Virtues of Patience in Strategic Queuing Systems

JASON GAITONDE∗, Cornell University
ÉVA TARDOS†, Cornell University

We consider the problem of selfish agents in discrete-time queuing systems, where competitive queues try to get their packets served. In this model, a queue gets to send a packet each step to one of the servers, which will attempt to serve the oldest arriving packet, and unprocessed packets are returned to each queue. We model this as a repeated game where queues compete for the capacity of the servers, but where the state of the game evolves as the length of each queue varies, resulting in a highly dependent random process. In classical work for learning in repeated games, the learners evaluate the outcome of their strategy in each step—in our context, this means that queues estimate their success probability at each server. Earlier work by the authors [in EC’20] shows that with no-regret learners, the system needs twice the capacity as would be required in the coordinated setting to ensure queue lengths remain stable despite the selfish behavior of the queues. In this paper, we demonstrate that this myopic way of evaluating outcomes is suboptimal: if more patient queues choose strategies that selfishly maximize their long-run success rate, stability can be ensured with just \( \frac{e}{e-1} \approx 1.58 \) times extra capacity, strictly better than what is possible assuming the no-regret property.

As these systems induce highly dependent random processes, our analysis draws heavily on techniques from the theory of stochastic processes to establish various game-theoretic properties of these systems. Though these systems are random even under fixed stationary policies by the queues, we show using careful probabilistic arguments that surprisingly, under such fixed policies, these systems have essentially deterministic and explicit asymptotic behavior. We show that the growth rate of a set can be written as the ratio of a submodular and modular function, and use the resulting explicit description to show that the subsets of queues with largest growth rate are closed under union and non-disjoint intersections, which we use in turn to prove the claimed sharp bicriteria result for the equilibria of the resulting system. Our equilibrium analysis relies on a novel deformation argument towards a more analyzable solution that is quite different from classical price of anarchy bounds. While the intermediate points in this deformation will not be Nash, the structure will ensure the relevant constraints and incentives similarly hold to establish monotonicity along this continuous path.

CCS Concepts: • Theory of computation → Quality of equilibria; Algorithmic game theory.

Additional Key Words and Phrases: price of anarchy, queuing systems

ACM Reference Format:
Jason Gaitonde and Éva Tardos. 2021. Virtues of Patience in Strategic Queuing Systems. In Proceedings of the 22nd ACM Conference on Economics and Computation (EC ’21), July 18–23, 2021, Budapest, Hungary. ACM, New York, NY, USA, 21 pages. https://doi.org/10.1145/3465456.3467640

∗Supported by NSF grant CCF-1408673 and AFOSR grant FA9550-19-1-0183. Email: jsg355@cornell.edu.
†Supported in part by NSF grants CCF-1408673, CCF-1563714 and AFOSR grant FA9550-19-1-0183. Email: eva.tardos@cornell.edu.
1 INTRODUCTION

A fundamental aim at the intersection of economics and computer science is to understand the efficiency of systems when the dynamics are governed by the actions of strategic and competitive agents. A large body of work has established bounds on the price of anarchy of various well-studied games, which measures the gap between the social welfare at the worst-case Nash equilibrium with that of the social optimum attainable via coordination by the agents [7]; moreover, recent work has shown that in many cases, price of anarchy bounds seamlessly extend when agents employ simple no-regret learning algorithms in repeated games [2, 9, 13].

However, a critical assumption of these models is that in each round, agents play an “independent” copy of the same game; that is, the past sequence of play does not fundamentally change the nature of the game except through their learning. In some settings, this approximation might be valid, like in modeling routing in the context of the morning rush-hour traffic. In other applications, this assumption clearly does not hold; in modeling packet routing in computer networks, if a packet gets dropped, then this packet will be resent in future rounds and thus increase congestion. Therefore, developing a deeper theory of the efficiency of strategic agents in repeated games that retain state is of large practical importance.

Recent work by the authors [5] initiates the study of price-of-anarchy-style bounds when the repeated games carry state in a discrete-time queuing system. In this model, queues must compete for servers to clear their own packets which arrive at heterogeneous rates, while servers give priority to older packets. This priority scheme induces strong dependencies between rounds because a queue that fails to clear packets will have priority in future rounds. The main result of [5] is that if the queues use no-regret algorithms, then just a factor of 2 higher service rates than what is necessary for centralized stability is sufficient to ensure that this selfish system also remains stable.

In this paper, we continue this line of work towards understanding the price of anarchy in such games. In particular, we show that no-regret behavior can look short-sighted when considering the long-run dependencies at play in this queuing system, as seen in Example 1.1. While no-regret learning is reasonably well-suited to “independent” repeated games, this need not hold when adding state: players can improve their outcome when considering the long-term effect of their behavior. We explore the outcomes of this queuing setting when queues are sufficiently patient in evaluating the results of their strategies and select fixed randomizations over servers optimally conditioned on the policies of the other queues. We connect the game-theoretic incentives of this system with the asymptotic properties of these random dynamics by enforcing that each queue aims to minimize their linear rate of growth under these dynamics. Using a novel deformation argument, we show that just $e^{\frac{1}{e}} < 2$ times extra server capacity suffices to ensure every queue is stable in any Nash equilibrium of this game, strictly better than what is possible under no-regret dynamics.

1.1 Overview of Results and Techniques

In this paper, we consider a discrete-time queuing system where $n$ queues receive packets at heterogeneous rates $\lambda = (\lambda_1, \ldots, \lambda_n) \in (0, 1)^n$. In each round, any queue that has any remaining packets must select exactly one of $m$ servers with heterogeneous success probabilities $\mu = (\mu_1, \ldots, \mu_m) \in [0, 1]^m$ to attempt to clear a single packet. Each server can only succeed in clearing at most one packet in each round, and most importantly, returns each unprocessed packet to the original queue. We assume for simplicity that servers have no buffer. We further assume that servers attempt to serve the oldest packet it receives in each round\(^1\), thus giving priority to queues that have not been able

\(^1\)An alternate choice is to assume that servers select packets uniformly at random among packets it receives in a given round to attempt to serve. However, it is shown in [5] that adding even $\text{poly}(n)$ slack to the central feasibility conditions (given by (1)) cannot guarantee stability in general, and patience by the queues does not help in this case either.
to clear packets efficiently. Queue lengths can grow arbitrarily, so the efficiency we consider is under what conditions on the service and arrival rates can it be ensured that the system remain stable? See Section 2.1 for precise definitions and the full specification.

The main result of this paper is to show that with patience by the queues, a factor of $\frac{e}{e-1} \approx 1.58$ extra server capacity over what is needed in the centralized setting suffices to guarantee the stability of the system despite selfishness. This is in contrast to the main result of [5], where it is shown that if queues use no-regret learning, the system needs a factor of 2 extra server capacity. In this setting, the no-regret property implies that queues send to queues with the highest empirical success rates, without accounting for how alternate choices affect these rates. The analysis there leverages the no-regret property along with concentration and a potential function argument to ensure that older queues use the most efficient servers sufficiently many times on long enough windows. The resulting factor 2 arises in a natural way from these considerations and can be shown to be essentially unimprovable just assuming the no-regret property. While this result provides a constant factor bicriteria result, Example 1.1 illustrates the value of patience in our model.

Example 1.1. Suppose there are two queues with arrival rates $\lambda_1 = \lambda_2 = .51$ and two servers with $\mu_1 = 1$ and $\mu_2 = .49$. In this case, each queue receives a new packet roughly once every two periods on average. One can show that if both servers send to the top rate server every period, the sequence of play will satisfy the no-regret property, as they roughly split the top server equally. Each server then roughly clears at a rate of $1/2$ which is strictly better than deviating to the lower server, but this system will not be stable; packets arrive at a total rate of 1.02 while are cleared at a rate of 1 in expectation, leading to linear growth. In this example, there is a unique fixed no-regret policy for each agent, given the behavior of the other agent, but either agent would be better off in the long-term by slightly deviating to the inferior server even if the other stayed the same. In the classical setting with “independent” repeated games, this cannot occur.

If queues are viewed as sufficiently patient and could choose fixed strategies that optimize long-run stability by sometimes sending to suboptimal servers, they could experience better long-run stability. In the above example, this happens because if one queue starts mixing slightly on the slow server, then the other queue clears faster as she faces less competition on the high rate server. Although this first queue sends to the better server less often, she will tend to have priority more often when she does so, thus actually increasing the long-run effectiveness of this server for this agent. These effects cause both servers to simultaneously clear. Another way to phrase it is that no-regret is myopic in evaluating sending to a server: sending to the inferior server will necessarily induce regret without accounting for how this behavior will change their success rate on the superior server. Our main result is that with this more patient form of evaluation, the system requires less capacity to ensure stability than when queues use no-regret learning.

1.1.1 Patient Selfishness. While vanilla no-regret learning seemed to be the “correct” notion to study repeated games without carryover, examples like this suggest that perhaps this is not so when outcomes from previous rounds of the game directly affect the nature of the games in future iterations. Towards furthering our understanding of the price of anarchy in stateful repeated games, we formulate and study a patient version of this model where queues optimize over the long-term effect of fixed randomized strategies over servers. Each queue $i$’s choice over servers can be described by a fixed vector $p_i \in \Delta^{m-1}$, where $\Delta^{m-1}$ is the probability simplex over the $m$ servers. We study this as a traditional game and consider the resulting Nash equilibria when each queue aims to choose their fixed randomization to minimize their long-run aging rate (equivalently, their long-run growth rate, see Section 2.1) conditioned on the others. Our main interest is understanding
under what conditions on the service and arrival rates will the system remain stable in *every Nash equilibrium*?

To study this, we face significant probabilistic and game-theoretic challenges: probabilistic challenges to establish the closed form of asymptotic growth rates for given strategies, and game-theoretic challenges in bounding the quality of the resulting Nash equilibria of this game. The techniques we use will prove useful in addressing these conceptually distinct difficulties, thereby unifying the game-theoretic and probabilistic properties of our systems.

1.1.2 Asymptotic Growth Rates. In the above discussion, we stated that each queue aims to select a fixed randomization over servers to minimize their long-run aging rate in this system given the randomizations of the others. Our first task, to do any game-theoretic analysis of this system, is to analyze the long-run properties of this random process of queue ages (which typically will not even be recurrent). A major technical component of our work is showing that for any fixed, independent randomizations $p$ by the queues over servers, not only do these long-run growth rates exist almost surely, they are *deterministic* and can be explicitly computed as a function of the strategies:

**Theorem 1.2 (Theorem 4.1, informal).** There exists an explicit, continuous function $r : (Δ^{m−1})^n → \mathbb{R}_{≥0}^n$ such that, if queues independently randomize over servers according to $p ∈ (Δ^{m−1})^n$, then the (random) long-run growth rate of each queue $i$ is $r_i(p)$ almost surely.

To prove this result, we provide an alternate, *algorithmic* description of the long-run rates in Section 3.1. Working just with this alternative definition, we show that the queues partition into groups such that all queues in a group age asymptotically at the same rate. We then return to the task of establishing that the true, long-run asymptotic aging rates of the queues for any choice of strategies coincides with the output of the algorithm. To do this, we repeatedly appeal to concentration bounds to show that each subset in the partition grows at the desired rate; as the priority structure changes rapidly round-to-round, we do so via a delicate argument that accounts for these changes. We then carefully apply the Borel-Cantelli lemma to establish the result. After more formally defining the various parameters of our queuing process, we prove this result in Section 4, but defer some of the highly nontrivial and technical details to the appendix.

1.1.3 Game-Theoretic Properties: Equilibria and Price of Anarchy. Once we show that these limits almost surely are equal to an explicit, deterministic function of $p$, it might still not be the case that a Nash equilibrium exists in the induced game. However, we show that the cost function exhibits significant analytic properties which lets us reason about the structure of the sets that arise in the partition for any fixed strategy profile. Concretely, we show that each level set of the cost function corresponds to the minimizing subset of the ratio of a submodular and modular set function; this significant structure allows us to show that the subsets that minimize this ratio are closed under union and non-empty intersection. In particular, these considerations will be enough to show continuity as a function of the strategies (Theorem 3.4) which along with other properties will enable us to show that an equilibrium exists via Kakutani’s Theorem (Theorem 3.5). While we show that the cost function of our game has significant structure, the correspondence between actions (randomizations) and costs is quite nonlinear, imposing new technical challenges.

Recall that our goal is to ensure stability in any Nash equilibrium, assuming some relationship on the service rates to the arrival rates. Our main result shows that the correct constant of system slack is $\frac{e}{e−1} ≈ 1.58$, beating the best achievable constant of 2 in the no-regret setting of [5]:

**Theorem 1.3 (Main, Corollary 5.2, informal).** If the service capacity is large enough so that the system would remain feasible when centrally managed even if capacities are scaled down by $\frac{e}{e−1}$, then in every equilibrium of this game, all queues are stable.
This result is tight: in the symmetric system where $m = n$, each queue has the same arrival rate, each server has the same success rate, and each queue chooses to uniformly randomize over servers, a simple balls-in-bins analysis yields this constant as $m,n \to \infty$.

To prove this theorem, we provide a novel argument that establishes the result by continuously deforming any Nash profile towards a carefully constructed strategy profile, while only monotonically decreasing the rate at which the top group clears. We then analyze the resulting profile to give a lower bound on the value of the Nash profile. The key difficulty is that the relevant incentives for each queue correspond to possibly many different subsets of queues that have maximal aging rates, subsets that collectively clear packets at the lowest rate relative to the rate they receive them. These constraints are difficult to directly compare; different choices of deviations in the strategy by a queue at any Nash equilibrium may violate distinct constraints, making it unclear how to argue about the quality of these equilibria. In particular, there does not seem to be a direct analogue of the Nash indifference principle in finite-action games where utilities are affine in the randomizations of each agent (recall Example 1.1, where the queue moving to the lesser server will still appear to prefer the better server).

To overcome these difficulties, we show that one can significantly reduce the number of incentive constraints one must consider for each queue (Proposition 5.3). We can carefully perform our deformation of the collective strategy vector of the queues according to the structure of these sparsified constraints, and show that our deformation only hurts the quality of the Nash solution to provide a valid lower bound. We elaborate on these difficulties and prove Theorem 1.3 in Section 5.

In contrast, almost every known price-of-anarchy-style result can be viewed via the very general smoothness framework of Roughgarden [13], which connects an equilibrium with the social optimum via discrete changes in the strategy profile. Our argument instead relies on a careful equilibrium analysis that smoothly interpolates between the equilibrium and a “good” profile which is easy to explicitly bound; however, during these deformations, these intermediate strategy profiles will not be equilibria. To prove the monotonicity of this deformation, we connect the incentives at Nash to the structure of the subset of maximizers of the long-run rate function, and show that the Nash constraints still hold in the directions we deform.

The organization of this paper is as follows: after formalizing the strategic queuing model in Section 2.1 and the relevant game in Section 2.2, we describe how to compute the resulting aging rates for each fixed strategy $p$ by the queues in Section 3.1. Roughly speaking, given any fixed $p$, the set of queues partitions into subsets that all age at equal asymptotic rates determined by fractional bottlenecks in the system. Using this algorithmic description, we then turn to establishing analytic properties in Section 3.2; from these properties, we can deduce the existence of pure equilibria in this queuing game in Section 3.3. We then prove that the algorithmic description of long-run aging rates given in the previous sections indeed holds with probability one in Section 4; as the details are rather involved, however, we defer some of the more technical auxiliary claims to the appendix. Finally, we establish our tight bound on the price of anarchy of $\frac{e}{e-1}$ in Section 5.

1.2 Related Work

Our work falls in a long tradition of establishing price of anarchy bounds for various games [7], which roughly quantifies the difference in the performance of a competitive and selfish system with the social optimum that can be achieved through explicit coordination. Our work is most closely related to the aforementioned prior work by the authors [5]. However, our price of anarchy analysis shows that the no-regret condition alone is too myopic when facing a game with carry-over state—our techniques and analysis are completely different as well. While our stability objective differs from usual objectives in this literature, our results qualitatively also resemble the bicriteria result of Roughgarden and Tardos [14], which shows that in nonatomic routing, the cost incurred at
We will later repeatedly use the following simple fact:

We study the strategic queuing model introduced in previous work by the authors [5], which is a smoothness framework [13], our argument is an equilibrium analysis that is more similar to Johari and Tsitsiklis [6], who establish equilibrium conditions and modify their problem while maintaining the equilibrium condition to arrive at a version that is easy to analyze. In our argument, we also modify the equilibrium itself towards a more tractable solution, but the intermediate points in this deformation will not be Nash, requiring additional arguments.

While the goal of our work is in establishing price-of-anarchy-style bounds in dependent systems, this necessitates a careful understanding of the analytic properties of our random queuing dynamics. In the probability literature, the long-run aging rates that form the incentives are known as (linear) escape rates of random walks [10]. A rich theory has emerged to study this in special networks, but it is unclear how to apply these techniques in our setting. Our work relies on careful, self-contained estimates levering concentration, coupling, and supermartingale arguments.

Our queuing setting bears resemblance to stochastic games, a generalization of Markov decision processes where multiple players competitively and jointly control the actions and transitions (see, for instance [4, 11]). However, our work differs from this line in multiple ways: in our model, queues are unaware of the system state and parameters, and most importantly, we are interested in explicit bounds to derive price-of-anarchy-style results for stability.

2 PRELIMINARIES

Notation. We use the following fractional sum operation $\oplus : \mathbb{R}^2 \times \mathbb{R}_{\geq 0}^2 \rightarrow \mathbb{R}_{\geq 0}$:

$$\frac{a}{b} \oplus \frac{c}{d} = \frac{a+c}{b+d}.$$  

We will later repeatedly use the following simple fact:

**Fact 2.1.** For all $a_1, \ldots, a_n \geq 0$ and $b_1, \ldots, b_n > 0$,

$$\min_{i \in [n]} \frac{a_i}{b_i} \leq \frac{a_1}{b_1} \oplus \ldots \oplus \frac{a_n}{b_n} = \frac{\sum_{i=1}^n a_i}{\sum_{i=1}^n b_i} \leq \max_{i \in [n]} \frac{a_i}{b_i}.$$  

Moreover, equality holds in either of the inequalities if and only if both inequalities are tight.

Given a $n \times m$-dimensional vector $p = (p_1, \ldots, p_n)$, where $p_i \in \mathbb{R}^m$, we will write $p_{ij}$ for the $j$th element of $p_i$. Given a vector $x \in \mathbb{R}^n$ and a subset $I \subseteq [n]$, we write $x_I$ to denote the vector restricted to the components in $I$. As is standard, we say “almost sure” to mean with probability one. Given a set $S$, we will write $P(S)$ to denote the power set. We write $\text{Bern}(\lambda)$ to denote a Bernoulli random variable that is 1 with probability $\lambda$ and 0 with probability $1 - \lambda$. We write $\text{Geom}(\lambda)$ to denote a geometric random variable with parameter $\lambda$.

2.1 Queuing Model

We study the strategic queuing model introduced in previous work by the authors [5], which is a competitive version of a queuing model considered by Krishnasamy et al [8]. As described above, queues receive packets with some fixed probability, and must select a server with heterogeneous success rates to send their oldest packet to. Each server chooses only the oldest packet it receives to attempt to clear, and returns each unprocessed packet to the original queue (as well as the packet it attempted to clear if it fails). Each queue receives only bandit feedback of whether their packet was cleared or not. In this work, we instead work with an equivalent, deferred-decisions version of this model that keeps track only of the oldest packet at each queue:

1. Time progresses in discrete steps $t = 0, 1, \ldots$. At each time $t$, $T^i_t$ is the timestamp of the oldest unprocessed packet of queue $i$ at time $t$. $T^i_t = \max\{0, t - T^i_t\}$ is the age of the current
oldest packet of queue $i$ at time step $t$. In particular, $T^i_t$ measures how old the current oldest unprocessed packet for queue $i$ is.  

(2) Queue $i$ can send a packet to any server $j$ in this time step if $t - T^i_t \geq 0$. Each server $j$ attempts to serve only the oldest packet it receives, and succeeds with probability $\mu_j$. If queue $i$’s packet is successfully served, set $T^i_{t+1} = T^i_t + X^i$, where $X^i \sim \text{Geom}(\lambda_i)$ is independent of all past events, and otherwise $T^i_{t+1} = T^i_t$.

See [5] for a more formal explanation of the equivalence: it follows because the gap between a sequence of Bern($\lambda$) trials follows a Geom($\lambda$) distribution.

**Definition 2.1.** The queuing system under some dynamics is **stable** if for each $i \in [n]$, $T^i_t/t \to 0$ almost surely. The system is strongly stable if, for any $r \geq 0$ and any $i \in [n]$, the random process $\{T^i_t\}_{t=0}^{\infty}$ satisfies $\mathbb{E}[\{T^i_t\}^r] \leq C_r$ where $C_r$ is a fixed constant depending only on $r$, not on $t$.

It is shown in [5] that strong stability is indeed stronger than stability, for here it implies that almost surely $T^i_t = o(t^c)$ for every $c > 0$.

To set a baseline measure of when stability is possible under any coordinated policy, it is shown in [5] that a simple criteria relating $\lambda$ and $\mu$ is both necessary and sufficient.

**Theorem 2.2 (Theorem 2.1 of [5]).** Suppose that $1 > \lambda_1 \geq \ldots \geq \lambda_n > 0$ and $1 \geq \mu_1 \geq \ldots \geq \mu_m \geq 0$. Then the above queuing system is strongly stable for some centralized (coordinated) scheduling policy if and only if for all $1 \leq k \leq n$,

$$\sum_{j=1}^{k} \mu_j > \sum_{i=1}^{k} \lambda_i. \quad (1)$$

When (1) holds, we say that the queuing system is (centrally) feasible.

Until now, we have mostly left the manner in which queues choose servers unspecified. One natural model is that each queue uses a standard no-regret learning algorithm that learns from their previous history of successes at each server to make a (randomized) choice of server in each subsequent round. In [5], it is shown that if the queues each use sufficiently good no-regret algorithms, then under mild technical restrictions, the system remains stable with just an extra factor of 2 on the right side of (1):

**Theorem 2.3 (Theorem 3.1 of [5], informal).** Suppose each queue uses sufficiently good no-regret learning algorithms, and that for all $1 \leq k \leq n$,

$$\sum_{j=1}^{k} \mu_j > 2 \sum_{i=1}^{k} \lambda_i. \quad (2)$$

Then the random process $T^i_t$ under these dynamics is strongly stable.

### 2.2 Patient Queuing Systems

We now formally define the patient queuing game that is the focus of this work. To begin, we formulate the game in a manner that is well-defined *a priori*:

---

2While $T^i_t \geq 0$ by definition, it is possible that $\bar{T}^i_t > t$. The interpretation is that the queue has cleared all of her packets at time $t$ and will receive her next one at $t = \bar{T}^i_t$, or equivalently, in $\bar{T}^i_t - t$ steps in the future from the perspective at time $t$.

3The assumption that $\lambda < 1$ is merely to avoid simple edge cases that can be separately handled easily.
We now construct a function that will help establish various game-theoretic properties of this system. In particular, we show that the expected aging rate is equivalent to the cost function defined above is clearly well-defined as the limsup of expected values.

Note that the cost function defined above is closely related to the limsup of expected values. However, we will actually show that the limit of the random quantity \( \frac{T_i}{t} \) (without expectations) is almost surely equal to a deterministic constant depending on \( \mathbf{p}, \lambda, \mu \) (see Theorem 4.1 and the discussion in Section 3). By deriving an alternate, explicit characterization of these values, we show that Nash equilibria exist in Theorem 3.5. See also Remark 4.1 for an alternate equivalent formulation of the game via long-run growth rates.

Our main focus in this work will be to give guarantees on the quality of all Nash equilibria. In a slight abuse of the price of anarchy terminology, we make the following definition:

Definition 2.5. Let \( \lambda \) and \( \mu \) satisfy the conditions of Theorem 2.2, so that \( \mu \) strictly majorizes \( \lambda \). For \( \alpha \geq 1 \), let \( \mathcal{G}(\alpha) = ([n], (c_i)^n_{i=1}, \mu, \alpha^{-1} \lambda) \), and let \( \mathcal{N}(\alpha) \) denote the set of Nash equilibria of \( \mathcal{G}(\alpha) \). The price of anarchy of \( \mathcal{G} = \mathcal{G}(1) \) is defined as the supremum of \( \alpha \) values such that there exists a Nash equilibrium \( \mathbf{p}^* \in \mathcal{N}(\alpha) \) and some \( i \in [n] \) such that \( c_i(\mathbf{p}^*) > 0 \) in \( \mathcal{G}(\alpha) \).

3 LONG-RUN STRATEGIES

In this section, we extensively study the properties of the cost function \( c(\mathbf{p}) \), which is currently written as the limsup of the expected value of the random linear aging rate of each queue. By taking the limsup and expected values, the cost function is well-defined, albeit quite unwieldy at present. Our first task is thus to provide an alternative, algorithmic description of \( c(\mathbf{p}) \), which we initially denote \( r(\mathbf{p}) \) (for "rates") in Section 3.1. We then show that \( r \) has significant analytic structure that will help establish various game-theoretic properties of this system. In particular, we show that the level subsets (in \( [n] \)) of \( r(\mathbf{p}) \) enjoy convenient closure properties, which will be enough to establish continuity and other properties, which will help to prove the existence of Nash equilibria.

We finally return to the proof that this function is actually equal to the cost function \( c \) to Section 4. In fact, we show a significantly stronger result in Theorem 4.1: the limit of the random process \( T_i^t / t \) itself exists and is equal to \( r(\mathbf{p}) \) almost surely. As such, defining the game via the limsup of the expected aging rate is equivalent to defining it via the true linear aging rate.

3.1 Algorithmic Description of Cost Function

We now construct a function \( r : (\Delta^{n-1})^n \rightarrow [0, 1] \) that we later prove is equivalent to \( c \). We will show that for any fixed \( \mathbf{p} \), the set \([n]\) of queues partitions into subsets \( S_1, S_2, \ldots \), where each queue in \( S_l \) group has the same aging rate and \( S_1 \) ages the fastest, then \( S_2 \), etc., according to \( r \).

For intuition about the quantities that will arise before considering the general case, consider the simplest setting of a single queue and a single server with rates \( \lambda > \mu \). Such a system has no

\[ c(\mathbf{p}) = \limsup_{t \to \infty} \mathbb{E} \left[ \frac{T_i^t}{t} \right] \]

where \( T_i^t \) is again the age of queue \( i \) at time \( t \) in the random queuing process induced by running the queuing system with \( \mu \) and \( \lambda \) the system parameters when each queue chooses a server by independently randomizing according to \( \mathbf{p} \) in each time step she has a packet.

We say \( \mathbf{p} \) is a Nash equilibrium of \( \mathcal{G} \) if for all \( i \in [n], p_i = \arg \min_{p' \in \Delta^{n-1}} c_i(p', p_{-i}) \), i.e. each queue chooses \( p_i \) to minimize their cost function conditioned on the strategies \( p_{-i} \) of the other queues.

In words, the price of anarchy of a centrally feasible system is the supremum of values of \( \alpha \) such that, when all queue arrival rates are scaled down by \( \alpha \), there still nonetheless exists a Nash equilibrium and some queue that suffers nonzero linear aging.
competition nor nontrivial strategies. In any round where the queue has an uncleared packet, the age will first increase by 1 deterministically. With probability \( \mu \), the queue will succeed in clearing this packet, and the age will go down in expectation by \( \mathbb{E}[X] = 1/\lambda \), where \( X \sim \text{Geom}(\lambda) \) is independent of whether or not the server succeeds. Therefore, the expected change in this queue’s age will be \( 1 - \mu/\lambda > 0 \), and we expect that the queue will asymptotically age at this rate.

In general, with multiple queues and servers, the actual values of \( c_i \) are best described via a recursive algorithm that computes the rates, which we give below. The intuition is that \( S_1(p) \) will be the subset that minimizes the ratio of expected packets they clear collectively given \( p \), assuming they have priority over all other queues, divided by their sum of arrival rates. This quantity arises by viewing this subset as a single large queue as in the single queue example above. Conditioned on this set \( S_1 \) of queues growing fastest, they will typically have priority, and then we recurse to find the lower groups. The algorithm begins by initializing \( k = 1 \) and \( I = [n] \):

1. Compute the minimum value over all nonempty subsets \( S \subseteq I \) of
   \[
   \frac{\sum_{j=1}^{m} \mu_j (1 - \prod_{i \in S} (1 - p_{i,j}))}{\sum_{i \in S} \lambda_i}.
   \]
   This gives the expected number of packets cleared by \( S \) if all queues in \( S \) send in a time step and they have priority over all other queues, divided by their sum of arrival rates.

2. If this value is at least 1, then no subset of queues will have linear aging, so set \( S_k = I \), \( r_i(p) = 0 \) for all \( i \in S_k \), and terminate. Otherwise, set \( S_k \) to be the minimizer of the previous quantity over all nontrivial subsets of \( I \), chosen to be of largest cardinality in the case of degeneracies.\(^5\) In this case, for each \( i \in S_k \), \( r_i(p) \) gets set to
   \[
   1 - \frac{\sum_{j=1}^{m} \mu_j (1 - \prod_{i \in S_k} (1 - p_{i,j}))}{\sum_{i \in S_k} \lambda_i}.
   \]
   For \( k = 1 \), we refer to any subset with the minimum ratio as a **tight**, or minimizing, subset.

3. Update the server rates \( \mu_j \) as \( \mu_j \leftarrow \mu_j \prod_{i \in S_k} (1 - p_{i,j}) \). That is, \( \mu_j \) gets discounted by the probability a queue from \( S_k \) sends to server \( j \) (assuming all these queues are sending). Update \( I \leftarrow I \setminus S_k \), \( k \leftarrow k + 1 \), and recurse on \( I \) with \( \mu \) and \( p_I \) if nonempty.

As many of these quantities will appear often, we make the following conventions: for any subsets \( S, S' \) such that \( S \subseteq [n] \setminus S' \), define \( \lambda(S) \) as the sum of arrival rates of packets to a set of queues \( S \), and \( \alpha(S|p, \mu, S') \) as the expected number of packets cleared from queues in \( S \) with service rates \( \mu \), if the queues in \( S' \) have priority, \( S \) has priority over all other queues, and all queues in \( S \cup S' \) send packets in the round:

\[
\alpha(S|p, \mu, S') \triangleq \sum_{j=1}^{m} \mu_j \prod_{i \in S'} (1 - p_{i,j}) (1 - \prod_{i \in S} (1 - p_{i,j}))
\]

\[
\lambda(S) \triangleq \sum_{i \in S} \lambda_i,
\]

and then let

\[
f(S|p, \mu, \lambda, S') \triangleq \frac{\sum_{j=1}^{m} \mu_j \prod_{i \in S'} (1 - p_{i,j}) (1 - \prod_{i \in S} (1 - p_{i,j}))}{\sum_{i \in S} \lambda_i} = \frac{\alpha(S|p, \mu, S')}{\lambda(S)},
\]

denote the ratio of expected number of packets cleared by \( S \) when having priority over all members but \( S' \), normalized by the expected number of new packets received in each round by \( S \). Let \( S_k(p, \mu, \lambda) \) be the \( k \)th set output by the above algorithm. When \( p, \mu, \lambda \) are clear from context, we will suppress them. We write \( U_k = \cup_{t=1}^{k} S_t \) as the set of queues in the top \( k \) groups outputted by the algorithm, with \( U_0 = \emptyset \). We will write \( f_k = f(S_k|U_{k-1}) \), and we use \( g_k = \max\{0, 1 - f_k\} \) for the rate

\(^5\)We show in Lemma 3.2 that this choice is unique and canonical.
of the $k$th outputted set, which is equal to $r_i(p, \mu, \lambda)$ for any $i \in S_k(p, \mu, \lambda)$. From the recursive construction,
\[ S_{k+1}(p, \mu, \lambda) = S_1(p[n]\setminus U_k, \mu', \lambda[n]\setminus U_k) \]  
(2)
where $\mu'_j = \mu_j \prod_{i \in U_k(p, \mu, \lambda)} (1 - p_i)$ for all $j \in [m]$. In words, having found $U_k$, $S_{k+1}$ is the largest minimal set among the remaining elements, but where the $\mu$ rates have been reweighed by the probability no element of $U_k$ sends to each server. For the reader’s convenience, we collect these quantities in Table 1 for easy reference. When the values of, or dependencies on, $p, \mu, \lambda$ are clear from context, we omit them.

### 3.2 Properties of Rate Function

We now record basic properties of the output of the algorithm that will be useful in studying the analytic properties and in proving that this algorithm gives the correct asymptotic rates, though deferring their proofs to the appendix. Clearly, for fixed $S$, the function $f(S|T)$ is nonincreasing in $T$ as a set function. We repeatedly use the following fact, which can be seen simply by expanding the definition of $f$:

**FACT 3.1.** Suppose $S, S', T$ are such that $S, S' \subseteq [n] \setminus T$ and are disjoint. Writing $f$ in the form of the quotient $\alpha/\lambda$, then
\[ f(S \cup S'|T) = f(S|T) \oplus f(S'|S \cup T). \]

Throughout this paper, we will view $f$ as the quotient $\alpha/\lambda$ when invoking Fact 3.1.

Next, we characterize some structure in the minimizing subsets at each step of the algorithm, which will allow us to choose the $S_k$ canonically as the largest cardinality maximizer. To do this, we first show that the function $\alpha(\cdot)$ is submodular [15]:

**LEMMA 3.1 (SUBMODULAR).** For fixed $S'$, $p, \mu, \lambda$ the function $\alpha(S|p, \mu, \lambda, S')$ is submodular is $S$, i.e. for any $S, T \subseteq [n] \setminus S'$, $\alpha(S \cap T|S') + \alpha(S \cup T|S') \leq \alpha(S|S') + \alpha(T|S')$.

Now, recall that the relevant functions in the construction of the above algorithm is the set function $f = \alpha/\lambda$. As a consequence of the fact that this function is the ratio of a submodular function with a modular function, we will be able to gain significant closure properties of the tight subsets (as defined above), which will end up being critical in establishing both game-theoretic and probabilistic properties of our systems.\footnote{In an unrelated context, similar ideas were used by Benjamini, et al. to show that the edge-isoperimetric ratio is not attained in certain infinite networks $[1, 10]$.}

**LEMMA 3.2 (CLOSURE).** For each fixed $p$ and $k \geq 1$, the set of minimizers of $f(\cdot|U_{k-1})$ in $\mathcal{P}([n]\setminus U_{k-1})$ are closed under union and non-disjoint intersection; that is, if $S, S' \subseteq [n]\setminus U_{k-1}$ are minimizers, then so is $S \cup S'$, as well as $S \cap S'$ if nonempty. Moreover, if $S \cap S'$ is empty, then the queues in $S$ and $S'$ must send to disjoint subsets of servers.\footnote{While not necessary for our results, one can use the proof of Lemma 3.2 to show that the maximizing subsets can be computed in strongly polynomial time using the fact that the function $\alpha(\cdot) - y \cdot \lambda(\cdot)$ is submodular.}

In particular, the minimizing set with largest cardinality is unique, and is the union of all minimizing sets at step $k$. If $S$ is considered at step $k$ of the algorithm, but $S$ is not a subset of $S_k$, then $f(S|U_{k-1}) > f(S_k|U_{k-1})$.

From Lemma 3.2, it nearly immediately follows that the outputted rates are strictly monotonic decreasing in the groups: as mentioned, $[n] = S_1 \cup S_2 \cup \ldots$ is meant to give a partition into groups that age together, where $S_1$ is the fastest aging group, $S_2$ the next fastest, etc. As such, the disjoint subsets iteratively output by the algorithm satisfy the intuition that motivates the construction.
Table 1. Table of notation for quantities appearing in main text.

| Symbol | Formula | Definition |
|--------|---------|------------|
| $\lambda$ | Vector of length $n$ of queue arrival rates in descending order. |
| $\Delta^{m-1}$ | Probability simplex over $m$ element set. |
| $\mu$ | Vector of length $m$ of server success rates in descending order. |
| $p$ | Vector of queue randomizations over servers in $(\Delta^{m-1})^n$. |
| $\bar{T}_t^i$ | Timestamp of oldest packet at queue $i$ at time step $t$. |
| $T_i^t$ | Age of queue $i$ at time $t$. |
| $\alpha(S|p, \mu, S')$ | Expected number of packets cleared by queues in $S$ if all have packets in a round and have priority over all queues except for those in $S'$ and each such queue also has packets in the round. |
| $\lambda(S)$ | Sum of arrival rates of queues in $S$. |
| $f(S|p, \mu, \lambda, S')$ | Ratio of expected packets cleared by $S$ with priority over all queues except $S'$ to total arrival rate of $S$. |
| $S_k(p, \mu, \lambda)$ | $k$th subset output in the algorithm of Section 3.2. |
| $U_k(p, \mu, \lambda)$ | Union of first $k$ outputted subsets in the algorithm of Section 3.2. |
| $r_i(p, \mu, \lambda)$ | Outputted aging rate of queue $i$ in the algorithm of Section 3.2. |
| $f_k(p, \mu, \lambda)$ | Value of $f$ for $S_k$ when $U_{k-1}$ has priority. |
| $g_k(p, \mu, \lambda)$ | Outputted rate for $S_k$; equivalently, value of $r_i$ for any $i \in S_k$. |

**Lemma 3.3 (Monotonicity).** Let $S_1, S_2, \ldots$ be the outputs of the algorithm in order. Then $g_k > g_{k+1}$ for each $k \geq 1$.

With these two basic properties, we can obtain an important structural result that will prove fruitful in establishing the existence of equilibria in the next section:
With these structural results, we can turn to showing our first game-theoretic property of this game, for now assuming that the costs are given by $r$, the output of the algorithm of Section 3.1: namely, that equilibria exist. While the cost functions are not quite convex, by restricting each component to a line that varies only a single queue’s strategy, one can deduce enough structure that allows for an application of Kakutani’s Theorem. We record this result here, while deferring the proof to the appendix:

**Theorem 3.5 (Existence of Nash Equilibria).** There exists a pure equilibrium of the game with costs given by $r : (\Delta^{m-1})^n \to [0, 1]^n$.

### 3.3 Existence of Equilibria

With these structural results, we can turn to showing our first game-theoretic property of this game, for now assuming that the costs are given by $r$, the output of the algorithm of Section 3.1: namely, that equilibria exist. While the cost functions are not quite convex, by restricting each component to a line that varies only a single queue’s strategy, one can deduce enough structure that allows for an application of Kakutani’s Theorem. We record this result here, while deferring the proof to the appendix:

**Theorem 3.5 (Existence of Nash Equilibria).** There exists a pure equilibrium of the game with costs given by $r : (\Delta^{m-1})^n \to [0, 1]^n$.

### 4 ASYMPTOTIC CONVERGENCE TO RATE FUNCTION

Having established game-theoretic properties of the patient queuing game assuming that the costs are given by the function $r$, rather than the asymptotic limiting behavior $c$, we now return to the task of showing that these quantities are equal. Our main technical result asserts a rather strong form of this:

**Theorem 4.1 (Almost Sure Asymptotic Convergence).** Let $G = ([n], (c_i)_{i=1}^n, \mu, \lambda)$ be a one-shot queuing game. For each fixed $\mathbf{p}$ and all $i \in [n]$, almost surely it holds that

$$c_i(\mathbf{p}) = \lim_{t \to \infty} \frac{T_i}{t} = r_i(\mathbf{p}).$$

Before providing the proof, we give an overview of the details: the high-level idea is to show that this identity holds for all $i \in S_1$, then $S_2$, and so on. We first show that the maximum queue age grows at most the desired rate on each long-enough window with high probability (Proposition 4.2). The key insight is that if a subset of much older queues $S$ has priority on a long window of length $w$, the quantity $w \cdot f(S_1) \cdot \lambda(S)$ is a lower bound on the expected number of packets cleared collectively by $S$ on this window by definition of $S_1$. The analysis is fairly straightforward when there is a single old queue but gets considerably more complicated when there are multiple old queues: while we know these queues collectively have priority over all young queues, we must argue about priorities within this subset to bound the growth of the maximum queue age. We use concentration bounds (from [5, 16]) with a careful induction that chains together large windows to obtain a win-win analysis.

Once we have established this upper bound on aging on all queues, we then argue that the average queue in $S_1$ ages at a rate of at least $g_1$ almost surely (Proposition 4.3). Combined with the upper bound, we conclude that because the average queue and oldest queue in $S_1$ ages at the desired rate almost surely, all queues in $S_1$ must age at this rate almost surely. To extend this analysis to lower groups $S_2$, etc, a similar argument shows that the maximum age of any queue not in $S_1$ grows at rate at most $g_2$. Because every queue in $S_1$ grows at rate $g_1 > g_2$, almost surely every queue in $S_1$ will eventually be much older than every queue not in $S_1$, giving priority. We leverage this fact to show that the average queue in $S_2$ must grow by at least $g_2$, and therefore every queue in $S_2$ grows at this rate almost surely. The argument for $S_3, \ldots$ is completely analogous.

We now carry out this high-level plan, though deferring the more technical intermediate results to the appendix. Our main intermediate claim asserts that with high probability, the maximum queue age increases at a rate of at most $(1 - (1 - \epsilon) \cdot f_1)$ on the next $w$ steps for a large enough $w$. In fact, more generally, the following holds:
PROPOSITION 4.2. Fix $\epsilon > 0$. For any integer $a \in \mathbb{N}$, let $w = a \cdot \lceil \frac{6}{\epsilon} \rceil^{n-1}$. Suppose it holds at time $t$ that $\max_{i \in [n]} T^i_t \geq w \cdot f_i$. Then

$$\max_{i \in [n]} T^i_{t+w} - \max_{i \in [n]} T^i_t \leq (1 - (1 - \epsilon) \cdot f_i) \cdot w$$

with probability at least $1 - C_1 \exp(-C_2 a)$, where $C_1, C_2 > 0$ are absolute constants depending only on $n, \epsilon, \lambda, \mu, p$, but not on $a$.

More generally, for each $s \geq 1$, if $\max_{i \in U_{s+1}} T^i_t \geq w \cdot f_s$, then

$$\max_{i \in [n]} T^i_{t+w} - \max_{i \in U_{s+1}} T^i_t \leq (1 - (1 - \epsilon) \cdot f_s) \cdot w$$

with probability at least $1 - C_1 \exp(-C_2 a)$, where $C_1, C_2 > 0$ are absolute constants depending only on $n, \epsilon, \lambda, \mu, p$, but not on $a$.

For Proposition 4.2 to yield anything useful, we will need a corresponding lower bound that asserts roughly that if groups have separated according to what the algorithm asserts, then the aging rate of the average queue in a group grows at the conjectured rate. We show the following result: if we have the conjectured separation between groups $U_{k-1}$ and $S_k$ (i.e. each queue in the former is significantly older than each queue in the latter), then some weighted combination of the queue ages in $S_k$ (whose significance will prove apparent momentarily) must rise significantly.

PROPOSITION 4.3. For any $s \geq 1$ and any fixed $\epsilon > 0$, the following holds: suppose that at time $t$, it holds that

$$\min_{i \in U_s} T^i_t - \max_{i \in U_{s+1}} T^i_t \geq 2 \cdot \frac{w}{\lambda_n}.$$ 

Then with probability $1 - A \exp(-B w)$ where $A, B > 0$ are absolute constants not depending on $w$, we have

$$\sum_{i \in U_{s+1}} \lambda_i T^i_{t+w} - \sum_{i \in U_{s+1}} \lambda_i T^i_t \geq (1 - (1 + \epsilon) f_{s+1}) \cdot w \cdot \left( \sum_{i \in U_{s+1}} \lambda_i \right).$$

Moreover, for any fixed $\epsilon > 0$, with probability at least $1 - A \exp(-B w)$ it holds that

$$\sum_{i \in S_1} \lambda_i T^i_t \geq (1 - (1 + \epsilon) f_1) \cdot w \cdot \left( \sum_{i \in S_1} \lambda_i \right).$$

With these two results in hand, we may return to the proof of Theorem 4.1.

PROOF OF THEOREM 4.1. By the Dominated Convergence Theorem, it suffices to show the second equality. We will show the desired statement holds for each $i \in S_1$, then $S_2$, and so on. We first treat the case that the last outputted group $S_k$ satisfies $g_k = 0$, or equivalently that $f_k \geq 1$. Fix $\epsilon > 0$ and partition time into consecutive windows of size $w_\ell = \ell \cdot \lceil \frac{6}{\epsilon} \rceil^{n-1}$. Let $W_\ell = \sum_{q=1}^{\ell-1} w_q$ be the time period at the beginning of the $\ell$th window, and note that $w_\ell = \Theta(W_\ell^{1/2})$.

Consider the following events for $\ell = 1, 2, \ldots$

$$A_\ell = \left\{ \max_{i \in [n]} T^i_{W_\ell} - \max_{i \in [n]} T^i_{W_\ell-1} \geq (1 - (1 - \epsilon) \cdot f_k) \cdot w_\ell \right\}$$

$$B_\ell = \left\{ \max_{i \in [n]} T^i_{W_\ell} \geq w_\ell \cdot f_k \right\}$$

$$C_\ell = A_\ell \cap B_\ell.$$

Clearly, $\Pr(C_\ell) \leq \Pr(A_\ell|B_\ell)$. But by Proposition 4.2, we know that for some constants $C_1, C_2 > 0$ independent of $\ell$, that

$$\Pr(A_\ell|B_\ell) \leq C_1 \exp(-C_2 \cdot \ell).$$
Therefore, we have that
\[
\sum_{t=1}^{\infty} \Pr(C_t) \leq \sum_{t=1}^{\infty} C_1 \exp(-C_2 \cdot t) < \infty.
\]

The first Borel-Cantelli lemma (Theorem 2.3.1 of [3]) thus implies that almost surely at most finitely many of the \(C_t\) occur, or equivalently, almost surely for all but finitely many of the \(\ell\), either
\[
\max_{i \in [n] \setminus U_{k-1}} T^i_{W_{t+1}} - \max_{i \in [n] \setminus U_{k-1}} T^i_{W_t} \leq (1 - (1 - \epsilon) \cdot f_k) \cdot w_t \quad \text{or} \quad \max_{i \in [n] \setminus U_{k-1}} T^i_{W_t} < w_t \cdot f_k.
\]

Observe that for each of the intervals where the latter holds, the value during the interval is at most \(w_t \cdot f_k + w_{t+1} = O(W^t_{1/2})\). In particular, it is not difficult to see that almost surely \(\max_{i \in [n] \setminus U_{k-1}} T^i_{W_t}\) is either \(o(W_t)\), in which case we are done, or grows by at most a rate of \((1 - (1 - \epsilon) \cdot f_k) \cdot w_t\). Either way, as \(\epsilon > 0\) was arbitrary, we may take \(\epsilon \to 0\) to deduce the desired result that almost surely\(^8\)
\[
\limsup_{t \to \infty} \frac{\max_{i \in [n] \setminus U_{k-1}} T^i_t}{t} = 0 = g_k,
\]
using \(f_k \geq 1\). As ages of queues are nonnegative, the lower bound of 0 is trivial.

Now we turn to the rest of the groups, and we now assume that \(g_k > 0\). We do this inductively. For \(S_1\), fix \(\epsilon > 0\) and define
\[
A_t = \left\{ \sum_{i \in S_1} \lambda_i T^i_t < (1 - (1 + \epsilon) f_1) \cdot t \cdot \left( \sum_{i \in S_1} \lambda_i \right) \right\}.
\]

By Proposition 4.3, we know \(\Pr(A_t) \leq A \exp(-Bt)\) for some constants \(A, B > 0\) independent of \(t\). Therefore, \(\sum_{t=1}^{\infty} \Pr(A_t) < \infty\), from which the Borel-Cantelli lemma implies that almost surely, for all but finitely many \(t\),
\[
\sum_{i \in S_1} \lambda_i T^i_t \geq (1 - (1 + \epsilon) f_1) \cdot t \cdot \left( \sum_{i \in S_1} \lambda_i \right).
\]

Taking \(\epsilon \to 0\), we obtain almost surely
\[
\liminf_{t \to \infty} \frac{\sum_{i \in S_1} \lambda_i T^i_t}{t} \geq g_1 \cdot \left( \sum_{i \in S_1} \lambda_i \right). \tag{3}
\]

Next, note that deterministically, we have from Fact 2.1
\[
\min_{i \in S_1} T^i_t \leq \frac{\sum_{i \in S_1} \lambda_i T^i_t}{\sum_{i \in S_1} \lambda_i} \leq \max_{i \in S_1} T^i_t. \tag{4}
\]

In particular, we deduce that almost surely,
\[
\liminf_{t \to \infty} \frac{\max_{i \in S_1} T^i_t}{t} \geq g_1 > 0. \tag{5}
\]

\(^8\)For any \(\epsilon > 0\), we have directly shown that the statement holds for \(t\) of the form \(t = W_t\) for \(t \geq 1\). For any \(t\) such that \(W_t \leq t < W_{t+1}\), \(T^i_t\) cannot be more than \(w_t\) from the value at \(W_t\) as ages can increase by at most 1 in each period. This implies that on any such intermediate time, the difference in the numerator from the value at \(t = W_t\) is \(O(w_t) = O(t^{1/2}) = o(t)\) and thus vanishes in the limit when divide by \(t\), so the limit sup may be taken over all \(t\), not just the sparsified sequence.
For the upper bound, let the \( w_t \) and \( W_t \) be as before, and now define

\[
A_t = \left\{ \max_{i \in [n]} T_{W_{i+1}}^i - \max_{i \in [n]} T_{W_t}^i > (1 - (1 - \epsilon)) \cdot f_t \cdot w_t \right\}
\]

\[
B_t = \left\{ \max_{i \in [n]} T_{W_t}^i \geq w_t \cdot f_t \right\}
\]

\[
C_t = A_t \cap B_t.
\]

Again, \( \Pr(C_t) \leq \Pr(A_t|B_t) \). By Proposition 4.2, a now routine application of the Borel-Cantelli lemma implies that almost surely, for all but finitely many \( \ell \), either

\[
\max_{i \in [n]} T_{W_{i+1}}^i - \max_{i \in [n]} T_{W_t}^i < (1 - (1 - \epsilon)) \cdot f_t \cdot w_t \quad \text{or} \quad \max_{i \in [n]} T_{W_t}^i < w_t \cdot f_t.
\]

But the latter event cannot happen infinitely often with positive probability, as this would imply \( \max_{i \in [n]} T_{W_t}^i = o(W_t) \) infinitely often with nonzero probability, which violates (5). Therefore, it must be the case that almost surely, for all but finitely many \( \ell \), the former event holds. This implies that almost surely

\[
\limsup_{t \to \infty} \frac{\max_{i \in [n]} T_{W_t}^i}{t} \leq (1 - (1 - \epsilon)) \cdot f_t;
\]

taking \( \epsilon \to 0 \) implies that

\[
\limsup_{t \to \infty} \frac{\max_{i \in [n]} T_{W_t}^i}{t} \leq g_1.
\]

As clearly the left side is an upper bound for the lim sup of only those queues in \( S_1 \), almost surely

\[
\limsup_{t \to \infty} \frac{\max_{i \in S_1} T_{W_t}^i}{t} \leq g_1.
\]

Combining this with (5), we finally deduce that almost surely

\[
\lim_{t \to \infty} \frac{\max_{i \in S_1} T_{W_t}^i}{t} = g_1.
\]

Finally, using (3) and (4), we can also conclude that almost surely

\[
\lim_{t \to \infty} \frac{\min_{i \in S_1} T_{W_t}^i}{t} = g_1.
\]

As \( r_i(p) = g_1 \) for all \( i \in S_1(p) \) by definition of \( g_1 \), this proves the theorem for all queues in \( S_1 \).

We now show how to extend this inductively to higher values of \( k \) with \( g_k > 0 \). Suppose that we have shown for all \( i \in U_{k-1} \) that the desired almost sure limit holds, and now consider \( S_k \). A completely analogous argument using the windows \( w_t \) as above with Proposition 4.2 via the Borel-Cantelli lemma implies that almost surely

\[
\limsup_{t \to \infty} \frac{\max_{i \in [n]\setminus U_{k-1}} T_{W_t}^i}{t} \leq g_k.
\]

Now, with these same windows, fix \( \epsilon > 0 \) and let

\[
A_t = \left\{ \sum_{i \in S_k} \lambda_i T_{W_{i+1}}^i - \sum_{i \in S_k} \lambda_i T_{W_t}^i < (1 - (1 + \epsilon)) f_k \cdot w_t \cdot \left( \sum_{i \in S_k} \lambda_i \right) \right\}
\]

\[
B_t = \left\{ \min_{i \in U_{k-1}} T_{W_t}^i - \max_{i \in S_k} T_{W_t}^i \geq 2 \cdot \frac{w_t}{\lambda_n} \right\}
\]

\[
C_t = A_t \cap B_t.
\]
Another completely analogous application of Proposition 4.3 and the Borel-Cantelli lemma implies that almost surely, at most finitely many of the $C_i$ occur. That is, almost surely, for all but finitely many $\ell$, either

$$\sum_{i \in S_k} \lambda_i T^i_{W_{t+1}} - \sum_{i \in S_k} \lambda_i T^i_{W_{t}} \geq (1 - (1 + \epsilon) f_k) \cdot \omega_t \cdot \left( \sum_{i \in S_k} \lambda_i \right) \quad \text{or} \quad \min_{i \in U_{k-1}} T^i_{W_{t}} - \max_{i \in S_k} T^i_{W_{t}} < 2 \cdot \frac{\omega_t}{\lambda_n}.$$  

But the latter event cannot happen infinitely often with any nonzero probability by virtue of the inductive hypothesis and (6), as $g_{k-1} > g_k$ by Lemma 3.3, which implies that these timestamps cannot be so close infinitely often. Therefore, it must be the case that for all but finitely many of $\ell$, the former event holds. As usual, this immediately implies that

$$\liminf_{t \to \infty} \frac{\sum_{i \in S_k} \lambda_i T^i_{t}}{t} \geq (1 - (1 + \epsilon) f_k) \left( \sum_{i \in S_k} \lambda_i \right).$$

Again taking $\epsilon \to 0$ thus implies that almost surely

$$\liminf_{t \to \infty} \frac{\sum_{i \in S_k} \lambda_i T^i_{t}}{t} \geq g_k \left( \sum_{i \in S_k} \lambda_i \right),$$

which again coupled with (6) and Fact 2.1 yields that almost surely

$$\lim_{t \to \infty} \min_{i \in S_k} T^i_{t} = \lim_{t \to \infty} \max_{i \in S_k} T^i_{t} = g_k.$$  

The extension to all $i \in S_k$ follows in the same manner as before by comparing with the average.  

Observe that Theorem 4.1 rather strongly characterizes the linear almost sure asymptotic growth rates of each queue for any choices of randomizations. Our main result in Theorem 5.1 will show that, with a small slack in the system capacity, each queue will be guaranteed sublinear asymptotic growth almost surely in any equilibrium. While the cost function emphasizes the physical interpretation as asymptotic linear growth rates, these incentives impose that queues are indifferent between sublinear growth rates. One could instead define the game just using the $f_k$ quantities directly, rather than taking the max with 0 as is needed to argue about the asymptotic growth rates via $r$. If queues started out equally backed up, the $f_k$ quantities measure the linear speed at which their ages descend to zero. In this setting, we can provide the following stronger conclusion whose proof relies on a supermartingale argument of [12] and is deferred to the appendix.

**Theorem 4.4.** Fix $p$ and suppose that for some group $S_k$ output by the algorithm, $f_k > 1$, so that $1 - f_k < 0$. Then, for each $i \in S_k$, $T^i_t$ is strongly stable.

**Remark 4.1.** While we have formulated the patient queuing game in terms of linear aging rates, all of our results, both probabilistic and game-theoretic, still hold if we had instead done so in terms of linear growth rates with only minor modification. If $Q^i_t$ is the number of uncleared packets of queue $i$ at time $t$, then $\lim_{t \to \infty} Q^i_t / t = \lambda_i \cdot \lim_{t \to \infty} T^i_t / t$ almost surely. This can be deduced as follows: note that $Q^i_t$ is precisely equal to the number of packets queue $i$ received between time $t - T^i_t$ and $t$. Under the natural coupling between Bernoulli and geometric random variables as in [5], the conditional expectation of this, given $T^i_t$, is exactly $\lambda_i \cdot T^i_t$. From standard Chernoff bounds and the Borel-Cantelli lemma, one can show that almost surely, for all $t$, $Q^i_t = \lambda_i \cdot T^i_t + O(\sqrt{\ln t})$, where the implicit constant is random but finite. Dividing by $t$ and taking limits implies the claim. Thus, if cost functions are instead defined via the long-run growth rates, then this differs from the given cost functions by a positive scalar. This preserves the set of Nash equilibria, so all of our game-theoretic results also transfer.
5 PRICE OF ANARCHY

Having established the almost sure asymptotic convergence of this system for any fixed strategies and the existence of equilibria, we finally turn to the game-theoretic task of understanding what condition ensures the stability at any equilibrium profile. By considering deviations by a queue at the Nash equilibrium to a single other server, it is possible to show that the price of anarchy is always at most 2, matching the learning bound of [5]. We show that this factor is actually suboptimal and that \( \frac{e}{e-1} \approx 1.58 \) is the right factor by considering continuous deviations.

The following simple example shows that this is the best possible constant factor: fix \( \epsilon > 0 \) small and suppose there are \( n \) queues and \( n \) servers, with \( \lambda = (1-1/e+\epsilon, \ldots, 1-1/e+\epsilon) \) and \( \mu = (1, \ldots, 1) \), and \( p \) has every queue uniformly mixing among the servers. It is easy to see by symmetry that this system is Nash with \( S_1 = [n] \), for if a queue deviates from this uniform distribution, this does not change the worst ratio. Moreover, for any fixed \( \epsilon > 0 \), as \( n \to \infty \), this system becomes unstable. One can check that

\[
 f(S_1|p) = f([n]|p) = \frac{\sum_{j=1}^{n}(1 - \prod_{i=1}^{n}(1 - 1/n))}{n(1-1/e+\epsilon)} \to \frac{1-1/e}{1-1/e+\epsilon} < 1,
\]

so that \( r(S_1) = \max_i c_i(p) > 0 \). Our main result asserts that this is the worst case, where every queue is maximally colliding subject to being Nash. Concretely, we prove the following instance-dependent bound from which the claimed factor immediately follows:

**Theorem 5.1 (Main).** Let \( p \) be any Nash equilibrium of \( G \), and let \( S_1 \) be as defined before. Then

\[
 f(S_1|p) \geq \min \left\{ 1, \min_{k \leq n} \max_{x \in \mathbb{R}_+^m, \sum_{j=1}^{m} x_j = k} \sum_{j=1}^{m} \mu_j(1 - (1 - x_j/k)^k) \right\}.
\]

**Corollary 5.2.** Let \( p \) be a Nash equilibrium of \( G \), and suppose that for each \( 1 \leq k \leq n \),

\[
 \sum_{j=1}^{k} \mu_j > \left( \frac{e}{e-1} \right) \sum_{i=1}^{k} \lambda_i.
\]

Then every queue is stable at \( p \). In particular, the price of anarchy of the patient queuing game is exactly \( \frac{e}{e-1} \).

**Proof.** In Theorem 5.1, for any \( k \leq n \), we may set \( x_j = 1 \) for \( 1 \leq j \leq k \). Note that \( (1 - x_j/k)^k < e^{-1} \), so if \( \mu \) majorizes \( \lambda \) by a factor of at least \( \frac{e}{e-1} \). Theorem 5.1 implies that \( f(S_1|p) \geq 1 \). From Lemma 3.3 and Theorem 4.1, we conclude that all queues are stable. \( \square \)

We now prepare for the proof of Theorem 5.1. The idea will be to continuously deform the Nash profile towards a highly symmetrized strategy vector while only weakly decreasing \( f(S_1) \). At the end of this process, we obtain a lower bound on this value at Nash. To do this deformation and ensure monotonicity of the growth rate, we must at some point use the Nash property. The difficulty in proving the tightness of this example lies in the form of the \( f \) functions; recall that as \( S_1 \) is the set of all queues growing at the fastest rate as the union of all tight subsets, it can have many proper tight subsets, and each queue \( i \in S_1 \) thus has to locally optimize all of the functions \( f(S|p) \) with \( S \supset i \) simultaneously at Nash (see Figure 1 for an interesting example). In particular, whenever a queue \( i \in S_1 \) is at Nash, one possible deviation may weakly decrease \( f(S) \) for some tight subset \( S \supset i \), while another deviation may be unprofitable because \( f(S') \) weakly decreases for some different tight subset \( S' \supset i \). That is, each queue may be constrained by multiple different

\footnotesize
\[\text{For any fixed } k, \text{ it is not difficult to determine the optimal value of } x \text{ to give the tightest lower bound. It suffices to maximize the numerator, which is concave. By standard KKT conditions at optimality, for all } j, j' \in [m] \text{ such that } x_j > 0, \text{ we must have } \mu_j(1 - x_j/k)^{k-1} = \mu_{j'}(1 - x_{j'}/k)^{k-1}, \text{ and } x_\ell = 0 \text{ for all lower indices.}\]

---

536
Fig. 1. Four queues compete for three servers, with maximally tight sets marked. The two outer queues exclusively send to the outer servers, and the inner queues send to two servers each, as indicated by the figure. One can verify that this profile is Nash and the subsets marked are tight. Consider whether one of the inner queues would want to deviate from the current profile. If she shifts probability to the inner server, the smallest tight set will rise in aging rate. However, if she instead attempts to deviate to one of the outer servers, the rate of all four queues will rise.

objective functions at Nash, making it difficult to generically argue about why any given deviation decreases performance. We overcome this barrier by connecting the incentives for each queue in $S_1$ with the structure guaranteed by Lemma 3.2. More concretely, we reduce the number of constraints we must consider for each queue by showing in Proposition 5.3 that there exists a much smaller, ordered set of tight subsets that completely characterizes the incentives of any queue (whose proof is deferred to the appendix):

**Proposition 5.3.** Let $p$ be any arbitrary strategy vector by the queues, and without loss of generality, let $[k]$ be the maximal tight subset after relabeling. Then, for some $s > 0$, there exists a level partition of $[k]$ into $s$ levels with the following property: if a queue $i \in [k]$ belongs to a level-$\ell$ subset, then for any deviation by $i$ that shifts probability mass from one server to another and does not increase $f(S)$ for some tight subset $S \ni i$, there exists a tight subset $S'$ containing all queues at all levels $j \leq \ell$ such that $f(S')$ must not increase.

With this result, we may finally return to the proof of Theorem 5.1.

**Proof of Theorem 5.1.** Let $p$ be any Nash equilibrium, and suppose that $S_1$ is the maximal tight subset. If $f(S_1) \geq 1$, then we are done, so suppose that $f(S_1) < 1$. The Nash assumption then implies that any deviation by a queue in $S_1$ cannot decrease the value of each tight subset it is part of; note that we need the $f(S_1) < 1$ assumption, as incentives are about the rates, and when $f(S_1) \geq 1$, the rate may remain 0 even if $f$ decreases.

For convenience, reindex and relabel so that $|S_1| = k$ and $S_1 = [k]$. Fix any $x \in \mathbb{R}_{\geq 0}^m$ such that $\sum_{i=1}^m x_i = k$. It suffices to show that

$$f([k]|p) \geq \frac{\sum_{j=1}^m \mu_j (1 - (1 - x_j/k)^k)}{\sum_{j=1}^k \lambda_j}.$$

From now on, we omit the dependence on $p$ in $f$ unless explicitly needed.

Consider the level partition of $[k]$ guaranteed by Proposition 5.3 and suppose that there are $s$ levels. We continuously deform the Nash solution while monotonically decreasing $f([k])$, so that at the end of this process, we have a lower bound on the value of Nash. Given any strategy profile $p$, we say a server $j$ is oversaturated if $\sum_{i \in [k]} p_{ij} > x_j$ and undersaturated if $\sum_{i \in [k]} p_{ij} < x_j$. We will continuously move probability mass from the queues from oversaturated to undersaturated
servers. If no server is oversaturated, we will be done; notice that if a server is oversaturated, an easy averaging argument implies some server must be undersaturated.

Suppose that there exists some oversaturated server. Let \( i \in [k] \) be a queue at level-\( s \), the top level. If \( i \) nontrivially sends to an oversaturated server \( j \) so that \( p_{ij} > 0 \), we continuously decrease \( p_{ij} \) and increase \( p_{ij'} \) for some undersaturated server \( j' \), until either \( j \) stops being oversaturated, \( j' \) stops being undersaturated, or \( p_{ij} \) hits zero. We claim that this deformation cannot increase \( f([k]) \).

To see this, observe that because \( p \) is Nash, we know that any deviation by \( i \) from one server it is nontrivially mixing at to another cannot increase \( f(S) \) for all tight subsets \( S \) containing \( i \), hence there must be some \( S \ni i \) such that \( f(S) \) does not increase. But then Proposition 5.3 implies that some subset \( S' \) containing all queues up to level-\( s \) must have \( f(S') \) not increase either. As \( [k] \) is the only such subset, this deformation could not have actually increased \( f([k]) \).

Moreover, we claim that we can do this for all level-\( s \) queues one-by-one without increasing \( f([k]) \). While the intermediate profiles are not Nash, because we only move probability mass from oversaturated to undersaturated servers, each oversaturated queue only has at most the same probability mass as it did at Nash while each undersaturated queue only has additional probability mass compared to what it had at Nash. As we have shown any such deviation by a level-\( s \) server from an oversaturated queue to an undersaturated queue at Nash cannot increase \( f([k]) \), and now deviations are only worse at this intermediate stage while deforming the level-\( k \) queue strategies, each such deformation still cannot increase \( f([k]) \). Therefore, we can continuously shift all probability mass from level-\( s \) queues at oversaturated servers to undersaturated servers while never increasing \( f([k]) \).

Suppose we have now done this for all levels at least \( \ell + 1 \) for some \( \ell < s \) while not increasing \( f([k]) \), and we want to continue this process at level-\( \ell \). Let \( p' \) be this intermediate strategy vector, where we note that for any queue \( i \) below level \( \ell + 1 \), \( p'_{ij} = p_i \). Again, if no server is oversaturated, we are done. Otherwise, suppose some queue \( i \) at level-\( \ell \) still sends to an oversaturated server \( j \), and we again try to decrease \( p_{ij} \) and increase \( p_{ij'} \) for some undersaturated server \( j' \) as before until the same stopping criterion. We must show that this too cannot increase \( f([k]) \).

Suppose otherwise that it did indeed increase \( f([k]) \) with respect to \( p' \). For a contradiction, it suffices to show that this implies that this same deviation, with respect to the original Nash solution \( p \), must have increased \( f(S) \) for every subset \( S \) containing all queues up to level-\( \ell \). This is sufficient to obtain a contradiction as then Proposition 5.3 implies that every tight subset containing \( i \) improves at Nash with respect to this deviation, which violates the Nash property.

To prove this claim, let \( S \subseteq [k] \) be an arbitrary tight subset at Nash containing all queues up to level-\( \ell \). Because we assume that this deviation improves \( f([k]) \) with respect to \( p' \), it follows from taking partial derivatives that

\[
\mu_j \prod_{r \in [k] \setminus \{i\}} (1 - p'_{rj}) < \mu'_j \prod_{r \in [k] \setminus \{i\}} (1 - p'_{rj'}). \]

However, note that at \( p' \), as \( j \) is still oversaturated, \( p'_{rj} = 0 \) for all queues \( r \) that are at strictly higher levels. As all queues at level-\( \ell \) and below have \( p'_{ij} = p_i \) and \( S \) contains all such queues, this inequality implies

\[
\mu_j \prod_{r \in S \setminus \{i\}} (1 - p_{rj}) < \mu'_j \prod_{r \in S \setminus \{i\}} (1 - p'_{rj'}). \]

Moreover, as \( j' \) is undersaturated, we must have \( p'_{rj'} \geq p_{rj'} \) for all \( r \in [k] \) from the construction of this process, and removing terms in the product only increases the right side. Therefore, we deduce that

\[
\mu_j \prod_{r \in S \setminus \{i\}} (1 - p_{rj}) < \mu'_j \prod_{r \in S \setminus \{i\}} (1 - p_{rj'}). \]
This implies that this deviation also increases \( f(S) \) at Nash. As \( S \) was an arbitrary tight subset containing all queues up to level-\( \ell \), the claim is proved. By the reduction described above, this is a contradiction, and therefore \( f([k]) \) must further decrease with respect to \( \mathbf{p}' \). The argument extends analogously at all intermediate points of this process at level-\( \ell \) by the same reasoning as before.

Therefore, by induction, it follows that we may continuously deform probability mass from oversaturated servers to undersaturated servers while only decreasing \( f([k]) \). At the end of this process, there cannot be any oversaturated servers, otherwise the process could have continued. In particular, if \( \mathbf{p}'' \) is the final probability vector at the end of this process, we have shown that \( \sum_{i \in [k]} p_{ij}'' = x_j \) for all servers \( j \) and that \( f([k])|\mathbf{p}'' \leq f([k])|\mathbf{p} \). We have

\[
  f([k])|\mathbf{p} = \frac{\sum_{j=1}^m \mu_j (1 - \prod_{i=1}^k (1 - p_{ij}))}{\sum_{i=1}^k \lambda_i} \quad \text{(by definition)}
\]

\[
  \geq \frac{\sum_{j=1}^m \mu_j (1 - \prod_{i=1}^k (1 - p_{ij}''))}{\sum_{i=1}^k \lambda_i} \quad \text{(by construction)}
\]

\[
  \geq \frac{\sum_{j=1}^m \mu_j (1 - (1 - x_j/k)^k)}{\sum_{i=1}^k \lambda_i} \quad \text{(as symmetric profile maximizes product)}.
\]

The second inequality holds because given \( \sum_{i \in [k]} p_{ij}'' \), the maximizer of \( \prod_{i \in [k]} (1 - p_{ij}'') \) is attained when each term is equal. As \( x \) was arbitrary, we may take the maximum of the right side over all \( x \) satisfying the constraints, and then the minimum over \( k \). As \( \mathbf{p} \) was an arbitrary Nash profile, this concludes the proof.

\[\square\]

6 DISCUSSION

In this paper, we have studied a patient version of the queuing system of [5]; using careful probabilistic arguments to establish the incentive structure of the game, along with exploiting the analytic structure of the long-run rates of the induced Markov chain, we show that the correct bicriteria factor in this setting is \( e - 1 \) via a novel deformation argument, strictly better than the factor of 2 obtained in the no-regret learning setting. While the current result is not explicitly a learning result, this gap nevertheless suggests that no-regret behavior is not necessarily the correct notion of agent behavior in repeated games that carry strong interdependencies between rounds as in the priority structure here. It is an interesting question as to whether a natural form of learning can arrive at a Nash equilibrium of the patient version or at least result in stable outcomes without reaching an equilibrium.

Moreover, though our restriction to time-independent policies exhibits quite rich behavior while enabling us to completely characterize the game-theoretic properties, perhaps there is a larger space of strategies where similar results hold. To that end, it may be necessary to explore the theoretical properties of more powerful learning algorithms in such settings that get the best of both worlds, namely balancing current rewards while maintaining long-run perspective. Whether such results are possible is an exciting open direction in resolving some of the deficiencies of traditional price of anarchy results; we leave a more systematic investigation of this direction to future work.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their insightful comments and suggestions.

REFERENCES

[1] Itai Benjamini, Russell Lyons, Yuval Peres, and Oded Schramm. 1999. Group-invariant percolation on graphs. Geometric & Functional Analysis GAFA 9, 1 (1999), 29–66.
[2] Avrim Blum, MohammadTaghi Hajiaghayi, Katrina Ligett, and Aaron Roth. 2008. Regret minimization and the price of total anarchy. In Proceedings of the 40th Annual ACM Symposium on Theory of Computing. ACM, 373–382. https://doi.org/10.1145/1374376.1374430

[3] Rick Durrett. 2019. Probability: Theory and Examples. Vol. 49. Cambridge University Press.

[4] Jerzy Filar and Koos Vrieze. 2012. Competitive Markov Decision Processes. Springer Science & Business Media.

[5] Jason Gaitonde and Éva Tardos. 2020. Stability and Learning in Strategic Queuing Systems. In EC ’20: The 21st ACM Conference on Economics and Computation. ACM, 319–347. https://doi.org/10.1145/3391403.3399491

[6] Ramesh Johari and John N. Tsitsiklis. 2004. Efficiency Loss in a Network Resource Allocation Game. Math. Oper. Res. 29, 3 (2004), 407–435. https://doi.org/10.1287/moor.1040.0091

[7] Elias Koutsoupias and Christos H. Papadimitriou. 1999. Worst-case Equilibria. In STACS 99, 16th Annual Symposium on Theoretical Aspects of Computer Science. Springer, 404–413. https://doi.org/10.1007/3-540-49116-3_38

[8] Subhashini Krishnasamy, Rajat Sen, Ramesh Johari, and Sanjay Shakkottai. 2016. Regret of Queueing Bandits. In Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain. 1669–1677. http://papers.nips.cc/paper/6370-regret-of-queueing-bandits

[9] Thodoris Lykouris, Vasilis Syrgkanis, and Éva Tardos. 2016. Learning and Efficiency in Games with Dynamic Population. In Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016. SIAM, 120–129. https://doi.org/10.1137/1.9781611974331.ch9

[10] Russell Lyons and Yuval Peres. 2016. Probability on Trees and Networks. Cambridge Series in Statistical and Probabilistic Mathematics, Vol. 42. Cambridge University Press, New York. xv+699 pages. https://doi.org/10.1017/9781316672815

[11] Abraham Neyman and Sylvain Sorin. 2003. Stochastic Games and Applications. Vol. 570. Springer Science & Business Media.

[12] Robin Pemantle and Jeffrey S Rosenthal. 1999. Moment conditions for a sequence with negative drift to be uniformly bounded in $L^r$. Stochastic Processes and their Applications 82, 1 (1999), 143–155.

[13] Tim Roughgarden. 2015. Intrinsic Robustness of the Price of Anarchy. J. ACM 62, 5 (2015), 32:1–32:42. https://doi.org/10.1145/2806683

[14] Tim Roughgarden and Éva Tardos. 2002. How bad is selfish routing? J. ACM 49, 2 (2002), 236–259. https://doi.org/10.1145/506147.506153

[15] Alexander Schrijver. 2003. Combinatorial Optimization: Polyhedra and Efficiency. Vol. 24. Springer Science & Business Media.

[16] Carsten Witt. 2014. Fitness levels with tail bounds for the analysis of randomized search heuristics. Inform. Process. Lett. 114, 1 (2014), 38–41. https://doi.org/10.1016/j.ipl.2013.09.013