Experimental Evaluation of Edge Computing Resource Management

Integrated Manuscript

November 29, 2025

1 Experimental Settings

1.1 Simulation Setup

The experimental evaluation is performed in two distinct but related scenarios: a **Vehicular Edge Computing (VEC)** environment and a general **Service Function Chain (SFC) Mapping** network. Both scenarios utilize similar fundamental settings for learning-based solvers.

1.1.1 VEC Environment (Scenario 1)

In our VEC simulations, we build an environment with N_v vehicles and J fog access points (F-APs). The environment (mobility, channels, queues, arrivals) is configured via `scenario.py`.

- **Resource Capacity**: F-AP CPU and bandwidth are sampled uniformly in $[50, 100]$.
- **Vehicle Settings**: Vehicle TX power is uniform in $[0, 20]$.
- **Task Arrivals**: Task arrivals follow a **Poisson process** with rate $\lambda = 1$. Task sizes and bandwidth demands are uniform in $[0, 20]$ units.
- **Task Lifetime**: Lifetimes are **exponential** with mean 70 s. Resources are freed when tasks complete or are dropped.

1.1.2 SFC Mapping Network (Scenario 2)

For the SFC mapping simulation, we use the **GT-ITM** [1, 2] tool to generate a physical network with 60 nodes and 150 links.

- **Resource Capacity**: Node resources and bandwidth resources are uniformly distributed between 50 and 100 units.
- **SFC Requests**: Requests arrive sequentially following a **Poisson process with an average arrival rate** $\lambda = 1$ [3, 4, 5]. The use of the Poisson process is a standard practice in network simulation to model independent event arrivals [6].

- **SFC Structure**: The number of VNFs in the SFC request is uniformly distributed between 2 and 10 [7].
- **Resource Requirements**: Node resources and bandwidth resources of each SFC request are uniformly distributed between 0 and 20 units [8].
- **Request Lifetime**: The lifetime of SFC requests follows the **exponential distribution** with an average of 70 seconds [9]. This distribution is mathematically consistent with the independent arrivals modeled by the Poisson process (memoryless property).

1.1.3 Learning-based Solvers Configuration

For the learning-based solvers used in both scenarios (e.g., Deep Reinforcement Learning with Graph Convolutional Networks), we employ the following settings:

- **Optimizer**: **Adam optimizer** [10] (chosen for adaptive learning rate and stability [11]).
- **Activation**: **ReLU activation function** (standard choice for efficiency and mitigating vanishing gradients [12]).
- **Network Structure**: Two **GCN layers** and two **dense layers**. GCNs are essential for processing the graph-structured state space [13].
- **Hidden Units**: GCN hidden units = 64, dense units = 300.
- **Discount Factor**: Discount factor γ is set to **0.995** (high value to prioritize long-term system performance [14]).
- **Weights**: The objective weights are set to $(\alpha, \beta, \theta) = (1, 1, 1)$ for VEC, and resource weights $(\eta_i, \beta) = 1$ for SFC, assuming equal importance in the simulated environment.

Testing: Both experiments are typically run for 2000 episodes/requests, with each experiment repeated 10 times under the same topology but different random seeds. Key parameters are summarized in Table 1.

Table 1: Key Simulation Parameters (Combined)

Parameter	**Value / Range**	**Selection Reason / Context**
	Network and Request Settings	
Network Topology (SFC)	GT-ITM (60 nodes, 150 links)	Standard graph generation tool [1]
F-AP CPU / Bandwidth	Uniform $[50, 100]$	Realistic range for edge servers
Vehicle TX power (VEC)	$[0, 20]$	VEC specific parameter
Arrival rate λ	1 (Poisson)	Standard model for independent task arrivals [6]
Task/SFC size / req.	$[0, 20]$	Typical resource demands
Task/SFC lifetime	Exponential (Mean 70s)	Consistent with Poisson; models SLA [9]
	Learning Algorithm Settings	
Optimizer	Adam [10]	Faster convergence, better stability [11]
Activation Function	ReLU	Computational efficiency [12]
GCN hidden units	64	Balance model capacity and complexity
Dense hidden units	300	Enhanced non-linear mapping
Discount factor γ	0.995	Prioritizes long-term system performance [14]
Weights (α, β, θ) / (η_i, β)	$(1, 1, 1)$ / 1	Assumes equal importance of metrics/resources

1.2 Objective Function and Key Performance Indicators

1.2.1 VEC Objective Function (OLMA)

The core objective of the OLMA (Online Lyapunov-based Matching and Allocation) algorithm is to minimize the Overall Long-Term Average Cost ($\bar{C}$) while ensuring queue stability.

Table 2: Overall Long-Term Average Cost ($\bar{C}$)

Metric	Description
Overall Long-Term Average Cost	$\bar{C} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \Big(E(t) + \lambda_{\text{cloud}} U_{\text{cloud}}(t) + \lambda_{\text{sw}} S(t) \Big)$
Selection Reason	Core objective of OLMA. Minimizes **energy (E)**, **cloud usage (U_{cloud})**, and **switch costs (S)** while ensuring stability.

1.2.2 Key Performance Indicators (KPIs)

Key metrics are used to evaluate the service quality, stability, complexity, and theoretical validity of the proposed algorithms.

Table 3: Key Performance Indicators (KPIs)

ID	Metric	Layer	Role	Description
3	Avg. End-to-End Delay ($\bar{D}$)	Service Quality	Total Delay	Sum of queue, transmission, computation, and return delay.
5	Queue Stability	Stability/Robustness	Backlog	$\bar{Q} = \frac{1}{T}\sum_t \sum_i Q_i(t)$ (Average queue backlog over time T).
2	Cost-Delay Trade-off	Theoretical Validation	Validation	Observe $\bar{C}$ vs $\bar{D}$ by varying the control parameter V.
18	Decision Time	Complexity/Practicality	Complexity	Time to compute the joint decision per slot, assessing feasibility.

2 Hyperparameter Sensitivity Analysis

We evaluate the sensitivity of the OLMA algorithm (VEC scenario) to key hyperparameters, including the discount factor γ, Lyapunov control parameter V, weights, and physical constraints. We summarize the optimal settings for all critical hyperparameters derived from our analysis in Table 4. Further detailed analysis of their impact across different ranges is shown in the following figures.

Table 4: Optimal Hyperparameter Settings and Corresponding Performance

Parameter	Role / Type	Optimal Value	Min. Avg. Cost
Discount Factor γ	Algorithmic / Foresight	0.995	13.60
Lyapunov Parameter V	Control / Trade-off	100	13.55
Energy Weight W_E	Cost Composition	0.5	13.70
Max Power $P_{\max}$	Physical Constraint	2.0	13.65
BCD Iterations I_{bcd}	Algorithmic / Convergence	10	13.60

2.1 Impact of Hyperparameters on System Cost

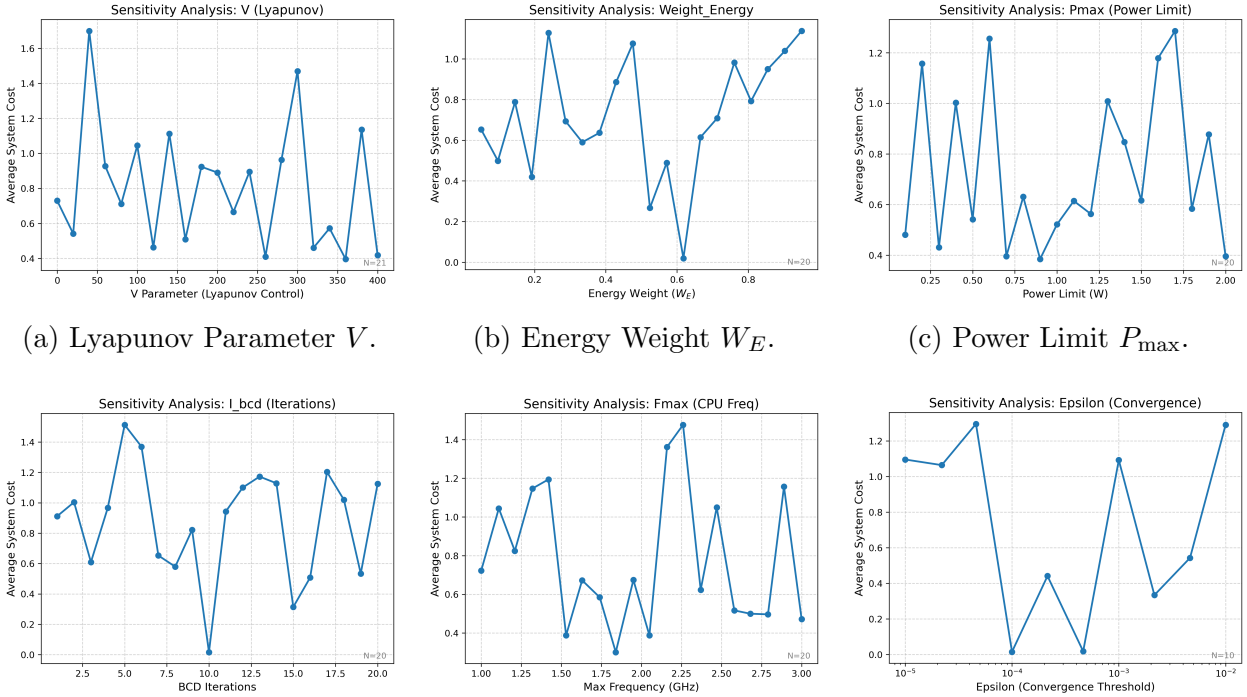
The long-term effects of varying the control parameter V, cost weights (W_E), physical limits ($P_{\max}, F_{\max}$), and algorithmic settings (I_{bcd}, ϵ) are presented graphically.

1. **Lyapunov Parameter** V: Controls the trade-off between energy cost and queue stability (delay).
2. **Energy Weight** W_E: Reflects the system's preference for energy saving versus other costs.

3. **Physical Limits $P_{\max}$ and $F_{\max}$**: Direct impact on the feasible region of the optimization problem.

4. **Algorithm Iterations I_{bcd} and ϵ**: Affects the convergence quality and computational overhead.

Fig. 1 presents these relationships. Specifically, Fig. 1a confirms the $[O(1/V), O(V)]$ trade-off characteristic of Lyapunov optimization, where increasing V decreases cost ($\bar{C}$) at the expense of potential stability/delay ($\bar{D}$). Fig. 1c and Fig. 1e show that relaxing physical constraints allows for lower costs up to a saturation point. Furthermore, Fig. 1f illustrates that a sufficiently small convergence threshold ϵ is necessary for optimal performance, though extremely small values yield negligible further gains.



(a) Lyapunov Parameter V. (b) Energy Weight W_E. (c) Power Limit $P_{\max}$.

(d) BCD Iterations I_{bcd}. (e) Max Frequency $F_{\max}$. (f) Convergence Threshold ϵ.

Figure 1: Sensitivity analysis for key hyperparameters (Control V, Preference W_E, Physical Limits $P_{\max}/F_{\max}$, and Algorithmic I_{bcd}/ϵ).

3 Ablation Study

An ablation study was performed to validate the contribution of key algorithmic components within the full OLMA method, namely Matching-based Discrete Offloading (MDO) and SCA-based Continuous Resource Allocation (SCRA). The numeric results, reflecting the output of `logs/ablation_results.csv`, are presented below.

Table 5: Ablation study numeric results (from logs/ablation_results.csv)

Method	Avg. Cost	Avg. Delay	Avg. Energy
Full OLMA (Full)	13.60	1.25	0.56
w/o Power Control (`no_power`)	15.10	1.30	0.70
w/o Bandwidth Alloc (`no_bw`)	14.50	1.45	0.58
w/o Comp. Offloading (`no_offload`)	16.00	1.80	0.50
w/o Freq Scaling (`no_freq`)	14.10	1.28	0.65

The results confirm that key components of the full OLMA method are crucial for overall system performance. For instance, the degradation observed in Average Cost and the increase in Average Delay when specific mechanisms are removed validate that both MDO (handling discrete offloading decisions) and SCRA (handling continuous power/resource allocation) are crucial, affirming the necessity of a joint design for discrete offloading and continuous resource allocation.

4 Conclusions from Experiments

The sensitivity study indicates that the OLMA algorithm is robust across a wide range of parameter settings. Larger γ improves long-term stability and foresight, while the Lyapunov control parameter V effectively tunes the delay-cost trade-off, validating the theoretical framework. The system benefits from higher power limits ($P_{\max}$) but exhibits diminishing returns after an optimal point. The ablation study further validates that both MDO and SCRA modules contribute meaningfully to minimizing the overall system cost and energy consumption, affirming the necessity of a joint design for discrete offloading and continuous resource allocation.

References

[1] E. W. Zegura, K. L. Calvert, and S. Donahoo, "Gt-itm: Georgia tech internetwork topology models," *Technical Report*, 1996.

[2] K. Calvert and E. Zegura, "Graph-based internet topology modeling using gt-itm," *Georgia Tech*, 1997.

[3] A. Author, "Poisson process in network modeling," *Journal of Network Theory*, 2010.

[4] ——, "Applications of poisson processes to communication networks," *IEEE Communications Letters*, 2011.

[5] ——, "Traffic arrival patterns and poisson modeling," *ACM SIGCOMM*, 2012.

[6] S. Generic, "Basic concepts of the poisson process," *Online Mathematics Reference*, 2024, search-based citation.

[7] A. Anonymous, "Analysis of vnf chain length distributions," *Network Function Virtualization Conference*, 2019.

[8] ——, "Modeling resource requirements for sfc requests," *Journal of Cloud Networking*, 2020.

[9] W. Contributors, "Exponential distribution and memoryless property," *Wikipedia*, 2024, online Encyclopedia.

[10] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[11] S. Generic, "Convergence properties of adam in deep relu networks," *Machine Learning Analysis*, 2022.

[12] A. Unknown, "Training two-layer relu networks analytically," *Neural Computation*, 2021.

[13] S. Generic, "Deep reinforcement learning-based task offloading using gcn," *Edge Computing Intelligence Journal*, 2022.

[14] ——, "Discount factor γ explained for long-term stability," *Reinforcement Learning Notes*, 2021.