# Tail Optimality of the Nudge-M Scheduling Algorithm

Nils Charlet, Benny Van Houdt
Dept. Computer Science, University of Antwerp, Belgium
{nils.charlet,benny.vanhoudt}@uantwerpen.be

## ABSTRACT

Recently it was shown that the response time of First-Come-First-Served (FCFS) scheduling can be stochastically and asymptotically improved upon by the *Nudge* scheduling algorithm in case of light-tailed job size distributions. Such improvements are feasible even when the jobs are partitioned into two types and the scheduler only has information about the type of incoming jobs (but not their size).

In this paper we introduce Nudge-$M$ scheduling, where basically any incoming type-1 job is allowed to pass any type-2 job that is still waiting in the queue given that it arrived as one of the last $M$ jobs. We prove that Nudge-$M$ has an asymptotically optimal response time within a large family of Nudge scheduling algorithms when job sizes are light-tailed. Simple explicit results for the the prefactor of Nudge-$M$ are derived as well as explicit results for the optimal parameter $M$. An expression for the prefactor that only depends on the type-1 and type-2 mean job sizes and the fraction of type-1 jobs is presented in the heavy traffic setting.

## 1. INTRODUCTION

First-Come-First-Served (FCFS) scheduling, where jobs are served in their order of arrival, is generally considered to be fair and is in fact known to be weakly tail optimal for class-I job size distributions [2]. A class-I job size distribution $X$ is a light-tailed distribution for which some mild technical conditions hold such that the response time in an M/G/1 queue, where $X$ represents the service time, has exponential decay [1]. These distributions include all phase-type distributions [5] as well as any distribution with finite support. Recall that a distribution is light-tailed if there exists an $\epsilon > 0$ such that $E[e^{-\epsilon X}]$ is finite. Thus for any class-I job size distribution $X$, we have

$$P[R_{FCFS} > t] \sim c_{FCFS} e^{-\theta_Z t},$$

where $R_{FCFS}$ is the job response time in an M/G/1 queue with job size distribution $X$ under the FCFS scheduling discipline [1, Section 5]. The fact that FCFS is weakly tail optimal means that FCFS has the highest possible decay rate $\theta_Z$ of all scheduling disciplines.

In a recent paper [4] it was shown that the *prefactor* $c_{FCFS}$ is however not minimal among all scheduling algorithms (meaning FCFS is not strongly tail optimal) and a lower prefactor can be achieved by the so-called *Nudge* scheduling algorithm. The Nudge algorithm in [4] was such that the scheduler needs to known whether the size of an incoming job exceeds certain thresholds. It was subsequently shown in [6] that similar results can be obtained in a much more relaxed setting where jobs are partitioned into two types and the scheduler only needs to know the type of incoming jobs. This was done by introducing the Nudge-$K$ scheduling algorithm. Nudge-$K$ operates as follows: when a type-1 job arrives at time $t$ and the previous $k \leq K$ *arrivals* were type-2, then the incoming type-1 job is served before any of these $k$ type-2 jobs that are still waiting in the queue at time $t$. Note that under Nudge-$K$ a type-1 job can pass up to $K$ type-2 jobs, but a type-2 job can be passed at most once.

Let $R_{Nudge-K}$ be the response time in an M/G/1 queue using the Nudge-$K$ scheduling algorithm. It was shown in [6] that

$$P[R_{Nudge-K} > t] \sim c_K e^{-\theta_Z t},$$

for some prefactor $c_K$ and this prefactor is minimized over $K \geq 0$ by setting $K = \max(0, K_{opt})$ with

$$K_{opt} = \left\lfloor \log\left(\frac{\tilde{S}_1(-\theta_Z)(\tilde{S}_2(-\theta_Z)-1)}{\tilde{S}_2(-\theta_Z)(\tilde{S}_1(-\theta_Z)-1)}\right) \Big/ \log(\tilde{S}(-\theta_Z)) \right\rfloor, \quad (1)$$

where $\tilde{S}(s)$ is the Laplace transform of a random job and $\tilde{S}_i(s)$ of a type-$i$ job, for $i = 1, 2$. When $K_{opt} \leq 0$, then $c_{FCFS} < c_K$ for any $K > 0$, while otherwise $c_{FCFS} > c_K$ for $K = K_{opt}$.

The objective of this paper, which is a short version of [3], is to answer the following question:

*Given that jobs are partitioned in two types, which Nudge-like scheduling algorithm that only uses job types and their arrival order minimizes the prefactor?*

Our answer is the Nudge-$M$ scheduling algorithm. When a type-1 job arrives at time $t$ under Nudge-$M$, it passes any type-2 job still waiting in the queue provided that this type-2 job was among the last $M$ arrivals (before time $t$). We first derive a simple expression for the prefactor $c_M$ of the Nudge-$M$ scheduling algorithm and rather surprisingly show that minimizing $c_M$ is done by setting $M = \max(0, M_{opt})$ with $M_{opt} = K_{opt}$ given by (1).

More importantly, we introduce a large family $\mathcal{F}$ of Nudge-like scheduling algorithms (see Section 2) and show that Nudge-$M$ with $M = M_{opt}$ minimizes the prefactor among all the scheduling algorithms in this family $\mathcal{F}$, meaning Nudge-$M$ with $M = M_{opt}$ is strongly tail optimal within $\mathcal{F}$.

The Nudge-$M$ scheduling algorithm only uses the type of arriving jobs and the order in which they arrive. One may wonder to what extent the prefactor can be further reduced if more information is used, such as the exact arrival times. In a closely related paper [7] that was written concurrently to this paper, the authors introduce the $\gamma$-Boost scheduling algorithm. This algorithm minimizes the prefactor for class-I job size distributions in an M/G/1 queue among all scheduling policies in case the job size $s$ of each individual job as well as the arrival time of each job is known. Moreover the authors also propose a $\gamma$-Boost algorithm in case the jobs are partitioned into several types and the scheduler only has information about the job types and arrival time, but not the individual sizes. In this setting the boost of a job depends on its type only. The authors of [7] prove that $\gamma$-Boost achieves a lower prefactor than the Nudge-$M$ algorithm by exploiting the additional arrival time information. In this paper we prove that the prefactor of $\gamma$-Boost and Nudge-$M$ coincide in the heavy traffic limit. This indicates that the gain offered by the additional arrival time information vanishes as the load tends to one.

This short paper is structured as follows. Section 2 presents the model as well as the family of scheduling algorithms under consideration, while Section 3 lists the main results.

## 2. MODEL AND ALGORITHMS

We consider a queuing system with two types of jobs. Arriving jobs are either type-1 with probability $p$ or type-2 with probability $1-p$, and consecutive types are independent. Jobs arrive following a Poisson process with parameter $\lambda$. Let $E[X_i]$ be the mean job size of a type-$i$ job, for $i = 1, 2$. Without loss of generality, assume that $pE[X_1] + (1-p)E[X_2] = 1$ so that the load of the system is $\lambda$. We demand that the job size distribution is a class-I distribution.

We consider the following family $\mathcal{F}_M$ of Nudge scheduling algorithms. Let $t$ be the function that counts in number of twos in a string of any length consisting of ones and twos, e.g., $t(12122) = 3$. A Nudge scheduling algorithm belonging to $\mathcal{F}_M$ is characterized by a function $n$ from $\{1,2\}^M$ to $\{0,\ldots,M\}$ that obeys the following two conditions:

(C1) $n(s) \leq t(s)$,

(C2) $n(s_0 s_1 \ldots s_{M-1}) \leq n(s) + 1(s_0 = 2)$

for all $s = s_1 \ldots s_M \in \{1,2\}^M$, where $1(A) = 1$ if $A$ is true and $1(A) = 0$ otherwise. The interpretation of this function is as follows. Whenever a type-1 job arrives it looks at the types of the last $M$ arrivals. Assume these $M$ types are characterized by the string $s$, then the type-1 job passes the $n(s)$ most recent type-2 arrivals if they are still waiting in the queue. For instance, if $n(s) = 3$, but there are only two type-2 jobs waiting in the queue, then the type-1 job passes only these two type-2 jobs. Note that the condition (C2) on $n(s)$ guarantees that if a type-1 job may pass a type-2 job, all intermediate arrivals of type-1 may also pass this type-2 job. We clearly have that $\mathcal{F}_M \subset \mathcal{F}_{M+1}$, for any $M$, as for any $n(s)$ in $\mathcal{F}_M$, we can define $n'(s)$ in $\mathcal{F}_{M+1}$ such that $n'(s_1 \ldots s_M 1) = n'(s_1 \ldots s_M 2) = n(s_1 \ldots s_M)$.

The family $\mathcal{F}$ is defined as $\mathcal{F} = \bigcup_{M \geq 0} \mathcal{F}_M$ and contains a large variety of scheduling policies, such as Nudge-$K$ and Nudge-$M$, that use the last $M$ job arrival types for some $M$ to make scheduling decisions (see [3] for more examples). More specifically, for Nudge-$M$ the function $n(s)$ is such that $n(s) = t(s)$ as all type-2 jobs among the last $M$ may be passed. For Nudge-$K$ [6] we set $M = K$ and $n(s)$ equal to the number of leading twos in $s$ before encountering a one, as a type-2 job can be passed at most once.

## 3. RESULTS

The first result gives a simple expression for the ratio between the prefactor $c_M$ of Nudge-$M$ and the prefactor $c_{FCFS}$ of FCFS.

**Theorem 1.** *For Nudge-$M$ we have*

$$\frac{c_M}{c_{FCFS}} = (w_1 + (1-w_1)\tilde{S}(-\theta_Z)^M)(w_1 + w)^M, \qquad (2)$$

*with $w_1 = p\tilde{S}_1(-\theta_Z)/\tilde{S}(-\theta_Z)$ and $w = (1-p)/\tilde{S}(-\theta_Z)$. Further, $c_M/c_{FCFS}$ is convex in $M$ and achieves a unique minimum in $M_{opt} = K_{opt}$ defined by (1).*

PROOF. (Sketch) We limit ourselves to the expression for $c_M$. As we are analyzing the asymptotic behaviour, one can show that we may assume that an arriving type-1 job sees a queue with at least $M$ jobs still waiting, and that a type-2 job is waiting long enough that the next $M$ arrivals happen before it goes into service. Let $c_{W^{(i)}}$ be the prefactor of the type-$i$ waiting time. For a tagged type-1 job, we look at the workload without the last $M$ jobs, which has prefactor $c_Z/\tilde{S}(-\theta_Z)^M$. The waiting time consists of this work plus the workload associated with the type-1 jobs in the last $M$ arrivals as these are not passed. With probability $\binom{M}{k}(1-p)^k p^{M-k}$ we need to add the work of $M-k$ type-1 jobs, which yields (using the final value theorem)

$$\begin{aligned} c_{W^{(1)}}(M) &= \frac{c_Z}{\tilde{S}(-\theta_Z)^M} \sum_{k=0}^{M} \binom{M}{k} (1-p)^k p^{M-k} \tilde{S}_1(-\theta_Z)^{M-k} \\ &= c_Z \sum_{k=0}^{M} \binom{M}{k} \left(\frac{(1-p)}{\tilde{S}(-\theta_Z)}\right)^k \left(\frac{p\tilde{S}_1(-\theta_Z)}{\tilde{S}(-\theta_Z)}\right)^{M-k} \\ &= c_Z (w_1 + w)^M. \end{aligned}$$

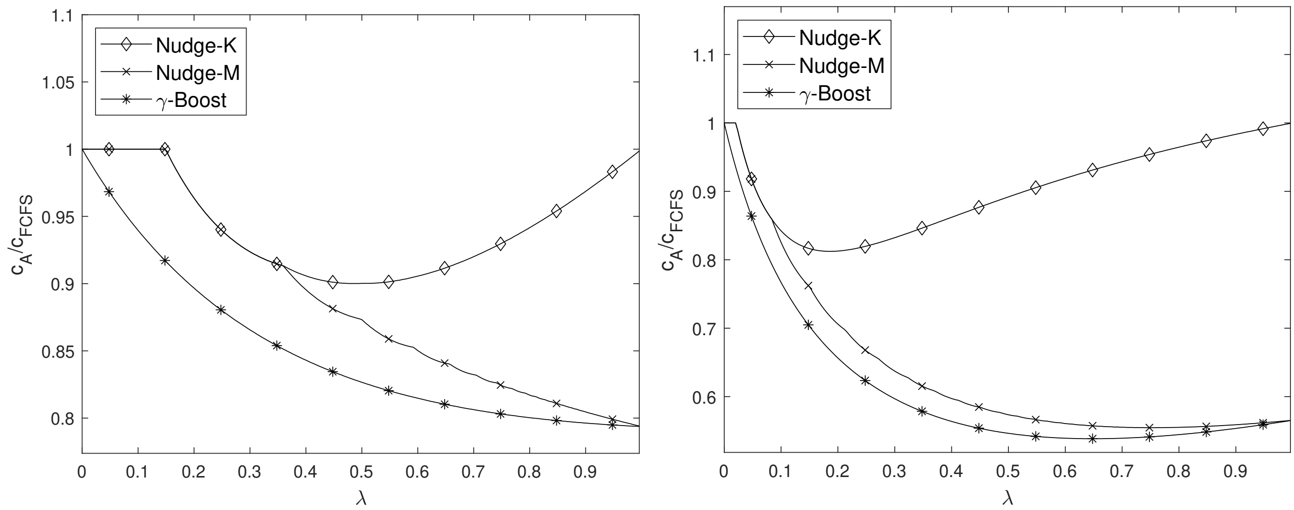
As for a tagged type-2 job, with probability $\binom{M}{k}(1-p)^{M-k}p^k$ there are $k$ type-1 jobs among the next $M$ arrivals and these $k$ jobs pass the tagged job. The waiting time of the type-2 job therefore equals the FCFS workload plus the work associated with $k$ type-1 jobs. Hence, by the final value theorem we find

$$\begin{aligned} c_{W^{(2)}}(M) &= c_Z \sum_{k=0}^{M} \binom{M}{k} (1-p)^{M-k} p^k \tilde{S}_1(-\theta_Z)^k \\ &= c_Z (w_1 + w)^M \tilde{S}(-\theta_Z)^M. \end{aligned}$$

Finally, $c_M = pc_{W^{(1)}}\tilde{S}_1(-\theta_Z) + (1-p)c_{W^{(2)}}\tilde{S}_2(-\theta_Z)$ and $c_{FCFS} = c_Z\tilde{S}(-\theta_Z)$. □

The next theorem establishes the optimality result for Nudge-$M$ in $\mathcal{F}$.

**Theorem 2.** *The Nudge-$M$ scheduling algorithm with $M = M_{opt} \geq 0$ has the smallest prefactor in $\mathcal{F}$. Further, Nudge-$M$ minimizes the prefactor in $\mathcal{F}_M$ for $M \leq M_{opt}$.*




(a) $E[X_2]/E[X_1] = 4$, $p = 2/3$

(b) $E[X_2]/E[X_1] = 10$, $p = 9/10$

Figure 1: The prefactor ratio of Nudge-$K$, Nudge-$M$, and $\gamma$-Boost (with optimized $K$, $M$, and boost) compared to FCFS with exponential job sizes. Nudge-$M$ significantly outperforms Nudge-$K$ for higher loads, while $\gamma$-Boost outperforms the different Nudge algorithms by exploiting the arrival time information.

PROOF. (Sketch) The proof proceeds in three steps. First a general expression for the prefactor $c_\pi$ of any scheduling algorithm $\pi \in \mathcal{F}$ is derived. This rather involved expression is subsequently used to show that for any $\pi \in \mathcal{F}$ and $s \in \{1,2\}^M$, if we increase $n(s)$ by one such that the conditions (C1) and (C2) on the function $n$ remain valid, then this leads to a reduction in the prefactor if and only if the position of the $n(s)+1$-st two in $s$ is in the first $M_{opt}$ positions. This result suffices to see that Nudge-$M$ is optimal in $\mathcal{F}_M$ for $M \leq M_{opt}$ as $n(s) = t(s)$ for any $s$ under Nudge-$M$. To prove optimality of Nudge-$M$ with $M = M_{opt}$ in $\mathcal{F}$ one shows that a sequence of operations can be found, where an operation increases or decreases $n(s)$ by one for some $s$, that gradually transforms any policy $\pi \in \mathcal{F}$ to Nudge-$M$ such that the prefactor reduces during each operation. ☐

**Theorem 3.** *For Nudge-M with $E[X_2] \geq E[X_1]$, we have*

$$\lim_{\lambda \to 1^-} \frac{c_M}{c_{FCFS}} = E[X_1]^{-pE[X_1]} E[X_2]^{-(1-p)E[X_2]} \quad (3)$$

*where*

$$M_{opt} \approx \left\lfloor \frac{\log(E[X_2]/E[X_1])E[X^2]}{2(1-\lambda)} \right\rfloor \quad (4)$$

*for $\lambda$ close to one.*

This result shows that the heavy traffic limit of $c_M/c_{FCFS}$ is insensitive to the shape of the job size distributions $X_1, X_2$ and $X$. We further note that the optimal $M$ for the heavy traffic regime given by (4) is expressed in terms of $\lambda, E[X_1]$, $E[X_2]$ and $E[X^2]$ only and may therefore be easier to estimate in practice than (1).

**Corollary 1.** *Let $c_\gamma$ denote the prefactor of $\gamma$-Boost [7], then*

$$\lim_{\lambda \to 1^-} c_M = \lim_{\lambda \to 1^-} c_\gamma,$$

*with $M = M_{opt}$ meaning the prefactors of $\gamma$-Boost and Nudge-M coincide in the heavy traffic limit.*

This result indicates that the additional arrival time information does not yield a tail improvement in the heavy traffic limit. It does however yield an improvement for $\lambda < 1$ as illustrated in Figure 1 which compares the prefactor ratio of three scheduling algorithms with optimized parameters and exponential job sizes. It shows significant gains for Nudge-$M$ over Nudge-$K$ when the load is sufficiently high. Note that Nudge-$K$ and Nudge-$M$ coincide when $M_{opt} \leq 1$. Finally the $\gamma$-Boost algorithm outperforms Nudge-$M$ as proven in [7] by exploiting arrival time information.

## Acknowledgment

This work was supported by the FWO project G0A9823N.

## 4. REFERENCES

[1] J. Abate, L. Choudhury, and W. Whitt. Waiting-time tail probabilities in queues with long-tail service-time distributions. *Queueing Systems*, 16:311—338, 1994.

[2] O. Boxma and B. Zwart. Tails in scheduling. *SIGMETRICS Perform. Eval. Rev.*, 34(4):13–20, mar 2007.

[3] N. Charlet and B. Van Houdt. Tail optimality and performance analysis of the Nudge-M scheduling algorithm. https://arxiv.org/abs/2403.06588, 2024.

[4] I. Grosof, K. Yang, Z. Scully, and M. Harchol-Balter. Nudge: Stochastically improving upon FCFS. *Proc. ACM Meas. Anal. Comput. Syst.*, 5(2), jun 2021.

[5] G. Latouche and V. Ramaswami. *Introduction to Matrix Analytic Methods and stochastic modeling.* SIAM, Philadelphia, 1999.

[6] B. Van Houdt. On the stochastic and asymptotic improvement of first-come first-served and nudge scheduling. *Proc. ACM Meas. Anal. Comput. Syst.*, 6(3):1–22, dec 2022.

[7] G. Yu and Z. Scully. Strongly tail-optimal scheduling in the light-tailed M/G/1. *To appear in Proc. ACM Meas. Anal. Comput. Syst.*, 2024.