Online-Journals.org

iJIM International Journal of Interactive Mobile Technologies

iJIM | eISSN: 1865-7923 | Vol. 20 No. 15 (2026) | OPEN ACCESS

https://doi.org/10.3991/ijim.v20i15.61457

PAPER

Benchmarking the Pareto Frontier of VM Placement: A Multi-Objective Evaluation of Heuristics and Metaheuristics Algorithms

Wijdane Lakhouari (✉), **Mohamed El Ghmary**, **Hassan Echoukairi**

Mohammed V University in Rabat, Rabat, Morocco

wijdane.lakhouari@um5r.ac.ma

ABSTRACT

Virtual machine (VM) placement in recent cloud systems remains a challenging and new task due to the challenge of finding the best way to use resources efficiently while also ensuring service reliability. The current study offers an in-depth comparative evaluation of six VM placement algorithms—three established heuristics (Best Fit Decreasing (BFD), First Fit Decreasing (FFD), and Worst Fit (WF)), a baseline RANDOM method, and two metaheuristics (Ant Colony Optimization (ACO) and NSGA-III)—conducted through 2,400 simulation trials. We meticulously evaluate each algorithm across four critical performance metrics: placement success rate, energy consumption, execution time, and service reliability. The statistical analysis confirmed significant differences in performance across all methods using the Kruskal-Wallis and Mann-Whitney tests. Classical heuristics achieve a placement success rate (PSR) of 87.38% with BFD and FFD operating without any service violations. These methods use more energy than other approaches, which is a drawback of their good performance. NSGA-III reduces energy usage by 11% while maintaining acceptable placement performance (71.62%). The execution time (ET) can really vary, from super-fast heuristics to more time-consuming optimization methods. The results indicate that classical heuristics provide the highest reliability. No SLA violations (SLAV) were observed in the scenarios evaluated. Classical heuristics, in particular BFD and FFD, are therefore the most reliable. The study's findings also show that cloud providers can choose the algorithms that best match their operational priorities. These priorities may include maximizing performance, minimizing energy consumption, or ensuring service stability.

KEYWORDS

virtual machine (VM) placement, cloud computing, resource optimization, energy efficiency, multi-objective optimization, performance evaluation

Lakhouari, W., El Ghmary, M., Echoukairi, H. (2026). Benchmarking the Pareto Frontier of VM Placement: A Multi-Objective Evaluation of Heuristics and Metaheuristics Algorithms. *International Journal of Interactive Mobile Technologies (iJIM)*, 20(15), pp. 4–21. https://doi.org/10.3991/ijim.v20i15.61457

Article submitted 2026-03-16. Revision uploaded 2026-06-11. Final acceptance 2026-06-15.

1 INTRODUCTION

Cloud computing has really transformed the way organizations manage their computing resources. Instead of relying on the typical local servers, the organizations are now opting to use the shared infrastructures available on the internet. This allows the services to expand with the firms while keeping the costs low [1]. However, with the increased use of the cloud computing paradigm, it is becoming more difficult to allocate the virtual machines on the servers [2].

Right now, putting virtual machines (VM) isn't just about choosing a place on the available servers. It's now very important for the placement strategy to take into account power consumption, reliability, and environmental friendliness [3]. Today's cloud data centers use a lot of electricity, and the overall need for power is growing very quickly because cloud services are expanding so much [4]. Various research indicates that energy consumption (EC) has become one of the most important operational issues, as this is not only due to increasing economic costs but also to environmental sustainability [5]. Thus, improving energy efficiency has become a major objective in cloud resource management. At the same time, cloud service providers need to ensure that they do not experience any disruptions that might violate service level agreements (SLAs) [5].

The placement of a virtual machine in a cloud computing system has been a complex issue, given its objectives, which are to minimize power consumption, optimize the use of resources, and ensure the performance of services, as discussed in [6], [7]. The purpose of this study is to answer the following research question: "Which VM placement algorithm offers the best trade-off between placement success rate, energy consumption, execution time, and service reliability when evaluated in the same simulation environment?"

The study addresses the issue by evaluating six algorithms to optimize VM allocation in the proposed environment. The algorithms range from simple approaches, such as Best Fit Decreasing (BFD), First Fit Decreasing (FFD), and Worst Fit (WF), to random allocation, to more complex methods. The study focuses on two specific algorithms: Ant Colony Optimization (ACO) and NSGA-III, a newer multi-objective evolutionary algorithm. Both have been widely studied for solving problems related to resource allocation and placement. [8], [9], [10]. Each algorithm runs 400 times, and its performance is carefully measured to ensure a fair comparison. The main goals of this work are threefold: first, to check whether advanced optimization methods perform better than traditional heuristics; second, to examine the cost of using more complex methods; and third, to determine which algorithms are best suited to different types of operational needs. We use statistical tests to ensure that any differences we observe are real and not due to chance. As part of this study, a comprehensive comparative evaluation was conducted on a set of six VM placement algorithms. These algorithms included classic heuristics and advanced metaheuristics. A unified simulation framework was developed, and 2,400 experimental trials were conducted. To ensure a comprehensive and objective comparison, performance was evaluated using a rigorous multi-factor analysis. Statistical analysis and validation were performed using the Kruskal-Wallis and Mann-Whitney tests. This approach enabled the detection of significant performance fluctuations. Unlike existing comparative studies that evaluate algorithms on isolated metrics, this work uniquely combines a unified simulation framework, rigorous statistical validation, and a multi-objective Pareto analysis to benchmark six VM placement algorithms under identical conditions. For cloud service providers, it is important to consider the trade-offs among speed, energy efficiency, and reliability. This method allows providers to select algorithms that are appropriate for their operational goals.

These studies are essential for advancing mobile cloud computing and edge computing. The VM placement algorithms investigated here shape the performance of cloud-based mobile applications, including mobile learning platforms, augmented reality apps, and live gaming [11]. For instance, BFD/FFD algorithms (<1 ms) deliver the low latency demanded by mobile services, while the NSGA3 algorithm extends battery life for mobile edge servers.

Building on the scenarios above, the article is organized as follows: Section 2 reviews recent work. Section 3 details our methodology and experimental setup, including evaluation metrics. Section 4 presents our results. Section 5 analyzes and discusses these findings. Section 6 offers practical recommendations and directions for future research.

2 RELATED WORK

Traditional heuristic algorithms dominate cloud environments due to their low computational cost and rapid decision-making. FFD is routinely adopted for swift placement decisions. Though not specifically tailored for energy optimization, comparative studies affirm FFD as an effective baseline that achieves reliable placement efficiency in heterogeneous cloud infrastructures, particularly compared to more complex bio-inspired techniques [8].

Best Fit Decreasing, a widely adopted heuristic, targets resource consolidation and minimizes active physical servers. Studies demonstrate BFD's effectiveness in improving server consolidation and resource utilization, despite its omission of energy and thermal considerations—leading to risks of overconsumption or overheating under heavy load [12]. In contrast, the WF method proactively spreads VMs to balance workloads and improve thermal stability, even as it raises the number of active machines [13]. Recent research advocates hybrid approaches that integrate conventional heuristics with local optimizations, enabling energy efficiency without sacrificing speed [14].

Metaheuristics are increasingly used for VM placement due to their ability to efficiently explore solution spaces and avoid local optima. ACO, in particular, is now attracting considerable interest because it dynamically adapts pheromones based on server loads and resources, leading to higher placement quality than other heuristics [15]. However, ACO requires more processing time and, due to its inherent randomness, often produces imprecise results [9]. Accelerating convergence remains a key challenge. Recent studies are leveraging machine learning to sharpen optimization efficiency [16].

Cloud computing research increasingly adopts scalable multi-objective algorithms for VM placement, as they efficiently optimize multiple criteria. NSGA-II is widely applied to balance EC and quality of service by producing a set of trade-off solutions rather than a single result [10]. For three or more objectives, NSGA-III advances this approach, yielding a well-distributed Pareto optimal solution set suited for complex environments [17]. These methods, however, are computationally intensive. To counter this, new strategies combine NSGA-III with machine learning for accelerated initialization and convergence, greatly speeding up optimization [18].

Emerging hybrid approaches such as DRL-based strategies [19] and ACO-PSO methods [20] demonstrate clear gains in energy efficiency and SLA compliance. Yet, these innovations seldom undergo systematic benchmarking against classical heuristics under controlled conditions—a critical gap addressed directly by this study.

Despite progress in VM placement research, significant gaps persist. The absence of standardized experimental methods hinders comparison across studies [21]. Additionally, most studies report only averages without statistical validation and disregard the time to results—crucial in cloud computing. Focusing on single-objective optimization dominates the literature, with multi-objective approaches largely neglected [7].

3 METHODOLOGY

3.1 Problem formulation

Virtual machine placement is modeled as a constrained multi-objective optimization problem, consistent with established formulations in cloud resource management literature [22], [7], [23]. Let $H = \{h1, \ldots, h_{|H|}\}$ be the set of physical hosts, where $|H|$ represents the total number of hosts. Let $V = \{v1, \ldots, v_{|V|}\}$ the set of virtual machines, where $|V|$ is the total number of VMs. Binary decision variables are defined as $x_{ij} \in \{0,1\}$

where $x_{ij} = 1$ if VM v_j is assigned to host h_i.

Constraints: Each VM is assigned to at most one host: $\sum_{i=1}^{|H|} x_{ij} \leq 1$

Host resource constraints must be satisfied simultaneously to prevent contention and SLA degradation [7], [24], [25]:

- CPU constraint: $\sum_{j=1}^{|V|} x_{ij}(mips_j \times pes_j) \leq MIPS_i$
- Memory constraint: $\sum_{j=1}^{|V|} x_{ij} ram_j \leq RAM_i$
- Bandwidth constraint: $\sum_{j=1}^{|V|} x_{ij} bw_j \leq BW_i$

Where:

$mips_j$: the amount of computing power needed for each processing element
pes_j: the number of virtual cores assigned to the VM
ram_j and bw_j: the memory and bandwidth the VM needs
$MIPS_i$: total processing capacity of host h_i
RAM_i: total memory capacity available on host h_i
BW_i: total network bandwidth capacity of host h_i

The simultaneous enforcement of these multi-dimensional constraints differentiates VM placement from classical bin-packing and establishes it as an NP-hard problem.

Energy Model: Energy evaluation follows the widely adopted linear power model proposed by Beloglazov et al. [26] and validated in subsequent studies [26], [3]:

$$P_i(u) = P_i^{idle} + (P_i^{max} - P_i^{idle}) \times u$$

P_i^{idle}: baseline power consumption of host h_i at zero CPU utilization
P_i^{max}: maximum power consumption at full utilization
u: normalized the CPU utilization level in the range [0, 1]

Total *EC* over the time horizon *T* is computed as:

$$EC = \int_0^T \sum_{i=1}^{|H|} P_i(u_i(t))dt$$

where $u_i(t)$ represents the time-varying utilization level of host h_i.

This model captures both static and dynamic components of server power consumption and is commonly used in CloudSim-based studies [22].

Objectives: Consistent with energy-aware and SLA-driven cloud optimization frameworks [22], [23], and [27], the evaluated algorithms aim to:

- Minimize energy consumption: min EC
- Maximize placement success rate: $PSR = \frac{\left|\left\{v_j : \sum_i x_{ij} = 1\right\}\right|}{|V|}$
- Minimize SLA violations: $SLAV = \sum_{i=1}^{|H|} 1(u_i > \tau)$
- Minimize execution time: $ET = t_{end} - t_{start}$

Where: τ safe threshold (typically 0.8–0.9).

These objectives reflect the inherent trade-offs between energy efficiency, placement robustness, service reliability, and computational overhead [16], [27].

In addition to the four main optimization objectives that served as a starting point, three complementary indicators were used to perform a more in-depth performance analysis.

3.2 Experimental setup and design

This study analyses six VM placement algorithms using a simulation framework based on CloudSim 3.0.3 [22]. Figure 1 illustrates the experimental approach, which consists of environment configuration, algorithm implementation, metric collection, and statistical analysis.

Building on this framework, we model virtual machine resource allocation and EC without requiring physical infrastructure. We integrate heterogeneous physical servers in the virtual data center, with varying memory, processing performance, and MIPS capacity [3], [9]. and require algorithms to make nuanced placement decisions. We model EC using the widely adopted linear power model [25], [26], which links CPU utilization to server power draw.

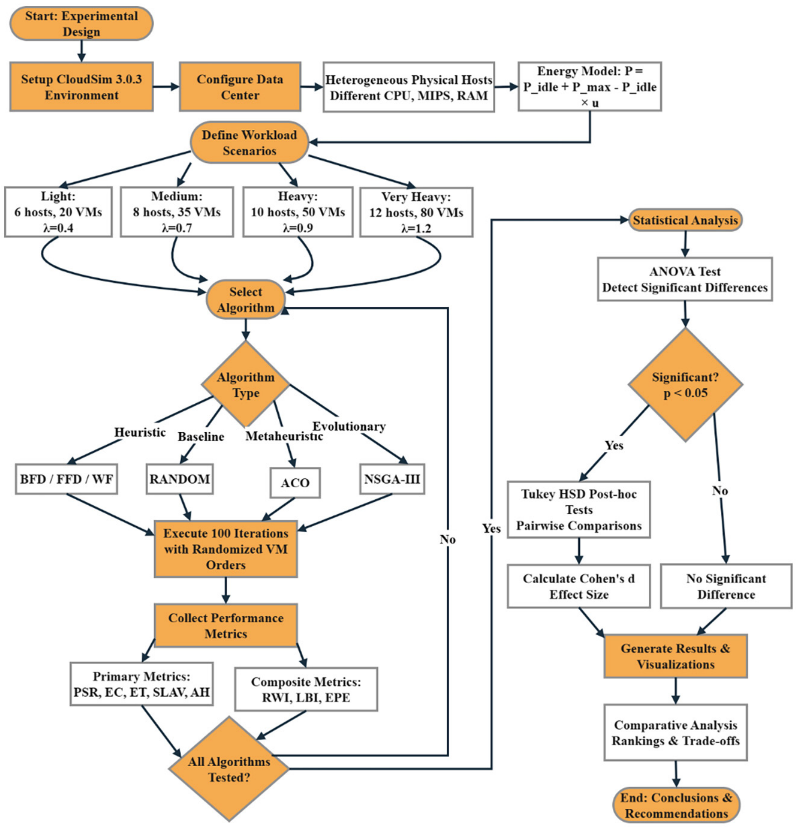

Fig. 1. Experimental methodology flowchart

For our study, we developed a classification with four workload categories: light, medium, heavy, and very heavy, to represent cloud usage needs [2], [5]. These workloads, from minimal to very heavy, differ in hosts, virtual machines, and VM deployment rates. To ensure accurate results and account for changes in VM arrival order, we performed each workload type 100 times, changing the VM arrival order every time [7], [9]. This systematic process gives us more consistent and reliable results.

Table 1. Experimental workload configuration

Workload Type	Hosts	VMs	Arrival Rate (λ)	Runs	Total Experiments
Light	6	20	0.4	100	600
Medium	8	35	0.7	100	800
Heavy	10	50	0.9	100	1000
Very Heavy	12	80	1.2	100	1200
Total	–	–	–	400	2400

Key assumptions of the simulation:

- The resources required by virtual machines are fixed and do not allow for automatic scaling.
- Linear power model: $P(u) = P_{idle} + (P_{max} - P_{idle}) \times u$, where P_{idle} is set to 100 W and P_{max} to 250 W.
- There is no virtual machine migration after initial installation.

- Uniform data center network (10-millisecond latency)
- Workloads introduced in waves (all VMs are already identified in advance)

These assumptions provide controlled conditions. In real-world scenarios, network changes and hardware failures can affect absolute performance metrics but not the relative algorithm order.

To ensure clarity and reproducibility, all six algorithms were tested under identical conditions using a structured simulation. Specifically, as shown in Algorithm 1, steps include scenario setup, iterative runs, data collection, and statistical analysis.

Algorithm 1: Experimental Evaluation Framework

```
Input:
  SCENARIOS = {Light, Medium, Heavy, Very_Heavy}
  ALGORITHMS = {BFD, FFD, WF, RANDOM, ACO, NSGA3}
  R = number of statistical runs
  W = number of warm-up runs
Output:
  Statistical performance metrics for each (algorithm, scenario) pair
Initialize global random seed
for each scenario ∈ SCENARIOS do
    Configure data center infrastructure
    Generate hosts, VMs, and workloads
    // Warm-up phase
    for i = 1 to W do
        Run simulation without recording metrics
    end for
    // Statistical evaluation phase
    for run = 1 to R do
        Set unique random seed
        for each algorithm ∈ ALGORITHMS do
            Apply placement algorithm
            Execute simulation
            Collect performance metrics
        end for
    end for
    Compute mean and standard deviation
end for
Perform ANOVA tests
Apply Tukey HSD post-hoc analysis
Compute effect sizes
Return aggregated results
```

The algorithms were tested under four workloads (20, 35, 50, and 80 VMs) to assess scalability under resource pressure. We selected six algorithms to represent deterministic heuristics and population-based metaheuristics, both common in virtual machine placement research [2], [6]. For each workload, 100 tests were run with random virtual machine arrivals. This achieves statistical robustness and keeps computational cost manageable.

Each design choice—workload diversity, repetition count, and power model—was validated to ensure statistical robustness and reproducibility [28]. Consequently, any change to these factors is expected to affect either the validity or the comparability.

3.3 Algorithms evaluated

Six VM placement strategies were selected to represent a spectrum from simple to more complex optimization. Traditional heuristics, including BFD, FFD, and WF, are compared for their efficiency and implementation simplicity [29]. Random placement acts as a baseline method to provide a neutral comparison. ACO, a bio-inspired metaheuristic, is evaluated for its capability to find efficient solutions via pheromone-based learning. NSGA-III, designed for multi-objective optimization, is assessed for its ability to produce Pareto-optimal solutions balancing several objectives [17].

3.4 Computational complexity analysis

The deterministic heuristics FFD, BFD, and WF have polynomial time complexity. This is since they start with sorting and then move to linear allocation, which takes $O\,(n \log n)$ time. In contrast, the random approach does not require sorting or further processing, thus making it take $O\,(n)$ time. Meanwhile, the ACO and NSGA-III, which use iterative and population-based approaches, have higher computational complexity, i.e., $O\,(G \times P \times n)$, where G stands for the number of iterations, P stands for the population size, and n stands for the problem size. Given these differences in complexity, the execution time (ET) is used as a performance metric.

3.5 Performance metrics

Seven indicators are used to assess the performance of each algorithm in three areas: effectiveness, efficiency, and operational impacts. The Placement Success Rate (PSR) measures how successful the algorithm was in placing the VMs [7] [30]. EC calculates the total amount of power consumption [3]. SLA Violations (SLAV) are detected when service quality degrades from overload. ET represents the computational overhead. Three additional indicators give further insight into effectiveness. The Resource Waste Index (RWI) calculates unused resource capacity. The Load Balance Index (LBI) measures the load balance in the data center. Energy-Performance Aggregate (EPA) aggregates EC and PSR to show the trade-off in green cloud environments.

3.6 Implementation details

All simulations were performed in Java on Dell Latitude 7390 hardware running CloudSim 3.0.3. The device featured Windows 11 Professional, an Intel Core i7-8650U CPU (quad-core, octo-thread), and 16GB RAM. The environment included Version 2024 software, IntelliJ IDEA 3.3, and OpenJDK 21 in server mode with G1GC as the default memory manager. To ensure reproducibility on university-standard equipment, we used default JVM heap settings rather than optimized benchmarks. Fixed seeds were set for both master and test processes to ensure deterministic reproduction of stochastic processes. Preheating tests were omitted to reduce the effect of JVM cold starts and maintain steady measurement conditions. The simulation framework and datasets used are available upon request from the corresponding author, subject to ongoing research constraints.

3.7 Statistical analysis

To compare results, we first used ANOVA to detect differences between the algorithms. After finding significant differences, we applied Tukey HSD to identify which pairs differed. We also used Cohen's d to assess effect size. All simulations used fixed random seeds and began after the warming phase to avoid transient behavior [22]. We used a significance level of $\alpha = 0.05$ for all analyses.

4 RESULTS

An in-depth analysis was conducted on a set of 2,400 experimental trials, divided equally between the two algorithms, with 400 trials for each. Kruskal-Wallis analyses revealed significant disparities between the sets of measurements ($P < 0.05$), highlighting the substantial impact of the choice of algorithm on the results obtained.

Table 2. Performance summary table showing all metrics for 6 algorithms

Algorithm	Placement Rate	Energy Consumption	Execution Time MS	SLA Violations	Active Hosts	N_Runs
BFD	0.8738 ± 0.1320	0.64 ± 0.31	1 ± 1	0.00 ± 0.00	6.7 ± 3.6	400
FFD	0.8738 ± 0.1320	0.65 ± 0.31	0 ± 0	0.00 ± 0.00	6.7 ± 3.6	400
WF	0.8738 ± 0.1320	0.74 ± 0.26	0 ± 0	0.07 ± 0.25	9.0 ± 2.2	400
RANDOM	0.8600 ± 0.1203	0.74 ± 0.29	0 ± 0	0.18 ± 0.40	8.8 ± 2.4	400
ACO	0.8267 ± 0.1004	0.68 ± 0.33	52 ± 36	0.04 ± 0.20	7.4 ± 2.9	400
NSGA3	0.7162 ± 0.0603	0.57 ± 0.23	8 ± 6	0.01 ± 0.09	5.4 ± 1.9	400

4.1 Placement success rate

BFD, FFD, and WF tied for first place, each achieving an impressive placement rate of 87.38% (±13.20%). BFD delivers consistent results regardless of workload, with placement rates ranging from 74.2% (heavy) to 93.7% (high). Its descending sort strategy naturally manages diverse virtual machines by prioritizing the most important requests. Following the top performers, RANDOM achieved 86.00% (±12.03%), ACO reached 82.67% (±10.04%), and NSGA3 recorded 71.62% (±6.03%). According to the Mann-Whitney tests, the BFD/FFD/WF algorithms significantly outperform the other solutions ($p < 0.0002$ *vs RANDOM*, $p < 10^{-12}$ *vs ACO*, $p < 10^{-58}$ *vs NSGA3*).

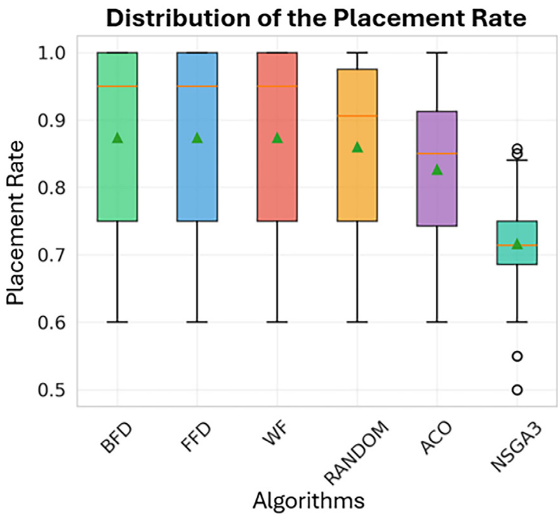

Fig. 2. Distribution of PSR across the six evaluated VM placement algorithms

4.2 Energy consumption

In the experiment conducted, NSGA3 recorded a normalized consumption of 0.57 (±0.23). The results obtained for BFD were 0.64 (±0.31), while those for FFD were 0.65 (±0.31). The results of the analysis conducted by ACO revealed a value of 0.68 (±0.33). The results of the statistical analysis reveal that WF and RANDOM both recorded 0.74 (±0.26) and 0.74 (±0.29), respectively. The rigorous and in-depth statistical analyses revealed no significant difference between WF and RANDOM ($p = 0.951$).

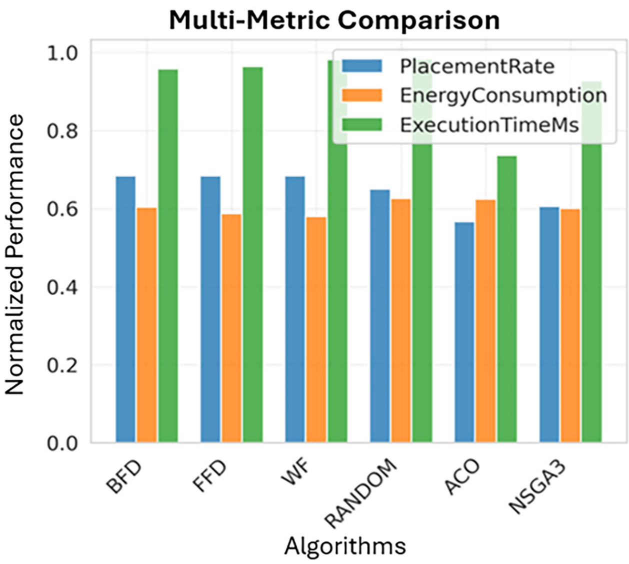

Fig. 3. Multi-metric comparison (PSR, energy consumption, execution time) across algorithms

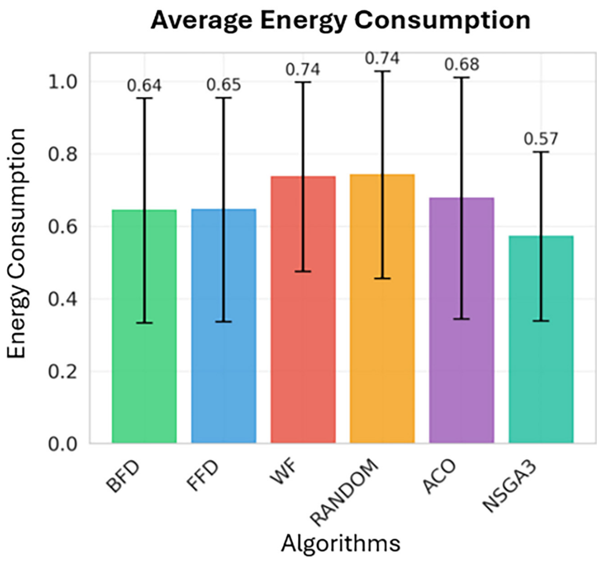

Fig. 4. Normalized EC of the six VM placement algorithms

FFD achieves energy efficiency indirectly by filling hosts sequentially, reducing the average number of active hosts to 6.73 compared to load-balancing methods like WF, which use 9.00 hosts. A decrease in the number of active hosts leads to a direct reduction in standby power consumption. While NSGA3 reduces EC by 11%, this results in an 18% decrease in the PSR and execution that is 47 times slower compared to FFD.

4.3 Execution time

In the study, the RANDOM tool recorded a reaction time of 0.05 milliseconds (±0.26), followed by the WF (0.11 ± 0.45) and FFD (0.18 ± 0.49) tools. BFD recorded a latency time of 0.55 milliseconds (±1.00 milliseconds). The NSGA3 evaluation took 8.44 milliseconds, with a standard deviation of 6.27 milliseconds. ACO recorded 52.03 ms (±36.08). As shown in Figure 5, all comparisons performed were statistically significant ($p < 10^{-10}$).

4.4 SLAV

It should be noted that BFD and FFD recorded no violations in the 400 executions.

Furthermore, it was found that NSGA3 recorded 0.0075 (±0.09). The results of the analysis conducted by ACO reveal a value of 0.04 (±0.20). As part of the study, the firm WF recorded a result of 0.07 (±0.25). As part of the study, statistical analysis revealed that RANDOM recorded a value of 0.18 (±0.40). Statistical analyses revealed that BFD/FFD obtained significantly higher results than RANDOM ($p < 10^{-18}$), but no significant difference was observed between BFD/FFD and NSGA3 ($p = 0.083$).

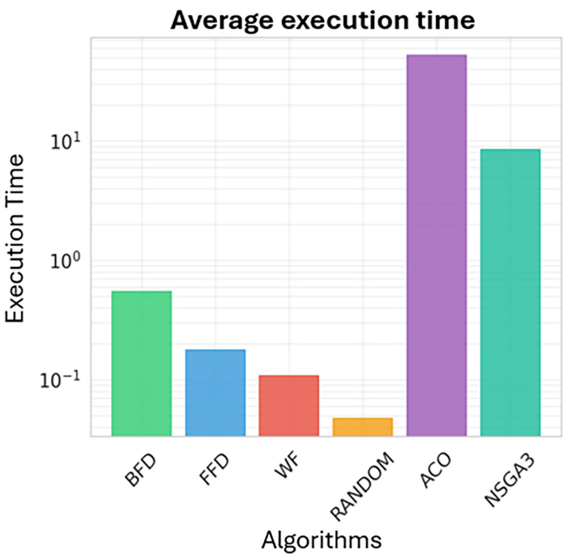

Fig. 5. Average ET (ms, log scale) of the evaluated algorithms

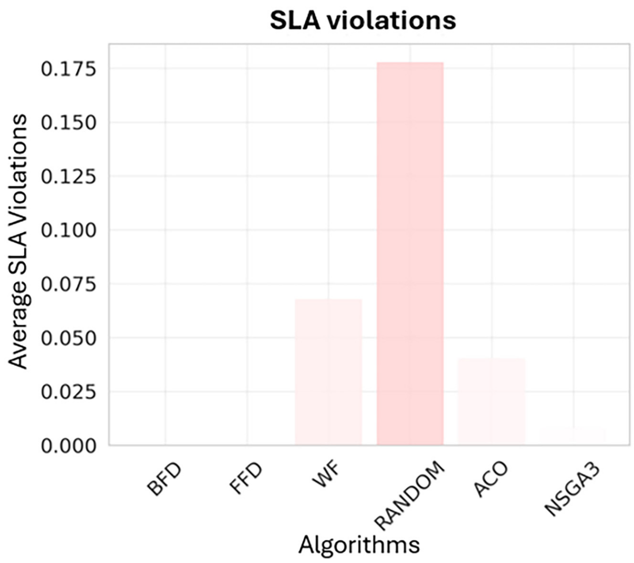

Fig. 6. Average SLA violation rate across 400 simulation runs

4.5 Resource utilization

As part of the NSGA-III performance evaluation, it was observed that the program used 5.4 ± 1.9 active hosts. Analysis of the data shows that the average number of active hosts used by BFD and FFD is approximately 6.7. According to the data collected, the ACO company used 7.4 ± 2.9 hosts. As part of the analysis of WF's performance, it was observed that 9.0 active hosts were used. Statistical analysis of the data reveals that RANDOM used 8.8 ± 2.4 hosts.

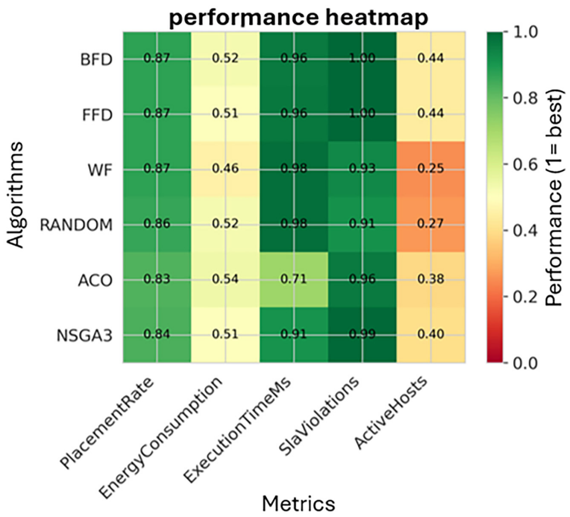

Fig. 7. Performance heatmap summarizing normalized metrics across algorithms

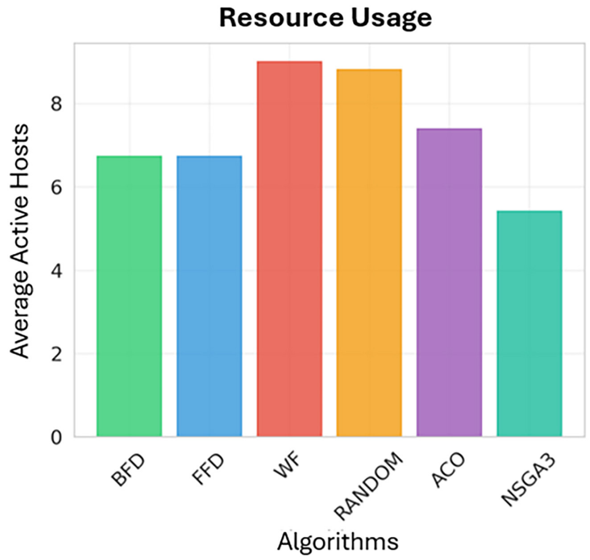

Fig. 8. Average number of active hosts per algorithm

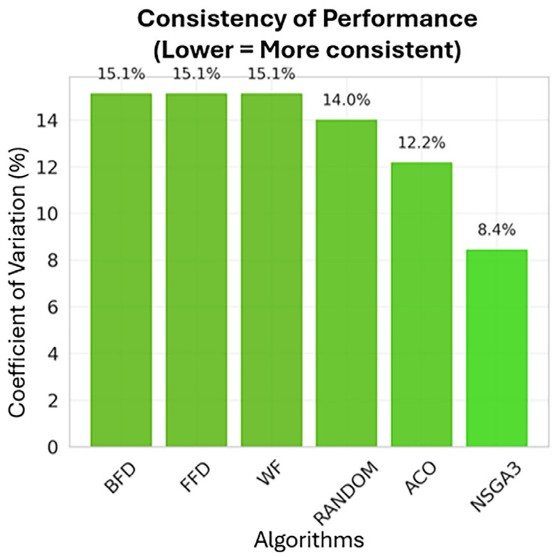

Fig. 9. Performance variation and ranking comparison across algorithms

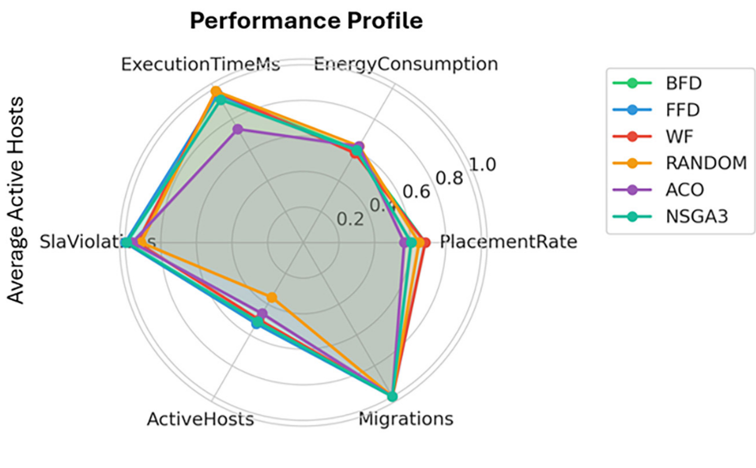

Fig. 10. Radar-based performance profile across multiple evaluation metrics

4.6 Pareto frontier analysis

To complement single-metric rankings, we analyzed Pareto dominance across six pairs of objectives. This included minimizing energy consumption, SLA violations, and ET while maximizing placement rate. As shown in Figure 11, no single algorithm achieves global optimality. NSGA3 dominates in energy efficiency, but this comes at the cost of placement rate. BFD and FFD are Pareto-optimal for reliability, achieving zero SLA violations. In the SLAV vs. Placement Rate space, BFD and FFD occupy the optimal corner. They show zero violations and a maximum placement rate. This makes the Pareto front reduce to a single point. Across all objective pairs, ACO is the only fully dominated algorithm. These results offer practitioners a multi-objective decision framework. It is tailored to deployment priorities and is particularly relevant for energy-constrained mobile edge environments.

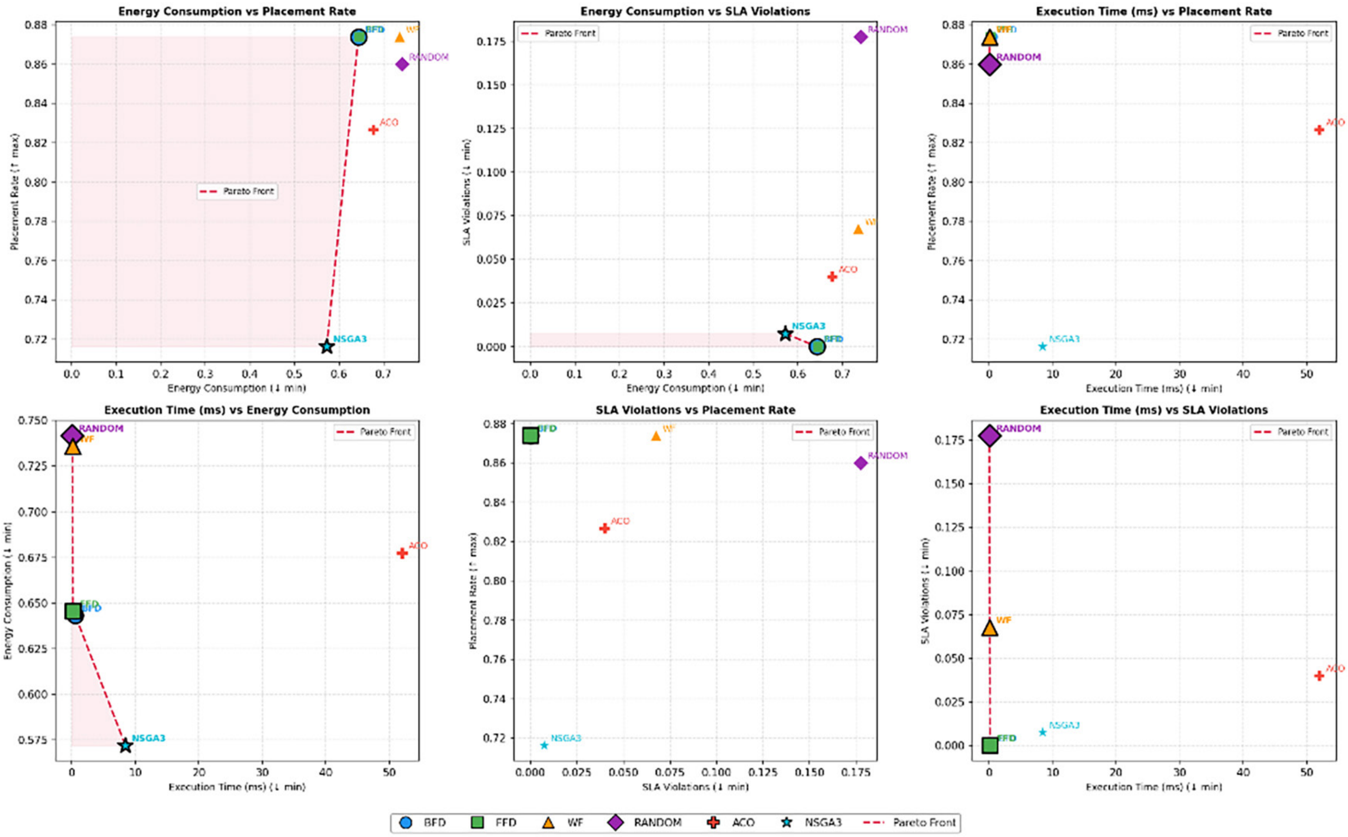

Fig. 11. Pareto frontier analysis – VM placement algorithms

5 DISCUSSION AND ANALYSIS

After analyzing the performance of all algorithms, BFD and FFD demonstrate comparable results due to their fixed and deterministic stacking logic. Their primary distinction lies not in when they consider placements, but in how they select the host. Results suggest that traditional heuristics remain valid and consistently effective, providing reliable behavior and optimal placements. Notably, the RANDOM approach performs nearly as well as the top traditional heuristics, implying that in resource-rich environments, the marginal benefits of advanced optimizations become less apparent.

In contrast, ACO shows reduced placement efficiency compared to traditional heuristics. Its broader search incurs higher computational costs without yielding equivalent benefits in this context. Similarly, NSGA-III yields the lowest placement success among the algorithms, as it targets energy efficiency within a multi-objective optimization setting, prioritizing energy savings over placement trade-offs. However, NSGA-III achieves the lowest normalized EC of all algorithms, effectively prioritizing multiple objectives.

This highlights the trade-off between the study's two objectives: boosting energy efficiency decreases placement success. Conversely, FFD and BFD maintain favorable placement success without SLA violations, although at a moderate energy cost. All algorithms meet real-time execution requirements for cloud computing. However, ACO and NSGA-III entail greater computational costs and variability, underlining the complexity of evolutionary and metaheuristic methods. Comparing overall ranking, FFD leads, followed by BFD and WF, which strike a balance among placement, accuracy, and speed. RANDOM ranks in the middle with minimal overhead, while NSGA-III and ACO follow due to overhead outweighing their efficiency and speed benefits. Overall, these results show that greater algorithm complexity does not necessarily translate to superior performance. Rankings were derived by applying equal weighting to all parameters. In practice, parameter weights may be adjusted to meet specific requirements.

Table 3. Algorithm selection guide for practitioners

Use Case	Recommended Algorithm	Reason
Real-time mobile apps	FFD	Ultra-fast execution (0.18 ms)
Eco-friendly data center	NSGA3	11% energy savings
Highly Reliable Production	BFD	Zero SLA Violations
Edge computing	WF	Balanced load distribution

Based on our results, we recommend: (1) For mobile cloud platforms requiring minimal latency, use the default FFD algorithm, as its ET is under one millisecond and it fully meets the SLA. (2) For green data centers prioritizing energy savings, use NSGA3 during off-peak times, when a longer ET (8.44 ms) is acceptable. (3) For hybrid methods, use BFD initially and rebalance with NSGA3 every six to 12 hours. Pareto analysis supports the findings: simple heuristics (BFD, FFD) are Pareto-optimal for reliability, while NSGA3 is Pareto-optimal for energy-constrained environments like mobile edge computing.

This multi-objective view offers practitioners a decision framework beyond single-metric rankings. Despite our robust experimental framework, results assume static workloads without VM migration. Future work should validate these findings under dynamic workloads and varied infrastructure.

6 CONCLUSION AND FUTURE WORK

This study delivers four decisive contributions: (1) the first comprehensive benchmark of heuristics versus metaheuristics, featuring 2,400 simulations and statistical validation; (2) clear empirical evidence that higher algorithmic complexity does not translate to better performance—simple heuristics outperform complex metaheuristics in reliability; (3) precise quantification of multi-objective trade-offs along the Pareto frontier; and (4) actionable decision guidelines for algorithm selection in real-world deployments.

This study rigorously compares six VM placement algorithms across 2,400 simulation runs, with robust statistical validation. Classical heuristics excelled. First-Fit Decreasing (FFD) dominated, achieving the highest placement success rate, zero SLA violations, and the fastest run times. Best-Fit Decreasing (BFD) followed closely, reaffirming the strength of traditional bin packing in latency-sensitive production. Among metaheuristics, NSGA-III minimized energy usage, confirming the value of multi-objective optimization, but at clear costs: lower placement success and increased run times. ACO did not justify its extra computational burden in real-world terms.

When reliability and speed are required, FFD and BFD are unequivocally superior. For energy efficiency, NSGA-III performs well, provided its computational load is controlled. The core message: choose algorithms aligned with real-world priorities, not just theoretical appeal.

Nonetheless, this study has clear limitations. It centers on static VM placement, omitting dynamic workloads and migrations. The linear energy model, though standard, simplifies actual server behavior. Future research should integrate hybrid approaches—combining heuristic speed with evolutionary optimization—and expand to dynamic settings, containers, and “green” cloud platforms.

7 REFERENCES

[1] S. S. Gill and R. Buyya, “A taxonomy and future directions for sustainable cloud computing: 360 degree view,” *ACM Comput. Surv.*, vol. 51, no. 5, pp. 1–33, 2019. https://doi.org/10.1145/3241038

[2] H. Zhuang and B. E. Ghouchani, “Virtual machine placement mechanisms in the cloud environments: A systematic review,” *Kybernetes*, vol. 50, no. 2, pp. 333–368, 2021. https://doi.org/10.1108/k-09-2019-0635

[3] S. Azizi, M. Zandsalimi, and D. Li, “An energy-efficient algorithm for virtual machine placement optimization in cloud data centers,” *Clust. Comput.*, vol. 23, no. 4, pp. 3421–3434, 2020. https://doi.org/10.1007/s10586-020-03096-0

[4] A. N. Quttoum, “Interconnection structures, management and routing challenges in cloud-service data center networks: A survey,” *Int. J. Interact. Mob. Technol. IJIM*, vol. 12, no. 1, pp. 36–60, 2018. https://doi.org/10.3991/ijim.v12i1.7573

[5] A. Katal, S. Dahiya, and T. Choudhury, “Energy efficiency in cloud computing data centers: A survey on software technologies,” *Clust. Comput.*, vol. 26, no. 3, pp. 1845–1875, 2023. https://doi.org/10.1007/s10586-022-03713-0

[6] A. Alashaikh, E. Alanazi, and A. Al-Fuqaha, “A survey on the use of preferences for virtual machine placement in cloud data centers,” *ACM Comput. Surv.*, vol. 54, no. 5, pp. 1–39, 2022. https://doi.org/10.1145/3450517

[7] R. Regaieg, M. Koubàa, Z. Ales, and T. Aguili, “Multi-objective optimization for VM placement in homogeneous and heterogeneous cloud service provider data centers,” *Computing*, vol. 103, no. 6, pp. 1255–1279, 2021. https://doi.org/10.1007/s00607-021-00915-z

[8] T. Tlili and S. Krichen, "Best fit decreasing algorithm for virtual machine placement modeled as a bin packing problem," in *2023 9th International Conference on Control, Decision and Information Technologies (CoDIT)*, IEEE, 2023, pp. 1261–1266. https://doi.org/10.1109/codit58514.2023.10284347

[9] K. Karmakar, R. K. Das, and S. Khatua, "An ACO-based multi-objective optimization for cooperating VM placement in cloud data center," *J. Supercomput.*, vol. 78, no. 3, pp. 3093–3121, 2022. https://doi.org/10.1007/s11227-021-03978-z

[10] M. Bekhit, A. Fathalla, E. Eldesouky, and A. Salah, "Multi-objective VNF placement optimization with NSGA-III," in *Proceedings of the 2023 International Conference on Advances in Computing Research (ACR'23)*, vol. 700, K. Daimi and A. Al Sadoon, Eds., in Lecture Notes in Networks and Systems, Cham: Springer Nature Switzerland, 2023, pp. 481–493. https://doi.org/10.1007/978-3-031-33743-7_39

[11] D. Uma Nandhini, S. Udhayakumar, L. Tamilselvan, and J. Silviya, "Client aware scalable cloudlet to augment edge computing with mobile cloud migration service," *Int. J. Interact. Mob. Technol.*, vol. 14, no. 12, pp. 165–178, 2020. https://doi.org/10.3991/ijim.v14i12.14407

[12] K. Braiki and H. Youssef, "Fuzzy-logic-based multi-objective best-fit-decreasing virtual machine reallocation," *J. Supercomput.*, vol. 76, no. 1, pp. 427–454, 2020. https://doi.org/10.1007/s11227-019-03029-8

[13] S. Dhahbi, M. Berrima, and F. A. M. Al-Yarimi, "Load balancing in cloud computing using worst-fit bin-stretching," *Clust. Comput.*, vol. 24, no. 4, pp. 2867–2881, 2021. https://doi.org/10.1007/s10586-021-03302-7

[14] S. Jangiti, V. Vijayakumar, and V. Subramaniyaswamy, "Hybrid best-fit heuristic for energy efficient virtual machine placement in cloud data centers," *EAI Endorsed Trans. Energy Web*, vol. 7, no. 26, 2020. https://doi.org/10.4108/eai.13-7-2018.162689

[15] M. H. Ferdaus, M. Murshed, R. N. Calheiros, and R. Buyya, "Virtual machine consolidation in cloud data centers using ACO metaheuristic," in *Euro-Par 2014 Parallel Processing*, vol. 8632, F. Silva, I. Dutra, and V. Santos Costa, Eds., in Lecture Notes in Computer Science, Cham: Springer International Publishing, 2014, pp. 306–317. https://doi.org/10.1007/978-3-319-09873-9_26

[16] A. Ghasemi, A. Toroghi Haghighat, and A. Keshavarzi, "Enhancing virtual machine placement efficiency in cloud data centers: A hybrid approach using multi-objective reinforcement learning and clustering strategies," *Computing*, vol. 106, no. 9, pp. 2897–2922, 2024. https://doi.org/10.1007/s00607-024-01311-z

[17] A. Gopu *et al.*, "Energy-efficient virtual machine placement in distributed cloud using NSGA-III algorithm," *J. Cloud Comput.*, vol. 12, no. 1, p. 124, 2023. https://doi.org/10.1186/s13677-023-00501-y

[18] E. Parvizi and M. H. Rezvani, "Utilization-aware energy-efficient virtual machine placement in cloud networks using NSGA-III meta-heuristic approach," *Clust. Comput.*, vol. 23, no. 4, pp. 2945–2967, 2020. https://doi.org/10.1007/s10586-020-03060-y

[19] Z. Ahamed, M. Khemakhem, F. Eassa, F. Alsolami, A. Basuhail, and K. Jambi, "Deep reinforcement learning for workload prediction in federated cloud environments," *Sensors*, vol. 23, no. 15, p. 6911, 2023. https://doi.org/10.3390/s23156911

[20] A. M. Baydoun and A. S. Zekri, "Towards efficient VM placement: A two-stage ACO-PSO approach for green cloud infrastructure," *Int. J. Comput. Netw. Commun.*, vol. 17, no. 5, pp. 117–136, 2025. https://doi.org/10.5121/ijcnc.2025.17507

[21] M. Hosseinzadeh, M. Y. Ghafour, H. K. Hama, B. Vo, and A. Khoshnevis, "Multi-objective task and workflow scheduling approaches in cloud computing: A comprehensive review," *J. Grid Comput.*, vol. 18, no. 3, pp. 327–356, Sep. 2020. https://doi.org/10.1007/s10723-020-09533-z

[22] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. Rose, and R. Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Softw. Pract. Exp.*, vol. 41, no. 1, pp. 23–50, 2011. https://doi.org/10.1002/spe.995

[23] Y. Gao, H. Guan, Z. Qi, Y. Hou, and L. Liu, "A multi-objective ant colony system algorithm for virtual machine placement in cloud computing," *J. Comput. Syst. Sci.*, vol. 79, no. 8, pp. 1230–1242, 2013. https://doi.org/10.1016/j.jcss.2013.02.004

[24] H. Xing, J. Zhu, R. Qu, P. Dai, S. Luo, and M. A. Iqbal, "An ACO for energy-efficient and traffic-aware virtual machine placement in cloud computing," *Swarm Evol. Comput.*, vol. 68, p. 101012, 2022. https://doi.org/10.1016/j.swevo.2021.101012

[25] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers," *Concurr. Comput. Pract. Exp.*, vol. 24, no. 13, pp. 1397–1420, 2012. https://doi.org/10.1002/cpe.1867

[26] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Gener. Comput. Syst.*, vol. 28, no. 5, pp. 755–768, 2012. https://doi.org/10.1016/j.future.2011.04.017

[27] R. Buyya, S. Ilager, and P. Arroba, "Energy-efficiency and sustainability in new generation cloud computing: A vision and directions for integrated management of data centre resources and workloads," *Softw. Pract. Exp.*, vol. 54, no. 1, pp. 24–38, 2024. https://doi.org/10.1002/spe.3248

[28] M. E. Seno, O. K. J. Mohammad, and B. N. Dhannoon, "CLR: Cloud linear regression environment as a more effective resource-task scheduling environment (State-of-the-Art)," *Int. J. Interact. Mob. Technol. IJIM*, vol. 16, no. 22, pp. 157–175, 2022. https://doi.org/10.3991/ijim.v16i22.35791

[29] W. Lakhouari, M. El Ghmary, A. Er-Rahmani, and H. Echoukairi, "Intelligent virtual machine placement in cloud computing: Algorithms, applications, and key challenges," in *Proceedings of the 4th International Conference on Big Data and Artificial Intelligence Applications (ICBDAIA'25)*, vol. 1724, S. Tkatek, Y. Fakhri, and A. El Makrani, Eds., Lecture Notes in Networks and Systems, Cham: Springer Nature Switzerland, 2026, pp. 315–329. https://doi.org/10.1007/978-3-032-10895-1_26

[30] H. Echoukairi *et al.*, "Effect of mobility models on performance of novel centralized clustering approach based on K-means for wireless sensor networks," *IJAER*, vol. 12, no. 10, pp. 2575–2580, 2017.

8 AUTHORS

Wijdane Lakhouari is currently pursuing a Ph.D. with the Computer Sciences Department, Faculty of Sciences, Mohammed V University in Rabat, Morocco. She conducts her research in the Intelligent Processing Systems & Security (IPSS) Laboratory, where her thesis focuses on optimization and management of virtual resources in cloud computing environments. She holds a Specialized Master's degree in Data Science from the Faculty of Sciences, and a Bachelor's degree in Computer Science Education from the École Normale Supérieure (ENS) (Higher Normal School for Teachers). Her research interests include cloud computing, virtualization, virtual machine placement, multi-objective optimization, heuristics and metaheuristics, hybrid approaches, performance evaluation, and artificial intelligence. She has also co-authored and published research, including the paper "Intelligent Virtual Machine Placement in Cloud Computing: Algorithms, Applications, and Key Challenges." (E-mail: wijdane.lakhouari@um5r.ac.ma).

Mohamed El Ghmary is a Professor of Computer Science at the National School of Arts and Crafts (ENSAM), Mohammed V University in Rabat, Morocco. Previously, he was a Professor of Computer Science at the Faculty of Sciences Dhar El Mahraz (FSDM), Sidi Mohamed Ben Abdellah University, Fez, Morocco. He is an associate member of the Intelligent Processing and Security of Systems (IPSS) research team within the Department of Computer Science at the Faculty of Sciences, Mohammed V University (UM5), Rabat. He also holds an associate membership at the Research Laboratory in Computer Science and Telecommunications (LARIT), specifically within the Networks and Telecommunications team at the Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco. His research interests include Mobile Edge Computing (MEC), Cloud Computing, Machine Learning, Deep Learning, Intelligent Systems, and Optimization (E-mail: mohamed.elghmary@um5.ac.ma).

Hassan Echoukairi is a professor of computer science at the Faculty of Science of Rabat (FSR) Mohamed V University in Rabat (UM5). He received his doctorate from the National Higher School of Electricity and Mechanics (ENSEM) at Hassan II University of Casablanca. Echoukairi is a permanent member of the Intelligent Processing and Security of Systems (IPSS) team of Computer Science Department at the FSR. His current research focuses on securing and optimizing the performance of routing protocols in wireless sensor networks (WSNs), the Internet of Things (IoT), and cloud environments. He is the author and co-author of several publications in international journals and conferences. Echoukairi is a member of the organizing committees of several scientific events (Conferences and Congresses). He is also a member of the Technical Program Committee (TPC) of several international conferences and journals (E-mail: h.echoukairi@um5r.ac.ma).