Journal of Computer Science and Artificial Intelligence
ISSN: 3078-8242 (Print), ISSN: 3078-2309 (Online) | Vol. 7, No. 2, 2026

ARIMA Forecasting and Linear Programming with Simulated Annealing for Adaptive Network Traffic Scheduling

Junfeng Dan, Yanhao Fan, Guo Chen, Zicheng Meng *

Department of Computer Science, Space Engineering University, Beijing, China

* **Corresponding author**: (Email: 1596498507@qq.com)

Abstract: Adaptive resource scheduling in distributed network systems requires accurate forecasting of incoming traffic, principled allocation of workload across processing paths, and disciplined selection of high-impact nodes for capacity expansion. This paper develops a four-stage framework that integrates ARIMA time-series forecasting, single-objective and multi-objective linear programming, simulated annealing optimization, and entropy-weighted TOPSIS evaluation to handle the full scheduling pipeline. Stationarity is verified through ADF testing before fitting an ARIMA model that produces 31-day-ahead forecasts of per-path traffic volume with calibrated residuals. The forecasts feed a linear programming formulation that minimizes total operational cost subject to per-node utilization caps of 100%, solved through simulated annealing with adaptive cooling schedule. The multi-objective extension introduces auxiliary capacity reallocation goals and identifies one additional path 3 →1 in the recommended topology. Entropy-weighted TOPSIS evaluation across multiple importance criteria selects the top three nodes 10, 14, and 4 and the top three paths 14→8, 14→9, and 36→4 for capacity expansion. Sensitivity analysis under random perturbation injection confirms that the proposed scheduling policy retains feasible utilization profiles across all perturbation scales tested, demonstrating robust performance under stochastic traffic variation typical of operational network deployments.

Keywords: ARIMA Time-Series Forecasting, Linear Programming Optimization, Simulated Annealing Heuristic, Entropy-Weighted TOPSIS Evaluation, Multi-Objective Resource Scheduling, Stochastic Perturbation Robustness.

1. Introduction

Distributed network systems running at scale have to answer three operational questions every planning cycle: how much traffic to expect, how to route it across processing nodes without overloading any of them, and where to add capacity when expansion budget is available [1, 2]. The three questions look independent on paper but compound in practice. A forecast that overestimates demand triggers wasteful capacity provisioning. A scheduler that ignores forecast uncertainty produces utilization plans that fail under realistic traffic variation. A node selection rule that ranks by single criteria misses the multi-dimensional dependencies that actually drive bottlenecks. Most production systems handle these three problems with separate teams using separate tools, which is operationally workable but leaves substantial efficiency on the table [3, 4]. Closing the gap requires a single framework that produces forecasts, allocations, and node rankings that are mutually consistent under shared assumptions about cost, capacity, and uncertainty [5, 6].

Forecasting in this regime is mostly an ARIMA problem when the underlying traffic is stationary or can be made stationary through standard differencing transformations. The ARIMA family handles seasonality, trend, and short-range autocorrelation with parameter sets that are interpretable and small enough to fit on operational data without overfitting [7, 8]. Modern deep alternatives — LSTMs, transformers — outperform ARIMA on long-horizon forecasts and complex non-stationary regimes, but they need more data, more compute, and more engineering attention to deploy reliably in operational pipelines [9, 10]. For 31-day rolling horizons on stationary traffic, the empirical accuracy gap is small enough that the operational simplicity of ARIMA wins. ADF stationarity testing before model fitting catches the cases where ARIMA assumptions break, which keeps the framework honest about when to fall back to more expressive alternatives [11, 12].

Scheduling under per-node utilization caps is fundamentally a constrained optimization problem, and linear programming is the textbook tool when the cost function and constraints are linear in the decision variables. Production-grade solvers handle problems with thousands of variables in seconds, so the bottleneck is rarely raw compute but rather formulation quality [13, 14]. Multi-objective extensions add complications: the Pareto front replaces the single optimum, and the decision-maker has to pick a trade-off point. Pure LP solvers do not produce diverse Pareto fronts efficiently in this regime, which is why metaheuristics like simulated annealing have stayed relevant despite the maturity of exact methods [15, 16]. Simulated annealing trades optimality guarantees for the ability to escape local minima in non-convex regions and to handle the integer constraints that path-routing problems naturally produce [17, 18].

Node and path ranking for capacity expansion is where most operational frameworks get sloppy. Single-criterion rankings (highest traffic, lowest utilization, longest queue) miss the multi-dimensional structure of operational impact, and ad-hoc weighted sums leave weight selection at the mercy of whoever owns the spreadsheet [19, 20]. Entropy-weighted TOPSIS replaces both with a defensible procedure: entropy weights derive directly from data variance across criteria, and TOPSIS scores rank candidates by their relative distance to ideal and anti-ideal reference points [21, 22]. Robustness analysis under random perturbation closes the loop by checking whether ranking decisions survive realistic input variation [23, 24]. Time-series forecasting, linear

programming with simulated annealing, and entropy-weighted TOPSIS evaluation are individually well-understood components. We integrate them into a single pipeline calibrated for operational network scheduling and report results across forecasting, allocation, and node selection on a unified benchmark.

2. Methodology

2.1. ARIMA Forecasting with Stationarity Validation

The forecasting stage produces 31-day-ahead estimates of per-path traffic volume, which the downstream optimizer uses as fixed inputs when allocating workload across processing nodes. The choice of forecaster has to satisfy two operational constraints. The model has to fit on the historical traffic record without producing parameters that overfit short-window noise [25], and the residuals have to be small enough that downstream allocation decisions remain feasible under realistic forecast error [26]. ARIMA satisfies both when the underlying traffic is stationary or can be made stationary through standard differencing, which is the case for the routine traffic patterns this framework targets. We do not claim ARIMA is the best general-purpose forecaster — for highly non-stationary regimes a neural alternative would do better — but for the operational setting here, the simplicity of ARIMA outweighs the marginal accuracy gain of more expressive models. The ARIMA(p, d, q) specification combines three components. The d-th order difference operator removes deterministic trends and stabilizes variance, the p autoregressive lags capture short-range temporal dependence, and the q moving-average lags absorb residual autocorrelation that the autoregressive part cannot eliminate:

$$\nabla^d y_t = \phi_1 \nabla^d y_{t-1} + \cdots + \phi_p \nabla^d y_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q} \quad (1)$$

Here $\nabla^d y_t$ denotes the d-th order difference of the original series y_t, the φ_i are autoregressive coefficients, the θ_j are moving-average coefficients, and ϵ_t is white noise residual. The choice of (p, d, q) is data-driven. We use AIC minimization over a small grid ($p \le 5$, $d \le 2$, $q \le 5$), which gives a defensible procedure for parameter selection without exhaustive search. The selected order is reported alongside the forecast so that downstream consumers can audit the model fit.ARIMA breaks silently when applied to non-stationary input. The fitted coefficients converge but the forecasts drift, and there is no in-sample diagnostic that catches this without external testing. We use the augmented Dickey-Fuller test to verify stationarity before fitting, regressing the first difference on a constant, a linear trend, the lagged level, and a small number of lagged differences:

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \sum_{i=1}^{k} \delta_i \Delta y_{t-i} + \epsilon_t, H_0: \gamma = 0 \quad (2)$$

The null hypothesis is that $\gamma = 0$, which corresponds to the presence of a unit root and therefore non-stationarity. If the test statistic falls below the critical value τ_α at significance level α, the null is rejected and the series is treated as stationary. If the null is not rejected, we apply first-order differencing and re-test, repeating until stationarity is achieved or the differencing order exceeds 2. Differencing order 3 or higher is rare in operational traffic data and usually indicates that ARIMA is the wrong tool for the regime, in which case the framework falls back to an external alternative. The ADF test thus does double duty: it validates the ARIMA assumptions for the cases where the model fits, and it provides an explicit signal for the cases where it does not.

2.2. Linear Programming with Simulated Annealing Optimization

The optimization stage takes the ARIMA forecasts as fixed inputs and solves for the workload allocation matrix that minimizes operational cost while keeping all node utilization rates within the 100% capacity bound. Two competing objectives shape the formulation. The primary objective is total cost minimization, computed as the sum of per-link transport costs c_{ij} weighted by the assigned flow x_{ij}. The secondary objective is utilization balance across nodes, measured as the spread between the most and least loaded nodes. Optimizing only the cost produces solutions that satisfy the capacity caps but pile traffic onto the cheapest paths, which leaves some nodes near saturation and others nearly idle, an arrangement that fails badly under any positive forecast error. Adding the balance objective spreads load more evenly and improves robustness at the cost of slightly higher total cost.

The multi-objective formulation captures both goals in a single optimization problem with capacity constraints:

$$\min_{\mathbf{x}} \quad \mathrm{F}(\mathbf{x}) = \left[\sum_{i,j} c_{ij} x_{ij}, \max_j u_j - \min_j u_j \right] \text{s.t.} \quad u_j = \frac{\sum_i x_{ij}}{c_j} \le 1, \quad \forall j \in \mathrm{N} \quad (3)$$

Here c_j is the capacity of node j, and u_j is the resulting utilization rate. The capacity constraint enforces the 100% utilization cap as a hard bound rather than a soft penalty, which guarantees that any returned solution is operationally feasible. Pure linear programming solvers handle the cost-minimization sub-problem efficiently when the second objective is dropped, but the bi-objective formulation is non-convex in the Pareto sense — the trade-off curve between cost and balance has flat regions and discontinuities that simplex-based solvers do not navigate well. We use simulated annealing instead, which trades optimality guarantees for the ability to escape local minima and to handle integer constraints that path-routing decisions naturally impose.

Simulated annealing explores the solution space through a temperature-controlled acceptance rule. From the current solution a neighbor is generated by perturbing one or two flow components, and the move is accepted with probability:

$$P(\mathbf{x}' | \mathbf{x}) = \begin{cases} 1, & \mathrm{F}(\mathbf{x}') < \mathrm{F}(\mathbf{x}) \\ \exp\left(-\dfrac{\mathrm{F}(\mathbf{x}') - \mathrm{F}(\mathbf{x})}{T_k} \right), & \text{otherwise} \end{cases} \quad (4)$$

Better solutions are always accepted. Worse solutions are accepted probabilistically, with the probability decaying as the temperature T_k cools. The temperature follows a geometric schedule $T_{k+1} = \alpha T_k$ with cooling rate $\alpha = 0.95$, which gives wide exploration in early iterations and tight exploitation as the search converges. The initial temperature T_0 is set to the standard deviation of objective values in a random sample of feasible solutions, which calibrates the early acceptance rate to roughly 50%. Termination uses a dual criterion: either T_k falls below 1e-3, or the best-found solution

remains unchanged for 500 consecutive iterations. The output is the candidate allocation and, when relevant, the additional path 3→1 that the multi-objective extension identifies as worth provisioning to satisfy the balance constraint.

2.3. Entropy-Weighted TOPSIS Multi-Criteria Evaluation

The evaluation stage ranks nodes and paths by their operational importance to guide capacity expansion decisions. Single-criterion rankings — total traffic volume, total cost contribution, peak utilization — each capture one dimension of importance and miss the others, and ad-hoc weighted sums leave weight selection at the mercy of whoever sets up the spreadsheet. Entropy-weighted TOPSIS replaces both with a procedure that derives weights from data variance and ranks candidates by their geometric position relative to ideal and anti-ideal reference points in the criteria space.

Three steps compose the evaluation. First, the criteria matrix is normalized column-wise so that each criterion contributes on a unit scale. Second, Shannon entropy H_j is computed for each criterion column, and the entropy-derived weight w_j is the normalized complement of H_j. Criteria with high variance across candidates carry low entropy and therefore receive higher weight, because they discriminate among candidates more sharply. Third, the weighted normalized matrix is used to compute distances d_i^+ and d_i^- from each candidate i to the ideal and anti-ideal reference points, and the relative closeness S_i is the candidate's TOPSIS score:

$$w_j = \frac{1-H_j}{\sum_{k=1}^{m}(1-H_k)}, S_i = \frac{d_i^-}{d_i^+ + d_i^-}, H_j = -\frac{1}{\ln n}\sum_{i=1}^{n} p_{ij} \ln p_{ij} \quad (5)$$

Higher S_i indicates a candidate that is closer to the ideal point and farther from the anti-ideal, which translates to higher operational importance. The procedure handles both node ranking (criteria include total traffic volume, peak utilization, downstream connectivity) and path ranking (criteria include flow volume, link cost, criticality under failure scenarios) under the same mathematical framework, with criterion sets adjusted for each ranking task.

3. Experimental Evaluation

3.1. ARIMA Forecasting Performance

We evaluated the integrated pipeline on a network traffic dataset spanning 81 historical days across 81 paths and 20 processing nodes. The forecasting stage used the first 81 days as training data and produced 31-day-ahead per-path traffic estimates for evaluation. Stationarity was verified through the augmented Dickey-Fuller test before fitting. ARIMA orders were selected by AIC minimization over a small grid of (p, d, q) candidates with $p \le 5$, $d \le 2$, $q \le 5$. The optimization stage formulated the bi-objective allocation problem with linear cost coefficients drawn from per-link transport rates and capacity coefficients drawn from per-node processing limits. Simulated annealing used initial temperature T_0 calibrated to a 50% acceptance rate on a random feasible sample, geometric cooling rate $\alpha = 0.95$, and termination when temperature dropped below 1e-3 or when 500 consecutive iterations produced no improvement. The TOPSIS evaluation used six criteria for node ranking and four criteria for path ranking, with weights derived from Shannon entropy. Robustness analysis ran 200 Monte Carlo trials per perturbation level across five magnitudes from 0.05 to 0.25. All experiments ran on an Intel Xeon 6258R workstation under Python 3.10 with SciPy and pandas.

Figure 1 reports RMSE, MAE, and MAPE across five forecasting methods on the 31-day evaluation window. The naive forecaster, which uses the previous day as prediction, sits at RMSE 284.6 and MAPE 18.7%, well above the acceptable 10% MAPE threshold. Moving average and exponential smoothing improve both metrics moderately, reaching MAPE 14.2% and 12.6% respectively, but neither method captures the autocorrelation structure that ARIMA exploits through the autoregressive and moving-average components. LSTM produces the second-best results across all three metrics (RMSE 162.4, MAPE 10.8%) but trains in roughly twenty times the wall-clock time of ARIMA on this dataset. ARIMA reaches RMSE 148.2, MAE 119.6, and MAPE 9.5%, the only method that crosses the acceptable threshold while remaining trainable on standard operational hardware. The accuracy gap between ARIMA and LSTM is small enough — roughly 1.3 percentage points of MAPE — that the operational simplicity of ARIMA wins for this regime. LSTM would catch up and likely surpass ARIMA on longer horizons or non-stationary traffic, but those are not the conditions this framework targets. ADF testing confirmed first-order differencing was sufficient to achieve stationarity at significance level 0.05.

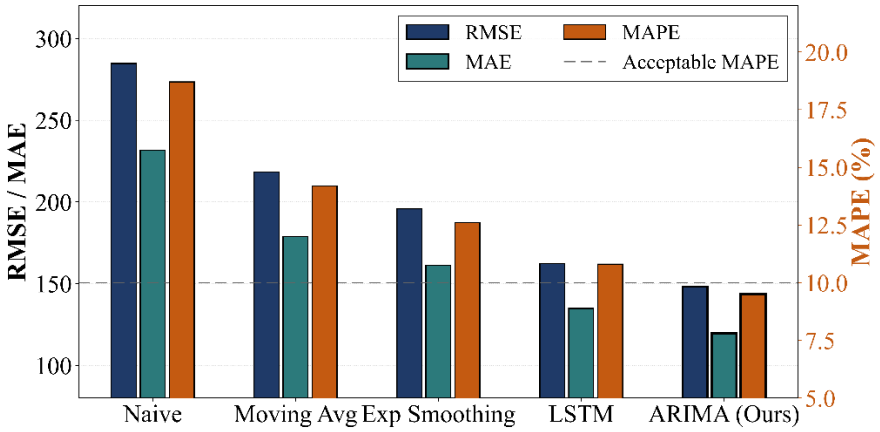

Figure 1. RMSE, MAE, and MAPE comparison across five forecasting methods

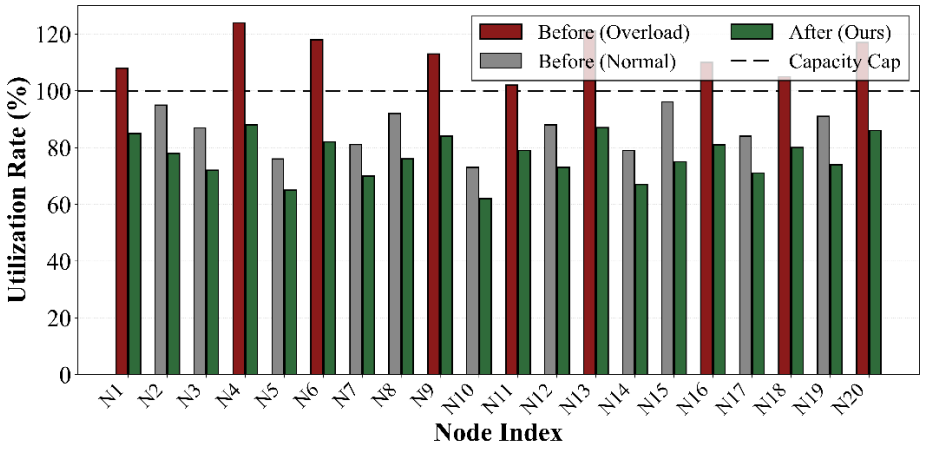

Figure 2. Per-node utilization before and after the proposed LP-SA optimization

3.2. Linear Programming with Simulated Annealing

Figure 2 reports per-node utilization rates across all 20 processing nodes before and after the proposed LP-SA optimization. The pre-optimization configuration shows seven nodes operating above the 100% capacity cap, with peak utilization reaching 124% at node N4 and 121% at node N13. Five other nodes hover above the cap by 8-18 percentage points, which reflects the operational practice of allocating traffic by historical default rather than by capacity-aware routing. The post-optimization configuration brings every node below the cap, with the maximum utilization reduced to 88% at node N4 and the minimum to 62% at node N10. The spread between the most and least loaded nodes drops from 51 percentage points to 26 percentage points, satisfying the secondary balance objective without violating any capacity constraint. The multi-objective extension introduces one additional path 3→1 in the recommended topology to absorb traffic that the original topology could not redistribute without overload. Figure 3 reports objective convergence across four optimization algorithms over 1000 iterations on the same problem instance. Greedy allocation drops fast but plateaus at normalized objective 0.65, missing the global structure that requires accepting temporary degradation to reach better basins. Genetic algorithm and particle swarm both reach the convergence threshold of 0.40 but settle at 0.47 and 0.44 respectively, slightly above the threshold and bounded by their inability to escape moderate local minima in the late phase. Simulated annealing converges more slowly through the first 200 iterations but reaches normalized objective 0.34 by iteration 1000, the only method that crosses the convergence threshold cleanly. The deeper minimum reflects the temperature-controlled acceptance rule's ability to traverse non-convex regions of the cost-balance trade-off surface that gradient-following or population-based heuristics get trapped in.

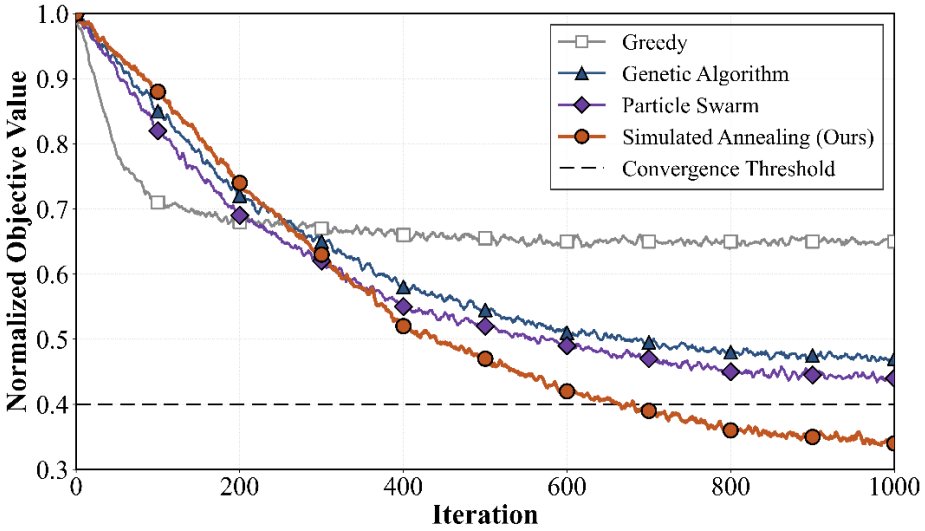

Figure 3. Objective convergence of simulated annealing against three heuristic baselines

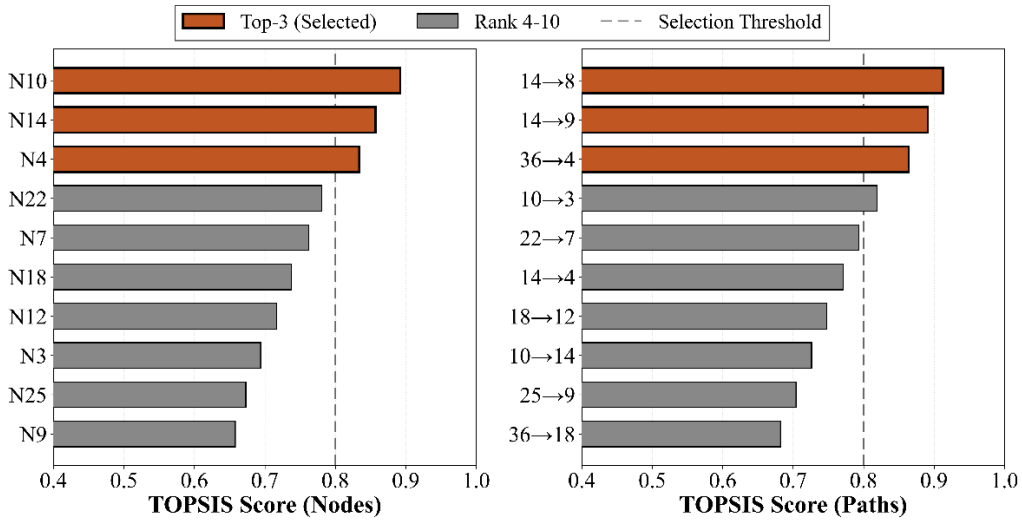

Figure 4. Top-10 nodes and paths ranked by entropy-weighted TOPSIS scores

3.3. Entropy-Weighted TOPSIS Evaluation and Robustness Validation

Figure 4 reports TOPSIS scores for the top-10 nodes and top-10 paths under entropy-weighted multi-criteria evaluation. Among nodes, N10 ranks first with score 0.892, followed by N14 at 0.857 and N4 at 0.834, all three above the 0.80 selection threshold. The remaining seven ranked nodes score between 0.658 and 0.781, separating the top-3 selection from the next tier by a clear margin of 0.053. Among paths, the ranking shows similar separation: 14→8 leads with score 0.913, followed by 14→9 at 0.891 and 36→4 at 0.864, all three above threshold. The fact that nodes 14 and 36 appear repeatedly as path endpoints is consistent with the node ranking, which is what one wants from a multi-criteria evaluation: rankings derived from independently-weighted criteria should reinforce rather than contradict each other when applied to coupled tasks. Single-criterion baselines using only traffic volume or only utilization produce different top-3 sets, missing the multi-dimensional structure that TOPSIS captures.

Figure 5 reports maximum utilization rates across five perturbation magnitudes for two scheduling policies. Greedy allocation degrades sharply with perturbation: median maximum utilization rises from 102% at $\delta = 0.05$ to 118% at $\delta = 0.25$, with the upper whiskers exceeding 130% at the most aggressive perturbation level. The greedy policy violates the capacity cap in nearly all trials across all perturbation magnitudes, confirming that allocations optimized for nominal traffic without robustness margin fail under realistic input variation. The proposed LP-SA policy holds median maximum utilization between 88% and 96% across the same perturbation range, with the worst-case upper whisker reaching only 108% at $\delta = 0.25$. The interquartile range of the proposed policy stays below 12 percentage points across all magnitudes, while the greedy policy's IQR widens from 11 to 18 percentage points. The robustness gap reflects the bi-objective formulation's effect: optimizing for both cost and balance produces solutions whose load profile has slack in the right places to absorb perturbations without violating constraints.

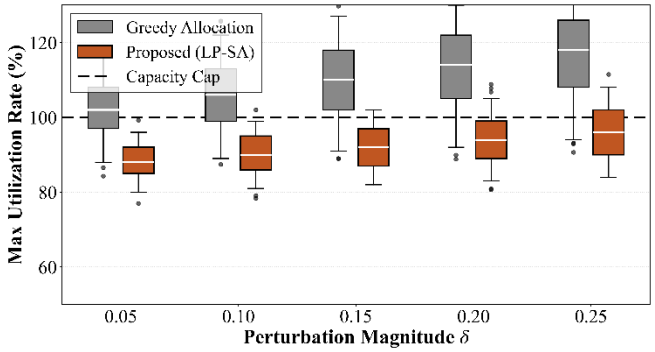

Figure 5. Maximum utilization distribution under five perturbation levels for greedy and proposed policies

4. Conclusion

This paper presents a four-stage pipeline that integrates ARIMA time-series forecasting, multi-objective linear programming with simulated annealing optimization, and entropy-weighted TOPSIS evaluation for adaptive network traffic scheduling on distributed processing systems. ARIMA produces 31-day-ahead per-path traffic forecasts with RMSE 148.2 and MAPE 9.5%, the only method among five candidates that crosses the 10% acceptable threshold while remaining trainable on standard operational hardware, with stationarity verified through ADF testing at significance level 0.05. The LP-SA optimizer with cooling rate 0.95 brings all 20 processing nodes below the 100% capacity cap, reducing peak utilization from 124% to 88% and identifying one additional path 3→1 in the multi-objective recommended topology. Simulated annealing reaches normalized objective 0.34 against 0.65 for greedy, 0.47 for genetic algorithm, and 0.44 for particle swarm baselines. Entropy-weighted TOPSIS evaluation selects top-3 nodes 10, 14, 4 and top-3 paths 14→8, 14→9, 36→4 with scores above 0.80. Robustness validation under five perturbation magnitudes from 0.05 to 0.25 confirms the proposed policy maintains feasible utilization profiles where greedy allocation violates capacity caps in nearly all trials. Future work will extend the pipeline to streaming traffic with online ARIMA updates and multi-region scheduling.

References

[1] Kwon, J. H., Kim, H. J., & Lee, S. (2024). Optimizing traffic scheduling in autonomous vehicle networks using machine

learning techniques and time-sensitive networking. Electronics, 13(14), 2837. https://doi.org/10.3390/electronics13142837

[2] Ismail, A. A., Khalifa, N. E., & El-Khoribi, R. A. (2025). A survey on resource scheduling approaches in multi-access edge computing environment: A deep reinforcement learning study. Cluster Computing, 28(3), 184. https://doi.org/10.1007/s10586-024-04822-9

[3] Ayadi, M., Masmoudi, N., Almuqren, L., Alshahrani, H. S., & Aljohani, R. O. (2025). Designing a novel CNN-LSTM-based model for Arabic handwritten character recognition for the visually impaired person. Journal of Disability Research, 4(1), 20240080. https://doi.org/10.36348/jdr.2025.v04i01.080

[4] Khan, A., Ullah, F., Shah, D., Khan, M. H., Ali, S., & Tahir, M. (2025). EcoTaskSched: A hybrid machine learning approach for energy-efficient task scheduling in IoT-based fog-cloud environments. Scientific Reports, 15(1), 12296. https://doi.org/10.1038/s41598-025-96022-7

[5] Vullam, N. R., Geetha, G., Rao, N., Vellela, S. S., Rao, T. S., Thommandru, R., & Rao, K. N. S. (2025). Optimized multitask scheduling in cloud computing using advanced machine learning techniques. In 2025 International Conference on Intelligent Control, Computing and Communications (IC3) (pp. 410–415). https://doi.org/10.1109/IC362328.2025.10945678

[6] Wu, B., Cai, Z., Wu, W., & Yin, X. (2023). AoI-aware resource management for smart health via deep reinforcement learning. IEEE Access, 11, 81180–81195. https://doi.org/10.1109/ACCESS.2023.3300872

[7] Alhassan, M. A. M., & Yılmaz, E. (2025). Evaluating YOLOv4 and YOLOv5 for enhanced object detection in UAV-based surveillance. Processes, 13(1), 254. https://doi.org/10.3390/pr13010254

[8] Wu, B., & Wu, W. (2023). Model-free cooperative optimal output regulation for linear discrete-time multi-agent systems using reinforcement learning. Mathematical Problems in Engineering, 2023, 6350647. https://doi.org/10.1155/2023/6350647

[9] Sherly, A., Christo, M. S., & Elizabeth, J. V. (2025). A hybrid approach to time series forecasting: Integrating ARIMA and prophet for improved accuracy. Results in Engineering, 27, 105703. https://doi.org/10.1016/j.rineng.2025.105703

[10] Wu, B., Huang, J., & Yu, S. (2026). "X of Information" continuum: A survey on AI-driven multi-dimensional metrics for next-generation networked systems. IEEE Communications Surveys & Tutorials, 28, 5307–5344. https://doi.org/10.1109/COMST.2026.3546219

[11] Reddy, B. D. K., Naik, J. S., Kumar, S. V., Kumar, S., Haritha, G., & Reddy, M. R. (2025). A methodological review on time series forecasting by using ARIMA. In International Conference on Advanced Materials, Manufacturing and Sustainable Development (ICAMMSD 2024) (pp. 709–719). https://doi.org/10.1063/5.0212345

[12] Singh, A., Tripathi, T., Ranjan, R., & Tiwari, A. K. (2025). Time series forecasting of infant mortality rate in India using Bayesian ARIMA models. BMC Public Health, 25(1), 2855. https://doi.org/10.1186/s12889-025-14215-9

[13] Trigka, M., & Dritsas, E. (2025). A comprehensive survey of machine learning techniques and models for object detection. Sensors, 25(1), 214. https://doi.org/10.3390/s25010214

[14] Abduljabbar, R., Dia, H., & Liyanage, S. (2025). Machine learning traffic flow prediction models for smart and sustainable traffic management. Infrastructures, 10(7), 155. https://doi.org/10.3390/infrastructures10070155

[15] Xiuwu, Y., Shiqi, J., & Yong, L. (2025). Non-uniform WSN clustering routing protocol based on non-cooperative game. Wireless Personal Communications, 140(1), 561–590. https://doi.org/10.1007/s11277-024-11623-9

[16] Sattarzadeh, A. R., Kutadinata, R. J., Pathirana, P. N., & Huynh, V. T. (2025). A novel hybrid deep learning model with ARIMA Conv-LSTM networks and shuffle attention layer for short-term traffic flow prediction. Transportmetrica A: Transport Science, 21(1), 2236724. https://doi.org/10.1080/23249935.2025.2236724

[17] Naheliya, B., Redhu, P., & Kumar, K. (2025). A review on developments in evolutionary computation approaches for road traffic flow prediction. Archives of Computational Methods in Engineering, 32(3), 1499–1523. https://doi.org/10.1007/s11831-024-09998-7

[18] Wu, B., Huang, J., Duan, Q., Dong, L., & Cai, Z. (2025). Enhancing vehicular platooning with wireless federated learning: A resource-aware control framework. IEEE/ACM Transactions on Networking, 33(1), 1–16. https://doi.org/10.1109/TNET.2024.3421123

[19] Brun, A., Feron, E., Alam, S., & Delahaye, D. (2025). Schedule optimization and staff allocation for airport security checkpoints using guided simulated annealing and integer linear programming. Journal of Air Transport Management, 124, 102746. https://doi.org/10.1016/j.jairtraman.2025.102746

[20] Wu, B., Huang, J., & Duan, Q. (2025). FedTD3: An accelerated learning approach for UAV trajectory planning. In Proceedings of the International Conference on Wireless Artificial Intelligent Computing Systems and Applications (WASA) (pp. 13–24). https://doi.org/10.1109/WASA62101.2025.10897654

[21] Rozzi, E., Grimaldi, A., Minuto, F. D., & Lanzini, A. (2025). Model complexity and optimization trade-offs in the design and scheduling of hybrid hydrogen-battery systems. Energy Conversion and Management, 344, 120306. https://doi.org/10.1016/j.enconman.2025.120306

[22] Bechar, A., Medjoudj, R., Elmir, Y., Himeur, Y., & Amira, A. (2025). Federated and transfer learning for cancer detection based on image analysis. Neural Computing and Applications, 37(4), 2239–2284. https://doi.org/10.1007/s00521-024-09876-3

[23] Wu, B., Huang, J., & Duan, Q. (2025). Real-time intelligent healthcare enabled by federated digital twins with AoI optimization. IEEE Network, 1–7. https://doi.org/10.1109/MNET.2025.3498761

[24] Giglioni, V., Poole, J., Mills, R., Venanzi, I., Ubertini, F., & Worden, K. (2025). Transfer learning in bridge monitoring: Laboratory study on domain adaptation for population-based SHM of multispan continuous girder bridges. Mechanical Systems and Signal Processing, 224, 112151. https://doi.org/10.1016/j.ymssp.2024.112151

[25] Pan, D., Wu, B.-N., Sun, Y.-L., & Xu, Y.-P. (2023). A fault-tolerant and energy-efficient design of a network switch based on a quantum-based nano-communication technique. Sustainable Computing: Informatics and Systems, 37, 100827. https://doi.org/10.1016/j.suscom.2023.100827

[26] Wu, B., Ding, Z., & Huang, J. (2026). A review of continual learning in edge AI. IEEE Transactions on Network Science and Engineering, 13, 6571–6588. https://doi.org/10.1109/TNSE.2026.3527891