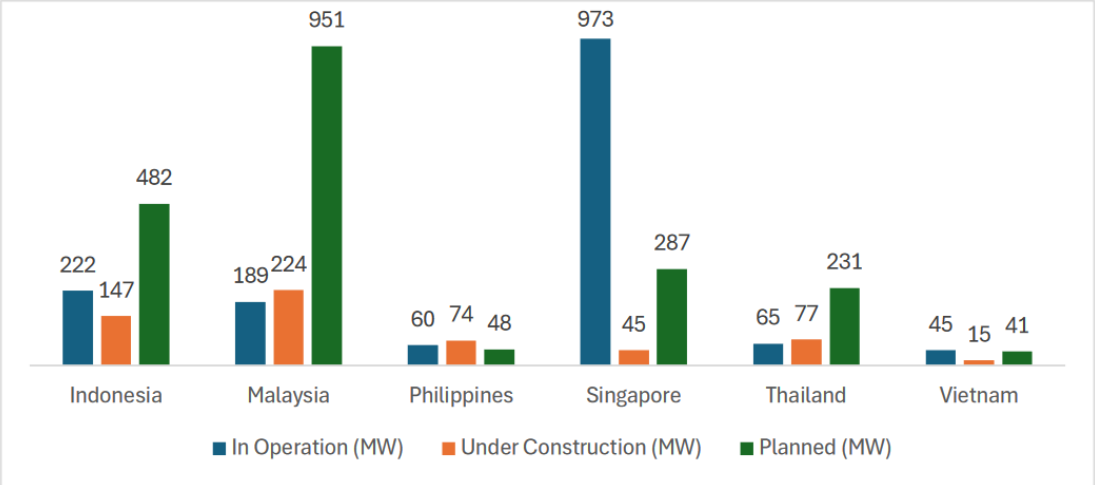

Figure 1-2: Data centre capacity (in Megawatts) in selected ASEAN countries for the year 2024 [1].

To drive energy efficiency in data centres, virtualisation technology is adopted as the main strategy to optimise resource utilisation and reduce energy consumption. It involves creating Virtual Machines (VMs) based on the user's chosen operating system and specified resource needs and running on physical servers to host applications [3]. The advantage of virtualisation is improved utilization of hardware as it allows the dynamic sharing of physical resources and resource pools such as CPU, memory and storage [4]. This sharing of resources enables the consolidation of VMs, where VMs are migrated or allocated into the minimum number of physical servers. With this, inactive servers with no workload can be switched off to help lower the energy consumption of data centres.

The consolidation of VMs from underutilised servers to others with higher utilisation enables the underutilised server to be shut down, thereby efficiently reducing the power consumption in data centres. However, it also introduced a computational problem called the Virtual Machine Placement (VMP) problem. As the name suggests, the VMP problem is concerned with optimising the placement of VMs into physical

servers to improve energy efficiency [5]. The main objective of this optimisation is to find a best solution which places all the VMs into a minimum number of physical servers. The VMP problem is known to be an NP-hard problem, which means that computationally complex and hard to solve efficiently [3], [6], [5]. Traditional algorithms are often impractical and consume high computational time when it comes to solving complex problems like the VMP problem. As a result, various bio-inspired algorithms have been introduced to tackle these NP-hard problems.

In addition to consolidation, another strategy used to address the VMP problem is load balancing. It focuses on distributing workloads evenly across available physical servers to avoid the overloading or underloading of any single server [7]. A balanced resource usage across servers not only improves performance but also contributes to energy efficiency by minimising resource wastage and reducing the number of idle servers that remain powered on without doing useful work. Like server consolidation, bio-inspired algorithms have been applied to load balancing strategies due to their ability to handle complex and dynamic environments effectively.

In this paper, three bio-inspired algorithms based on Ant Colony Optimisation (ACO), Particle Swarm Optimisation (PSO) and Modified Genetic Algorithm (MGA) are proposed to allocate and migrate VMs to minimum number of physical servers to reduce the energy consumption of data centres. The proposed algorithms mainly perform server consolidation based on CPU utilisation. The algorithms are compared to the data centre's baseline VM allocation policy to evaluate their effectiveness in reducing power consumption.

1.2 Problem Statement and Motivation

The rapid growth in data centre demands has led to a significant increase in energy consumption. The increase in energy consumption of data centres can cause several key challenges. From an economic perspective, the energy consumption of an average data centre is as much as 25,000 households and it is expected to double every five years [8]. This increase in energy consumption has resulted in escalating operational costs, with power bills becoming one of the significant expenses for data centre operators [8]. Furthermore, high data centre energy consumption also leads to several environmental challenges. The International Energy Agency has estimated that

data centres and data transmission networks are responsible for around 1% of energy-related greenhouse gas (GHG) emissions [9]. On the other hand, Google has revealed in their 2024 environmental report that their total GHG emissions have increased by 48% over 5 years, mainly due to increases in data centre energy consumption [10]. Lastly, data centre servers consume a considerable amount of energy even when operating in idle mode. Significant energy savings can be achieved by shutting down these idle servers [8]. While virtualisation has allowed for better resource sharing through Virtual Machines (VMs), the optimal placement of VMs across physical servers remains a significant challenge. This issue is known as the Virtual Machine placement (VMP) problem, which is a computationally complex NP-hard problem. The problem becomes increasingly difficult as the scale of data centres increases.

There are many past studies that have tried to tackle the issue of high power consumption of data centres. Various methods have been proposed to manage power consumption in data centres. They can be broadly classified into server-consolidation-based approaches, workload management or task scheduling techniques, and thermal-aware power management techniques [11]. However, several gaps remain in existing approaches. Many fail to evaluate their effectiveness under varying workload conditions, limiting their real-world applicability. Others focus on optimising a single resource, typically CPU while overlooking other critical resources such as RAM. Additionally, some methods are not context-aware and fail to consider how resource demands change and depend on each other. As a result, these systems struggle to adapt efficiently to real-time fluctuations in workload and resource availability.

To address these limitations, this project explores three context-aware bio-inspired algorithms based on Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), and Modified Genetic Algorithm (MGA), designed to optimise power consumption through efficient VM placement and migration. Unlike traditional approaches, these methods dynamically adapt to varying workload intensities by considering both CPU and RAM requirements simultaneously. By combining these diverse optimisation strategies, the project aims to enhance resource utilisation and minimise energy consumption in data centres.

1.3 Research Objectives

1. Design a simulation platform for data centre environments.

- Develop a simulation platform to evaluate power management strategies in cloud data centres, leveraging the CloudSim Plus framework for modelling components and metrics.
- Utilise built-in CloudSim Plus classes for simulating physical servers, VMs, workloads, and network environments, while configuring them to suit both homogeneous and heterogeneous setups.
- Integrate essential performance metrics such as CPU utilisation, RAM utilisation, and power consumption to enable detailed evaluation.
- Incorporate multiple workload scenarios with progressively increasing numbers of VMs and cloudlets to evaluate algorithm adaptability under low, medium, and high resource demands.

2. Deploy Bio-Inspired Algorithms to Optimise Power Management

- Deploy and evaluate three bio-inspired metaheuristic algorithms: Ant Colony Optimisation (ACO), Particle Swarm Optimisation (PSO), and a Modified Genetic Algorithm (MGA) for VM allocation and migration in both homogeneous and heterogeneous data centre environments.
- Apply these algorithms across multiple workload scenarios to optimise power consumption by consolidating VMs, balancing workloads, and reducing idle energy waste.
- Assess algorithm performance against the data centre's baseline VM allocation policy, focusing on improvements in energy efficiency, resource utilisation, and overall operational performance.

1.4 Project Scope and Directions

This project focuses on the development of novel power management methods that involve server consolidation to enhance pod-to-pod power delivery in data centres to reduce power consumption. The solution involves designing and evaluating bio-inspired algorithms, such as Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), and Modified Genetic Algorithm (MGA), to optimise Virtual Machine Placement (VMP) and migration. The reason bio-inspired algorithms are chosen for this project is due to their effectiveness in solving optimisation problems, which is also encountered in server consolidation. By consolidating workloads onto a minimum number of active servers, the approach seeks to maximise resource utilisation while minimising overall energy usage.

The proposed methods consider two key resource dimensions: CPU and memory utilisation. ACO and PSO dynamically adapt to varying workload intensities by optimising VM allocation based on both CPU and RAM, whereas MGA focuses primarily on CPU utilisation to detect underutilised and overutilised hosts. This combined optimisation strategy ensures efficient workload distribution, balanced resource usage, and improved VM consolidation, ultimately reducing the number of active servers.

To evaluate the effectiveness of these algorithms, the project uses simulation-based testing to measure improvements in power consumption, resource utilisation, and overall system performance. The evaluation is conducted across two data centre setups: a homogeneous environment, where all servers share identical configurations, and a heterogeneous environment, where servers have diverse hardware capabilities. Within each setup, the algorithms are tested under four distinct scenarios that vary in the number of virtual machines (VMs) and cloudlets (workloads) to assess performance under different resource demand levels.

For both setups, the proposed algorithms (ACO, PSO, and MGA) are benchmarked against the data centre's baseline VM allocation method to evaluate their effectiveness. Key performance metrics include total power consumption, average power consumption and resource (CPU and RAM) utilisation. By testing across multiple configurations and workload intensities, the evaluation provides a comprehensive assessment of the algorithms' robustness, adaptability, and energy-saving capabilities in diverse operational environments.

1.5 Contributions

The main contributions of this project are highlighted as follows:

- **Integration of bio-inspired algorithms for power management:** The project implements and evaluates Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), and a Modified Genetic Algorithm (MGA) to optimise VM placement and server consolidation in data centres.
- **Energy-efficient and context-aware VM allocation:** The proposed methods dynamically allocate and migrate VMs by considering CPU and RAM utilisation (for ACO and PSO) and CPU-based host status (for MGA), aiming to reduce power consumption while improving resource utilisation.
- **Comprehensive evaluation across multiple scenarios:** The algorithms are tested on both homogeneous and heterogeneous data centre setups under four different workload scenarios, and their performance is benchmarked against the baseline VM allocation policy to validate effectiveness in diverse operational conditions.

1.6 Report Organization

The report is structured into five chapter. Chapter 2 reviews related work on bio-inspired algorithms and data centre power management methods. Chapter 3 describes the proposed Ant Colony Optimisation (ACO)-based VM allocation and migration approach. Chapter 4 presents the preliminary work and simulation results. Chapter 5 concludes the report and provide directions for future research.

Chapter 2

Literature Review

This chapter will present a comprehensive review of existing works and literatures related to methods to reduce power consumption in data centres that involve bio-inspired algorithms. It is structured as follows: Section 2.1 presents the concept of bio-inspired algorithms, Section 2.2 discusses existing methods for Server Consolidation, Section 2.3 reviews existing Workload Balancing/Task Scheduling methods, Section 2.4 reviews existing Thermal-aware Power Management Techniques.

2.1 Bio-inspired Algorithms

As we move further into the digital age, the surge in data volume has made it increasingly difficult to extract valuable insights and knowledge using conventional algorithms due to the growing complexity of analysis. Identifying the best solutions has become increasingly difficult, if not impossible, given the vast and dynamic range of potential solutions and the computational complexity involved [12]. This is especially true for NP-hard problems, where identifying the optimal solution is computationally expensive or even infeasible within a limited timeframe [12], as there are no efficient algorithms to solve them [13]. Therefore, many of the problems have to be solved using trial-by-error approach using different optimisation techniques [13]. This is where bio-inspired algorithms offer a promising and innovative approach to address these challenges.

Bio-inspired algorithms are computational methods that draw inspiration from natural processes and biological systems to solve complex problems. In general, bio-inspired algorithms are widely classified into few categories, with the two most widely recognised categories being evolutionary-based algorithms inspired by the natural evolution process and swarm-based/swarm-intelligence algorithms inspired by animals' collective behaviour [14]. Other categories include ecology-based [14], multi-objective [14], and physics and chemistry-based [13] algorithms.

Evolutionary-based algorithms are optimisation techniques inspired by the principle of natural evolution and biological processes. They imitate the mechanisms of biological evolution to find optimal or near-optimal solutions to complex problems. Example of evolutionary-based algorithms include artificial neural network, genetic

algorithm, evolution strategies, differential evolution and paddy field algorithm [14]. Among them, one of the most popular method used to address data centres' energy consumption problem is the genetic algorithm (GA). GA is inspired by Darwin's theory of natural selection, whereby it uses several nature-inspired operators to evolve a population of potential solutions over generations [12]. Its key components involve a fitness function, which evaluates how well a solution solves the problem to guide the selection process, and multiple operators such as inheritance, crossover, reproduction and mutation [12]. These operators are used to develop "child" solutions from a selected pair of pre-optimised "parent" solutions that retain positive characteristics of the "parent" while reducing the less positive characteristics [12].

On the other hand, swarm-based/swarm-intelligence algorithms are inspired by the collective behaviour of social insects and animals, which uses multiple agents to solve optimisation problems. Some popular example of swarm-based/swarm-intelligence algorithms are the Ant Colony Optimisation (ACO) algorithm that uses social interactions of ants, Particle Swarm Optimisation (PSO) algorithm that mimics swarming behaviour of fish and birds, Cuckoo Search algorithm (CSA) models the brooding paratism of cuckoo species, Firefly algorithm inspired by the flashing behaviour of swarming fireflies and so on [13]. These swarm-based/swarm-intelligence algorithms are highly popular and widely adopted due to several key reasons. One such reason is that they involve multiple agents sharing information, which fosters self-organisation, co-evolution and learning over iterations, thereby enhancing their efficiency [13]. Another reason is that these algorithms can easily be parallelized, which makes them suitable for large-scale optimisation tasks to solve complex problems [13].

In short, bio-inspired algorithms provide innovative and adaptive strategies for addressing the energy consumption problem in data centres. Their ability to adaptively search for near-optimal solutions in complex and dynamic environments make them particularly suitable for solving NP-hard problems, where traditional algorithms may fall short. This makes them an invaluable tool for creating more sustainable and efficient data centres. Table 2-1 summarises Section 2.1.

Algorithm	Inspiration Source	Key Features	Common Use in Data Centres
Genetic Algorithm (GA)	Natural Evolution (Darwin's Theory)	Selection, crossover and mutation	Server consolidation, resource optimisation
Ant Colony Optimisation (ACO)	Ant foraging behaviour	Pheromone trails, heuristic search, shortest path finding	Server consolidation, load balancing
Cuckoo Search Algorithm (CSA)	Cuckoo brood parasitism	Levy flights, random nest selection, host nest replacement	VM placement, resource allocation
Particle Swarm Optimisation (PSO)	Swarming of bird/fish	Velocity and position update, global and local variants	VM placement, resource optimisation
Firefly Algorithm (FA)	Firefly light attraction	Attractiveness is proportional to brightness	Resource allocation and scheduling

Table 2-1: Summary of Section 2.1.

2.2 Server consolidation Methods

Server consolidation refers to the process of reducing the number of active physical servers in a data centre by running multiple applications on fewer servers. At the heart of server consolidation is virtualisation technology, which has become increasingly important in enhancing the energy efficiency of data centres [15]. Virtualisation technology enables multiple Virtual Machines (VM) to run on a single physical server, allowing for shared use of hardware resources. As a result, VMs can be consolidated to run on the fewest physical servers required, while unused servers can be shut down to reduce energy consumption and save energy costs [3]. Most methods employing server consolidation involves solving the VM placement (VMP) problem, which is a computational problem that aims to determine the most optimal allocation of VMs onto physical servers [3]. Figure 2-1 and 2-2 shows how server

consolidation can optimize server utilization by allocating VMs to the minimum number of servers required, while shutting down the servers that are unused.

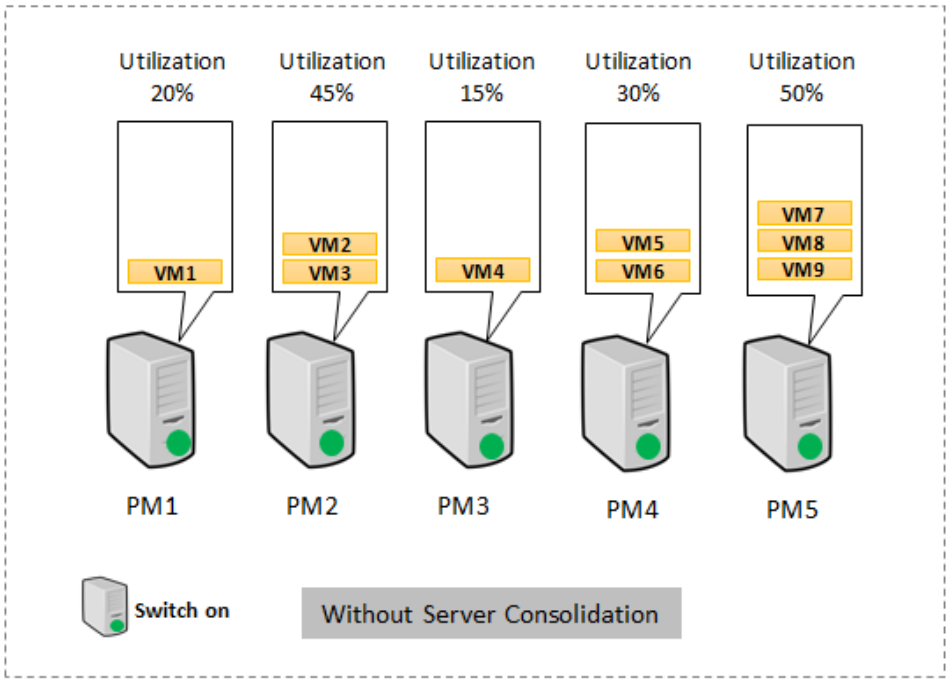

Figure 2-1: Servers utilisation without server consolidation [16].

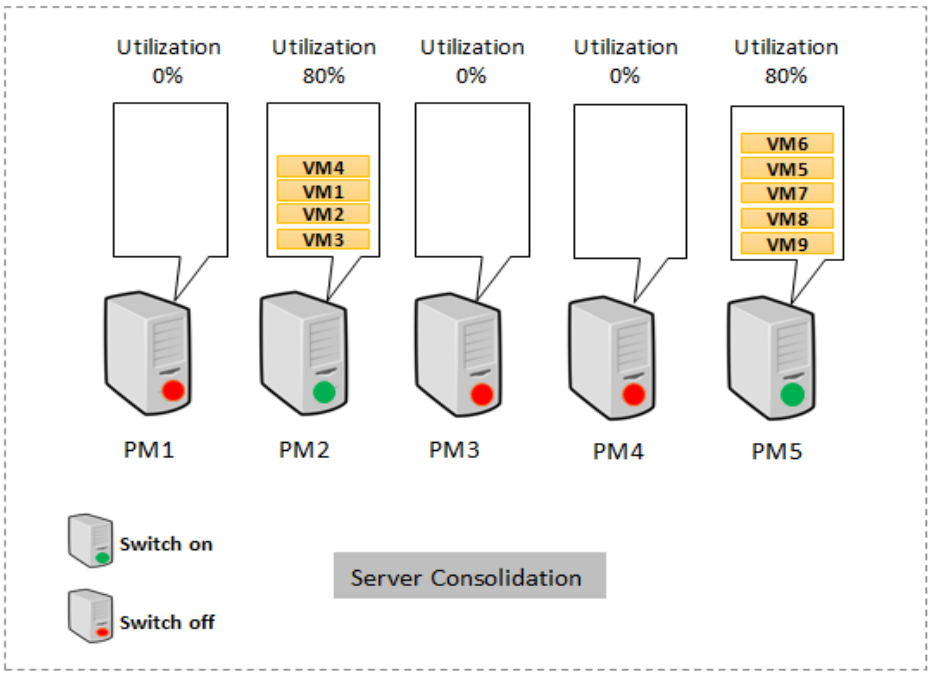

Figure 2-2: Server utilisation with server consolidation [16].

In recent years, researchers have proposed various algorithms to enhance the efficiency of server consolidation. For example, Liu et al. [3] proposed an efficient Ant Colony System (ACS) to solve the VMP problem in cloud computing. ACS is based on Ant Colony Optimisation (ACO) algorithm, which mimics the behaviour of real ants. The ACS in [3] is inspired by real ants' ability to find the shortest paths to food source using pheromones and applies this concept to the VMP problem. By using pheromones to record historical search information and heuristic information to guide decisions, ACS can sequentially assign virtual machines to the most suitable servers, making it an effective method for solving VMP challenges. Moreover, the ACS in [3] is paired with the Order Exchange and Migration (OEM) local search techniques, and the resulting algorithm is termed as an OEMACS. The OEM local search plays a crucial role in converting an infeasible solution into a feasible one. It is applied when the current best solution is found to be infeasible and it involves two steps, an ordering exchange operation followed by a migration operation. These steps aim to adjust the VM assignments to alleviate or eliminate server overloads. The experiment results showed that the OEMACS outperforms conventional heuristic and evolutionary-based approaches such as First-fit Decreasing (FFD), reordering grouping Genetic Algorithm (RGGA) and ACO-based method in minimising the number of active servers and energy consumption and maximising resource utilisation [3]. However, this approach only considers two dimensions of resource usage, which is CPU and RAM requirements. Consequently, the potential energy savings are limited to the power consumed by these two resources alone.

Other than that, Tang and Pan [17] proposed a hybrid genetic algorithm (HGA) to solve the VMP problem in data centres. Unlike other existing VMP approaches, which often overlook power consumption associated with the communication networks within data centres, this method recognises this significant energy usage and factors it into VMP strategies to enhance the overall energy efficiency of data centres. In Genetic Algorithm (GA), potential solutions to a specific problem are encoded as chromosomes within a data structure, and recombination operators are then applied to these chromosomes to evolve towards better solutions over time [16]. The HGA proposed by [17] is based on a previous GA proposed by the Wu et al. [15]. It makes use of the same encoding scheme, fitness function, selection strategy and genetic operators as the original GA. However, it improves the original GA by adding an infeasible solution

repairing procedure and a local optimisation procedure, both of which aim to improve the algorithm's exploitation capacity and ability to converge more effectively. The approach in [17] starts by generating solutions with a Genetic Algorithm (GA). If the solution violates VMP constraints, it undergoes a repair procedure where VMs are reassigned to other physical machines (PMs) until constraints are satisfied. Feasible solutions are then optimized using the local optimisation procedure to reassign all VMs from a PM to other PMs, allowing the initial PM to be turned off, thereby reducing power consumption. In the evaluations, the HGA in [17] significantly outperforms the original GA in [15] that already have a better performance when compared to the FFD algorithm. However, while the HGA also has a lower mean computation time compared to the original GA, its computation time is still significantly larger compared to other heuristic algorithms.

On the other hand, Kurdi et al. [18] proposed a scheduling algorithm inspired by the behaviour of locusts, known as Locust-inspired scheduling Algorithm to reduce energy consumption in Cloud datacenters (LACE). Locusts demonstrate flexible behaviours that transitions between two phases, the solitary phase and gregarious phase. The LACE algorithm in [18] emulates these behaviours with two phases of its own, the mapping phase and the consolidation/migration phase. During the mapping phase, servers behave like solitary locusts, accepting only unallocated VMs. In contrast, during the consolidation/migration phase, the servers mimic the gregarious behaviour of locusts and aggressively search for VMs, including those on other servers. The algorithm in [18] also classifies servers into heavily-loaded servers and lightly-loaded servers based on their processor utilisation level. It then applies global and local migration rules to always migrate VMs from lightly-loaded servers to heavily-loaded servers. The LACE algorithm is compared against three well-established benchmarks, namely Dynamic Voltage Frequency Scaling (DVFS), Energy aware Scheduling using the Workload-aware Consolidation Technique (ESWCT) and the static Threshold with Minimum Utilization policy (ThrMu) [18]. The LACE algorithm outperforms latter two in resource utilization and energy consumption and performs similarly to DVFS. It excels in energy efficiency across most data center scales, though ThrMu is more efficient in large-scale centers but less reliable. LACE also improves Service Level Agreement (SLA) response times compared to latter two benchmarks, while matching DVFS's performance. However, initial VM allocation in [18] is done in a First-Come-

First-Serve (FCFS) basis to all available servers, which results in inefficient resource usage as the VMs will eventually have to be reallocated to suitable servers afterwards. This can be solved by achieving full utilisation of servers in the mapping phase before progressing to the next phase [19].

Furthermore, Salami et al. [6] tackled the VMP problem by proposing a new Cuckoo Search Algorithm (CSA) termed newCSA. This newCSA is based on the modified CSA of Walton et al. [20]. The CSA, as the name suggests, is inspired by the brood parasitism behaviour of cuckoos, where they lay eggs in other birds' nests. This requires the cuckoos' eggs to evolve to avoid being detected and discarded by the host birds [20]. Generally, the CSA is based on three idealised rules [21]: (1) Each cuckoo lays a single egg at a time and deposits it in a randomly selected nest. (2) The nests with the highest quality eggs are retained for the next generation. (3) The number of available host nests remains constant, and there is a probability $p_a \in [0,1]$ that the host bird will discover the cuckoo's egg. If that is the case, the host bird can either discard the egg or abandon its nest to build an entirely new nest. The newCSA algorithm proposed in [6] introduced a novel cost function for the placement solution, three new perturbation functions used to search the design space and a new, computationally cheap method for updating the cost of solutions. In the CSA, each nest represents a solution that indicates which server host which VM and the best net will be chosen. The newCSA algorithm is tested against the RGGA, FFD, best-fit decreasing (BFD) and a prior CSA-based method termed multiCSA and the results showed that newCSA is better in terms of number of servers required for VM placement, power consumption, and execution time [6]. Nevertheless, one limitation of this method is that it only considers two dimensions/resources, namely memory and CPU. There is an improvement that can be made by including more dimensions/resources in the method.

Moreover, Singh et al. [22] proposed a bio-inspired VMP framework that aims to maximise resource utilisation and minimise power consumption and carbon emissions. It proposes a novel FP-NSO algorithm that combines the concepts of Nondominated Sorting technique-based Genetic Algorithm (NSGA-II) and Flower Pollination Optimisation (FPO). The FPO in [22] generates an initial population of solutions by randomly allocating VMs. Each individual solution, which represents a VM allocation is considered as a flower or pollen. In [22], Random-Fit (RF) and First-Fit (FF) algorithms is used to perform the VM-PM mapping process. After the mapping process,

the FP-NSO utilises the modules in NSGA-II to generate the optimal VM allocations and FPO algorithm to achieve optimal assignment of VMs. In the end, the best flower which is the optimal solution is obtained. The FP-NSO algorithm is evaluated against nine existing approaches, including the original NGSA-II, GA, FF, RF and more. The algorithms are tested on two different scenarios, static VMP and dynamic VMP. In the static VMP scenario, FP-NSO enables resource utilisation up to 69%, surpassing other algorithms by a margin of 9.29% to 67.08% [22]. Evaluation also shows that FP-NSO can significantly reduce the number of active PMs and is one of the best performers in reducing power consumption and carbon emissions among the algorithms. On other hand, FP-NSO achieved a significant reduction in power consumption, execution time, and carbon emission over other algorithms in the dynamic VMP scenario, which is up to 16.69%, 75.87% and 48.60% respectively [22]. It also improves resource utilisation up by 78.18% compared to other algorithms [22]. However, there is still some reliability concern with the FP-NSO algorithm. Further refinements can be made to improve its reliability.

Additionally, Liu et al. [5] proposed a Multi-population Ant Colony System (ACS) Algorithm with the Extreme Learning Machine (ELM) prediction called ELM_MPACS. As the name suggests, the ELM_MPACS uses ELM, which is a single hidden layer feed-forward neural network to predict the state of each host in the data centre. On the other hand, the multi-population ACS algorithm is employed to determine the destination host for VM migration based on the prediction of the ELM. Each population's migration scheme is evaluated using an objective function that considers both power consumption and number of migrations, and the best solution is selected. Like other ACS-based algorithms like [3], the ELM_MPACS in [5] also rely on pheromones and heuristic information to select destination hosts for VMs. Furthermore, the ELM_MPACS utilises local search strategy to improve the migration plan and prevent SLA violations. The VM migration process of the ELM_MPACS is also quite different from other server consolidation algorithms. In ELM_MPACS, VMs on overloaded hosts are moved to normal hosts, while VMs on underloaded hosts are moved to other underloaded hosts with higher utilisation. A constraint is set whereby destination hosts' utilisation must be lower than source hosts' after migration and VMs from underloaded hosts can only be moved to destination hosts with higher utilisation than the source [5]. This approach minimises unnecessary migration and speeds up the

migration process. ELM_MPACS is evaluated against four benchmark algorithms in the CloudSim simulator. Experimental results demonstrate that ELM_MPACS is more effective than those algorithms in reducing data centre power consumption, VM migration time and SLA violations [5]. However, ELM_MPACS only considers CPU utilization during VM migration. The effectiveness of the consolidation technique can be enhanced by incorporating multiple resources in the VM integration process. Table 2-2 summarises Section 2.2.

Work	Objective	Method	Results	Limitation
Liu et al. [3]	Efficiently allocate VMs across the fewest number of PMs to minimise energy consumption for cloud computing.	Ant Colony System (ACS)-based approach paired with Order Exchange and Migration (OEM) local search techniques. (OEMACS)	OEMACS outperforms FFD, RGGA and ACO-based algorithms in terms of average energy consumption and server utilisation.	Only consider two dimensions of resource usage, CPU and RAM requirements.
Tang and Pan [17]	Enhance energy efficiency of data centres by considering the network power consumption in VMP.	Hybrid Genetic Algorithm (HGA) that incorporates infeasible solution repair and local optimisation procedure.	HGA significantly outperforms the original GA and FFD approach in reducing energy consumption.	HGA has a significantly larger computation time compared to other heuristic algorithms.
Kurdi et al. [18]	Reduce energy consumption in cloud data centres.	Locust-inspired scheduling algorithm called LACE with mapping phase and	LACE outperforms ESCWT and ThrMu, while matching DVFS	Initial VM allocation in FCFS basis results in resource wastage as

		consolidation/migration phase.	in resource utilisation and energy consumption, and excels in energy efficiency across different data centre scales.	VMs will have to be reallocated.
Salami et al. [6]	Solve the VMP problem by developing a new Cuckoo Search Algorithm (newCSA).	newCSA algorithm that introduced a novel cost function, three new perturbation functions and an efficient method to update the cost of solutions in VM placement.	newCSA performs better than compared methods in terms of number of servers required, power consumption and execution time.	Only considers two dimensions of resource usage, CPU and memory.
Singh et al. [22]	Maximise resource utilisation, minimise power consumption and carbon emission of data centres.	Bio-inspired VMP framework that uses FP-NSO algorithm that combines Flower Pollination Optimisation (FPO) and Nondominated Sorting technique-based Genetic Algorithm (NSGA-II).	FP-NSO significantly reduces power consumption and carbon emission and improves resource utilisation compared to other algorithms.	The proposed framework has some reliability concerns.
Liu et al. [5]	Improve VM consolidation efficiency and	ELM_MPACS algorithm combining Multi-population Ant	ELM_MPACS algorithm is more effective	Only considers one resource/dimension, which is CPU

	achieve balance between reducing energy consumption and SLA violations.	Colony system (ACS) for VM migration/consolidation with Extreme Learning Machine (ELM) for predicting host states.	in reducing energy consumption, VM migration time, and SLA violations compared to four other benchmark algorithms	utilisation during VM migration

Table 2-2: Summary of Section 2.2.

2.3 Workload Balancing/Task Scheduling Methods

A considerable amount of energy is wasted when energy is distributed across computing nodes and when these nodes handle application workloads [23]. As a result, it is necessary to efficiently distribute workloads between the computing nodes. Load balancing is one of the key methods employed to reduce energy consumption in data centres. It involves distributing the workload evenly between participating nodes and ensuring all nodes share the load equally [23]. Load balancing aims to prevent any single node from being overloaded or underutilised . By redistributing tasks effectively, load balancing not only optimises resource utilisation, but also minimises energy consumption, ultimately leading to lower operational costs and improved system performance.

Load balancing algorithms are techniques used to distribute workloads evenly across multiple computing resources. Some traditional load balancing algorithms include round-robin, least connection, weighted round-robin, IP hash and least response time [7]. Although load balancing shares some similarities with server consolidation such as making real-time decisions on where to place workloads, they are quite different in nature. Load balancing distributes the workloads/tasks evenly among available servers or VMs to ensure optimal performance, while server consolidation involves reallocating workloads/tasks to the fewest possible servers to reduce energy consumption. There have been several studies that implemented bio-inspired algorithms to enhance load balancing strategies.

For instance, Gupta and Deshpande [24] introduced a load balancing technique for cloud data centres that is based on Ant Colony Optimisation (ACO). As mentioned above, ACO is inspired by ants' foraging behaviour. In this technique, servers are treated as nodes and artificial ants utilise two types of pheromones to guide their movements, foraging pheromones (FP) and trailing pheromones (TP). The pheromones are updated depending on the direction of the ants' search. Foraging pheromone is updated when the ants move from an underloaded node to an overloaded node, and trailing pheromone is updated when the opposite occurs. The technique in [24] involves artificial ants that move between nodes to balance the load, guided by pheromone levels. If a node is overloaded, they look for an underloaded neighbor and update the Trailing Pheromone (TP). If the node is underloaded, they search for an overloaded neighbor, updating the Foraging Pheromone (FP). Load balancing only happens when the conditions match; otherwise, the search continues. The load redistribution is carried out based on the proposed redistribution policy, which determines how many requests each node involved should handle. Figure 2-3 below shows the load balancing process when the ant found an overloaded node first.

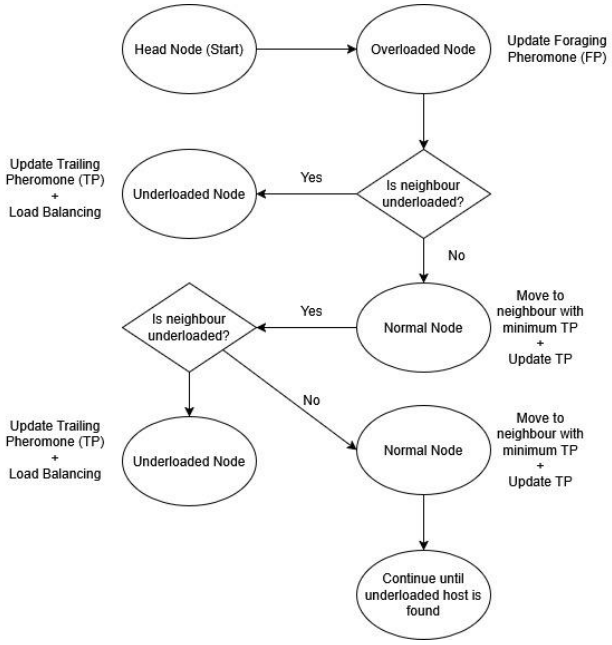

Figure 2-3: ACO load balancing process in [24].

Experimental results showed that the ACO load balancing technique improves the resource utilisation of the nodes and decreases the number of underloaded and overloaded nodes [24]. However, the study did not compare its results with other load

balancing algorithms, which limit the ability to evaluate its effectiveness relative to existing methods. At the same time, the experiment did not address the energy savings achieved by the ACO load balancing technique, which is crucial for evaluating its overall efficiency and effectiveness in reducing energy consumption in data centres.

Furthermore, Das et al. [25] proposed a novel load balancing algorithm by combining Weighted Round Robin algorithm with the Honeybee Inspired load balancing approach. The approach is used to remove and migrate tasks between VMs by considering the priority. It begins by checking for underloaded and overloaded VMs and then removes tasks from VMs with excessive load and checks for priority. The Honeybee inspired load balancing algorithm is responsible for assigning weight to VMs and reallocating non-pre-emptive tasks to underloaded VMs when priority exists. On the other hand, if priority does not exist, Weighted Round Robin algorithm is used to allocate tasks. The hybrid algorithm in [25] assigns weights to VMs based on their capacity and evaluates the load on each machine. If load imbalance is detected, it identifies underloaded and overloaded VMs. High-priority tasks are assigned to appropriate VMs, while lower-priority tasks use a Round Robin policy. Finally, the load on each VM is updated. The algorithm is compared with the Honeybee Inspired algorithm and Weighted Round Robin algorithm using a cloud analyst simulator. The results showed that the average response time and data centre processing time of the hybrid algorithm is faster than both the individual Honeybee Inspired and Weighted Round Robin algorithm [25]. However, the approach does not consider other Quality of Service (QoS) factors such as waiting time, migration time costs and so on. Moreover, like [24], the approach did not measure the energy saved by using the hybrid algorithm.

Additionally, Lawanya Shri et al. [26] developed a load balancing model using a Firefly algorithm to maximise resource utilisation and ensure even distribution of load across all resources in cloud servers. The three idealised rules of the Firefly algorithm include [26]: (1) Any firefly can be attracted to another as they are unisexual. (2) Attractiveness of a firefly is directly proportional to its brightness, where a dimmer firefly is attracted to a brighter one, but the attraction diminishes as distance between them increases. (3) If a firefly cannot find any other firefly brighter than itself, it moves randomly. The brightness of a firefly is determined by the objective function of the algorithm. The proposed approach in [26] is termed as Fuzzy Hybrid Firefly Algorithm

based on Simulated Annealing (FFA-SA). It combines Firefly algorithm with Simulated Annealing optimisation algorithm to enhance optimisation accuracy and convergence speed. During the selection process of VMs, a fuzzy approach is applied to allocate tasks effectively by using fuzzy rules to define control policies for fireflies. In [26], dominant fireflies represent VMs in the data centre, while submissive fireflies represent jobs or tasks assigned to these VMs. If a particular dominant firefly (VM) is overwhelmed with submissive fireflies (tasks), submissive fireflies will be redirected to another dominant firefly to ensure balanced distribution. The FFA-SA is compared against existing algorithms such as Honeybee Behaviour Load Balancing algorithm (HBB-LB), Particle Swarm Optimisation (PSO) and Energy-aware Fruit Fly algorithm (EFOA-LB). FFA-SA outperformed other algorithms in reducing makespan and energy consumption in data centers through effective load balancing [26]. However, the approach did not consider other factors such as resource utilisation and number of overloaded or underloaded servers, which could further enhance the efficiency and performance of the approach.

Moreover, Gamal et al. [27] proposed a hybrid artificial bee and ant colony load balancing algorithm for cloud computing environments named OH_BAC. The algorithm is based on osmotic behaviour, which refers to the way cells or systems respond to the process of osmosis. In [27], VMs migrate from heavily loaded PMs to lightly loaded PMs like water moving from a region of lower solute concentration to higher solute concentration in the osmosis process. The OH_BAC combines key behaviours of Ant Colony Optimisation (ACO) and Artificial Bee Colony (ABC) algorithms, where ACO's rapid solution discovery at diversity systems and ABC's waggle dance for information sharing are integrated. A knowledge base, which is a central resource is used to guide the ABC and ACO in the VM migration process. The process in [27] begins with the ABC component, where a scout bee calculates the standard deviation to identify underutilized and overutilized hosts. Once identified, the employed bee selects a suitable VM for migration, which is then executed by the onlooker bee to a suitable Physical Machine (PM). Concurrently, the ACO component generates a list of osmotic PMs using a knowledge base enhanced with osmosis techniques. The ACO then calculates the fitness function to find the best PM for the selected VM migration. The final migration step ensures that the selected PM is compatible with the osmotic list of hosts from the knowledge base before executing the

migration. The performance OH_BAC is compared in two experiments with fixed and variable loads against ACO, ABC, H_BAC and other host overloading detection algorithms. OH_BAC significantly improved energy consumption, SLA violations, VM migrations, and host shutdowns compared to other algorithms under both fixed and variable loads, although it has a higher Service Level Agreement Time per Active Host (SLATAH) [27]. Table 2-3 summarises Section 2.3.

Work	Objective	Method	Result	Limitation
Gupta and Deshpande [24]	Develop a load balancing technique that improves performance in cloud data centres.	Load balancing technique based on Ant Colony Optimisation (ACO).	ACO-based load balancing improves resource utilisation of servers and decreases the number of overloaded and underloaded servers.	Did not compare performance with other algorithm to evaluate effectiveness. Did not address power savings achieved by the algorithm.
Das et al. [25]	Enhance load balancing in cloud environment.	Hybrid algorithm that combines Honeybee algorithm and Weighted Round Robin algorithm.	Hybrid algorithm demonstrates faster average response time and data centre processing time than both the individual algorithms.	Did not consider other QoS factors such as waiting time and migration time costs. Did not measure energy saved by hybrid algorithm.
Lawanya Shri et al. [26]	Maximise resource utilisation and ensure even distribution of load across all resources in cloud servers.	Fuzzy Hybrid Firefly Algorithm based on Simulated Annealing (FFA-SA)	FFA-SA reduced makespan and energy consumption, outperforming HBB-LB, PSO, and EFOA-LB algorithms.	Did not consider other factors such as resource utilisation and number of overloaded or underloaded servers.
Gamal et al. [27]	Develop a load balancing	Hybrid Artificial Bee	OH_BAC improved energy	OH_BAC algorithm has a

	algorithm to optimize energy usage and system performance in cloud environments.	Colony (ABC) and Ant Colony Optimization (ACO) algorithm based on osmotic behavior (OH_BAC).	consumption, SLA violations, VM migrations, and host shutdowns compared to other methods under fixed and variable loads.	higher Service Level Agreement Time per Active Host (SLATAH) than the compared algorithms.

Table 2-3: Summary for Section 2.3.

2.4 Thermal-aware Power Management Techniques

As mentioned above, much of the energy costs of data centres are associated with the cooling process [2]. Therefore, an effective way to reduce power consumption in data centres is by minimizing the burden placed on cooling systems to maintain the temperature of the computing infrastructure [28]. Thermal-aware power management methods, as the name suggests, are power management methods designed to manage power consumptions in systems while considering their thermal behaviour. These methods are crucial in environments like data centres, where maintaining optimal temperature is essential for the longevity and reliability of hardware components.

For example, Chen et al. [29] proposed a power and thermal-aware VM placement scheme to reduce the power consumptions of data centres. This VM placement scheme, also known as power and thermal-aware VM dynamic scheduling scheme (PTDS) includes a new host load detection algorithm termed Average Median Deviation (AMD), Minimisation Algorithm (MM) for migrating VMs during the VM selection phase and VM placement algorithm based on enhanced Ant Colony Optimisation (ACO) called PTOACO. Unlike other VM placement schemes, PTDS's objective is not only to minimise the energy consumption of computing equipment but also to maintain the temperature control of the hosts to prevent host damage due to high temperature. The system model of PTDS consists of three sub-models [29]: (1) the linear computing system power model, which explains the linear relationship between host's power consumption and change in time. (2) the cooling system power model, which examines the utilisation of cooling energy. (3) the server temperature model that utilises CPU temperature to evaluate the connection between server utilisation, computer room air conditioning (CRAC) cooling capacity and thermal characteristics. These models work

together to define the power and thermal management strategies in PTDS. The PTDS scheme is compared with seven benchmark scheduling schemes to evaluate its effectiveness. They are evaluated on four standard metrics including energy consumption, hotspots, SLA violation, and active hosts. The PTDS scheme achieved the second-lowest average energy consumption and hotspots, ranked fourth in average SLA violations, and had the lowest number of active hosts per hour compared to seven other scheduling schemes [29]. However, the PTACO algorithm is prone to local optimsation, where it converges on suboptimal solutions while potentially ignoring the globally optimal solution.

Furthermore, Yang et al. [30] presented a novel power model to link task assignment, heat recirculation, inlet temperature and cooling costs in homogenous and heterogenous data centres with under-floor air supply. The model comprises four stages. The first stage connects cold air temperature to the power consumption in the data centre, while the second stage represents inlet temperature using power consumption by using an abstract heat recirculation model. The third stage relates inlet temperature to task placement and power profile and the final stage uses peak inlet temperature to determine the maximum temperature of supplied cold air. This model, along with a Genetic Simulated Annealing (GSA) algorithm is used to assign tasks in the data centre. The GSA algorithm in [30] is an enhancement of the traditional Genetic Algorithm (GA), incorporating simulated annealing. Like the approach in [26], simulated annealing conducts single point search using solution transformation and is included in the GSA algorithm to improve the performance of the solution searching process. By combining the parallelization capabilities of GA with the solution transformation and selection mechanisms of simulated annealing, the GSA algorithm significantly lowers the risk of getting trapped in a local optimum during the search process. In the experiments, the proposed approach is compared with the traditional GA and the Ant Colony (AC) algorithm. Results demonstrated that it outperforms GA and AC algorithms in decreasing the temperature requirement of supplied cold air and reducing the power consumption of cooling systems [30]. However, the approach primarily focuses on reducing cooling costs in data centres, overlooking the computing costs. To achieve a more comprehensive solution, both cooling and computing costs should be considered. Table 2-4 summarises Section 2.4.

Work	Objective	Method	Result	Limitation
Chen et al. [29]	Reduce power consumption of data centres and SLA violation rate while preventing hotspots.	Power and thermal-aware VM dynamic scheduling scheme (PTDS) combining Average Median Deviation (AMD), Minimisation Algorithm (MM), and PTOACO (enhanced ACO algorithm).	PTDS achieved second lowest energy consumption and hotspots, ranked fourth in SLA violations, and had the lowest number of active hosts per hour compared to seven other benchmark schemes.	PTACO algorithm is prone to local optimisation, potentially overlooking the global optimal solution.
Yang et al. [30]	Reduce power consumption of cooling systems in data centre by intelligent task assignment.	Power model linking task assignment, heat recirculation, inlet temperature, and cooling costs in homogenous and heterogeneous data centers with under-floor air supply combined with Genetic Simulated Annealing (GSA) algorithm.	GSA algorithm outperformed GA and Ant Colony based algorithm in reducing cooling system power consumption and lower the temperature of cold air.	Approach primarily focused on reducing cooling costs, overlooking computing costs.

Table 2-4: Summary of Section 2.4.

Chapter 3
System Method/Approach

This chapter outlines the methodology used to design and evaluate VM allocation and migration methods in data centres. It begins with the overall design specifications, including the workflow and tools used, followed by the system model that simulates both homogeneous and heterogeneous data centre setups under varying workload scenarios. The chapter then details the implementation of three bio-inspired optimisation algorithms, explains their core principles and demonstrating the complete VM allocation and migration processes aimed at improving power efficiency and resource utilisation.

3.1 Design Specifications

3.1.1 General Work Procedure

This project employs a simulation-based development methodology to design and evaluate the Virtual Machine (VM) allocation and migration algorithms based on Ant Colony Optimisation (ACO), Particle Swarm Optimisation (PSO), and Modified Genetic Algorithm (MGA). This approach involves using CloudSim Plus, a cloud simulation framework, to model and simulate data centre behaviour under different workloads. The general work procedure includes problem formulation, development environment setup, system modelling, algorithm implementation, testing through simulation, performance evaluation, and documentation.

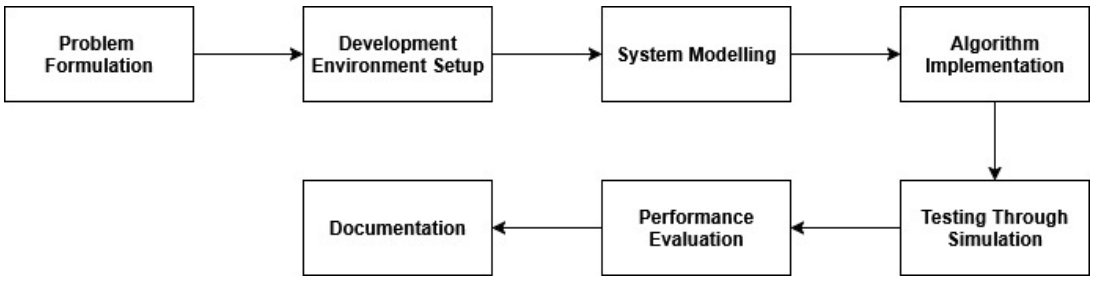

Figure 3-1: General Work Procedure of the Project.

Problem formulation phase defines the research problem and objectives. It involves identifying issues related to VM allocation and migration in data centre environments

and establishing key performance indicators such as power consumption and resource utilisation. Next, the development environment is set up by installing and configuring CloudSim Plus within Eclipse Integrated Development Environment (IDE), along with the required dependencies using Apache Maven. Once the environment is ready, system modelling is carried out by designing a simulated cloud infrastructure that includes a data centre, a data centre broker, physical hosts/servers, VMs and cloudlets that represent user workloads. Following this, the implementation phase involves developing three custom VM allocation and migration algorithms. They are the ACO-based algorithm that mimics the foraging behaviour of ants using pheromone trails and heuristic information to make VM placement and migration decisions, the PSO-based algorithm inspired by swarm intelligence and uses particles to explore the solution space and iteratively adjust VM placement based on personal best and global best positions and Modified Genetic Algorithm (MGA) which applies evolutionary principles with a problem-specific crossover strategy to optimise VM placement by migrating VMs from overutilised or underutilised hosts while prioritising resource efficiency. After that, simulation and testing are carried out to verify and evaluate the performance of the algorithm. After implementation, simulation and testing are conducted to verify and evaluate the performance of the algorithms. Multiple test cases are created with varying workloads, VM sizes, and server configurations under both homogeneous and heterogeneous data centre setups. Finally, all aspects of the project will be compiled into a report to conclude the project work.

3.1.2 Tools to use

This section shows the tools/software used for this project.

- **CloudSim Plus 8.5.5:**

CloudSim Plus is a Java-based cloud simulation framework that enables the modelling and simulation of data centre and cloud computing environments [31]. It is a modified version derived from the original CloudSim simulation tool. It is selected as the simulation tool for this project because it offers powerful, flexible and modern features beyond the original CloudSim framework. For example, CloudSim Plus provides interfaces and classes for implementing heuristic algorithms such as Ant Colony Systems, Simulated Annealing and Tabu Search [31]. It also features precise power

consumption calculations as well as built-in calculations of CPU utilisation history and energy consumptions for both VMs and hosts [31]. The CloudSim Plus framework also supports dynamic and realistic workloads. It enables the dynamic creation of VM and cloudlets at runtime, along with the delayed submission of created VMs and cloudlets [31]. This allows realistic and dynamic workload modelling in data centre environments. Given these features, CloudSim Plus provides a powerful platform for developing and validating the proposed power management methods.

- **Eclipse IDE:**

Eclipse IDE is a free, open-source Java-based integrated development environment (IDE) primarily used for Java development. It offers a platform for creating, debugging and testing applications by offering a wide range of tools and plugins. Since CloudSim Plus framework is Java-based, Eclipse is chosen to develop and simulate cloud-computing scenarios using the framework. Furthermore, Eclipse also offers robust support for Maven projects, which simplifies dependency management and project configuration. Therefore, it makes it easier to integrate and manage the CloudSim Plus libraries and other required components in the simulation environment.

- **Apache Maven:**

Maven is a software project management and build automation tool. It is mainly used in Java-based development. It automates tasks such as compilation, testing, packaging and dependency management. In this project, Maven is used to efficiently manage CloudSim Plus framework and its related libraries. Maven simplifies the setup of CloudSim Plus by automatically downloading and integrating its dependencies through the pom.xml configuration file.

- **Visual Studio Code (VS Code):**

Visual Studio Code (VS Code) is a lightweight, open-source code editor developed by Microsoft. It supports multiple programming languages and offers different features like syntax highlighting and intelligent code completion. It is a convenient tool for quick development and debugging. Although Eclipse is the main IDE used to build and run the CloudSim Plus simulations, VS Code is occasionally used for writing and

editing parts of the code. Its speed and simplicity make it a handy alternative for writing and debugging code.

3.2 System Model

The system model represents the interaction between the main system components within the simulated data centre environment. It is designed to evaluate the performance of the proposed VM allocation and migration policies using CloudSim Plus as the simulation framework. Figure 3-2 shows the system model for this project.

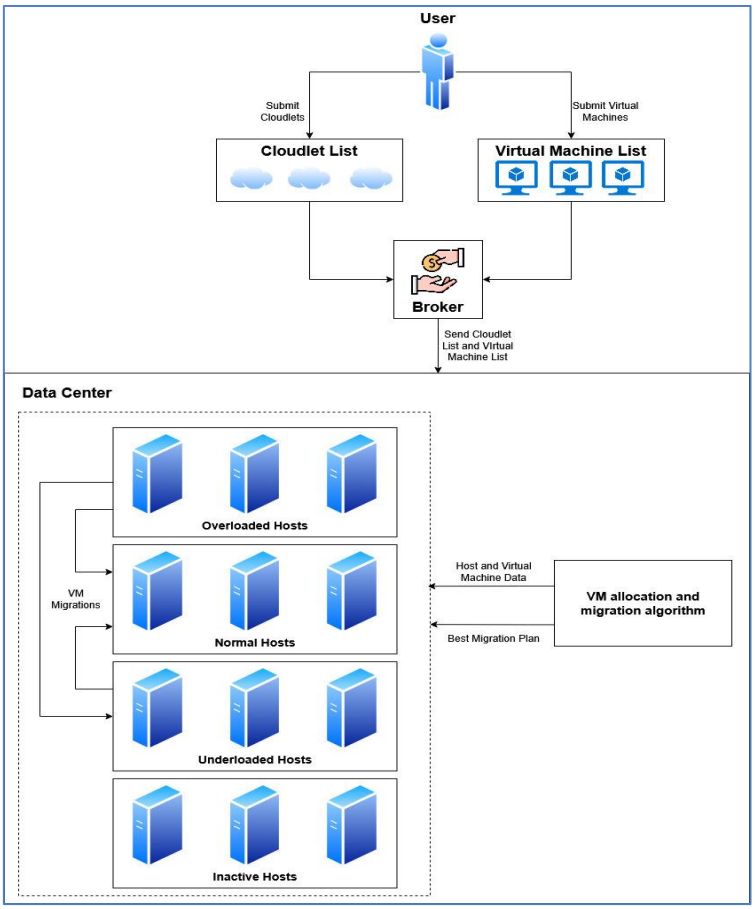

Figure 3-2: System Model for this Project.

At the core of the model lies the data centre, which consists of a set of physical hosts capable of hosting multiple VMs. A data centre broker act as an intermediary between simulated users and the data centre and is responsible for VM management steps such as VM and cloudlet creation, submission and destruction. Initially, the system receives a set of VMs, $V = \{VM_1, VM_2, VM_3, \ldots, VM_{|V|}\}$ and a set of cloudlets (user workloads),

$C = \{C_1, C_2, C_3, \ldots, C_{|c|}\}$. Next, the set of VMs and cloudlets are submitted to the data centre broker. The data centre broker will then allocate the set of VMs, V across a set of physical servers/hosts, $P = \{PM_1, PM_2, PM_3, \ldots, PM_{|P|}\}$. The initial allocation allocates VMs to active hosts with the minimum number of free processing elements (CPU cores). At this stage, no migration occurs and VMs are statically mapped to hosts to provide a starting point for execution.

As the simulation progresses and cloudlets begin executing, certain hosts may become overloaded or underloaded. To address this, the custom VM allocation policy (based on ACO, PSO or MGA) is triggered to determine whether reallocation or migration of VMs is necessary. The goal is to balance the load across hosts or consolidate VMs onto fewer hosts, thereby improving resource utilisation and reducing overall power consumption. During this process, the algorithm identifies the potential source hosts (underloaded and overloaded hosts), VMs eligible for migration, and suitable target hosts based on pre-defined criteria. The objective is to construct an optimal migration plan by selecting the best combination of these migration options that results in the most efficient utilisation of resources and minimises power consumption.

3.3 Algorithms

Efficient virtual machine (VM) allocation and migration are critical for optimizing resource utilization and reducing power consumption in modern data centres. Traditional static or rule-based approaches often fail to adapt effectively under dynamic workloads. To address these limitations, this project leverages bio-inspired algorithms to intelligently determine optimal VM placement and migration strategies.

In this study, three distinct algorithms are implemented and evaluated within a simulated CloudSim Plus environment: Ant Colony Optimisation (ACO) algorithm, Particle Swarm Optimisation (PSO) algorithm and Modified Genetic Algorithm (MGA).

3.3.1 Ant Colony Optimisation (ACO) Algorithm

3.3.1.1 Overview of Ant Colony Optimisation (ACO) Algorithm

Ant Colony Optimisation (ACO) draws its inspiration from the natural foraging behaviour observed in real ant colonies [24]. In nature, ants find the shortest path between their nest and a food source by laying down pheromone trails along the paths

they travel. Other ants would follow these pheromone trails, and the shortest paths accumulate stronger pheromone levels as time goes on, which makes them more attractive for other ants. This concept can be adapted to solve the Virtual Machine Placement (VMP) problem. Ants would iteratively build a solution by choosing a VM to be assigned to a server based on the combination of pheromone levels as well as heuristic values, until all VMs have been assigned to servers. The approach places each VM on the most appropriate server, thus ensuring efficient resource utilisation.

In this project, the ACO algorithm is adapted to address the VM placement and migration problem within data centres. The focus is to optimise resource utilisation and reduce power consumption by balancing workload and consolidating VMs onto fewer active servers. The algorithm operates in iteration, with artificial ants constructing feasible migration plans based on current system state. Each ant builds a solution by selecting a sequence of migration tuples, where a tuple consists of a source host, VM to migrate and a target host. These tuples are selected based on pheromone value, which represents learned experience from previous iterations, and heuristic information that reflects the current desirability of each tuple. Over successive iterations, the algorithm gradually converges towards an optimal or near-optimal migration plan, which is then used to migrate VMs to their designated target hosts for improved resource utilisation and energy efficiency.

3.3.1.2 Initialisation of parameters of ACO algorithm

In the ACO-based VM allocation and migration algorithm, there are several key parameters that influence the behaviour and performance of the solution construction process. These parameters are 32nderutiliz at the beginning of the algorithm. Table 3-1 shows the main parameters in this algorithm:

Parameters	Value	Description
α	1.0	This parameter represents the influence of the pheromone value (τ) in the selection of migration tuples. A larger α places more emphasis on the learned experience of previous ants, which promotes the exploitation of known good solutions

β	2.0	This parameter represents the influence of the heuristic information (η), which is the problem-specific knowledge that guides ants towards more promising solutions. A larger β increases the impact of heuristics and encourages more informed exploration.
q_0	0.7	This parameter represents the exploitation parameter, and it determines whether an ant will exploit the best-known path or explore new path probabilistically. A higher q_0 encourages exploitation, while a smaller q_0 encourages exploration.
ρ	0.1	This parameter represents the pheromone evaporation rate, where $\rho \in (0,1)$. It determines how quickly the pheromone trail fades over time. A small ρ value is favoured to prevent pheromones from evaporating too quickly so that past experiences can be retained in the pheromone values.
τ_0	1.0	This parameter represents the initial pheromone value deposited on all migration tuples. It helps initialise the search space and prevents any bias towards specific solutions in early stages of the algorithm.
Number of ants	5	This parameter defines how many ants are used per iteration. Each ant will construct a local migration plan which will then be evaluated based on the objective functions. A larger number can improve the solution quality but may increase computational time.
Number of iterations	5	This parameter defines how many times the ant colony will repeat the solution construction process. More iterations improve the chances of finding optimal or near-optimal plans. However, this also increases computational time and resource consumption, so a balanced approach is favoured.

K_i	0.5	This parameter represents the fraction of a host's maximum power that is consumed when the host is idle (not hosting any VMs but still powered on). An idle but powered-on server can consume approximately 50% to 70% of the power used by a fully loaded server. Therefore, the fraction parameter is set to 0.5, representing idle or static power as 50% of the server's maximum power consumption.

Table 3-1: Parameters of the ACO-based Algorithm.

3.3.1.3 Heuristic Information

In the ACO-based VM allocation and migration algorithm, heuristic information is denoted as η_{ijk} and plays a crucial role in guiding the decision-making process of ants during the solution construction phase. It provides the estimate of the desirability or suitability of migrating a virtual machine VM_j from a source host PM_i to a target host PM_k, based on resource availability and balance. The equation (1) represents the formula to calculate heuristic information for each migration tuple and it is based on the heuristic calculation in [3].

$$H_{ijk} = \frac{1.0 - \left| \frac{PM_k^c - PM_k^{cu} - VM_j^c}{PM_k^c} - \frac{PM_k^m - PM_k^{mu} - VM_j^m}{PM_k^m} \right|}{\left| \frac{PM_k^c - PM_k^{cu} - VM_j^c}{PM_k^c} \right| + \left| \frac{PM_k^m - PM_k^{mu} - VM_j^m}{PM_k^m} \right| + 1.0} \quad (1)$$

Where PM_k^c and PM_k^m represents the total CPU capacity and memory/RAM capacity of the target host respectively, while PM_k^{cu} and PM_k^{mu} is the currently used CPU and memory/RAM of the target host. VM_j^c is the CPU requirement of the VM to be migrated and VM_j^m is the memory/RAM requirement of the VM to be migrated. The denominator indicates how much of the host's resources including CPU and memory are being used [3]. It captures the extent to which resources are utilised after placing the VM to the target host. On the other hand, the numerator represents how evenly the remaining resources are distributed within that host [3]. A balanced distribution across different resources avoids creating bottlenecks and ensures more efficient usage of the target host.

3.3.1.4 Solution Construction

The solution construction process is the core mechanism of the ACO-based VM allocation and migration algorithm. It involves generating and evaluating potential VM migration plan to achieve better resource utilisation and lower power consumption within data centre. The process begins by identifying whether hosts are overutilised or underutilised based on the pre-defined thresholds. Any hosts with CPU utilisation, μ_i greater than 0.8 will be categorized as overloaded hosts, while hosts with μ_i less than 0.2 will be categorized as underloaded hosts. These thresholds help us to identify candidates for source and target hosts for VM migration. Once the overloaded and underloaded hosts are identified, the algorithm proceeds to construct a set of migration candidates. Each candidate is represented as a tuple as shown in equation (2) where PM_i is the source host, VM_j is the VM selected for migration PM_k is the target host, and T is the set of all migration tuples

$$T_{ijk} = \left(PM_i, VM_j, PM_k\right) \mid PM_i, PM_j \in P, VM_j \in V, x_{ij} = 1 \quad (2)$$

The source host, PM_i is chosen from the underloaded hosts and overloaded hosts, with the aim to balance the workload (from overloaded hosts) or consolidate VMs onto fewer servers (from underloaded hosts). The VM_j is the virtual machine currently running on the source host, while the targe host PM_k is selected from a set of non-overloaded active servers. Once the set of migration tuples T_{ijk} is generated, the ACO algorithm begins its iterative process to construct a migration plan. In each iteration, artificial ants traverse the solution space by probabilistically selecting migration tuples based on two factors, pheromone trail (τ_{ijk}) and heuristic information (η_{ijk}). The probability of selecting a tuple T_{ijk} is computed using equation (3).

$$p\left(T_{ijk}\right) = \frac{(\tau_{ijk})^{\alpha}(\eta_{ijk})^{\beta}}{\sum_{(i,j,k)\in T}(\tau_{ijk})^{\alpha}(\eta_{ijk})^{\beta}} \quad (3)$$

Another key mechanism in the solution construction phase is the balance between exploration and exploitation. Exploration involves trying new migration options regardless of the selection weight, while exploitation involves reinforcing the best-known options. The key to constructing high-quality VM migration plans is to find a balance between exploration and exploitation. To achieve this, an exploitation parameter (q_0) is set to control whether the ants perform exploration or exploitation. Equation (4) below shows the exploration vs exploitation rule:

$$T_{ijk}^{*} = \begin{cases} \arg max_{i,j,k \epsilon T}\left[(\tau_{ijk})^{\alpha}(\eta_{ijk})^{\beta}\right] if\ q \leq q_0\ (Exploitation) \\ Select\ T_{ijk}\ with\ probability\ p(T_{ijk}), if\ q > q_0\ (Exploration) \end{cases} \quad (4)$$

A random number q is generated to determine the ant's behaviour, whereby the ant will choose the tuple whose product of pheromone value τ and heuristic information η are maximal when the q value is less than or equal to the exploitation parameter, q_0 – this is the exploitation behaviour. Otherwise, the ant will select tuple T_{ijk} based on the probability computed in equation (2) – this is the exploration behaviour.

Each ant builds a complete migration plan by selecting a sequence of feasible migration tuples. These plans are known as the local migration plan, M^a. Once a feasible migration tuple is added to the local migration plan M^a, a local pheromone update is immediately applied to the pheromone trail associated with that tuple. The construction of the local migration plan continues until all migratable VMs have migrated or if there are no more feasible migration candidates. Each constructed local migration plan is then evaluated using the objective function that evaluates the score of the plan. The best performing plan among all ants in an iteration is designated as the iteration-best migration plan, denoted as M^*. If this iteration-best plan outperforms the current global-best plan M^B, it replaces it as the new global best. At the end, the globally best migration plan M^B is translated into a VM-host mapping to server as the execution plan for the actual migration process. The mapping specifies which VM to migrate from its current host to a new target host that it is mapped with. This global best migration plan is also used in the global pheromone update phase of the algorithm. Figure 3-3 shows a simple flowchart of one iteration cycle of the ACO algorithm.

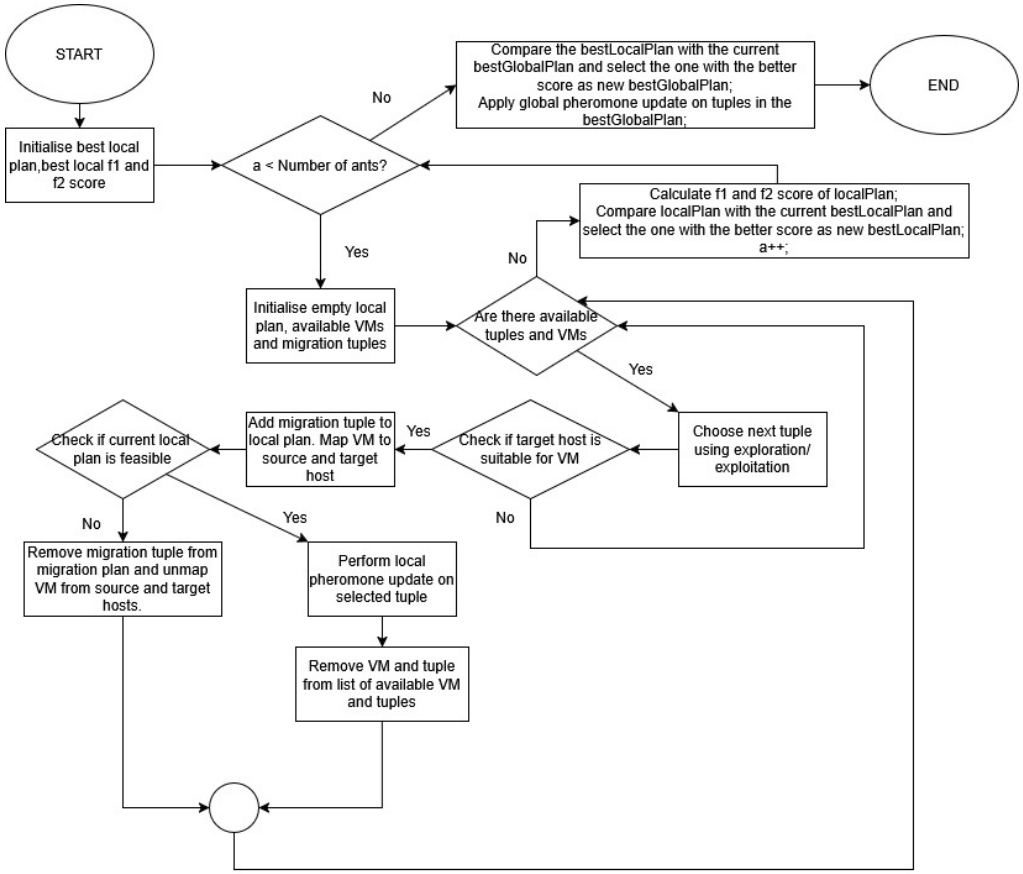

Figure 3-3: Flowchart of one iteration cycle of the ACO algorithm.

3.3.1.5 Objective Function

To evaluate the quality of each migration plan generated by artificial ants in the algorithm, two objective functions – f_1 score and f_2 score is used. The equations for calculating both scores are adopted from [3]. The f_1 score aims to minimising the number of active hosts by consolidating VMs onto fewer servers. It counts the number of active physical servers required to host all VMs under the given migration plan, M^a. A host is considered active if it hosts at least one VM. Equation (5) and (6) show the calculation of f_1 score:

$$y_i = \begin{cases} 1, & if \sum_{j \in V} x_{ij} \geq 1 \\ 0, & otherwise \end{cases}, \quad \forall i \in P \tag{5}$$

$$f_1(M^a) = \sum_{i \in P} y_i \tag{6}$$

Here y_i is a binary indicator that shows whether host I is active. Unlike in [3], where the f_1 score is assigned a value equal to the total number of physical servers plus one for infeasible migration plans (to distinguish them from feasible ones), the algorithm in this project takes a different approach by filtering out infeasible tuples from the plan, specifically those where the target host cannot accommodate the VM. The migration plan with the lower f_1 score is considered to perform better and is selected as the best local migration plan. The best local migration plan will then be compared to the best global migration plan, and it replaces the global one if it performs better.

In case where there are two migration plans with the same f_1 score, the algorithm calculates the f_2 score to measure how well the resources are utilised across all active servers. A smaller f_2 score indicates a more balanced resource utilisation across servers. Equation (7) shows the formula for calculating f_2 score:

$$f_2(M^a) = \sum_{i \in P} \left(\left(\frac{|PM_i^c - PM_i^{cu}|}{PM_i^c} + \frac{|PM_i^m - PM_i^{mu}|}{PM_i^m} \right) y_i \right) \quad (7)$$

Together, the two scores guide the algorithm towards migration plans that have fewer active servers and better resource utilisation.

3.3.1.6 Pheromone Update Rule

The pheromone update rule is another critical component of the ACO-based VM allocation and migration algorithm. It guides the collective learning behaviour of ants by reinforcing good solutions and gradually letting poorer ones fade away. The pheromone trail/value of each migration tuple, τ_{ijk} is updated through two processes: local pheromone update and global pheromone update.

Local Pheromone Update

Each ant updates the pheromone value of the tuples it has selected during the construction of its local migration plan. This process helps promote exploration by slightly reducing the pheromone intensity of frequently chosen tuples by the last ant, which encourages other ants to explore alternative paths. The local pheromone update rule is defined in equation (8):

$$\tau_{ijk} = (1 - \rho) \cdot \tau_{ijk} + \rho \cdot \tau_0 \quad (8)$$

Where ρ represents the pheromone decay parameter and $0 < \rho < 1$. T_{ijk} represents the pheromone value of the tuple T_{ijk} while τ_0 is the initial pheromone value.

Global Pheromone Update

After all the ants have constructed their local migration plans in one iteration, the algorithm will identify the best-performing plan M^* based on the objective functions. This plan is then compared with the current global best migration plan (M^b) and replaces it if it performs better in overall objectives. The global pheromone update rule in this algorithm is a modified version of the one described in [3]. The pheromone values for the tuples in the global best plan are updated using the equations (9) and (10) defined for global pheromone updates.

$$\tau_{ijk} = (1 - \rho) . \tau_{ijk} + \rho . \Delta \tau_{ijk} \quad (9)$$

$$\Delta\tau_{ijk} = \frac{1}{f_1(M^b)} + \frac{1}{\frac{PM_k^c - PM_k^{cu}}{PM_k^c} + \frac{PM_k^m - PM_k^{mu}}{PM_k^m} + 1} \quad (10)$$

$\Delta \tau_{ijk}$ represents the delta pheromone value. The first component of $\Delta \tau_{ijk}$ ensures that better solutions with a lower f_1 score receive a higher pheromone increment. The f_1 score measures the number of active hosts, so migration plans the involve fewer active servers are favoured. On the other hand, the second component of $\Delta \tau_{ijk}$ promotes the consolidation of VMs onto fewer active servers by rewarding target hosts with less remaining CPU and memory capacity. In other words, it encourages the selection of target hosts that are more fully utilised.

3.3.1.7 Complete ACO-based VM Allocation and Migration Algorithm

This section presents the complete flow of the proposed ACO-based VM Allocation and Migration Algorithm. The following flowchart in Figure 3-4 illustrates the step-by-step process of the complete algorithm.

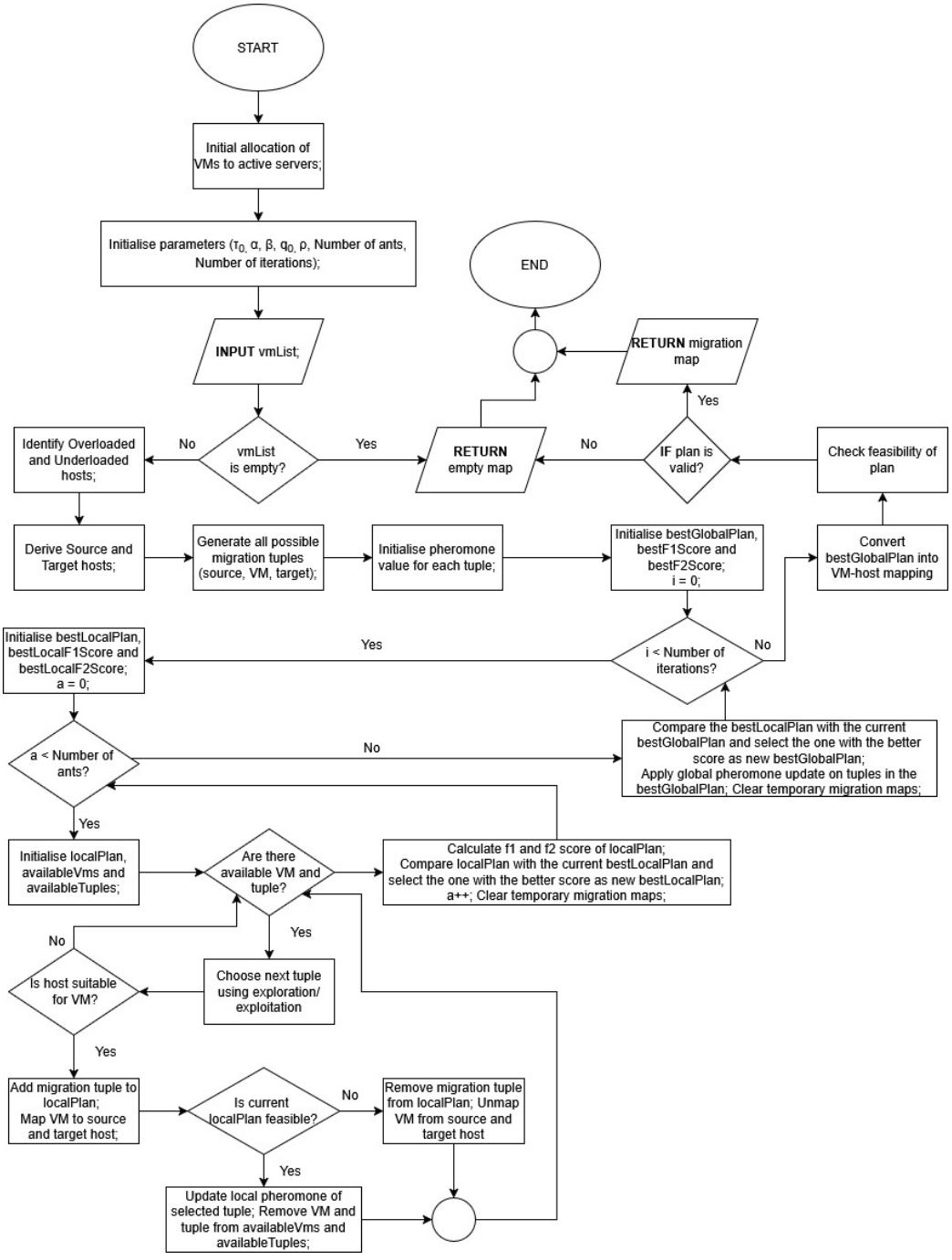

Figure 3-4: Complete ACO-based Algorithm Flowchart.

3.3.2 Particle Swarm Optimisation (PSO) Algorithm

3.3.2.1 Overview of Particle Swarm Optimisation (PSO) Algorithm

Particle Swarm Optimisation (PSO) is an optimisation technique inspired by the collective behaviour of bird flocking and fish schooling [32]. In the context of this algorithm, a particle represents a single VM-to-host mapping, while a swarm consists of multiple particles, collectively forming a candidate migration plan. Multiple swarms are maintained simultaneously, allowing the algorithm to explore several alternative migration strategies in parallel.

The algorithm begins by identifying VMs that need to be migrated from both overloaded and underloaded hosts. Once the set of migratable VMs and available target hosts is determined, the PSO process is initialised by generating multiple swarms. Within each swarm, the position of a particle represents the selected host for a particular VM, while its velocity determines how the mapping changes between iterations.

During each iteration, the algorithm evaluates the fitness of each swarm based on how effectively its overall migration plan balances CPU and RAM utilisation across the available hosts. Each swarm maintains a personal best migration plan, which is the most efficient configuration it has discovered so far, while the global best migration plan is tracked across all swarms. Using these best-known solutions, combined with randomised exploration factors, the particles within each swarm update their velocities and positions to refine the VM-to-host mappings, progressively improving the quality of their migration plans.

After completing the specified number of iterations, the global best swarm is selected, and its corresponding migration plan is applied to produce the final VM allocation strategy, optimising resource utilisation and improving overall data centre performance.

3.3.2.2 Initialisation of parameters of PSO algorithm

In the PSO-based VM allocation and migration algorithm, there are several key parameters that influence the behaviour and performance of the solution construction process. Table 3-2 shows the main parameters in this algorithm:

Parameters	Value	Description
iterations	10	Number of optimisation cycles, represents the total number of velocity and position updates to refine solutions.
swarmNo	10	Number of swarms (solutions)
c_1	2.0	Parameter c1 is known as the cognitive acceleration coefficient. It controls how much a particle is influenced by its own best-known position
c_2	2.0	Parameter c2 is known as the social acceleration coefficient. It controls how much a particle is influenced by the global best-known position.
w	0.5	Inertia weight w scales the particle's current velocity to maintain momentum from the previous step.
r_1, r_2	random.nextDouble()	Parameters r1 and r2 are random scalars in [0, 1], Parameter r1 adds stochasticity to the cognitive component to promote exploration, while parameter r2 adds stochasticity to the social component to diversify movement.

Table 3-2: Parameters of the PSO-based Algorithm.

3.3.2.3 VM allocation and migration rule (PSO algorithm)

A. Swarm Initialisation

In this PSO-based VM allocation approach, multiple swarms are created. Each swarm represents a potential solution space and consists of multiple particles, with the number of particles in each swarm determined by the number of migratable VMs. For each swarm, every particle corresponds to a VM and has two key attributes: position and velocity. The position represents the index of the host to which a particular VM is

assigned. On the other hand, the velocity defines the rate and direction of change for the particle's position in subsequent iterations, influencing how the particle explores the solution space.

During initialization, each particle's position is randomly assigned to a host from the available host list, ensuring diversity in the initial solutions. Similarly, each particle's velocity is randomly initialized between 0 and 1, allowing different particles to move with varying dynamics. Finally, the global best swarm is initialized to store the best overall solution discovered across all swarms during the optimization process. This setup ensures a broad and diverse exploration of the search space, improving the chances of finding an optimal VM allocation strategy.

B. Fitness Function (PSO algorithm) and Personal/Global Best Update

In the Particle Swarm Optimization (PSO) algorithm for VM allocation, the fitness function evaluates how optimal a given VM-to-host mapping is for each swarm. The goal is to minimize resource imbalance and ensure that VM allocations are feasible based on host capacities. Additionally, the algorithm updates each swarm's personal best and the overall global best solutions to guide future particle movements.

The fitness function plays a key role in evaluating how well a given solution maps VMs to available hosts. Each VM is assigned to a host based on the particle's position, which represents a candidate solution. For every allocation, the algorithm checks whether the selected host has sufficient available resources to accommodate the VM's requirements. If any of the resource constraints (CPU, memory, bandwidth, or storage) are violated, the solution is marked as infeasible and is penalized with a very high fitness value. For feasible solutions, the fitness value of the particle is calculated based on how balanced the workload is across all hosts. This is done by measuring the variance in CPU and RAM usage relative to their averages across the data center. A lower variance indicates a more balanced and efficient allocation, resulting in a lower fitness value, which the algorithm seeks to minimize. Equation (11) below shows the fitness function:

$$F = \sum_{i=1}^{n} \frac{PM_i^{cu} - AvgCPU}{PM_i^{cu}} - \frac{PM_i^{mu} - AvgRAM}{PM_i^{mu}} \quad (11)$$

Where PM_i^{cu} and PM_i^{mu} is the currently used CPU and memory/RAM of i^{th} host, while AvgCPU and AvgRAM represents the average CPU and memory/RAM usage of all available hosts.

Once the fitness value is computed, the PSO algorithm updates two key metrics: the personal best and the global best. Each particle, representing a specific VM-to-host mapping within a swarm, maintains its own personal best position, which reflects the most efficient allocation it has achieved so far. If the particle's current allocation yields a better fitness value than its historical best, it updates its personal best accordingly. At the same time, the algorithm evaluates all particles across all swarms to identify the global best solution, which represents the most optimal allocation discovered so far.

C. Velocity and Position Update Rule

In the PSO algorithm, the velocity and position update rule govern how particles (each representing a VM-to-host mapping) move through the search space to explore better allocation strategies. The equations involved in the update rule are adopted from [32]. Each particle's position X and velocity V at time (t) is represented as shown in equation (12) and (13) below:

$$X_i(t) = x_1, x_2, \dots \dots x_n \quad (12)$$

$$V_i(t) = v_1, v_2, \dots \dots v_n \quad (13)$$

Each swarm consists of n particles, which is equivalent to the number of migratable VMs. Each particle's personal best position and global best position is represented as pBest and gBest. For each swarm, the algorithm iterates through all particles and updates their velocities based on three key components:

1. Inertia Component (w) – This term controls the particle's tendency to continue moving in its current direction. A higher inertia weight encourages exploration of the search space, while a lower weight promotes exploitation around known good solutions.
2. Cognitive Component (c_1 * $r_1(t)$ * (pBest – currentPosition)) – This factor represents the particle's personal learning. It pulls the particle toward its own personal best position, guiding it based on its individual experience.
3. Social Component (c_2 * $r_2(t)$ * (gBest – currentPosition)) – This factor reflects collective learning. It attracts the particle toward the global best position found

across all swarms, encouraging collaboration among particles to converge on the most promising solutions.

The updated velocity is calculated as shown in equation (14):

$$V_i(t+1) = wV_i(t) + \big(pBest_i - X_i(t)\big)c_1r_1(t) + \big(gBest_i - X_i(t)\big)c_2r_2(t) \quad (14)$$

Where v_i, x_i, $pBest_i$ *and* $gBest_i$ is the velocity, the current position, the personal best position and the global best position of particle I respectively. Parameters $r_1(t)$ and $r_2(t)$ are random numbers in the range [0, 1], while c_1, and c_2 are acceleration coefficient as mentioned in the parameters section.

After updating the velocity, the particle's position is adjusted as shown in equation (15):

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (15)$$

Through repeated updates over multiple iterations, the PSO algorithm balances exploration (searching for new solutions) and exploitation (refining around the best-known solutions), gradually converging toward an optimal VM-to-host allocation strategy.

3.3.2.4 Single Iteration of PSO Algorithm

In a single iteration of PSO, each swarm is evaluated to determine how effectively it balances VM placement across the available hosts. The algorithm first computes the fitness of each swarm based on resource utilisation, then updates the personal best migration plan for each swarm if its current configuration outperforms previous attempts. Next, the global best migration plan is updated by comparing all swarms to identify the most optimal solution found so far. Finally, the velocity and position of each particle within the swarms are adjusted based on their personal best, the global best, and random exploration factors. Figure 3-5 shows the flowchart of one iteration cycle of the PSO algorithm.

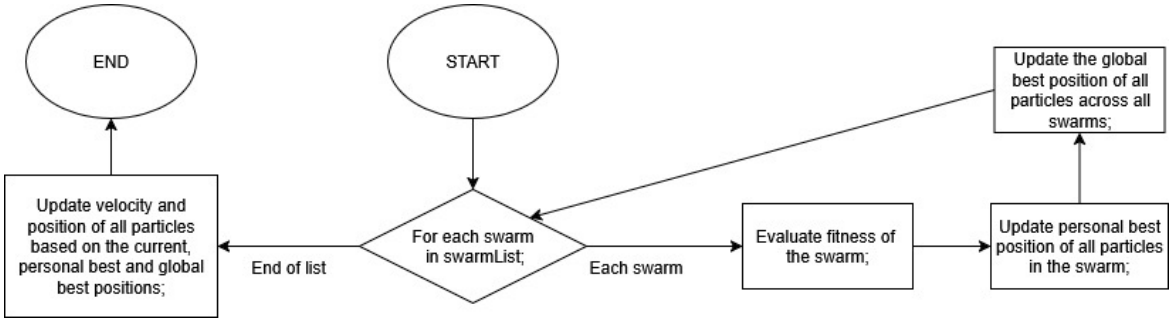

Figure 3-5: Flowchart of one iteration cycle of the PSO algorithm.

3.3.2.5 Complete PSO-based VM Allocation and Migration Algorithm

This section presents the complete flow of the proposed PSO-based VM Allocation and Migration Algorithm. The following flowchart in Figure 3-6 illustrates the step-by-step process of the complete algorithm.

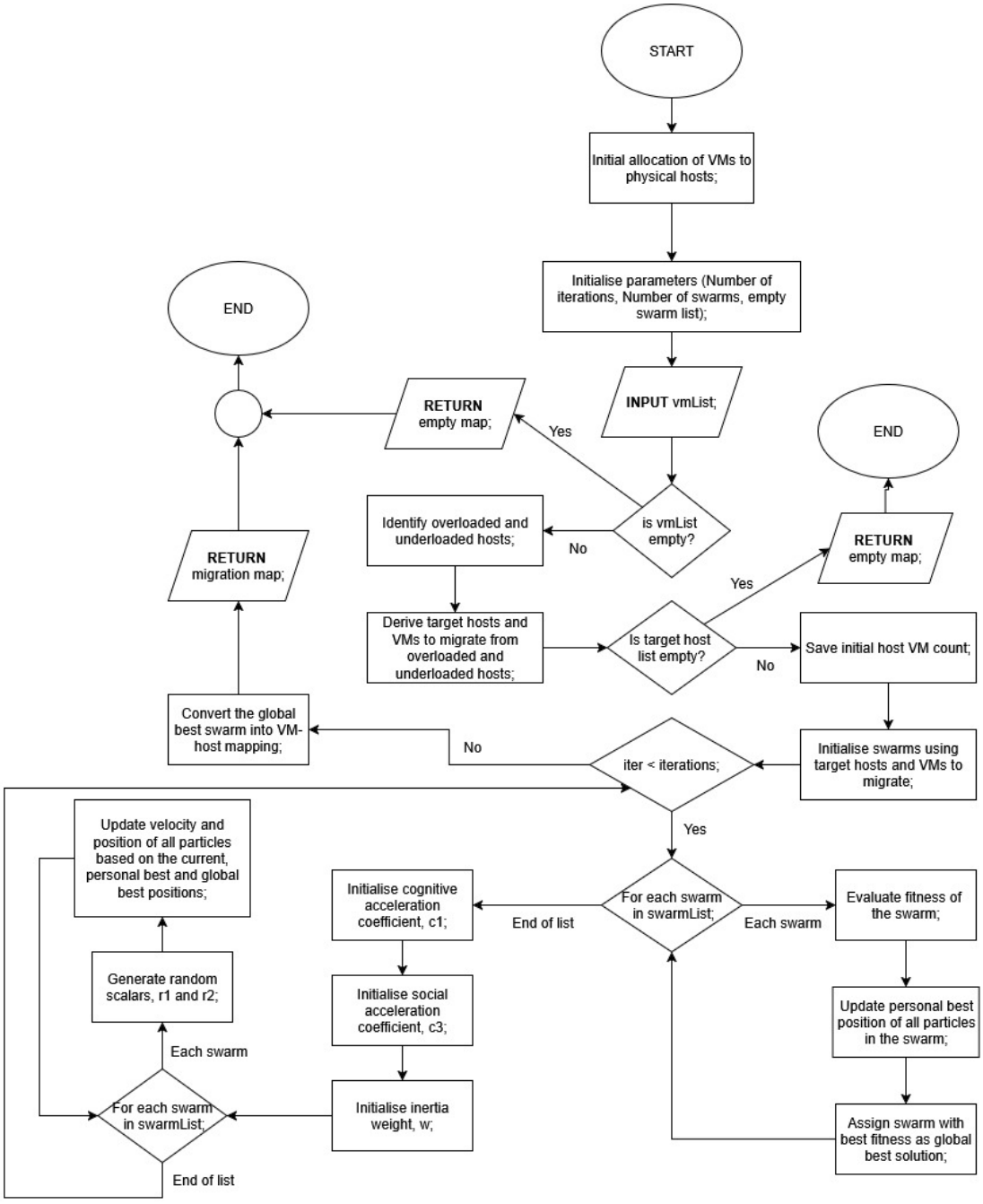

Figure 3-6: Complete PSO-based Algorithm Flowchart.

3.3.3 Modified Genetic Algorithm (MGA)

3.3.3.1 Overview of Modified Genetic Algorithm (MGA)

The Modified Genetic Algorithm (MGA) is inspired by Darwin's theory of natural selection, where the fittest individuals are selected for reproduction to produce offspring of the next generation [33]. In the context of Virtual Machine Placement (VMP), MGA evolves a population of candidate solutions to find an optimal or near-optimal mapping of VMs to physical hosts in a data center. Each solution is called a chromosome, and it represents a specific VM placement configuration.

The algorithm operates over multip"e ge'erations, where each generation evolves through the core genetic operations: selection, crossover, and mutation. In each generation, parent chromosomes are selected based on their fitness scores, which aims to minimise the number of overutilised and underutilised. The crossover operation then combines two parent solutions to create new offspring by exchanging VM placement segments. To maintain diversity and prevent premature convergence, mutation introduces small random changes in the offspring's VM assignments. After generating a new population, all individuals are re-evaluated, and the process continues for a predefined number of generations. Ultimately, the best chromosome from the final generation is decoded into an optimal or near-optimal VM-to-host mapping.

3.3.3.2 Initialisation of parameters of MGA

In the MGA-based VM allocation and migration algorithm, there are several key parameters that influence the behaviour and performance of the solution construction process. Table 3-3 shows the main parameters in this algorithm:

Parameters	Value	Description
iterations/generations	100	Number of generations the MGA algorithm runs.
populationSize	10	Number of chromosomes (solutions) in each generations.
W_1	0.5	Weight assigned to the number of underutilized hosts in the fitness function.

W_2	0.5	Weight assigned to the number of overutilized hosts in the fitness function.
ε	0.2	ε controls how strictly hosts are classified as over- or underutilized by shrinking the utilization thresholds toward the average.
mutationRate	0.1	Probability of applying mutation to a chromosome.
hasMigratedOnce	false	Flag to restrict migration to once per simulation.

Table 3-3: Parameters of the MGA-based Algorithm.

3.3.3.3 VM allocation and migration rule (MGA)

A. Identification and Classification of Overutilised and Underutilised Hosts

Unlike the other two algorithms (ACO and PSO), where overutilisation and underutilisation thresholds are set from the beginning, MGA performs dynamic identification and classification of overutilized and underutilized hosts using a statistical approach based on the Moving Range (MR) technique. First, it computes the average utilization and standard deviation from the collected CPU utilisation of each physical hosts to capture the central tendency and variability of resource usage. After that, the Upper Control Limit (UCL) and Lower Control Limit (LCL) are determined using the 3-sigma rule as shown in equation (16) and (17), where μ represents the average utilisation and σ represents the standard deviation:

$$UCL = \mu + 3\sigma \quad (16)$$

$$LCL = \mu - 3\sigma \quad (17)$$

These limits represent the natural operating bounds for host utilization under normal conditions. The limits are further adjusted using an epsilon (ε) factor, which is used to increase or reduce the sensitivity to workload changes. The thresholds are calculated as shown in equation (18) and (19):

$$Overutilisation\ Threshold = (1 - \varepsilon) \times UCL \quad (18)$$

$$Underutilisation\ Threshold = (1 - \varepsilon) \times LCL \quad (19)$$

Finally, hosts with utilization above the upper threshold are classified as overutilized, while those below the lower threshold are classified as underutilized. Hosts within the thresholds are considered fairly loaded.

B. Encoding and Initialisation of Population

In the MGA, the VM placement solution is encoded as chromosomes. A chromosome consists of multiple genes, each representing the physical host assigned to a specific VM, directly mapping all VMs to their allocated hosts. The Initial Population Generation Strategy initializes the chromosome population by dividing VMs into two groups: those hosted on underutilized or overutilized hosts, and those on normally utilized hosts. Those in the second group retain their original placement to preserve high-fitness individuals, while the first group undergoes a random search to maintain diversity.

C. Fitness Function (MGA algorithm)

The fitness function in the MGA is designed to reduce energy consumption and improve resource utilization. This is achieved using a weighted sum of two key metrics: the number of underutilised physical hosts (NUU) and the number of overutilised physical hosts (NOU). For each chromosome, the algorithm calculates the total CPU utilisation per host based on the VM-to-host mapping. It then compares each host's utilisation against predefined thresholds to count underutilised and over-utilised hosts. The objective value is computed as shown in equation (20):

$$F_{obj}(x) = W_1 * NUU(x) + W_2 * NOU(x) \quad (20)$$

In the equation, W_1 and W_2 are the weights assigned to NUU and NOU respectively, while x represents a chromosome in the population. The final fitness score is inversely proportional to the objective value, ensuring that solutions with fewer underutilised and overutilised hosts receive higher fitness values.

D. Parent Selection

Parent selection in the Modified Genetic Algorithm (MGA) is performed using the roulette wheel selection method, where the probability of selecting an individual as a parent is proportional to its fitness value. This ensures that fitter chromosomes have a higher likelihood of being chosen for reproduction, thereby promoting the propagation of high-quality solutions in subsequent generations. The selection process calculates

the cumulative fitness across the population and chooses a parent when the cumulative value surpasses a randomly generated threshold. This probabilistic mechanism maintains diversity while guiding the search toward optimal or near-optimal solutions.

E. VM Placement Crossover

Crossover is performed in MGA to improve VM distribution by combining allocation patterns from two parent solutions. The process begins by identifying the high-fitness parent (better solution) and the low-fitness parent (worse solution) between the two chosen parents. From the low-fitness parent, physical hosts in the underutilized and overutilized categories are selected, along with the VMs hosted on them. The selected VMs are then migrated to the corresponding hosts of the high-fitness parent, provided that the resource constraints are satisfied. This targeted reassignment ensures that poorly placed VMs from the low-fitness parent benefit from the better allocation decisions in the high-fitness parent, leading to improved offspring quality.

F. VM Placement Mutation

In Genetic Algorithms (Gas), mutation introduces small random changes to candidate solutions to maintain diversity and prevent premature convergence. The mutation approach in the MGA targets physical hosts classified as underutilised or overutilised and relocates their VMs to the most suitable host in the fair group that meets the VM's resource constraints. During this process, each VM on underutilized or overutilized hosts is considered for mutation with a fixed mutation rate. Candidate PMs from the fair group are evaluated based on available resources, and the VM is migrated to the one with the highest free CPU capacity while ensuring feasibility. This strategy helps maintain population diversity while improving VM placement efficiency in subsequent generations.

3.3.3.4 Single Iteration of MGA

A single iteration of the Modified Genetic Algorithm (MGA) begins with a population of candidate solutions, each represented as a chromosome encoding a VM-to-host placement. The process starts by generating an initial population and evaluating the fitness of each chromosome according to the defined objective function. Within each generation, two parent chromosomes are selected using a selection method that favours higher-fitness solutions. These parents undergo the problem-specific crossover operation, where VM placements from underutilized and overutilized PMs in the lower-fitness parent are reassigned to corresponding PMs in the higher-fitness parent. The resulting offspring chromosome is then subjected to mutation, which may relocate selected VMs from overloaded or underloaded PMs to suitable PMs in the fair group to maintain diversity. After all offspring for the new generation are created, their fitness values are recalculated, and the population is replaced by this new set of solutions. This process is repeated for a predefined number of generations. At the end of the iterations, the best-performing chromosome is decoded to produce the final VM-to-PM allocation map. Figure 3-7 below shows the flowchart of one iteration cycle of the MGA.

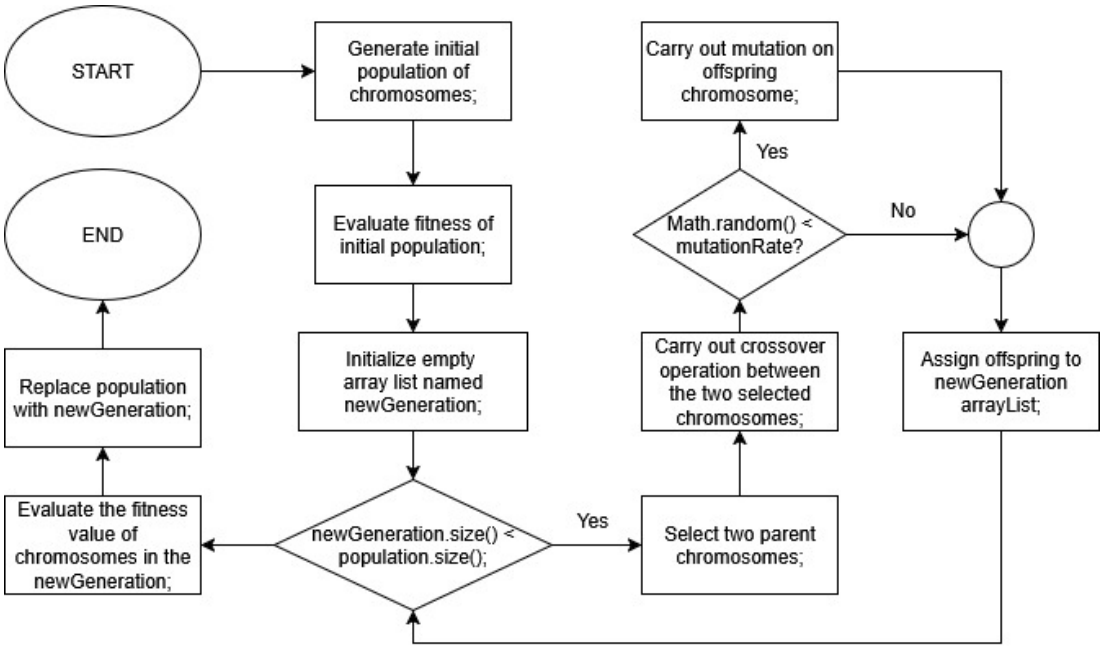

Figure 3-7: Flowchart of one iteration cycle of the MGA.

3.3.3.5 Complete MGA-based VM Allocation and Migration Algorithm

This section presents the complete flow of the proposed MGA-based VM Allocation and Migration Algorithm. The following flowchart in Figure 3-8 illustrates the step-by-step process of the complete algorithm.

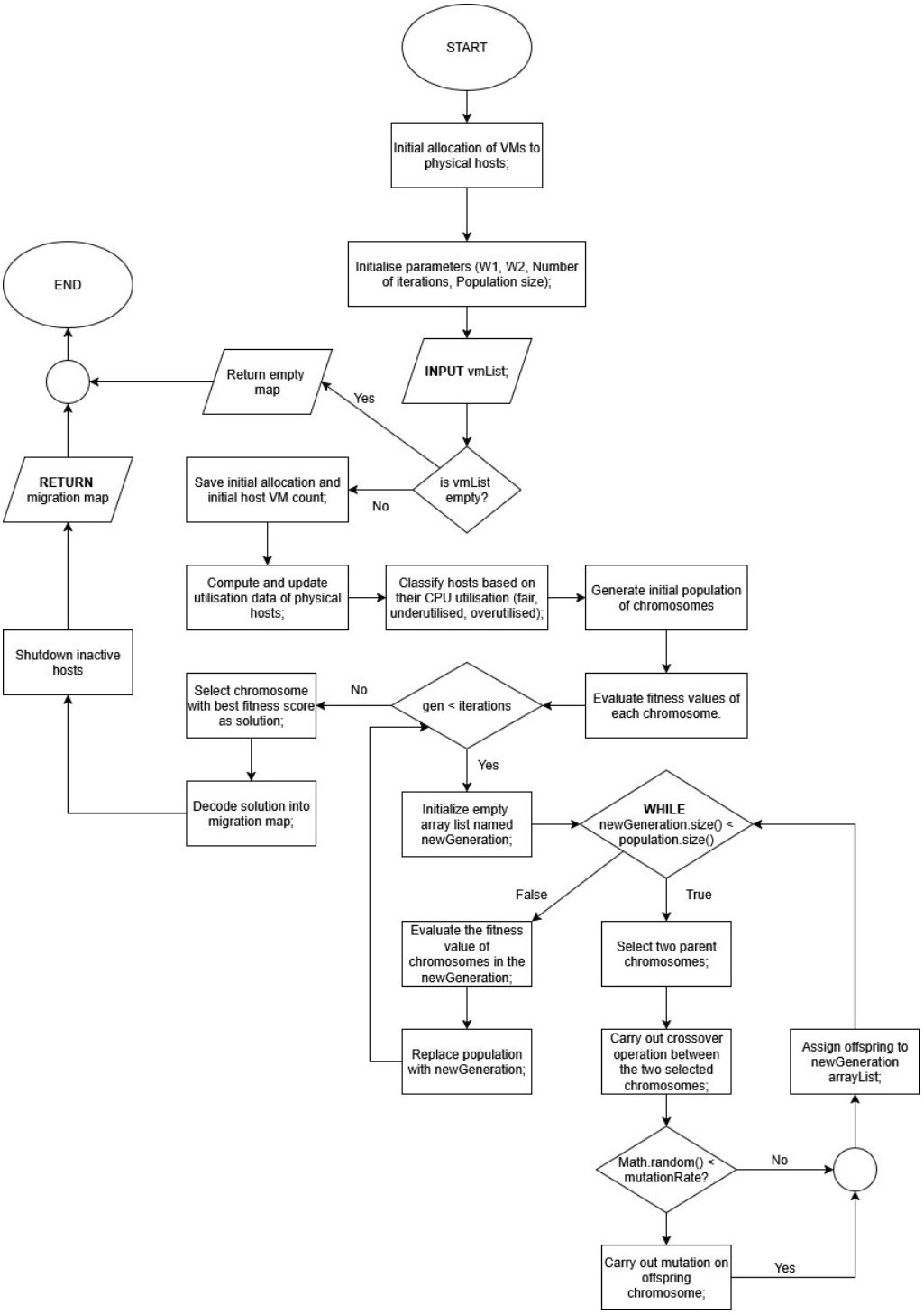

Figure 3-8: Complete MGA-based Algorithm Flowchart.

Chapter 4
System Design

This chapter presents the overall design of the proposed system for energy-efficient VM allocation and migration in data centres. It begins with the problem formulation, outlining the optimisation objectives and constraints. The main system components are then described, followed by a system block diagram to illustrate the data flow and interactions. Finally, the chapter provides a visualisation of algorithm behaviour, showcasing the ideal behaviour of the implemented bio-inspired algorithms.

4.1 Problem Formulation

The core problem addressed in this project is the efficient allocation and migration of Virtual Machines (VMs) to physical servers within a data centre environment. The goal is to optimise resource utilisation while minimising power consumption of physical servers. To achieve this, the problem is formulated as an optimisation task where a set of VMs, each with specific resource requirements (CPU, memory, storage, bandwidth), must be assigned to a set of available physical servers/hosts in a way such that the resource utilisation (CPU, memory utilisation) is optimised, and power consumption is minimised.

Inputs:

1. A list of physical servers/hosts P, where $P = \{PM_i \mid 1 \leq i \leq |P|\}$. Each physical server/host is represented as PM_i and its CPU, memory, storage and bandwidth capacity is represented as PM_i^c, PM_i^m, PM_i^s, and PM_i^b respectively. The current CPU, memory, storage and bandwidth usage of PM_i is represented as PM_i^{cu}, PM_i^{mu}, PM_i^{su}, and PM_i^{bu}.
2. A list of Virtual Machines V, where $V = \{VM_j \mid 1 \leq j \leq |V|\}$. Each VM is represented as VM_i and its CPU, memory, storage and bandwidth capacity is represented as VM_i^c, VM_i^m, VM_i^s, and VM_i^b respectively.
3. A list of Cloudlets C, where $C = \{C_t \mid 1 \leq t \leq |C|\}$.

Objectives:

The main objective is to determine an optimal virtual machine (VM) placement plan where the power consumption across all physical servers are minimised. Thus, a power model to measure power consumption is constructed. The total power consumption of all physical servers is defined in equation (21):

$$\sum_{i=1}^{|P|} E(PM_i) \tag{21}$$

where $E(PM_i)$ is the power consumed by PM_i as shown by the power model in equation (22).

$$E(PM_i) = k_i * e_i^{max} + (1 - k_i) * e_i^{max} * \mu_i \tag{22}$$

where e_i^{max} is the maximum power consumed by PM_i when the server's utilisation is maximum; k_i is the fraction of power consumption when the server is in idle or static mode, whereas μ_i is the CPU utilisation of PM_i. The CPU utilisation of PM_i, μ_i is defined as shown in equation (23):

$$\mu_i = \frac{PM_i^{cu}}{PM_i^{c}}, 0 \leq \mu_i \leq 1 \tag{23}$$

An overutilisation and underutilisation threshold is set to identify physical servers that are either overloaded/overutilised or underloaded/underutilised. A host is considered overutilised if its CPU utilisation is over 80%, while a host is considered underutilised if its CPU utilisation falls below 20% as shown in equation (24).

$$PM_i \text{ is overutilised if } \mu_i > 0.8 \text{ and underutilised if } \mu_i < 0.2 \tag{24}$$

In this project, the average power consumption, CPU utilisation and RAM utilisation is used as key metrics to measure the performance of the proposed method. They are computed using equation (25), (26) and (27).

$$AvgPower = \frac{1}{|P|}\sum_{i=1}^{|P|} E(PM_i) \tag{25}$$

$$AvgCPU = \frac{1}{|P|}\sum_{i=1}^{|P|} \mu_i \tag{26}$$

$$AvgRAM = \frac{1}{|P|}\sum_{i=1}^{|P|} \frac{PM_i^{mu}}{PM_i^{m}} \tag{27}$$

Constraints:

VM_j can only be placed in PM_i if and only if PM_i satisfies the resource requirements (CPU, memory, storage and bandwidth) of VM_j as shown in equation (29), (30), (31) and (32). The total usage, including the assigned VM, must not exceed the PM's resource capacity. Additionally, each VM can only be placed in one and only one physical server (PM) as shown in equation (28).

$$x_{ij} = \begin{cases} 1, & if\ VM_j\ is\ placed\ in\ PM_i \\ 0, & Otherwise \end{cases} \quad \forall i \in P\ and\ \forall j \in V \tag{28}$$

$$\sum_{j=1}^{|V|} VM_j^c \,.\, x_{ij} + PM_i^{cu} \leq PM_i^c \tag{29}$$

$$\sum_{j=1}^{|V|} VM_j^m \,.\, x_{ij} + PM_i^{mu} \leq PM_i^m \tag{30}$$

$$\sum_{j=1}^{|V|} VM_j^s \,.\, x_{ij} + PM_i^{su} \leq PM_i^s \tag{31}$$

$$\sum_{j=1}^{|V|} VM_j^b \,.\, x_{ij} + PM_i^{bu} \leq PM_i^b \tag{32}$$

4.2 Main System Components

This project simulates a data centre environment to evaluate the performance of the proposed VM allocation and migration algorithms based on ACO, PSO and MGA. The system consists of the following core components:

1. Data Centre

The data centre is the central component of the simulation environment and represents the physical infrastructure in a data centre environment. It is responsible for hosting multiple physical servers (hosts) and managing the allocation and execution of VMs and cloudlets (user workloads). In this project, CloudSim Plus provides a built-in Datacenter class, which is used to model the data centre. This class supports the configuration of the data centre, including the lists of hosts and the VM allocation policy that defines how VMs are distributed across the available hosts.

2. Data Centre Broker

The data centre broker act as the intermediary between users and the data centre/cloud infrastructure. Its main role is to manage the submission and scheduling of VMs and cloudlets on behalf of the users. In this project, the responsibility of broker is to receive a list of VMs and cloudlets and decide where to place them based on the allocation policy. The DataCenterBroker class in CloudSim Plus is used to represent this component.

3. Hosts (Physical Servers)

In CloudSim Plus, hosts represent the physical servers in a data centre. These hosts are responsible for providing VMs with necessary computational resources such as CPU, memory, bandwidth and storage. Each host can be configured using the Host class in CloudSim Plus. Possible configurations include the number of CPU cores, the Million instructions per second (MIPS) capacity of each core, memory (RAM) capacity, storage capacity and network bandwidth. The power model of the hosts can also be configured by extending the PowerModelHostAbstract class. Multiple VMs can run concurrently on a single host if the VmScheduler attribute of the host is set to VmSchedulerTimeShared(). The scheduler allows for time-sharing where each VM is allocated a slice of the host's CPU resources.

4. Virtual Machines (VMs)

Virtual Machines (VMs) represent the logical abstraction of computing resources within a physical host. A VM provides a platform for running user workloads, such as cloudlets (tasks or jobs). In CloudSim Plus, VMs are configured using the Vm class and can be configured with specific attributes such as number of CPU cores, MIPS capacity per core, RAM capacity, storage capacity and network bandwidth. After they are configured, the list of VMs is submitted to the data centre broker to be allocated to physical hosts. A VM can only be allocated to a host if the host has sufficient resources to host the VM. Like the VmScheduler attribute of hosts, if the CloudletScheduler attribute of the VM is set to CloudletSchedulerTimeShared(), multiple cloudlets can run concurrently on a single VM.

5. Cloudlets (Tasks/User Workloads)

A cloudlet in CloudSim Plus represents a unit of workload or task that is submitted by a user to be executed on a VM. Like a real-world application or computational job, cloudlets consume resources such as CPU time, memory, and bandwidth. In this project, cloudlets are used to simulate user jobs running on VMs hosted in a data centre. CloudSim Plus provides a Cloudlet class which allows users to configure cloudlet attributes such as cloudlet length in MIPS, number of Processing Elements which is the number of CPU cores required to execute the task, and input and output sizes in bytes. Cloudlets are submitted to the broker and assigned to VMs by the broker after they are configured.

6. VM Allocation Policy

The VM allocation policy in CloudSim Plus defines the strategy used to allocate and migrate VMs to physical hosts within the data centre. In this project, three custom VM allocation and migration policy is implemented by extending CloudSim Plus's VmAllocationPolicyMigrationAbstract class.

4.3 System Block Diagram

The block diagram in Figure 4-1 illustrates the architecture of a cloud data centre system configured in CloudSim Plus. The system includes multiple layers, including user interaction, VM management and physical infrastructure.

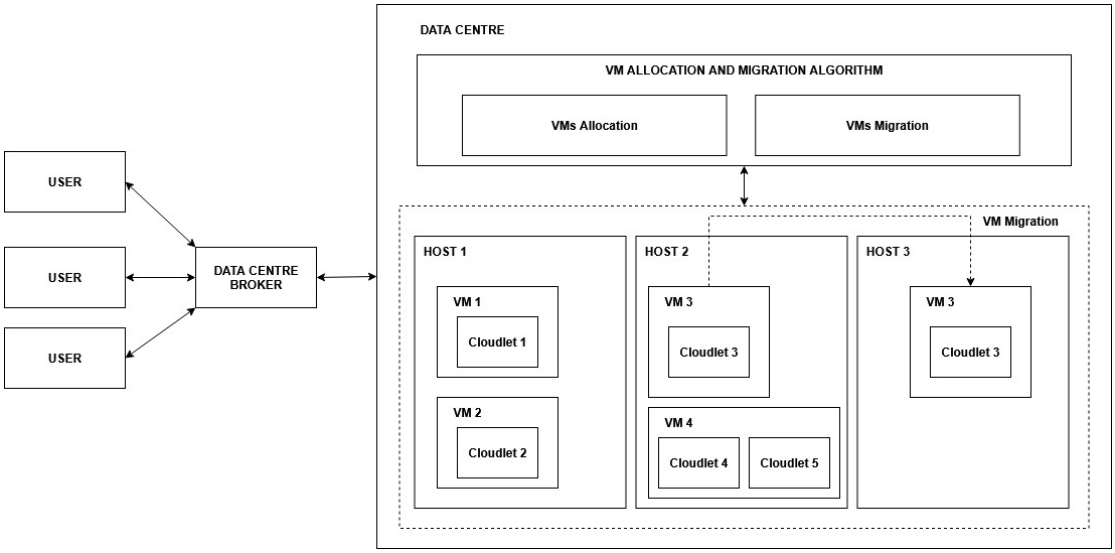

Figure 4-1: Block Diagram of System Configuration in CloudSim Plus.

1. User layer
    - Users interact with the cloud data centre by submitting cloudlets (tasks)
    - The Data Centre Broker acts as the intermediary, managing user requests and assigning cloudlets to appropriate VMs.
2. VM management
    - The system employs a bio-inspired algorithm (ACO, PSO or MGA) to optimise VM allocation and migration. The algorithm receives hosts' and VMs' information such as CPU utilisation, memory utilisation, bandwidth and storage requirements, and computes an optimal migration plan to reduce power consumption.
3. Physical infrastructure
    - The physical infrastructure consists of a data centre that consists of multiple hosts/physical servers. Each server can host multiple VMs and each VMs can execute multiple cloudlets/tasks.

4.4 Visualisation of algorithm behaviour

Figures 4-2 and 4-3 illustrate the VM placement and behaviour of the ACO-, PSO- and MGA-based VM allocation and migration algorithms before and after the migration process. Although each algorithm follows a different optimisation strategy, all three share the same goal: to balance resource utilisation, reduce number of active hosts and reduce overall data center power consumption.

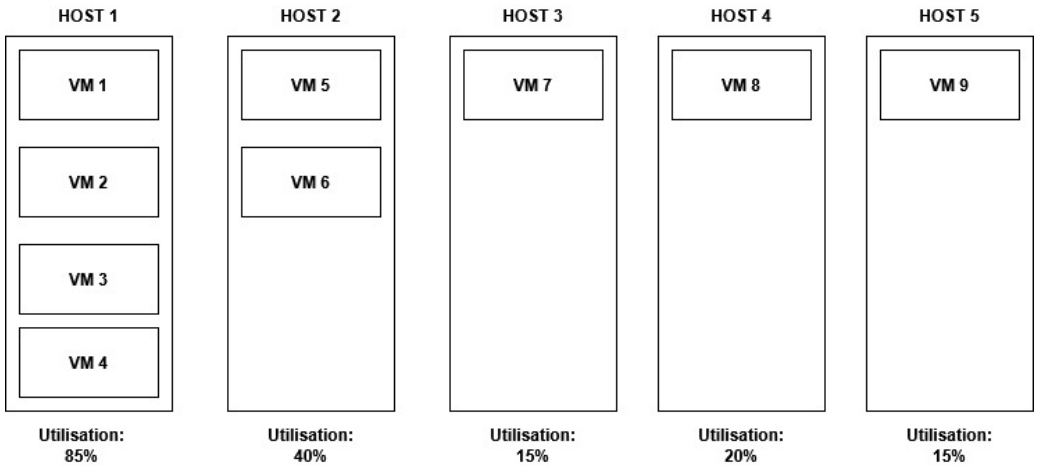

Figure 4-2: VM placement before migration process.

- Before migration:

The data centre consists of five hosts with the following utilisation levels: Host 1 (85%), Host 2 (40%), Host 3 (15%), Host 4 (20%), Host 5 (15%). Here, Host 1 is considered overloaded (utilisation > 80%), while Host 3 and Host 4 are considered underloaded (utilisation < 20%). These source hosts are candidates for VM migration to improve overall resource balance and energy efficiency.

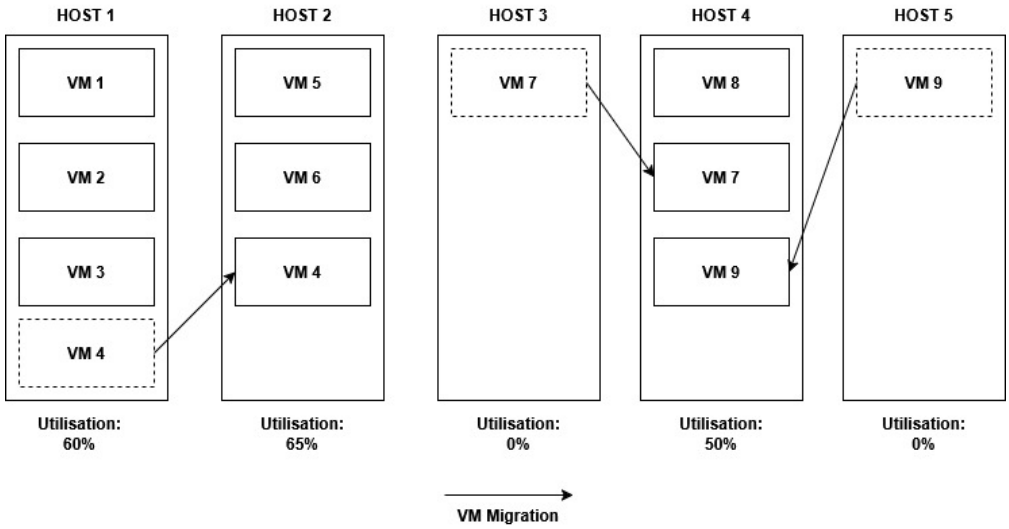

Figure 4-3: VM placement after migration process.

- After migration:

After applying the ACO-based VM allocation and migration algorithm, the new utilisation values are: Host 1 (60%), Host 2 (65%), Host 3 (0%), Host 4 (50%), Host 5 (0%). The algorithm migrated VMs from the overloaded Host 1 and underloaded Hosts 3 and 5 to more suitable target hosts (e.g., Host 2), which had moderate utilisation and could absorb additional workloads without becoming overloaded.

Chapter 5
Experiment/Simulation

This chapter discusses the experimental setup and simulation process used to evaluate the performance of the proposed bio-inspired VM allocation and migration algorithms. It begins with the initial setup and configuration, followed by the verification plan, detailing server specifications, VM configurations, and cloudlet workloads. Two main test cases are considered: a homogeneous data centre setup and a heterogeneous data centre setup, each designed to assess the algorithms under different infrastructure conditions and workload scenarios. Finally, the chapter highlights implementation issues and challenges encountered during the simulation process.

5.1 Initial Setup and Configuration

The preliminary work of this project begins with the initial setup and configuration of the development environment. The project uses Java as the primary programming language and is built using Apache Maven, a widely used project management and build automation tool for Java-based projects. Eclipse IDE was selected as the development platform as it supports Maven integration. The initial setup involves creating a Maven-based project in Eclipse and installing the necessary dependencies through the pom.xml file. This setup ensures that the project structure follows standard conventions, and all required libraries are automatically managed.

- **Creating a Maven Project in Eclipse**: Create a Maven project in Eclipse, specifying details like Group Id, Artifact Id, and Version.
- **Libraries and Dependencies Required for Project**: This project utilises several essential libraries and dependencies and manages them through Maven. The dependencies are CloudSim Plus 8.5.5, org.json version 20220320, and SLF4J (Simple Logging Facade for Java) version 2.0.17. These dependencies are declared in the pom.xml file as shown in Figure 5-1.

```
<dependencies>
  <dependency>
    <groupId>org.cloudsimplus</groupId>
    <artifactId>cloudsimplus</artifactId>
    <version>8.5.5</version>
  </dependency>

  <dependency>
      <groupId>org.json</groupId>
      <artifactId>json</artifactId>
      <version>20220320</version>
  </dependency>

  <dependency>
      <groupId>org.slf4j</groupId>
      <artifactId>slf4j-api</artifactId>
      <version>2.0.17</version>
  </dependency>
</dependencies>
```

Figure 5-1: Dependencies of the Project.

5.2 Verification Plan

A comprehensive verification plan is established to ensure that proposed VM allocation and migration algorithm functions correctly under different conditions and configurations. The test plan is designed to evaluate the algorithm's behaviour with different server specifications, VM types and cloudlet workloads. The primary objective is to verify the effectiveness of the algorithm in terms of power consumption and resource utilisation across varying scenarios.

5.2.1 Server Specification

Three types of servers are simulated in the CloudSim Plus framework to represent a realistic data centre environment. Each server model is based on currently popular server models in the data centre space and has different specifications in terms of CPU cores, Million instructions per second (MIPS), memory and power characteristics. Table 5-1 below shows the server specifications.

Server Model	Dell PowerEdge R740	HPE ProLiant DL380 Gen10	Supermicro SuperServer 1029U-TR4
Processor	Xeon Gold 6248R – 2 x 24 cores (48 total)	Intel Xeon Platinum 8280M – 2 x 28 cores (56 total)	Xeon Silver 4214R – 2 x 12 cores (24 total)

RAM	256 GB	256 GB	256 GB
Storage	15TB SSD	15TB SSD	20TB SSD
Bandwidth	100 Gbps	100 Gbps	100 Gbps
MIPS/Core	~3000 MIPS/Core (3.0GHz)	~2700 MIPS/Core (2.7GHz)	~2400 MIPS/Core (2.4GHz)
Power Characteristics	• Static Power: 300 W • Max Power: 600 W • Startup Power: 400 W • Shutdown Power: 50 W	• Static Power: 350 W • Max Power: 700 W • Startup Power: 450 W • Shutdown Power: 50 W	• Static Power: 200 W • Max Power: 400 W • Startup Power: 300 W • Shutdown Power: 50 W

Table 5-1: Server Specifications.

5.2.2 Virtual Machine Specifications and Cloudlet Configuration

To evaluate the performance of the proposed VM allocation and migration algorithms, three distinct types of VMs, each representing different workload intensities are configured. They are categorized into three VM types – LIGHT, MEDIUM and HIGH and they are designed with varying computational and memory demands. The user workloads in CloudSim Plus are represented as cloudlets and each cloudlet has its own characteristics such as instruction length (in MIPS), processing elements (number of cores), input file size and output file size. In this simulation, cloudlets are also categorized into three types – LIGHT, MEDIUM and HIGH to represent its resource demands. Table 5-2 shows the VM specifications and cloudlet requirements.

Component	Attribute	LIGHT	MEDIUM	HIGH
VM	Number of Processing Elements (Cores)	1	2	4
	MIPS/Core	1000	2000	2400
	RAM (GB)	2	4	8
	Storage (GB)	50	100	200
	Bandwidth (Mbps)	500	1000	2000

Cloudlet	Instruction Length (MIPS)	100000	500000	1000000
	Number of Processing Elements (Cores)	1	2	4
	Input File Size (Bytes)	10000	25000	50000
	Output File Size (Bytes)	2500	5000	10000

Table 5-2: Virtual Machine Specifications and Cloudlet Requirements.

5.2.3 Test Case 1: Homogeneous Data Centre Setup

To evaluate the performance and behaviour of the proposed VM allocation and migration algorithms, a homogeneous data centre configuration is used. This setup ensures that all physical servers/hosts are of the same type, with the same specifications to provide a controlled environment for testing algorithm behaviour under consistent hardware conditions. To assess the system under varying workload intensities, four separate scenarios are defined, each introducing a combination of LIGHT, MEDIUM, and HIGH cloudlets and corresponding VMs. The number of VMs and cloudlets increases progressively across scenarios to simulate different load levels. This test case will be tested on the data centre's baseline VM allocation policy and the three proposed VM allocation and migration algorithms. Table 5-3 shows the parameters and details of Test Case 1.

Parameters	**Details**
Data Centre Type	Homogeneous
Number of physical hosts	20 (To simulate a data centre pod)
Host Model	Dell PowerEdge R740
Host Specifications	(Refer to Server Specifications section)
Number of VMs	• Scenario 1: 100 LIGHT VMs, 50 MEDIUM VMs, 10 HIGH VMs • Scenario 2: 250 LIGHT VMs, 100 MEDIUM VMs, 30 HIGH VMs • Scenario 3: 500 LIGHT VMs, 200 MEDIUM VMs, 60 HIGH VMs • Scenario 4: 700 LIGHT VMs, 250 MEDIUM VMs, 60 HIGH VMs
VM Specifications	(Refer to VM Specifications section)

Number of Cloudlets	• Scenario 1: 100 LIGHT Cloudlets, 100 MEDIUM Cloudlets, 100 HIGH Cloudlets • Scenario 2: 250 LIGHT Cloudlets, 250 MEDIUM Cloudlets, 250 HIGH Cloudlets • Scenario 3: 500 LIGHT Cloudlets, 500 MEDIUM Cloudlets, 500 HIGH Cloudlets • Scenario 4: 800 LIGHT Cloudlets, 600 MEDIUM Cloudlets, 600 HIGH Cloudlets
Cloudlet Specifications	(Refer to Cloudlet Specifications section)
Evaluation Algorithms	• Baseline VM Allocation Policy • Proposed ACO-based Policy • Proposed PSO-based Policy • Proposed MGA-based policy

Table 5-3: Homogeneous Data Centre Test Case.

5.2.4 Test Case 2: Heterogeneous Data Centre Setup

To evaluate the performance and behaviour of the proposed VM allocation and migration algorithms, a heterogeneous data centre configuration is used. In this setup, physical servers are of different models and specifications, including Dell PowerEdge R740, HPE ProLiant DL380 Gen10, and Supermicro SuperServer 1029U-TR4. This reflects a more realistic data centre environment where hardware diversity exists. The same four scenarios in Test Case 1 are reused in Test Case 2. Similarly, this test case will be tested on the data centre's baseline VM allocation policy and the three proposed VM allocation and migration algorithms. Table 5-4 shows the parameters and details of Test Case 2.

Parameters	**Details**
Data Centre Type	Heterogeneous
Number of physical hosts	20 (To simulate a data centre pod)
Host Model	• Dell PowerEdge R740 (8 hosts)

	• HPE ProLiant DL380 Gen10 (6 hosts) • Supermicro SuperServer 1029U-TR4 (6 hosts)
Host Specifications	(Refer to Server Specifications section)
Number of VMs (per scenario)	• Scenario 1: 100 LIGHT VMs, 50 MEDIUM VMs, 10 HIGH VMs • Scenario 2: 250 LIGHT VMs, 100 MEDIUM VMs, 30 HIGH VMs • Scenario 3: 500 LIGHT VMs, 200 MEDIUM VMs, 60 HIGH VMs • Scenario 4: 700 LIGHT VMs, 250 MEDIUM VMs, 60 HIGH VMs
VM Specifications	(Refer to VM Specifications section)
Number of Cloudlets	• Scenario 1: 100 LIGHT Cloudlets, 100 MEDIUM Cloudlets, 100 HIGH Cloudlets • Scenario 2: 250 LIGHT Cloudlets, 250 MEDIUM Cloudlets, 250 HIGH Cloudlets • Scenario 3: 500 LIGHT Cloudlets, 500 MEDIUM Cloudlets, 500 HIGH Cloudlets • Scenario 4: 800 LIGHT Cloudlets, 600 MEDIUM Cloudlets, 600 HIGH Cloudlets
Cloudlet Specifications	(Refer to Cloudlet Specifications section)
Evaluation Algorithms	• Baseline VM Allocation Policy • Proposed ACO-based Policy • Proposed PSO-based Policy • Proposed MGA-based policy

Table 5-4: Heterogeneous Data Centre Test Case.

5.3 Implementation issues and Challenges

Implementing ACO, PSO, and MGA-based VM allocation and migration algorithms in a simulated data centre environment presents several implementation issues and challenges. One of the primary challenges is the simulation complexity. Simulating a realistic data centre environment using CloudSim Plus requires a deep understanding of its architecture, classes, and event-driven simulation model. Misconfigurations in essential components such as hosts, VMs, or cloudlets can lead to inaccurate results, unexpected behaviour, or even simulation failures. Since the three algorithms are tested under both homogeneous and heterogeneous data centre environments, achieving accurate modelling becomes even more demanding.

Another significant challenge lies in the validation and benchmarking process. Evaluating the effectiveness of the ACO, PSO, and MGA algorithms requires extensive testing and comparison against the baseline VM allocation policy provided by CloudSim Plus. To ensure fair and meaningful comparisons, all simulations must be conducted under identical conditions with consistent workloads, VM configurations, and server setups. Designing, running, and analysing these benchmarks is a time-consuming and complex process.

Additionally, the abundance of migration possibilities poses a critical challenge. All three algorithms need to evaluate numerous potential migration plans by considering every possible combination of source hosts, VMs, and target hosts. As the number of VMs and hosts increases, the number of possible solutions grows exponentially, which significantly impacts execution time and scalability. ACO relies on pheromone trails to guide optimal migration, PSO continuously adjusts positions based on personal and global bests, and MGA explores multiple crossover-based combinations. Despite their unique approaches, all three algorithms face computational challenges when handling large-scale data centre environments.

Chapter 6
System Evaluation and Discussion

This chapter presents the evaluation and discussion of the proposed bio-inspired VM allocation and migration algorithms. It begins by defining the system performance metrics used for assessment and then analyses the simulation results across both homogeneous and heterogeneous data centre setups under various workload scenarios. Detailed comparisons are made for CPU utilisation, RAM utilisation, average power consumption, and total power consumption across different scenarios. The chapter further summarises the energy savings achieved by the ACO, PSO, and MGA policies compared to the baseline. Finally, it highlights the limitations of the simulation, evaluates the achievement of the project objectives, and discusses the novel contributions introduced by this work.

6.1 System Performance Definition

System performance in this project is evaluated based on key metrics such as resource utilisation and power consumption in data centre.

- **Resource Utilisation:**

Resource utilisation measures how efficiently resources such as CPU, memory and bandwidth are used in the physical servers, also known as hosts. These metrics are important in data centre/cloud computing environments as they directly impact the performance and energy consumption of data centres. In this project, resource utilisation is measured using the built-in features of the CloudSim Plus framework. The framework provides the utilisation statistics of the CPU usage and memory allocation of each host and virtual machine in the data centre. This enables continuous monitoring and assessment of how well the proposed algorithm in improving resource utilisation across physical servers to ensure better energy efficiency and performance.

- **Power consumption:**

Power consumption is a key metric for this project as the project's main objective is to reduce power consumption and improve energy efficiency in data centres. In this

project, a linear power model is developed to measure the power consumption of each physical servers/hosts in the data centre simulation environment. The study in [3] pointed out that power consumption of servers can be assumed to increase linearly with its CPU utilization while an idle but powered on server can consume around 50-70% of the energy used when operating at maximum capacity. Thus, a power model based on the above information is implemented within the CloudSim Plus framework to estimate the mean power consumption by each host based on its mean CPU utilisation. This analysis helps determine the feasibility of the proposed algorithm in reducing the overall power consumption in data centres.

6.2 Simulation results

To evaluate the effectiveness of the proposed metaheuristic-based VM allocation and migration algorithms: Ant Colony Optimisation (ACO), Particle Swarm Optimisation (PSO), and Modified Genetic Algorithm (MGA), a series of simulations were conducted across two test cases: a homogeneous data centre configuration and a heterogeneous data centre configuration. For each test case, four scenarios were defined with progressively increasing numbers of VMs and cloudlets to simulate varying workload intensities (as detailed in Section 5.2.3: Test Case 1 and Section 5.2.4: Test Case 2). In each scenario, simulations utilised 20 physical servers and were repeated 30 times to ensure statistical reliability. The performance of the three algorithms was compared against the baseline data centre VM allocation policy across key evaluation metrics, including: Average CPU utilisation and Average RAM utilisation of all active servers, Average power consumption of all active servers and Total power consumption of all servers. The results presented in this section provide insights into how each algorithm performs under different workload intensities.

6.2.1 Simulation Results for Homogeneous Data Centre Test Case

This section presents the simulation results for the homogeneous data centre test case, where all physical servers have identical hardware configurations. The performance of the proposed ACO, PSO, and MGA-based VM allocation and migration policies is compared against the baseline VM allocation policy across four workload scenarios. Key metrics such as average CPU and RAM utilisation, average power consumption

and total power consumption are analysed to evaluate the effectiveness of each algorithm in optimising resource usage and improving energy efficiency.

6.2.1.1 Average CPU utilisation of all active servers across different scenarios

The simulation results for average CPU utilisation across all active servers show that two bio-inspired policies (PSO and MGA) consistently outperform the baseline VM allocation policy in all scenarios. The baseline policy records the highest CPU utilisation, reaching 37.85% in Scenario 4, indicating inefficient resource distribution under heavy workloads. Among the proposed algorithms, MGA achieves the lowest CPU utilisation across most scenarios, closely followed by PSO. The ACO-based policy shows comparatively higher utilisation in lighter workload scenarios because it consolidates VMs onto fewer active servers, allowing underutilised hosts to shut down and thereby saving power. Despite this, ACO still demonstrates significant improvements over the baseline in heavier workload scenarios. Overall, all three algorithms enhance resource efficiency, with MGA and PSO achieving the most substantial reductions in CPU utilisation. Table 6-1 and Figure 6-1 show the Average CPU Utilisation (%) of all active servers across different scenarios in Homogeneous Data Centre Setup.

Policy	Scenario	Average CPU Utilisation (%) of all active servers ± stdev
Baseline VM allocation policy	Scenario 1	6.88
	Scenario 2	16.29
	Scenario 3	32.57
	Scenario 4	37.85
ACO-based VM allocation and migration policy	Scenario 1	20.98 ± 7.63
	Scenario 2	17.80 ± 6.55
	Scenario 3	10.95 ± 3.90
	Scenario 4	9.36 ± 2.93
PSO-based VM allocation and migration policy	Scenario 1	3.16 ± 2.68
	Scenario 2	4.76 ± 1.94
	Scenario 3	9.44 ± 2.97
	Scenario 4	7.53 ± 2.78

MGA-based VM allocation and migration policy	Scenario 1	2.26 ± 0.17
	Scenario 2	10.88 ± 1.71E-15
	Scenario 3	7.69 ± 4.38E-15
	Scenario 4	6.32 ± 4.10E-15

Table 6-1: Average CPU Utilisation (%) of all active servers across different scenarios in Homogeneous Data Centre Setup.

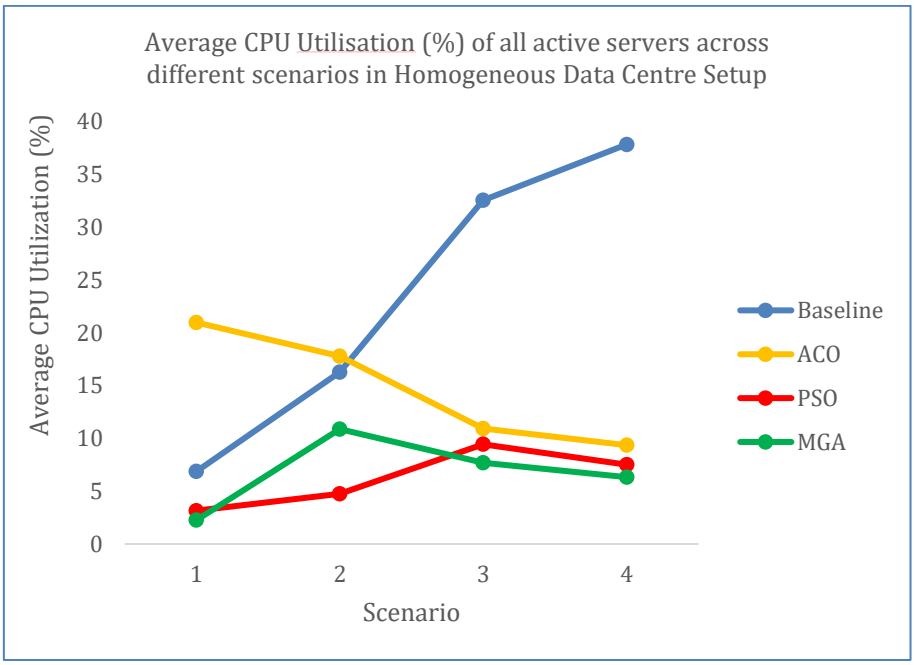

Figure 6-1: Average CPU Utilisation (%) of all active servers across different scenarios in Homogeneous Data Centre Setup.

6.2.1.2 Average RAM utilisation of all active servers across different scenarios

The results for average RAM utilisation across all active servers indicate that the three proposed algorithms (ACO, PSO, and MGA) exhibit distinct behaviours compared to the baseline policy. The baseline VM allocation policy shows steadily increasing RAM usage, reaching 55.56% in Scenario 4, suggesting less efficient resource balancing under heavy workloads. The ACO-based policy maintains relatively high but stable RAM utilisation across all scenarios (53–56%), primarily because it consolidates VMs onto fewer active servers to shut down underutilised hosts and save power. Similarly, PSO achieves higher RAM usage under lighter workloads due to VM consolidation but approaches the baseline levels in higher workload scenarios. MGA shows the lowest utilisation in Scenario 1 but records the highest RAM usage in Scenarios 2–4, indicating

that it prioritises VM consolidation and CPU optimisation over memory efficiency. Overall, while all three algorithms outperform the baseline in CPU optimisation, ACO leverages high RAM utilisation for energy savings, PSO balances both CPU and RAM, and MGA focuses primarily on CPU efficiency. Table 6-2 and Figure 6-2 show the Average RAM Utilisation (%) of all active servers across different scenarios in Homogeneous Data Centre Setup.

Policy	**Scenario**	**Average RAM Utilisation (%) of all active servers ± stdev**
Baseline VM allocation policy	Scenario 1	9.26
	Scenario 2	22.00
	Scenario 3	43.99
	Scenario 4	55.56
ACO-based VM allocation and migration policy	Scenario 1	55.53 ± 1.12
	Scenario 2	53.82 ± 2.84
	Scenario 3	53.41 ± 1.22
	Scenario 4	55.81 ± 1.09
PSO-based VM allocation and migration policy	Scenario 1	13.68 ± 2.65
	Scenario 2	22.88 ± 1.60
	Scenario 3	41.02 ± 0.76
	Scenario 4	54.21 ± 0.35
MGA-based VM allocation and migration policy	Scenario 1	12.33 ± 0.82
	Scenario 2	62.88 ± 2.46E-14
	Scenario 3	62.96 ± 3.67E-14
	Scenario 4	65.51 ± 1.69E-14

Table 6-2: Average RAM Utilisation (%) of all active servers across different scenarios in Homogeneous Data Centre Setup.

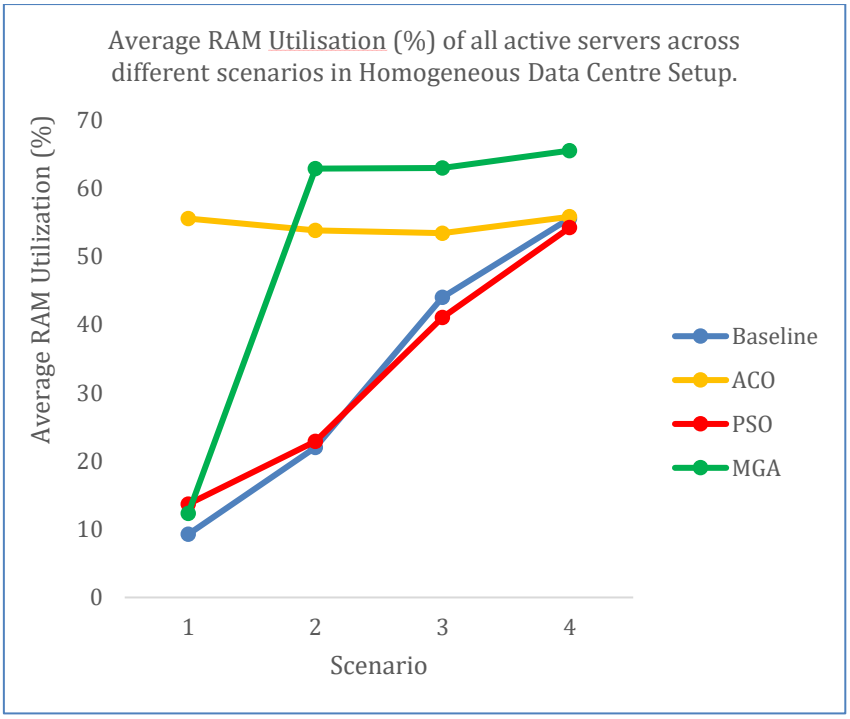

Figure 6-2: Average RAM Utilisation (%) of all active servers across different scenarios in Homogeneous Data Centre Setup.

6.2.1.3 Average power consumption of all active servers across different scenarios

The average power consumption results reflect the direct relationship between CPU utilisation and server energy usage, where higher CPU utilisation translates to higher dynamic power consumption. The baseline policy records the highest overall power usage, peaking at 413.55 W in Scenario 4 due to inefficient VM distribution and a larger number of active servers. The ACO-based policy shows a clear downward trend in power consumption across scenarios (From 362.95 W in Scenario 1 to 328.09 W in Scenario 4), driven by its strategy of consolidating VMs and shutting down underutilised hosts. PSO consistently maintains lower power usage than both the baseline and ACO, achieving balanced CPU loads while keeping energy consumption stable. MGA records the lowest power usage in light workloads (306.79 W in Scenario 1) and remains energy-efficient in heavier scenarios. Table 6-3 and Figure 6-3 below demonstrate the Average Power Consumption (Watts) of all active servers across different scenarios in Homogeneous Data Centre Setup.

Policy	Scenario	Average Power Consumption (Watts) of all active servers ± stdev
Baseline VM allocation policy	Scenario 1	320.63
	Scenario 2	348.86
	Scenario 3	397.71
	Scenario 4	413.55
ACO-based VM allocation and migration policy	Scenario 1	362.95 ± 22.90
	Scenario 2	353.41 ± 19.66
	Scenario 3	332.85 ± 11.71
	Scenario 4	328.09 ± 8.80
PSO-based VM allocation and migration policy	Scenario 1	309.49 ± 8.03
	Scenario 2	314.27 ± 5.82
	Scenario 3	328.33 ± 8.90
	Scenario 4	322.58 ± 8.34
MGA-based VM allocation and migration policy	Scenario 1	306.79 ± 0.52
	Scenario 2	332.65 ± 6.76E-14
	Scenario 3	323.08 ± 1.49E-14
	Scenario 4	318.96 ± 1.05E-13

Table 6-3: Average Power Consumption (Watts) of all active servers across different scenarios in Homogeneous Data Centre Setup.

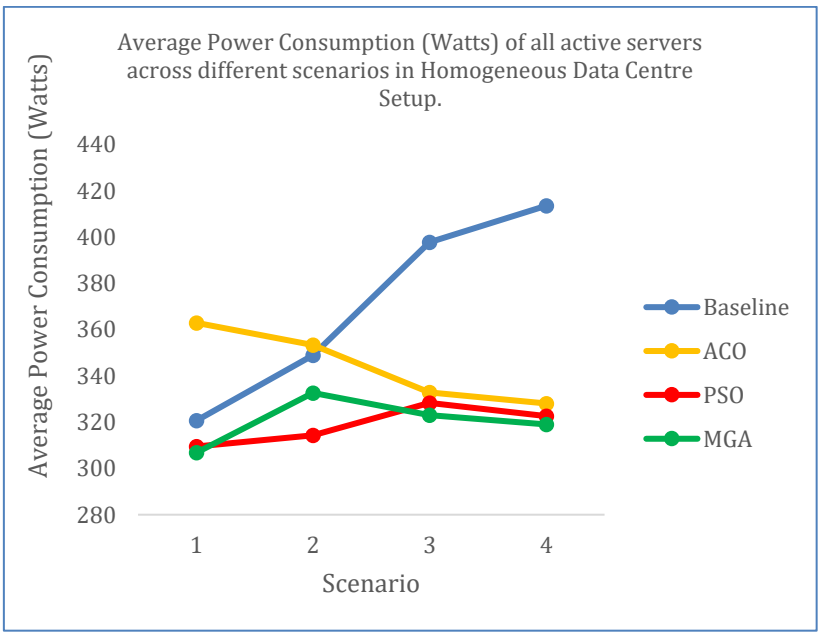

Figure 6-3: Average Power Consumption (Watts) of all active servers across different scenarios in Homogeneous Data Centre Setup.

6.2.1.4 Total power consumption of all servers across different scenarios

The total power consumption results highlight the overall energy efficiency of each policy across increasing workload scenarios. The baseline policy consistently records the highest total power consumption, rising from 788.82 MW in Scenario 1 to 1017.43 MW in Scenario 4, due to inefficient VM allocation and limited host consolidation. The ACO-based policy demonstrates the largest energy savings, especially under light to moderate workloads, starting at just 166.32 MW in Scenario 1 and maintaining lower consumption in Scenarios 2 (346.99 MW) and 3 (620.71 MW). However, in Scenario 4, its consumption rises sharply (851.22 MW) as more servers remain active to handle the heavier workload. PSO shows a balanced trend, consuming moderately across all scenarios by maintaining stable CPU utilisation and avoiding excessive server activation. MGA, meanwhile, performs inconsistently: it consumes high power in Scenario 1 due to less aggressive host consolidation, but achieves significant energy savings in Scenarios 2 and 3 (350.71 MW and 680.42 MW), before increasing again in Scenario 4 (815.50 MW). Table 6-4 and Figure 6-4 demonstrate the Total Power Consumption (MegaWatts) of all servers across different scenarios in Homogeneous Data Centre Setup.

Policy	Scenario	Total Power Consumption (MegaWatts)
Baseline VM allocation policy	Scenario 1	788.82
	Scenario 2	858.28
	Scenario 3	978.47
	Scenario 4	1017.43
ACO-based VM allocation and migration policy	Scenario 1	166.32
	Scenario 2	346.99
	Scenario 3	620.71
	Scenario 4	851.22
PSO-based VM allocation and migration policy	Scenario 1	410.68
	Scenario 2	688.05
	Scenario 3	781.77
	Scenario 4	790.92
	Scenario 1	764.44
	Scenario 2	350.71

MGA-based VM allocation and migration policy	Scenario 3	680.42
	Scenario 4	815.50

Table 6-4: Total Power Consumption (MegaWatts) of all servers across different scenarios in Homogeneous Data Centre Setup.

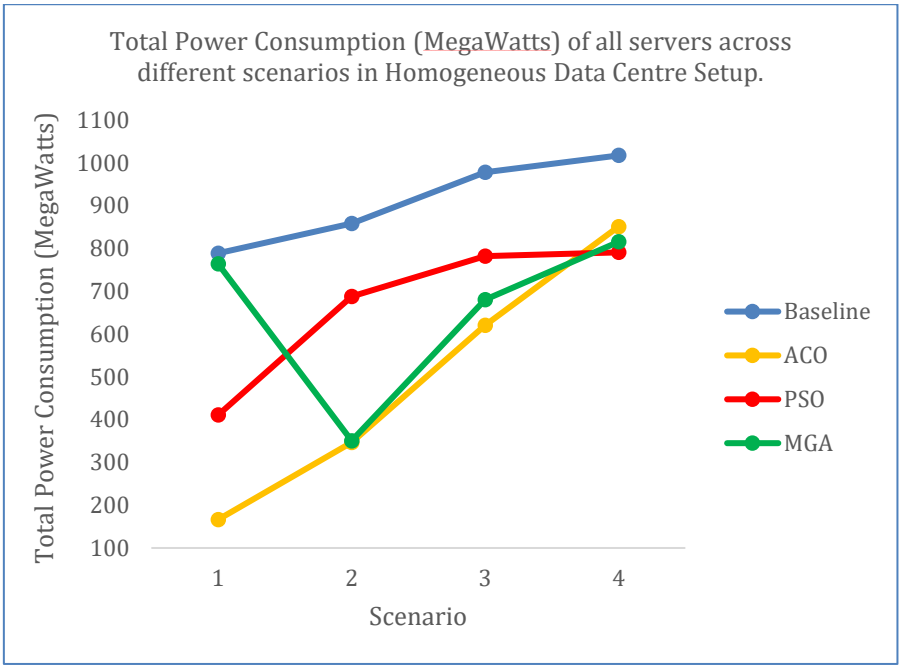

Figure 6-4: Total Power Consumption (MegaWatts) of all servers across different scenarios in Homogeneous Data Centre Setup.

6.2.2 Simulation Results for Heterogeneous Data Centre Test Case

This section presents the simulation results for the heterogeneous data centre test case, where physical servers have different hardware configurations, including varying processing capacities, memory sizes, and power characteristics. The performance of the proposed ACO, PSO, and MGA-based VM allocation and migration policies is compared against the baseline VM allocation policy across four workload scenarios. Key performance metrics, including average CPU and RAM utilisation, average power consumption and total power consumption, are analysed to evaluate the effectiveness of each algorithm in optimising resource allocation, balancing workloads across diverse server types, and improving overall energy efficiency in a heterogeneous environment.

6.2.2.1 Average CPU utilisation of all active servers across different scenarios

In the heterogeneous data centre configuration, where servers differ in terms of processing capacity, memory, and power profiles, the average CPU utilisation patterns vary significantly across the four workload scenarios. The baseline policy shows a steady increase in utilisation as workloads grow, peaking at 45.18% in Scenario 4. In contrast, the ACO-based policy maintains moderate utilisation levels between 7.82% and 12.23%, primarily due to shutting down underutilised servers and migrating VMs more aggressively to fewer active hosts. The PSO-based policy demonstrates better balancing under light workloads (Scenarios 1 and 2) but maintains relatively low utilisation in Scenarios 3 and 4. Meanwhile, the MGA-based policy achieves the lowest CPU utilisation across most scenarios, especially under higher workloads. Table 6-5 and Figure 6-5 below show the Average CPU Utilisation (%) of all active servers across different scenarios in Heterogeneous Data Centre Setup.

Policy	Scenario	Average CPU Utilisation (%) of all active servers ± stdev
Baseline VM allocation policy	Scenario 1	6.70
	Scenario 2	17.33
	Scenario 3	34.18
	Scenario 4	45.18
ACO-based VM allocation and migration policy	Scenario 1	12.23 ± 4.32
	Scenario 2	12.05 ± 4.18
	Scenario 3	7.82 ± 2.34
	Scenario 4	11.70 ± 3.74
PSO-based VM allocation and migration policy	Scenario 1	5.76 ± 2.51
	Scenario 2	8.72 ± 3.08
	Scenario 3	19.89 ± 5.32
	Scenario 4	6.99 ± 0.15
MGA-based VM allocation and migration policy	Scenario 1	14.08 ± 8.88E-15
	Scenario 2	7.28 ± 8.88E-16
	Scenario 3	5.56 ± 9.76E-16
	Scenario 4	4.06 ± 1.78E-15

Table 6-5: Average CPU Utilisation (%) of all active servers across different scenarios in Heterogeneous Data Centre Setup.

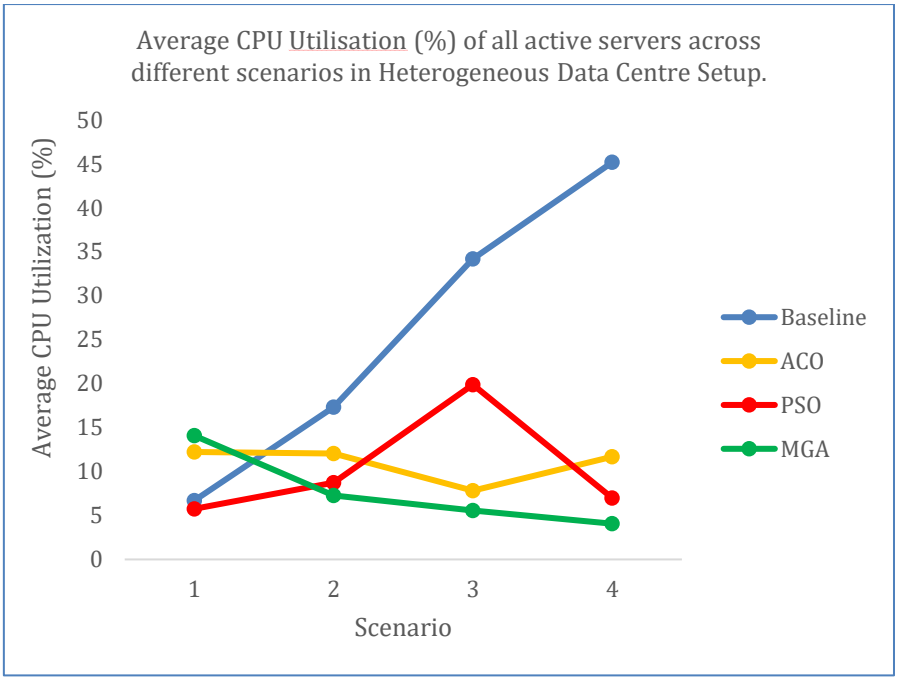

Figure 6-5: Average CPU Utilisation (%) of all active servers across different scenarios in Heterogeneous Data Centre Setup.

6.2.2.2 Average RAM utilisation of all active servers across different scenarios

In the heterogeneous data centre scenario, where servers vary in hardware specifications, the average RAM utilisation trends differ significantly across the four workload levels. The baseline policy shows a steady rise in RAM usage, starting at 9.28% in Scenario 1 and reaching 55.57% in Scenario 4. This reflects its static VM placement strategy, which fails to consolidate workloads efficiently, resulting in higher active server counts under heavy demand. The ACO-based policy achieves higher and more balanced RAM utilisation in lighter workloads (24.12% in Scenario 1 and 35.23% in Scenario 2) by migrating VMs aggressively and shutting down underutilised servers. As workloads increase, RAM usage remains controlled (42.82% in Scenario 3 and 45.79% in Scenario 4), showing its ability to maintain efficiency even under heavier loads. The PSO-based policy demonstrates stable and moderate RAM utilisation across all scenarios. It performs close to the baseline under higher workloads (39.51% and 45.22% in Scenarios 3 and 4) but shows higher RAM usage in lighter workloads (11.10% in Scenario 1 vs. 9.28% baseline, and 22.14% vs 22.04% in Scenario 2) due to VM consolidation. The MGA-based policy exhibits a more adaptive behaviour. It achieves the lowest RAM utilisation in Scenario 2 (8.49%). However, under heavier workloads, it utilises more available memory than other policies (51.86% in Scenario

3 and 55.69% in Scenario 4). Table 6-6 and Figure 6-6 show the Average RAM Utilisation (%) of all active servers across different scenarios in Heterogeneous Data Centre Setup.

Policy	Scenario	Average RAM Utilisation (%) of all active servers ± stdev
Baseline VM allocation policy	Scenario 1	9.28
	Scenario 2	22.04
	Scenario 3	44.08
	Scenario 4	55.57
ACO-based VM allocation and migration policy	Scenario 1	24.12 ± 2.12
	Scenario 2	35.23 ± 2.01
	Scenario 3	42.82 ± 1.03
	Scenario 4	45.79 ± 0.56
PSO-based VM allocation and migration policy	Scenario 1	11.10 ± 1.07
	Scenario 2	22.14 ± 0.71
	Scenario 3	39.51 ± 1.93
	Scenario 4	45.22 ± 0.34
MGA-based VM allocation and migration policy	Scenario 1	26.41 ± 1.39E-14
	Scenario 2	8.49 ± 7.11E-15
	Scenario 3	51.86 ± 1.40E-14
	Scenario 4	55.69 ± 2.34E-14

Table 6-6: Average RAM Utilisation (%) of all active servers across different scenarios in Heterogeneous Data Centre Setup.

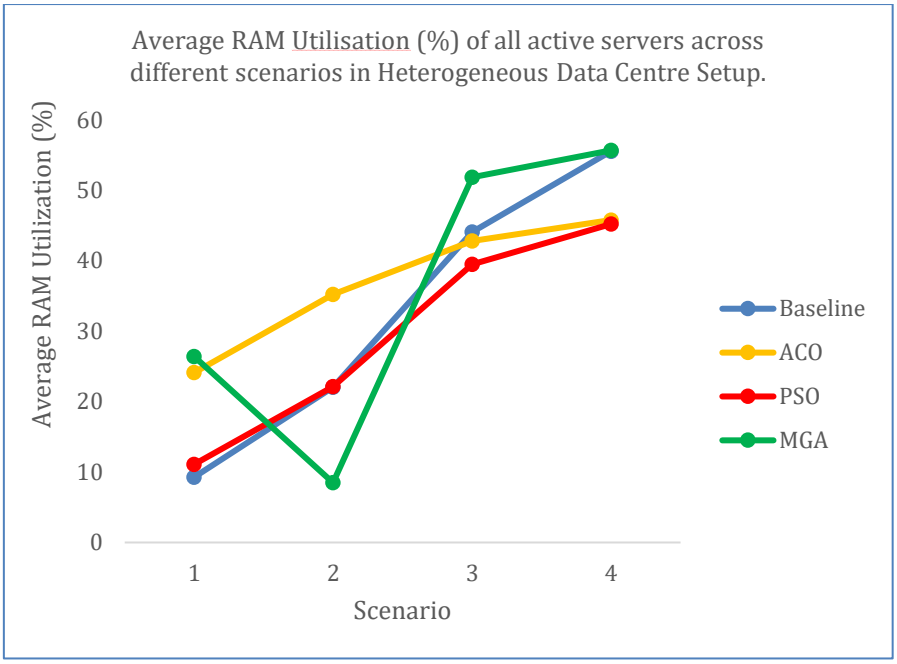

Figure 6-6: Average RAM Utilisation (%) of all active servers across different scenarios in Heterogeneous Data Centre Setup.

6.2.2.3 Average power consumption of all active servers across different scenarios

While the servers in the Heterogeneous Data Centre have different specifications and configurations, the average power consumption is still largely related to the average CPU utilisation. Thus, the baseline policy consistently exhibits the highest energy usage, rising from 306.82 W in Scenario 1 to 417.18 W in Scenario 4. This is primarily due to its static allocation approach, which keeps more servers active even under lighter workloads, leading to unnecessary energy overheads. The ACO-based policy achieves the lowest and most stable average power consumption among all strategies. By aggressively consolidating VMs and shutting down underutilised hosts, it reduces the average power usage to 242.10 W in Scenario 1 and maintains energy efficiency as workloads grow. This demonstrates ACO's strong capability to balance workload placement with power savings. The PSO-based policy achieves moderate energy savings compared to the baseline, with 282.55 W in Scenario 1 and 302.70 W in Scenario 2. However, it shows less consistent optimisation at higher workloads, consuming 325.69 W in Scenario 3 before slightly dropping to 305.16 W in Scenario 4. The MGA-based policy performs competitively with ACO in light workloads but demonstrates a more adaptive approach under increasing demand. While its energy

consumption rises moderately (298.64 W in Scenario 2 and 299.21 W in Scenario 4), it remains lower than both the baseline and PSO. The Average Power Consumption (Watts) of all active servers across different scenarios in Heterogeneous Data Centre Setup is shown in Table 6-7 and Figure 6-7 below.

Policy	**Scenario**	**Average Power Consumption (Watts) of all active servers ± stdev**
Baseline VM allocation policy	Scenario 1	306.82
	Scenario 2	336.66
	Scenario 3	385.99
	Scenario 4	417.18
ACO-based VM allocation and migration policy	Scenario 1	242.10 ± 8.69
	Scenario 2	276.28 ± 12.24
	Scenario 3	299.04 ± 7.32
	Scenario 4	319.93 ± 9.61
PSO-based VM allocation and migration policy	Scenario 1	282.55 ± 8.42
	Scenario 2	302.70 ± 8.71
	Scenario 3	325.69 ± 14.68
	Scenario 4	305.16 ± 0.98
MGA-based VM allocation and migration policy	Scenario 1	242.66 ± 1.36E-13
	Scenario 2	298.64 ± 1.71E-13
	Scenario 3	290.22 ± 1.14E-13
	Scenario 4	299.21 ± 3.66E-14

Table 6-7: Average Power Consumption (Watts) of all active servers across different scenarios in Heterogeneous Data Centre Setup.

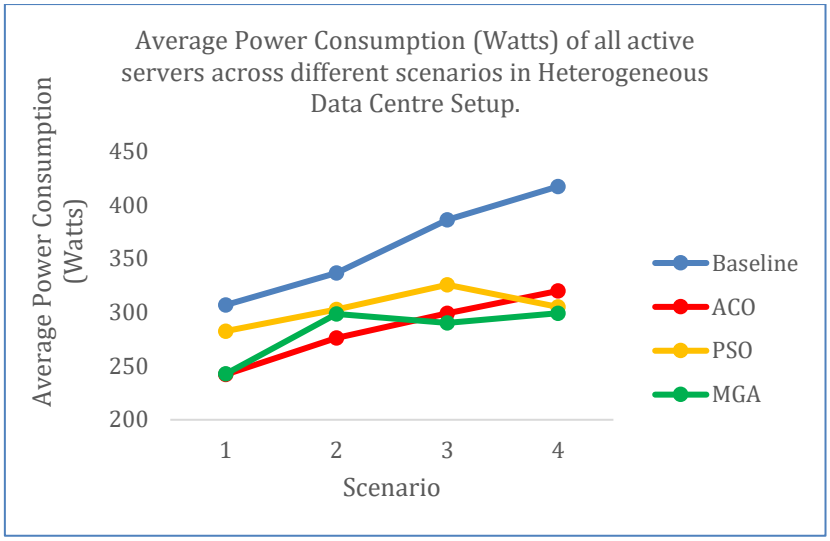

Figure 6-7: Average Power Consumption (Watts) of all active servers across different scenarios in Heterogeneous Data Centre Setup.

6.2.2.4 Total power consumption of all servers across different scenarios

In the heterogeneous data centre, the total power consumption across all servers highlights how each policy manages energy efficiency under increasing workloads. The baseline policy consistently records the highest overall energy usage, rising from 922.79 MW in Scenario 1 to 1,160.88 MW in Scenario 3, before slightly dropping to 1,026.38 MW in Scenario 4. Its static VM allocation approach keeps many servers active even during low demand, leading to significant energy inefficiency. The ACO-based policy delivers the most efficient energy utilisation across all scenarios. It consumes only 211.31 MW in Scenario 1 and 389.83 MW in Scenario 2, which represents a reduction of over 60% compared to the baseline. Even under heavier workloads, ACO maintains lower consumption (791.89 MW in Scenario 3 and 724.87 MW in Scenario 4) by consolidating VMs effectively and shutting down underutilised hosts.

The PSO-based policy achieves moderate savings compared to the baseline across all scenarios, demonstrating stable performance under varying workloads. The MGA-based policy demonstrates competitive performance in light workloads (209.35 MW in Scenario 1, nearly matching ACO) but shows less stability under heavier demand, where its total power consumption spikes to 712.80 MW in Scenario 2 and 899.87 MW in Scenario 4. The Total Power Consumption (MegaWatts) of all servers across

different scenarios in Heterogeneous Data Centre Setup are recorded in Table 6-8 and Figure 6-8 below.

Policy	Scenario	Total Power Consumption (MegaWatts)
Baseline VM allocation policy	Scenario 1	922.79
	Scenario 2	1012.51
	Scenario 3	1160.88
	Scenario 4	1026.38
ACO-based VM allocation and migration policy	Scenario 1	211.31
	Scenario 2	389.83
	Scenario 3	791.89
	Scenario 4	724.87
PSO-based VM allocation and migration policy	Scenario 1	496.42
	Scenario 2	709.01
	Scenario 3	693.90
	Scenario 4	741.39
MGA-based VM allocation and migration policy	Scenario 1	209.35
	Scenario 2	712.80
	Scenario 3	742.03
	Scenario 4	899.87

Table 6-8: Total Power Consumption (MegaWatts) of all servers across different scenarios in Heterogeneous Data Centre Setup.

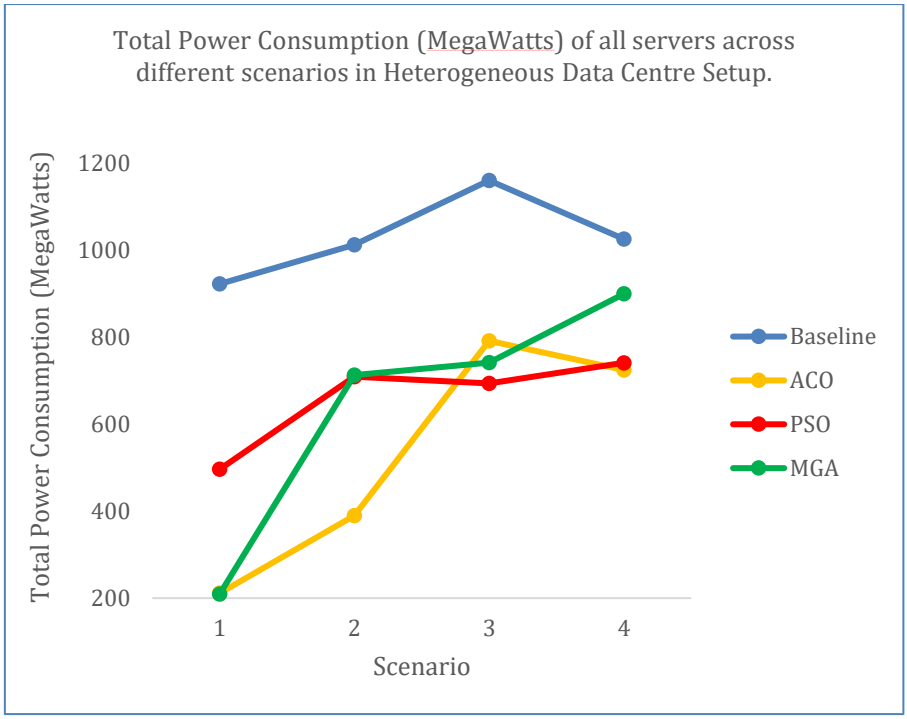

Figure 6-8: Total Power Consumption (MegaWatts) of all servers across different scenarios in Heterogeneous Data Centre Setup.

6.2.3 Summary of Simulation Results

This section summarises the energy savings achieved by the proposed ACO, PSO, and MGA-based VM allocation and migration policies compared to the baseline policy across both homogeneous and heterogeneous data centre setups. The results highlight the extent to which each algorithm reduces total power consumption under varying workload scenarios.

6.2.3.1 Energy Savings Achieved by ACO Policy

In the homogeneous data centre, the ACO policy consistently demonstrates significant reductions in total power consumption across all scenarios. Under light workloads (Scenario 1), power usage drops from 788.82 MW in the baseline to 166.32 MW with ACO, resulting in a savings of 622.50 MW or 78.92%. Similarly, under moderate workloads (Scenario 2), the policy achieves a 59.57% reduction, consuming only 346.99 MW compared to the baseline's 858.28 MW. Although the energy savings gradually decrease as workloads increase, ACO still delivers substantial reductions of 36.56% and 16.34% in Scenarios 3 and 4, respectively. On average, ACO achieves a 47.85% reduction in total energy consumption compared to the baseline, highlighting

its efficiency in homogeneous environments. The results are shown in Table 6-9 and Figure 6-9.

Scenario	Total power consumption of all servers for baseline policy (MW)	Total power consumption of all servers for ACO policy (MW)	Energy Saved (MW)	Percent of energy saved (%)
1	788.82	166.32	622.50	78.92
2	858.28	346.99	511.28	59.57
3	978.47	620.71	357.76	36.56
4	1017.43	851.22	166.21	16.34
Average	910.75	496.31	414.44	47.85

Table 6-9: Energy savings achieved by ACO Policy compared to baseline policy in homogeneous data centre setup.

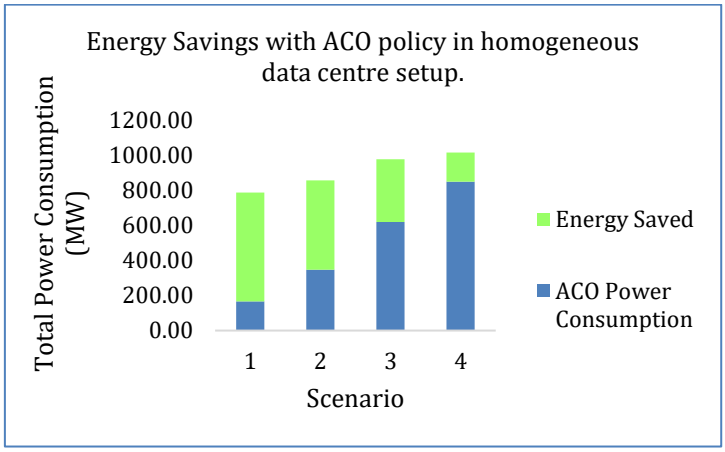

Figure 6-9: Energy Savings with ACO policy in homogeneous data centre setup.

In the heterogeneous data centre, a similar trend is observed. Under light workloads (Scenario 1), ACO reduces consumption from 922.79 MW to just 211.31 MW, achieving a savings of 711.48 MW or 77.10%. In Scenario 2, the savings remain strong at 61.50%, dropping from 1,012.51 MW to 389.83 MW. However, as workloads intensify, the percentage of energy saved decreases: 31.79% in Scenario 3 and 29.38% in Scenario 4. Across all scenarios, the ACO policy achieves an average energy saving

of 49.94% compared to the baseline. Table 6-10 and Figure 6-10 below demonstrates the results.

Scenario	Total power consumption of all servers for baseline policy (MW)	Total power consumption of all servers for ACO policy (MW)	Energy Saved (MW)	Percent of energy saved (%)
1	922.79	211.31	711.48	77.10
2	1012.51	389.83	622.68	61.50
3	1160.88	791.89	368.99	31.79
4	1026.38	724.87	301.51	29.38
Average	1030.64	529.47	501.17	49.94

Table 6-10: Energy savings achieved by ACO Policy compared to baseline policy in heterogeneous data centre setup.

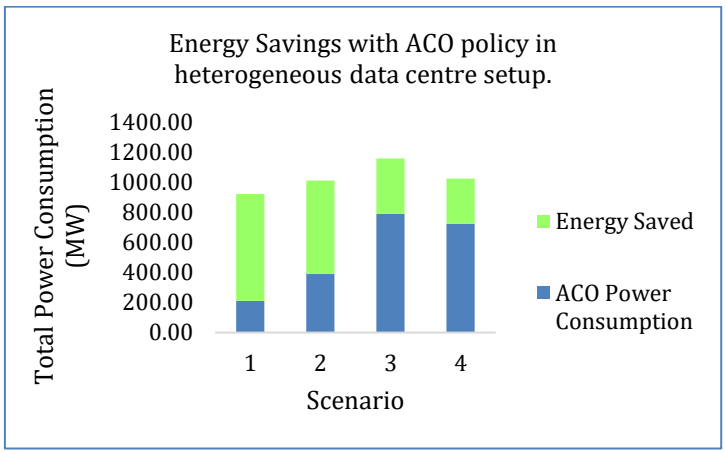

Figure 6-10: Energy Savings with ACO policy in heterogeneous data centre setup.

Overall, these results confirm that the ACO-based VM allocation and migration policy is highly effective in improving energy efficiency by consolidating workloads and shutting down underutilised hosts. While both environments benefit significantly, heterogeneous data centres allow ACO to achieve slightly greater average energy savings because high-power servers can be selectively turned off when underutilised, amplifying the overall reduction in energy consumption.

6.2.3.2 Energy Savings Achieved by PSO Policy

In the homogeneous data centre, the PSO policy achieves moderate energy reductions across all workload scenarios. Under light workloads (Scenario 1), power consumption drops from 788.82 MW in the baseline to 410.68 MW with PSO, resulting in savings of 378.15 MW or 47.94%. However, under medium to heavy workloads (Scenarios 2, 3, and 4), the savings are significantly lower, at 19.83%, 20.10%, and 22.26%, respectively. This reduction occurs because fewer servers can be consolidated and shut down when utilisation levels increase. On average, the PSO policy achieves 27.53% energy savings compared to the baseline in homogeneous environments. Table 6-11 and Figure 6-11 demonstrate the energy savings achieved by PSO Policy compared to baseline policy in homogeneous data centre setup.

Scenario	Total power consumption of all servers for baseline policy (MW)	Total power consumption of all servers for PSO policy (MW)	Energy Saved (MW)	Percent of energy saved (%)
1	788.82	410.68	378.15	47.94
2	858.28	688.05	170.23	19.83
3	978.47	781.77	196.70	20.10
4	1017.43	790.92	226.51	22.26
Average	910.75	667.85	242.90	27.53

Table 6-11: Energy savings achieved by PSO Policy compared to baseline policy in homogeneous data centre setup.

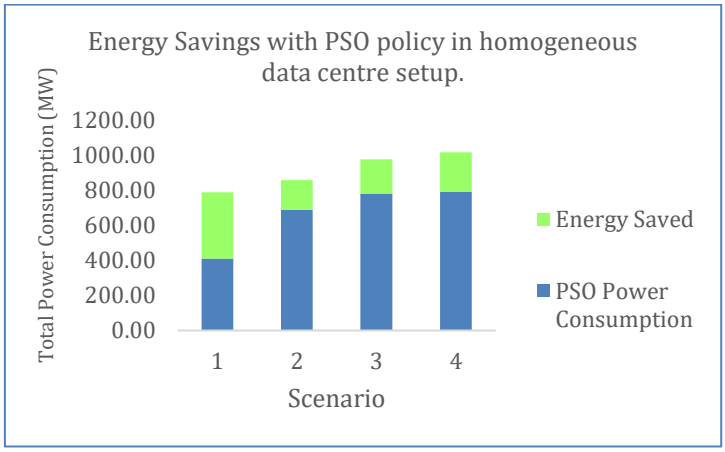

Figure 6-11: Energy Savings with PSO policy in homogeneous data centre setup.

In the heterogeneous data centre setup, the PSO policy performs better overall, benefiting from the ability to leverage differences in server power ratings. Under light workloads (Scenario 1), energy consumption reduces from 922.79 MW to 496.42 MW, saving 426.37 MW or 46.20%. For Scenario 2, the savings are 29.98%, while in Scenario 3, PSO achieves its highest reduction of 40.23%, lowering energy use from 1,160.88 MW to 693.90 MW. In Scenario 4, the savings drop slightly to 27.77% due to high utilisation limiting consolidation opportunities. On average, the PSO policy delivers 36.04% energy savings in heterogeneous setups, which is a notable improvement compared to the homogeneous configuration. Table 6-12 and Figure 6-12 shows the energy savings achieved by PSO Policy compared to baseline policy in heterogeneous data centre setup.

Scenario	**Total power consumption of all servers for baseline policy (MW)**	**Total power consumption of all servers for PSO policy (MW)**	**Energy Saved (MW)**	**Percent of energy saved (%)**
1	922.79	496.42	426.37	46.20
2	1012.51	709.01	303.50	29.98
3	1160.88	693.90	466.98	40.23
4	1026.38	741.39	284.99	27.77
Average	1030.64	660.18	370.46	36.04

Table 6-12: Energy savings achieved by PSO Policy compared to baseline policy in heterogeneous data centre setup.

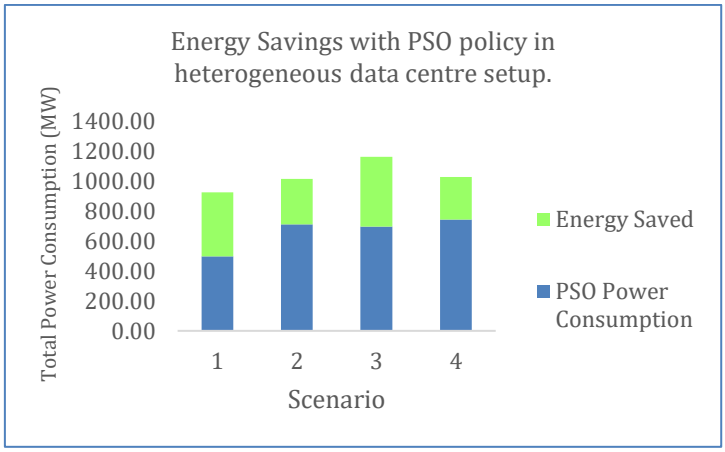

Figure 6-12: Energy Savings with PSO policy in heterogeneous data centre setup.

6.2.3.3 Energy Savings Achieved by MGA Policy

In the homogeneous data centre, MGA exhibits inconsistent energy savings across the four workload scenarios. Under Scenario 1 (light workload), the savings are minimal at just 3.09%, with power consumption dropping slightly from 788.82 MW to 764.44 MW. However, in Scenario 2 (moderate workload), MGA achieves its highest reduction, lowering consumption from 858.28 MW to 350.71 MW, a saving of 59.14%. In Scenario 3, savings stand at 30.46%, while Scenario 4 achieves only 19.85% due to higher utilisation levels limiting VM consolidation. On average, the MGA policy delivers 28.13% energy savings in homogeneous environments, comparable to PSO's performance but significantly lower than ACO's.

Scenario	Total power consumption of all servers for baseline policy (MW)	Total power consumption of all servers for MGA policy (MW)	Energy Saved (MW)	Percent of energy saved (%)
1	788.82	764.44	24.38	3.09
2	858.28	350.71	507.57	59.14
3	978.47	680.42	298.04	30.46
4	1017.43	815.50	201.93	19.85
Average	910.75	652.77	257.98	28.13

Table 6-13: Energy savings achieved by MGA Policy compared to baseline policy in homogeneous data centre setup.

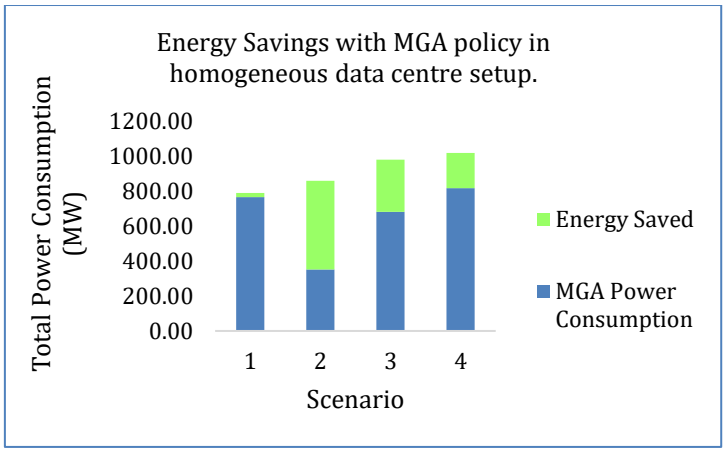

Figure 6-13: Energy Savings with MGA policy in homogeneous data centre setup.

In the heterogeneous data centre (Table 6-14), MGA performs better overall, benefiting from the availability of servers with diverse power ratings. In Scenario 1, energy consumption is reduced drastically from 922.79 MW to just 209.35 MW, resulting in 77.31% savings, which is the highest across all policies and setups. However, the gains are less dramatic under higher workloads: 29.60% savings in Scenario 2, 36.08% in Scenario 3, and only 12.33% in Scenario 4 due to limited consolidation opportunities under heavy utilisation. On average, MGA achieves 38.83% energy savings in heterogeneous setups, an improvement over the homogeneous case and slightly higher than PSO's 36.04%, but still trailing ACO's performance.

Scenario	**Total power consumption of all servers for baseline policy (MW)**	**Total power consumption of all servers for MGA policy (MW)**	**Energy Saved (MW)**	**Percent of energy saved (%)**
1	922.79	209.35	713.44	77.31
2	1012.51	712.80	299.72	29.60
3	1160.88	742.03	418.85	36.08
4	1026.38	899.87	126.51	12.33
Average	1030.64	641.01	389.63	38.83

Table 6-14: Energy savings achieved by MGA Policy compared to baseline policy in heterogeneous data centre setup.

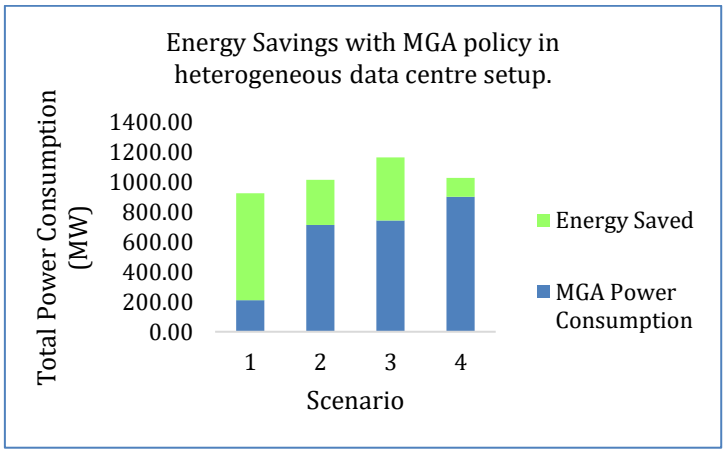

Figure 6-14: Energy Savings with MGA policy in heterogeneous data centre setup.

6.3 Limitations of Simulation

This project faces several limitations that may affect the accuracy and generalizability of its results. Firstly, the data centre topology is simplified by assuming a flat server placement, where all servers are positioned side-by-side without considering the hierarchical structure of real-world architectures. This abstraction overlooks important network-related constraints like latency, bandwidth bottlenecks, and inter-rack communication overheads, which can significantly influence VM migration costs and energy efficiency. Secondly, simulation inaccuracies arise from CloudSim Plus's abstractions. While it is flexible, it may fail to fully capture real-world complexities. Key factors such as hardware heterogeneity, I/O delays, and network congestion are either simplified or ignored, potentially leading to overestimated energy savings and underestimated migration costs. Lastly, the energy models used in the simulations account only for server power consumption, neglecting significant contributors such as cooling systems, power distribution losses, and support infrastructure. Consequently, the reported energy savings may not fully represent the total operational power efficiency achievable in real data centres.

6.4 Objectives Evaluation

The objectives of this project were successfully achieved through the design, deployment, and evaluation of a comprehensive simulation platform and bio-inspired optimisation algorithms for energy-efficient data centre management. For the first objective, a simulation platform was designed and implemented using the CloudSim Plus framework, where its extensive libraries and classes are used to model data centre components and performance metrics. The platform was configured to simulate both homogeneous and heterogeneous data centre setups, allowing for a thorough evaluation of algorithms under environments with identical and diverse server configurations. Additionally, four distinct workload scenarios were created to represent varying levels of resource demand. The platform enabled accurate modelling of physical hosts, VMs, workloads, and VM migration events, while tracking critical metrics such as CPU utilisation, RAM utilisation, and power consumption. By providing a scalable and flexible simulation environment, the first objective was fully accomplished, enabling the evaluation of power management methods without the need for costly physical infrastructure.

For the second objective, three bio-inspired optimisation algorithms were successfully implemented and deployed as custom VM allocation and migration policies within the simulation platform. They are based on Ant Colony Optimisation (ACO), Particle Swarm Optimisation (PSO), and Modified Genetic Algorithm (MGA) respectively. Through comprehensive testing across homogeneous and heterogeneous data centre setups under various workload scenarios, all three algorithms demonstrated substantial improvements in energy efficiency and resource utilisation compared to the baseline static allocation policy. Among them, ACO consistently achieved the most significant power savings, particularly under high-load conditions, making it highly suitable for large-scale and energy-conscious environments. MGA delivered strong and balanced performance, proving effective across both homogeneous and heterogeneous infrastructures, while PSO offered competitive results in specific scenarios despite being slightly less efficient overall. These findings confirm that the objective of deploying bio-inspired algorithms to enhance power management and resource allocation in data centres has been successfully achieved.

Overall, the project successfully achieved its objectives by developing a robust simulation platform and deploying multiple bio-inspired optimisation algorithms to

evaluate their effectiveness in optimising power consumption and resource allocation. The comparative results highlight the strengths of each algorithm, demonstrating their potential applicability to different types of data centre environments and workload conditions.

6.5 Novel Aspects of this project

The novelty of this project lies in the application and comparison of three bio-inspired algorithms: Ant Colony Optimisation (ACO), Particle Swarm Optimisation (PSO), and a Modified Genetic Algorithm (MGA), for intelligent VM allocation and migration within a simulated data centre environment. Unlike conventional static or rule-based heuristic approaches, these algorithms dynamically explore, evaluate, and select optimal migration plans in response to fluctuating workloads and resource demands.

The ACO-based VM allocation and migration introduces a bio-inspired mechanism where artificial "ants" construct migration plans by considering combinations of source hosts, VMs, and target hosts. The decision-making process is guided by pheromone trails and heuristic information, enabling the system to balance workload distribution or consolidate VMs to minimise power consumption while optimising resource utilisation. The PSO-based optimisation mimics the swarm intelligence observed in bird flocking or fish schooling. Each "particle" represents a VM-to-host mapping, and particles iteratively update their positions based on both their personal best and the global best solution. This behaviour allows PSO to efficiently search and converge toward near-optimal VM placement strategies, adapting effectively to dynamic workload variations. The MGA-based VM allocation introduces a problem-specific crossover strategy that enhances traditional genetic algorithms. The algorithm selectively migrates VMs from over-utilised or under-utilised hosts in the low-fitness parent solution to the high-fitness parent configuration, resulting in improved offspring solutions. This modified crossover mechanism accelerates convergence and improves exploration while maintaining solution diversity.

By testing and comparing these three algorithms in the same simulation environment, this project provides a new comparative analysis of VM allocation and migration strategies. The goal is to find out which approach works best for reducing power consumption, improving resource usage, and making the data centre more efficient when workloads keep changing.

Chapter 7
Conclusion and Recommendation

This chapter concludes the project by summarising the key findings, outcomes, and contributions made throughout the study. It highlights how the proposed ACO, PSO, and MGA-based VM allocation and migration algorithms were evaluated across homogeneous and heterogeneous data centre setups under various workload scenarios, demonstrating significant improvements in energy efficiency and resource utilisation compared to baseline policies. Furthermore, the chapter provides recommendations for future work, including potential enhancements to the algorithms, integration with predictive and hybrid techniques, and the adoption of more realistic data centre models to further improve performance and applicability in real-world environments.

7.1 Summary of the project

The rapid growth in demand for data centre services has significantly increased power consumption, resulting in substantial economic, environmental, and operational challenges. While virtualisation has enabled efficient resource sharing through Virtual Machines (VMs), determining the optimal placement and migration of VMs across physical servers remains an NP-hard problem. Traditional techniques such as static server consolidation and workload balancing have provided improvements but struggle to cope with the scale, heterogeneity, and dynamic resource demands of modern data centres. Therefore, there is a pressing need for adaptive, intelligent, and energy-aware VM allocation strategies.

This project investigates and evaluates three bio-inspired optimisation algorithms: Ant Colony Optimisation (ACO), Particle Swarm Optimisation (PSO), and Modified Genetic Algorithm (MGA), to address VM allocation and migration challenges in homogeneous and heterogeneous data centre environments. The ACO-based VM allocation and migration policy leverages the foraging behaviour of ants to explore efficient migration paths, identifying overloaded and underloaded hosts and selecting optimal VMs for migration to minimise idle server usage and improve overall energy efficiency. The PSO-based policy models the social behaviour of swarms, where candidate solutions (particles) iteratively converge towards optimal VM placement by

balancing exploration and exploitation of the solution space. The MGA-based policy incorporates a problem-specific crossover mechanism called VM placement, where VMs from under- or over-utilised hosts in low-fitness solutions are migrated to higher-fitness configurations, enabling more effective resource utilisation while reducing power consumption.

Key Findings of the project:

- The ACO-based policy achieved the highest energy savings, reducing total power consumption by an average of 47.85% in homogeneous setups and 49.94% in heterogeneous setups compared to the baseline.
- The MGA-based policy delivered significant improvements as well, achieving average energy savings of 28.13% in homogeneous and 38.83% in heterogeneous environments, demonstrating its suitability across varying infrastructure types. However, it displayed inconsistencies in certain workload scenarios, such as the notable spike in energy consumption observed in Scenario 1 of the homogeneous data centre setup.
- The PSO-based policy provided moderate energy reductions (27.53% in homogeneous and 36.04% in heterogeneous environments), showing competitive performance in specific scenarios but was generally outperformed by ACO in both data centre setups.

Overall, this project demonstrates that bio-inspired optimisation algorithms are highly effective in reducing energy consumption while maintaining efficient VM placement across diverse data centre environments. Among the three, the ACO-based policy proved to be the most effective, making it a promising approach for large-scale, heterogeneous, and energy-conscious cloud infrastructures.

7.2 Recommendations

While the implementation of ACO, PSO, and MGA-based VM allocation and migration algorithms has demonstrated significant improvements in energy efficiency and resource utilisation, there are still several directions for future enhancement.

First, algorithmic optimisation can be explored to further improve power savings and adaptability under more complex and dynamic environments. This includes advanced parameter tuning strategies, adaptive control of exploration–exploitation balance, and the integration of machine learning techniques to enhance decision-making during VM placement and migration.

Second, the simulation environment can be made more realistic by incorporating complex data centre topologies, multi-tier network structures, and realistic server placement models. Considering these factors would provide a more accurate assessment of migration costs and overall power consumption, improving the real-world applicability of the proposed solutions.

Third, future work can investigate hybrid bio-inspired approaches by combining the strengths of different algorithms. For example, integrating ACO's adaptability with PSO's faster convergence or MGA's strong exploration capabilities could yield more robust and scalable solutions. Additionally, exploring multi-objective optimisation techniques would allow balancing between energy savings, migration overheads, and SLA compliance.

Finally, predictive workload management represents a promising direction. Leveraging Large Language Models (LLMs) or other AI-based predictors could enable proactive VM allocation by forecasting workload spikes and minimising unnecessary migrations. Furthermore, request pre-filtering mechanisms could reduce data centre workloads and optimise energy usage.

By addressing these directions, future work can significantly enhance the scalability, adaptability, and efficiency of VM allocation and migration strategies in modern cloud data centres.

REFERENCES

[1] ASEAN Centre for Energy, “Building Next Generation Data Center Facility in ASEAN” aseanenergy.org. https://aseanenergy.org/wp-content/uploads/2024/05/Building-Next-Generation-Data-Center-Facility-in-ASEAN.pdf

[2] International Energy Agency, “Electricity 2024,” IEA.org. https://www.iea.org/reports/electricity-2024

[3] X. -F. Liu, Z. -H. Zhan, J. D. Deng, Y. Li, T. Gu and J. Zhang, “An Energy Efficient Ant Colony System for Virtual Machine Placement in Cloud Computing,” in *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 113-128, Feb. 2018, doi: 10.1109/TEVC.2016.2623803.

[4] Huawei Technologies Co., Ltd., “Virtualization Technology,” *Cloud Computing Technology*, pp. 97–144, Oct. 2022, doi: https://doi.org/10.1007/978-981-19-3026-3_3.

[5] F. Liu, Z. Ma, B. Wang and W. Lin, “A Virtual Machine Consolidation Algorithm Based on Ant Colony System and Extreme Learning Machine for Cloud Data Center,” in *IEEE Access*, vol. 8, pp. 53-67, 2020, doi: 10.1109/ACCESS.2019.2961786.

[6] H. O. Salami, A. Bala, S. M. Sait, and I. Ismail, “An energy-efficient cuckoo search algorithm for virtual machine placement in cloud computing data centers,” *Journal of Supercomputing*, vol. 77, no. 11, Article ID 13330, 2021, doi: https://doi.org/10.1007/s11227-021-03807-3.

[7] N. Kumar Rajpoot, P. Singh and B. Pant, "Nature-Inspired Load Balancing Algorithms for Resource Allocation in Cloud Computing," *2023 International*

Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES), Greater Noida, India, 2023, pp. 827-832, doi: 10.1109/CISES58720.2023.10183630.

[8] M. Dayarathna, Y. Wen and R. Fan, "Data Center Energy Consumption Modeling: A Survey," in *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 732-794, Firstquarter 2016, doi: 10.1109/COMST.2015.2481183.

[9] International Energy Agency, "Data Centres & Networks," IEA.org. https://www.iea.org/energy-system/buildings/data-centres-and-data-transmission-networks

[10] Google, "Google Environmental Report 2024," sustainability.google. https://sustainability.google/reports/google-2024-environmental-report/

[11] S. Mittal, "A survey of techniques for improving energy efficiency in embedded computing systems," *Int. J. Comput. Aided Eng. Technol.*, vol. 6, no. 4, pp. 440-459, 2014. [Online]. Available: https://arxiv.org/pdf/1401.0765

[12] A. K. Kar, "Bio inspired computing—A review of algorithms and scope of applications," *Expert Syst. Appl.*, vol. 59, pp. 20-32, Oct. 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095741741630183X

[13] I. Fister, Jr., X.-S. Yang, I. Fister, J. Brest, and D. Fister, "A brief review of nature-inspired algorithms for optimization," *Elektrotehniški Vestnik*, vol. 80, no. 3, pp. 116–122, 2013. [Online]. Available: https://arxiv.org/pdf/1307.4186

[14] X. Fan, W. Sayers, S. Zhang, Z. Han, L. Ren, and H. Chizari, "Review and classification of bio-inspired algorithms and their applications," *Journal of Bionic Engineering*, vol. 17, no. 3, pp. 611–631, 2020. [Online]. Available: https://eprints.glos.ac.uk/8468/1/8468%20Fan%2C%20X.%2C%20Sayers%2

C%20W.%2C%20Zhang%2C%20S.%20et%20al.%20%282020%29%20Review-and-Classification-of%20Bio-inspired-Algorithms-and-Their-Applications.pdf

[15] G. Wu, M. Tang, Y.-C. Tian, and W. Li, "Energy-efficient virtual machine placement in data centers by genetic algorithm," in *Proc. Int. Conf. Neural Inf. Process*. Berlin, Germany: Springer, 2012, pp. 315–323. [Online]. Available: https://eprints.qut.edu.au/53767/18/246.pdf

[16] C. Sonklin, Minimising the energy consumption of data centres by genetic algorithms, 2020. [Online]. Available: https://eprints.qut.edu.au/198050/1/Chanipa_Sonklin_Thesis.pdf

[17] M. Tang and S. Pan, "A hybrid genetic algorithm for the energy-efficient virtual machine placement problem in data centers," *Neural Process. Lett.*, vol. 41, no. 2, pp. 1–11, Apr. 2014. [Online]. Available: https://eprints.qut.edu.au/67718/1/NPL-246-New.pdf

[18] H. A. Kurdi, S. M. Alismail and M. M. Hassan, "LACE: A Locust-Inspired Scheduling Algorithm to Reduce Energy Consumption in Cloud Datacenters," in *IEEE Access*, vol. 6, pp. 35435-35448, 2018, doi: 10.1109/ACCESS.2018.2839028.

[19] M. A. Ala'anzy and M. Othman, "Mapping and Consolidation of VMs Using Locust-Inspired Algorithms for Green Cloud Computing," *Neural Process. Lett.*, vol. 54, no. 1, pp. 405–421, Feb. 2022, doi: https://doi.org/10.1007/s11063-021-10637-0.

[20] S. Walton, O. Hassan, K. Morgan, and M. Brown, ''Modified cuckoo search: A new gradient free optimisation algorithm,'' *Chaos, Solitons Fractals*, vol. 44,

no. 9, pp. 710–718, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S096007791100107X

[21] X. -S. Yang and Suash Deb, "Cuckoo Search via Lévy flights," *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, Coimbatore, India, 2009, pp. 210-214, doi: 10.1109/NABIC.2009.5393690.

[22] A. K. Singh, S. R. Swain, D. Saxena and C. -N. Lee, "A Bio-Inspired Virtual Machine Placement Toward Sustainable Cloud Resource Management," in *IEEE Systems Journal*, vol. 17, no. 3, pp. 3894-3905, Sept. 2023, doi: 10.1109/JSYST.2023.3248118.

[23] A. Goyal and N.S. Chahal, "A Proposed Approach for Efficient Energy Utilization in Cloud Data Center", *International Journal of Computer Applications*, vol.115, pp. 11, April 2015. [Online]. Available: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=79d86c5576948690dd616ed0835ac5d94855385a

[24] E. Gupta and V. Deshpande, "A Technique Based on Ant Colony Optimization for Load Balancing in Cloud Data Center," *2014 International Conference on Information Technology*, Bhubaneswar, India, 2014, pp. 12-17, doi: 10.1109/ICIT.2014.54.

[25] N. K. C. Das, M. S. George and P. Jaya, "Incorporating weighted round robin in honeybee algorithm for enhanced load balancing in cloud environment," *2017 International Conference on Communication and Signal Processing (ICCSP)*, Chennai, India, 2017, pp. 0384-0389, doi: 10.1109/ICCSP.2017.8286383.

[26] M. Lawanya Shri, E. Ganga Devi, B. Balusamy, S. Kadry, S. Misra, and M. Odusami, "A fuzzy based hybrid firefly optimization technique for load balancing in cloud datacenters," in *Proceedings of the International Conference*

on Innovations in Bio-Inspired Computing and Applications, pp. 463–473, New York, NY, USA, 2018, doi: https://doi.org/10.1007/978-3-030-16681-6_46.

[27] M. Gamal, R. Rizk, H. Mahdi and B. E. Elnaghi, "Osmotic Bio-Inspired Load Balancing Algorithm in Cloud Computing," in *IEEE Access*, vol. 7, pp. 42735-42744, 2019, doi: 10.1109/ACCESS.2019.2907615.

[28] M. T. Chaudhry, T. C. Ling, A. Manzoor, S. A. Hussain, and J. Kim, "Thermal-Aware Scheduling in Green Data Centers," *ACM Computing Surveys (CSUR)*, vol. 47, no. 3, p. 39, 2015. [Online]. Available: https://www.researchgate.net/profile/Muhammad-Tayyab-Chaudhry/publication/272822782_Thermal-Aware_Scheduling_in_Green_Data_Centers/links/552cf0870cf21acb09210ac6/Thermal-Aware-Scheduling-in-Green-Data-Centers.pdf

[29] R. Chen, B. Liu, W. Lin, J. Lin, H. Cheng, and K. Li, "Power and thermal-aware virtual machine scheduling optimization in cloud data center," *Future Generation Computer Systems*, vol. 145, pp. 578–589, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X23001346

[30] L. Yang, Y. Deng, L. T. Yang and R. Lin, "Reducing the Cooling Power of Data Centers by Intelligently Assigning Tasks," in *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1667-1678, June 2018, doi: 10.1109/JIOT.2017.2783329.

[31] "CloudSim Plus," *CloudSim Plus*. https://cloudsimplus.org/

[32] A. K. Pandey and S. Singh, "An Energy Efficient Particle Swarm Optimization based VM Allocation for Cloud Data Centre: EEVMPSO", *EAI Endorsed Scal Inf Syst*, vol. 10, no. 5, Aug. 2023.

[33] M. Radi, A. A. Alwan and Y. Gulzar, "Genetic-Based Virtual Machines Consolidation Strategy With Efficient Energy Consumption in Cloud Environment," in *IEEE Access*, vol. 11, pp. 48022-48032, 2023, doi: 10.1109/ACCESS.2023.3276292.

APPENDIX

A.1 POSTER