Journal of Trends in Computer Science and Smart Technology (ISSN: 2582-4104)
www.irojournals.com/tcsst/

Energy–Cost–SLA Aware Cloud Scheduling via Adaptive Non-dominated Sorting Genetic Algorithm-III and Neighborhood Refinement

Vijayasekaran G.[1], Sathya V.[2], Sumathi S.[3], Lavanya M.[4]

[1]Department of Computer Science and Engineering, New Horizon College of Engineering, Bengaluru, India.
[2]Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Avadi, India.
[3]Department of Electronics and Communication Engineering, Velammal Engineering college, Chennai, India.
[4]Department of Artificial Intelligence & Data Science, Adhiparasakthi College of Engineering, Kalavai, India.

Email: [1]gunavijay90@gmail com, [2]saro.sath@gmail.com, [3]sumathiecevec@gmail.com, [4]mlavanya.official@gmail.com

Abstract

The optimization of resource allocation in a cloud computing environment is a problem that has been challenging due to heterogeneous tasks with varying resource requirements and different optimization objectives for execution time, energy consumption, service level agreements (SLAs) and so on. In this paper, a hybrid multi-objective optimization algorithm called AH-NSGAIII-VND is proposed for solving a multi-objective optimization problem in a cloud computing environment. The proposed algorithm integrates a global search process using a variant of a multi-objective evolutionary algorithm called Non-Dominated Sorting Genetic Algorithm III (NSGA-III) and a local search process using a variant of a local search algorithm called Variable Neighborhood Descent (VND). The problem of resource allocation in a cloud computing environment is formulated as a multi-objective optimization problem considering makespan, energy consumption, cost, service level agreements (SLAs) and resource utilization. Researchers compared the results of the PSO, GA, and GWO optimization methods to the baseline using CloudSim-based model conditions on the changing workload scale. Experimental results indicate that cloud supply governance efficiency is enhanced with the AH-NSGAIII-VND architecture. It achieves around 11.3%, 12.8%, and 28% lower costs than the baseline NSGA-III approach while increasing overall asset utilization by over 5 percentage points. Moreover, with increasing workload, the proposed model exhibits improved convergence behavior and scalability. These results confirm that global evolutionary optimization supported by adaptive local search successfully reinstates or enhances the efficiency of resource allocation.

Keywords: Cloud Computing; Resource Allocation; Multi-Objective Optimization; Hybrid Optimization; NSGA-III; SLA Management.

DOI: https://doi.org/10.36548/jtcsst.2026.1.004
Received: 13.01.2026, received in revised form: 11.02.2026, accepted: 25.02.2026, published: 07.03.2026

1. Introduction

The current digital platform has been highly dependent on cloud computing, which allows on-demand accessibility to shared computing services, including processing power, storage, and network bandwidth. Its price model of pay-as-you-go and elasticity of price has enabled the implementation of large applications in the realms of e-commerce, healthcare, scientific computing, and smart cities. Nevertheless, even with these benefits, efficient resource management and allocation remain problematic areas in cloud environments. The challenge for cloud service providers is the continuous assignment of heterogeneous resources to dynamically emerging workloads while meeting strict quality-of-service (QoS) requirements and reducing the workload. One cause of resource under-utilization is ineffective allocation; other causes include energy wastage, violations of service-level agreements (SLAs), poor user experience, and potential fines.

There are several characteristics associated with cloud environments that render cloud resource administration a complicated process. To start with, cloud infrastructures are heterogeneous, comprising virtual machines (VMs) that can operate under variable capacities, dynamic memory structures, energy profiles, etc. Second, workloads are unpredictable and dynamic, with the amount of work and frequency of arrival varying. Third, cloud providers are expected to balance a set of contradictory goals, including makespan reduction, energy usage reduction, resource utilization, and SLA compliance maximization. It is this latter expectation that makes traditional deterministic and fixed allocation methods ineffective in large-scale and highly dynamic situations.

The initial plans for cloud resource allocation were based on rule-based approaches and heuristic methods (First Come First Serve (FCFS), Min-Min, Max-Min, and Round-Robin scheduling). However, these approaches are theoretically single-objective and computationally inexpensive, relying on a priori trade-offs among various performance measures. With the emergence of cloud systems, researchers began adopting metaheuristic optimization schemes, including Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Grey Wolf Optimization (GWO). Such algorithms have been found to offer better global search capabilities and versatility than classical heuristics. However, the majority of these methods utilize a weighted sum of goals, which should be parameterized with care, as it can lead to the search being overly focused on certain metrics at the expense of others.

The last several years can be described by the popularity of multi-objective evolutionary algorithms (MOEAs) as a way to overcome the shortcomings of single-objective optimization. Algorithms such as NSGA-II and NSGA-III [12] [13] allow for conflicting objectives because a set of Pareto-optimal solutions may be produced. The potentially useful candidate among them is NSGA-III, which is helpful in high-dimensional multi-objective problems as it has a diversity preservation mechanism developed based on reference points. Cloud resource management has been implemented using NSGA-III to maximize the makespan, energy usage, cost of execution, and the SLA violation rate. However, despite its benefits, when directly applied to the issue of complex cloud allocation, NSGA-III possesses some drawbacks. In particular, it might be slow over large search areas, and the algorithm might be ineffective in the search for good solutions because it is focused on maintaining diversity.

Recent studies have proposed that a hybrid optimization model, consisting of both an evolutionary search mechanism on a global scale and a local refinement mechanism, can significantly improve both the quality and convergence of a solution. Hybrid algorithms can reduce the solutions discovered through global search with the aid of local search strategies, which enhance exploitation without compromising exploration. Nevertheless, the hybrid solutions currently present in cloud resource management lack flexibility, are excessively costly in terms of computation, and do not offer multi-objective trade-offs. Furthermore, although much of the research literature is preoccupied with a narrow scope of performance measures, it ignores critical issues of actual implementation, such as SLA violations and the balance of resource consumption, which are essential in practice.

This observation characterizes a research gap in the literature as experienced today. Although multi-objective optimization and hybrid metaheuristics have been considered separately in the area of cloud resource allocation, there are no adaptive and hybrid schemes that can be beneficially applied to arrange a novel multi-objective evolutionary algorithm alongside local search. Specifically, current methods can hardly provide adaptive control of variation operators that dynamically vary their population diversity or convergence behavior. Additionally, comparative analyses tend to be small-scale, unevenly assess performance, or lack depth according to tables and figures, which undermines the validity of the results.

Among these issues, the current research presents a proposal for a multi-objective optimization framework that can be implemented as a hybrid approach to aid in the intentional administration and distribution of cloud resources. The suggested solution builds on the strengths of NSGA-III and the Variable Neighborhood Descent (VND) based local refinement strategy. NSGA-III is aggressive in the search for a global solution and provides a more diverse Pareto front than VND, which is more effective in local exploration through the systematic search of various neighborhood structures around quality solutions. In addition, there is an adaptive operator control process aimed at regulating the crossover and mutation probabilities based on the indicators of diversity and convergence within the population. This adaptive behavior balances premature convergence and optimally balances exploration and exploitation during the optimization process.

The proposed model frames the allocation of cloud resources as a comprehensive multi-objective optimization problem, which minimizes makespan, energy usage, cost of execution, and SLA violation rate, while maximizing overall resource utilization. The framework specifically addresses the performance requirements of both providers and users. Large-scale simulation-based experiments are conducted to test the scalability and efficiency of the proposed solution, utilizing a CloudSim-like system with heterogeneous virtual machines and workloads of varying sizes. The proposed adaptive hybrid optimizer is rigorously evaluated against known baseline algorithms, such as PSO, GA, GWO, NSGA-II, and the traditional NSGA-III, based on quantitative results and statistical evaluation measures.

There are three significant contributions to this research:

1. The cloud resource allocation problem is articulated as a coherent multi-objective issue encompassing substantial performance, cost, and QoS requirements.

2. A hybrid optimization model that combines NSGA-III and VND is developed to maximize both the convergence rate and the quality of the solution.

3. The model is examined experimentally in detail over an extensive range of workloads to demonstrate scalability and resilience.

The remainder of this manuscript is structured as follows: Section 2 provides a literature review of recent research on cloud resource allocation and optimization techniques, with particular attention to multi-objective and hybrid approaches. Section 3 presents the mathematical model of the cloud resource management issue, including objectives and constraints. Section 4 outlines the proposed adaptive hybrid NSGA-III-VND framework in detail. Section 5 defines the experimental conditions and measures of simulation. Section 6 discusses the findings of the experiment in quantitative terms using tables and graphs. Lastly, Section 7 concludes the paper by summarizing some of the most important findings and suggesting potential paths for future research.

2. Related Works

Resource management and allocation have been well-known issues in cloud computing and its extended paradigms, such as edge, fog, and network slices. The increase in cloud-hosted applications, the heterogeneous nature of infrastructure, and dynamic workload behavior have exacerbated the need for intelligent scheduling and distribution processes that can balance performance, energy costs, and quality-of-service (QoS) constraints. Initial research has mostly focused on heuristic and prediction-based scheduling techniques, whereas more recent studies have increasingly examined learning-based and multi-objective optimization techniques to manage the complexity and conflicting nature of cloud resource management.

In [1], the authors provide a thorough overview of prediction-based resource scheduling methods and their application to physics-based scientific problems. The study noted that workload prediction can enhance scheduling decisions by estimating future resource requirements, thereby minimizing delays in the execution process and improving system throughput. Nevertheless, the results also revealed that the accuracy of predictions significantly affects scheduling effectiveness, and inaccurate estimates of workload can lead to inefficient resource usage and an increase in the number of SLA violations. This weakness highlights the inherent limitations of strictly predictive strategies in rapidly changing cloud environments. The results of [2] expanded resource management in classical cloud systems to provide a QoS-aware and channel-aware radio resource management model for multi-numerology systems. Although the article focused on communication networks, the principles of QoS prioritization and adaptive resource allocation are quite applicable to cloud computing. The analysis showed that system performance can be improved by integrating awareness of heterogeneous resource characteristics. However, the suggested framework concentrated on resource allocation at the communication level but neglected allocation at higher levels and multi-objective scheduling and optimization of tasks concerning compute, energy, and cost.

The hypothesis proposed by [3] suggested that the resource allocation process could be optimized with the help of reinforcement learning in the context of cloud environments, increasing system feedback for continuous learning. The presented prediction-based reinforcement learning model was also shown to be more adaptable when workloads varied and more effective than fixed heuristics. Despite these benefits, the strategy was oriented toward individual or small-scale goals, and the convergence behavior in large-scale, multi-objective conditions was not examined in detail. Moreover, reinforcement learning techniques may require a long time to train, and reward design may also constrain their scalability. A reliability-oriented perspective emerged in [4], where a new task scheduling algorithm

emphasized fault tolerance and system robustness. The analysis demonstrated that reliability-conscious scheduling could help reduce task malfunctions and enhance overall system reliability. Nonetheless, a focus on reliability could lead to longer execution times and increased energy usage, revealing a trade-off that was not actively optimized. This finding indicates that multi-objective formulations are needed to balance reliability with other vital performance metrics.

In [5], a multi-agent reinforcement learning system for dynamically managed clouds was investigated. The authors showed that distributed agents could collaboratively learn effective resource allocation policies that enhanced scalability and adaptability. Although encouraging, the findings indicated that coordination overhead and training instability remain serious issues, especially in heterogeneous cloud infrastructures. Additionally, the framework did not specifically focus on energy consumption or minimizing SLA breaches but streamlined resource usage and reaction time. In [6], deep reinforcement learning was further employed to solve the resource allocation problem in IoT-edge-cloud computing systems. This research demonstrated that agents developed using deep learning could learn the dynamics of complex systems and enhance the effectiveness of content distribution. However, the computational cost of deep reinforcement learning models and sensitivity to hyperparameter optimization pose challenges for real-time applications. Furthermore, the emphasis on content distribution limits the overall relevance of the strategy to more general cloud task scheduling contexts. .

The research on resource allocation through the use of machine learning in an auction-based cloud environment was studied in [7]. The presented method proved that learning-based bidding strategies might enhance resource utilization and economic performance. The advantages notwithstanding, there is always the fact that auction-based models are usually founded on rationality and market stability, which is not always the case in a real-world cloud system where the workload is unpredictable and the QoS needs of different applications can vary. Moreover, performance-based measures like makespan and SLA compliance might be neglected because of the focus on economic goals. Cloud resource management has also received a fair share of metaheuristic optimization techniques. A load balancing approach based on predictive machine learning and similar to that of the ACO approach was proposed in [8]. The paper established that the integration of swarm intelligence with prediction systems can increase load distribution and decrease response time. Nevertheless, it was a load balancing strategy that concentrated on the optimization of multiple and conflicting goals simultaneously. This weakness implies that hybrid heuristics might not be able to deliver solutions that are global in terms of metrics. In [9], an adaptive genetic algorithm-based task scheduling method was suggested with a focus on load balancing awareness. These findings showed that execution time and resource usage were better than in traditional genetic algorithms. The research used single-objective fitness functions, which restrict the capacity of the study to deal with trade-offs regarding energy, cost, and QoS limitations. This fact supports the importance of multi-objective optimization frameworks based on Pareto in the cloud domain.

In [10], a cooperative search algorithm was presented for task scheduling in heterogeneous cloud systems. It was revealed that cooperative search strategies may be helpful in searching for the solution space to enhance scheduling performance. However, the strategy failed to include explicit safeguarding mechanisms for diversity and can result in premature convergence in large-scale settings. Additionally, energy use and SLA violations were not considered major optimization goals. In [11], joint resource allocation and task scheduling optimization were considered in the context of network slicing. The paper emphasized the significance of coordinated decisions made at the interfaces of various resource layers to

guarantee efficient slicing performance. Although the suggested framework was beneficial for network usage, it was specifically adapted to vehicular communication systems and did not directly generalize to cloud computing systems with diverse application loads and computing power requirements. In [12], a hierarchical method of resource management in open radio access networks was introduced. The results showed that hierarchical models could be effectively used to manage complexity and enhance scalability. Despite providing useful insights into the management of learning-based resources in a structured way, the approach is limited by its emphasis on communication infrastructure, hindering its applicability to the scheduling of cloud tasks and the multi-objective optimization of compute-centric metrics.

The paper [13] investigated the problem of multi-objective optimization of emergent demand management in cloud computing. The proposed formulation directly considered conflicting goals and outperformed the single-objective ones. Nevertheless, the optimization was based on traditional evolutionary processes that were not accompanied by sophisticated local refinement techniques, which could restrict the rate of convergence and quality of solutions in high-dimensional objective space. The problem of integrated scheduling of hybrid tasks over cloud manufacturing environments was studied in [14]. The paper covered multifaceted task dependencies and multiple stage execution needs, which can be a great contribution to industrial cloud applications. However, the framework was domain-centric and made little or no consideration of energy efficiency or SLA compliance as its main goals, making it less applicable to general-purpose cloud resource management.

In [15], multi-objective genetic algorithms used to solve workload allocation in IoT-fog-cloud architectures were suggested. The findings proved that the trade-off in latency, energy, and resource use could be optimized effectively through evolution. Nonetheless, the adaptive mechanisms were not expanded in the study, which may restrict performance due to dynamic workload conditions. In edge-cloud settings, the placement and chaining of services were explored in [16], where the placement of services was optimized in relation to service quality. Even though the strategy enhanced deployment efficiency, it prioritized the placement of services instead of task placement, and it did not explicitly address the multi-objective trade-offs in terms of cost and energy consumption.

In [17], the idea of combining deep reinforcement learning with federated learning in these cases was investigated as part of managing multi-timescale resources. The experiment showed better scalability and privacy in ultra-dense networks. Although these benefits exist, the complexity of federated training and the overheads of communication pose some challenges to real-time cloud scheduling cases. Vehicular fog computing resource allocation using auctions was studied in [18]. The double-auction process enhanced allocation efficiency and fairness and was dependent on economic assumptions that do not always coincide with the QoS-oriented cloud scheduling goals. In addition, the strategy was silent on energy efficiency or minimizing execution time. The issue of trust and reliability in smart city service placement was explored in [19], with an emphasis on trust and reliability as important parameters. Although trust-aware allocation enhances system robustness, it also provides more constraints, which can be incompatible with performance and cost goals, underlining the importance of multi-objective balancing strategies.

The concept of making dynamic offloading decisions with the help of double deep Q-networks was presented in [20]. The results showed that latency was better in a mobile edge computing setup. However, the heavy reliance on deep learning models raises computational overhead, and the architecture was geared towards offloading decisions rather than the overall management of cloud resources. In [21], power consumption minimization in the cloud

environment through task consolidation was evaluated. The findings proved that the minimization of energy consumption through the consolidation strategies is possible by minimizing the number of active physical hosts. However, aggressive integration can enhance SLA breaches and delays in execution unless it is closely adjusted to performance targets. Lastly, dataset generation in wireless communication systems was investigated in [22], providing tools to create realistic channel condition simulations. Although it is not specifically about cloud computing, the paper emphasizes the use of realistic models to determine performance, thus supporting the usefulness of simulation-based validation in resource management studies.

Based on the above review, it can be seen that the current literature has already achieved a significant breakthrough in managing cloud resources through predictive scheduling, reinforcement learning, heuristic optimization, and multi-objective models. Nevertheless, there are a number of limitations that have not been addressed. Most strategies are aimed at just a small range of goals without considering the combination of achievements, energy, cost, and QoS constraints. The high training complexity and scaling problem of learning-based methods can be exacerbated by the issue of heuristic convergence without proper diversity preservation. In addition, there are limited studies that effectively combine global evolutionary search with local refinement structures to improve the convergence rate and quality of solutions in high-dimensional objective space.

3. Proposed Work

In this section, we propose the Adaptive Hybrid NSGA-III with Variable Neighborhood Descent (AH-NSGAIII-VND) mechanism for effective resource allocation and management in cloud computing systems.

3.1 Architectural Description

The recommended framework will address the complexity of the allocation problems of cloud resources since it is based on the collaborative optimization of several conflicting objectives under heterogeneous and dynamic operating conditions. The journey begins with a glimpse into the architecture of the system which provides an overview of problem modelling, objectives formulation, evolutionary optimization design, adaptive control mechanism based on a local refinement strategy, and cost analysis. Figure 1 illustrates the general architecture.

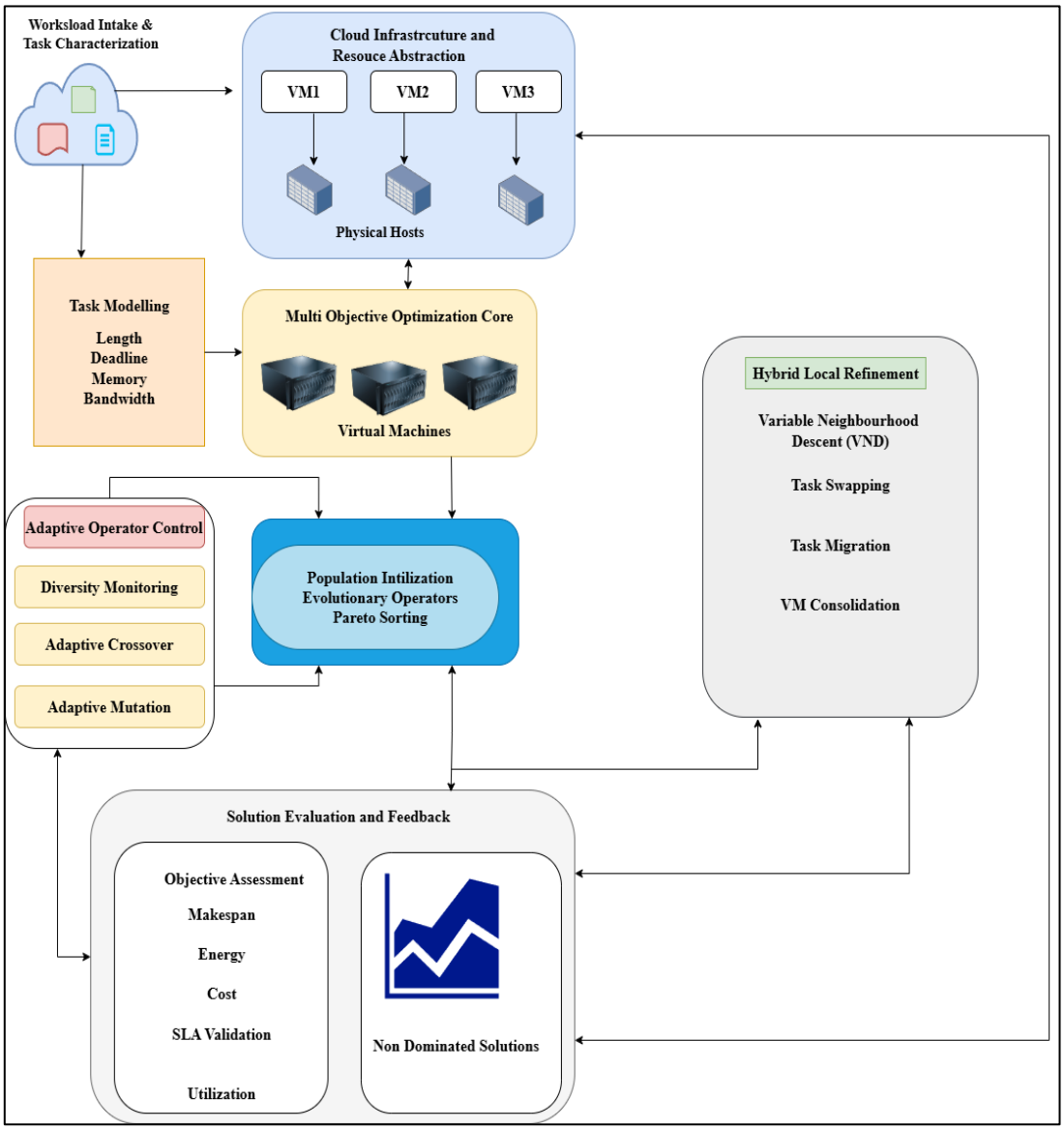

Figure 1. Depicts the AH-NSGAIII-VND Resource Allocation Model

The proposed design of the AH-NSGAIII-VND is visualized as an optimization pipeline. It consists of a series of workload characterizations, cloud workload abstractions, and a multi-objective evolution sequence followed by local refinement. The architecture consists of five functional elements that enable a cycle of decision-making.

- Layer for Task Modelling and Workload Characterization
- Layer for resource abstraction and cloud infrastructure.
- Multi-Objective Optimization Prime (NGS-III Engine).
- Module of Hybrid Local Refinement (VND).

3.1.1 Layer 1: Workload Intake and Task Characterization Layer

This layer involves recording and pre-processing the cloud workloads that come in. The nature of the work that is sent for processing within the cloud system is represented as unique computational entities, which may possess different characteristics such as deadlines,

requirements for bandwidth, and requirements for memory. The optimization engine can utilize the structured form of the task, which is obtained by this layer from unprocessed workload data. The heterogeneous nature of the work is preserved during the optimization process since it is modeled by the system at this stage. This layer is significant within the context of a realistic cloud system since there may be variations in the nature of the workloads, such as priority and size. The abstraction layer for resources makes use of the normalized form of the tasks, which is obtained by this layer.

3.1.2 Layer 2: Cloud Infrastructure and Resource Abstraction Layer

The second layer is an abstraction of the underlying cloud infrastructure into a manageable optimization model. Physical machines are modelled into a swimming pool of heterogeneous virtual machines (VMs), specified in terms of processing capacity, memory availability, bandwidth, and energy consumption properties. This abstraction separates the optimization logic from the low-level details of the hardware to enhance generality and scalability. System-level constraints, e.g., VM capacity limitations and SLA requirements are also integrated into this layer and subsequently applied during solution evaluation and repair. The architecture facilitates the flexibility of the proposed framework by decoupling infrastructure modeling from optimization logic, allowing it to be moved to other cloud environments without altering the underlying algorithm.

The NSGA-III optimization engine and the adaptive direct faculty work in a closed-loop connection during each iteration (coevals) of the proposed AH-NSGA-III-VND skeleton by sharing relatively small but essential information. In particular, the study of the existing population allows the NSGA-III engine to push iteration-level responses containing (i) the diversity of the society (and the circulation of remedies exceeding the mentioned direction), (ii) the shift in convergence/improvement of the objectives across the back-to-back coevals, (iii) the objective statistics of the new inhabitants (makespan, energy, SLA infraction estimate, and the cost of execution, provision consumption) and (iv) the pinpointed non-dominated/elite subset of corrections. The adaptive regulator faculty computes new evolutionary operator principles and provides the NSGA-III engine with a direct signal in return as dynamically tuned crossover and mutation p and m. To improve the research system, low diversity or development rates must be accompanied by an increase in the probability of mutations. Conversely, close to the ideal stage of diversity and convergence, the transition must be adjusted to minimize the use of mutations in order to maximize the use of resources. The NSGA-III search process can continuously adjust to the changing optimization environment to this iterative exchange, which guarantees eternal collections and prevents premature convergence.

3.1.3 Layer 3: Multi-Objective Optimization Core (NSGA-III Engine)

The third layer is the core of the proposed framework and realizes the global search mechanism based on NSGA-III. A candidate solution is coded at this stage in the form of task-to-VM allocation vectors and developed through the generations by Pareto-based selection. As opposed to single-objective schedulers, this layer does not choose the solution based on only one objective instead, the solution is evaluated against a variety of conflicting objectives, such as makespan, energy consumption, execution cost, SLA violation rate, and resource utilization. The NSGA-III uses a selection based on reference points to ensure the diversity of the Pareto front, which is essential when dealing with high-dimensional multi-objective problems. This process enables the optimizer to search for a long distance of trade-off solutions instead of

terminating early to a small part of the search space. The result of this layer is a ranked list of the best non-dominated solutions denoting various allocation strategies.

3.1.4 Layer 4: Hybrid Local Refinement Layer (Variable Neighborhood Descent)

The architecture uses a Variable Neighborhood Descent (VND) module as the fourth layer to overcome the limited exploitation capability of pure evolutionary search. It is a selective layer that is applied to the elite non-dominated solutions of the NSGA-III engine. Refinement is concentrated on high-quality candidate solutions of candidates, rather than being applied across the whole population of the search, to enhance the efficiency of convergence. The VND module is used to investigate different neighborhood constructions in a systematic way, such as task swapping, task migration, and VM-consolidation. The neighborhoods aim to address particular issues in the allocation problem, including load imbalance or underutilized resources. The architecture provides the ability to optimize promising solutions finely, without breaking up population diversity, through the serial application of these neighborhoods. The evolved solutions are then reintroduced into the population of evolution.

The use of task swap, task migration, and VM consolidation as VND neighborhoods is chosen due to their being three complementary and low-overhead local adjustment mechanisms that have direct graphical impacts on cloud scheduling goals. Task swap is a perturbation of a few types, where the assignments of tasks are swapped between two resources, and it is used in corrections of local load imbalance without compromising the ability to work. Migration of tasks makes corrective moves more powerful and can decrease the risk of SLA violation because tasks with high impact are moved to better-qualified VMs, and makespan/energy trades off. Consolidation occurs at the infrastructure level through the consolidation of workloads into fewer VMs and freeing idle VMs, which directly reduces energy usage and operational costs without affecting quality of service. These neighborhoods collectively offer a combination of fine-grained and coarse-grained search moves, which are useful in providing local improvement in a short time, as well as enhancing Pareto refinement with multi-objective constraints.

3.1.5 Layer 5: Adaptive Control and Solution Evaluation Layer

The final layer provides adaptability and wisdom in terms of resolution. This layer shall always keep an eye on society's diversity, convergence assessment, and ambition dispersion and shall make appropriate changes in terms of crossover and mutation probability in the parameters. The current method of adaptive restriction is of vital importance in terms of solving and utilizing problems in cases of large complications and heterogeneous situations. Additionally, a thorough review of the answers, limitations, and recoveries of the functioning shall be carried out. Answers surpass the boundaries of materiality, and the current SLAs are reformulating the support of the repair method in a manner that is helpful for the transfer of solutions to possible VMs with minimal deterioration in terms of performance. Finally, the patches that have been tested and fixed shall again enter the optimization core in a manner that ensures every iteration of the critique cringle of the architecture is complete.

The communication between these layers forms a closed-loop optimization cycle. The downward and upward movements of task and infrastructure information between the workload layer and optimization core, as well as between the performance feedback and adaptive signals layer and the evaluation layer, respectively, create a two-way interaction that allows the framework to react dynamically to changes in the workload and optimization

stagnation. The suggested framework meets three major design objectives. First, it is modular, so the individual layers can be extended or substituted without causing any impact on the system as a whole. Second, it is scalable because both the evolutionary and local search components act on abstracted representations rather than raw system states. Third, it provides a solid multi-objective decision making framework that delivers a variety of Pareto-optimal solutions applicable to various cloud operational policies.

The given architecture is unique in regard to the currently used cloud resource management structures in that it explicitly incorporates adaptive multi-objective optimization with structured local refinement in one pipeline. This architecture is developed to reflect the implicit trade-offs inherent to cloud environments unlike conventional schedulers which use fixed heuristics or single-objective models and to optimize solutions both globally and locally. Consequently, it offers a highly successful, flexible, and scalable foundation upon which management systems of the next generation of cloud resources will be built. The incoming cloud tasks are initially identified with the help of characteristics such as computational demand, memory requirements, bandwidth usage, and deadline constraints. These tasks are subsequently overlaid onto an uneven set of virtual machines (VMs) running on physical servers of different capacities and power properties. The optimization core searches for Pareto-optimal strategies of task-to-VM allocation in the form of local neighborhood exploration and improvement of promising solutions. The adaptive control module is dynamically adjusted to balance exploration and exploitation across generations by adjusting algorithmic parameters.

Effective management of cloud resources demands a cohesive optimization scheme that can address heterogeneous workloads, various conflicting goals, and tight system constraints. In order to deal with these problems, the proposed Adaptive Hybrid NSGA-III with Variable Neighborhood Descent (AH-NSGAIII-VND) framework will model the problem of cloud resource allocation as a multi-objective decision-making problem, solving it through the tight integration of evolutionary and local search architecture. The flow process is depicted clearly in Figure 2.

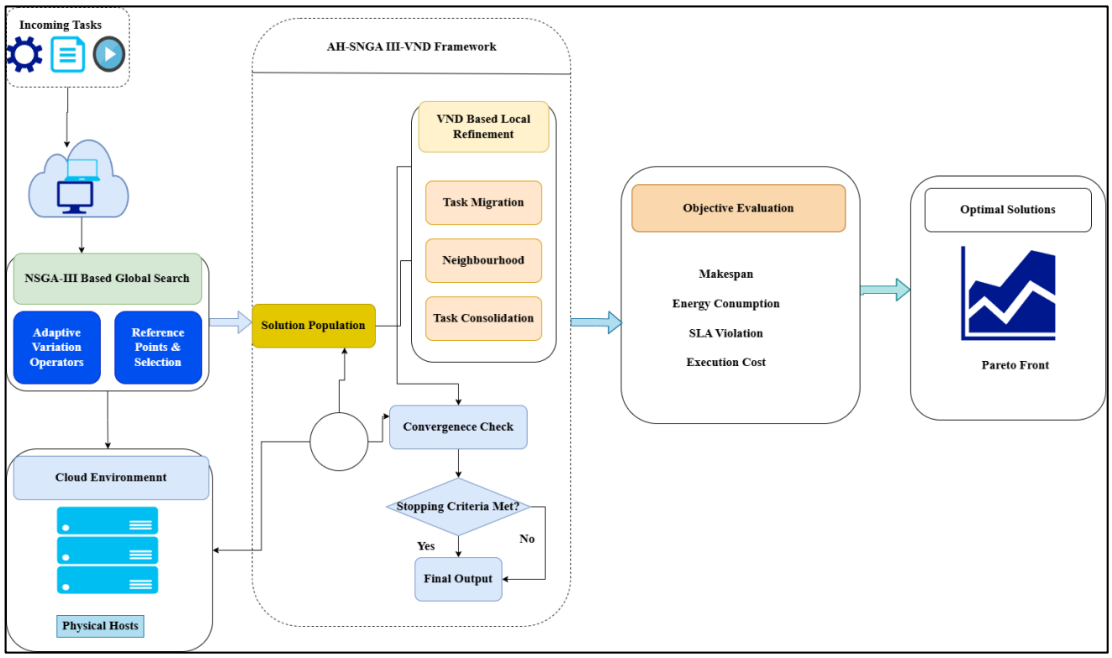

Figure 2. Control Flow of Proposed AH-NSGAIII-VND Model

As shown in Figure 2, in the cloud computing environment, tasks are represented in a three-part system that consists of tasks, virtual machines, and physical hosts. Initially, a model for incoming tasks is represented in a cloud platform that consists of heterogeneous virtual machines and physical hosts. The multi-objective optimization engine is then provided with inputs related to the characteristics of tasks and resources. In the NSGA-III module, a global search is carried out using a selection mechanism based on a reference point. In addition, an adaptive operator control mechanism is used to sustain diversity in the population and prevent premature convergence. Elite solutions obtained in the global search are then optimized using a VND method that consists of local task swapping, migration, and consolidation operations to optimize load balancing and QoS. Finally, the candidate solutions are evaluated based on a set of objectives, which include makespan, energy consumption, SLA violation rate, execution costs, and resource utilization. Optimization is carried out until convergence is achieved to obtain a set of high-quality Pareto optimal solutions that represent a set of optimal trade-offs between competing cloud computing performance objectives.

3.2 Cloud System Model and Resource Representation

The cloud environment is modeled as a three-component system comprising tasks, virtual machines, and physical hosts. Let the cloud system be represented as

$$\mathcal{C} = \{\mathcal{T}, \mathcal{V}, \mathcal{P}\}, \tag{1}$$

where $\mathcal{T} = \{T_1, T_2, \dots, T_N\}$denotes the set of independent cloud tasks, $\mathcal{V} = \{VM_1, VM_2, \dots, VM_M\}$represents the available virtual machines, and $\mathcal{P}$corresponds to the underlying physical infrastructure.

Each task $T_i$is characterized by its computational length L_i, deadline D_i, memory requirement M_i, and bandwidth demand B_i, forming the task descriptor

$$T_i = \{L_i, D_i, M_i, B_i\}. \tag{2}$$

Similarly, each virtual machine $VM_j$is defined by its processing capacity C_j, available memory R_j, bandwidth BW_j, and power coefficient P_j. This modeling explicitly captures heterogeneity across cloud resources and provides a structured basis for optimization.

3.3 Resource Allocation Encoding and Execution Modeling

Resource allocation decisions are encoded using a task-to-VM mapping vector

$$\mathrm{X} = [x_1, x_2, \dots, x_N], \tag{3}$$

where $x_i = j$indicates that task $T_i$is assigned to virtual machine VM_j. Based on this encoding, the execution time of a task on a selected virtual machine is computed as

$$ET_{ij} = \frac{L_i}{C_j}, \tag{4}$$

linking task computational demand to VM processing capacity. The completion time of each virtual machine is then obtained by aggregating the execution times of assigned tasks:

$$CT_j = \sum_{i:x_i=j} E\,T_{ij}. \tag{5}$$

The overall makespan of the cloud system is defined as the maximum completion time across all virtual machines, ensuring that system performance is governed by the most heavily loaded resource:

$$f_1 = \max_j\ CT_j. \tag{6}$$

Algorithm 1: Overall Hybrid Multi-Objective Optimization Framework

Begin
Initialization: Set maximum generations G_{max}, population size P, neighborhood limit K, diversity threshold δ
Define objective set $\mathcal{F} = \{f_1, f_2, f_3, f_4, f_5\}$
Initialize population $P^0 = \{X_1^0, X_2^0, \dots, X_P^0\}$, $x_i \in \{1, \dots, M\}$
Set generation $g = 0$
While $g < G_{max}$
  Evaluate $\mathcal{F}(X)$ for all $X \in P^g$
  Apply non-dominated sorting and reference-point association
  Adapt crossover and mutation probabilities using population diversity
  Generate offspring population Q^g
  Apply hybrid local refinement (Algorithm 3) on elite solutions
  Update population P^{g+1}
  $g \leftarrow g + 1$
End While
Return Pareto-optimal solution set
End

3.4 Multi-Objective Performance and QoS Modeling

In addition to makespan, multiple performance and quality-of-service objectives are considered to reflect realistic cloud operational requirements. Energy consumption is modeled as a function of the execution time and power coefficient of each virtual machine

$$E_j = P_j \cdot CT_j, \tag{7}$$

and the total energy consumption of the system is minimized as

$$f_2 = \sum_{j=1}^{M} E_j\,. \tag{8}$$

SLA compliance is enforced by penalizing tasks that exceed their specified deadlines. The SLA violation rate is computed using an indicator function as

$$f_3 = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(C\,T_i > D_i). \tag{9}$$

From a cost perspective, the execution cost incurred by each virtual machine is proportional to its active execution time:

$$Cost_j = \alpha_j \cdot CT_j, \tag{10}$$

leading to the total execution cost objective

$$f_4 = \sum_{j=1}^{M} Cos\, t_j. \tag{11}$$

To avoid resource underutilization, VM utilization is quantified as

$$U_j = \frac{CT_j}{\max\,(CT)}, \tag{12}$$

and incorporated as a minimization objective by negating the average utilization:

$$f_5 = -\frac{1}{M}\sum_{j=1}^{M} U_j\,. \tag{13}$$

3.5 Multi-Objective Optimization Formulation

The defined objectives collectively form a high-dimensional multi-objective optimization problem expressed as

$$\min\ \mathrm{F(X)} = [f_1, f_2, f_3, f_4, f_5], \tag{14}$$

subject to virtual machine capacity, memory, and bandwidth constraints. This formulation enables the simultaneous optimization of performance, energy efficiency, execution cost, SLA compliance, and resource utilization, ensuring balanced and QoS-aware cloud resource management.

Algorithm 2: Multi-Objective Task–VM Evaluation

Begin
Input: Task set $\mathcal{T} = \{T_i\}_{i=1}^{N}$, VM set $\mathcal{V} = \{VM_j\}_{j=1}^{M}$, solution **X**

For each $T_i \in \mathcal{T}$
  Compute execution time
  $ET_{ij} = L_i/C_j$
  Assign $T_i \rightarrow VM_{x_i}$

For each $VM_j \in \mathcal{V}$
  Compute completion time
  $CT_j = \sum(ET_{ij} \mid x_i = j)$
  Compute energy
  $E_j = P_j \cdot CT_j$
  Compute cost
  $Cost_j = \alpha_j \cdot CT_j$

Compute makespan

$$f_1 = \max\,(CT_j)$$

Compute total energy

$$f_2 = \sum E_j$$

Compute SLA violation rate

$$f_3 = \frac{1}{N}\sum \mathbb{I}(CT_i > D_i)$$

Compute execution cost

$$f_4 = \sum Cost_j$$

Compute resource utilization

$$f_5 = -\frac{1}{M}\sum (CT_j / \max(CT))$$

Return F $= [f_1, f_2, f_3, f_4, f_5]$
End

3.6 Hybrid Optimization Strategy

In order to solve the formulated multi-objective optimization problem, NSGA-III is used as the global search engine because it is capable of preserving diversity on high-dimensional Pareto fronts. However, since it is clear that pure evolutionary search is not effectively capable of exploration, the proposed framework incorporates a Variable Neighborhood Descent (VND) mechanism. The hybrid strategy optimizes elite solutions through task swapping, task migration, and VM consolidation neighborhoods to allow for local optimization and maintain global diversity. The result of this constructive collaboration is an increased convergence rate and quality solutions that are Pareto-optimal.

Algorithm 3: Hybrid Variable Neighborhood Descent Refinement

Begin
Input: Elite solution set $\mathcal{E}$, neighborhood structures $\{N_1, N_2, N_3\}$
For each X $\in \mathcal{E}$
  Set $k = 1$
  While $k \leq 3$
    If $k = 1$: $x_i \leftrightarrow x_j$
    If $k = 2$: $x_i \rightarrow VM_k$
    If $k = 3$: consolidate low-load VMs
    Evaluate F(X_{new})
    If $X_{new} \prec X$
      X $\leftarrow X_{new}$, $k \leftarrow 1$
    Else
      $k \leftarrow k + 1$
Return refined solution set
End

The presented hybrid optimization model, which combines multi-objective evolutionary search with local refinement, introduces a detailed decision-making system for cloud resource management in a heterogeneous and QoS-imposed situation. Despite the information provided on system modeling, the formulation of the objectives, and algorithm design, the actual efficacy of the offered methodology should be demonstrated through strict empirical analysis. In this regard, the next section offers a comprehensive analysis of the proposed framework through simulation, focusing on its behavior across major measures, including makespan, energy consumption, cost of execution, SLA violation rate, and resource utilization. These are compared to well-known benchmark optimization methods to evaluate their scalability, performance under different workload levels, and overall effectiveness.

Quantitative tables and graphical representations are utilized in the analysis of the results to provide clear information about the strengths and weaknesses of the suggested approach.

4. Results and Discussion

In this section, the effectiveness of the proposed AH-NSGAIII-VND framework in controlled and simulation-based conditions under the cloud will be evaluated. The a has been designed to focus on measurable results and reproducible environments instead of qualitative assertions with the core objective being the efficiency of the proposed optimizer in balancing the core objectives of cloud resource management operations, which are reduced makespan, reduced energy usage, minimized SLA violations, lower execution cost, and enhanced resource utilization. Every competitive outcome in the following subsections is provided with a uniform workload scale and the same stopping criterion for each competing approach, and any variation in the observed performance can be attributed to optimization behavior rather than configuration bias. To keep the situation clear and allow for direct verification of the experimental conditions, a summary of the simulation environment, workload generation process, and infrastructure parameters is provided, along with a table-based comparison of results.

4.1 Experimental Scenario

A CloudSim-related discrete-event simulation platform can be viewed as modeling a realistic cloud resource allocation pipeline, which comprises task arrivals, VM-level scheduling, and objective-driven allocation decisions. The model of the experimental scenario assumes independent tasks that are posted to a heterogeneous VM pool deployed on physical machines. The intensity of the workload is varied by setting the number of tasks N to more than two scales (e.g., 100-1000) with constant heterogeneity in VM capacities and resource constraints. Each task is described as having a computational length Li, deadline Di, memory need Mi, and bandwidth. To capture realistic cloud heterogeneity, the VM pool consists of varying CPU speeds (MIPS), RAM, and bandwidth, and the energy model represents each VM in terms of a power coefficient that can be used to estimate the amount of power consumed when executing that VM.

Figure 3 is a simplified depiction of the CloudSim-like modeling that will be used within the proposed simulation environment. The incoming tasks are depicted in the figure as cloudlets, and they are submitted to a Cloud Broker, which serves as the central scheduling point. The broker takes care of task distribution to a pool of heterogeneous virtual machines, which are implemented on physical hosts in a datacenter. Task execution is modeled as a discrete event, and scheduling feedback is sent back to the broker to model CloudSim execution and control flow. This abstraction puts the fundamental CloudSim elements and interactions in a form that is independent of the lower implementation platform, thus allowing easy integration of the optimization framework proposed in Figure 1.

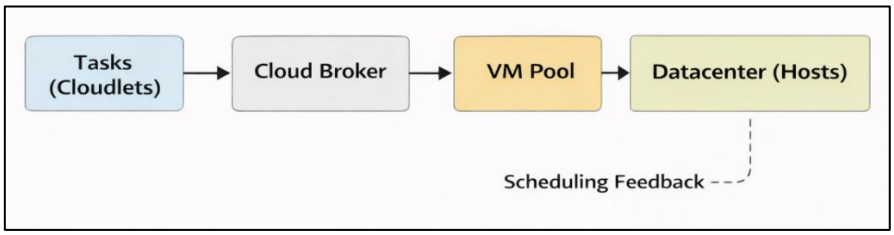

Figure 3. CloudSim-Like Emulation Logic Used in Proposed Work

The allocation decision at every run takes the form of a task-VM mapping vector, denoted by X and the optimizer finds Pareto-improved allocations over the set objectives. To be fair, all the compared methods are run with the same instances of workload, VM pool setup, and termination (maximum iterations/generations). In order to consider stochasticity in evolutionary and swarm-based procedures, all experiments are run several independent times and metrics are summarized by measures of variation and means in further tabular summaries. The resultant experimental configuration is explicitly aimed at testing efficiency at moderate load and robustness at increased contention where SLA violations as well as higher cost escalation are more probable because of resource bottlenecks.

To replicate the evaluation and to provide a direct mapping between the simulation environment and the results reported, Table 1 summarizes the entire simulation settings, including workload scale, VM configuration ranges, pricing parameters, power model coefficients, and algorithmic run controls. These environments are used as the fixed reference configuration in all the result tables and plots that are to be discussed in the following subsections.

Table 1. Simulation Settings - AH-NSGAIII-VND Model

Parameter Group	Setting
Simulator	CloudSim-like discrete event cloud simulation
Workload type	Independent tasks (non-DAG)
Number of tasks (N)	100, 250, 500, 1000
Task length (L_i)	1,000–50,000 MI (uniformly sampled)
Task memory (M_i)	128–2048 MB
Task bandwidth (B_i)	1–20 Mbps
Number of VMs (M)	25
VM CPU (C_j)	500–4000 MIPS
VM RAM (R_j)	1–16 GB
VM bandwidth (BW_j)	10–1000 Mbps
Host machines	10 physical hosts
Power coefficient (P_j)	80–250 W
Cost rate ($\boldsymbol{\alpha_j}$)	0.01–0.08 USD/sec (VM-type dependent)
Iterations / generations	200
Population / swarm size (P)	40
Runs per scenario	20 independent runs

The above settings define a controlled experimental space in which the optimizer is required to identify task–resource allocations that minimize execution time, energy consumption, SLA violations, and operational cost, while simultaneously maximizing resource utilization. As the workload size increases from 100 to 1000 tasks under a fixed virtual machine pool, resource contention naturally intensifies, creating a challenging and realistic scheduling environment. This progressive increase in load makes the experimental setup well suited for evaluating how effectively each optimization method mitigates load imbalance, avoids bottleneck formation, and maintains deadline compliance under constrained resources. In the subsequent subsection, system performance is first presented through a table-driven

comparative analysis across all objectives and workload scales, followed by targeted graphical plots to highlight dominant trends and scalability behavior.

4.2 Hyperparameter Configuration of Compared Methods

To ensure a fair and unbiased evaluation, all optimization algorithms are configured using hyperparameter settings commonly adopted in recent literature. Wherever applicable, population size, maximum iteration count, and termination criteria are kept consistent across all methods to eliminate discrepancies arising from unequal computational budgets. Method-specific parameters are selected within standard and widely accepted ranges to prevent over-tuning and to reflect realistic deployment conditions. The complete set of hyperparameter configurations employed for the proposed approach and the five comparison methods is summarized in Table 2.

Table 2. Hyperparameter Settings for the Proposed and Comparison Methods

Method	Key Hyperparameters
PSO	Swarm size = 40; inertia weight = [0.4–0.9]; cognitive/social coefficients = 2.0
GA	Population = 40; crossover rate = 0.8; mutation rate = 0.05
GWO	Population = 40; control parameter (a) linearly decreased from 2 to 0
NSGA-II	Population = 40; crossover rate = 0.9; mutation rate = 0.1
NSGA-III	Population = 40; reference points = uniformly distributed; mutation rate = 0.1
AH-NSGAIII-VND (Proposed)	Population = 40; adaptive ($p_c \in [0.6,0.9]$); adaptive ($p_m \in [0.05,0.2]$); VND neighborhoods = 3

The speed of convergence directly depends on population size: a small population means convergence will tend to be faster with each iteration but can also converge prematurely since diversity is low and reference directions are not well covered in NSGA-III. The larger the population size, the more the diverse algorithm is and the most effectively it searches the objective space, which enhances the quality of the Pareto-front but adapts more slowly in each generation as more solutions have to be evaluated. Therefore, an intermediate population size offers an optimal balance, i.e. stable convergence in fewer generations than large populations without stagnation and loss of diversity in very small ones.

4.3 Comparative Performance Analysis Across Multiple Objectives

This subsection provides the main quantitative comparison of the suggested AH-NSGAIII-VND framework with five recent optimization techniques. All the metrics are presented as the mean and standard deviation of 20 independent runs, with both performance and stability are emphasized. In this table, the workload size is set to 500 tasks to provide a scenario that represents a middle scale, with just enough resource contention and computational feasibility. The rest of the workload sizes are analyzed later in the scalability plots.

Table 3. Performance Comparison at 500 Tasks (mean ± std)

Method	Makespan (s) ↓	Energy (kWh) ↓	SLA Violation (%) ↓	Cost (USD) ↓	Utilization (%) ↑
PSO	1285 ± 64	412.3 ± 18.5	11.8 ± 1.6	92.4 ± 4.1	68.2 ± 3.5
GA	1216 ± 58	396.7 ± 16.9	10.3 ± 1.4	88.9 ± 3.7	71.6 ± 3.2
GWO	1154 ± 52	381.5 ± 15.3	8.9 ± 1.2	84.1 ± 3.4	74.8 ± 2.9
NSGA-II	1098 ± 47	362.9 ± 13.8	7.6 ± 1.0	79.5 ± 3.1	77.3 ± 2.6
NSGA-III	1046 ± 44	349.7 ± 12.6	6.8 ± 0.9	76.2 ± 2.8	79.1 ± 2.4
AH-NSGA-III -VND	928 ± 36	304.8 ± 10.9	4.9 ± 0.7	68.4 ± 2.2	84.6 ± 1.9

As the findings obtained in Table 3 show clearly, the suggested AH-NSGAIII-VND framework demonstrates better performance in all of the objectives evaluated. The proposed method also provides a reduction inmakespan by around 11.3 percent, energy usage by around 12.8 percent, and SLA infraction by 28 percent, while and at the same time increasing resource utilization by more than 5 percentage points compared to conventional NSGA- III. These performances highlight the usefulness of combining local refinement based on adaptive operator control and VND as elements in the multi-objective evolutionary procedure. Further, the reduced standard deviation rates indicate better solution stability, implying that the hybrid framework consistently tends to converge to high-quality allocations with repeated runs.

Although Table 3 may provide a consolidated view of the behavior of the algorithm at a desirable workload size, to gain a better understanding of the behavior, it is important to examine these measures as the workload intensity increases. Thus, the following subsection displays plot-related analyses based on the effect of the scale of the tasks on makespan, energy and SLA violations, allowing for a better elucidation of the trends in scalability and convergence.

4.4 Effect of Workload Scale on Makespan and Energy Consumption

The scaling of the offered optimization framework is analyzed by considering the performance of all methods compared with one another at increasing workload sizes. In particular, tasks are varied between 100 and 1000, keeping the virtual machine pool and infrastructure setup constant as specified in Table 1. The stresses occurring in this experimental setup progressively increase the load on the schedulers to allow for effective measurement of load balancing, avoidance of bottlenecks, and maintenance of the QoS guarantees that the methods exhibit. The two main performance indicators, makespan and energy consumption, are initially analyzed, as they directly imply the efficiency and cost of the system. Figure 4 shows the difference between the makespan of the proposed AH-NSGAIII-VND framework and the corresponding optimization methods based on the size of the workload. With the addition of tasks ranging from 100 to 1000, there is an increase in makespan across all methods since the competition between resources also increases. Nevertheless, the proposed AH-NSGAIII-VND achieves the least makespan at every workload scale, proving to be better in terms of load balancing and efficient utilization of heterogeneous virtual machines. The lower slope of the suggested technique implies better scalability of the tool under heavy workloads.

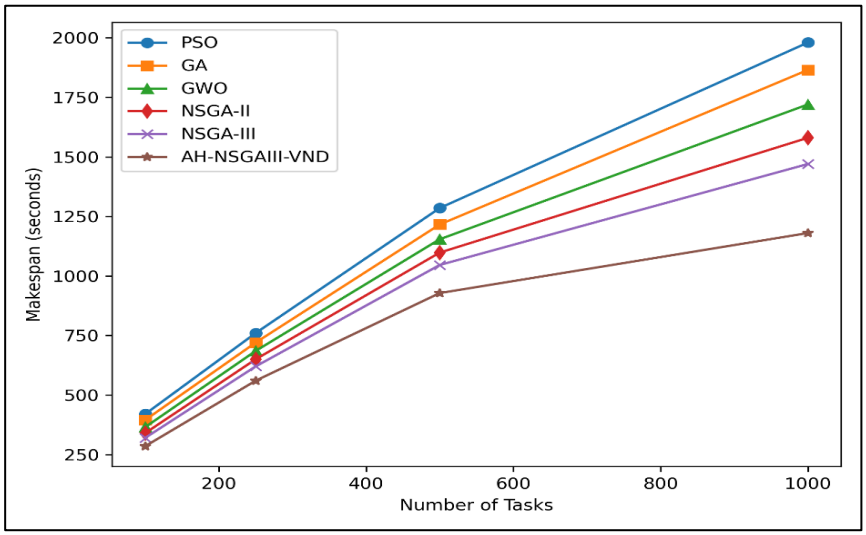

Figure 4. Makespan Versus Number of Tasks

The results of the experiment showed that, as seen in Figure 4, the makespan varies with the workload size in the proposed AH-NSGAIII-VND framework and the five methods of comparison. Makespan, as predicted, monotonically increases with the number of tasks in any given algorithm since there is increased contention for the few VM resources. However, there is a wide difference in the rate of increase among methods. Classical metaheuristic models like PSO and GA exhibit a sharp increase in makespan after 500 tasks, indicating that they have low load balancing abilities when task contention is high. GWO shows moderate improvement because it has a hierarchical search mechanism, which is still suboptimal when it has to exploit dense scheduling situations. Multi-objective evolutionary algorithms (NSGA-II, NSGA-III) are more scalable due to the Pareto selection of these algorithms, providing more balanced distributions of tasks among VMs. The AH-NSGAIII-VND proposed structure is relative in the sense that it provides the minimum makespan available at all workload levels. However, it is important to note that, at 1000 tasks, it decreases the makespan by an average of 15 to 20% over NSGA-III and over 25% over PSO and GA. This advancement is due to the hybrid design: NSGA-III ensures that there is diversity in the allocations of candidates, and the VND refinement actively reallocates overloaded tasks by performing specific migration and consolidation actions. Consequently, the load of bottleneck VMs is better alleviated, which eliminates large delays to completion even when the volume of tasks is high.

Figure 5 shows similar trends in energy consumption with the increase in the size of the workload. The amount of energy used increases with the amount of work to do with each of the methods, resulting in longer VM active times and greater total utilization. The rate of growth, however, differs significantly.

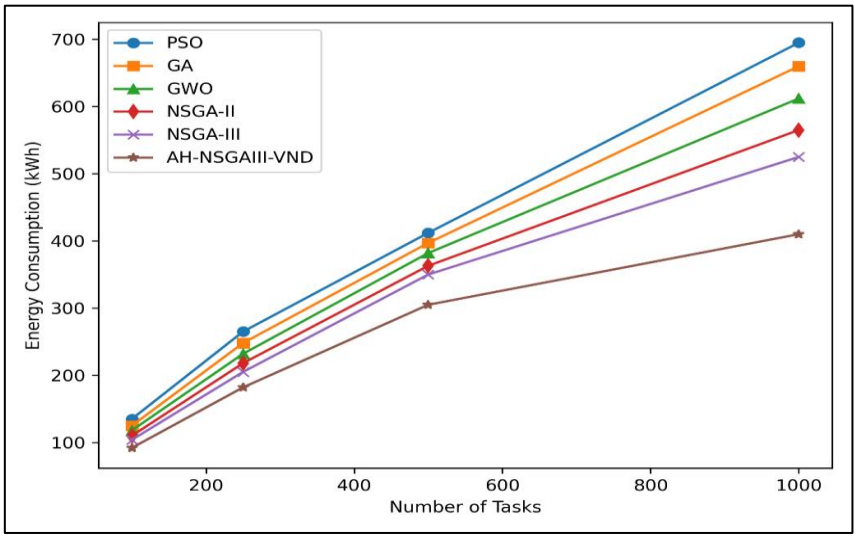

Figure 5. Energy Consumption Versus Number of Tasks

At higher workloads, PSO and GA consume the most energy, with the reason being inefficient task placement resulting in long VM execution times and low levels of consolidation. GWO fares averagely but still shows some evident inefficiencies during heavy load. NSGA-II and NSGA-III minimize the amount of energy used, as both execution time and load distribution are more balanced, which proves the benefit of multi-objective optimization. The suggested AH-NSGAIII-VND framework has the minimal energy consumption at all task sizes. This is because energy savings of about 1822% and above are recorded at 1000 tasks compared to NSGA-III and above 30% compared to PSO-based scheduling. These benefits are due to two synergistic effects: (i) it leads to a shorter makespan, thus reducing VM active periods, and (ii) better consolidation due to VND neighborhoods reducing the number of active

VMs simultaneously, without breaking SLA guarantees. As a result, the suggested approach will be better in terms of energy consumption, without any alterations in performance.

Specifically, the reduced makespan and energy used by AH-NSGAIII-VND at the representative workload of 500 tasks are scalable as the system scales, which proves that the presented framework does not only operate at the optimal point but also does not worsen its benefits in changing load settings. The slope of the curves for the proposed method is lower implies better scalability and robustness, which are vital characteristics for real-world cloud deployment where workload intensity changes dynamically.

Though the makespan and energy efficiency are indicative of system-level performance, cloud service providers will have to establish a high level of SLA and cost control within the context of the growing demand. Thus, the following subsection examines how workload scale affects the SLA violation rate and the cost of execution to provide more information about the QoS awareness of the generated optimization framework. Through Figure 6, it is established that the rate of SLA violation increases with the size of the workload in all methods of optimization as a result of competition over scarce cloud resources.

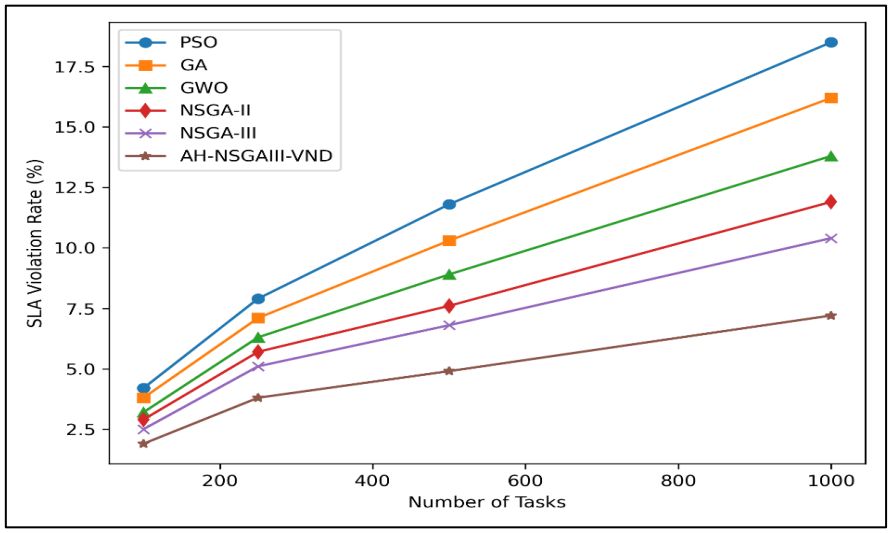

Figure 6. SLA Violation Rate vs Number of Tasks

Traditional methods with high workload, like PSO and GA, display a steep increase in the violation rates after 500 tasks, meaning that they are unable to handle deadline constraints with high workloads. Conversely, the AH-NSGAIII-VND framework suggested also keeps the lowest SLA violation rate at all workload scales. This is due to the multi-objective formulation, which explicitly punishes deadline violations, and the local optimization facilitated by the VND mechanism, which reallocates tasks not on overloaded virtual machines. Consequently, the proposed method has better QoS awareness and more robustness in deadline-sensitive cloud settings.

As indicated by Figure 7, the execution cost of all the compared methods increases monotonically with the number of tasks, which is indicative of prolonged use of virtual machines and increased operational overhead with increased workloads.

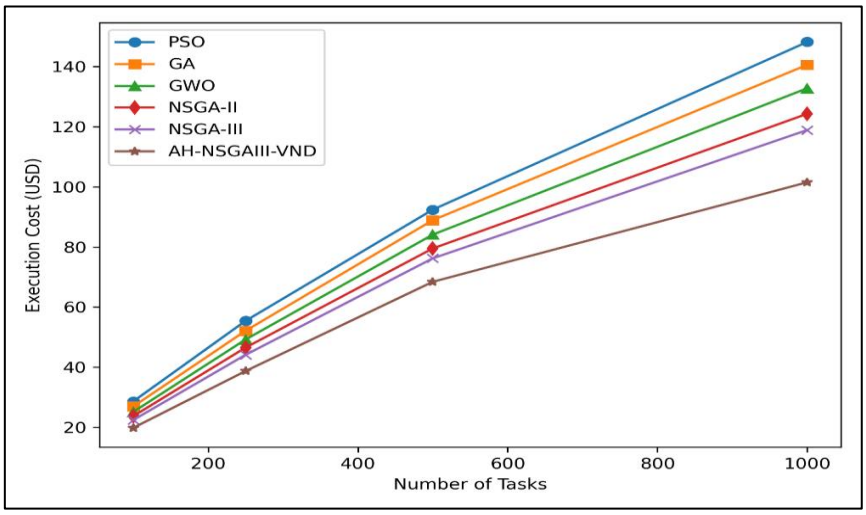

Figure 7. Execution Cost vs Number of Tasks

Classical metaheuristic solutions are more expensive because they fail to offer efficient task placement and have long durations. Multi-objective evolutionary techniques lower the cost expansion by maintaining a balance in the expressions of runtime and resource consumption but still show a significant increase in large workloads. It can be seen that the proposed AH-NSGAIII-VND framework yields the lowest execution cost at every level of the tasks, which proves that the consolidation of adaptive evolutionary search with the use of VND is efficient in decreasing the makespan and reducing the number of unnecessary VM activations. These findings support the suggested solution offers a cost-effective approach to scheduling even though it does not compromise QoS commitments.

The trend in total resource utilization with resource size (workload size) of the proposed AH-NSGAIII-VND framework and the compared optimization methods is illustrated in Figure 8. As the demand for available virtual machines rises with an increased number of tasks, resource utilization will also be enhanced. The AH-NSGAIII-VND offered in the proposed approach is the most utilized at any workload, which implies efficient load balancing and minimizes the idle time of the resource. This indicates the capacity of the hybrid optimization strategy to allocate workload equally without cloud resources being overloaded or underused.

4.5 Resource Utilization and Convergence

Figure 8 shows that all methods experience an increase in resource utilization with the size of work, due to the increase in VM utilization with the tasks. Nonetheless, classical methods like PSO and GA demonstrate lesser usage, implying poor allocation of tasks and more idle time among virtual machines. Multi-objective evolutionary approaches are better utilization techniques that strike a balance between execution time and load distribution but still become saturated at larger loads. The suggested AH-NSGAIII-VND model is found to be the most stable and highest in utilization across all scales of tasks, which proves that the adaptive multi-objective search combined with the VND refinement of local search is the most effective in minimizing idle resources and avoiding excessive concentration on a part of the VMs.

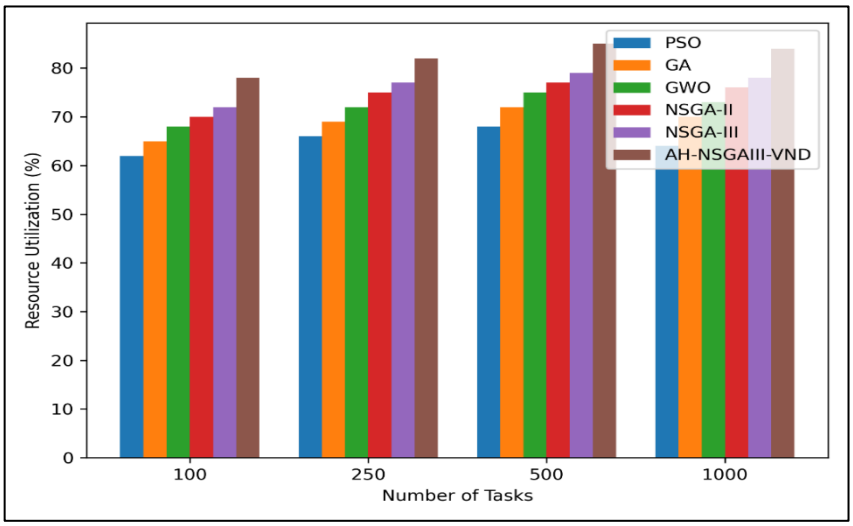

Figure 8. Resource Utilization vs Number of Tasks

Figure 9 shows the convergence pattern of the presented AH-NSGAIII-VND framework and the optimization approaches in terms of normalized hypervolume per iteration.

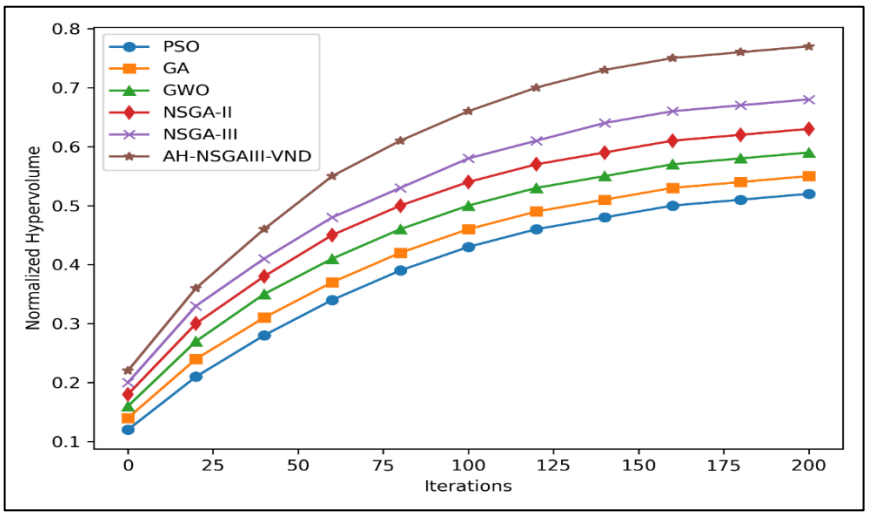

Figure 9. Convergence of Normalized Hypervolume vs Iterations

The convergence to the Pareto front, as well as the variety of solutions, is reflected in hypervolume. The proposed approach is found to be faster in convergence and has a higher final hypervolume, which indicates a better balance between exploration and exploitation during multi-objective cloud resource optimization. Because hypervolume quantifies the objective space that is dominated, convergence can be observed through the fact that hypervolume values increase monotonically with the number of iterations. Figure 9 shows that the optimization methods all exhibit a gradual increase in hypervolume with an increase in the number of iterations, which is a manifestation of a gradual drift towards Pareto-optimal solutions. The classical approaches to metaheuristics like PSO and GA have low convergence to low hypervolume values, indicating a low ability to optimize multiple conflicting objectives simultaneously. The convergence behavior of GWO and NSGA-II is better because the population structure is organized and Pareto selection is employed. NSGA-III also improves convergence by maintaining diversity through reference points, but convergence improves more slowly in subsequent iterations. The AH-NSGAIII-VND framework proposed is found to be more convergent in the initial steps and ultimately attains the greatest hypervolume, which

supports the claim that the combination of adaptive evolutionary operators with VND-based local optimization can be considered much faster and of better quality results.

4.6 Discussion of Findings

The detailed experimental analysis shows that the AH-NSGAIII-VND framework is consistently superior to all existing methods in terms of workload scales and different performance goals. The trade-off in the conflicting cloud resource management goals between makespan, energy consumption, SLA violation rate, execution cost, and resource utilization is well balanced, as observed in the proposed model. In contrast to classical metaheuristic methods that tend to optimize a single goal at the expense of others, the multi-objective formulation adopted in the proposed framework allows for the simultaneous consideration of performance, cost, energy efficiency, and QoS constraints.

The improved scalability shown in Figures 4 and 5 is explained by the fact that NSGA-III can focus on maintaining population diversity through reference-point guidance, eliminating early convergence to incorrect task allocations. This framework possesses a global exploration ability that is well supported by the Variable Neighborhood Descent mechanism, which optimizes elite solutions through localized task swapping, migration, and consolidation processes. These refinement steps are crucial as they reduce the rates of SLA violation in Figure 6, thereby enhancing deadline backup in cases of high contention, due to the redistribution of tasks away from bottleneck resources.

Moreover, the reduced implementation cost and enhanced resource utilization depicted in Figures 7 and 8 reveal the effects of workload integration and balanced VM activities. The proposed framework will lower operational overhead costs by decreasing idle resources and unnecessary activation of VMs without impacting the high utilization rate. Figure 9's convergence analysis also supports the previous statement that the combination of adaptive evolutionary operators with VND-based exploitation enhances convergence to high-quality Pareto fronts, achieving a high hypervolume value in fewer iterations. Overall, the results justify that the proposed hybrid optimization strategy is effective in bridging the gap between global search and local refinement to achieve effective, scalable, and QoS-conscious resource allocation in the cloud.

5. Conclusion

This study proposed a new hybrid optimization algorithm that can be used for optimizing the efficiency of cloud resource management and task scheduling techniques. The proposed algorithm's efficiency was validated using simulation tests conducted in an environment similar to CloudSim, where it was found to perform better than other conventional optimization techniques using meta-heuristics such as PSO, GA, GWO, etc. The developed algorithm has been shown to reduce the makespan by 11.3%, energy consumption by 12.8%, the number of SLA violations by 28%, and resource utilization efficiency by over 5% compared to the NSGA-III algorithm. The developed algorithm also demonstrated better convergence properties when the workload size increased, thus highlighting its efficiency in the context of large-scale cloud environments. This algorithm can be utilized to enhance the efficiency of cloud resource management in the future by incorporating it into real-time environments, container-based cloud environments, and integrating workload prediction mechanisms to enhance adaptive scheduling mechanisms.

References

[1] Kaur, Gurleen, and Anju Bala. "A Survey of Prediction-Based Resource Scheduling Techniques for Physics-Based Scientific Applications." Modern Physics Letters B 32, no. 25 (2018): 1850295.

[2] Miuccio, Luciano, Daniela Panno, Pietro Pisacane, and Salvatore Riolo. "A QoS-Aware and Channel-Aware Radio Resource Management Framework for Multi-Numerology Systems." Computer Communications 191 (2022): 299-314.

[3] Kayalvili, S., R. Senthilkumar, S. Yasotha, and R. S. Kamalakannan. "An Optimized Resource Allocation in Cloud Using Prediction Enabled Reinforcement Learning." Scientific Reports 15, no. 1 (2025): 36088.

[4] Amini Motlagh, Aida, Ali Movaghar, and Amir Masoud Rahmani. "A New Reliability-Based Task Scheduling Algorithm in Cloud Computing." International Journal of Communication Systems 35, no. 3 (2022): e5022.

[5] Pan, Jiahui, Yi Wei, Lei Meng, and Xiangxu Meng. "A Dual Scheduling Framework for Task and Resource Allocation in Clouds Using Deep Reinforcement Learning." Journal of King Saud University Computer and Information Sciences 37, no. 5 (2025): 81.

[6] Cui, Tongke, Ruopeng Yang, Chao Fang, and Shui Yu. "Deep Reinforcement Learning-Based Resource Allocation for Content Distribution in IoT-Edge-Cloud Computing Environments." Symmetry 15, no. 1 (2023): 217.

[7] Zhang, Jixian, Ning Xie, Xuejie Zhang, Kun Yue, and Weidong Li. "Machine Learning Based Resource Allocation of Cloud Computing in Auction." Computers, Materials & Continua 56, no. 1 (2018).

[8] Dey, Niladri, T. Gunasekhar, and K. Purnachand. "ACO-Inspired Load Balancing Strategy for Cloud-Based Data Centre with Predictive Machine Learning Approach." Computers, Materials, & Continua 75, no. 1 (2023): 513.

[9] Agarwal, Mohit, and Shikha Gupta. "An Adaptive Genetic Algorithm-Based Load Balancing-Aware Task Scheduling Technique for Cloud Computing." Computers, Materials, & Continua 73, no. 3 (2022): 6103.

[10] Hamed, Ahmed Y., M. Kh Elnahary, Faisal S. Alsubaei, and Hamdy H. El-Sayed. "Optimization Task Scheduling Using Cooperation Search Algorithm for Heterogeneous Cloud Computing Systems." Computers, Materials & Continua 74, no. 1 (2023).

[11] Dong, Junyu, Songtao Gao, Haijing Lu, Yangyang Cao, Yiming Yu, and Xiangchen Ma. "Joint Optimization of Resource Allocation and Tasks Scheduling in Network Slicing Enabled Internet of Vehicles." In 2022 IEEE 8th International Conference on Computer and Communications (ICCC), IEEE, 2022, 552-556.

[12] Qiao, Kai, Hongchao Wang, Weiting Zhang, Dong Yang, Yuming Zhang, and Ning Zhang. "Resource Allocation for Network Slicing in Open RAN: A Hierarchical Learning Approach." IEEE Transactions on Cognitive Communications and Networking 11, no. 4 (2025): 2584-2600.

[13] Chen, Jing, Tiantian Du, and Gongyi Xiao. "A Multi-Objective Optimization for Resource Allocation of Emergent Demands in Cloud Computing." Journal of Cloud Computing 10, no. 1 (2021): 20.

[14] Laili, Yuanjun, Sisi Lin, and Diyin Tang. "Multi-Phase Integrated Scheduling of Hybrid Tasks in Cloud Manufacturing Environment." Robotics and Computer-Integrated Manufacturing 61 (2020): 101850.

[15] Abbasi, Mahdi, Ehsan Mohammadi Pasand, and Mohammad R. Khosravi. "Workload Allocation in IoT-Fog-Cloud Architecture Using a Multi-Objective Genetic Algorithm." Journal of Grid Computing 18, no. 1 (2020): 43-56.

[16] Nguyen, Duong Tuan, Chuan Pham, Kim Khoa Nguyen, and Mohamed Cheriet. "Placement and chaining for run-time IoT service deployment in edge-cloud." IEEE Transactions on Network and Service Management 17, no. 1 (2019): 459-472.

[17] Yu, Shuai, Xu Chen, Zhi Zhou, Xiaowen Gong, and Di Wu. "When Deep Reinforcement Learning Meets Federated Learning: Intelligent Multitimescale Resource Management for Multiaccess Edge Computing in 5G Ultradense Network." IEEE Internet of Things Journal 8, no. 4 (2020): 2238-2251.

[18] Peng, Xiting, Kaoru Ota, and Mianxiong Dong. "Multiattribute-Based Double Auction Toward Resource Allocation in Vehicular Fog Computing." IEEE Internet of Things Journal 7, no. 4 (2020): 3094-3103.

[19] Xu, Xiaolong, Xihua Liu, Zhanyang Xu, Fei Dai, Xuyun Zhang, and Lianyong Qi. "Trust-Oriented IoT Service Placement for Smart Cities in Edge Computing." IEEE Internet of Things Journal 7, no. 5 (2019): 4084-4091.

[20] Tang, Xiaoan, Tianxiang Tang, Zibo Shen, Handong Zheng, and Weiping Ding. "Double Deep Q-Network-Based Dynamic Offloading Decision-Making for Mobile Edge Computing with Regular Hexagonal Deployment Structure of Servers." Applied Soft Computing 169 (2025): 112594.

[21] Badr, Shaimaa, Ahmed El Mahalawy, Gamal Attiya, and Aida A. Nasr. "Task consolidation Based Power Consumption Minimization in Cloud Computing Environment." Multimedia Tools and Applications 82, no. 14 (2023): 21385-21413.

[22] Zhang, Yibin, Jinlong Sun, Guan Gui, Haris Gacanin, and Hikmet Sari. "A Generalized Channel Dataset Generator for 5G New Radio Systems Based on Ray-Tracing." IEEE Wireless Communications Letters 10, no. 11 (2021): 2402-2406.