A Multi-Objective Archimedes Optimizer for VM Selection in Healthcare Hybrid Cloud-IoT Systems

Ahmed Yosreddin Samti[1], Ines Ben Jaafar[1], and Issam Nouaouri[2]

***Abstract*— Healthcare applications deployed over hybrid Cloud-IoT infrastructures must satisfy strict latency and reliability requirements while remaining energy- and cost-efficient. In such environments, virtual machine (VM) selection becomes a challenging multi-objective optimization problem involving conflicting criteria. This paper formulates healthcare-aware VM selection as a discrete multi-objective task-to-VM assignment problem and proposes MO-AOA, a multi-objective extension of the Archimedes Optimization Algorithm. The proposed method combines buoyancy-inspired search dynamics with Pareto dominance ranking, crowding-distance diversity preservation, an external archive of non-dominated solutions, and a repair mechanism for feasible discrete allocations. MO-AOA is evaluated using both CloudSim Plus simulations and real workload traces derived from the Google Borg dataset, and is compared with NSGA-II, PSO, and ACO. Experimental results show that MO-AOA achieves up to 18% latency reduction, 22% energy savings, and 25% higher SLA compliance, while also improving Pareto-front quality. These results demonstrate that MO-AOA is an effective optimization framework for healthcare-aware resource orchestration in hybrid Cloud-IoT systems.**

Keywords: Multi-objective optimization, Archimedes Optimization Algorithm, VM selection, hybrid Cloud-IoT, healthcare computing, SLA-aware scheduling.

I. INTRODUCTION

The integration of the Internet of Things (IoT) and Cloud Computing has profoundly transformed modern healthcare, enabling large-scale intelligent data analysis for applications such as patient monitoring, diagnostics, and real-time decision-making. This convergence has facilitated the emergence of connected, data-driven healthcare ecosystems that increasingly rely on efficient computational resource management and intelligent workload distribution.

For instance, Fakhouri et al. [1] proposed a multi-objective task scheduling model for IoT systems operating in the edge-cloud continuum, with particular emphasis on energy efficiency and real-time responsiveness. Similarly, Pournazari et al. [2] introduced a multi-objective optimization model for energy-centric offloading in fog computing, demonstrating that hybrid architectures can contribute to reducing both latency and energy consumption for healthcare IoT workloads. Collectively, these studies suggest that the coordinated integration of cloud and IoT layers plays a key role in improving performance and sustainability in modern healthcare systems.

Within this context, Virtual Machine (VM) selection and task scheduling emerge as fundamental challenges in cloud-IoT environments.

The problem is inherently multi-objective, involving the simultaneous consideration of cost, latency, energy consumption, and Service Level Agreement (SLA) compliance.
Recent works, such as Qasim et al. [3], explored improved genetic algorithms combined with multi-criteria decision-making (MCDM) techniques for Industrial IoT workloads, reporting improved trade-offs in resource utilization and processing time. Likewise, Maldonado Carrascosa and Seddiki [4] investigated multi-objective optimization of virtual machine migration to enhance performance and reduce operational costs in heterogeneous cloud systems. In this work, VM selection is formulated as a task-to-VM assignment problem, where each candidate solution encodes a mapping of healthcare tasks to available virtual machines. While classical metaheuristics such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO) have demonstrated effectiveness in addressing various cloud optimization problems, prior studies indicate that their performance may degrade when dealing with highly dynamic, large-scale cloud-IoT workloads. In particular, maintaining an appropriate balance between exploration and exploitation remains challenging under conflicting objectives and rapidly changing system conditions.
As a result, recent research has increasingly explored physics-inspired optimization algorithms, which model natural physical processes to achieve more adaptive and stable search behaviors.

Motivated by these observations, this paper proposes MO-AOA, a multi-objective extension of AOA for healthcare-aware VM selection in hybrid Cloud-IoT systems. The proposed approach makes four main contributions:

- A discrete multi-objective task-to-VM assignment formulation considering cost, latency, energy, and SLA violation.
- A multi-objective AOA design using Pareto ranking, crowding distance, external archive management, and a repair operator.
- Validation in both CloudSim Plus and Google Borg-derived settings to cover controlled and realistic workloads.
- Comparative evidence against NSGA-II, PSO, and ACO using objective metrics and Pareto-front quality indicators.

[1]SMART Lab, University of Tunis, Tunis, Tunisia.
[2]LGI2A, Universite d'Artois, Arras, France.

The remainder of this paper is organized as follows. Section 2 formulates the VM selection problem and introduces the proposed MO-AOA framework. Section 3 describes the experimental setup. Section 4 presents and discusses the results. Section 5 concludes the paper.

II. PROBLEM FORMULATION AND PROPOSED MO-AOA

A. Problem Formulation

We define the VM selection problem as minimizing latency and energy while maximizing SLA compliance and minimizing cost. Formally, this can be modeled as a multi-objective optimization problem. Let $T = \{t_1, \ldots, t_n\}$ be the set of healthcare tasks and $V = \{v_1, \ldots, v_m\}$ be the set of available VMs. A candidate solution maps each task t_i to a VM v_j. The optimization goal is:

$$\min F = (f_1, f_2, f_3, f_4) \tag{1}$$

where f_1 is execution cost, f_2 is end-to-end latency, f_3 is energy consumption, and $f_4 = 1 -$ SLA compliance is SLA violation. Resource-capacity and deadline constraints make the problem discrete and constrained.

This formulation is well suited to healthcare workloads, where latency reflects response time for patient-critical services, energy affects infrastructure sustainability, cost impacts cloud operation, and SLA violation captures service reliability.

B. Multi-Objective Archimedes Optimization

The Archimedes Optimization Algorithm models each candidate solution as an object immersed in a fluid. The movement of each solution is driven by buoyancy-related variables such as density, volume, and acceleration. At iteration tt, the buoyant acceleration of solution ii is expressed t is:

$$A_i^{(t)} = g \cdot \frac{\rho_f - \rho_i^{(t)}}{\rho_f} + \lambda^{(t)} \cdot V_i^{(t)} \tag{2}$$

where g is the gravitational constant, ρ_f is fluid density, $\rho_i^{(t)}$ is object density, $V_i^{(t)}$ is object volume, and $\lambda^{(t)}$ is an adaptive control parameter.

To support multi-objective, discrete VM allocation, MO-AOA introduces:

- Pareto dominance ranking for non-dominated sorting.
- Crowding distance for diversity preservation.
- An external archive of non-dominated solutions.
- A repair mechanism that converts continuous updates into feasible discrete assignments and fixes constraint violations.

The optimization process begins with a random population of feasible allocations. At each iteration, candidate solutions are updated through buoyancy-driven dynamics, discretized, repaired if necessary, and re-evaluated according to the four objectives.

The population is then ranked using non-dominated sorting and crowding distance, while the external archive is updated with the best trade-off solutions. Termination occurs either after a maximum number of iterations or when Pareto-front improvement becomes negligible.

By embedding physical dynamics within a multi-objective optimization framework, MO-AOA achieves a natural balance between convergence precision and solution diversity. Its adaptive buoyancy mechanism enables continuous refinement of VM allocations while preserving responsiveness to changing workload conditions.

This combination of physical modeling, Pareto-based optimization, and domain-aware prioritization makes MO-AOA an efficient and scalable approach for intelligent resource allocation in hybrid Cloud-IoT environments supporting real-time medical applications.

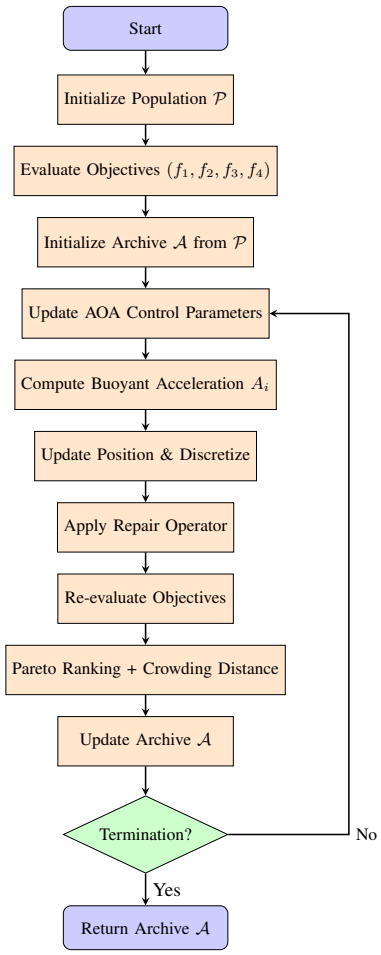

Fig. 1. Flow diagram of the MO-AOA optimization process.

The proposed approach is structured into four key phases, each of which is detailed below.

C. Algorithm Pseudocode

Algorithm 1 MO-AOA for Multi-Objective VM Selection

Require: Tasks $\mathcal{T} = \{t_1, \ldots, t_n\}$ with resource demand and deadline; VM set $\mathcal{V} = \{v_1, \ldots, v_m\}$ with capacity and power model; Objectives $\{f_1, f_2, f_3, f_4\}$; Population size N, max iterations $Iter_{\max}$; AOA parameters g, λ; Archive size $A_{\max}$; HV threshold ε_{HV} and window W.

Ensure: External archive $\mathcal{A}$ (Pareto-front approximation).

— Step 1: Initialization —

Initialize population $\mathcal{P} = \{X^1, \ldots, X^N\}$ with $X^k = [x_1^k, \ldots, x_n^k]$, $x_i^k \sim \mathcal{U}\{1, \ldots, m\}$.
Initialize AOA state variables (ρ, Vol, Acc) for each solution.
Apply REPAIR(X^k) to each X^k to ensure feasibility.
Evaluate objectives f_1, f_2, f_3, f_4 for each X^k.
Initialize archive $\mathcal{A}$ with non-dominated solutions from $\mathcal{P}$.

— Step 2: Main Optimization Loop —

for $iter = 1$ **to** $Iter_{\max}$ **do**
  Update AOA control parameters.
  for each $X^k \in \mathcal{P}$ **do**
    Compute acceleration Acc^k using Eq. (2).
    Update continuous position: $\tilde{X}^k \leftarrow X^k + r \cdot Acc^k$, $r \sim \mathcal{U}(0, 1)$.
    $x_i \leftarrow \max(1, \min(m, \text{round}(\tilde{x}_i)))$ ▷ discretize and bound
    Apply REPAIR(X^k) for capacity/deadline feasibility.
    Evaluate objectives for repaired X^k.
  end for
  Merge $\mathcal{P}$ into archive $\mathcal{A}$; remove dominated solutions.
  if $|\mathcal{A}| > A_{\max}$ **then** prune by crowding distance.
  end if
  Select next generation $\mathcal{P}$ via non-dominated sorting and crowding distance.
  if HV improvement over last W iterations $< \varepsilon_{\text{HV}}$ **then break**
  end if
end for
return $\mathcal{A}$

function REPAIR(X)
  for each VM v_j **do** compute total assigned load (CPU, RAM, bandwidth).
  end for
  while any VM violates capacity **do**
    Identify task t_i contributing most to violation.
    Reassign t_i to feasible VM $v_{j'}$ minimizing $\text{Penalty}(j') = \alpha\,\Delta\text{Latency} + \beta\,\Delta\text{Cost} + \gamma\,\Delta\text{Energy}$.
  end while
  if excessive deadline violations remain **then** reassign tasks with largest lateness to faster feasible VMs.
  end if
  return repaired X.
end function

D. Algorithmic Flow

The complete process of MO-AOA can be summarized as follows:

1) **Initialization:** Generate the population of task-to-VM mappings.
2) **Evaluation:** Compute cost, latency, energy, and SLA for each mapping.
3) **Buoyancy-Based Update:** Update densities, volumes, and positions based on Archimedes' law.
4) **Pareto Ranking:** Perform non-dominated sorting to form the current Pareto front.
5) **Crowding Distance:** Maintain diversity along the Pareto surface.
6) **Archiving:** Store non-dominated solutions in an external archive.
7) **Termination:** After reaching the maximum iteration, return $\mathcal{P}^*$ and select the healthcare-prioritized best configuration.

III. EXPERIMENTAL SETUP

The proposed MO-AOA is evaluated in two complementary settings in order to assess both controlled performance and robustness under realistic workloads. The first setting uses CloudSim Plus to simulate a three-layer healthcare hybrid Cloud-IoT architecture composed of IoT devices, edge/fog nodes, and cloud data centers. The cloud layer hosts 100 heterogeneous VMs divided into small, medium, and large configurations with different CPU, RAM, bandwidth, storage, and power characteristics.

Healthcare workloads were generated using both synthetic workloads in CloudSim Plus and real traces derived from Google Borg cluster logs. Each task represents a healthcare processing job such as patient monitoring data analysis or diagnostic computation.

Parameter	Distribution / Value
Number of tasks	5 000–20 000
CPU demand	200–2 000 MIPS
Task length	1 000–10 000 instructions
Arrival process	Poisson distribution
Deadline	200–500 ms
Task priority	Uniform distribution

TABLE I

WORKLOAD PARAMETERS.

This configuration simulates heterogeneous healthcare workloads including real-time monitoring and batch medical analytics.

The second setting uses a Google Borg-derived workload dataset, where cloud tasks are mapped to healthcare-like processing jobs such as patient monitoring analytics or diagnostic workloads.

Since the original traces do not explicitly provide energy, cost, and SLA values, these metrics are derived using standard power, pricing, and deadline-compliance models. Latency is computed from submission and completion times, cost is estimated from CPU usage, energy is estimated from execution duration and utilization, and a task is considered SLA-compliant if it completes successfully within 500 ms.
For comparative evaluation, MO-AOA is benchmarked against NSGA-II, PSO, and ACO using identical population sizes and iteration limits, as reported in the full manuscript. All algorithms are executed over multiple independent runs to reduce stochastic bias.
The performance metrics are latency, energy consumption, SLA compliance, hypervolume (HV), and inverted generational distance (IGD). These metrics jointly quantify execution quality, reliability, and Pareto-front performance.

IV. RESULTS AND DISCUSSION

Table 2 summarizes the main results obtained in the simulation-based and Borg-derived settings. In both environments, MO-AOA provides the best overall trade-off among latency, energy, SLA compliance, and Pareto-front quality.

A. Comparative Performance

In CloudSim Plus, MO-AOA achieves an average latency of 210 ms, compared with 254 ms, 272 ms, and 290 ms for NSGA-II, PSO, and ACO, respectively. Similarly, energy consumption is reduced to 7.8 kWh, compared with 9.3 kWh, 9.0 kWh, and 10.2 kWh for the baseline algorithms.

SLA compliance reaches 98.3%, outperforming NSGA-II (93.1%), PSO (91.8%), and ACO (89.6%). The hypervolume (HV) metric improves to 0.91, compared with 0.76, 0.73, and 0.68 for NSGA-II, PSO, and ACO. The table 2 focuses on the operational efficiency of the algorithms, specifically how they handle time, power, and service level agreements.

TABLE II
SYSTEM PERFORMANCE METRICS

Algorithm	Latency (ms)	Energy (kWh)	SLA (%)
MO-AOA	210±12 / 228±14	7.8±0.4 / 8.1±0.5	98.3±1.1
PSO	272±18 / 278±20	9.0±0.6 / 9.3±0.6	91.8±2.0
NSGA-II	254±16 / 261±17	9.3±0.5 / 9.5±0.6	93.1±1.8
ACO	290±21 / 296±23	10.2±0.7 / 10.4±0.8	89.6±2.3

Latency & Energy: Lower values are better. MO-AOA significantly outperforms the others, showing the lowest delay and power consumption.

SLA (Service Level Agreement): Higher values are better. This represents the reliability of the system. Again, MO-AOA maintains the highest compliance (above 97%). The table 3 focuses on the quality of the Pareto-optimal solutions found by each algorithm using standard mathematical benchmarks.

TABLE III
OPTIMALITY QUALITY (HV AND IGD)

Algorithm	Hypervolume (HV) ↑	IGD ↓
MO-AOA	0.91±0.02 / 0.89±0.03	0.0031±0.0004
PSO	0.73±0.05 / 0.71±0.04	0.0110±0.0013
NSGA-II	0.76±0.04 / 0.75±0.04	0.0094±0.0011
ACO	0.68±0.05 / 0.66±0.05	0.0120±0.0140

Hypervolume (HV): Measures the diversity and convergence of the solution set. A higher value (closer to 1.0) indicates a better spread of solutions. MO-AOA (0.91) shows the best performance.

Inverted Generational Distance (IGD): Measures how far the results are from the true optimal "front." A lower value is better. MO-AOA has the lowest IGD, suggesting it is the most accurate algorithm in this set.

B. Pareto-Front Quality and Convergence

The superiority of MO-AOA is not limited to single-metric averages. The high HV and low IGD values indicate that MO-AOA generates a Pareto front that is both well distributed and closer to the reference optimum than those produced by NSGA-II, PSO, and ACO. This suggests that the proposed combination of buoyancy-based search, Pareto ranking, and archive management improves both convergence quality and solution diversity.
Figure 2 illustrates the comparative Pareto front quality using Hypervolume (HV) and Inverted Generational Distance (IGD). MO-AOA attains the highest HV (0.91) and the lowest IGD (0.0031), signifying a well-distributed and converged Pareto front.

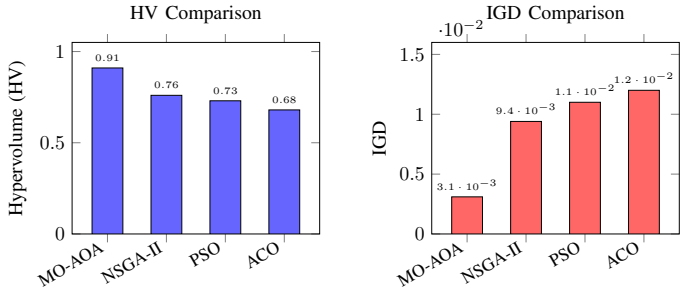

Fig. 2. Pareto front quality indicators: HV and IGD comparison.

Another important observation is the faster convergence of MO-AOA. According to the reported convergence curves, MO-AOA in Figure 3 clearly indicate within 80–100 iterations, while NSGA-II and PSO require more iterations to approach similar fitness levels.

This faster stabilization is particularly desirable in healthcare Cloud-IoT systems, where resource allocation must remain responsive under dynamic workload changes.

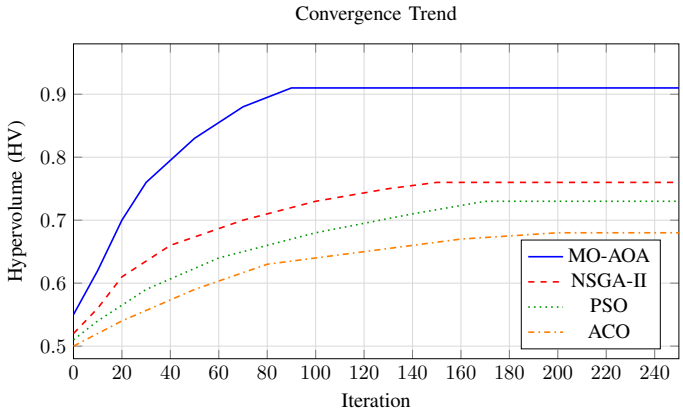

Fig. 3. Convergence trend: objective minimization over iterations.

Collectively, the simulation and empirical data validate MO-AOA as a high-performance framework capable of navigating the complex trade-offs inherent in sustainable healthcare computing.

The stability of the algorithm in both real and simulated workloads underscores its potential to reduce the carbon footprint of hospital data centers while maintaining strict quality-of-service standards for clinicians. With these performance results established, the subsequent discussion contextualizes these improvements within the broader landscape of Cloud-IoT research and explores the broader clinical implications of the study.

C. Discussion

The superiority of MO-AOA is not limited to single-metric averages. The high HV and low IGD values indicate that MO-AOA generates a Pareto front that is both well distributed and closer to the reference optimum than those produced by NSGA-II, PSO, and ACO. This suggests that the proposed combination of buoyancy-based search, Pareto ranking, and archive management improves both convergence quality and solution diversity.

Another important observation is the faster convergence of MO-AOA. According to the reported convergence curves, MO-AOA generally stabilizes within 80–100 iterations, while NSGA-II and PSO require more iterations to approach similar fitness levels. This faster stabilization is particularly desirable in healthcare Cloud-IoT systems, where resource allocation must remain responsive under dynamic workload changes.

In summary, the empirical results and visual analyses demonstrate that MO-AOA transcends theoretical optimization. By delivering shorter, more consistent response times and enhancing the quality of service, MO-AOA offers tangible operational benefits for tele-ICU and healthcare Cloud-IoT ecosystems, ultimately fostering more sustainable hospital cloud infrastructures.

V. CONCLUSION

This paper proposed MO-AOA, a multi-objective Archimedes-based optimization framework for VM selection in healthcare hybrid Cloud-IoT systems. The method formulates VM selection as a discrete multi-objective problem and combines buoyancy-driven search with Pareto ranking, crowding distance, an external archive, and a repair mechanism. Evaluations performed with both CloudSim Plus and Borg-derived traces show that MO-AOA consistently outperforms NSGA-II, PSO, and ACO in terms of latency, energy efficiency, SLA compliance, and Pareto-front quality.

These results suggest that MO-AOA is a promising solution for intelligent resource orchestration in healthcare-aware cloud infrastructures. In future work, the framework can be extended with predictive workload models and tested in real hospital cloud environments.

REFERENCES

[1] H. Fakhouri, F. Alkhabbas, and S. Alawadi, "Multi-objective optimisation for energy-centric offloading in fog computing" *IEEE Access*, 2025.

[2] J. Pournazari and A. Ullah and A. Al-Dubai and others "An optimized multi-objective task scheduling approach for IoT systems in the edge-cloud continuum," *Computing*, 2025.

[3] M. Qasim and M. Sajid and M. Lapina and M. Shahid "COBGA: MCDM-assisted improved genetic algorithm for scheduling Industrial Internet of Things jobs in cloud computing" *Cluster Computing*, 2025.

[4] M. Qasim and M. Sajid and M. Lapina and M. Shahid "COBGA: MCDM-assisted improved genetic algorithm for scheduling Industrial Internet of Things jobs in cloud computing" *Cluster Computing*, 2025.

[5] F. J. Maldonado Carrascosa and D. Seddiki and A. Jiménez Sánchez, "Multi-objective optimization of virtual machine migration among cloud data centers," *Soft Computing*, 2024.

[6] N. Khaledian, S. Razzaghzadeh, S. Moazzami, and P. N. Kivi, "TM-MO-AOA: A two-stage task scheduling approach using TOPSIS and multi-objective Archimedes optimization in fog-cloud environments," *Computing*, 2025.

[7] S. Kushwaha, R. S. Singh, and K. Prajapati, "Multi-objective workflow scheduling in cloud using Archimedes optimization algorithm," *Concurrency and Computation: Practice and Experience*, 2025.

[8] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[9] A. Y. Samti, I. Ben Jaafar, and I. Nouaouri, "A novel NSGA-III-GKM++ framework for multi-objective cloud resource brokerage optimization," *Mathematics*, 2025.

[10] F. A. Hashim *et al.*, "Archimedes optimization algorithm: A new metaheuristic algorithm for solving optimization problems," *Applied Intelligence*, vol. 51, pp. 1531–1551, 2021.

[11] L. Abualigah and A. Diabat, "Archimedes optimizer: Theory, analysis, improvements, and applications," *Archives of Computational Methods in Engineering*, 2023.

[12] M. C. Silva Filho *et al.*, "CloudSimPlus: A cloud computing simulation framework pursuing software engineering principles for improved modularity, extensibility, and correctness," in *Proc. IFIP/IEEE IM*, 2017, pp. 400–406.

[13] A. Verma *et al.*, "Large-scale cluster management at Google with Borg," in *Proc. EuroSys*, 2015.
[14] M. H. Sayadnavard, A. T. Haghighat, and A. M. Rahmani, "Adaptive multi-objective virtual machine consolidation for energy-efficient and SLA-aware cloud data centers," *Journal of Grid Computing*, 2025.