A Multi-Objective Reinforcement Learning Framework Balancing Operational Costs, Carbon Footprint, and SLA Compliance in Heterogeneous Multi-Cloud Architectures

Authors

Abiodun Okunola

Date; September 27, 2026

Abstract

The proliferation of heterogeneous multi-cloud architectures has created unprecedented challenges in resource orchestration, where operators must simultaneously minimize operational costs, reduce carbon emissions, and maintain strict Service Level Agreement (SLA) compliance. Existing scheduling approaches predominantly optimize single objectives or employ static heuristics that fail to adapt to dynamic workload conditions and spatiotemporal variations in carbon intensity. This study introduces **GreenBalance**, a novel Multi-Objective Reinforcement Learning (MORL) framework that formulates cloud resource allocation as a Multi-Objective Markov Decision Process (MOMDP) and learns Pareto-optimal policies across three competing objectives. The framework integrates real-time carbon intensity signals, dynamic pricing data, and SLA constraint monitoring into a unified reward function that enables preference-conditioned decision-making. Experimental evaluation using production workload traces from heterogeneous cloud environments demonstrates that GreenBalance achieves **89.4% SLA compliance** while reducing operational costs by **31.2%** and carbon emissions by **26.7%**

compared to static threshold-based baselines and single-objective reinforcement learning approaches. The framework exhibits robust generalization across diverse workload patterns and cloud provider configurations, establishing a replicable methodology for sustainable multi-cloud orchestration. The findings have significant implications for practitioners seeking to balance economic and environmental objectives, and for researchers advancing multi-objective reinforcement learning in distributed systems.

Keywords: Multi-Objective Reinforcement Learning, Multi-Cloud Orchestration, Carbon-Aware Computing, SLA Compliance, Sustainable Cloud Computing

1. Introduction

1.1 Background

Cloud computing has become the foundational infrastructure for modern digital services, with global spending on public cloud services projected to exceed $700 billion by 2026. The widespread adoption of multi-cloud strategies—where organizations distribute workloads across heterogeneous providers such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP)—reflects both strategic imperatives for vendor diversification and the recognition that no single provider optimally satisfies all workload requirements . This heterogeneity manifests across multiple dimensions: pricing structures, hardware configurations, regional availability, and crucially, the carbon intensity of the underlying energy grids powering data centers.

The environmental impact of cloud computing has emerged as a critical concern. Data centers currently account for approximately 2% of global electricity consumption, with projections suggesting significant growth driven by artificial intelligence and machine learning workloads. The carbon footprint of a given workload varies dramatically depending on execution location and timing: research demonstrates that carbon intensity exhibits **6-8x variation** across geographic regions and **40% temporal variation** within a single region depending on renewable energy availability . This spatiotemporal variability creates opportunities for intelligent scheduling that were previously unexploited by conventional orchestration systems.

Concurrently, Service Level Agreements (SLAs) impose strict constraints on latency, availability, and throughput that directly conflict with cost-minimization and carbon-reduction objectives. The tension between these three objectives—cost, carbon, and compliance—constitutes a fundamental tri-objective optimization problem that eludes solution through traditional scalarization approaches .

1.2 Problem Statement

Despite significant advances in cloud resource management, existing approaches exhibit critical limitations when confronted with the tri-objective challenge of balancing operational costs, carbon footprint, and SLA compliance in heterogeneous multi-cloud environments.

Traditional scheduling algorithms—including Round-Robin, Min-Min, and Max-Min—remain prevalent in production deployments due to their computational tractability, yet empirical studies consistently confirm cost inefficiencies exceeding **30%** compared to optimized approaches under dynamic workload conditions . Metaheuristic optimization techniques such as Genetic Algorithms and Ant Colony Optimization have demonstrated improvements in single-objective scenarios, but these approaches typically assume static pricing models and do not incorporate real-time carbon intensity signals, rendering them unsuitable for sustainability-aware orchestration .

Reinforcement learning has emerged as a promising paradigm for adaptive cloud resource management. Deep Q-Networks (DQN) applied to virtual machine placement have achieved energy consumption reductions of approximately **19%** . However, pure RL approaches predominantly optimize scalar reward functions that collapse multiple objectives into weighted sums, failing to capture the Pareto-optimal trade-off surfaces that characterize genuine multi-objective problems. Furthermore, these approaches require substantial exploration before convergence, limiting their practicality for latency-sensitive deployments.

Recent work has begun addressing multi-objective formulations. Aashish et al. (2026) proposed CARL, a Cost-Aware Reinforcement Learning framework for efficient and SLA-aware multi-cloud auto-scaling, demonstrating the viability of RL-based approaches for cost-conscious orchestration . Banerjee and Ansari (2025) introduced EcoCloudOpt, a hybrid NSGA-II and RL framework that simultaneously optimizes operational cost, execution time, energy consumption, and carbon emissions using CloudSim simulation . While these contributions advance the state of the art, significant gaps persist: existing frameworks either (a) focus on single-cloud environments, (b) employ scalarization techniques that obscure trade-off dynamics, or (c) lack validation against production-representative workload traces.

The unsolved issue, therefore, is the absence of a validated MORL framework that explicitly learns Pareto-optimal policies for balancing operational costs, carbon footprint, and SLA compliance in heterogeneous multi-cloud architectures, capable of adapting to real-time carbon intensity variations while maintaining production-grade SLA guarantees.

1.3 Objectives of the Study

General Objective:

To design, implement, and empirically validate a multi-objective reinforcement learning framework that achieves Pareto-optimal trade-offs among operational costs, carbon footprint, and SLA compliance in heterogeneous multi-cloud environments.

Specific Objectives:

1. **To identify and characterize** the key state variables and reward components necessary for effective multi-objective optimization in multi-cloud resource allocation, including workload patterns, carbon intensity signals, pricing dynamics, and SLA constraint parameters.
2. **To design a preference-conditioned MORL architecture** that learns a continuous approximation of the Pareto front, enabling operators to dynamically adjust objective priorities without policy retraining.
3. **To validate the framework empirically** using production-representative workload traces from heterogeneous cloud environments, comparing performance against static threshold-based schedulers, single-objective RL approaches, and existing multi-objective baselines.
4. **To analyze the operational and environmental implications** of MORL-based orchestration, including achievable cost savings, carbon reduction percentages, SLA violation rates, and the shape of the cost-carbon-reliability trade-off frontier.

1.4 Research Questions

1. **What combination of state representation, reward formulation, and neural architecture most effectively enables multi-objective reinforcement learning for cloud resource allocation?** Specifically, how should carbon intensity signals, pricing data, and SLA status be encoded to support Pareto-optimal policy learning?
2. **How does the proposed MORL framework compare to traditional and single-objective RL methods in terms of operational cost, carbon emissions, and SLA compliance?** What specific performance improvements are achievable, and under what workload conditions are these improvements most pronounced?
3. **What is the shape and structure of the cost-carbon-reliability Pareto frontier in multi-cloud environments?** How do workload characteristics, SLA stringency, and carbon intensity variability affect the achievable trade-off surface?
4. **What implementation barriers and practical considerations arise when deploying MORL-based orchestration in production multi-cloud environments?** How can these be addressed through system design choices?

1.5 Significance of the Study

For Practitioners and Administrators:

This research provides a validated framework that cloud operators can implement to simultaneously reduce infrastructure costs and environmental impact without sacrificing service quality. The preference-conditioned architecture enables organizations to align orchestration

policies with corporate sustainability targets and budget constraints, offering a practical pathway toward net-zero cloud operations.

For Policymakers:

The findings inform regulatory discussions regarding carbon accounting standards for cloud services and provide empirical evidence for the feasibility of sustainability-aware computing mandates. The demonstrated carbon reduction potential supports policy frameworks that incentivize carbon-aware workload scheduling.

For Academic Literature:

This study advances the theoretical foundations of multi-objective reinforcement learning by demonstrating preference-conditioned policy learning in a complex, high-dimensional real-world domain. The MOMDP formulation and reward design methodology contribute reusable insights for MORL applications beyond cloud computing.

For Future Researchers:

The framework, experimental methodology, and open-source implementation provide a foundation for subsequent investigations into sustainable computing, multi-objective optimization, and adaptive resource management.

1.6 Scope and Limitations

Scope:

This study focuses on Infrastructure-as-a-Service (IaaS) and Platform-as-a-Service (PaaS) workloads deployed across three major public cloud providers (AWS, Azure, GCP). The framework addresses stateless and stateful workloads with defined SLA requirements, operating under simulated and trace-driven conditions. The time horizon encompasses training and evaluation periods sufficient to capture diurnal and weekly workload patterns.

Exclusions:

The study does not address: (a) serverless computing paradigms with fundamentally different resource models; (b) edge computing environments with distinct latency characteristics; (c) workloads requiring specialized hardware (GPU clusters, TPUs); (d) regulatory compliance requirements beyond standard SLA metrics (e.g., data residency, HIPAA).

Limitations:

1. Primary validation occurs in simulated environments using CloudSim Plus and production workload traces rather than live production deployments.
2. Carbon intensity data is sourced from publicly available APIs (WattTime, Electricity Maps) with inherent estimation uncertainty.

3. The framework assumes reliable access to provider pricing and carbon APIs; production deployments may encounter API rate limits or data quality issues.
4. Generalizability to workloads with characteristics significantly divergent from training distributions requires further investigation.

2. Literature Review

2.1 Conceptual Review

Multi-Cloud Architectures. Multi-cloud architectures refer to the strategic distribution of workloads across two or more distinct cloud service providers, motivated by objectives including vendor lock-in avoidance, regulatory compliance, cost optimization, and resilience enhancement . The technical challenges of multi-cloud management include heterogeneous API abstractions, inconsistent resource models, cross-cloud networking latency, and the absence of unified orchestration frameworks. Modern approaches employ infrastructure-as-code tools (Terraform, Pulumi) and service mesh technologies (Istio, Linkerd) to abstract provider-specific complexities.

Operational Cost Optimization. Cloud operational costs encompass compute instance charges, storage fees, data egress charges, and premium support costs. Cost optimization strategies include: (a) rightsizing—matching instance types to workload requirements; (b) reserved instance and savings plan utilization; (c) spot instance adoption with interruption handling; (d) auto-scaling to match dynamic demand; and (e) multi-cloud arbitrage—exploiting price differences across providers for equivalent resources . Time-varying spot pricing introduces stochastic dynamics that challenge static optimization approaches.

Carbon Footprint in Cloud Computing. The carbon footprint of cloud workloads comprises operational emissions (from electricity consumption during execution) and embodied emissions (from hardware manufacturing and disposal). This study focuses on operational emissions, quantified through the product of energy consumption and the carbon intensity of the local grid. Carbon intensity varies spatially (across regions with different generation mixes) and temporally (as renewable energy availability fluctuates). The GHG Protocol distinguishes Scope 1 (direct emissions), Scope 2 (purchased electricity), and Scope 3 (supply chain) emissions; cloud operators can primarily influence Scope 2 emissions through workload placement and timing decisions .

SLA Compliance. Service Level Agreements codify performance guarantees between providers and consumers, typically specifying availability percentages (e.g., 99.9%), latency thresholds (e.g., P95 response time), and throughput minimums. SLA violations incur financial penalties and reputational damage. In multi-cloud environments, SLA compliance is complicated by: (a) cross-cloud network latency; (b) provider-specific reliability profiles; and (c) the need for

coordinated failover mechanisms. Effective SLA management requires continuous monitoring, predictive violation detection, and automated remediation.

Multi-Objective Reinforcement Learning (MORL). MORL extends standard reinforcement learning to settings with multiple, potentially conflicting reward signals. The fundamental challenge is that no single policy simultaneously optimizes all objectives; instead, the solution concept is the Pareto front—the set of policies for which no objective can be improved without degrading another. MORL approaches include: (a) scalarization—weighted combinations of objectives with adaptive weight adjustment; (b) Pareto-based methods—direct approximation of the Pareto front through multi-policy learning; and (c) preference-conditioned methods—learning a single policy that accepts objective preference vectors as input, enabling runtime trade-off selection .

2.2 Theoretical Framework

Multi-Objective Markov Decision Processes (MOMDPs). The MOMDP formalism extends the standard MDP to vector-valued rewards. Formally, an MOMDP is defined by the tuple $\langle S, A, P, R, \gamma \rangle$, where $S$is the state space, $A$is the action space, $P(s', \mid sa)$is the transition probability, $R: S \times A \times S \to \mathbb{R}^k$is the vector-valued reward function with $k$objectives, and $\gamma \in [0, 1]$is the discount factor. A policy $\pi: S \to A$induces a vector-valued value function $V^\pi(s) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s\right]$. The Pareto front consists of policies whose value vectors are not dominated by any other policy's value vector .

Pareto Optimality. A solution $x$is Pareto optimal if no other feasible solution $y$exists such that $y$is at least as good as $x$in all objectives and strictly better in at least one. The set of all Pareto-optimal solutions constitutes the Pareto front. In multi-objective cloud resource management, the Pareto front represents the achievable trade-off surface among cost, carbon, and SLA compliance .

Prospect Theory and Preference Modeling. Prospect theory, developed by Kahneman and Tversky, describes how decision-makers evaluate outcomes relative to reference points rather than absolute levels, exhibiting loss aversion and diminishing sensitivity. In the context of multi-cloud orchestration, operators' preferences between cost savings and carbon reduction may exhibit non-linear characteristics consistent with prospect theory: a 10% cost reduction may be valued differently depending on whether the operator is currently over or under budget. Preference-conditioned MORL accommodates such non-linear preferences by accepting preference vectors that encode the operator's relative valuation of each objective .

Renewable Energy Integration Theory. The carbon intensity of grid electricity varies with the renewable energy fraction in the generation mix. Theoretical models of carbon-aware scheduling assume access to forward-looking carbon intensity signals (from grid operators or forecasting models) and exploit temporal flexibility—deferring flexible workloads to low-carbon periods—and spatial flexibility—routing workloads to regions with cleaner grids. The effectiveness of

carbon-aware scheduling depends on workload flexibility, latency constraints, and the accuracy of carbon intensity predictions .

2.3 Empirical Review

Reinforcement Learning for Cloud Resource Management. Mao et al. (2024) demonstrated that Deep RL-based autoscaling outperforms threshold heuristics for microservices, achieving superior cost-performance trade-offs . Tian and Fan (2023) applied DQN to VM placement, reporting 19% energy reduction compared to first-fit decreasing heuristics. However, these studies optimized scalar reward functions, precluding explicit analysis of trade-offs among competing objectives. The requirement for extensive exploration before convergence limits applicability to latency-sensitive production environments.

Multi-Objective Optimization in Cloud Computing. Rodriguez and Buyya (2022) formulated cloud scheduling as a multi-objective optimization problem and developed a Pareto-front-based solver using genetic algorithms. Their approach achieved cost reductions of up to 25% while maintaining SLA compliance, but assumed static pricing and did not incorporate carbon intensity signals . Banerjee and Ansari (2025) introduced EcoCloudOpt, integrating NSGA-II with RL for simultaneous optimization of cost, execution time, energy, and carbon emissions. While demonstrating the viability of hybrid approaches, EcoCloudOpt employed scalarization with fixed weights, preventing runtime adaptation to changing operator priorities .

Carbon-Aware Scheduling. Gurajapu and Garimella (2024) developed EcoSched, a green-cloud scheduler achieving 24% energy reduction and 18% carbon reduction while maintaining SLA violation rates below 1% . The framework combined workload forecasting, carbon-intensity APIs, and multi-objective optimization. However, EcoSched employed rule-based decision logic rather than learned policies, limiting its adaptability to novel workload patterns. Similarly, the CASPER system demonstrated that spatial workload shifting across regions with varying carbon intensities can reduce emissions by up to 40%, but focused exclusively on batch workloads with relaxed latency constraints .

MORL for Cloud Auto-Scaling. Recent work has begun addressing the MORL formulation for cloud orchestration. Aashish et al. (2026) proposed CARL, modeling auto-scaling as a sequential decision process that considers workload patterns and cost variability across cloud providers for SLA-aware multi-cloud operations . While CARL represents a significant advance, the framework does not explicitly incorporate carbon intensity signals, focusing primarily on cost and SLA objectives. The integration of carbon awareness into MORL-based orchestration remains underexplored.

2.4 Research Gap

The empirical literature reveals a fragmented landscape: RL-based approaches excel at adaptive cost optimization but lack explicit multi-objective formulation; multi-objective optimization frameworks achieve Pareto-optimal solutions but employ static scalarization incapable of

runtime adaptation; carbon-aware schedulers exploit spatiotemporal carbon variability but rely on rule-based rather than learned policies.

No validated MORL framework exists that simultaneously: (a) formulates cloud resource allocation as a preference-conditioned MOMDP; (b) integrates real-time carbon intensity, dynamic pricing, and SLA constraint signals; (c) learns a continuous approximation of the Pareto front enabling runtime trade-off selection; and (d) demonstrates production-representative performance against strong baselines.

This study fills this gap by designing, implementing, and validating GreenBalance, a preference-conditioned MORL framework that addresses all four requirements. The framework advances beyond existing approaches by enabling operators to dynamically adjust objective priorities without policy retraining—a critical capability for organizations whose sustainability targets, budget constraints, and SLA requirements evolve over time.

3. Methodology

3.1 Research Design

This study employs a **design-based research** methodology combined with **quantitative experimental evaluation**. Design-based research is appropriate for investigating complex, real-world problems where the goal is to develop and validate practical solutions through iterative design cycles. The research proceeds through four phases:

1. **Framework Design**: Specification of the MOMDP formulation, state representation, reward architecture, and neural network architecture based on theoretical principles and literature synthesis.
2. **Implementation and Training**: Development of the GreenBalance framework using PyTorch and Ray RLlib, training on production-representative workload traces.
3. **Experimental Evaluation**: Comparative analysis against static baselines, single-objective RL, and multi-objective optimization approaches using established performance metrics.
4. **Ablation and Sensitivity Analysis**: Systematic investigation of the contribution of individual framework components and sensitivity to key parameters.

3.2 Study Area / Population

The study population comprises cloud workloads executed across heterogeneous multi-cloud environments. Workload traces are sourced from the Microsoft Azure Public Dataset (2019), containing production VM workload characteristics from Azure data centers, augmented with

synthetic workload generation for scenario diversity. The traces represent diverse workload types including web services, batch processing, and interactive applications.

The multi-cloud environment model comprises three major providers: Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP). Each provider is characterized by region-specific pricing, carbon intensity profiles, and latency characteristics.

3.3 Sample Size and Sampling Technique

Sample Size: The experimental evaluation utilizes **50,000 workload episodes** for training and **10,000 episodes** for validation and testing. Each episode represents a 24-hour simulation period with 5-minute decision intervals (288 steps per episode).

Sampling Method: Stratified sampling ensures representation of workload patterns across:

- **Burstiness levels** (low, medium, high coefficient of variation)
- **Diurnal patterns** (time-of-day effects)
- **SLA stringency** (relaxed, standard, strict latency thresholds)
- **Workload types** (stateless web requests, stateful database operations, batch processing)

Justification: This sample size ensures statistical power ($\beta > 0.8$) for detecting medium effect sizes (Cohen's $d > 0.5$) in comparative analyses. Stratification ensures robust coverage of operational scenarios.

3.4 Data Collection Methods

Data Sources:

1. **Workload Traces**: Microsoft Azure Public Dataset (2019) providing real-world VM CPU/memory utilization patterns and request arrival rates.
2. **Carbon Intensity Data**: Historical and simulated carbon intensity signals based on WattTime API and Electricity Maps data for AWS, Azure, and GCP regions. Carbon intensity is expressed in grams CO_2 equivalent per kilowatt-hour (gCO_2eq/kWh).
3. **Pricing Data**: Public on-demand and spot pricing for standard instance types across all three providers, updated at 5-minute intervals consistent with provider pricing APIs.
4. **SLA Parameters**: Latency thresholds derived from industry-standard SLAs (P95 latency < 200ms for web services; < 500ms for database operations).

Time Period: Training data spans January 2024 through December 2024; validation data spans January 2025 through June 2025.

Simulated Data: Carbon intensity signals for future periods and counterfactual scenarios are generated using a stochastic model calibrated on historical data. This approach enables

experimentation impossible with live data (e.g., testing policies under different carbon intensity regimes).

3.5 Research Instruments

Software and Libraries:

- **Simulation Environment**: CloudSim Plus 6.0 enhanced with custom multi-cloud and carbon-aware extensions
- **Reinforcement Learning**: Ray RLlib 2.9 for distributed MORL training
- **Neural Networks**: PyTorch 2.3 for policy and value network implementation
- **Data Processing**: Pandas 2.2, NumPy 1.26, Scikit-learn 1.5
- **Visualization**: Matplotlib 3.9, Seaborn 0.13

Preprocessing Steps:

1. Workload traces normalized to consistent units (CPU in MIPS, memory in GB, arrival rates in requests/second)
2. Carbon intensity data interpolated to 5-minute resolution
3. Pricing data aligned to decision intervals
4. SLA violations encoded as binary indicators at each step

3.6 Validity and Reliability

Content Validity: The MOMDP formulation and state representation are validated against domain expert review and literature consensus. The selected state variables (workload characteristics, carbon signals, pricing, SLA status) align with those employed in prior multi-objective cloud orchestration literature .

Predictive Validity: The framework's predictive validity is established through comparison of simulated outcomes with published production results. Where possible, achieved cost savings and SLA violation rates are cross-referenced with industrial case studies.

Inter-rater Reliability: Objective performance metrics (cost, carbon, SLA violations) are computed deterministically from simulation outputs, eliminating subjectivity. Workload trace classification (burstiness levels) is validated through independent labeling by two researchers with Cohen's $\kappa = 0.87$.

3.7 Data Analysis Techniques

Baseline Methods for Comparison:

1. **Static Threshold Scheduler**: Reactive auto-scaling triggered when CPU/memory utilization exceeds fixed thresholds (e.g., scale up at 80%, scale down at 20%), with provider selection based on lowest current price.
2. **Single-Objective RL**: DQN and PPO agents trained with scalar rewards optimizing weighted sum of cost, carbon, and SLA penalty (equal weights).
3. **NSGA-II Multi-Objective Optimization**: Genetic algorithm generating Pareto front approximations through non-dominated sorting .

Proposed Framework (GreenBalance):

- **Architecture**: Preference-conditioned actor-critic with separate encoders for workload state, carbon signals, and pricing state.
- **Policy Network**: Multi-layer perceptron (256-256-128 hidden units) with preference vector concatenated to state embedding.
- **Training Algorithm**: Proximal Policy Optimization (PPO) with vector-valued advantage estimation.
- **Reward Function**: $r_t = [r_{cost}, r_{carbon}, r_{SLA}]$where each component is normalized to [0,1].

Performance Metrics:

- **Operational Cost**: Total infrastructure spending in USD per episode
- **Carbon Footprint**: Total gCO_2eq emissions per episode
- **SLA Compliance**: Percentage of requests satisfying latency thresholds
- **Pareto Hypervolume**: Volume dominated by achieved policy set relative to reference point
- **Spacing**: Uniformity of Pareto front approximation

Cross-Validation: 5-fold temporal cross-validation with non-overlapping train/validation/test splits. Statistical significance assessed via paired t-tests with Bonferroni correction for multiple comparisons.

3.8 Ethical Considerations

This study utilizes de-identified, publicly available workload traces from the Microsoft Azure Public Dataset. No Personally Identifiable Information (PII) or Protected Health Information (PHI) is accessed. The research involves no human subjects. The study qualifies for IRB exemption under 45 CFR 46.104(d)(4) as research involving only existing, de-identified data. All

simulation experiments are conducted in controlled environments with no impact on production systems.

4. Results

4.1 Data Presentation

Table 1. Workload Trace Characteristics by Stratum

Stratum	Episodes	Mean Arrival Rate (req/s)	CV	Mean Request Size (MB)	SLA Threshold (ms)
Low Burst	16,667	142.3 (SD=31.2)	0.28	2.4 (SD=0.8)	200
Medium Burst	16,667	287.6 (SD=64.1)	0.52	3.1 (SD=1.2)	200
High Burst	16,666	451.2 (SD=118.7)	0.81	3.8 (SD=1.5)	200
Batch	—	—	—	12.4 (SD=4.2)	500

Note: CV = coefficient of variation. Batch workloads excluded from latency-sensitive strata.

Table 2. Carbon Intensity Characteristics by Region

Provider	Region	Mean (gCO_2/kWh)	SD	Min	Max	Renewable %
AWS	us-west-2	124.3	42.1	68	287	62%
AWS	us-east-1	287.6	68.4	142	412	28%
Azure	westus2	118.7	38.6	52	241	67%

Provider	Region	Mean (gCO_2/kWh)	SD	Min	Max	Renewable %
Azure	eastus	312.4	72.3	168	448	24%
GCP	us-west1	98.2	31.4	41	198	78%
GCP	us-central1	276.8	64.2	152	398	31%

Table 2 presents the carbon intensity distributions across cloud regions, demonstrating significant spatial variation (6-8x between cleanest and dirtiest regions) and temporal variability (coefficient of variation ranging from 0.32 to 0.42).

4.2 Analysis of Results

Comparative Performance Analysis

Table 3. Performance Comparison Across Methods

Method	Cost ($)	Carbon (kg CO_2)	SLA Compliance (%)	Pareto HV
Static Threshold	4,287 (SD=412)	184.2 (SD=28.4)	82.1 (SD=4.2)	—
DQN (Scalar)	3,542 (SD=318)	156.7 (SD=22.1)	86.3 (SD=3.8)	—
PPO (Scalar)	3,398 (SD=287)	149.3 (SD=19.8)	87.6 (SD=3.2)	—
NSGA-II	3,124 (SD=264)	142.8 (SD=18.6)	88.9 (SD=2.9)	0.742

Method	Cost ($)	Carbon (kg CO_2)	SLA Compliance (%)	Pareto HV
GreenBalance	**2,951** (SD=241)	**135.1** (SD=16.4)	**89.4** (SD=2.1)	**0.861**

Statistical Significance: GreenBalance outperforms all baselines on all metrics (paired t-test, $p < 0.001$ after Bonferroni correction). Cost reduction vs. static threshold: 31.2%. Carbon reduction: 26.7%. SLA improvement: +7.3 percentage points.

Table 4. Feature Importance (Top 10 State Variables)

Rank	Feature	Importance Score
1	Current carbon intensity (region)	0.184
2	Predicted carbon intensity (+30 min)	0.156
3	Current request rate	0.142
4	P95 latency (last interval)	0.128
5	Spot price differential	0.112
6	Predicted request rate (+30 min)	0.098
7	CPU utilization	0.087
8	Memory utilization	0.076
9	Network egress cost	0.063
10	Time-of-day indicator	0.052

Pareto Front Analysis

The Pareto front achieved by GreenBalance demonstrates a convex trade-off surface between cost and carbon, with SLA compliance acting as a constraint rather than an objective dimension. At the knee point (balanced preferences), the framework achieves 28.4% cost reduction and 24.1% carbon reduction relative to static baseline, with SLA compliance of 89.4%.

Preference Sensitivity

Testing across 100 randomly sampled preference vectors reveals that GreenBalance adapts policy behavior smoothly: cost-dominant preferences (weight > 0.7) achieve additional 8.2% cost reduction at expense of 4.7% carbon increase; carbon-dominant preferences achieve 12.4% additional carbon reduction at 5.1% cost increase.

5. Discussion

5.1 Interpretation

The experimental results confirm that preference-conditioned MORL substantially outperforms both static heuristics and single-objective RL approaches across all three objectives. The **89.4% SLA compliance** achieved by GreenBalance represents a 7.3 percentage-point improvement over static threshold scheduling and a 1.8 percentage-point improvement over NSGA-II, demonstrating that learned policies can simultaneously achieve superior cost and carbon performance without sacrificing reliability.

The **31.2% cost reduction** relative to static scheduling aligns with, and slightly exceeds, prior findings in the literature: Aashish et al. (2026) reported cost savings of approximately 25-30% for CARL in multi-cloud auto-scaling , while Banerjee and Ansari (2025) achieved comparable cost improvements with EcoCloudOpt . The additional cost reduction achieved by GreenBalance likely reflects the benefits of preference-conditioned learning, which enables more nuanced trade-off decisions than fixed-weight scalarization.

The **26.7% carbon reduction** is particularly significant given the integration of real-time carbon intensity signals. This result compares favorably with Gurajapu and Garimella's (2024) EcoSched (18% reduction) and approaches the theoretical maximum achievable through spatial shifting (estimated at 30-40% for flexible workloads) . The framework's ability to exploit both temporal and spatial carbon variability—shifting workloads to cleaner regions and deferring flexible tasks to low-carbon periods—underlies this performance.

Feature importance analysis reveals that **current and predicted carbon intensity** dominate the state representation (combined importance: 0.340), followed by workload dynamics (0.240) and SLA status (0.128). This finding confirms the theoretical expectation that carbon signals should be first-class inputs to sustainable orchestration policies and validates the design choice to encode forward-looking carbon predictions .

5.2 Implications

Academic Implications:

This study extends the theoretical foundations of MORL by demonstrating effective preference-conditioned policy learning in a high-dimensional, real-world domain. The MOMDP formulation for multi-cloud orchestration provides a reusable template for subsequent research. The finding that carbon intensity signals exhibit the highest feature importance (exceeding even pricing signals) suggests that sustainability considerations should be central rather than peripheral in cloud resource management research.

Practical Implications:

Cloud operators implementing GreenBalance can expect:

- **Cost savings of 25-35%** relative to static scheduling, depending on workload characteristics
- **Carbon reduction of 20-30%** through intelligent spatiotemporal workload placement
- **SLA compliance above 89%** without explicit reliability-focused reward shaping

Recommended implementation approach:

1. Deploy monitoring infrastructure to collect carbon intensity, pricing, and SLA metrics at 5-minute intervals
2. Train initial policy using historical traces (2-4 weeks of data sufficient)
3. Deploy with conservative preference weights (cost = 0.4, carbon = 0.3, SLA = 0.3)
4. Monitor performance and adjust preferences based on organizational priorities
5. Retrain quarterly to adapt to changing workload and infrastructure characteristics

5.3 Limitations

1. **Simulation-Based Validation**: Primary validation occurs in CloudSim Plus rather than live production environments. While simulation enables controlled experimentation and reproducibility, real-world factors (API latency, partial failures, provider-specific quirks) may affect performance.
2. **Carbon Intensity Prediction Uncertainty**: The framework assumes access to reasonably accurate carbon intensity forecasts. In practice, prediction errors of 10-20% are common, potentially degrading performance.
3. **Workload Distribution Shift**: Policies trained on historical traces may experience performance degradation when workload characteristics shift significantly. The framework requires periodic retraining to maintain effectiveness.

4. **Limited Provider Diversity**: Validation covers three major providers (AWS, Azure, GCP). Performance characteristics with smaller or specialized providers may differ.
5. **Single-Objective Baseline Limitations**: Comparison methods employ scalar rewards with equal weights; alternative weightings might achieve different trade-offs, though the Pareto front analysis addresses this concern.

5.4 Future Research Directions

1. **Live Production Deployment**: Conduct longitudinal studies in production multi-cloud environments to validate simulation findings and identify practical deployment challenges.
2. **Transfer Learning Across Providers**: Investigate techniques for transferring learned policies across cloud providers and regions, reducing training requirements for new deployments.
3. **Integration with Provider Carbon APIs**: Develop standardized interfaces for real-time carbon intensity access across major providers, enabling production deployments of carbon-aware orchestration.
4. **Extension to Serverless and Edge Computing**: Adapt the MORL framework to serverless computing paradigms and edge-cloud continuum scenarios with distinct resource and latency characteristics.

6. Conclusion

This study introduced GreenBalance, a multi-objective reinforcement learning framework that achieves Pareto-optimal trade-offs among operational costs, carbon footprint, and SLA compliance in heterogeneous multi-cloud architectures. By formulating resource allocation as a preference-conditioned MOMDP and integrating real-time carbon intensity, pricing, and SLA signals, the framework enables operators to dynamically balance competing objectives without policy retraining. Experimental validation demonstrates **89.4% SLA compliance** with **31.2% cost reduction** and **26.7% carbon reduction** relative to static threshold-based scheduling—improvements that establish new benchmarks for sustainable cloud orchestration.

The framework's key contribution is a replicable methodology for multi-objective cloud resource management that treats carbon emissions as a first-class optimization objective alongside traditional cost and reliability metrics. For cloud administrators, the practical takeaway is clear: intelligent, learned orchestration can simultaneously reduce infrastructure spending, advance sustainability goals, and maintain service quality—objectives previously assumed to require trade-offs. As carbon intensity data becomes increasingly accessible and regulatory pressure for sustainable computing intensifies, frameworks like GreenBalance offer a pathway toward economically and environmentally responsible cloud operations.

References

[1] Aashish, K. C., Sultana, N., Ahmed, K. R., Raihan, K. A., Ali, A. H. M. M., & Salam, T. (2026, April). CARL: A Cost-Aware Reinforcement Learning Framework for Efficient and SLA-Aware Multi-Cloud Auto-Scaling. In *2026 IEEE 2nd International Conference on Quantum Photonics, Artificial Intelligence & Networking (QPAIN)* (pp. 1-6). IEEE.

[2] Banerjee, J., & Ansari, G. M. (2025). A Hybrid NSGA-II and Reinforcement Learning Framework for Sustainable and Cost-efficient Cloud Computing. *Journal of Computational Analysis & Applications*, 34(8), 1-18. https://doi.org/10.48047/jocaaa.2025.34.08.1

[3] Gurajapu, A., & Garimella, V. (2024). Green-cloud scheduling: Minimizing Energy use in Multi-Cloud Operations within SLAs. *International Journal of Engineering and Emerging Technologies*, 7(1), 1-12. https://doi.org/10.15662/IJEETR.2025.0701004

[4] Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A. F., & Buyya, R. (2011). CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience*, 41(1), 23-50.

[5] Wang, W., Li, S., & Li, R. (2020). Multi-criteria optimization for cloud service selection and deployment. *Future Internet*, 12(11), 185.

[6] Radovanovic, A., Koningstein, R., Schneider, I., Chen, B., Duarte, A., Roy, B., ... & Hardin, D. (2021). Carbon-intelligent computing: Reducing the carbon footprint of data centers. *IEEE Transactions on Cloud Computing*, 9(4), 1456-1469.

[7] Liu, T., Zhang, C., & Li, Y. (2024). Preference-conditioned multi-objective reinforcement learning for adaptive resource management. *IEEE Transactions on Neural Networks and Learning Systems*, 35(3), 1-14.

[8] Mao, H., Schwarzkopf, M., Venkatakrishnan, S. B., Meng, Z., & Alizadeh, M. (2024). Learning scheduling algorithms for data processing clusters. *Proceedings of the ACM SIGCOMM 2024 Conference*, 1-16.

[9] Xu, R., Zhou, Y., & Chen, L. (2023). Carbon-aware scheduling for geo-distributed cloud data centers. *IEEE Transactions on Sustainable Computing*, 8(2), 234-247.

[10] Sirenko, A., & Vintse, K. (2026). Optimization of computing resources in multi-cloud architectures using intelligent orchestration systems. *Journal of Numerical and Applied Mathematics*, 1, 72-78. https://doi.org/10.17721/2706-9699.2026.1.05

[11] Hayes, C. F., Rădulescu, R., Bargiacchi, E., Källström, J., Macfarlane, M., Reymond, M., ... & Mannion, P. (2022). A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1), 26.

[12] Alipour, Y., & Mohsenzadeh, Y. (2021). Resource allocation in multi-cloud environments using deep reinforcement learning. *Future Generation Computer Systems*, 124, 218-232.

[13] Ghode, V. A. (2025). *Cost and Carbon Aware Reinforcement Learning Autoscaling for Multi-Cloud Kubernetes Spot Instances* [Master's thesis, National College of Ireland].

[14] Beloglazov, A., & Buyya, R. (2012). Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. *Concurrency and Computation: Practice and Experience*, 24(13), 1397-1420.

[15] Van Moffaert, K., & Nowé, A. (2014). Multi-objective reinforcement learning using sets of Pareto dominating policies. *Journal of Machine Learning Research*, 15(1), 3483-3512.