Finite-Time Minimax Bounds in Queueing Control

Yujie Liu   Vincent Y. F. Tan   Yunbei Xu
National University of Singapore
yj-liu@nus.edu.sg, vtan@nus.edu.sg, yunbei@nus.edu.sg

Abstract

We establish the first *finite-time* minimax lower bounds—and derive new policies that achieve them—for the total queue length in scheduling problems over stochastic processing networks with adversarial arrivals. Prior analyses of MaxWeight guarantee only stability and asymptotic optimality in heavy traffic; we prove that, at finite horizons, MaxWeight can incur strictly larger backlog by problem-dependent factors which we identify. Our main innovations are 1) a *minimax framework* that pinpoints the precise problem parameters governing any policy's finite-time performance; 2) a minimax lower bound on total queue length; 3) fundamental limitation of MaxWeight that it is suboptimal in finite time; and 4) a new scheduling rule that minimizes the *full* Lyapunov drift—including its second-order term—thereby matching the lower bound under certain conditions, up to universal constants. These findings reveal a fundamental limitation on "drift-only" methods and points the way toward principled, non-asymptotic optimality in queueing control.

1 Introduction

We tackle the classical scheduling problem in a single–hop stochastic processing network (SPN), deciding at each time step which queue to serve so as to minimize total delay. Scheduling underpins virtually every modern communication and cloud-computing platform: from the landmark MaxWeight policies of Tassiulas and Ephremides [26] and Stolyar [25] to Google's Borg data-center [28]. More recently, it has become a linchpin of AI infrastructure, enabling throughput-optimized schedulers for GPU clusters that drive large language model training and inference [2, 13]. Whereas classical analyses focus on long-run averages under stationary traffic, today's applications demand rigorous, finite-time guarantees in the face of bursty, non-stationary workloads—e.g. NVIDIA's Run:AI scheduler [5] and similar systems [21]. Developing scheduling policies with provable delay bounds in these adversarial, time-critical regimes is crucial for next-generation responsiveness and efficiency.

The celebrated *MaxWeight scheduling policy* is widely regarded as the de facto solution for SPNs, due to its throughput-optimality [26] and asymptotic guarantees based on elegant Lyapunov drift analysis and diffusion approximations [25]. However, in finite-horizon regimes, MaxWeight can incur substantial backlogs, as demonstrated in prior works [23, 3] and validated by our experiments. Despite its prevalence, we know little about its worst-case performance and fundamental limitations in such non-asymptotic settings, leaving a crucial gap in both theory and practice.

In this work, we ask the following central questions: **What is the minimum achievable queue length by time T in single-hop SPNs? Can MaxWeight attain this minimum? If not, what alternative scheduling policies can possibly achieve it, and under what conditions?**

Contributions. We provide complete answers to our central questions. **First**, We establish a *minimax framework* that identifies the key problem parameters governing the finite-time performance of any policy (Section 2). **Second**, we derive the first *finite-time minimax lower bound* for any scheduling policy in single-hop SPNs (Section 3); this bound scales as $\Omega(\sqrt{T})$ at time T, with

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: MLxOR: Mathematical Foundations and Operational Integration of Machine Learning for Uncertainty-Aware Decision-Making.

two explicit constants depending on the arrival variance and the number of queues. **Third**, we show that MaxWeight is *not* minimax-optimal: its backlog can exceed the lower bound by a factor determined by the scheduling capacity (Section 4.2). **Finally**, we introduce $\mathrm{LyapOpt}$, a novel scheduling policy that minimizes the *full* Lyapunov drift, incorporating both first- and second-order terms (Section 4.1). We prove that $\mathrm{LyapOpt}$ achieves the minimax lower bound up to universal constants, thereby establishing the first finite-time optimal policy for single-hop SPNs (Section 4.1). Extensive simulations show that the $\mathrm{LyapOpt}$ significantly outperforms MaxWeight across various scenarios.

Bridging Queueing Control and Machine Learning. Traditional dynamic programming (DP) methods emphasize asymptotic analysis. We develop a minimax framework for finite-horizon analysis in structured control, expanding the theoretical foundations of both learning theory and DP, and opening new directions for short-horizon decision making. Unlike reinforcement learning regret analyses that benchmark against an optimal DP policy, our study characterizes the inherent performance of the policy class itself, complementing parameter-estimation frameworks.

2 Problem Setup and Minimax Framework

2.1 Problem Setup of Queueing Control

We consider a discrete-time single-hop SPN with n parallel queues, where jobs arrive exogenously and depart after a single service. Let $Q(t) \in \mathbb{R}_+^n$ denote the queue length vector at discrete time t, and $A(t) \in \mathbb{R}_+^n$ the arrival vector, where $A_i(t)$ is the number of jobs arriving to queue i for $1 \le i \le n$.

Queueing Dynamics and Scheduling Sequence. The system's dynamics are governed by the scheduling sequence $\mathcal{D} := \{\mathcal{D}_t\}_{t\in\mathbb{N}_0}$. Each element of $\mathcal{D}_t$, termed a *schedule*, specifies the number of jobs that can depart from each queue at time t. The *capacity region* associated with $\mathcal{D}_t$ is

$$\Pi(\mathcal{D}_t) \;=\; \big\{\gamma \in \mathbb{R}_+^n : \exists\, d \in \mathrm{conv}(\mathcal{D}_t) \text{ with } \gamma \le d\big\}.$$

Thus, $\Pi(\mathcal{D}_t)$ consists of all vectors dominated by convex combinations of schedules in $\mathcal{D}_t$. At each time t, the decision maker selects a schedule $D(t) \in \mathcal{D}_t$, and the queue lengths evolve according to the recursion below, with initial condition $Q(0) = \mathbf{0}$.

$$Q(t+1) = \max\{Q(t) - D(t), \mathbf{0}\} + A(t+1), \quad \text{for } t \in \mathbb{N}_0. \tag{1}$$

Arrival Process. We consider the general adversarial arrivals: The arrivals $A := \{A(t)\}_{t\in\mathbb{N}}$ may be chosen by an adversary, possibly with arbitrary dependencies across queues and across time. Let $\lambda(t) = \mathbb{E}[A(t)]$ denote the mean arrival rate vector at time t.

Policy. A *policy* Φ is defined as a sequence of scheduling rules, $\Phi = \{\phi_t\}_{t\ge 0}$, where each ϕ_t is a mapping from the history up to time t, denoted by

$$\mathcal{H}_t = \{Q(0), D(0), A(1), \ldots, Q(t-1), D(t-1), A(t), Q(t)\},$$

into a probability distribution over the scheduling set $\mathcal{D}_t$. At each time t, given the history $\mathcal{H}_t$, a schedule $D(t)$ is chosen randomly according to the distribution $\phi_t(\mathcal{H}_t)$.

Our goal is to minimize cumulative queue length. To achieve this, we analyze the fundamental limits of existing scheduling policies (notably the MaxWeight policy), and develop a novel Lyapunov-based policy that explicitly accounts for second-order terms of the Lyapunov drift, demonstrating optimality against minimax lower bound in the finite-time horizon regime.

2.2 Minimax Criteria

The minimax criterion is a standard approach to studying the intrinsic difficulty of problems in statistics and machine learning [29]. In this work, we extend this criterion to the domain of queueing control, aiming to bridge queueing theory with learning-theoretic methodologies and to motivate the use of statistical decision-theoretic tools in a broader class of DP problems.

Performance Metric. We use the total queue length as the performance metric (see [19]), which quantifies the overall system backlog. Formally, for $T \ge 1$, it is given by $\mathbb{E}[\sum_{i=1}^n Q_i(T)]$.

Model Classes: Arrival Process and Scheduling Sequence. The queueing system under consideration is defined by an arrival process A and a scheduling sequence $\mathcal{D}$. Adopting a minimax perspective, we define the following model class that includes adversarial variance-constrained arrival processes and capacity-constrained scheduling sequences:

$$\mathcal{M}(C,B) = \{(A,\mathcal{D}) : \lambda(t) \in \Pi(\mathcal{D}_t), \frac{1}{n}\sum_{i=1}^{n} \text{Var}(A_i(t)) \le C^2, t \in \mathbb{N}; \frac{1}{n}\sum_{i=1}^{n} d_i^2 \le B^2, \forall d \in \mathcal{D}_t, t \in \mathbb{N}_0\}$$

where $C \ge 0$ bounds arrival variability across queues and $B > 0$ bounds the scheduling capacity. We aim to find the fundamental minimax expected total queue length at time T:

$$\inf_{\Phi} \sup_{(A,\mathcal{D})\in\mathcal{M}(C,B)} \mathbb{E}_{\Phi,A}\left[\sum_{i=1}^{n} Q_i(T)\right]. \tag{2}$$

3 Minimax Lower Bounds

In this section, we derive a finite-time minimax lower bound on the expected total queue length for the system in Section 2. Formally, we seek a bound for (2). Intuitively, this lower bound captures the worst-case scenario an adversary can induce under variance constraints on arrivals and capacity constraints on scheduling, thereby establishing fundamental performance limits for any policy in single-hop SPNs.

Theorem 1 (Minimax Lower Bound)**.** *For any scheduling policy, and for arrival processes and scheduling sequences within the model class $\mathcal{M}(C,B)$, the following lower bound holds whenever $T > \left(\frac{2B^2}{C^2} + \frac{2C^2}{B^2} + 4\right)\mathbb{1}_{\{C>0\}} + 1$:*

$$\inf_{\Phi} \sup_{(A,\mathcal{D})\in\mathcal{M}(C,B)} \mathbb{E}_{\Phi,A}\left[\sum_{i=1}^{n} Q_i(T)\right] \ge \frac{nC\sqrt{T-2}}{4\sqrt{2e\pi}} + nB. \tag{3}$$

Theorem 1 reveals that no policy can guarantee a better scaling than $nC\sqrt{T}$ in finite horizon settings. This establishes a fundamental benchmark against which any scheduling algorithm can be compared.

4 Finite-Time Performance Guarantees

4.1 Optimal Lyapunov Policy

In this subsection, we propose LyapOpt, a scheduling policy that matches the lower bound (3) up to universal constants by optimizing both first- and second-order terms of the quadratic Lyapunov function $V(x) = \|x\|_2^2$. Given the queue length vector $Q(t) \in \mathbb{R}_+^n$, the schedule is chosen as

$$D^{\text{LyapOpt}}(t) \in \operatorname*{argmin}_{d\in\mathcal{D}_t} \sum_{i=1}^{n} \big(\max\{Q_i(t) - d_i, 0\}\big)^2,$$

with ties broken arbitrarily. This objective, derived from the queue dynamics (1), serves as a surrogate for the Lyapunov drift $\Delta V(t) = \mathbb{E}[V(Q(t+1) - A(t+1)) - V(Q(t) - A(t)) \mid \mathcal{H}_t]$. Now we state its performance guarantee.

Theorem 2 (Finite-Time Performance of the LyapOpt Policy)**.** *Within the model class $\mathcal{M}(C,B)$, the* LyapOpt *policy achieves the following bound on the expected total queue length:*

$$\mathbb{E}\Big[\sum_{i=1}^{n} Q_i(T)\Big] \le n\sqrt{\sum_{t=1}^{T-1} \mathbb{E}\Big[\min_{d\in\mathcal{D}_t} f(Q(t),d)\Big]/n + (T-1)C^2} + \sum_{i=1}^{n} \mathbb{E}[A_i(T)] \tag{4}$$

$$\text{with } f(Q(t),d) = \mathbb{E}\Bigg[\underbrace{2\sum_{i=1}^{n} Q_i(t)\big(\lambda_i(t) - d_i\big)}_{\text{first-order term}} + \underbrace{\sum_{i=1}^{n}\big(d_i^2 - \lambda_i(t)^2\big)}_{\text{second-order term}} \;\Bigg|\; \mathcal{H}_t\Bigg]. \tag{5}$$

If $\lambda(t) \in \mathcal{D}_t$ for all $t \in \mathbb{N}$, the expected total queue length satisfies

$$\mathbb{E}\Big[\sum_{i=1}^{n} Q_i(T)\Big] \le nC\sqrt{T-1} + \sum_{i=1}^{n} \mathbb{E}[A_i(T)]. \tag{6}$$

From (4), the advantage of $\mathrm{LyapOpt}$ is that it simultaneously optimizes first- and second-order terms by minimizing the full Lyapunov drift, enabling the departure vector to closely track the arrival rate. As shown in (6), when the arrival rate lies in $\mathcal{D}_t$, $\mathrm{LyapOpt}$ matches it exactly and achieves the lower bound up to a constant factor, thereby establishing finite-time optimality.

4.2 Limitation of MaxWeight

The MaxWeight policy is well known for throughput optimality and asymptotic guarantees, but it can exhibit notable performance limitations in finite time. In this subsection, we review established upper bounds [26, 25, 19, 30] and present a new lower bound showing that the factors in these bounds are necessary, thereby revealing fundamental limitations of the policy.

4.2.1 Upper Bound and Physical Meaning of MaxWeight

Consider the MaxWeight scheduling policy defined by selecting schedules according to:

$$D^{\text{MaxWeight}}(t) \in \operatorname*{argmax}_{d \in \mathcal{D}_t} \langle Q(t), d \rangle.$$

By optimizing the first-order Lyapunov term, MaxWeight prioritizes queues with larger backlogs. The finite-time upper bound for MaxWeight in $\mathcal{M}(C, B)$ is given below.

Theorem 3 (Upper Bound of MaxWeight Policy)**.** *Under the model class $\mathcal{M}(C, B)$, the MaxWeight policy satisfies the following upper bound on the expected total queue length:*

$$\mathbb{E}\Big[\sum_{i=1}^{n} Q_i(T)\Big] \leq n\sqrt{(B^2 + C^2)(T-1)} + \sum_{i=1}^{n} \mathbb{E}[A_i(T)]. \tag{7}$$

The upper bound reveals a key limitation of MaxWeight—and drift-based methods more broadly—namely, that they optimize only the first-order term. By always selecting extreme points, MaxWeight ignores the geometry of the arrival rate captured in the second-order term, which can lead to queue accumulation.

4.2.2 Lower Bound of MaxWeight for Dimension 2

The upper bound suggests that MaxWeight's performance depends on the number of queues n, the variance parameter C, and the capacity parameter B. While the lower bound in Theorem 1 highlights the roles of n and C, the following lower bound further reveals the inherent dependence on B.

Proposition 1. *There exists a family of instances in $\mathcal{M}(0, B)$ with $B \geq 3\sqrt{2}$, for which the expected total queue lengths under* $\mathrm{LyapOpt}$ *and* $\mathrm{MaxWeight}$ *satisfy:*

$$\sum_{i=1}^{2} Q_i^{\mathrm{LyapOpt}}(T) \leq 2 - \frac{1}{\sqrt{2}B}, \qquad T \geq 1;$$

$$\sum_{i=1}^{2} Q_i^{\mathrm{MaxWeight}}(T) \geq \frac{\sqrt{BT}}{2^{\frac{5}{4}}}, \qquad \left\lceil \frac{2B^2}{\sqrt{2}B - 1} \right\rceil \leq T \leq \left\lceil \Big(\frac{\sqrt{2}B}{2} - 1\Big)^3 \right\rceil + 1. \tag{8}$$

The $\sqrt{BT}$ gap in inequality (8) is evident for practical values of T and B (Appendix A.3.1). Moreover, extensive simulations show that $\mathrm{LyapOpt}$ consistently outperforms $\mathrm{MaxWeight}$ across diverse scenarios (Appendix A.3.2).

Consequently, the MaxWeight policy, which optimizes solely the first-order Lyapunov drift, inherently neglects higher-order terms crucial for achieving optimal finite-time performance. This neglect leads to excessive queue lengths, particularly pronounced in finite horizons and near-capacity scenarios. Both our theoretical insights and numerical results directly motivate the development of enhanced scheduling algorithms that explicitly incorporate second-order Lyapunov terms.

5 Conclusion and Future Directions

We revealed a fundamental finite-time gap between MaxWeight and the minimax lower bound for SPNs, and closed it with a second-order Lyapunov optimization policy. Future avenues include multi-hop networks, partial observability, and reinforcement learning approximations of $\mathrm{LyapOpt}$.

References

[1] J. Abate and W. Whitt. Transient behavior of the M/M/1 queue: Starting at the origin. *Queueing systems*, 2:41–65, 1987.

[2] R. Ao, G. Luo, D. Simchi-Levi, and X. Wang. Optimizing LLM inference: Fluid-guided online scheduling with memory constraints. *arXiv preprint arXiv:2504.11320*, 2025.

[3] M. Bramson, B. D'Auria, and N. Walton. Stability and instability of the MaxWeight policy. *Mathematics of Operations Research*, 46(4):1611–1638, 2021.

[4] J. G. Dai and W. Lin. Asymptotic optimality of maximum pressure policies in stochastic processing networks. *The Annals of Applied Probability*, 2008.

[5] R. Dar and E. Karabulut. NVIDIA open sources run:AI scheduler to foster community collaboration.

[6] A. Eryilmaz and R. Srikant. Asymptotically tight steady-state queue length bounds implied by drift conditions. *Queueing Systems*, 72:311–359, 2012.

[7] D. Freund, T. Lykouris, and W. Weng. Quantifying the cost of learning in queueing systems. *Advances in Neural Information Processing Systems*, 36:6532–6544, 2023.

[8] D. Gamarnik and D. A. Goldberg. On the rate of convergence to stationarity of the M/M/N queue in the Halfin–Whitt regime. *The Annals of Applied Probability*, 2013.

[9] G. R. Gupta and N. B. Shroff. Delay analysis for multi-hop wireless networks. In *Proceedings of the 28th IEEE International Conference on Computer Communications (INFOCOM)*, pages 2356–2364. IEEE, 2009.

[10] W. Kang and R. J. Williams. Diffusion approximation for an input-queued switch operating under a maximum weight matching policy. *Stochastic Systems*, 2(2):277–321, 2013.

[11] S. Krishnasamy, A. Arapostathis, R. Johari, and S. Shakkottai. On learning the cμ rule in single and parallel server networks. In *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 153–154. IEEE, 2018.

[12] S. Krishnasamy, R. Sen, R. Johari, and S. Shakkottai. Learning unknown service rates in queues: A multiarmed bandit approach. *Operations research*, 69(1):315–330, 2021.

[13] Y. Li, J. Dai, and T. Peng. Throughput-optimal scheduling algorithms for LLM inference and AI agents. *arXiv preprint arXiv:2504.07347*, 2025.

[14] M. J. Luczak and C. McDiarmid. On the maximum queue length in the supermarket model. *The Annals of Probability*, 2006.

[15] Y. Ma and S. T. Maguluri. Convergence rate analysis of the Join-the-Shortest-Queue system. *arXiv preprint arXiv:2503.15736*, 2025.

[16] S. T. Maguluri and R. Srikant. Heavy traffic queue length behavior in a switch under the Maxweight algorithm. *Stochastic Systems*, 6(1):211–250, 2016.

[17] A. Mandelbaum and A. L. Stolyar. Scheduling flexible servers with convex delay costs: Heavy-traffic optimality of the generalized cμ-rule. *Operations Research*, 52(6):836–855, 2004.

[18] M. Mundhenk, J. Goldsmith, C. Lusena, and E. Allender. The complexity of finite-horizon Markov decision process problems. *Journal of the ACM*, 47(4):681–720, 2000.

[19] M. Neely. *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool Publishers, 2010.

[20] C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of Markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.

[21] Y. Qiao, S. Anzai, S. Yu, H. Ma, Y. Wang, M. Kim, and H. Xu. Conserve: Harvesting GPUs for low-latency and high-throughput large language model serving. *arXiv preprint arXiv:2410.01228*, 2024.

[22] P. Robert. *Stochastic Networks and Queues*, volume 52. Springer Science & Business Media, 2013.

[23] D. Shah and D. Wischik. Optimal scheduling algorithms for input-queued switches. In *Proceedings of the 25th IEEE International Conference on Computer Communications: INFOCOM 2006*, pages 1–11. IEEE Computer Society, 2006.

[24] T. Stahlbuhk, B. Shrader, and E. Modiano. Learning algorithms for minimizing queue length regret. *IEEE Transactions on Information Theory*, 67(3):1759–1781, 2021.

[25] A. L. Stolyar. Maxweight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic. *The Annals of Applied Probability*, 14(1):1–53, 2004.

[26] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. In *29th IEEE Conference on Decision and Control*, 1990.

[27] J. S. Van Leeuwaarden and C. Knessl. Transient behavior of the Halfin–Whitt diffusion. *Stochastic Processes and their Applications*, 121(7):1524–1545, 2011.

[28] A. Verma, L. Pedrosa, M. Korupolu, D. Oppenheimer, E. Tune, and J. Wilkes. Large-scale cluster management at Google with Borg. In *Proceedings of the Tenth European Conference on Computer Systems (EuroSys)*, 2015.

[29] A. Wald. Statistical decision functions which minimize the maximum risk. *Annals of Mathematics*, 46(2):265–280, 1945.

[30] N. Walton and K. Xu. Learning and information in stochastic networks and queues. In *Tutorials in Operations Research: Emerging Optimization Methods and Modeling Techniques with Applications*, pages 161–198. INFORMS, 2021.

[31] Z. Yang, R. Srikant, and L. Ying. Learning while scheduling in multi-server systems with unknown statistics: Maxweight with discounted UCB. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 206 of *Proceedings of Machine Learning Research*, pages 4275–4312. PMLR, 2023.

A Technical Appendices and Supplementary Material

A.1 Related Works

MaxWeight and asymptotic optimality. The MaxWeight (or backpressure) policy, first proposed by Tassiulas and Ephremides [26], is celebrated for its throughput-optimality in general SPNs. Subsequent work has refined its delay guarantees in various settings via fluid and diffusion approximations [25, 17, 4, 10]. However, these analyses are fundamentally asymptotic and do not provide direct performance guarantees in finite-horizon settings.

Finite-horizon analyses in queueing. Analyzing queueing behavior over finite horizons remains challenging, even for simple models like the $M/M/1$ queue [1]. Most existing work addresses uncontrolled systems, using tools such as coupling [22] and spectral methods [8, 27] to study convergence to steady state. In contrast, finite-horizon analysis for controlled systems is far more limited, with only a few results for routing policies like Join-the-Shortest-Queue (JSQ) [14, 15]. The finite-time behavior of general scheduling policies remains largely unexamined.

Parameter learning in queueing. A growing body of work studies queueing systems where key parameters—such as arrival and service rates—are unknown and must be learned online. A common performance metric is queueing regret, which quantifies the excess queue length incurred by a learning algorithm relative to an oracle policy with complete knowledge of the system [12, 11, 24, 7]. Other works use the time-averaged queue length over a finite horizon to evaluate learning efficiency [31].

In contrast, our work assumes full knowledge of the system parameters and aims to characterize the fundamental gap between the finite-horizon delay incurred by a scheduling policy and the ideal baseline of zero.

Lower bounds for structured dynamic programming. Queueing control is a structured dynamic programming problem, yet solving such problems is often intractable due to the exponential growth of state and action spaces—the so-called "curse of dimensionality" [20, 18]. These computational barriers motivate the study of fundamental performance limits. Despite its importance, prior work on delay lower bounds in queueing systems is limited; a notable exception is Gupta and Shroff [9], who derive lower bounds on delay in multi-hop networks by reducing to the delay of a $G/D/1$ queue, though this remains challenging to analyze in finite time. We establish the first minimax lower bound on finite-horizon queueing delay, setting a benchmark for evaluating scheduling policies.

Drift-method limitations and alternatives. The Lyapunov drift framework underlies most stability and steady-state analyses in queueing systems [6, 16], supporting the asymptotic optimality of many drift-based methods. However, these methods typically rely on coarse first-order approximations, which often obscure transient inefficiencies and fail to directly optimize key performance metrics. Extensions such as drift-plus-penalty [19] introduce auxiliary objectives (e.g., delay or energy), but still lack explicit finite-time performance guarantees. In contrast, we propose a scheduling policy that minimizes the full Lyapunov drift—capturing both first- and second-order terms—to optimize the performance metric in finite time.

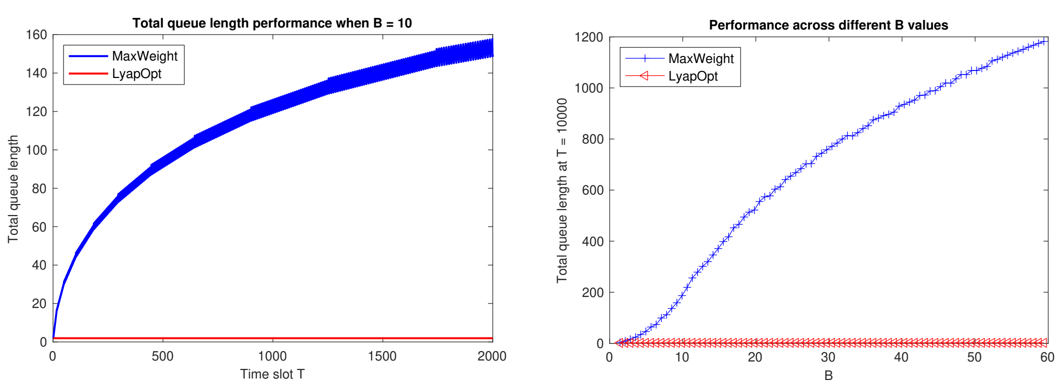
A.2 Notation

Let $\mathbb{N}_0 := \{0, 1, 2, \dots\}$ and $\mathbb{R}^n_+ := \{x \in \mathbb{R}^n : x_i \geq 0,\ 1 \leq i \leq n\}$. For any finite set $S \subset \mathbb{R}^n$, $\mathrm{conv}(S)$ denotes the convex hull of S. We use $\mathbf{0}$ to denote the zero vector, with the dimension understood from the context. Given two vectors $x, y \in \mathbb{R}^n$ $\max\{x, y\}$ denotes their componentwise maximum, that is, $(\max\{x, y\})_i = \max\{x_i, y_i\}$ for each i. Let $\mathrm{Var}(X)$ denote the variance of the random variable X.

A.3 Numerical Experiments

A.3.1 Performance Gap Between $\mathrm{LyapOpt}$ and MaxWeight for Dimension 2

The $\sqrt{BT}$ gap as seen in in inequality (8) is evident for practical values of T and B, as demonstrated in Figures 1a and 1b, where $b = \sqrt{2}B$. However, as shown in Figures 2a and 2b, the total queue length under MaxWeight eventually becomes bounded (i.e., $O(1)$ in T) as T increases. The lower bound in Proposition 1 is established under the assumption of deterministic arrivals (i.e., $C = 0$). Deriving a similar lower bound for the total queue length under stochastic arrival processes remains an open problem.

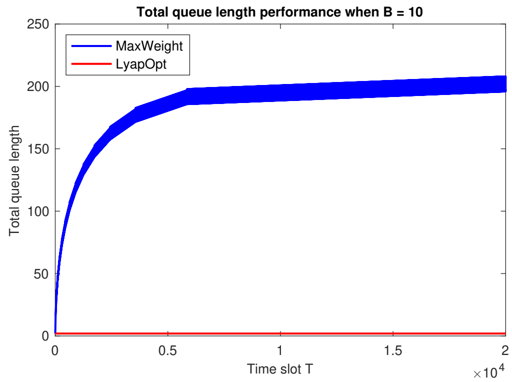


(a) Total queue length when $B = 10$

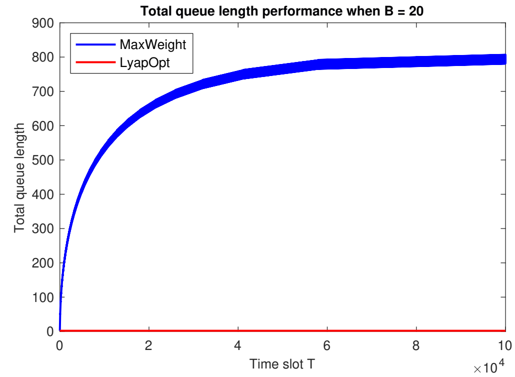
(b) Total queue length across different B

Figure 1: Performance comparison of MaxWeight and LyapOpt policies versus B

The model in (16) often arises in systems like wireless networks or data centers, where two queues–$Q_1(t)$ and $Q_2(t)$–have similar arrival rates $(1,\ 1 - \frac{1}{b})$ with $b \gg 1$, and share a resource whose service



(a) Total queue length when $B = 10$



(b) Total queue length when $B = 20$

Figure 2: Performance comparison of MaxWeight and LyapOpt policies versus B

shifts between rate b for Q_1 and rate 1 for Q_2. MaxWeight tends to over-prioritize Q_1 by selecting extreme points, causing backlog in the harder-to-serve $Q_2(t)$ and revealing its finite-time inefficiency. In contrast, LyapOpt adapts to the arrival rate and avoids backlog in this deterministic setting.

A.3.2 Additional Experiments with More Queues

We compare the finite-time performance of LyapOpt and MaxWeight as the number of parallel queues n increases from 2 to 8. For each n, the scheduling set $\mathcal{D}_t = \mathcal{D}_\star$ is generated by uniformly sampling $10n$ vectors with integer entries between 1 and 10. Then, 2000 arrival rate vectors—each representing a distinct scenario—are sampled uniformly from the boundary of the capacity region $\Pi(\mathcal{D}_\star)$. Arrivals follow Binomial distributions with fixed variance 1 per queue, calibrated to match the sampled arrival rate vectors. Each simulation run spans 1000 time slots, and results are averaged over 100 independent runs. Table 1 reports, for each dimension n, the proportion of scenarios where the ratio

$$\text{ratio} = \frac{\text{Total Queue Length } (\text{LyapOpt}) \text{ at } t = 1000}{\text{Total Queue Length } (\text{MaxWeight}) \text{ at } t = 1000}$$

falls below 1, 0.9, and 0.5, respectively—indicating cases where LyapOpt outperforms MaxWeight. These results show that for a wide range of queues, LyapOpt consistently outperforms MaxWeight, sometimes substantially so.

Figures 3a and 3b show a representative case with $n = 8$ queues and an arrival rate on the boundary of the capacity region, where LyapOpt significantly outperforms MaxWeight. In Figure 3a, both policies appear to exhibit $\sqrt{T}$ growth, but LyapOpt maintains consistently lower total queue length. Figure 3b further shows reduced queue imbalance under LyapOpt, as reflected in the lower sum of squared queue lengths.

Table 1: Scenarios with ratios $\leq 1, 0.9, 0.5$

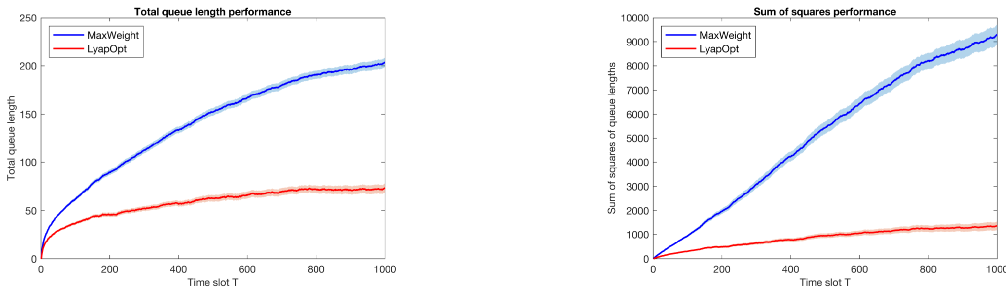
n	≤ 1	≤ 0.9	≤ 0.5
2	84.7%	25.9%	0%
3	97.5%	54.1%	36.3%
4	99.9%	78.5%	46.1%
5	100%	67.0%	31.3%
6	97.4%	71.3%	26.5%
7	100%	90.0%	45.9%
8	100%	80.7%	35.9%

A.4 Proof of Theorem 1

A.4.1 Proof Sketch

We outline the key ideas in establishing the lower bound, with the formal proof provided in the Appendix A.4.3. Fix a time-invariant scheduling sequence $\mathcal{D}$ with $\mathcal{D}_t = \mathcal{D}_\star$ for all $t \in \mathbb{N}_0$, and assume that $\mathcal{D}$ satisfies the condition in model class $\mathcal{M}(C, B)$.

1. **A policy-independent lower bound: reducing from policy vectors to capacity values.** Let $M = \max\{\sum_{i=1}^n d_i, d \in \mathcal{D}_\star\}$ denote the maximum total number of jobs that can depart from the system in a single time slot. Comparing the cumulative arrivals with the maximum possible cumulative



(a) Total queue length.

(b) Squared queue length.

Figure 3: Finite-time comparison of MaxWeight and LyapOpt policies ($n = 8$).

departures up to time T, we define $S(T-1) = \sum_{t=1}^{T-1} X(t)$ with $X(t) = \sum_{i=1}^{n} A_i(t) - M$. Then by (1), for any arrival process A and any scheduling policy, the total queue length satisfies

$$\sum_{i=1}^{n} Q_i(T) \geq \max\{S(T-1), 0\} + \sum_{i=1}^{n} A_i(T). \tag{9}$$

2. **Constructing hard instances with dependent binary-valued arrivals.** Inspired by expert problem lower bounds, we construct A as to be i.i.d. over t. When the $A_i(t)$ are independent, $\text{Var}(X(t)) \propto n$; if they are fully dependent, $\text{Var}(X(t)) \propto n^2$. To maximize the fluctuation of $S(T-1)$, we choose arrivals that are perfectly correlated across queues. Consider the following construction of the arrival process A: for each t,

$$\mathbb{P}(A_1(t) = K\lambda_1, A_2(t) = K\lambda_2, \ldots, A_n(t) = K\lambda_n) = 1/K,$$
$$\mathbb{P}(A_1(t) = 0, A_2(t) = 0, \ldots, A_n(t) = 0) = (K-1)/K.$$

With this construction, $\mathbb{E}[A(t)] = \lambda$ and $\{X(t)\}_{t\in\mathbb{N}}$ forms an i.i.d. sequence of binary random variables. We then select $\lambda \in \mathcal{D}_\star$ such that $\sum_{i=1}^{n} \lambda_i = M$, and choose the tuning parameter K such that the variance constraint in the model class $\mathcal{M}(C, B)$ holds with equality (see Appendix A.4.3 for the details). Consequently, $\mathbb{E}[X(t)] = 0$ and $\text{Var}(X(t)) = M^2(K-1)$. The construction of such binary random variables allows us to derive explicit finite-time lower bounds, rather than relying on asymptotic approximations via the central limit theorem.

3. **Right-Tail Overshoot Lower Bound for Sums of Binary Random Variables.** We establish a right-tail lower bound for binary sums using nontrivial anti-concentration arguments, leveraging precise control over discrete fluctuations. Specifically, by Lemma 2 in Appendix A.4.2, we obtain a lower bound on $\mathbb{E}[\max\{S(T-1), 0\}]$ simply by scaling the corresponding bound by M.

4. **Constructing hard instance for the scheduling set.** Construct the scheduling set

$$\mathcal{D}_\star = \left\{(x_1, x_2, \ldots, x_n) | x_1^2 + x_2^2 + \cdots + x_n^2 = nB^2, x_i \geq 0 \text{ for } 1 \leq i \leq n\right\}.$$

This choice of $\mathcal{D}_\star$ ensures that the scheduling sequence $\mathcal{D}$ satisfies the constraint of the model class $\mathcal{M}(C, B)$. In this case, we can choose $\lambda = (B, B, \ldots, B)$ so that $M = nB$, and set $K - 1 = \frac{C^2}{B^2}$ (see Appendix A.4.3 for the details). Applying the scaling factor M to the lower bound of Lemma 2 and then invoking (9) yields (3). □

A.4.2 Supporting Lemmas

Lemma 1. *Given the arrival process A and scheduling sequence $\mathcal{D}$, for any scheduling policy, the total queue length satisfies*

$$\sum_{i=1}^{n} Q_i(T) \geq \max\left\{\sum_{s=1}^{T-1}\Big(\sum_{i=1}^{n} A_i(s) - M_s\Big), \sum_{s=2}^{T-1}\Big(\sum_{i=1}^{n} A_i(s) - M_s\Big), \ldots, \sum_{i=1}^{n} A_i(T-1) - M_{T-1}, 0\right\} + \sum_{i=1}^{n} A_i(T),$$

where $M_t = \max\left\{\sum_{i=1}^{n} d_i, d \in \mathcal{D}_t\right\}$.

Lemma 2. *Let $\{Z(t)\}_{t\in\mathbb{N}}$ be a sequence of i.i.d. binary random variables, where each $Z(t)$ takes the value $K-1$ with probability $1/K$ and -1 with probability $(K-1)/K$, for some $K>1$. Define the partial sum $S(t)=\sum_{s=1}^{t} Z(s)$. Then we have for all $t>\frac{K}{K-1}$,*

$$\mathbb{E}\big[\max\{S(t),0\}\big] \geq \frac{1-\delta(t;K)}{\sqrt{2\pi}} \exp\left(\frac{-K^2(t-1)}{12((K-1)t-K)(t-K)}\right)\sqrt{(K-1)(t-1)}, \quad (10)$$

with $\delta(t;K)=\max\left\{\frac{K}{t},\frac{K}{t(K-1)}\right\}$. In particular, for all $t>2K^2/(K-1)$,

$$\mathbb{E}\big[\max\{S(t),0\}\big] \geq \frac{1}{2\sqrt{2e\pi}}\sqrt{(K-1)(t-1)}. \quad (11)$$

A.4.3 Formal Proof

Proof of Theorem 1. Fix a time-invariant scheduling sequence $\mathcal{D}$ with $\mathcal{D}_t=\mathcal{D}_\star$ for all $t\in\mathbb{N}_0$, and assume that $\mathcal{D}$ that satisfies the condition in model class $\mathcal{M}(C,B)$. By Lemma 1, for any arrival process A and any scheduling policy, the total queue length satisfies

$$\sum_{i=1}^{n} Q_i(T) \geq \max\left\{\sum_{t=1}^{T-1}\left(\sum_{i=1}^{n} A_i(t)-M\right),0\right\}+\sum_{i=1}^{n} A_i(T),$$

where $M=\max\left\{\sum_{i=1}^{n} d_i, d\in\mathcal{D}_\star\right\}$. Define $S(T-1)=\sum_{t=1}^{T-1} X(t)$ with $X(t)=\sum_{i=1}^{n} A_i(t)-M$. We can rewrite the lower bound as

$$\sum_{i=1}^{n} Q_i(T) \geq \max\{S(T-1),0\}+\sum_{i=1}^{n} A_i(T).$$

Now consider the following construction of the arrival process A: for each t,

$$\mathbb{P}\Big(A_1(t)=K\lambda_1, A_2(t)=K\lambda_2,\ldots,A_n(t)=K\lambda_n\Big)=\frac{1}{K},$$
$$\mathbb{P}\Big(A_1(t)=0, A_2(t)=0,\ldots,A_n(t)=0\Big)=\frac{K-1}{K},$$

where

$$\lambda \in \operatorname*{argmin}_{\substack{\gamma\in\Pi(\mathcal{D}_\star)\\ \sum_{i=1}^{n}\gamma_i=M}} \left\{\sqrt{\frac{1}{n}\sum_{i=1}^{n}\gamma_i^2}\right\} \quad \text{and} \quad K=\frac{nC^2}{\sum_{i=1}^{n}\lambda_i^2}+1. \quad (12)$$

With this construction, we have

$$\mathbb{E}[A(t)]=\lambda, \quad \text{and} \quad \mathrm{Var}(A_i(t))=\frac{(K-1)^2\lambda_i^2+\lambda_i^2(K-1)}{K}=(K-1)\lambda_i^2. \quad (13)$$

Such a construction ensures that $\frac{1}{n}\sum_{i=1}^{n}\mathrm{Var}(A_i(t))=C^2$ and $\{X(t)\}_{t\in\mathbb{N}}$ forms an i.i.d. sequence of binary random variables, taking the value $M(K-1)$ with probability $1/K$ and $-M$ with probability $(K-1)/K$, with variances of $M^2(K-1)$. Thus, by Lemma 2, we obtain a lower bound on $\mathbb{E}[\max\{S(T-1),0\}]$ simply by scaling the corresponding bound in (11) by M.

Now we consider the scheduling set

$$\mathcal{D}_\star=\left\{(x_1,x_2,\ldots,x_n)|x_1^2+x_2^2+\cdots+x_n^2=nB^2, x_i\geq 0 \text{ for } 1\leq i\leq n\right\}.$$

This choice of $\mathcal{D}_\star$ ensures that the scheduling sequence $\mathcal{D}$ satisfies the constraint of the model class $\mathcal{M}(C,B)$. In this case, we can choose $\lambda=(B,B,\ldots,B)$ so that $M=nB$, and set $K-1=\frac{C^2}{B^2}$. Applying the scaling factor M to the lower bound (11) of Lemma 2 and then invoking (9) yields (3). ☐

A.5 Formal Proofs of Theorems 2 and 3

Before presenting the specific proof, we first develop a general Lyapunov drift analysis applicable to any scheduling policy. For any arrival processes and scheduling sequences in $\mathcal{M}(C, B)$, and any scheduling policy Φ, consider the one-step Lyapunov drift:

$$
\begin{aligned}
&\mathbb{E}\left[V\big(Q(t+1) - A(t+1)\big) - V\big(Q(t) - A(t)\big) \mid \mathcal{H}_t\right] \\
&= \mathbb{E}\left[\sum_{i=1}^{n}(\max\{Q_i(t) - D_i(t), 0\})^2 - \sum_{i=1}^{n}(Q_i(t) - A_i(t))^2 \middle| \mathcal{H}_t\right] \\
&\leq \sum_{i=1}^{n} \mathbb{E}\left[(Q_i(t) - D_i(t))^2 - Q_i(t)^2 + 2Q_i(t)A_i(t) - A_i(t)^2 \mid \mathcal{H}_t\right] \\
&= \sum_{i=1}^{n} \mathbb{E}\left[D_i(t)^2 - A_i(t)^2 + 2Q_i(t)\big(A_i(t) - D_i(t)\big) \mid \mathcal{H}_t\right] \\
&= f(Q(t), D(t)) + r(Q(t), A(t)),
\end{aligned}
$$

where

$$
f(Q(t), d) = \mathbb{E}\Big[\underbrace{2\sum_{i=1}^{n} Q_i(t)\big(\lambda_i(t) - d_i\big)}_{\text{first-order term}} + \underbrace{\sum_{i=1}^{n}\big(d_i^2 - \lambda_i(t)^2\big)}_{\text{second-order term}} \Big| \mathcal{H}_t\Big],
$$

and

$$
r(Q(t), A(t)) = \sum_{i=1}^{n}\left[2Q_i(t)(A_i(t) - \lambda_i(t)) + \lambda_i(t)^2 - A_i(t)^2\right].
$$

Note that

$$
\begin{aligned}
&\mathbb{E}\big[Q_i(t)A_i(t)\big] \\
&= \mathbb{E}\left[\big(Q_i(t) - A_i(t)\big)A_i(t)\right] + \mathbb{E}\big[A_i(t)^2\big] \\
&\overset{(a)}{=} \mathbb{E}\big[Q_i(t) - A_i(t)\big]\mathbb{E}\big[A_i(t)\big] + \mathbb{E}\big[A_i(t)^2\big] \\
&\overset{(b)}{=} \mathbb{E}\big[Q_i(t)\big]\lambda_i(t) + \mathrm{Var}(A_i(t)),
\end{aligned}
$$

where (a) follows from the independence between $Q_i(t) - A_i(t)$ and $A_i(t)$, and (b) follows from the relation $\mathbb{E}\big[A_i(t)^2\big] = \lambda_i(t)^2 + \mathrm{Var}(A_i(t))$. Then we have

$$
\mathbb{E}[r(Q(t), A(t))] = \sum_{i=1}^{n}\mathrm{Var}(A_i(t)).
$$

By taking expectation and summing over for $1 \leq t \leq T-1$, we have

$$
\begin{aligned}
\mathbb{E}[V(Q(T) - A(T)] &\leq \mathbb{E}[V(Q(1) - A(1))] + \sum_{t=1}^{T-1}\mathbb{E}[f(Q(t), D(t))] + \sum_{t=1}^{T-1}\sum_{i=1}^{n}\mathrm{Var}(A_i(t)) \\
&\leq \sum_{t=1}^{T-1}\mathbb{E}[f(Q(t), D(t))] + (T-1)nC^2, \qquad (14)
\end{aligned}
$$

where the last inequality follows from the relation $Q(1) = A(1)$. Note that

$$
\begin{aligned}
\left(\mathbb{E}\Big[\sum_{i=1}^{n}\big(Q_i(T) - A_i(T)\big)\Big]\right)^2 &\overset{(c)}{\leq} \mathbb{E}\Big[\big(\sum_{i=1}^{n}\big(Q_i(T) - A_i(T)\big)\big)^2\Big] \\
&\overset{(d)}{\leq} n\mathbb{E}\Big[\sum_{i=1}^{n}\big(Q_i(T) - A_i(T)\big)^2\Big] \\
&= n\mathbb{E}\big[V\big(Q(T) - A(T)\big)\big].
\end{aligned}
$$

Here (c) follows from Jensen's inequality and (d) follows from the Cauchy-Schwartz inequality. Substituting this expression into (14), we have

$$\mathbb{E}\Big[\sum_{i=1}^{n} Q_i(T)\Big] \leq \sqrt{\sum_{t=1}^{T-1} n\mathbb{E}[f(Q(t), D(t))] + (T-1)n^2C^2} + \sum_{i=1}^{n} \mathbb{E}[A_i(T)]. \tag{15}$$

Proof of Theorem 2. For the LyapOpt policy, the one-step Lyapunov drift

$$\begin{aligned}\Delta V(t) &= \min_{d \in \mathcal{D}_t} \mathbb{E}\Big[\sum_{i=1}^{n}(\max\{Q_i(t) - d, 0\})^2 - \sum_{i=1}^{n}(Q_i(t) - A_i(t))^2 \,\Big|\, \mathcal{H}_t\Big] \\ &\leq \min_{d \in \mathcal{D}_t} f(Q(t), d) + r(Q(t), A(t)).\end{aligned}$$

Applying a similar derivation as in the proof of (15) yields the bound in (4). When $\lambda(t) \in \mathcal{D}_t$ for all $t \geq 0$, the bound in (6) holds since $\min_{d \in \mathcal{D}_t} f(Q(t), d) \leq 0$. □

Proof of Theorem 3. For the MaxWeight policy, recall (5). The MaxWeight policy minimizes the first-order term and ensures that this term remains non-positive at each step, since $\lambda(t) \in \Pi(\mathcal{D}_t)$ and, by definition of $\Pi(\mathcal{D}_t)$, $D^{\text{MaxWeight}}(t)$ maximizes $\langle Q(t), d\rangle$ over all $d \in \Pi(\mathcal{D}_t)$. Besides, the second-order term is bounded by nB^2. Substituting into (15), we obtain (7). □

A.6 Proof of Proposition 1

A.6.1 Proof Sketch

Proof sketch. To show MaxWeight's finite-time suboptimality, we consider the following arrivals and time-invariant scheduling sequence $\mathcal{D}$,

$$\begin{aligned}&\mathcal{D}_t = \mathcal{D}_\star = \{d \in \mathbb{R}^2 : d = x(b, 0) + (1-x)(0, 1), 0 \leq x \leq 1\}, t \in \mathbb{N}_0 \\ &A(t) = (1, (b-1)/b) \text{ for all } t \geq 2, \text{ and } A(1) = (1, (b-1)/b - \varepsilon),\end{aligned} \tag{16}$$

where $b = \sqrt{2}B$, and $\varepsilon = 0$ if b is irrational; otherwise, $\varepsilon > 0$ is a small irrational constant (so that no two schedules tie under MaxWeight at finite time, and it always selects either $(0, 1)$ or $(b, 0)$). Under MaxWeight, one can show: At each time t, MaxWeight selects $(b, 0)$ unless $\frac{Q_2(t)}{Q_1(t)} \geq b$; As a result, $Q_2(t)$ must accumulate to b before it forces the policy to use $(0, 1)$; however, doing so causes $Q_1(t)$ to increase to 2, requiring $Q_2(t)$ to build up to $2b$ before $(0, 1)$ can be used again more frequently. This alternating pattern causes $Q_2(t)$ to grow at a rate of approximately $\sqrt{bT}$ over a finite horizon T. In contrast, our policy always chooses the "true arrival" schedule $(1, (b-1)/b)$, maintaining constant queue lengths $O(1)$. See Appendix A.6.2 for details. □

A.6.2 Formal Proof

Proof. Since the arrival rate lies in the scheduling set $\mathcal{D}_\star$ and the arrival variance is $C = 0$, Theorem 2 and (6) yield

$$\mathbb{E}\Big[\sum_{i=1}^{2} Q_i^{\text{LyapOpt}}(T)\Big] = \sum_{i=1}^{2} \mathbb{E}[A_i(T-1)] = 2 - \frac{1}{b} = 2 - \frac{1}{\sqrt{2}B}.$$

Now we consider the MaxWeight policy under

$$\begin{aligned}&\mathcal{D}_t = \mathcal{D}_\star = \{d \in \mathbb{R}^2 : d = x(b, 0) + (1-x)(0, 1), 0 \leq x \leq 1\}, t \in \mathbb{N}_0, \\ &A(t) = (1, (b-1)/b) \text{ for all } t \geq 2, \text{ and } A(1) = (1, (b-1)/b - \varepsilon),\end{aligned} \tag{17}$$

where $b = \sqrt{2}B$, and $\varepsilon = 0$ if b is irrational; otherwise, $\varepsilon > 0$ is a small irrational constant. Here we choose ε such that $\lceil b^2/(b-1) + \varepsilon\rceil = \lceil b^2/(b-1)\rceil$, which is always possible since $b^2/(b-1)$ is never an integer for $b \geq 6$.

By the queue dynamics, $Q(1) = (1, 1-1/b-\varepsilon)$. MaxWeight repeatedly selects $(b, 0)$ until $Q_2(t) \geq b$. Thus $Q(t) = (1, (1-1/b)t - \varepsilon)$ for $1 \leq t \leq \lceil b^2/(b-1) + \varepsilon\rceil = \lceil b^2/(b-1)\rceil$. And it is clear to see that $Q_2(t)$ can never be smaller than $b - 1/b$, so the $-\varepsilon$ term is always preserved in $Q_2(t)$, making it irrational for all t. In the following, we consider only time horizons where $Q_1(t) < b$ holds throughout, ensuring that $Q_1(t)$ always remains an integer. Consequently, there is never a tie in MaxWeight decisions, and it either selects $(b, 0)$ or $(0, 1)$. Therefore, we henceforth restrict our analysis to the MaxWeight policy under the system

$$\begin{aligned}&\mathcal{D}_t = \mathcal{D}_\star = \{(b,0),(0,1)\}, t \in \mathbb{N}_0 \text{ for } b \geq 6,\\ &A(t) = (1,(b-1)/b) \text{ for all } t \geq 2, \text{ and } A(1) = (1,(b-1)/b - \varepsilon),\end{aligned} \tag{18}$$

where $b = \sqrt{2}B$, and $\varepsilon = 0$ if b is irrational; otherwise, $\varepsilon > 0$ is a small irrational constant such that $\lceil b^2/(b-1) + \varepsilon \rceil = \lceil b^2/(b-1) \rceil$.

At each time t, the MaxWeight policy will choose $(0, 1)$ only when $Q_2(t) > bQ_1(t)$ and $(b, 0)$ when $bQ_1(t) > Q_2(t)$. Let $t_0 = 0$ and

$$t_k = \min\{t : Q_2(t) \geq kb\} \text{ for } 1 \leq k \leq \lfloor b/2 \rfloor.$$

We will show, by induction, that

(i) t_k is well defined;

(ii) $Q_1(t_k) = 1$;

(iii) For $t_{k-1} \leq t < t_k$, $1 \leq Q_1(t) \leq k$; $(k-1)b - (k-1)/b \leq Q_2(t) < kb$.

For $k = 1$, by the above discussion, $t_1 = \lceil b^2/(b-1) \rceil$ is well defined, $Q_1(t_1) = 1$, and (iii) naturally holds. And it is clear to see that $Q_2(t)$ can never be smaller than $b - 1/b$.

Now suppose that (i)-(iii) hold for some $k - 1$ with $k \geq 2$. We show (i)-(iii) hold for k. Note the fact that at any time t, choosing $(0, 1)$ gives

$$Q_1(t+1) = Q_1(t) + 1, Q_2(t+1) = Q_2(t) - \tfrac{1}{b},$$

whereas choosing $(b, 0)$ yields

$$Q_1(t+1) = \max\{Q_1(t) - b, 0\} + 1 = 1, Q_2(t+1) = Q_2(t) + 1 - \tfrac{1}{b}.$$

Thus Q_1 increases only when $(0, 1)$ is used, and Q_2 increases only when $(b, 0)$ is used. Starting from $t \geq t_{k-1}$, and noting $Q_1(t_{k-1}) = 1$, consider the next k time slots. There must be at least one use of $(b, 0)$, because if the first $k - 1$ departures were all $(0, 1)$, then $Q_1(t_{k-1} + k - 1) = k$, forcing the kth departure to be $(b, 0)$. Moreover, if only consider $k - 1$ time slots after t_{k-1}, at most one $(b, 0)$ can be used, because for $s \leq k$, $Q_2(t + s) \geq (k - 1)b - s/b > (k - 2)b$. Note that $Q_2(t_{k-1} + k) \geq (k - 1)b + 1 - k/b > (k - 1)b$. By the same reasoning and the fact that $Q_2(t_{k-1} + k + s) \geq (k - 1)b + 1 - (k + s)/b > (k - 1)b$ for $s \leq k$, in each subsequent block of k slots, Q_2 increases by exactly $1 - k/b > 0$. Hence t_k is well defined, and since Q_2 only increases when $(b, 0)$ is applied, we conclude $Q_2(t_k) = 1$. Finally, (iii) follows directly from the analysis above. And we complete the induction. In addition, we also have

$$\left\lceil \frac{k(b-1)}{1-k/b} \right\rceil - 1 \leq t_k - t_{k-1} \leq \left\lceil \frac{kb}{1-k/b} \right\rceil.$$

Therefore, by (iii), for all $t_{k-1} \leq t \leq t_k$, we have $Q_1(t) + Q_2(t) \geq (k - 1)b$ and

$$(k-1)b = \sqrt{t}b\frac{k-1}{\sqrt{t}} \geq \sqrt{t}b\frac{k-1}{\sqrt{t_k}}.$$

Note that

$$t_k = t_1 + \sum_{i=2}^{k-1}(t_k - t_{k-1}) \leq \sum_{i=1}^{k-1}\left\lceil \frac{ib}{1-i/b} \right\rceil + 1.$$

Claim 1. For $b \geq 4$ and $k \geq 2$,

$$\frac{k-1}{\sqrt{t_k}} \geq \frac{1}{2\sqrt{b+1}}.$$

With the help of Claim 1, for $t_1 \le t \le t_{\lfloor \frac{b}{2} \rfloor}$,

$$Q_1(t) + Q_2(t) \ge \frac{b\sqrt{t}}{2\sqrt{b+1}} \ge \frac{\sqrt{bt}}{3}.$$

Claim 2. For $b \ge 6$,

$$t_{\lfloor \frac{b}{2} \rfloor} \ge \frac{b^3}{16}.$$

Therefore, for $t_1 \le t \le b^3/16$,

$$Q_1(t) + Q_2(t) \ge \frac{\sqrt{bt}}{3}.$$

And for $t \le t_1 = \left\lceil \frac{b^2}{1-b} \right\rceil$,

$$Q_1(t) + Q_2(t) \ge 1 + (1 - 1/b)t - \varepsilon \ge (1 - 1/b)t.$$

□

Proof of Claim 1. For $2 \le k \le \lfloor b/2 \rfloor$ and each $1 \le i \le k-1$ we have $1 - i/b > 1/2$. Hence

$$\left\lceil \frac{ib}{1 - i/b} \right\rceil \le \frac{ib}{1 - i/b} + 1 \le 2ib + 1.$$

Summing over $i = 1, 2, \ldots, k-1$ yields

$$t_k \le \sum_{i=1}^{k-1} (2ib + 1) = (k-1)(bk+1).$$

It follows that

$$\sqrt{t_k} \le \sqrt{(k-1)(b\,k+1)} \le \sqrt{k^2\,(b+1)} \le k\sqrt{b+1}.$$

Hence

$$\frac{k-1}{\sqrt{t_k}} \ge \frac{k-1}{k\,\sqrt{b+1}} = \frac{1 - \frac{1}{k}}{\sqrt{b+1}}.$$

Finally, since $k \ge 2$ implies $1 - \frac{1}{k} \ge \frac{1}{2}$, we conclude

$$\frac{k-1}{\sqrt{t_k}} \ge \frac{1}{2\sqrt{b+1}}.$$

□

Proof of Claim 2. Note that

$$t_{\lfloor \frac{b}{2} \rfloor} = t_1 + \sum_{i=2}^{\lfloor \frac{b}{2} \rfloor} (t_k - t_{k-1}) \ge \sum_{i=1}^{\lfloor \frac{b}{2} \rfloor} \left(\left\lceil \frac{ib}{1 - i/b} \right\rceil - 1 \right).$$

Let $k = \lfloor \frac{b}{2} \rfloor$. Since $1 \le i \le k \le \frac{b}{2}$, we have

$$\sum_{i=1}^{k} \Big(\left\lceil \tfrac{ib}{1-i/b} \right\rceil - 1 \Big) \ge \sum_{i=1}^{k} \Big(\frac{ib^2}{b-i} - 1 \Big) \ge \frac{b^2}{b-1} \cdot \frac{k(k+1)}{2} - k.$$

Since $k = \lfloor b/2 \rfloor \ge \frac{b-2}{2}$, we get

$$\frac{b^2}{b-1} \cdot \frac{k(k+1)}{2} \ge \frac{b^2}{b-1} \cdot \frac{(b-2)b}{8} \ge \frac{b^3(b-2)}{8(b-1)}.$$

Therefore

$$\sum_{i=1}^{k} \Big(\left\lceil \tfrac{ib}{1-\frac{i}{b}} \right\rceil - 1 \Big) \ge \frac{b^3(b-2)}{8(b-1)} - k \ge \frac{b^3(b-2)}{8(b-1)} - \frac{b}{2} \ge \frac{b^3}{16}.$$

□

A.7 Proofs of Lemmas 1 and 2

Proof of Lemma 1. Let $D(t) \in \mathcal{D}$ denote the schedule selected by the scheduling policy at time t. Let $\Delta(t) = A(t) - D(t)$ for all $t \geq 1$. By (1), for $1 \leq i \leq n$ and $T \geq 1$, we have

$$\begin{aligned}
&Q_i(T) - A_i(T) \\
&= \max\{Q_i(T-1) - D_i(T-1), 0\} \\
&= \max\{\max\{Q_i(T-2) - D_i(T-2), 0\} + A_i(T-1) - D_i(T-1), 0\} \\
&= \max\{Q_i(T-2) - D_i(T-2) + \Delta_i(T-1), \Delta_i(T-1), 0\} \\
&= \max\left\{Q_i(1) - D_i(1) + \sum_{s=2}^{T-1} \Delta_i(s), \sum_{s=2}^{T-1} \Delta_i(s), \ldots, \Delta_i(T-1), 0\right\} \\
&= \max\left\{\sum_{s=1}^{T-1} \Delta_i(s), \sum_{s=2}^{T-1} \Delta_i(s), \ldots, \Delta_i(T-1), 0\right\}, \qquad (19)
\end{aligned}$$

where the last equation follows that $Q_i(0) = 0$ and then $Q_i(1) = A_i(1)$.

Since $\sum_{i=1}^{n} D_i(t) \leq M$ for all t, by equation (19), we have

$$\begin{aligned}
\sum_{i=1}^{n} Q_i(T) &= \sum_{i=1}^{n} \max\left\{\sum_{s=1}^{T-1} \Delta_i(s), \sum_{s=2}^{T-1} \Delta_i(s), \ldots, \Delta_i(T-1), 0\right\} + \sum_{i=1}^{n} A_i(T) \\
&\geq \max\left\{\sum_{s=1}^{T-1}\sum_{i=1}^{n} \Delta_i(s), \sum_{s=2}^{T-1}\sum_{i=1}^{n} \Delta_i(s), \ldots, \sum_{i=1}^{n} \Delta_i(T-1), 0\right\} + \sum_{i=1}^{n} A_i(T) \\
&\geq \max\left\{\sum_{s=1}^{T-1}\left(\sum_{i=1}^{n} A_i(s) - M_s\right), \sum_{s=2}^{T-1}\left(\sum_{i=1}^{n} A_i(s) - M_s\right), \ldots, \right. \\
&\qquad \left. \sum_{i=1}^{n} A_i(T-1) - M_{T-1}, 0\right\} + \sum_{i=1}^{n} A_i(T).
\end{aligned}$$

□

Proof of Lemma 2. A simple calculation shows

$$\mathbb{E}\big[\max\{S(t), 0\}\big] = \sum_{k=k_0}^{t} (kK - t)\binom{t}{k}\left(\tfrac{1}{K}\right)^k\left(\tfrac{K-1}{K}\right)^{t-k} = \frac{t\,(K-1)^{m_0+1}}{K^t}\binom{t-1}{m_0}, \qquad (20)$$

where $k_0 = \left\lfloor \frac{t}{K} \right\rfloor + 1$ and $m_0 = t - k_0$. In fact, we can prove by induction on m that

$$G_m := \sum_{k=t-m}^{t} (kK - t)\binom{t}{k}\Big(K-1\Big)^{t-k} = t\,(K-1)^{m+1}\binom{t-1}{m}$$

for $0 \leq m \leq t-1$. Note that when $m = 0$, we have $G_0 = t(K-1)$. So the identity holds in the base case. Now assume the equality holds for some $m > 0$, and we prove it for $m+1$.

$$\begin{aligned}
G_{m+1} &= \big((t-m-1)K - t\big)\binom{t}{t-m-1}\Big(K-1\Big)^{m+1} + G_m \\
&= \Big(K-1\Big)^{m+1}\left(\big((t-m-1)K - t\big)\frac{t!}{(m+1)!(t-m-1)!} + \frac{t!}{(t-m-1)!m!}\right) \\
&= \Big(K-1\Big)^{m+1}\frac{t!}{(m+1)!(t-m-2)!}\left(\frac{\big((t-m-1)K - t\big)}{t-m-1} + \frac{m+1}{t-m-1}\right) \\
&= \Big(K-1\Big)^{m+2}\frac{t!}{(m+1)!(t-m-2)!} \\
&= t\,(K-1)^{m+2}\binom{t-1}{m+1}.
\end{aligned}$$

Thus, we have completed the proof by induction.

We focus on the case $m_0 > 0$, or equivalently, $t > K/(K-1)$, for the expression (20). By the Stirling approximation, for any integer $r \geq 1$,

$$\sqrt{2\pi r}\left(\frac{r}{e}\right)^r \leq r! \leq \sqrt{2\pi r}\left(\frac{r}{e}\right)^r e^{\frac{1}{12r}}.$$

Apply the lower bound to $(t-1)!$ and the upper bound to $m_0!$ and $(t-1-m_0)!$. Then

$$\binom{t-1}{m_0} = \frac{(t-1)!}{m_0!\,(t-1-m_0)!} \geq \frac{C_0(t)}{\sqrt{2\pi}} \sqrt{\frac{t-1}{m_0\,(t-1-m_0)}}\; \frac{(t-1)^{\,t-1}}{m_0^{m_0}\,(t-1-m_0)^{\,t-1-m_0}}$$

with $C_0(t) = \exp\left(-\frac{1}{12m_0} - \frac{1}{12(t-1-m_0)}\right)$. Hence

$$\mathbb{E}\big[\max\{S(t),0\}\big] \geq \frac{C_0(t)t\,(K-1)^{m_0+1}}{\sqrt{2\pi}K^t} \sqrt{\frac{t-1}{m_0\,(t-1-m_0)}}\; \frac{(t-1)^{\,t-1}}{m_0^{m_0}\,(t-1-m_0)^{\,t-1-m_0}}.$$

And we can show that for $t > K/(K-1)$,

$$\frac{(K-1)^{m_0+1}}{K^t}\frac{(t-1)^{\,t-1}}{m_0^{m_0}\,(t-1-m_0)^{\,t-1-m_0}} \geq \frac{K-1}{K}\min\left\{1-\frac{K}{t}, 1-\frac{K}{t(K-1)}\right\}. \tag{21}$$

We defer its proof to later, assuming it holds, we immediately obtain

$$\mathbb{E}\big[\max\{S(t),0\}\big] \geq \min\left\{1-\frac{K}{t}, 1-\frac{K}{t(K-1)}\right\}\frac{K-1}{K}\frac{C_0(t)t}{\sqrt{2\pi}}\sqrt{\frac{t-1}{m_0\,(t-1-m_0)}}. \tag{22}$$

Note that

$$m_0 = t-k_0 \;\leq\; t-\frac{t}{K} = t\,\frac{K-1}{K}, \quad t-1-m_0 \;\leq\; \frac{t}{K}.$$

We have

$$m_0(t-1-m_0) \leq t\frac{K-1}{K}\frac{t}{K} = \frac{K-1}{K^2}t^2,$$

and therefore

$$\sqrt{\frac{t-1}{m_0\,(t-1-m_0)}} = \sqrt{\frac{t-1}{\frac{K-1}{K^2}\,t^2}} = \frac{K}{\sqrt{K-1}}\frac{\sqrt{t-1}}{t}.$$

Plug this into (22), we have

$$\mathbb{E}\big[\max\{S(t),0\}\big] \geq \frac{C_0(t)}{\sqrt{2\pi}}\min\left\{1-\frac{K}{t}, 1-\frac{K}{t(K-1)}\right\}\sqrt{(K-1)(t-1)}.$$

It is clear to see that

$$\begin{aligned} C_0(t) &= \exp\left(\frac{-(t-1)}{12m_0(t-1-m_0)}\right) = \exp\left(\frac{-(t-1)}{12(t-1-\lfloor t/K\rfloor)\lfloor t/K\rfloor}\right) \\ &\geq \exp\left(\frac{-(t-1)}{12(t-1-t/K)(t/K-1)}\right) = \exp\left(\frac{-K^2(t-1)}{12((K-1)t-K)(t-K)}\right). \end{aligned}$$

Therefore, we have (10). To complete the proof, it suffices to show (21).

Write $p_0 = \frac{m_0}{t-1}$ and $H(p) = -\big[p\ln p + (1-p)\ln(1-p)\big]$. Then

$$\frac{(t-1)^{\,t-1}}{m_0^{m_0}\,(t-1-m_0)^{\,t-1-m_0}} = \exp\big((t-1)\,H(p_0)\big)$$

and

$$\frac{(K-1)^{m_0+1}}{K^t} = \exp\big((m_0+1)\ln(K-1) - t\ln K\big).$$

To show (21), we only need to prove that for $t > K/(K-1)$,

$$\begin{aligned} F(p_0, K) &:= (t-1)\, H(p_0) \; + \; (m_0+1)\ln(K-1) \; - \; t\ln K \\ &\geq \min\left\{\ln\frac{(t-K)(K-1)}{tK}, \ln\frac{t(K-1)-K}{tK}\right\}. \end{aligned} \tag{23}$$

Note that

$$F(p_0, K) \; = \; (t-1)\Big[\, H(p_0) + p_0\ln(K-1) - \ln K\Big] \; + \; \ln(K-1) - \ln K.$$

A direct algebraic check shows, for $p_1 = 1 - 1/K$,

$$H(p_1) \; + \; p_1\ln(K-1) \; - \; \ln K \; = \; 0.$$

By the mean value theorem,

$$\begin{aligned} &(H(p_0) + p_0\ln(K-1) - \ln K) - (H(p_1) + p_1\ln(K-1) - \ln K) \\ &= (p_0 - p_1)(H'(p) + \ln(K-1)) \end{aligned}$$

for some $p \in [p_{\min}, p_{\max}]$ with $p_{\min} = \min\{p_0, p_1\}$ and $p_{\max} = \max\{p_0, p_1\}$. Note that $H'(p) = \ln\frac{1-p}{p}$ is decreasing, therefore,

$$\begin{aligned} (t-1)\Big[\, H(p_0) + p_0\ln(K-1) - \ln K\Big] &\geq (t-1)(p_0 - p_1)(H'(p_0) + \ln(K-1)) \\ &= (\frac{t-1}{K} - \lfloor\tfrac{t}{K}\rfloor)\ln\frac{\lfloor\frac{t}{K}\rfloor(K-1)}{t-1-\lfloor\frac{t}{K}\rfloor}. \end{aligned}$$

For the first term, since $\frac{t}{K} - 1 \leq \lfloor\frac{t}{K}\rfloor \leq \frac{t}{K}$, it follows that $\frac{t-1}{K} - 1 \leq \lfloor\frac{t}{K}\rfloor - \frac{1}{K} \leq \frac{t-1}{K}$. Hence

$$\Big|\frac{t-1}{K} - \lfloor\tfrac{t}{K}\rfloor\Big| \leq 1.$$

For the second term, since

$$\frac{t-K}{t} = \frac{(\frac{t}{K}-1)(K-1)}{t-\frac{t}{K}} \leq \frac{\lfloor\frac{t}{K}\rfloor(K-1)}{t-1-\lfloor\frac{t}{K}\rfloor} \leq \frac{\frac{t}{K}(K-1)}{t-1-\frac{t}{K}} = \frac{t(K-1)}{t(K-1)-K},$$

and

$$\frac{t-K}{t} < 1; \;\; \frac{t(K-1)}{t(K-1)-K} > 1,$$

then

$$\Big|\ln\frac{\lfloor\frac{t}{K}\rfloor(K-1)}{t-1-\lfloor\frac{t}{K}\rfloor}\Big| \leq \max\left\{\ln\frac{t}{t-K}, \ln\frac{t(K-1)}{t(K-1)-K}\right\}.$$

Thus,

$$\begin{aligned} F(p_0, K) &\geq -\max\left\{\ln\frac{t}{t-K}, \ln\frac{t(K-1)}{t(K-1)-K}\right\} + \ln(K-1) - \ln K \\ &= \min\left\{\ln\frac{t-K}{t}, \ln\frac{t(K-1)-K}{t(K-1)}\right\} + \ln(K-1) - \ln K \\ &= \min\left\{\ln\frac{(t-K)(K-1)}{tK}, \ln\frac{t(K-1)-K}{tK}\right\}. \end{aligned}$$

We complete the proof of (23).

For (11), a straightforward calculation shows that whenever $t > 2K^2/(K-1)$, both

$$\frac{K^2(t-1)}{12((K-1)t-K)(t-K)} < \frac{1}{2}, \text{ and } \delta(t;K) < \frac{1}{2}.$$

Hence the proof is complete. □