Exact FCFS matching rates for two infinite multi-type sequences

EURANDOM PREPRINT SERIES
2010-025

Exact FCFS matching rates for two infinite multi-type sequences

I. Adan, G. Weiss
ISSN 1389-2355

Exact FCFS matching rates for two infinite multi-type sequences

Ivo Adan* Gideon Weiss†

June 1, 2010

Abstract

We consider an infinite sequence of items of types $\mathcal{C} = \{c_1, \ldots, c_I\}$, and another infinite sequence of items of types $\mathcal{S} = \{s_1, \ldots, s_J\}$, and a bipartite graph G of allowable matches between the types. Matching the two sequences on a first come first served basis defines a unique infinite matching between the sequences. For $(c_i, s_j) \in G$ we define the matching rate r_{c_i,s_j} as the long term fraction of (c_i, s_j) matches in the infinite matching, if it exists. We assume that the types of items in the two sequences are i.i.d. with given probability vectors α, β. We describe this system by a Markov chain, obtain conditions for ergodicity, and derive its stationary distribution which is of product form. We show that if the chain is ergodic, then the matching rates exist almost surely, and give a closed form formula to calculate them.

Keywords: Service system; first come first served policy; multi type customers and servers; infinite bipartite matching; infinite bipartite matching rates; Markov chains; product form solution.

2000 Mathematics Subject Classification: Primary 60J10; Secondary 90B22; 68M20.

1 Introduction

We consider the model suggested by Caldentey, Kaplan and Weiss [4]. We have an infinite sequence of customers, $c^1, \ldots, c^N, \ldots$ and of servers, $s^1, \ldots, s^M, \ldots$. Customers are of types $\{c_1, \ldots, c_I\}$, servers are of types $\{s_1, \ldots, s_J\}$. Customers of type c_i can be served by a subset $\mathcal{S}(c_i)$ of the servers, servers of type s_j can serve a subset $\mathcal{C}(s_j)$ of the customers. A bipartite graph G describes possible matches of customers and servers, where an arc (c_i, s_j) in G indicates that $c_i \in \mathcal{C}(s_j)$, and $s_j \in \mathcal{S}(c_i)$.

A unique first come first served (FCFS) infinite bipartite matching is defined between the two sequences: customer c^N is matched to the first server in the sequence that can serve it and that has not been matched to any of the customers $c^1, \ldots, c^{N-1}$. Equivalently, server s^M is matched to the first customer in the sequence that he can serve, and which has not been matched to any of the previous servers in the sequence. It is easy to see that these two constructions result

*Department of Mathematics and Computer Science, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, the Netherlands, and Department of Quantitative Economics, University of Amsterdam, P.O.Box 19268, 1000 GG Amsterdam, the Netherlands; email iadan@win.tue.nl Research supported in part by the Netherlands Organization for Scientific Research (NWO).

†Department of Statistics, The University of Haifa, Mount Carmel 31905, Israel; email gweiss@stat.haifa.ac.il Research supported in part by Israel Science Foundation Grants 454/05 and 711/09, hospitality of the Newton Institute on Mathematics is gratefully acknowledged.

in the same infinite matching, and indeed the roles of servers and customers in this model are completely symmetric. For each N we denote by $r^N_{c_i,s_j}$ the fraction of (c_i, s_j) matches created between $c^1, \ldots, c^N$ and $s^1, \ldots, s^N$.

We assume that the sequences of customers and servers are randomly generated. The types of customers are i.i.d. drawn from a probability vector α, and the types of the servers are i.i.d. drawn from a probability vector β, with the two sequences independent. This defines a probability distribution on the matches, and in particular on $r^N_{c_i,s_j}$.

For given G, α, β we define the matching rates $r_{c_i,s_j} = \lim_{N\to\infty} r^N_{c_i,s_j}$ if these limits exist almost surely. Obviously the matching rates must satisfy:

$$\begin{aligned} \sum_{c_i \in \mathcal{C}(s_j)} r_{c_i,s_j} &= \beta_{s_j}, \qquad \text{for all } s_j, \\ \sum_{s_j \in \mathcal{S}(c_i)} r_{c_i,s_j} &= \alpha_{c_i}, \qquad \text{for all } c_i. \end{aligned} \tag{1}$$

We refer to these as *the total resource pooling linear equations.* If these equations do not have a non-negative solution, then rates cannot exist, and we say that in this case there can be no complete resource pooling in the system. Unfortunately, these equations are not enough to determine the rates, since in many cases (depending on the structure of the graph G) they may have many nonnegative solutions. In cases when the solution is unique the question of convergence still remains.

Let $\mathcal{C}$, resp. $\mathcal{S}$ denote a subset of customer, resp. server types, and let $\mathcal{S}(\mathcal{C}) = \bigcup_{c_i\in\mathcal{C}} \mathcal{S}(c_i)$, $\mathcal{C}(\mathcal{S}) = \bigcup_{s_j\in\mathcal{S}} \mathcal{C}(s_j)$, and let also $\alpha_{\mathcal{C}} = \sum_{c_i\in\mathcal{C}} \alpha_{c_i}$, $\beta_{\mathcal{S}} = \sum_{s_j\in\mathcal{S}} \beta_{s_j}$. Caldentey, Kaplan and Weiss [4] have shown that the following condition is necessary for the existence of matching rates:

$$\alpha_{\mathcal{C}} \le \beta_{\mathcal{S}(\mathcal{C})}, \text{ for all subsets } \mathcal{C}, \qquad \beta_{\mathcal{S}} \le \alpha_{\mathcal{C}(\mathcal{S})}, \text{ for all subsets } \mathcal{S}.$$

They conjectured that the sharpened condition:

$$\alpha_{\mathcal{C}} < \beta_{\mathcal{S}(\mathcal{C})}, \text{ for all non trivial subsets } \mathcal{C}, \qquad \beta_{\mathcal{S}} < \alpha_{\mathcal{C}(\mathcal{S})}, \text{ for all non trivial subsets } \mathcal{S}. \tag{2}$$

is sufficient for existence of the matching rates. They have also suggested a Markovian description for the matching of each successive server s^N, or for each successive pair c^N, s^N. Using this Markovian description they confirmed the conjecture for some special types of graphs G and calculated the matching rates for some of those. Recently, Busic, Gupta and Mairess [2] have shown that the sharper condition (2) is necessary for the existence of rates, and discussed some related models.

In the current paper we show that indeed (2) is necessary and sufficient for the existence of rates, and obtain a closed form formula (7) for calculating the matching rates. We do so by refining the Markovian description in [4], to obtain a new Markov chain which is associated with the matching of each successive server. For this Markov chain we find a product form stationary distribution (6). The form of this stationary distribution confirms that (2) is sufficient for ergodicity of the chain, and hence proves (see Theorem 2 in [4]) that (2) is sufficient for the existence of the rates. The formula (7) is then derived from the stationary distribution. The Markov chain which we use to describe the matching process uses the same idea which was used by Visschers et al. [9, 10] to describe a queueing system with multi-type customers and multi-type servers.

The motivation for this model can be found in assigning tenants to housing projects (cf. Kaplan [6, 7]), adopting couples to adoptive children, kidney transplants, etc. In a queueing context it relates to situations where servers and customers play symmetric roles, e.g. if both

arrive in independent Poisson streams and an arriving customer (resp. server) is matched to the longest waiting compatible server (resp. customer), and both are then immediately removed from the system. This model is also relevant to skill based routing in call centers. A recent paper of Talreja and Whitt [8] derives further results for such a call center type model, including some matching rates under first come first served.

The rest of the paper is structured as follows: In Section 2 we define the Markov chain and derive its stationary distribution. In Section 3 we obtain the formula (7) for the matching rates. In Sections 5 and 6 we explore the relationship between our model and the manufacturing type queueing system of Visschers et al. [9, 10] and the call center skilled based routing type model of Whitt and Talreja [8].

Remark 1. We assume without loss of generality that in the graph G no two nodes have exactly the same connections. The reason is that our matching mechanism does not distinguish between such nodes. Therefore, if we have for example two server types s', s'' with $\mathcal{C}(s') = \mathcal{C}(s'')$, we will merge them to a single type s and calculate the matching rates for the merged server type s with $\beta_s = \beta_{s'} + \beta_{s''}$. Once we can calculate $r_{s,c}$ for any customer type c, we can retrieve $r_{s',c} = \frac{\beta_{s'}}{\beta_s} r_{s,c},\ r_{s'',c} = \frac{\beta_{s''}}{\beta_s} r_{s,c}$.

2 The Markov chain

We now define a discrete time Markov chain Z_N associated with the matching of successive servers, so that Z_N summarizes the state after the matching of $s^1, \ldots, s^N$. Assume that N is large enough so that $s^1, \ldots, s^N$ contains at least one server of each type s_j for $j = 1, \ldots, J$. Let s^{k_j} be the last server of type s_j among $s^1, \ldots, s^N$, and let c^{l_j} be the customer which is matched to server s^{k_j}. Note that because matching is FCFS, if we look at the customers which were matched to $s^1, \ldots, s^N$, then c^{l_j} is the last of them which is matched to a type s_j server. Let $l_{(1)} < l_{(2)} < \cdots < l_{(J)}$ be the ordered string of $l_1, \ldots, l_J$. This defines a (random) permutation of server types, $S_1, \ldots, S_J$, where S_j is the server that matched customer $c^{l_{(j)}}$. Consider now the customers $c^{l_{(j)}+1}, \ldots, c^{l_{(j+1)}-1}$ (this may be an empty string). Some of them may have been matched to servers s^M where $1 \le M \le N$ and where $M \notin \{k_1, \ldots, k_J\}$. Let n_j be the number of unmatched customers between $c^{l_{(j)}}$ and $c^{l_{(j+1)}}$. We define the state of Z_N as $\mathfrak{s} = (S_1, n_1, S_2, n_2, \ldots, S_{J-1}, n_{J-1}, S_J)$. Figure 1 illustrates a typical state of Z_N. There are five types of customers and five types of servers. The system graph G at the top of the figure has $\mathcal{S}(c_1) = \{s_1, s_5\}$ and $\mathcal{S}(c_i) = \{s_{i-1}, s_i\}, i = 2, \ldots, 5$. The figure illustrates the state Z_N which is seen by server s^{N+1} when he is searching for his match. All previous servers $s^1, \ldots, s^N$, represented by gray dots, have been matched to customers, represented by gray dots and by the five black dots which are $c^{l_{(1)}}, \ldots, c^{l_{(5)}}$, the last customers matched by each type of server. The oblongs around those 5 black dots spell out the type of server that matched each of them. Server s^{N+1} is represented by a black dot, and the white dots represent the remaining unmatched servers and customers. This state is $Z_N = \mathfrak{s} = (s_5, 0, s_1, 3, s_4, 2, s_2, 3, s_3)$.

We will use the following notation:

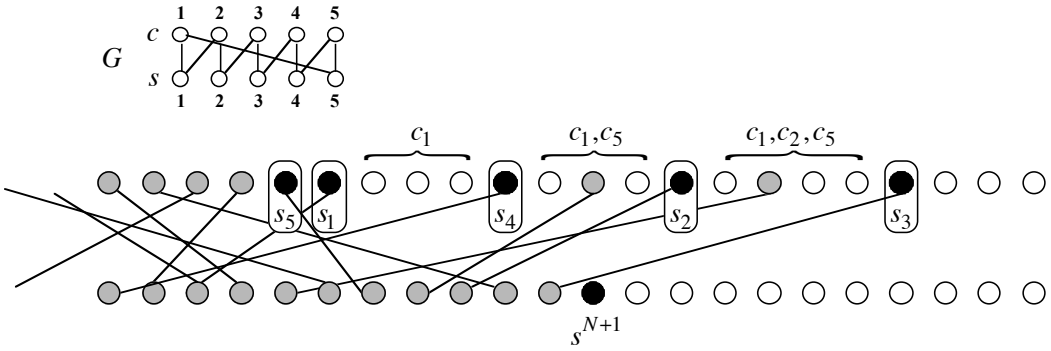

Figure 1: Illustration of the system's Markovian state

- S : an arbitrary server type from the set of server types $\{s_1, \ldots, s_J\}$. The capitalized S points to one of the server types, and in particular in an arbitrary state $\mathfrak{s} = (S_1, n_1, \ldots, n_{J-1}, S_J)$, the sequence $S_1, \ldots, S_J$ is the permutation of the server types as they appear in the order of $c^{l(1)}, \ldots, c^{l(J)}$. Note that the actual server types s_j are not capitalized.
- $\mathcal{C}$: a subset of customer types.
- $\mathcal{S}$: a subset of server types.
- $\mathcal{C}(\mathcal{S})$: the subset of customer types which can be matched to at least one server type in $\mathcal{S}$, equals $\bigcup_{s\in\mathcal{S}} \mathcal{C}(s)$.
- $\mathcal{U}(\mathcal{S})$: the set of customer types which are uniquely served by the set of server types $\mathcal{S}$. Customer types in $\mathcal{U}(\mathcal{S})$ cannot be served by any type of server which is not in $\mathcal{S}$. It is equal to $\overline{\mathcal{C}}(\overline{\mathcal{S}})$, the complement set of all the customers which can be matched to server types in the complement of $\mathcal{S}$. We let by convention $\mathcal{U}(\emptyset) = \emptyset$.
- $\alpha_{\mathcal{C}}$: sum of α_c over $c \in \mathcal{C}$. By convention, $\alpha_\emptyset = 0$.
- $\beta_{\mathcal{S}}$: sum of β_s over $s \in \mathcal{S}$.

Returning to Figure 1 we look at the unmatched customers. There are 3 unmatched customers following directly after the last s_1 match. Clearly those are customers which can only be matched to servers s_5, s_1. Hence, looking at G, they are all of type c_1. Similarly the two unmatched customers following the last s_4 match cannot be matched to s_2, s_3 and hence must belong to $\mathcal{U}(\{s_5, s_1, s_4\}) = \{c_1, c_5\}$, and the last three unmatched customers must be of types $\{c_1, c_2, c_5\} = \mathcal{U}(\{s_5, s_1, s_4, s_2\}) = \overline{\mathcal{C}}(\{s_3\})$.

In general, for a state $\mathfrak{s} = (S_1, n_1, S_2, n_2, \ldots, n_{J-1}, S_J)$ the n_j unmatched customers following directly after the last match of S_j will all belong to $\mathcal{U}(\{S_1, \ldots, S_j\})$. Those unmatched customers include all the customers of types in $\mathcal{U}(\{S_1, \ldots, S_j\})$ which were in the original infinite sequence of customers between $c^{l(j)}$ and $c^{l(j+1)}$. As a result, if $c \in \mathcal{U}(\{S_1, \ldots, S_j\})$ then each of the n_j unmatched customers can be of type c with probability $\frac{\alpha_c}{\alpha_{\mathcal{U}(\{S_1,\ldots,S_j\})}}$. Clearly, $\mathcal{U}(\{S_1, \ldots, S_j\}) \subseteq \mathcal{U}(\{S_1, \ldots, S_j, S_{j+1}\})$, with possibility of equality. Also, it is possible that $\mathcal{U}(\{S_1\}) = \mathcal{U}(\{S_1, S_2\}) = \cdots = \mathcal{U}(\{S_1, \ldots, S_j\}) = \emptyset$ in which case $n_1 = \cdots = n_j = 0$ for all sample paths; states for which $n_j > 0$ but $\mathcal{U}(\{S_1, \ldots, S_j\}) = \emptyset$ are not feasible. Let $\mathfrak{S}$ be the state space of Z_N. Hence, if $\mathcal{P}_J$ is the set of all permutations of $\{s_1, \ldots, s_J\}$, and $\mathbb{Z}_+$ the non-negative integers, then the state space of Z_N is the subset of $\mathcal{P}_J \times \mathbb{Z}_+^{J-1}$ of all (feasible) states $\mathfrak{s}$ satisfying

$n_j > 0$ only if $\mathcal{U}(\{S_1, \ldots, S_i\}) \neq \emptyset$ for all $j = 1, \ldots, J-1$, so

$$\mathfrak{S} = \{(S_1, n_1, \ldots, n_{J-1}, S_J) | (S_1, \ldots, S_J) \in \mathcal{P}_J, n_i \geq 0, n_i = 0 \text{ if } \mathcal{U}(\{S_1, \ldots, S_i\}) = \emptyset, i = 1, \ldots, J\}$$

It is worth noting the following: In the paper of Caldentey, Kaplan and Weiss [4] the matching process of successive servers is described by the Markov chain X_N which lists the ordered string of unmatched customers, and the countable state space of X_N consists of finite ordered strings of customer types. This Markov chain turned out to be intractable, and we believe that it does not in general have a product form stationary distribution. Our current process Z_N is based on Visschers et al. [9, 10], which analyze and obtain product form solutions for a continuous time Markov chain describing a multi-type customer multi-type server queueing system (this will be discussed in Section 5). The process Z_N retains information different from that retained by X_N about the matching process. It records the last match for each type of server, which is not included in the state description of [4], but it does not specify the types of unmatched customers, only how many there are following directly after the last match of each type of server.

We now describe the transition mechanism of Z_N. If the chain is in state $Z_N = \mathfrak{s}$, and s^{N+1} is of type S_i then none of the first $n_1 + \cdots + n_{i-1}$ unmatched customers can match him. He will then consider the n_i unmatched customers following $c^{l(i)}$, and look for a match, and take the first match. The probability for each one of them to provide a match is $\frac{\alpha_{\mathcal{U}(\{S_1,\ldots,S_i\}) \cap \mathcal{C}(S_i)}}{\alpha_{\mathcal{U}(\{S_1,\ldots,S_i\})}}$, and the successive trials are independent. If no match is found among these n_i customers, server s^{N+1} will continue searching along the remaining $n_{i+1} + \cdots + n_{J-1}$ customers, to look for a match, and if none is found he will then search the rest of the infinite sequence following $c^{l(J)}$, where he will eventually find a match after a geometrically distributed number of trials. Recall that all of the n_j unmatched customers following $c^{l(j)}$ are of types $\mathcal{U}(\{S_1, \ldots, S_j\})$, so the probability that one of these n_j unmatched customers following $c^{l(j)}$ will provide a match for s^{N+1} is $\frac{\alpha_{\mathcal{U}(\{S_1,\ldots,S_j\}) \cap \mathcal{C}(S_i)}}{\alpha_{\mathcal{U}(\{S_1,\ldots,S_j\})}}$, and the trials are independent. We denote by $\delta_j(S_i)$ the probability of no match between s^{N+1} of type S_i and one of the n_j unmatched customers between $c^{l(j)}$ and $c^{l(j+1)}$.

The effect of s^{N+1} finding a match among the n_j customers following S_j is that the permutation $S_1, \ldots, S_i, \ldots, S_j, \ldots, S_J$ is replaced by a permutation in which S_i moves to the right and is inserted between S_j and S_{j+1}. In the special case that a match is found among the n_i customers following $c^{l(i)}$ the permutation is unchanged and only the counts change. In the special case that no match is found among $n_1 + \ldots + n_{J-1}$ customers, S_i moves to the rightmost position in the permutation. If the type of s^{N+1} is S_1 and $n_1 > 0$, server s^{N+1} will be matched to the first unmatched customer following $c^{l(1)}$, and the only change in state will be that n_1 is reduced by 1.

This concludes the description of the Markov chain Z_N. Before formulating the global balance equations we establish the following properties of Z_N.

Theorem 1. *Z_N is an irreducible and aperiodic Markov chain.*

Proof. It is obvious from the foregoing description of the states and transitions that the transitions probabilities do not depend on any of the states prior to Z_N, so Z_N is a Markov chain.

It is possible to move with positive probability from any state $(S_1, n_1, \ldots, n_{J-1}, S_J)$ to a state with no unmatched customers between them and (possibly) some other permutation of the server types, say $(\bar{S}_1, 0, \ldots, 0, \bar{S}_J)$, in $\sum_{i=1}^{J-1} n_i$ steps, by having consecutive servers each of which can match a consecutive unmatched customer. One can also move with positive probability from any state $(\bar{S}_1, 0, \ldots, 0, \bar{S}_J)$ to the state $(S_1, n_1, \ldots, n_{J-1}, S_J)$ in $J + \sum_{i=1}^{J-1} n_i$ steps. This is done if successive servers are of type $S_1, \ldots, S_J$ and the infinite sequence of customers starts with a customer of S_1 followed by n_1 customers of types in $\mathcal{U}(\{S_1\})$, and then a customer of S_2, followed by n_2 customers of $\mathcal{U}(\{S_1, S_2\})$, etc. Hence the chain is irreducible.

The chain is aperiodic, since from any state $(S_1, n_1, \ldots, n_{J-1}, S_J)$ one can stay in the same state in the next step if s^{N+1} is of type S_J and the first customer following $c^{l(J)}$ can be matched to S_J. □

To formulate the global balance equations we need to specify the precise transitions *into* state $\mathfrak{s} = (S_1, n_1, \ldots, n_{J-1}, S_J)$. For $j = 1, \ldots, J$, if s^{N+1} is of type S_j then state $\mathfrak{s}$ will be reached from an originating state in which S_j follows S_k, and there are $n_k - l$ unmatched customers between S_k and S_j. Here $k \leq j-1$, with $0 \leq l \leq n_k$. We denote this originating state $\text{swap}_{k,l}^{S_j}(\mathfrak{s})$. A typical transition from $\text{swap}_{k,l}^{S_j}(\mathfrak{s})$ to $\mathfrak{s}$ is illustrated in Figure 2. Note that in the originating state S_{j-1} and S_{j+1} are in consecutive positions in the permutation, with $n_{j-1} + 1 + n_j$ unmatched customers between them, one of which is then matched to s^{N+1}.

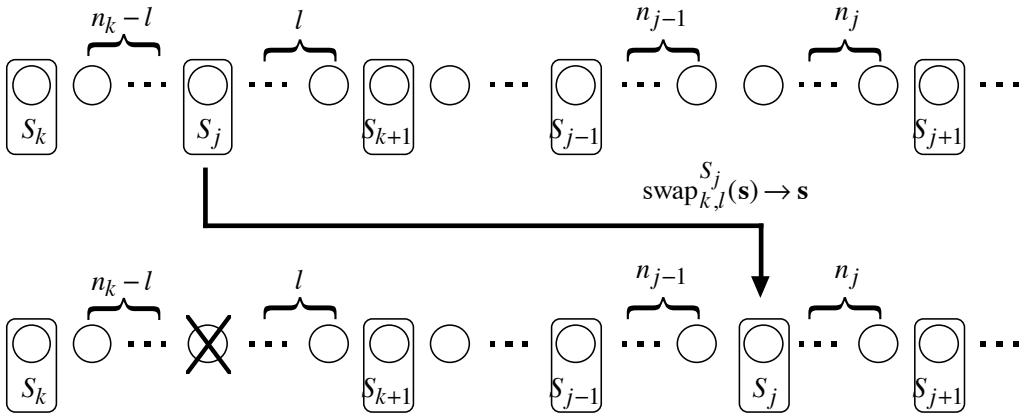

Figure 2: Transition from state $\text{swap}_{k,l}^{S_j}(\mathfrak{s})$ to state $\mathfrak{s}$

To clarify we illustrate some special cases in Figure 3. In the transition $\text{swap}_{k,l}^{S_J}(\mathfrak{s})$ to $\mathfrak{s}$ (Figure 3a), there is obviously no S_{J+1}, and S_J moves from its originating position in the permutation to the last position. If $k = j-1$ we have the transition $\text{swap}_{j-1,l}^{S_j}(\mathfrak{s})$ to state $\mathfrak{s}$ (Figure 3b), in which the permutation remains the same, but the counts of unmatched customers between S_{j-1}, S_j, S_{j+1} change, from $n_{j-1} - l, l + 1 + n_j$ in the originating state to n_{j-1}, n_j in $\mathfrak{s}$. The case of $k = 0$ means that s^{N+1} is of the same type as the leftmost server in the originating state. There are now two possibilities. If there are any unmatched customers following the first server in the originating state, then s^{N+1} would match with the first of them, and the transition would be from $\text{swap}_{0,0}^{S_1}(\mathfrak{s})$ to $\mathfrak{s}$, with the permutation remaining the same and the number of unmatched customers in the first interval reducing from $n_1 + 1$ to n_1 (Figure 3d); in this case $j = 1$. If there are no unmatched customers following the first server in the originating state, then the transition will be from $\text{swap}_{0,0}^{S_j}(\mathfrak{s})$ to $\mathfrak{s}$ (Figure 3c); in this case $j > 1$.

Note that, if $j < J$, the originating state $\text{swap}_{k,l}^{S_j}(\mathfrak{s})$ always has one additional unmatched customer in front of S_{j+1}, i.e. the one matching S_j in state $\mathfrak{s}$. However, if $\mathcal{U}\{S_1, \ldots, S_j\} = \emptyset$, such an additional customer is not possible and thus the state $\text{swap}_{k,l}^{S_j}(\mathfrak{s})$ is not feasible; this means that $\mathfrak{s}$ can not be reached by a match of S_j. Hence, the transition $\text{swap}_{k,l}^{S_j}(\mathfrak{s})$ to $\mathfrak{s}$ is feasible only if $\mathcal{U}\{S_1, \ldots, S_j\} \neq \emptyset$.

We denote the probability of the transition from $\text{swap}_{k,l}^{S_j}(\mathfrak{s})$ to $\mathfrak{s}$ by $q_{k,l}^{S_j}(\mathfrak{s})$, conditional on the

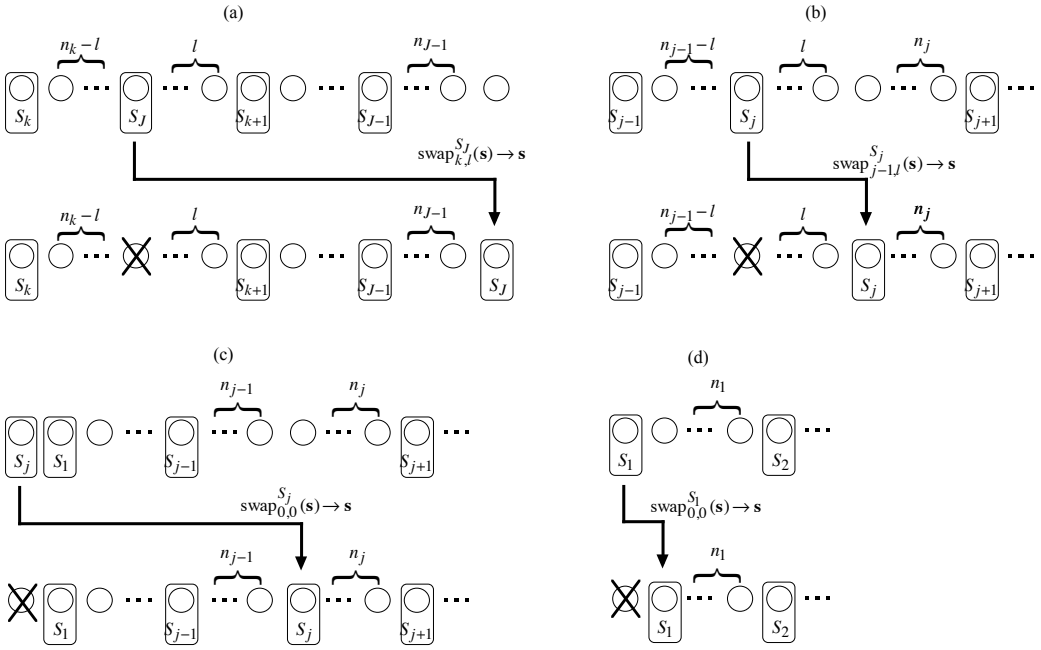

Figure 3: Some additional swap transitions

event that s^{N+1} is of type S_j. For $k > 0, k < j-1$ (see Figures 2, 3(a)) it is given by:

$$q^{S_j}_{k,l}(\mathfrak{s}) = (\delta_k(S_j))^l \, (\delta_{k+1}(S_j))^{n_{k+1}} \cdots (\delta_{j-1}(S_j))^{n_{j-1}} \, (1 - \delta_{j-1}(S_j)), \qquad 0 < k < j-1.$$

In the remaining cases (Figure 3(b,c,d)) it is:

$$\begin{aligned} q^{S_j}_{j-1,l}(\mathfrak{s}) &= (\delta_{j-1}(S_j))^l \, (1 - \delta_{j-1}(S_j)), \qquad j > 1, \\ q^{S_j}_{0,0}(\mathfrak{s}) &= q^{S_j}_{1,n_1}(\mathfrak{s}), \qquad j > 1, \\ q^{S_1}_{0,0}(\mathfrak{s}) &= 1, \end{aligned}$$

where we use

$$\delta_i(S_j) = \frac{\alpha_{\mathcal{U}(\{S_1,\ldots,S_i\})}}{\alpha_{\mathcal{U}(\{S_1,\ldots,S_i,S_j\})}}, \qquad 0 < i < j,$$

to denote the probability of no match for $s^{N+1} = S_j$ with one of the unmatched customers following S_i in the originating state. We set by convention, $\delta_i(S_j) = 0$ if $\mathcal{U}(\{S_1,\ldots,S_i\}) \subseteq \mathcal{U}(\{S_1,\ldots,S_i,S_j\}) = \emptyset$.

Equipped with the above notations, the global balance equations can be formulated as follows:

$$\pi(\mathfrak{s}) = \sum_{j:\mathcal{U}\{S_1,\ldots,S_j\}\neq\emptyset} \beta_{S_j} Q_{S_j}(\mathfrak{s}), \qquad \mathfrak{s} \in \mathfrak{S}, \tag{3}$$

where

$$Q_{S_1}(\mathfrak{s}) = q^{S_1}_{0,0}(\mathfrak{s})\pi(\mathrm{swap}^{S_1}_{0,0}(\mathfrak{s})), \tag{4}$$

$$Q_{S_j}(\mathfrak{s}) = q^{S_j}_{0,0}(\mathfrak{s})\pi(\mathrm{swap}^{S_j}_{0,0}(\mathfrak{s})) + \sum_{k=1}^{j-1}\sum_{l=0}^{n_k} q^{S_j}_{k,l}(\mathfrak{s})\pi(\mathrm{swap}^{S_j}_{k,l}(\mathfrak{s})), \qquad j > 1. \tag{5}$$

Note that $Q_{S_j}(\mathfrak{s})$, $j = 1, \ldots, J$ is the probability that the state $Z_{N+1} = \mathfrak{s}$ has been reached by match of $s^{N+1} = S_j$.

We are now ready to state our main theorem:

Theorem 2. *The global balance equations (3) for the Markov chain Z_N are solved by:*

$$\pi(\mathfrak{s}) = \pi(S_1, n_1, S_2, n_2, \ldots, S_{J-1}, n_{J-1}, S_J) = B \prod_{k=1}^{J-1} \left(\frac{1}{\beta_{\{S_1,\ldots,S_k\}}} \right) \left(\frac{\alpha_{\mathcal{U}\{S_1,\ldots,S_k\}}}{\beta_{\{S_1,\ldots,S_k\}}} \right)^{n_k}, \quad \mathfrak{s} \in \mathfrak{S}, \tag{6}$$

where B is a constant, $\mathcal{U}\{S_1, \ldots, S_k\}$ is the set of customer types which are served exclusively by the servers $\{S_1, \ldots, S_k\}$, and

$$\alpha_{\mathcal{U}\{S_1,\ldots,S_k\}} = \sum_{c \in \mathcal{U}\{S_1,\ldots,S_k\}} \alpha_c, \qquad \beta_{\{S_1,\ldots,S_k\}} = \sum_{j=1}^{k} \beta_{S_j}.$$

A necessary and sufficient condition for ergodicity of Z_N is condition (2), or equivalently, for each subset $\{S_1, \ldots, S_j\}$ of the server types $s_1, \ldots, s_J$:

$$\alpha_{\mathcal{U}\{S_1,\ldots,S_j\}} < \beta_{\{S_1,\ldots,S_j\}}, \qquad j = 1, \ldots, J,$$

in which case $\pi(\mathfrak{s})$ is the stationary distribution with the normalizing constant:

$$B^{-1} = \sum_{\mathcal{P}_J} \frac{1}{(\beta_{\{S_1\}} - \alpha_{\mathcal{U}\{S_1\}})(\beta_{\{S_1,S_2\}} - \alpha_{\mathcal{U}\{S_1,S_2\}}) \cdots (\beta_{\{S_1,\ldots,S_{J-1}\}} - \alpha_{\mathcal{U}\{S_1,\ldots,S_{J-1}\}})}$$

Proof. We will substitute expression (6) into (3) and check that global balance holds. First, for each state $\mathfrak{s} \in \mathfrak{S}$ and $j \in \{1, \ldots, J\}$ such that $\mathcal{U}\{S_1, \ldots, S_j\} \neq \emptyset$, we calculate by substitution of (6) the quantities $\beta_{S_j} q_{k,l}^{S_j}(\mathfrak{s}) \pi(\mathrm{swap}_{k,l}^{S_j}(\mathfrak{s}))/\pi(\mathfrak{s})$ appearing in (3) – (5). For $1 \leq k < j$ we have:

$$\beta_{S_j} q_{k,l}^{S_j}(\mathfrak{s}) \frac{\pi(\mathrm{swap}_{k,l}^{S_j}(\mathfrak{s}))}{\pi(\mathfrak{s})} = \beta_{S_j} \left(1 - \delta_{j-1}(S_j)\right) \left(\delta_k(S_j)\right)^l \frac{\left(\frac{1}{\beta_{\{S_1,\ldots,S_k,S_j\}}}\right)\left(\frac{\alpha_{\mathcal{U}\{S_1,\ldots,S_k,S_j\}}}{\beta_{\{S_1,\ldots,S_k,S_j\}}}\right)^l}{\left(\frac{\alpha_{\mathcal{U}\{S_1,\ldots,S_k\}}}{\beta_{\{S_1,\ldots,S_k\}}}\right)^l}$$

$$\left\{ \prod_{i=k+1}^{j-1} \left(\delta_i(S_j)\right)^{n_i} \frac{\left(\frac{1}{\beta_{\{S_1,\ldots,S_i,S_j\}}}\right)\left(\frac{\alpha_{\mathcal{U}\{S_1,\ldots,S_i,S_j\}}}{\beta_{\{S_1,\ldots,S_i,S_j\}}}\right)^{n_i}}{\left(\frac{1}{\beta_{\{S_1,\ldots,S_i\}}}\right)\left(\frac{\alpha_{\mathcal{U}\{S_1,\ldots,S_i\}}}{\beta_{\{S_1,\ldots,S_i\}}}\right)^{n_i}} \right\} \frac{\frac{\alpha_{\mathcal{U}\{S_1,\ldots,S_{j-1},S_j\}}}{\beta_{\{S_1,\ldots,S_{j-1},S_j\}}}}{\frac{1}{\beta_{\{S_1,\ldots,S_{j-1},S_j\}}}}$$

$$= \beta_{S_j} \left(1 - \delta_{j-1}(S_j)\right) \frac{\alpha_{\mathcal{U}\{S_1,\ldots,S_{j-1},S_j\}}}{\beta_{\{S_1,\ldots,S_k,S_j\}}} \left(\frac{\beta_{\{S_1,\ldots,S_k\}}}{\beta_{\{S_1,\ldots,S_k,S_j\}}} \right)^l \prod_{i=k+1}^{j-1} \left(\frac{\beta_{\{S_1,\ldots,S_i\}}}{\beta_{\{S_1,\ldots,S_i,S_j\}}} \right)^{n_i+1}$$

$$= \left(\alpha_{\mathcal{U}\{S_1,\ldots,S_{j-1},S_j\}} - \alpha_{\mathcal{U}\{S_1,\ldots,S_{j-1}\}}\right) \frac{\beta_{S_j}}{\beta_{\{S_1,\ldots,S_k,S_j\}}} \left(\frac{\beta_{\{S_1,\ldots,S_k\}}}{\beta_{\{S_1,\ldots,S_k,S_j\}}} \right)^l \prod_{i=k+1}^{j-1} \left(\frac{\beta_{\{S_1,\ldots,S_i\}}}{\beta_{\{S_1,\ldots,S_i,S_j\}}} \right)^{n_i+1}$$

$$= \left(\alpha_{\mathcal{U}\{S_1,\ldots,S_{j-1},S_j\}} - \alpha_{\mathcal{U}\{S_1,\ldots,S_{j-1}\}}\right) (1 - \theta_{k,j}) \theta_{k,j}^l \prod_{i=k+1}^{j-1} \theta_{i,j}^{n_i+1} .$$

The first equality is obtained after canceling all the common terms of $\pi(\mathrm{swap}_{k,l}^{S_j}(\mathfrak{s}))$ and $\pi(\mathfrak{s})$. The second and third equalities follow from canceling the α terms with the corresponding $\delta_i(S_j)$ terms. Finally, for the last equality, we denote

$$\theta_{i,j} = \frac{\beta_{\{S_1,\ldots,S_i\}}}{\beta_{\{S_1,\ldots,S_i,S_j\}}}, \qquad 1 \le i < j\,.$$

Similarly we obtain for the term $\beta_{S_j} q_{0,0}^{S_j}(\mathfrak{s})\pi(\mathrm{swap}_{0,0}^{S_j}(\mathfrak{s}))/\pi(\mathfrak{s})$ by substitution of (6), for $j > 1$:

$$\begin{aligned}
\beta_{S_j} q_{0,0}^{S_j}(\mathfrak{s})\frac{\pi(\mathrm{swap}_{0,0}^{S_j}(\mathfrak{s}))}{\pi(\mathfrak{s})} &= \beta_{S_j}\left(1-\delta_{j-1}(S_j)\right)\frac{1}{\beta_{S_j}} \\
&\left\{\prod_{i=1}^{j-1}\left(\delta_i(S_j)\right)^{n_i}\frac{\left(\frac{1}{\beta_{\{S_1,\ldots,S_i,S_j\}}}\right)\left(\frac{\alpha_{\mathcal{U}\{S_1,\ldots,S_i,S_j\}}}{\beta_{\{S_1,\ldots,S_i,S_j\}}}\right)^{n_i}}{\left(\frac{1}{\beta_{\{S_1,\ldots,S_i\}}}\right)\left(\frac{\alpha_{\mathcal{U}\{S_1,\ldots,S_i\}}}{\beta_{\{S_1,\ldots,S_i\}}}\right)^{n_i}}\right\}\frac{\frac{\alpha_{\mathcal{U}\{S_1,\ldots,S_{j-1},S_j\}}}{\beta_{\{S_1,\ldots,S_{j-1},S_j\}}}}{\frac{1}{\beta_{\{S_1,\ldots,S_{j-1},S_j\}}}} \\
= \quad &\left(\alpha_{\mathcal{U}\{S_1,\ldots,S_{j-1},S_j\}} - \alpha_{\mathcal{U}\{S_1,\ldots,S_{j-1}\}}\right)\prod_{i=1}^{j-1}\theta_{i,j}^{n_i+1}
\end{aligned}$$

Performing the summation in (5) we get, for $j > 1$:

$$\begin{aligned}
\beta_{S_j} Q_{S_j}(\mathfrak{s}) &= \beta_{S_j} q_{0,0}^{S_j}(\mathfrak{s})\pi(\mathrm{swap}_{0,0}^{S_j}(\mathfrak{s})) + \beta_{S_j}\sum_{k=1}^{j-1}\sum_{l=0}^{n_k} q_{k,l}^{S_j}(\mathfrak{s})\pi(\mathrm{swap}_{k,l}^{S_j}(\mathfrak{s})) \\
&= \pi(\mathfrak{s})\left(\alpha_{\mathcal{U}\{S_1,\ldots,S_{j-1},S_j\}} - \alpha_{\mathcal{U}\{S_1,\ldots,S_{j-1}\}}\right) \\
&\quad \left[\prod_{i=1}^{j-1}{\theta_{i,j}}^{n_i+1} + \sum_{k=1}^{j-1}\sum_{l=0}^{n_k}(1-\theta_{k,j})\theta_{k,j}^{l}\prod_{i=k+1}^{j-1}\theta_{i,j}^{n_i+1}\right] \\
&= \pi(\mathfrak{s})\left(\alpha_{\mathcal{U}\{S_1,\ldots,S_{j-1},S_j\}} - \alpha_{\mathcal{U}\{S_1,\ldots,S_{j-1}\}}\right).
\end{aligned}$$

To see that the sums of products of all the $\theta_{i,j}$ add up to 1, note that they represent probabilities for Bernoulli trials, of which there are altogether $\sum_{i=1}^{j-1}(n_i+1)$ trials, starting with $n_{j-1}+1$ trials with success probability of $(1-\theta_{j-1,j})$, followed by n_i+1 trials with success probability $(1-\theta_{i,j})$, for $i = j-2,\ldots,2,1$. The summation of terms $\sum_{k=1}^{j-1}\sum_{l=0}^{n_k}$ sums up the probabilities that the first success will be on the first, the second, ... or the last of the trials, while the first term in the square brackets is the probability of no success at all. These obviously add up to 1.

For $j = 1$ the substitution gives:

$$\begin{aligned}
\beta_{S_1} Q_{S_1}(\mathfrak{s}) &= \beta_{S_1} q_{0,0}^{S_1}(\mathfrak{s})\pi(\mathrm{swap}_{0,0}^{S_1}(\mathfrak{s})) \\
&= \beta_{S_1} 1\frac{\alpha_{\mathcal{U}\{S_1\}}}{\beta_{\{S_1\}}}\pi(\mathfrak{s}) \\
&= \pi(\mathfrak{s})\left(\alpha_{\mathcal{U}\{S_1\}} - \alpha_{\mathcal{U}\{\emptyset\}}\right),
\end{aligned}$$

where we used that $\alpha_{\mathcal{U}\{\emptyset\}} = 0$.

Finally, summing up over $j \in \{1,\ldots,J\}$ satisfying $\mathcal{U}\{S_1,\ldots,S_j\} \neq \emptyset$, we get from substitut-

ing (6) in the global balance equations (3) that:

$$\begin{aligned}\sum_{j=1}^{J}\beta_{S_j}Q_{S_j}(\mathfrak{s}) &= \pi(\mathfrak{s})\sum_{j:\mathcal{U}\{S_1,\ldots,S_j\}\neq\emptyset}\left(\alpha_{\mathcal{U}\{S_1,\ldots,S_{j-1},S_j\}}-\alpha_{\mathcal{U}\{S_1,\ldots,S_{j-1}\}}\right)\\ &= \pi(\mathfrak{s})\sum_{j=1}^{J}\left(\alpha_{\mathcal{U}\{S_1,\ldots,S_{j-1},S_j\}}-\alpha_{\mathcal{U}\{S_1,\ldots,S_{j-1}\}}\right)\\ &= \pi(\mathfrak{s})\left(\alpha_{\mathcal{U}\{S_1,\ldots,S_J\}}-\alpha_{\mathcal{U}\{\emptyset\}}\right)\\ &= \pi(\mathfrak{s}),\end{aligned}$$

since clearly $\alpha_{\mathcal{U}\{S_1,\ldots,S_J\}}=1$ and $\alpha_{\mathcal{U}\{\emptyset\}}=0$. The second equality is valid since if $\mathcal{U}\{S_1,\ldots,S_j\}=\emptyset$ then $\alpha_{\mathcal{U}\{S_1,\ldots,S_i\}}=0, i=1,\ldots,j$. This confirms that (6) solves the global balance equations.

If $\alpha_{\mathcal{U}(\mathcal{S})}<\beta_{\mathcal{S}}$, for every non-trivial subset of servers $\mathcal{S}$, the solution of the balance equations converges. This implies, by Theorem 1 in [5], that the Markov process Z_N is ergodic and its stationary distribution is obtained by normalization of the solution (6). Hence, the condition $\alpha_{\mathcal{U}(\mathcal{S})}<\beta_{\mathcal{S}}$, for every non-trivial subset of servers $\mathcal{S}$, is sufficient for ergodicity of Z_N. Using $\alpha_{\mathcal{U}(\mathcal{S})}=\alpha_{\overline{\mathcal{C}}(\overline{\mathcal{S}})}=1-\alpha_{\mathcal{C}(\overline{\mathcal{S}})}$ and $\beta_{\mathcal{S}}=1-\beta_{\overline{\mathcal{S}}}$, we see that this condition is equivalent to condition (2). So (2) is sufficient for ergodicity, and Busic, Gupta and Mairess [2] have shown that it is also necessary.

To finally derive the normalizing constant B the sum of the terms (6) over all states in $\mathfrak{S}$ is set to 1. For a single permutation $S_1,\ldots,S_J$ the number of unmatched customers n_i between S_i and S_{i+1} can take any value in $\mathbb{Z}_+$ if $\mathcal{U}(\{S_1,\ldots,S_i\})\neq\emptyset$, and otherwise, if $\mathcal{U}(\{S_1,\ldots,S_i\})=\emptyset$ only $n_i=0$ is feasible (and also $\alpha_{\mathcal{U}(\{S_1,\ldots,S_i\})}=0$). Hence, taking the sum over all feasible values of $n_1,\ldots,n_{J-1}$ for permutation $S_1,\ldots,S_J$ yields

$$\pi(S_1,\cdot,\ldots,\cdot,S_J)=\frac{B}{(\beta_{\{S_1\}}-\alpha_{\mathcal{U}\{S_1\}})(\beta_{\{S_1,S_2\}}-\alpha_{\mathcal{U}\{S_1,S_2\}})\cdots(\beta_{\{S_1,\ldots,S_{J-1}\}}-\alpha_{\mathcal{U}\{S_1,\ldots,S_{J-1}\}})}.$$

The normalizing constant B readily follows by adding $\pi(S_1,\cdot,\ldots,\cdot,S_J)$ over all permutations of $\{s_1,\ldots,s_J\}$. This completes the proof. □

3 The matching rates

We now calculate the matching rate between server type s_j and customer type c_i, where $c_i\in\mathcal{C}(s_j)$. We will first calculate the probability of a (c_i,s_j) match, conditional on server s^N being of type s_j and on the system Z_N being in state $\mathfrak{s}=(S_1,n_1,S_2,\ldots,n_{J-1},S_J)\in\mathfrak{S}$. We denote this as $r_{c_i,s_j}(S_1,n_1,S_2,\ldots,n_{J-1},S_J)$.

For convenience we define, relative to the permutation $S_1,S_2,\ldots,S_J$,

$$\alpha_{(k)}=\alpha_{U\{S_1,\ldots,S_k\}},\qquad \beta_{(k)}=\beta_{\{S_1,\ldots,S_k\}}=\beta_{S_1}+\cdots+\beta_{S_k},$$

and $\mathfrak{S}_{(k)}$ as the set of feasible values for n_k, so $\mathfrak{S}_{(k)}=\mathbb{Z}_+$ if $\mathcal{U}(\{S_1,\ldots,S_k\})\neq\emptyset$, and $\mathfrak{S}_{(k)}=\{0\}$ otherwise, where $U\{S_1,\ldots,S_k\}$ are the customer types which can be served only by server types $\{S_1,\ldots,S_k\}$. Note that $\alpha_{(k)}=0$ when $\mathfrak{S}_{(k)}=\{0\}$. Further, let

$$\phi_k=\frac{\alpha_{U\{S_1,\ldots,S_k\}\cap\{c_i\}}}{\alpha_{U\{S_1,\ldots,S_k\}}},\qquad \psi_k=\frac{\alpha_{U\{S_1,\ldots,S_k\}\cap(C(s_j)\setminus\{c_i\})}}{\alpha_{U\{S_1,\ldots,S_k\}}},\qquad \chi_k=1-\phi_k-\psi_k\ .$$

Here ϕ_k is the probability that a customer in the list of n_k unmatched customers between S_k and S_{k+1} will be type c_i and hence allow a (c_i,s_j) match. ψ_k is the probability that such a customer

will be of a type different from c_i and will allow a match with s_j. χ_k is the probability that such a customer is incompatible with s_j. In other words, ϕ_k, ψ_k, χ_k are the probabilities that when server s^N of type s_j examines one of the unmatched customers between S_k and S_{k+1}, this will result in an immediate (c_i, s_j) match, or in an immediate match with a customer of a type different from c_i, or in a continuation of the search for a match among the following customers. Note that either one or both of ϕ_k, ψ_k may be zero, or that one of them may be 1. In particular, $\phi_k = \psi_k = 1 - \chi_k = 0$ when $\mathcal{U}\{S_1, \ldots, S_k\} = \emptyset$, and $\phi_J = \alpha_{c_i}$ and $\psi_J = \alpha_{C(s_j)\setminus\{c_i\}}$.

We have:

$$
\begin{aligned}
&r_{c_i,s_j}(S_1, n_1, S_2, \ldots, n_{J-1}, S_J) = \\
&\quad (1-\chi_1^{n_1})\,\frac{\phi_1}{\phi_1+\psi_1} + \chi_1^{n_1}\left[(1-\chi_2^{n_2})\,\frac{\phi_2}{\phi_2+\psi_2} + \chi_2^{n_2}\left[(1-\chi_3^{n_3})\,\frac{\phi_3}{\phi_3+\psi_3} + \cdots\right.\right. \\
&\qquad \left.\left.\chi_{J-2}^{n_{J-2}}\left[\left(1-\chi_{J-1}^{n_{J-1}}\right)\frac{\phi_{J-1}}{\phi_{J-1}+\psi_{J-1}} + \chi_{J-1}^{n_{J-1}}\left[\frac{\phi_J}{\phi_J+\psi_J}\right]\right]\cdots\right]\right],
\end{aligned}
$$

where it is understood that if ϕ_k, ψ_k are both zero, then $(1-\chi_k^n)\,\frac{\phi_k}{\phi_k+\psi_k} = 0$ for all $n \geq 0$.

We next calculate the probability of a (c_i, s_j) match conditional on server s^N being of type s_j, and on the event $Z_N = \mathfrak{s} \in \{(S_1, n_1 \in \mathfrak{S}_{(1)}, S_2, n_2 \in \mathfrak{S}_{(2)}, \ldots, S_{J-1}, n_{J-1} \in \mathfrak{S}_{(J-1)}, S_J)\}$, i.e the permutation of server types is $S_1, \ldots, S_J$, with an arbitrary and feasible number of leftover unmatched customers between them. We denote this as $r_{c_i,s_j}(S_1, S_2, \ldots, S_J)$

Conditional on the permutation, using our convenient notation,

$$
\pi(n_1, \ldots, n_{J-1} \,|\, S_1, \ldots, S_J) \quad = \quad \prod_{k=1}^{J-1}\left(1-\frac{\alpha_{(k)}}{\beta_{(k)}}\right)\left(\frac{\alpha_{(k)}}{\beta_{(k)}}\right)^{n_k},
$$

we get by performing the summations:

$$
\begin{aligned}
&r_{c_i,s_j}(S_1, \ldots, S_J) = \sum_{n_1 \in \mathfrak{S}_{(1)}, \ldots, n_{J-1} \in \mathfrak{S}_{(J-1)}} \pi(n_1, \ldots, n_{J-1} \,|\, S_1, \ldots, S_J)\, r_{c_i,s_j}(S_1, n_1, \ldots, n_{J-1}, S_J) \\
&= \sum_{n_1 \in \mathfrak{S}_{(1)}, \ldots, n_{J-1} \in \mathfrak{S}_{(J-1)}} \prod_{k=1}^{J-1}\left(1-\frac{\alpha_{(k)}}{\beta_{(k)}}\right)\left(\frac{\alpha_{(k)}}{\beta_{(k)}}\right)^{n_k} \\
&\quad \left\{(1-\chi_1^{n_1})\,\frac{\phi_1}{\phi_1+\psi_1} + \chi_1^{n_1}\left[(1-\chi_2^{n_2})\,\frac{\phi_2}{\phi_2+\psi_2} + \chi_2^{n_2}\left[(1-\chi_3^{n_3})\,\frac{\phi_3}{\phi_3+\psi_3} + \cdots\right.\right.\right. \\
&\qquad \left.\left.\left.\chi_{J-2}^{n_{J-2}}\left[\left(1-\chi_{J-1}^{n_{J-1}}\right)\frac{\phi_{J-1}}{\phi_{J-1}+\psi_{J-1}} + \chi_{J-1}^{n_{J-1}}\left[\frac{\phi_J}{\phi_J+\psi_J}\right]\right]\cdots\right]\right]\right\} \\
&= \sum_{n_1 \in \mathfrak{S}_{(1)}}\left(1-\frac{\alpha_{(1)}}{\beta_{(1)}}\right)\left(\frac{\alpha_{(1)}}{\beta_{(1)}}\right)^{n_1}\left\{(1-\chi_1^{n_1})\,\frac{\phi_1}{\phi_1+\psi_1}\right. \\
&\quad + \chi_1^{n_1}\sum_{n_2 \in \mathfrak{S}_{(2)}}\left(1-\frac{\alpha_{(2)}}{\beta_{(2)}}\right)\left(\frac{\alpha_{(2)}}{\beta_{(2)}}\right)^{n_2}\left\{(1-\chi_2^{n_2})\,\frac{\phi_2}{\phi_2+\psi_2}\right. \\
&\qquad + \quad \ddots \\
&\qquad\quad + \chi_{J-2}^{n_{J-2}}\sum_{n_{J-1} \in \mathfrak{S}_{(J-1)}}\left(1-\frac{\alpha_{(J-1)}}{\beta_{(J-1)}}\right)\left(\frac{\alpha_{(J-1)}}{\beta_{(J-1)}}\right)^{n_{J-1}}\left\{\left(1-\chi_{J-1}^{n_{J-1}}\right)\frac{\phi_{J-1}}{\phi_{J-1}+\psi_{J-1}}\right. \\
&\qquad\qquad \left.\left. + \chi_{J-1}^{n_{J-1}}\left[\frac{\phi_J}{\phi_J+\psi_J}\right]\right\}\cdots\right\}\Bigg\}
\end{aligned}
$$

$$= \sum_{k=1}^{J-1} \frac{\phi_k}{\phi_k + \psi_k} \frac{\alpha_{(k)}(1-\chi_k)}{\beta_{(k)} - \alpha_{(k)}\chi_k} \prod_{l=1}^{k-1} \frac{\beta_{(l)} - \alpha_{(l)}}{\beta_{(l)} - \alpha_{(l)}\chi_l} + \frac{\phi_J}{\phi_J + \psi_J} \prod_{l=1}^{J-1} \frac{\beta_{(l)} - \alpha_{(l)}}{\beta_{(l)} - \alpha_{(l)}\chi_l}$$
$$= \sum_{k=1}^{J-1} \phi_k \frac{\alpha_{(k)}}{\beta_{(k)} - \alpha_{(k)}\chi_k} \prod_{l=1}^{k-1} \frac{\beta_{(l)} - \alpha_{(l)}}{\beta_{(l)} - \alpha_{(l)}\chi_l} + \frac{\phi_J}{\phi_J + \psi_J} \prod_{l=1}^{J-1} \frac{\beta_{(l)} - \alpha_{(l)}}{\beta_{(l)} - \alpha_{(l)}\chi_l}.$$

Finally we need multiply each of these $r_{c_i,s_j}(S_1,\ldots,S_J)$ by β_j and by the probability of the permutation, which is:

$$\pi(S_1,\ldots,S_J) = B \prod_{k=1}^{J-1} (\beta_{(k)} - \alpha_{(k)})^{-1}$$

and then add up over all the permutations, to get:

Theorem 3. *For each pair (c_i, s_j), the matching rate r_{c_i,s_j} is given by*

$$\begin{aligned} r_{c_i,s_j} &= \beta_{s_j} \sum_{\mathcal{P}_J} B \prod_{k=1}^{J-1} (\beta_{(k)} - \alpha_{(k)})^{-1} \\ &\quad \left(\sum_{k=1}^{J-1} \phi_k \frac{\alpha_{(k)}}{\beta_{(k)} - \alpha_{(k)}\chi_k} \prod_{l=1}^{k-1} \frac{\beta_{(l)} - \alpha_{(l)}}{\beta_{(l)} - \alpha_{(l)}\chi_l} + \frac{\phi_J}{\phi_J + \psi_J} \prod_{l=1}^{J-1} \frac{\beta_{(l)} - \alpha_{(l)}}{\beta_{(l)} - \alpha_{(l)}\chi_l} \right). \end{aligned} \tag{7}$$

Note that inside the parentheses of (7) each term in the summation over k is the probability of a match in the kth interval between the servers S_k, S_{k+1}, and the last term is the probability that the match occurs in the infinite remainder of the sequence of customers.

4 Calculating the matching rates

We now give some examples and demonstrate calculation of the matching rates. For some special system graphs it is possible to derive the matching rates quite easily.

If the graph G is complete, i.e. all customer types are compatible with all server type, then c^N will be matched to S^N for all N, and $r_{c_i,s_j} = \alpha_{c_i}\beta_{c_j}$. This result is due to Talreja and Whitt [8].

Another tractable example are the almost complete graphs. In these graphs every server type is connected to all customer types except at most one, and every customer type is connected to all server types except at most one. Without loss of generality we assume that each customer and server node in G has exactly one missing arc. This is without loss of generality, since if server type s' is connected to all customer types then we can add a fictitious customer type c' with $\alpha_{c'} = 0$, which is connected to all except s'. We then have an equal number of customer and server types. We label the server types as $s_1,\ldots,s_J$, and the customer types as $c_1,\ldots,c_J$ where s_j is incompatible with c_j, $j = 1,\ldots,J$. The matching rates for this case were derived by Caldentey, Kaplan and Weiss [4], and are given by:

$$r_{c_i,s_j} = \alpha_{c_i}\beta_{s_j} \frac{\big((1-\alpha_{c_i})(1-\beta_{s_j}) - \alpha_{c_j}\beta_{s_i}\big)}{(1-\alpha_{c_i}-\beta_{s_i})(1-\alpha_{c_j}-\beta_{s_j})} \pi(\emptyset), \tag{8}$$

where $\pi(\emptyset)$ is the probability that $n_1 = \cdots = n_{J-1} = 0$, given by:

$$\pi(\emptyset) = \sum_{\mathcal{P}_J} B \prod_{j=1}^{J-1} \frac{1}{\beta_{\{S_1,\ldots,S_j\}}} = \sum_{\mathcal{P}_J} B \prod_{j=1}^{J} \frac{1}{\beta_{\{S_1,\ldots,S_j\}}} = \frac{B}{\beta_{s_1}\cdots\beta_{s_J}}.$$

To obtain (8) from (7), note that for the almost complete graph $\mathcal{U}(S_1,\ldots,S_k)=\emptyset$, $k<J-1$, so that $\alpha_{(k)}=0$, $k<J-1$, and a matching of s_j with c_i can happen only in states with (i) no unmatched customers, i.e. $n_1=\cdots=n_{J-1}=0$, (ii) $n_{J-1}>0$ and $S_J=s_j$ and (iii) $n_{J-1}>0$ and $S_J=s_i$. The calculation then is straightforward, by using the identity

$$\sum_{\mathcal{P}_k}\prod_{l=1}^{k}\frac{1}{\beta_{s_1}+\cdots+\beta_{s_l}}=\frac{1}{\beta_{s_1}\cdots\beta_{s_k}},$$

where $\mathcal{P}_k$ denotes the set of all permutations of $s_1,\ldots,s_k$. This identity is verified by induction.

We have used (7) also to derive the matching rates for complete minus two graphs, in which each customer type is connected to all but two of the server types, and each server type is connected to all but two of the customer types. In this case $\mathcal{U}(S_1,\ldots,S_k)=\emptyset$, $k<J-2$, so in the formula (7) one needs to sum over only two terms inside the parenthesis. Nevertheless the formulas quickly become very lengthy and unilluminating, and seem to offer no advantage over the general expression (7).

If the graph G is a tree, with no loops, one can derive the matching rates directly from the complete resource pooling linear equations (1), which in that case have a unique solution. The condition for stability is then that the solution is all positive. This result is also due to Talreja and Whitt [8].

In general formula (7) gives explicit expressions for the matching rates. However, it is not an easy formula to calculate, as it requires for every pair (c_i,s_j) the calculation of several quantities separately for every permutation of $s_1,\ldots,s_J$. It is not obvious that any short cuts could be used to reduce the computational complexity, since to obtain r_{c_i,s_j} the formula requires addition of non-negative terms for each permutation. Recall that calculation of the permanent of an $n\times n$ matrix, which requires addition of non-negative terms for each of the $n!$ permutations is known to be $\sharp P$. We are aware of some efforts to represent the matching rates as solutions to some optimization problem. Such a method could present an attractive alternative to the direct use of our formula (7).

We have programmed the formula (7), and we conclude this section on computing matching rates by presenting one numerical example. It is for a system with 6 types of customers and 6 types of servers, where each node in the bipartite graph G is of degree 3, every type is connected to exactly 3, and incompatible with 3, see Figure 4. We have used:

$$\alpha=(.04,\ .25,\ .06,\ .27,\ .08,\ .30)$$

$$\beta=(.14,\ .15,\ .16,\ .17,\ .18,\ .20)$$

The following Table 1 gives the matching rates as calculated from formula (7). We have also

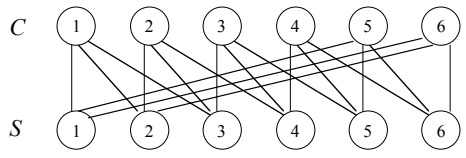

Figure 4: A 3 connected bipartite graph with 6 customer and 6 types

simulated the system, running 100 realizations of $\approx 10,000$ customer/server pairs each. To obtain better estimates from the simulation we continued each run beyond 10,000 until we reached a

state $(S_1, 0, \ldots, 0, S_J)$ so that all customers and servers were matched. Note that each of these states is a regeneration point of the Markov chain, and so our simulation did not require warm-up and is unbiased. We give in the table approximate 95% confidence interval for the matching rates (mean of the hundred $\pm$ 2 standard deviations). Note that all 18 values with the exception of r_{c_4,s_4} are within the confidence interval given by the simulation — this is just as a sanity check.

Two algorithms which were proposed to calculate the matching rates are discussed in [4]: the algoirthm of Caldentey and Kaplan [3], and the quasi independent algorithm. It is found there that they do not always give the correct values. We note that they do not give correct values of the matching rates for this example.

5 Relation to the manufacturing system

Visschers et al [9, 10] consider the following queueing model to describe a manufacturing system. There are jobs of types $\{c_1, \ldots, c_I\}$ and a total of J machines $\{s_1, \ldots, s_J\}$, where job of type c_i can be processed by machine s_j if $(c_i, s_j) \in G$. Customers arrive in independent Poisson streams of rates λ_{c_i}, $i = 1, \ldots, I$, the processing times of jobs by machine s_j are independent and exponentially distributed with rate μ_{s_j}, $j = 1, \ldots, J$. Service discipline is first come first served, so that when machine s_j finishes processing of a job it will take the longest waiting job in the system which it can serve. Arriving jobs of type c_i will join the end of the queue if they find no available idle machine. An arriving job of type c_i which will find one or more idle machines that can serve him will go into service immediately at one of the machines. The choice of machine is random according to an assignment probability distribution, where $P(c_i, s_j|\{S_1, \ldots, S_i\})$ is the probability that a job of type c_i is assigned to the idle machine s_j which can serve it, when the set of busy machines is $\{S_1, \ldots, S_i\}$. These assignment probability distributions determine assignment rates: $\lambda_{S_j}(\{S_1, \ldots, S_i\})$ is the rate at which idle machine S_j is activated when $\{S_1, \ldots, S_i\}$ is the set of busy machines.

The state of the manufacturing system is given as $\tilde{\mathfrak{s}} = (S_1, n_1, \ldots, S_i, n_i)$, where there are a total of $i + n_1 + \cdots + n_i$ jobs in the system, i of which are being processed, where machine S_k is serving the $k + n_1 + \cdots + n_{k-1}$-th job in the queue, for $k = 1, \ldots, i$, with $n_1, \ldots, n_{i-1}$ jobs waiting between the machines, and n_i jobs waiting after the last machine. The remaining $J - i$ machines are idle.

Visschers et al. [9, 10] using the results of [1], show that there exist unique assignment rates $\lambda_{S_j}(\{S_1, \ldots, S_i\})$ with the property that for any subset of machines the product

$$\lambda_{S_1}(\emptyset)\lambda_{S_2}(\{S_1\}) \cdots \lambda_{S_i}(\{S_1, \ldots, S_{i-1}\})$$

is independent of the permutation of $\{S_1, \ldots, S_i\}$, and there exist assignment probability distributions which achieve these assignment rates. Furthermore, by employing partial balance arguments (that directly lead to a candidate product form solution), they show that these assignment rates dictate a product form stationary distribution of the system:

$$\tilde{\pi}(\tilde{\mathfrak{s}}) = \tilde{B} \frac{\lambda_{S_1}(\emptyset)\lambda_{S_2}(\{S_1\}) \cdots \lambda_{S_i}(\{S_1, \ldots, S_{i-1}\})}{\mu_{\{S_1\}}\mu_{\{S_1,S_2\}} \cdots \mu_{\{S_1,\ldots,S_i\}}} \prod_{j=1}^{i} \left(\frac{\lambda_{\mathcal{U}\{S_1,\ldots,S_j\}}}{\mu_{\{S_1,\ldots,S_j\}}} \right)^{n_j},$$

where $\tilde{B}$ is a normalizing constant and $\lambda_{\mathcal{C}} = \sum_{c \in \mathcal{C}} \lambda_c$, $\mu_{\mathcal{S}} = \sum_{s \in \mathcal{S}} \mu_s$. The system is stable if and only if $\lambda_{U\{S_1,\ldots,S_j\}} < \mu_{\{S_1,\ldots,S_j\}}$ for $j = 1, \ldots, J$ and every permutation of machines $S_1, \ldots, S_J$.

This manufacturing system is obviously very similar to our matching model, if we replace the arrival and processing rates λ, μ with the probabilities α, β. Assume now that the total traffic intensity of the manufacturing system approaches 1. Then machines are busy almost all

	1	2	3	4	5	6
1	0.014465 0.01455 ± 0.00025	0.100031 0.09998 ± 0.00066	0.0255038 0.02536 ± 0.00034	0. 0.	0. 0.	0. 0.
2	0. 0.	0.060297 0.06036 ±0.00055	0.0123971 0.01253 ±0.00023	0.077306 0.07689 ± 0.00061	0. 0.	0. 0.
3	0. 0.	0. 0.	0.0220991 0.02198 ± 0.00030	0.109884 0.10963 ± 0.00061	0.0280165 0.02788 ± 0.00040	0. 0.
4	0. 0.	0. 0.	0. 0.	0.0828097 0.08363 ± 0.00068	0.0159407 0.01613 ± 0.00026	0.0712496 0.07095 ± 0.00071
5	0.0157665 0.01583 ± 0.00027	0. 0.	0. 0.	0. 0.	0.0360428 0.03606 ± 0.00040	0.128191 0.12813 ± 0.00062
6	0.00976846 0.00967 ± 0.00019	0.0896718 0.09000 ± 0.00075	0. 0.	0. 0.	0. 0.	0.10056 0.10042 ± 0.00076

Table 1: Matching rates for a 6×6, 3 connected example. Shown are the exact rates, and simulation results with 95% confidence intervals.

the time. This implies that machines become available at times given by independent Poisson processes of rates μ_{s_j}. Furthermore, arriving jobs will almost never find idle servers, so they will almost always join the queue. In particular the assignment probability distributions will become almost irrelevant. We now provide an alternative proof to Theorem 2, based on the stationary distribution of the manufacturing system of [9, 10], when the total traffic intensity in the system approaches 1.

Alternative proof of Theorem 2. We note that the state space and transitions of Z_N are exactly those of the multi-type customer and multi-type server queueing system of [9, 10], when we condition on all servers being busy, and aggregate over the number of customers queued up behind the last machine.

In the manufacturing system of [9, 10], if all machines are busy the state is given by: $\tilde{\mathfrak{s}} = (S_1, n_1, S_2, n_2, \ldots, S_{J-1}, n_{J-1}, S_J, n_J)$, where $S_1, \ldots, S_J$ is a permutation of the machines, and n_j the numbers queued between machine S_j and S_{j+1} for $1 \leq j < J$, and n_J is the number queued up behind the last machine.

We start from the stationary probabilities of the system in [9, 10] and go through several steps to reach our result. We explain notation and steps following this derivation:

$$\begin{aligned}
\tilde{\pi}(\tilde{\mathfrak{s}}) &= \tilde{B}\frac{\lambda_{S_1}(\emptyset)\lambda_{S_2}(\{S_1\})\cdots\lambda_{S_J}(\{S_1,\ldots,S_{J-1}\})}{\mu_{\{S_1\}}\mu_{\{S_1,S_2\}}\cdots\mu_{\{S_1,\ldots,S_J\}}}\prod_{j=1}^{J}\left(\frac{\lambda_{U\{S_1,\ldots,S_j\}}}{\mu_{\{S_1,\ldots,S_j\}}}\right)^{n_j} \\
&= \tilde{B}\rho^{J+n_J}\,\frac{\alpha_{S_1}(\emptyset)\alpha_{S_2}(\{S_1\})\cdots\alpha_{S_J}(\{S_1,\ldots,S_{J-1}\})}{\beta_{\{S_1\}}\beta_{\{S_1,S_2\}}\cdots\beta_{\{S_1,\ldots,S_{J-1}\}}}\prod_{j=1}^{J-1}\left(\frac{\rho\alpha_{U\{S_1,\ldots,S_j\}}}{\beta_{\{S_1,\ldots,S_j\}}}\right)^{n_j} \\
&= \tilde{B}\rho^{J+n_J}\,\alpha_{S_1}(\emptyset)\alpha_{S_2}(\{S_1\})\cdots\alpha_{S_J}(\{S_1,\ldots,S_{J-1}\})\prod_{j=1}^{J-1}\left(\frac{1}{\beta_{\{S_1,\ldots,S_j\}}}\right)\left(\frac{\rho\alpha_{U\{S_1,\ldots,S_j\}}}{\beta_{\{S_1,\ldots,S_j\}}}\right)^{n_j} \\
&= \tilde{B}\rho^{J+n_J}\,\Psi\prod_{j=1}^{J-1}\left(\frac{1}{\beta_{\{S_1,\ldots,S_j\}}}\right)\left(\frac{\rho\alpha_{U\{S_1,\ldots,S_j\}}}{\beta_{\{S_1,\ldots,S_j\}}}\right)^{n_j}.
\end{aligned}$$

The first expression is taken from [10], where $\lambda_{s_j}(\mathcal{S})$ is defined as the rate at which server s_j is activated when $\mathcal{S}$ is the subset of busy servers. We then define $\lambda = \sum_{i=1}^{I}\lambda_{c_i}$, $\mu = \sum_{j=1}^{J}\mu_{s_j}$, $\rho = \frac{\lambda}{\mu}$, and let: $\alpha_{c_i} = \frac{\lambda_{c_i}}{\lambda}$, $\beta_{s_j} = \frac{\mu_{s_j}}{\mu}$. We divide each term in the numerator by λ and each term in the denominator by μ, to obtain the second equality, where $\alpha_{S_j}(\{S_1,\ldots,S_i\}) = \lambda_{S_j}(\{S_1,\ldots,S_i\})/\lambda$. Note that $\alpha_{U\{S_1,\ldots,S_J\}} = \beta_{\{S_1,\ldots,S_J\}} = 1$, so we can drop them from the product, which now goes from 1 to $J-1$. The third equality is straight forward. We now note that, as is required in [10], the product $\lambda_{S_1}(\emptyset)\lambda_{S_2}(\{S_1\})\cdots\lambda_{S_J}(\{S_1,\ldots,S_{J-1}\})$ is the same for all the permutations of $\{s_1,\ldots,s_J\}$, and define the constant value

$$\Psi = \frac{\lambda_{S_1}(\emptyset)\lambda_{S_2}(\{S_1\})\cdots\lambda_{S_J}(\{S_1,\ldots,S_{J-1}\})}{\lambda^J} = \alpha_{S_1}(\emptyset)\alpha_{S_2}(\{S_1\})\cdots\alpha_{S_J}(\{S_1,\ldots,S_{J-1}\})$$

which is the last equality.

Note that n_J appears only in the exponent of ρ. Summing over $n_J = 0, 1, \ldots$ we obtain the marginal stationary probabilities:

$$\tilde{\pi}(\mathfrak{s},\rho) = \tilde{B}\frac{\rho^J}{1-\rho}\,\Psi\prod_{j=1}^{J-1}\left(\frac{1}{\beta\{S_1,\ldots,S_j\}}\right)\left(\frac{\rho\alpha_{U\{S_1,\ldots,S_j\}}}{\beta\{S_1,\ldots,S_j\}}\right)^{n_j}.$$

Hence the conditional probabilities given that all servers are busy are:

$$\pi(\mathfrak{s},\rho) = B(\rho) \prod_{j=1}^{J-1} \left(\frac{1}{\beta\{S_1,\ldots,S_j\}} \right) \left(\frac{\rho\alpha_{U\{S_1,\ldots,S_j\}}}{\beta_{\{S_1,\ldots,S_j\}}} \right)^{n_j}.$$

with the new normalizing constant

$$B(\rho)^{-1} = \sum_{\mathcal{P}_J} \frac{1}{(\beta_{\{S_1\}} - \rho\alpha_{\mathcal{U}\{S_1\}})(\beta_{\{S_1,S_2\}} - \rho\alpha_{\mathcal{U}\{S_1,S_2\}}) \cdots (\beta_{\{S_1,\ldots,S_{J-1}\}} - \rho\alpha_{\mathcal{U}\{S_1,\ldots,S_{J-1}\}})}.$$

Substituting $\rho = 1$ we get the result (6). □

6 Conjectured matching rates for a call center system

Talreja and Whitt [8] considered a queueing system which provides a model for call centers with skill based routing. The model is similar to the manufacturing model of Section 5. Customers of types c_i, $i = 1,\ldots,I$ arrive as independent ergodic point processes (not necessarily Poisson) with rates λ_{c_i}. They are served by pools of servers of various types s_j, $j = 1,\ldots,J$, with M_{s_j} servers of each type, that have i.i.d service times distributed as G_{s_j} (not necessarily exponential), so that the service capacity of the whole type s_j pool is at total rate μ_{s_j}. Server of type s_j can serve customer of type c_i if $(c_i, s_j) \in G$. Service discipline is first come first served. The added feature here is that the system is overloaded so that there is not enough service capacity to serve all the customers, and customers of type c_i have patience distribution F_{c_i}, so that customers abandon the queue without service if their patience limit is reached.

Talreja and Whitt consider this system under many server heavy traffic scaling (uniform accelaration), where one thinks of a sequence of systems in which for system n the arrival rates and the number of servers are scaled up by a factor of n, and the queue lengths are then rescaled through division by n. Since the system is overloaded servers will be busy almost all the time, and queues of customers of all types will be non-empty almost all the time. Also, two consecutive customers will have arrived almost at the same time, and when a server becomes available for one of them, a server will become available for the next one (if resource pooling holds) almost immediately, irrespective of their types, and it is conjectured in [8] that global first come first served occurs on a fluid scale, so that all customers which do not abandon get served after a global waiting time of W. Assuming patience distributions are absolutely continuous W is uniquely determined by

$$\sum_{i=1}^{I} \lambda_{c_i}(1 - F_{c_i}(W)) = \sum_{j=1}^{J} \mu_{s_j}. \tag{9}$$

It is then possible to write down the following equations for the matching rates ν_{c_i,s_j}:

$$\begin{aligned} \sum_{c_i \in \mathcal{C}(s_j)} \nu_{c_i,s_j} &= \mu_{s_j}, && \text{for all } s_j, \\ \sum_{s_j \in \mathcal{S}(c_i)} \nu_{c_i,s_j} &= \lambda_{c_i}(1 - F_{c_i}(W)), && \text{for all } c_i. \end{aligned}$$

These are again *the total resource pooling linear equations* for this system.

Talreja and Whitt obtain the matching rates for systems where G is a tree, or when G is complete, and for hybrid cases of these. For the complete graph case they prove convergence of the stochastic system to these rates. No further results exist for this model so far.

Consider now this model, and assume that arrivals are Poisson and processing times are exponential, let $\mu = \sum \mu_{s_j}$, $\lambda = \sum \lambda_{c_i}$, and let $\rho = \lambda/\mu$. If we let $\rho \searrow 1$ the solution of (9) will have $W = 0$, and the total resource pooling linear equations are identical to our model. It seems that just as the infinite matching model corresponds to the manufacturing system of Section 5 when $\rho \nearrow 1$ so it can also correspond to the overloaded system with abandonments, as $\rho \searrow 1$.

For $\rho \geq 1$, under many server heavy traffic scaling (uniform acceleration), the following limiting behavior appears plausible: Most customers will wait a time which is close to W before being served. When a server will look at the queue he will therefore encounter enough customers of the various types which have all been waiting approximately W, and are now close to the head of the queue. He will choose the first one of those which he can serve. Assume now that arrivals are Poisson, which is a reasonable assumption for a high arrival rate call center. This implies that customers which get served are i.i.d. of type c_i with probability $\alpha_{c_i} = \lambda_{c_i}(1 - F_i(W))/\lambda$. Also, since servers are busy almost all the time, with many servers this results in servers of type s_j becoming available as independent Poisson streams with rates μ_{s_j}, so consecutive servers will be i.i.d of type s_j with probability $\beta_{s_j} = \mu_{s_j}/\mu$. We formulate the following conjecture:

Conjecture 1. *In the model of Talreja and Whitt [8], under uniform acceleration (many server heavy traffic scaling), the matching rates are given by the formula (7).*

References

[1] Adan, I.J.B.F., Hurkens, C.A.J., Weiss, G. (2010) A reversible multi-class multi-server loss system. *Probability in Engineering and Informational Sciences*. To appear.

[2] Busic, A., Gupta, V. and Mairesse, J. (2010) Stability of the bipartite matching model, Preprint, arxiv:1003.3477v1 [cs.DM].

[3] Caldentey, R.A. and Kaplan, E.H. (2002) A Heavy Traffic Approximation for Queues with Restricted Customer-Service Matchings. Unpublished manuscript.

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

[5] Foster, F.G. (1953) On the stochastic matrices associated with certain queuing processes. *Ann. Math. Stat.* 24:355–360.

[6] Kaplan, E.H. (1984) Managing the demand for public housing. ORC technical report # 183, MIT.

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

[8] Talreja, R. and Whitt, W. (2007) Fluid Models for Overloaded Multi-class many-service queueing systems with FCFS routing. *Management Science* 54:1513–1527.

[9] Visschers, J.W.C.H.. (2000) *Random walks with geometric jumps*. Ph.D. Thesis, Eindhoven University of Technology.

[10] Visschers, J.W.C.H., Adan, I.J.B.F., Weiss, G. (2010). A product form solution to a system with multi-type customers and multi-type servers. Preprint.