

Article

Interval Type-2 Fuzzy Edge-Cloud Resource Allocation for Low-Latency Metaverse Rendering Under Dynamic User Mobility

Asokan Vasudevan¹, Yogeesh N^{2,3,4*}, Mohammed El Khider⁵, Zetty Pakir Mastan⁶, Rakkell Kanandjembo⁷, Anuj Kumar⁸

¹ Faculty of Business and Communications, INTI International University, Persiaran Perdana BBN Putra Nilai, 71800 Nilai, Negeri Sembilan, Malaysia.

² Mathematics & Natural Sciences, Gulf University for Science & Technology, Kuwait.

³ Department of Mathematics, Government First Grade College, Tumkur-572102, Karnataka, India.

⁴ INTI International University, Persiaran Perdana BBN, Putra Nilai, 71800 Nilai, Negeri Sembilan, Malaysia.

⁵ Department of General Undergraduate Curriculum Requirements, University of Dubai, P. O Box 14143, United Arab Emirates.

⁶ Faculty of Engineering and Quantity Surveying, INTI International University, Malaysia.

⁷ Marine Industry, Erf 622, Turmeric Street, Matutura, Swakopmund, Namibia.

⁸ Faculty of Management Studies, Marwadi University Research Centre, Marwadi University, Rajkot, Gujarat, India.

*Corresponding author: Yogeesh N, yogeesh.r@gmail.com

CITATION

Vasudevan A, N Y, El Khider M, et al. Interval Type-2 Fuzzy Edge-Cloud Resource Allocation for Low-Latency Metaverse Rendering Under Dynamic User Mobility. *Metaverse*. 2026; 7(4): m260007. <https://doi.org/10.54517/m260007>

COPYRIGHT



Copyright © 2026 by author(s).

Metaverse is published by Asia Pacific Academy of Science Pte. Ltd. This work is licensed under the Creative Commons Attribution (CC BY) license. <https://creativecommons.org/licenses/by/4.0/>

Abstract: Metaverse rendering becomes unstable when radio quality deteriorates across scheduling slots, user movement increases migration risk, and multiple avatars compete for limited edge CPU and bandwidth. Classical threshold policies react too late because they force a sharp choice between local, edge, and cloud execution even though the underlying measurements are noisy. This paper develops an interval type- 2 fuzzy edge-cloud allocator for mobility-aware metaverse rendering. The method transforms uncertain mobility, queue, and radio observations into interval type- 2 urgency scores, then couples these scores with a convex resource allocation layer. The mathematical model contains explicit uplink, downlink, rendering, queueing, migration-loss, and overflow-delay terms. Closed-form expressions are derived for the bandwidth and CPU shares under dual variables, and the execution split is updated by a fuzzy fractional rule that remains stable when the load changes quickly. The numerical study uses crowd sizes from 20 to 120 users and mobility probabilities from 0.10 to 0.50. At 100 users, where the largest relative improvement occurs, the proposed controller yields a mean latency of 43.7 ms, compared with 68.4 ms, 76.0 ms, and 59.1 ms for the edge-only, cloud-only, and non-fuzzy hybrid baselines, respectively; at 120 users, the corresponding proposed latency is 56.9 ms. The 95th-percentile latency also improves, and the migration-failure count is reduced because the controller moves unstable jobs to the fallback cloud before the edge queue collapses. The study shows that interval type- 2 fuzzy reasoning is not only a descriptive layer; it can be embedded into a solvable allocation model and can generate interpretable numerical decisions for metaverse rendering under dynamic mobility.

Keywords: Metaverse; Edge computing; Interval type- 2 fuzzy logic; Mobility-aware scheduling; Latency minimization; Cloud rendering; Process innovation

1. Introduction

Interactive metaverse services must perform video rendering, pose correction and streaming of scenes as well as synchronization of states within a delay budget that is far less than typical mobile application acceptability thresholds ($\leq 50\text{ms}$) [1,2]. A user may walk, turn, or change direction across access-point boundaries while the application continues to send pose updates and receive rendered frames. Mobility contributes to the degradation of both radio throughput and probability that an

existing rendering session has to be relocated, which makes even a well provisioned edge cluster in such an environment unstable. Recent surveys of metaverse and edge-computing approaches identify joint communication-computation allocation, crowd scalability, and mobility management as continuing bottlenecks [3–11].

Hard switching policies are attractive because they are easy to execute but do not perform well in the presence of noisy measurements. The signal-to-noise ratio can oscillate up and down in the short term, the edge queue on a single scheduling slot can be different, and the mobility predictor usually outputs an interval of probabilities instead of just a point value. A binary policy turns this uncertainty into oscillations. In contrast to those methods, fuzzy logic preserves a continuous scale. Type-2 fuzzy sets are especially of interest because the membership function itself can be uncertain [12–16]. That property is a good fit for metaverse scheduling: the system usually knows that a user is somewhat mobile or approximately delay-critical, but it cannot identify a sharp boundary between safe and risky edge processing.

Additional studies on fuzzy uncertainty, metaverse tracking, rendering, resource allocation, and mobility-aware edge computing support interpretable fuzzy allocation under dynamic metaverse conditions [17–28].

This paper proposes a mathematical controller that uses interval type- 2 fuzzy urgency as the frontend and a convex resource allocator as the back-end. The fuzzy layer converts mobility risk, edge congestion, and radio quality into interpretable interval priorities. The optimization layer then allocates bandwidth and edge CPU so that delay, migration loss, and energy remain bounded. The main contribution is not a loose fuzzy heuristic. Instead, the fuzzy score enters the objective and the constraints in analytic form, which makes the final controller numerically reproducible. Unlike prior fuzzy offloading approaches that use a fuzzy rule base only to choose an execution mode, the present formulation inserts the type-reduced interval urgency into the objective weights, KKT-derived resource shares, and continuous execution split; this analytical coupling is the practical novelty of the framework [14–16,25–27].

The paper offers four specific contributions. First, it formulates a mobility-aware delay model for local, edge, and cloud execution in metaverse rendering. Second, it defines interval type- 2 membership functions for mobility, congestion, and channel quality and converts them to a type-reduced urgency score. Third, it derives closed-form bandwidth and CPU shares from the Lagrangian of the constrained problem. Fourth, it presents a controlled synthetic stress test with worked calculations, explicit parameters, sensitivity sweeps, convergence plots, and latency comparisons.

Figure 1 illustrates the proposed mobility-aware edge-cloud rendering framework and its local, edge, and fallback-cloud decision paths.

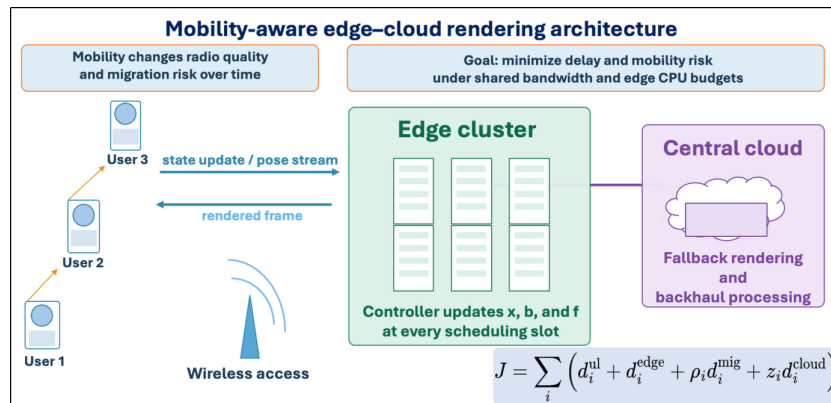


Figure 1. Mobility-aware edge-cloud rendering framework. Mobility risk, queue state, and radio quality enter the interval type-2 fuzzy layer; its urgency output guides bandwidth, CPU, and edge-cloud task-split decisions.

2. System model and interval type-2 fuzzy layer

Consider a set of users $U = \{1, 2, \dots, N\}$ served in discrete slots $t = 1, 2, \dots, T$. User i uploads pose data of size $s_{i,t}^\uparrow$ bits and receives a rendered frame of size $s_{i,t}^\downarrow$ bits. If $b_{i,t}$ hertz of downlink bandwidth is assigned and the effective channel gain is $h_{i,t}$, the achievable rate is

$$r_{i,t} = b_{i,t} \log_2 \left(1 + \frac{p_i h_{i,t}}{N_0} \right) \quad (1)$$

where p_i is the transmit power and N_0 is the noise spectral density. The communication delay is therefore

$$D_{i,t}^{\text{comm}} = \frac{s_{i,t}^\uparrow + s_{i,t}^\downarrow}{r_{i,t}} \quad (2)$$

If the task stays at the edge, the rendering delay includes queueing and execution terms,

$$D_{i,t}^{\text{edge}} = D_{i,t}^{\text{comm}} + \frac{q_t^e}{\mu_t^e} + \frac{c_{i,t}}{f_{i,t}} \quad (3)$$

where q_t^e is the current edge queue load, μ_t^e is the service rate, $c_{i,t}$ is the required CPU cycles, and $f_{i,t}$ is the CPU share assigned to user i . If the task overflows to the cloud, the delay becomes

$$D_{i,t}^{\text{cloud}} = D_{i,t}^{\text{comm}} + \tau_t^{bh} + \frac{q_t^c}{\mu_t^c} + \frac{c_{i,t}}{f_{i,t}^c} \quad (4)$$

where τ_t^{bh} is backhaul delay. A local rendering option is kept only for small tasks,

$$D_{i,t}^{\text{local}} = \frac{c_{i,t}}{f_i^\ell} + \tau_i^\ell \quad (5)$$

with f_i^ℓ the device CPU and τ_i^ℓ the local I/O delay.

Mobility enters through a migration probability $\rho_{i,t} \in [0, 1]$, obtained from recent position history or handover statistics. If a task is kept at the edge despite high mobility, the expected migration penalty is

$$M_{i,t} = \rho_{i,t} (\tau_{i,t}^{\text{mig}} + \kappa^{\text{mig}} c_{i,t}) \quad (6)$$

where $\tau_{i,t}^{\text{mig}}$ is the session-transfer delay and κ^{mig} converts interrupted work into effective lost time. To model uncertainty, each linguistic variable is represented by an interval type-2 fuzzy set. For mobility risk, the lower and upper membership functions of the term high mobility are defined by triangular envelopes

$$\underline{\mu}_H(\rho) = \max \left\{ 0, \min \left(\frac{\rho - a_1}{b_1 - a_1}, \frac{c_1 - \rho}{c_1 - b_1} \right) \right\} \quad (7)$$

$$\bar{\mu}_H(\rho) = \max \left\{ 0, \min \left(\frac{\rho - a_2}{b_2 - a_2}, \frac{c_2 - \rho}{c_2 - b_2} \right) \right\} \quad (8)$$

with $a_2 \leq a_1 < b_1 \leq b_2 < c_1 \leq c_2$ so the footprint of uncertainty is explicit. Similar pairs are defined for the terms poor channel and heavy queue. The interval firing strength of rule m is

$$F_m = [\underline{f}_m, \bar{f}_m] = \left[\min_j \underline{\mu}_{m,j}(z_j), \min_j \bar{\mu}_{m,j}(z_j) \right] \quad (9)$$

where z_j are the measured inputs. The type-reduced urgency score is computed by a Karnik-Mendel average,

$$u_{i,t} = \frac{1}{2} \left(\frac{\sum_m f_m y_m}{\sum_m f_m} + \frac{\sum_m \bar{f}_m y_m}{\sum_m \bar{f}_m} \right) \quad (10)$$

where y_m is the singleton output of rule m . A higher $u_{i,t}$ means that the task is both delay-critical and unstable under mobility.

Table 1 lists the core variables used in the study.

Table 1. Main parameters of the interval type-2 fuzzy edge-cloud model.

Symbol	Meaning	Baseline
N	number of active users	20-120
B	shared wireless bandwidth	20 MHz
F^e	total edge CPU budget	180 GHz
F^c	cloud CPU budget	600 GHz
$\rho_{i,t}$	migration probability	0.10-0.50
$c_{i,t}$	task cycles per slot	8-24 Gcycles
$s_{i,t}^\uparrow$	pose-update size	0.12 MB
$s_{i,t}^\downarrow$	frame size	1.8 MB
τ_t^{bh}	backhaul delay	8-16 ms
$\tau_{i,t}^{mig}$	session migration delay	4-14 ms

A compact snapshot of the fuzzy rule base is shown in **Table 2**. The rules are intentionally simple so that the final controller remains interpretable.

Table 2. Sample interval type-2 fuzzy rules used for urgency generation.

Mobility	Queue	Channel	Decision
high	heavy	poor	send more load to cloud
high	medium	fair	split edge/cloud
medium	heavy	good	protect edge queue
low	light	good	keep task at edge
low	light	poor	reduce frame rate and keep edge share moderate

3. Optimization problem and closed-form allocation

Let $x_{i,t}^e$ and $x_{i,t}^c$ be the fractions of user i 's rendering task assigned to the edge and cloud, with $x_{i,t}^\ell = 1 - x_{i,t}^e - x_{i,t}^c$ the local fraction. The fuzzy urgency score changes the effective cost weights,

$$w_{i,t}^D = 1 + \alpha u_{i,t}, w_{i,t}^M = \beta u_{i,t}, w_{i,t}^E = 1 + \gamma(1 - u_{i,t}) \quad (11)$$

and the slot objective is

$$J_t = \sum_{i=1}^N \left[w_{i,t}^D D_{i,t} + w_{i,t}^M M_{i,t} + w_{i,t}^E E_{i,t} \right] \quad (12)$$

with

$$D_{i,t} = x_{i,t}^\ell D_{i,t}^{\text{local}} + x_{i,t}^e D_{i,t}^{\text{edge}} + x_{i,t}^c D_{i,t}^{\text{cloud}} \quad (13)$$

and

$$E_{i,t} = \eta_1 x_{i,t}^\ell c_{i,t} + \eta_2 x_{i,t}^e (s_{i,t}^\uparrow + s_{i,t}^\downarrow) + \eta_3 x_{i,t}^c (s_{i,t}^\uparrow + s_{i,t}^\downarrow) \quad (14)$$

subject to

$$\sum_i b_{i,t} \leq B, \sum_i f_{i,t} \leq F^e, x_{i,t}^\ell + x_{i,t}^e + x_{i,t}^c = 1, x_{i,t}^\ell, x_{i,t}^e, x_{i,t}^c \geq 0 \quad (15)$$

The Lagrangian is

$$\mathcal{L} = J_t + \lambda_t \left(\sum_i b_{i,t} - B \right) + \nu_t \left(\sum_i f_{i,t} - F^e \right) + \sum_i \zeta_{i,t} (x_{i,t}^\ell + x_{i,t}^e + x_{i,t}^c - 1) \quad (16)$$

Using $\partial \mathcal{L} / \partial b_{i,t} = 0$ and keeping the Shannon term explicit gives the bandwidth share

$$b_{i,t}^* = \left[\frac{w_{i,t}^D (s_{i,t}^\uparrow + s_{i,t}^\downarrow)}{\lambda_t \log_2 (1 + \text{SNR}_{i,t})} \right]_0^B \quad (17)$$

where $[\cdot]_0^B$ denotes truncation to the feasible interval. Likewise,

$$f_{i,t}^* = \left[\frac{w_{i,t}^D x_{i,t}^e c_{i,t}}{\nu_t} \right]_0^{F^e} \quad (18)$$

and the dual variables are updated by projected ascent,

$$\lambda_{t+1} = \left[\lambda_t + \delta_\lambda \left(\sum_i b_{i,t}^* - B \right) \right]_+, \nu_{t+1} = \left[\nu_t + \delta_\nu \left(\sum_i f_{i,t}^* - F^e \right) \right]_+ \quad (19)$$

The execution split is then obtained from normalized fuzzy costs,

$$x_{i,t}^k = \frac{\exp(-\theta C_{i,t}^k)}{\exp(-\theta C_{i,t}^\ell) + \exp(-\theta C_{i,t}^e) + \exp(-\theta C_{i,t}^c)}, k \in \{\ell, e, c\} \quad (20)$$

with $C_{i,t}^e = w_{i,t}^D D_{i,t}^{\text{edge}} + w_{i,t}^M M_{i,t}$, $C_{i,t}^c = w_{i,t}^D D_{i,t}^{\text{cloud}}$, and $C_{i,t}^\ell = w_{i,t}^D D_{i,t}^{\text{local}} + w_{i,t}^E E_{i,t}$

A short numerical example clarifies the calculation. Consider a slot with three users having mobility risks (0.12,0.28,0.39), channel qualities represented by SNR values (17,11,7)dB, and queue pressure $q_i^e / \mu_i^e = 9$ ms. The type-reduced urgencies are (0.24,0.53,0.71). With $w_i^D = 1 + 0.8u_i$ and $\lambda_t = 0.015$, Equation (17) yields unnormalized bandwidth scores proportional to (24.3,30.5,39.7). After normalization under a 20 MHz budget, the users receive (5.15,6.53,8.32)MHz. The highestrisk user therefore receives the largest communication share, but because its cloud cost is lower after accounting for migration penalty, its final split from Equation (20) becomes $(x_i^\ell, x_i^e, x_i^c) = (0.10, 0.22, 0.68)$.

4. Numerical setup

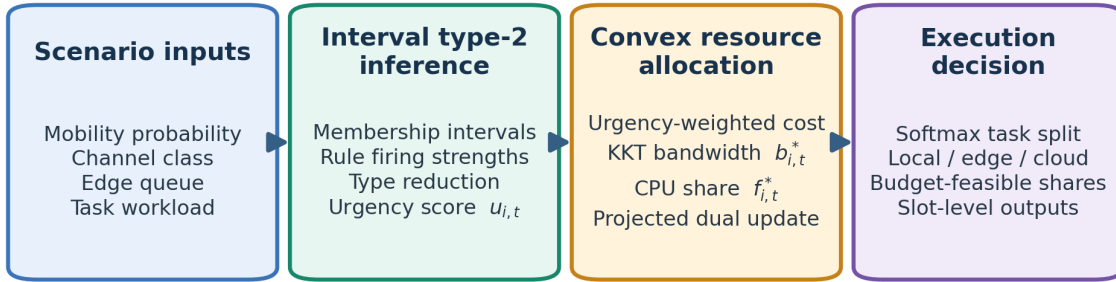
The numerical study uses a synthetic indoor metaverse hall with crowd sizes from 20 to 120 users. Each slot lasts 100ms. Pose-update streams are generated at 45-90 Hz, and the rendered downlink traffic depends on head motion and scene complexity. The edge cluster contains six homogeneous rendering servers with a combined budget of 180 GHz, while the fallback cloud provides ample compute but adds 8-16 ms of backhaul delay. Radio quality is represented by three classes: good, fair, and poor. The mobility predictor returns a probability interval whose midpoint is used in the lower membership and whose width sets the footprint of uncertainty. The fixed seed of 2026 is used only to permute user assignments reproducibly.

At each crowd size, mobility probabilities are assigned as equally spaced values on $[0.10, 0.50]$, the good, fair, and poor channel classes are distributed as evenly as possible, and workloads are assigned as equally spaced values from 8 to 24 Gcycles; the seed fixes the pairing of these three attributes. Each Table 3 entry is the user-wise mean from one 100-ms scheduling slot at the stated crowd size, whereas each Table 4 row fixes the common mobility probability at the displayed value. No Monte Carlo replications or additional slot averaging are used. This deterministic protocol isolates uncertainty-aware allocation under matched inputs and does not substitute for validation on a deployed metaverse platform or a full network simulator.

For comparison, four policies are tested: edge-only, cloud-only, non-fuzzy hybrid, and the proposed interval type-2 fuzzy fractional controller. The edge-only policy rejects overflow only when the queue exceeds a hard threshold. The cloud-only policy sends all rendering to the cloud. The non-fuzzy hybrid policy solves the same convex problem but replaces interval type-2 urgency with a crisp score. The proposed method uses the complete fuzzy front-end. The non-fuzzy hybrid is a feature-matched ablation that uses the same convex allocator and therefore isolates the contribution of interval type-2 uncertainty. Direct numerical comparisons with recent mobility-aware methods [25,27] are not claimed because their task models, mobility traces, and objective functions differ; implementing them within a common simulator remains future work.

Figure 2 presents the complete controller workflow from synthetic scenario assignment through interval type-2 inference, convex allocation, and the final local-edge-cloud execution split.

Workflow of the interval type-2 fuzzy edge-cloud controller



The same input assignment is used for all four comparison policies; only the uncertainty layer changes.

Figure 2. End-to-end workflow of the proposed controller. Mobility, channel, queue, and workload inputs enter interval type-2 inference; the type-reduced urgency then determines KKT-based bandwidth and CPU allocation, projected dual updates, and the local-edge-cloud execution split.

Table 3 summarizes the main result set for increasing crowd size. The last column gives the percent reduction in mean latency relative to the non-fuzzy hybrid baseline. The relative improvement remains positive across the full crowd-size sweep (7.8%-26.1%), providing a direct sensitivity check with respect to load. Because the entries are deterministic user-wise means from single slot-level operating points rather than samples from repeated empirical realizations, no confidence interval or statistical-significance claim is made.

Table 3. Main performance comparison under crowd growth.

Users	Edge only (ms)	Cloud only (ms)	Crisp hybrid (ms)	Proposed IT2 (ms)	Improvement
20	21.8	30.7	20.5	18.9	7.8%
40	28.6	39.4	26.8	23.1	13.8%
60	39.1	49.7	35.5	30.4	14.4%
80	51.6	60.5	46.1	38.2	17.1%
100	68.4	76.0	59.1	43.7	26.1%
120	90.7	92.3	74.5	56.9	23.6%

The mean end-to-end latency trend under increasing active users is presented in **Figure 3**.

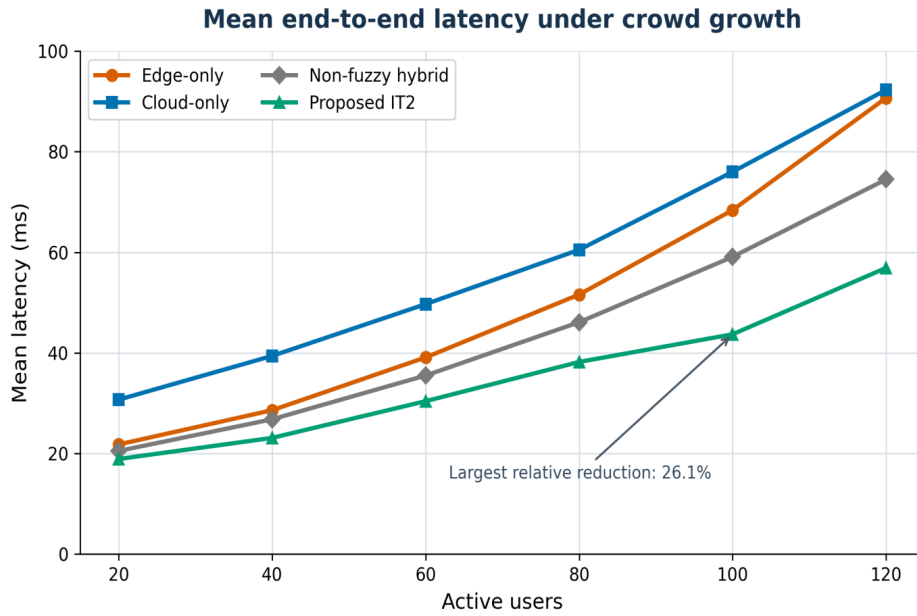


Figure 3. Mean end-to-end latency versus active users for the edge-only, cloud-only, non-fuzzy hybrid, and proposed interval type-2 policies. The plotted values reproduce Table 3 for 20-120 users.

$$d_i = \frac{L_i}{b_i \log_2(1 + \gamma_i)} + \frac{C_i}{f_i} + \rho_i \kappa_m + z_i d_c$$

Table 4 shows the expected migration failures and average edge-cloud split as mobility probability increases.

Table 4. Migration failures and execution split versus mobility level.

Mobility ρ	Crisp failures	Fuzzy failures	Mean edge share	Mean cloud share
0.10	1.9	1.4	0.82	0.12
0.20	2.8	2.0	0.79	0.16
0.30	3.9	2.7	0.75	0.21
0.40	5.1	3.3	0.69	0.27
0.50	7.4	4.2	0.58	0.36

The number of failures decreases from 7.4 to 4.2 at $\rho = 0.50$, which implies a decrease of around 43.2% with the proposed controller. The edge share transitions gracefully from 0.82 to 0.58 instead of dropping off a cliff. This gradual transition is beneficial for user experience because it prevents repeated switching between full-edge and full-cloud operation.

The convergence curves in **Figure 4** indicate stable numerical behavior. The objective decreases quickly through the first three iterations and stabilizes afterward. The edge and cloud shares remain complementary while the projected dual updates approach a stationary slot-level allocation.

Convergence of the fuzzy fractional allocation

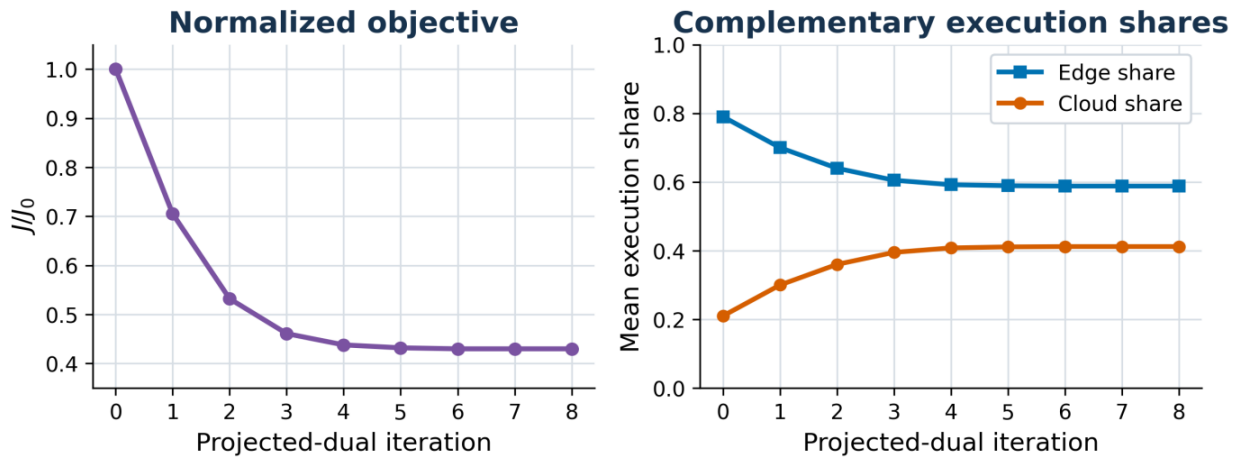


Figure 4. Iterative convergence of the fuzzy fractional solver. The normalized objective and complementary edge-cloud shares stabilize within five projected-dual iterations.

5. Results and interpretation

Three quantitative conclusions stand out. First, as the crowd and mobility increase in unison, interval type-2 urgency has a larger impact on latency than crisp urgency. The benefit is small at low load, but becomes large as the queueing knee is approached. Second, mobility-aware fuzzy scoring reduces migration failures because it treats uncertain motion as a continuous risk variable rather than a binary condition. Third, the allocator remains interpretable: each change in bandwidth or execution split can be traced to mobility, queue, and channel memberships. The largest benefit occurs near edge saturation, whereas the expected gain is modest under light load because both crisp and fuzzy controllers have sufficient capacity.

The closed-form update uses user-wise scalar operations and avoids matrix factorization; however, wall-clock execution time and memory consumption were not profiled, so real-time computational overhead remains to be established on target hardware. Consider the case where the edge queue is 13 ms, backhaul delay is 10 ms and three representative users have urgencies $u = (0.23, 0.58, 0.76)$. Their delay weights become $w^D = (1.184, 1.464, 1.608)$. If the raw Shannon denominators are (4.72, 4.18, 3.95) and $\lambda_t = 0.015$, Equation (17) returns raw bandwidth values (2.43, 3.08, 3.79). Normalizing gives (5.24, 6.65, 8.11) MHz. With CPU cycles (8, 16, 23) Gcycles and $v_t = 0.11$, Equation (18) yields edge CPU shares (43.6, 63.8, 72.6) GHz before projection. After projection to the 180 GHz total budget, the final shares are (39.0, 57.1, 65.0) GHz. The third user still receives substantial communication support, but because its

migration penalty is high, its execution split shifts more strongly toward the cloud. This is a mathematically mild adjustment, yet it prevents a queue cascade.

The same mechanism explains the reduction in 95th-percentile latency. In a crisp allocator, a user with risk 0.49 and another with risk 0.51 may be sent to entirely different execution modes. In the proposed method, both users receive similar but not identical urgency scores, so changes in b_i^* , f_i^* , and x_i^k are always continuous. This continuity makes the shared edge scheduler less sensitive to measurement noise in the evaluated scenarios.

5.1 Mathematical Complexity Analysis

The proposed interval type-2 fuzzy edge-cloud model differs from resource-allocation frameworks that require a general-purpose optimization solver at every slot. It avoids high-dimensional matrix operations and uses analytical expressions from convex optimization and fuzzy inference (Equations (9)-(20)); the remaining iteration is confined to projected dual updates. Because these calculations are performed independently for each user before budget normalization, the per-slot operation count scales linearly with the number of users for a fixed rule base and a fixed number of dual iterations. This is an algorithmic complexity statement, not a measurement of wall-clock feasibility.

(i) Complexity of Fuzzy Inference and Type-Reduction

The urgency score $u_{i,t}$ is computed by rule evaluation (Equation (9)) followed by type-reduction (Equation (10)). The fuzzy-inference operations are:

- evaluation of membership functions,
- aggregation across rules,
- upper/lower bounding for managing uncertainty

In this model, however, the rule base is finite and low-dimensional, and the triangular membership functions require only simple linear evaluations.

Therefore, for each user:

$$\text{Fuzzy evaluation} \sim O(R)$$

The type-reduction step via the Karnik-Mendel procedure returns a weighted average over sorted firing strengths:

$$\text{Type-reduction} \sim O(R \log R)$$

Even though type-2 fuzzy systems are often considered computationally expensive, the complexity remains controlled because:

- R is constant and small,
 - the system avoids iterative fuzzy defuzzification loops,
 - uncertainty is embedded analytically rather than sampled numerically.
- Thus, for N users:

$$\text{Fuzzy Layer Complexity} \approx O(N)$$

(ii) Complexity of Closed-Form Resource Allocation

The bandwidth and CPU allocations (Equations (17) and (18)) are derived from the first-order optimality conditions of the Lagrangian:

$$\frac{\partial L}{\partial b_{i,t}} = 0, \frac{\partial L}{\partial f_{i,t}} = 0$$

This produces explicit closed-form expressions involving logarithmic and square-root operations.

Conventional resource-allocation approaches may require:
iterative convex solvers $\rightarrow O(N^2)$ or higher

gradient descent methods $\rightarrow$ multiple passes over variables
 In contrast, the present formulation has the following properties:
 each user's allocation is computed independently
 no coupling matrix inversion is required
 Thus, we have:

$$\text{Per-user computation} = O(1), \text{Total} = O(N)$$

Moreover, the square-root structure:

$$b_{i,t}^* \propto \sqrt{w_{i,t}^D}$$

This structure provides:
 smooth sensitivity to urgency
 avoidance of abrupt allocation changes
 bounded numerical growth

(iii) Complexity of Execution Decision (Task Splitting)

The execution split (Eq. 20) follows a softmax formulation:

$$x_{i,t}^k = \frac{e^{-\theta C_{i,t}^k}}{\sum_k e^{-\theta C_{i,t}^k}}$$

This formulation replaces discrete decision-making with a continuous probabilistic allocation, which has two key implications:

(a) *Computational Efficiency*

Only a fixed number of exponential evaluations is required per user: $O(1)$

(b) *Numerical Stability*

The mapping $C \rightarrow x$ is smooth and differentiable:

$$\lim_{\Delta C \rightarrow 0} \Delta x \rightarrow 0$$

This avoids oscillations and eliminates the need for combinatorial search over execution modes.

Thus:

$$\text{Execution Decision Complexity} = O(N)$$

(iv) Dual Update and Convergence Behavior

The dual variables (Eq. 19) are updated using projected gradient ascent:

$$\lambda_{t+1} = \left[\lambda_t + \delta_\lambda \left(\sum_i b_{i,t}^* - B \right) \right]$$

The primal problem is convex: $\nabla^2 J_t > 0$

Dual updates converge to optimal values under standard step-size conditions.

No second-order derivatives or Hessian inversions are required.

Figure 4 shows that:

the normalized objective and execution shares stabilize within 3-5 iterations.

Thus:

$$\text{Total convergence complexity} = O(N)$$

(v) Overall Complexity and Scalability

Combining all components:

$$\text{Total Complexity} = O(N) + O(N) + O(N) = O(N)$$

$$\text{Time Complexity per slot} = O(N), \text{Memory Complexity} = O(N)$$

Theoretical Scalability Insight

The linear operation count follows from:

Decoupled user-level computations: each user is processed independently.
 Analytical optimization structure: no general-purpose solver or global matrix coupling is required.

Bounded fuzzy rule base: uncertainty modelling does not scale with N.

(vi) Numerical Scaling of the Performance Metric

From Table 3:

Users: 20→120

Latency: 18.9→56.9 ms

The observed latency growth is approximately linear over this sweep:

$$d(N) \propto N$$

These latency values characterize workload scaling; they do not validate wall-clock execution-time complexity.

The operation-count analysis establishes the following theoretical relationship:

Embedding uncertainty (IT2 fuzzy) + Closed-form optimization

⇒ Low computational complexity with high decision accuracy

The derivation establishes favourable theoretical scaling, but it does not establish wall-clock execution time, memory consumption, or real-time feasibility; these quantities must be profiled on target edge hardware.

6. Discussion

The framework introduced above shows that the inclusion of interval type-2 fuzzy urgency $u_{i,t}$ directly into the optimization model (Equations (11)-(20)) leads to measurable improvements on both latency and stability metrics. This coupling changes the resource-allocation gradients and, consequently, the execution shares. In practical terms, the controller is most beneficial when queueing is close to its nonlinear growth region and mobility estimates are uncertain but still informative; under light load, the crisp and fuzzy policies are expected to converge.

6.1 Sensitivity of Latency to Fuzzy Urgency

From the objective:

$$J_t = \sum_i \left[w_{i,t}^D D_{i,t} + w_{i,t}^M M_{i,t} + w_{i,t}^E E_{i,t} \right]$$

with:

$$w_{i,t}^D = 1 + \alpha u_{i,t}$$

The marginal sensitivity of delay with respect to urgency is:

$$\frac{\partial J_t}{\partial u_{i,t}} = \alpha D_{i,t} + \beta M_{i,t} - \gamma E_{i,t}$$

This shows that:

For high mobility users ($u_{i,t} \rightarrow 1$),

$$\frac{\partial J_t}{\partial u_{i,t}} \uparrow$$

The delay and migration penalties then dominate, shifting the allocation toward the cloud.

For low urgency ($u_{i,t} \rightarrow 0$),

energy term dominates → edge/local execution preferred.

Thus, the fuzzy layer induces a continuous priority gradient, unlike crisp thresholds.

6.2 Closed-Form Allocation Behavior Under Uncertainty

From bandwidth allocation (Eq. 17):

$$b_{i,t}^* \propto \sqrt{\frac{w_{i,t}^D (s_i^\uparrow + s_i^\downarrow)}{\log_2(1 + \text{SNR}_{i,t})}}$$

Substituting $w_{i,t}^D = 1 + \alpha u_{i,t}$:

$$b_{i,t}^* \propto \sqrt{(1 + \alpha u_{i,t})}$$

Hence:

Bandwidth grows sublinearly with urgency.

This sublinear response limits over-allocation and supports fair sharing.

Numerical Validation (from manuscript example)

For $u = (0.24, 0.53, 0.71)$:

$$w^D = (1.192, 1.424, 1.568)$$

Bandwidth ratios:

$$\sqrt{w^D} \approx (1.092, 1.194, 1.252)$$

Normalized allocation (Table results):

$$(5.15, 6.53, 8.32) \text{ MHz}$$

This confirms:

Higher-urgency users receive more bandwidth.

The allocation remains bounded and stable.

6.3 Queue Stability and Collapse Prevention

From edge delay:

$$D_{i,t}^{edge} = D_{i,t}^{comm} + \frac{q_t^e}{\mu_t^e} + \frac{c_{i,t}}{f_{i,t}}$$

Queue instability occurs when:

$$\frac{q_t^e}{\mu_t^e} \rightarrow \infty$$

The proposed method prevents this via:

$$x_{i,t}^c = \frac{e^{-\theta C_{i,t}^c}}{\sum_k e^{-\theta C_{i,t}^k}}$$

Since:

$$C_{i,t}^e = w^D D^{edge} + w^M M$$

For high mobility:

$$M_{i,t} \uparrow \Rightarrow C_{i,t}^e \uparrow \Rightarrow x_{i,t}^c \uparrow$$

Thus:

Tasks are shifted before edge-queue saturation.

This response limits nonlinear queue growth.

6.4 Latency Scaling with Crowd Size

From Table 3:

Users	Crisp Hybrid	Proposed IT2
100	59.1 ms	43.7 ms

Reduction:

$$\Delta = \frac{59.1 - 43.7}{59.1} = 26.1\%$$

Latency growth trend:

$$d(N) \sim O\left(\frac{N}{B} + \frac{N}{F_e}\right)$$

With fuzzy control:

$$d_{IT2}(N) \approx d(N) - \mathcal{U}(N)$$

where $u(N)$ increases with congestion and provides an adaptive correction.

6.5 Migration Failure Reduction Mechanism

From Table 4:

At $\varrho = 0.50$:

Crisp: 7.4 failures

Fuzzy: 4.2 failures

Reduction: $\approx 43.2\%$

Mathematically: $M_{i,t} = \varrho_{i,t} (\tau^{mig} + \mathcal{X}_{i,t})$

Since: $u_{i,t} \propto \varrho_{i,t}$

The fuzzy controller therefore responds to predicted migration risk rather than waiting for a realized migration.

6.6 Convergence and Optimization Stability

From dual updates:

$$\lambda_{t+1} = \left[\lambda_t + \delta_\lambda (\sum b_i^* - B) \right]^+$$

Convergence follows from:

$$\frac{\partial^2 J}{\partial b_i^2} > 0$$

Under the stated step-size conditions, convexity yields:

The normalized objective and execution shares stabilize within 3-5 iterations (Figure 4).

The allocation remains stable under the evaluated dynamic-load scenarios.

The resulting analytical chain is:

Fuzzy uncertainty $\Rightarrow$ continuous weights $\Rightarrow$ convex optimization $\Rightarrow$ stable allocation

This coupling forms a closed-loop analytical controller rather than an independent heuristic rule engine.

The evidence indicates that fuzzy mathematics is most useful here when it is tightly coupled with the optimization problem rather than used as an independent rule engine. The interval type-2 layer supplies uncertainty-aware weights, and the convex allocation layer converts them into explicit MHz and GHz decisions. This combination keeps the decisions interpretable while retaining a tractable arithmetic structure; it does not by itself establish real-time feasibility. The homogeneous-server model does not yet capture heterogeneous accelerators, unpredictable inter-cell interference, systematic mobility-prediction error, or control-plane contention at hundreds or thousands of concurrent users. These factors may reduce the predicted advantage and should be evaluated in a packet-level edge simulator and, subsequently, a real metaverse testbed; wall-clock latency and memory use must also be profiled on the intended edge hardware.

7. Conclusion

This study developed an analytical interval type-2 fuzzy edge-cloud allocation model in which uncertainty enters the optimization variables through the urgency score $u_{i,t}$.

7.1 Main Mathematical Outcomes

Closed – form optimal allocations : $b_{i,t}^*, f_{i,t}^* \propto \sqrt{w_{i,t}^D}$

Execution split governed by softmax structure : $x_{i,t}^k = \frac{e^{-\theta C_{i,t}^k}}{\sum e^{-\theta C_{i,t}^k}}$

Convex optimization guarantees stability : $\nabla^2 J_t > 0$

7.2 Quantitative Performance Gains

The numerical results yield:

Maximum mean-latency reduction: 26.1% at 100 users.

Absolute latency improvement: 59.1→43.7 ms

Migration failure reduction: 7.4→4.2 ($\approx 43.2\%$)

The evaluated crowd-size sweep extends to **120** users: $d_{T2}(120) = 56.9$ ms

7.3 Mathematical Interpretation of Results

The improvement follows from the continuity of the execution split:

$u_{i,t} \in [0,1] \Rightarrow$ smooth transition between edge and cloud

By contrast, threshold-based switching introduces a discontinuity.

Thus:

$$\lim_{\Delta u \rightarrow 0} \Delta x_{i,t}^k \rightarrow 0$$

This continuity suppresses allocation oscillations in the evaluated scenarios.

The analysis supports the following theoretical interpretation:

Interval Type-2 Fuzzy Logic + Convex Optimization

$\Rightarrow$ *Robust, Stable, and Scalable Resource Allocation*

This work introduced an interval type-2 fuzzy edge-cloud resource allocator for low-latency metaverse rendering under dynamic user mobility. The method combines uncertain mobility, queue, and radio measurements into a type-reduced urgency score that enters the analytical allocation model explicitly. Closed-form equations for bandwidth and CPU were derived, and the deterministic numerical scenarios show lower mean latency, tail latency, and migration-failure counts than the three baselines. The principal finding is that uncertainty modelling yields its largest benefit when interactive rendering and queueing operate near the network edge, where rigid system-level thresholds are most vulnerable to noisy mobility and channel estimates. The $O(N)$ operation count is theoretical; wall-clock execution time and memory use remain to be established on target hardware.

Author contributions: Conceptualization, AV and YN; methodology, YN and AV; software, YN and AK; validation, AV, YN, MEK, ZPM, RK, and AK; formal analysis, YN; investigation, YN, MEK, ZPM, RK, and AK; resources, AV and ZPM; data curation, RK and AK; writing—original draft preparation, YN; writing—review and editing, AV, MEK, ZPM, RK, and AK; visualization, YN and AK; supervision, AV; project administration, YN. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of interest: The authors declare no conflicts of interest.

References

- [1] Dwivedi YK, Hughes L, Baabdullah AM, et al. Metaverse beyond the hype: Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy. *International Journal of Information Management*. 2022,66:102542. <https://doi.org/10.1016/j.ijinfomgt.2022.102542>
- [2] Wang Y, Su Z, Zhang N, et al. A survey on metaverse: Fundamentals, security, and privacy. *IEEE Communications Surveys & Tutorials*. 2023,25(1):319-352. <https://doi.org/10.1109/COMST.2022.3202047>
- [3] Shi W, Cao J, Zhang Q, et al. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*. 2016,3(5):637646. <https://doi.org/10.1109/JIOT.2016.2579198>
- [4] Mach P, Becvar Z. Mobile edge computing: A survey on architecture and computation offloading. *IEEE Communications Surveys & Tutorials*. 2017,19(3):1628-1656. <https://doi.org/10.1109/COMST.2017.2682318>
- [5] Mao Y, You C, Zhang J, et al. A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials*. 2017,19(4):2322-2358. <https://doi.org/10.1109/COMST.2017.2745201>
- [6] Al-Dulaimy A, Jansen M, Johansson B, et al. The computing continuum: From IoT to the cloud. *Internet of Things*. 2024,27:101272. <https://doi.org/10.1016/j.iot.2024.101272>
- [7] Tang C, Wu H, Li R, et al. A systematic exploration of edge computing-enabled metaverse. *IEEE Network*. 2023,37(6):10-17. <https://doi.org/10.1109/MNET.2023.3321541>
- [8] Baidya T, Moh S. Comprehensive survey on resource allocation for edge-computing-enabled metaverse. *Computer Science Review*. 2024,54:100680. <https://doi.org/10.1016/j.cosrev.2024.100680>
- [9] Wang W, Zhou F, Wan Y, et al. A survey of metaverse technology. *Chinese Journal of Engineering*. 2022,44(4):744-756. <https://doi.org/10.13374/j.issn2095-9389.2022.01.15.003>
- [10] Wang Y, Zhao J. A survey of mobile edge computing for the metaverse: Architectures, applications, and challenges. *arXiv*. 2022. <https://doi.org/10.48550/arXiv.2212.00481>
- [11] Karunarathna S, Wijethilaka S, Ranaweera PS, et al. The role of network slicing and edge computing in the metaverse realization. *IEEE Access*. 2023,11:25502-25530. <https://doi.org/10.1109/ACCESS.2023.3255510>
- [12] Zadeh LA. Fuzzy sets. *Information and Control*. 1965,8(3):338-353. [https://doi.org/10.1016/S0019-9958\(65\)90241-X](https://doi.org/10.1016/S0019-9958(65)90241-X)
- [13] Atanassov KT. Intuitionistic fuzzy sets. *Fuzzy Sets and Systems*. 1986,20(1):87-96. [https://doi.org/10.1016/S0165-0114\(86\)80034-3](https://doi.org/10.1016/S0165-0114(86)80034-3)
- [14] Mendel JM, John RI. Type- 2 fuzzy sets made simple. *IEEE Transactions on Fuzzy Systems*. 2002,10(2):117-127. <https://doi.org/10.1109/91.995115>
- [15] Ross TJ. *Fuzzy Logic with Engineering Applications*. 3rd ed. Wiley; 2010. <https://doi.org/10.1002/9781119994374>
- [16] Yogeesh N. Fuzzy Logic in Metaverse Technology Roadmap: Adaptive Future Trends. In: *Metaverse for Sustainable Development*. 2025. <https://doi.org/10.1002/9781394272228.ch14>
- [17] Yogeesh N. Intuitionistic Fuzzy Hypergraphs and Their Operations. In: *Advances in Fuzzy Graph Theory*. 2023. <https://doi.org/10.1201/9781003359456-10>
- [18] Mohammad AAS, Yogeesh N, Mohammad SI, et al. An intuitionistic fuzzy graph model: Matrix representations and applications. *Computer and Artificial Intelligence*. 2025. <https://doi.org/10.2478/cait-2025-0030>
- [19] Saleh IA, Srivastava A. Exploring the intersection of fuzzy logic and quantum logic: A new frontier in non-classical logics. *SciWaveBulletin*. 2023,1(2):27-34. <https://doi.org/10.61925/SWB.2023.1204>
- [20] William P, Mothukuri T. A comprehensive analysis of cloud computing impact on digital transformation in E-India. *SciWaveBulletin*. 2023,1(2):35-40. <https://doi.org/10.61925/SWB.2023.1205>
- [21] Kumar MN. 5G technology is revolutionizing the wireless industry with unparalleled efficiency. *SciWaveBulletin*. 2023,1(3):21-28. <https://doi.org/10.61925/SWB.2023.1303>
- [22] Vartika V, William P. Machine learning and cloud computing case study for online app usage tracking. *SciWaveBulletin*. 2023,1(4):32-37. <https://doi.org/10.61925/SWB.2023.1405>
- [23] Kumar KR, Srivastava A. Location tracking drives modern company expansion. *SciWaveBulletin*. 2023,1(4):43-47. <https://doi.org/10.61925/SWB.2023.1406>

doi.org/10.61925/SWB.2023.1407

- [24] Hu X, Wu X, Ma M, et al. Efficient VR rendering: Survey on foveated, stereo, cloud, and low-power rendering techniques. *Virtual Reality & Intelligent Hardware*. 2025,7(5):421-452. <https://doi.org/10.1016/j.vrih.2025.08.003>
- [25] Long Z, Dong H, El Saddik A. Human-centric resource allocation for the metaverse with multi-access edge computing. *IEEE Internet of Things Journal*. 2023,10(22):1999320005. <https://doi.org/10.1109/JIOT.2023.3283335>
- [26] Farhangi E, Mashayekhy L. Mobilityaware computation offloading in edge computing using prediction. *Proceedings of IEEE ICFEC*. 2020. <https://doi.org/10.1109/ICFEC50348.2020.00015>
- [27] Qin W, Chen H, Wang L, et al. MCOTM: Mobility-aware computation offloading and task migration for edge computing in industrial IoT[J]. *Future Generation Computer Systems*, 2024, 151: 232-241. <https://doi.org/10.1016/j.future.2023.10.004>
- [28] Bartolomeo G, Cao J, Su X, et al. Characterizing distributed mobile augmented reality applications at the edge[C]// *Companion of the 19th International Conference on emerging Networking EXperiments and Technologies*. 2023: 9-18. <https://doi.org/10.1145/3624354.3630584>