

Tilburg University

## Heavy-traffic universality of redundancy systems with assignment constraints

Cardinaels, Ellen; Borst, Sem; van Leeuwen, Johan S.H.

*Published in:*  
Operations Research

*DOI:*  
[10.1287/opre.2022.2385](https://doi.org/10.1287/opre.2022.2385)

*Publication date:*  
2022

*Document Version*  
Publisher's PDF, also known as Version of record

[Link to publication in Tilburg University Research Portal](#)

*Citation for published version (APA):*  
Cardinaels, E., Borst, S., & van Leeuwen, J. S. H. (2022). Heavy-traffic universality of redundancy systems with assignment constraints. *Operations Research*. Advance online publication. <https://doi.org/10.1287/opre.2022.2385>

### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.



## Operations Research

Publication details, including instructions for authors and subscription information:  
<http://pubsonline.informs.org>

### Heavy-Traffic Universality of Redundancy Systems with Assignment Constraints

Ellen Cardinaels , Sem Borst , Johan S. H. van Leeuwen

To cite this article:

Ellen Cardinaels , Sem Borst , Johan S. H. van Leeuwen (2022) Heavy-Traffic Universality of Redundancy Systems with Assignment Constraints. Operations Research

Published online in Articles in Advance 05 Dec 2022

. <https://doi.org/10.1287/opre.2022.2385>

Full terms and conditions of use: <https://pubsonline.informs.org/Publications/Librarians-Portal/PubsOnLine-Terms-and-Conditions>

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact [permissions@informs.org](mailto:permissions@informs.org).

The Publisher does not warrant or guarantee the article's accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

Copyright © 2022, INFORMS

Please scroll down for article—it is on subsequent pages



With 12,500 members from nearly 90 countries, INFORMS is the largest international association of operations research (O.R.) and analytics professionals and students. INFORMS provides unique networking and learning opportunities for individual professionals, and organizations of all types and sizes, to better understand and use O.R. and analytics tools and methods to transform strategic visions and achieve better outcomes. For more information on INFORMS, its publications, membership, or meetings visit <http://www.informs.org>



## Methods

# Heavy-Traffic Universality of Redundancy Systems with Assignment Constraints

Ellen Cardinaels,<sup>a,\*</sup> Sem Borst,<sup>a</sup> Johan S. H. van Leeuwen<sup>b</sup>

<sup>a</sup>Department of Mathematics and Computer Science, Eindhoven University of Technology, 5600 MB Eindhoven, Netherlands; <sup>b</sup>Department of Econometrics and Operations Research, Tilburg University, 5000 LE Tilburg, Netherlands

\*Corresponding author

Contact: e.cardinaels@tue.nl,  <https://orcid.org/0000-0002-1350-9346> (EC); s.c.borst@tue.nl,  <https://orcid.org/0000-0003-3306-6447> (SB); J.S.H.vanLeeuwen@tilburguniversity.edu (JSHvL)

Received: May 20, 2021

Revised: January 7, 2022; August 16, 2022

Accepted: August 22, 2022

Published Online in Articles in Advance:  
December 5, 2022

Area of Review: Stochastic Models

<https://doi.org/10.1287/opre.2022.2385>

Copyright: © 2022 INFORMS

**Abstract.** Service systems often face task-server assignment constraints because of skill-based routing or geographical conditions. Redundancy scheduling responds to this limited flexibility by replicating tasks to specific servers in agreement with these assignment constraints. We gain insight from product-form stationary distributions and weak local stability conditions to establish a state space collapse in heavy traffic. In this limiting regime, the parallel-server system with redundancy scheduling operates as a multiclass single-server system, achieving full resource pooling and exhibiting strong insensitivity to the underlying assignment constraints. In particular, the performance of a fully flexible (unconstrained) system can be matched even with rather strict assignment constraints.

**Funding:** The work of S. Borst was partly supported by the Netherlands Organization for Scientific Research (NWO) through Gravitation [Grant NETWORKS-024.002.003]. The work of J. S. H. van Leeuwen was partly supported by VICI [Grant 202.068].

**Supplemental Material:** The e-companion is available at <https://doi.org/10.1287/opre.2022.2385>.

**Keywords:** assignment constraints • heterogeneity • parallel-server systems • load balancing • redundancy scheduling • heavy-traffic limit • state space collapse • resource pooling

## 1. Introduction

Task-server assignment constraints are ubiquitous in a broad range of everyday service systems. In contrast to fully flexible scenarios, where any task can be carried out by any server, assignment of tasks in these systems is restricted to a subset of the servers depending on the underlying features of the tasks and servers. This may potentially degrade the system performance and raises the question of how much flexibility in the task-server assignment constraints is needed to achieve performance levels comparable with those in a fully flexible system.

Assignment constraints play a particularly critical role in dynamic matching scenarios. For example, customers in a ride-sharing network will mainly be matched with drivers in their vicinity, and blood transfusions or organ transplants can only take place whenever there are both available and compatible donors and patients. Assignment constraints are also prevalent in customer contact centers. Skills of the various agents, like the spoken language or particular problem-solving skills, narrow the options for forwarding an incoming call. Skill-based resource management also plays a crucial role in health-care operations, where there are patients with specific conditions and specialized medical staff members.

Conceptually similar to these skill-based service models are computer systems, such as data center environments and cloud computing platforms. These systems support a continually evolving variety of applications, which involve not just increasing amounts of traffic but also, a growing diversity in task types. Even when tasks or servers are not intrinsically different, some servers tend to be better equipped to perform particular tasks because of data locality and network topology constraints. This notion is even more pronounced in manufacturing systems where available supply, expected demand, and involved costs may be dominant factors to decide which production plants should be able to produce which products.

Motivated by these observations, we set out to explore how assignment constraints impact the system performance in terms of queue lengths and delays. We assume that the assignment constraints between task types and servers are described in terms of a general bipartite graph, a so-called compatibility graph, and additionally, we allow for heterogeneous server speeds. In particular, we focus on parallel-server systems in a redundancy scheduling setting where replicas are created for each arriving task and then assigned to different servers. These could be either a subset of  $d$

servers selected uniformly at random in power-of- $d$  policies or an arbitrary subset of eligible servers in the context of the assignment constraints as described. As soon as the first of these replicas either starts service or completes service, the remaining ones are abandoned (referred to as the “cancel-on-start” (c.o.s.) and “cancel-on-completion” (c.o.c.) versions, respectively).

Dispatching replicas of the same task to several servers increases the chance for one of the replicas to find a short queue and thus, start service fast. A well-known application of this paradigm is *multiple listing*, where patients in need of an organ transplant register at multiple waiting lists. It is, for instance, shown by Zheng et al. (2022) that multiple listing for lung transplants reduces the waiting times and improves the probability of finding a suitable donor without affecting the waiting list mortality. The c.o.s. version of redundancy scheduling indeed resembles a Join-the-Smallest-Workload (JSW) policy with partial selection of servers (Adan et al. 2018, Ayesta et al. 2018). The c.o.c. version additionally increases the chance for one of the replicas to have a short run time (assuming independent run times on different servers). These relatively small jobs that experience a disproportionately longer run time in computer systems are referred to as *stragglers*; see, for instance, Ananthanarayanan et al. (2013) and Joshi (2018). On the flip side, the possibly concurrent execution of replicas creates a risk of potential wastage of capacity, depending on run time distributions. Moreover, the cost to cancel jobs in progress can be nonnegligible, urging caution in the implementation of the c.o.c. version; see, for instance, Shah et al. (2016), Joshi et al. (2017), and Lee et al. (2017), where performance trade-offs are investigated.

The launchpad for our analysis is provided by product-form distributions for redundancy systems with arbitrary compatibility graphs as obtained in the seminal papers by Gardner et al. (2016) and Ayesta et al. (2018). Although closed-form results for such complex systems are a rare luxury, the expressions in the literature depend on the compatibility graph in a highly intricate fashion and are unfortunately not particularly transparent. We, therefore, adopt a heavy-traffic perspective in order to extract the essential elements and obtain explicit insight into the impact of the assignment constraints on the performance. The heavy-traffic results reveal a remarkable universality property and in particular, indicate that the performance of a fully flexible system can asymptotically be matched even with rather strict assignment constraints as further discussed.

### 1.1. Related Literature

As mentioned, product-form distributions for the c.o.c. and c.o.s. versions of redundancy systems with general compatibility graphs were established by Gardner et al. (2016) and Ayesta et al. (2018), respectively. The latter results extended product-form distributions derived

earlier under more restrictive conditions by Visschers et al. (2012). Related product-form distributions for similar systems with assignment constraints and assign to longest idle server first (ALIS) policies were obtained by Adan and Weiss (2014). In fact, all these results turn out to be connected to product-form distributions for order-independent queues (Krzyszewski 2011), the concept of balanced fairness (Bonald and Comte 2017, Bonald et al. 2017), and token-based central queues (Ayesta et al. 2021). A recent overview of queueing models with task-server assignment constraints that yield product-form distributions is provided by Gardner and Righter (2020).

Although such product-form distributions are specified in closed form for general compatibility graphs, the expressions are unwieldy and yield no illuminating formulas for performance metrics, like mean queue lengths or delays. The expressions do not even readily lend themselves for computational purposes, except in specific scenarios where the compatibility graphs satisfy particular structural properties. For instance, for “nested structures,” the mean delays can be computed in an inductive fashion as shown by Gardner et al. (2017a) and Gardner and Righter (2020).

Fairly tractable expressions for response time distributions and mean response times have also been derived by Gardner et al. (2017b) and Hellekens and Van Houdt (2018) for the supermarket model with full flexibility, identical servers, and power-of- $d$  policies, which replicate jobs to each of the subsets of  $d$  servers with equal probability. Even though this notion of selective randomized replication is conceptually different from replication under assignment constraints, it can mathematically be described as a special instance. Because these scenarios are inherently symmetric and constrained to a specific structural family, however, they do not provide generic insight in the performance impact of assignment constraints.

In a different and broader strand of work, stochastic systems have extensively been considered in heavy-traffic regimes to provide greater tractability. Indeed, heavy-traffic limits have been established for a wide variety of multiclass parallel-server systems, with a broad range of both task assignment (routing, load balancing) policies and server allocation (scheduling, sequencing) strategies (Bramson 1998, Harrison 1998, Harrison and López 1999, Bell and Williams 2001). This body of literature is vast, and an exhaustive review is beyond the scope of this paper. In particular, the notion of state space collapse has been observed as a common phenomenon in a heavy-traffic regime in, for instance, stochastic processing networks and switched networks (Shah and Wischik 2012, Maguluri and Srikant 2015, Sharifnassab et al. 2020). A stochastic system is said to exhibit state space collapse if its multiclass limiting process has a lower-dimensional, often one-dimensional description in contrast to the original prelimiting process. The notion

of state space collapse has often been observed under natural complete resource pooling (commonly abbreviated as CRP) conditions that guarantee the system to behave asymptotically as a single-server queue with pooled resources. These combined concepts are studied, for instance, by Harrison and López (1999), Stolyar (2004, 2005), and Dai and Lin (2008).

By comparison, heavy-traffic results for redundancy systems have remained scarce, and the c.o.c. version with concurrent execution of tasks in fact seems to move beyond the conventional dynamics considered in the heavy-traffic literature. The few heavy-traffic results that do exist pertain to the c.o.s. version. In power-of- $d$  settings, which can be interpreted as special instances with highly symmetric compatibility graphs as noted, Atar et al. (2019a, b) obtain process-level Brownian limits for these models and demonstrate a state space collapse. Ayesta et al. (2018) use the product-form distributions to establish that the total queue length when properly scaled tends to a unit-exponential random variable but do not consider joint queue-length dynamics. Afèche et al. (2021) investigate the general setting with assignment constraints, but they focus on the expected delay and do not consider multidimensional queue-length processes and the associated state space collapse.

As mentioned earlier, the c.o.s. version of redundancy scheduling essentially mimics a JSW policy, which in turn, closely resembles a Join-the-Shortest-Queue (JSQ) strategy, especially in heavy-traffic conditions. We will discuss further related literature threads in the realm of JSQ strategies after presenting a detailed model description in Section 2.

## 1.2. Main Contributions

We examine how assignment constraints impact the performance of redundancy systems in terms of queue lengths and delays. We demonstrate that the performance impact tends to be limited, provided the assignment constraints and traffic composition satisfy a mild and natural assumption comparable with the mentioned CRP conditions. In particular, if the assignment constraints leave sufficient flexibility for the full service capacity to be used given the load proportions of the various task types, then it is ensured that the assignment constraints create no local capacity bottlenecks and that no subset of the servers can get overloaded as long as the total load is less than the total service capacity.

We establish that when the latter condition holds and traffic is Markovian, the system occupancy exhibits state space collapse in heavy traffic and asymptotically behaves as in a multiclass single-server first come, first served (FCFS) queue. Informally speaking, the number of tasks of each type remains in strict proportion to the arrival rates in the limit, whereas the total number of tasks, properly scaled, weakly converges to an

exponential random variable. Thus, the number of tasks of each type has an exponential limiting distribution as well. By virtue of the distributional form of Little's law, this means that job delay, after scaling, also has an exponential limiting distribution. Moreover, we extend the results to scenarios where local capacity bottlenecks could occur, and the queue lengths in a critically loaded subsystem exhibit state space collapse.

In order to prove the results for the c.o.c. mechanism, we start from the product-form distributions for arbitrary assignment constraints as studied by Gardner et al. (2016). We construct a specific enumeration of all the possible task configurations to write the joint probability-generating function (PGF) in a convenient form that facilitates a heavy-traffic analysis. The results for the c.o.s. mechanism are established by exploiting the relation between the product-form expressions and those studied by Ayesta et al. (2018).

The results reveal a remarkable universality property in the sense that the system achieves complete resource pooling and exhibits the same behavior across a broad range of scenarios, as long as no local capacity bottlenecks occur. In particular, the performance of a fully flexible system can be asymptotically matched, even under quite stringent assignment constraints. These results translate into several practical implications and guidelines. First, they indicate that a limited degree of flexibility, when properly designed, is sufficient to achieve full resource pooling. Adding greater flexibility provides only limited performance gains, although it may improve robustness if there is uncertainty in the server speeds or load proportions of the various job types. Second, under a fairly mild condition, the system obeys qualitatively similar scaling laws as if it were fully flexible, so that dimensioning approaches for such scenarios can be adopted without accounting for the assignment relations in great detail.

## 1.3. Organization of the Paper

In Section 2, we present a detailed model description and discuss some broader context and preliminaries, such as the product-form distributions that provide the starting point for our analysis. The main results are stated in Sections 3.1 and 3.2. Numerical results and observations are provided in Section 3.3. Section 4 contains the proofs for the c.o.c. mechanism. Some details and the proofs for the c.o.s. mechanism are deferred to the e-companion. We conclude with an outlook for further research in Section 5.

# 2. Model Description and Preliminaries

## 2.1. Model Description

We consider a system with  $N$  parallel servers with speeds  $\mu_1, \dots, \mu_N$  and several job types that correspond to (nonempty) subsets  $S \subseteq \{1, \dots, N\}$  of the servers.

Type  $S$  jobs arrive as a Poisson process of rate  $\lambda_S$  and are replicated to the servers in the subset  $S$ . This setup fits the premise that job assignment is subject to some constraints, like compatibility relations or data locality issues, as discussed in the introduction.

If we denote by  $\mathcal{S} = \{S \in 2^{\{1, \dots, N\}} : \lambda_S > 0\}$  the collection of job types, then the assignment constraints can be represented in terms of a bipartite graph, with nodes for each of the  $K = |\mathcal{S}|$  different job types on the one hand and nodes for each of the  $N$  servers on the other hand. A job-type node and a server node are connected by an edge in this compatibility graph whenever a job of this type is replicated to this server (Visschers et al. 2012, Adan and Weiss 2014, Gardner et al. 2016).

We distinguish between two different versions of redundancy scheduling, referred to as c.o.c. and c.o.s. In the c.o.c. version, as soon as the first replica of a particular job finishes service, the remaining replicas are discarded. The sizes of the replicas are independent and exponentially distributed with unit mean. In the c.o.s. version, the redundant replicas are already abandoned as soon as the first replica starts its service. The sizes of the replicas are also exponentially distributed with unit mean but do not need to be independent. In either case, each of the servers follows a FCFS discipline.

For compactness, denote by  $\lambda_{\text{tot}} := \sum_{S \subseteq \{1, \dots, N\}} \lambda_S$  the total arrival rate, and define  $\lambda := \lambda_{\text{tot}}/N$ . Because of the Poisson splitting property, we can equivalently think of the system as receiving jobs that arrive at rate  $\lambda_{\text{tot}}$  and are replicated to the servers in  $S \subseteq \{1, \dots, N\}$  with probability  $p_S := \lambda_S/\lambda_{\text{tot}}$ .

**Remark 1** (Power-of- $d$  Policies). We observe that so-called power-of- $d$  policies are subsumed as an important special case of our setup. These policies replicate jobs to a randomly selected subset of  $d$  servers and have been widely considered in the context of the supermarket model with full flexibility and without any assignment constraints (Gardner et al. 2017b). From a modeling perspective, however, they can be recovered as the special case where the job types correspond to all  $K = \binom{N}{d}$  different subsets of  $\{1, \dots, N\}$  of size  $d \leq N$  and  $p_S = 1/K$  for all such  $S$ . Our analysis is entirely cast in terms of the probabilities  $p_S$ , and it is immaterial whether these arise from selective replication of jobs (possibly nonuniform and/or randomized), underlying assignment constraints, or a combination of these two factors.

**Remark 2** (Fully Flexible System). We will explore how the system performance is impacted by the heterogeneity of the server speeds  $\mu_1, \dots, \mu_N$  and the assignment constraints in terms of the probabilities  $(p_S)_{S \in \mathcal{S}}$ . We examine under what conditions on  $\mu_1, \dots, \mu_N$  and  $p_S$  performance of a fully flexible system can be

approached. The fully flexible system corresponds to a scenario with homogeneous jobs that can be replicated to all servers (i.e.,  $\lambda_{\text{tot}} = \lambda_{\{1, \dots, N\}}$  and  $p_{\{1, \dots, N\}} = 1$ ). With  $M/M/C$  denoting a  $C$  server system subject to Markovian arrival and departures processes, the system then behaves as either an  $M/M/1$  FCFS queue with service rate  $\mu_{\text{tot}} := \sum_{n=1}^N \mu_n$  under the c.o.c. mechanism or as an  $M/M/N$  FCFS queue with heterogeneous server speeds under the c.o.s. mechanism.

The first-order performance criterion is the stability condition (Adan and Weiss 2014, Gardner et al. 2016).

**Assumption 1** (Stability Conditions). *Throughout, we assume that*

$$N\lambda \sum_{S \in \mathcal{T}} p_S < \sum_{n \in \cup_{S \in \mathcal{T}} S} \mu_n \quad (1)$$

for all (nonempty)  $\mathcal{T} \subseteq \mathcal{S}$  or equivalently,

$$N\lambda \sum_{S \subseteq U} p_S < \sum_{n \in U} \mu_n \quad (2)$$

for all (nonempty)  $U \subseteq \{1, \dots, N\}$ , which have been shown to be necessary and sufficient conditions for the system to be stable under the c.o.c. or c.o.s. mechanisms.

In particular, taking  $\mathcal{T} = \mathcal{S}$  or  $U = \{1, \dots, N\}$ , we see that  $\lambda < \mu$  is a necessary condition, with  $\mu := \mu_{\text{tot}}/N$  denoting the average service rate across all servers. Note that the left-hand side of (1) represents the total arrival rate of job types  $S \in \mathcal{T}$ , whereas the right-hand side measures the aggregate service rate of the servers that can help in handling jobs of these types. Likewise, the right-hand side of (2) represents the aggregate service rate of the servers in the set  $U$ , whereas the left-hand side captures the total arrival rate of job types that can be handled by these servers only.

As noted earlier, the c.o.s. version amounts to a JSW policy with partial selection of servers, which is covered by the framework of state-dependent assignment strategies by Foss and Chernova (1998). Their stability criteria for systems with “partial accessibility” indicate that the conditions are in fact necessary and sufficient for arbitrary job size distributions. Because of the possibly concurrent execution of replicas under the c.o.c. policy, the corresponding stability conditions for nonexponential job size distributions are challenging and have remained elusive so far.

In the power-of- $d$  scenario discussed, the stability conditions in (1) and (2) reduce to a set of just  $N - d + 1$  inequalities

$$N\lambda \frac{\binom{j}{d}}{\binom{N}{d}} < \sum_{n=1}^j \mu_{(n)}$$

for  $j = d, \dots, N$ , with  $\mu_{(n)}$  denoting the  $n$ th smallest service rate among the  $N$  servers. These inequalities reflect that the aggregate service rate of the  $j$  slowest servers should exceed the total arrival rate of jobs that are replicated to these servers only. In the case of identical service rates  $\mu_n \equiv \mu$  for all  $n = 1, \dots, N$ , the inequality for  $j = N$  is the most stringent one, yielding the stability condition  $\lambda < \mu$ , independent of the value of  $d$ , which is consistent with the result obtained by Gardner et al. (2017b) and Anton et al. (2021).

## 2.2. Further Related Literature

The c.o.s. version of redundancy scheduling basically emulates a JSW policy, which in turn, roughly behaves as a (weighted) JSQ policy, especially in a heavy-traffic regime. This is reflected by the fact that the mentioned framework of state-dependent assignment strategies by Foss and Chernova (1998) not only covers the JSW policy but also, the JSQ policy. Indeed, the stability conditions for the JSQ policy coincide with (1) and (2) for the JSW policy; see also Bramson (2011) and Cruise et al. (2020). The resemblance further manifests itself in the similarity between the process-level limits and state space collapse results for the c.o.s. mechanism in power-of- $d$  settings and those for JSQ( $d$ ) policies (i.e., power-of- $d$  versions of JSQ strategies) (Atar et al. 2019a, b). Further heavy-traffic results for JSQ( $d$ ) policies are obtained by Hurtado-Lange and Maguluri (2021) in a discrete-time setup as well as by Chen and Ye (2012) and Sloothaak et al. (2021) for nonuniform sampling variants.

Unlike the situation for redundancy scheduling, the existing literature does contain some results on the performance impact of assignment constraints on JSQ strategies that go beyond power-of- $d$  settings, albeit in a many-server scenario rather than a heavy-traffic regime. Specifically, Turner (1998), Gast (2015), Mukherjee et al. (2018), and Budhiraja et al. (2019) consider JSQ policies in network scenarios where the servers are arranged in a graph structure and each receive arriving jobs, which can be forwarded to their neighbors. In other words, the selection of subsets of servers is not done uniformly at random as in power-of- $d$  policies but governed by the neighborhood sets in a network graph with the servers as nodes. These network models can be viewed as a further class of special instances within the framework that we consider, with identical server speeds, uniform loads across the various job types, and a one-to-one correspondence between job types and servers.

The results of Turner (1998), He and Down (2008), and Gast (2015) pertain to certain fixed-degree graphs, in particular line graphs (He and Down 2008) and ring topologies (Turner 1998, Gast 2015). Their results demonstrate that the performance sensitively depends on the underlying graph topology and that sampling from

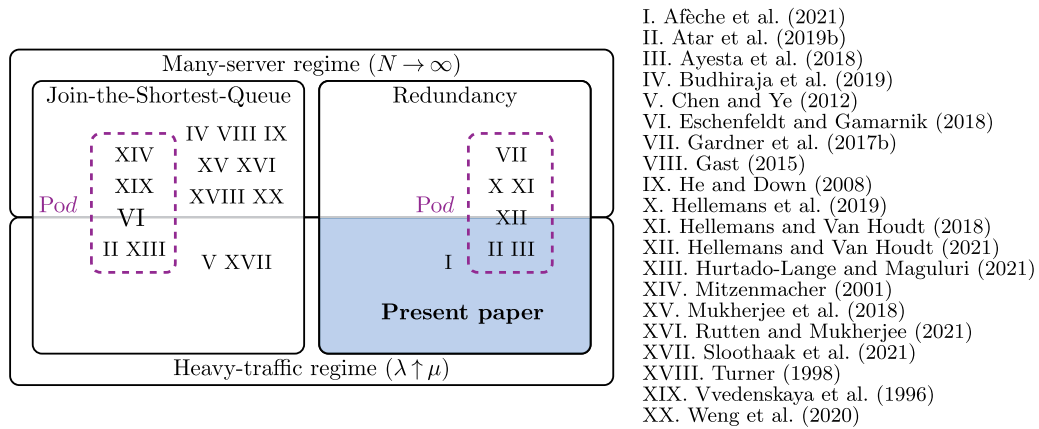
fixed neighborhood sets is typically outperformed by resampling the same number of alternate servers across the entire system.

In contrast, the results by Mukherjee et al. (2018) and Budhiraja et al. (2019) focus on cases where the degrees may grow with the total number of servers. Their results establish for a many-server regime conditions in terms of the density and topology of the network graph in order for JSQ and JSQ( $d$ ) policies to achieve asymptotically similar performance as in a fully connected graph. From a high level, conceptually related graph conditions for asymptotic optimality were examined using quite different techniques by Tsitsiklis and Xu (2013, 2017) in a dynamic scheduling framework (as opposed to a load-balancing context).

The recent papers by Weng et al. (2020) and Rutten and Mukherjee (2022) consider the same general setup with a bipartite compatibility graph as in the present paper, but they pursue a many-server regime and obtain results extending those by Mukherjee et al. (2018) and Budhiraja et al. (2019) to this more general setup. Informally speaking, both studies identify conditions in terms of the connectivity properties of the bipartite compatibility graph for similar performance to be achievable as in a fully flexible system. More specifically, Rutten and Mukherjee (2022) focus on scenarios with identical server speeds and uniform loads across the various job types, and they establish process-level limits indicating convergence of the system occupancy under JSQ( $d$ ) policies to that in the supermarket model with full flexibility. Weng et al. (2020) allow for heterogeneous server speeds and arbitrary load distributions, and they demonstrate that speed-aware extensions of the JSQ and Join-the-Idle-Queue strategies achieve vanishing waiting times and minimum expected sojourn times. Interestingly, the results by Weng et al. (2020) and Rutten and Mukherjee (2022) also entail a certain notion of universality, with similar achievable performance as in a fully flexible system under relatively sparse assignment constraints. However, in the many-server regime, this universality property does not manifest itself in terms of a state space collapse of the queue lengths but rather, the fluid-scaled system occupancy showing no queue buildup.

In summary, we map out the existing work in Figure 1 along two dimensions, with JSQ versus redundancy policies as the vertical axis and the many-server versus heavy-traffic regime as the horizontal axis. Literature focusing on the power-of- $d$  setting for the JSQ and redundancy policies, demarcated by dashed lines, is present in all four quadrants and reveals interesting similarities and contrasting features. The results by Eschenfeldt and Gamarnik (2017) and Hellemans and Van Houdt (2021) are positioned at the border of both

**Figure 1.** (Color online) A Taxonomy of the Literature on Scheduling Policies with Assignment Constraints in Asymptotic Regimes



Notes. The literature on power-of- $d$  (Pod) settings is demarcated by the dashed lines. (I. Afèche et al. 2021; II. Atar et al. 2019b; III. Ayesta et al. 2018; IV. Budhiraja et al. 2019; V. Chen and Ye 2012; VI. Eschenfeldt and Gamarnik 2017; VII. Gardner et al. 2017b; VIII. Gast 2015; IX. He and Down 2008; X. Hellemans et al. 2019; XI, XII. Hellemans and Van Houdt 2018, 2021; XIII. Hurtado-Lange and Maguluri 2021; XIV. Mitzenmacher 2001; XV. Mukherjee et al. 2018; XVI. Rutten and Mukherjee 2022; XVII. Sloothaak et al. 2021; XVIII. Turner 1998; XIX. Vvedenskaya et al. 1996; XX. Weng et al. 2020).

limiting regimes as they cover systems operating under the heavy-traffic many-server regime for the JSQ( $d$ ) policy and c.o.s. policy, respectively. However, the performance impact of general assignment constraints and heterogeneous server speeds, which falls outside the dashed lines in Figure 1, has mainly been pursued for JSQ-like strategies in a many-server regime and has barely received any attention so far for redundancy strategies or in a heavy-traffic regime. The only exception is the work by Afèche et al. (2021), who consider the design of reward-optimal bipartite compatibility graphs. However, their study is focused on expected delay as an optimization criterion, and they do not consider convergence of multidimensional queue-length processes for arbitrary assignment constraints and associated state space collapse properties.

### 2.3. Product-Form Distributions

In preparation for our analysis, we review in this subsection the existing product-form distributions for redundancy systems with assignment constraints as described.

**2.3.1. Redundancy Cancel-on-Completion.** The occupancy of the system at time  $t$  under the redundancy c.o.c. policy may be represented in terms of a vector  $(c_1, \dots, c_{M(t)})$ , with  $M(t)$  denoting the total number of jobs, including the ones in service, in the system at time  $t$  and  $c_m \in \mathcal{S}$  indicating the type of the  $m$ th oldest job at that time. It was shown by Gardner et al. (2016) that, if the stability conditions (1) and (2) are satisfied, the stationary distribution of the system occupancy is

$$\pi_{c.o.c.}(c) = \pi_{c.o.c.}(c_1, \dots, c_M) = C \prod_{i=1}^M \frac{N \lambda p_{c_i}}{\mu(c_1, \dots, c_i)}, \quad (3)$$

with  $C$  a normalization constant corresponding to the state without any job present and

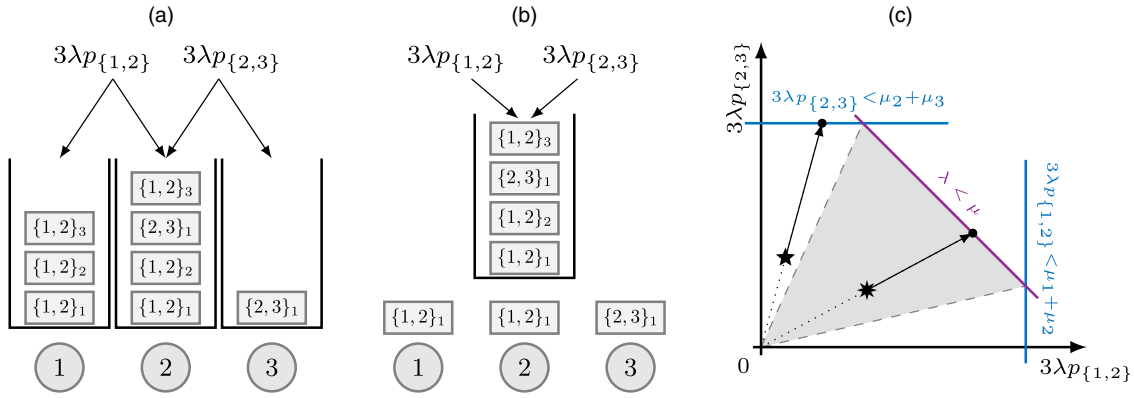
$$\mu(c_1, \dots, c_i) = \sum_{n \in \bigcup_{m=1}^i \{c_m\}} \mu_n. \quad (4)$$

Figure 2 visualizes two different representations of a particular state for a system with  $n = 3$  servers and the assignment constraints depicted at the top of Figure 2(a). Figure 2(a) shows how the replicas, belonging to the jobs in state  $c$ , are stored in separate queues in front of each of the compatible servers. Figure 2(b) visualizes the same state  $c$  from a modeling perspective where all the jobs are stored in a virtual central queue in order of arrival.

**2.3.2. Redundancy Cancel-on-Start.** As soon as a copy starts service at one of its compatible servers, the remaining replicas are instantaneously discarded from the system under the redundancy c.o.s. policy. In a situation where all servers are busy, a server that becomes available selects the longest waiting compatible job. Whenever a new job arrives and several compatible servers are idle, an assignment rule must be specified. Within the literature, three prominent rules can be distinguished: (i) assign uniformly at random, (ii) assign according to the so-called *assignment condition* (Visschers et al. 2012), and (iii) ALIS. In the present paper, we will focus on assignment rule (iii). The product form of an FCFS-ALIS policy for general assignment constraints was first derived by Adan and Weiss (2014); later, it was argued by Adan et al. (2018) and Ayesta et al. (2018) that it is equivalent to the product form



**Figure 2.** (Color online) Representations of the Redundancy c.o.c. Policy



Notes. (a) Representation with one queue per server and replicas. (b) Central queue representation. (c) The capacity region enclosed by the stability conditions (1) and (2). With the assignment constraints as indicated in panel (a) with  $n = 3$  servers, panels (a) and (b) give two different representations of the state  $c = (\{1, 2\}, \{1, 2\}, \{2, 3\}, \{1, 2\})$  of the system operating according to the redundancy c.o.c. policy. The notation  $\{i, j\}_k$  stands for the  $k$ th arrival of a type  $\{i, j\}$  job. For the initial arrival rate vectors  $3\lambda(p_{\{1,2\}}, p_{\{2,3\}})$  given by  $*$  and  $\star$  in panel (c), the critical normalized arrival rates  $\lambda^*$  are equal to  $\mu$  and  $(\mu_2 + \mu_3)/(3p_{\{2,3\}})$ , respectively. (a) Representation with one queue per server and replicas. (b) Central queue representation. (c) The capacity region enclosed by the stability conditions in (1) and (2).

under the redundancy c.o.s. policy with assignment rule (iii).

The occupancy of the system at time  $t$  under assignment rule (iii) can be represented as  $(c_1, \dots, c_{\tilde{M}(t)}; u_1, \dots, u_{L(t)})$ , with  $\tilde{M}(t)$  and  $L(t)$  denoting the total number of waiting jobs and the number of idle servers at time  $t$ , respectively. The type of the  $m$ th oldest waiting job is given by  $c_m \in \mathcal{S}$  and  $u_l \in \{1, \dots, N\}$  represents the  $l$ th longest waiting idle server. Note that none of the waiting jobs can be compatible with one of the idle servers and that the state descriptor omits the types of jobs that are in service. It was shown by Gardner and Righter (2020) that, if the stability conditions (1) and (2) are satisfied, the stationary distribution of the system occupancy is

$$\begin{aligned} \pi_{\text{c.o.s.}}(\mathbf{c}, \mathbf{u}) &= \pi_{\text{c.o.s.}}(c_1, \dots, c_{\tilde{M}}; u_1, \dots, u_L) \\ &= C' \prod_{i=1}^{\tilde{M}} \frac{N\lambda p_{c_i}}{\mu(c_1, \dots, c_i)} \prod_{l=1}^L \frac{\mu_{u_l}}{\lambda_{C(u_1, \dots, u_l)}}, \end{aligned} \quad (5)$$

with  $C'$  a normalization constant corresponding to the state where all servers are busy and no jobs are waiting,  $\mu(c_1, \dots, c_i)$  as in (4), and  $\lambda_{C(u_1, \dots, u_l)}$  the arrival rate of jobs that are compatible with (at least one of) the idle servers  $(u_1, \dots, u_l)$ : that is,

$$\lambda_{C(u_1, \dots, u_l)} = N\lambda \sum_{S: S \cap \{u_1, \dots, u_l\} \neq \emptyset} p_S.$$

### 3. Main Results

We now provide an overview of the main results, relegating the proofs to later sections. We first introduce some useful notation. Let  $(Q_S)_{S \in \mathcal{S}}$  and  $(\tilde{Q}_S)_{S \in \mathcal{S}}$  be random vectors with the stationary distribution of the total number of jobs of each type and the total number of

waiting jobs of each type, respectively. Let  $(R_n)_{n=1, \dots, N}$  be a random vector with the joint stationary distribution of the number of replicas assigned to each server. Note that not all  $R_n$  replicas assigned to server  $n$  will necessarily receive (let alone complete) service at that server before being discarded from the system because of the redundancy policy. The random variables with the stationary distribution of the sojourn time and waiting time of an arbitrary type  $S$  job are denoted by  $V_S$  and  $W_S$ , respectively.

For compactness, with minor abuse of notation, define  $p_{\mathcal{T}} := \sum_{S \in \mathcal{T}} p_S$  as the fraction of jobs that belong to the subset of job types  $\mathcal{T} \subseteq \mathcal{S}$ . Also, define

$$\mu_{\mathcal{T}} := \sum_{n \in \cup_{S \in \mathcal{T}} S} \mu_n$$

as the aggregate service rate of the servers that are compatible with the subset of job types  $\mathcal{T} \subseteq \mathcal{S}$ .

**Definition 1** (Critical Subset and Critical Arrival Rate). The ratio  $\rho_{\mathcal{T}} := p_{\mathcal{T}}/\mu_{\mathcal{T}}$  is called the capacity ratio of the subset of job types  $\mathcal{T} \subseteq \mathcal{S}$ . The subset  $\mathcal{T}^* := \arg \max_{\mathcal{T} \subseteq \mathcal{S}} \rho_{\mathcal{T}}$  with the maximum capacity ratio is called the critical subset, and  $\lambda^* = 1/(N\rho_{\mathcal{T}^*}) = \mu_{\mathcal{T}^*}/(Np_{\mathcal{T}^*})$  is called the critical (normalized) arrival rate.

Note that the stability conditions (1) and (2) may be equivalently written as  $\lambda < \lambda^*$ .

#### 3.1. Heavy-Traffic Regime Under the Local Stability Conditions

In this subsection, we assume that the probabilities  $(p_S)_{S \in \mathcal{S}}$  and the server speeds  $\mu_1, \dots, \mu_N$  satisfy the so-called local stability conditions.

**Assumption 2** (Local Stability Conditions). For all (nonempty)  $T \subsetneq S$ ,

$$p_T < \frac{1}{N\mu} \mu_T \quad (6)$$

with  $\mu = \frac{1}{N} \mu_{\text{tot}}$  the average service rate as defined earlier, or equivalently, for all (nonempty)  $U \subsetneq \{1, \dots, N\}$ ,

$$\sum_{S: S \subseteq U} p_S < \frac{1}{N\mu} \sum_{n \in U} \mu_n. \quad (7)$$

Note that we restrict to strict subsets  $T \subsetneq S$  and  $U \subsetneq \{1, \dots, N\}$ , as (6) and (7) hold with equality in cases  $T = S$  and  $U = \{1, \dots, N\}$ , respectively, by definition of the average service rate  $\mu$ . Then,  $\rho_T < 1/(N\mu) = \rho_S$  for all  $T \subsetneq S$ , so that the critical “subset” is  $T^* = S$  and the critical normalized arrival rate is  $\lambda^* = \mu$ .

Also, the inequalities in (6) and (7) imply that the stability conditions in (1) and (2) are satisfied for any  $\lambda < \mu$  (i.e., as long as the total arrival rate is strictly less than the aggregate service rate). More specifically, the strict inequalities in (6) ensure that as  $\lambda \uparrow \mu$ , for all (nonempty)  $T \subsetneq S$ , the total arrival rate of the job types in  $T$  remains bounded away from the aggregate service rate of the servers that can help in handling jobs of these types. More formally, it can be deduced that there exists an  $\epsilon > 0$  such that for all  $T \subsetneq S$ , we have that  $N\mu p_T + \epsilon < \mu_T$  and hence, also  $N\lambda p_T + \epsilon < \mu_T$  for all  $\lambda \leq \mu$ . Although the expression is closely related to the form of the inequalities in (6) and (7), it can be rewritten in a form that is better suited for application in the proofs later on. There exists an  $\epsilon > 0$  such that for all  $T \subsetneq S$ , we have that  $(N\mu p_T)/\mu_T < 1 - \epsilon$ . Similarly, the strict inequalities in (7) guarantee that as  $\lambda \uparrow \mu$ , for all (nonempty)  $U \subsetneq \{1, \dots, N\}$ , the aggregate service rate of the servers in the set  $U$  remains bounded away from the total arrival rate of job types that can be handled by these servers only. Thus, the inequalities in (6) and (7) ensure that there are no local capacity bottlenecks and that as  $\lambda \uparrow \mu$ , only the inequalities in (1) and (2) for  $T = S$  and  $U = \{1, \dots, N\}$ , respectively, are tight in the limit. We will henceforth refer to the inequalities in (6) and (7) as the *local stability conditions*.

For later use, we observe that the inequalities in (6) may also be written in the less intuitive but equivalent form

$$\frac{1}{N\mu} \sum_{n \in (\cup_{S: S \not\subseteq T'} S)^c} \mu_n < \sum_{S \in T'} p_S$$

for all (nonempty)  $T' = T^c \subsetneq S$ . These inequalities reflect that for all (nonempty)  $T'$ , the total arrival rate of job types  $S \in T'$  as  $\lambda \uparrow \mu$  should become strictly higher than the aggregate service rate of the servers that are

able to handle jobs of these types only. In particular, taking  $T' = \{S_0\}$ , we obtain

$$\frac{1}{N\mu} \sum_{n \in \mathcal{N}_{S_0}^c} \mu_n < p_{S_0}, \quad (8)$$

with  $\mathcal{N}_{S_0} = \cup_{S \in \mathcal{S} \setminus S_0} S$  and  $S_0$  an arbitrary job type belonging to  $\mathcal{S}$ . Likewise, the conditions in (7) may be rewritten as

$$\frac{1}{N\mu} \sum_{n \in U'} \mu_n < \sum_{S: S \cap U' \neq \emptyset} p_S$$

for all (nonempty)  $U' = U^c \subsetneq \{1, \dots, N\}$ . These inequalities indicate that for all (nonempty)  $U' \subsetneq \{1, \dots, N\}$ , the aggregate service rate of the servers in the set  $U$  should become strictly lower than the total arrival rate of the job types that can be handled by at least one of these servers as  $\lambda \uparrow \mu$ . In particular, taking  $U' = \{n_0\}$ , we see that any server  $n_0$  (with a strictly positive speed  $\mu_{n_0} > 0$ ) must be able to handle at least one job type  $S$  (with a strictly positive arrival probability  $p_S > 0$ ).

The next theorem establishes that detailed performance metrics, such as the queue lengths and delays, are not strongly affected by the exact values of the probabilities  $(p_S)_{S \in \mathcal{S}}$  as long as the local stability conditions (6) and (7) hold. We will write  $\xrightarrow{d}$  to denote convergence in distribution of random variables, and  $\text{Exp}(1)$  will represent a unit-mean exponentially distributed random variable.

**Theorem 1.** If the local stability conditions (6) and (7) hold, then for both the c.o.c. and c.o.s. mechanisms,

$$\left(1 - \frac{\lambda}{\mu}\right) (Q_S)_{S \in \mathcal{S}} \xrightarrow{d} \text{Exp}(1) (p_S)_{S \in \mathcal{S}},$$

as  $\lambda \uparrow \mu$ .

The theorem shows that the joint distribution of the number of jobs of each type, under both the c.o.c. and c.o.s. mechanisms, exhibits state space collapse in a heavy-traffic regime. Moreover, the joint stationary distribution coincides in the limit with the joint distribution of a multiclass M/M/1 queue with arrival rate  $N\lambda$ , service rate  $N\mu$ , and class probabilities  $(p_S)_{S \in \mathcal{S}}$ , provided the local stability conditions (6) and (7) are satisfied. In particular, the total number of jobs, properly scaled, tends to an exponential random variable in the limit, and thus, the number of jobs of each type has an exponential limiting distribution as well.

Theorem 1 may be interpreted as follows. It is highly unlikely that any server is idling when the total number of jobs is large because of the natural diversity in job types and the fact that any server is able to handle at least one job type as noted. In fact, although the

parallel-server system with a c.o.c. mechanism is not work conserving, the probability of any server being idle will vanish in the heavy-traffic limit. This can be formalized based on the arguments used to prove Theorem 1. For the c.o.s. mechanism, this is already shown in theorem 3.1 of Adan and Weiss (2014). Hence, the system will operate at the full aggregate service rate  $N\mu$  with high probability when the total number of jobs is sufficiently large. This explains why the total number of jobs behaves asymptotically the same as in an M/M/1 queue with arrival rate  $N\lambda$  and service rate  $N\mu$ , and in particular, follows the well-known scaled exponential distribution in the limit. In the special case of the power-of- $d$  policy, this result was already shown by Ayesta et al. (2018) for the c.o.s. mechanism.

What is far less evident, however, is the state space collapse (i.e., the proportion of type  $S$  jobs present in the system coincides with the corresponding arrival probability  $p_S$  for each job type  $S \in \mathcal{S}$ , like in a multi-class M/M/1 queue with an FCFS discipline). In order to provide an informal explanation, suppose that for a particular type  $S_0$ , the proportion of jobs is significantly lower than the corresponding arrival probability  $p_{S_0}$ , whereas the total number of jobs is large. Hence, in order to observe this unexpectedly low fraction of type  $S_0$  jobs, a large number of these type  $S_0$  jobs have been completed that arrived after jobs of other types that are still in the system. This, in turn, implies that type  $S_0$  jobs will only receive a nonvanishing fraction of the capacity of the servers in  $\mathcal{N}_{S_0}^c$  that cannot handle jobs of any other types, as all other compatible servers are processing earlier arrived jobs of different types. Here,  $\mathcal{N}_{S_0}^c$  is defined as  $\cup_{S \in \mathcal{S} \setminus \{S_0\}} S$ , the set of all servers compatible with the job types  $\mathcal{S} \setminus \{S_0\}$ . Because the type  $S_0$  jobs are not accumulating in the system, the aggregate service rate of the servers in  $\mathcal{N}_{S_0}^c$  is no less than the arrival rate of the type  $S_0$  jobs (i.e., the inequality in (8) is violated), which yields a contradiction. Hence, we conclude that for none of the job types, the proportion of jobs can be significantly lower than the corresponding arrival probability, meaning that the state space collapse occurs.

The local stability conditions (6) and (7) may be interpreted as so-called CRP conditions. Such conditions have emerged as an ubiquitous concept in the heavy-traffic behavior of stochastic networks and in particular, play a paramount role in the occurrence of a (one-dimensional) state space collapse. CRP conditions have been formulated in roughly three different, yet related, ways in the literature: (i) linear programming characterizations, (ii) geometric interpretations, and (iii) systems of linear inequalities. All three notions pertain to the relative position of the critical load vector on the boundary of the capacity region of the system and obviously, involve the relevant system parameters

(e.g., service rates, compatibility constraints between job types and servers, and proportions of job types). However, the various representations differ in the degree to which either the parameter values or the intrinsic properties of interest are explicitly covered.

The first type of characterization as introduced by Harrison (1998) and Harrison and López (1999) requires that the solution to a certain linear program (the dual version of a linear program describing feasible resource allocation options) is unique. The second representation stipulates that the normal vector to the boundary of the capacity region at the critical load vector is unique with strictly positive components; see, for instance, Mandelbaum and Stolyar (2004), Stolyar (2004), and Hurtado-Lange and Maguluri (2020). The third kind of characterization involves a system of linear equalities in terms of the system parameters; see, for instance, the seminal papers by Laws (1992) and Kelly and Laws (1993), which refer to these inequalities as “generalized cut constraints,” and the more recent studies by Shi et al. (2019), Banerjee et al. (2020), and Varma and Maguluri (2021), which develop related notions. The local stability conditions (6) and (7) are similar in spirit as the ones in the latter category, which take a particularly convenient form in the context of a parallel-server system. We have opted to state the CRP condition in this form because the linear equalities provide an explicit characterization in terms of the system parameters, which is straightforward to verify and additionally captures how the CRP condition is used in the proof arguments. However, the local stability conditions can be shown to be equivalent to a conceptually similar linear programming construction as in (i) or a geometric representation in a similar vein as in (ii). The latter two characterizations, both specialized to the setting of a parallel-server system with compatibility constraints, are provided and further discussed in Section EC.1 of the e-companion, where the equivalence with the inequalities in (6) and (7) is also established.

### 3.2. Heavy-Traffic Regime in a General Scenario

Although the local stability conditions (6) and (7) are natural and desirable design objectives, one can definitely conceive scenarios where these inequalities may not be satisfied. In order to illustrate this, Figure 2(c) visualizes the capacity region for the system depicted in Figure 2(a). As can be seen here, for probabilities  $(p_S)_{S \in \mathcal{S}}$  for which the vector  $\lambda(p_S)_{S \in \mathcal{S}}$  lies within the cone region (indicated in gray), this vector  $\lambda(p_S)_{S \in \mathcal{S}}$  will indeed reach the boundary of the capacity region corresponding to the stability condition  $\lambda < \mu$  when  $\lambda$  increases. In other words, the local stability conditions are met. This implies that the critical subset is  $T^* = \mathcal{S}$  and the normalized critical relative arrival rate is  $\lambda^* = \mu$ . For probabilities  $(p_S)_{S \in \mathcal{S}}$  with corresponding arrival rate vectors

outside this cone region, however, the vector  $\lambda(p_S)_{S \in \mathcal{S}}$  could already reach the boundary of the capacity region when  $\lambda$  approaches some  $\lambda^* < \mu$ , and hence,  $\mathcal{T}^*$  is a strict subset of  $\mathcal{S}$ . In general, starting from the stability conditions in (1) and increasing  $\lambda$  will result in a subset  $\mathcal{T}^* \subseteq \mathcal{S}$  of job types such that the inequality  $N\lambda p_{\mathcal{T}^*} < \mu_{\mathcal{T}^*}$  becomes tight when  $\lambda \uparrow \lambda^*$ . Note that  $\mathcal{T}^*$  and  $\lambda^*$  are precisely the critical subset and critical normalized arrival rate as defined in Definition 1.

We now extend Theorem 1 to such a scenario with  $\lambda^* < \mu$  and  $\mathcal{T}^* \neq \mathcal{S}$  under the following mild assumption.

**Assumption 3.** Let the capacity ratio  $\rho_{\mathcal{T}}$  for all  $\mathcal{T} \subseteq \mathcal{S}$  and the critical subset  $\mathcal{T}^*$  be as specified in Definition 1. Then, the subset  $\mathcal{T}^*$  that maximizes the capacity ratio  $\rho_{\mathcal{T}}$  is unique (i.e., the vector of critical arrival rates can reach any point on the boundary of the capacity region except for its extreme vertices).

**Theorem 2.** Let  $\mathcal{T}^*$ ,  $p_{\mathcal{T}^*}$ , and  $\lambda^*$  be as in Definition 1, satisfying Assumption 3. Then, for both the c.o.c. and c.o.s. mechanisms,

$$\left(1 - \frac{\lambda}{\lambda^*}\right)(Q_S)_{S \in \mathcal{S}} = \left(1 - \frac{\lambda}{\lambda^*}\right)\left((Q_S)_{S \in \mathcal{T}^*}, (Q_S)_{S \notin \mathcal{T}^*}\right) \xrightarrow{d} \left(\text{Exp}(1)\left(\frac{p_S}{p_{\mathcal{T}^*}}\right)_{S \in \mathcal{T}^*}, (0)_{S \notin \mathcal{T}^*}\right),$$

as  $\lambda \uparrow \lambda^*$ .

Hence, a similar state space collapse occurs as in Theorem 1 for the number of jobs of each type in  $\mathcal{T}^*$ . On the other hand, the number of jobs of types not in  $\mathcal{T}^*$  (i.e., job types for which the aggregate arrival rate to the compatible servers is subcritical) becomes negligible after scaling.

From Theorem 2, we can derive the following two corollaries.

**Corollary 1.** Let  $\mathcal{T}^*$ ,  $p_{\mathcal{T}^*}$ , and  $\lambda^*$  be as in Definition 1, satisfying Assumption 3. Then, for both the c.o.c. and c.o.s. mechanisms,

$$\left(1 - \frac{\lambda}{\lambda^*}\right)(R_n)_{n=1, \dots, N} \xrightarrow{d} \text{Exp}(1)(q_n)_{n=1, \dots, N},$$

as  $\lambda \uparrow \lambda^*$ , with  $q_n = \sum_{S \in \mathcal{T}^* : n \in S} \frac{p_S}{p_{\mathcal{T}^*}}$ ,  $n = 1, \dots, N$ .

Note that the summation in the definition of  $q_n$  is equal to zero when server  $n$  is not compatible with any of the job types in  $\mathcal{T}^*$ . Hence, its queue length becomes negligible, after scaling, in the heavy-traffic regime.

By virtue of the distributional form of Little's law (Keilson and Servi 1988), the sojourn time and waiting time of a type  $S$  job, properly scaled, also have an exponential distribution in the limit when  $S \in \mathcal{T}^*$ . When  $S \notin \mathcal{T}^*$ , we know from the previous result that the (scaled) queue lengths at its compatible servers are

zero; hence, its (scaled) sojourn time and waiting time are zero as well.

**Corollary 2.** Let  $\mathcal{T}^*$ ,  $p_{\mathcal{T}^*}$ , and  $\lambda^*$  be as in Definition 1, satisfying Assumption 3. Then, for both the c.o.c. and c.o.s. mechanisms, for  $S \in \mathcal{T}^*$ ,

$$\left(1 - \frac{\lambda}{\lambda^*}\right)V_S \xrightarrow{d} \rho_{\mathcal{T}^*} \text{Exp}(1), \left(1 - \frac{\lambda}{\lambda^*}\right)W_S \xrightarrow{d} \rho_{\mathcal{T}^*} \text{Exp}(1),$$

and for  $S \notin \mathcal{T}^*$ ,

$$\left(1 - \frac{\lambda}{\lambda^*}\right)V_S \xrightarrow{d} 0, \left(1 - \frac{\lambda}{\lambda^*}\right)W_S \xrightarrow{d} 0,$$

as  $\lambda \uparrow \lambda^*$ .

This result shows that, for instance, when the local stability conditions in (6) and (7) hold (i.e.,  $\mathcal{T}^* \equiv \mathcal{S}$ ), the sojourn time and waiting time of a particular type of job asymptotically do not depend on the probabilities  $(p_S)_{S \in \mathcal{S}}$  in any way as  $\rho_{\mathcal{T}^*} = p_{\mathcal{T}^*} / \mu_{\mathcal{T}^*} = 1 / (N\mu)$ . Thus, surprisingly, jobs of types with more compatible servers do not enjoy significantly shorter waiting times or sojourn times in the limit.

The proof of Theorem 2 for the c.o.c. mechanism, presented in Section 4, relies on a specific enumeration of all possible job configurations, which yields a particularly convenient form of the joint PGF of the number of jobs of each type as provided in the next proposition.

**Proposition 1.** Assuming that the stability conditions (1) and (2) are satisfied, the joint PGF of the number of jobs of each type for the redundancy c.o.c. policy is given by

$$\mathbb{E} \left[ \prod_{S \in \mathcal{S}} z_S^{Q_S} \right] = \frac{f(\mathbf{z})}{f(\mathbf{1})}, \quad (9)$$

where  $\mathbf{z}$  and  $\mathbf{1}$  are  $|\mathcal{S}|$ -dimensional vectors with entries  $|z_S| \leq 1$  and

$$f(\mathbf{z}) = 1 + \sum_{m=1}^{|\mathcal{S}|} \sum_{S \in \mathcal{S}_m} \prod_{j=1}^m \frac{N\lambda p_{S_j} z_{S_j}}{\mu(S_1, \dots, S_j)} \prod_{j=1}^m \left( 1 - \frac{N\lambda}{\mu(S_1, \dots, S_j)} \sum_{i=1}^j p_{S_i} z_{S_i} \right)^{-1}.$$

The  $m$ -dimensional vector  $\mathbf{S}$  consists of  $m$  different job types, and the set consisting of all these vectors is denoted by  $\mathcal{S}_m$ .

An interpretation of this PGF in terms of the ordered vectors  $\mathbf{S}$  of job types and geometrically distributed random variables can be found in Section EC.2 of the e-companion. The proof of Proposition 1 for the c.o.c. mechanism and the close relationship between the product-form expressions (3) and (5) can be exploited

to prove Theorem 2 for the c.o.s. mechanism. This derivation is given in Section EC.3.1 of the e-companion. Moreover, a proof technique similar to the one outlined for the c.o.c. mechanism can also be applied to prove the results for the c.o.s. mechanism, where first, the joint PGF of the number of waiting jobs is derived before taking the appropriate limit. This proof is deferred to Section EC.3.2 of the e-companion.

Having exact expressions for the PGFs of the number of jobs of each type as in Proposition 1 and Proposition EC.1 in the e-companion allows us to establish convergence of the (scaled)  $n$ th moments of both the total number of jobs,  $Q$ , and  $Q_S$ , besides the convergence in distribution in Theorem 1.

**Theorem 3.** *If the local stability conditions (6) and (7) hold, then for both the c.o.c. and c.o.s. mechanisms,*

$$\lim_{\lambda \uparrow \lambda^*} \mathbb{E} \left[ \left( \left( 1 - \frac{\lambda}{\mu} \right) Q \right)^n \right] = n! = \mathbb{E}[(\text{Exp}(1))^n]$$

and

$$\lim_{\lambda \uparrow \mu} \mathbb{E} \left[ \left( \left( 1 - \frac{\lambda}{\mu} \right) Q_S \right)^n \right] = n! (p_S)^n = \mathbb{E}[(\text{Exp}(1)p_S)^n]$$

for any  $n \geq 1$  and  $S \in \mathcal{S}$ .

The proofs of Theorem 3, including explicit expressions for the  $n$ th moments of  $Q$  and  $Q_S$ , are deferred to Section EC.4 of the e-companion.

The results focus on the job types that experience critical load and show that the contribution of the noncritical job types to the total number of jobs becomes negligible after scaling. Hence, the full system comprises two subsystems. The critical subsystem consists of all job types in  $\mathcal{T}^*$  and their compatible servers, and the noncritical subsystem consists of all job types  $\mathcal{S} \setminus \mathcal{T}^*$  and their compatible servers. Note that the server sets of both subsystems do not need to be disjoint. However, along the same lines as the proof of Theorem 2, it can be shown that the noncritical subsystem in the heavy-traffic regime operates as an isolated system with servers that are only compatible with types outside of  $\mathcal{T}^*$ , referred to as the *truncated* system, as formalized in the following theorem.

**Theorem 4.** *Let  $\mathcal{T}^*$ ,  $p_{\mathcal{T}^*}$ , and  $\lambda^*$  be as in Definition 1, satisfying Assumption 3. For the c.o.c. policy,*

$$\left( \left( 1 - \frac{\lambda}{\lambda^*} \right) (Q_S)_{S \in \mathcal{T}^*}, (Q_S)_{S \notin \mathcal{T}^*} \right) \xrightarrow{d} \left( \text{Exp}(1) \left( \frac{p_S}{p_{\mathcal{T}^*}} \right)_{S \in \mathcal{T}^*}, (Q_S^*)_{S \notin \mathcal{T}^*} \right)$$

as  $\lambda \uparrow \lambda^*$ , and for the c.o.s. policy,

$$\left( \left( 1 - \frac{\lambda}{\lambda^*} \right) (\tilde{Q}_S)_{S \in \mathcal{T}^*}, (\tilde{Q}_S)_{S \notin \mathcal{T}^*} \right) \xrightarrow{d} \left( \text{Exp}(1) \left( \frac{p_S}{p_{\mathcal{T}^*}} \right)_{S \in \mathcal{T}^*}, (\tilde{Q}_S^*)_{S \notin \mathcal{T}^*} \right)$$

as  $\lambda \uparrow \lambda^*$ . Here,  $Q_S^*$  and  $\tilde{Q}_S^*$  denote the number of type  $S$  jobs and the number of waiting type  $S$  jobs, respectively, in a truncated system as described.

The theorem implies that the full system decomposes into two disjoint and independent subsystems in the heavy-traffic regime. A more formal definition of the truncated system, additional information, and a sketch of the proof are deferred to Section EC.5 of the e-companion.

### 3.3. Numerical Results and Discussion

In this subsection, we present numerical results to illustrate the heavy-traffic limits and discuss some design implications.

**3.3.1. Numerical Illustration.** We focus on a system with three identical servers and two job types labeled  $\{1, 2\}$  and  $\{2, 3\}$  as schematically represented in Figure 2(a). For each arriving job, replicas are assigned to either servers 1 and 2 or servers 2 and 3 depending on its type. The joint stationary distribution of the number of jobs under the c.o.c. mechanism can be derived through some straightforward but tedious calculations from the more detailed stationary distribution (3). Specifically,  $\mathbb{P}\{(Q_{\{1,2\}}, Q_{\{2,3\}}) = (q, q')\}$  equals

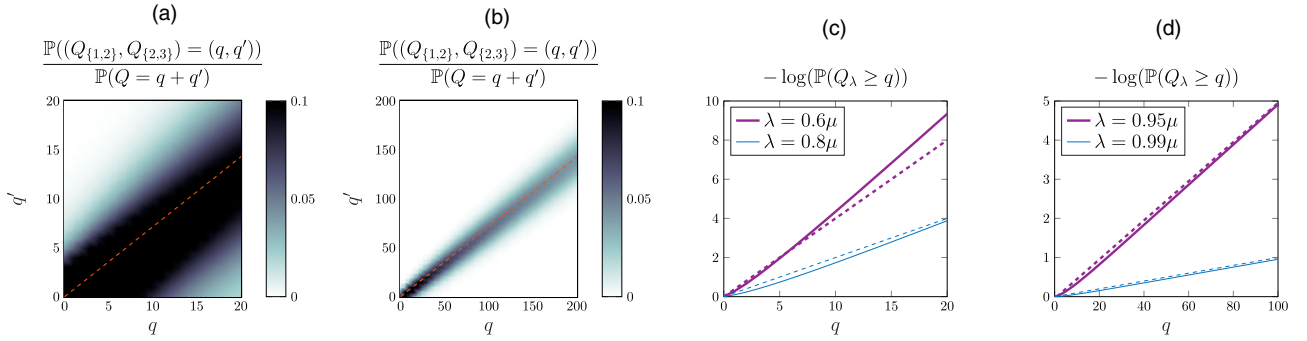
$$\begin{cases} C & \text{if } q = q' = 0 \\ C \left( \frac{N\lambda p_{\{1,2\}}}{2\mu} \right)^q & \text{if } q \geq 1, q' = 0 \\ C \left( \frac{N\lambda p_{\{2,3\}}}{2\mu} \right)^{q'} & \text{if } q = 0, q' \geq 1 \\ C \left( \frac{N\lambda p_{\{1,2\}}}{3\mu} \right)^q \left( \frac{N\lambda p_{\{2,3\}}}{3\mu} \right)^{q'} \left[ \sum_{k=1}^q \binom{q+q'-k-1}{q'-1} \left( \frac{3}{2} \right)^k \right. \\ \quad \left. + \sum_{k=1}^{q'} \binom{q+q'-k-1}{q-1} \left( \frac{3}{2} \right)^k \right] & \text{if } q, q' \geq 1. \end{cases}$$

The normalization constant  $C$  is given by

$$\frac{3(2\mu - N\lambda p_{\{1,2\}})(2\mu - N\lambda p_{\{2,3\}})(\mu - \lambda)}{(N\lambda)^2 p_{\{1,2\}} p_{\{2,3\}} \mu + 12\mu^2(\mu - \lambda)},$$

as was also established by Gardner et al. (2016). In order to produce numerical results, we set the values of the job type fractions to  $p_{\{1,2\}} = 7/12$  and  $p_{\{2,3\}} = 5/12$ , satisfying the local stability conditions in (6) and (7), which in this particular case, read  $p_{\{1,2\}}, p_{\{2,3\}} < 2/3$ .

Figure 3, (a) and (b) corroborates the state space collapse result. The (scaled) probability mass function strongly concentrates around the dashed line  $\{(p_{\{1,2\}}, p_{\{2,3\}})x : x \geq 0\}$  with slope 5/7, which represents the heavy-traffic limit. Remarkably, this not only holds for a high load of 0.99 but already manifests itself for a

**Figure 3.** (Color online) Numerical Illustration of the Result in Theorem 1

*Notes.* The system consists of three identical servers, has type fractions  $(p_{\{1,2\}}, p_{\{2,3\}}) = (7/12, 5/12)$  and a compatibility graph as indicated in Figure 2. Panels (a) and (b) visualize the (scaled) joint probability mass function  $\mathbb{P}\{(Q_{\{1,2\}}, Q_{\{2,3\}}) = (q, q')\}$  when  $\lambda = 0.6\mu$  and  $\lambda = 0.99\mu$ , respectively. The (dashed) line with slope  $5/7$  depicts the limiting regime. Panels (c) and (d) give a comparison between the cumulative distributions of the total number of jobs (solid line) and the random variable  $\text{Exp}(1)\left(1 - \frac{\lambda}{\mu}\right)^{-1}$  (dashed line) for various values of  $\lambda$ .

moderate load value of 0.6 (note the 10-fold difference in scale).

Figure 3, (c) and (d) confirms that the total number of jobs, properly scaled, converges to a unit exponential random variable (i.e.,  $\left(1 - \frac{\lambda}{\mu}\right)Q_\lambda \rightarrow \text{Exp}(1)$  as  $\lambda \uparrow \mu$ ), with  $Q_\lambda = Q_{\{1,2\}} + Q_{\{2,3\}}$ . We observe that the heavy-traffic approximation

$$\begin{aligned} \mathbb{P}\{Q_\lambda \geq q\} &= \mathbb{P}\left\{\left(1 - \frac{\lambda}{\mu}\right)Q_\lambda \geq \left(1 - \frac{\lambda}{\mu}\right)q\right\} \\ &\approx \mathbb{P}\left\{\text{Exp}(1) \geq \left(1 - \frac{\lambda}{\mu}\right)q\right\} = e^{-(1-\frac{\lambda}{\mu})q} \quad (10) \end{aligned}$$

is not only nearly exact for high load values, like 0.99 and 0.95, but also fairly close for moderate load values, like 0.8 and 0.6, outside the asymptotic regime. Only for the tail probabilities does the accuracy starts to diminish. Note that for a load value of 0.6, the probability that, for instance,  $q = 15$  is exceeded is around  $4.7 \times 10^{-4}$ .

The system with three servers and two job types is admittedly a toy model. The product-form distributions in (3), in fact, hold for systems of any size and arbitrary compatibility constraints. As noted earlier, however, these expressions do not lend themselves well for computational purposes in more complex situations.

**3.3.2. Simulation Results.** For the c.o.s. mechanism, we conducted simulations for systems with  $n = 4, 10,$  or  $20$  homogeneous servers and a variety of compatibility graphs. These compatibility graphs range from fairly sparse (e.g.,  $N$  job types with type  $n$  jobs only compatible with servers  $n$  and  $n + 1$ ) to fairly dense (e.g., power-of- $d$  setting with  $d$  close to  $N$ ). It is observed that

for a small system, the heavy-traffic limit in (10) provides a reasonable approximation for the total number of jobs in the system (i.e.,  $Q_\lambda \approx \text{Exp}(1)\left(1 - \frac{\lambda}{\mu}\right)^{-1}$ ).

For larger systems, this still seems to hold true when the compatibility graph is rather dense, meaning that each job type is compatible with a large fraction of servers. However, when a type  $S$  job arrives with probability  $p_S$ , we observe that the number of type  $S$  jobs is approximately equal to  $p_S Q_\lambda$ . More detailed simulation results are provided in Section EC.6 of the e-companion.

**3.3.3. Design Implications.** The universality property embodied in Theorem 1 implies that the performance of a fully flexible system can asymptotically be matched as long as the local stability conditions are satisfied. In practical terms, this means that creating a limited amount of flexibility in the assignment constraints, if done judiciously, is sufficient and that an excess of flexibility yields only minor performance gains. Similar observations have been made in the context of process flexibility in manufacturing systems; see, for instance, Jordan and Graves (1995), Graves and Tomlin (2003), Simchi-Levi and Wei (2012), and Shi et al. (2019).

To illustrate the observation, suppose that we set the probabilities as

$$\begin{aligned} p_n &= \frac{N-1-\epsilon}{N(N-1)\mu} \mu_n \quad \text{for all } n = 1, \dots, N \quad \text{and} \\ p_{\{1, \dots, N\}} &= \frac{\epsilon}{N-1}. \end{aligned}$$

Hence, most jobs are assigned to a single server, and a few jobs are replicated to all servers. Then, the average

replication degree is  $1 + \epsilon$ , whereas for all (nonempty) subsets  $U \subsetneq \{1, \dots, N\}$ , we have for any  $\epsilon > 0$ ,

$$\sum_{S: S \subseteq U} p_S = \sum_{n \in U} p_n = \frac{N-1-\epsilon}{N(N-1)\mu} \sum_{n \in U} \mu_n < \frac{1}{N\mu} \sum_{n \in U} \mu_n,$$

implying that the local stability conditions in (7) are satisfied.

Note that for  $\epsilon = 0$  and a setting with identical servers, the described scenario reduces to a system of  $N$ -independent M/M/1 queues with arrival rate  $\lambda$  and service rate  $\mu$ . In that case, the number of jobs at each server has a geometric stationary distribution with parameter  $\lambda/\mu$  and when scaled with  $1 - \lambda/\mu$ , tends to a unit-exponential distribution as  $\lambda \uparrow \mu$ . Thus, the scaled total number of jobs converges to the sum of  $N$ -independent unit-exponential random variables. In contrast, for any  $\epsilon > 0$ , Theorem 1 implies that the total number of jobs tends to a single-unit exponential random variable and is, hence, smaller by a factor  $N$ .

In the example, we observed that the performance at high load is roughly similar for all values  $\epsilon > 0$  and significantly better than for  $\epsilon = 0$ . This observation is reminiscent of the finding that a power-of-two policy provides a significant improvement over purely random assignment to a single server in a regime where the number of servers  $N$  grows large, whereas  $\lambda$  remains fixed (Gardner et al. 2017b). (Interestingly, the significant benefit of “just a little” flexibility has also been encountered in different resource-sharing contexts; see, for instance, Fleming and Simon (1999) and Tsitsiklis and Xu (2012).) The fact that the heavy-traffic regime and a many-server scenario point to similar behavior also suggests that it would be interesting to explore joint scalings.

## 4. Proofs

The proof of the heavy-traffic result stated in Theorem 2, which implies the result in Theorem 1, relies on a well-suited expression for the joint PGF of the number of jobs of each type. This expression, which may be of independent interest, is provided in Proposition 1 and involves a specific enumeration of all possible job configurations as explained in the proof.

**Proof of Proposition 1.** Using the stationary distribution given in (3), the joint PGF of the number of jobs of each type may be written as

$$\mathbb{E} \left[ \prod_{S \in \mathcal{S}} z_S^{Q_S} \right] = \sum_{M=0}^{\infty} \sum_{(c_1, \dots, c_M) \in \mathcal{S}^M} \pi_{c.o.c.}(c_1, \dots, c_M) \prod_{S \in \mathcal{S}} z_S^{q_S^c}, \quad (11)$$

with  $z$  an  $|\mathcal{S}|$ -dimensional vector with entries  $|z_S| \leq 1$  and  $q_S^c$  the total number of type  $S$  jobs in state  $c = (c_1, \dots, c_M)$ . The summation on the right-hand side in (11) over all possible states can be conducted in three steps.

i. Fix the number of different job types  $m$  that occur in the state  $c$ ,  $m = 1, \dots, |\mathcal{S}|$ .

ii. Fix  $m$  distinct job types and the order in which they occur,  $S = [S_1, \dots, S_m]$ . This implies that the oldest job in the system is of type  $S_1$ ; that the possible following jobs are of the same type; and that the first time a different type is observed, it will be a job of type  $S_2$ , etc. The set containing all these vectors of length  $m$  is denoted by  $\mathcal{S}_m$ .

iii. Sum over all states with this particular order. For instance, for the vector  $[S_1, S_2, S_3] \in \mathcal{S}_3$  with only three job types, one has to sum over all states

$$c = (S_1, \underbrace{S_1, S_1}_{k_1}, S_2, \underbrace{\times, \times, \times}_{k_2}, S_3, \underbrace{\circ, \circ, \circ}_{k_3}),$$

where  $\times$  denotes jobs of types  $S_1$  and/or types  $S_2$  and  $\circ$  denotes jobs of types  $S_1$ ,  $S_2$ , and/or  $S_3$ . The values  $k_1$ ,  $k_2$ , and  $k_3$  can be any natural number.

The third step might warrant some illustration. For example, the contribution of the ordered vector  $[S_1, S_2, S_3]$  to (11) can be computed by first determining how many jobs there are present in total,  $M \geq 3$ . Then, the values of  $k_1$ ,  $k_2$ , and  $k_3 = M - k_1 - k_2 - 3$  are set. Finally, the  $k_2$  and  $k_3$  intermediate jobs are labeled as type  $S_1$  or type  $S_2$  jobs and type  $S_1$ , type  $S_2$ , or type  $S_3$  jobs, respectively. Relying on (3), the total contribution to (11) is then given by

$$\begin{aligned} & C \sum_{M=3}^{\infty} \frac{N\lambda p_{S_1 z_{S_1}}}{\mu(S_1)} \sum_{k_1=0}^{M-3} \left[ \left( \frac{N\lambda p_{S_1 z_{S_1}}}{\mu(S_1)} \right)^{k_1} \frac{N\lambda p_{S_2 z_{S_2}}}{\mu(S_1, S_2)} \right. \\ & \quad \left. \sum_{k_2=0}^{M-3-k_1} \left[ \left( \frac{N\lambda}{\mu(S_1, S_2)} \right)^{k_2} \sum_{l=0}^{k_2} \binom{k_2}{l} (p_{S_1 z_{S_1}})^l (p_{S_2 z_{S_2}})^{k_2-l} \right. \right. \\ & \quad \left. \left. \frac{N\lambda p_{S_3 z_{S_3}}}{\mu(S_1, S_2, S_3)} \left( \frac{N\lambda}{\mu(S_1, S_2, S_3)} \right)^{k_3} \right. \right. \\ & \quad \left. \left. \sum_{l_1=0}^{k_3} \sum_{l_2=0}^{k_3-l_1} \left[ \binom{k_3}{l_1, l_2, k_3-l_1-l_2} (p_{S_1 z_{S_1}})^{l_1} \right. \right. \right. \\ & \quad \left. \left. \left. (p_{S_2 z_{S_2}})^{l_2} (p_{S_3 z_{S_3}})^{k_3-l_1-l_2} \right] \right] \right]. \end{aligned}$$

Applying the multinomial of Newton leads to

$$\begin{aligned} & C \prod_{j=1}^3 \frac{N\lambda p_{S_j z_{S_j}}}{\mu(S_1, \dots, S_j)} \sum_{M=3}^{\infty} \sum_{k_1=0}^{M-3} \left[ \left( \frac{N\lambda}{\mu(S_1)} p_{S_1 z_{S_1}} \right)^{k_1} \right. \\ & \quad \left. \sum_{k_2=0}^{M-3-k_1} \left[ \left( \frac{N\lambda}{\mu(S_1, S_2)} \right)^{k_2} (p_{S_1 z_{S_1}} + p_{S_2 z_{S_2}})^{k_2} \right. \right. \\ & \quad \left. \left. \left( \frac{N\lambda}{\mu(S_1, S_2, S_3)} \right)^{k_3} (p_{S_1 z_{S_1}} + p_{S_2 z_{S_2}} + p_{S_3 z_{S_3}})^{k_3} \right] \right]. \end{aligned}$$

Interchanging the order of summation results in

$$C \prod_{j=1}^3 \frac{N\lambda p_{S_j z_{S_j}}}{\mu(S_1, \dots, S_j)} \left[ \sum_{k_1=0}^{\infty} \left( \frac{N\lambda p_{S_1 z_{S_1}}}{\mu(S_1)} \right)^{k_1} \right] \\ \left[ \sum_{k_2=0}^{\infty} \left( \frac{N\lambda}{\mu(S_1, S_2)} \right)^{k_2} (p_{S_1 z_{S_1}} + p_{S_2 z_{S_2}})^{k_2} \right] \\ \left[ \sum_{k_3=0}^{\infty} \left( \frac{N\lambda}{\mu(S_1, S_2, S_3)} \right)^{k_3} (p_{S_1 z_{S_1}} + p_{S_2 z_{S_2}} + p_{S_3 z_{S_3}})^{k_3} \right].$$

Because of the stability conditions in (1) and (2), the expression for the infinite geometric sum may be applied to obtain

$$C \prod_{j=1}^3 \frac{N\lambda p_{S_j z_{S_j}}}{\mu(S_1, \dots, S_j)} \\ \prod_{j=1}^3 \left( 1 - \frac{N\lambda}{\mu(S_1, \dots, S_j)} (p_{S_1 z_{S_1}} + \dots + p_{S_j z_{S_j}}) \right)^{-1}.$$

Generalizing the reasoning and applying the mentioned three steps will give an expression for (11), namely

$$C \left[ 1 + \sum_{m=1}^{|S|} \sum_{S \in \mathcal{S}_m} \prod_{j=1}^m \frac{N\lambda p_{S_j z_{S_j}}}{\mu(S_1, \dots, S_j)} \right. \\ \left. \prod_{j=1}^m \left( 1 - \frac{N\lambda}{\mu(S_1, \dots, S_j)} \sum_{i=1}^j p_{S_i z_{S_i}} \right)^{-1} \right].$$

The three steps only consider states with at least one job, and the contribution of the empty state can be seen in the additional one. Now, substituting  $z_S = 1$  in (11) for all  $S \in \mathcal{S}$  should give one as a result, and this yields an expression for the normalization constant  $C$ . This concludes the derivation of the joint PGF (9).  $\square$

A similar result is derived by Ayesta et al. (2021) concerning the generating function of both the total and the waiting numbers of jobs of each type in the token-based central queue setting, which includes among others matching models and redundancy models. Because of a slightly different state description and an alternative enumeration of the states, these generating functions still consist of infinite sums, including general expressions for the probability that particular servers are processing particular job types.

Circumventing the obstacles by directly using the product-form expressions for the stationary distribution in (3) allows us to study the heavy-traffic limit in Theorem 2 for redundancy c.o.c. by interchanging the summation and limit operator in the expression provided in Proposition 1.

**Proof of Theorem 2** (c.o.c. Mechanism). Let  $\mathcal{T}^* \subseteq \mathcal{S}$  be the critical subset,  $p_{\mathcal{T}^*}$ ,  $\mu_{\mathcal{T}^*}$ , and  $\lambda^* = \mu_{\mathcal{T}^*} / (Np_{\mathcal{T}^*})$  as

defined in Definition 1. We will prove the following heavy-traffic behavior of the moment-generating function (MGF) of the number of jobs of each type  $(Q_S)_{S \in \mathcal{S}}$ :

$$\mathbb{E} \left[ \exp \left( - \left( 1 - \frac{\lambda}{\lambda^*} \right) \sum_{S \in \mathcal{S}} t_S Q_S \right) \right] \rightarrow \left( 1 + \sum_{S \in \mathcal{T}^*} \frac{p_S}{p_{\mathcal{T}^*}} t_S \right)^{-1}, \quad (12)$$

as  $\lambda \uparrow \lambda^*$  and  $t_S \geq 0$  for all  $S \in \mathcal{S}$ . Moreover, it can easily be seen that the MGF of the random vector  $\mathbf{X} := (\text{Exp}(1)(p_S/p_{\mathcal{T}^*})_{S \in \mathcal{T}^*}, (0)_{S \notin \mathcal{T}^*})$  is given by

$$\mathbb{E} \left[ \prod_{S \in \mathcal{T}^*} \exp \left( -t_S \left( \frac{p_S}{p_{\mathcal{T}^*}} \text{Exp}(1) \right) \right) \right] \\ = \mathbb{E} \left[ \exp \left( - \left( \sum_{S \in \mathcal{T}^*} \frac{p_S}{p_{\mathcal{T}^*}} t_S \right) \text{Exp}(1) \right) \right] \\ = \left( 1 + \sum_{S \in \mathcal{T}^*} \frac{p_S}{p_{\mathcal{T}^*}} t_S \right)^{-1}. \quad (13)$$

By Feller's convergence theorem (Feller 1971), the nonnegative random vector  $\left( 1 - \frac{\lambda}{\lambda^*} \right) (Q_S)_{S \in \mathcal{S}}$  converges in distribution to the random vector  $\mathbf{X}$  when its MGF converges pointwise to the MGF in (13). Hence, it is sufficient to show that (12) holds in order to conclude the result stated in Theorem 2.

To obtain the MGF of  $\left( 1 - \frac{\lambda}{\lambda^*} \right) (Q_S)_{S \in \mathcal{S}}$ , define  $z_S := \exp \left( - \left( 1 - \frac{\lambda}{\lambda^*} \right) t_S \right)$ , and use the expression for the PGF in (9). As we allow  $t_S \geq 0$ , it follows that  $|z_S| \leq 1$ . For any  $m = 1, \dots, |S|$ , for any  $S = [S_1, \dots, S_m] \in \mathcal{S}_m$ , and for any  $j = 1, \dots, m$ , it can be observed that

$$\lim_{\lambda \uparrow \lambda^*} \frac{N\lambda}{\mu(S_1, \dots, S_j)} p_{S_j z_{S_j}} = \frac{\mu_{\mathcal{T}^*}}{\mu(S_1, \dots, S_j)} \frac{p_{S_j}}{p_{\mathcal{T}^*}} \in (0, \infty).$$

From Assumption 3, it can be deduced that there exists some  $\epsilon > 0$  such that for all  $\mathcal{T} \subsetneq \mathcal{T}^*$ , there holds that  $\frac{N\lambda p_{\mathcal{T}}}{\mu_{\mathcal{T}}} = \frac{\mu_{\mathcal{T}^*} p_{\mathcal{T}}}{\mu_{\mathcal{T}} p_{\mathcal{T}^*}} < 1 - \epsilon$ . Hence,

$$\lim_{\lambda \uparrow \lambda^*} \left( 1 - \frac{N\lambda}{\mu(S_1, \dots, S_j)} \sum_{i=1}^j p_{S_i z_{S_i}} \right)^{-1} \\ = \begin{cases} \infty & \text{if } \{S_1, \dots, S_j\} = \mathcal{T}^*, \\ \left( 1 - \frac{\mu_{\mathcal{T}^*}}{\mu(S_1, \dots, S_j)} \sum_{i=1}^j \frac{p_{S_i}}{p_{\mathcal{T}^*}} \right)^{-1} \in (0, \infty) & \text{otherwise.} \end{cases}$$

Therefore, the dominating terms in both the numerator and denominator of (9) in the heavy-traffic regime are those with  $S \in \mathcal{S}_m$  such that  $m \geq |\mathcal{T}^*|$  and  $\{S_1, \dots, S_{|\mathcal{T}^*}|\} = \mathcal{T}^*$ . Let  $\mathcal{S}^{\mathcal{T}^*}$  denote the set of all vectors  $S$  that satisfy this property. Note that if  $\mathcal{T}^*$  is contained in  $S$  but not as the  $|\mathcal{T}^*|$  first occurring job types, this  $S$  will only have a finite contribution to the value in the numerator and



denominator because of the observations. This leads to

$$\lim_{\lambda \uparrow \lambda^*} \mathbb{E} \left[ \exp \left( - \left( 1 - \frac{\lambda}{\lambda^*} \right) \sum_{S \in \mathcal{S}} t_S Q_S \right) \right]$$

$$= \lim_{\lambda \uparrow \lambda^*} \frac{\sum_{S \in \mathcal{S}^{T^*}} \prod_{j=1}^{|S|} \frac{N \lambda p_{S_j} z_{S_j}}{\mu(S_1, \dots, S_j)} \prod_{j=1}^{|S|} \left( 1 - \frac{N \lambda}{\mu(S_1, \dots, S_j)} \sum_{i=1}^j p_{S_i} z_{S_i} \right)^{-1}}{\sum_{S \in \mathcal{S}^{T^*}} \prod_{j=1}^{|S|} \frac{N \lambda p_{S_j}}{\mu(S_1, \dots, S_j)} \prod_{j=1}^{|S|} \left( 1 - \frac{N \lambda}{\mu(S_1, \dots, S_j)} \sum_{i=1}^j p_{S_i} \right)^{-1}}.$$

Because  $\mu(S_1, \dots, S_{|T^*|}) = \mu_{T^*}$  and  $\sum_{i=1}^{|T^*|} p_{S_i} = p_{T^*}$ , the fraction can be rewritten as

$$\lim_{\lambda \uparrow \lambda^*} \frac{\sum_{S \in \mathcal{S}^{T^*}} \prod_{j=1}^{|S|} \frac{N \lambda p_{S_j} z_{S_j}}{\mu(S_1, \dots, S_j)} \prod_{j=1}^{|S|} \left( 1 - \frac{N \lambda}{\mu(S_1, \dots, S_j)} \sum_{i=1}^j p_{S_i} z_{S_i} \right)^{-1}_{j \neq |T^*|}}{\sum_{S \in \mathcal{S}^{T^*}} \prod_{j=1}^{|S|} \frac{N \lambda p_{S_j}}{\mu(S_1, \dots, S_j)} \prod_{j=1}^{|S|} \left( 1 - \frac{N \lambda}{\mu(S_1, \dots, S_j)} \sum_{i=1}^j p_{S_i} \right)^{-1}_{j \neq |T^*|}}$$

$$\cdot \lim_{\lambda \uparrow \lambda^*} \frac{1 - \frac{N \lambda}{\mu_{T^*}} p_{T^*}}{1 - \frac{N \lambda}{\mu_{T^*}} \sum_{S \in T^*} p_S z_S}.$$

The first limit evaluates to one because of the observations, and after applying l'Hôpital's rule, the second limit indeed evaluates to the right-hand side of (13). This concludes the proof of Theorem 1.  $\square$

The proofs of Corollaries 1 and 2 are deferred to Section EC.7 of the e-companion.

## 5. Outlook

The broader lay of the land and paucity of results for parallel-server systems with arbitrary assignment constraints as visualized in Figure 1 suggest a few natural directions for further research.

First of all, the papers of, in particular, Weng et al. (2020) and Rutten and Mukherjee (2022) are the primary counterparts of the present paper for JSQ-type strategies in a many-server scenario rather than redundancy scheduling in a heavy-traffic regime. Surprisingly, the results in these two papers also entail a certain notion of universality, with similar achievable performance as in a fully flexible system under relatively stringent assignment constraints. Although this universality property manifests itself in a different form in a many-server scenario, it suggests that this paradigm may not just apply to a given policy in a given limiting regime but in fact, unifies and spans across different conditions and different policies, with JSW providing a natural bridge between JSQ and redundancy scheduling as mentioned earlier. In particular, we strongly conjecture that similar heavy-traffic results as derived in the present paper for redundancy scheduling hold for JSQ policies (up to speed-dependent weight factors), except that these would need to be established in terms of process-level limits as they are

obtained by Atar et al. (2019b) for power-of- $d$  settings in the absence of any explicit stationary distribution. Likewise, it would be interesting to investigate whether similar many-server asymptotics as obtained by Weng et al. (2020) and Rutten and Mukherjee (2022) apply for redundancy scheduling models, for which the same product-form distributions as used in the present paper would provide a natural tool set.

A further research direction would be to use the PGFs to establish convergence rates and refined approximations with improved accuracy in prelimit scenarios of moderate instead of high load. Although the PGFs exist in closed form, the analysis would be both numerically and analytically challenging because of the intricate dependencies on the assignment constraints.

## Acknowledgments

The authors thank the two anonymous reviewers and the associate editor for their insightful comments and further thank Céline Comte for helpful discussions that have led to the generalized result in Theorem 2.

## References

- Adan I, Weiss G (2014) A skill based parallel service system under FCFS-ALIS—steady state, overloads, and abandonments. *Stochastic Systems* 4(1):250–299.
- Adan I, Kleiner I, Richter R, Weiss G (2018) FCFS parallel service systems and matching models. *Performance Evaluation* 127–128: 253–272.
- Afèche P, Caldentey R, Gupta V (2021) On the optimal design of a bipartite matching queueing system. *Oper. Res.* 70(1): 363–401.
- Ananthanarayanan G, Ghodsi A, Shenker S, Stoica I (2013) Effective straggler mitigation: Attack of the clones. *Proc. 10th USENIX Conf. Networked Systems Design Implementation* (USENIX, Lombard, IL), 185–198.
- Anton E, Ayesta U, Jonckheere M, Verloop IM (2021) On the stability of redundancy models. *Oper. Res.* 69(5):1540–1565.
- Atar R, Keslassy I, Mendelson G (2019a) Replicate to the shortest queues. *Queueing Systems* 92(1–2):1–23.
- Atar R, Keslassy I, Mendelson G (2019b) Subdiffusive load balancing in time-varying queueing systems. *Oper. Res.* 67(6): 1678–1698.
- Ayesta U, Bodas T, Verloop IM (2018) On a unifying product form framework for redundancy models. *Performance Evaluation* 127–128:93–119.
- Ayesta U, Bodas T, Dorsman JL, Verloop IM (2021) A token-based central queue with order-independent service rates. *Oper. Res.* 70(1):545–561.
- Banerjee S, Kanoria Y, Qian P (2020) Dynamic assignment control of a closed queueing network under complete resource pooling. Preprint, submitted June 25, <https://arxiv.org/abs/1803.04959>.
- Bell SL, Williams RJ (2001) Dynamic scheduling of a system with two parallel servers in heavy traffic with resource pooling: Asymptotic optimality of a threshold policy. *Ann. Appl. Probab.* 11(3):608–649.
- Bonald T, Comte C (2017) Balanced fair resource sharing in computer clusters. *Performance Evaluation* 116:70–83.
- Bonald T, Comte C, Mathieu F (2017) Performance of balanced fairness in resource pools: A recursive approach. *Proc. ACM Measurement Anal. Comput. Systems* 1(2):1–25.

- Bramson M (1998) State space collapse with application to heavy traffic limits for multiclass queueing networks. *Queueing Systems* 30(1-2):89–148.
- Bramson M (2011) Stability of join the shortest queue networks. *Ann. Appl. Probab.* 21(4):1568–1625.
- Budhiraja A, Mukherjee D, Wu R (2019) Supermarket model on graphs. *Ann. Appl. Probab.* 29(3):1740–1777.
- Chen H, Ye HQ (2012) Asymptotic optimality of balanced routing. *Oper. Res.* 60(1):163–179.
- Cruise J, Jonckheere M, Shneer S (2020) Stability of JSQ in queues with general server-job class compatibilities. *Queueing Systems* 95(3):271–279.
- Dai JG, Lin W (2008) Asymptotic optimality of maximum pressure policies in stochastic processing networks. *Ann. Appl. Probab.* 18(6):2239–2299.
- Eschenfeldt P, Gamarnik D (2017) Supermarket queueing system in the heavy traffic regime. Short queue dynamics. Preprint, submitted January 17, <https://arxiv.org/abs/1610.03522>.
- Feller W (1971) *An Introduction to Probability Theory and Its Applications*, 2nd ed., Wiley Series in Probability and Statistics, vol. 2 (Wiley, Hoboken, NJ).
- Fleming PJ, Simon B (1999) Heavy-traffic approximations for a system of infinite servers with load balancing. *Probab. Engrg. Inform. Sci.* 13(3):251–273.
- Foss SG, Chernova NI (1998) On the stability of a partially accessible multi-station queue with state-dependent routing. *Queueing Systems* 29(1):55–73.
- Gardner K, Righter R (2020) Product forms for FCFS queueing models with arbitrary server-job compatibilities: An overview. *Queueing Systems* 96(1–2):3–51.
- Gardner K, Harchol-Balter M, Hyytiä E, Righter R (2017a) Scheduling for efficiency and fairness in systems with redundancy. *Performance Evaluation* 116:1–25.
- Gardner K, Harchol-Balter M, Scheller-Wolf A, Velednitsky M, Zbarsky S (2017b) Redundancy- $d$ : The power of  $d$  choices for redundancy. *Oper. Res.* 65(4):1078–1094.
- Gardner K, Zbarsky S, Doroudi S, Harchol-Balter M, Hyytiä E, Scheller-Wolf A (2016) Queueing with redundant requests: Exact analysis. *Queueing Systems* 83(3-4):227–259.
- Gast N (2015) The power of two choices on graphs: The pair approximation is accurate? *ACM SIGMETRICS Performance Evaluation Rev.* 43(2):69–71.
- Graves SC, Tomlin BT (2003) Process flexibility in supply chains. *Management Sci.* 49(7):907–919.
- Harrison JM (1998) Heavy traffic analysis of a system with parallel servers: Asymptotic optimality of discrete-review policies. *Ann. Appl. Probab.* 8(3):822–848.
- Harrison JM, López MJ (1999) Heavy traffic resource pooling in parallel server systems. *Queueing Systems* 33(4):339–368.
- He YT, Down DG (2008) Limited choice and locality considerations for load balancing. *Performance Evaluation* 65(9):670–687.
- Hellemans T, Van Houdt B (2018) On the power-of- $d$ -choices with least loaded server selection. *Proc. ACM Measurement Anal. Comput. Systems* 2(2):1–22.
- Hellemans T, Van Houdt B (2021) Mean waiting time in large-scale and critically loaded power of  $d$  load balancing systems. *Proc. ACM Measurement Anal. Comput. Systems* 5(2):1–34.
- Hellemans T, Bodas T, Van Houdt B (2019) Performance analysis of workload dependent load balancing policies. *Proc. ACM Measurement Anal. Comput. Systems* 3(2):1–35.
- Hurtado-Lange D, Maguluri ST (2020) Transform methods for heavy-traffic analysis. *Stochastic Systems* 10(4):275–309.
- Hurtado-Lange D, Maguluri ST (2021) Throughput and delay optimality of power-of- $d$  choices in inhomogeneous load balancing systems. *Oper. Res. Lett.* 49(4):616–622.
- Jordan WC, Graves SC (1995) Principles on the benefits of manufacturing process flexibility. *Management Sci.* 41(4):577–594.
- Joshi G (2018) Synergy via redundancy: Boosting service capacity with adaptive replication. *ACM SIGMETRICS Performance Evaluation Rev.* 45(3):21–28.
- Joshi G, Soljanin E, Wornell G (2017) Efficient redundancy techniques for latency reduction in cloud systems. *ACM Trans. Modeling Performance Evaluation Comput. Systems* 2(2):12.
- Keilson J, Servi LD (1988) A distributional form of Little's law. *Oper. Res. Lett.* 7(5):223–227.
- Kelly FP, Laws CN (1993) Dynamic routing in open queueing networks: Brownian models, cut constraints and resource pooling. *Queueing Systems* 13(1–3):47–86.
- Krzesinski AE (2011) Order independent queues. Boucherie R, van Dijk N, eds. *Queueing Networks*, International Series in Operations Research and Management Science, vol. 154 (Springer, Boston), 85–120.
- Laws CN (1992) Resource pooling in queueing networks with dynamic routing. *Adv. Appl. Probab.* 24(3):699–726.
- Lee K, Pedarsani R, Ramchandran K (2017) On scheduling redundant requests with cancellation overheads. *IEEE/ACM Trans. Networking* 25(2):1279–1290.
- Maguluri ST, Srikant R (2015) Heavy-traffic behavior of the Max-weight algorithm in a switch with uniform traffic. *ACM SIGMETRICS Performance Evaluation Rev.* 43(2):72–74.
- Mandelbaum A, Stolyar AL (2004) Scheduling flexible servers with convex delay costs: Heavy-traffic optimality of the generalized  $c\mu$ -rule. *Oper. Res.* 52(6):836–855.
- Mitzenmacher M (2001) The power of two choices in randomized load balancing. *IEEE Trans. Parallel Distributed Systems* 12(10):1094–1104.
- Mukherjee D, Borst SC, van Leeuwen JSH (2018) Asymptotically optimal load balancing topologies. *Proc. ACM Measurement Anal. Comput. Systems* 2(1):1–29.
- Rutten D, Mukherjee D (2022) Load balancing under strict compatibility constraints. *Math. Oper. Res.*, ePub ahead of print April 20, <https://doi.org/10.1287/moor.2022.1258>.
- Shah D, Wischik D (2012) Switched networks with maximum weight policies: Fluid approximation and multiplicative state space collapse. *Ann. Appl. Probab.* 22(1):70–127.
- Shah NB, Lee K, Ramchandran K (2016) When do redundant requests reduce latency? *IEEE Trans. Comm.* 64(2):715–722.
- Sharifnassab A, Tsitsiklis JN, Golestani SJ (2020) Fluctuation bounds for the Max-weight policy with applications to state space collapse. *Stochastic Systems* 10(3):223–250.
- Shi C, Wei Y, Zhong Y (2019) Process flexibility for multiperiod production systems. *Oper. Res.* 67(5):1300–1320.
- Simchi-Levi D, Wei Y (2012) Understanding the performance of the long chain and sparse designs in process flexibility. *Oper. Res.* 60(5):1125–1141.
- Sloothak F, Cruise J, Shneer S, Vlasiou M, Zwart B (2021) Complete resource pooling of a load-balancing policy for a network of battery swapping stations. *Queueing Systems* 99(1–2):65–120.
- Stolyar AL (2004) MaxWeight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic. *Ann. Appl. Probab.* 14(1):1–53.
- Stolyar AL (2005) Optimal routing in output-queued flexible server systems. *Probab. Engrg. Inform. Sci.* 19(2):141–189.
- Tsitsiklis JN, Xu K (2012) On the power of (even a little) resource pooling. *Stochastic Systems* 2(1):1–66.
- Tsitsiklis JN, Xu K (2013) Queueing system topologies with limited flexibility. *ACM SIGMETRICS Performance Evaluation Rev.* 41(1):167–178.
- Tsitsiklis JN, Xu K (2017) Flexible queueing architectures. *Oper. Res.* 5(65):1398–1413.
- Turner SRE (1998) The effect of increasing routing choice on resource pooling. *Probab. Engrg. Inform. Sci.* 12(1):109–124.
- Varma SM, Maguluri ST (2021) Transportation polytope and its applications in parallel server systems. Preprint, submitted August 11, <https://arxiv.org/abs/2108.13167>.

- Vischers J, Adan I, Weiss G (2012) A product form solution to a system with multi-type jobs and multi-type servers. *Queueing Systems* 70(3):269–298.
- Vvedenskaya ND, Dobrushin RL, Karpelevich FI (1996) Queueing system with selection of the shortest of two queues: An asymptotic approach. *Problemy Peredachi Informatsii* 32(1):20–34.
- Weng W, Zhou X, Srikant R (2020) Optimal load balancing with locality constraints. *Proc. ACM Measurement Anal. Comput. Systems* 4(3):1–37.
- Zheng L, Chandratre S, Ali A, Szabo A, Durham L, Joyce LD, Joyce DL (2022) How does multiple listing affect lung transplantation? A retrospective analysis. *Seminars Thoracic Cardiovascular Surgery* 34(1):326–335.

---

**Ellen Cardinaels** is a PhD candidate in the Department of Mathematics & Computer Science at Eindhoven University of Technology. Her research interests include performance modeling and the asymptotic analysis of stochastic service systems. In particular, in her PhD research, she focuses on the analysis of load balancing strategies in systems subject

to assignment constraints induced by, for instance, job-server compatibility relations or data locality.

**Sem Borst** is a full professor in stochastic operations research in the Department of Mathematics & Computer Science at Eindhoven University of Technology. His main research interests are in the area of performance evaluation and resource allocation algorithms for large-scale stochastic networks, in particular computer-communication systems. Sem was recipient of the 2017 ACM SIGMETRICS Achievement Award and delivered the 2020 INFORMS APS (Applied Probability Society) Markov Lecture.

**Johan S. H. van Leeuwen** is a full professor in stochastic operations research in the econometrics and operations research department at the Tilburg School of Economics and Management. As a mathematician, his research interests include probability theory, complex analysis, stochastic processes, queueing theory, random graphs, and stochastic optimization. Some of his research is applied for decision making under uncertainty and the design of large-scale systems.