International Journal of Science and Research Archive
eISSN: 2582-8185
Cross Ref DOI: 10.30574/ijsra
Journal homepage: https://ijsra.net/

(REVIEW ARTICLE)

Check for updates

Cost-aware scheduling of ML pipelines in heterogeneous cloud environments

Ramkinker Singh *

Carnegie Mellon University, USA.

International Journal of Science and Research Archive, 2025, 16(02), 728-735

Publication history: Received on 24 June 2025; revised on 29 July 2025; accepted on 01 August 2025

Article DOI: https://doi.org/10.30574/ijsra.2025.16.2.2288

Abstract

The rise of complexity and scale in machine learning (ML) workflows and increasing adoption of heterogeneous cloud infrastructures has made cost-effective scheduling of pipelines challenging. Traditional scheduling mechanisms often don't account for variabilities in pricing, efficiency in energy consumption, heterogeneity in resources, or interoperability across clouds, which results in suboptimal costs and inefficiencies in resource utilization. In this paper, we will review the current literature and newer methods that focus on cost-aware scheduling of ML pipelines, in the aforementioned environments. We will focus on intelligent scheduling mechanisms based on reinforcement learning, AI-based scheduling, and optimization, energy aware scheduling policies, and global orchestration. In particular, we will review the recent advances in the use of evolutionary algorithms in scheduling, cloud agnostic scheduling frameworks, and carbon aware scheduling and infrastructure management, to provide a large perspective on how heterogeneous computing environments can be harnessed to increase the performance and cost-effectiveness in ML workflows. We will aim to provide a state-of-the-art overview of methods and approaches that help researchers and practitioners optimize deployment strategies for large-scale ML in multi-cloud and hybrid architecture like exist today.

Keywords: Cost-Aware Scheduling; Machine Learning Pipelines; Heterogeneous Cloud; Resource Optimization

1. Introduction

As data-driven applications experience exponential growth, machine learning (ML) is playing an increasingly prominent role in the computational workloads executed in clouds. This growth in data-driven applications typically implicates complex multi-stage ML pipelines that may consist of stages such as data preprocessing, training, validation, and deployment with each stage capitalizing on unique operating resources among digestible clouds, spanning geographies and ethical heterogeneity, likely to incur significant operational expenses if scheduled incorrectly. Given that costs and cost-aware scheduling have become an area of focus, scheduling in environments with an assortment of resources and varying pricing mechanisms, and load variability remains important [1].

ML workloads can also include several components, for example, extract-transform-load (ETL) operations, model training, and deployment, and each of these workload components may create distinct resource requirements that will need to be accounted for with scheduling. Furthermore, scheduling needs to take into account compute resource costs, data transfer costs, carbon footprint, and latency. As the trend of multi-cloud models continues many of these scheduling challenges may not only be compounded by scalability of multi-cloud implementation but with the variation of resource types and service-level agreements (SLAs). To address the scheduling challenges above, researchers are starting to propose intelligent, adaptive, and energy-efficient scheduling algorithms that take cost into consideration while optimally maintaining [or improving] overall system performance.

Cloud providers like AWS, Azure, and Google Cloud have different pricing models (on-demand, spot, reserved) as well as different hardware, and these can be utilized differently based on the work characteristics. Traditional scheduling

* Corresponding author: Ramkinker Singh.

frameworks, however, are not naturally built to respond dynamically to the complexities of cloud resource allocations. For this reason, recent frameworks have incorporated AIs, and cost-prediction strategies in scheduling, which have offered improved efficiency in performance and spend [2].

2. Cost-aware scheduling fundamentals in ml pipelines

Cost-aware scheduling seeks to sequence and assign the ML pipeline tasks across cloud resources, in an optimal manner, that minimizes spend, but allows for usability of computational resources. In the context of heterogeneous clouds, this covers CPU and GPU nodes, CPU and GPU nodes are not only heterogeneous in their compute abilities, but they offer varying capacity and cost structures. For example, in ETL tasks that are inherently a critical component of ML pipelines, inefficient execution of tasks occurs specifically because a resource is provisioned statically. In the case of ML-Frameworks, the introduction of AI-based optimization can significantly reduce cost, while also improving performance across big data tasks [1].

One of the key challenges is managing data movement across distributed environments. For pipelines that involve large data, the cost and delay of network transfers can add up significantly in terms of overall cost. A cloud-agnostic framework taking into account these activities will be able at runtime to dynamically select the most cost-effective configuration of resources to use [2]. This type of framework can abstract provider-specific characteristics to allow for easy portability of applications between providers/platforms.

Even traditional planners/schedulers are leveraging modern scheduling systems to improve task-resource mappings iteratively using performance feedback based on reinforcement learning and evolutionary algorithms. These systems can also predict execution time and cost based on different configurations to determine when to change the resource allocation strategy. For example, using reinforcement learning along with multi-objective planning/scheduling, systems can minimize overall cost, latency and energy use, in a real-time manner [3].

3. Reinforcement learning and evolutionary strategies for scheduling

Reinforcement learning (RL) represents one of the most promising approaches to cost-aware dynamic scheduling. RL enables cloud scheduling environments with a large number of configurable options and variable costs to use past executions to learn future scheduling decisions. The RL agent can learn to optimize adaptive scheduling policies, which will improve over time, account for state changes in a system, and account for environmental contingency effects. There is evidence to suggest that evolutionary reinforcement learning frameworks can develop and iterate one step further than their heuristic-based cousins – by adapting and tuning their scheduling strategies based on feedback from the environment in a constant cycle of evolution and improvement [3].

The purpose of these methods is to handle several concurrent workflows, a common situation when workflows are deploying ML applications. They allow for multi-workflow scheduling, which can appropriately consider not only the scheduling cost (a priority), but task constraints (deadlines, resource requirements, deadlines, data dependencies) that impose scheduling limitations. Using task recommendations in this way and termed as multi-workflow will enable better resource sharing and avoid contention on high demand and scarce nodes.

The benefit of such algorithms is that they can also offer the facility to solve multiple objectives. In contrast to static algorithms, RL-based methods provide a strategy for dealing with the immediate cost and for the long-term benefits; however, the RL-based schedulers still have their own unique challenges from an interpretability and resource standpoint. Deterministic mapping of tasks and the low computational cost of heuristic-based methods means RL algorithms may be reintroducing uncertainty as a side effect of the stochastic learning, which can also result in a lot of hyperparameter tuning and potential exploration strategies, which only adds to the complexity of the operations. While RL based algorithms are effective within automated workflows described as high count and able to learn and define repetitive works, their real-time abilities are still limited in cases with narrow histories or dynamic workloads. An example may be, moving a deferred task of no urgency into a cheaper time window or moving a compute-heavy task to a GPU-based node with lower utilization which provides a lot of savings over time.

Energy consumption is also important apart from costs. The connection between energy efficiency and cost-efficiency is becoming more commonplace, especially in large data centers. Scheduling strategies must not only consider the price of resources but also the energy profiles of resources, and the carbon-intent of those resources [4]. Cost-aware scheduling schemes are applying energy (and carbon) aware policies to save energy, reduce impact on the environment, and to align with the sustainability objectives of their organizations.

Energy-aware scheduling strategies may also have possible conflict with latency and throughput objectives as energy-efficient nodes may be under-resourced or located some distance away. The use of energy-aware decisions may be translated to higher data transfer times or longer job completions. Similarly, the use of real-time carbon-intensity metric will require good management of resources or to establish that a cloud provider is providing real-time metrics. This is something that most cloud providers will not provide. Energy-intensive systems tend to work in priority of efficient lifecycle workload allocation where as carbon-focused systems do not consider workload allocation effect on overall performance will get presented with more urgent throughput and latency objectives.

4. Energy-aware and carbon-conscious scheduling

When developing scheduling strategies that focus on sustainability and cost savings, energy awareness needs to be present in every decision-making step. This is critical when considering heterogeneous systems of varied power consumption between different nodes. Energy aware scheduling models are built on reducing energy use through allocating tasks efficiently, scaling resources and distributing workloads thermally. These energy aware scheduling models ultimately lower operating costs due to lower energy costs [4].

Recent trends in carbon aware cloud computing offer new types of frameworks that schedule tasks based on carbon intensity metrics. In these frameworks, the carbon intensity of how energy is generated for powering data centres is analysed, the scheduling frameworks allocate tasks at times when the data centre is running on energy from regions or servers that produce greener energy. Thus, in these models, real time carbon intensity allows the system to move workloads at different times of the day, aiding in a reduction to carbon footprint of ML processes while assisting in lower financial costs [5].

Table 1 shown below details a comparison of various scheduling strategies, presented in recent literature. This table highlights designated focus areas for each strategy, and the types of cost savings mechanisms.

Table 1 Comparison of Cost-Aware Scheduling Strategies in ML Pipelines

Scheduling Strategy	Primary Focus	Cost Optimization Technique	Environment Type
AI-Driven ETL Scheduling [1]	Data pipeline optimization	AI-based task allocation	Multi-cloud
Cloud-Agnostic Framework [2]	Dynamic adaptability	Cross-provider resource selection	Multi-cloud
RL-Based Multi-Workflow Scheduling [3]	Dynamic learning and adaptation	Evolutionary reinforcement learning	Single and hybrid cloud
Energy-Aware Scheduling [4]	Power efficiency	Thermal-aware task migration	Heterogeneous cluster
Carbon-Aware Scheduling [5]	Sustainability and cost balance	Carbon intensity analysis	Green cloud infrastructure

Although Table 1 provides a tabulated snapshot of scheduling strategies, a closer inspection reveals nuances that involve trade-offs and synergies. In particular, RL-based systems are able to adapt learning and respond to dynamic workloads, however require much more training data and also diverge over longer time periods creating delayed convergence - thus, customer cold-start is not very ideal. Conversely, heuristic-based methods are easier to derive and build, and interpret, however they lack the required adaptability worth mention in changing pricing corridors. Energy-aware and carbon-conscious strategies often conflict with performance optimization, especially when greener nodes are located in regions with higher network latency. Interestingly, cloud-agnostic frameworks complement global optimization strategies well, as they offer the abstraction needed for scalable cross-provider deployment, though they may abstract away critical low-level tuning opportunities. Understanding these interplays is crucial when selecting or designing a scheduler for a specific ML workload.

Below is a schematic overview of a typical cost-aware ML pipeline scheduling architecture in heterogeneous cloud environments. It includes the decision layers, learning modules, and feedback mechanisms for adaptive scheduling.

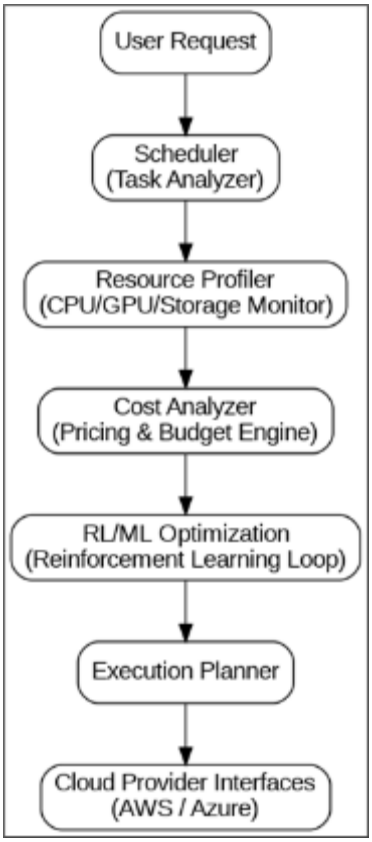

Figure 1 Cost-Aware ML Pipeline Scheduling Architecture. Created based on the conceptual synthesis of sources 1-5

The following graph illustrates the performance (in throughput) against cost (USD per job) of different scheduling strategies based on synthesized data from the studies reviewed.

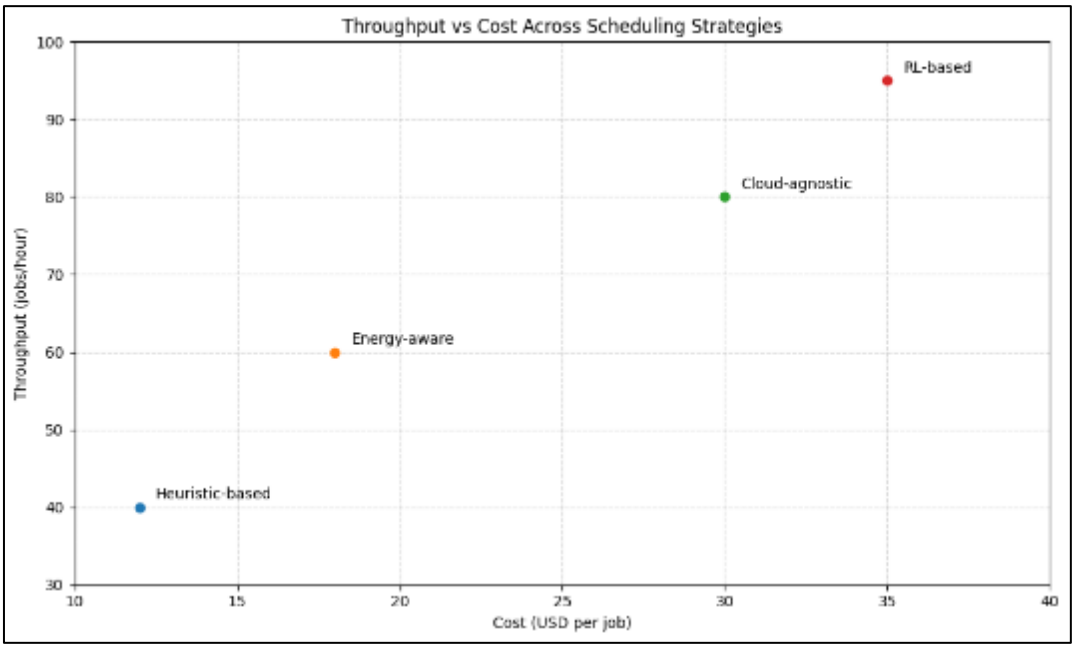

Figure 2 Cost vs. Performance of Scheduling Strategies. Simulated based on average data comparisons across refs 1–5

5. Global optimization in heterogeneous cloud pipelines

With the rise of increasingly distributed ML applications, the need for global optimization across heterogeneous environments has become imperative. ML pipelines today may span multiple data centers and even multiple cloud vendors. Each platform comes with different virtual machine types, billing models, availability zones, and data transfer policies. These disparate concerns present a non-trivial hurdle for cost-aware scheduling functions aiming to reach global optimization while still considering performance, consistency, and reliability.

Global optimization techniques aim to account for each of the three constraints holistically. For example, scheduling based on global optimization techniques stresses the amount of throughput maximally achieved while minimizing costs and operational efficiency. A well-documented example of a global optimization technique is revealing a workflow on any machine learning framework that pipeline restructures workflows into modular sub-tasks. Once decomposed into modular sub-tasks, the sub-tasks can be mapped to optimal resources while taking into consideration, for example, a cost function with multiple goals relating to cost, availability, and performance [6]. Furthermore, we can apply this method of de-composition and pipeline restructuring to implement fine-grained scheduling in a context-sensitive schedule optimization that can change based on the operational context.

An important component of global optimization frameworks is data locality awareness. Many ML pipelines are heavily data-centric; therefore, the proximity of compute resources to data sources significantly impacts both latency and cost. A global optimizer must analyze data gravity, bandwidth limitations, and storage costs to ensure efficient task-to-resource binding. These considerations are especially critical for tasks like model training and data preprocessing, where large-scale datasets are frequently accessed and transformed.

In addition, the field of latency-sensitive applications, e.g. real-time inference or adaptive learning systems, will be positively impacted by optimization approaches accounting for inter-cloud costs and time-delays of data transit. Given the optimization scheduling model predicted the execution times and data transfer overheads, as well any dynamic pricing models, whether to co-locate assembled tasks or otherwise distribute them could be assessed. The challenge in practice is discovering a balance between locality, costs, and scalability while managing fluctuating system loads.

For example, when running multiple models concurrently, within a federated cloud context (Multiple ML Models), a cost-aware scheduler must investigate the appropriate deployment position for each model component segment. The scheduler called upon a knowlege graph of resource metadata and utilized real-time price feeds and machine profiles to apply global optimizations heuristics to the assignment of tasks. Each decision was refined by any telemetry data stored from execution and fed to the optimization loop for future iterations [6].

This is increasingly useful when task dependencies are non-linear or when performance at one stage of a pipeline has an impact on subsequent stages. Global optimization of execution sequences will not only save direct operational costs, but also increase reliability and scalability of the overall pipeline.

While global optimization frameworks have matured to enable reasoning about a larger number cost and performance metrics, there are several important challenges which remain under-explored. One specific challenge is dynamic multi-objective optimization across diverse and globally distributed resources. The variation in pricing, availability, and capabilities of resources across regions introduces new levels of complexity that existing optimizers do not account for. Furthermore, all systems tend to underestimate hidden operational costs, such as cold-start penalties for lightly used VMs, or redundancy overheads in replicated environments. A significant shortcoming is that no current system is taking into consideration the real-time market algorithmic nature of resources other than by the inclusion of spot instance selection into the optimization model. Current initiatives in this space do not make it possible to incorporate more advanced financial indicators such as, pricing trends on preemptible instances, or SLA penalties and carbon credits, into a more financially and environmentally efficient scheduling decision.

6. Geo-distributed scheduling and streaming workflows

Many of today's machine-learning (ML) applications ultimately source data from a stream of data points—data streams can originate from an IoT device, real-time analytics system, or from an edge sensor. The act of scheduling workflows for these streaming data types over geographically-distributed data centers adds further constraints and challenges. Cost optimization in this context involves not only compute and storage costs but also factors such as network latency, data duplication, and data egress costs.

Cost-efficient streaming workflow allocation frameworks have been designed to address these issues by optimizing scheduling across cloud regions with geo-distributed architectural components. Such frameworks utilize the model of minimizing communication between regions while ensuring consistency and responsiveness is consistent with data produced from each of the data architectures. Incorporating network topology awareness, resource availability, and energy consumption into the scheduling logic is essential [7].

Such systems are able to allow real-time task reallocation without disrupting story continuity when performance service-level objectives (SLOs) are maintained in the even precise contingencies such as congestion, data input bursting or unexpected cloud pricing rates. The inherently dynamic characteristics of streaming workflows often require the scheduling engine to be reactive rather than indicative making the integration of adaptive learning models essential. For example, should a region encounter sudden bandwidth throttling or pricing surge, the scheduler must reliably offload non- {critical components) otherwise known as minimal components to alternate non-task-ondemand nodes without interrupting the overall continuity of the workflow through data capacity.

Geo-distributed scheduling also facilitates fault tolerance by enabling redundancy-aware replication. As a result of this redundancy, ML services of high-priority, such as a recommendation engine's inference models, maintain high availability and can recover quickly from known partial part of the system failure. Nevertheless, having multiple replicas per shards across regions comes at an extra cost. Hence, cost-aware schedulers must decide in real-time when and where they want to instantiate replicas to shoulder that additional cost.

The trade-off between redundancy due to fault-tolerant implementation, against cost savings, calls for a layered approach to streaming workflow management. Time-sensitive tasks receive higher priority and greater redundancy with near instant fail-over while lower priority or ancillary tasks are postponed or stereotyped based upon cost-benefit. Time-zone-aware scheduling capabilities are native to geo-distributed scheduling, and also enable scheduling tasks so they can be shifted towards regions with the smallest current demand and least rush-hour. Non-time-sensitive jobs such as batch jobs can always be scheduled later when they can be scheduled through cheaper regions but still within their acceptable deadlines.

7. Strategic approaches for public cloud cost optimization

Opportunities for savings exist in public cloud environments through reserved instances, spot instances, autoscaling, and pricing tiers, but these opportunities are leveraged effectively in ML pipelines through effective scheduling decisions made with both historical use patterns and predictive analytics in mind.

As prior yellow papers have suggested, one way to think about this is to create two pipeline classes, critical and non-critical. Non-critical components (e.g., data archiving, batch training, etc.) can be scheduled on spot instances that can be interrupted but are cheaper. Critical components can then be scheduled on reserved or on-demand instances that provide a guaranteed solution for scheduling. This classification-mapping of resources serves as the basis for hybrid cost optimization strategies in public cloud platforms [8].

One additional approach is scheduling autoscaling with a cost limiter. With this approach, the auto-scheduler dynamically provisions additional resources during peak load times, but enforces a cost cap. The machine learning models in the scheduler compound predictions of future load based on historical patterns, thus allowing the scheduler to provision based on anticipated load and availability, rather than historical load that highlights reaction-based provisioning. Doing so improves both cost control and resource utilization simultaneously.

Further cost-aware scheduling could be using instance pre-selecting, where virtual machines can be ranked in terms of price/performance, selected for scheduling by the scheduling algorithms for the various pipeline components using valid performance benchmarks that have been pre-calibrated with real-time market price streams. This further precision allows for a more granular allocation of tasks since each task is allocated to the least expensive node that satisfies the task requirements.

Further optimization could be achieved due to the visibility of resource consumption by using 'cloud-native' monitoring tools, and the resource provider billing APIs over the public cloud. This data is ingested by the scheduling engine so it may make periodic adjustments. It can shut-down resources that are not being used, or it can consolidate the tasks with similar characteristics from completion times, minimizing the fragmentation of the billing.

In summary, the issue for effective public cloud scheduling is adaptive, intelligent, and finely-grained control that aligns the provisioning of resources with the workload and pricing behaviours [8].

8. Embracing heterogeneity through adaptable frameworks

Due to the variety of cloud platforms, which include CPUs, GPUs, FPGAs, TPUs, and different storage models - it must be flexible, extensible, and pragmatic so it can respond to the evolving context. Flexible resource management frameworks are designed to exploit heterogeneity - dynamically matching one task's characteristics to a resource profile. These frameworks will provide profiling engines which analyze workload attributes - such as memory intensity, parallel-ability, data dependency, or tolerance to delay [9]. After profiling work into these characteristics, mapping and classifying workloads are made - tasks like deep learning training which are GPU intensive go to a GPU accelerated node - memory improved instances for memory heavy tasks like data aggregation. Not only does the equivalent matching improve the user's performance; it reduces idle compute cycles, improving overall efficiency.

Also, as a result of the modular design, either new resources or service models can easily plug into the framework as they are developed. These frameworks rely on learning and feedback loops to improve on scheduling decisions over time. This means that as cloud platforms change or workloads shift, the scheduler will continue to make optimal decisions without manual intervention.

Frameworks that accept heterogeneity can also provide fault tolerance. When workloads are distributed across heterogeneous nodes without being overly reliant on any single component from hardware, the chance of suffering from a fault (node becomes unavailable) or performance degradation (loss of computing resources) becomes greater. Redundancy methods and checkpointing can improve resiliency without excessive costs since tasks can be duplicated or resumed from a checkpoint.

Heterogeneity is even more meaningful when it offers adaptability to support the wide variety of ML workloads—from batch processing to edge inference—and ensuring that consistent performance and cost remain stable as scales or operating conditions change [9].

9. Conclusion

Cost-efficient scheduling of machine learning pipelines over a mix of cloud services is a significant step forward in computing and the development and provisioning of infrastructure services through cloud environments. In particular, the use of AI-driven optimized scheduling techniques, energy-wise and carbon-aware systems, reinforcement learning methods, and global orchestration have the potential to increase the cost-effectiveness, scalability, and sustainability of ML workflows.

This review has looked at major techniques, challenges, and architecture from recent work, and is hopeful there will continue to be a greater move to embodied, adaptive scheduling, and situation-aware scheduling systems. With the continuing development and rollout of cloud technologies, and the increasing need for sustainability, the next generation of cloud scheduling systems must incorporate even more predictive analytics, telemetry, and resource provisioning and optimization in a decentralized manner to remain relevant in a more complex operating environment.

Although there has been considerable work in scheduling mechanisms that consider cost, there is still significant room for ongoing research. For instance, the absence of standardized benchmarks and datasets for testing scheduling algorithms across various cloud contexts decreases the ability to generalize the results of existing research. Additionally, many of the existing models, which use cost as an underpinning, assume that work is invariant in time and predictably used, which is inconsistent with the stochastic nature and unpredictable workload of real-world ML pipelines. Furthermore, multi-objective optimization processes predicated on cost do not yet tackle the miserly optimization challenge of co-optimizing potentially opposing cost-related objectives like minimizing cost—in the context of acceptable output—maximizing throughput, and minimizing energy use in large-scale scenarios. Little work has also been carried out on quantifying hidden costs of a cloud application — for example the cost of resource provisioned but otherwise idle or the cost of storing intermediate data or delaying elements of the pipeline as a result of network latency

In many cases the implications of integrating real-time market-driven pricing models—beyond spot/on-demand pricing—remain unexplored. However, further lines of research are required on semi-decentralized and federated scheduling models, particularly for edge-cloud hybrid systems, or on whether and how adaptive models can use carbon trading markets, SLA-based pricing variability and data sovereignty constraints in real time.

References

[1] Krishnama, C., Puchhakayala, R., Kotha, S., and Gouri, F. (2025, April). Cost-Optimized Cloud Scheduling for ETL and Big Data Using AI. In 2025 13th International Symposium on Digital Forensics and Security (ISDFS) (pp. 1-6). IEEE.

[2] Jiang, F., Ferriter, K., and Castillo, C. (2020, April). A cloud-agnostic framework to enable cost-aware scheduling of applications in a multi-cloud environment. In NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium (pp. 1-9). IEEE.

[3] Huang, V., Wang, C., Ma, H., Chen, G., and Christopher, K. (2022, November). Cost-aware dynamic multi-workflow scheduling in cloud data center using evolutionary reinforcement learning. In International Conference on Service-Oriented Computing (pp. 449-464). Cham: Springer Nature Switzerland.

[4] Kumar, N., and Vidyarthi, D. P. (2017). An energy aware cost-effective scheduling framework for heterogeneous cluster system. Future Generation computer systems, 71, 73-88.

[5] Beena, B. M., Prashanth, C. S. R., Manideep, T. S. S., Saragadam, S., and Karthik, G. (2025). A Green Cloud-Based Framework for Energy-Efficient Task Scheduling Using Carbon Intensity Data for Heterogeneous Cloud Servers. IEEE Access.

[6] Lin, E., Xu, L., Bramhavar, S., de Oca, M. M., Gorsky, S., Yi, L., ... and Chou, J. (2022). Global optimization of data pipelines in heterogeneous cloud environments. arXiv preprint arXiv:2202.05711.

[7] Chen, W., Paik, I., and Li, Z. (2016). Cost-aware streaming workflow allocation on geo-distributed data centers. IEEE Transactions on Computers, 66(2), 256-271.

[8] Aydoğan, M., and Batan, A. (2024). Cost Optimization Strategies for Big Data Analytics in Public Cloud Infrastructures. Applied Science, Engineering, and Technology Review: Innovations, Applications, and Directions, 14(10), 1-13.

[9] Thinakaran, P. (2021). Embracing Heterogeneity in Cloud Platforms through Adaptable Resource Management Framework.