Minimax Theory and its Applications

Volume 11 (2026), No. 3s, 192 to 201

Variational Optimization Methods for Resource-Aware Scheduling in Cloud Computing Environments

Dr. Franciskus Antonius Alijoyo

Principal, Center for Risk Management and Sustainability Indonesia, Lecturer, School of Business and Information Technology LIKMI Bandung, Indonesia Parahyangan Catholic University Bandung, Indonesia
Email: franciskus.antonius.alijoyo63@gmail.com, Orchid Id: 0000-0002-5624-873X

Abstract
Resource scheduling methods for dynamic cloud computing environments must be robust in nature to adjust to workload uncertainty, resource contention, execution-time variation and deadline requirements. Deadline reliability is not guaranteed with existing deterministic scheduling techniques under changing execution conditions, and fixed chance constraints are not flexible enough to the changing priority of tasks and the uncertainty level. In this paper, we introduce an Adaptive Chance-Constrained Alternating Direction Method of Multipliers (AC-ADMM) framework for uncertainty-aware cloud resource scheduling. The framework combines a distributed ADMM optimization with probabilistic deadline requirements and proposes using adaptive risk tolerance based on the priority of the tasks and the uncertainty in the amount of work executed. ADMM penalty parameter is dynamically adjusted based on primal and dual residuals to improve consensus and convergence. The deterministic equivalent of the chance constraint is calculated, and a deadline reliability theorem is stated and proven in order to ensure the desired schedule reliability (probability of meeting the task deadlines) under the assumptions of a distribution for the execution time. Deadline success rate, SLA violation rate, makespan, energy consumption, scheduling cost, resource utilization and convergence behaviour are used to measure the performance of the framework. Convergence stability under increasing workload uncertainty and an 89.7% deadline success rate with a 10.3% SLA violation rate at AC-ADMM. The experiments show how theoretical deadline guarantees can be successfully combined with adaptive distributed optimization to guarantee reliable and resource-aware cloud scheduling.

Keywords: Cloud Computing, Resource Scheduling, Chance Constraints, ADMM, Uncertainty-Aware Optimization, Deadline Reliability, SLA, Distributed Optimization.

I. Introduction

Cloud computing has become a widely adopted paradigm for providing scalable processing, storage, and application services. The increasing diversity of cloud applications requires scheduling methods that efficiently allocate resources while satisfying performance and service requirements. In real-time environments, scheduling decisions are affected by workload intensity, task arrivals, execution time, resource availability, and service-level agreement (SLA) requirements. Therefore, multiple objectives must be considered rather than task completion time alone. Recent studies have investigated deadline-constrained scheduling, energy-aware allocation, virtual machine scheduling, and SLA-aware resource management (Janjani*et al.*, 2024; Kumar *et al.*, 2024; Manam *et al.*, 2023; Mehor*et al.*, 2024; Prathiba& Sankar, 2024). However, reliable scheduling under changing workload and resource conditions remains challenging.

A major challenge is uncertainty in task execution time caused by resource contention, workload fluctuations, virtualization overhead, and changing computational capacity. Deterministic models based on expected execution time may satisfy deadlines during planning but fail during execution. Existing deadline-, energy-, and SLA-based approaches address resource efficiency and service requirements but do not fully capture probabilistic execution-time uncertainty (Kumar *et al.*, 2024; Li *et al.*, 2024; Manam *et al.*, 2023).

Chance-constrained optimization incorporates uncertainty by specifying a minimum probability of meeting a deadline. Such formulations have been applied to stochastic cloud scheduling, online resource allocation, and energy-aware data-centre scheduling (Chen *et al.*, 2023; Liu *et al.*, 2024; Yan *et al.*, 2022). However, a fixed probability level may not provide suitable reliability for tasks with different priorities, workload uncertainty, and

SLA requirements. This motivates an adaptive risk mechanism that changes according to current scheduling conditions.

Cloud infrastructures also contain multiple resource clusters with heterogeneous capacities, making centralized optimization computationally demanding. Distributed optimization can divide the problem into local subproblems while maintaining global coordination. ADMM supports this process through local and consensus updates, and recent studies have investigated distributed and accelerated ADMM methods (Rokade & Kalaimani, 2025; Wang *et al.*, 2025). Its convergence can further benefit from adaptive penalty selection based on optimization residuals (Crisci *et al.*, 2023).

Although cloud scheduling, chance constraints, and distributed ADMM have been studied separately, their integrated use for dynamic real-time scheduling remains limited. Existing methods often focus on individual objectives such as energy, cost, SLA compliance, or task completion. Reinforcement-learning-based approaches have also addressed adaptive scheduling and SLA violation reduction (Krishna & Dudekula, 2025; Mangalampalli*et al.*, 2024; Yamsani& Chenna Reddy, 2026), but they do not provide the proposed mathematical integration of adaptive probabilistic deadline constraints and residual-based ADMM control.

To address this gap, an Adaptive Chance-Constrained Alternating Direction Method of Multipliers (AC-ADMM) framework is proposed for resource-aware scheduling in dynamic cloud environments. The formulation jointly considers execution time, energy consumption, and operational cost with resource-capacity and probabilistic deadline constraints. The risk tolerance, $\varepsilon_i(t)$, is dynamically adjusted according to task priority, resource utilization, workload uncertainty, and SLA criticality. ADMM decomposes scheduling across resource clusters, while the penalty parameter, ρ_k, is updated using primal and dual residuals to improve convergence under changing conditions.

The main contributions of this study are as follows:

1. An uncertainty-aware formulation of multi-objective scheduling is presented by considering execution time and energy consumption, along with operational cost and resource constraints.
2. An adaptive chance constraint mechanism, whereby the deadline risk tolerance varies based on task priority, workload uncertainty, resource utilization, and SLA criticality, is presented.
3. A distributed scheduling mechanism based on ADMM is constructed to divide the allocation of resources across the clusters of clouds while ensuring global consensus.
4. A Residual-driven Adaptive Penalty (RAP) mechanism is implemented to adaptively adjust the ADMM penalty parameter based on the primal and dual residual activity.

To evaluate the proposed AC-ADMM, it is compared with the deterministic ADMM and chance-constrained ADMM with respect to deadline success rate, SLA violation rate, makespan, energy consumption, cost, resource utilization, convergence time, and scheduling overhead, under varying dynamic workload and resource conditions.

II. Cloud Scheduling And Optimization Background

Cloud resource scheduling is defined as the action of assigning tasks to the available resources so that the execution time, energy consumption, cost and resource utilization are balanced. Scheduling is challenging for dynamic workloads and resource contention due to changing execution conditions during the task's processing. Most scheduling methods consider deadline constraints, SLA requirements, energy efficiency and resource allocation.

Uncertainty-aware optimization takes a probabilistic approach to deadline requirements, enhancing the reliability of the scheduling. Chance constraints can be exploited to control the probability of violating deadline constraints, and multi-objective optimization can be used to handle multiple scheduling requirements. But a static risk tolerance might not be sufficient to mirror variance in task priority and workload uncertainty.

Distributed optimization is helpful when your cloud consists of many resource clusters. With ADMM, local decisions on resources can be optimized independently of one another but with global consensus. The convergence of ADMM can be further enhanced by adaptively adjusting the penalty parameter on the basis of primal and dual residuals. These requirements also drive towards combining the adaptive chance constraints with the adaptive ADMM into a single resource-aware scheduling framework.

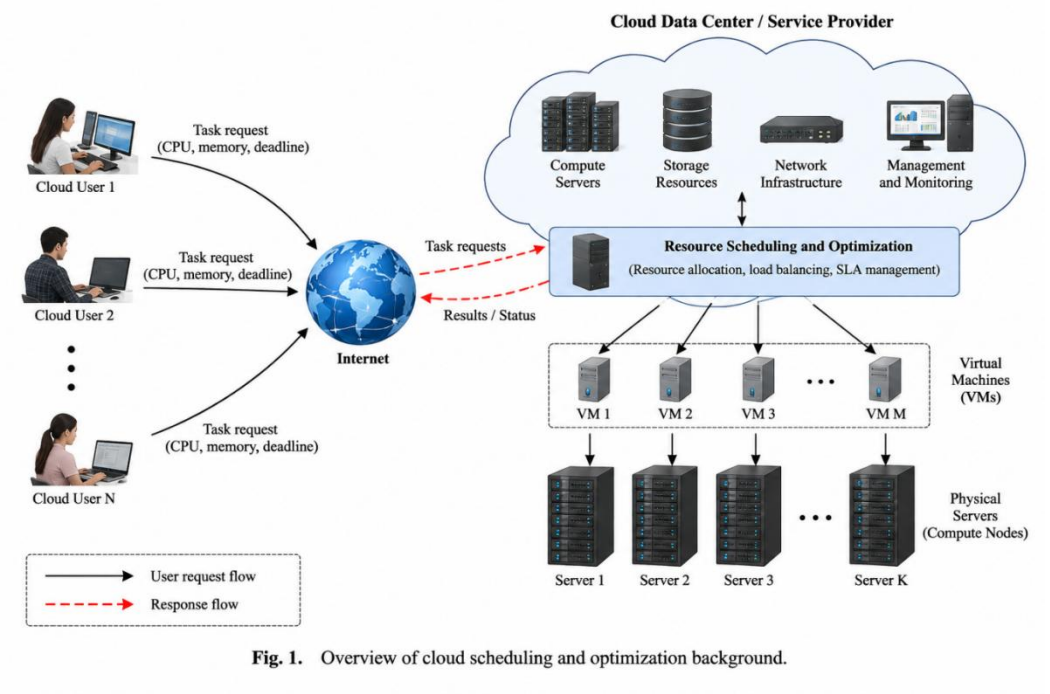


Fig. 1. Overview of cloud scheduling and optimization background.

Fig. 1. Overview of cloud scheduling and optimization background.

III. Uncertainty-Aware Scheduling Model

A Heterogeneous cloud has N real-time tasks and M available resources. There is a specific resource requirement, execution time, and deadline for each task and a limited processing capacity for each resource, along with five specific corresponding priorities. The binary variable $x_{ij}$is defined as the assignment of task i to resource j, with indicators thatx_{ij}=1 if there is an assignment and x_{ij}=0 if the task is not assigned to that resource. Resources have a strict one-task-per-resource allocation, and the total resource demand allocated to a resource does not exceed its capacity.

This scheduling objective takes into account energy consumption, operation cost and execution time. This leads to a multi-objective formulation, which is then written as

$$\min_{x} \; w_1 T(x) + w_2 E(x) + w_3 C(x) \qquad (1)$$

subject to

$$\sum_{j=1}^{M} x_{ij} = 1, \; x_{ij} \in \{0, 1\}, \qquad (2)$$

$$\sum_{i=1}^{N} x_{ij} \, R_i \leq R_j. \qquad (3)$$

Here, $T(x), E(x)$,and $C(x)$represent total execution time, energy consumption, and operational cost, respectively, while w_1, w_2, w_3determine their relative importance.

Time for execution is considered uncertain due to the variability of workload, resource contention and changes in resource capacity of resources, which can alter the execution time of the tasks. Denote the time it takes to execute a task.$i$be by $T_i(x, \xi)$,and its average value and standard deviation by $\mu_i(x)$and $\sigma_i(x)$, respectively. Consequently, the deadline requirement is specified in terms of a probability statement:

$$P(T_i(x, \xi) \leq D_i) \geq 1 - \epsilon_i(t). \qquad (4)$$

The probabilistic constraint translates to a deterministic one for normally distributed execution time:

$$\mu_i(x) + \Phi^{-1}\big(1 - \epsilon_i(t)\big)\sigma_i(x) \le D_i. \qquad (5)$$

The formulation enables a direct link between the uncertainty of a deadline execution time and the deadline reliability. A smaller value of $\epsilon_i(t)$sets a more stringent reliability requirement, while a larger value allows scheduling to take a greater risk. However, as a result, the risk value is not regarded as a fixed value, and it is adaptively determined in the proposed AC-ADMM framework.

Theoretical Analysis

Theorem 1 (Deadline Reliability Guarantee).
Suppose that $T_i \sim \mathcal{N}(\mu_i,\ \sigma_i^2)$and assume the scheduling solution satisfies (5), then the probability of completing task i on time t is lower bounded by $1 - \epsilon_i(t)$.

Proof: Since $T_i$follows a normal distribution,

$$P(T_i \le D_i) = \Phi\left(\frac{D_i - \mu_i}{\sigma_i}\right). \qquad (6)$$

From (5),

$$\frac{D_i - \mu_i}{\sigma_i} \ge \Phi^{-1}\big(1 - \epsilon_i(t)\big). \qquad (7)$$

Because $\Phi(\cdot)$is monotonically increasing,

$$P(T_i \le D_i) \ge \Phi\left(\Phi^{-1}\big(1 - \epsilon_i(t)\big)\right) = 1 - \epsilon_i(t). \qquad (8)$$

Therefore, the chance-constrained formulation ensures that at least.$1 - \epsilon_i(t)$of the time meets the deadline. The scheduling model's reliability property is proved in the theorem.

IV. Adaptive Chance-Constrained Admm Framework

The proposed Adaptive Chance-Constrained Alternating Direction Method of Multipliers (AC-ADMM) has been presented for reliable resource scheduling in dynamic cloud environments. Uncertain workloads and bounded resources, coupled with varying service requirements, make it crucial to capture probabilistic deadline protection and distribute optimization within the framework. The overall process includes uncertainty estimation, adaptive risk control, chance-constrained scheduling, distributed optimization and real-time feedback, which are depicted in Fig. 2.

A. Adaptive Risk Control

The first component is the acceptable deadline risk of each task. The framework, however, does not assume the same level of risk tolerance for all tasks, but it varies the risk-tolerance level based on the task priority, workload uncertainty, resource utilization, and criticality of SLA. High-priority or SLA-sensitive tasks are thus given more stringent deadline reliability requirements, and tasks with low criticality can accept a higher level of deadline uncertainty in comparison. Whenever workload variations or resource utilization changes, the risk level is also updated. This adaptive mechanism is used to overcome the disadvantages of having one set probability threshold during the scheduling process.

B. Distributed ADMM Scheduling

The scheduling problem is broken down into resource clusters following risk estimation. Local optimizations are carried out at each cluster with the available computing resources, and a consensus mechanism is used to coordinate the local optimizations with the overall scheduling requirement. To tackle the large-scale scheduling problem, ADMM is applied to divide it into a set of small subproblems so that the required computation is reduced compared to centralized optimization. Local resource allocation can, therefore, be tailored to cluster-level conditions, while keeping with the global resource scheduling objective.

C. Residual-Based Penalty Adaptation

The convergence behaviour of ADMM can be controlled via the relationship between the primal and dual residuals. The change of the consensus solution between successive iterations is called the dual residual, while the difference between the local resource decision and the consensus solution is called the primal residual. The penalty parameter is increased if the disagreement of the primal problem prevails and decreased if the disagreement of the dual problem prevails. In this adaptive adjustment, the optimization will adjust to the convergence conditions, rather than use a penalty that is fixed. This ADMM adaptation based on residuals agrees with well-known methods of adaptive selection of ADMM parameters.

D. Real-Time Scheduling Feedback

The last one is a feedback mechanism for continual adaptation of scheduling. During the scheduling process, workload intensity, execution-time uncertainty, resource utilization and resource availability all change. The scheduling decision is revised based on the new information provided by the updated information of uncertainty estimation and risk-control stages when the operating conditions change. This closed-loop mode is especially critical in cloud settings, where resources can't be guaranteed to persist in the same state.

The whole AC-ADMM process is as follows: Dynamic Workload → Uncertainty Estimation → Adaptive Risk Control → Chance-Constrained Scheduling → Distributed ADMM → Residual-Based Penalty Adaptation → Resource Allocation → Real-Time Feedback. The proposed framework consists primarily of task-level risk adaptation, as well as iteration-level convergence adaptation. This design ensures an end-to-end approach of maintaining deadline reliability while considering the execution time (runtime), overall energy consumption, cost, and resource utilization under the dynamic cloud environment.

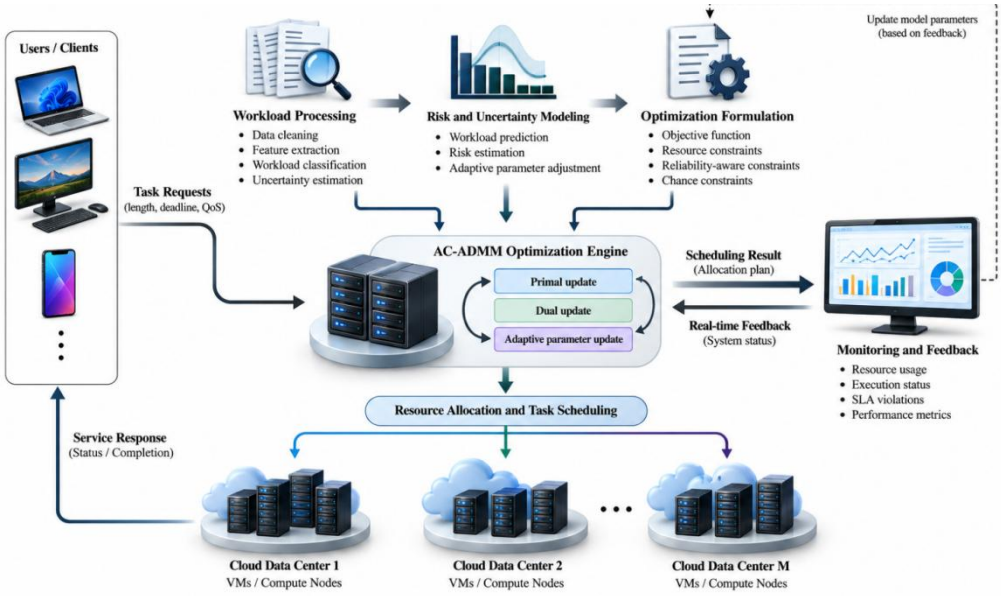


Fig. 2. Integrated workflow of the proposed AC-ADMM framework for adaptive and reliable cloud resource scheduling.

V. Experimental Methodology

A. Experimental Setup

A cloud scheduling scenario of 5K independent tasks across 8 clusters of resources was used for evaluating the proposed AC-ADMM framework. For each task, the features of execution time, CPU requirement, memory requirement, priority and execution-time uncertainty were defined. A varying processing capability was allocated to the resource clusters to cater to varying competencies of computing resources. Task arrival and resource availability were varied during the experiments to look at scheduling performance under varying operating conditions.

Depending on which approach is proposed, three types of scheduling were evaluated and used for comparison: Deterministic ADMM, Chance-Constrained ADMM, and Adaptive Chance-Constrained ADMM (AC-ADMM).

Deterministic ADMM schedules under fixed execution-time constraints, and Chance-Constrained ADMM schedules under probabilistic deadline constraints, with a fixed risk tolerance. AC-ADMM also dynamically adjusts the risk tolerance based on the priority of tasks, workload uncertainty, resource usage, and SLA criticality. In the optimization process, the Penalty parameter of ADMM, focusing on its primal and dual residuals, was also dynamically adjusted.

The experimental workload comprised a mix of heterogeneous task demands and varying uncertainty of task-execution times. The uncertainty level was varied in different experiments to test the scheduling methods in order to understand their capacities for maintaining deadline reliability. To create contention for resources, the number of tasks being accessed simultaneously was increased, and workload-spike experiments were performed to increase task arrival rates within a brief time window. Resource-failure cases were tested by decreasing the operational capacity of chosen clusters.

B. Parameter and Dimension Settings

The basic experimental parameters are listed in Table I. The workload dimension was set to 5,000 tasks for the main comparison, distributed across 8 clusters of resources. Task priorities were specified through normalized priority levels, and the uncertainty of execution times through the standard deviation of these task-execution-time estimates. The initial ADMM parameter was defined as a fixed parameter for all three methods, and for AC-ADMM, the parameter was tuned within the optimization based on the obtained primal and dual residuals.

Table 1. Experimental Parameter Settings

Parameter	**Setting**
Tasks	5,000
Resource clusters	8
Priority range	0–1
Initial ρ	1.0
μ	10
τ	2
Risk range	0.01–0.20
Max. iterations	150
Resources	CPU, Memory
Methods	ADMM, CC-ADMM, AC-ADMM

The impact of changing problem size by varying the number of tasks, while keeping the same resource cluster structure, was explored. More experiments tested various workload sparsity and uncertainty levels to see how well the optimization performed under different and more sparse workloads. To make it convenient to compare the three methods, the same task and resource characteristics were kept the same in all comparative experiments.

C. Evaluation Metrics

Deadline reliability, resource efficiency, computational cost and convergence behaviour were considered as measures for evaluation of scheduling performance. Deadline Success Rate (DSR) is the percentage of tasks completed on/ before deadline and is equal to

$$DSR = \frac{N_{\text{success}}}{N_{\text{total}}} \times 100.$$

The corresponding **SLA Violation Rate (SVR)** is given by

$$SVR = \frac{N_{\text{violation}}}{N_{\text{total}}} \times 100.$$

Since deadline violations represent failures to satisfy the scheduling requirement, the two measures are complementary.

$DSR + SVR = 100\%.$

Makespan is the amount of time needed to complete the scheduled workload. The less time a master takes to complete a workload, the smaller the makespan. Energy consumption and scheduling cost are used to denote the energy needed by the allocated resources and the resource cost needed to execute the tasks, respectively. Resource utilisation indicates the percentage of resources used while running a task.

The convergence performance was assessed by the number of ADMM iterations and convergence time. The disagreement between the local and consensus resource decision was measured by the primal residual, while the consensus variables' change between successive iterations was measured by the dual residual. Such measures proved especially significant to have tested the adaptive penalty mechanism of AC-ADMM.

The experiments were also studied at various uncertainty levels, resource contention conditions, workload spikes, and resource-failure scenarios. This allowed them to assess the scheduling efficiency and reliability of their proposed scheme to allocate resources to cope with the varying cloud environment.

VI. Performance Evaluation

A. Convergence Performance

The convergence behaviour of Deterministic ADMM, CC-ADMM, and AC-ADMM was evaluated using the number of iterations, convergence time, and final primal and dual residuals. These results are summarized in Table II. With the inclusion of the probabilistic deadline constraint, CC-ADMM is found to be requiring 126 iterations, and deterministic ADMM is found to require 118 iterations. AC-ADMM with lower residual converge within 103 iterations, as can be observed,showing the impact of adaptive penalty adjustment in the distributed optimization process.

Table 2. convergence Performance

Method	Iterations	Time (s)	Primal Residual	Dual Residual
Deterministic ADMM	118	3.84	0.0087	0.0091
CC-ADMM	126	4.27	0.0072	0.0080
AC-ADMM	103	3.51	0.0049	0.0053

B. Deadline Reliability Analysis

The influence of uncertainty in the execution time has been studied at four different levels of uncertainty. This is because the levels of uncertainty make it hard to satisfy deadlines, and the more uncertain, the lower the chance that the deadline will be met. The Table III results indicate that at all uncertainty levels, AC-ADMM has greater deadline reliability. The performance of all algorithms at very high uncertainty is 81.9% for AC-ADMM, 70.6% for CC-ADMM and 58.1% for Deterministic ADMM.

Table 3. deadline Success Under Execution-Time Uncertainty

Uncertainty Level	Deterministic ADMM (%)	CC-ADMM (%)	AC-ADMM (%)
Low	86.4	91.2	94.8
Moderate	78.7	86.5	91.7
High	69.3	79.8	87.6
Very High	58.1	70.6	81.9

C. Overall Comparative Analysis

Deadline success, SLA violation, makespan, energy consumption, scheduling cost, and resource utilization were used to evaluate the total scheduling performance. A comparison is made in Table IV. AC-ADMM gives the maximum deadline success ratio and resource utilisation, and at the same time it creates minimum SLA violations, makespan, energy consumption and cost.

Table 4. overall Scheduling Performance

Method	Deadline Success (%)	SLA Violation (%)	Makespan (s)	Energy	Cost	Utilization (%)
Deterministic ADMM	72.6	27.4	612.8	215.7	502.4	19.4
CC-ADMM	81.9	18.1	574.6	207.3	481.6	21.8
AC-ADMM	89.7	10.3	548.2	198.6	462.9	24.1

The results show that probabilistic deadline constraints can be used to increase reliability in the presence of uncertain execution conditions. An extra adaptive mechanism is added in AC-ADMM, which is used to further control risk and consensus optimization, leading to better scheduling performance based on the discussed metrics. This reduces SLA violation rate and makespan, along with better resource utilization shows a balanced allocation of cloud resources in uncertain workloads.

VII. Comparative And Robustness Analysis

The comparative study indicates that the three scheduling methods react differently to uncertainty and varying resource conditions. Distributed resource coordination is offered by a deterministic ADMM, but it is based on fixed deadline conditions, lacking adaptability to deal with variations in task execution. CC-ADMM provides probabilistic deadline protection and thus enhances its reliability, though its fixed risk tolerance is not adaptive to the change of workload characteristics. AC-ADMM is a probabilistic scheduling algorithm that integrates adaptive risk and penalty control, enabling the scheduling process to react to the uncertainty of the tasks as well as optimisation behaviour.

The overall results show that AC-ADMM achieves a deadline success rate of 89.7%, with SLA violations limited to 10.3%. The improvement is more noticeable as the execution risk rises. With a very high level of uncertainty, the deadline success rate stands at 81.9% for the case of AC-ADMM, 70.6% for the case of CC-ADMM and 58.1% for Deterministic ADMM. This means that the adaptive risk control affords better protection against workload deadline degradation when uncertainty in the workload is greater.

The robustness analysis includes enlarging the workload size. An increased number of tasks places a greater demand on the optimization problem, as more tasks and resources are allocated to each other. AC-ADMM does not exhibit any type of unstable convergence and gives 5,000 tasks a runtime of 3.51 s. The adaptive penalty approach makes the consensus process adapt to the changes in the primal and dual residuals, and therefore does not have to keep the same penalty value during the optimization.

Another observation that is very important is the balance between reliability and the efficiency of resources. Attempts to increase meeting of deadlines should not lead to unproductive uses of resources or overly constraining resource allocations. At the same time, the scheduling risk is reduced where possible in the process of adapting to that risk, and resources are allocated accordingly across clusters in a coordinated manner by AC-ADMM. The reduction in SLA violations and makespan, combined with the increased resource utilization, suggests that the framework is capable of ensuring reliable scheduling with efficient resource usage.

The comparative and robustness analysis showed that the combined adaptation of scheduling risk and optimization parameters is the primary benefit of AC-ADMM. In this way, the framework's scheduling behaviour will be more stable when both the execution uncertainty and workload size fluctuate, which is suitable in dynamic cloud resource management.

VIII. CONCLUSION AND FUTURE WORK

AC-ADMM is a new framework called Adaptive Chance-Constrained ADMM, which offers a solution for handling uncertainties in resource scheduling in a dynamic cloud computing environment. The framework also includes probabilistic deadline constraints, associated with the distributed optimization via ADMM, as well as adaptive risk control and penalty adjustments based on the residual. A Deadline Reliability Guarantee theorem was formulated to demonstrate that if the deterministic equivalent of the chance constraint is satisfied, then all the tasks can meet their deadlines with a specified target probability, given the assumed execution-time distribution. This gives a theoretical foundation for the reliability constraint in the scheduling framework.

The evaluation results indicate that compared to the other scheduling approaches, AC-ADMM is better in terms of deadline reliability, SLA compliance, convergence behaviour, and resource utilization. Through simulation results, the proposed method achieves an 89.7% deadline success rate with a 10.3% SLA violation rate and consistently converges under heavier workload uncertainty and task dimensions. The theoretical proof and experimental analysis show the contribution of probabilistic deadline control and adaptive distributed optimization in reliable scheduling of cloud resources.

Future work entails expanding the framework to heterogeneous virtual machines and other resources, including storage and network bandwidth. Continuously arriving tasks, real-time workload prediction capabilities, and more complex SLA requirements can also be applied online. Future research may involve large-scale cloud and edge-cloud platforms with dynamic resource failures and investigate the scalability and practical usability of the proposed framework.

References

1. Crisci, S., De Simone, V., & Viola, M. (2023). On the adaptive penalty parameter selection in ADMM. *Algorithms, 16*(6), 264. https://doi.org/10.3390/a16060264

2. Manam, S., Moessner, K., & Vural, S. (2023). Deadline-constrained cost minimisation for cloud computing environments. *IEEE Access, 11*, 38514–38522. https://doi.org/10.1109/ACCESS.2023.3258682

3. Wang, G., Wang, D., Li, C., Li, C.-F., & Lei, Y. (2025). The fast inertial ADMM optimization framework for distributed machine learning. *Future Generation Computer Systems, 164*, 107575. https://doi.org/10.1016/j.future.2024.107575

4. Rokade, K., & Kalaimani, R. K. (2025). Distributed ADMM with linear updates over directed networks. *IEEE Transactions on Network Science and Engineering, 12*(2), 1396–1407. https://doi.org/10.1109/TNSE.2025.3529703

5. Yan, J., Lu, Y., Chen, L., Qin, S., Fang, Y., Lin, Q., Moscibroda, T., Rajmohan, S., & Zhang, D. (2022). Solving the batch stochastic bin packing problem in cloud: A chance-constrained optimization approach. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

6. Chen, Y., Deng, Z., Zhou, Y., Chen, Z., Chen, Y., & Hu, H. (2023). An online algorithm for chance-constrained resource allocation. *arXiv*. https://arxiv.org/abs/2303.03254

7. Yamsani, Y., & Chenna Reddy, P. (2026). SLA aware deep reinforcement learning for adaptive EdgeCloud task scheduling. *Scientific Reports, 16*, 10037.

8. Krishna, M. S. R., & Dudekula, K. V. (2025). Multi-objective enhanced task scheduling algorithm for SLA violation mitigation in cloud computing using deep reinforcement learning. *International Journal of Grid and Utility Computing, 16*(5–6), 588–603. https://doi.org/10.1504/IJGUC.2025.148552

9. Mangalampalli, S., Karri, G. R., Kumar, M., *et al.* (2024). DRLBTSA: Deep reinforcement learning-based task-scheduling algorithm in cloud computing. *Multimedia Tools and Applications, 83*, 8359–8387. https://doi.org/10.1007/s11042-023-16008-2

10. Janjani, H., Agarwal, T., Gopinath, M. P., Sharma, V., & Raja, S. P. (2024). Designing energy-aware scheduling and task allocation algorithms for online reinforcement learning applications in cloud environments. *IEEE Transactions on Computational Social Systems*. https://doi.org/10.1109/TCSS.2024.3508089

11. Mehor, Y., Rebbah, M., & Smail, O. (2024). Energy-aware scheduling of tasks in cloud computing. *Informatica, 48*(16). https://doi.org/10.31449/inf.v48i16.5741

12. Prathiba, S., & Sankar, S. (2024). Energy-efficient resource allocation in cloud infrastructure using L3F-MGA and E-ANFIS. *Measurement: Sensors, 31*, 100965. https://doi.org/10.1016/j.measen.2023.100965

13. Vhatkar, K., Kathole, A. B., Lonare, S., *et al.* (2024). Energy-aware optimal virtual machine scheduling in cloud environment using hybridized egret swarm with sea lion optimization algorithm. *Neural Processing Letters, 56*, 248. https://doi.org/10.1007/s11063-024-11706-w

14. Kumar, V., Ali, A., Mittal, P., Aqeel, I., Shuaib, M., Alam, S., *et al.* (2024). E2SVM: Electricity-efficient SLA-aware virtual machine consolidation approach in cloud data centres. *PLOS ONE, 19*(6), e0303313. https://doi.org/10.1371/journal.pone.0303313

15. Li, H., Zhang, X., Li, H., Duan, X., & Xu, C. (2024). SLA-based task offloading for energy consumption-constrained workflows in fog computing. *Future Generation Computer Systems, 156*, 64–76. https://doi.org/10.1016/j.future.2024.03.013

16. Liu, W., Yan, Y., Wang, P., Wang, Z., & Ding, Z. (2024). Flexibility quantification for energy-aware data centre job scheduling: A chance-constrained bi-level model. In *Proceedings of the 2024 IEEE/IAS Industrial and Commercial Power System Asia* (pp. 760–764). IEEE. https://doi.org/10.1109/ICPSAsia61913.2024.10761863

17. Zhou, G., Tian, W., Buyya, R., Xue, R., & Song, L. (2024). Deep reinforcement learning-based methods for resource scheduling in cloud computing: A review and future directions. *Artificial Intelligence Review, 57*, 124. https://doi.org/10.1007/s10462-024-10756-9

18. Wang, G., Wang, D., Li, C., Li, C.-F., & Lei, Y. (2025). The fast inertial ADMM optimization framework for distributed machine learning. *Future Generation Computer Systems, 164*, 107575. https://doi.org/10.1016/j.future.2024.107575

19. Rokade, K., & Kalaimani, R. K. (2025). Distributed ADMM with linear updates over directed networks. *IEEE Transactions on Network Science and Engineering, 12*(2), 1396–1407. https://doi.org/10.1109/TNSE.2025.3529703

20. Crisci, S., De Simone, V., & Viola, M. (2023). On the adaptive penalty parameter selection in ADMM. *Algorithms, 16*(6), 264. https://doi.org/10.3390/a16060264