

LB/TH/28/2026

TH6272

ENERGY-EFFICIENT CLOUD TASK SCHEDULING USING DEEP LEARNING

K. D. S. A. Chandrasiri

248054V

Master of Science (Major Component Research)

Department of Computer Science & Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

April 2026

ENERGY-EFFICIENT CLOUD TASK SCHEDULING USING DEEP LEARNING

K. D. S. A. Chandrasiri

248054V

Thesis submitted in partial fulfillment of the requirements for the degree
Master of Science (Major Component Research)

Department of Computer Science & Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

April 2026

DECLARATION

I declare that this is my own work and this Thesis does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature:

Date: 2026-04-22

The supervisor should certify the Thesis with the following declaration.

The above candidate has carried out research for the Master of Science (Major Component Research) Thesis under my supervision. I confirm that the declaration made above by the student is true and correct.

Name of Supervisor: Prof. Dulani Meedeniya

Signature of the Supervisor:

Date: 22/04/2026

DEDICATION

I dedicate this thesis to my parents, whose unconditional love, guidance, and sacrifices have been the foundation of my educational journey and personal growth. I am deeply grateful to my beloved wife for her constant encouragement, patience, and understanding, which sustained me throughout the challenges of this work.

I also extend my sincere appreciation to my supervisor, Prof. Dulani Meedeniya, for her invaluable guidance, insightful feedback, and continuous support throughout the completion of this thesis. Her mentorship has been instrumental in shaping my research and academic development.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my supervisor, Prof. Dulani Mee-deniya, for her invaluable guidance, continuous encouragement, and insightful feedback throughout the course of this research. Her expertise and support have been instrumental in shaping this work and in strengthening my academic and research skills.

I am also grateful to the academic and administrative staff of the department for their support and assistance during my postgraduate studies. Their contributions have created a conducive environment for learning and research.

Finally, I would like to thank my family and friends for their unwavering support and encouragement, which have been a source of strength and motivation throughout this journey.

ABSTRACT

This thesis presents a deep learning-based approach for energy-efficient cloud workflow scheduling by modeling scheduling as a multi-objective reinforcement learning problem that simultaneously optimizes makespan, energy consumption, and Quality of Service (QoS). The work first introduces GNN-Flat, a baseline Graph Neural Network (GNN) scheduler that represents workflow dependencies using Directed Acyclic Graph (DAG) structures and demonstrates the feasibility of applying graph-based learning to cloud task scheduling. Building on insights from this implementation, the research proposes 2SD-GAT, a two-stage deep reinforcement learning scheduler based on Graph Attention Networks (GAT), which forms the core contribution of the thesis. The 2SD-GAT architecture separates task selection and resource allocation decisions while leveraging attention mechanisms to capture inter-task dependencies and preference-based reward optimization, enabling improved Pareto trade-offs across competing objectives. Extensive experiments using synthetic and real-world datasets show superior performance compared with heuristic and learning-based baselines. The results show that the proposed approach achieves a 26.8% hypervolume gain with a 4.5x lower Inverted Generational Distance (IGD), along with a makespan improvement of 14.08%. Finally, the research bridges theory and practice by developing a pluggable RL-based scheduling framework, integrations with Apache Airflow and Kubernetes, and supporting tools including workflow engines, execution components, and a web-based GUI, demonstrating how the proposed scheduler can be deployed in realistic cloud orchestration environments.

Keywords: cloud, cloud-task-scheduling, deep-reinforcement-learning, energy-consumption, multi-objective optimization

TABLE OF CONTENTS

Declaration of the Candidate & Supervisor	i
Dedication	ii
Acknowledgement	iii
Abstract	iv
Table of Contents	v
List of Figures	x
List of Tables	xiii
List of Abbreviations	xiv
List of Appendices	xvii
1 Introduction	1
1.1 Background	1
1.2 Motivation	1
1.3 Research Question	2
1.4 Objectives	2
1.5 Problem Statement	2
1.6 Research Statement	3
1.7 Overview of Proposed Solution	3
1.8 Novelty and Contribution	3
1.9 Practical Implications	3
2 Literature Survey	5
2.1 Background	6
2.1.1 Energy Consumption in Cloud Computing	6
2.1.2 Cloud Architecture and Cloud Task Scheduling	8
2.1.3 DAG-based Cloud Workflow Scheduling	8
2.1.4 Deep Reinforcement Learning	9
2.1.5 Graph Neural Networks	9
2.1.6 Multi-Objective Optimization	10

2.2	Related Cloud Scheduling Taxonomies	11
2.2.1	Types of Cloud Workloads	11
2.2.2	Cloud Scheduling Constraints	12
2.2.3	Objectives in Cloud Workflow Scheduling	13
2.2.4	Static vs. Dynamic Scheduling	14
2.2.5	Preemptive vs. Non-preemptive Scheduling	15
2.2.6	Online vs. Offline Scheduling	15
2.3	Energy-Efficient Cloud Workflow Scheduling Techniques	16
2.4	Pareto Optimality & Evaluation Metrics	29
2.4.1	Evaluation Metrics of Related Studies	30
2.5	Experimental Datasets and Simulations	32
2.5.1	Real-World Datasets	32
2.5.2	Simulation Frameworks	36
2.5.3	Comparison Algorithms in Reinforcement Learning Approaches	37
2.6	Comparison Tables of Related Studies	37
2.7	Graph Neural Networks for Cloud Scheduling	38
2.8	Popular DAG based Workflow Engines	43
2.8.1	Apache Airflow	43
2.8.2	Prefect	44
2.8.3	Kubernetes as a Workflow Execution Platform	44
2.9	Limitations of Current Research	45
3	Methodology	47
3.1	Methodology Overview	47
3.2	Graph Neural Networks for Cloud Workflow Scheduling	48
3.2.1	Problem Modeling	48
3.2.2	Lower Bound Heuristics	51
3.2.3	Data Preprocessing	53
3.2.4	Reinforcement Learning Framework	53
3.2.5	Agent Architecture and Model Design	56
3.2.6	Dataset Generation	59
3.2.7	Agent Training	60

3.2.8	Cloud Computing Simulation Model	61
3.2.9	Agent Evaluation	63
3.3	Multi-Objective Cloud Workflow Scheduling with Two-Stage Deep Reinforcement Learning	65
3.3.1	Problem Modeling	66
3.3.2	Lower Bound Heuristics	67
3.3.3	Data Preprocessing	68
3.3.4	Reinforcement Learning Framework	68
3.3.5	Preference-Based Reward Function	70
3.3.6	Agent Architecture and Model Design	72
3.3.7	Synthetic Dataset Generation	78
3.3.8	Real-World Dataset Generation	79
3.3.9	Tuning Reward Function Parameters	81
3.3.10	Simulation Environment	81
3.3.11	Agent Training	81
3.3.12	Agent Evaluation	84
3.3.13	Comparison Algorithms	85
3.3.14	Evaluation Metrics	86
3.4	Apache Airflow Integration	88
3.4.1	Airflow Architecture Overview	88
3.4.2	Challenges in Integrating Adaptive Scheduling	89
3.4.3	Plugin-Based Scheduling Architecture	90
3.4.4	Execution and Evaluation Setup	91
3.5	Integration with Kubernetes Scheduling	92
3.5.1	Existing Kubernetes Scheduling Architecture	92
3.5.2	Proposed Kubernetes-Based Scheduling Architecture	94
3.5.3	Execution Flow	95
3.5.4	Execution and Evaluation Setup	96
3.6	Implementation of the RL-Based Pluggable Scheduler Prototype	96
3.6.1	System Overview	97
3.6.2	Core Components and Data Models	97

3.6.3	Execution Flow	98
3.6.4	RL-Based Scheduling Integration	98
3.6.5	Design Rationale	100
3.6.6	Prototype Limitations and Extensibility	100
3.7	Tool Development	100
3.7.1	Workflow Submission	101
3.7.2	Worker Implementation	102
3.7.3	Engine Implementation	104
3.7.4	GNN Agent (2SD-GAT) for Scheduling	107
3.7.5	Web GUI for Workflow Simulation	115
4	Results and Analysis	117
4.1	Results and Analysis for GNN-Flat	117
4.1.1	Results and Discussion	117
4.1.2	Algorithm Runtime Analysis	120
4.1.3	Statistical Analysis	121
4.1.4	Multi-objective solution Analysis	122
4.1.5	Limitations and Required Improvements	125
4.2	Results and Analysis for 2SD-GAT	126
4.2.1	Results for Synthetic Datasets	126
4.2.2	Ablation Study	134
4.2.3	Sensitivity Analysis	136
4.2.4	Computational Cost and Training Time	138
4.3	Results for Scheduler Implementations and Integrations	138
4.3.1	Results for Apache Airflow Integration	138
4.3.2	Results for Kubernetes-Based Scheduling Integration	140
4.3.3	Results for the RL-Based Pluggable Scheduler Prototype	141
5	Discussion	145
5.1	Baseline Research	145
5.2	Contributions and Features	146
5.3	Experimental Evaluation and Results	147
5.3.1	Comparative Analysis with Existing Studies	148

5.4	Challenges and Future Work	148
6	Conclusion	152
	References	153
	Appendix A Publications	167
	Appendix B 2SD-GAT and Tool User Manual	168
	B.1 Setting up the Environment	168
	B.2 Training the Agent	168
	B.3 Tool Execution via CLI	170
	B.4 Tool Execution via GUI	172
	Appendix C Code & Data availability	176

LIST OF FIGURES

Figure	Description	Page
Figure 2.1	CAGR of the Cloud Computing Market and Energy Consumption Share of Data Centers in Selected Regions	7
Figure 2.2	Interaction flow of a reinforcement learning framework	10
Figure 2.3	Different types of cloud workloads and the basic structure of a cloud scheduler	11
Figure 2.4	Overview of Cloud Workflow Scheduling Algorithm Taxonomy	16
Figure 2.5	Pareto Front Visualization	30
Figure 2.6	SIPHT Workflow Example (from Pegasus)	35
Figure 2.7	Overview of RLKube Approach	45
Figure 3.1	Graphical Abstract of the Research Methodology	47
Figure 3.2	RL methodology used for the proposed algorithm	49
Figure 3.3	Sample workflows represented as DAGs	49
Figure 3.4	Agent architecture used for the proposed algorithm	56
Figure 3.5	Structure of the encoder	57
Figure 3.6	An example of a graph constructed from a workflow	58
Figure 3.7	Episodic return during training	62
Figure 3.8	Cloud-computing infrastructure and components assumed in the model architecture used for simulation developed in CloudSim	63
Figure 3.9	Overview of the 2SD-GAT reinforcement learning methodology	65
Figure 3.10	RL Framework used for 2SD-GAT	72
Figure 3.11	Task Actor Network Architecture	73
Figure 3.12	VM Actor Network Architecture	76
Figure 3.13	Critic Network Architecture	78
Figure 3.14	Linear-branch-merge-sink DAG structure	80
Figure 3.15	Episodic return during training of the agent with equal weights	83
Figure 3.16	Episodic return during training of the some other agents	84
Figure 3.17	Apache Airflow Architecture (Local Executor)	89
Figure 3.18	Modified Apache Airflow Architecture (Celery Executor with RL-Driven Scheduling)	90
Figure 3.19	Apache Airflow executing workflows in RL-Driven mode	92
Figure 3.20	Kubernetes architecture overview with native scheduling components	93
Figure 3.21	Kubernetes-based scheduling architecture with RL-driven scheduler pod	95
Figure 3.22	Architecture of the RL-Based Pluggable Scheduler Prototype (Using queue-based connectors for communication)	97

Figure 3.23	Sequence diagram illustrating the execution flow of the RL-Based Plug-gable Scheduler Prototype	99
Figure 3.24	Screenshot of the web GUI for workflow simulation and scheduling visualization	116
Figure 4.1	Comparison of Makespan and Energy Consumption Across Algorithms	119
Figure 4.2	Runtime and Decision latency variation with different parameters	120
Figure 4.3	Pareto front illustrating the trade-offs between makespan and active energy consumption achieved by the proposed GNN-Flat framework compared to baseline algorithms	123
Figure 4.4	Pareto front illustrating the trade-offs between makespan and active energy consumption achieved by the proposed GNN-Flat framework compared to baseline algorithms and MOHEFT algorithm	124
Figure 4.5	Pareto fronts of schedulers for Synthetic datasets	127
Figure 4.6	Hypervolume/IGD/Decision Latency of schedulers for Synthetic datasets	128
Figure 4.7	Pareto fronts of schedulers for Real and Dynamic datasets	131
Figure 4.8	Hypervolume of Schedulers for Real and Dynamic datasets	131
Figure 4.9	Pareto fronts/Hypervolume/IGD/Decision Latency of ablation study	135
Figure 4.10	Sensitivity of agent performance to preference weights (Top) and Sensitivity analysis of agent behaviour under varying preference weights (bottom).	137
Figure 4.11	Comparison of RL and Default Schedulers Across Experiment Instances for Airflow Integration	139
Figure 4.12	Comparison of RL and Default Schedulers Across Experiment Instances for Kubernetes Integration	140
Figure 4.13	Comparison of task execution timelines for the proposed scheduler prototype under different scheduling algorithms	141
Figure 4.14	Gantt chart of task execution timelines for the proposed scheduler prototype under Random scheduler	142
Figure 4.15	Gantt chart of task execution timelines for the proposed scheduler prototype under Round-Robin scheduler	143
Figure 4.16	Gantt chart of task execution timelines for the proposed scheduler prototype under HEFT scheduler	143
Figure 4.17	Gantt chart of task execution timelines for the proposed scheduler prototype under Weighted Dynamic scheduler	144
Figure 5.1	The MPGN architecture for the FJSSP	146
Figure B.1	Training script execution	168
Figure B.2	Tool execution via CLI	171
Figure B.3	Screenshot of the scheduling GUI, with numbered elements corresponding to different functionalities	173

Figure B.4	Screenshot of the VM/workflow visualizer in the web GUI, showing the defined hosts, VMs, and task DAGs	174
Figure B.5	Screenshot of the scheduling status visualizer in the web GUI, showing a Gantt chart of task execution	175

LIST OF TABLES

Table	Description	Page
Table 2.1	Selected Approaches using Traditional Algorithms - Not Energy-Aware	17
Table 2.2	Selected Approaches using Heuristics	19
Table 2.3	Selected Approaches using Meta-Heuristics	23
Table 2.4	Reward Function Design Categories in ML-based Scheduling	26
Table 2.5	Selected Approaches using Machine Learning	27
Table 2.6	Summary of Performance Metrics in Multi-Objective Optimization	31
Table 2.7	Summary of Evaluation Metrics of Selected Heuristic Studies	32
Table 2.8	Summary of Evaluation Metrics of Selected Meta-Heuristic Studies	33
Table 2.9	Summary of Evaluation Metrics of Selected Reinforcement Learning Studies	33
Table 2.10	Workflows in Pegasus Workflow Structure	34
Table 2.11	Simulation Tool Usage among Related Literature	37
Table 2.12	Comparison Algorithms in RL-based Workflow Scheduling Approaches	38
Table 2.13	Comparison of existing RL-based Approaches	39
Table 2.14	Comparison of existing RL-based Approaches Cont.	40
Table 2.15	Comparison of existing RL-based Approaches Cont.	41
Table 2.16	Feature Comparison of RL-Based Scheduling Algorithms	42
Table 3.1	Summary of Input Features	54
Table 3.2	Host characteristics used for the dataset generation	60
Table 3.3	Training dataset generation parameters	61
Table 3.4	Hyperparameters used for PPO training	61
Table 3.5	Evaluation dataset generation parameters	63
Table 3.6	Summary of Comparison Algorithms	64
Table 3.7	Summary of Input Features	69
Table 3.8	Summary of configuration parameters derived from the Google cluster-usage traces for real-world dataset generation	80
Table 3.9	Training dataset generation parameters	82
Table 3.10	Summary of PPO hyperparameters used for training the agent and key reproducibility settings	82
Table 3.11	Reward scaling factors used for training the agent on the synthetic dataset	83
Table 3.12	Reward scaling factors used for training the agent on the real-world dataset	84
Table 3.13	Evaluation synthetic dataset generation parameters	85
Table 3.14	Evaluation real-world dataset generation parameters	85
Table 3.15	Evaluation dynamic real-world dataset generation parameters	85

Table 3.16	Summary of Comparison Algorithms	87
Table 4.1	Performance Comparison of the Proposed Algorithm and Baselines	118
Table 4.2	Statistical Test Results for Scheduling Algorithms Across Datasets	122
Table 4.3	Performance and Solution Comparison of the Proposed Algorithm and MOHEFT	124
Table 4.4	Best Objective Values of 2SD-GAT and Baselines for Synthetic Datasets	129
Table 4.5	Pareto Quality Metrics of 2SD-GAT and Baselines for Synthetic Datasets	130
Table 4.6	Best Objective Values of 2SD-GAT and Baselines for Real-World Datasets	132
Table 4.7	Pareto Quality Metrics of 2SD-GAT and Baselines for Real-World Datasets	133
Table 4.8	Performance Comparison of 2SD-GAT, 2SD-GIN, and 2SD-MLP schedulers (Best value is highlighted in bold text and second-best value is highlighted in <u>underline</u>)	136
Table 4.9	Linear sensitivity models relating preference weights ($\lambda_1, \lambda_2, \lambda_3$) to performance metrics.	136
Table 4.10	Comparison of RL and Default Across Experiment Instances	139
Table 4.11	Comparison of Scheduling Algorithms	142
Table 5.1	Feature Comparison of RL-Based Scheduling Algorithms	149
Table B.1	CLI configuration parameters for PPO training	170
Table B.2	Functional elements of the scheduling GUI	173

LIST OF ABBREVIATIONS

Abbreviation	Description
ACO	Ant Colony Optimization
AI	Artificial Intelligence
CAGR	Compound Annual Growth Rate
CIS	Cloud Information Service
CLI	Command Line Interface
CP	Constraint programming
CRDs	Custom Resource Definitions
CSO	Cat Swarm Optimization
DAG	Directed Acyclic Graph
DDQN	Double Deep Q-Network
DFJSSP	Dynamic FJSSP
DRL	Deep Reinforcement Learning
DS	Deep Set
DVFS	Dynamic Voltage and Frequency Scaling
EA	Evolutionary Algorithms
EDF	Earliest Deadline First
ETL	Extract-Transform-Load
FCFS	First-Come-First-Serve
FJSSP	Flexible Job Shop Scheduling Problem
GA	Genetic Algorithms
GAT	Graph Attention Networks
GIN	Graph Isomorphism Network
GNN	Graph Neural Network
GSA	Gravitational Search Algorithm
HEFT	Heterogeneous Earliest Finish Time
HV	Hypervolume
IGD	Inverted Generational Distance
ILP	Integer Linear Programming
KinD	Kubernetes in Docker
LSTM	Long Short-Term Memory
MDP	Markov Decision Process
ML	Machine Learning
MLP	Multi-Layer Perceptron

Abbreviation	Description
MOHEFT	Multi-Objective Heterogeneous Earliest Finish Time
MOO	Multi-Objective Optimization
MTBF	Mean Time Between Failures
NIST	National Institute of Standards and Technology
PPO	Proximal Policy Optimization
PSO	Particle Swarm Optimization
QoS	Quality of Service
RL	Reinforcement Learning
RNNs	Recurrent Neural Networks
RR	Round Robin
SPEC	Standard Performance Evaluation Corporation
VM	Virtual Machine

LIST OF APPENDICES

Appendix	Description	Page
Appendix -A	Publications	167
Appendix -B	2SD-GAT and Tool User Manual	168
Appendix -C	Code & Data availability	176