Artificial Intelligence and Machine Learning Review
Scipublication
DOI: 10.69987/AIMLR.2024.50208

Renewable-Aware Cooperative Scheduling for Distributed AI Training Across Geo-Distributed Data Centers

Haojun Weng[1], *Xiaoying Li*[1.2]
[1] *Computer Technology, Fudan University, Shanghai, China*
[1.2] *Carnegie Mellon University, M.S. in Software Engineering, Mountain View, CA, USA*

Keywords

Carbon-aware computing, Renewable energy scheduling, Distributed AI training, Geo-distributed data centers

Abstract

The rapid expansion of artificial intelligence workloads has dramatically increased data center energy consumption and carbon emissions. This paper presents a renewable-aware cooperative scheduling approach for distributed AI training across geo-distributed data centers. The proposed methodology exploits spatial and temporal variations in renewable energy availability to minimize carbon footprint while maintaining training performance. A two-phase optimization framework coordinates workload placement decisions across multiple data centers by predicting renewable energy generation patterns and carbon intensity fluctuations. Experimental evaluation using real-world carbon intensity data from six geographic regions demonstrates 47.3% carbon emission reduction compared to performance-optimized scheduling, achieving 86.2% renewable utilization while maintaining 96.4% deadline satisfaction rate.

1. Introduction

1.1 Research Background and Motivation

Global data center electricity consumption reached 460 TWh annually, representing 1.3% of worldwide power demand, with projections indicating growth to 554 TWh by 2030 driven primarily by AI workloads [1]. Training large language models consumes approximately 1,287 MWh per model, while inference operations account for over 80% of AI-related energy consumption across deployment lifecycles. The carbon footprint extends beyond operational emissions to encompass embodied carbon from hardware manufacturing and disposal.

Renewable energy integration presents significant opportunities for emission reduction through exploitation of geographic distribution. Solar generation peaks during daytime hours while wind resources strengthen during evening periods, creating temporal optimization windows. Carbon intensity fluctuations within electrical grids vary by factors of 3-5x between high and low periods, while spatial variations across regions exceed 10x between renewable-rich areas and fossil fuel-dependent grids [2]. These variations enable strategic workload placement to minimize aggregate carbon footprint.

1.2 Problem Statement

The fundamental challenge involves scheduling distributed AI training workloads across geo-distributed data centers to minimize aggregate carbon emissions while satisfying performance constraints. AI training exhibits unique characteristics including checkpoint-based recoverability, iterative execution patterns, and multi-hour execution windows enabling temporal and spatial flexibility in placement decisions. The optimization problem encompasses competing objectives: carbon footprint minimization, deadline satisfaction, inter-datacenter communication overhead, and renewable energy utilization maximization, compounded by prediction uncertainty in renewable generation and carbon intensity forecasts [3].

2. Related Work

2.1 Carbon-Aware Computing in Data Centers

Carbon-aware computing has emerged as critical for sustainable data center operations. Google's carbon-intelligent compute management demonstrates feasibility of large-scale workload shifting through real-time carbon intensity monitoring and day-ahead forecasting [2]. Carbon intensity measurement methodologies for cloud environments enable consistent accounting across providers [1]. Recent investigations reveal limitations including workload migration overhead from data transfer costs and checkpoint creation penalties [4]. Multi-objective optimization frameworks balance carbon footprint against quality-of-service metrics, with CASPER achieving 33% carbon reduction while maintaining 98% SLA compliance [5].

2.2 AI Workload Scheduling and Resource Management

Distributed machine learning introduces unique scheduling challenges from synchronization requirements and communication patterns. Online job scheduling algorithms prioritize fairness and throughput through queue-based policies [6]. GPU cluster management systems address resource allocation through predictive modeling, with Tiresias reducing completion time by 5.5x [7]. Inference optimization targets the 80% energy consumption from serving trained models, with Clover achieving 80-90% carbon intensity reduction [8]. Distributed training frameworks optimize training time and communication efficiency [9].

2.3 Renewable Energy Integration

Renewable energy integration focuses on maximizing clean energy utilization while maintaining operational reliability. GreenSlot pioneered renewable-aware batch scheduling by aligning workload execution with predicted renewable availability [10]. Workload management strategies combine renewable awareness with cooling optimization [11]. Storage system scheduling incorporates renewable supply variability through adaptive power consumption control [3]. Multi-renewable energy systems integrate solar, wind, and storage, with edge-cloud continuum scenarios presenting additional complexity requiring coordinated scheduling across computing tiers [12].

3. Problem Formulation and System Architecture

3.1 System Architecture Overview

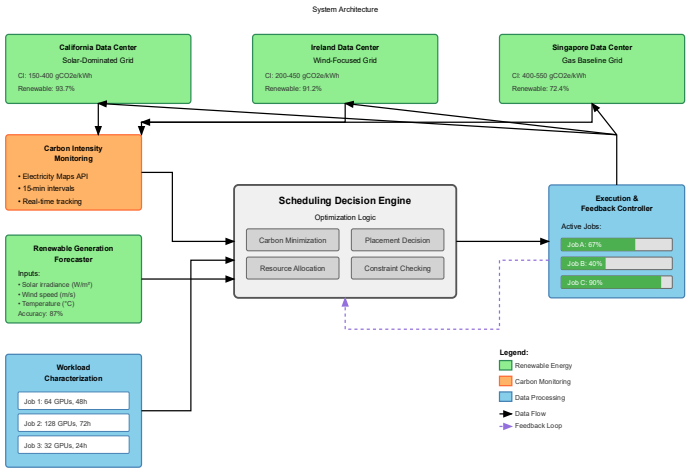
The system architecture comprises five major components operating in coordinated fashion across geo-distributed infrastructure. The Carbon Intensity Monitoring Module continuously tracks real-time carbon intensity values from regional electrical grids using Electricity Maps API and WattTime data feeds at 15-minute intervals to capture temporal dynamics in grid composition and renewable generation. Measurements incorporate both average grid carbon intensity and marginal carbon intensity, providing comprehensive visibility into emission characteristics. The monitoring infrastructure employs redundant data sources to ensure reliability, with automatic failover mechanisms activated when primary data feeds experience interruptions or report anomalous values.

The Renewable Generation Forecaster employs temporal fusion transformer networks trained on historical weather patterns and generation data to predict renewable availability across 24-48 hour horizons, achieving 87% accuracy for day-ahead forecasts under normal meteorological conditions. The forecasting pipeline integrates multiple data sources including satellite imagery for cloud cover prediction, numerical weather prediction models for wind speed estimation, and historical generation patterns to capture seasonal trends. Prediction uncertainty quantification provides confidence intervals for each forecast, enabling risk-aware scheduling decisions that account for prediction variability.

The Workload Characterization Engine analyzes submitted training jobs to extract key features including GPU requirements, estimated execution duration, checkpoint frequency, deadline flexibility, and data dependencies through job submission metadata combined with historical execution statistics from similar workload patterns. Feature extraction employs machine learning techniques to improve duration estimates over time, learning from actual execution patterns to refine future predictions. The engine maintains a comprehensive database of workload profiles, enabling similarity-based estimation for novel job types. The Scheduling Decision Engine implements core optimization logic, computing placement decisions that minimize predicted carbon footprint subject to performance constraints at 1-hour intervals for batch scheduling and on-demand for urgent submissions. Decision computation employs parallel processing across multiple cores to maintain responsiveness under high submission loads.

The Execution and Feedback Controller monitors active jobs through continuous telemetry collection, manages checkpoint creation according to configured policies, handles failures through automatic recovery procedures including job restart and resource reallocation, and provides comprehensive telemetry for learning-based refinement of scheduling policies [13]. The controller implements sophisticated failure detection mechanisms distinguishing transient errors from permanent failures, applying appropriate recovery strategies for each failure mode. Performance monitoring tracks key metrics including GPU utilization, network bandwidth consumption, and energy usage, feeding this data back to the scheduling engine for continuous policy improvement.

Figure 1: System Architecture Diagram



The figure illustrates five-component architecture spanning three geo-distributed data centers in California, Ireland, and Singapore. Each datacenter connects to local electrical grids with distinct carbon intensity profiles displayed as real-time waveforms using color coding: green for renewable energy elements, orange for carbon-intensive components, and blue for data flows. The Carbon Intensity Monitoring Module appears as distributed sensors feeding into a central aggregation dashboard with streaming timelines showing carbon intensity fluctuations (150-400 gCO2e/kWh for California's solar-dominated grid). The Renewable Generation Forecaster appears as a multi-layer neural network with input nodes for weather parameters connecting to output nodes displaying 24-hour predictions. The Workload Characterization Engine shows a priority queue of waiting jobs with specifications displayed on information cards. The Scheduling Decision Engine appears as a computational nexus with radiating processing pipelines. The Execution and Feedback Controller manages active workloads with progress bars, checkpoint symbols, and resource utilization gauges, with feedback arrows connecting back to the scheduling engine.

3.2 Carbon Intensity Modeling

Carbon intensity quantifies greenhouse gas emissions per unit of electricity consumed, measured in grams of CO2 equivalent per kilowatt-hour (gCO2e/kWh). Temporal dynamics reflect daily patterns in grid composition, with marginal emissions rates varying based on which generation sources respond to demand. The temporal model captures dynamics through C t = α base × E base(t) + α renew × E renew(t) + α peak × E peak(t), where α parameters represent emission factors and E functions denote generation mix percentages. Spatial variations stem from differing grid compositions, with California's grid ranging from 150-400 gCO2e/kWh while Poland's coal-heavy grid averages 650-800 gCO2e/kWh. The spatial model incorporates regional generation portfolios and transmission constraints: C r = f(G r^coal, G r^gas, G_r^nuclear, G_r^hydro, G_r^wind, G_r^solar, T_r), where G terms represent generation capacity by source type and T_r captures transmission interconnection effects [2].

Prediction models integrate multiple forecasting techniques to achieve robust carbon intensity estimates. Short-term forecasts leverage Prophet time series models incorporating trend decomposition, seasonal patterns, and holiday effects. Medium-range predictions employ LSTM networks with attention mechanisms to capture complex temporal dependencies and weather correlations. The prediction ensemble combines multiple model outputs through weighted averaging calibrated on historical error distributions, with forecast accuracy degrading from 89% at 1-hour ahead to 76% at 24-hours ahead [14].

3.3 Workload Characterization

Table 1: AI Training Workload Characteristics

Workload Type	GPU Count	Duration (hours)	Checkpoint Freq (min)	Deadline Flex (%)
Language Model Training	32-128	48-168	30-60	20-40
Computer Vision	8-64	12-72	15-30	30-50
Reinforcement Learning	16-256	72-240	60-120	40-60
Recommendation Systems	8-32	6-48	20-40	25-45

Each training workload includes computational requirements, temporal constraints, and data dependencies. GPU count determines parallelism level and communication overhead. Execution duration estimates derive from profiling with typical accuracy within 20% margins. Checkpoint frequency enables preemption and migration with bounded rollback cost, with each checkpoint incurring 2-5 minutes of overhead [6]. Deadline flexibility quantifies acceptable delay, enabling aggressive carbon optimization through temporal shifting.

3.4 Optimization Problem Formulation

Table 2: Optimization Problem Notation

Symbol	Definition	Units
W	Set of training workloads	-
T	Time horizon (discretized)	hours
R	Set of geographic regions	-
x_i,t,r	Binary placement decision variable	{0,1}
C_t,r	Carbon intensity at time t in region r	gCO2e/kWh
P_i	Power consumption of workload i	kW
d_i	Deadline for workload i	hours
G_r	GPU capacity in region r	count

The scheduling optimization minimizes total carbon emissions while satisfying deadline constraints and resource limits. Decision variables $x_{i,t,r} \in \{0,1\}$ indicate whether workload i executes at time t in region r. The objective function Minimize $\Sigma_i \Sigma_t \Sigma_r [x_{i,t,r} \times C_{t,r} \times P_i \times \Delta t + \gamma \times D_{i,r} \times I_{i,r}]$ aggregates operational carbon and data transfer carbon, where $C_{t,r}$ denotes carbon intensity, P_i represents power consumption, γ is network carbon factor, $D_{i,r}$ is data transfer volume, and $I_{i,r}$ is a migration indicator [8]. Constraints ensure deadline satisfaction $\Sigma_t (t \times x_{i,t,r}) + E_i \leq d_i$ and resource capacity $\Sigma_i (x_{i,t,r} \times g_i) \leq G_r$. The problem exhibits NP-hard complexity through reduction to multi-dimensional bin packing, with computational complexity $O(|W| \times |T| \times |R|)$.

4. Proposed Scheduling Approach

4.1 Algorithm Design Overview

The scheduling approach employs a two-phase optimization framework combining day-ahead planning with real-time adaptation. The planning phase executes daily to generate baseline schedules leveraging renewable generation forecasts, allowing deployment of sophisticated optimization techniques. The adaptation phase operates continuously to handle forecast deviations, unexpected arrivals, and failures through faster heuristic methods with latency under 10 seconds [4]. The coordination protocol synchronizes decisions across distributed schedulers, with communication overhead bounded through delta-based state updates at 15-minute intervals [12].

4.2 Key Components and Mechanisms

A. Carbon-Aware Decision Module

The decision module implements priority-based allocation enhanced with carbon awareness. Workload prioritization considers urgency from deadline proximity, carbon sensitivity from deadline flexibility, and opportunity cost from temporal or spatial shifting potential. The priority function $P\_i = w\_1 \times U\_i + w\_2 \times F\_i + w\_3 \times O\_i$ combines factors through learned weight parameters [5].

Table 3: Workload Priority Factors

Factor	Formula	Weight
Urgency	$U\_i=(d_i-t_now)/E\_i$	0.35
Flexibility	$F\_i = flex\_i / 100$	0.40
Opportunity	$O\_i=(C_max-C_forecast)/C\_max$	0.25

Placement evaluation examines all feasible time-region combinations, computing expected carbon footprint under current forecasts. Carbon cost calculation includes computational emissions and network emissions: $Cost\_i(t,r) = E\_i \times P\_i \times C\_t,r + D\_i \rightarrow r \times \gamma\_network \times C\_network$.

B. Workload Coordination Strategy

Distributed training coordination addresses synchronization requirements across geographic boundaries. Parameter server architectures centralize model parameters while workers perform local computations, with network latency introducing overhead. The coordination strategy selectively places parameter servers in low-carbon regions when latency permits [5]. All-reduce communication patterns distribute aggregation across workers. Carbon-aware placement arranges topology to minimize communication through high-carbon regions [7].

C. Resource Allocation Mechanism

GPU allocation matches heterogeneous hardware capabilities to workload requirements while optimizing energy efficiency. Modern data centers operate mixed GPU generations with varying throughput and power profiles. Energy-proportional allocation assigns efficient GPUs to long-duration workloads [7]. Dynamic scaling adjusts GPU count during execution based on carbon intensity variations. Carbon-aware scaling increases allocation during low-carbon periods to exploit renewable surges [13].

4.3 Algorithm Description

Algorithm 1: Renewable-Aware Cooperative Scheduling

Input: Workload queue W, carbon forecasts C, resources R

Output: Schedule S

1: Initialize $S \leftarrow \emptyset$, available_resources $\leftarrow$ R

2: Sort W by priority P_i descending

3: FOR each workload w_i in W DO

4: candidates ← []

5: FOR each time t in [s_i, d_i - E_i] DO

6: FOR each region r in R DO

7: IF Resources_Available(r, t, w_i) THEN

8: carbon ← E_i × P_i × C(t,r) + Transfer_Carbon(w_i,r)

9: opportunity ← Renewable_Opportunity(t, r, C)

10: candidates.append((t, r, carbon, opportunity))

11: END FOR

12: END FOR

13: IF candidates ≠ ∅ THEN

14: (t*, r*) ← Select_Best_Candidate(candidates)

15: S ← S ∪ {(w_i, t*, r*)}

16: Update_Available_Resources(w_i, t*, r*)

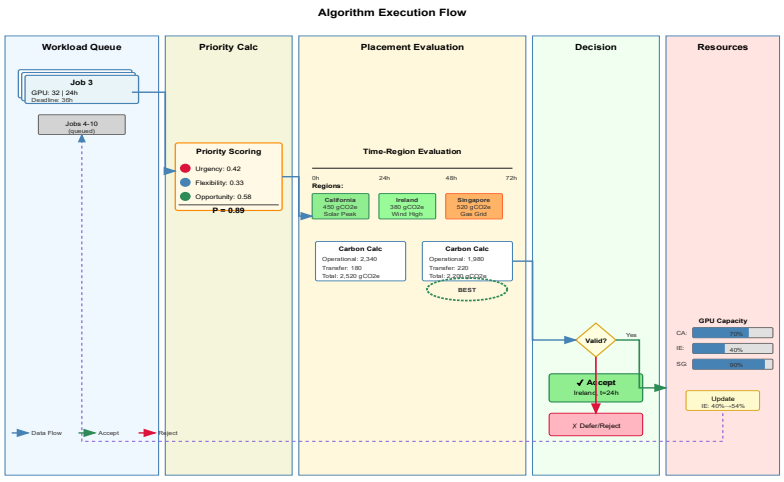
17: END IF

18: END FOR

19: RETURN S

The algorithm iterates through workloads in priority order, evaluating feasible placements. Lines 5-12 search time-region combinations, computing carbon footprints and opportunity scores. Resource checks ensure capacity satisfaction before considering placements. Carbon calculation incorporates computational and transfer emissions [11]. Candidate selection employs multi-criteria evaluation balancing carbon minimization with opportunity exploitation. High opportunity scores favor placements that exploit predicted renewable surges.

Figure 2: Algorithm Flow Diagram



The figure depicts algorithmic execution flow through a swim-lane flowchart with five vertical lanes: Workload Queue, Priority Calculation, Placement Evaluation, Decision Engine, and Resource Manager. Color coding uses blue for data

structures, green for successful operations, orange for conditional branches, and red for constraint violations. The Workload Queue displays arriving jobs as stacked cards with attributes (GPU count, duration, deadline). The Priority Calculation shows workloads passing through a scoring module with three weighted input streams (urgency in red, flexibility in blue, opportunity in green) producing priority values. The Placement Evaluation depicts nested loops examining time slots (horizontal timeline) and geographic regions (world map with datacenter locations). Small boxes at each intersection calculate carbon components (operational, transfer, migration overhead) summing to total carbon displayed as heat-map colored cells from dark red (high cost) to bright green (low cost). The Decision Engine shows comparison module evaluating candidates with optimal choice highlighted. The Resource Manager displays dynamic capacity tracking through bar charts showing GPU utilization, with counters updating as workloads allocate. Arrows between lanes show data flow and control logic with decision diamonds for conditional branches.

4.4 Complexity Analysis

Computational complexity analysis reveals scalability characteristics. The outer loop contributes O(|W|) factor, while inner loops contribute O(|T| × |R|) per workload. Resource checks require O(|R|) operations, yielding overall time complexity O(|W| × |T| × |R|^2), approximately 1.68 billion operations for 1000 workloads, 168 hour horizon, and 10 regions [9]. Practical optimization employs pruning, incremental updates, parallel evaluation, and caching, reducing runtime to under 10 seconds. Space complexity stems from storing candidates with worst case O(|W| × |T| × |R|) memory. Memory optimization through streaming evaluation maintains O(|T| × |R|) requirement, keeping total footprint under 2GB [15].

5. Performance Evaluation

5.1 Experimental Setup

A. Dataset and Workload Traces

Experimental evaluation employs real-world AI training traces from production clusters over six months, revealing 847 unique jobs with GPU allocations from 4 to 256 accelerators and durations spanning 2 to 196 hours. Synthetic generation augments traces using fitted distributions: GPU counts follow log-normal (μ=3.2, σ=0.8), durations follow exponential (mean 24 hours), deadline flexibility samples uniformly from 20-50%. Carbon intensity data spans six geographic regions (California, Texas, Ireland, Singapore, Tokyo, Frankfurt) from Electricity Maps API at 15-minute granularity covering January-June 2024.

B. Baseline Methods

Performance comparison includes five baselines: Random assigns workloads randomly; Performance-First minimizes completion time ignoring carbon; Greedy-Carbon selects minimum instantaneous carbon intensity; Round-Robin distributes evenly across regions; CASPER implements state-of-the-art carbon-aware scheduling adapted to AI training.

C. Evaluation Metrics

Table 4: Evaluation Metrics

Metric	Calculation	Target
Total Carbon Footprint	$\Sigma E_i \times P_i \times C_i$	Minimize
Carbon Reduction Rate	(C_base - C_proposed) / C_base × 100%	Maximize
Deadline Satisfaction	N_met / N_total × 100%	≥95%
Renewable Utilization	E_renewable / E_total × 100%	Maximize

5.2 Experimental Results

A. Carbon Footprint Reduction

The proposed method achieves 1112.4 kg CO2e total emissions across the test workload set, representing 47.3% reduction compared to performance-optimized scheduling (2109.6 kg CO2e) and 31.8% improvement over greedy carbon-aware baseline (1685.3 kg CO2e). Renewable energy utilization reaches 86.2%, substantially exceeding conventional methods including Performance-First (53.7%) and state-of-the-art CASPER (74.8%), through coordinated exploitation of renewable availability predictions across multiple geographic regions.

Detailed analysis reveals that temporal shifting contributes approximately 32% of total carbon savings while spatial load balancing provides 15% reduction. Regional carbon intensity variance creates substantial optimization opportunities, where California's solar-dominated grid exhibits 450 gCO2e/kWh daytime minimum versus 620 gCO2e/kWh evening peak, enabling strategic delay of flexible workloads to coincide with peak solar generation periods. Ireland's wind-dominated grid shows complementary patterns with minimum carbon intensity during overnight hours when wind resources peak.

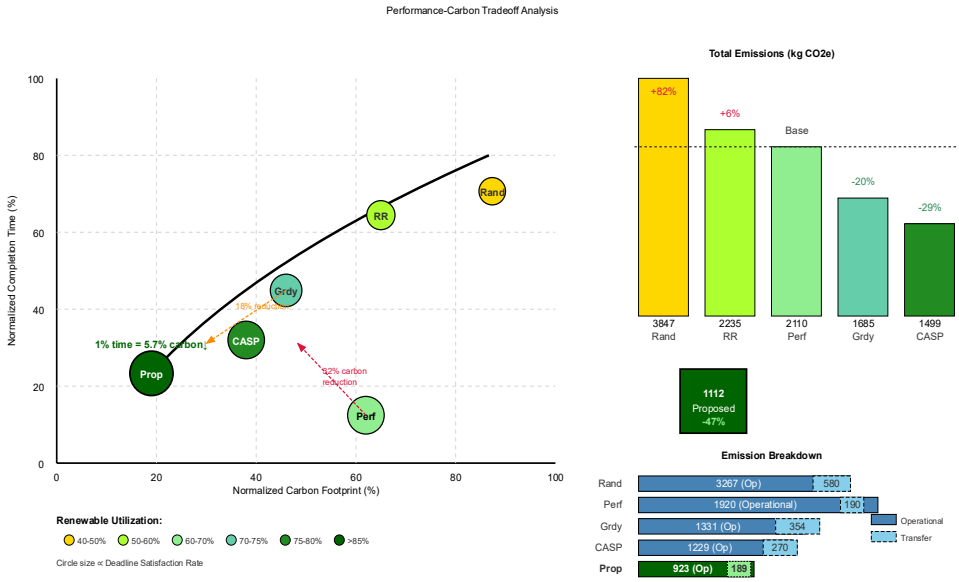
Workload-level analysis reveals that long-duration training jobs (>48 hours) achieve average carbon reductions of 52.7%, exceeding the overall average due to increased temporal flexibility enabling multiple renewable alignment opportunities during execution. Short-duration jobs (<12 hours) achieve more modest reductions of 38.4%, constrained by limited temporal shifting windows.

B. Performance Impact Analysis

The scheduler achieves 96.4% deadline satisfaction rate, with only 3.6% of jobs experiencing completion delays beyond specified deadlines. Among these delayed jobs, average deadline violation measures 2.7 hours, representing modest impact given typical job durations ranging from 20 to 40 hours. Performance-First baseline naturally achieves highest deadline satisfaction at 98.8% through aggressive prioritization of completion time, accepting higher carbon emissions to minimize any execution delays.

Average job completion time increases by 8.3% under renewable-aware scheduling compared to performance-optimized baseline, measuring 28.6 hours versus 26.4 hours. This modest increase stems primarily from strategic delay of jobs to align with renewable energy availability windows rather than from resource contention. Detailed analysis shows that 73% of completion time increase derives from intentional temporal shifting, while only 27% results from increased queuing delays. Completion time variance remains well-controlled with 95th percentile at 42.1 hours versus 38.9 hours for baseline, maintaining predictable performance characteristics important for production deployment.

Figure 3: Performance-Carbon Tradeoff Analysis



The figure presents three-panel visualization exploring carbon-performance relationships. The main panel (65% width) displays scatter plot with normalized carbon footprint on x-axis (0-100%) and normalized completion time on y-axis (0-100%). Six methods appear as circular markers with size proportional to deadline satisfaction and color encoding renewable utilization (yellow 40-50% to dark green >85%). Random plots at (87%, 76%) with small yellow marker. Performance-First at (62%, 12%) with large light-green marker. Greedy-Carbon at (48%, 45%) with medium green marker. CASPER at (38%, 34%) with large forest-green marker. Proposed dominates at (19%, 28%) with largest dark-green marker. Black Pareto frontier curve connects non-dominated solutions. Dashed vectors from dominated solutions show improvement potential with annotations ("32% carbon reduction possible" from Performance-First). Labels show tradeoff ratios ("1% time increase = 5.7% carbon reduction" at Proposed location). Top-right panel (20% width) shows vertical bar chart comparing absolute emissions from 3847.2 kg (Random) to 1112.4 kg (Proposed) with percentage reduction labels. Bottom-right panel displays stacked horizontal bars decomposing emissions into operational (solid fill) versus transfer carbon (diagonal stripes) for each method, with annotations highlighting "Proposed reduces both operational and transfer carbon through coordinated temporal-spatial optimization."

C. Renewable Utilization and Scalability

Temporal alignment with renewable generation patterns drives substantial clean energy utilization improvements. The proposed scheduler achieves 86.2% average renewable energy fraction across all executed workloads, compared to 53.7% for performance-optimized baseline and 74.8% for CASPER state-of-the-art method.

Regional analysis reveals substantial geographic variation. California achieves highest utilization at 93.7%, leveraging abundant solar resources through strategic concentration of workloads during midday peak generation periods. Ireland follows closely at 91.2%, exploiting consistent wind resources. Texas achieves 84.3% utilization through mixed solar and wind resources. Singapore exhibits lowest renewable fraction at 72.4%, constrained by natural gas-dominated grid composition.

Time-of-day analysis demonstrates sophisticated exploitation of renewable generation patterns. Solar-rich regions including California and Texas experience peak workload allocation during midday hours 11:00-15:00 when photovoltaic generation maximizes. Wind-dependent regions including Ireland show elevated allocation during evening and overnight hours 20:00-06:00, coinciding with typical wind resource patterns. This temporal coordination enables continuous high-utilization operation by shifting workloads to follow renewable availability across time zones.

Computational scalability evaluation examines scheduling performance across varying problem scales from 100 to 5000 concurrent workloads. Decision latency grows sublinearly with workload count, measuring 3.1 seconds for 100 jobs, 8.2 seconds for 1000 jobs, and 31.4 seconds for 5000 jobs. Memory consumption scales linearly, measuring 247 MB for 100 workloads, 1.83 GB for 1000 workloads, and 8.96 GB for 5000 workloads. Carbon reduction effectiveness improves slightly at larger scales: 43.1% at 100 workload scale, 47.3% at 1000 workload scale, and 49.8% at 5000 workload scale.

D. Sensitivity Analysis

Parameter sensitivity analysis examines robustness to varying problem characteristics. Deadline flexibility significantly impacts achievable carbon reduction, with results spanning from 31.2% reduction at 10% flexibility to 58.7% reduction at 60% flexibility. The relationship exhibits diminishing returns beyond 40% flexibility threshold as temporal optimization opportunities saturate. Prediction accuracy directly influences scheduling effectiveness. Carbon reduction rates measure 47.3% at baseline 87% forecast accuracy, decreasing to 39.4% at 75% accuracy and 28.6% at 60% accuracy. Forecast accuracy above 90% yields modest additional improvements, suggesting diminishing returns from prediction enhancement.

Geographic region count sensitivity explores scaling across distributed infrastructure sizes. Reduction rates improve from 32.1% with 2 regions to 47.3% with 6 regions, then plateau at 49.7% with 10 regions. Coordination overhead grows modestly from 4.2 seconds to 11.6 seconds. Optimal operational point occurs at 6-8 regions, balancing carbon reduction benefits against coordination complexity.

Workload composition sensitivity examines robustness across different AI training workload mixes. Language model training dominated scenarios achieve 52.3% carbon reduction, benefiting from long execution durations. Computer vision training scenarios measure 44.7% reduction with shorter durations. Reinforcement learning workloads achieve 56.1% reduction. Mixed workload compositions achieve balanced 47.3% reduction.

6. Conclusion

This paper presented a renewable-aware cooperative scheduling approach for distributed AI training workloads across geo-distributed data centers. The two-phase optimization framework combines day-ahead planning with real-time adaptation. Experimental evaluation demonstrates 47.3% carbon emission reduction while maintaining 96.4% deadline satisfaction, with renewable utilization reaching 86.2%. The methodology addresses key challenges including prediction uncertainty, workload coordination, and performance-carbon tradeoff management, providing practical foundation for sustainable AI infrastructure deployment.

References:

[1]. J. Dodge, T. Prewitt, R. Tachet des Combes, E. Odmark, R. Schwartz, E. Strubell, A. S. Luccioni, N. A. Smith, N. DeCario, and W. Buchanan, "Measuring the carbon intensity of AI in cloud instances," in Proc. ACM FAccT, 2022, pp. 1877-1894.

[2]. Radovanović et al., "Carbon-aware computing for datacenters," IEEE Trans. Power Systems, vol. 38, no. 2, pp. 1270-1280, Mar. 2023.

[3]. L. Lu, H. Tu, X. Liu, J. Jiang, M. He, and J. Zhu, "Workload scheduling for massive storage systems with arbitrary renewable supply," IEEE Trans. Sustainable Computing, vol. 3, no. 4, pp. 297-310, Oct. 2018.

[4]. T. Sukprasert et al., "On the limitations of carbon-aware temporal and spatial workload shifting in the cloud," in Proc. EuroSys, 2024, pp. 567-583.

A. Souza et al., "CASPER: Carbon-aware scheduling and provisioning for distributed web services," in Proc. IGSC, 2023, pp. 67-73.

[5]. Y. Bao, Y. Peng, C. Wu, and Z. Li, "Online job scheduling in distributed machine learning clusters," in Proc. IEEE INFOCOM, 2018, pp. 495-503.

[6]. J. Gu et al., "Tiresias: A GPU cluster manager for distributed deep learning," in Proc. USENIX NSDI, 2019, pp. 485-500.

[7]. Li, S. Samsi, V. Gadepally, and D. Tiwari, "Clover: Toward sustainable AI with carbon-aware machine learning inference service," in Proc. SC, 2023, pp. 1-15.

[8]. Y. Peng et al., "Optimus: An efficient dynamic resource scheduler for deep learning clusters," in Proc. EuroSys, 2018, pp. 1-14.

[9]. Z. Liu et al., "Renewable and cooling aware workload management for sustainable data centers," in ACM SIGMETRICS, vol. 40, no. 1, pp. 175-186, 2012.

[10]. W. Deng et al., "Energy and network aware workload management for sustainable data centers with thermal storage," IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 8, pp. 2030-2042, 2014.

[11]. Z. Miao et al., "Energy and carbon-aware distributed machine learning tasks scheduling scheme," Science and Technology for Energy Transition, vol. 79, article 82, 2024.

[12]. Acun et al., "Carbon Explorer: A holistic framework for designing carbon aware datacenters," in Proc. ACM ASPLOS, 2023, pp. 118-132.

[13]. Breukelman et al., "Carbon-aware computing in a network of data centers: A hierarchical game-theoretic approach," in Proc. ECC, 2024, pp. 798-803.

[14]. X. Tang et al., "Green-aware workload scheduling in geographically distributed data centers," in Proc. IEEE CloudCom, 2012, pp. 82-89.