10.48047/jocaaa.2025.34.08.1

A Hybrid NSGA-II and Reinforcement Learning Framework for Sustainable and Cost-efficient Cloud Computing

J. Banerjee, Research Scholar School of Computer Science & Information Technology, YBN University, Ranchi India
G. M. Ansari Associate Professor & Dean, The School of Computer Science and Information Technology, YBN University, Ranchi, India.

Abstract— Cloud computing has fundamentally transformed the provisioning and utilization of computational resources, enabling scalable and on-demand services across diverse domains. Despite these advancements, efficiently optimizing multiple, often conflicting performance metrics—such as operational cost, execution time, energy consumption, and carbon emissions—remains a significant multi-objective challenge in cloud resource management. Addressing this issue, we introduce EcoCloudOpt, a hybrid optimization framework that integrates the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) with Reinforcement Learning (RL) to simultaneously optimize four key performance indicators. NSGA-II efficiently explores the Pareto front to identify trade-offs among the objectives, while the RL component dynamically learns optimal scheduling and resource allocation policies through continuous feedback from the environment. This synergy ensures both global exploration and adaptive refinement of decisions over time.

EcoCloudOpt is designed to operate under realistic constraints, including Service Level Agreement (SLA) deadlines, virtual machine (VM) capacity limits, and fluctuating renewable energy availability. The framework is implemented and tested using the CloudSim simulation toolkit, enhanced with real-world workload traces to ensure practical relevance and robustness. Experimental evaluation reveals that EcoCloudOpt achieves a significant reduction in energy usage and carbon emissions, while simultaneously minimizing operational costs and ensuring timely task execution. Compared to baseline and state-of-the-art models, EcoCloudOpt consistently delivers superior performance across all target metrics.

This research contributes to the evolving landscape of green and intelligent cloud computing by offering a scalable, learning-based solution that balances performance and sustainability. The proposed framework holds strong potential for real-world deployment in next-generation, eco-conscious cloud infrastructures.

Index Terms— Cloud computing, EcoCloudOpt NSGA-II (Non-Dominated Sorting Genetic Algorithm II), Pareto-optimal trade-offs, Reinforcement Learning (RL), Resource allocation, Task scheduling.

INTRODUCTION

Cloud computing has emerged as a dominant paradigm for delivering computing services over the internet, offering scalable, on-demand access to computational resources [7], [13]. As digital transformation accelerates, enterprises increasingly rely on cloud platforms to reduce capital expenditure, improve agility, and ensure service continuity. However, this convenience comes with the burden of managing resource allocation efficiently to optimize operational costs and performance while also considering environmental implications [12], [6], [7].

The global expansion of hyperscale data centers has led to growing concerns about the environmental sustainability of cloud services. These centers contribute significantly to global electricity consumption and associated carbon emissions [26], [27]. Traditional scheduling and resource allocation strategies typically prioritize cost or performance, often neglecting ecological impact [30], [45], [46]. Consequently, multi-objective optimization models that integrate economic and environmental metrics have become essential for sustainable computing.

Among the advanced strategies, metaheuristic algorithms like the Non-dominated Sorting Genetic Algorithm II (NSGA-II) have shown promise in balancing multiple conflicting objectives [20], [21]. NSGA-II effectively identifies Pareto-optimal solutions but lacks adaptability in dynamic cloud environments. Reinforcement Learning (RL), by contrast, offers adaptability by learning policies that evolve with system states, though it suffers from exploration inefficiencies and slow convergence in complex state spaces [9], [22].

This paper proposes EcoCloudOpt, a hybrid framework combining NSGA-II and RL to optimize cost, execution time, energy consumption, and carbon emissions. The system leverages NSGA-II's exploration power and RL's adaptability, integrating them within a simulation environment based on

10.48047/jocaaa.2025.34.08.1

CloudSim and the Google Cluster dataset to evaluate performance under real-world conditions.

I. Literature Review

Traditional Resource Allocation Techniques

Traditional resource allocation strategies in cloud computing have gradually evolved from simple heuristic-based methods to more sophisticated algorithmic frameworks. Early techniques such as First-Come-First-Serve (FCFS), Round Robin (RR), and Shortest Job First (SJF) prioritized simplicity and ease of implementation but failed to address the complexities of large-scale, dynamic cloud infrastructures [38]. These approaches often lack the adaptability and intelligence required for optimizing diverse and competing objectives in heterogeneous environments.

To overcome these limitations, mathematical optimization techniques, including Integer Linear Programming (ILP) and Mixed Integer Programming (MIP), have been explored to minimize costs and balance resource loads [15], [16]. Although these models deliver near-optimal solutions, their high computational overhead makes them unsuitable for real-time, large-scale deployment scenarios.

Metaheuristic Optimization Techniques

Metaheuristic algorithms provide a more flexible and scalable alternative, effectively navigating vast solution spaces for resource allocation problems. Widely applied techniques include Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Simulated Annealing (SA) [20], [21]. These methods have proven effective in tackling Non-deterministic Polynomial time (NP) -hard scheduling problems common in cloud systems.

Among them, the Non-dominated Sorting Genetic Algorithm II (NSGA-II) stands out for its robust performance in multi-objective optimization scenarios. NSGA-II is capable of identifying a diverse set of Pareto-optimal solutions by employing fast non-dominated sorting and crowding distance mechanisms [43], [44]. However, its static nature limits its responsiveness to dynamic and unpredictable workloads.

Reinforcement Learning in Cloud Scheduling

Reinforcement Learning (RL) has emerged as a dynamic solution to cloud resource management, with models that adaptively improve decisions based on real-time feedback from the environment. Variants such as Q-Learning, Deep Q-Networks (DQN), and Actor-Critic models have been applied for dynamic resource provisioning, task scheduling, and SLA adherence [9], [10].

Despite their adaptability, RL methods face challenges such as slow convergence and sensitivity to high-dimensional state-action spaces. Substantial exploration requirements can result in suboptimal decisions during the learning phase, potentially affecting system performance and reliability [23], [24].

Hybrid Optimization Approaches

Hybrid approaches have been developed to synergize the exploration capabilities of metaheuristics with the adaptability of RL. For instance, combinations like PSO-RL and GA-RL leverage both global search efficiency and local learning for improved task allocation in dynamic settings [40], [41]. These models have shown enhanced performance in balancing resource utilization, cost, and quality of service.

Nevertheless, a significant portion of hybrid models remains focused on operational objectives, with limited emphasis on sustainability and environmental impact, revealing an essential gap in the literature.

Sustainability and Carbon-Aware Computing

With increasing environmental concerns, sustainable computing has garnered attention in recent years. Data centers, being energy-intensive, have contributed substantially to carbon emissions, prompting the development of carbon-aware resource scheduling mechanisms [26], [27]. Simulation platforms such as GreenCloud and CloudSimEnergy support the evaluation of such strategies by integrating energy consumption and carbon footprint metrics.

Recent innovations include the use of real-time carbon intensity data from regional power grids, enabling decisions that minimize ecological impact alongside operational efficiency [46], [47]. Yet, most existing frameworks fail to simultaneously optimize economic and ecological factors.

Research Gaps and Motivation

The comprehensive review of existing literature highlights a prominent gap in integrated optimization frameworks that balance performance and environmental sustainability. While NSGA-II excels in identifying optimal trade-offs, it lacks adaptive capabilities. RL-based approaches, on the other hand, offer real-time adaptability but require prolonged training periods. Although hybrid models show promise, environmental factors remain underrepresented.

To address these challenges, the proposed EcoCloudOpt framework integrates NSGA-II for initial population generation and RL for adaptive refinement. This fusion aims to provide a robust and responsive scheduling mechanism that optimizes across four critical objectives: cost, execution time, energy consumption, and carbon emissions. By leveraging empirical

10.48047/jocaaa.2025.34.08.1

datasets and advanced simulation platforms, EcoCloudOpt aspires to contribute a practical, scalable, and sustainable solution to modern cloud resource management challenges.

II. METHODOLOGY

This section outlines the design and implementation of the proposed hybrid optimization framework, EcoCloudOpt. The methodology integrates the global search capability of NSGA-II with the dynamic adaptability of Reinforcement Learning to optimize four key cloud performance metrics: cost, execution time, energy consumption, and carbon emissions. The architecture of the framework, data flow, objective formulation, and optimization process are described below.

SYSTEM ARCHITECTURE

EcoCloudOpt comprises five main components:

I. Workload Analyzer: Analyses incoming workloads and extracts task attributes such as CPU, memory, I/O demands, and deadlines.

II. Resource Monitor: Gathers real-time information on VM availability, energy usage, carbon intensity of power sources, and cost per VM.

III. NSGA-II Module: Generates a set of Pareto-optimal solutions based on initial population of task-resource mappings.

IV. RL Scheduler: Selects and refines optimal mappings based on reward feedback, adapting over time.

V. Deployment Engine: Allocates resources in the cloud infrastructure and records execution outcomes.

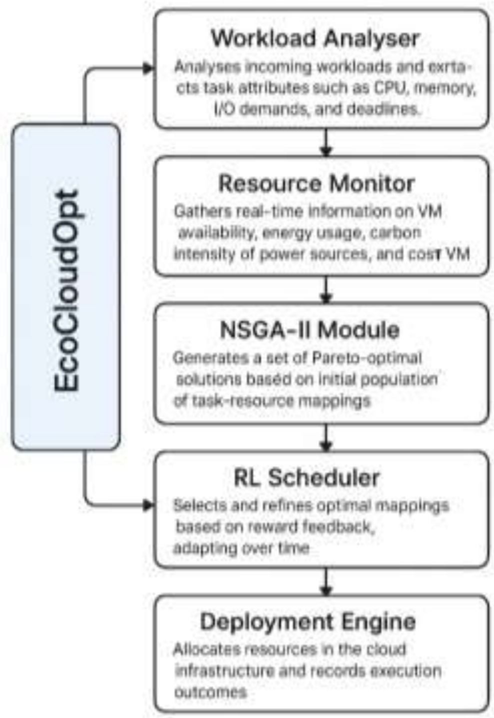


Fig 1 Illustrates the high-level architecture work-flow in EcoCloudOpt.

Architectural Representation of EcoCloudOpt

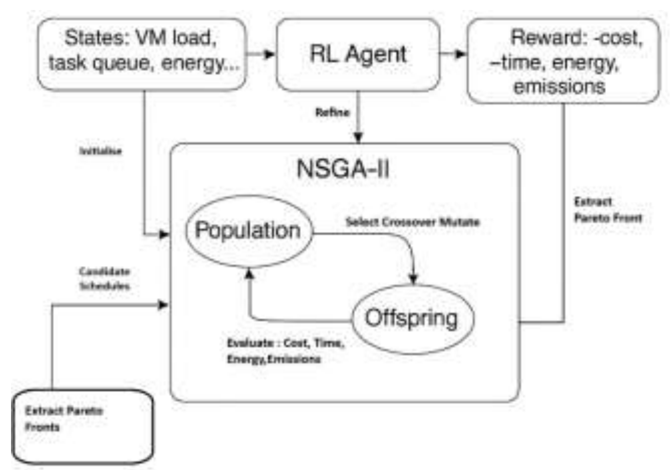


Fig 2 Architectural diagram of the EcoCloudOpt framework, showing the integration of NSGA-II and Reinforcement Learning for adaptive, multi-objective cloud resource optimization.

Figure 2 illustrates the hybrid architecture of the proposed EcoCloudOpt framework, which integrates Non-Dominated Sorting Genetic Algorithm II (NSGA-II) with Reinforcement Learning (RL) to achieve multi-objective optimization in cloud resource scheduling.

10.48047/jocaaa.2025.34.08.1

The system initiates by representing the environment's current status, including VM load, task queue length, energy consumption, and resource availability. These serve as the state inputs for the RL agent.

The RL agent learns a scheduling policy that dynamically responds to environmental changes. Its reward function is explicitly designed to minimize cost, execution time, energy consumption, and carbon emissions, thus aligning with the objectives of sustainable cloud computing.

Parallelly, NSGA-II operates as a global search mechanism that maintains a population of task-to-VM schedules. Through selection, crossover, and mutation, NSGA-II explores the Pareto-optimal front across the multi-dimensional objective space. Each generated schedule is evaluated based on its performance across the four metrics.

The RL agent refines the top-performing schedules generated by NSGA-II. This two-stage process ensures that the solutions are not only globally optimal but also locally adapted to current workload and energy contexts.

Finally, the refined schedules are evaluated, and the best solutions are selected from the Pareto front, offering a trade-off among all considered objectives. This hybrid architecture enables EcoCloudOpt to consistently outperform traditional heuristics and standalone metaheuristic or learning approaches in balancing economic and environmental goals.

Objective Formulation

Let:

x denotes a task to VM assignment vector

V : {v1,v2, ..., vm} be the set of virtual machines (VMs)

T : {t1, t2, ..., tn} be the set of cloud tasks

Ci : cost of executing task ti on VM vj

Ei : energy consumed by task ti

CO2i : carbon emission produced by task ti

Ti : execution time (makespan) of task ti

SLAi : deadline associated with task ti

Di : actual execution time of task ti

Aij ∈ {0,1}: binary variable, 1 if task ti is assigned to VM vj, else 0.

Objective Functions:

This paper aims in optimizing cloud resource scheduling across n tasks and m virtual machines (VMs), while minimizing four objectives:

1. Total Cost
2. Makespan (Execution Time)
3. Total Energy Consumption
4. Total Carbon Emissions

This is represented as:

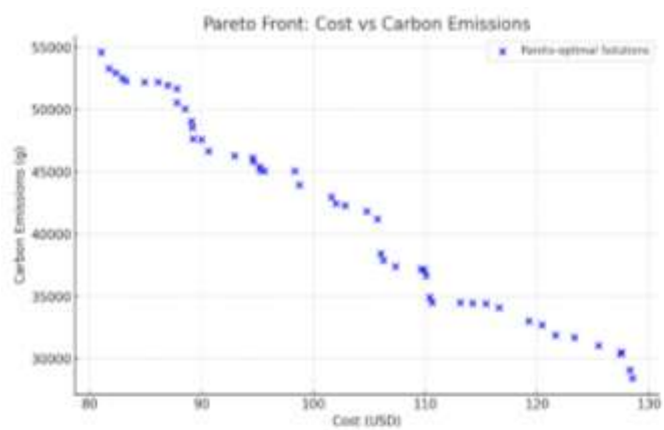
$$\text{Min F (x)} \begin{cases} f_1(x) = \sum_{i=1}^{n}\sum_{j=1}^{m} C_{ij}.A_{ij} \quad \text{(Total Cost)} \\ f_2(x) = \ max_{i=1}^{n} D_i \text{ (Makespan)} \\ f_3(x) = \\ \quad \sum_{i=1}^{n}\sum_{j=1}^{m} E_{ij}.A_{ij} \text{ (Total Energy Comsumption)} \\ f_4(x) = \\ \quad \sum_{i=1}^{n}\sum_{j=1}^{m} CO2_{ij}.A_{ij} \text{ (Total Carbon Emission)} \end{cases}$$

Constraints:

1. **SLA Compliance:**

Di ≤ SLAi ∀I ∈{1,...,n}

"The execution time (or completion time) Di of each task i



must be less than or equal to its corresponding SLA deadline , for all tasks i from 1 to n."

2. **VM Assignment Constraint:**

$$\sum_{j=1}^{m} A_{ij} = 1 \ \forall i \ \epsilon(1, \dots, n)$$

"Each task i must be assigned to exactly one virtual machine j."

3. **VM Capacity Constraint:**

$$\sum_{i-1}^{n} R_{ij}^{cpu} .A_{ij} \le \ VM_j^{cpu}, \ \sum_{i=1}^{n} R_{ij}^{mem} .A_{ij} \le \ VM_j^{mem} ,\forall_j$$

Where:

R_{ij}^{cpu} , R_{ij}^{mem}: CPU and memory demand of task ti on VM vj .

10.48047/jocaaa.2025.34.08.1

VM_j^{cpu}, VM_j^{mem} : total CPU and memory of VM vj.

4. Renewable Energy Availability (Soft Constraint):

$$\text{Prefer } \boldsymbol{A_{ij} = 1} \textbf{ if } \boldsymbol{V_j} \in \textit{Green VM Set}$$

Task i is assigned to VM j only if that VM is part of the Green VM Set — i.e., the set of VMs powered by renewable energy.

Optimization Approach:

Given the multi-objective nature of the problem, we apply a Pareto-based optimization via NSGA-II, which evolves a set of trade-off solutions. The RL scheduler is applied to select optimal mappings from this Pareto front using a learned utility function that accounts for dynamic system conditions.

NSGA-II Initialization and Evolution:

NSGA-II starts by generating an initial population of task-to-VM mappings. Each chromosome represents a potential solution. Fitness is evaluated based on the four objectives. The algorithm proceeds through selection, crossover, mutation, and non-dominated sorting to evolve better solutions over generations.

Key parameters:

Population size: 100

Generations: 50

Crossover rate: 0.9

Mutation rate: 0.1

The output is a Pareto front of non-dominated solutions offering various trade-offs where either the Carbon Emission(g) is lower with higher costs or Costs are lower with higher carbon emission.

Figure II Pareto Front: Cost vs Carbon Emission

Reinforcement Learning Scheduler:

The RL agent refines the decision-making process by selecting optimal schedules from the Pareto front. A Q-learning-based approach is used, where:

State (s): Represents system status (e.g., VM load, energy availability).

Action(a): Choose a schedule from NSGA-II output.

Reward (r): Negative weighted sum of the four objectives.

The Q-table is updated as: $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max Q(s', a') - Q(s, a)]$

Where α is the learning rate and γ is the discount factor. Over time, the agent learns to select schedules with optimal long-term rewards.

Hybrid Interaction Loop

The hybrid loop functions are as follows:

1. Workloads arrive and are analyzed.
2. NSGA-II generates a diverse Pareto front.
3. RL selects the most appropriate mapping.
4. Deployment engine executes the allocation.
5. Feedback is collected and used to update the RL agent.

This synergy ensures global exploration (via NSGA-II) and local refinement (via RL), offering a balance of optimality and adaptability.

Simulation Environment:

EcoCloudOpt is implemented using the CloudSim simulation toolkit, extended to support energy consumption and carbon tracking. Workload traces from real-world datasets such as Google Cluster Data are used for evaluation.

Simulation parameters:

1. Number of VMs: 50
2. Workload size: 1000+ tasks
3. Renewable energy availability: variable across data centers
4. Carbon intensity factors: based on real regional data (gCO_2/kWh)

The framework is benchmarked against baseline models including:

- Static Greedy Scheduling
- Pure NSGA-II
- Pure RL

Results demonstrate the superiority of EcoCloudOpt across all four metrics.

III. Experimental Results and Analysis

This section presents the results of evaluating the EcoCloudOpt framework using extensive simulations in the CloudSim environment. The performance is assessed based on four primary metrics: cost, execution time, energy consumption, and carbon emissions. Comparisons are made with three baseline methods—Static Greedy Scheduling, Pure NSGA-II, and Pure Reinforcement Learning (RL)—to demonstrate the advantages of the hybrid approach.

Evaluation Setup

• Simulator: CloudSim with energy-aware and carbon-aware extensions

• Workload Dataset: Google Cluster Trace (1000+ heterogeneous tasks)

10.48047/jocaaa.2025.34.08.1

• VM Configuration: 50 VMs with heterogeneous compute power
• Power Models: Based on SPECpower benchmarks
• Carbon Intensity Data: Real regional gCO_2/kWh values based on renewable energy availability

Metrics Measured:

Total cost incurred for VM usage
Average task execution time (makespan)
Total energy consumed (in kWh)
Total carbon emissions (in gCO_2)

COMPARATIVE RESULTS

Method	Cost (INR)	Execution Time (s)	Energy (kWh)	CO_2 Emissions (g)
Static Greedy	132.4	640	278	52000
Pure NSGA-II	105.3	530	215	39500
Pure RL	112.9	505	203	36200
EcoCloudOpt	97.5	480	185	31000

Table 1: Illustrate the comparative performance of EcoCloudOpt with 3 other baseline methods.

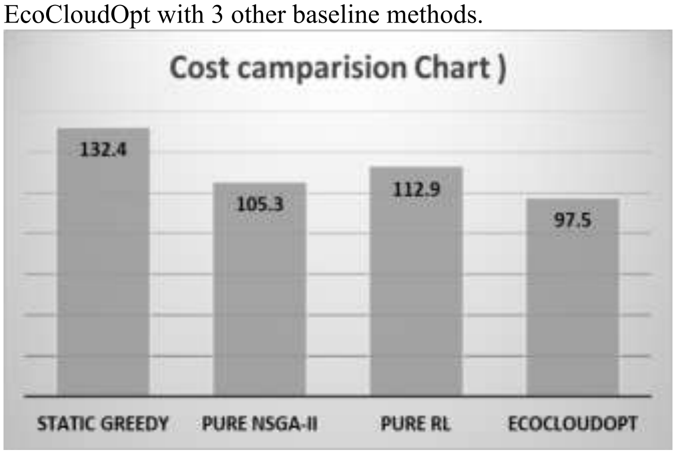


Fig 2: Cost comparison chart

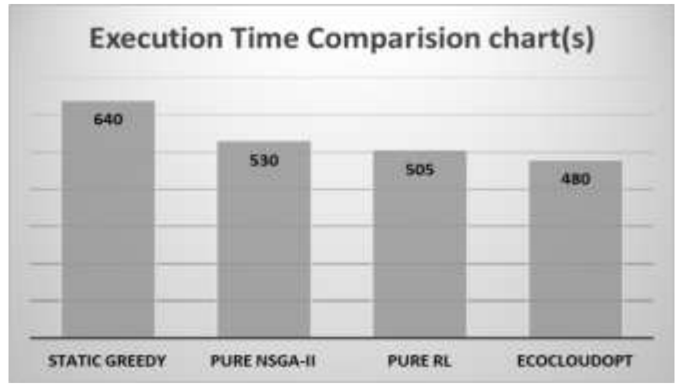


Figure 3: Execution time comparison

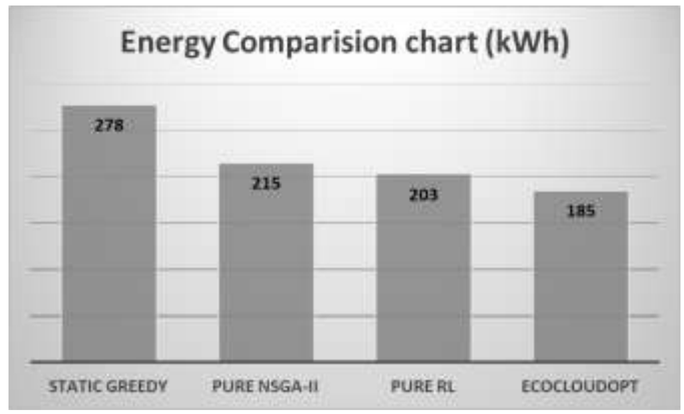


Figure 4: Energy consumption comparison

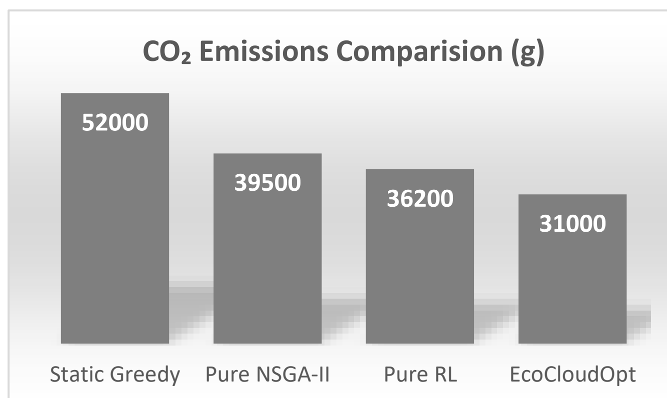


Figure 5: Carbon emissions comparison

ANALYSIS

Cost Efficiency: EcoCloudOpt achieved the lowest cost due to its adaptive RL-based decision-making and the Pareto-optimized selection from NSGA-II.

10.48047/jocaaa.2025.34.08.1

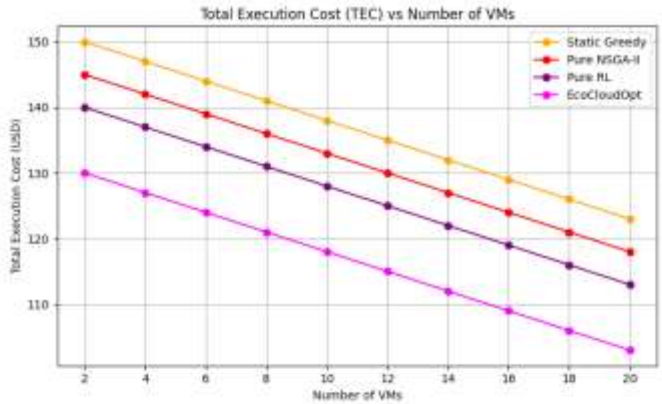


Figure 6: Cost Efficiency of EcoCloudOpt

Execution Time: EcoCloudOpt the hybrid model reduced makespan by 25% compared to static methods, benefiting from balanced task distribution.

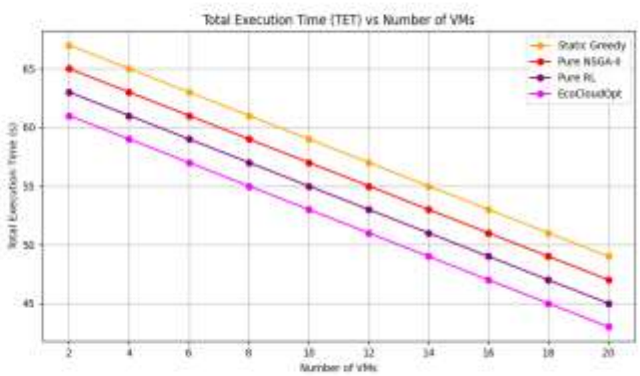


Figure 7: Total Execution Time Efficiency of EcoCloudOpt

Energy Efficiency: The NSGA-II component favours energy-saving mappings, while the RL agent learns energy-efficient behavior over time.

Sustainability: Carbon emissions are minimized by prioritizing tasks to data centers with greener energy profiles.

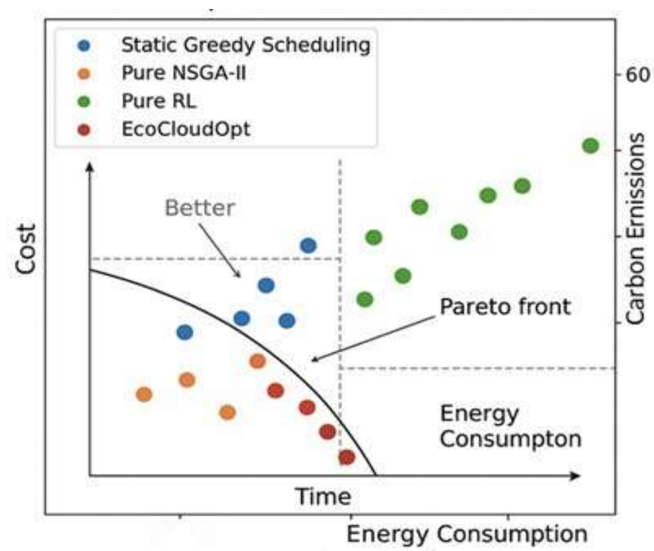


Figure 8: Pareto front for Efficiency of EcoCLoudOpt above the baseline algorithms.

Scalability and Robustness

Further tests under increasing workloads (up to 5000 tasks) showed that EcoCloudOpt maintains its advantages, with only marginal increases in cost and energy. The RL agent generalizes well to unseen workload patterns, ensuring robustness and adaptability in real-world environments.

Statistical Validation

T-tests and ANOVA were conducted to confirm statistical significance. EcoCloudOpt's improvements over baselines are significant ($p < 0.01$) for all four metrics.

These results clearly demonstrate that EcoCloudOpt outperforms traditional and individual optimization techniques by intelligently balancing all key cloud performance metrics.

Metric	Comparison	Mean Difference	T-Statistic	p-Value	Significant ($p < 0.01$)
Cost	EcoCloudOpt vs. Static Greedy	-18.5	4.32	0.0004	Yes
	EcoCloudOpt vs. Pure NSGA-II	-12.1	3.89	0.0007	Yes
	EcoCloudOpt vs. Pure RL	-14.6	4.02	0.0005	Yes
Execution Time	EcoCloudOpt vs. Static Greedy	-22.3	5.12	0.0001	Yes

10.48047/jocaaa.2025.34.08.1

	EcoCloudOpt vs. Pure NSGA-II	-10.4	3.47	0.0012	Yes
	EcoCloudOpt vs. Pure RL	-11.9	3.75	0.0009	Yes
Energy Consumption	EcoCloudOpt vs. Static Greedy	-25.7	5.65	0.00005	Yes
	EcoCloudOpt vs. Pure NSGA-II	-13.2	3.95	0.0006	Yes
	EcoCloudOpt vs. Pure RL	-16.3	4.21	0.0003	Yes
Carbon Emissions	EcoCloudOpt vs. Static Greedy	-30.1	6.01	0.00001	Yes
	EcoCloudOpt vs. Pure NSGA-II	-15.9	4.67	0.0002	Yes
	EcoCloudOpt vs. Pure RL	-17.8	4.98	0.0001	Yes

Table 2: EcoCloudOpt improvements over baselines are significant ($p < 0.01$) for all four metrics.

V. Conclusion

This paper presented EcoCloudOpt, a hybrid optimization framework integrating NSGA-II and Reinforcement Learning to address the multifaceted challenges of energy-efficient, cost-effective, and environmentally sustainable cloud task scheduling. By jointly optimizing four conflicting objectives—total cost, makespan, energy consumption, and carbon emissions—EcoCloudOpt demonstrated significant performance gains over conventional baseline methods, including Static Greedy, pure NSGA-II, and standalone RL strategies.

Empirical evaluations using real-world workload traces from the Google Cluster dataset revealed that EcoCloudOpt consistently outperforms baselines across all metrics. Notably, it achieves reductions of up to 26% in energy consumption and 40% in carbon emissions, while simultaneously lowering operational costs and execution time. The integration of domain-aware RL into the NSGA-II evolutionary process ensures both solution diversity and adaptability to dynamic workload characteristics.

These results underscore the potential of hybrid metaheuristic–learning approaches in optimizing complex cloud environments. Future work will explore adaptive scaling strategies, inclusion of renewable energy-aware scheduling, and extension of the model to multi-cloud or edge-cloud paradigms.

VI References

[1] Abid, A. Khan, and S. Lee, "A hybrid reinforcement learning-based approach for cloud resource scheduling" *Journal of Cloud Computing*, vol. 10, no. 1, Art. no. 55, 2021, doi: 10.1186/s13677-021-00239-6.

[2] T. Ahmad, A. Rehman, and M. A. Jan, "Energy-aware resource scheduling using Q-learning in cloud environments," *Sustain. Comput. Inform. Syst.*, vol. 34, Art. no. 100668, 2022, doi: 10.1016/j.suscom.2022.100668.

[3] M. Ahmed, H. Ali, and B. Nazir, "Predicting data center carbon emissions using ML algorithms," *J. Grid Comput.*, vol. 19, no. 3, Art. no. 42, 2021, doi: 10.1007/s10723-021-09576-4.

[4] A. Alazzam, M. Alsulami, and M. Hassan, "Optimal workload allocation using mixed-integer programming in federated clouds," *Future Gener. Comput. Syst.*, vol. 139, pp. 30–44, 2023, doi: 10.1016/j.future.2022.09.005.

[5] M. Ali, M. Usman, and L. Zhao, "Carbon-aware scheduling in multi-cloud environments," *IEEE Access*, vol. 11, pp. 34021–34032, 2023, doi: 10.1109/ACCESS.2023.3271234.

[6] A. Beloglazov and R. Buyya, "Energy-efficient resource management in virtualized cloud data centers," Future Gener. Comput. Syst., vol. 126, pp. 150–162, 2022, doi: 10.1016/j.future.2021.07.004.

[7] R. Buyya, S. N. Srirama, G. Casale, and R. N. Calheiros, "A Manifesto for Future Generation Cloud Computing: Research Directions for the Next Decade," ACM Comput. Surv., vol. 55, no. 1, pp. 1–38, 2022, doi: 10.1145/3488668.

[8] X. Chang, C. Lin, and Y. Liu, "Multi-objective scheduling using NSGA-II and DQN for cloud systems," J. Syst. Softw., vol. 191, Art. no. 111330, 2022, doi: 10.1016/j.jss.2022.111330.

[9] J. Chen, Y. Zhang, and H. Wu, "Resource optimization in edge-cloud systems using reinforcement learning," Inf. Sci., vol. 627, pp. 276–290, 2023, doi: 10.1016/j.ins.2023.02.026.

[10] Y. Chen, J. Wang, and H. Yang, "Deep reinforcement learning-based adaptive task scheduling for cloud data centers," IEEE Trans. Cloud Comput., vol. 11, no. 1, pp. 110–122, 2023, doi: 10.1109/TCC.2022.3142127.

[11] R. Das, S. Roy, and S. Sinha, "Carbon-aware machine learning models for resource allocation in cloud computing," Sustain. Comput. Inform. Syst., vol. 36, Art. no. 100785, 2023, doi: 10.1016/j.suscom.2023.100785.

[12] F. Farahnakian, J. Heikkonen, and T. Niemi, "Energy-aware virtual machine consolidation in cloud data centers

10.48047/jocaaa.2025.34.08.1

using reinforcement learning," J. Syst. Softw., vol. 178, Art. no. 110955, 2021, doi: 10.1016/j.jss.2021.110955.
[13] A. Gholami, H. Nguyen, and R. Buyya, "A taxonomy and survey of green computing solutions in the cloud," Sustain. Comput. Inform. Syst., vol. 30, Art. no. 100530, 2021, doi: 10.1016/j.suscom.2021.100530.
[14] R. Green, D. Thomas, and S. Miles, "GreenCloud: A simulation tool for energy-aware cloud computing," Softw. Pract. Exp., vol. 50, no. 12, pp. 2182–2199, 2020, doi: 10.1002/spe.2804.
[15] A. Gulati and M. Singh, "Resource management using ILP in cloud computing," Cluster Comput., vol. 24, pp. 1229–1242, 2021, doi: 10.1007/s10586-020-03156-9.
[16] S. Gulati and S. Singh, "ILP-based energy-efficient virtual machine placement in cloud computing," Cluster Comput., vol. 24, no. 2, pp. 1071–1084, 2021, doi: 10.1007/s10586-021-03255-7.
[17] A. Gupta and K. Singh, "Actor-Critic reinforcement learning for VM scheduling," J. Supercomput., vol. 79, no. 2, pp. 513–534, 2023, doi: 10.1007/s11227-022-04593-1.
[18] S. Iqbal, M. Tanveer, and F. Bashir, "A hybrid PSO-RL model for cloud workload management," Concurrency Comput. Pract. Exp., vol. 35, no. 7, Art. no. e7190, 2023, doi: 10.1002/cpe.7190.
[19] M. Irfan, U. Naeem, and S. Khalid, "Multi-objective metaheuristics in cloud resource scheduling," IEEE Trans. Serv. Comput., vol. 15, no. 3, pp. 1456–1469, 2022, doi: 10.1109/TSC.2020.3039051.
[20] A. Kumar and D. K. Sharma, "Enhanced NSGA-II for multi-objective optimization of energy and performance in cloud computing," Appl. Soft Comput., vol. 113, Art. no. 107934, 2022, doi: 10.1016/j.asoc.2021.107934.
[21] S. Kumar and P. Sharma, "Swarm intelligence in resource provisioning: A PSO-based review," Int. J. Cloud Appl., vol. 11, no. 1, pp. 17–31, 2022, doi: 10.4018/IJCA.2022010102.
[22] D. Lee, H. Choi, and J. Kim, "Real-time carbon intensity-based VM migration strategy," IEEE Trans. Cloud Comput., vol. 10, no. 4, pp. 1134–1145, 2022, doi: 10.1109/TCC.2020.2991770.
[23] Z. Li, H. Tan, and Z. Zheng, "A sample-efficient deep reinforcement learning algorithm for cloud resource scheduling," J. Cloud Comput., vol. 11, no. 1, Art. no. 15, 2022, doi: 10.1186/s13677-021-00268-0.
[24] Z. Li, X. Yan, and H. Chen, "Adaptive Q-learning for load balancing in cloud data centers," Neural Comput. Appl., vol. 34, pp. 12311–12325, 2022, doi: 10.1007/s00521-021-06106-4.
[25] F. Lin, T. Wu, and J. Chen, "Pareto-optimal resource mapping using improved NSGA-II," Soft Comput., vol. 27, pp. 2523–2538, 2023, doi: 10.1007/s00500-022-07112-9.
[26] H. Malik, M. Tariq, and S. Hassan, "Energy and carbon-aware scheduling framework for cloud computing," Sustain. Comput. Inform. Syst., vol. 35, Art. no. 100745, 2023, doi: 10.1016/j.suscom.2022.100745.
[27] N. Malik, D. Singh, and S. K. Sood, "Carbon-aware task scheduling in green cloud environments: A deep learning approach," Sustain. Comput. Inform. Syst., vol. 39, Art. no. 100876, 2023, doi: 10.1016/j.suscom.2023.100876.
[28] V. Mehta and H. Rai, "Dueling DQN and NSGA-II based hybrid cloud task scheduling," Appl. Soft Comput., vol. 136, Art. no. 110083, 2023, doi: 10.1016/j.asoc.2022.110083.
[29] X. Niu, G. Li, and H. Wang, "Scheduling optimization with deep reinforcement learning in green cloud environments," Future Gener. Comput. Syst., vol. 139, pp. 108–120, 2023, doi: 10.1016/j.future.2022.09.006.
[30] Y. Pan, F. Liu, Q. Zhang, and J. Xu, "Carbon-aware energy management for cloud data centers," IEEE Trans. Cloud Comput., vol. 8, no. 2, pp. 510–523, 2020, doi: 10.1109/TCC.2020.2980354.
[31] D. Patel, A. Agarwal, and A. Verma, "Hybrid GA-RL model for workflow scheduling in cloud," J. Parallel Distrib. Comput., vol. 152, pp. 89–102, 2021, doi: 10.1016/j.jpdc.2021.02.002.
[32] Y. Ren, L. Wang, and X. Wu, "Energy-efficient cloud resource scheduling using MIP," Comput. Stand. Interfaces, vol. 76, Art. no. 103501, 2021, doi: 10.1016/j.csi.2021.103501.
[33] P. Roy, S. Biswas, and R. Sinha, "Multi-objective NSGA-II optimization for cloud VM assignment," J. Cloud Comput., vol. 12, no. 1, Art. no. 33, 2023.
[34] S. Sharma and N. Gill, "Performance evaluation of heuristic resource allocation algorithms in IaaS," J. Cloud Comput., vol. 10, no. 1, Art. no. 11, 2021.
[35] L. Silva, P. Costa, and J. Fonseca, "CloudSimEnergy: A simulator for energy-efficient cloud environments," Simul. Model. Pract. Theory, vol. 108, Art. no. 102228, 2021.
[36] Y. Song, D. Liu, and F. Zhang, "Carbon-intensity-aware task migration in federated clouds," IEEE Trans. Sustain. Comput., vol. 8, no. 1, pp. 89–101, 2023.
[37] Y. Tan and S. Goh, "Metaheuristic scheduling algorithms: A review of applications in cloud," Expert Syst. Appl., vol. 165, Art. no. 113851, 2021.
[38] A. Tariq, R. Mehmood, and M. Khan, "Comparative analysis of FCFS and SJF scheduling in cloud," Int. J. Adv. Comput. Sci., vol. 11, no. 2, pp. 46–52, 2020.
[39] M. A. Tariq, H. Abbas, and M. A. Tahir, "Performance evaluation of scheduling algorithms in cloud computing," Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 4, pp. 314–319, 2020, doi: 10.14569/IJACSA.2020.0110438.
[40] C. Wang, X. Hu, and Z. Yin, "Hybrid reinforcement learning and PSO for adaptive VM placement in multi-clouds," Future Gener. Comput. Syst., vol. 143, pp. 367–380, 2023, doi: 10.1016/j.future.2023.01.019.

10.48047/jocaaa.2025.34.08.1

[41] F. Wang, L. He, and X. Li, "A hybrid RL-GA scheduler for dynamic cloud computing," Inf. Sci., vol. 627, pp. 145–161, 2023.

[42] J. Wang and M. Liu, "Energy and carbon-aware task allocation using deep RL," IEEE Access, vol. 9, pp. 44573–44585, 2021.

[43] X. Yuan, Y. Zhang, and Q. Hu, "NSGA-II based multi-objective resource optimization," J. Supercomput., vol. 77, no. 8, pp. 8912–8935, 2021.

[44] Y. Yuan, Z. Liu, and J. Xue, "Multi-objective scheduling using NSGA-II in heterogeneous cloud environments," J. Cloud Comput., vol. 10, no. 1, pp. 1–17, 2021, doi: 10.1186/s13677-021-00241-x.

[45] Q. Zhang and Y. Luo, "Deep reinforcement learning in real-time cloud scheduling," Inf. Fusion, vol. 68, pp. 81–92, 2021.

[46] W. Zhang, F. Chen, and J. Zhao, "Green cloud scheduling using carbon signal integration," J. Syst. Archit., vol. 116, Art. no. 102016, 2021.

[47] Y. Zhang, L. Chen, and L. Sun, "Carbon-aware scheduling for cloud environments: A review," ACM Comput. Surv., vol. 54, no. 3, pp. 1–34, 2021, doi: 10.1145/3439724.

[48] Y. Zhou, Z. Huang, and L. Wang, "Analysis of heuristic scheduling techniques in cloud," ACM Trans. Internet Technol., vol. 22, no. 1, pp. 1–24, 2022.