# Lexi: Priority-First Service Orchestration

Xuan-Bach Le[1] ✉*, Phat T. Tran-Truong[1], Duy Nguyen[2], and Ngoc Khanh Tran Nguyen[3]

[1] Ho Chi Minh City University of Technology (HCMUT), VNU-HCM, Vietnam
{lexuanbach,phatttt}@hcmut.edu.vn
[2] University of Economics Ho Chi Minh City, Vietnam
duynguyen.726202321440@st.ueh.edu.vn
[3] University of Science, VNU-HCM, Vietnam
22120378@student.hcmus.edu.vn

**Abstract.** A service spanning cloud, edge, and serverless nodes must balance latency, data exposure, carbon, and cost, which operators may rank strictly, e.g., meeting a latency SLO first. Weighted sums approximate such a ranking only with deployment-specific weights, and their decisions change when an objective's numerical scale is misestimated. We present Lexi, a continuum orchestrator applying a thresholded lexicographic rule to predictions of an off-policy counterfactual outcome model, which assumes objectives decompose across stages. We instantiate the rule as SLO $\succ$ privacy $\succ$ carbon $\succ$ cost and use a simple topological privacy proxy only to separate placements already tied on the SLO. In a 15-seed simulation, Lexi needs no weights and reaches **1.46**% SLO violations with PII exposure **0.05**, comparable to hand-calibrated and SLO-constrained weighted sums. An untuned sum reaches 23% SLO violations and 1.94 PII exposure, and only one of 84 tested weightings meets the target. Under-scaling privacy 10× raises the tuned and constrained baselines' exposure to **1.85** and **1.84** of two stages, respectively. Lexi is unchanged. The evidence covers one simulated application under the additive no-interference assumption.

**Keywords:** Service orchestration · Cloud-edge-serverless continuum · Lexicographic optimisation · Scale invariance.

## 1 Introduction

Modern services run across a *continuum* [3] of regional cloud, on-premises edge, and serverless resources, where placement decides each stage's latency, data exposure, carbon, and cost. Operators often rank these objectives, for example by requiring the latency service-level objective (SLO) to be met before they consider privacy, carbon, or cost. A weighted sum encodes such a ranking as exchange

* X.-B. Le and P. T. Tran-Truong contributed equally to this work.
✉ Corresponding author: Xuan-Bach Le, lexuanbach@hcmut.edu.vn.
Artifact and extended version: https://github.com/lexuanbach/lexi.

rates between objectives. Widely separated weights can approximate it, but suitable values occupy a small, deployment-specific region (one of the 84 weightings tested here) and depend on units that change with hardware, carbon intensity, and budgets. Lexicographic selection needs no exchange rates and is invariant to strictly increasing per-objective transformations (Sect. 4).

We design Lexi to rank candidate placements on the highest-priority objective and break ties with lower priorities. This *thresholded lexicographic* rule (Fig. 1) reads an off-policy *counterfactual outcome model* that estimates untaken placements from executed ones. The evaluated controller makes no deliberate exploratory migration, though estimates beyond the observed assignments depend on model adequacy and feature coverage [14,4]. We instantiate it on SLO $\succ$ privacy $\succ$ carbon $\succ$ cost, where privacy is *the number of PII-tagged stages on a publicly reachable node*. This deliberately coarse *topological* proxy is no compliance certificate and misses the misconfiguration failures dominating real exposure [6]. Ranked second, it never overrides the SLO and only separates SLO-tied candidates (Sect. 2), and another second-priority objective could replace it with the rule unchanged.

We simulate one application (Sect. 5) and make no physical-testbed claim. The extended version adds proofs, the input-generation derivation, a real-trace rerun and RQ5–RQ7 (slack, stability, safety margins, coverage probes, scaling).

## 2 The Lexi Model and Controller

**Definition 1 (Continuum, service DAG, placement, objectives).** *A* continuum *is a set of nodes $\mathcal{N}$, each with a tier, a region, a* privacy tier *(edge private, others public) and the parameters of Sect. 5.1. A* service DAG *has stages $1, \ldots, K$, each with a demand and a PII tag, and a* placement $\pi$ *maps stages to nodes, giving for request $r$ the vector $\mathbf{g}^{\mathrm{raw}}(\pi, r) \in \mathbb{R}^4$ of critical-path* latency*,* privacy *(PII-tagged stages on a public node),* carbon *and* cost*, all* lower is better*.*

*Normalisation.* Objective 0 is the SLO *hinge* $\mathbf{g}_0 = \min(1, \max(0, \text{latency} - \text{SLO})/\rho)$, $\rho = 60$ ms here. It treats latency as a threshold, which ties every SLO-feasible placement at $\mathbf{g}_0 = 0$ and every latency at or above 320 ms at $\mathbf{g}_0 = 1$. The other three are divided by an operator-declared magnitude and clamped to $[0, 1]$. While no candidate saturates the (only non-decreasing) clamp, changing those divisors is a strictly increasing reparameterisation, which leaves strict Lexi selection unchanged (Prop. 1), unlike a weighted sum.

**Definition 2 (Priority order and selection rule).** *The operator specifies a strict priority order on the normalised coordinates, here SLO $\succ$ privacy $\succ$ carbon $\succ$ cost. The* thresholded-lexicographic optimum *with slack $\boldsymbol{\eta}$ keeps, for $k = 0, 1, 2$ in turn, the candidates within $\eta_k$ of the current best on objective $k$, then breaks the final tie on cost. With $\boldsymbol{\eta} = \mathbf{0}$ it is the strict lexicographic optimum, consuming only the* order *of each objective's values. With positive slack, additive distances and their units matter, and a lower priority may choose within the*

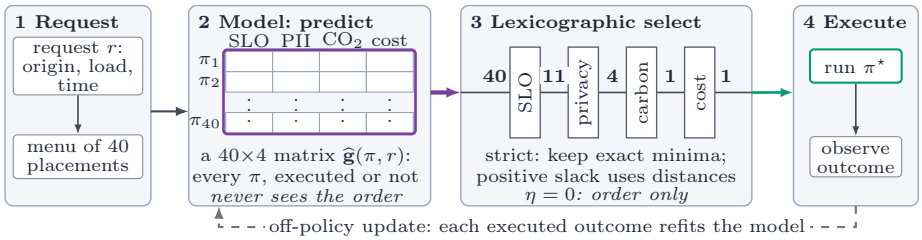


**Fig. 1.** LEXI's decision loop: the model predicts every objective for every candidate (purple: the only interface). The evaluated strict rule ($\boldsymbol{\eta} = \mathbf{0}$) reads order only (counts: Table 1).

*tolerated higher-priority band. Under the strict rule, objective $k$ is consulted only on the set objectives $0, \ldots, k-1$ failed to separate, and a lower priority cannot cause a higher-priority regression.*

The rule is a filter. Each level keeps the (near-)best on one column and passes them down. Per request LEXI predicts $\widehat{\mathbf{g}}(\pi, r_t)$ for every candidate, applies Def. 2, executes and updates the model in $O(K)$. We evaluate the strict rule, $\boldsymbol{\eta} = \mathbf{0}$, since slack on a threshold objective becomes SLO violations directly. Carbon slack, SLO safety margin and coverage probe are disabled (the extended version).

*Privacy coordinate and scope.* $\mathbf{g}_1$ is a *topological* count, read only from the placement and a static sensitivity tag. Two of six stages carry the tag, putting "PII exposure" in $[0, 2]$. Ranked second, $\mathbf{g}_1$ is consulted only among SLO-tied placements (Sect. 3). It is no compliance verdict and ignores residency, encryption, transit security and the *misconfiguration* failures that dominate real exposure [6]. Hard prohibitions belong in the candidate generator or a residency pre-filter. $\mathbf{g}_1$ is a *preference* among otherwise-acceptable placements, *integer-valued* (hence tie-prone) and *unit-ambiguous*, and results resting only on these properties hold for any such objective.

*Outcome-model interface.* Under Assumption 1 LEXI keeps one online per-node estimator per objective, with error $\varepsilon_t$ over the candidate menu. Model and rule share only a $|\mathcal{C}| \times 4$ matrix of normalised predictions. The model never learns the priority order, and the strict selector sees only each column's ordering, without uncertainty or units. The error *can* still let a truly SLO-violating placement survive level 0, the band-crossing error $\varepsilon_t$ measures in Thm. 1.

## 3 A Worked Decision: From Signals to a Placement

The *relationship between the prediction model and the selection rule* is clearest on a real decision. Table 1 traces step $t = 904$ of seed 0 verbatim (node ids as in Table 2), and its "true" latency is for exposition only. From three exogenous signals (origin region 2, load $\ell = 0.765$, time $t = 904$), the model returns

**Table 1.** Step $t$=904 (seed 0): six of the eleven level-0 survivors, including all four level-1 survivors. **Bold**: the choice. †: admitted at level 0 by prediction error.

| | | Latency (ms) | | Normalised $\widehat{\mathbf{g}}$ (what the selector reads) | | | | |
|---|---|---|---|---|---|---|---|---|
| # | Placement | pred. | true | $\mathbf{g}_0$ | $\mathbf{g}_1$ | $\mathbf{g}_2$ | $\mathbf{g}_3$ | Elim. |
| 2 | E2/E2/E2/E2/E2/E2 | 252.5 | 251.9 | 0 | 0.0 | 0.673 | 1.000 | L2 |
| 6 | S2/S2/S2/S2/S2/S2 | 228.9 | 228.8 | 0 | 1.0 | 0.166 | 0.216 | L1 |
| 19 | C2/E2/C2/C2/C2/E2 | 230.3 | 245.8 | 0 | 0.0 | 0.476 | 0.848 | L2 |
| 20 | S1/E2/S1/S1/S1/E2 | 259.1 | 259.0 | 0 | 0.0 | 0.387 | 0.608 | L2 |
| **21** | S2/E2/S2/S2/S2/E2 | **235.7** | **235.5** | **0** | **0.0** | **0.314** | **0.592** | – |
| 24† | C0/C2/S2/S1/C2/C0 | 256.2 | 268.2 | 0 | 1.0 | 0.371 | 0.496 | L1 |

$\widehat{\mathbf{g}}(\pi, r_{904})$ for *all 40* candidates, including the 39 never executed, via per-node models fitted on steps 1–903. This $40 \times 4$ matrix is *all* the selector sees.

**Level 0, the SLO hinge: 40 → 11.** Every feasible placement scores exactly 0, which leaves 11 of 40 candidates tied at the predicted minimum $\mathbf{g}_0 = 0$ for the next priority to break. Only 9 of the 11 truly meet the 260 ms SLO, as under-prediction admitted #24 and one other (not shown). This is the only way estimation error reaches the decision ($\varepsilon_t$ in Thm. 1), and both fall at level 1.

**Level 1, privacy: 11 → 4.** Of the eleven, the minimum predicted exposure is $\mathbf{g}_1 = 0$, and the four with both PII stages on a private edge node survive, $\{2, 19, 20, 21\}$: **privacy is consulted only on the set the SLO has already tied, and can only ever choose among SLO-equivalent placements**.

**Levels 2–3.** #21 wins the near-continuous carbon level outright, leaving the cost tie-break unused. A band $\eta_2$=0.08 would keep #20 as a near-tie ($0.387 \leq 0.314{+}0.08$) and defer to cost (again #21). On the *same* matrix the untuned sum picks #6, exposing **both** PII stages, because #6 wins on latency, carbon and cost and the sum prices two exposed stages below that saving. That exchange rate *is* the weighting. The tuned sum (0.030 vs. 0.111 for #6) and Constr-WS pick #21, yet a 10× privacy under-scaling switches the tuned sum to #6 (0.021 vs. 0.030). No such rescaling changes the lexicographic choice (Prop. 1).

## 4 Theoretical Analysis

Full proofs are in the extended version. Both results hold for any ordered bundle of bounded objectives, whatever the coordinates mean. A decision costs $O(|\mathcal{C}|K)$.

**Proposition 1 (Scale-invariance).** *Let $\phi = (\phi_0, \ldots, \phi_3)$ be any strictly increasing per-objective reparameterisations, and replace each $\mathbf{g}_k$ by $\phi_k \circ \mathbf{g}_k$ in the estimates the selector sees. Strict lexicographic selection returns the* same *placement. A* fixed-weight *scalariser is not invariant in general: there exist candidate sets and positive coordinate rescalings that change its selected optimum. A pair crosses only when the scaled-coordinate difference and weighted remainder have opposite signs. Another candidate may still preserve the global argmin.*

Such invariance is classical [8]. Pricing makes any method scale-*dependent*, including a constrained formulation beyond the coordinate it constrains (Sect. 6.2).

**Assumption 1 (Additive no-interference / stable unit treatment value assumption (SUTVA)).** *Each objective decomposes as a sum (latency: critical path) of per-(stage, node) terms that depend on the stage's assignment and the exogenous request state and not on which* other *stages are co-placed. Executed placements update their per-node terms. Prediction elsewhere also requires model adequacy and node-feature coverage.*

Co-location contention breaks Assumption 1 for every additive-model method, and Sect. 6.3 finds a graded degradation largely preserving the ordering. Heavy shared-node interference is untested.

**Theorem 1 (Top-priority tracking, under assumptions).** *Let* $\pi_t^\star$ *be the priority-optimal placement in menu* $\mathcal{C}_t$*, and let* $\varepsilon_t = \max_{\pi\in\mathcal{C}_t,k} |\widehat{\mathbf{g}}_{t,k}(\pi) - \mathbf{g}_{t,k}(\pi)|$*. Then* $\mathbf{g}_0(\hat{\pi}_t) \le \mathbf{g}_0(\pi_t^\star) + \eta_0 + 2\varepsilon_t$*. If also* (A1) *the per-node models are realizable up to a floor* $\varepsilon_{\text{mis}}$*,* (A2) *features and parameters are bounded, noise sub-Gaussian, and* (A3) *every per-node Gram matrix is persistently excited (*$\lambda_{\min} \ge ct$*), then with probability* $1-\delta$ *online ridge estimation [1] makes the cumulative top-priority gap* $\eta_0 T + O(\sqrt{T\log T}) + 2\varepsilon_{\text{mis}} T$*. Because a no-exploration policy does not guarantee (A3), that rate is* conditional*. The bound is on normalised hinge loss and covers neither raw latency nor the SLO-violation probability.*

## 5 Experimental Setup

We use a purpose-built discrete-event simulator (released with the artifact) with $M$=7 nodes over 3 regions, a $K$=6-stage DAG with a 260 ms SLO, and a geo-diurnal workload varying load and origin. Every method ranks the same fixed 40-candidate menu. The primary comparison uses synthetic load and carbon signals and averages **15 seed-level post-warm-up summaries** over steps 800–1599. 95% bootstrap intervals and paired tests resample those summaries. Sweeps use 6–8 seeds. A separate rerun, without re-tuning, substitutes UK Carbon Intensity API forecasts [16] and the Azure Functions 2019 trace [17] for those two signals.

### 5.1 How the evaluation inputs are generated, and why

Energy, cost, round-trip time (RTT), queueing, and PII tags are authorial choices: DeathStarBench [10] and Alibaba provide DAG shapes but no continuum placements. Table 2 deliberately creates tension among private but weak edge nodes, powerful remote cloud nodes, and cheap public serverless nodes [3]. Values are simulator units, with no provider measurements behind them. The synthetic intensity ordering prevents privacy and carbon from aligning trivially.

*The DAG, the PII tags, and the menu.* The evaluated DAG is `frontend`(8) → `auth`(16) → {`search`(30), `recommend`(25)} → `order`(20) → `pay`(18), with per-stage compute demand in parentheses. Stage $s$ on node $n$ has latency $\text{RTT}(\text{origin}, n) + b_n\,\text{demand}_s + q_n\ell$, with $\text{RTT}_{\text{in}}$ within the node's region and $\text{RTT}_{\text{out}}$ across, $b$ the compute per unit demand (as is energy) and $q$ the queueing

**Table 2.** Nodes of config A, verbatim from the artifact. Edge nodes are private.

| | | | Latency terms | | | | Cost and carbon | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Id | Tier | Reg. | $b$ ms/unit | $\mathrm{RTT}_{\mathrm{in}}$ ms | $\mathrm{RTT}_{\mathrm{out}}$ ms | $q$ ms/load | Cost sim./inv. | Energy Wh, $d$=20 | $\bar{I}$ g/kWh | Priv. |
| `E0` | edge | 0 | 2.20 | 4 | 70 | 9.0 | 1.15 | 0.62 | 520 | ✓ |
| `E1` | | 1 | 2.00 | 5 | 72 | 8.5 | 1.10 | 0.58 | 540 | ✓ |
| `E2` | | 2 | 2.10 | 5 | 71 | 8.8 | 1.12 | 0.60 | 430 | ✓ |
| `C0` | cloud | 0 | 0.50 | 35 | 46 | 2.0 | 0.55 | 0.95 | 360 | |
| `C2` | | 2 | 0.55 | 38 | 50 | 2.2 | 0.50 | 0.90 | 190 | |
| `S1` | serverless | 1 | 1.10 | 18 | 30 | 4.5 | 0.20 | 0.36 | 260 | |
| `S2` | | 2 | 1.20 | 20 | 32 | 4.8 | 0.18 | 0.34 | 175 | |

sensitivity per unit load. Its carbon (g) is $\sum_s \mathrm{energy}_n(\mathrm{demand}_s/20) I_n(t)/1000$, where $I_n$ is a synthetic node-specific effective intensity oscillating around $\bar{I}_n$ with an amplitude of 60–120 $\mathrm{gCO_2/kWh}$ on a 240-step diurnal cycle, and queueing and carbon noise are seeded at execution. The DAG's shape (serial head, compute-heavy parallel fan-out, join, serial tail) follows DeathStarBench [10] and the Alibaba traces [13]. We set demands so that the parallel branches dominate the critical path. The two PII-tagged stages, `auth` and `pay`, also contribute latency while determining privacy. Thus the objectives are coupled, and $\mathbf{g}_1$ is integer-valued and tie-prone. The queueing sensitivities $q$ have the least external grounding. We set them proportional to inverse compute capacity (the weak edge queues hardest) and large enough that load re-orders the tiers over a day, and the SLO sweep of Sect. 6.3 serves as the sensitivity analysis for this choice. The deduplicated menu holds 22 *structured* placements and 18 seeded random fills. The structured ones are 7 homogeneous placements, 3 latency-greedy ones (one per origin region), the cost-minimal one, and 12 privacy-aware ones enumerating every (private edge, public backend) pair. These last guarantee that the privacy objective always has something to choose, which pins any privacy failure on the *rule*. The menu is not neutral, however, and Sect. 6.3 shows that it is the largest determinant of the absolute numbers.

*Baselines.* All methods share the menu, model architecture, and initial priors. Each learned method maintains its own model and therefore acquires a policy-dependent training trajectory. Lexi has no weights, and the weight-based baselines get favourable calibrated settings: *WS-default* (untuned $0.30, 0.12, 0.30, 0.28$), *WS-tuned* (calibrated, SLO-heavy $0.85, 0.10, 0.00, 0.05$, better than the best of the 84-point sweep), three *scale-robust* scalarisers reusing those weights [15], *Lex-LP*, and *Constr-WS*. Constr-WS, a one-step selector with no learned policy, admits only predicted-SLO-feasible candidates and then minimises an equal-weight sum of the rest. If none is predicted feasible, it chooses the candidate with minimum predicted SLO hinge. Lex-LP implements the same discrete lexicographic rule as Lexi on the same estimates. We also

**Table 3.** RQ2 operator-facing outcomes (15 seeds, means, lower is better). Inv% is the priority-inversion rate. The text summarises the three scale-robust scalarisers.

| Method | SLO%↓ | PII↓ | Carbon↓ | Cost↓ | Churn↓ | Inv%↓ |
|---|---|---|---|---|---|---|
| **LEXI (no weights)** | **1.46** | 0.05 | 0.98 | 3.21 | 1.35 | **0.0** |
| Constr-WS (SLO filter + sum) | **1.32** | 0.05 | 0.98 | 3.19 | 1.32 | 0.7 |
| WS-tuned (calibrated) | 3.19 | **0.01** | 0.99 | 3.20 | 1.28 | 3.7 |
| WS-default (untuned) | 23.02 | 1.94 | 0.45 | 1.22 | 0.65 | 94.7 |

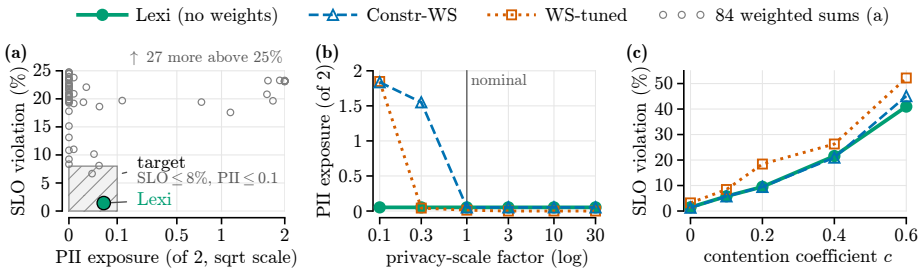


**Fig. 2.** (a) RQ1 weight sweep, hatched: target (6 seeds). (b) RQ3 privacy-scale miscalibration (8 seeds). (c) RQ4 co-location contention, breaking Assumption 1 (15 seeds).

report the *priority-inversion rate*, the share of decisions lexicographically worse than the strict priority optimum on shared estimates.

## 6 Results

### 6.1 RQ1/RQ2: tuning and objective satisfaction

A weighted sum reaches the paper's target operating region only after a deployment-specific search. Of 84 weightings swept over the objective simplex (Fig. 2(a)), only **one** meets a modest target (SLO $\leq$ 8% *and* PII $\leq$ 0.1) and **none** dominate LEXI (1.42% SLO in this 6-seed sweep). In the 15-seed main run (Table 3), LEXI has **1.46%** SLO violations (95% confidence interval (CI) $[1.31, 1.60]$) at PII exposure **0.05** from the order alone. It is on par with Constr-WS and better on SLO than the hand-tuned sum (paired $p < 0.001$), which is better on privacy. These methods occupy a nearby target region with different residual trade-offs. LEXI needs no weight tuning. Scale-robust sums approach LEXI's privacy at 2.3–3.3× its SLO violation, and only LEXI never inverts the order (0.0% vs. 0.7–5.0%).

### 6.2 RQ3: sensitivity to objective scaling

We next rescale the *units* of the privacy objective as the optimiser sees them, leaving true outcomes unchanged. LEXI's PII exposure stays **invariant** at 0.052 across a 300× swing (Fig. 2(b)), because selection depends only on the objective

*order* (Prop. 1). The tuned sum matches LEXI at nominal scale but silently re-prioritises. Under-weighting privacy 10× drives its PII from 0.01 to **1.85 of 2 stages** (augmented Tchebycheff: 1.86), and such a scale drifts and cannot be measured exactly. The rank- and min-max-normalised sums are invariant too, yet they sit at ≈ 3.5% SLO because they still *price* what they rank.

**Comparison with Constr-WS.** Constr-WS also needs no tuning for its SLO constraint and is close to LEXI on the headline row (1.32% vs 1.46%, with identical PII and carbon, and cost within 0.02). On the top priority the two use the same feasible set when at least one candidate is predicted SLO-feasible, because the hard constraint is the strict $\eta_0$=0 top-level band. Below the filter it applies a price where LEXI applies three more levels. Constr-WS inverts the priority order on 0.68% of decisions here and on **14.3**% under the real Azure load (the extended version), against exactly 0% for LEXI in both, and a 10× privacy under-scaling raises its exposure from 0.052 to **1.84 of 2 stages** (Fig. 2(b)). Its hard SLO constraint is scale-free, but the blend beneath it is not.

### 6.3 RQ4: robustness to curation, configuration and additivity

Curation matters most: a random menu moves LEXI from 1.44% to **43.6**% SLO violations (8 seeds). At a typical request 13.7% of the 40 curated candidates meet the SLO, 45.0% keep both PII stages private and 5.6% do both, and in the random menu these shares collapse to 3.9%, 19.4% and **0.2**%. The *expected-objective oracle* selects the priority optimum from noise-free simulator expectations and is then scored on a realized outcome. It violates the SLO on 1.48% of requests with the curated menu vs. LEXI's 1.44%. This ordering is possible because realized draws differ. **With the random menu the oracle itself violates on 43.53**% vs. LEXI's 43.56%. Thus absolute outcomes depend on the menu, while method ordering survives (43.6% LEXI, 45.8% WS-tuned, 59.8% WS-default).

The ordering also holds over an SLO sweep of 220–340 ms and an independent 10-node, 8-stage configuration with three PII stages (LEXI 6.78% < Constr-WS 7.17% < WS-tuned 8.14% < WS-default 13.81%). Co-location contention breaks Assumption 1 (Fig. 2(c)) and degrades *every* additive-model method together and gradually (LEXI 1.46 → 40.97% at $c$=0.6), preserving the ordering except at the extreme. It damages both SLO and privacy: LEXI's SLO violation rises from 1.46% to 40.97% and PII exposure from 0.05 to 1.49. Over the *whole* menu, including the 39 never-executed candidates per step, the model's mean absolute SLO-hinge error is **0.0286** normalized, and LEXI selects the expected-objective priority optimum on **99.7**% of decisions at under 0.1 ms each.

## 7 Discussion and Threats to Validity

We deliberately create priority conflict in the simulator (Sect. 5.1). The candidate menu therefore determines the achievable absolute outcomes (Sect. 6.3), while additivity (Assumption 1) defines the model boundary. The evaluation isolates

the decision rule from control-plane overhead, migration cost, telemetry delay, admission control, cold starts, and uninjected failures. It covers one application, and the privacy coordinate is a ranked topological preference rather than a compliance model. Hardware validation with concurrent applications is the next step from decision-level evidence to deployment performance.

## 8 Related Work

In continuum scheduling [3], three recent systems are closest. *HuntMS* [5] geo-distributes microservices for carbon and cost under an SLO *constraint*, *Cluster-Less* [7] orchestrates serverless workflows against a *deadline*, and *Festina* [18] minimises serverless energy under a latency-SLO constraint. Each optimises one scalar under a constraint without ranking the rest. We compare formulations rather than running these systems: their substrates are absent from our simulator, while *Constr-WS* (Sect. 6.2) captures the shared decision pattern of filtering on a threshold objective and then scalarising the feasible set. This excludes their engineering and limits the comparison. Constrained MDPs [2] likewise make "meet the SLO first" a hard constraint (level 0 of a lexicographic rule) and then price the rest. Lexicographic RL [9] ranks objectives but targets tabular control. Temporal-logic objectives also avoid hand-set weights and reduce optimally to average rewards [12], but constrain behaviour over time, whereas LEXI orders objectives in one decision. The same priority-first pattern governs post-quantum profile selection, with admissibility gates before cost ranking [11].

## 9 Conclusion

Operators express priorities as an order. We encode that order directly instead of hiding it inside exchange rates. LEXI applies strict lexicographic selection to predictions from an off-policy counterfactual outcome model. In our simulation, it reaches the weighted and constrained baselines' target region without tuning and remains unchanged under strictly increasing per-objective transformations. This result belongs to strict selection: positive slack introduces unit-dependent bands, and the candidate menu determines what any selector can achieve.

**Acknowledgments.** We acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this study.

## References

1. Abbasi-Yadkori, Y., Pál, D., Szepesvári, C.: Improved algorithms for linear stochastic bandits. In: Proc. NIPS. pp. 2312–2320 (2011)
2. Altman, E.: Constrained Markov Decision Processes. Chapman and Hall/CRC (1999)
3. Bonomi, F., Milito, R., Zhu, J., Addepalli, S.: Fog computing and its role in the Internet of things. In: Proc. ACM MCC. pp. 13–16 (2012)

4. Buesing, L., Weber, T., Zwols, Y., Racanière, S., Guez, A., Lespiau, J.B., Heess, N.: Woulda, coulda, shoulda: Counterfactually-guided policy search. In: Proc. ICLR (2019)
5. Christofidi, G., Álvarez-Terribas, F., Roumpos, I., Kourtellis, N., Omaña Iglesias, J., Doudali, T.D.: HuntMS: A framework for microservice geo-distribution for carbon and cost reduction. arXiv preprint arXiv:2603.10768v2 (2026), v1 titled "Aceso"
6. Continella, A., Polino, M., Pogliani, M., Zanero, S.: There's a hole in that bucket! a large-scale analysis of misconfigured S3 buckets. In: Proc. 34th Annual Computer Security Applications Conference (ACSAC). pp. 702–711 (2018). https://doi.org/10.1145/3274694.3274736
7. Farahani, R., Colosi, M., Murturi, I., Nastic, S., Villari, M., Dustdar, S., Prodan, R.: ClusterLess: Deadline-aware serverless workflow orchestration on federated edge clusters. In: Proc. IEEE ICDCS (2026), arXiv:2605.04310
8. Fishburn, P.C.: Lexicographic orders, utilities and decision rules: A survey. Manag. Sci. **20**(11), 1442–1471 (1974)
9. Gábor, Z., Kalmár, Z., Szepesvári, C.: Multi-criteria reinforcement learning. In: Proc. ICML. pp. 197–205 (1998)
10. Gan, Y., Zhang, Y., Cheng, D., Shetty, A., Rathi, P., Katarki, N., Bruno, A., Hu, J., Ritchken, B., Jackson, B., Hu, K., Pancholi, M., He, Y., Clancy, B., Colen, C., Wen, F., Leung, C., Wang, S., Zaruvinsky, L., Espinosa, M., Lin, R., Liu, Z., Padilla, J., Delimitrou, C.: An open-source benchmark suite for microservices and their hardware-software implications for cloud & edge systems. In: Proc. ASPLOS. pp. 3–18 (2019)
11. Le, X.B., Tran-Truong, P.T., Nguyen, D., Nguyen, N.K.T., Ha, X.S.: QuantumTrustKG: Provenance-aware orchestration for quantum-safe microservices. In: Proc. ICSOC. LNCS, Springer (2026), to appear
12. Le, X.B., Wagner, D., Witzman, L., Rabinovich, A., Ong, L.: Reinforcement learning with LTL and $\omega$-regular objectives via optimality-preserving translation to average rewards. In: Advances in Neural Information Processing Systems 37 (NeurIPS) (2024). https://doi.org/10.52202/079017-3718
13. Luo, S., Xu, H., Lu, C., Ye, K., Xu, G., Zhang, L., Ding, Y., He, J., Xu, C.: Characterizing microservice dependency and performance: Alibaba trace analysis. In: Proc. ACM SoCC. pp. 412–426 (2021)
14. Mao, H., Schwarzkopf, M., Venkatakrishnan, S.B., Meng, Z., Alizadeh, M.: Learning scheduling algorithms for data processing clusters. In: Proc. ACM SIGCOMM. pp. 270–288 (2019)
15. Miettinen, K.: Nonlinear Multiobjective Optimization, International Series in Operations Research & Management Science, vol. 12. Kluwer Academic Publishers (1999)
16. National Energy System Operator: Carbon intensity API. https://carbonintensity.org.uk/ (2026), regional half-hourly forecasts; accessed 24 September 2026
17. Shahrad, M., Fonseca, R., Goiri, Í., Chaudhry, G., Batum, P., Cooke, J., Laureano, E., Tresness, C., Russinovich, M., Bianchini, R.: Serverless in the wild: Characterizing and optimizing the serverless workload at a large cloud provider. In: Proc. USENIX ATC. pp. 205–218 (2020)
18. Wang, T., Rattihalli, G., Dhakal, A., Shangguan, L., Milojicic, D.: Energy-aware scheduling for serverless LLM serving on shared GPUs. arXiv preprint arXiv:2606.30391 (2026)