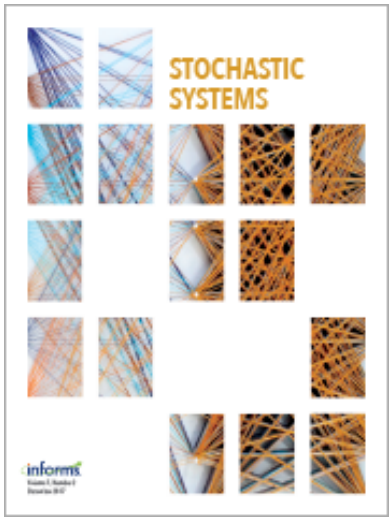

Stochastic Systems

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

A Skill Based Parallel Service System Under FCFS-ALIS — Steady State, Overloads, and Abandonments

Ivo Adan, Gideon Weiss

To cite this article:
Ivo Adan, Gideon Weiss (2014) A Skill Based Parallel Service System Under FCFS-ALIS — Steady State, Overloads, and Abandonments. Stochastic Systems 4(1):250-299. https://doi.org/10.1287/13-SSY117

Stochastic Systems
2014, Vol. 4, No. 1, 250–299
DOI: 10.1214/13-SSY117

A SKILL BASED PARALLEL SERVICE SYSTEM UNDER FCFS-ALIS — STEADY STATE, OVERLOADS, AND ABANDONMENTS

By Ivo Adan[*,‡], and Gideon Weiss[†,§]

Eindhoven University of Technology[‡] and The University of Haifa[§]

We consider a queueing system with J parallel servers $\mathcal{S} = \{m_1, \ldots, m_J\}$, and with customer types $\mathcal{C} = \{a, b, \ldots\}$. A bipartite graph G describes which pairs of server-customer types are compatible. We consider FCFS-ALIS policy: A server always picks the first, longest waiting compatible customer, and a customer is always assigned to the longest idle compatible server. We assume Poisson arrivals and server dependent exponential service times. We derive an explicit product-form expression for the stationary distribution of this system when service capacity is sufficient. We also calculate fluid limits of the system under overload, to show that local steady state exists. We distinguish the case of complete resource pooling when all the customers are served at the same rate by the pooled servers, and the case when the system has a unique decomposition into subsets of customer types, each of which is served at its own rate by a pooled subset of the servers. Finally, we discuss possible behavior of the system with generally distributed abandonments, under many server scaling. This paper complements and extends previous results of Kaplan, Caldentey and Weiss [18], and of Whitt and Talreja [34], as well as previous results of the authors [4, 35] on this topic.

1. Introduction.

1.1. *Background and motivation.* It is an inconvenient fact that most queueing models are complicated and often intractable. This is true even for the single server queue, with the shining exception of the M/M/1 queue, which in stationary form has exponential sojourn times and geometric $(1-\rho)\rho^n$ queue length distribution, and for which many other quantities can

Received June 2013.

[*]Research supported in part by the Netherlands Organization for Scientific Research (NWO).

[†]Research supported in part by Israel Science Foundation Grants 711/09 and 286/13.

MSC 2010 subject classifications: Primary 60K25; secondary 90B22.

Keywords and phrases: Service systems, multi type customers, multi type skill based servers, matching of infinite sequences, product form solution, first come first served policy, assign longest idle server policy, complete resource pooling, local steady state, overloaded queues, abandonment.

be calculated by explicit formulae. Research on queueing networks would have gone nowhere if it weren't for Jackson's discovery that Jackson networks have a stationary distribution given by $\prod(1-\rho_i)\rho_i^{n_i}$. From that time onwards, product form results have been keenly sought after, as they seem the best way to get explicit solutions and useful insight for more general models.

In recent years it has become very important to investigate service systems which serve customers of several types, and which employ servers of various skills. Such service systems are referred to in current literature as *queues with skill based parallel service*. These systems have a bipartite compatibility graph to indicate which types of servers can serve which types of customers. Applications include such varied fields as call centers, outsourcing, manufacturing processes, cloud computing and health systems. As one might expect, studying these systems immediately gives rise to very complex models. Most queueing papers in this area deal with very limited simple compatibility graphs, such as "V" systems, "Λ" systems, "N" systems, and results are mostly obtained as approximations using various types of scaling. The behavior of queues with skill based parallel service is highly dependent on the type of policy which is used to assign servers to customers, and much effort has gone into trying to define objectives and identify optimal or near optimal policies, see e.g. [12, 15, 27, 31, 36].

In this paper we present the first general product form results for an important class of useful skill based parallel service queues. Our focus is the FCFS-ALIS — first come first served, assign longest idle server — policy. This means that, whenever a server becomes available, he will pick the longest waiting customer in the system which he can serve, and whenever a customer arrives to find several idle servers, he will be assigned to the longest idle compatible server. This policy has several attractive features: First and foremost it is fair to both customers and servers — in many systems, e.g. organ donations, public housing assignment, FCFS is dictated by law. ALIS is the best way to equalize the efforts of the servers, and thus it encourages diligent service. The policy is also very natural and easy to implement, and it requires minimal information about the parameters of the system and its current state. As a result, it is useful also when the parameters are time varying, e.g. for systems with periodic fluctuations of the workload.

Our results in this paper complement and extend earlier results, which we have obtained in two recent papers. In [35] we have derived a product form solution for the same model under a different policy, in which service is FCFS, but arriving customers that find several idle servers are assigned to a compatible server randomly. However, those results did not give rise

to a practical policy, since product form only held for a singular choice of the assignment distribution. In [4] we considered FCFS matching of an infinite sequence of customers and an infinite sequence of servers, giving rise to a model that is much simpler than a queue, but provides a great deal of insight. Both models are closely related to the model of this paper. Two motivating precursors to these papers were the paper of Talreja and Whitt [34], which introduced FCFS skill based routing in an overloaded system with abandonments, and a paper of Caldentey, Kaplan and Weiss [18], which introduced the infinite matching model. Product form results for skill based routing in a loss system were obtained in [3, 4].

Our model in this paper is described in the standard customer server language used for queueing models. However, it should find as much use also to describe the flow of jobs in a manufacturing system with non-homogenous machines, skill based routing of calls to agents in a call center, wireless messages to ad hoc nodes, evaluation threads to computing processors, and so on. See in particular [6, 7, 9, 11, 24, 25, 26, 30, 32, 37].

1.2. *System description.* We study the following system: There are J parallel servers, labeled $m_1, \ldots, m_J$, and there are several types of customers, labeled $a, b, c, \ldots$. We denote the set of servers by $\mathcal{S}$ and the set of customer types by $\mathcal{C}$. Service is skill based, in the sense that each server m_j has a non-empty subset of customer types which he can serve, given by $\mathcal{C}(m_j)$, the union of which is $\mathcal{C}$, and customers of type c have a non-empty set of servers which can serve them, given by $\mathcal{S}(c)$. This is described by a bipartite graph between the servers and the customer types, with directed arc (c, m_j) if c can be served by m_j. We assume that this graph is connected. The motivating advantage of such systems is that while they provide custom tailored service to the various types of customers, the overlap of server skills allows for resource pooling and reduced congestion.

We assume that arrivals are Poisson and service is exponential. Customers of type c arrive at the system in independent Poisson streams with rates λ_c, $c \in \mathcal{C}$. Service times of server m_j are independent and exponentially distributed with rate μ_{m_j}, $j \in \mathcal{S}$. Note that service durations of customers depend on the server providing the service, and *not* on the customer type. In the literature, this case is also referred to as *pool dependent* service rates, see e.g. [16].

As stated before, the service policy is FCFS-ALIS. Under this combined FCFS-ALIS policy we adopt the following, somewhat non-standard, Markovian system description. Typically, parallel service systems are described by several queue of waiting customers and the set of parallel servers which are

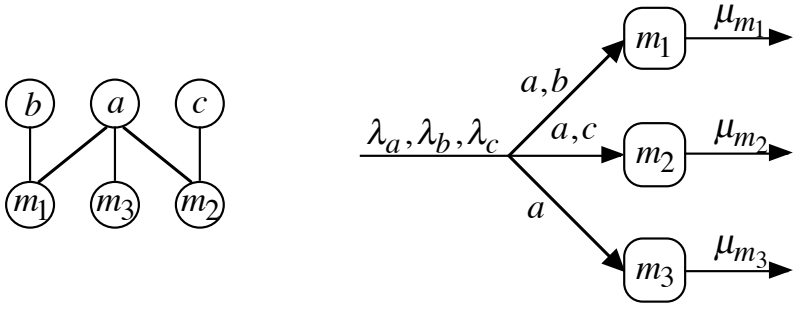

FIG 1. *A system with three types of customers and three parallel servers.*

either busy or idle. Here, we imagine all the customers to be waiting in a single queue according to their order of arrival, each of the customers being distinguished by his type. The servers move through the queue to provide service to successive customers, and customers leave their place only on completion of service. A full description of the state of the system consists of three parts: first the list of all the customers in the system in their order of arrivals, including customers which are being served (since they stay in line), second the location in this list of each of the busy servers, where we imagine that the servers are situated in the queue at the position of the customer that they are serving, and third the list of all idle servers, ordered by increasing idle time and waiting at the head of the line for new customers to arrive.

The following illustration of the system, and description of its state are similar to those in [35]. We use the same example here to point out how the current system differs from [35]. Consider a system with three servers and three customer types, as shown in Fig. 1: There are three customer types a, b, c and three servers with $\mathcal{C}(m_1) = \{a, b\}$, $\mathcal{C}(m_2) = \{a, c\}$, $\mathcal{C}(m_3) = \{a\}$, the left side of the figure is the compatibility graph, the right side shows the routing of arrivals to the servers.

Three possible states of this system are depicted in Fig. 2, which employs the following way to describe the state of the system: The customers are denoted by circles and the servers by rectangles. Customers in service have a rectangle drawn around them with the identity of the server inside. Idle servers are denoted by rectangles with a $*$ instead of a circle. The customers are ordered from left to right by increasing time of arrival, followed on the right by the idle servers, ordered from left to right by increasing idle time. One can visualize the dynamics of the system with customers arriving from the right, scanning the idle servers to pick the rightmost (longest idle) one

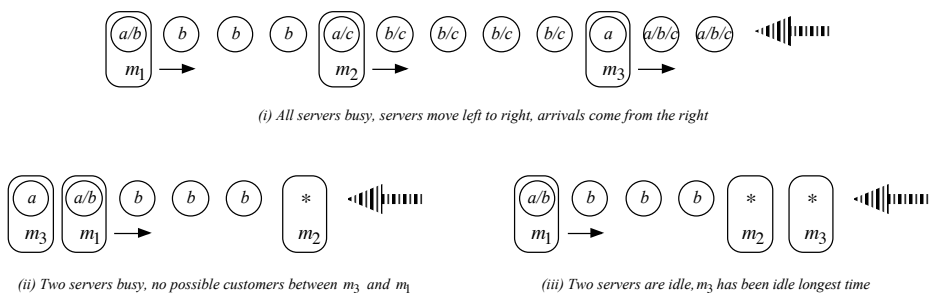

FIG 2. *A possible state for the system in Fig. 1.*

that is compatible, and then joining the queue at the rightmost position with this server, or without a server if none is compatible. Concurrently when a service is completed the customer is removed from the queue, and starting from its current position, the server that completed service moves to the right looking for the earliest waiting customer which is compatible, and starts serving it, or if no compatible customer is found the server joins the idle servers in the leftmost position. Note that waiting customers to the left of a server in this picture must be incompatible with that server, because of the FCFS rule. The difference with the system in [35] is that in the current system we need to record the list of idle servers, ordered by increasing idle time, whereas in [35] we only need to know which servers are idle. For product form systems, however, it is not uncommon that apparently small changes immediately render the system intractable. Surprisingly, in this case it appears that product form is preserved.

In Fig. 2 part (i) there are 12 customers in the system, and all the servers are busy. Server m_1 is serving the first customer in line, which must therefore be either of type a or of type b. Following the queue to the right, server m_2 is serving the first customer in the line which he can serve, which is the 5th customer in the line, and must therefore be either type a or type c. Server m_3 is serving the first customer in the line (apart from customers 1 and 5) which he can serve, and must be of type a. There are 3 customers waiting between servers m_1 and m_2. These customers cannot be served by either server m_2 or by server m_3, so they must be type b customers. There are 4 customers waiting between m_2 and m_3, those cannot be served by server m_3, so they must be of types b or c. Finally, there are 2 customers at the tail of the queue, behind server m_3, which may be of types a, b, or c. The situation in part (ii) of Fig. 2 is that servers m_3, m_1 are busy, with server m_3 serving the earlier customer, while server m_2 is idle. There can be no customers

waiting after server m_3 and before the next server, because servers m_1, m_2 combined can serve all types, and would have picked up the next customer after server m_3. The situation in part (*iii*) of Fig. 2 is that only server m_1 is busy, with 3 type b customers waiting and servers m_2, m_3 are idle, with server m_3 the one that became idle first. If the next arriving customer is of type a he will go to server m_3 (and not to m_2, because of ALIS). If he is of type c, he will go to server m_2. If he is of type b, the two servers will remain idle and the customer will join the queue in the last position.

We will actually aggregate some of the states in this detailed description, to simplify the model while retaining the Markovian property.

1.3. *Overview of results.* In the first part of this paper (Section 2) we examine our system when it is stable, i.e. positive recurrent. We define a Markov chain which describes the system, derive its transition rates, set up partial balance equations, and solve these equations to obtain conditions for ergodicity and an explicit product-form expression for the stationary distribution of the system. The derivation of the product form solution is similar to that in [35], but the difference in the state description, by including the list of idle servers ordered by increasing idle time, makes it necessary to present the full proof again. We feel that this is also important for the readability of the paper. We then compare, in Section 2.3, the behavior of the system under ALIS with the behavior under the random assignment policy of [35]. Surprisingly, it turns out that they both possess the *same stationary distribution*. The distribution of the waiting time in the queue is briefly mentioned in Section 2.4, as it is the same as the one derived in [35].

In the second part of the paper (Section 3) we analyze the system under overload conditions. The behavior under FCFS for an overloaded system with several types of customers is trivial for systems with uniform servers – all the servers work all the time and customers are served one after the other, while the queue of customers continues to grow at a linear rate. For systems with skill based parallel servers, the behavior under overload is much more intricate: Different types of servers may move through the queue at different rates, and different types of customers will be served at different rates, even though the policy is FCFS. We are not aware of any previous papers which address this question. The main tool in the analysis of the overloaded system is the derivation of the fluid scale dynamics under FCFS-ALIS. The novel derivations of these dynamics are of interest in their own rights, and may prove to be useful in future research. We also require a lemma on local steady state of a Markovian system, which is given in [2]. To analyze the behavior of overloaded systems under FCFS-ALIS, we introduce the concept of *complete*

resource pooling of the system: under complete resource pooling, the system is stable under FCFS-ALIS whenever the total arrival rate $\lambda = \sum_{c\in C} \lambda_c$ is less than the total service rate $\mu = \sum_{j=1}^{J} \mu_{m_j}$. We derive an explicit condition for complete resource pooling. Under complete resource pooling, if the system is overloaded, i.e. $\lambda > \mu$, the system is transient with the number of customers in the system growing without bound as $t \to \infty$. However, we show that at the same time, as $t \to \infty$, the servers stay together, and only the queue behind the last server grows without bound, while the state of the servers and the customers waiting between them converges to a stationary distribution. Remarkably, this stationary distribution is the same as that obtained for FCFS infinite matching in [4]. We also consider the case when complete resource pooling does not hold. In that case we show, by considering a network maximal flow problem, that the system under FCFS-ALIS policy will, when overloaded, *decompose in a unique way* to a partition of the servers, where each subset of servers stay together and serve a subset of the customers, while queues between these subsets of servers will grow without bound. Again, the state of each subset of servers, and the customers waiting between them, will converge to a stationary distribution given by [4].

Finally, in Section 4 we outline heuristic conjectures on many server behavior of the FCFS-ALIS parallel service system, under several generalizations, including abandonments, general arrival streams and customer-server type dependent service times. These conjectures may stimulate further research. At the time of writing we do not yet have rigorous proofs of this behavior under many server scaling.

Note: We chose to put most of the proofs in an appendix, for various reasons: The proofs for Section 2 mainly follow results in previous papers [35, 4], which the reader may be familiar with. The proofs for Section 3.2 employ techniques of fluid limits similar to [17, 21], while the proofs for Section 3.4 employ techniques of flows in networks [23], and the reader may not wish to pursue those. We hope this may improve the readability of the paper.

1.4. *Some comments on resource pooling for skill based parallel service.* As stated before, the performance of systems with skill based parallel service is highly dependent on the service policy, and on the traffic intensity. When resource pooling occurs, the performance of the system is essentially similar to what could be achieved with a homogeneous pool of servers of similar capacity. In the case of systems which have excess capacity, referred to as QD (Quality Driven) regime, pooling usually reduces waiting times: lack of pooling results in the partition of the servers into several pools, with separate

queues at each, so that service is not uniform, and waiting times are much higher. In critically loaded systems, for example in the QED (Quality and Efficiency Driven) regime, lack of pooling will result in some parts of the system being stable and others being unstable. For overloaded systems, which are stabilized by abandonments (in the ED, Efficiency driven regime), resource pooling will as a rule lessen the abandonments. These topics are discussed in [12, 13, 14, 15, 16, 28], where various policies, seeking optimal performance are discussed. In these papers results are obtained mainly for heavy traffic, and the analysis is based on diffusion approximations. In the current paper, in contrast to all the above papers, we focus on the FCFS-ALIS policy. We obtain resource pooling in the stable case (Section 2), and also in the overloaded case, when our condition for complete resource pooling holds (see Section 3.1 for the definition, and Section 3.3 for the pooling result). The important thing to note is that our condition for complete resource pooling is a minimal condition. It is a necessary condition under any of the policies in the above references. Furthermore, we believe it is a necessary condition for resource pooling under any policy, for any reasonable definition of resource pooling. As we show in the current paper, this minimal necessary condition is also sufficient for resource pooling when the policy is FCFS-ALIS.

2. The stable system. In this section we define a continuous time Markov chain $X(t)$ to describe the dynamics of our queueing system. We derive conditions for ergodicity for this Markov chain, and we obtain its stationary distribution, which is of product form (Sections 2.1, 2.2). We then compare it with the stationary distribution of the system under the policy of [35] (Section 2.3), and mention the derivation of the waiting time distribution (Section 2.4).

To define the Markov chain we aggregate some of the states in the detailed description, to simplify the model while retaining the Markovian property. We retain the identity and location of the busy servers, but we do not specify the type of customer that each of them is serving. Also we only record the number of customers between the busy servers, and do not specify the string of customer types. Finally, we retain the order of the idle servers. Thus the state of the system in Fig. 2(i) is denoted $(m_1, 3, m_2, 4, m_3, 2)$, the state in Fig. 2(ii) is $(m_3, 0, m_1, 3, m_2)$, and the state in Fig. 2(iii) is $(m_1, 3, m_2, m_3)$. Note that each busy server is followed by a number which counts how many (could be zero) customers are waiting behind him, while idle servers are not followed by a number. Also note that the location of the servers implicitly contains information on the type of customers waiting in line between them: they have been skipped and thus cannot be served by the servers further

down the line. For example, the three customers behind m_1, in the state $(m_1, 3, m_2, 4, m_3, 2)$ in Fig. 2(i), can only be handled by m_1 and not by m_2 or m_3. This means that they have to be of type b. Similarly, each of the four customers behind m_2 are either of type b or type c. We can tell that they are of type b with probability $\frac{\lambda_b}{\lambda_b+\lambda_c}$ and of type c with probability $\frac{\lambda_c}{\lambda_b+\lambda_c}$. Customers behind m_3 have not been scanned yet and thus may be of any type.

We introduce the following notation:

$$
\begin{aligned}
M &:= \text{an arbitrary server } M \text{ from the set of servers } \mathcal{S} = \{m_1, \ldots, m_J\}. \text{ The capitalised } M \text{ points to one of the servers } m_j. \text{ Note that the names (or labels) of the servers } m_j \text{ are not capitalised.}\\
\lambda_{\mathcal{X}} &:= \sum_{c\in\mathcal{X}} \lambda_c, \text{ where } \mathcal{X} \subseteq \mathcal{C}\\
\mu_{\mathcal{Y}} &:= \sum_{m_j\in\mathcal{Y}} \mu_{m_j}, \text{ where } \mathcal{Y} \subseteq \mathcal{S}\\
\mathcal{C}(\mathcal{Y}) &:= \text{total set of customer types that can be handled by the servers in } \mathcal{Y} \subseteq \mathcal{S}, \text{ which is equal to } \textstyle\bigcup_{m_j\in\mathcal{Y}} \mathcal{C}(m_j).\\
\mathcal{U}(\mathcal{Y}) &:= \text{set of customer types unique to the servers in } \mathcal{Y} \subseteq \mathcal{S}, \text{ thus the set of customer types that cannot be served by servers outside } \mathcal{Y}. \text{ We have } \mathcal{U}(\mathcal{Y}) = \overline{\mathcal{C}(\overline{\mathcal{Y}})}, \text{ where } \overline{\mathcal{A}} \text{ denotes the complement of set } \mathcal{A}.\\
\mathcal{S}(\mathcal{X}) &:= \text{total set of servers that can serve customer types in } \mathcal{X} \subseteq \mathcal{C}, \text{ which is equal to } \textstyle\bigcup_{c\in\mathcal{X}} \mathcal{S}(c).
\end{aligned}
$$

2.1. *Definition of the system state and the Markov chain.* We define the Markov chain $X(t)$ as the process which records the state of the system at time t, where the state in general is:

$\boldsymbol{(M_1, n_1, \ldots, M_i, n_i, M_{i+1}, \ldots, M_J)}$: State in which there are i busy servers and $J - i$ idle servers with corresponding numbers of customers waiting between the busy servers. Here $M_1, \ldots, M_J$ is a permutation of $m_1, \ldots, m_J$. Servers $M_1, \ldots, M_i$ are serving customers of increasing arrival times, with n_j customers, ordered by arrival times, waiting between servers M_j and M_{j+1}, and servers $M_{i+1}, \ldots, M_J$ are idle, with increasing idle times. There is a total of $i + \sum_{j=1}^{i} n_j$ customers in the system, i of which are being served.

The state space is denoted by $\mathfrak{S}$ and to simplify the notation we use $\mathfrak{s}$ to denote an arbitrary state $\mathfrak{s} = (M_1, n_1, \ldots, M_i, n_i, M_{i+1}, \ldots, M_J) \in \mathfrak{S}$. Fig. 3 shows a system in state $\mathfrak{s}$.

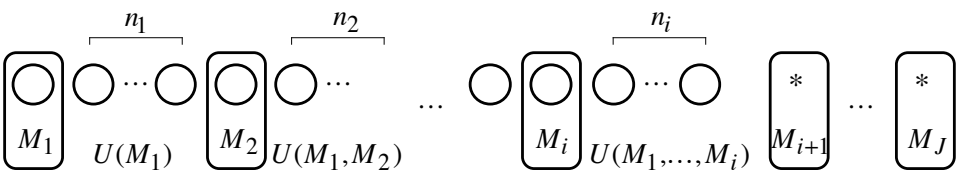

FIG 3. *General system in state* $\mathfrak{s} = (M_1, n_1, \ldots, M_i, n_i, M_{i+1}, \ldots, M_J)$.

There are a few things that are important to note about this state description:

First, the waiting customers between servers M_j and M_{j+1} can only be handled by some of the servers $M_1, \ldots, M_j$ and not by any of the servers $M_{j+1}, \ldots, M_J$. This is due to the FCFS serving order. Thus waiting customers between servers M_j and M_{j+1} can only be of type $c \in \mathcal{U}(\{M_1, \ldots, M_j\})$. The n_i waiting customers in the back of the queue cannot be handled by any of the idle servers and have to be of type $c \in \mathcal{U}(\{M_1, \ldots, M_i\})$.

Second, since each part of the queue between two servers contains customers from different subsets of customer types, it is necessary to keep these parts separated in the state description. It is not possible to aggregate the state description any further without losing the Markov property. Because the state description does not specify the types of the n_j customers between M_j and M_{j+1} we cannot tell the type of each of them, but we do know that he is of type c where $c \in \mathcal{U}(\{M_1, \ldots, M_j\})$ with probability $\frac{\lambda_c}{\lambda_{\mathcal{U}(\{M_1,\ldots,M_j\})}}$, independent of all the others. This is because the queue between servers M_j and M_{j+1} contains all the customers of types $\mathcal{U}(\{M_1, \ldots, M_j\})$ that arrived between the two customers served by M_j, M_{j+1}, in their original order of arrival.

Third, it is possible that the set of customer types $\mathcal{U}(\{M_1, \ldots, M_j\})$ is empty for a certain set of servers $\{M_1, \ldots, M_j\}$. In this case there are no customers which cannot be handled by any of the servers $M_{j+1}, \ldots, M_J$. Thus there can be no waiting customers between M_j and M_{j+1}, and therefore n_j can only be equal to zero. In that case one also has that $n_1, n_2, \ldots, n_{j-1} = 0$.

Fourth, it is important to note that in this state description we lose customer type information about the customers that are in service, since we only denote the server that is handling the customer and not the type of the customer. This aggregation preserves the Markov property since all types are processed by server m_j at rate μ_{m_j}. We conjecture that specifying the customer types in service will destroy the possibility of a product form solution. This was also indicated in Proposition 8, Section 2, of [35], which illustrates the subtlety of the definition of the system state, being crucial to the existence of product forms.

2.2. *Dynamics of the Markov chain, ergodicity, and product form stationary distribution.* The dynamics of the Markov chain $X(t)$ are as follows: When the system is in state $\mathfrak{s}$, the following transitions are possible:

(i) When a customer of type c arrives, he will activate server M_J if $c \in \mathcal{C}(M_J)$, and he will activate server M_j for $i+1 \leq j < J$ if $c \in \mathcal{C}(M_j)\backslash\mathcal{C}(\{M_{j+1}, \ldots, M_J\})$. The customer will move to the end of the queue with the activated server which will then become M_{i+1} with $n_{i+1} = 0$. If $c \in \mathcal{U}(\{M_1, \ldots, M_i\})$, then the customer will move to the end of the queue and wait, the idle servers will remain unchanged, and n_i will increase by one.

(ii) When server M_j completes service, he will scan the customers in queue from left to right, starting with the n_j customers queued behind it, and continuing with the queues behind servers $M_{j+1}, \ldots, M_i$. It will skip a customer in the queue behind M_k, $k \geq j$, if the customer type is $c \in \mathcal{U}(\{M_1, \ldots, M_k\})\backslash\mathcal{C}(M_j)$, and will pick up the first customer of type $c \in \mathcal{U}(\{M_1, \ldots, M_k\}) \cap \mathcal{C}(M_j)$. If he finds no customer to serve, he will join the idle servers, in the leftmost position.

It is readily verified that the process $X(t)$ on $\mathfrak{S}$ is a continuous time Markov chain. Furthermore it is irreducible (cf. Section 3.1 in [35]). The following theorem states the stationary distribution of $X(t)$ which is of product form.

THEOREM 2.1. *The solution to the equilibrium equations for the process $X(t)$ is given by*

$$\pi_X(\mathfrak{s}) = B \prod_{j=1}^{i} \frac{\lambda_{\mathcal{U}(\{M_1,\ldots,M_j\})}{}^{n_j}}{\mu_{\{M_1,\ldots,M_j\}}{}^{n_j+1}} \prod_{j=i+1}^{J} \lambda_{\mathcal{C}(\{M_j\ldots,M_J\})}{}^{-1}. \tag{2.1}$$

The Markov chain is ergodic if and only if the two equivalent conditions hold:

$$\begin{aligned} &\lambda_C < \mu_{\mathcal{S}(C)}, && \text{for all } C \subseteq \mathcal{C}, \\ &\lambda_{\mathcal{U}(S)} < \mu_S, && \text{for all } S \subseteq \mathcal{S}. \end{aligned} \tag{2.2}$$

After normalisation this solution becomes the stationary distribution, with normalization constant:

(2.3)

$$B = \left(\sum_{M_1,\ldots,M_J \in \mathcal{P}} \sum_{i=0}^{J} \prod_{j=1}^{i} \left(\mu_{\{M_1,\ldots,M_j\}} - \lambda_{\mathcal{U}(\{M_1,\ldots,M_j\})}\right)^{-1} \prod_{j=i+1}^{J} \left(\lambda_{\mathcal{C}(\{M_j,\ldots,M_J\})}\right)^{-1} \right)^{-1},$$

where $\mathcal{P}$ is the set of all the permutations of $\mathcal{S}$ (by convention empty products are 1).

Note that if $\mathcal{U}(\{M_1,\dots,M_j\}) = \emptyset$, then $\lambda_{\mathcal{U}(\{M_1,\dots,M_j\})} = 0$ and hence, $\pi(\mathfrak{s}) = 0$ for all $\mathfrak{s} = (M_1, n_1, \dots, M_j, n_j, \dots, M_J)$ with $n_j > 0$, as it should. Setting up the equilibrium equations and verifying Theorem 2.1 is similar to [35]. The proof is given in Appendix A.1

2.3. *Comparison with random assignment model.* In a recent paper [35], the same queueing system was analyzed, but under a different policy. Surprisingly, we find that the stationary distribution under both policies is the same.

The policy used in [35] is as follows: When a server becomes available, he will take the longest waiting compatible customer, so FCFS. However, when a customer arrives and finds several compatible idle servers, he will not go to the longest idle server, but will instead choose one of them randomly, using a random assignment probability. Thus, if a customer of type c arrives to find servers $M_1,\dots,M_i$ busy, he will go to server M_j which is idle and for which $c \in \mathcal{C}(M_j)$ with probability $P(c, M_j|\{M_1,\dots,M_i\})$. It is shown in [35] that these probabilities can always be chosen in such a way that the queueing system will have a product form solution, provided it is stable (which holds if and only if (2.2) is satisfied). While these special assignment probabilities may not be unique, they will determine unique values of $\lambda_{M_j}(\{M_1,\dots,M_i\})$, the rate at which idle server M_j will be activated when servers $\{M_1,\dots,M_i\}$ are busy. These unique activation rates can be calculated recursively from:

$$\lambda_{\mathcal{C}(\mathcal{S}\setminus\{M_1,\dots,M_i\})} = \sum_{M\notin\{M_1,\dots,M_i\}} \lambda_M(\{M_1,\dots,M_i\}), \tag{2.4}$$

$$\frac{\lambda_{M_j}(\{M_1,\dots,M_i\})}{\lambda_{\mathcal{C}(\mathcal{S}\setminus\{M_1,\dots,M_i\})}} = \Bigl(1 + \sum_{M_k\notin\{M_1,\dots,M_i,M_j\}} \frac{\lambda_{M_k}(\{M_1,\dots,M_i,M_j\})}{\lambda_{M_j}(\{M_1,\dots,M_i,M_k\})}\Bigr)^{-1},$$
$$i \le J-2,\ M_j \notin \{M_1,\dots,M_i\}.$$

These activation rates have the special property that:

$$\prod_{j=1}^{i} \lambda_{M_j}(\{M_1,\dots,M_{j-1}\}) = \prod_{j=1}^{i} \lambda_{\overline{M}_j}(\{\overline{M}_1,\dots,\overline{M}_{j-1}\}) \tag{2.5}$$

for every permutation $\overline{M}_1,\dots,\overline{M}_i$ of $M_1,\dots,M_i$. In [35], this property is referred to as the *assignment condition*.

The dynamics of the system under the random assignment policy are described by a Markov chain $Y(t)$, whose states list the busy servers in order,

and the number of customers queueing between them, given by $M_1, n_1, \ldots, n_{i-1}, M_i, n_i$, where the idle servers are $\mathcal{S}\backslash\{M_1, \ldots, M_i\}$ (and now there is no need to record the list of idle servers, in order of increasing idle time). The stationary distribution, when the assignment probabilities and the activation rates are as described above, are given by (see [35], Theorem 2)

$$\begin{aligned}(2.6)\qquad &\pi_Y(M_1, n_1, \ldots, n_{i-1}, M_i, n_i)\\ &\quad = \pi_Y(0) \prod_{j=1}^{i} \frac{\lambda_{\mathcal{U}(\{M_1,\ldots,M_j\})}^{n_j}}{\mu_{\{M_1,\ldots,M_j\}}^{n_j+1}} \prod_{j=1}^{i} \lambda_{M_j}(\{M_1, \ldots, M_{j-1}\}).\end{aligned}$$

THEOREM 2.2. *The system under random assignment, satisfying the assignment condition of [35], and the system under FCFS-ALIS share the same stationary behavior in the sense that:*

$$\begin{aligned}&\pi_Y(M_1, n_1, \ldots, n_{i-1}, M_i, n_i)\\ &\quad = \sum_{\overline{M}_{i+1},\ldots,\overline{M}_J \in \mathcal{P}(\{M_{i+1},\ldots,M_J\})} \pi_X(M_1, n_1, \ldots, n_{i-1}, M_i, n_i, \overline{M}_{i+1}, \ldots, \overline{M}_J)\end{aligned}$$

where $\mathcal{P}(\{M_{i+1}, \ldots, M_J\})$ *denotes the set of all the permutations of* $M_{i+1}, \ldots, M_J$.

The proof of Theorem 2.2 is given in Appendix A.2, and it is based on the proof of a similar result for a skill based service Erlang loss system [3, 5].

2.4. *Waiting time distribution.* The waiting time distribution for a customer of type c that arrives at the system is derived for the random assignment policy in [35], by using the distributional form of Little's law. By Theorem 2.2 in the previous section, a customer of type c that arrives at the system when the system is governed by the FCFS-ALIS policy, will, under steady state, see exactly the same distribution of states as under the random assignment policy of [35]. As a consequence, the waiting time distribution for a customer of type c will be exactly the one derived in [35], Theorem 3.

3. The overloaded system. In this section we consider the system as before, with total arrival rate $\lambda = \sum_{c\in C} \lambda_c$ and total service rate $\mu = \sum_{j=1}^{J} \mu_{m_j}$. We introduce the following notations: $\rho = \lambda/\mu$ is the total traffic intensity, $\alpha_c = \lambda_c/\lambda$ is the fraction of arrivals of type c customers, and $\beta_{m_j} = \mu_{m_j}/\mu$ is the fraction of service capacity of server m_j. Also, for subsets, $\alpha_{\mathcal{X}} = \lambda_{\mathcal{X}}/\lambda$, $\mathcal{X} \subseteq \mathcal{C}$, and $\beta_{\mathcal{Y}} = \mu_{\mathcal{Y}}/\mu$, $\mathcal{Y} \subseteq \mathcal{S}$.

It is convenient in this section to rewrite the stationary probabilities of $X(t)$ as:

$$\pi_X(\mathfrak{s}) = \tilde{B}(\rho) \prod_{j=1}^{i} \frac{\left(\rho\alpha_{\mathcal{U}(\{M_1,\ldots,M_j\})}\right)^{n_j}}{\beta_{\{M_1,\ldots,M_j\}}{}^{n_j+1}} \prod_{j=i+1}^{J} \left(\rho\alpha_{\mathcal{C}(\{M_j\ldots,M_J\})}\right)^{-1} \tag{3.1}$$

with

$$\begin{aligned}\tilde{B}(\rho) &= B/\mu^J \\ &= \left(\sum_{M_1,\ldots,M_J \in \mathcal{P}(\mathcal{S})} \sum_{i=0}^{J} \left(\prod_{j=1}^{i} \left(\beta_{\{M_1,\ldots,M_j\}} - \rho\alpha_{\mathcal{U}(\{M_1,\ldots,M_j\})}\right)^{-1} \right.\right. \\ &\qquad \left.\left. \times \prod_{j=i+1}^{J} \left(\rho\alpha_{\mathcal{C}(\{M_j,\ldots,M_J\})}\right)^{-1} \right) \right)^{-1}.\end{aligned}$$

We will study what happens to $X(t)$ as the total traffic intensity increases. We will assume that α, β are fixed, μ is fixed, and the total arrival rate λ increases. Under these conditions, for $\rho > 1$ the system becomes unstable with some of the queues growing without bounds. We will discover that when some of the queues grow without bounds, the rest of the system stabilizes and has a stationary behavior, which is identical to that observed for FCFS matching of two infinite sequences (of servers and of customers) as discussed in [4]. We will distinguish a case of complete resource pooling and a case of incomplete resource pooling. For the latter we will find a unique decomposition of the system.

We state the infinite matching model of [4] here (see also [18]): There are two infinite sequences, a sequence of customers $c^1, c^2, \ldots$ chosen i.i.d. from $\mathcal{C}$ with probabilities α, and a sequence of servers (or of services) $s^1, s^2, \ldots$ chosen i.i.d. from $\mathcal{S}$ with probabilities β, and a bipartite graph of compatibilities. Servers and customers are matched on a FCFS basis. Note that this is not a queueing model - each server is used only once, when he is matched to a customer, and there are no service durations and no arrival times, only the order of the customers and of the servers is relevant. The roles of the servers and customers in this model are entirely symmetric. The policy is FCFS in the ordinal sense and not in the temporal sense: a customer that precedes another customer in the sequence of customers is matched to the server in the earliest possible position in the sequence of servers.

A discrete time markov chain, $I = (I(N), N = 0, 1, \ldots)$ is associated with the infinite matching model. $I(N)$ describes the state of the system after matching the first N servers, and it is defined as follows: After matching

the first N servers to customers, the sequence of customers has an initial segment in which all customers are matched, followed by a middle segment in which some are matched and some are not, followed by a last infinite segment where none are matched. $I(N)$ describes this middle segment. The state is of the form $(M_1, n_1, \ldots, M_{J-1}, n_{J-1}, M_J)$, where $M_1, \ldots, M_J$ is a random permutation of $\mathcal{S}$ that lists the last matched server of each type according to its order of appearance in the sequence of customers, and n_j is the number of unmatched customers between M_j and M_{j+1}. It is shown in [4] that I is an ergodic Markov chain and has a product form stationary distribution given by:

$$\pi_I(M_1, n_1, \ldots, n_{J-1}, M_J) = B_I \prod_{j=1}^{J-1} \frac{\alpha_{\mathcal{U}(\{M_1,\ldots,M_j\})}{}^{n_i}}{\beta_{\{M_1,\ldots,M_j\}}{}^{n_i+1}}, \tag{3.2}$$

with

$$B_I = \left(\sum_{M_1,\ldots,M_J \in \mathcal{P}(\mathcal{S})} \prod_{j=1}^{J-1} \left(\beta_{\{M_1,\ldots,M_j\}} - \alpha_{\mathcal{U}(\{M_1,\ldots,M_j\})} \right)^{-1} \right)^{-1}, \tag{3.3}$$

whenever B_I is positive and finite. The analogy between I and our system is clear: The last matched server of type m_j in $I(N)$ corresponds to the position of server m_j in the queue, the permutation $M_1, \ldots, M_J$ in $I(N)$ is the order of the servers in the queue, and the unmatched customers in $I(N)$ correspond to the queues between the servers. The infinite sequence of customers that follow after M_J in the infinite matching model corresponds to waiting customers and to all future arrivals in the queueing model.

The rest of this section is structured as follows: In Section 3.1 we define complete resource pooling, under which $X(t)$ is stable for all $\lambda < \mu$, and show that as $\lambda \nearrow \mu$ the stationary distribution of $X(t)$ converges to that of $I(N)$. In Section 3.2 we study the fluid approximation to our queueing system. This contains information on the dynamics of the system which cannot be gleaned from its stationary distribution. The rest of the section deals with overloaded systems: in Section 3.3 we study the limiting behavior of the overloaded system under complete resource pooling. We show that while the last queue grows to infinity, the rest of the system converges in law to the stationary distribution of $I(N)$. In particular, while the last queue grows to infinity, the remaining queues between the servers remain well behaved, and the servers stay close together. In Section 3.4 we study a network maximal flow problem that is related to the stability of our system, and derive a unique decomposition of the system, in the case of incomplete resource pooling. In

Section 3.5 we study the limiting behavior of the overloaded system under incomplete resource pooling.

3.1. *Complete resource pooling.* We say that the system satisfies *complete resource pooling* if the following three equivalent statements hold:

$$\begin{aligned} &\beta_S > \alpha_{\mathcal{U}(S)}, && \text{for } S \subset \mathcal{S}, \quad S \neq \emptyset, \mathcal{S}, \\ &\beta_S < \alpha_{\mathcal{C}(S)}, && \text{for } S \subset \mathcal{S}, \quad S \neq \emptyset, \mathcal{S}, \\ &\alpha_C < \beta_{\mathcal{S}(C)}, && \text{for } C \subset \mathcal{C}, \quad C \neq \emptyset, \mathcal{C}. \end{aligned} \tag{3.4}$$

As we state in the next theorem, under complete resource pooling, the process $X(t)$ will be stable for all $\rho < 1$, and as $\rho \nearrow 1$ its stationary distribution will converge to the stationary distribution of $I(N)$, the process describing FCFS matching of two infinite sequences in [4].

We will use the following notation for the marginal distributions :

$$\begin{aligned} &\pi_X(M_1, n_1, \ldots, M_j, n_j, M_{j+1}, \cdot, \ldots, M_i, \cdot, M_{i+1}, \ldots, M_J) \\ &\qquad = \sum_{n_{j+1}, \ldots, n_i = 0}^{\infty} \pi_X(M_1, n_1, \ldots, , M_i, n_i, M_{i+1}, \ldots, M_J). \end{aligned}$$

THEOREM 3.1. *Consider the system with fixed μ, α, β, and let λ vary. Assume that complete resource pooling holds.*

(i) The process $X(t)$ is ergodic for all $\lambda < \mu$.

(ii) For states when some servers are idle,

$$\lim_{\rho \nearrow 1} \pi_X(M_1, n_1, \ldots, , M_i, n_i, M_{i+1}, \ldots, M_J) = 0, \text{ for } i < J.$$

(iii) For states when all servers are busy,

$$\begin{aligned} &\lim_{\rho \nearrow 1} \pi_X(M_1, n_1, \ldots, M_{J-1}, n_{J-1}, M_J, \cdot) \\ &\qquad = B_I \prod_{j=1}^{J-1} \frac{\alpha_{\mathcal{U}(\{M_1, \ldots, M_j\})}{}^{n_i}}{\beta_{\{M_1, \ldots, M_j\}}{}^{n_i+1}} \\ &\qquad = \pi_I(M_1, n_1, \ldots, M_{J-1}, n_{J-1}, M_J), \end{aligned}$$

with $\lim_{\rho \nearrow 1} \tilde{B}(\rho)(1-\rho)^{-1} = B_I$.

(iv) The same results with the same limiting values hold also for the stationary distribution of the discrete-time Markov chain of jumps.

The proof of Theorem 3.1 is given in Appendix B.1.

3.2. *Fluid limits of the FCFS-ALIS queueing system with multitype customers and servers.* The stationary distribution of a Markovian system provides complete information about long term performance measures, and it is therefore very useful. However, it does not provide any information about the dynamics of the system. The dynamics of a Markovian system are of interest when the system is stable, as they provide information about its short term behavior. Furthermore, for unstable systems they provides information on the transient behavior of the system. Exact analysis of the dynamics is almost always intractable, however the dynamics can be approximated by studying fluid or diffusion approximations to the system. In this section we study the fluid approximation of our system. For an introduction to the study of fluid limits, see [17, 19, 20, 21].

We introduce a different notation for our Markov process $X(t)$. We let $X(t) = (S(t), Q_1(t), \ldots, Q_J(t))$. Here $S(t) = (M_1(t), \ldots, M_J(t))$ is a permutation of $\mathcal{S}$. The total number of customers in the system is $|Q(t)| = \sum Q_j(t)$, ordered in order of arrival with Q_1 earlier than Q_2 earlier than Q_3, and so on. Of the queues at time t, the first $k(t)$ queues are non-empty, the remaining are empty. Server $M_j(t)$ is serving the first of the customers of $Q_j(t)$ (the $1 + \sum_{i<j} Q_i(t)$ customer in the queue), where $j = 1, \ldots, k(t)$, and servers $M_j(t)$, $j = k(t) + 1, \ldots, J$ are idle, ordered by increasing idle time. The customer in service at $M_j(t)$ is of course of type $c \in \mathcal{C}(M_j)$. The remaining $Q_j(t)-1$ waiting customers are all of them of types in $\mathcal{U}(\{M_1(t), \ldots, M_j(t)\})$.

We also introduce the following processes that give an alternative description of the system. Assume each customer upon arrival is given a number which counts its position in the arrival stream. We let $A(t)$ be the total number of arrivals by time t, where the last arrival was numbered $A(t)$. We also let $Y_1(t) < \cdots < Y_J(t)$ be the positions of the servers in the sequence of customers up to time t, so that $Y_i(t)$ is the sequence number of the customer currently served by the ith server, which is server $M_i(t)$. If the number of busy servers is $k(t) < J$, we will let the positions of the idle servers $M_{k(t)+1}(t), \ldots, M_J(t)$ be $Y_{k(t)+1}, \ldots, Y_J(t) = A(t) + 1, \ldots, A(t) + J - k(t)$. We denote $Y(t) = (Y_1(t), \ldots, Y_J(t))$. Note that $A(t), Y_1(t), \ldots, Y_J(t)$ are all monotone non-decreasing in t. We shall let $A(0), Y_1(0), \ldots, Y_J(0) \geq 0$ and $S(0) = (M_1(0), \ldots, M_J(0))$ be an initial state.

The fluid scaling of an arbitrary function $z(t)$ is denoted $\bar{z}^n(t) = z(nt)/n$. Let $z(t, \omega)$ be a stochastic process with paths in D^d (real vector functions in the d dimensional Euclidean space which are right continuous with left limits). Let ω denote a fixed sample path, and let r be a divergent subsequence of integers. If $\bar{z}^r(t, \omega)$ converges to a deterministic function $\bar{z}(t)$ as $r \to \infty$, then we call $\bar{z}(t)$ a fluid limit. Convergence here is in the Skorohod

J_1 topology on $\mathbb{D}^d$, but if $\bar{z}(t)$ is continuous it is equivalent to convergence uniformly on compacts (u.o.c.). To obtain fluid limits with $\bar{z}(0) \neq 0$ one needs to consider a sequence of functions $z^n(t,\omega)$ which are all defined on the same probability space, and which differ in their initial values $z^n(0,\omega)$.

To study fluid limits of our system we now define a sequence of systems, by describing the primitives in the probability space which are common to all of them and by defining the sequence of their initial states. For simplicity take $\lambda + \mu = 1$ and think of the evolution of our systems for all $n = 1, 2, \ldots$ and $t > 0$ as powered by a common Poisson process with rate 1. Each event of the process is either an arrival with probability λ or a service completion with probability μ. An arrival is an arrival of a customer of type c with probability α_c. A service completion is by server m_j with probability β_{m_j}. A customer departs at a service completion only if server m_j is not idle. To define the state of the system at time 0 we also have a multi-Bernoulli process of customer types, denoted $C = c^{-1}, c^{-2}, \ldots, c^{-k}, \ldots$, which represent customers that arrived in the past, prior to time $t = 0$, ordered by order of arrival, with type c occurring with probability α_c. We exclude from this list of previous customers the $\leq J$ customers which are in service at time 0. These stochastic primitives are common to all the systems.

Next we describe the initial states of the sequence of systems: They only differ by the location of the servers at time 0, within the sequence C: Let $q_{1,n}, \ldots, q_{J,n}$ be some deterministic non-negative integers, with $q^n = \sum_{k=1}^{J} q_{k,n}$, we then set $Y_j^n(0) = j + \sum_{k=1}^{j-1} q_{k,n}$, $A^n(0) = J + q^n$. The q^n customers waiting for service at time 0, between the positions of the servers, are of types $c^{-q^n}, \ldots, c^{-1}$, so that c^{-q^n} is the type of the earliest arrival customer, and c^{-1} is the type of the latest arrival prior to $t = 0$. To fix our initial limiting state we assume that $q_{j,n}/n$ converge to some fixed values, which determines the fluid limit values of $0 = \bar{Y}_1(0) \leq \cdots \leq \bar{Y}_{k(0)}(0) < \bar{A}(0) = \bar{Y}_{k(0)+1}(0) = \cdots = \bar{Y}_J(0)$. For simplicity we let $S^n(0) = S^0$ where S^0 is a fixed initial permutation which defines $\bar{S}(0)$. Note that the sequence $q_{n,j}$ and S^0 are fixed deterministic, while the sequence of customers that arrived prior to time 0 is random i.e. it depends on ω.

Fluid limits of A, Y, and S may be obtained when we take $\bar{A}^n(t,\omega) = A^n(nt,\omega)/n$ and $\bar{Y}^n(t,\omega) = Y^n(nt,\omega)/n$, and $\bar{S}^n(t,\omega) = S^n(nt,\omega)$ and let $n \to \infty$. Note that $S^n(\cdot,\omega)$ is a function from $[0,\infty)$ to the finite set of permutations $\mathcal{P}$, so there is no division by n in the definition of $\bar{S}^n(t,\omega)$. The following complication does however arise: For all ω, n, t, we have that $\bar{Y}_1^n(t,\omega) < \cdots < \bar{Y}_J^n(t,\omega)$, and there is a unique permutation $\bar{S}_1^n(t,\omega) < \cdots < \bar{S}_J^n(t,\omega)$ associated with them. However, in the limit we may have $\bar{Y}_1(t,\omega) \leq \cdots \leq \bar{Y}_J(t,\omega)$, and so the fluid limits of $\bar{S}^n(t,\omega)$

will not be a permutation, but will be a partition of a permutation, i.e. an ordered partition. For example, if $\bar{Y}_1(t) < \bar{Y}_2(t) = \bar{Y}_3(t) = \bar{Y}_4(t) < \bar{Y}_5(t) = \bar{Y}_6(t) < \bar{Y}_7(t)$, then a possible fluid limit will be the ordered partition $\bar{S}(t) = (M_1, \{M_2, M_3, M_4\}, \{M_5, M_6\}, M_7)$. In general we will associate with a fluid limit $\bar{Y}(t)$ of $\bar{Y}^r(t,\omega)$ a fluid limit for $\bar{S}^r(t,\omega)$ that will be given by a corresponding ordered partition $\bar{S}(t) = (S_1, \ldots, S_K)$ where each S_i is a subset of servers, and $S_1, \ldots, S_K$ is a partition of $\mathcal{S}$. This is not standard in the literature on fluid limits and we shall make the definition of such a fluid limit precise in Appendix B.2.

When discussing ordered partitions, we may sometimes suppress some information and group some of the subsets of the partition, and replace $(S_1, \ldots, S_K)$ by say (S', S_k, S''), where $S' = \bigcup_{l=1}^{k-1} S_l$ and $S'' = \bigcup_{l=i+1}^{K} S_l$. We extend the definition of complete resource pooling (3.4) to subsets.

DEFINITION 3.2. Consider a partition of the servers into subsets S', S, S''. We say that S has complete resource pooling between S' and S'' (the order of S' before S'' is important here), if the subsystem which consists of servers $m_i \in S$, and the customer types $c \in \mathcal{U}(S' \cup S) \backslash \mathcal{U}(S')$, with $\tilde{\beta}_{m_i} = \beta_{m_i}/\beta_S$, $\tilde{\alpha}_c = \alpha_c / \alpha_{\mathcal{U}(S' \cup S) \backslash \mathcal{U}(S')}$ has complete resource pooling.

We now have the following theorem, which summarizes the fluid dynamics of the system:

THEOREM 3.3. *Consider a sequence of systems as above. Then* $\bar{A}^n(t,\omega)$, $\bar{Y}^n(t,\omega)$, $\bar{S}^n(t,\omega)$, $\bar{Q}^n(t,\omega)$ *converge almost surely to fluid limits* $\bar{A}(t)$, $\bar{Y}(t)$, $\bar{S}(t)$, $\bar{Q}(t)$ *as* $n \to \infty$. *The fluid limits satisfy:*

(i)

$$\bar{A}(t) = \bar{A}(0) + \lambda t \tag{3.5}$$

(ii) Let $\bar{Y}_{k-1}(t) < \bar{Y}_k(t) = \cdots = \bar{Y}_l(t) < \bar{Y}_{l+1}(t)$ *for some* $k < l$ *and for some* t, *and let* $\bar{S}(t) = (S', \{M_k, \ldots, M_l\}, S'')$ *be the corresponding partition of the servers. Then* $\bar{Y}_k, \ldots \bar{Y}_l$ *will move together at rates:*

$$\frac{d}{dt}\bar{Y}_i(t) = \mu \frac{\beta_{\{M_k, \ldots, M_l\}}}{\alpha_{\mathcal{U}(S' \cup \{M_k, \ldots, M_l\})} - \alpha_{\mathcal{U}(S')}}, \quad i = k, \ldots, l. \tag{3.6}$$

during $t < \tau < t + \Delta$ *for some* $\Delta > 0$, *if and only if* $\{M_1, \ldots, M_k\}$ *have complete resource pooling between* S' *and* S''.

(iii)

$$\begin{aligned} \bar{Q}_i(t) &= \big(\bar{Y}_{i+1}(t) - \bar{Y}_i(t)\big)\alpha_{\mathcal{U}(\{M_1, \ldots M_i\})}, \quad i = 1, \ldots, J-1, \\ \bar{Q}_J(t) &= \bar{A}(t) - \bar{Y}_J(t). \end{aligned} \tag{3.7}$$

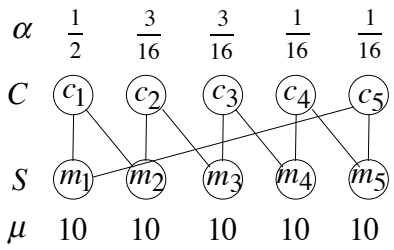

FIG 4. *A five server five customer types system.*

The proof of Theorem 3.3 requires several propositions, which derive the dynamics of fluid limits of our system. These propositions and the proof of the theorem are given in Appendix B.2

Theorem 3.3 implies that the processes $\bar{Y}_j(t)$ will move at any time with constant rates, and in coalescent subsets, so that whenever one subset of servers overtakes another, they either coalesce into a single subset, which is the union of both, or the union of the two subsets breaks up into new subsets, which move away form each other. These coalescent subsets always satisfy complete resource pooling between their predecessors and successors. This can continue until all the servers coalesce and all move together, or until they are partitioned into subsets, which drift apart at constant rates. The front subset of servers may move at the rate of λ if that subset has enough capacity to serve all of its customers, or at a rate $< \lambda$ if it is overloaded. We show in Section 3.4 that these subsets are uniquely determined, irrespective of the initial fluid state $\bar{Z}(0)$.

We now present an example that illustrates how Theorem 3.3 determines the dynamics of the fluid limits.

Example. We consider a system with 5 types of customers and 5 servers, where each node in the bipartite graph is of degree 2. The system and values of α, μ are displayed in Fig. 4. The total arrival rate is $\lambda = 60$, which is larger than the total service capacity of 50, so the system is overloaded.

Because any four servers can serve all types of customers, $\bar{Q}_1(t) = 0$ for all t. We assume that the initial partition of the servers is $(\{m_4, m_5\}, m_3, m_2, m_1)$, and the queues $\bar{Q}_2(0), \bar{Q}_3(0), \bar{Q}_4(0), \bar{Q}_5(0)$ consist of customers of types $c_4, \{c_3, c_4\}, \{c_2, c_3, c_4\}$, and {all types}, respectively. The initial values of $\bar{Y}$ are $\bar{Y}(0) = (0, 0, 250, 750, 825)$ and $\bar{A}(0) = 850$. This corresponds to $\bar{Q}(0) = (0, 15.625, 125, 32.8125, 25)$.

The system evolves in 5 time intervals. At the end of each interval some servers overtake other servers, and the new subset S of servers then stays together or splits, according to whether the servers in S have complete resource

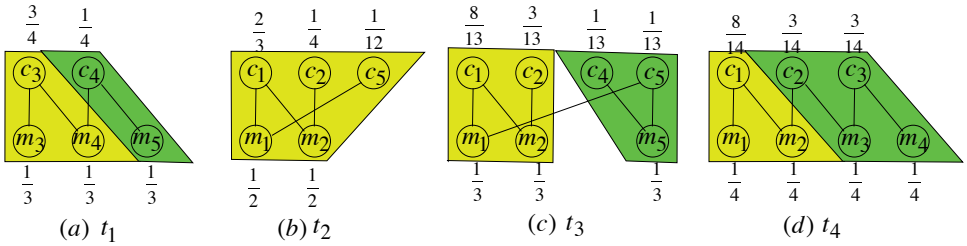

FIG 5. *Resource pooling in partitioned system.*

pooling between the servers preceding and succeeding them. The calculations to determine whether colliding servers coalesce or split up again, according to Definition 3.2, are described in Fig. 5. In that figure the fractions are the conditional probabilities for the subsystems of servers and customers. The five stages, at which queues empty out and collisions occur, are as follows:

At t_1 servers $\{m_4, m_5\}$ overtake server m_3 ($\bar{Q}_2$ empties of customers of type c_4). There is no complete resource pooling of $\{m_3, m_4, m_5\}$, and server m_5 splits off and leaves $\{m_3, m_4\}$ behind. $\bar{Q}_2$ fills up in the next interval, with customers of type c_3 (Fig. $5_{(a)}$)

At t_2 server m_2 overtakes server m_1, there is complete resource pooling of $\{m_1, m_2\}$, and they coalesce and continue together. Here $\bar{Q}_4$ empties, and stays empty in the next interval (Fig. $5_{(b)}$).

At t_3 server m_5 overtakes $\{m_1, m_2\}$, there is no resource pooling, and m_5 moves ahead (Fig. $5_{(c)}$).

At t_4 servers $\{m_3, m_4\}$ overtake servers $\{m_1, m_2\}$, there is no complete resource pooling, and the two subsets change positions (Fig. $5_{(d)}$).

At t_5 server m_5 catches up with the input, and thereafter $\bar{Q}_5$ remains empty, and server m_5 is underloaded

Following t_5 there are no more collisions, the queue of server m_5 is stable with fluid 0, and the remaining two fluid queues continue to fill up at constant rates.

The fluid dynamics of $\bar{Y}(t)$ can be calculated from Equation (3.6) of Theorem 3.3, and the fluid dynamics of $\bar{Q}(t)$ can be calculated from Equation (3.7) of Theorem 3.3. Table 1 lists the the configurations in each interval, the collision times, the fluid queue lengths at those times, and the rate of change in the queue lengths in each interval. Overlined numbers denote repeated fractions. Fig. 6 plots the fluids in the system over time.

In the long run, after t_5, servers m_1, m_2 will lag behind, serving customers of type c_1, servers m_3, m_4 will be ahead of them serving customers of types

TABLE 1
Dynamics of $\bar{Q}(t)$

$(0, t_1)$		m_4, m_5	c_4	m_3	c_3, c_4	m_2	c_2, c_3, c_4	m_1	$\mathcal{C}$
$t_0 = 0$	$\bar{Q}(0)$		15.63		125		32.81		25
	$d\bar{Q}(t)/dt$		$-16.\bar{6}$		0		$-15.\bar{5}$		$42.\bar{2}$
(t_1, t_2)		m_3, m_4	c_3	m_5	c_3, c_4	m_2	c_2, c_3, c_4	m_1	$\mathcal{C}$
$t_1 = 0.94$	$\bar{Q}(t_1)$		0		125		18.23		64.58
	$d\bar{Q}(t)/dt$		10		$-26.\bar{6}$		$-15.\bar{5}$		$42.\bar{2}$
(t_2, t_3)			m_3, m_4	c_3	m_5	c_3, c_4	m_1, m_2	$\mathcal{C}$	
$t_2 = 2.11$	$\bar{Q}(t_2)$			11.72		93.75		114.06	
	$d\bar{Q}(t)/dt$			10		$-33.\bar{3}$		$33.\bar{3}$	
(t_3, t_4)			m_3, m_4	c_3	m_1, m_2	c_1, c_2, c_3	$m_5,$	$\mathcal{C}$	
$t_3 = 4.92$	$\bar{Q}(t_3)$			39.84		0		207.81	
	$d\bar{Q}(t)/dt$			$-14.\overline{54}$		$44.\overline{54}$		-20	
(t_4, t_5)			m_1, m_2	c_1	m_3, m_4	c_1, c_2, c_3	m_5	$\mathcal{C}$	
$t_4 = 7.66$	$\bar{Q}(t_4)$			0		122.02		153.03	
	$d\bar{Q}(t)/dt$			$6.\bar{6}$		$23.\bar{3}$		-20	
(t_5, ∞)			m_1, m_2	c_1	m_3, m_4	c_1, c_2, c_3	m_5	$\mathcal{C}$	
$t_5 = 15.31$	$\bar{Q}(t_5)$			51.01		300.55		0	
	$d\bar{Q}(t)/dt$			$6.\bar{6}$		$5.8\bar{3}$		0	

c_2, c_3 and skipping customers of type c_1. The queues of customers of these types will continue to fill up. Server m_5 will be ahead, serving customers of type c_4, c_5, skipping all other types, and keeping the queue behind him stable. Server m_5 will be processing at rate 7.5 and will be idle a quarter of the time.

3.3. *The overloaded system under complete resource pooling.* We consider now the behavior of the system under complete resource pooling, when $\lambda > \mu$. Here clearly the Markov process is transient. However, what we will see is that while the queue behind the last server grows without bound, the servers and the queues between them tend to a limiting distribution, which is again that of the FCFS infinite matching model. We will use the notation and the results on the fluid dynamics from Section 3.2. We will also need the following lemma, the proof of which is given in [2].

LEMMA 1. *Let* $X(n) = (X_1(n), X_2(n))$ *be a Markov chain on countable state space with* $X_i(n) \in \mathbb{Z}^+$. *Assume the following:*

1. $\lim_{n\to\infty} X_2(n) = \infty$ *almost surely.*

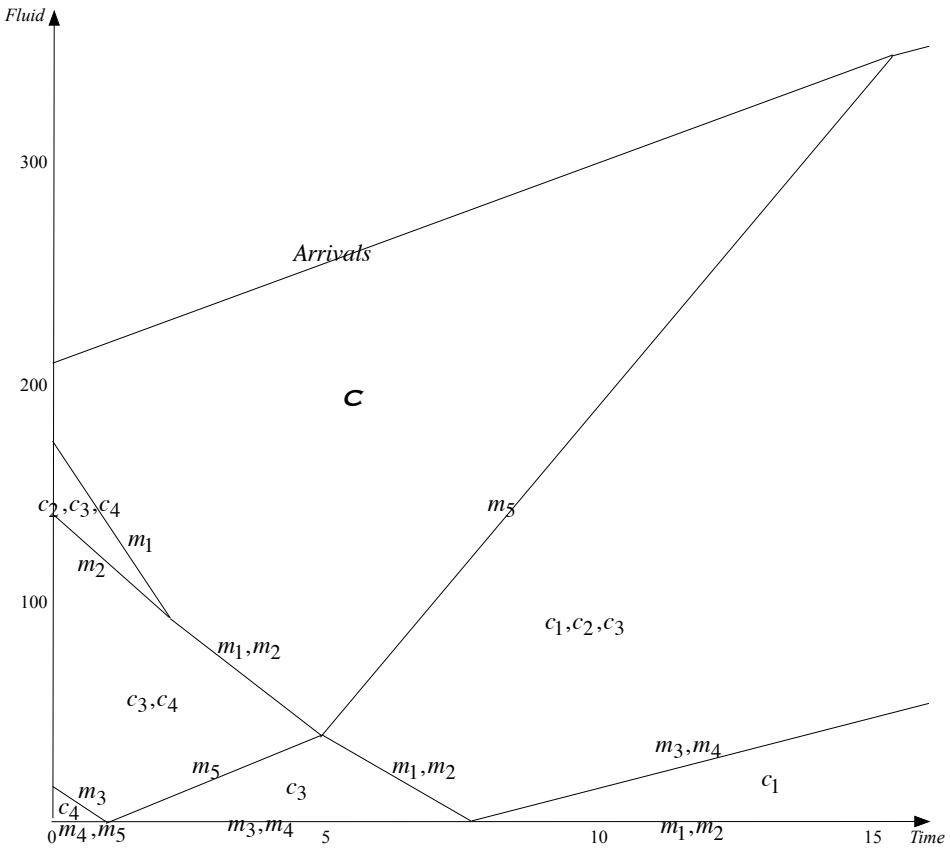

FIG 6. *Dynamics of the fluid queues.*

2. $P(X_1(n+1)=j|X_1(n)=i,\ X_2(n)=l)=P_{i,j}$, *for all values of* $l>0$, *where* $P_{i,j}$ *are transition probabilities of an ergodic Markov chain with stationary probabilities* π_j.

Then for all initial i_0, j_0*:*

$$\sup_j \left| P\big(X_1(n)=j \,|\, X_1(0)=i_0, X_2(0)=j_0\big) - \pi_j \right| \to 0, \ \text{as } n\to\infty,$$

i.e. $X_1(n)$ *converges in distribution to* π *in total variation norm.*

We can now prove the following theorem:

THEOREM 3.4. *Assume complete resource pooling and* $\lambda>\mu$.

(i) *From any fixed initial state, as* $t \to \infty$*:*

$$k(t) \to J, \quad a.s. \qquad Q_J(t)/t \to (\lambda - \mu), \quad a.s. \tag{3.8}$$

(ii) *As* $t \to \infty$, $P(S(t) = (m_1, m_2, \ldots, m_J), Q_1(t) = n_1 + 1, \ldots, Q_{J-1}(t) = n_{J-1} + 1)$ *converges to* $\pi_I(m_1, n_1, \ldots, n_{J-1}, m_J)$ *in total variation distance, where* π_I *is the stationary distribution of* I*, the Markov chain describing the FCFS infinite matching model, given in (3.2), (3.3).*

PROOF. We consider first the fluid limits for the overloaded system. By the results of the previous section, we have that $\frac{d}{dt}\bar{Y}_J(t) \le \mu$. This will hold, because by Proposition B.9, the front subset of servers S will move at a rate $\mu\frac{\beta_S}{\alpha_{\mathcal{C}(S)}}$, and by complete resource pooling, $\beta_S < \alpha_{\mathcal{C}(S)}$. Since $\bar{Q}_J(t) = \bar{A}(t) - \bar{Y}_J(t)$, we have:

$$\frac{d}{dt}\bar{Q}_J(t) = \frac{d}{dt}\bar{A}(t) - \frac{d}{dt}\bar{Y}_J(t) = \lambda - \mu\frac{\beta_S}{\alpha_{\mathcal{C}(S)}} \ge \lambda - \mu > 0.$$

Hence $\bar{Q}_J(t) \to \infty$ as $t \to \infty$, and then of course $Q_J(t)$ will diverge almost surely. In particular, this implies that all servers will be busy, so $k(t) \to J$ almost surely as $t \to \infty$.

Consider now the behavior of our Markovian system when $Q_J(t) > 0$. When $Q_J(t) > 0$, all the servers are busy, and the queues $Q_j(t), j = 1, \ldots, J-1$ have transitions which occur as a Poisson process of rate μ, irrespective of the current state. The sequence of states following each transition form a discrete time process, with Markovian transition probabilities, which are exactly those of the FCFS infinite matching model, and do not depend on the value of $Q_J(t)$. Hence, the conditions of Lemma 1 are fulfilled, and the discrete time jump process of states will converge in law to π_I. As a result, the continuous time process will also converge in law to π_I.

Because $Q_1(t), \ldots, Q_{J-1}(t)$ converge to a stationary distribution, $\bar{Q}_1, \ldots, \bar{Q}_{J-1}$ converge almost surely to 0. Hence, in the fluid limit all the servers will move together at rate μ and for any fixed initial state, $\bar{Q}_J(t) = (\lambda - \mu)t$. Hence, $Q_J(t)/t \to \lambda - \mu$ almost surely. □

3.4. *Unique decomposition under incomplete resource pooling.* We again consider the system with fixed α, β, μ and let λ increase, but we now consider the case that complete resource pooling does not hold. We show that there exists a unique decomposition of the system when it is overloaded. To do so, we associate with our system the following network (see Fig. 7): The nodes are $c \in \mathcal{C}$, $m_j \in \mathcal{S}$, a source node o, and a sink node t. The arcs are

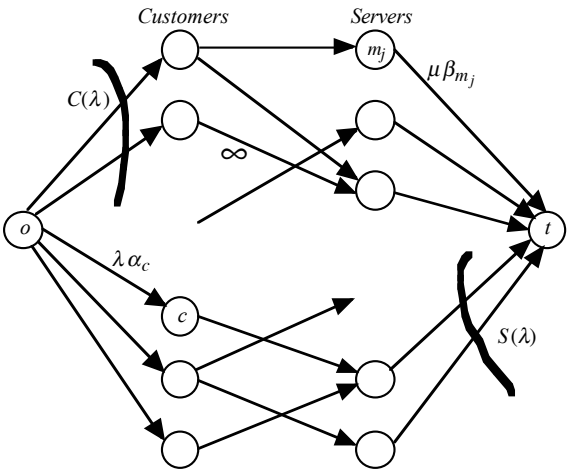

FIG 7. *s–t maximal flow problem with minimum cut through* $C(\lambda)$, $S(\lambda)$.

(o, c) of capacity λ_c for all $c \in \mathcal{C}$, (m_j, t) of capacity μ_{m_j} for all $m_j \in \mathcal{S}$, and (c, m_j) of infinite capacity for all (c, m_j) in the bipartite compatibility graph. The proofs for propositions and theorems in this section are given in Appendix B.3, throughout here and in the appendix we use the terminology of network flows, as in [10, 23].

The maximal flow in the network is related to the stability of our system through the following proposition.

PROPOSITION 3.5. *A necessary and sufficient condition for stability is that the maximal flow from* o *to* t *is* λ*, and that the cut through arcs* $(o, c), c \in \mathcal{C}$ *is the unique minimal cut.*

The following theorem describes the solution of the o to t maximum network flow problem, as a function of λ. Fig. 8 illustrates this theorem as well as the following Corollary 3.7.

THEOREM 3.6. *Consider the o–t maximum network flow problem as* λ *increases. Then:*

(i) The maximal flow $f(\lambda)$ *is a continuous piecewise linear non-decreasing concave function of* λ*, with breakpoints* $0 = \lambda^{(0)} < \lambda^{(1)} < \cdots < \lambda^{(L)} < \lambda^{(L+1)} = \infty$*, which has slope* 1 *for* $0 < \lambda < \lambda^{(1)}$*, and is constant and equal to* μ *for* $\lambda > \lambda^{(L)}$.

(ii) For each interval $(\lambda^{(i-1)}, \lambda^{(i)})$ *there exist a set of customer types* $C(\lambda^{(i-1)}, \lambda^{(i)})$ *and a set of servers* $S(\lambda^{(i-1)}, \lambda^{(i)})$ *such that they form a cut, which is the unique minimal cut for all* λ *in the interval.*

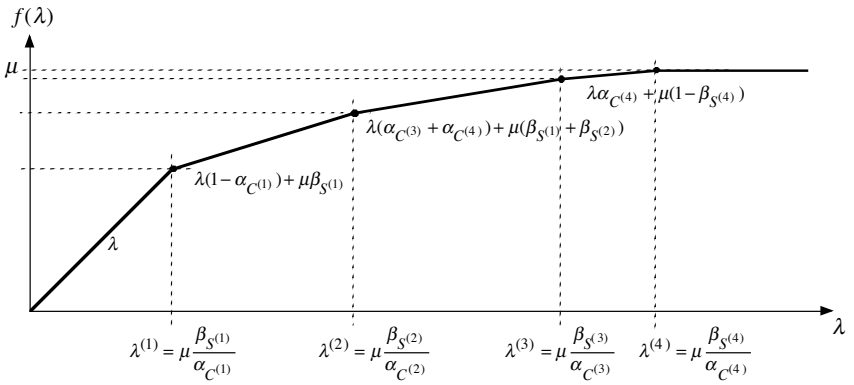

FIG 8. *Maximal flow as a function of* λ.

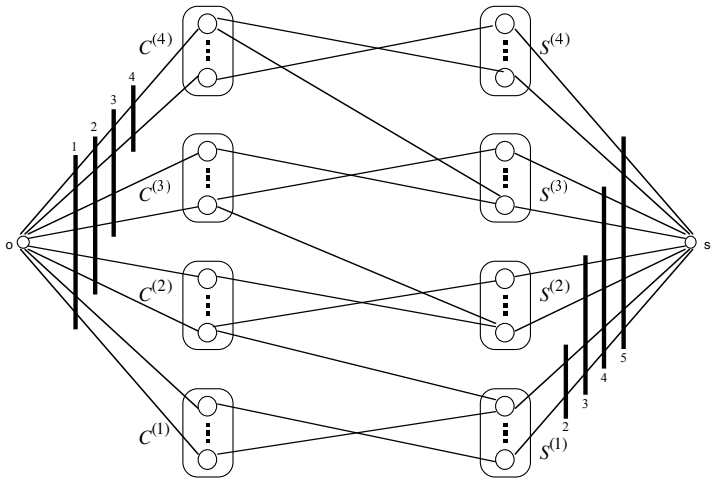

FIG 9. *Decomposition of servers and of customer types.*

(iii) The sets $C(\lambda^{(i-1)},\lambda^{(i)})$ *are decreasing in* i *and the sets* $S(\lambda^{(i-1)},\lambda^{(i)})$ *are increasing in* i*, in the sense that:* $C(\lambda^{(0)},\lambda^{(1)}) \supset C(\lambda^{(1)},\lambda^{(2)}) \supset \cdots \supset C(\lambda^{(L)},\lambda^{(L+1)})$ *and* $S(\lambda^{(0)},\lambda^{(1)}) \subset S(\lambda^{(1)},\lambda^{(2)}) \subset \cdots \subset S(\lambda^{(L)},\lambda^{(L+1)})$.

Theorem 3.6 induces a decomposition of the servers and of the customer types, as detailed in the next corollary. The decomposition is illustrated in Fig. 9. In this figure we have minimal cuts, which belong to a case where there are 4 breakpoints in $f(\lambda)$, which correspond to 5 intervals, and 5 minimal cuts as numbered.

COROLLARY 3.7. *If the maximal flow $f(\lambda)$ has breakpoints $0 = \lambda^{(0)} < \lambda^{(1)} < \cdots < \lambda^{(L)} < \lambda^{(L+1)} = \infty$, then there is a unique partition of the set of servers into non-empty subsets $\mathcal{S}^{(1)}, \ldots, \mathcal{S}^{(L)}$ and of the set of customer types into non-empty subsets $\mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(L)}$, such that:*

(i) The minimal cut for $\lambda^{(i-1)} < \lambda < \lambda^{(i)}$ consists of

$$\mathcal{C}(\lambda^{(i-1)}, \lambda^{(i)}) = \bigcup_{k \geq i} \mathcal{C}^{(k)}, \qquad \mathcal{S}(\lambda^{(i-1)}, \lambda^{(i)}) = \bigcup_{k<i} \mathcal{S}^{(k)},$$

for $i = 1, 2, \ldots, L+1$.

(ii) The maximal flow $f(\lambda)$ where $\lambda^{(i-1)} < \lambda < \lambda^{(i)}$, is:

$$f(\lambda) = \lambda \sum_{k \geq i} \alpha_{\mathcal{C}^{(k)}} + \mu \sum_{k<i} \beta_{\mathcal{S}^{(k)}},$$

for $i = 1, \ldots, L+1$.

(iii) The values of the breakpoints are:

$$\lambda^{(1)} = \mu \frac{\beta_{\mathcal{S}^{(1)}}}{\alpha_{\mathcal{C}^{(1)}}} < \lambda^{(2)} = \mu \frac{\beta_{\mathcal{S}^{(2)}}}{\alpha_{\mathcal{C}^{(2)}}} < \cdots < \lambda^{(L)} = \mu \frac{\beta_{\mathcal{S}^{(L)}}}{\alpha_{\mathcal{C}^{(L)}}}.$$

(iv) Consider the subsystem composed of servers $\mathcal{S}^{(i)}$ and of customer types $\mathcal{C}^{(i)}$, with arrival rates $\lambda\alpha_c$ for customers of type c. Then this system is stable for $\lambda < \lambda^{(i)}$ and unstable for $\lambda \geq \lambda^{(i)}$.

(v) Consider the subsystem composed of servers $\bigcup_{k=i}^{L} \mathcal{S}^{(k)}$, and of customer types $\bigcup_{k=i}^{L} \mathcal{C}^{(k)}$, with arrival rates $\lambda\alpha_c$ for customers of type c. Then this system is stable for $\lambda < \lambda^{(i)}$ and unstable for $\lambda \geq \lambda^{(i)}$.

(vi) For $\lambda^{(i-1)} < \lambda < \lambda^{(i)}$ the maximal flow solution of the whole system will have zero flow on arcs from customers $c \in \mathcal{C}^{(k)}$ to servers $m_j \in \mathcal{S}^{(l)}$ for all $k \geq i > l$.

The following corollary gives another way of solving the max flow problem and decomposing the sets of servers and customer types, and pinpoints the nature of this decomposition:

COROLLARY 3.8. *The sets $\mathcal{C}^{(i)}, \mathcal{S}^{(i)}$ have the following characterization:*

$$\mathcal{C}^{(i)} = \arg\min_{C \subseteq \mathcal{C} \setminus \bigcup_{k<i} \mathcal{C}^{(k)}} \frac{\beta_{\mathcal{S}(C) \setminus \bigcup_{l<i} \mathcal{S}^{(l)}}}{\alpha_C}, \qquad \mathcal{S}^{(i)} = \mathcal{S}(\mathcal{C}^{(i)}) \setminus \bigcup_{l<i} \mathcal{S}^{(l)}.$$

Intuitively, the picture is as follows: Under complete resource pooling $\mathcal{C}^{(1)} = \mathcal{C}$, $\mathcal{S}^{(1)} = \mathcal{S}$. When there is no resource pooling, for some subsets of customers, the requirement that $\alpha_C < \beta_{\mathcal{S}(C)}$ is violated. As λ increases,

the subset of customers, which have the least value of $\beta_{\mathcal{S}(C)}/\alpha_C$, becomes overloaded when λ reaches $\mu\frac{\beta_{\mathcal{S}(C)}}{\alpha_C}$, and this defines $\mathcal{C}^{(1)}$ and $\mathcal{S}^{(1)}$. This leaves servers $\mathcal{S}\backslash\mathcal{S}^{(1)}$ to serve the remaining customer types $\mathcal{C}\backslash\mathcal{C}^{(1)}$. Note that even if $c \in \mathcal{C}\backslash\mathcal{C}^{(1)}$ can be served by a server in $\mathcal{S}^{(1)}$, this will not happen in the max flow solution, since all the servers in $\mathcal{S}^{(1)}$ are fully occupied by $\mathcal{C}^{(1)}$. The remaining servers and customer types now behave like a subsystem with $\tilde{\alpha}_c = \frac{\alpha_c}{1-\alpha_{\mathcal{C}^{(1)}}}$, $\tilde{\beta}_{m_j} = \frac{\beta_{m_j}}{1-\beta_{\mathcal{S}^{(1)}}}$. If in the remaining subsystem, $\frac{\tilde{\beta}_{\mathcal{S}(C)}}{\tilde{\alpha}_C} \geq 1$ for all subsets, then $L = 2$ and the minimum of these ratios will be reached by $\mathcal{S}\backslash\mathcal{S}^{(1)}$, $\mathcal{C}\backslash\mathcal{C}^{(1)}$. Else, if the minimum is again < 1, the subsets $C, \mathcal{S}(C)$ with minimal ratio of $\beta_{\mathcal{S}(C)\backslash\mathcal{S}^{(1)}}/\alpha_C$ will be $\mathcal{S}^{(2)}, C^{(2)}$, and the servers in $\mathcal{S}^{(2)}$ will become overloaded when $\lambda = \mu\frac{\beta_{\mathcal{S}^{(2)}}}{\alpha_{C^{(2)}}}$, and so on (see Figs. 8 and 9).

Example (continued). We return to the 5 server 5 customer types system of the example in Section 3.2. We consider long term behavior as a function of the input rate λ.

The system is stable for $0 < \lambda < 40$.

At $\lambda = 40$ the inflow of customers of type c_1 reaches $\lambda_{c_1} = 20$ which equals the capacity of $\mathcal{S}(c_1) = \{m_1, m_2\}$. For all $\lambda > 40$, servers m_1, m_2 are fully occupied by a queue of customers of type c_1, which grows at rate $\frac{1}{2}\lambda - 20$.

At $\lambda = 53.\bar{3}$ the inflow of customers of types c_2, c_3 reaches $\lambda_{\{c_2,c_3\}} = 20$ which is the capacity of the two remaining servers that can serve them, m_3, m_4. For all $\lambda > 53.\bar{3}$ servers m_3, m_4 are fully occupied by a queue of customers of types c_2, c_3, which grows at rate $\frac{6}{16}\lambda - 20$.

At $\lambda = 80$ the inflow of customers of types c_4, c_5 reaches $\lambda_{\{c_4,c_5\}} = 10$ which is the capacity of the last remaining server that can serve them, m_5. For all $\lambda > 80$ server m_5 is fully occupied by a queue of customers of types c_4, c_5, which grows at rate $\frac{2}{16}\lambda - 10$.

For $\lambda > 80$ all servers are overloaded, the system is split to three subsystems: $\mathcal{S}^{(1)}, \mathcal{C}^{(1)} = (c_1, m_1, m_2)$, $\mathcal{S}^{(2)}, \mathcal{C}^{(2)} = (c_2, c_3, m_3, m_4)$, and $\mathcal{S}^{(3)}, \mathcal{C}^{(3)} = (c_4, m_5, m_5)$.

3.5. *Limiting behavior of overloaded system under incomplete resource pooling.* Following our study of the fluid limits of our system, and the decomposition of the set of servers and of customer types when there is no resource pooling, we can now describe the limiting behavior of our system as $t \to \infty$ in the case that there is no resource pooling, as a function of the total arrival rate λ. In the following theorem we use the notation developed in Section 3.4.

THEOREM 3.9. *Assume the system has incomplete resource pooling, as in Section 3.4 and that the total arrival rate is* $\lambda^{(l)} < \lambda < \lambda^{(l+1)}$. *Then as* $t \to \infty$ *the following convergences will occur for the state of the system,* $\mathfrak{s} = (S(t), Q_1(t), \ldots, Q_J(t))$:

(i) The permutation $S(t)$ *as* $t \to \infty$ *will consist of a permutation of* $S^{(1)}$ *followed by a permutation of* $S^{(2)}$ *and so on up to a permutation of* $S^{(l)}$, *followed by a permutation of the remaining servers.*

(ii) As $t \to \infty$, *the queue between the last server of* $S^{(k)}$ *and the first server of* $\mathcal{S}^{(k+1)}$, *for* $k = 1, \ldots, \min\{l, L-1\}$ *will diverge, growing at a rate* $\mu(\frac{\beta_{\mathcal{S}^{(k+1)}}}{\alpha_{\mathcal{C}^{(k+1)}}} - \frac{\beta_{\mathcal{S}^{(k)}}}{\alpha_{\mathcal{C}^{(k)}}})\alpha_{\bigcup_{j\le k}\mathcal{C}^{(j)}}$.
If $l = L$, *i.e.* $\lambda > \lambda_L$, *the queue after the last server will grow at rate:* $\lambda - \mu\frac{\beta_{\mathcal{S}^{(L)}}}{\alpha_{\mathcal{C}^{(L)}}}$.

(iii) For $k = 1, \ldots, l$, *the probability distribution of the permutation of* $S^{(k)}$ *and the queue length between the servers* $\mathcal{S}^{(k)}$ *will converge to the stationary distribution of the FCFS infinite matching model for the subsystem of* $\mathcal{C}^{(k)}, \mathcal{S}^{(k)}$.

(iv) If $l < L$, *then the probability distribution of the permutation of the remaining servers and the queue lengths between them and behind the last of them, and the ordered set of idle servers, will converge to the stationary distribution of the stable system consisting of* $\bigcup_{k>l}\mathcal{C}^{(k)}, \bigcup_{k>l}\mathcal{S}^{(k)}$, *as given by Theorem 2.1.*

The diverging queues will diverge almost surely. The convergence of the probabilities to stationary probabilities will be in total variation distance. The overloaded subsystems and the remaining stable system will converge in distribution to independent processes.

PROOF. It follows from the results of Sections 3.2 and 3.4 that the fluid limits of the processes Y_k for the various servers will eventually coalesce into subsets, which will move together, with the servers of the subsets $S^{(k)}$ moving together at rates increasing with k for $k = 1, \ldots, l$, and the remaining servers will move together with $A(t)$ at the rate λ. This implies (i). The rates in (ii) are obtained directly from (3.6) and (3.7).

It is then seen that for each subsystem the transition rates, conditional on the diverging queues being > 0 are exactly those of the FCFS matching model for the subsystems $k = 1, \ldots, l$, and for the remaining subsystem they are equal to those of a stable system, consisting of $\bigcup_{k>l}\mathcal{C}^{(k)}, \bigcup_{k>l}\mathcal{S}^{(k)}$. Parts (iii) and (iv) then follow by Lemma 1.

The independence follows, since the various subsystems have independent transitions, given that the diverging queues are > 0. □

4. Discussion of many server scaling. In this section we give some indications on the behavior of the parallel service system under FCFS-ALIS, when the number of servers is large. Our assumption now is that there are n_j servers of type m_j, $j = 1, \ldots, J$, so that the total service capacity is $\mu = \sum n_j \mu_{m_j}$ of which a fraction $\beta_j = n_j \mu_{m_j}/\mu$ belongs to servers of type m_j. The total arrival rate is $\lambda = \sum_c \lambda_c$ and a fraction α_c of them are of type c. We also assume that customers of type c have patience distribution F_c. We fix $\alpha_c, F_c, \beta_j, \mu_{m_j}$ and let n_j and λ increase. The following discussion is heuristic, and will require further research to verify it.

Our main premise is that the behavior of the system with many servers will be on two time scales:

- The total number of customers waiting, the rate of abandonment, and the number of idle servers behave similar to a system with a single customer type and a single server type. This is also indicated in the seminal paper of Talreja and Whitt [34].
- The allocation of servers to customers will behave as in the infinite matching model, similar to the limiting behavior that we found in Section 3.

We assume first that complete resource pooling holds, and discuss many server behavior under the three regimes, of QD (quality driven, total offered load < 1), ED (efficiency driven, overloaded system), and QED (quality and efficiency driven, critically loaded). Stability in all these regimes is guaranteed by the abandonments. When there is no resource pooling we then discuss decomposition similar to Sections 3.4, 3.5. Finally, we discuss some generalizations.

4.1. *Resource pooled system under QD regime.* When the number of servers is large and the system is underloaded ($\lambda < \mu$), there will always be some idle servers, there will be no waiting time and no customers will abandon. In that case, our main premise is that, because of ALIS, all servers, regardless of their types, will have the same idle time distribution. Hence, a server of type m_j will have cycles of serving at rate μ_{m_j}, and then idling for a random time T which has the same distribution for all j. Furthermore, the types of the servers which become idle in sequence are i.i.d.

As the $\lambda, n_j \to \infty$, the idle time T will become constant, and the fraction of services of type m_j will then be

$$\tilde{\beta}_j = n_j \Big/ \lambda \left(\frac{1}{\mu_{m_j}} + T \right) \tag{4.1}$$

The sequence of matches will then behave as in the infinite bipartite matching model, with α and $\tilde{\beta}$.

4.2. *Resource pooled system under ED regime.* When the number of servers is large and the system is overloaded ($\lambda > \mu$), there will always be customers waiting in the system. In that case, our main premise is that, because of FCFS, all customers, regardless of their types, will have the same waiting time distribution (see [34]), from which the abandonment rates can be obtained. Furthermore, the types of the customers that are served in sequence will be i.i.d.

As the $\lambda, n_j \to \infty$, the waiting time W will become constant. The total service rate $\tilde{\lambda}$, of which a fraction $\tilde{\alpha}_c$ are of type c, will then be:

$$\tilde{\lambda} = \lambda \sum_{c \in \mathcal{C}} \alpha_c (1 - F_c(W)), \qquad \tilde{\alpha}_c = \frac{\lambda \alpha_c (1 - F_c(W))}{\tilde{\lambda}}. \tag{4.2}$$

The sequence of matches will then behave as in the infinite bipartite matching model, with $\tilde{\alpha}$ and β.

4.3. *Resource pooled system under QED regime.* When λ and n_j are balanced, so that $\mu - \lambda = \kappa\sqrt{\lambda}$, and we let $\lambda, n_j \to \infty$, then servers will be busy almost all the time and almost no customers will abandon. Our premise then is that customer types and server types in sequence will be i.i.d. The sequence of matches will then behave as in the infinite bipartite matching model, with α and β.

4.4. *No resource pooling.* In that case the system will decompose into subsystems in a unique way, as in Sections 3.4, 3.5. Each of those subsystems will have complete resource pooling, and will be in one of the above regimes and behave accordingly.

4.5. *Generalizations.* We believe that these results also hold under the following conditions:

(i) Replace Poisson arrivals by general independent stationary arrival streams of rates λ_c. Arrivals will no longer be i.i.d., but dependence between successive arrivals will be short range.

(ii) Replace independent exponential service times by independent general service times of rates μ_{m_j}. The individual servers will then become available at times which will form a stationary process, and they will be almost independent, so that again, types of serves will be i.i.d. in sequence.

Furthermore, the sequence of matches will still behave as in the infinite bipartite matching model, even if we allow customer-server type dependent service times, i.e., the service duration of a customer depends on both the customer type and the server type. In that case the calculation of the cor-

responding rates $\tilde{\alpha}_c$ and $\tilde{\beta}_j$ will be more involved. In a recent paper [1] we provide some evidence that this is the case by simulation studies.

Acknowledgements. We are very grateful to the referees for pointing out a gap in our proofs and for the valuable suggestions to improve the presentation of the paper.

APPENDIX A: PROOFS FOR SECTION 2 OF THE PAPER

A.1. Proof of Theorem 2.1. We proceed in the following steps: We first obtain the transition rates into state $\mathfrak{s}$, we then set up the partial balance equilibrium equations, finally we check that they are satisfied by $\pi_X(\mathfrak{s})$, and calculate the normalizing constant. The ergodicity condition follows directly from the expression for the normalizing constant.

Because the proof is quite similar to the proof of product form solution in [35], we skip some details.

A.1.1. *Transitions and transition rates into a state* $\mathfrak{s}$. Denote

(A.1)
$$\delta_j(M) = \begin{cases} \frac{\lambda_{\mathcal{U}(\{M_1,\ldots,M_j\})}}{\lambda_{\mathcal{U}(\{M_1,\ldots,M_j,M\})}}, & \text{if } \mathcal{U}(\{M_1,\ldots,M_j,M\}) \neq \emptyset, \\ 0, & \text{if } \mathcal{U}(\{M_1,\ldots,M_j,M\}) = \emptyset, \end{cases} \qquad j = 1,2,\ldots,J.$$

With $\mathfrak{s} = (M_1, n_1, \ldots, M_i, n_i, M_{i+1}, \ldots, M_J)$ we have:

(i) Transition due to a departure, where server becomes idle. The originating state is

$$idle_{kl}(\mathfrak{s}) = (M_1, n_1, \ldots, M_k, n_k - l, M_{i+1}, l, \ldots, M_i, n_i, M_{i+2}, \ldots, M_J),$$

and the transition probability, conditional on service completion by server M_{i+1}, is

$$\begin{aligned} p_{k,l}(\mathfrak{s}) &= \delta_k(M_{i+1})^l \delta_{k+1}(M_{i+1})^{n_{k+1}} \ldots \delta_i(M_{i+1})^{n_i}, \quad k \geq 1,\ l = 0,\ldots,n_k, \\ p_{0,0}(\mathfrak{s}) &= p_{1,n_1}(\mathfrak{s}). \end{aligned}$$

This transition is illustrated in Fig. 10

(ii) Transition due to a departure and start of a new service. The originating state is

$$\begin{aligned} &swap_{k,l,j}(\mathfrak{s}) \\ &\quad = (M_1, n_1, \ldots, M_k, n_k - l, M_j, l, \ldots, M_{j-1}, n_{j-1} + 1 + n_j, M_{j+1}, \ldots, M_J), \end{aligned}$$

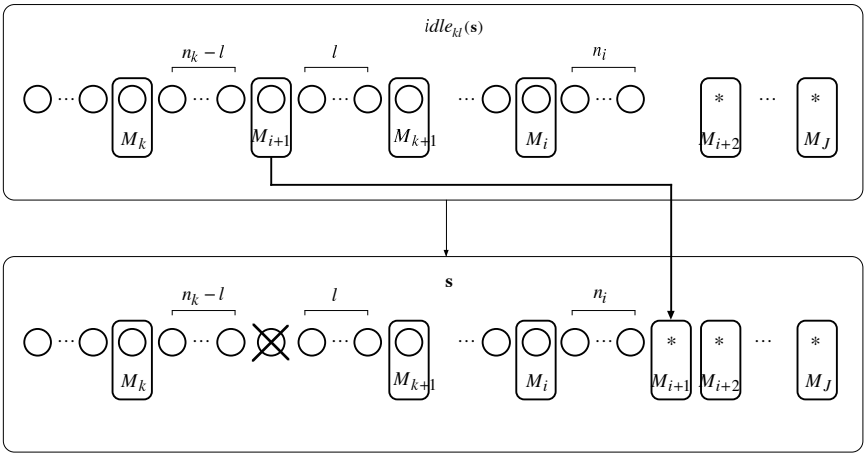

FIG 10. *Transition from state* $idle_{k,l}(\mathfrak{s})$ *to state* $\mathfrak{s}$.

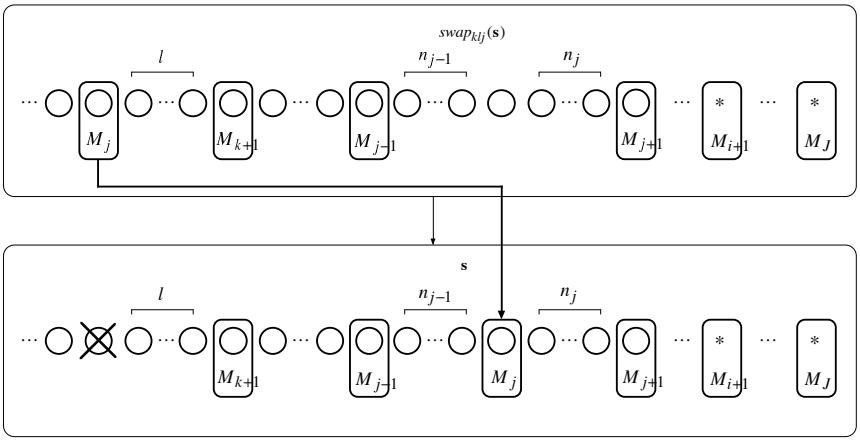

FIG 11. *Transition from state* $swap_{k,l,j}(\mathfrak{s})$ *to state* $\mathfrak{s}$.

and the transition probability, conditional on service completion by server M_j, is

$$\begin{aligned} q_{k,l,j}(\mathfrak{s}) &= \delta_k(M_j)^l \delta_{k+1}(M_j)^{n_{k+1}} \dots \delta_{j-1}(M_j)^{n_{j-1}} \left(1 - \delta_{j-1}(M_j)\right), \\ &\qquad j = 2, \dots, J, 1 \le k < j, l = 0, \dots, n_k, \\ q_{0,0,j}(\mathfrak{s}) &= q_{1,n_1,j}(\mathfrak{s}), \\ q_{0,0,1}(\mathfrak{s}) &= 1. \end{aligned}$$

This transition is illustrated in Fig. 11.

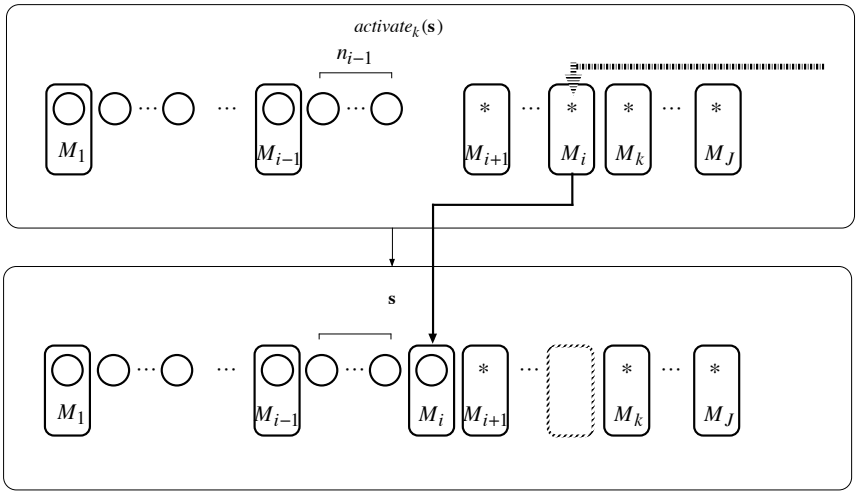

FIG 12. *Transition from state* $activate_k(\mathfrak{s})$ *to state* $\mathfrak{s}$.

(iii) Transition due to an arrival that joins the queue. The originating state is

$$wait(\mathfrak{s}) = (M_1, n_1, \ldots, \ldots, M_i, n_i - 1, M_{i+1}, \ldots, M_J), \quad n_i > 0,$$

and the transition rate is $\lambda_{\mathcal{U}(\{M_1,\ldots,M_i\})}$.

(iv) Transition due to an arrival that activates one of the servers. The originating state is

$$activate_k(\mathfrak{s}) = (M_1, n_1, \ldots, M_{i-1}, n_{i-1}, M_{i+1}, \ldots, M_{k-1}, M_i, M_k, \ldots, M_J),$$

and the transition rate is $\lambda_{\mathcal{C}(M_i)\setminus\mathcal{C}(\{M_k,\ldots,M_J\})}$ for $k = i+1, \ldots, J$, and in the case that M_J is activated, we use the convention that $k = J+1$, and the rate is $\lambda_{\mathcal{C}(M_J)}$. This transition is illustrated in Fig. 12.

A.1.2. *Partial balance equations.* Define:

$$\mathcal{P}_{M_{i+1}}(\mathfrak{s}) = \sum_{k=1}^{i}\sum_{l=0}^{n_k} p_{k,l}(\mathfrak{s})\pi(idle_{k,l}(\mathfrak{s})) + p_{1,n_1}(\mathfrak{s})\pi(idle_{0,0}(\mathfrak{s})),$$

$$\mathcal{Q}_{M_j}(\mathfrak{s}) = \begin{cases} \displaystyle\sum_{k=1}^{j-1}\sum_{l=0}^{n_k} q_{k,l,j}(\mathfrak{s})\pi(swap_{k,l,j}(\mathfrak{s}))+ \\ \qquad +q_{0,0,j}\pi(swap_{0,0,j}(\mathfrak{s})), & \text{if } \mathcal{U}(\{M_1,\ldots,M_j\}) \neq \emptyset, \\ 0, & \text{otherwise.} \end{cases}$$

This system actually satisfies *partial balance equations* which are:

(i) The total probability flux out of state $\mathfrak{s}$ due to an arrival that activates a server equals the total probability flux into state $\mathfrak{s}$ due to a departure which idles a server:

$$\lambda_{C(\{M_{i+1},\ldots,M_J\})}\pi(\mathfrak{s}) \quad = \quad \mu_{M_{i+1}}\mathcal{P}_{M_{i+1}}(\mathfrak{s}). \tag{A.2}$$

(ii) The total probability flux out of state $\mathfrak{s}$, due to an arrival that joins the queue, equals the total probability flux into state $\mathfrak{s}$, due to a departure which is followed by another start of service (so that the set of idle servers is unchanged):

$$\lambda_{\mathcal{U}(\{M_1,\ldots,M_i\})}\pi(\mathfrak{s}) \quad = \quad \sum_{j=1}^{i}\mu_{M_j}\mathcal{Q}_{M_j}(\mathfrak{s}). \tag{A.3}$$

(iii) The total probability flux out of state $\mathfrak{s}$ in which $n_i = 0$ due to a departure, equals the total probability flux into state $\mathfrak{s}$, due to an arrival of a customer which activates server M_i:

$$\begin{aligned}\mu_{\{M_1,\ldots,M_i\}}\pi(\mathfrak{s}) \quad = \quad & \sum_{k=i+1}^{J}\lambda_{\mathcal{C}(M_i)\setminus\mathcal{C}(\{M_k,\ldots,M_J\})}\pi(activate_k(\mathfrak{s})) \\ & + \lambda_{\mathcal{C}(M_i)}\pi(activate_{J+1}(\mathfrak{s})), \qquad n_i = 0.\end{aligned} \tag{A.4}$$

(iv) The total probability flux out of state $\mathfrak{s}$ in which $n_i > 0$ due to a departure, equals the total probability flux into state $\mathfrak{s}$, due to an arrival of a customer which joins the queue:

$$\mu_{\{M_1,\ldots,M_i\}}\pi(\mathfrak{s}) \quad = \quad \lambda_{\mathcal{U}(\{M_1,\ldots,M_i\})}\pi(wait(\mathfrak{s})), \qquad n_i > 0. \tag{A.5}$$

A.1.3. *Verification of the balance equations, calculation of the normalization constant, and ergodicity conditions.* To prove Theorem 2.1 it remains to verify that the expression (2.1) satisfies the partial balance equations (A.2)–(A.5). This is quite straightforward, involving at one step a summation over coefficients of a binomial distribution. The verification is similar to the corresponding verification of Theorem 2 in [35].

Calculation of the normalization constant (2.3) is straightforward from summation of the geometric sequences. Note that it is the inverse of a sum of $J!$ terms.

When calculating the normalizing constant, it is clear that it is finite only if all the geometric sums are finite, and that is the case if and only if the ergodicity condition (2.2) holds.

This completes the proof.

A.2. Proof of Theorem 2.2.

PROOF. Comparing (2.1) and (2.6), and recalling that π_X and π_Y both sum to 1, what we have to show is that for some constant D (which is the same for any $M_1, \ldots, M_i$):

$$\prod_{j=1}^{i} \lambda_{M_j}(\{M_1, \ldots, M_{j-1}\}) = D \sum_{\overline{M}_{i+1}, \ldots, \overline{M}_J \in \mathcal{P}(\{M_{i+1}, \ldots, M_J\})} \prod_{j=i+1}^{J} \left(\lambda_{\mathcal{C}(\{\overline{M}_j, \ldots, \overline{M}_J\})}\right)^{-1}.$$

We take

$$D = \prod_{j=1}^{J} \lambda_{M_j}(\{M_1, \ldots, M_{j-1}\}).$$

By the assignment condition (2.5), D is the same for all permutations of $M_1, \ldots, M_J$, and so it is the same for all choices of $M_1, \ldots, M_i$. We note that

$$\prod_{j=1}^{i} \lambda_{M_j}(\{M_1, \ldots, M_{j-1}\}) = D\Big(\prod_{j=i+1}^{J} \lambda_{M_j}(\{M_1, \ldots, M_{j-1}\})\Big)^{-1}.$$

Hence we need to show that:

$$\begin{aligned} &\Big(\prod_{j=i+1}^{J} \lambda_{M_j}(\{M_1, \ldots, M_{j-1}\})\Big)^{-1} \\ &\quad = \sum_{\overline{M}_{i+1}, \ldots, \overline{M}_J \in \mathcal{P}(\{M_{i+1}, \ldots, M_J\})} \prod_{j=i+1}^{J} \left(\lambda_{\mathcal{C}(\{\overline{M}_j, \ldots, \overline{M}_J\})}\right)^{-1}. \end{aligned} \tag{A.6}$$

This is exactly statement (8) in [5], with the following change of notation:

$(M_{i+1}, \ldots, M_J)$	is	$(j_m, j_{m-1}, \ldots, j_1)$	in [5],
$\lambda_{\mathcal{C}(S)}$	is	$\eta(S)$	in [5],
$\lambda_M(\overline{S})$, $M \in S$	is	$\eta_j(S)$, $j \in S$	in [5].

This statement is proved there. □

APPENDIX B: PROOFS FOR SECTION 3 OF THE PAPER

B.1. Proof of Theorem 3.1.

PROOF. We note that $\alpha_{\mathcal{U}(\{M_1, \ldots, M_i\})} < \beta_{\{M_1, \ldots, M_i\}}$ implies that $\lambda_{\mathcal{U}(\{M_1, \ldots, M_i\})} < \mu_{\{M_1, \ldots, M_i\}}$ whenever $\lambda < \mu$, and so the system is ergodic for all λ such that $\lambda < \mu$, by Theorem 2.1. This proves (i).

Next we observe that the value of $\tilde{B}(\rho)$ is:

$$\tilde{B}(\rho) = \Bigg[\sum_{M_1,\ldots,M_J \in \mathcal{P}(\mathcal{S})} \sum_{i=0}^{J-1} \Big(\prod_{j=1}^{i} \Big(\beta_{\{M_1,\ldots,M_j\}} - \rho\alpha_{\mathcal{U}(\{M_1,\ldots,M_j\})} \Big)^{-1} \times \prod_{j=i+1}^{J} \Big(\rho\alpha_{\mathcal{C}(\{M_j,\ldots,M_J\})} \Big)^{-1} \Big) + \sum_{M_1,\ldots,M_J \in \mathcal{P}(\mathcal{S})} \Big((1-\rho)^{-1} \prod_{j=1}^{J-1} \Big(\beta_{\{M_1,\ldots,M_j\}} - \rho\alpha_{\mathcal{U}(\{M_1,\ldots,M_j\})} \Big)^{-1} \Big) \Bigg]^{-1}.$$

The term $(1-\rho)^{-1}$ is there, because of $\beta_{\{M_1,\ldots,M_J\}} = \alpha_{\mathcal{U}(\{M_1,\ldots,M_J\})} = 1$. All the expressions inside the square brackets remain bounded as $\rho \nearrow 1$, except for $(1-\rho)^{-1}$, which tends to ∞. Hence:

$$\lim_{\rho\nearrow 1} \tilde{B}(\rho) = 0,$$

$$\lim_{\rho\nearrow 1} \tilde{B}(\rho)(1-\rho)^{-1} = \left(\sum_{M_1,\ldots,M_J \in \mathcal{P}(\mathcal{S})} \prod_{j=1}^{J-1} \Big(\beta_{\{M_1,\ldots,M_j\}} - \rho\alpha_{\mathcal{U}(\{M_1,\ldots,M_j\})} \Big)^{-1} \right)^{-1} = B_I.$$

For a single permutation $M_1,\ldots,M_J$ with servers $M_1,\ldots,M_i$ busy, $i < J$, we calculate:

$$\begin{aligned} &\pi(M_1,\cdot,\ldots,M_i,\cdot,M_{i+1},\ldots,M_J) \\ &\quad = \sum_{n_1,\ldots,n_i=0}^{\infty} \pi(M_1,n_1,\ldots,M_i,n_i,M_{i+1},\ldots,M_J) \\ &\quad = \tilde{B}(\rho) \prod_{j=1}^{i} (\beta_{\{M_1,\ldots,M_j\}} - \rho\alpha_{\mathcal{U}(\{M_1,\ldots,M_j\})})^{-1} \prod_{j=i+1}^{J} (\rho\alpha_{\mathcal{C}(\{M_j\ldots,M_J\})})^{-1}. \end{aligned}$$

All terms in this expression, except $\tilde{B}(\rho)$, remain bounded, and hence the whole expression tends to 0 as $\rho \nearrow 1$. This proves (ii).

On the other hand, when all J servers are busy, we calculate:

$$\begin{aligned} \pi(M_1,n_1,\ldots,M_{J-1},n_{J-1},M_J,\cdot) &= \sum_{n_J=0}^{\infty} \pi(M_1,n_1,\ldots,M_J,n_J) \\ &= \tilde{B}(\rho) \prod_{j=1}^{J-1} \frac{\big(\rho\alpha_{\mathcal{U}(\{M_1,\ldots,M_j\})}\big)^{n_j}}{\beta_{\{M_1,\ldots,M_j\}}^{\,n_j+1}} (1-\rho)^{-1}, \end{aligned}$$

where we again used $\beta_{\{M_1,\ldots,M_J\}} = \alpha_{\mathcal{U}(\{M_1,\ldots,M_J\})} = 1$. As $\rho \nearrow 1$, $\tilde{B}(\rho)(1-\rho)^{-1} \to B_I$, and we obtain (iii).

To prove (iv) we note that the transition rates of the queueing process $X(t)$ are bounded between λ and $\lambda+\mu$, and so the jump chain and the continuous process are both ergodic or both non-ergodic at the same time. Furthermore, all the stationary probabilities of states, when not all the servers are busy, will tend to zero for the jump chain of the queueing process as well as for the continuous queueing process, as $\rho \nearrow 1$. Finally, the transition rate at which jumps occur, when all the servers are busy, is $\lambda+\mu$, independent of the state. Hence the stationary probabilities for the jump chain and for the continuous time process, for states when all servers are busy, will tend to the same limits as $\rho \nearrow 1$. This proves (iv). □

B.2. The fluid model and proof of Theorem 3.3. The proof of Theorem 3.3 proceeds in three logical steps. First we prove that fluid limits exist. Next we show that every fluid limit obeys a set of rules that determine its dynamics. Finally, because these dynamics determine a unique fluid limit, the convergence to this fluid limit follows. We start with some preliminaries.

We add another component to our description of the system: We let $T_{S,j}(t)$ be the accumulated time over $(0,t]$ that the order of the servers is S and the j server in the order of servers is serving a customer, for all $j = 1,\ldots,J$ and all $S \in \mathcal{P}$ where $\mathcal{P}$ is the set of all the permutations of servers $m_1,\ldots,m_J$. We let $T(t)$ denote the vector of all the $T_{S,j}(t)$. The dynamics of our sequence of systems are fully described by $Z(t) = (A(t), T(t), Y(t), S(t), Q(t))$.

We study the sequence of systems with random paths $Z^n(t,\omega) = (A^n(t,\omega), T^n(t,\omega), Y^n(t,\omega), S^n(t,\omega), Q^n(t,\omega))$, and use the scaling:

$$\begin{aligned}\bar{Z}^n(t,\omega) &= (\bar{A}^n(t,\omega), \bar{T}^n(t,\omega), \bar{Y}^n(t,\omega), \bar{S}^n(t,\omega), \bar{Q}^n(t,\omega))\\ &= (A^n(nt,\omega)/n, T^n(nt,\omega)/n, Y^n(nt,\omega)/n, S^n(nt,\omega), Q^n(nt,\omega)/n),\end{aligned}$$

as $n \to \infty$. We defined initial conditions that guarantee $\bar{A}^n(0,\omega) \to \bar{A}(0)$ and $\bar{Y}^n(0,\omega) \to \bar{Y}(0)$, independent of ω. We also have by definition that $T^n(0,\omega) = 0$ so $\bar{T}(0) = 0$.

For the times of events, the types of customers, and the identities of servers completing service, the functional strong law of large numbers (FSLLN) applies. We now consider only the set of sample paths ω for which there is FSLLN convergence. We exclude all other sample paths, which are a set of measure zero, hence all our following statements hold almost surely. This means in particular that the number of type c customers among $c^{-q^n},\ldots,c^{-1}$, divided by n converges to $\bar{A}(0)\alpha_c$. Our first proposition follows immediately from the FSLLN:

PROPOSITION B.1. *Equations (3.5) and (3.7) hold almost surely.*

PROOF. (3.5) is immediate. To show (3.7) recall that $Q_j(t)$ consists of those customers between $Y_j(t)$ and $Y_{j+1}(t)$ which belong to $\mathcal{U}(\{M_1,\ldots,M_i\})$, and so $Q_i(t) \sim \text{Binomial}\big(Y_{i+1}(t) - Y_i(t), \alpha_{\mathcal{U}(\{M_1,\ldots,M_i\})}\big)$. Hence $Q_i^n(nt,\omega)/n$ converges by FSLLN to the r.h.s of (3.7) almost surely. □

The next three propositions examine the existence of fluid limits.

PROPOSITION B.2. *Fluid limits of $\bar{T}^n(t), \bar{Y}^n(t)$ exist and are absolutely continuous.*

PROOF. The arguments are similar to Dai and Lin [21]. We consider first $T^n(t,\omega)$. We have for every S, j that the cumulative time $T^n_{S,j}(t,\omega)$ is Lipschitz continuous (with constant 1), and this property is retained by $\bar{T}^n_{S,j}(t,\omega) = T^n_{S,j}(nt,\omega)/n$. Hence for every sample path ω we have a sequence of equi-continuous functions, and therefore (this is the argument of [21], based on the Arzela-Ascoli Theorem, see Royden [33]) there exists a subsequence, indexed by divergent r such that $\bar{T}^r_{S,j}(t,\omega)$ converges to $\bar{T}_{S,j}(t)$ as $r \to \infty$, u.o.c.. Since the number of components of $T^n(t)$ is finite (in fact $J \times J!$), we can by a standard argument select successive subsequences and end up with a subsequence r such that for all components simultaneously, $\bar{T}^r(t,\omega) \to \bar{T}(t)$.

To examine $Y^r(t,\omega)$ we partition the time during which $Y^r_j(t,\omega)$ is evolving into subsets of time given by the $T^r_{S,j}(t,\omega)$. Consider now the service completions of the server in position j during the time accumulated by $T^r_{S,j}(t,\omega)$: The server in position j is M_j, with service times that are exponentially distributed with rate μ_{M_j}. By the memoryless property of the exponential distribution, the total number of service completions by server M_j conditional of the value of $T^r_{S,j}(t)$ is distributed as $N^r_{S,j} \sim \text{Poisson}\big(\mu_{M_j} T^r_{S,j}(t,\omega)\big)$, or to put it more generally, $N^r_{S,j}(t)$ is a Poisson process with time dependent rate $\mu_{M_j} \frac{d}{dt} T^r_{S,j}(t)$. To find the next customer to serve, while the permutation of the servers is S, server M_j will skip all the customers which have already been served by the servers which are ahead of him in the permutation, and he will also skip all the customers which can only be served by servers which are behind him in the permutation. For a fixed permutation S, counting from one customer that M_j can serve to the next, the number of customers is distributed as a Geometric random variable. Again, by the memoryless property of the Geometric distribution, we have that for every job that server M_j will complete during $T^r_{S,j}(t,\omega)$, he will move ahead over a random number of customers which is geometrically

distributed, with a parameter which we denote $p_{S,j}$ (we will calculate it later).

We let $Y^r_{S,j}(t,\omega)$ be the total number of customers that the server in position j passes (serving or skipping them) during the accumulated time $T^r_{S,j}(t,\omega)$. Then by Wald's equation this will have expected value $\mu_{M_j} T^r_{S,j}(t,\omega)/p_{S,j}$ (in fact it will be distributed as a Poisson random variable with that parameter). This means that $Y^r_{S,j}(t)$ is a non-homogenous compound Poisson process. Hence, by the functional strong law of large numbers, we get that $\bar{Y}^r_{S,j}(t,\omega) = Y^r_{S,j}(rt,\omega)/r \to \bar{T}_{S,j}(t)\mu_{M_j}/p_{S,j} = \bar{Y}_{S,j}(t)$ as $r \to \infty$ u.o.c. This is for the same sequence of r that defines all the $\bar{T}_{S,j}$. Finally, $Y^r_j(t,\omega) = Y^r_j(0) + \sum_{S\in\mathcal{P}} Y^r_{S,j}(t,\omega)$, so $\bar{Y}^r_j(t,\omega) \to \bar{Y}_j(t) = \bar{Y}_j(0) + \sum_{S\in\mathcal{P}} \bar{Y}_{S,j}(t)$ as $r \to \infty$ u.o.c.

To complete the proof, we note that since $\bar{T}_j(t)$ are Lipschitz continuous with constant 1, $\bar{Y}_j(t)$ are Lipschitz continuous, with a constant $\max_{\{S,j\}} \mu_{M_j}/p_{S,j}$. □

Limiting fluid behavior of $S^n(nt,\omega)$ is more involved. We noted that while $S(t,\omega)$ is a function from $[0,\infty)$ to the permutations $\mathcal{P}$, fluid limits $\bar{S}(t)$ need to be partitioned permutations. To discuss fluid limits of $\bar{S}^n(t,\omega) = S^n(nt,\omega)$ we need to make some definitions.

DEFINITION B.3. For a fluid limit $\bar{Y}_1(t) \le \cdots \le \bar{Y}_J(t)$ the order $O(t)$ is a partition of $1,\ldots,J$ so that i,j belong to the same subset if and only if $\bar{Y}_i(t) = \bar{Y}_j(t)$. In addition, the subsets are ordered by ascending indexes.

An order O' is a refinement of an order O if every subset in O' is contained in a subset of O.

A simple collision of $\bar{Y}(\cdot)$ and $O(\cdot)$ happens at t if $O(\tau)$ is constant for $\tau \in (t-\epsilon,t)$ and for $\tau \in (t,t+\epsilon)$ for some $\epsilon > 0$, but the partitions in the two intervals are different. In that case the partitions in the two intervals are refinements of $O(t)$.

DEFINITION B.4. A function $\bar{S}(t)$ from $\mathbb{R}_+$ to the set of partitions of $s_1,\ldots,s_J$ is said to be a fluid limit compatible with $\bar{Y}(t)$ if:

- For every t, $\bar{S}(t)$ is an ordered partition of a permutation $M_1,\ldots,M_J$ according to $O(t)$. That means that $\bar{S}(t)$ consists of subsets $(S_1,\ldots,S_L)$, $O(t)$ consists of subsets $(O_1,\ldots,O_L)$, and $M_j \in S_l$ if $j \in O_l$.
- In every interval in which $O(t)$ is constant, $\bar{S}(t)$ is a constant partition of a constant permutation.
- If $O(t)$ has a simple collision at t, then $\bar{S}(t)$ is a partition of the same permutation as in the interval preceding t, and of the same permutation as in the interval succeeding t.

DEFINITION B.5. We say that $\bar{S}^r(t,\omega)$ converges to $\bar{S}(t)$ as $r \to \infty$ if for any interval $t \in [0,T]$ there is an n_0, so that $\bar{S}(t)$ is an ordered partition of $\bar{S}^r(t,\omega)$ for all $r > n_0$.

The following two proposition summarize what we can say about fluid limits of $S^n(nt,\omega)$.

PROPOSITION B.6. *Assume that for a divergent sub-sequence r, $\bar{Y}^r(t,\omega) \to \bar{Y}(t)$, and consider a closed interval $[t',t'']$ in which $O(t)$ is constant. Then there exists a constant partition $\bar{S} = \bar{S}(t)$ of the servers according to $O(t)$, such that for some divergent subsequence r' of r, the partition of $S^{r'}(r't,\omega)$ according to $O(t)$ is equal to $\bar{S}$ for all $t \in [t',t'']$ and for all r'.*

PROOF. Because $O(t)$ is constant over $[t',t'']$ and $\bar{Y}(t)$ are continuous, there will be an $\epsilon > 0$ such that components of $\bar{Y}(t)$ that belong to different subsets in $O(t)$ are never closer than 3ϵ. By the u.o.c. convergence of $\bar{Y}^r(t,\omega)$ there is a n_0 such that for all $r > n_0$, $|\bar{Y}^r(t,\omega) - \bar{Y}(t)| < \epsilon$ for all $t \in [t',t'']$, so components of $\bar{Y}^r(t,\omega)$ for $r > n_0$ which belong to different parts of $O(t)$ never meet in the interval $[t',t'']$. But this implies that for every $r > n_0$, the partition of the permutation of servers $S^r(rt,\omega)$ into the subsets determined by $O(t)$ is a constant partition for all $t \in [t',t'']$ (we are not saying it is the same constant partition for different values of r). But the total number of possible partitions is finite (it is $\frac{J}{J+1}\frac{(2J)!}{J!}$). So at least one of them appears in infinitely many $r > n_0$, which defines the constant partition $\bar{S}$ and the subsequence r'. □

PROPOSITION B.7. *Consider a fluid limit $\bar{Y}^r(t,\omega) \to \bar{Y}(t)$. If the number of 'collisions', instants where $O(t)$ changes, is finite, then for arbitrarily large T, there exists a fluid limit $\bar{S}(t)$ which is compatible with $\bar{Y}(t)$ over $[0,T]$, and a divergent subsequence r' of r such that $\bar{S}^{r'}(t,\omega)$ converges to $\bar{S}(t)$ as $r' \to \infty$.*

PROOF. Assume $0 < t_1 < t_2 < \cdots < t_M < \infty$ are all the collision times. Because there is only a finite number of them they are all simple. Consider the closed intervals $[t_i - \delta, t_i + \delta]$, $i = 1,\ldots,M$, and $[0, t_1 - \delta]$, $[t_i + \delta, t_{i+1} - \delta]$, $[t_M + \delta, T]$, $i = 1,\ldots,M-1$, for some small $\delta > 0$ and arbitrarily large T. Consider the interval $[0,T]$. By the argument of the previous Proposition B.6, we can find $\epsilon > 0$ such that all components of $\bar{Y}(t)$ which belong to different parts of $O(t)$ are never closer than 3ϵ, and we can then find n_0 such that $|\bar{Y}_j^r(t,\omega) - \bar{Y}_j(t)| < \epsilon$ for all $j = 1,\ldots,J$, $0 \le t \le T$ and $r > n_0$.

It is now easy to see that for every $r > n_0$, when we let $\delta \to 0$, that there is a fluid limit candidate $\bar{S}^r(t)$ which is compatible with $\bar{Y}(t)$, so that $\bar{S}^r(t)$ is an ordered partition of the permutation $\bar{S}^r(t,\omega)$ for all t.

We now use again the argument of the previous Proposition B.6: Since there are only $2M+1$ intervals, the total possible sets of $2M+1$ partitions is a finite number. Hence there is one candidate $\bar{S}(t) = \bar{S}^r(t)$ which appears with infinitely many r. This gives us the fluid limit, and the divergent subsequence r', for which $\bar{S}^{r'}(t,\omega) = \bar{S}(t)$ for all r'. □

To summarize, we now know that fluid limits $\bar{A}(t), \bar{Y}(t), \bar{T}(t), \bar{Q}(t)$ exist and are Lipschitz continuous for all $t \geq 0$. We also know, by Proposition B.7, that if $\bar{Y}(t)$ have only a finite number of collisions then also a partition $\bar{S}(t)$ compatible with $\bar{Y}(t)$ exists for all $t \geq 0$, and by the same argument, if $\bar{Y}(t)$ have only a finite number of collisions in $[0,T]$ then a partition $\bar{S}(t)$ compatible with $\bar{Y}(t)$ exists for $t \in [0,T]$.

Once we established existence of fluid limits we will in the next propositions investigate their dynamics. It will turn out that these determine the fluid limits uniquely for all $t \geq 0$.

PROPOSITION B.8. *Consider a fluid limit for which* $\bar{Y}_1(t) < \bar{Y}_2(t) < \cdots < \bar{Y}_J(t)$, *and let* $\bar{S}(t) = (M_1, \ldots, M_J)$. *Then almost surely at all regular* t*:*

$$\frac{d}{dt}\bar{Y}_i(t) = \mu \frac{\beta_{M_i}}{\alpha_{\mathcal{U}(\{M_1,\ldots,M_i\})} - \alpha_{\mathcal{U}(\{M_1,\ldots,M_{i-1}\})}}, \quad i = 1, \ldots, J. \tag{B.1}$$

PROOF. Consider $\bar{Y}(t) = \lim_{r\to\infty} \bar{Y}^r(t,\omega)$. Because $\bar{Y}_i$ are continuous, there exists $\Delta > 0$ such that for all $\tau \in (t-\Delta, t+\Delta)$ the order of $\bar{Y}_i$ is unchanged, $\bar{Y}_1(\tau) < \bar{Y}_2(\tau) < \cdots < \bar{Y}_J(\tau)$, and so for r large enough, for $\tau \in (t-\Delta/2, t+\Delta/2)$, $\bar{Y}_1^r(\tau,\omega) < \bar{Y}_2{}^r(\tau,\omega) < \cdots < \bar{Y}_J^r(\tau,\omega)$, and so we have, for large enough r that $Y_1^r(s,\omega) < Y_2^r(s,\omega) < \cdots < Y_J^r(s,\omega)$, $s \in I = (rt - r\Delta/2, rt + r\Delta/2)$. In particular this means that servers do not overtake each other, and the order of the servers is given by the permutation $S^r(s,\omega) = (M_1, \ldots, M_J)$ and it is unchanged over $s \in I$. Also, by (3.7), $\bar{Q}_i(s) > 0$, $i = 1, \ldots, J-1$, for $s \in I$.

Consider now the movement of server M_i, in the time interval $s \in I$. Since $\bar{Q}_i(t) > 0$, we have that $Q_i^r(s,\omega) > 0$ for $s \in I$, and so server M_i will be busy all the time. Hence, during the time interval I, he will complete a total of L services, where $L \sim \text{Poisson}(\mu\beta_{M_i} r\Delta)$. Server M_i will serve customers, which are in $\mathcal{C}(M_i)$, and which have not been served by any of the servers $M_{i+1}, \ldots, M_J$, i.e he will serve customers in $\mathcal{U}(\{M_1, \ldots, M_i\}) \backslash \mathcal{U}(\{M_1, \ldots, M_{i-1}\})$. Hence, he will skip all customers which are not in $\mathcal{U}(\{M_1, \ldots, M_i\}) \backslash$

$\mathcal{U}(\{M_1,\ldots,M_{i-1}\})$, and so at each service completion, he will move a random number of places G in the sequence of customers, where G is a geometric random variable with parameter (probability of success) $\alpha_{\mathcal{U}(\{M_1,\ldots,M_i\})} - \alpha_{\mathcal{U}(\{M_1,\ldots,M_{i-1}\})}$. Hence, the total change in Y_i^r over the interval I will be:

$$Y_i(rt + r\Delta/2) - Y_i(rt - r\Delta/2) = \sum_{l=1}^{L} G_l, \text{ with } G_l \text{ i.i.d distributed like } G.$$

Hence:

$$\bar{Y}_i^r(t + \Delta/2) - \bar{Y}_i^r(t - \Delta/2) = \frac{\sum_{l=1}^{L} G_l}{r},$$

which, by Wald's equation and the FSLLN, converges as $r \to \infty$ to

$$\bar{Y}_i(t + \Delta/2) - \bar{Y}_i(t - \Delta/2) = \Delta\mu\beta_{M_i} \frac{1}{\alpha_{\mathcal{U}(\{M_1,\ldots,M_i\})} - \alpha_{\mathcal{U}(\{M_1,\ldots,M_{i-1}\})}}$$

from which (B.1) follows. □

Proposition B.8 clarifies how single isolated servers move in the fluid limits. The next proposition studies movement of servers which stay together in the fluid limit.

PROPOSITION B.9. *Consider a fluid limit for which* $\bar{Y}_{k-1}(\tau) < \bar{Y}_k(\tau) = \cdots = \bar{Y}_l(\tau) < \bar{Y}_{l+1}(\tau)$ *for some* $k < l$ *and for all* $\tau \in (t - \Delta, t + \Delta)$. *Let* $\bar{S}(\tau) = (S', \{M_k,\ldots,M_l\}, S'')$ *for the same range of* τ, *where* S', S'' *are the subsets of servers preceding and succeeding* $M_k,\ldots,M_l$ *(their order may be known, but it is irrelevant here). Then:*

$$\frac{d}{dt}\bar{Y}_i(t) = \mu \frac{\beta_{\{M_k,\ldots,M_l\}}}{\alpha_{\mathcal{U}(\{M_1,\ldots,M_l\})} - \alpha_{\mathcal{U}(\{M_1,\ldots,M_{k-1}\})}}, \quad i = k,\ldots,l. \tag{B.2}$$

PROOF. We consider the processes $Y_k^r(s,\omega),\ldots,Y_l^r(s,\omega)$, $s \in I = (rt - r\Delta/2, rt + r\Delta/2)$, and their fluid scaling, $\bar{Y}_k^r(\tau,\omega),\ldots,\bar{Y}_l^r(\tau,\omega)$, $\tau \in (t - \Delta/2, t + \Delta/2)$. As before, because $\bar{Q}_{k-1}(\tau) > 0$, $\bar{Q}_l(\tau) > 0$, for r large enough these processes move in isolation from the other Y_j during $s \in I$, and they consist of the movement of the fixed set of servers $S = \{M_k,\ldots,M_l\}$. Note that these servers may change their order many times during the time interval I. For r large enough, $\bar{Y}_j^r$ is arbitrarily close to $\bar{Y}_j$ uniformly over $(t - \Delta/2, t + \Delta/2)$. Hence for r large enough, we have that $\bar{Y}_k^r(t - \Delta/2) \approx \cdots \approx \bar{Y}_l^r(t - \Delta/2)$, and also $\bar{Y}_k^r(t + \Delta/2) \approx \cdots \approx \bar{Y}_l^r(t + \Delta/2)$, in the sense that $\approx$ will be a distance, which is negligible relative to Δ.

The servers in S are working all the time, so they will process a total of $L \sim \text{Poisson}(\mu\beta_{\{M_k,\ldots,M_l\}} r\Delta)$. They will all start approximately at the same place, and end up approximately at the same place, processing all the customers of types in $\mathcal{U}(\{M_1,\ldots,M_l\})\backslash\mathcal{U}(\{M_1,\ldots,M_{k-1}\}$, and skipping all the other customers, between their approximately common starting and ending positions. The total distance travelled by all the processors in S will therefore be approximately equal to $\sum_{l=1}^{L} G_l$ where G_l are again i.i.d. geometric random variables with probability of success $\alpha_{\mathcal{U}(\{M_1,\ldots,M_l\})} - \alpha_{\mathcal{U}(\{M_1,\ldots,M_{k-1}\})}$.

Thus, proceeding to the limit as in the proof of Proposition B.8, we get that

$$\bar{Y}_i(t-\Delta/2) - \bar{Y}_i(t+\Delta/2) = \Delta\mu \frac{\beta_{\{M_k,\ldots,M_l\}}}{\alpha_{\mathcal{U}(\{M_1,\ldots,M_l\})} - \alpha_{\mathcal{U}(\{M_1,\ldots,M_{k-1}\})}}$$

and (B.2) follows. □

For the next proposition we will make use of the following elementary lemma, the proof of which may be found in [22]

LEMMA 2. *Let $g(t)$ be an absolutely continuous nonnegative function on $t \geq 0$ and let $\dot{g}(t)$ denote its derivative whenever it exists.*

(i) If $g(t) = 0$ and $\dot{g}(t)$ exists, then $\dot{g}(t) = 0$.

(ii) Assume the condition that for some $\epsilon > 0$, whenever $g(t) > 0$ and $\dot{g}(t)$ exists, then $\dot{g}(t) < -\epsilon$. Then $g(t) = 0$ for all $t > \delta$ where $\delta = g(0)/\epsilon$. Furthermore $g(\cdot)$ is nonincreasing and hence, once it reaches zero, it stays there forever.

PROPOSITION B.10. *Assume that complete resource pooling holds, and that we start from $\bar{Q}_i(0) = 0$, $i = 1,\ldots,J-1$, $\bar{Q}_J(0) > 0$. Then for some $\Delta > 0$, we will have $\bar{Q}_i(t) = 0$, $i = 1,\ldots,J-1$, and $\frac{d}{dt}\bar{Y}_i(t) = \mu$, $i = 1,\ldots,J$, for $0 < t < \Delta$.*

PROOF. By continuity of $\bar{Q}$ we can find $\Delta > 0$ such that $\bar{Q}_J(t) > 0$, $0 < t < \Delta$, and so during $0 < t < \Delta$ all servers will be busy. We wish to show that all the servers move at the same rate. We will show that indeed, if $\bar{Q}_i(t) > 0$ at some $0 < t < \Delta$, then $\frac{d}{dt}\bar{Q}_i(t) < -\epsilon < 0$ which by Lemma 2 implies that $\bar{Q}_i(t) = 0$, $i = 1,\ldots,J-1$ for $0 < t < \Delta$. Applying formula (B.2) to $\mathcal{S}$, we then get $\frac{d}{dt}\bar{Y}_i(t) = \mu$ for $0 < t < \Delta$.

Assume then that $\bar{Q}_k(t) > 0$ for some k, at a regular time t (i.e. all derivatives exist at t). Let i be such that $i = \min\{k : \bar{Q}_k(t) > 0\}$ and $\frac{d}{dt}\bar{Y}_1(t) = \cdots = \frac{d}{dt}\bar{Y}_i(t)$, and let j be such that $j = \max\{k : \bar{Q}_k(t) > 0\}$ and

$\frac{d}{dt}\bar{Y}_{j+1}(t) = \cdots = \frac{d}{dt}\bar{Y}_J(t)$, where $1 \le i \le j \le J-1$. Consider the partition $\bar{S}(t) = (S_1, S_2, S_3)$ where $S_1 = \{M_1, \ldots, M_i\}$, $S_2 = \{M_{i+1}, \ldots, M_j\}$, $S_3 = \{M_{j+1}, \ldots, M_J\}$. Then the servers S_1 move together and the servers S_3 move together at t. Specializing Proposition B.9 to S_1 and S_3 we have:

$$\frac{d}{dt}\bar{Y}_1(t) = \mu\frac{\beta_{S_1}}{\alpha_{\mathcal{U}(S_1)}}, \qquad \frac{d}{dt}\bar{Y}_J(t) = \mu\frac{\beta_{S_3}}{\alpha_{\mathcal{C}(S_3)}}.$$

By complete resource pooling, $\beta_{S_1} > \alpha_{\mathcal{U}(S_1)}$ and $\alpha_{\mathcal{C}(S_3)} > \beta_{S_3}$, and therefore $\frac{d}{dt}\bar{Y}_1(t) > \frac{d}{dt}\bar{Y}_J(t)$. Looking at all possible sets S_1, S_3 we can find $\epsilon > 0$ such that

$$\frac{d}{dt}\bar{Y}_J(t) - \frac{d}{dt}\bar{Y}_1(t) = \sum_{k=1}^{J-1}\frac{d}{dt}\bar{Q}_k(t) < -\epsilon.$$

The proposition follows. □

Complete resource pooling is also necessary for all the servers to move together as the next proposition shows:

PROPOSITION B.11. *Assume that there is no complete resource pooling. Assume we start from $\bar{Q}_i(0) = 0$, $i = 1, \ldots, J-1$, $\bar{Q}_J(0) > 0$. Then immediately the set of servers will split into more than one subset, which will move at different rates.*

PROOF. Assume to the contrary that all servers move together. Then by Proposition B.10, for all servers $\frac{d}{dt}\bar{Y}_i(t) = \mu$, $0 < t < \Delta$.

If there is no resource pooling, then there is a subset of servers S such that $\beta_S < \alpha_{\mathcal{U}(S)}$. We will show that for any time t, some of the servers in S will move at a rate which is $< \mu$. This will prove the proposition.

Assume first that the servers in S move together, and are behind all other servers. Then the rate at which they move will be, by Proposition B.9, $\frac{d}{dt}\bar{Y}_i(t) = \mu\frac{\beta_S}{\alpha_{\mathcal{U}(S)}} < \mu$. Assume next that the servers in S split into $S = M_1 \cup \cdots \cup M_L$ subsets, each of which moves together, and that all these subsets are behind all the servers in $\overline{S}$. Then as argued in the proof of Proposition B.10, the servers of the last subset M_1 will move at a rate slower than $\mu\frac{\beta_S}{\alpha_{\mathcal{U}(S)}} < \mu$.

Finally, if these subsets of S, which move together, are not behind all the servers in $\overline{S}$, then the servers of the last subset M_1 will have more customers to serve than if they were moving in the very back. Hence the rate of moving for $i \in M_1$ will be:

$$\frac{d}{dt}\bar{Y}_i(t) \le \frac{\beta_{M_1}}{\alpha_{\mathcal{U}(M_1)}} \le \frac{\beta_S}{\alpha_{\mathcal{U}(S)}} < \mu.$$ □

We now use the extended definition of complete resource pooling for subsets, Definition 3.2, to prove part (ii) of Theorem 3.3, and equation (3.6), and to complete the proof of the theorem.

Proof of part (ii) of Theorem 3.3. Because $\bar{Q}_{k-1}(t) > 0$ and $\bar{Q}_l(t) > 0$, the servers $S = \{M_k, \ldots, M_l\}$ will move in isolation of all the other servers during a time interval $t < \tau < t + \Delta$, and will be between servers S' and S''. In their movement, they will process all the customers in $\mathcal{U}(S \cup S')\backslash\mathcal{U}(S')$, and only those customers. The condition that S has complete resource pooling between S' and S'' means that in a system that would consist only of servers S and customer types $\mathcal{U}(S \cup S')\backslash\mathcal{U}(S')$, with $\tilde{\beta}, \tilde{\alpha}$ as defined above, there would be complete resource pooling. Hence, by Propositions B.10, B.11, it would be necessary and sufficient for the servers of these subsystems to move together. But the movement of the servers S when they are between S', S'' and $\bar{Q}_{k-1}(t) > 0$ and $\bar{Q}_l(t) > 0$, is exactly as if they were a separate system, with the only difference that they will actually also skip over all the customers types in $\mathcal{C}(S'')$. Hence, by Propositions B.10, B.11 they will stay together if and only if S has complete resource pooling between S' and S''. □

Completing the proof of Theorem 3.3. We have shown that fluid limits exist for $t \in [0, T]$ as long as they include only a finite number of simple isolated collisions. We have also shown that they satisfy (3.5), (3.6), (3.7). In Section 3.4 we show, employing results form flows in networks which are independent of the stochastic model, that the decomposition of a set of servers into subsets which have complete resource pooling is unique. Equations (3.5), (3.6), (3.7) and the uniqueness of the decomposition completely determine the evolution of $\bar{Z}(t) = (\bar{A}(t), \bar{T}(t), \bar{Y}(t), \bar{S}(t), \bar{Q}(t))$, and shows it to have only a finite number of collisions. This means that every fluid limit that starts from given $\bar{Z}(0) = (\bar{A}(0), \bar{T}(0), \bar{Y}(0), \bar{S}(0), \bar{Q}(0))$ has to follow the same unique trajectory, for almost all ω, and for any subsequence r for which there is convergence to a fluid limit. But this implies that $\bar{Z}^n(t, \omega) = (\bar{A}^n(t, \omega), \bar{T}^n(t, \omega), \bar{Y}^n(t, \omega), \bar{S}^n(t, \omega), \bar{Q}^n(t, \omega))$ converges almost surely to the unique fluid limit. □

B.3. Maxflow network analog, and proofs for Section 3.4. In what follows we use the terms, notation, and results as formulated in Ford and Fulkerson's book [23], pages 1–14, see also [10]. In a directed network with origin and terminal o, t, a cut is given by a partition of the nodes into two sets, one of which includes o and the other includes t. For our network it is given by $\{o, \overline{C}, S\}$ and $\{t, C, \overline{S}\}$, for some subset of customer types C and some subset of servers S, where $\overline{C}$, $\overline{S}$ are the complements of C, S.

The capacity of the cut is the sum of the capacities of arcs directed from the o part to the t part. For our network this will be the sum of the capacities of the arcs from o to the nodes in C, the arcs from the nodes of S to t, and the arcs from $\overline{C}$ to $\overline{S}$. For any cut with finite capacity, there are no arcs from $\overline{C}$ to $\overline{S}$, hence $S(\overline{C}) \subseteq S$, and $C(\overline{S}) \subseteq C$. We will only consider cuts with finite capacity. Rather than talk about the cut as a partition, we will describe a cut by the sets C and S. The capacity of a (C, S) cut is then $\lambda_C + \mu_S$. The celebrated max-flow min-cut theorem states that the maximal o to t flow through the network equals the capacity of the minimal cut.

Proof of proposition 3.5. The proof is similar to proofs that were given in [8, 18], we give it here for completeness.

Sufficiency: We note first that if max flow is λ and the unique minimal cut consists of the arcs $(o, c), c \in \mathcal{C}$, then there exists $\epsilon > 0$ such that for capacities $\lambda_c(1 + \epsilon)$ the max flow is $\lambda(1 + \epsilon)$. But then, for any subset of customer types C, the total flow from o to the nodes in C equals $(1+\epsilon)\lambda_C$, so the total flow from C to the server nodes $\mathcal{S}(C)$ equals at least $(1+\epsilon)\lambda_C$ (it may be more, since $\mathcal{S}(C)$ may receive flow from additional customer nodes), and so the total flow from server nodes $\mathcal{S}(C)$ to t equals at least $(1 + \epsilon)\lambda_C$. But the capacity of the arcs from server nodes $\mathcal{S}(C)$ to t equals $\mu_{\mathcal{S}(C)}$, so we must have have $\mu_{\mathcal{S}(C)} \geq (1 + \epsilon)\lambda_C > \lambda_C$. Hence the condition (2.2) for stability of the queueing system with total arrival rate λ holds, and the system is stable.

Necessity: Assume that the maximal flow is $< \lambda$ or that the maximal flow is λ but the cut $(\mathcal{C}, \emptyset)$ consisting of arcs $(o, c), c \in \mathcal{C}$ is not the unique minimal cut. Then there exists a minimal cut (C, S) with capacity $\leq \lambda$ where $C \neq \mathcal{C}$ and $S \neq \emptyset$. But then, as observed above, $\mathcal{S}(\overline{C}) \subseteq S$. Hence $\mu_{\mathcal{S}(\overline{C})} \leq \mu_S \leq \lambda - \lambda_C = \lambda_{\overline{C}}$, which contradicts the condition (2.2). □

Proof of Theorem 3.6. (i) The maximal flow problem for each fixed λ is a linear program with feasible and bounded solutions, and when it is considered with varying λ, it is a parametric linear program. As such it will have intervals in which the same basis is optimal, and such intervals will cover the whole line of $\lambda > 0$. Within such an interval, the flows of the optimal solution will be affine functions of λ. The optimal maximal flow objective $f(\lambda)$ is clearly a continuous non-decreasing function of λ. Consider now the optimal flows for λ' and λ'' with $\lambda' < \lambda''$ and look at $\lambda = (1 - \theta)\lambda' + \theta\lambda''$. The convex combination of the optimal flows for λ' and for λ'' is a feasible flow for λ with objective value $(1-\theta)f(\lambda') + \theta f(\lambda'')$, which can only be suboptimal. This proves the concavity. Finally, if $0 < \lambda < \min\{\mu_{m_1}, \ldots, \mu_{m_J}\}$, the maximal flow is λ, so the slope of the initial interval

of $f(\lambda)$ is 1, and for λ such that $\min_{c\in C}\lambda_c > \mu$, the maximal flow is μ, so the slope of $f(\lambda)$ in the last half infinite interval is 0.

(ii) Consider an interval $\lambda^{(i-1)} < \lambda < \lambda^i$ in which the maximal flow is $f(\lambda) = a + b\lambda$. For fixed λ_0 in the interval, consider a minimum cut, so its capacity is $a+b\lambda_0$. For any other λ in the interval the capacity of this cut is an affine function of λ, and it will be at least $f(\lambda)$ because any cut capacity is an upper bound on the flow. This implies that the capacity of the cut is equal to $a + b\lambda$ for all $\lambda^{(i-1)} < \lambda < \lambda^i$, and hence the cut is a minimal cut for all $\lambda^{(i-1)} < \lambda < \lambda^i$. Hence we have shown that any minimal cut for λ_0 is in fact a minimal cut for the whole range of values $\lambda^{(i-1)} < \lambda < \lambda^i$.

Assume now that there are two different minimal cuts, (C_1, S_1), and (C_2, S_2). The capacity of these minimal cuts will be $\lambda\alpha_{C_1} + \mu\beta_{S_1}$ and $\lambda\alpha_{C_2} + \mu\beta_{S_2}$ respectively, where both expressions are equal to $a + b\lambda$. By Corollary 5.4 in [23], the cut formed by $C_1 \cap C_2$ and $S_1 \cup S_2$ will also be a minimal cut, with capacity $\lambda\alpha_{C_1\cap C_2} + \mu\beta_{S_1\cup S_2}$. Recall that $\alpha_c > 0$ for all customer types $c \in \mathcal{C}$. Hence we cannot have equal capacities for the three cuts for a range of values of λ unless $C_1 = C_2$. Once we have that $C_1 = C_2$, we see that we cannot have equality of the capacities of the three cuts unless also $S_1 = S_2$. This proves the uniqueness in each interval.

(iii) We consider $\lambda' < \lambda''$. Let (C, S) be the minimal cut for λ'. We partition the network into two subnetworks, $\{o, \overline{C}, S, t\}$ and $\{o, C, \overline{S}, t\}$, with respective max flows $\mu\beta_S$ and $\lambda'\alpha_C$, and min cuts $(\emptyset, S)$ and $(C, \emptyset)$. We now look for the max flow in the two networks, when we go from λ' to λ''. The flow for $\{o, \overline{C}, S, t\}$ remains unchanged at $\mu\beta_S$. The flow for $\{o, C, \overline{S}, t\}$ may increase, with a new minimal cut (C_1, S_1), where $C_1 \subseteq C$, $S_1 \subseteq \overline{S}$ and maximal flow $\lambda''\alpha_{C_1} + \mu\beta_{S_1}$. We claim that $(C_1, S\cup S_1)$ is a cut for the combined network, because there are no arcs from $\overline{C}$ to $\overline{S}$. This cut has a combined feasible flow of $\lambda''\alpha_{C_1} + \mu\beta_{S_1} + \mu\beta_S$, and is therefore the minimal cut for λ''. This is the monotonicity we needed to show. Strict monotonicity holds if λ', λ'' belong to different intervals. □

Proof of Corollary 3.7. Parts (i) and (ii) and (iii) follow directly from the construction of minimal cuts in Theorem 3.6. Parts (iv) and (v) then follow from Proposition 3.5 and the expression in (iii). Finally, for part (vi), we note that when $\lambda^{(i-1)} < \lambda < \lambda^{(i)}$, then the servers $\mathcal{S}^{(j)}$ receive flow $\mu\beta_{\mathcal{S}^{(j)}}$ from customer types $\mathcal{C}^{(j)}$ for $j = 1, \ldots, i-1$, so there can be no additional flow to any of them from nodes of customer types in $\mathcal{C}^{(i)}, \ldots, \mathcal{C}^{(L)}$. □

Proof of Corollary 3.8. One way to see this is that, by Corollary 3.7 (iv), for each $C \subset \mathcal{C}^{(i)}$ we have $\frac{\beta_{S(C)}}{\alpha_C} < \frac{\beta_{\mathcal{S}^{(i)}}}{\alpha_{\mathcal{C}^{(i)}}}$, and by (iv), $\frac{\beta_{\mathcal{S}^{(i)}}}{\alpha_{\mathcal{C}^{(i)}}}$ are monotone increasing in i. □

REFERENCES

[1] Adan, I. J. B. F., Boon, M. A. A., Weiss, G. (2013). Design and evaluation of overloaded service systems with skill based routing, under FCFS policies. *Performance Evaluation*, to appear.

[2] Adan, I. J. B. F., Foss, S., Weiss, G. (2012). Local stability in a transient Markov chain, working paper.

[3] Adan, I. J. B. F., Hurkens, C., Weiss, G. (2010). A reversible Erlang loss system with multitype customers and multitype servers. *Probability in Engineering and Informational Sciences* **24** 535–548. MR2725348

[4] Adan, I. J. B. F., Weiss, G. (2011). Exact FCFS matching rates for two infinite multi-type sequences. *Operations Research* **60** 475–489. MR2935072

[5] Adan, I. J. B. F., Weiss, G. (2011). A loss system with skill based servers under assign to longest idle server policy. *Probability in Engineering and Informational Sciences* **26** 307–321. MR2943331

[6] Adan, I. J. B. F., Wessels, J., Zijm, W. H. M. (1989). Queuing analysis in a flexible assembly system with a job-dependent parallel structure. *Operations Research Proceedings 1988*, Springer-Verlag, Berlin, 551–558.

[7] Adan, I. J. B. F., Wessels, J., Zijm, W. H. M. (1991). Flexible assembly and shortest queue problems. *Modern Production Concepts, Theory and Applications*, G. Fandel, G. Zaepfel (eds.), Springer-Verlag, Berlin, 644–659.

[8] Adan, I., Foley, R., McDonald, D. (2009). Exact asymptotics of the stationary distribution of a Markov chain: A production model. *Queueing Systems* **62** 311–344. MR2546420

[9] Aerts, J., Korst, J., Verhaegh, W. (2007). Load balancing for redundant storage strategies: Multiprocessor scheduling with machine eligibility. *Journal of Scheduling* **4** 245–257. MR2017535

[10] Ahuja, R. K., Magnanti, T. L., Orlin, J. B. (1993). *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, NJ. MR1205775

[11] Aksin, Z., Armony, M., Mehrotra, V. (2007). The modern call-center, a multi-disciplinary perspective on operations management research. *Production and Operations Management* **16** 665–688.

[12] Armony, M. (2005). Dynamic routing in large-scale service systems with heterogeneous servers. *Queueing Systems* **51** 287–329. MR2189596

[13] Armony, M., Ward, A. R. (2010). Fair dynamic routing in large-scale heterogeneous-server systems. *Operations Research* **58**(3) 624–637. MR2680568

[14] Armony, M., Mandelbaum, A. (2011). Routing and staffing in large-scale service systems: The case of homogeneous impatient customers and heterogeneous servers. *Operations Research* **59**(1) 50–65. MR2814218

[15] Atar, R. (2005). Scheduling control for queueing systems with many servers: Asymptotic optimality in heavy traffic. *The Annals of Applied Probability* **15** 2606–2650. MR2187306

[16] Atar, R., Mandelbaum, A., Shaikhet, G. (2009). Simplified control problems for multiclass many-server queueing systems. *Mathematics of Operations Research* **34** 795–812. MR2573496

[17] Bramson, M. (2008). *Stability of Queueing Networks*, Springer. MR2445100

[18] Caldentey, R., Kaplan, E. H., Weiss, G. (2009). FCFS infinite bipartite matching of servers and customers. *Advances in Applied Probability* **41** 695–730. MR2571314

[19] Chen, H., Mandelbaum, A. (1994). Hierarchical modeling of stochastic networks, Part I: Fluid models. *Stochastic Modeling and Analysis of Manufacturing Systems*, Springer, 47–105.

[20] Dai, J. G. (1995). On positive Harris recurrence of multiclass queueing networks: A unified approach via fluid limit models. *Annals of Applied Probability* **5** 49–77. MR1325041

[21] Dai, J. G., Lin, W. (2005). Maximum pressure policies in stochastic processing networks. *Operations Research* **53** 197–218. MR2131925

[22] Dai, J., Weiss, G. (1996). Stability and instability of fluid models for certain re-entrant lines. *Mathematics of Operations Research* **21** 115–134. MR1385870

[23] Ford, L. R., Jr., Fulkerson, D. R. (1962). *Flows in Networks*, Princeton University Press, Princeton. MR0159700

[24] Foss, S., Chernova, N. (1998). On the stability of a partially accessible multi-station queue with state-dependent routing. *Queueing Systems* **29** 55–73. MR1643598

[25] Garnett, O., Mandelbaum, A. (2000). An introduction to skill-based routing and its operational complexities. http://iew3.technion.ac.il/serveng/Lectures/SBR.pdf.

[26] Green, L. (1985). A queueing system with general-use and limited-use servers. *Operations Research* **33** 168–182. MR0786054

[27] Gurvich, I., Whitt, W. (2010). Service-level differentiation in many-server service system via queue-ratio routing. *Operations Research* **58** 316–328. MR2674799

[28] Harrison, J. M., Lopez, M. J. (1999). Heavy traffic resource pooling in parallel-server systems. *Queueing Systems* **33** 339–368 MR1742575

[29] Jennings, O. B., Reed, J. E. (2012). An overloaded multiclass FIFO queue with abandonments. *Operations Research* **60** 1282–1295. MR2998896

[30] Kaplan, E. H. (1988). A public housing queue with reneging and task-specific servers. *Decision Sciences* **19** 383–391.

[31] Mandelbaum, A., Stoylar, A. L. (2004). Scheduling flexible servers with convex delay costs: Heavy-traffic optimality of the generalized cμ-rule. *Operations Research* **52** 836–855. MR2104141

[32] McManus, M. L., Long, M. C., Copper, A., Litvak, E. (2004). Queueing theory accurately models the need for critical care resources. *Anesthesiology* **100** 1271–1276.

[33] Royden, H. L. (1988). *Real Analysis*, 3rd ed, Prentice Hall, New York. MR0928805

[34] Talreja, R., Whitt, W. (2007). Fluid models for overloaded multi-class many-service queueing systems with FCFS routing. *Management Science* **54** 1513–1527.

[35] Visschers, J., Adan, I. J. B. F., Weiss, G. (2012). A product form solution to a system with multi-type customers and multi-type servers. *Queueing Systems* **70** 269–298. MR2886485

[36] Wallace, R. B., Whitt, W. (2005). A staffing algorithm for call centers with skill-based routing. *Manufacturing and Service Operations Management* **7** 276–294.

[37] Zijm, W. H. M., Laarhoven, P. J. M. (1993). Production preparation and numerical control in PCB assembly. *International Journal of Flexible Manufacturing Systems* **5** 187–207.

Department of Mechanical Engineering
Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven, the Netherlands
E-mail: iadan@tue.nl

Department of Statistics
The University of Haifa
Mount Carmel 31905, Israel
E-mail: gweiss@stat.haifa.ac.il