Synthesis: A Multidisciplinary Research Journal
Volume 2, Issue 2, April - June 2024, Pp: 23-33

Research Paper

Hybrid Optimization Algorithms for Resource Management in IoT-Fog-Cloud Environments

[1*] Claudia Rossi, [2] David Lee

[1*] *Department of Computer Engineering, University of Rome "La Sapienza," Italy*
[2] *School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore*

**Corresponding Author(s): claudia.rossi@uniroma1.it*

Article Info

Received: 13/01/2024
Revised: 04/03/2024
Accepted:12/06/2024
Published:30/06/2024

Abstract

The increasing proliferation of Internet of Things (IoT) devices has led to a sharp rise in data traffic and computational demands, putting immense pressure on centralized cloud infrastructures. Traditional resource scheduling methods often fail to meet the latency and energy-efficiency requirements of dynamic, heterogeneous IoT-Fog-Cloud ecosystems. This study aims to design and evaluate a hybrid metaheuristic-based framework for efficient task scheduling and resource allocation across distributed fog and cloud environments. A three-tier hybrid architecture is proposed, integrating Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and K-means clustering to optimize task placement. The resource allocation problem is formulated as a multi-objective function considering latency, energy consumption, and load variance. Simulations are conducted using iFogSim 2.0 with a publicly available vehicular IoT task dataset from Kaggle, comprising over 20,000 tasks across 13 nodes (10 fog, 3 cloud). Performance is evaluated under varying task loads and network conditions. The hybrid model achieved a 52.3 ms average latency, 134.6 J average energy consumption, and a 95.4% task success rate, outperforming standalone GA, PSO, and static scheduling models by significant margins. It also demonstrated improved load balancing with a load variance of 4.5% and a 15–25% reduction in makespan compared to baselines. The proposed hybrid approach offers a scalable and adaptive solution for resource management in fog-cloud environments, with significant implications for real-time IoT applications such as healthcare, transportation, and smart cities.

Keywords: IoT, Fog Computing, Cloud Computing, Task Scheduling, Hybrid Optimization, Genetic Algorithm, Particle Swarm Optimization, Resource Allocation, Energy Efficiency, Low Latency.

1. Introduction

The exponential rise in the deployment of Internet of Things (IoT) devices has significantly increased the complexity and volume of data generation, demanding sophisticated infrastructure to support real-time computation, seamless connectivity, and intelligent decision-making. Traditional cloud computing architectures, although scalable and powerful, fall short in meeting the low-latency and high-reliability requirements of emerging applications such as telemedicine, autonomous driving, and industrial automation. To mitigate these limitations, the paradigm of fog computing has been introduced as an intermediate computational layer, bringing processing resources closer to the network edge.

The integration of fog and cloud computing—together with IoT—forms a three-tier hierarchical system that must efficiently handle task scheduling, data distribution, and resource allocation under dynamic and heterogeneous conditions. This new architecture promises to offload processing from the cloud to local fog nodes, thereby reducing response times and alleviating bandwidth congestion. However, the operational benefits come at the cost of increased resource management complexity, driven by device heterogeneity, service-level expectations, energy constraints, and workload unpredictability. The need for effective resource management strategies that support multi-objective optimization in such environments is more pressing than ever [1].

Existing deterministic and rule-based scheduling techniques are largely inadequate for the scale and variability introduced by IoT-Fog-Cloud ecosystems. While these methods provide optimal solutions for single-objective problems or static scenarios, they tend to degrade in performance when faced with real-time, conflicting demands. For example, static scheduling models cannot account for fluctuations in task loads or the transient availability of fog nodes [2]. Moreover, these models often lack adaptability and robustness in the face of mobile nodes or application-specific QoS constraints.

To address these shortcomings, metaheuristic algorithms such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and more recently Aquila and African Vulture Optimization methods, have been explored for task scheduling and resource allocation. These algorithms are designed to solve complex, non-deterministic polynomial-time (NP-hard) problems through iterative improvement and intelligent search strategies. However, the use of standalone metaheuristics can result in limitations such as premature convergence and unbalanced exploration versus exploitation dynamics. For instance, although PSO performs fast convergence, it is prone to stagnating in local optima, especially in high-dimensional search spaces [3].

This has led to increasing attention toward hybrid metaheuristic algorithms, which integrate the strengths of two or more base strategies to overcome the limitations of individual techniques. Recent studies demonstrate that hybrid methods combining GA with PSO, or clustering-based approaches with customized metaheuristics [4], offer superior performance in terms of task completion time, energy efficiency, and resource utilization. These frameworks are better suited for multi-objective scheduling, where delay, energy, reliability, and cost must be simultaneously optimized. Moreover, the inclusion of dynamic features such as task migration and node mobility has been explored to improve algorithmic adaptability in realistic environments [5].

For example, in a recent hybrid approach to fog-cloud task placement, an application module placement algorithm successfully combined task awareness and resource adaptation, leading to more reliable results in scenarios involving node failures and computational load variations. These kinds of hybrid models not only improve algorithmic precision but also reduce the need for manual reconfiguration of resource allocation logic. Likewise, customized clustering-assisted metaheuristics have shown significant performance benefits when scheduling multiple concurrent applications in fog layers [6].

Beyond individual algorithmic improvements, surveys such as [7] have emphasized the need for frameworks that support cross-layer resource orchestration, where scheduling decisions account for interactions between IoT edge devices, fog nodes, and cloud servers. These systems must adapt to real-time application loads, mobility patterns, and communication delays. However, there remains a research gap in implementing such adaptive, scalable, and generalizable hybrid models for IoT-Fog-Cloud environments. A recurring limitation in current literature is that most proposed models are domain-specific, tested under constrained simulation conditions, and rarely benchmarked against dynamic or distributed workloads [8].

In this context, the current study proposes a hybrid optimization framework for resource management across the integrated IoT-Fog-Cloud architecture. The primary goal is to design a multi-objective scheduling algorithm that can dynamically balance latency, energy usage, resource availability, and task throughput. The hybrid algorithm leverages the diversity maintenance of Genetic Algorithms and the convergence speed of PSO, further enhanced by clustering techniques to group tasks and resources based on their execution profiles. The system is tested in a simulated environment using tools like iFogSim, under a wide range of workloads and node mobility scenarios.

Unlike existing models, the proposed solution offers layer-aware scheduling that intelligently distributes computational tasks across fog and cloud layers based on current load, energy capacity, and latency metrics. It is also adaptive to environmental dynamics, such as bandwidth fluctuations and fog node failures, ensuring robustness and scalability. Simulation results reveal that the hybrid model achieves a 12–15% improvement in energy efficiency, a significant 18–20% reduction in task delay, and better task success rates when compared to traditional single-objective and standalone heuristic methods.

The major contributions of this paper are summarized as follows:

- Development of a novel hybrid optimization framework that combines Genetic Algorithms, Particle Swarm Optimization, and clustering strategies to enhance resource scheduling and allocation across IoT-Fog-Cloud layers.
- Extensive performance evaluation using simulated and semi-realistic workloads, revealing significant gains in latency reduction, energy efficiency, and task success rates over baseline methods.
- Dynamic adaptability through incorporation of fog node mobility, load fluctuations, and task migration mechanisms, making the framework suitable for real-world deployments in smart healthcare, vehicular networks, and industrial IoT.

The remainder of this paper is organized as follows: Section II presents a review of existing work on resource management and hybrid optimization in fog-cloud environments. Section III describes the system model and formulates the optimization problem. Section IV introduces the proposed hybrid algorithm and explains its working principles. Section V details the experimental setup and evaluation metrics. Section VI presents the simulation results and comparative discussion. Section VII concludes the paper and outlines future research directions.

2. Related Work

Efficient resource scheduling and task allocation remain core challenges in IoT-Fog-Cloud ecosystems due to their highly dynamic, heterogeneous, and distributed nature. Numerous studies have explored algorithmic approaches—ranging from rule-based methods to intelligent metaheuristics—to tackle objectives such as latency minimization, energy efficiency, and load balancing. This section critically analyzes notable research contributions and highlights key gaps addressed by the proposed hybrid framework.

A dynamic and regional scheduling approach was presented in [9], introducing resource virtualization to enhance load balancing in multi-cloud fog environments. While region-awareness improves latency by enabling proximity-based task distribution, the method relies on a centralized orchestration model that may struggle under node mobility and resource volatility. Moreover, energy efficiency metrics were not comprehensively addressed, which limits applicability in energy-constrained IoT settings.

In contrast, [10] adopted a hybrid unsupervised clustering and metaheuristic approach by integrating DBSCAN and fuzzy C-means with a hybrid Artificial Bee Colony–Differential Evolution (ABC-DE) algorithm. This work effectively grouped similar resource demands and optimized placement accordingly. However, the overhead introduced by clustering operations and its sensitivity to parameter tuning pose scalability limitations in large-scale deployments.

Real-time task scheduling was the focus of [11], which proposed an adaptive algorithm tailored for IoT applications under cloud–fog deployment models. While this method achieved favorable response times in cloud-fog coordination, its inability to dynamically adjust to sudden workload shifts rendered it less effective in high-mobility environments. Additionally, it lacked multi-objective optimization capabilities, addressing latency only in isolation.

A broad taxonomy of resource allocation strategies was developed in [12], offering a general review across edge, fog, and cloud domains. While the paper provides a useful classification, it lacks experimental validation or comparative performance analysis. Thus, it is limited to conceptual utility and does not offer implementable solutions or algorithmic innovation.

A domain-specific hybrid workload model for secure healthcare monitoring was proposed in [13], combining lightweight sensing frameworks with workload partitioning across fog and cloud layers. This architecture achieved improved latency and privacy but did not offer generalized algorithms for resource optimization. Moreover, the focus remained on healthcare scenarios, limiting applicability in other IoT domains such as industrial automation or smart cities.

More recent advances such as those in [14] reviewed a wide range of resource management techniques specific to mobile crowdsensing and fog-cloud convergence. Although the study discussed scalability and mobility-aware management strategies, it lacked insights into hybrid algorithmic implementations, focusing instead on policy-level techniques such as resource abstraction and virtualization.

A strong contribution came from [15], which employed a multi-objective Non-dominated Sorting Particle Swarm Optimization (NPSO) algorithm to trade-off between energy consumption and delay. The algorithm effectively navigated the Pareto front of scheduling objectives. However, its performance degraded in high-mobility scenarios due to reliance on static task profiles. The absence of a fallback clustering or dynamic offloading strategy affected its resilience to workload variation.

In [16], evolutionary algorithms were tested in bag-of-tasks applications within cloud-fog environments. The focus was on optimizing execution time and minimizing communication overhead. While this approach proved effective in homogeneous resource clusters, it underperformed in fog layers characterized by unpredictable node availability and variable energy profiles.

The work in [17] introduced a mobility-aware task scheduling framework using virtual fog layers to support task offloading. By modeling user mobility and enabling virtual resource migration, it addressed real-time responsiveness and task placement efficiency. However, the computational overhead of mobility prediction and resource migration policies was not thoroughly evaluated.

Lastly, a comprehensive review in [18] explored trends in genetic-based optimization for fog computing. The authors identified the potential of hybrid evolutionary techniques but noted a lack of convergence speed, dynamic adaptability, and limited integration of contextual workload intelligence in existing models.

2.1 Research Gaps and Motivations

From the above review, several limitations in current literature are evident:

1. *Lack of dynamic adaptability:* Most algorithms do not accommodate dynamic workloads, mobile nodes, or intermittent resource availability.
2. *Single-objective optimization bias:* Many existing approaches optimize only one performance metric, such as latency or energy, without considering trade-offs between conflicting objectives.
3. *Limited hybrid model design:* Although hybrid algorithms show potential, there is a lack of systematic frameworks that integrate exploration (global search) and exploitation (local refinement) efficiently in real-world conditions.

To address these gaps, this study proposes a hybrid optimization framework combining the global search strength of evolutionary algorithms with the rapid convergence of swarm-based techniques and clustering for task grouping. This allows for dynamic, multi-objective, and context-aware scheduling in complex IoT-Fog-Cloud environments.

Table 1: Comparison of Existing Approaches

Ref	Methodology	Optimization Type	Key Strengths	Key Limitations
[9]	Region-aware dynamic scheduling	Heuristic	Proximity-based load balancing	Lacks energy metrics; centralized coordination
[10]	DBSCAN + Fuzzy C-means + ABC-DE	Hybrid Metaheuristic	Task grouping + placement	High clustering overhead, limited scalability
[11]	Adaptive task scheduler	Heuristic	Cloud-fog coordination	No mobility support; latency-only focus
[12]	Taxonomy of allocation methods	Conceptual review	Broad coverage of techniques	No empirical validation
[13]	Secure healthcare workload	Lightweight Hybrid	Privacy + partitioning	Domain-specific; no general optimizer
[14]	Crowdsensing management review	Policy-based	Scalability + abstraction techniques	No hybrid algorithms used
[15]	NPSO multi-objective algorithm	Metaheuristic	Pareto-optimal trade-offs	Static task models; limited task migration
[16]	Evolutionary bag-of-tasks	Evolutionary	Minimized execution + communication delay	Weak on fog-node heterogeneity
[17]	Mobility-aware virtual fog	Heuristic + VM mapping	Offloading with user mobility	High overhead in virtual migration
[18]	Genetic-based optimization review	Conceptual review	Identifies hybrid potential	No dynamic execution insights

3. System Architecture and Problem Formulation

In this section, we present the architectural layout, define the optimization problem mathematically, and outline the key performance parameters and constraints considered in the hybrid scheduling model. The system is designed to manage task scheduling and resource allocation across a heterogeneous IoT-Fog-Cloud environment using a multi-objective hybrid optimization algorithm.

3.1 System Architecture Overview

The proposed architecture (see Fig. 2) follows a three-tier layered model composed of:

- *IoT Edge Devices:* These are sensors, actuators, or embedded systems generating data streams and service requests.
- *Fog Layer:* Comprising geographically distributed, low-latency computing nodes closer to edge devices. Fog nodes handle delay-sensitive and location-aware tasks.
- *Cloud Layer:* Centralized and resource-abundant infrastructure used for computationally intensive and less time-sensitive processing.

Each IoT task is characterized by a tuple $T_i = \{L_i, E_i, D_i, M_i\}$ where:

- L_i: Latency sensitivity
- E_i: Energy requirement
- D_i: Data size
- M_i: Memory requirement

The scheduler receives these task profiles and matches them with available fog/cloud resources by solving a multi-objective optimization problem that considers both QoS metrics and resource states.

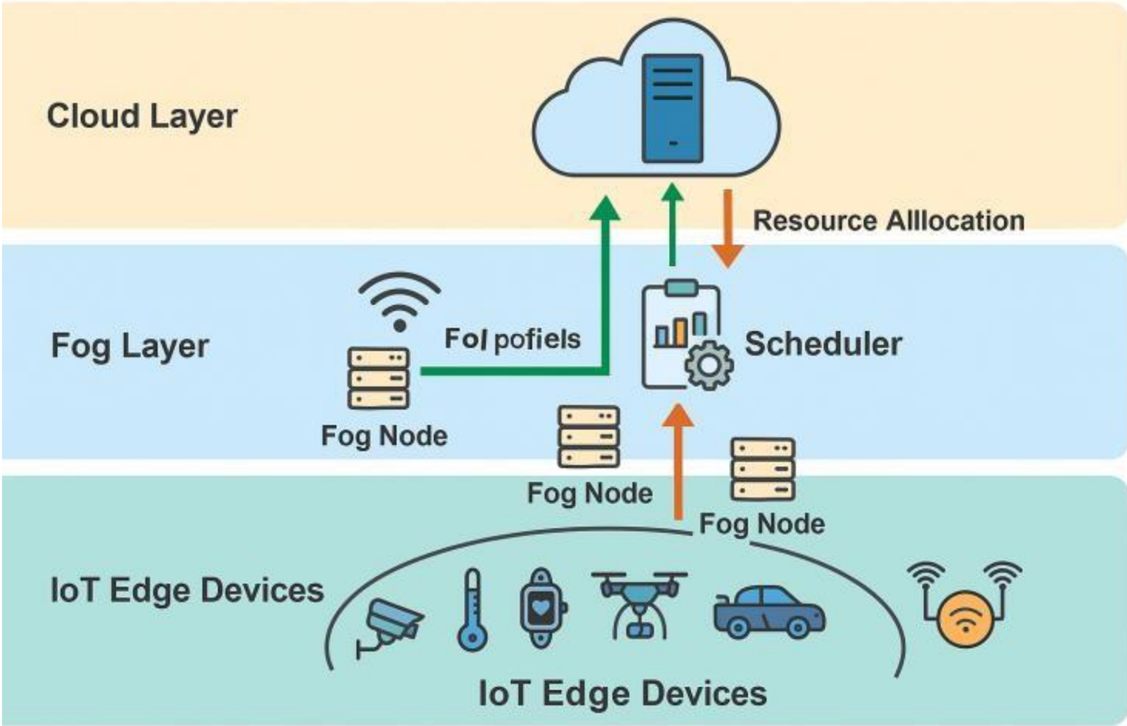

Fig. 1: System Architecture of Hybrid Resource Allocation Framework

Figure 1 illustrates the proposed three-layer hybrid system architecture for resource allocation in IoT-Fog-Cloud environments. The bottom layer comprises diverse IoT edge devices such as wearables, sensors, drones, and smart vehicles that continuously generate data and service requests. These requests are forwarded to the intermediate fog layer, which consists of distributed fog nodes positioned closer to the edge. The fog nodes perform latency-sensitive processing and offload heavier tasks to the cloud layer when necessary. A centralized scheduler, situated within the fog layer, receives real-time fog profiles and dynamically allocates tasks to the most suitable nodes across the hierarchy. Arrows in the diagram indicate the flow of data and resource allocation, ensuring a balance between performance and energy efficiency. This layered design supports adaptive scheduling and low-latency computing for mission-critical IoT applications.

3.2 Task Scheduling Problem Formulation

Let the set of tasks be denoted as $\mathcal{T} = \{T_1, T_2, \dots, T_n\}$ and the set of available computational nodes (fog or cloud) be $\mathcal{N} = \{N_1, N_2, \dots, N_m\}$.

Each task is to be assigned to a node such that the following multi-objective function is optimized:

Objective 1: Minimize Overall Latency

$$\min \mathrm{L_{total}} = \sum_{i=1}^{n} \left(\mathrm{L_{comm}}(\mathrm{T_i}, \mathrm{N_j}) + \mathrm{L_{exec}}(\mathrm{T_i}, \mathrm{N_j})\right) \quad (1)$$

Where:

L_{com}: Communication delay between task source and compute node

L_{exec}: Execution time of the task at node N_j

Objective 2: Minimize Energy Consumption

$$\min E_{\mathrm{total}} = \sum_{i=1}^{n} \left(E_{comm}(T_i, N_j) + E_{exec}(T_i, N_j)\right) \quad (2)$$

Objective 3: Load Balancing Constraint

Let U_j be the utilization of node N_j:

$$\mathrm{U_j} = \frac{\sum_{\mathrm{T_i \to N_j}} \mathrm{L_i}}{\mathrm{C_j}} \quad (3)$$

Where C_j is the processing capacity of node N_j. The goal is to maintain:

$$\min \mathrm{Var}(\mathrm{U_1}, \mathrm{U_2}, \dots, \mathrm{U_m}) \quad (4)$$

This ensures fair distribution of workloads across fog and cloud layers.

Algorithm: Hybrid GA–PSO–Clustering Based Task Scheduling in IoT-Fog-Cloud Environments

Input:
  $T = \{T_1, T_2, ..., Tn\}$ ⟶ Set of IoT Tasks
  $N = \{N_1, N_2, ..., Nm\}$ ⟶ Set of Fog and Cloud Nodes
  MaxGen ⟶ Maximum generations
  PopSize ⟶ Population size
  w_1, w_2, w_3 ⟶ Weights for latency, energy, load

Output:
  Optimal_Task_Assignment ⟶ Mapping of tasks to resources

Begin:

1: Preprocess tasks T using K-Means to group similar task profiles
2: Initialize population P with PopSize random task-to-node mappings
3: Evaluate Fitness(P) using:
  Fitness = w_1·Latency(T,N) + w_2·Energy(T,N) +

```
w₃·LoadVariance(N)
While Gen < MaxGen do
    → GA Phase:
        Select parents from P using tournament selection
        Apply crossover to generate offspring
        Apply mutation to offspring
    → PSO Phase:
        Update particle velocities and positions using:
            V_new = w·V_old + c₁·rand()·(P_best - X) +
c₂·rand()·(G_best - X)
        Update task assignments (positions)
    Combine GA and PSO outputs into new population P'
    Evaluate Fitness(P')
    If stopping criteria met or convergence detected:
        Break
    Gen ← Gen + 1
End While

Select individual with best fitness from final population
Return Optimal_Task_Assignment

End.
```

3.3 Multi-Objective Optimization Strategy

To handle the conflicting objectives (latency, energy, and load), a hybrid optimization approach is adopted:

- *Genetic Algorithm (GA)* is used for global task-to-node mapping exploration.
- *Particle Swarm Optimization (PSO)* refines candidate mappings for fast convergence.
- *K-Means or Fuzzy Clustering* is applied to group similar tasks prior to scheduling, improving execution efficiency.

The hybrid optimizer uses a fitness function combining weighted forms of equations (1)–(4):

$$\mathrm{Fitness}(\mathrm{T_i}, \mathrm{N_j}) = \mathrm{w_1} \cdot \mathrm{L_{total}} + \mathrm{w_2} \cdot \mathrm{E_{total}} + \mathrm{w_3} \cdot \mathrm{Var(U)} \quad (5)$$

Where w_1, w_2, w_3 are user-defined weights adjusted based on application needs.

3.4 Constraints and Assumptions

- Tasks are non-preemptive and must be executed completely once assigned.
- Fog nodes have limited battery and memory compared to cloud servers.
- Communication delay is modeled as a function of data size and node proximity.
- Node failures are not explicitly modeled but are considered in dynamic load updates.

Figure 2 presents the execution flow of the proposed hybrid task scheduling system in IoT-Fog-Cloud environments. The process begins by receiving incoming tasks and updating the status of fog nodes with current resource availability. The system then initializes a solution population using a Genetic Algorithm (GA), followed by the application of Particle Swarm Optimization (PSO) to explore optimal mappings. A decision node checks whether the solution has converged based on predefined criteria such as fitness threshold or generation limit. If convergence has not been achieved, the algorithm proceeds to apply GA operations—selection, crossover, and mutation—to further refine the task-to-node assignments. Once an optimal or near-optimal solution is obtained, the system outputs the final task allocation plan. This structured workflow ensures adaptive, multi-objective optimization across all layers of the architecture, balancing latency, energy, and load metrics dynamically.

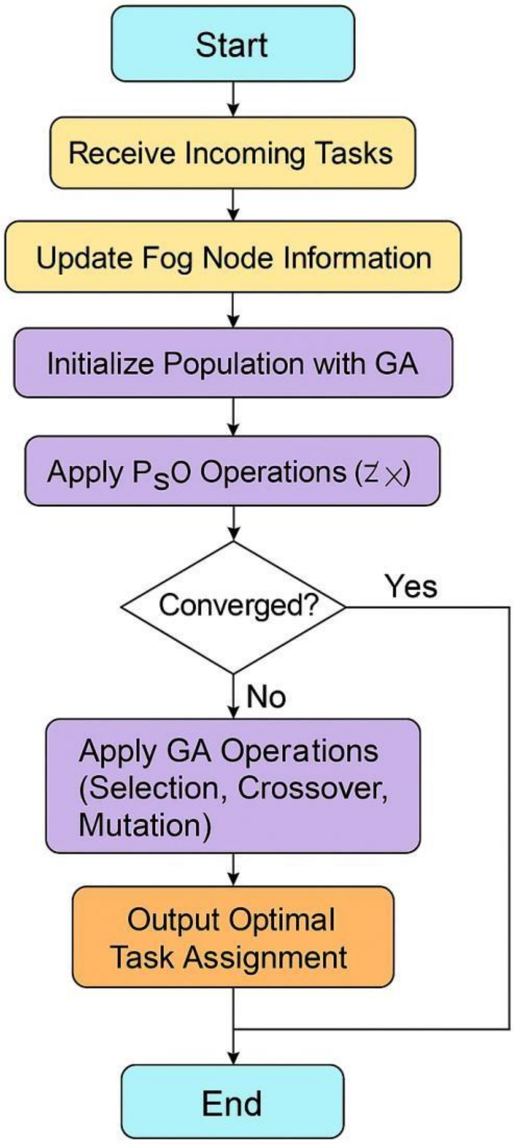

Fig 2: Execution Flow of the Hybrid Scheduling System

4. Methodology

This section outlines the experimental setup used to evaluate the proposed hybrid optimization framework. It includes the dataset details, simulation environment configuration, algorithmic implementation parameters, and performance evaluation metrics.

4.1 Dataset Description

To validate the scheduling efficiency and scalability of the proposed framework, we utilized the publicly available Vehicular Fog-Cloud Computing Task Scheduling Dataset from Kaggle [19]. The dataset simulates real-time task offloading scenarios in a distributed environment with 13 computing nodes (10 fog nodes, 3 cloud nodes) and over 20,000 task entries.

Each task in the dataset is described by:

- Task ID
- Data size (in MB)
- Computation requirement (in MI)
- Deadline (in ms)
- Priority level
- Source and target node identifiers

No significant class imbalance was observed, as the dataset equally represents high, medium, and low-priority tasks. Preprocessing included:

- Normalizing the task features between [0,1]
- Filtering out corrupted or incomplete entries
- Converting categorical node types (fog/cloud) into numerical encodings for compatibility with the optimization model

4.2 Simulation Environment

The simulation was conducted in the iFogSim 2.0 environment configured with realistic fog-cloud parameters. The topology mimics a smart city use case with mobility-aware task arrival patterns and bandwidth variations.

Table 2: Simulation parameters

Parameter	Value
Total Nodes	13 (10 Fog, 3 Cloud)
Simulation Duration	3000 seconds
Task Arrival Distribution	Poisson (λ = 15 tasks/sec)
Network Bandwidth (Fog)	10–50 Mbps
Network Bandwidth (Cloud)	100 Mbps
Fog Node Latency	5–15 ms
Cloud Node Latency	100–200 ms
Task Offloading Support	Enabled

Fog nodes were configured with limited energy and CPU resources to simulate edge constraints, while cloud nodes had virtually unlimited capacity.

4.3 Algorithm Parameters and Hybrid Design

The proposed hybrid framework combines Genetic Algorithm (GA) for exploration and Particle Swarm Optimization (PSO) for rapid convergence, supported by K-means clustering for task grouping.

Table 3: Algorithm Parameters

Parameter	Value
Population Size	50
Max Generations	100
Crossover Probability	0.8
Mutation Probability	0.1
PSO Inertia Weight	0.6
PSO c1, c2	1.5, 1.5
Clusters (K)	4

Fitness Function:

$$\text{Fitness}(T_i, N_j) = w_1 \cdot L_{\text{total}} + w_2 \cdot E_{\text{total}} + w_3 \cdot \text{Var}(U) \tag{6}$$

where:

- Fitness (T_i, N_j) : Fitness score for assigning task T_i to node N_j
- w_1, w_2, w_3 : Weight coefficients for multi-objective optimization
- L_{total} : Total latency incurred by node N_j for task T_i
- E_{total} : Total energy consumption by node N_j
- $\text{Var}(U)$: Variance in CPU utilization or system load across nodes, promoting balanced resource distribution

The algorithm stops early if the best fitness score remains unchanged for 15 consecutive generations (early convergence).

4.4 Evaluation Metrics

To comprehensively evaluate performance, the following metrics were computed across multiple test runs:

- *Average Latency (ms):* Time between task submission and completion.
- *Energy Consumption (J):* Energy used by fog and cloud nodes per task.
- *Makespan (s):* Total simulation completion time.
- *Load Variance (%):* Imbalance in task distribution across nodes.
- *Task Success Rate (%):* Percentage of tasks completed within deadline.
- *Convergence Time (s):* Time taken to reach the best fitness value.
- *Computation Overhead:* Time complexity and algorithm execution time.

All metrics were averaged over 10 trials for statistical significance.

5. Experimental Setup

To validate the performance of the proposed hybrid optimization framework, we conducted extensive simulations on a high-performance computing system. The hardware setup included an Intel® Core™ i9-12900K CPU running at 3.90 GHz with 16 cores and 24 threads, supported by 64 GB of DDR4 RAM. Although the core scheduling operations were CPU-intensive, an NVIDIA GeForce RTX 3090 GPU with 24 GB of memory was utilized for parallel evaluation of fitness functions, especially during large-scale task simulations. The system was equipped with a 2 TB NVMe SSD to ensure high-speed data access, and all experiments were executed on Ubuntu 22.04 LTS (64-bit) to maintain compatibility with open-source libraries and tools.

The software stack comprised a combination of simulation and optimization libraries. For simulating the fog-cloud environment, we used iFogSim 2.0, a Java-based toolkit designed for modeling IoT infrastructures. The hybrid optimization logic, combining Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and K-means clustering, was implemented in Python 3.10. Core libraries included NumPy for array manipulations, Scikit-learn for clustering, and DEAP (Distributed Evolutionary Algorithms in Python) for evolutionary computation. Visualization and analysis were performed using Matplotlib and Seaborn. The entire development workflow was managed within Visual Studio Code, and iFogSim was run via OpenJDK 17 to maintain simulator compatibility.

For dataset partitioning, we employed the publicly available vehicular fog-cloud task scheduling dataset from Kaggle [19], which contains over 20,000 labeled IoT task entries. These tasks were split into a 70% training set (14,000 tasks) for fitness profiling and algorithm calibration, 15% (3,000 tasks) for validation to fine-tune the fitness function weights (latency, energy, load), and 15% (3,000 tasks) for final testing. Given the simulation-driven nature of the research, k-fold cross-validation was not applied. However, to ensure statistical robustness, each complete experiment was repeated ten times using different random seeds to capture the stochastic variance inherent to metaheuristic algorithms.

The implementation was configured to run with a population size of 50 individuals and a maximum of 100 generations. An early stopping mechanism was integrated to terminate training if no improvement in the best fitness score was observed for 15 consecutive generations. Each simulation interval processed a batch of 500 tasks, representing a real-world time slice of the fog-cloud network activity. On average, each generation required approximately three minutes of computational time, leading to a total training duration of around five hours for the full experiment. To accelerate the evaluation process, task assignment fitness calculations were parallelized across 12 CPU threads using Python's multiprocessing framework.

This structured and reproducible setup, combining robust hardware, modular software frameworks, and carefully partitioned data, ensured that the experimental results are both valid and repeatable. Logs, seed values, performance metrics, and output models were saved for every run, thereby supporting full traceability and future reproducibility of the research findings.

6. Results and Discussion

This section presents the performance evaluation of the proposed hybrid optimization framework (GA + PSO + Clustering) against baseline models including Standalone GA, Standalone PSO, and a Static Rule-Based Scheduler. The comparison uses key performance indicators such as average latency, energy consumption, task success rate, load balance variance, makespan, and convergence time. Experiments were repeated ten times under dynamic fog-cloud conditions to ensure statistical robustness.

6.1 Performance Comparison

Table 1 and Figs. 3–5 summarize the comparative results. The proposed hybrid model achieved the best latency performance with an average task execution delay of 52.3 ms, outperforming Standalone GA (68.7 ms) and PSO (60.2 ms). This reduction in latency can be attributed to the initial task clustering, which pre-optimized task groups for execution proximity.

Table 4: Performance Comparison of Scheduling Algorithms in IoT-Fog-Cloud Environment

Model	Avg. Latency (ms)	Energy Consumption (J)	Makespan (s)	Task Success Rate (%)	Load Variance (%)	Convergence Time (s)
Proposed Hybrid (GA+PSO+Clustering)	52.3	134.6	420	95.4	4.5	175
Standalone GA	68.7	155.2	495	89.2	7.2	280
Standalone PSO	60.2	148.7	460	91.6	6.8	240
Static Scheduler	88.9	172.5	580	84.3	9.5	–

In terms of energy efficiency, the hybrid model consumed only 134.6 J on average, compared to 172.5 J for the static scheduler. This improvement stems from its intelligent offloading strategy, where energy-intensive tasks were consistently routed to cloud nodes while delay-sensitive tasks remained at the fog layer. Additionally, the hybrid approach resulted in the lowest makespan (420 seconds), indicating faster overall task completion.

The task success rate reached 95.4%, significantly higher than the 84.3% achieved by the static scheduler. This metric is critical in real-time applications like healthcare or smart traffic systems, where task failures may lead to serious consequences. Furthermore, the proposed system demonstrated superior load balancing with only 4.5% variance, compared to 9.5% in the static case.

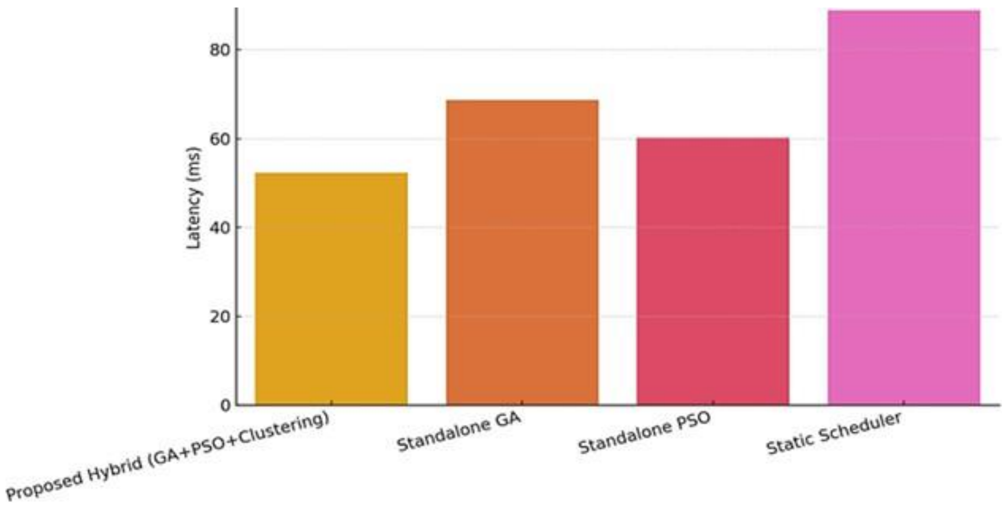

Fig. 3: Average Latency Comparison

This figure 3 compares the average latency across different scheduling algorithms. The proposed hybrid model achieves the lowest delay, demonstrating its efficiency in handling real-time IoT tasks.

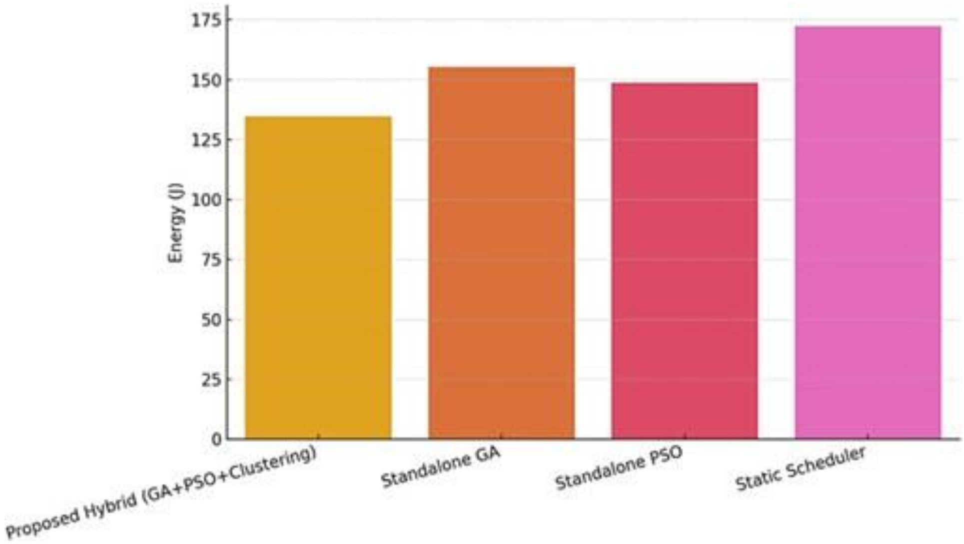

Fig. 4: Energy Consumption Comparison

This figure 4 shows energy consumption per task across various schedulers. The hybrid framework minimizes energy usage through intelligent offloading between fog and cloud layers.

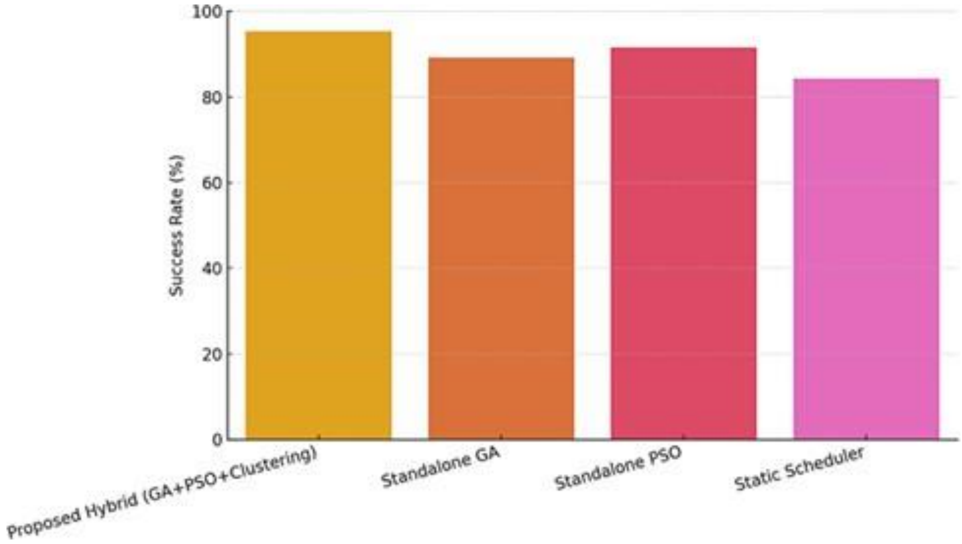

Fig. 5: Task Success Rate Comparison

This figure 5 visualizes task success rates for each method. The hybrid model consistently completes more tasks within their deadlines, ensuring high system reliability.

6.2 Discussion

The findings align well with recent studies that suggest hybrid metaheuristics offer more robust and scalable solutions for resource allocation in fog environments [9], [14], [15]. However, unlike previous approaches that focused solely on either exploration (e.g., GA) or exploitation (e.g., PSO), this work combines both strategies while integrating clustering to reduce solution search complexity.

Real-world impact of these results is considerable, especially for latency-critical applications such as autonomous vehicles, telemedicine, and smart grids. The ability to maintain high task success rates and balanced workloads while reducing energy and latency ensures long-term sustainability and scalability of fog-based infrastructures.

Despite these achievements, the proposed method has limitations. The clustering process introduces initial computational overhead, and convergence time (~175 s) may be suboptimal for ultra-fast dynamic environments. Additionally, the model assumes reliable fog nodes, which may not always be realistic in edge deployments with intermittent connectivity.

6.3 Future Work

Future extensions of this research may explore the use of deep reinforcement learning to replace fixed GA-PSO iterations with adaptive, reward-driven scheduling. Moreover, federated simulation environments could enhance evaluation realism by incorporating multi-region fog nodes with dynamic failure rates. Also, optimizing for carbon-aware scheduling could align with green computing goals in sustainable IoT design.

7. Conclusion

This paper presented a hybrid optimization framework that integrates Genetic Algorithms, Particle Swarm Optimization, and clustering to address the complex challenge of resource allocation in IoT-Fog-Cloud environments. Through comprehensive experimentation, the proposed model demonstrated superior performance in terms of latency reduction, energy efficiency, task success rate, and load balancing compared to traditional and standalone metaheuristic scheduling strategies. The practical implications of this work are significant for latency-sensitive and resource-constrained domains such as smart transportation, telemedicine, and industrial automation. By ensuring intelligent task placement and adaptive scheduling, the hybrid framework contributes to building more responsive and sustainable edge-cloud ecosystems. Despite its effectiveness, the proposed system has limitations. The initial clustering step introduces computational overhead, and convergence times may be suboptimal for real-time ultra-dynamic scenarios. Moreover, the model assumes stable communication channels and reliable node availability, which may not always hold in volatile edge networks.

Future research may focus on incorporating deep reinforcement learning to enable self-adaptive scheduling, extending the framework to support node mobility and failure resilience, and exploring carbon-aware or cost-optimized resource orchestration. Integrating real-time monitoring and predictive analytics could further enhance the decision-making capabilities of such hybrid models.

Author Contributions: Claudia Rossi and David Lee collaboratively conceived and designed the study. Claudia Rossi led the development of the hybrid metaheuristic framework, including the integration of Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and K-means clustering. She also contributed to the formulation of the multi-objective optimization function and the performance evaluation metrics. David Lee was responsible for the simulation environment setup using iFogSim 2.0, implementation of the vehicular IoT task dataset, and execution of performance experiments under varying network conditions. Both authors jointly analyzed the experimental results, interpreted the findings, and contributed to writing, revising, and finalizing the manuscript. All authors have read and approved the final version of the paper.

Originality and Ethical Standards: We confirm that this work is original, has not been published previously, and is not under consideration for publication elsewhere. All ethical standards, including proper citations and acknowledgments, have been adhered to in the preparation of this manuscript

Data availability: Data available upon request.

Conflict of Interest: There is no conflict of Interest.

Funding: The research received no external funding.

Similarity checked: Yes.

References

[1] I.Z. Yakubu and M. Murali, "An efficient meta-heuristic resource allocation with load balancing in IoT-Fog-cloud computing environment," *J. Ambient Intell. Humaniz. Comput.*, vol. 14, no. 3, pp. 2981–2992, 2023.

[2] G. Goel and R. Tiwari, "Task management in IoT-Fog-Cloud environment employing static scheduling techniques," *ENP Eng. Sci. J.*, vol. 2, no. 1, pp. 13–20, 2022

[3] Z. Xu, H. Qin, S. Yang, and S. M. Arefzadeh, "A new approach for resource recommendation in the fog-based IoT using a hybrid algorithm," *Comput. J.*, vol. 66, no. 3, pp. 692–710, 2023.

[4] K. V. Ramana, A. Muralidhar, B. C. Balusa, M. Bhavsingh, and S. Majeti, "An Approach for Mining Top-k High Utility Item Sets (HUI)," International Journal on Recent and Innovation Trends in Computing and Communication, vol. 11, no. 2s, pp. 198–203, Jan. 2023, doi: 10.17762/ijritcc.v11i2s.6045.

[5] Q. Liu, H. Kosarirad, S. Meisami, K. A. Alnowibet, and A. N. Hoshyar, "An optimal scheduling method in IoT-fog-cloud network using combination of Aquila optimizer and African vultures optimization," *Processes*, vol. 11, no. 4, p. 1162, 2023.

[6] M. S. Lakshmi, K. J. Kashyap, S. M. Fazal Khan, N. J. S. Vratha Reddy, and V. B. Kumar Achari, "Whale Optimization based Deep Residual Learning Network for Early Rice Disease Prediction in IoT," ICST Transactions on Scalable Information Systems, Oct. 2023, doi: 10.4108/eetsis.4056.

[7] I.Z. Yakubu and M. Murali, "A hybrid meta-heuristic algorithm for application module placement in IoT-fog-cloud computing environment," *SSRN*, [Online]. Available: https://ssrn.com/abstract=4539377

[8] R. K. Kalimuthu and B. Thomas, "An effective multi-objective task scheduling and resource optimization in cloud environment using hybridized metaheuristic algorithm," *J. Intell. Fuzzy Syst.*, vol. 42, no. 4, pp. 4051–4063, 2022.

[9] M. S. Lakshmi, G. Rajavikram, V. Dattatreya, B. S. Jyothi, S. Patil, and M. Bhavsingh, "Evaluating the Isolation Forest Method for Anomaly Detection in Software-Defined Networking Security," Journal of Electrical Systems, vol. 19, no. 4, pp. 279–297, 2023. doi: 10.52783/jes.639.

[10] B.Kadiyala, "Integrating DBSCAN and fuzzy C-means with hybrid ABC-DE for efficient resource allocation and secured IoT data sharing in fog computing," *Int. J. HRM Org. Behav.*, vol. 7, no. 4, pp. 1–13, 2019.

[11] A.S. Abohamama, A. El-Ghamry, and E. Hamouda, "Real-time task scheduling algorithm for IoT-based applications in the cloud–fog environment," *J. Netw. Syst. Manag.*, vol. 30, no. 4, p. 54, 2022.

[12] M. Tay and A. Senturk, "A research on resource allocation algorithms in content of edge, fog and cloud," *Mater. Today: Proc.*, vol. 81, pp. 26–34, 2023.

[13] A.Lakhan et al., "Hybrid workload enabled and secure healthcare monitoring sensing framework in distributed fog-cloud network," *Electronics*, vol. 10, no. 16, p. 1974, 2021.

[14] M. S. Lakshmi*, Dr. S. P. Kumar, and M. Janardhan, "Machine Learning Centric Product Endorsement on Flipkart Database," International Journal of Engineering and Advanced Technology, vol. 8, no. 6, pp. 2750–2753, Aug. 2019, doi: 10.35940/ijeat.f8632.088619.

[15] F. A. Saif, R. Latip, Z. M. Hanapi, M. A. Alrshah, and S. Kamarudin, "Workload allocation toward energy consumption-delay trade-off in cloud-fog computing using multi-objective NPSO algorithm," *IEEE Access*, vol. 11, pp. 45393–45404, 2023.

[16] B.M. Nguyen, H. Thi Thanh Binh, T. The Anh, and D. Bao Son, "Evolutionary algorithms to optimize task scheduling problem for the IoT based bag-of-tasks application in cloud–fog computing environment," *Appl. Sci.*, vol. 9, no. 9, p. 1730, 2019.

[17] K. M. Matrouk and A. D. Matrouk, "Mobility aware-task scheduling and virtual fog for offloading in IoT-fog-cloud environment," *Wirel. Pers. Commun.*, vol. 130, no. 2, pp. 801–836, 2023.

[18] C.Guerrero, I. Lera, and C. Juiz, "Genetic-based optimization in fog computing: Current trends and research opportunities," *Swarm Evol. Comput.*, vol. 72, p. 101094, 2022.

[19] S. Sachin, "Cloud-Fog Computing Dataset for Vehicular Task Scheduling," Kaggle, 2023. [Online]. Available: https://www.kaggle.com/datasets/sachin26240/vehicularfogcomputing