Optimal Multiserver Scheduling under General Service Constraints in Heavy Traffic

RUNHAN XIE, University of California, Berkeley, USA
ZIV SCULLY, Cornell University, USA
ISAAC GROSOF, Georgia Institute of Technology, USA and Northwestern University, USA

Many modern applications of queueing theory focus on multiserver systems with complex service constraints. Two such systems are compatibility scheduling, where jobs from each class can only be processed by a subset of the servers, and multiserver job scheduling where each job occupies multiple servers simultaneously. Previous theoretical work either focuses entirely on maximizing server utilization, with no guarantee on performance (e.g. mean response time), or establishes optimality results for mean response time, but only under simple service constraints.

In this paper, we focus on scheduling multiserver systems with general service constraints. In this more general setting, we develop the Smallest Equalizing Bucket (SEB) policy, which we prove is the first policy to achieve optimal mean response time in heavy traffic in this general setting, under mild assumptions.

The key difficulty in designing a good scheduling policy is that we must simultaneously prioritize small jobs and also keep all servers busy, all while fitting within the complex service constraints. In single-server queues, keeping the single server busy is trivial, so prioritizing small jobs yields optimal mean response time. However, in more complicated systems, greedily prioritizing the smallest jobs may leave servers idle and cause instability. Our SEB policy maintains an even mixture of jobs across all classes and sizes, while also serving jobs among the smallest in the system, thereby achieving both objectives and achieving heavy-traffic optimality under general service constraints.

1 INTRODUCTION

Traditional multiserver queueing theory focuses on models with homogeneous jobs, where every job needs the same kind and amount of resources. These homogeneous models represent a tradeoff between applicability to real-world settings and amenability to theoretical analysis. However, to model many important modern systems, such as computing systems and service systems, more general service constraints are required. Important models with general service constraints include multiserver job (MSJ) models, where different jobs require different numbers of servers, and compatibility models, where each class of jobs can only be served by a subset of the servers. MSJ models are ubiquitous in modern datacenters, especially for training large machine learning models like large language models, which consistently require and hold multiple GPU/CPU cores simultaneously throughout the training processes [1, 40, 48]. Compatibility models are used in various settings, such as in ride-sharing platforms, where drivers only take riders traveling to certain destinations; call-centers, where certain employees handle certain calls; and in computer systems, where servers handle only jobs for which the relevant data is pre-stored [53, 55].

Designing and analyzing effective scheduling algorithms has always been a central topic of queueing theory. A well-chosen scheduling policy can dramatically improve performance (e.g. mean response time[1]) with no additional resources. In homogeneous multiserver models, scheduling has been the subject of several recent results, including policies with optimal heavy traffic mean response time [15, 16, 47]. However, such results are scarce under general service constraints, with

[1]A job's response time, also known as delay or sojourn time, is the amount of time between its arrival and its completion.

Authors' addresses: Runhan Xie, runhan_xie@berkeley.edu, University of California, Berkeley, Department of Industrial Engineering and Operations Research, Berkeley, CA, USA; Ziv Scully, zivscully@cornell.edu, Cornell University, School of Operations Research and Information Engineering, Ithaca, NY, USA; Isaac Grosof, igrosof@cs.cmu.edu, Georgia Institute of Technology, School of Industrial and Systems Engineering, Atlanta, GA, USA and Northwestern University, Department of Industrial Engineering and Management Sciences, Evanston, IL, USA.

most results in the setting focusing instead on throughput optimality [33, 34, 41, 49]. Our goal is to devise a scheduling policy with optimal heavy traffic mean response time in this more general setting, covering the MSJ and compatibility scheduling settings as special cases.

There is one recent paper of this type in the MSJ model [18], which shows that the ServerFilling-SRPT policy achieves optimal heavy traffic mean response time, but the result is highly limited by its requirement that the server need of every job is a power of two, or at least is a perfect divisor of the number of servers. We seek to prove a similar caliber of result in a far more general setting. As explained in our literature review (Section 2 and Table 1), there are no results in the compatibility scheduling setting which prove optimal heavy-traffic mean response time. We therefore ask:

> How do we schedule jobs of different classes under general service constraints to minimize mean response time?

1.1 Challenges

When minimizing mean response time, the primary goal of a scheduling policy is to serve the jobs of smallest (remaining) size. In a homogeneous job model like the M/G/k, straightforwardly prioritizing such jobs with the SRPT-k policy is sufficient to achieve an asymptotically optimal mean response time [15]. However, under our general service constraints, greedily serving small jobs may fail to use the full capacity of the system, or may leave us with gaps that can only be filled by large jobs. This occurs when there is a dramatic imbalance within the set of small jobs. For instance, in the compatibility scheduling setting, if all of the small jobs are the same class, which only half the servers can serve, there is no way to serve the small jobs.

We navigate this challenge by *preventing imbalance from arising in the first place*. We balance the set of small jobs in advance, and more generally balance all sets of jobs of similar sizes (Section 4).

1.2 Our Contribution

In this paper, we devise the Smallest Equalizing Bucket (SEB) policy, the first policy which provably achieves heavy-traffic optimal mean response time in multi-server systems with general service constraints. SEB's optimality holds under certain assumptions on the joint duration-class structure.

In order to both prioritize small jobs and balance the set of small jobs, SEB works as follows:

- We first divide job sizes into disjoint intervals (buckets). Each job is placed into a bucket based on its original size.
- Within each bucket, we find the ideal "equalizing" service option that maintains its balance
- We find the smallest-size bucket for which the ideal service option is available, and serve that bucket. We only move on to a larger-size bucket if the ideal service option cannot be fulfilled in the current bucket.

While our model is much more general than past settings in which similar optimality results have been proven, it is still restricted in certain ways. Key assumptions include (1) job size distributions are bounded; (2) the complete resource pooling condition is met in heavy traffic; (3) job size and job class are independent. We discuss these assumptions in more detail in Section 3.5, and leave lifting these restrictions to future work.

Another limitation of SEB is its empirical performance at lower loads. In Section 8, we compare SEB to several heuristics in simulation. We find that outside of the heavy-traffic regime SEB was designed to solve, the heuristics outperform SEB. Nevertheless, our simulations suggest that, with one exception, the heuristics are not heavy-traffic optimal. The exception is a heuristic called MaxWeight-Queue SRPT, which we identify as a strong candidate for further study (Appendix B).

The rest of the paper is organized as follows. Section 2 reviews prior work. Section 3 describes our model of size-based scheduling with service constraints. Section 4 describes the SEB policy

Policies \ Results	Throughput	Mean Response Time	Job-Server Configuration
FCFS	Not optimal	Analyzed, not optimal	General
MaxWeight	Optimal	Not analyzed	General
Randomized Timers	Optimal	Not analyzed	General
ServerFilling/ DivisorFilling	Optimal	Analyzed, not optimal	Restrictive: server needs divide number of servers
Idle-Avoid $c\mu/m$ Rule	Optimal	Optimal in many-server limit over a finite horizon	General
ServerFilling-SRPT/ DivisorFilling-SRPT	Optimal	Heavy-traffic optimal	Restrictive: server needs divide number of servers
Smallest Equalizing Bucket (SEB)	**Optimal**	**Heavy-traffic optimal**	**General**

Table 1. Comparison of optimality results for our paper and for prior work

and the intuition behind its design. Section 5 states our main result, namely SEB's heavy-traffic optimality, which we prove over the course of Sections 6 and 7.

2 PRIOR WORK

In this section, we review prior work on the subjects covered in this paper. In Section 2.1, we discuss prior work on MSJ scheduling, in Section 2.2, we discuss prior work on compatibility scheduling, and in Section 2.3, we discuss prior work on generalized service constraints.

2.1 Multiserver-job scheduling

Multiserver-job (MSJ) scheduling has become a topic of interest in the queueing theory community recently. Despite recent advances, theoretical results on MSJ scheduling remain limited [24]. We now give a brief overview of MSJ scheduling policies with theoretical guarantees. In Table 1, we compare our results in this paper to prior policies and results.

First-Come-First-Served (FCFS): FCFS with head-of-line-blocking is the most straightforward policy. However, FCFS is generally not throughput nor mean response time optimal due to blocking at the front of the queue. The stability region under FCFS is characterized under restrictive assumptions (e.g. [4, 38, 43]). Mean response time under FCFS is known exactly only in very specific settings [6, 12, 31]. A general bound and heavy-traffic characterization of the mean response time under FCFS is recently established in [19].

MaxWeight and Randomized Timers: MaxWeight is shown to be throughput optimal in both the known-size (i.e. the scheduling policy uses size information of jobs in system) [34] and unknown-size [33] settings. Randomized timers is a non-preemptive scheduling policy that does not require size information. It is shown that randomized timers is also throughput optimal [14, 41]. There is no theoretical result quantifying the mean response times under either policy, but based on simulation results, it is unlikely that either policy could be mean response time optimal.

ServerFilling/DivisorFilling: Grosof et al. [17] introduce a queueing framework called Work Conserving Finite Skip (WCFS) and propose the ServerFilling/DivisorFilling policies, which consider the minimal set of jobs necessary to fill all of the servers. A critical assumption in [17] is that the server needs of jobs must divide the number of servers, ensuring that it is possible to fill all

of the servers whenever enough jobs are present. It is shown that ServerFilling/DivisorFilling is throughput optimal and the heavy-traffic mean response time is comparable to that in M/G/1/FCFS.

ServerFilling/DivisorFilling-SRPT: Following work in ServerFilling/DivisorFilling, Grosof et al. [18] propose the ServerFilling/DivisorFilling-SRPT policies, which prioritize jobs of shortest remaining size. This is the first MSJ scheduling policy that is proved both throughput and heavy-traffic mean response time optimal. However, the restrictive assumption that the server needs of jobs must divide the number of servers is still in place.

Other policies: In addition to the policies above, a number of other scheduling policies have been proposed: variants of Backfilling e.g. [8, 30, 52], Shortest Area First [8], reinforcement learning based Shortest Job First [20], and the idle-avoid $c\mu/m$ rule [57], among many others. Mean response time optimality results have been shown in the many-servers limit [26, 57], but heavy-traffic optimality remains open in general, as we illustrate in Table 1.

A related class of models is called Multi-Resource Jobs (MRJ), where a job requires sufficient amount of multiple types of resources such as processors, memory, storage space, etc. before it can enter service. MaxWeight is shown to be throughput optimal and heavy-traffic optimal [35]. Randomized timers is another throughput optimal policy [14, 41]. Recently, a new and easy-to-implement policy is proposed in [9] and a bound on mean response time is established.

2.2 Queues with Compatibilities

In recent years, queues with compatibility between jobs and servers are extensively studied. In these queues, a job can only enter service at a subset of servers as specified by a compatibility graph. There are many variants of the compatibility model. The variant we consider in this paper has bipartite compatibility graphs, and each job can only receive service from one server at a time. For this model, it is shown that when job sizes are exponentially distributed, the stationary distributions are of product-forms (e.g. [3, 13, 51]) under specific position-based or random service policies. In a few restrictive settings, such product-form results hold for general job size distribution (e.g. [2, 22, 56]).

On the scheduling side, most existing work in similar settings treats jobs as indistinguishable (that is, the load-balancer does not have information on sizes of individual jobs) and focuses on load-balancing algorithms. Optimality results therein are established in mean-field limit or many-server limit (e.g. [39, 44, 50, 54]). We are aware of no prior study of scheduling known-size jobs in the compatibility scheduling setting.

2.3 MaxWeight under General Service Constraints (Generalized Switch)

The setting of general service constraints, which is the broad focus of this paper and which we define in Section 3.1, is also known in the literature as the *generalized switch* setting [49]. It has this name because it was originally introduced as a generalization of the $n \times n$ (input-queued) switch, but as our results focus on the complete-resource-pooling (CRP) setting [25], the $n \times n$ switch's most interesting behavior is not explored, so we use a more flexible name.

The most-studied policy in the generalized switch/generalized service constraints setting is the MaxWeight policy. Under a CRP assumption [25], and in a discrete time setting where jobs are indistinguishable, MaxWeight experiences state-space collapse, maximizes throughput and minimizes mean work in heavy traffic [49]. Note that minimizing mean work is not the same as minimizing mean response time in a setting where jobs are not indistinguishable, such as ours. Note that we make an equivalent CRP assumption: See Sections 3.2 and 3.5. Low complexity MaxWeight variants have also been shown to achieve mean-work-optimality in the same heavy traffic regime [28], as well as lower-information policies [10].

Outside of the CRP condition, the behavior of MaxWeight has also been studied recently, demonstrating collapse to a higher-dimensional subspace and characterizing the heavy-traffic stationary behavior [27]. Further empirical study also exists in the packet-switching setting, showing that policies can outperform MaxWeight on simple topologies [29].

Note that in all of the above models, jobs are either indistinguishable, or distinguished only by their classes. We are aware of no prior study of scheduling known-size jobs in the generalized switch setting, which is our focus.

3 MODEL

In this paper, we study optimal scheduling in the general service constraints (generalized switch) model. Because this is a highly abstract model, we also specialize our policy and our results to two motivating models: the *compatibility scheduling* model and the *multiserver-job scheduling* model. We define the *general service constraints scheduling* model in Section 3.1. We define several notions around stability which span all of these models in Section 3.2. We define the resource-pooled M/G/1 system in Section 3.4 and list the assumptions used throughout this paper in Section 3.5.

3.1 General service constraints scheduling (generalized switch)

The general service constraints (generalized switch) model we consider is as follows: There are n_s servers and n_c classes of jobs, and a set of service options $\widetilde{\mathcal{R}} = \{\widetilde{\mathbf{r}}\}$ to choose from. A service option $\widetilde{\mathbf{r}}$ specifies a number of jobs of each class which can be served at once.

For instance, in the multiserver-job (MSJ) setting, service options are any number of jobs with total server need up to n_s. In the compatibility scheduling setting, a service option is any set of jobs that can be assigned to all of the servers. For example, consider a MSJ setting with $n_s = 7$ servers, and where jobs can have server needs 1, 2, or 3. One possible service option is $\widetilde{\mathbf{r}} = [2, 1, 1]$, where two 1-server jobs are served, one 2-server job, and one 3-server job is served.

At any given time, the scheduling policy π selects any service option $\widetilde{\mathbf{r}} \in \widetilde{\mathcal{R}}$, and selects up to $\widetilde{\mathbf{r}}_i$ class-i jobs to be served, which can be any of the class-i jobs in the system.

Jobs of class i arrive according to a Poisson process with rate λ_i, for an overall arrival rate of λ. Each job has a service duration sampled i.i.d. from some general class-specific distribution with random variable D_i. The remaining service time and class of each job in the system is known to the scheduling policy at all times. The scheduler may choose any service option from the list at any moment in time. Jobs may be preempted and resumed with no overhead or loss of work.

3.2 Stability, facets, and sizes

In Section 3.1, we have associated with each job a duration. This is the total amount of time a job spends at the server(s) before it is completed. If the current service option is $\widetilde{\mathbf{r}}$, then the total remaining duration of jobs in class i is decreasing with rate $\widetilde{\mathbf{r}}_i$. Using the job duration, we define the system load as $\widetilde{\boldsymbol{\rho}}_i = \lambda_i \mathbb{E}[D_i]$. Then the stability region $\widetilde{\mathcal{S}}$ of the system is the open interior of the convex hull formed from all service options: $\widetilde{\mathcal{S}} = \text{Interior}(\text{ConvexHull}(\widetilde{\mathcal{R}}))$.

So far we have defined all quantities using the duration of jobs. In this paper, however, we will work with the *size* of a job, rather than duration. Where a duration is an amount of time, and might be measured in seconds, size is measured in system-seconds. Intuitively, a job's size is the product of the fraction of the system's capacity that a job uses and the job's duration. However, subtleties arise when attempting to rigorously define a job's "fraction of system capacity".

Some examples help illustrate the concept. In the compatibility scheduling setting, each job uses one of the n_s servers. Thus, a job of duration d has a size of d/n_s. In the MSJ setting, in the part of the capacity region $\widetilde{\mathcal{S}}$ where all servers are occupied, a job with duration d and server need k has a

size of kd/n_s. In more general MSJ systems, however, there are service options on the surface of $\widetilde{\mathcal{S}}$ that can't fill all servers. In these cases, it is not obvious how we should convert duration to size. The situation becomes even more subtle in our full setting of general service constraints.

In general, to define size, we define a facet-based mapping to convert duration to size. We associate to each facet on $\widetilde{\mathcal{S}}$ a set of conversion coefficients. Specifically, let $\widetilde{\mathcal{F}}$ be a facet and $\widetilde{\mathcal{R}}^{\widetilde{\mathcal{F}}}$ the set of service options on the facet. Then our conversion coefficients $\mathbf{k} = (k_1, \ldots, k_m)$ are the solution to the system of equations $\langle \mathbf{k}, \widetilde{\mathbf{r}} \rangle = 1$ for all $\widetilde{\mathbf{r}} \in \widetilde{\mathcal{R}}^{\widetilde{\mathcal{F}}}$. With these conversion coefficients defined, we define size of a class-i job with duration d to be $k_i d$, relative to a given facet $\widetilde{\mathcal{F}}$. The conversion coefficients guarantee that all service options in $\widetilde{\mathcal{R}}^{\widetilde{\mathcal{F}}}$ process total remaining size at rate 1 and all other service options process at rate no greater than 1.

For example, consider a MSJ setting with 11 servers, where jobs can have server needs 2 (class 1) or 3 (class 2). There are three facets on $\widetilde{\mathcal{S}}$: One with vertices $[5, 0]$ and $[4, 1]$, one with vertices $[4, 1]$ and $[1, 3]$, and one with vertices $[1, 3]$ and $[0, 3]$. The conversion coefficients associated with these two facets are $[1/5, 1/5]$, $[2/11, 3/11]$, and $[0, 1/3]$, respectively. Note that in the third case, one of the size conversion coefficients was 0. This happens when the facet is parallel to an axis. This represents a pathological edge-case, and our results do not focus on this case.

Given a load vector $\widetilde{\boldsymbol{\rho}}$ in the capacity region $\widetilde{\mathcal{S}}$, we define the load vector's *dominating facet* to be the facet that contains a scalar multiple of the load vector. For instance, in the 11-server MSJ setting, the dominating facet of the load vector $[2, 1]$ is the facet with endpoints $[4, 1]$ and $[1, 3]$. We avoid the edge-case where the load vector's scalar multiple is on the boundary of multiple facets.

For a duration-based load vector $\widetilde{\boldsymbol{\rho}}$ with dominating facet $\widetilde{\mathcal{F}}$, we define its corresponding size-based load vector: $\boldsymbol{\rho}_i = \lambda_i \mathbb{E}[S_i]$, where $S_i = k_i D_i$, and the size-based dominating facet $\mathcal{F}$. The presence of a $\sim$ indicates a duration-based quantity, while the absence indicates a size-based quantity. The total system load is $\rho = \|\boldsymbol{\rho}\|_1$. The system is stabilizable if and only if $\rho < 1$, which is an equivalent condition to $\widetilde{\boldsymbol{\rho}} \in \widetilde{\mathcal{S}}$. For instance, in the 11-server MSJ setting, the duration-based load vector $[2, 1]$ has conversion coefficient $[2/11, 3/11]$, from its facet, and size-based load vector $[4/11, 3/11]$, for a total system load of $7/11$.

We now specify the heavy-traffic regime we are interested in. Let $\widetilde{\mathcal{F}}$ be the dominating facet. Let $\widetilde{\boldsymbol{\nu}}$ be a point in the interior of $\widetilde{\mathcal{F}}$. We consider a sequence of systems, indexed by a stability gap ε, with load vectors $\widetilde{\boldsymbol{\rho}}^{(\varepsilon)} = (1 - \varepsilon)\widetilde{\boldsymbol{\nu}}$. In this sequence of systems, we hold durations D_i constant, and allow the arrival rates $\lambda_i^{(\varepsilon)}$ to scale linearly with ε. Let $\boldsymbol{\rho}^{(\varepsilon)}$ and $\boldsymbol{\nu}$ be the corresponding vectors after size conversion. These systems have linearly related load and arrival rates. We say that we are in the heavy-traffic regime when examining the limit $\varepsilon \to 0$. Note that we assume that $\widetilde{\boldsymbol{\nu}}$ is in the *interior* of $\widetilde{\mathcal{F}}$, not its boundary. This assumption is referred to in literature as the Complete-Resource-Pooling (CRP) [25] condition. Throughout the paper, we consider an arbitrary such vector $\widetilde{\boldsymbol{\nu}}$, and we will write $\lim_{\rho \to 1}$ to denote this heavy-traffic regime.

From now on we will work primarily with size unless specified otherwise.

3.3 Idleness

In this section, we defined the idleness and relevant idleness of the system. Idleness at time t, $\mathcal{I}(t)$, measures how much processing capacity is wasted when a given service option is adopted at time t. We assume here that for a duration-based service option $\widetilde{\mathbf{r}}$ adopted at time t, the number of class-i jobs in service, $\widetilde{\mathbf{r}}_i$, is no more than the total number of class-i jobs in systems. Let $\mathbf{r}$ be its corresponding size-based service vector, then the system idleness at time t is defined as $\mathcal{I}(t) = 1 - \|\mathbf{r}\|_1$.

Similarly, we can define the x-relevant idleness of the system at time t, $\mathcal{I}_{\leqslant x}(t)$, as the fraction of capacity wasted on processing jobs with remaining sizes no more than x. Let $\widetilde{\mathbf{r}}_{\leqslant x}$ be the x-relevant service option, where $\widetilde{\mathbf{r}}_{\leqslant x,i}$ is the number of class-i jobs selected by the service option that have remaining sizes no more than x. Let $\mathbf{r}_{\leqslant x}$ be its corresponding size-based service vector, then the x-relevant idleness at time t is defined as $\mathcal{I}_{\leqslant x}(t) = 1 - \|\mathbf{r}_{\leqslant x}\|_1$

3.4 Resource pooling

We consider an M/G/1 queue corresponding to the dominating facet, which we call the *resource-pooled* M/G/1. This resource-pooled M/G/1 queue is defined so that jobs arrive according to the same process as in the original system and the duration of a job equals the size of the job in the original system, defined relative to the given facet. Since we consider a single-server queue, the service constraints in the original system do not apply and the server can work on any job, or any combination of jobs. The Complete Resource Pooling condition guarantees that such an M/G/1 queue is unique. The two policies we consider for the resource-pooled queue are Preemptive-Shortest-Job-First (PSJF-1), which prioritizes the job with the smallest original size, and Shortest-Remaining-Processing-Time (SRPT-1), which prioritizes the job with the smallest remaining size.

3.5 Model Assumptions

The Markovian descriptor of the system is a triple: $(\mathbf{s}, \mathbf{s}_r, \mathbf{c})$, namely a vector of the original sizes of jobs in the system, a vector of their remaining sizes, and a vector of classes of these jobs.

We make the following assumptions about the system.

(1) Over the joint (class, duration) distribution, job class and job size are independent.
(2) The class-i job size duration distribution S_i is bounded. That is, the supports of all S_i's are contained in interval $[s_{\min}, s_{\max}]$ where $0 < s_{\min} < s_{\max} < \infty$.
(3) Complete resource pooling: The load vector $\widetilde{\boldsymbol{\rho}}^{(\varepsilon)}$ and its corresponding heavy-traffic limit $\widetilde{\boldsymbol{\nu}}$ are in the interior of the dominating facet $\widetilde{\mathcal{F}}$.

While Assumptions (2) and (3) are relatively standard in the literature, we now discuss Assumption (1) further. Assumption (1) is easiest to understand in the compatibility scheduling and MSJ scheduling settings. In the compatibility scheduling setting, a job's size is d/n_s, where d is the job's duration, so assumption (1) is equivalent to the assumption that a job's duration is independent of the job's class, or in other words that the duration distribution for each class is the same.

In the MSJ scheduling setting, consider the case in which the dominating facet is the one on which all service options utilize all servers. Note that if the numbers of servers n_s is large compared to the largest server need of a job, this facet will dominate the great majority of the stability region.

In this case, a job's size is kd/n_s, where k is the job's server need and d is its duration. Note that kd is often referred to as a job's *area* [8]. In this setting, Assumption (1) states that a job's area is independent of the job's class, or in other words that the area distribution for each class is the same.

Despite these assumptions, our result is still far more general than the prior state of the art: No previous paper has shown optimal mean response time under any general service option setting. Moreover, these assumptions do not remove the key challenge of the setting, namely that we must simultaneously prioritize serving small jobs and keeping sufficient balance among the job classes that we can keep all servers well-utilized.

4 OUR POLICY: SHORTEST EQUALIZING BUCKET (SEB)

In this section, we discuss the challenges that lie in the path of devising an optimal policy, the behavior of existing policies and the intuition behind our SEB policy.

4.1 What are the essential qualities of an optimal policy?

From examining prior optimality results in other settings within queueing and scheduling theory, one can observe two necessary prerequisites for minimizing mean response time.

(1) **Keeping all servers busy.** Before we analyze the mean response time of any policy in steady state, we must make sure that the policy stabilizes the system. Keeping all servers as busy as possible is key to stabilizing the system when the load approaches the total capacity.
(2) **Prioritizing smaller jobs.** To minimize mean response time, ideally, we would always prioritize jobs from smallest to largest (remaining) size, along the lines of SRPT. Prioritizing smaller jobs over larger jobs if possible is key to achieving optimal mean response time.

MaxWeight falls short because it focuses entirely on keeping all servers busy [49]. SRPT-k, ServerFilling-SRPT, and the idle-avoid $c\mu/m$ rule all focus on settings where keeping all servers busy is either easy or unnecessary [15, 18, 57]. Note however that there has been no analysis of policies that manage to keep all servers busy as well as MaxWeight, while simultaneously prioritize jobs based on size. Our SEB policy prioritizes both goals.

4.2 Intuition of our policy

To simplify the discussion, we assume for now that there are only two sizes of jobs $s_1 < s_2$, namely "small" s_1 and "large" s_2. We explain how to generalize the ideas to general bounded size distributions towards the end of this subsection.

The key idea behind our policy is to balance both the small jobs and the large jobs separately. Balance refers to keeping the work vector of small jobs, $\mathbf{w}_{\text{small}}$, and work vector of large jobs, $\mathbf{w}_{\text{large}}$, roughly parallel to the load vector $\boldsymbol{\rho}$. We define balance in this manner for two reasons: (1) Since $\boldsymbol{\rho}$ is not axis-parallel, keeping the work vector aligned with it prevents excessive depletion of work in certain classes. If too much work is drained from a specific class, there may not be enough jobs from that class to fulfill the preferred service option. (2) if a bucket does not receive any service, arrivals provide a drift in the direction of $\boldsymbol{\rho}$, so the bucket becomes more balanced.

To keep such a balance among the small or the large jobs, a preferred service option from the dominating facet is chosen for each, so that $\mathbf{w}_{\text{small}}$ and $\mathbf{w}_{\text{large}}$ would each become more aligned with $\boldsymbol{\rho}$ if those jobs were served.

To keep such balance *separately* for small jobs and for large jobs, we either serve only small jobs or only large jobs if we decide to serve jobs at all. In particular, we prioritize serving small jobs by only trying to serve the large jobs when the preferred service option for the small jobs is not available, because not enough small jobs of the correct classes are present. By maintaining our notion of balance, we will have an efficient service option for the small jobs as long as there are more than a few small jobs.

Note that our goal is not to squeeze out every drop of performance. Our goal is merely to achieve heavy traffic optimality. We therefore do not focus too much on the scheduling decisions when a bucket is near empty, as this is rare in heavy traffic. We take a simple option, skipping over that bucket, for ease of theoretical analysis.

Our intuition for small and large jobs generalizes to more sizes of jobs. For an arbitrary bounded job-size distribution, we divide the support of the distribution into disjoint intervals which we call "buckets". An arriving job falls into one of the buckets according to its original size and stays in the same bucket for the entirety of its time in system. Our policy maintains balance for all buckets separately, and gives smaller buckets (i.e. buckets corresponding to smaller sizes) higher priority if their preferred service option can be fulfilled.

Finally, we must decide how to choose which jobs go in which buckets. Here, we draw analogy to the dispatching setting [16], where it has been found that geometric buckets, where the ratio

of the smallest size and largest size in a bucket is set to be a properly chosen constant c, leads to heavy-traffic optimality. We show that such geometric buckets lead to heavy-traffic optimality in our setting as well.

4.3 Defining Our Policy

Our Shortest Equalizing Bucket (SEB) policy is defined as follows:

Preprocessing: Find the dominating duration-based facet $\widetilde{\mathcal{F}}$ that contains $\widetilde{\boldsymbol{\rho}}$ and solve for the duration-to-size conversion coefficients by solving for $\mathbf{k}$ from the systems of equations $\langle \mathbf{k}, \widetilde{\mathbf{r}} \rangle = 1$ for all $\widetilde{\mathbf{r}} \in \mathcal{R}^{\widetilde{\mathcal{F}}}$. After the duration-size conversion, we now define size buckets based on the support of the size distribution $[s_{\min}, s_{\max}]$. Bucket i is defined as an interval $[b_{i-1}, b_i)$ so that $b_i/b_{i-1} = c$ (we set $b_0 = s_{\min}$), where

$$c = 1 + \frac{1}{1 + \log\left(\frac{1}{1-\rho}\right)}.$$

Let n_b denote the total number of buckets. The above constant-ratio definition for b_i is our definition for b_0 through b_{n_b-1}. We set the last bucket to be $[b_{n_b-1}, b_{n_b}]$, where we define $b_{n_b} = s_{\max}$. When a job arrives, it is added to the bucket containing its original size and stays in the same bucket until it is completed. We now define SEB's online behavior.

The scheduler iterates through the size buckets in increasing order.

(1) For each bucket i, the scheduler first computes the bucket work vector $\mathbf{w}^{(i)}$, obtained by summing remaining sizes of jobs in bucket i by class. Then it finds the ideal service rate vector $\mathbf{r}^{(i)*}$ among all service rates at the corners of the size-based dominating facet $\mathcal{F}$: $\mathbf{r}^{(i)*} = \operatorname{argmax}_{\mathbf{r} \in \mathcal{R}^{\mathcal{F}}} \left\langle \mathbf{w}^{(i)}_{\perp \boldsymbol{\rho}}, \mathbf{r} \right\rangle$. The scheduler then checks if the corresponding service option $\widetilde{\mathbf{r}}^{(i)*}$ can be fulfilled using jobs in the bucket.

(2) If service option $\widetilde{\mathbf{r}}^{(i)*}$ can be performed using jobs in bucket i, the scheduler places those jobs in service, and the scheduler stops iterating through the buckets. The scheduler pick jobs among those in a given class within the bucket in FCFS order.

(3) Otherwise, the scheduler moves on to bucket $i + 1$. If no bucket can fulfill its preferred service options, all servers idle.

The scheduler reruns the algorithm if there is an arrival or departure. It's worth noting that the preferred service option $\widetilde{\mathbf{r}}^{(i)*}$ might change even when there are no arrivals or departures. This happens when, after some service, the work vector arrives at a point where multiple service options become equally preferable. In this case, the scheduler will rapidly alternate between those service options, resulting in a weighted-processor-sharing behavior. This emergent processor-sharing behavior is common in scheduling policies, with the notable example of Least Attained Service (Foreground-Background) [23].

4.4 Basic property of SEB

Before we analyze how the system behaves under SEB, we establish an important property of the system: For any workload, there exists a balancing service rate option.

Proposition 4.1. *If the service rate vectors on the corner of a facet $\mathcal{F}$ which dominates the load vector $\boldsymbol{\rho}$ are of full-rank, then for any $\mathbf{w}$ such that $\mathbf{w}_{\perp \boldsymbol{\rho}} \neq \mathbf{0}$, there exists an $\varepsilon_0 > 0$, independent of $\mathbf{w}$, such that* $\max_{\mathbf{r} \in \mathcal{R}^{\mathcal{F}}} \frac{\langle \mathbf{w}_{\perp \boldsymbol{\rho}}, \mathbf{r} \rangle}{\|\mathbf{w}_{\perp \boldsymbol{\rho}}\|} \geqslant \varepsilon_0$.

Proposition 4.1 roughly says that for any bucket work vector $\mathbf{w}$, the preferred service option always incurs a non-vanishing drift towards balancing the bucket further. We will formalize this intuition in Section 6.3. Before proving Proposition 4.1, we first establish a lemma.

Lemma 4.2. *If the service rate vectors on the corner of a facet $\mathcal{F}$ which dominates the load vector $\boldsymbol{\rho}$ are of full-rank, then for any $\mathbf{x} \neq \mathbf{0}$ such that $\langle \mathbf{x}, \boldsymbol{\rho} \rangle = 0$, we have $\max_{\mathbf{r} \in \mathcal{R}^{\mathcal{F}}} \langle \mathbf{x}, \mathbf{r} \rangle > 0$.*

Proof. Let $\mathcal{R}^{\mathcal{F}} = \{\mathbf{r}_1, \ldots, \mathbf{r}_n\}$ and consider the convex hull of $\mathcal{R}^{\mathcal{F}}$. Since $\boldsymbol{\rho}$ is in the interior of this convex hull, there exist strictly positive scalars $\alpha_1, \ldots, \alpha_n$ such that $\boldsymbol{\rho} = \sum_{i=1}^{n} \alpha_i \mathbf{r}_i$ (see Exercise 3.1 in [7]). Since $\langle \mathbf{x}, \boldsymbol{\rho} \rangle = 0$, for any nonzero $\mathbf{x}$, either there must exist $\mathbf{r} \in \mathcal{R}^{\mathcal{F}}$ such that $\langle \mathbf{x}, \mathbf{r} \rangle > 0$, or $\langle \mathbf{x}, \mathbf{r}_i \rangle = 0$ for all i. The latter implies that $\mathbf{x}$ is linearly independent of all $\mathbf{r}_i$, which is impossible since $\mathcal{R}^{\mathcal{F}}$ is of full rank. □

Proof of Proposition 4.1. If the claim is false, there exists a sequence $\{\mathbf{w}^{[n]}\}_{n=1}^{\infty}$ such that

$$\lim_{n \to \infty} \max_{\mathbf{r} \in \mathcal{R}^{\mathcal{F}}} \frac{\left\langle \mathbf{w}^{[n]}_{\perp \boldsymbol{\rho}}, \mathbf{r} \right\rangle}{\|\mathbf{w}^{[n]}_{\perp \boldsymbol{\rho}}\|} = 0.$$

Since the set $\{\mathbf{x} : \|\mathbf{x}\| = 1\} \cap \{\mathbf{x} : \langle \mathbf{x}, \boldsymbol{\rho} \rangle = 0\}$ is compact, there exists a limit point $\mathbf{y}$ of the sequence $\{\frac{\mathbf{w}^{[n]}}{\|\mathbf{w}^{[n]}\|}\}_1^{\infty}$ in the compact set. It follows that $\max_{\mathbf{r} \in \mathcal{R}^{\mathcal{F}}} \langle \mathbf{y}, \mathbf{r} \rangle = 0$, contradicting Lemma 4.2. □

4.5 General Notation

Under SEB, for $i = 1, \ldots, n_b$, we define the idleness of the first i buckets as

$$\mathcal{I}_{\leqslant i} := \frac{1}{n_s} \sum_{j=1}^{n_s} \mathbb{1}(\text{server } j \text{ is not working on any job in the first } i \text{ buckets})$$

We define the total work in the first i buckets as $W_{\leqslant i}$. We define the total load into the first i buckets as $\rho_{\leqslant i}$. We define $W_{\leqslant i}^{\mathrm{M/G/1}}$ as the work in an M/G/1 queue with job size distribution the original distribution conditioned on size $\leqslant b_i$ and load $\rho_{\leqslant i}$.

When referring to the work vector of a bucket, we will use superscripts for the bucket number and the subscripts for the class number. For instance, $\mathbf{w}_j^{(i)}$ denotes the work of class j jobs in bucket i and $\mathrm{S}^{(i)}$ denotes the sizes of jobs falling into bucket i.

5 MAIN RESULTS AND ROADMAP

Our main result in this paper is to prove the heavy traffic optimality of our Smallest Equalizing Bucket (SEB) policy, the first heavy traffic optimality result in the general service constraints setting. We do so by comparing against the resource-pooled M/G/1 system, with durations equal to the sizes of the jobs, as defined in Section 3.4.

We prove that in the heavy traffic limit, the mean response time of SEB converges to that of the resource-pooled Shortest Remaining Processing Time (SRPT-1) policy, and that the SRPT-1 policy is a lower bound on the optimal policy in this setting:

Theorem 5.1. *Our policy is heavy-traffic optimal, under the assumptions in Section 3.5: For any $\widetilde{\boldsymbol{\nu}}$ on the capacity region that is in the interior of a facet, our SEB policy converges to optimal as the load vector $\widetilde{\boldsymbol{\rho}}$ converges to $\widetilde{\boldsymbol{\nu}}$.*

$$\lim_{\rho \to 1} \frac{\mathbb{E}[T^{\mathrm{SEB}}]}{\mathbb{E}[T^{\text{SRPT-1}}]} = \lim_{\rho \to 1} \frac{\mathbb{E}[T^{\mathrm{SEB}}]}{\mathbb{E}[T^{\mathrm{OPT}}]} = 1,$$

where $T^{\text{SRPT-1}}$ is the response time under resource-pooled SRPT.

We discuss our proof structure in Section 5.1, and prove the theorem in Section 7.3.

Our optimality proof relies critically on our heavy traffic characterization of the mean response of the SEB policy:

Theorem 5.2. *Under SEB, in the heavy traffic limit and under the assumptions in Section 3.5, mean response time has the following asymptotic behavior:*

$$\mathbb{E}[T^{\text{SEB}}] \leqslant c \cdot \mathbb{E}\left[T^{\text{PSJF-1}}\right] + \Theta\left(\log^2\left(\frac{1}{1-\rho}\right)\right) \quad \text{as } \rho \to 1. \tag{1}$$

where $T^{\text{PSJF-1}}$ is the response time under resource-pooled Preemptive-Shortest-Job-First (PSJF-1), and where c is the bucket width multiplier.

Theorem 5.2 is the focus of the majority of the technical section of this paper, and we discuss the proof structure in detail in Section 5.1. We prove the theorem in Section 7.2.

5.1 Roadmap to Optimality

We begin by proving that the system is throughput optimal under SEB. That is, we prove the system is stable for all $\rho < 1$ (Theorem 6.1).

We then establish an upper bound on mean response time under SEB for any load $\rho < 1$. A job's response time under policy π can be written as $T^\pi = T^\pi_{\text{wait}} + T^\pi_{\text{res}}$, where T^π_{wait} (wait time) is the time between when a job arrives and when it first receives any service and T^π_{res} (residence time) is the time between when a job first receives service and when it leaves the system. In our system, bounding $\mathbb{E}[T^\pi_{\text{res}}]$ is a straightforward application of Little's law (Lemma 7.4), whereas bounding $\mathbb{E}[T^\pi_{\text{wait}}]$ is much more complicated. Note also that $\mathbb{E}[T^\pi_{\text{wait}}]$ dominates under heavy traffic.

One of the key tools we use to bound $\mathbb{E}[T^\pi_{\text{wait}}]$ is the Work Integral Number Equality (WINE) technique [5, 42, 46, 47], which converts the analysis of mean response time to the analysis of mean relevant work in system.

Proposition 5.3 (Scully [46] Theorem 15.3). *For an arbitrary stable queueing system and an arbitrary scheduling policy π,*

$$\mathbb{E}[T^\pi] = \frac{1}{\lambda}\int_0^\infty \frac{\mathbb{E}[W^\pi_{\text{remaining size}\leqslant x}]}{x^2}\,dx,$$

where $\mathbb{E}[W^\pi_{\text{remaining size}\leqslant x}]$ is the total remaining size of all jobs with remaining sizes no more than x.

When we apply Proposition 5.3 to bound the waiting time, we apply it to the subsystem consisting of jobs that have not yet received any service, in Lemma 7.5. Note that the set of jobs which have not yet received service and which have remaining size $\leqslant x$ is a subset of the jobs in the system with original size $\leqslant x$. Thus, for any given threshold x, it suffices to bound $\mathbb{E}[W^\pi_{\text{original size}\leqslant x}]$.

Recall that SEB assigns a job to a bucket based on the original size of the job. As a result, to bound $\mathbb{E}[W^\pi_{\text{original size}\leqslant x}]$, we focus on total work in buckets $\leqslant i$, where bucket i is the bucket that a job of size x is placed in. Our strategy is to apply the Work Decomposition Law [46, 47] to the first i buckets. We state the Work Decomposition Law here in a way that is specialized to our setting.

Proposition 5.4 (Scully [46] Theorem 8.2). *For any service policy π that stabilizes the system,*

$$\mathbb{E}[W^\pi_{\leqslant i}] - \mathbb{E}[W^{\text{M/G/1}}_{\leqslant i}] = \frac{\mathbb{E}[\mathcal{I}_{\leqslant i} W^\pi_{\leqslant i}]}{1-\rho_{\leqslant i}}$$

for any $i = 1, \ldots, n_b$.

The key term in Proposition 5.4 is $\mathbb{E}[\mathcal{I}_{\leqslant i} W^\pi_{\leqslant i}]$, which we subsequently refer to as the *waste*. Waste intuitively measures how much work is not being processed by the servers. We will analyze the waste bucket by bucket. If there's little work $W^\pi_{\leqslant i}$, the waste is small. The challenge is to show that the waste is small in the presence of substantial work.

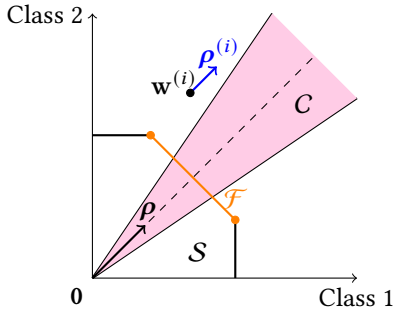


Fig. 1. Drift of bucket-i work vector in a two-class system, when service is not occurring.

Note that waste happens if we can't serve jobs in a bucket, so we need to show that when there is a lot of work in a bucket, it is very rare that the bucket is ineligible to receive service. To formalize this, we first define a cone around the load vector, then we consider two possible cases:

(1) When the bucket work vector is outside the cone, the vector on average drifts towards the cone. Thus, we show a state-space collapse result (Theorem 6.3). Using Theorem 6.3 we show that the probability that the work vector is so far from the cone that the bucket is ineligible for service is small (Theorem 7.2).
(2) When the bucket work vector is inside the cone, we show that the bucket must be eligible for service if there is a lot of work (Theorem 7.1).

Combining all bounds together, we obtain a bound on the mean response time relative to mean response time in the resource-pooled system (Theorem 5.2, proven in Section 7.2). Heavy-traffic optimality then follows from this bound, and the fact that the resource-pooled system forms a lower bound on optimal mean response time, as we show in Section 7.3.

6 SEB ANALYSIS: STABILITY AND BALANCING EACH BUCKET

6.1 Defining the Cone-based State-space Collapse

In this section, we give definitions relevant for cone-based state space collapse, which is a key idea in our optimality proof that we discuss further in Section 5.1.

Recall that in Section 4.2, we explain how SEB keeps all size buckets balanced by keeping the work vector of each bucket roughly parallel to the load vector $\boldsymbol{\rho}$. To formalize this intuition, we aim to establish a state-space collapse result for the bucket work vectors $\mathbf{w}^{(i)}$. However, we note that the work vectors do not collapse to the load vector itself. This is because when a bucket receives no service, drift due to arrivals moves the work vector parallel to $\boldsymbol{\rho}$, not towards it. Note that the conditional load $\boldsymbol{\rho}^{(i)}$ arriving to any bucket i is parallel to $\boldsymbol{\rho}$, due to our assumption of independence between job class and job size (Section 3.5).

The fact that the arriving load vector $\boldsymbol{\rho}^{(i)}$ is parallel to $\boldsymbol{\rho}$ is useful: it implies that the drift will move the work vector towards a *cone* around $\boldsymbol{\rho}$, as illustrated in Fig. 1. We will show that whether or not the bucket receives service, the system drifts towards the cone. The cone we consider is

$$C = \left\{ \mathbf{w} \in \mathbb{R}^{n_c} : \frac{\|\mathbf{w}_{\parallel \boldsymbol{\rho}}\|_2}{\|\mathbf{w}\|_2} \geqslant \cos \varphi \right\},$$

where $\varphi > 0$ is chosen so that the following conditions are satisfied:

- φ is small enough that C is contained in the interior of the convex cone generated by service rate vectors in $\mathcal{R}^{\mathcal{F}}$.

- φ is small compared to the constant ε_0 in Proposition 4.1: $\left(\max_{\mathbf{r}\in\mathcal{R}^{\mathcal{F}}} \frac{\langle \boldsymbol{\rho},\mathbf{r}\rangle}{\|\boldsymbol{\rho}\|_2}\right)\tan\varphi < \varepsilon_0$.
- φ is small compared to the lowest-load class in the system: $\tan\varphi < \frac{\boldsymbol{\rho}_{\min}}{\|\boldsymbol{\rho}\|_2}$, where we define $\boldsymbol{\rho}_{\min} = \min_i \boldsymbol{\rho}_i$.

Note that we use the same cone C for all buckets i. That is, we show in Section 6.3 that each bucket work vector $\mathbf{w}^{(i)}$ converges to the same cone C.

6.2 Stability

Theorem 6.1. *The system is stable under SEB for any $\rho < 1$.*

Full proof deferred to Appendix A.1.

Proof outline. Our main tool is the continuous-time Foster-Lyapunov Theorem in [36]. The key idea is to find a nonnegative Lyapunov function V that has bounded drift on a compact set and negative drift outside that set. Our choices of V rely on the following observations:

- When bucket i does not receive any service, the average drift of $\mathbf{w}^{(i)}$ is parallel to $\boldsymbol{\rho}$, due to arrivals. As a result, job arrivals push $\mathbf{w}^{(i)}$ towards the cone C whenever $\mathbf{w}^{(i)} \notin C$.
- Since job size in bucket i is upper bounded by b_i, if there is more than $n_s b_i$ amount of work of each job class in bucket i, then there are enough jobs to fulfill any service option. We show that if, for some bucket i, $\mathbf{w}^{(i)} \in C$ and if $\|\mathbf{w}^{(i)}\|_1$ is sufficiently large, then bucket i is eligible for service. Our policy then ensures that some bucket receives service at full capacity and the total work in the system decreases at rate $1-\rho$ in this scenario.

As a result of these observations, our Lyapunov function is a weighted sum of two terms: a term characterizing the distance to cone C, based on the H_i function defined in Section 6.3, and the total work in the system. □

6.3 Bucket State-space Collapse

Our goal in this section is to prove that the workload $\mathbf{w}^{(i)}$ of each bucket i collapses to the cone C described in Section 6.1. We first prove a generic continuous-time state-space collapse theorem, Theorem 6.2, which may be of independent interest. We them apply this theorem to proving our bucket-based state space collapse, in Theorem 6.3.

We now prove a state-space collapse theorem that can be viewed as a continuous-time equivalent of the classic drift-based discrete-time state-space collapse theorem first introduced in Eryilmaz and Srikant [11]. The challenge here is to do drift analysis in continuous-time. To this end, we employ the Rate Conservation Law [37] to a properly chosen test function.

Theorem 6.2. *Let $\mathbf{W}(t) \in \mathbb{R}_+^n$ be the workload process of a queueing system with Poisson arrivals with rate $\lambda > 0$ with a stationary distribution $\mathbf{W}$. Suppose $V : \mathbb{R}_+^n \to \mathbb{R}_+$ is a differentiable nonnegative-valued function and the following conditions are satisfied:*

(i) There exist $\alpha > 0$, $\beta > 0$, and $K < \infty$ such that for any $\mathbf{W}(t) = \mathbf{w}$,

$$\mathcal{G}V(\mathbf{w}) := D_t V(\mathbf{w}) + \lambda \mathbb{E}[\Delta V(\mathbf{w})] \leqslant -\alpha + \beta \cdot \mathbb{1}(V(\mathbf{w}) \leqslant K)$$

where $\mathcal{G}$ is the infinitesimal generator of the workload process, $D_t V(\mathbf{w})$ is the change in $V(\mathbf{w})$ due to service, and $\Delta V(\mathbf{w}) = V(\mathbf{w}_+) - V(\mathbf{w})$ is the change in V immediately after an arrival at state $\mathbf{W}(t) = \mathbf{w}$.

(ii) There exist $\theta > 0$ and $D < \infty$ such that for all $\mathbf{W}(t) = \mathbf{w}$, $\mathbb{E}\left[e^{\theta|\Delta V(\mathbf{w})|}\right] < D$,

then for any $0 < \eta < \min\left\{\frac{\alpha\theta^2}{\lambda D}, \theta\right\}$, we have

$$\mathbb{E}\left[e^{\eta V(\mathbf{W})}\right] \leqslant \frac{\theta^2 \beta e^{\eta K}}{\theta^2\alpha - \lambda\eta D} < \infty$$

Full proof deferred to Appendix A.2.

Proof outline. We apply the Rate Conservation Law [37] to $e^{\eta(V(\mathbf{W})\wedge n)}$ for some $0 < \eta < \min\left\{\frac{\alpha\theta^2}{\lambda D}, \theta\right\}$ and for a fixed positive integer n such that $\mathbb{E}\left[e^{\eta(V(\mathbf{W})\wedge n)}\right] < \infty$.

Then we bound the resulting terms using conditions (i) and (ii) to obtain a bound on $\mathbb{E}\left[e^{\eta(V(\mathbf{W})\wedge n)}\right]$, using straightforward algebraic manipulations. Finally, we send $n \to \infty$ to obtain the desired result. □

To demonstrate state-space collapse in the sense of Theorem 6.2, we consider the Lyapunov function $H_i(\mathbf{w}) = (\|\mathbf{w}^{(i)}_{\perp\boldsymbol{\rho}}\|_2 - \|\mathbf{w}^{(i)}_{\|\boldsymbol{\rho}}\|_2 \tan\varphi)^+$. Note that $H_i(\mathbf{w}) = 0$ whenever the workload vector $\mathbf{w}^{(i)}$ is in the cone C. Intuitively, H_i measures the distance from the workload vector to C.

Theorem 6.3. *For any bucket i, under our SEB policy, if the assumptions in Section 3.5 are met, then we have the following state-space collapse:*

$$\text{If } H_i(\mathbf{w}) = (\|\mathbf{w}^{(i)}_{\perp\boldsymbol{\rho}}\|_2 - \|\mathbf{w}^{(i)}_{\|\boldsymbol{\rho}}\|_2 \tan\varphi)^+ \text{ and } \eta_i = e^{-2}\frac{b_{i-1}}{b_i^2}\tan\varphi, \text{ then } \mathbb{E}\left[e^{\eta_i H_i(\mathbf{W})}\right] \leqslant \frac{8}{\tan\varphi}\frac{b_i}{b_{i-1}},$$

bounding the distance from $\mathbf{w}_i$ to the cone C.

Proof deferred to Appendix A.3. The proof is a straightforward application of Theorem 6.2, verifying conditions (i) and (ii) in this setting.

7 SEB ANALYSIS: MEAN RESPONSE TIME

7.1 Bounding Waste

In this section, we bound the expected waste $\mathbb{E}[\mathcal{I}_{\leqslant i}W^{\text{SEB}}_{\leqslant i}]$ for an arbitrary bucket i. Our bound is based on our cone-state-space-collapse (SSC) result, Theorem 6.3. We show in this section that whenever the workload vector is in the cone or near cone, waste must be small. From our cone-SSC result, this allows us to bound expected waste. This bound on expected waste directly translates into a bound on expected work in each bucket, giving a bound on waiting time and response time.

Theorem 7.1. *If $\mathbf{w}^{(i)} \in C$ and if $\|\mathbf{w}^{(i)}\|_1$ satisfies*

$$\|\mathbf{w}^{(i)}\|_1 \geqslant \frac{n_s\sqrt{n_c}b_i}{\frac{\boldsymbol{\rho}_{\min}}{\|\boldsymbol{\rho}\|_2}\cos\varphi - \sin\varphi}$$

then any service option in $\mathcal{R}^{\mathcal{F}}$ can be fulfilled using jobs in bucket i.

Proof deferred to Appendix A.4.

We now proceed to bound expected waste $\mathbb{E}[\mathcal{I}_{\leqslant i}W_{\leqslant i}]$, which is the key technical result of the paper. To ease the notation, we define two constants that do not scale with load.

$$\gamma = \frac{n_s\sqrt{n_c}b_{n_b}}{\frac{\boldsymbol{\rho}_{\min}}{\|\boldsymbol{\rho}\|_2}\cos\varphi - \sin\varphi} \quad \text{and} \quad \tau = \frac{\tan\varphi}{\sqrt{n_c}}\left(\frac{\boldsymbol{\rho}_{\min}}{\|\boldsymbol{\rho}\|_2}\cos\varphi - \sin\varphi\right).$$

Theorem 7.2. *Under SEB,*

$$\mathbb{E}[\mathcal{I}_{\leqslant i}W^{\text{SEB}}_{\leqslant i}] \leqslant \left(A_1 + A_2 + A_2\log\left(\frac{1}{1-\rho_{\leqslant i}}\right)\right)(1-\rho_{\leqslant i})\frac{c^i - 1}{c-1} + A_2(1-\rho_{\leqslant i})i$$

$$\text{where} \quad A_1 := \frac{8e^2}{\tau\tan^2\varphi}c^3 b_0 \quad \text{and} \quad A_2 := 2\left(\frac{8}{\tau\tan\varphi}c + 1\right)c\max\left\{\frac{e^2}{\tau\tan\varphi}b_0 c, n_s\frac{b_0}{\tau}, \gamma\right\}$$

Proof. Let $B_{ij} > \gamma$ be numbers that we will later specify, then

$$\begin{aligned}
\mathbb{E}[\mathcal{I}_{\leqslant i} W^{\mathrm{SEB}}_{\leqslant i}] &= \sum_{j=1}^{i} \mathbb{E}[\mathcal{I}_{\leqslant i} \|\mathbf{W}^{(j)}\|_1] \\
&= \sum_{j=1}^{i} \left(\mathbb{E}[\mathcal{I}_{\leqslant i} \|\mathbf{W}^{(j)}\|_1 \mathbb{1}(\|\mathbf{W}^{(j)}\|_1 > B_{ij})] + \mathbb{E}[\mathcal{I}_{\leqslant i} \|\mathbf{W}^{(j)}\|_1 \mathbb{1}(\|\mathbf{W}^{(j)}\|_1 \leqslant B_{ij})] \right) \\
&\overset{(a)}{\leqslant} \sum_{j=1}^{i} \left(\mathbb{E}[\mathcal{I}_{\leqslant j} \|\mathbf{W}^{(j)}\|_1 \mathbb{1}(\|\mathbf{W}^{(j)}\|_1 > B_{ij})] + \mathbb{E}[\mathcal{I}_{\leqslant i} \|\mathbf{W}^{(j)}\|_1 \mathbb{1}(\|\mathbf{W}^{(j)}\|_1 \leqslant B_{ij})] \right) \\
&\overset{(b)}{\leqslant} \sum_{j=1}^{i} \underbrace{\mathbb{E}[\mathcal{I}_{\leqslant j} \|\mathbf{W}^{(j)}\|_1 \mathbb{1}(\|\mathbf{W}^{(j)}\|_1 > B_{ij})]}_{\mathcal{T}} + \sum_{j=1}^{i} B_{ij}(1 - \rho_{\leqslant i})
\end{aligned}$$

where (a) follows from the fact that for any $j \leqslant i$, $\mathcal{I}_{\leqslant i} \leqslant \mathcal{I}_{\leqslant j}$ and (b) follows from the stability of the system. It remains to bound $\mathcal{T}$. First note that according to Theorem 7.1 and the definition of B_{ij}, if the workload vector is in the cone C and the total work is large, there is no waste,

$$\mathbb{E}[\mathcal{I}_{\leqslant j} \|\mathbf{W}^{(j)}\|_1 \mathbb{1}(\|\mathbf{W}^{(j)}\|_1 \geqslant B_{ij}) \mathbb{1}(\mathbf{W}^{(j)} \in C)] = 0$$

because bucket j is eligible for service. This implies that

$$\mathcal{T} = \mathbb{E}[\mathcal{I}_{\leqslant j} \|\mathbf{W}^{(j)}\|_1 \mathbb{1}(\|\mathbf{W}^{(j)}\|_1 \geqslant B_{ij}) \mathbb{1}(\mathbf{W}^{(j)} \notin C)].$$

Define $\mathcal{E}_j = \{\mathbf{w}^{(j)} \in \mathbb{R}^{n_c} : \mathbf{w}^{(j)}_k \geqslant n_s b_i \text{ for all } k\}$ to be the region of workload-space such that if $\mathbf{w}^{(j)}$ is in the region, the system can fully serve all service options using only jobs from bucket j.

We now bound $\mathcal{T}$.

$$\begin{aligned}
\mathcal{T} &\overset{(a)}{\leqslant} \mathbb{E}[\mathbb{1}(\text{bucket } j \text{ ineligible}) \|\mathbf{W}^{(j)}\|_1 \mathbb{1}(\|\mathbf{W}^{(j)}\|_1 \geqslant B_{ij}) \mathbb{1}(\mathbf{W}^{(j)} \notin C)] \\
&\overset{(b)}{\leqslant} \mathbb{E}[\mathbb{1}(\mathbf{W}^{(j)} \notin \mathcal{E}_j) \|\mathbf{W}^{(j)}\|_1 \mathbb{1}(\|\mathbf{W}^{(j)}\|_1 \geqslant B_{ij}) \mathbb{1}(\mathbf{W}^{(j)} \notin C)] \\
&\overset{(c)}{\leqslant} B_{ij} \cdot \mathbb{P}(\{\mathbf{W}^{(j)} \notin \mathcal{E}_j\} \cap \{\|\mathbf{W}^{(j)}\|_1 \geqslant B_{ij})\}) + \int_{u=B_{ij}}^{\infty} \mathbb{P}(\{\mathbf{W}^{(j)} \notin \mathcal{E}_j\} \cap \{\|\mathbf{W}^{(j)}\|_1 \geqslant u)\})\, du \\
&\overset{(d)}{\leqslant} B_{ij} \cdot \mathbb{P}(H_j(\mathbf{W}) \geqslant \tau B_{ij} - n_s b_j) + \int_{u=B_{ij}}^{\infty} \mathbb{P}(H_j(\mathbf{W}) \geqslant \tau u - n_s b_j)\, du \\
&\overset{(e)}{\leqslant} B_{ij} \frac{8}{\tan\varphi} \frac{b_j}{b_{j-1}} e^{\eta_j(n_s b_j - \tau B_{ij})} + \int_{u=B_{ij}}^{\infty} \frac{8}{\tan\varphi} \frac{b_j}{b_{j-1}} e^{-\eta_j(\tau u - n_s b_j)}\, du \\
&= \left(B_{ij} + \frac{1}{\tau \eta_j} \right) \frac{8}{\tan\varphi} \frac{b_j}{b_{j-1}} e^{\eta_j(n_s b_j - \tau B_{ij})}
\end{aligned}$$

Step (a) follows from $\mathcal{I}_{\leqslant j} \leqslant \mathbb{1}(\text{bucket } j \text{ ineligible})$ as a result of the following observations:

- If bucket j is ineligible for service, then the inequality holds because $\mathcal{I}_{\leqslant j} \leqslant 1$.
- If bucket j is eligible for service, then the inequality holds because $\mathcal{I}_{\leqslant j} = 0$.

Step (b) follows from the observation that the set on which bucket j is ineligible is a subset of $\mathcal{E}_j^c$. Step (c) follows from tail-sum formula and the observation that, following from Theorem 7.1 and $B_{ij} > \gamma$, if $\|\mathbf{w}^{(j)}\|_1 \geqslant B_{ij}$ and $\mathbf{w}^{(j)} \notin \mathcal{E}_j$, then $\mathbf{w}^{(j)} \notin C$. Step (d) follows from the following lemma, the proof of which is deferred to Appendix A.5.

Lemma 7.3. *For any* $\mathbf{w}^{(j)} \notin C \cup \mathcal{E}_i$, $H_j(\mathbf{w}) = \|\mathbf{w}^{(j)}_{\perp \boldsymbol{\rho}}\|_2 - \|\mathbf{w}^{(j)}_{\| \boldsymbol{\rho}}\|_2 \tan\varphi \geqslant \tau \|\mathbf{w}^{(j)}\|_1 - n_s b_j$.

Step (e) follows from the state-space collapse result in Theorem 6.3.

We note that $B_{ij} > \gamma$ can be arbitrary. We set B_{ij} as follows for any j such that $j \leqslant i$, giving the following bound on $\mathcal{T}$:

$$B_{ij} = \frac{1}{\tau\eta_j}\log\left(\frac{1}{1-\rho_{\leqslant i}}\right) + \frac{n_s b_j}{\tau} + \gamma \implies \mathcal{T} \leqslant \left(B_{ij} + \frac{1}{\tau\eta_j}\right)\frac{8}{\tan\varphi}\frac{b_j}{b_{j-1}}(1-\rho_{\leqslant i})$$

Since $b_j/b_{j-1} = c$, $b_j = b_0 c^j$ and $\eta_j = e^{-2}\tan\varphi\frac{1}{c}\frac{1}{b_0 c^j}$. We combine all bounds above to obtain

$$\begin{aligned}
\mathbb{E}[\mathcal{I}_{\leqslant i} W^{\text{SEB}}_{\leqslant i}] &\leqslant \sum_{j=1}^{i}\left(B_{ij} + \frac{1}{\tau\eta_j}\right)\frac{8}{\tan\varphi}\frac{b_j}{b_{j-1}}(1-\rho_{\leqslant i}) + (1-\rho_{\leqslant i})\sum_{j=1}^{i} B_{ij} \\
&\leqslant \frac{8e^2}{\tau\tan^2\varphi}c^2 b_0(1-\rho_{\leqslant i})\sum_{j=1}^{i} c^j + \\
&\quad 2\left(\frac{8}{\tau\tan\varphi}c+1\right)(1-\rho_{\leqslant i})\left[\frac{e^2}{\tau\tan\varphi}b_0 c\log\left(\frac{1}{1-\rho_{\leqslant i}}\right)\sum_{j=1}^{i} c^j + n_s\frac{b_0}{\tau}\sum_{j=1}^{i} c^j + i\gamma\right]
\end{aligned}$$

Using A_1 and A_2 defined in the theorem statement, we obtain

$$\mathbb{E}[\mathcal{I}_{\leqslant i} W^{\text{SEB}}_{\leqslant i}] \leqslant \left(A_1 + A_2\log\left(\frac{1}{1-\rho_{\leqslant i}}\right) + A_2\right)(1-\rho_{\leqslant i})\frac{c^i - 1}{c-1} + A_2(1-\rho_{\leqslant i})i. \qquad \square$$

7.2 Bounding Mean Response Time

In this section, we bound the mean response time under SEB. We begin by bounding the mean residence time, which is a straightforward application of Little's law.

Lemma 7.4. *Under SEB,*

$$\mathbb{E}[T^{\text{SEB}}_{res}] \leqslant \frac{1}{\lambda} n_b n_s n_c$$

Proof. By Little's law [23], it suffices for us to bound the mean number of jobs in residence. Since we assumed FCFS for jobs in each class in each bucket, n_s jobs per class per bucket have received service, and thus at most $n_b n_s n_c$ jobs in total have received service. As a result, $\mathbb{E}[T^{\text{SEB}}_{\text{res}}] = \frac{1}{\lambda}\mathbb{E}[N^{\text{SEB}}_{\text{res}}] \leqslant \frac{1}{\lambda} n_b n_s n_c$. $\square$

We now bound mean waiting time $\mathbb{E}[T^{\text{SEB}}_{\text{wait}}]$, which is the dominant component of mean response time in heavy traffic. We first use WINE (Proposition 5.3) to relate mean waiting time to the mean amount of work in each bucket.

Lemma 7.5. *For any policy π that stabilizes the system,*

$$\mathbb{E}[T^{\pi}_{wait}] \leqslant \frac{1}{\lambda}\sum_{i=1}^{n_b}\frac{c-1}{b_0 c^i}\mathbb{E}[W^{\pi}_{\leqslant i}] + \frac{1}{\lambda}\frac{\mathbb{E}[W^{\pi}]}{b_{n_b}}$$

where $W^{\pi}_{\leqslant i}$ is work in the first i buckets and W^{π} is the total work in the system.

Proof deferred to Appendix A.6. Now, we use Lemmas 7.4 and 7.5, as well as the work decomposition law Proposition 5.4 and our bound on waste Theorem 7.2, to bound response time relative to the expected work in the resource pooled M/G/1 at or below each bucket cutoff. To simplify our bound, we apply Lemma 7.5 to the resource pooled PSJF-1 policy, and incorporate a bound on its response time into our bound on *SEB*.

Proposition 7.6. *Under SEB,*

$$\mathbb{E}[T^{\text{SEB}}] \leqslant c\mathbb{E}[T^{\text{PSJF-1}}] + \frac{A}{\lambda}\log\left(\frac{1}{1-\rho}\right)\left(2n_b + \frac{1}{c-1}\right) + \frac{1}{\lambda}n_b n_s n_c$$

where

$$A = \frac{A_1 + 3A_2}{b_0}\frac{b_{n_b} - b_0}{b_0},$$

and where A_1 and A_2 are defined in Theorem 7.2.

Proof deferred to Appendix A.7. Now, we examine Proposition 7.6 in the heavy traffic:

Theorem 5.2. *Under SEB, in the heavy traffic limit and under the assumptions in Section 3.5, mean response time has the following asymptotic behavior:*

$$\mathbb{E}[T^{\text{SEB}}] \leqslant c \cdot \mathbb{E}\left[T^{\text{PSJF-1}}\right] + \Theta\left(\log^2\left(\frac{1}{1-\rho}\right)\right) \quad \text{as } \rho \to 1. \tag{1}$$

where $T^{\text{PSJF-1}}$ is the response time under resource-pooled Preemptive-Shortest-Job-First (PSJF-1), and where c is the bucket width multiplier.

Proof. Recall from Section 4.3 that we set

$$c = 1 + \frac{1}{1 + \log\left(\frac{1}{1-\rho}\right)}.$$

Since $b_{n_b} = b_0 c^{n_b}$, and $b_0 = s_{\min}$ and $b_{n_b} = s_{\max}$ are constants not depending on load, we have

$$n_b = \frac{\log\frac{s_{\max}}{s_{\min}}}{\log c} = \Theta\left(\log\left(\frac{1}{1-\rho}\right)\right) \quad \text{as } \rho \to 1.$$

The theorem follows from Proposition 7.6 by noting that as $\rho \to 1$, c converges 1, and A converges to a fixed constant that does not scale with ρ. □

7.3 Heavy-traffic Optimality

Theorem 7.7. *Under any system load $\rho < 1$, and for any scheduling policy π, $\mathbb{E}[T^{\pi}] \geqslant \mathbb{E}[T^{\text{SRPT-1}}]$.*

Proof. We consider the resource-pooled M/G/1 queue defined in Section 3.4. Since service rate in the original system after the size conversion is no more than 1, any scheduling policy in our original multi-server system with service constraints can be realized in the resource-pooled M/G/1 queue. The result then follows from the optimality of single-server SRPT [45]. □

Theorem 5.1. *Our policy is heavy-traffic optimal, under the assumptions in Section 3.5: For any $\widetilde{\boldsymbol{\nu}}$ on the capacity region that is in the interior of a facet, our SEB policy converges to optimal as the load vector $\widetilde{\boldsymbol{\rho}}$ converges to $\widetilde{\boldsymbol{\nu}}$.*

$$\lim_{\rho\to 1}\frac{\mathbb{E}[T^{\text{SEB}}]}{\mathbb{E}[T^{\text{SRPT-1}}]} = \lim_{\rho\to 1}\frac{\mathbb{E}[T^{\text{SEB}}]}{\mathbb{E}[T^{\text{OPT}}]} = 1,$$

where $T^{\text{SRPT-1}}$ is the response time under resource-pooled SRPT.

Proof. Because the job size distribution is bounded, by Theorem 1 in Lin et al. [32],

$$\mathbb{E}[T^{\text{PSJF-1}}] \geqslant \mathbb{E}[T^{\text{SRPT-1}}] = \Theta\left(\frac{1}{1-\rho}\right)$$

Thus, it follows from Theorem 5.2

$$\lim_{\rho\to 1}\frac{\mathbb{E}[T^{\text{SEB}}]}{\mathbb{E}[T^{\text{PSJF-1}}]} = \lim_{\rho\to 1}\left(c + \frac{\Theta\left(\log^2\left(\frac{1}{1-\rho}\right)\right)}{\mathbb{E}[T^{\text{PSJF-1}}]}\right) = 1$$

The last step in our optimality proof relies on the heavy-traffic optimality of PSJF-1 for bounded job size distributions: $\lim_{\rho\to 1}\frac{\mathbb{E}[T^{\text{PSJF-1}}]}{\mathbb{E}[T^{\text{SRPT-1}}]} = 1$. This is a standard result that follows from the following waiting time inequality: $\mathbb{E}[W^{\text{PSJF-1}}] \leqslant \mathbb{E}[W^{\text{SRPT-1}}]$ [23]. For a detailed proof, see Theorem 5 in Grosof et al. [15]. By Theorem 7.7, our policy is thus heavy traffic optimal. □

8 NUMERICAL EVALUATION

We have proven in Theorem 5.1 that SEB achieves heavy-traffic optimal mean response time in the compatibility scheduling setting, the multiserver-job (MSJ) scheduling setting, and more generally in the general service constraints setting.

In this section, we simulate our SEB policy in a variety of compatibility and multiserver-job settings to empirically validate our heavy-traffic results. We compare SEB against several policies:

MaxWeight-Queue: When a job arrives or departs, the inner products $\langle \widetilde{\mathbf{r}}^{(i)}, \mathbf{q}\rangle$ between available service options $\{\widetilde{\mathbf{r}}^{(i)}\}$ and queue lengths $\mathbf{q}$ (number of jobs in each class) are computed. The scheduler chooses the service option that maximizes this inner product. If there are more jobs in a class than required by the scheduler, jobs are picked in FCFS order.

MaxWeight-Queue SRPT: This is the same as MaxWeight-Queue except that jobs in a class are chosen in SRPT order, where SRPT is defined based on job size, using the bounding facet.

MaxWeight-Work SRPT: When a job arrives or departs, the inner products $\langle \widetilde{\mathbf{r}}^{(i)}, \mathbf{w}\rangle$ between available service options $\{\widetilde{\mathbf{r}}^{(i)}\}$ and work vector $\mathbf{w}$ (remaining work in each class) are computed. The scheduler chooses the service option that maximizes this inner product. Jobs in the same class are picked in SRPT order.

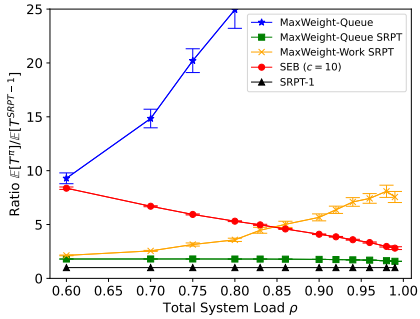
ServerFillingSRPT: This is an MSJ-specific policy [18]. Jobs are sorted in least remaining size order. The candidate set is defined to consist of the minimal initial sequence of jobs with total server need at least n_s, and jobs are served in most-server-need-first order among the candidate set, tie-broken by lower remaining size.

In our simulations, for simplicity, we only update a policy's service option at moments when arrivals and completions occur. This affects the SEB and MaxWeight-Work SRPT policies. Our initial exploration indicated that this did not have an appreciable impact on response times.

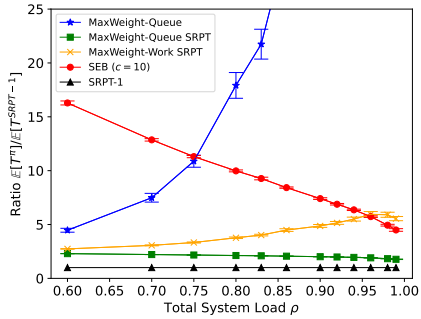
The MaxWeight-based policies are known to achieve optimal stability region, but their heavy-traffic optimality is open. ServerFilling-SRPT achieves heavy-traffic when all server needs are powers of two, but it has not been analyzed in the more general MSJ settings that we consider here.

8.1 Evaluation of compatibility scheduling

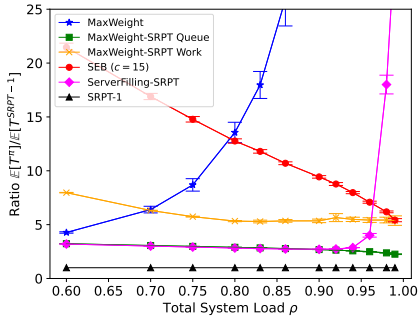
We start with a simple compatibility scheduling setting, the 2-server N setting, depicted in Fig. 2a. There are two servers and two classes of jobs. Jobs of class 1 can be served only by server 1 and jobs of class 2 can be served by either servers 1 or 2. The service options are $[2, 0]$ and $[1, 1]$.



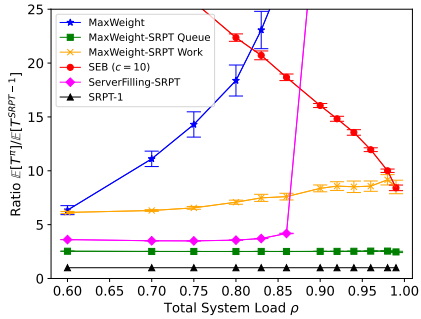
(a) Two-class compatibility model. Jobs are class 1 with probability 1/4 or class 2 with probability 3/4.



(b) Three-class compatibility model. Jobs are class 1 with probability 1/3, class 2 with probability 1/3, and class 3 with probability 1/3.

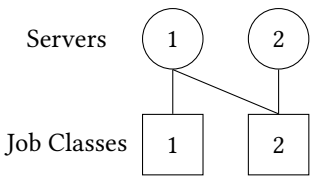


(c) Two-class MSJ model. Jobs are class 1 with probability 5/11 or class 2 with probability 6/11.

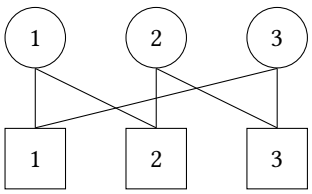


(d) Three-class MSJ model. Jobs are class 1 with probability 1/6, class 2 with probability 1/6, or class 3 with probability 4/6.

Fig. 2. We simulate our SEB policy, as well as our comparison policies, for a Bounded Pareto job size distribution with $C^2 = 99$, a relatively high-variance bounded job size distribution. 30 trials for each data point, with at least 2×10^6 jobs served for each load. 95% confidence intervals shown.



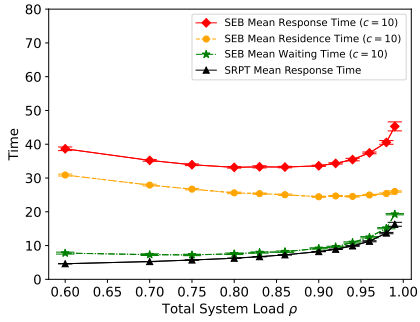
(a) The two-class compatibility model



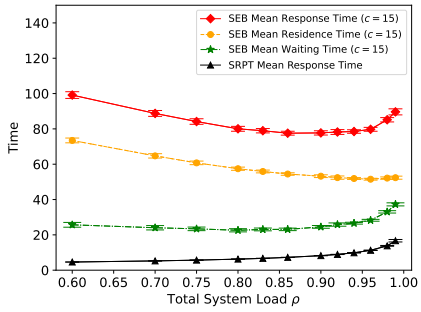
(b) The three-class compatibility model

Our simulation results for this setting are shown in Fig. 2a. For each simulated policy π, we evaluated the ratio $\mathbb{E}[T^{\pi}]/\mathbb{E}[T^{\text{SRPT-1}}]$. Recall that our heavy traffic result shows that this ratio converges to 1 as load ρ goes to 1.

Our results indicate that the SEB policy's ratio is descending throughout the range of loads simulated, in concert with our theoretical results. In contrast, the MaxWeight-Queue and MaxWeight-Work SRPT policies have growing ratios throughout the range of loads simulated, indicating an absence of heavy traffic optimality. The MaxWeight-Queue SRPT policy has a strong ratio throughout the range of loads simulated, and we discuss why we believe this occurs in Appendix B. Note



(a) Two-class compatibility model



(b) Two-class MSJ model

Fig. 3. Comparisons of mean residence and waiting times under SEB with the mean response time of SRPT-1

that we used a fixed bucket width multiplier $c = 10$ for all loads, to simplify the simulations, and because load was not high enough for varying the multiplier to improve the response time.

These results suggest a confirmation of SEB's heavy traffic optimality in this setting, though much higher loads would be needed for a full confirmation, which would require more simulation than we were able to conduct. To further investigate SEB's response time behavior, we separate its mean response time into mean residence time, and mean waiting time.

In Fig. 3a, we see that at all loads simulated, $\mathbb{E}[T^{SEB}]$ is dominated by residence time, rather than waiting time. This corresponds to the fact that SEB is an idling policy, so jobs may remain in residence longer than they would in non-idling policies. However, as can be seen in this figure, residence time is not increasing with load, and hence becomes negligible in heavy traffic. Waiting time dominates in heavy traffic, and our mean waiting time is closely comparable to the mean response time of SRPT-1, further validating our convergence to heavy traffic optimality.

We also evaluate our SEB policy and the comparison policies in a more complex compatibility scheduling setting, shown in Fig. 2b, with 3 servers, 3 job classes, and each server capable of serving two classes of jobs. Again, we evaluated the ratio $\mathbb{E}[T^{\pi}]/\mathbb{E}[T^{\text{SRPT-1}}]$ for each policy π.

In this setting, we likewise see that SEB's ratio falls across the entire range of loads considered, while MaxWeight-Queue and MaxWeight-SRPT Work have increasing ratios, and MaxWeight-SRPT Queue performs well across a variety of loads. Again, SEB's pre-heavy-traffic behavior is suggestive of its heavy traffic optimality.

8.2 Evaluation of multiserver-job scheduling

In the multiserver-job (MSJ) setting, we consider a setting with 11 servers, and two server need possibilities: One where jobs have server need either 2 or 3, and the dominating facet is bounded by the points $[4, 1]$ and $[1, 3]$, where both service options use all 11 servers. The other server need possibility has needs 2, 3, and 5, and we select a job size distribution with dominating facet bounded by the points $[3, 0, 1][0, 2, 1]$, and $[0, 0, 2]$. Note that not all of these service options use all 11 servers: $[0, 0, 2]$ uses only 10. This leads to different size-duration conversion coefficients, and our goal is to validate our results in this distinct setting.

Note that neither of these MSJ settings fall in the ServerFilling/DivisorFilling setting, as neither server need perfectly divides 11 [17, 18].

Our simulation results are shown in Figs. 2c and 2d. For each simulated policy π, we evaluated the ratio $\mathbb{E}[T^{\pi}]/\mathbb{E}[T^{\text{SRPT-1}}]$. Our empirical results indicate that in the MSJ setting, as in the prior compatibility scheduling setting, SEB's response time ratio falls consistently throughout the load

range simulated, adding validity to our results, Theorem 5.1, which states that our mean response time ratio converges to 1 as load $\rho \to 1$.

MaxWeight continues to perform poorly, and MaxWeight-SRPT Queue continues to perform well. MaxWeight-SRPT Work shows a roughly flat response time ratio across the range of loads simulated, increasing somewhat in Fig. 2d.

The ServerFilling-SRPT policy performs well for low load, but reaches the boundary of its stability region at load ρ significantly below 1, as it is not a throughput-optimal policy in this setting, and it experiences a rapid rise in response time ratio as a result.

Again, in Fig. 3b, we break SEB's mean response time down into its components, mean waiting time and mean residence time. Again, residence time does not increase asymptotically as we approach heavy traffic ($\rho \to 1$). Only waiting time increases in that limit. SEB's waiting time again parallels that of SRPT-1, lending credibility to our theoretical result of heavy traffic optimality.

9 CONCLUSION

In this paper, we design a new scheduling policy, called SEB (Smallest Equalizing Bucket) and show that it is heavy-traffic optimal, thus making SEB the first proven heavy-traffic optimal scheduling policy in multi-server systems with general service constraints. SEB strikes the right balance between prioritizing small jobs and keeping all servers busy, a critical component to optimality that no previous policies are able to achieve.

Despite the success of SEB, we have left in place a few assumptions. Most importantly, we have assumed bounded job size distribution and independence of job size and job class. Designing and analyzing a good policy without these assumptions is important both in theory and in practice. Our explorations suggest that substantial new ideas must be introduced to achieve optimality in the most general setting. We leave this to future work.

REFERENCES

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. TensorFlow: a system for Large-Scale machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pages 265–283, 2016.

[2] I. Adan and G. Weiss. A loss system with skill-based servers under assign to longest idle server policy. *Probability in the Engineering and Informational Sciences*, 26(3):307–321, 2012.

[3] I. Adan and G. Weiss. A skill based parallel service system under FCFS-ALIS—steady state, overloads, and abandonments. *Stochastic Systems*, 4(1):250–299, 2014.

[4] L. Afanaseva, E. Bashtova, and S. Grishunina. Stability analysis of a multi-server model with simultaneous service and a regenerative input flow. *Methodology and Computing in Applied Probability*, 22:1439–1455, 2020.

[5] S. Banerjee, A. Budhiraja, and A. L. Puha. Heavy traffic scaling limits for shortest remaining processing time queues with heavy tailed processing time distributions. *The Annals of Applied Probability*, 32(4):2587–2651, 2022.

[6] P. H. Brill and L. Green. Queues in which customers receive simultaneous service from a random number of servers: a system point approach. *Management Science*, 30(1):51–68, 1984.

[7] A. Brondsted. *An introduction to convex polytopes*, volume 90. Springer Science & Business Media, 2012.

[8] D. Carastan-Santos, R. Y. De Camargo, D. Trystram, and S. Zrigui. One can only gain by replacing EASY Backfilling: A simple scheduling policies case study. In *2019 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, pages 1–10. IEEE, 2019.

[9] Z. Chen, I. Grosof, and B. Berg. Simple policies for multiresource job scheduling. *MAthematical performance Modeling and Analysis (MAMA)*, 2024.

[10] W. Dai and S. Wang. Optimal control based on a general exponential scheduling rule for a generalized switch. In *2009 WRI International Conference on Communications and Mobile Computing*, volume 2, pages 332–336, 2009. doi: 10.1109/CMC.2009.144.

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

[12] D. Filippopoulos and H. Karatza. An M/M/2 parallel system model with pure space sharing among rigid jobs. *Mathematical and Computer Modelling*, 45(5-6):491–530, 2007.

[13] K. Gardner and R. Righter. Product forms for FCFS queueing models with arbitrary server-job compatibilities: an overview. *Queueing Systems*, 96(1):3–51, 2020.
[14] J. Ghaderi. Randomized algorithms for scheduling VMs in the cloud. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9. IEEE, 2016.
[15] I. Grosof, Z. Scully, and M. Harchol-Balter. SRPT for multiserver systems. *Performance Evaluation*, 127(128):154–175, 2018.
[16] I. Grosof, Z. Scully, and M. Harchol-Balter. Load balancing guardrails: Keeping your heavy traffic on the road to low response times. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 3(2):1–31, 2019.
[17] I. Grosof, M. Harchol-Balter, and A. Scheller-Wolf. WCFS: A new framework for analyzing multiserver systems. *Queueing Systems*, 102(1):143–174, 2022.
[18] I. Grosof, Z. Scully, M. Harchol-Balter, and A. Scheller-Wolf. Optimal scheduling in the multiserver-job model under heavy traffic. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 6(3):1–32, 2022.
[19] I. Grosof, Y. Hong, M. Harchol-Balter, and A. Scheller-Wolf. The RESET and MARC techniques, with application to multiserver-job analysis. *Performance Evaluation*, 162:102378, 2023.
[20] M. Guo, Q. Guan, and W. Ke. Optimal scheduling of VMs in queueing cloud computing systems with a heterogeneous workload. *IEEE Access*, 6:15178–15191, 2018.
[21] B. Hajek. Hitting-time and occupation-time bounds implied by drift analysis with applications. *Advances in Applied probability*, 14(3):502–525, 1982.
[22] B. Haji and S. M. Ross. A queueing loss model with heterogeneous skill based servers under idle time ordering policies. *Journal of Applied Probability*, 52(1):269–277, 2015.
[23] M. Harchol-Balter. *Performance modeling and design of computer systems: queueing theory in action*. Cambridge University Press, 2013.
[24] M. Harchol-Balter. The multiserver job queueing model. *Queueing Systems*, 100(3):201–203, 2022.
[25] J. M. Harrison and M. J. López. Heavy traffic resource pooling in parallel-server systems. *Queueing systems*, 33:339–368, 1999.
[26] Y. Hong and W. Wang. Sharp waiting-time bounds for multiserver jobs. In *Proceedings of the Twenty-Third International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, MobiHoc '22, page 161–170, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391658. doi: 10.1145/3492866.3549717. URL https://doi.org/10.1145/3492866.3549717.
[27] D. A. Hurtado Lange and S. T. Maguluri. Heavy-traffic analysis of queueing systems with no complete resource pooling. *Mathematics of Operations Research*, 47(4):3129–3155, 2022. doi: 10.1287/moor.2021.1248.
[28] P. R. Jhunjhunwala and S. T. Maguluri. Low-complexity switch scheduling algorithms: Delay optimality in heavy traffic. *IEEE/ACM Transactions on Networking*, 30(1):464–473, 2022. doi: 10.1109/TNET.2021.3116606.
[29] T. Ji, E. Athanasopoulou, and R. Srikant. On optimal scheduling algorithms for small generalized switches. *IEEE/ACM Transactions on Networking*, 18(5):1585–1598, 2010. doi: 10.1109/TNET.2010.2045394.
[30] J. P. Jones and B. Nitzberg. Scheduling for parallel supercomputing: A historical perspective of achievable utilization. In *Workshop on job scheduling strategies for parallel processing*, pages 1–16. Springer, 1999.
[31] S. S. Kim. *M/M/s queueing system where customers demand multiple server use*. Southern Methodist University, 1979.
[32] M. Lin, A. Wierman, and B. Zwart. Heavy-traffic analysis of mean response time under shortest remaining processing time. *Performance Evaluation*, 68(10):955–966, 2011.
[33] S. T. Maguluri and R. Srikant. Scheduling jobs with unknown duration in clouds. *IEEE/ACM Transactions on Networking*, 22(6):1938–1951, 2014.
[34] S. T. Maguluri, R. Srikant, and L. Ying. Stochastic models of load balancing and scheduling in cloud computing clusters. In *2012 Proceedings IEEE Infocom*, pages 702–710. IEEE, 2012.
[35] S. T. Maguluri, R. Srikant, and L. Ying. Heavy traffic optimal resource allocation algorithms for cloud computing clusters. *Performance Evaluation*, 81:20–39, 2014.
[36] S. P. Meyn and R. L. Tweedie. Stability of markovian processes III: Foster–Lyapunov criteria for continuous-time processes. *Advances in Applied Probability*, 25(3):518–548, 1993.
[37] M. Miyazawa. Rate conservation laws: a survey. *Queueing Systems*, 15:1–58, 1994.
[38] E. Morozov and A. Rumyantsev. Stability analysis of a MAP/M/s cluster model by matrix-analytic method. In *Computer Performance Engineering: 13th European Workshop, EPEW 2016, Chios, Greece, October 5-7, 2016, Proceedings 13*, pages 63–76. Springer, 2016.
[39] D. Mukherjee, S. C. Borst, and J. S. Van Leeuwaarden. Asymptotically optimal load balancing topologies. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 2(1):1–29, 2018.
[40] D. Narayanan, M. Shoeybi, J. Casper, P. LeGresley, M. Patwary, V. Korthikanti, D. Vainbrand, P. Kashinkunti, J. Bernauer, B. Catanzaro, et al. Efficient large-scale language model training on gpu clusters using megatron-lm. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–15, 2021.

[41] K. Psychas and J. Ghaderi. Randomized algorithms for scheduling multi-resource jobs in the cloud. *IEEE/ACM Transactions on Networking*, 26(5):2202–2215, 2018.
[42] R. Righter, J. G. Shanthikumar, and G. Yamazaki. On extremal service disciplines in single-stage queueing systems. *Journal of Applied Probability*, 27(2):409–416, 1990.
[43] A. Rumyantsev and E. Morozov. Stability criterion of a multiserver model with simultaneous service. *Annals of Operations Research*, 252:29–39, 2017.
[44] D. Rutten and D. Mukherjee. Load balancing under strict compatibility constraints. *Mathematics of Operations Research*, 48(1):227–256, 2023.
[45] L. Schrage. A proof of the optimality of the shortest remaining processing time discipline. *Operations Research*, 16(3): 687–690, 1968.
[46] Z. Scully. *A New Toolbox for Scheduling Theory*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, Aug. 2022. URL https://ziv.codes/pdf/scully-thesis.pdf.
[47] Z. Scully, I. Grosof, and M. Harchol-Balter. The gittins policy is nearly optimal in the M/G/k under extremely general conditions. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 4(3):1–29, 2020.
[48] M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.
[49] A. L. Stolyar. MaxWeight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic. *The Annals of Applied Probability*, 14(1):1 – 53, 2004. doi: 10.1214/aoap/1075828046. URL https://doi.org/10.1214/aoap/1075828046.
[50] J. N. Tsitsiklis and K. Xu. Queueing system topologies with limited flexibility. In *Proceedings of the ACM SIGMETRICS/international conference on Measurement and modeling of computer systems*, pages 167–178, 2013.
[51] J. Visschers, I. Adan, and G. Weiss. A product form solution to a system with multi-type jobs and multi-type servers. *Queueing Systems*, 70(3):269–298, 2012.
[52] J. Wang and W. Guo. The application of backfilling in cluster systems. In *2009 WRI International Conference on Communications and Mobile Computing*, volume 3, pages 55–59. IEEE, 2009.
[53] W. Wang, K. Zhu, L. Ying, J. Tan, and L. Zhang. Maptask scheduling in mapreduce with data locality: Throughput and heavy-traffic optimality. *IEEE/ACM Transactions On Networking*, 24(1):190–203, 2014.
[54] W. Weng, X. Zhou, and R. Srikant. Optimal load balancing with locality constraints. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 4(3):1–37, 2020.
[55] Q. Xie, A. Yekkehkhany, and Y. Lu. Scheduling with multi-level data locality: Throughput and heavy-traffic optimality. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9. IEEE, 2016.
[56] R. Xie, K. Gardner, and R. Righter. Insensitivity for loss systems with compatibilities. *ACM SIGMETRICS Performance Evaluation Review*, 51(2):12–14, 2023.
[57] N. Zychlinski, C. W. Chan, and J. Dong. Managing queues with different resource requirements. *Operations Research*, 71(4):1387–1413, 2023. doi: 10.1287/opre.2022.2284.

A DEFERRED PROOFS

A.1 Proof of Theorem 6.1

THEOREM 6.1. *The system is stable under SEB for any $\rho < 1$.*

PROOF. Our main tool is the continuous-time Foster Lyapunov Theorem in [36]. The key idea is to find a nonnegative Lyapunov function V that has bounded drift on a compact set $\mathcal{K}$ and negative drift outside $\mathcal{K}$.

Since SEB balances all buckets separately and induces a cone-based state-space collapse in each bucket, we show stability in the space $\mathbb{R}_+^{n_b \times n_c}$ which contains all possible values of bucket work vectors $\{\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(n_b)}\}$. For simplicity, we will use $\{\mathbf{w}^{(i)}\}$ to denote this collection of bucket work vectors and $\|\mathbf{w}\|_1$ to denote the total work in the system.

Consider the following Lyapunov function:

$$V(\{\mathbf{w}^{(i)}\}) = \sum_{i=1}^{n_b} H(\mathbf{w}^{(i)})^2 + M\|\mathbf{w}\|_1$$

where the function H and the constant M are defined as follows:

$$H(\mathbf{w}^{(i)}) := \left(\|\mathbf{w}^{(i)}_{\perp\boldsymbol{\rho}}\|_2 - \|\mathbf{w}^{(i)}_{\|\boldsymbol{\rho}}\|_2 \tan\varphi\right)^+, \qquad M := \frac{2\lambda n_b \sqrt{n_s} s_{\max}^2}{1-\rho}$$

We define the following compact set:

$$\mathcal{K} = \bigcap_{i=1}^{n_b} \{\mathbf{w}^{(i)} : H(\mathbf{w}^{(i)}) \leqslant C_i\} \cap \{\mathbf{w} : \|\mathbf{w}\|_1 \leqslant D\}$$

where

$$C_i := \frac{3\lambda n\sqrt{n}\max\{s_{\max}^3, 1\} n_b + M + 1}{\|\boldsymbol{\rho}^{(i)}\|_2 \tan\varphi}$$

$$D := \max\left\{\sqrt{n_c}\left(\frac{n_s b_{n_b} + \max_j C_j}{\frac{\rho_{\min}}{\|\boldsymbol{\rho}\|_2} - \tan\varphi} + \max_j C_j\right), \frac{n_s\sqrt{n_c} b_{n_b}}{\frac{\rho_{\min}}{\|\boldsymbol{\rho}\|_2}\cos\varphi - \sin\varphi}\right\}$$

Our goal is to show that the drift of the Lyapunov function V outside $\mathcal{K}$ is negative. We consider the following cases:

(1) Bucket i is in service for some $i \in \{1, \dots, n_b\}$.
(2) $H(\mathbf{w}^{(i)}) > C_i$ for some $i \in \{1, \dots, n_b\}$ and no bucket is in service.
(3) $\|\mathbf{w}\|_1 > D$ and $\mathbf{w} \in \bigcap_{i=1}^{n_b} \{\mathbf{w}^{(i)} : H(\mathbf{w}^{(i)}) \leqslant C\}$. In this case, we will show that some bucket must be eligible for service and we are thus back to case 1.

We now analyze each of the three cases separately.

Case 1(a): $\mathbf{w}^{(i)} \notin C$. Let $\mathbf{r}^*$ be the service vector chosen by the scheduler.

$$\begin{aligned}
\mathcal{G}V(\{\mathbf{w}^{(i)}\}) &\overset{(a)}{\leqslant} -\left\langle \nabla H(\mathbf{w}^{(i)})^2, \mathbf{r}^*\right\rangle - M(\|\mathbf{r}^*\|_1 - \|\boldsymbol{\rho}\|_1) + \lambda n_b n_c s_{\max}^2 \\
&\overset{(b)}{=} -\left\langle \nabla H(\mathbf{w}^{(i)})^2, \mathbf{r}^*\right\rangle + \lambda n_b n_c s_{\max}^2 - M(1-\rho) \\
&\overset{(c)}{=} -2\left\langle \mathbf{w}^{(i)}_{\perp\boldsymbol{\rho}}, \mathbf{r}^*\right\rangle + 2\tan\varphi\left(\frac{\left\langle \mathbf{w}^{(i)}_{\perp\boldsymbol{\rho}}, \mathbf{r}^*\right\rangle}{\|\mathbf{w}^{(i)}_{\perp\boldsymbol{\rho}}\|_2}\|\mathbf{w}^{(i)}_{\|\boldsymbol{\rho}}\|_2 + \frac{\left\langle \boldsymbol{\rho}, \mathbf{r}^*\right\rangle}{\|\boldsymbol{\rho}\|_2}\|\mathbf{w}^{(i)}_{\perp\boldsymbol{\rho}}\|_2\right) - 2\frac{\left\langle \boldsymbol{\rho}, \mathbf{r}^*\right\rangle}{\|\boldsymbol{\rho}\|_2}\|\mathbf{w}^{(i)}_{\|\boldsymbol{\rho}}\|_2\tan^2\varphi \\
&\quad + \lambda n_b n_c s_{\max}^2 - M(1-\rho) \\
&= -2(\|\mathbf{w}^{(i)}_{\perp\boldsymbol{\rho}}\|_2 - \|\mathbf{w}^{(i)}_{\|\boldsymbol{\rho}}\|_2\tan\varphi)\left(\frac{\left\langle \mathbf{w}^{(i)}_{\perp\boldsymbol{\rho}}, \mathbf{r}^*\right\rangle}{\|\mathbf{w}^{(i)}_{\perp\boldsymbol{\rho}}\|_2} - \frac{\left\langle \boldsymbol{\rho}, \mathbf{r}^*\right\rangle}{\|\boldsymbol{\rho}\|_2}\tan\varphi\right) + \lambda n_b n_c s_{\max}^2 - M(1-\rho) \\
&< \lambda n_b n_c s_{\max}^2 - M(1-\rho)
\end{aligned}$$

where (a) follows from

$$\lambda \sum_{i=1}^{n_b} \mathbb{E}\left(H(\mathbf{w}^{(i)} + \mathbf{S}^{(i)})^2 - H(\mathbf{w}^{(i)})^2\right) \leqslant \lambda n_b n_c s_{\max}^2$$

(b) follows from the fact that $\mathbf{r}^*$ processes total work at rate 1. (c) follows from the following calculations:

$$\nabla_{\mathbf{w}^{(i)}}\|\mathbf{w}^{(i)}_{\perp\boldsymbol{\rho}}\|_2^2 = 2\mathbf{w}^{(i)}_{\perp\boldsymbol{\rho}}, \qquad \nabla_{\mathbf{w}^{(i)}}\|\mathbf{w}^{(i)}_{\perp\boldsymbol{\rho}}\|_2 = \frac{\mathbf{w}^{(i)}_{\perp\boldsymbol{\rho}}}{\|\mathbf{w}^{(i)}_{\perp\boldsymbol{\rho}}\|_2}$$

$$\nabla_{\mathbf{w}^{(i)}}\|\mathbf{w}^{(i)}_{\|\boldsymbol{\rho}}\|_2^2 = 2\frac{\boldsymbol{\rho}}{\|\boldsymbol{\rho}\|_2}\|\mathbf{w}^{(i)}_{\|\boldsymbol{\rho}}\|_2, \quad \nabla_{\mathbf{w}^{(i)}}\|\mathbf{w}^{(i)}_{\|\boldsymbol{\rho}}\|_2 = \frac{\boldsymbol{\rho}}{\|\boldsymbol{\rho}\|_2}$$

Case 1(b): $\mathbf{w}^{(i)} \in C$. Let $\mathbf{r}^*$ be the service vector chosen by the scheduler.

$$\mathcal{G}V(\{\mathbf{w}^{(i)}\}) \leqslant -M(\|\mathbf{r}^*\|_1 - \|\boldsymbol{\rho}\|_1) + \lambda n_b n_c s_{\max}^2 < \lambda n_b n_c s_{\max}^2 - M(1-\rho)$$

Case 2:

$$\begin{aligned}
\mathcal{G}V(\{\mathbf{w}^{(i)}\}) &= \lambda \mathbb{E}[\Delta V(\{\mathbf{w}^{(i)}\})] \\
&= \lambda \sum_{j=1}^{n_b} \mathbb{E}[(H(\mathbf{w}^{(j)} + \mathbf{S}^{(j)})^2 - H(\mathbf{w}^{(j)})^2)] + \lambda M \mathbb{E}[\|\mathbf{S}\|_1] \\
&\leqslant \lambda(\mathbb{E}[H(\mathbf{w}^{(i)} + \mathbf{S}^{(i)})^2 - H(\mathbf{w}^{(i)})^2)]) + \lambda \sum_{j \neq i} \mathbb{E}[H(\mathbf{w}^{(j)} + \mathbf{S}^{(j)})^2 - H(\mathbf{w}^{(j)})^2)] + M\|\boldsymbol{\rho}\|_1 \\
&\leqslant \underbrace{\lambda(\mathbb{E}[H(\mathbf{w}^{(i)} + \mathbf{S})^2 - H(\mathbf{w}^{(i)})^2)])}_{\mathcal{T}_1} + \lambda(n_b - 1) n_c s_{\max}^2 + M
\end{aligned}$$

We now bound $\mathcal{T}_1$.

$$\begin{aligned}
\mathcal{T}_1 &= \lambda \mathbb{E}\left[\left(H(\mathbf{w}^{(i)} + \mathbf{S}^{(i)}) + H(\mathbf{w}^{(i)})\right)\left(H(\mathbf{w}^{(i)} + \mathbf{S}^{(i)}) - H(\mathbf{w}^{(i)})\right)\right] \\
&\leqslant (2\|\mathbf{w}_{\perp\boldsymbol{\rho}}^{(i)}\|_2 + \sqrt{n_c} s_{\max}) \lambda \mathbb{E}\left[H(\mathbf{w}^{(i)} + \mathbf{S}) - H(\mathbf{w}^{(j)})\right] \\
&\overset{(a)}{\leqslant} (2\|\mathbf{w}_{\perp\boldsymbol{\rho}}^{(i)}\|_2 + \sqrt{n_c} s_{\max}) \left(\frac{\lambda \mathbb{E}[\|\mathbf{S}_{\perp\boldsymbol{\rho}}^{(i)}\|_2^2]}{2\|\mathbf{w}_{\perp\boldsymbol{\rho}}^{(i)}\|_2} - \|\boldsymbol{\rho}^{(i)}\|_2 \tan\varphi\right) \\
&\leqslant (2\|\mathbf{w}_{\perp\boldsymbol{\rho}}^{(i)}\|_2 + \sqrt{n_c} s_{\max}) \left(\frac{\lambda n_c s_{\max}^2}{2\|\mathbf{w}_{\perp\boldsymbol{\rho}}^{(i)}\|_2} - \|\boldsymbol{\rho}^{(i)}\|_2 \tan\varphi\right)
\end{aligned}$$

where (a) follows from the proof of Theorem 6.3. Therefore, for

$$\|\mathbf{w}_{\perp\boldsymbol{\rho}}^{(i)}\|_2 \geqslant \frac{3\lambda n_c \sqrt{n_c} \max\{s_{\max}^3, 1\} n_b + M + 1}{\|\boldsymbol{\rho}^{(i)}\|_2 \tan\varphi} = C_i,$$

we have $\mathcal{G}V\{\mathbf{w}^{(i)}\}) < 0$

Case 3: In this case, there exists $j \in \{1, \ldots, n_b\}$ such that $\|\mathbf{w}^{(j)}\|_1 \geqslant D/n_b$. We now show that it must be the case that bucket j is eligible for service, hence this case is reduced to case 1. If $\mathbf{w}^{(j)} \notin C$, we have $\|\mathbf{w}_{\perp\boldsymbol{\rho}}^{(j)}\|_2 \leqslant C_j + \|\mathbf{w}_{\|\boldsymbol{\rho}}^{(j)}\|_2 \tan\varphi$. Since $\mathbf{w}^{(j)} = \mathbf{w}_{\|\boldsymbol{\rho}}^{(j)} + \mathbf{w}_{\perp\boldsymbol{\rho}}^{(j)}$, letting $\mathbf{e}_i$ be the i-th standard basis in $\mathbb{R}^{n_c}$, we have

$$\begin{aligned}
\left\langle \mathbf{w}^{(j)}, \mathbf{e}_i \right\rangle &= \left\langle \mathbf{w}_{\|\boldsymbol{\rho}}^{(j)}, \mathbf{e}_i \right\rangle + \left\langle \mathbf{w}_{\perp\boldsymbol{\rho}}^{(j)}, \mathbf{e}_i \right\rangle \\
&\geqslant \frac{\left\langle \mathbf{w}^{(j)}, \boldsymbol{\rho} \right\rangle}{\|\boldsymbol{\rho}\|_2^2} \boldsymbol{\rho}_i - \|\mathbf{w}_{\perp\boldsymbol{\rho}}^{(j)}\|_2 \\
&\geqslant \frac{\left\langle \mathbf{w}^{(j)}, \boldsymbol{\rho} \right\rangle}{\|\boldsymbol{\rho}\|_2^2} \boldsymbol{\rho}_i - C_j - \frac{\left\langle \mathbf{w}^{(j)}, \boldsymbol{\rho} \right\rangle}{\|\boldsymbol{\rho}\|_2^2} \|\boldsymbol{\rho}\|_2 \tan\varphi \\
&= \frac{\left\langle \mathbf{w}^{(j)}, \boldsymbol{\rho} \right\rangle}{\|\boldsymbol{\rho}\|_2^2} (\boldsymbol{\rho}_i - \|\boldsymbol{\rho}\|_2 \tan\varphi) - C_j \\
&= \|\mathbf{w}_{\|\boldsymbol{\rho}}^{(j)}\|_2 \left(\frac{\boldsymbol{\rho}_i}{\|\boldsymbol{\rho}\|_2} - \tan\varphi\right) - C_j
\end{aligned}$$

By norm inequality,

$$\|\mathbf{w}^{(j)}_{\|\boldsymbol{\rho}}\|_2 \geqslant \|\mathbf{w}^{(j)}\|_2 - \|\mathbf{w}^{(j)}_{\perp\boldsymbol{\rho}}\|_2 \geqslant \frac{1}{\sqrt{n_c}}\|\mathbf{w}^{(j)}\|_1 - C_j - \|\mathbf{w}^{(j)}_{\|\boldsymbol{\rho}}\|_2 \tan\varphi$$

The following bound follows:

$$\|\mathbf{w}^{(j)}_{\|\boldsymbol{\rho}}\|_2 \geqslant \frac{1}{1+\tan\varphi}\left(\frac{1}{\sqrt{n_c}}\|\mathbf{w}^{(j)}\|_1 - C_j\right)$$

Therefore, for

$$\|\mathbf{w}^{(j)}\|_1 \geqslant \sqrt{n_c}\left(\frac{n_s b_j + C_j}{\frac{\rho_{\min}}{\|\boldsymbol{\rho}\|_2} - \tan\varphi} + C_j\right)$$

we have $\mathbf{w}_i^{(j)} \geqslant n_s b_j$, which implies that the j-th bucket is eligible for service because there are at least n_s jobs in the bucket.

If $\mathbf{w}^{(j)} \in C$, then by Theorem 7.1, bucket j is eligible for service if

$$\|\mathbf{w}^{(j)}\|_1 \geqslant \frac{n_s\sqrt{n_c}b_j}{\frac{\rho_{\min}}{\|\boldsymbol{\rho}\|_2}\cos\varphi - \sin\varphi}$$

□

A.2 Proof of Theorem 6.2

The following lemma, which is proved in Hajek [21], is foundational to discrete-time state-space collapse and will play a critical role in our proof for continuous-time state-space collapse.

Lemma A.1 (Hajek [21] Lemma 2.2). *Suppose that X and Z are random variables such that $|X| \leqslant_{st} Z$ and $\mathbb{E}[e^{\lambda Z}] < \infty$ for some $\lambda > 0$. Then for $0 \leqslant \varepsilon \leqslant \lambda$,*

$$\mathbb{E}[e^{\varepsilon X}] \leqslant 1 + \varepsilon\mathbb{E}[X] + \varepsilon^2 c \tag{2}$$

where c is given by

$$c = \sum_{k=2}^{\infty} \frac{\lambda^{k-2}}{k!}\mathbb{E}[Z^k]$$

Theorem 6.2. *Let $\mathbf{W}(t) \in \mathbb{R}_+^n$ be the workload process of a queueing system with Poisson arrivals with rate $\lambda > 0$ with a stationary distribution $\mathbf{W}$. Suppose $V : \mathbb{R}_+^n \to \mathbb{R}_+$ is a differentiable nonnegative-valued function and the following conditions are satisfied:*

(i) There exist $\alpha > 0$, $\beta > 0$, and $K < \infty$ such that for any $\mathbf{W}(t) = \mathbf{w}$,

$$\mathcal{G}V(\mathbf{w}) := D_t V(\mathbf{w}) + \lambda\mathbb{E}[\Delta V(\mathbf{w})] \leqslant -\alpha + \beta \cdot \mathbb{1}(V(\mathbf{w}) \leqslant K)$$

where $\mathcal{G}$ is the infinitesimal generator of the workload process, $D_t V(\mathbf{w})$ is the change in $V(\mathbf{w})$ due to service, and $\Delta V(\mathbf{w}) = V(\mathbf{w}_+) - V(\mathbf{w})$ is the change in V immediately after an arrival at state $\mathbf{W}(t) = \mathbf{w}$.

(ii) There exist $\theta > 0$ and $D < \infty$ such that for all $\mathbf{W}(t) = \mathbf{w}$, $\mathbb{E}\left[e^{\theta|\Delta V(\mathbf{w})|}\right] < D$,

then for any $0 < \eta < \min\left\{\frac{\alpha\theta^2}{\lambda D}, \theta\right\}$, we have

$$\mathbb{E}\left[e^{\eta V(\mathbf{W})}\right] \leqslant \frac{\theta^2\beta e^{\eta K}}{\theta^2\alpha - \lambda\eta D} < \infty$$

Proof. Fix some positive integer n. Applying Rate Conservation Law [37] to $e^{\eta(V(\mathbf{W})\wedge n)}$ gives us

$$\mathbb{E}\left[\eta D_t(V(\mathbf{W})\wedge n)e^{\eta(V(\mathbf{W})\wedge n)}\right]+\lambda\mathbb{E}\left[e^{\eta(V(\mathbf{W}_+)\wedge n)}-e^{\eta(V(\mathbf{W})\wedge n)}\right]=0 \tag{3}$$

We first look into the second term on the LHS of (3) conditioned on $\mathbf{W}$.

$$\begin{aligned}
&\mathbb{E}\left[e^{\eta(V(\mathbf{W}_+)\wedge n)}-e^{\eta(V(\mathbf{W})\wedge n)}\mid\mathbf{W}\right]\\
=&\mathbb{E}\left[\left(e^{\eta(V(\mathbf{W}_+)\wedge n-V(\mathbf{W})\wedge n)}-1\right)e^{\eta(V(\mathbf{W})\wedge n)}\mid\mathbf{W}\right]\\
\overset{(a)}{\leqslant}&\mathbb{E}\left[\left(e^{\eta\Delta V(\mathbf{W})\cdot\mathbb{1}(V(\mathbf{W})\leqslant n)}-1\right)e^{\eta(V(\mathbf{W})\wedge n)}\mid\mathbf{W}\right]\\
=&\mathbb{E}\left[\left(e^{\eta\Delta V(\mathbf{W})}-1\right)e^{\eta V(\mathbf{W})}\mathbb{1}(V(\mathbf{W})\leqslant n)\mid\mathbf{W}\right]\\
=&\mathbb{E}\left[\eta\Delta V(\mathbf{W})e^{\eta V(\mathbf{W})}\mathbb{1}(V(\mathbf{W})\leqslant n)\mid\mathbf{W}\right]+\\
&\mathbb{E}\left[\left(e^{\eta\Delta V(\mathbf{W})}-\eta\Delta V(\mathbf{W})-1\right)e^{\eta V(\mathbf{W})}\mathbb{1}(V(\mathbf{W})\leqslant n)\mid\mathbf{W}\right]\\
\leqslant&\mathbb{E}\left[\eta\Delta V(\mathbf{W})\mid\mathbf{W}\right]e^{\eta V(\mathbf{W})}\mathbb{1}(V(\mathbf{W})\leqslant n)+\\
&\mathbb{E}\left[\left(e^{\eta\Delta V(\mathbf{W})}-\eta\Delta V(\mathbf{W})-1\right)\mid\mathbf{W}\right]e^{\eta V(\mathbf{W})}\mathbb{1}(V(\mathbf{W})\leqslant n)\\
\overset{(b)}{\leqslant}&\mathbb{E}\left[\eta\Delta V(\mathbf{W})\mid\mathbf{W}\right]e^{\eta V(\mathbf{W})}\mathbb{1}(V(\mathbf{W})\leqslant n)+\eta^2c(\mathbf{w})\,e^{\eta V(\mathbf{W})}\mathbb{1}(V(\mathbf{W})\leqslant n)
\end{aligned}$$

where
(a) follows because we have

$$V(\mathbf{W}_+)\wedge n-V(\mathbf{W})\wedge n\leqslant\Delta V(\mathbf{w})\mathbb{1}(V(\mathbf{w})\leqslant n)\quad\forall\,\mathbf{w}$$

(b) follows from assumption (ii) and (2) in Lemma A.1 in that for any $0<\eta\leqslant\theta$ we have

$$\mathbb{E}\left[e^{\eta\Delta V(\mathbf{W})}-\eta\Delta V(\mathbf{W})-1\mid\mathbf{W}\right]\leqslant\eta^2c(\mathbf{W})$$

where

$$c(\mathbf{W})=\sum_{k=2}^{\infty}\frac{\theta^{k-2}}{k!}\mathbb{E}\left[|\Delta V(\mathbf{W})|^k\right]$$

Here, $c(\mathbf{W})$ depends on $\mathbf{W}$. To obtain a bound independent of $\mathbf{W}$, we note that assumption (ii) gives

$$c(\mathbf{W})=\frac{\mathbb{E}\left[e^{\theta|\Delta V(\mathbf{W})|}\right]-(1+\theta\mathbb{E}[|\Delta V(\mathbf{W})|])}{\theta^2}\leqslant\frac{\mathbb{E}\left[e^{\theta|\Delta V(\mathbf{W})|}\right]}{\theta^2}\leqslant\frac{D}{\theta^2}$$

Taking expectation on both sides with respect to the stationary distribution of $\mathbf{W}(t)$ gives us

$$\begin{aligned}
\mathbb{E}\left[e^{\eta(V(\mathbf{W}_+)\wedge n)}-e^{\eta(V(\mathbf{W})\wedge n)}\right]\leqslant&\mathbb{E}\left[\mathbb{E}\left[\eta\Delta V(\mathbf{W})\mid\mathbf{W}\right]e^{\eta V(\mathbf{W})}\mathbb{1}(V(\mathbf{W})\leqslant n)\right]+\\
&\frac{\eta^2D}{\theta^2}\mathbb{E}\left[e^{\eta V(\mathbf{W})}\mathbb{1}(V(\mathbf{W})\leqslant n)\right]
\end{aligned}$$

Now we turn to analyze the first term on the LHS of 3. First observe that for any $\mathbf{w}$,

$$\eta D_t(V(\mathbf{w})\wedge n)e^{\eta(V(\mathbf{w})\wedge n)}\leqslant\eta D_t(V(\mathbf{w}))e^{\eta V(\mathbf{w})}\mathbb{1}(V(\mathbf{w})\leqslant n)$$

By assumption (i),

$$-\mathcal{G}V(\mathbf{w})=-D_tV(\mathbf{w})-\lambda\mathbb{E}[\Delta V(\mathbf{w})]\geqslant\alpha-\beta\cdot\mathbb{1}(V(\mathbf{w})\leqslant K)$$

Therefore, for any $\mathbf{w}$,

$$\begin{aligned}
&-\eta D_t(V(\mathbf{w}))e^{\eta V(\mathbf{w})}\mathbb{1}(V(\mathbf{w}) \leqslant n) \\
&\geqslant \eta(\alpha - \beta\mathbb{1}(V(\mathbf{w}) \leqslant K) + \lambda\mathbb{E}[\Delta V(\mathbf{w})])e^{\eta V(\mathbf{w})}\mathbb{1}(V(\mathbf{w}) \leqslant n) \\
&= (\alpha + \lambda\mathbb{E}[\Delta V(\mathbf{w})])\eta e^{\eta V(\mathbf{w})}\mathbb{1}(V(\mathbf{w}) \leqslant n) - \beta\eta e^{\eta V(\mathbf{w})}\mathbb{1}(V(\mathbf{w}) \leqslant n)\mathbb{1}(V(\mathbf{w}) \leqslant K) \\
&\geqslant (\alpha + \lambda\mathbb{E}[\Delta V(\mathbf{w})])\eta e^{\eta V(\mathbf{w})}\mathbb{1}(V(\mathbf{w}) \leqslant n) - \beta\eta e^{\eta K}
\end{aligned}$$

Taking expectation on both sides with respect to π gives us

$$\begin{aligned}
-\eta\mathbb{E}\left[D_t(V(\mathbf{W}))e^{\eta V(\mathbf{W})}\mathbb{1}(V(\mathbf{W}) \leqslant n)\right] \geqslant \alpha\eta\mathbb{E}\left[e^{\eta V(\mathbf{W})}\mathbb{1}(V(\mathbf{W}) \leqslant n)\right] + \\
\lambda\mathbb{E}\left[\mathbb{E}\left[\eta\Delta V(\mathbf{W}) \mid \mathbf{W}\right] e^{\eta V(\mathbf{W})}\mathbb{1}(V(\mathbf{W}) \leqslant n)\right] - \beta\eta e^{\eta K}
\end{aligned}$$

Rearranging (3) and applying all inequalities obtained above give us

$$\begin{aligned}
&\alpha\eta\mathbb{E}\left[e^{\eta V(\mathbf{W})}\mathbb{1}(V(\mathbf{W}) \leqslant n)\right] + \lambda\mathbb{E}\left[\mathbb{E}\left[\eta\Delta V(\mathbf{W}) \mid \mathbf{W}\right] e^{\eta V(\mathbf{W})}\mathbb{1}(V(\mathbf{W}) \leqslant n)\right] - \beta\eta e^{\eta K} \\
&\leqslant -\eta\mathbb{E}\left[D_t(V(\mathbf{W}))e^{\eta V(\mathbf{W})}\mathbb{1}(V(\mathbf{W}) \leqslant n)\right] \\
&= \lambda\mathbb{E}\left[e^{\eta(V(\mathbf{W}_+)\wedge n)} - e^{\eta(V(\mathbf{W})\wedge n)}\right] \\
&\leqslant \lambda\mathbb{E}\left[\mathbb{E}\left[\eta\Delta V(\mathbf{W}) \mid \mathbf{W}\right] e^{\eta V(\mathbf{W})}\mathbb{1}(V(\mathbf{W}) \leqslant n)\right] + \lambda\frac{\eta^2 D}{\theta^2}\mathbb{E}\left[e^{\eta V(\mathbf{W})}\mathbb{1}(V(\mathbf{W}) \leqslant n)\right]
\end{aligned}$$

Rearranging,

$$\left(\alpha - \lambda\frac{\eta D}{\theta^2}\right)\mathbb{E}\left[e^{\eta V(\mathbf{W})}\mathbb{1}(V(\mathbf{W}) \leqslant n)\right] \leqslant \beta e^{\eta K}$$

Take $\eta > 0$ so small that $\alpha - \lambda\frac{\eta D}{\theta^2} > 0$. We have

$$\mathbb{E}\left[e^{\eta V(\mathbf{W})}\mathbb{1}(V(\mathbf{W}) \leqslant n)\right] \leqslant \frac{\theta^2\beta e^{\eta K}}{\theta^2\alpha - \lambda\eta D}$$

Letting $n \to \infty$ and invoking monotone convergence theorem complete the proof.

□

A.3 Proof of Theorem 6.3

We would like to prove the bucket state-space collapse to the cone C result as discussed in Sections 6.1 and 6.3.

We first note that condition (ii) of Theorem 6.2 is satisfied because $|\Delta H_i(\mathbf{w})|$ is upper bounded by the largest job size allowed by the bucket. Before we verify condition (i), we would like to show the following lemma. The lemma is first presented in Eryilmaz and Srikant [11] under a slightly different setting. We show that this holds here as well.

Lemma A.2. *For vectors* $\mathbf{w}$ *and* $\mathbf{s}$*, we have*

$$\|(\mathbf{w}+\mathbf{s})_{\perp\boldsymbol{\rho}}\|_2 - \|\mathbf{w}_{\perp\boldsymbol{\rho}}\|_2 \leqslant \frac{1}{2\|\mathbf{w}_{\perp\boldsymbol{\rho}}\|_2}\left[(\|\mathbf{w}+\mathbf{s}\|_2^2 - \|\mathbf{w}\|_2^2) - (\|(\mathbf{w}+\mathbf{s})_{\|\boldsymbol{\rho}}\|_2^2 - \|\mathbf{w}_{\|\boldsymbol{\rho}}\|_2^2)\right]$$

Proof. Note that we have $\|\mathbf{x}\|_2 = \sqrt{\|\mathbf{x}\|_2^2}$ and the square root function $x \mapsto \sqrt{x}$ is concave. Thus,

$$\|(\mathbf{w}+\mathbf{s})_{\perp\boldsymbol{\rho}}\|_2 - \|\mathbf{w}_{\perp\boldsymbol{\rho}}\|_2 = \sqrt{\|(\mathbf{w}+\mathbf{s})_{\perp\boldsymbol{\rho}}\|_2^2} - \sqrt{\|\mathbf{w}_{\perp\boldsymbol{\rho}}\|_2^2}$$

$$\leqslant \frac{1}{2\|\mathbf{w}_{\perp\boldsymbol{\rho}}\|_2}\left(\|(\mathbf{w}+\mathbf{s})_{\perp\boldsymbol{\rho}}\|_2^2 - \|\mathbf{w}_{\perp\boldsymbol{\rho}}\|_2^2\right)$$

The lemma now follows from the Pythagorean theorem

$$\|(\mathbf{w}+\mathbf{s})_{\perp\boldsymbol{\rho}}\|_2^2 - \|\mathbf{w}_{\perp\boldsymbol{\rho}}\|_2^2 = \|\mathbf{w}+\mathbf{s}\|_2^2 - \|(\mathbf{w}+\mathbf{s})_{\|\boldsymbol{\rho}}\|_2^2 - \|\mathbf{w}\|_2^2 + \|\mathbf{w}_{\|\boldsymbol{\rho}}\|_2^2$$

□

Theorem 6.3. *For any bucket i, under our SEB policy, if the assumptions in Section 3.5 are met, then we have the following state-space collapse:*

$$\text{If } H_i(\mathbf{w}) = (\|\mathbf{w}^{(i)}_{\perp\boldsymbol{\rho}}\|_2 - \|\mathbf{w}^{(i)}_{\|\boldsymbol{\rho}}\|_2 \tan\varphi)^+ \text{ and } \eta_i = e^{-2}\frac{b_{i-1}}{b_i^2}\tan\varphi, \text{ then } \mathbb{E}\left[e^{\eta_i H_i(\mathbf{W})}\right] \leqslant \frac{8}{\tan\varphi}\frac{b_i}{b_{i-1}},$$

bounding the distance from $\mathbf{w}_i$ to the cone C.

Proof. Under SEB, a bucket alternates between two modes: pure work accumulation mode, where no service is given to the bucket, and service mode, where all servers work on the bucket. We will analyze the drifts under both modes separately.

In the work accumulation mode, the generator for the work vector $\mathbf{w}$ is

$$\mathcal{G}H_i(\mathbf{w}) = \lambda\mathbb{E}[\Delta H_i(\mathbf{w})]$$

We have

$$\Delta H_i(\mathbf{w}) = (\|(\mathbf{w}^{(i)}+\mathbf{s}^{(i)})_{\perp\boldsymbol{\rho}}\|_2 - \|\mathbf{w}^{(i)}_{\perp\boldsymbol{\rho}}\|_2) - (\|(\mathbf{w}^{(i)}+\mathbf{s}^{(i)})_{\|\boldsymbol{\rho}}\|_2 - \|\mathbf{w}^{(i)}_{\|\boldsymbol{\rho}}\|_2)\tan\varphi$$

In light of Lemma A.2, it suffices to look at $\|\mathbf{w}^{(i)}+\mathbf{s}^{(i)}\|_2^2 - \|\mathbf{w}^{(i)}\|_2^2$ and $\|(\mathbf{w}^{(i)}+\mathbf{s}^{(i)})_{\|\boldsymbol{\rho}}\|_2^2 - \|\mathbf{w}^{(i)}_{\|\boldsymbol{\rho}}\|_2^2$

$$\|\mathbf{w}^{(i)}+\mathbf{s}^{(i)}\|_2^2 - \|\mathbf{w}^{(i)}\|_2^2 = 2\left\langle \mathbf{w}^{(i)}, \mathbf{s}^{(i)}\right\rangle + \|\mathbf{s}^{(i)}\|_2^2$$

$$\|(\mathbf{w}^{(i)}+\mathbf{s}^{(i)})_{\|\boldsymbol{\rho}}\|_2^2 - \|\mathbf{w}^{(i)}_{\|\boldsymbol{\rho}}\|_2^2 = 2\left\langle \mathbf{w}^{(i)}_{\|\boldsymbol{\rho}}, \mathbf{s}^{(i)}_{\|\boldsymbol{\rho}}\right\rangle + \|\mathbf{s}^{(i)}_{\|\boldsymbol{\rho}}\|_2^2 = 2\frac{\left\langle \mathbf{w}^{(i)}, \boldsymbol{\rho}^{(i)}\right\rangle\left\langle \mathbf{s}^{(i)}, \boldsymbol{\rho}^{(i)}\right\rangle}{\|\boldsymbol{\rho}^{(i)}\|_2^2} + \|\mathbf{s}^{(i)}_{\|\boldsymbol{\rho}}\|_2^2$$

By Lemma A.2,

$$\|(\mathbf{w}^{(i)}+\mathbf{s}^{(i)})_{\perp\boldsymbol{\rho}}\|_2 - \|\mathbf{w}^{(i)}_{\perp\boldsymbol{\rho}}\|_2 \leqslant \frac{1}{\|\mathbf{w}^{(i)}_{\perp\boldsymbol{\rho}}\|_2}\left(\left\langle \mathbf{w}^{(i)}, \mathbf{s}^{(i)}\right\rangle - \frac{\left\langle \mathbf{w}^{(i)}, \boldsymbol{\rho}^{(i)}\right\rangle\left\langle \mathbf{s}^{(i)}, \boldsymbol{\rho}^{(i)}\right\rangle}{\|\boldsymbol{\rho}^{(i)}\|_2^2} + \frac{1}{2}\|\mathbf{s}^{(i)}_{\perp\boldsymbol{\rho}}\|_2^2\right)$$

and

$$\|(\mathbf{w}^{(i)}+\mathbf{s})_{\|\boldsymbol{\rho}}\|_2 - \|\mathbf{w}^{(i)}_{\|\boldsymbol{\rho}}\|_2 = \frac{\left\langle \mathbf{w}^{(i)}+\mathbf{s}^{(i)}, \boldsymbol{\rho}^{(i)}\right\rangle}{\|\boldsymbol{\rho}^{(i)}\|_2^2}\|\boldsymbol{\rho}^{(i)}\|_2 - \frac{\left\langle \mathbf{w}^{(i)}, \boldsymbol{\rho}^{(i)}\right\rangle}{\|\boldsymbol{\rho}^{(i)}\|_2^2}\|\boldsymbol{\rho}^{(i)}\|_2 = \frac{\left\langle \mathbf{s}^{(i)}, \boldsymbol{\rho}^{(i)}\right\rangle}{\|\boldsymbol{\rho}^{(i)}\|_2}$$

Thus,

$$\Delta H_i(\mathbf{w}) \leqslant \frac{1}{\|\mathbf{w}^{(i)}_{\perp\boldsymbol{\rho}}\|_2}\left(\left\langle \mathbf{w}^{(i)}, \mathbf{s}^{(i)}\right\rangle - \frac{\left\langle \mathbf{w}^{(i)}, \boldsymbol{\rho}\right\rangle\left\langle \mathbf{s}^{(i)}, \boldsymbol{\rho}^{(i)}\right\rangle}{\|\boldsymbol{\rho}^{(i)}\|_2^2} + \frac{1}{2}\|\mathbf{s}_{\perp\boldsymbol{\rho}}\|_2^2\right) - \frac{\left\langle \mathbf{s}^{(i)}, \boldsymbol{\rho}^{(i)}\right\rangle}{\|\boldsymbol{\rho}^{(i)}\|_2}\tan\varphi$$

Since $\lambda_i\mathbb{E}[\mathbf{S}^{(i)}] = \boldsymbol{\rho}^{(i)}$, we have

$$\lambda\mathbb{E}[\Delta H_i(\mathbf{w})] \leqslant \frac{\lambda\mathbb{E}[\|\mathbf{S}_{\perp\boldsymbol{\rho}}\|_2^2]}{2\|\mathbf{w}^{(i)}_{\perp\boldsymbol{\rho}}\|_2} - \|\boldsymbol{\rho}^{(i)}\|_2\tan\varphi$$

Now we show that, in the service mode, $\mathbf{w}^{(i)}$ also collapses to cone C. The generator for the bucket, when the bucket is in service mode, is

$$\mathcal{G}H_i(\mathbf{w}) = -\left\langle \nabla H_i(\mathbf{w}), \mathbf{r}^*\right\rangle + \lambda\mathbb{E}[\Delta H_i(\mathbf{w})]$$

In light of the drift analysis of $\lambda\mathbb{E}[\Delta H_i(\mathbf{w})]$, it suffices to show that $\langle \nabla H_i(\mathbf{w}), \mathbf{r}^* \rangle > 0$. We first compute $\nabla H_i(\mathbf{w})$.

$$\nabla H_i(\mathbf{w}) = \nabla \left(\|\mathbf{w}_{\perp\boldsymbol{\rho}}^{(i)}\|_2 - \|\mathbf{w}_{\|\boldsymbol{\rho}}^{(i)}\|_2 \tan\varphi \right)$$

We have

$$\nabla \|\mathbf{w}_{\|\boldsymbol{\rho}}^{(i)}\|_2 = \frac{\boldsymbol{\rho}}{\|\boldsymbol{\rho}\|_2^2} \frac{\left\langle \mathbf{w}^{(i)}, \boldsymbol{\rho} \right\rangle}{\|\mathbf{w}_{\|\boldsymbol{\rho}}^{(i)}\|_2} = \frac{\boldsymbol{\rho}}{\|\boldsymbol{\rho}\|_2}$$

$$\nabla \|\mathbf{w}_{\perp\boldsymbol{\rho}}^{(i)}\|_2 = \frac{\mathbf{w}_{\perp\boldsymbol{\rho}}^{(i)}}{\|\mathbf{w}_{\perp\boldsymbol{\rho}}^{(i)}\|_2}$$

Thus, by assumptions on φ in Section 6.1,

$$\langle \nabla H_i(\mathbf{w}), \mathbf{r}^* \rangle = \frac{\left\langle \mathbf{w}_{\perp\boldsymbol{\rho}}^{(i)}, \mathbf{r}^* \right\rangle}{\|\mathbf{w}_{\perp\boldsymbol{\rho}}^{(i)}\|_2} - \frac{\left\langle \boldsymbol{\rho}, \mathbf{r}^* \right\rangle}{\|\boldsymbol{\rho}\|_2} \tan\varphi > 0$$

Set

$$K_i = \frac{\lambda_i \mathbb{E}\left[\|\mathbf{S}_{\perp\boldsymbol{\rho}}^{(i)}\|_2^2 \right]}{\|\boldsymbol{\rho}^{(i)}\|_2} \cot\varphi$$

as in Theorem 6.2. Since $\lambda\mathbb{E}[\Delta H_i(\mathbf{w})] \leqslant \lambda_i b_i$ in Theorem 6.2, we have

$$\mathcal{G}H_i(\mathbf{w}) \leqslant -\frac{\|\boldsymbol{\rho}^{(i)}\|_2}{2} \tan\varphi + \left(\lambda_i b_i + \|\boldsymbol{\rho}^{(i)}\|_2 \right) \mathbb{1}(H(\mathbf{w}) \leqslant K_i)$$

We set $\alpha_i = -\frac{\|\boldsymbol{\rho}^{(i)}\|_2}{2} \tan\varphi$, $\beta_i = \lambda_i b_i + \|\boldsymbol{\rho}^{(i)}\|_2$, and $\theta_i = \frac{2}{b_i}$ as in Theorem 6.2. Note that $D = e^2$ as a result of $|\Delta H_i(\mathbf{w})| \leqslant b_i$ for all $\mathbf{w}$.

With these parameters in hand, we are ready to bound $\mathbb{E}[e^{\eta H(\mathbf{w})}]$ in terms of b_{i-1} and b_i. First note that we have

$$\sqrt{n_c}\lambda_i b_{i-1} \leqslant \|\boldsymbol{\rho}^{(i)}\|_2 \leqslant \sqrt{n_c}\lambda_i b_i$$

which immediately yields the following bounds

$$\frac{\sqrt{n_c}\lambda_i b_{i-1}}{2} \tan\varphi \leqslant \alpha_i \leqslant \frac{\sqrt{n_c}\lambda_i b_i}{2} \tan\varphi, \quad \beta_i \leqslant (\sqrt{n_c}+1)\lambda_i b_i \quad \text{and} \quad K_i \leqslant \frac{1}{\sqrt{n_c}} \frac{b_i^2}{b_{i-1}} \cot\varphi.$$

Recall that η_i in Theorem 6.2 needs to be taken so that $0 < \eta_i < \min\left\{ \frac{\alpha_i \theta_i^2}{\lambda_i D}, \theta \right\}$. Since we have

$$\frac{\alpha_i \theta_i^2}{\lambda_i D} \geqslant \frac{1}{\lambda_i e^2} \frac{\sqrt{n_c}\lambda_i b_{i-1}}{2} \tan\varphi \frac{4}{b_i^2} = 2\sqrt{n_c} e^{-2} \frac{b_{i-1}}{b_i^2} \tan\varphi,$$

$\eta_i = e^{-2} \frac{b_{i-1}}{b_i^2} \tan\varphi$ is a fine choice. Then we have

$$\theta_i^2 \beta_i e^{\eta_i K_i} \leqslant \frac{4}{b_i^2} (\sqrt{n_c}+1) \lambda_i b_i e^{e^{-2}/\sqrt{n_c}}$$

$$\theta_i^2 \alpha_i - \lambda_i \eta_i D \geqslant 2\sqrt{n_c} \frac{b_{i-1}}{b_i^2} \tan\varphi - \lambda_i \frac{b_{i-1}}{b_i^2} \tan\varphi \geqslant \lambda_i \sqrt{n_c} \frac{b_{i-1}}{b_i^2} \tan\varphi$$

Finally, we invoke Theorem 6.2 to obtain

$$\mathbb{E}\left[e^{\eta_i H(\mathbf{w})} \right] \leqslant \frac{4}{\tan\varphi} \frac{\sqrt{n_c}+1}{2\sqrt{n_c}-1} e^{e^{-2}/\sqrt{n_c}} \frac{b_i}{b_{i-1}} \leqslant \frac{8}{\tan\varphi} \frac{b_i}{b_{i-1}}$$

The last inequality follows from $\frac{\sqrt{n_c}+1}{2\sqrt{n_c}-1}e^{e^{-2}/\sqrt{n_c}} \leqslant 2$ for $n_c \geqslant 2$. □

A.4 Proof of Theorem 7.1

Theorem 7.1. *If $\mathbf{w}^{(i)} \in C$ and if $\|\mathbf{w}^{(i)}\|_1$ satisfies*

$$\|\mathbf{w}^{(i)}\|_1 \geqslant \frac{n_s\sqrt{n_c}b_i}{\frac{\boldsymbol{\rho}_{\min}}{\|\boldsymbol{\rho}\|_2}\cos\varphi - \sin\varphi}$$

then any service option in $\mathcal{R}^{\mathcal{F}}$ can be fulfilled using jobs in bucket i.

The theorem is a corollary of the following lemma, which relates the ℓ_1 and ℓ_∞ norms for workload vectors in the cone C.

Lemma A.3. *If $\mathbf{w}^{(i)} \in C$, then*

$$\min_{1\leqslant j\leqslant n_c} \mathbf{w}_j^{(i)} \geqslant \|\mathbf{w}^{(i)}\|_1 \cdot \frac{1}{\sqrt{n_c}}\left(\frac{\boldsymbol{\rho}_{\min}}{\|\boldsymbol{\rho}\|_2}\cos\varphi - \sin\varphi\right)$$

Proof. Let $\mathbf{e}_j$ be the j-th standard basis in $\mathbb{R}^n$, we have

$$\begin{aligned}
\left\langle \mathbf{w}^{(i)}, \mathbf{e}_j \right\rangle &= \left\langle \mathbf{w}^{(i)}_{\|\boldsymbol{\rho}}, \mathbf{e}_j \right\rangle + \left\langle \mathbf{w}^{(i)}_{\perp\boldsymbol{\rho}}, \mathbf{e}_j \right\rangle \\
&\geqslant \frac{\left\langle \mathbf{w}^{(i)}, \boldsymbol{\rho} \right\rangle}{\|\boldsymbol{\rho}\|_2^2}\boldsymbol{\rho}_j - \|\mathbf{w}^{(i)}_{\perp\boldsymbol{\rho}}\|_2 \\
&\overset{(a)}{\geqslant} \frac{\left\langle \mathbf{w}^{(i)}, \boldsymbol{\rho} \right\rangle}{\|\boldsymbol{\rho}\|_2^2}\boldsymbol{\rho}_i - \|\mathbf{w}^{(i)}\|_2 \sin\varphi \\
&\overset{(b)}{\geqslant} \frac{\boldsymbol{\rho}_j}{\|\boldsymbol{\rho}\|_2}\|\mathbf{w}^{(i)}\|_2\cos\varphi - \|\mathbf{w}^{(i)}\|_2\sin\varphi \\
&\overset{(c)}{\geqslant} \frac{1}{\sqrt{n_c}}\|\mathbf{w}^{(i)}\|_1\left(\frac{\boldsymbol{\rho}_{\min}}{\|\boldsymbol{\rho}\|_2}\cos\varphi - \sin\varphi\right)
\end{aligned}$$

where (a) and (b) follow from the assumption that $\mathbf{w}^{(i)} \in C$ and (c) follows from the norm inequality. Note further that

$$\frac{\boldsymbol{\rho}_{\min}}{\|\boldsymbol{\rho}\|_2}\cos\varphi - \sin\varphi > 0$$

because of assumption in Section 6.1 on φ. □

A.5 Proof of Lemma 7.3

Lemma A.4. *For any $\mathbf{w}^{(j)} \notin C \cup \mathcal{E}_i$, $H_j(\mathbf{w}) = \|\mathbf{w}^{(j)}_{\perp\boldsymbol{\rho}}\|_2 - \|\mathbf{w}^{(j)}_{\|\boldsymbol{\rho}}\|_2\tan\varphi \geqslant \tau\|\mathbf{w}^{(j)}\|_1 - n_s b_j$.*

Proof. We begin by showing that, for any $\mathbf{w}^{(j)} \notin C$, there exists $\overline{\mathbf{w}}^{(j)} \in C$ such that $H(\mathbf{w}^{(i)}) = \|\mathbf{w}^{(j)} - \overline{\mathbf{w}}^{(j)}\|_2$. One can verify that

$$\overline{\mathbf{w}}^{(j)} = \mathbf{w}^{(j)}_{\|\boldsymbol{\rho}} + \frac{\mathbf{w}^{(j)}_{\perp\boldsymbol{\rho}}}{\|\mathbf{w}^{(j)}_{\perp\boldsymbol{\rho}}\|_2}\|\mathbf{w}^{(j)}_{\|\boldsymbol{\rho}}\|_2\tan\varphi$$

is such a vector.

Since $\mathbf{w}^{(j)} \notin \mathcal{E}_i$, there exists $k = 1, \ldots, n_c$ such that $\mathbf{w}^{(j)}_k < n_s b_j$. Thus,

$$H(\mathbf{w}^{(i)}) = \|\mathbf{w}^{(j)} - \overline{\mathbf{w}}^{(j)}\|_2 \geqslant \overline{\mathbf{w}}^{(j)}_k - \mathbf{w}^{(j)}_k \geqslant \frac{\tau}{\tan\varphi}\|\overline{\mathbf{w}}^{(j)}\|_1$$

where the last inequality comes from Lemma A.3, which relates the ℓ_1 and ℓ_∞ norms for workload vectors in the cone C. It remains to lower bound $\|\overline{\mathbf{w}}^{(j)}\|_1$. Notice that we have

$$\|\overline{\mathbf{w}}^{(j)}\|_1 = \left\| \mathbf{w}^{(j)}_{\|\boldsymbol{\rho}} + \mathbf{w}^{(j)}_{\perp\boldsymbol{\rho}} \frac{\|\mathbf{w}^{(j)}_{\|\boldsymbol{\rho}}\|_2}{\|\mathbf{w}^{(j)}_{\perp\boldsymbol{\rho}}\|_2} \tan\varphi \right\|_1 \overset{(a)}{\geqslant} \min\{\|\mathbf{w}^{(j)}_{\|\boldsymbol{\rho}}\|_1, \|\mathbf{w}^{(j)}\|_1\}$$

where (a) follows from the following observations

- Note that $\frac{\|\mathbf{w}^{(j)}_{\|\boldsymbol{\rho}}\|_2}{\|\mathbf{w}^{(j)}_{\perp\boldsymbol{\rho}}\|_2}$ can be viewed as cotangent of the angle between $\mathbf{w}^{(j)}$ and $\boldsymbol{\rho}$. Since $\mathbf{w}^{(j)} \notin C$, this angle is larger than φ. This implies that $\frac{\|\mathbf{w}^{(j)}_{\|\boldsymbol{\rho}}\|_2}{\|\mathbf{w}^{(j)}_{\perp\boldsymbol{\rho}}\|_2} \tan\varphi \in (0,1)$.
- The ℓ_1 norm is a linear function, so the minimum is achieved whenever the constant $\frac{\|\mathbf{w}^{(j)}_{\|\boldsymbol{\rho}}\|_2}{\|\mathbf{w}^{(j)}_{\perp\boldsymbol{\rho}}\|_2} \tan\varphi$ is 0 or 1.

Analyzing $\|\mathbf{w}^{(j)}_{\|\boldsymbol{\rho}}\|_1$ further gives us

$$\|\mathbf{w}^{(j)}_{\|\boldsymbol{\rho}}\|_1 = \frac{\langle \mathbf{w}^{(j)}, \boldsymbol{\rho}\rangle}{\|\boldsymbol{\rho}\|_2^2}\|\boldsymbol{\rho}\|_1 \geqslant \|\mathbf{w}^{(j)}\|_1 \frac{\boldsymbol{\rho}_{\min}}{\|\boldsymbol{\rho}\|_2}\frac{\|\boldsymbol{\rho}\|_1}{\|\boldsymbol{\rho}\|_2} \overset{(b)}{\geqslant} \|\mathbf{w}^{(j)}\|_1 \frac{\boldsymbol{\rho}_{\min}}{\|\boldsymbol{\rho}\|_2} \overset{(c)}{\geqslant} \|\mathbf{w}^{(j)}\|_1 \tan\varphi$$

where (b) follows from the norm inequality and (c) follows from the assumption on φ in Section 3.5. This completes the proof of the lemma. □

A.6 Proof of Lemma 7.5

Lemma A.5. *For any policy π that stabilizes the system,*

$$\mathbb{E}[T^\pi_{wait}] \leqslant \frac{1}{\lambda}\sum_{i=1}^{n_b} \frac{c-1}{b_0 c^i}\mathbb{E}[W^\pi_{\leqslant i}] + \frac{1}{\lambda}\frac{\mathbb{E}[W^\pi]}{b_{n_b}}$$

where $W^\pi_{\leqslant i}$ is work in the first i buckets and W^π is the total work in the system.

Proof. To bound $\mathbb{E}[T^\pi_{\text{wait}}]$, we apply WINE (Proposition 5.3) to the queueing system consists only of jobs waiting to enter service. In this system, the remaining sizes of jobs as in Proposition 5.3 is their original sizes, as jobs leave the system as soon as they receive any service.

By Proposition 5.3,

$$\begin{aligned}
\mathbb{E}[T^\pi_{\text{wait}}] &\leqslant \frac{1}{\lambda}\int_0^\infty \frac{\mathbb{E}[W^\pi_{\text{original size}\leqslant x}]}{x^2}\,dx \\
&= \frac{1}{\lambda}\int_0^{b_{n_b}} \frac{\mathbb{E}[W^\pi_{\text{original size}\leqslant x}]}{x^2}\,dx + \frac{1}{\lambda}\int_{b_{n_b}}^\infty \frac{\mathbb{E}[W^\pi_{\text{original size}\leqslant x}]}{x^2}\,dx \\
&= \frac{1}{\lambda}\sum_{i=1}^{n_b}\int_{b_{i-1}}^{b_i} \frac{\mathbb{E}[W^\pi_{\text{original size}\leqslant x}]}{x^2}\,dx + \frac{1}{\lambda}\frac{\mathbb{E}[W^\pi_{\text{original size}\leqslant b_{n_b}}]}{b_{n_b}} \\
&\leqslant \frac{1}{\lambda}\sum_{i=1}^{n_b}\int_{b_{i-1}}^{b_i} \frac{\mathbb{E}[W^\pi_{\text{original size}\leqslant b_i}]}{x^2}\,dx + \frac{1}{\lambda}\frac{\mathbb{E}[W^\pi_{\text{original size}\leqslant b_{n_b}}]}{b_{n_b}} \\
&= \frac{1}{\lambda}\sum_{i=1}^{n_b}\mathbb{E}[W^\pi_{\text{original size}\leqslant b_i}]\left(\frac{1}{b_{i-1}} - \frac{1}{b_i}\right) + \frac{1}{\lambda}\frac{\mathbb{E}[W^\pi_{\text{original size}\leqslant b_{n_b}}]}{b_{n_b}}
\end{aligned}$$

$$= \frac{1}{\lambda}\sum_{i=1}^{n_b}\frac{c-1}{b_0c^i}\mathbb{E}[W^{\pi}_{\leqslant i}] + \frac{1}{\lambda}\frac{\mathbb{E}[W^{\pi}]}{b_{n_b}}.$$

□

A.7 Proof of Proposition 7.6

Proposition A.6. *Under SEB,*

$$\mathbb{E}[T^{\text{SEB}}] \leqslant c\mathbb{E}[T^{\text{PSJF-1}}] + \frac{A}{\lambda}\log\left(\frac{1}{1-\rho}\right)\left(2n_b + \frac{1}{c-1}\right) + \frac{1}{\lambda}n_b n_s n_c$$

where

$$A = \frac{A_1 + 3A_2}{b_0}\frac{b_{n_b} - b_0}{b_0},$$

and where A_1 and A_2 are defined in Theorem 7.2.

Proof. By Lemma 7.5,

$$\mathbb{E}[T^{\text{SEB}}_{\text{wait}}] \leqslant \underbrace{\frac{1}{\lambda}\sum_{i=1}^{n_b}\frac{c-1}{b_0c^i}\mathbb{E}[W^{\text{SEB}}_{\leqslant i}]}_{\mathcal{T}_1} + \underbrace{\frac{1}{\lambda}\frac{\mathbb{E}[W^{\text{SEB}}]}{b_{n_b}}}_{\mathcal{T}_2}$$

Recall from Proposition 5.4 that for any $i = 1, \ldots, n_b$,

$$\mathbb{E}[W^{\text{SEB}}_{\leqslant i}] - \mathbb{E}[W^{\text{M/G/1}}_{\leqslant i}] = \frac{\mathbb{E}[\mathcal{I}^{\text{SEB}}_{\leqslant i}W^{\text{SEB}}_{\leqslant i}]}{1-\rho_{\leqslant i}}$$

Using Proposition 5.4 and the waste bound in Theorem 7.2, we obtain

$$\begin{aligned}
\mathcal{T}_1 &\leqslant \frac{1}{\lambda}\sum_{i=1}^{n_b}\frac{c-1}{b_0c^i}\mathbb{E}[W^{\text{M/G/1}}_{\leqslant i}] + \frac{A_1+A_2}{\lambda}\sum_{i=1}^{n_b}\frac{c^i-1}{b_0c^i} + \frac{A_2}{\lambda}\sum_{i=1}^{n_b}\log\left(\frac{1}{1-\rho_{\leqslant i}}\right)\frac{c^i-1}{b_0c^i} + \frac{A_2}{\lambda}\sum_{i=1}^{n_b} i\frac{c-1}{b_0c^i} \\
&\leqslant \frac{1}{\lambda}\sum_{i=1}^{n_b}\frac{c-1}{b_0c^i}\mathbb{E}[W^{\text{M/G/1}}_{\leqslant i}] + \frac{1}{\lambda}\left(\frac{A_1+A_2}{b_0} + \frac{A_2}{b_0}\log\left(\frac{1}{1-\rho}\right)\right)\sum_{i=1}^{n_b}\frac{c^i-1}{c^i} + \frac{1}{\lambda}\frac{A_2}{b_0}n_b\sum_{i=1}^{n_b}\frac{c-1}{c^i} \\
&\leqslant \frac{1}{\lambda}\sum_{i=1}^{n_b}\frac{c-1}{b_0c^i}\mathbb{E}[W^{\text{M/G/1}}_{\leqslant i}] + \frac{1}{\lambda}\left(\frac{A_1+2A_2}{b_0} + \frac{A_2}{b_0}\log\left(\frac{1}{1-\rho}\right)\right)n_b \\
\mathcal{T}_2 &\leqslant \frac{1}{\lambda}\frac{\mathbb{E}[W^{\text{M/G/1}}]}{b_{n_b}} + \frac{1}{\lambda}\left[\left(\frac{A_1+A_2}{b_{n_b}} + \frac{A_2}{b_{n_b}}\log\left(\frac{1}{1-\rho}\right)\right)\frac{c^{n_b}-1}{c-1} + \frac{A_2}{b_{n_b}}n_b\right]
\end{aligned}$$

Combining bounds for $\mathcal{T}_1$ and $\mathcal{T}_2$ and noting that $b_{n_b} - b_0 = b_0(c^{n_b}-1)$ give us

$$\begin{aligned}
\mathbb{E}[T^{\text{SEB}}_{\text{wait}}] &\leqslant \frac{1}{\lambda}\sum_{i=1}^{n_b}\frac{c-1}{b_0c^i}\mathbb{E}[W^{\text{M/G/1}}_{\leqslant i}] + \frac{1}{\lambda}\frac{\mathbb{E}[W^{\text{M/G/1}}]}{b_{n_b}} + \frac{1}{\lambda}\left(\frac{A_1+A_2}{b_0} + \frac{A_2}{b_0}\log\left(\frac{1}{1-\rho}\right) + \frac{A_2}{b_0}\right)n_b + \\
&\quad \frac{1}{\lambda}\left(\frac{A_1+A_2}{b_{n_b}} + \frac{A_2}{b_{n_b}}\log\left(\frac{1}{1-\rho}\right)\right)\frac{b_{n_b}-b_0}{b_0}\frac{1}{c-1} + \frac{1}{\lambda}\frac{A_2}{b_{n_b}}n_b \\
&\leqslant \frac{1}{\lambda}\sum_{i=1}^{n_b}\frac{c-1}{b_0c^i}\mathbb{E}[W^{\text{M/G/1}}_{\leqslant i}] + \frac{1}{\lambda}\frac{\mathbb{E}[W^{\text{M/G/1}}]}{b_{n_b}} + \frac{A}{\lambda}\log\left(\frac{1}{1-\rho}\right)\left(2n_b + \frac{1}{c-1}\right) \qquad (4)
\end{aligned}$$

Next, we show that the first two terms involving $\mathbb{E}[W^{M/G/1}]$ are upper bounded by $c\mathbb{E}T^{\text{PSJF-1}}$.

We now consider a single-server Preemptive-Shortest-Job-First (PSJF) policy. Let $T^{\text{PSJF-1}}$ be its response time, then we have

$$\begin{aligned}
\mathbb{E}[T^{\text{PSJF-1}}] &\overset{(a)}{=} \frac{1}{\lambda}\int_0^\infty \frac{\mathbb{E}[W^{\text{PSJF-1}}_{\text{remaining size}\leqslant x}]}{x^2}\,dx \\
&\overset{(b)}{\geqslant} \frac{1}{\lambda}\int_0^\infty \frac{\mathbb{E}[W^{\text{PSJF-1}}_{\text{original size}\leqslant x}]}{x^2}\,dx \\
&\overset{(c)}{=} \frac{1}{\lambda}\frac{1}{c}\int_0^\infty \frac{\mathbb{E}[W^{\text{PSJF-1}}_{\text{original size}\leqslant cx}]}{x^2}\,dx \\
&\overset{(d)}{=} \frac{1}{\lambda}\frac{1}{c}\sum_{i=1}^{n_b}\int_{b_{i-1}}^{b_i} \frac{\mathbb{E}[W^{\text{PSJF-1}}_{\text{original size}\leqslant cx}]}{x^2}\,dx + \frac{1}{\lambda}\frac{1}{c}\int_{b_{n_b}}^\infty \frac{\mathbb{E}[W^{\text{PSJF-1}}_{\text{original size}\leqslant cx}]}{x^2}\,dx \\
&\overset{(e)}{\geqslant} \frac{1}{\lambda}\frac{1}{c}\sum_{i=1}^{n_b}\int_{b_{i-1}}^{b_i} \frac{\mathbb{E}[W^{\text{PSJF-1}}_{\text{original size}\leqslant b_i}]}{x^2}\,dx + \frac{1}{\lambda}\frac{1}{c}\int_{b_{n_b}}^\infty \frac{\mathbb{E}[W^{\text{PSJF-1}}_{\text{original size}\leqslant cx}]}{x^2}\,dx \\
&\overset{(f)}{=} \frac{1}{\lambda}\frac{1}{c}\sum_{i=1}^{n_b}\int_{b_{i-1}}^{b_i} \frac{\mathbb{E}[W^{\text{M/G/1}}_{\leqslant i}]}{x^2}\,dx + \frac{1}{\lambda}\frac{1}{c}\int_{b_{n_b}}^\infty \frac{\mathbb{E}[W^{\text{M/G/1}}]}{x^2}\,dx \\
&= \frac{1}{\lambda}\frac{1}{c}\sum_{i=1}^{n_b}\frac{c-1}{b_0 c^i}\mathbb{E}[W^{\text{M/G/1}}_{\leqslant i}] + \frac{1}{\lambda}\frac{1}{c}\frac{1}{b_{n_b}}\mathbb{E}[W^{\text{M/G/1}}] \qquad (5)
\end{aligned}$$

where (a) follows from the WINE identity. (b) follows from the fact that $W^{\text{PSJF-1}}_{\text{remaining size}\leqslant x}$ includes work from jobs with original sizes less than x (i.e. $W^{\text{PSJF-1}}_{\text{original size}\leqslant x}$) and jobs with original sizes larger than x but have remaining sizes less than x because they have received some service. (c) follows from a change of variable $x \mapsto cx$. To better compare $\mathbb{E}[T^{\text{PSJF-1}}]$ and $\mathbb{E}[T^{\text{SEB}}]$, we divide the integral into $b_0, b_1, \ldots, b_{n_b}, \infty$ in (d) the same way we define the buckets in MSJ scheduling. (e) follows from $b_{i-1} \leqslant x \leqslant b_i \leqslant cx$. (f) follows from the fact that PSJF is a work-conserving policy.

Comparing (5) with (4) completes the proof.

□

B MAXWEIGHT-QUEUE SRPT

We now discuss the strong empirical performance of the MaxWeight-Queue SRPT (MWQS) policy across the variety of settings we investigated in Sections 8.1 and 8.2.

Recall our intuition that optimal mean response time requires prioritizing two goals: Serving the jobs of smallest remaining size, and balancing the amount of small jobs of each class to maintain full server utilization.

MWQS accomplishes the first goal very well, because it is an SRPT-based policy, and does a good job for the second goal by balances the total number of jobs of each class. At relatively low loads, the total number of jobs of each class is a good proxy for the number of small jobs in each class.

However, we do not expect MWQS to be heavy-traffic optimal, because at very high load, most jobs in the system are much larger than a typical arriving job, because smaller jobs have been prioritized. As a result, the total number of jobs in a given class is no longer a good proxy for the number of small jobs of that class, which should hurt MWQS.

The strong empirical performance of MWQS is reminiscent of the strong empirical performance of JSQ dispatching to SRPT queues [16]. Further work is needed to understand the impressive performance of these policies.