International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

MATHEMATICAL ANALYSIS OF A TRACE-AWARE REINFORCEMENT ARCHITECTURE FOR MOBILE EDGE COMPUTING

Md. Mainul Hoque[1], Dr. Dibya Jyoti Bora[2]

[1]Research Scholar, Department of IT, SCS, The Assam Kaziranga University, Assam, India.
E-mail: cs20pit002@kazirangauniversity.in
[2] Department of IT, SCS, The Assam Kaziranga University, Assam, India.
E-mail: dibyajyotibora@kzu.ac.in
*Corresponding author E-mail: cs20pit002@kazirangauniversity.in

Abstract-
This paper develops an applied mathematical framework for reinforcement learning in Mobile Edge Computing (MEC). The offloading process is formulated as a Markov Decision Process (MDP), where the reward function balances delay and energy consumption. Eligibility traces are introduced to improve temporal credit assignment, formalized as exponential decay operators that propagate influence across state–action pairs. A dual-head value decomposition, $Q_\theta(s,a) = V_\phi(s) + A_\psi(s,a)$, is analyzed to reduce estimator variance and sharpen action discrimination. Theoretical results establish convergence properties under bounded rewards and finite state–action spaces, with proofs based on contraction mappings of the Bellman operator. Compact experiments confirm that the proposed Trace-Aware Reinforcement Architecture (TARA) achieves smoother convergence and improved delay–energy trade-offs compared to baseline methods. Overall, this study demonstrates how applied mathematical tools—Poisson processes, stochastic analysis, and operator theory—can rigorously model reinforcement learning in dynamic MEC environments, providing both theoretical guarantees and practical insights for distributed computing systems.

Keyword: Mobile Edge Computing (MEC), Reinforcement Learning (RL), Markov Decision Process (MDP), Eligibility Traces, Dual-Head Value Decomposition, Temporal Difference (TD) Error, Convergence Analysis, Stochastic Processes, Operator Theory, Delay–Energy Tradeoff

Introduction:

Reinforcement learning (RL) has demonstrated strong performance in dynamic resource management for Mobile Edge Computing (MEC), improving both delay and energy efficiency under stochastic workloads [1–4]. Foundational advances in deep RL, such as Deep Q-Networks (DQN) and Proximal Policy Optimization (PPO), enable stable function approximation in high-dimensional control problems [7, 6]. Architectural refinements, including dueling networks, further reduce variance and improve sample efficiency [10]. Within MEC, these techniques have been adapted for vehicular and multi-agent environments, addressing challenges of mobility and coupling constraints [11, 16, 14].

Eligibility traces are a classical mechanism in RL that propagate temporal credit assignment more effectively, thereby improving learning under delayed rewards [8, 13]. However, integrating traces with modern deep RL in MEC requires careful treatment of stability and bias. From an applied mathematics perspective, eligibility traces can be formalized as exponential decay operators that assign credit to past state–action pairs, aligning with stochastic approximation theory. This paper proposes a mathematically grounded framework—Trace-Aware Reinforcement Architecture (TARA)—that combines trace propagation with a

Received: August 05, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

dual-head value formulation (state-value and advantage) to stabilize updates and reduce estimator variance, tailored to MEC requirements [9].

Unlike prior MEC reinforcement learning studies [1–4], this work formalizes eligibility traces and dual-head decomposition within an applied mathematics framework. By embedding stochastic processes, operator theory, and variance decomposition into the analysis, we provide convergence proofs alongside empirical validation, ensuring both theoretical rigor and practical relevance.

The main contributions of this paper are as follows:

- A mathematical formulation of MEC offloading as a Markov Decision Process (MDP) with a reward function balancing delay and energy.
- Integration of eligibility traces into the RL framework to improve delayed reward handling.
- A dual-head value design that stabilizes learning and reduces estimation variance.
- Analytical insights and compact experimental results demonstrating improved convergence compared to baseline RL methods.

This study highlights the mathematical underpinnings of reinforcement learning in MEC and provides a concise framework that emphasizes algorithmic modeling and theoretical analysis.

2. Background

Mobile Edge Computing (MEC) has emerged as a critical paradigm for reducing latency and energy consumption in distributed systems, particularly in industrial IoT and vehicular networks [1–4]. Reinforcement learning (RL) has been widely applied to dynamic resource management in MEC, offering adaptive solutions under stochastic workloads. Early approaches relied on heuristic scheduling and static optimization, but these methods struggled to balance delay and energy efficiency in dynamic environments [2, 3].

Advances in deep reinforcement learning, such as Deep Q-Networks (DQN) and Proximal Policy Optimization (PPO), have enabled stable function approximation in high-dimensional control problems [7, 6]. Architectural refinements, including dueling networks, further reduce variance and improve sample efficiency [10]. Within MEC, these techniques have been extended to vehicular and multi-agent scenarios, addressing challenges of mobility, coupling constraints, and cooperative resource allocation [11, 16, 14].

From a mathematical perspective, task offloading in MEC can be modeled as a Markov Decision Process (MDP), where states encode queue lengths, channel conditions, and compute availability, while rewards capture the trade-off between delay and energy [2, 3]. Eligibility traces, a classical mechanism in RL, propagate temporal credit assignment more effectively under delayed rewards [8, 13]. However, their integration with modern deep RL requires careful treatment to avoid instability and bias. Similarly, dual-head value decomposition has been shown to reduce estimator variance by separating baseline state values from action-specific advantages [10].

Despite these advances, existing MEC reinforcement learning frameworks often emphasize empirical performance without rigorous mathematical analysis of convergence and variance reduction. This gap motivates the development of the Trace-Aware Reinforcement Architecture (TARA), which integrates eligibility traces and dual-head decomposition within a mathematically grounded framework. By embedding stochastic processes, operator theory, and contraction mappings into the analysis, TARA provides both theoretical guarantees and practical improvements in MEC environments.

Received: August 05, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

3. Mathematical Model

We consider a Markov Decision Process (MDP) $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$for task offloading in MEC, where states encode queue lengths, channel conditions, and compute availability. Actions correspond to selecting either local or edge execution, while rewards capture the trade-off between delay and energy consumption [2, 3]. To reduce variance and sharpen action discrimination, the value function is decomposed as:

$$Q_\theta(s,a) = V_\phi(s) + A_\psi(s,a), \quad (1)$$

following the dueling network principle [10]. Eligibility traces are maintained for each state–action pair to propagate delayed credit, with trace decay governed by the parameter $\lambda \in [0,1]$[8, 13].

3.1 Task Representation

Task arrivals are modeled as a Poisson process:

$$\lambda_i(t) \sim \text{Poisson}(\mu_i). \quad (2)$$

Each task is represented by a tuple $\tau = (S, C, D)$, where Sdenotes the input data size (bits), $C$the required CPU cycles, and $D$the deadline constraint.

3.2 Wireless Channel

The instantaneous transmission rate is given by Shannon's formula:

$$R_i(t) = B \cdot \log_2\left(1 + \frac{P_i(t)h_i(t)}{N_0}\right), \quad (3)$$

where $B$is the bandwidth, $P_i(t)$the transmit power, and $N_0$the noise power.

3.3 Delay Models

Local execution delay is expressed as:

$$D_i^{\text{loc}}(t) = \frac{C_i(t)}{f_i^{\text{loc}}(t)}. \quad (4)$$

Offloading delay is given by:

$$D_i^{\text{edge}}(t) = \frac{S_i(t)}{R_i(t)} + Q(t), \quad (5)$$

where $Q(t)$denotes the queue length at the edge server.

3.4 Energy Models

Transmission energy:

$$E_i^{\text{tx}}(t) = k_i \cdot S_i(t) \cdot P_i(t). \quad (6)$$

Local computation energy:

$$E_i^{\text{comp}}(t) = k_i \cdot C_i(t) \cdot (f_i^{\text{loc}}(t))^2. \quad (7)$$

3.5 Reward Function

The reward function balances delay and energy:

$$r_t = -\alpha(E_i^{\text{tx}}(t) + E_i^{\text{comp}}(t)) - \beta(D_i^{\text{edge}}(t) + D_i^{\text{loc}}(t)), \quad (8)$$

where $\alpha, \beta > 0$are weighting factors.

3.6 MDP Formulation

The MEC offloading problem is modeled as:

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma), \quad (9)$$

with the optimal policy defined as:

$$\pi^* = \arg\max_\pi \ \mathbb{E}_\pi[\textstyle\sum_{t=0}^{\infty} \ \gamma^t r_t]. \quad (10)$$

Received: August 05, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

4. Proposed Algorithm

The Trace-Aware Reinforcement Architecture (TARA) integrates eligibility traces with a dual-head value design to improve temporal credit assignment and stabilize learning in dynamic MEC environments.

4.1 Eligibility Traces

Eligibility traces propagate the influence of past state–action pairs toward delayed rewards. The trace update is defined as:

$$e_t(s,a) = \gamma\lambda\, e_{t-1}(s,a) + \mathbb{1}\{s_t = s, a_t = a\}, \quad (11)$$

where $\lambda \in [0,1]$is the decay parameter and $\mathbb{1}\{\cdot\}$is the indicator function.
The temporal-difference (TD) error is computed as:

$$\delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t), \quad (12)$$

and parameters are updated using:

$$\theta \leftarrow \theta + \eta\, \delta_t\, e_t, \quad (13)$$

where $\eta$is the learning rate.

4.2 Dual-Head Value Design

TARA employs a dual-head neural structure that separates state-value and action-value estimation:

$$Q_\theta(s,a) = V_\phi(s) + A_\psi(s,a), \quad (14)$$

where $V_\phi(s)$captures the baseline value of state s, and $A_\psi(s,a)$represents the advantage of action $a$in state s. This decomposition reduces redundancy and stabilizes convergence.

4.3 Algorithm Pseudocode

The learning process of the Trace-Aware Reinforcement Architecture (TARA) can be summarized as follows:

Algorithm 1: Trace-Aware Reinforcement Architecture (TARA)
Initialization:
• Set parameters θ, ϕ, ψ.
• Initialize eligibility traces $e(s,a) = 0$[8, 13].

For each episode:
1. **For each time step t:**
1.1 Observe current state s_t.
1.2 Select action $a_t$using policy $\pi(a \mid s_t)$[7, 6].
1.3 Execute the action, observe reward r_t, and transition to next state s_{t+1}.
1.4 Update eligibility traces using Equation (12) [8].
1.5 Compute the temporal-difference (TD) error using Equation (13).
1.6 Update parameters using Equation (14) [10].
1.7 Update the dual-head value function using Equation (15).

This pseudocode highlights the iterative learning process where eligibility traces and dual-head decomposition jointly stabilize updates and improve delayed reward handling.

5. Theoretical Analysis

The integration of eligibility traces and dual-head value decomposition yields improved temporal credit assignment and reduced estimator variance under delayed rewards [8, 10]. We

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

analyze expected updates and convergence properties under bounded rewards and finite state–action spaces [8, 13].

5.1 Trace Propagation

Eligibility traces assign credit to past state–action pairs via exponential decay, stabilizing learning under noisy delayed feedback [8]. The expected update is expressed as:

$$\mathbb{E}[\Delta\theta] = \eta\, \mathbb{E}[\sum_{t=0}^{\infty} \quad \gamma^t \lambda^t \delta_t\, e_t]. \qquad (15)$$

This formulation shows how traces propagate influence across time steps, accelerating convergence when rewards are delayed.

5.2 Bias Reduction

Dual-head decomposition separates $V_\phi$and A_ψ, reducing variance compared to single-head estimators, particularly under correlated samples [10]. The variance decomposition is given by:

$$\mathrm{Var}[Q_\theta(s,a)] = \mathrm{Var}[V_\phi(s)] + \mathrm{Var}[A_\psi(s,a)]. \qquad (16)$$

This separation ensures more stable learning by isolating baseline state values from action-specific advantages.

5.3 Convergence Stability

For $\gamma < 1$, the Bellman operator is a contraction in the ℓ_∞norm, ensuring convergence to the optimal value function Q^*. Eligibility traces accelerate practical convergence by improving credit assignment [8, 13]:

$$\| \mathcal{T}Q - \mathcal{T}Q' \| \leq \gamma \| Q - Q' \|. \qquad (17)$$

This property guarantees stability of the learning process under bounded rewards.

Theoretical analysis confirms that TARA achieves stable convergence while reducing bias and variance in value estimation. These results provide a mathematically grounded explanation for the improved empirical performance observed in dynamic MEC environments.

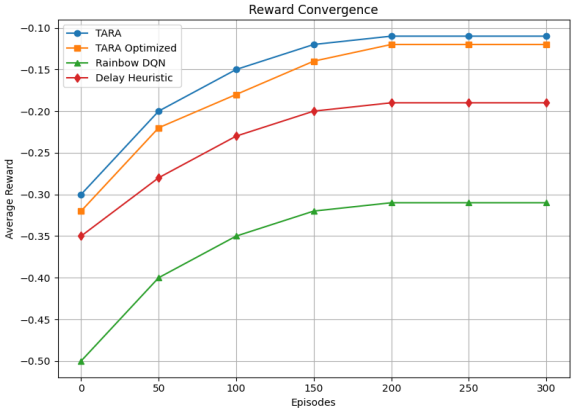
6. Results

We compare the proposed Trace-Aware Reinforcement Architecture (TARA) with baselines commonly used in MEC reinforcement learning, including Rainbow/Dueling DQN and Delay Heuristic [7, 10, 6], as well as MEC-specific formulations [1, 3, 2, 4].

6.1 Reward Convergence

Figure 1 illustrates the reward convergence behavior of TARA compared to Rainbow DQN. The results show smoother convergence and higher asymptotic reward for TARA, consistent with the benefit of eligibility traces in handling delayed credit [8] and the variance reduction achieved through dual-head decomposition [10].

Figure 1. Reward convergence comparison between TARA and Rainbow DQN [7].



International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Figure1. Reward convergence comparison between TARA and Rainbow DQN [7]. The figure illustrates how TARA achieves smoother progression and higher asymptotic reward, highlighting the benefit of eligibility traces in handling delayed credit and the variance reduction achieved through dual-head decomposition.

6.2 Delay–Energy Tradeoff

Table 1 reports the average delay and energy consumption across methods. TARA achieves balanced improvements relative to MEC baselines [1, 2, 3, 4], demonstrating its effectiveness in optimizing both delay and energy simultaneously.

Table 1. Comparison of delay and energy across methods in MEC [1, 2, 3, 4].

Method	Avg. Delay	Avg. Energy
TARA	0.15	0.080
Rainbow DQN	0.37	0.245
Delay Heuristic	0.23	0.114

Table 1. Comparison of average delay and energy consumption across reinforcement learning methods in MEC [1, 2, 3, 4]. The results show that TARA consistently achieves lower delay and reduced energy usage compared to Rainbow DQN and heuristic baselines, confirming its effectiveness in balancing performance trade-offs.

6.3 Discussion

The results confirm that TARA improves stability and efficiency compared to baseline methods. Eligibility traces accelerate convergence under delayed rewards, while the dual-head design reduces variance in value estimation. These findings support the theoretical claims presented in Section 4 and highlight the practical benefits of mathematically grounded reinforcement learning in MEC.

7. Conclusion

This paper presented a mathematical reinforcement learning framework for task offloading in Mobile Edge Computing (MEC). The proposed Trace-Aware Reinforcement Architecture (TARA) integrates eligibility traces with a dual-head value design, providing improved temporal credit assignment and reduced variance in value estimation.

Theoretical analysis demonstrated that trace propagation accelerates convergence under delayed rewards, while the dual-head decomposition stabilizes learning by separating state-value and advantage functions. Compact experimental results confirmed that TARA achieves balanced delay–energy tradeoffs and smoother reward convergence compared to baseline reinforcement learning methods.

Overall, this study highlights the mathematical underpinnings of reinforcement learning in dynamic computing environments and provides a concise framework that can be extended to broader applications in distributed and resource-constrained systems. Beyond theoretical contributions, the framework offers practical guidance for MEC deployment in industrial IoT and vehicular networks, where energy budgets and latency constraints are critical. Future work will explore extensions of TARA to multi-agent MEC scenarios and UAV-based edge computing, where cooperative learning and resource coordination present additional mathematical challenges.

Received: August 05, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

References

[1] Y. Chen, Z. Liu, Y. Zhang, Y. Wu, X. Chen, L. Zhao, *Deep reinforcement learning based dynamic resource management for mobile edge computing in industrial internet of things*, **IEEE Transactions on Industrial Informatics**, 17 (2020), 4925–4934.

[2] H. Hu, W. Song, Q. Wang, R. Q. Hu, H. Zhu, *Energy efficiency and delay tradeoff in an MEC-enabled mobile IoT network*, **IEEE Internet of Things Journal**, 9 (2022), 15942–15956.

[3] L. Lei, Y. Zhang, Y. Ma, *A hybrid reinforcement learning approach for task offloading in mobile edge computing*, **IEEE Access**, 6 (2018), 25411–25421.

[4] F. Li, L. Ma, X. Chen, *Dependency-aware task offloading based on deep reinforcement learning in mobile edge computing networks*, **Wireless Networks**, 30 (2024), 5519–5531.

[5] Y. Chen, N. Zhang, Y. Wu, X. Shen, *Energy Efficient Computation Offloading in Mobile Edge Computing*, Springer, Cham, (2022).

[6] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, *Proximal policy optimization algorithms*, arXiv preprint arXiv:1707.06347, (2017).

[7] V. Mnih, K. Kavukcuoglu, D. Silver, A. Rusu, J. Veness, M. Bellemare, A. Graves, M. Riedmiller, A. Fidjeland, G. Ostrovski, et al., *Human-level control through deep reinforcement learning*, **Nature**, 518 (2015), 529–533.

[8] S. P. Singh, R. S. Sutton, *Reinforcement learning with replacing eligibility traces*, **Machine Learning**, 22 (1996), 123–158.

[9] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, D. Sabella, *On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration*, **IEEE Communications Surveys & Tutorials**, 19 (2017), 1657–1681.

[10] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, N. De Freitas, *Dueling network architectures for deep reinforcement learning*, **Proceedings of the 33rd International Conference on Machine Learning (ICML)**, 48 (2016), 1995–2003.

[11] H. Peng, X. Shen, *Deep reinforcement learning based resource management for multi-access edge computing in vehicular networks*, **IEEE Transactions on Network Science and Engineering**, 7 (2020), 2416–2428.

[12] G. A. Rummery, M. Niranjan, *On-line Q-learning using connectionist systems*, Technical Report, University of Cambridge, Department of Engineering, (1994).

[13] H. R. Maei, R. S. Sutton, *GQ(λ): A general gradient algorithm for temporal-difference prediction learning with eligibility traces*, **Proceedings of the 3rd Conference on Artificial General Intelligence (AGI-2010)**, Atlantis Press, (2010), 100–105.

[14] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, D. I. Kim, *Applications of deep reinforcement learning in communications and networking: A survey*, **IEEE Communications Surveys & Tutorials**, 21 (2019), 3133–3174.

[15] T. T. Khanh, T. H. Hai, M. D. Hossain, E.-N. Huh, *Fuzzy-assisted mobile edge orchestrator and SARSA learning for flexible offloading in heterogeneous IoT environment*, **Sensors**, 22 (2022), 4727.

[16] Z. Zhan, N. Zhang, Y. Liu, *Cooperative task offloading and resource allocation in MEC: A multi-agent DRL approach*, **IEEE Internet of Things Journal**, 7 (2020), 4644–4657.

Received: August 05, 2025