FCFS Parallel Service Systems and Matching Models

Ivo Adan
Eindhoven University of Technology
iadan@win.tue.nl

Rhonda Righter
University of California at Berkeley
rrighter@berkeley.edu

Gideon Weiss
The University of Haifa
gweiss@stat.haifa.ac.il

ABSTRACT

We consider three parallel service models in which customers of several types are served by several types of servers subject to a bipartite compatibility graph, and the service policy is first come first served. Two of the models have a fixed set of servers. The first is a queueing model in which arriving customers are assigned to the longest idling compatible server if available, or else queue up in a single queue, and servers that become available pick the longest waiting compatible customer, as studied by Adan and Weiss, 2014. The second is a redundancy service model where arriving customers split into copies that queue up at all the compatible servers, and are served in each queue on FCFS basis, and leave the system when the first copy completes service, as studied by Gardner et al., 2016. The third model is a matching queueing model with a random stream of arriving servers. Arriving customers queue in a single queue and arriving servers match with the first compatible customer and leave the system at the moment of arrival, or they leave without a customer. The last model is relevant to organ transplants, to housing assignments, to adoptions and many other situations.

We study the relations between these models, and show that they are closely related to the FCFS infinite bipartite matching model, in which two infinite sequences of customers and servers of several types are matched FCFS according to a bipartite compatibility graph, as studied by Adan et al., 2017.

CCS CONCEPTS

• **Mathematics of computing → Queueing theory**; **Markov processes**;

KEYWORDS

Parallel service; FCFS; redundancy service; matching

ACM Reference Format:
Ivo Adan, Rhonda Righter, and Gideon Weiss. 2017. FCFS Parallel Service Systems and Matching Models. In *VALUETOOLS 2017: 11th EAI International Conference on Performance Evaluation Methodologies and Tools, December 5–7, 2017, Venice, Italy*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3150928.3150951

The work of Gideon Weiss is supported in part by Israel Science Foundation Grant 286.13.

VALUETOOLS 2017, December 5–7, 2017, Venice, Italy

ACM ISBN 978-1-4503-6346-4/17/12…$15.00
https://doi.org/10.1145/3150928.3150951

1 INTRODUCTION

We consider three parallel service models in which customers of several types, indexed by $c_i \in \mathcal{C} = \{c_1, \ldots, c_I\}$ are served by several types of servers, indexed by $s_j \in \mathcal{S} = \{s_1, \ldots, s_J\}$, subject to a bipartite compatibility graph, $\mathcal{G} = (\mathcal{S}, \mathcal{C}, \mathfrak{E})$, $\mathfrak{E} \subseteq \mathcal{S} \times \mathcal{C}$, such that $(s_j, c_i) \in \mathfrak{E}$ if customer type c_i can be served by server type s_j. We focus on first come first served (FCFS) policy in all the models, i.e. customers are prioritized by their order of arrivals, and servers are prioritized by the order in which they become available. Two of the models have a fixed set of servers, while the third model has a random stream of arriving servers. Briefly stated the models are as follows:

- *FCFS-ALIS Queueing Model:* There are J servers of types $\mathcal{S}$ and a stream of customers of types $\mathcal{C}$. An arriving customer is assigned to the longest idle server which is compatible with it (ALIS - assign longest idle server) if such is available, or else he joins the queue of waiting customers. A server that completes a service picks up the longest waiting customer which is compatible with him (FCFS), if such is available, or else he joins the queue of idle servers. This model was studied by Adan and Weiss [3].
- *A Redundancy Service Model:* There are J servers of types $\mathcal{S}$, each with his own FCFS queue, and a stream of arriving customers of types $\mathcal{C}$. An arriving customer splits upon arrival into several copies that join the queues of the servers which are compatible with it. Service of a customer can then proceed simultaneously at several compatible servers. The customer leaves the system when the first of his copies completes service. This model was studied by Gardner et al. [5].
- *A Matching Service Model:* There is an arrival stream of customers of types $\mathcal{C}$, and an independent arrival stream of servers of types $\mathcal{S}$. When a customer arrives he joins a queue of customers waiting for service. When a server arrives he scans the queue of customers and matches with the longest waiting customer that is compatible with his type, and the matched customer then leaves the system with the server. If the server does not find a match he leaves immediately without a match.

The matching queue model is relevant to many types of service systems: It can describe organ transplants, where patients are waiting to receive organs, and donated organs arrive in a random stream, and organs are assigned to compatible recipients in FCFS order, or are lost if no compatible recipient is waiting [9]. It can also describe an adoption process, where families are waiting for available babies to be adopted (this may only be approximate since unmatched babies do not disappear). It was used to model assignment of project houses to families in Boston public housing, by Kaplan [6, 7]. Another application is to call centers with inbound and outbound calls, where differently skilled agents (servers) start outbound calls if there are no waiting inbound calls that match their skill sets. Here the state would be the set of customers waiting in the queue, and would not include those

in service. Our matching queue model, although it seems very relevant to the study of organ transplants and to various other systems, has not to the best of our knowledge, been analyzed in any level of detail.

We assume Poisson arrivals and exponential service times for all three models so that their evolution is Markovian and can be described by various discrete space continuous time Markov chains.

These models are closely related to a fourth model:

- *The FCFS infinite bipartite matching model:* This was introduced in [2, 4] and studied in more detail recently by Adan, Busic, Mairesse and Weiss [1]. In this model there are two infinite sequences, drawn independently, one is drawn i.i.d. from C the other from $\mathcal{S}$, and the two sequences are then matched FCFS according to the compatibility graph $\mathcal{G}$. This model is much simpler than either of the above models since it does not involve arrival times and service times, and servers and customers play a completely symmetric role.

In this paper we explore the relations between the three service models, and their connections to the infinite matching model. Our results here are:

- The continuous time Markov chains that describe all three service models share the same stationary distribution. This leads the way to comparing their performance measures.
- We note that the redundancy service model and the matching queue are equivalent, in that they share the same continuous time Markov chain.
- We conjecture that the average sojourn time in the redundancy service system is shorter than in the FCFS-ALIS system.
- We discuss embedding of the three service models in the infinite matching model.
- We introduce a new discrete infinite matching model, which we call the infinite interleaved matching model, that is similar to the model of [1].
- We derive properties of this new infinite matching model, and obtain from those properties some surprising corollaries on the behavior of the match queue model.

The rest of the paper is structured as follows: In Section 2 we describe the three queueing models. In Section 3 we describe the relevant properties of the infinite matching model. In Section 4 we compare the three service models. In Section 5 we embed the service models in versions of the infinite matching model.

Notation

We let: $\mathcal{S}(c_i)$ denote the subset of server types compatible with c_i, $C(s_j)$ denote the subset of customer types compatible with s_j. For $C \subset \mathcal{C}$, $S \subset \mathcal{S}$ we denote $\mathcal{S}(C) = \bigcup_{c_i \in C} \mathcal{S}(c_i)$, $C(S) = \bigcup_{s_j \in S} C(s_j)$, and denote by $\mathcal{U}(S) = (C(S^c))^c$ those customer types that are compatible only with server types in S.

We associate with c_i a rate λ_{c_i}, and with s_j a rate μ_{s_j}, these are rates for exponential distributions. We also let $\bar{\lambda} = \sum_{i=1}^{I} \lambda_{c_i}$, $\bar{\mu} = \sum_{j=1}^{J} \mu_{s_j}$, and define $\alpha_{c_i} = \lambda_{c_i}/\bar{\lambda}$, $\beta_{s_j} = \mu_{s_j}/\bar{\mu}$. For a subsets $C \subset \mathcal{C}$, $S \subset \mathcal{S}$ we let $\lambda_C = \sum_{c_i \in C} \lambda_{c_i}$, $\mu_S = \sum_{s_j \in S} \mu_{s_j}$, and define α_C, β_S similarly.

In what follows we will denote quantities related to the FCFS-ALIS model by a superscript ^{q}, those related to the Redundancy model by a superscript ^{r}, those related to the Matching model by a superscript ^{m}. Finally, we denote quantities related to the infinite matching model by a superscript $^\infty$.

2 THE SERVICE MODELS

2.1 A stability condition

Theorem 2.1. *All three service models are stable, in the sense that Markov chains describing them are ergodic, if and only if the following condition holds:*

$$\lambda_C < \mu_{S(C)}, \qquad \text{for every } C \subseteq \mathcal{C}. \tag{2.1}$$

Proof. This follows from the form of the stationary distributions, which converge if and only if (2.1) holds. □

Figure 1 illustrates the compatibility graph for an example we will use throughout the paper. In this example there are 3 types of customers and 3 types of servers, customers of type c_2 (type c_3) can only be served by server of type s_2 (type s_3), while customers of type c_1 can be served by all types of servers.

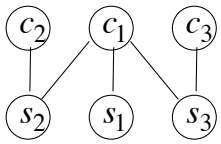

Figure 1: Compatibility graph for customer and server types

The stability condition for this example is:

$$\lambda_2 < \mu_2, \quad \lambda_3 < \mu_3, \quad \bar{\lambda} < \bar{\mu}.$$

2.2 The FCFS-ALIS parallel queueing model

Customers arrive in independent Poisson streams, with rate λ_{c_i} for type c_i. There are J servers of types $\{s_1, \ldots, s_J\}$, and service by server s_j is exponential with rate μ_{s_j}. The service policy as described in the introduction is FCFS-ALIS. Figure 2 illustrates a possible state for our example. In it all customers in the system are

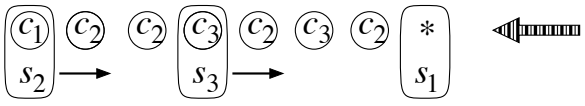

Figure 2: A current state under FCFS-ALIS

displayed in order of arrival, with earlier arrivals more to the left, and customers in service are shown together with their server. The oldest customer in the system is of type c_1 and it is served by server s_2, server s_3 is serving a customer of type c_3 after skipping two incompatible customers of type c_2. Server s_1 is idle. New customers are arriving from the right and on completion of service servers scan waiting customers from left to right.

In [3] the system is described by the process $Y^q(t) = (S_1, n_1, \ldots, S_i, n_i, S_{i+1}, \ldots, S_J)$ where $S_1, \ldots, S_J$ is a permutation of the servers, servers $S_1, \ldots, S_i$ are busy with S_1 serving the oldest customer in the system, S_2 has skipped n_1 customers and is serving the second oldest customer currently in service, and so on. n_j is the number of skipped customers between S_j and S_{j+1}. The remaining

servers, $S_{i+1}, \ldots, S_J$ are idle, ordered by length of time they were idle, with S_J the longest idle. This describes the system at time t. They proved:

Theorem 2.2 (Adan and Weiss [3]). *The process $Y^q(t)$ is a continuous time discrete state Markov chain. It is ergodic if and only if (2.1) holds. Its stationary distribution is given, up to a normalizing constant, by*

$$P^q(S_1, n_1, \ldots, S_i, n_i, S_{i+1}, \ldots, S_J) \propto \prod_{j=1}^{i} \frac{(\lambda_{\mathcal{U}(\{S_1,\ldots,S_j\})})^{n_j}}{(\mu_{\{S_1,\ldots,S_j\}})^{n_j+1}} \times \prod_{j=i+1}^{J} \frac{1}{\lambda_{C(\{S_j,\ldots,S_J\})}} \tag{2.2}$$

Adan and Weiss [3] also calculated the normalizing constant.

We introduce an alternative process to describe the system, $X^q(t) = (c^1, c^2, \ldots, c^L, s^1, \ldots, s^K)$ where c^ℓ is the random type of the ℓth oldest customer in the system that is waiting and has not started service yet, and s^k is the type of the kth longest idling server in the system, all this at time t. Note that L, the number of waiting customers corresponds to $n_1 + \cdots + n_i$ of $Y^q(t)$, and can take any value ≥ 0, while $s^1, \ldots, s^K$ correspond to $S_J, \ldots, S_{i+1}$ of $Y^q(t)$, which are the subset of idle servers, with no replications, so that $K \leq J$. We then have:

Theorem 2.3. *The process $X^q(t)$ is a continuous time discrete state Markov chain. It is ergodic if and only if (2.1) holds. Its stationary distribution is given, up to a normalizing constant, by:*

$$P^q(c^1, c^2, \ldots, c^L, s^1, \ldots, s^K) \propto \prod_{\ell=1}^{L} \frac{\lambda_{c^\ell}}{\mu_{\mathcal{S}(\{c^1,\ldots,c^\ell\})}} \times \prod_{k=1}^{K} \frac{\mu_{s^k}}{\lambda_{C(\{s^1,\ldots,s^k\})}} \tag{2.3}$$

The proof of this theorem is by partial balance, we present it in the appendix. In particular, the following corollary is immediate:

Corollary 2.4. *The process $X^q(t)$ conditional on the event that all servers are busy, has the stationary distribution given up to a normalizing constant by:*

$$P^q(c^1, c^2, \ldots, c^L \mid \textit{all busy}) \propto \prod_{\ell=1}^{L} \frac{\lambda_{c^\ell}}{\mu_{\mathcal{S}(\{c^1,\ldots,c^\ell\})}} \tag{2.4}$$

2.3 The redundancy service model

There are servers $s_1, \ldots, s_J$, and each of them has his own FCFS queue of compatible customers, and service of server s_j is exponential rate μ_{s_j}. Customers arrive in independent Poisson streams, with rate λ_{c_i} for customers of type c_i. Each arriving customer, upon arrival, splits into copies of the same type, and one copy joins the queue of each of the servers with which it is compatible. Service of a customer can then be performed at several compatible servers simultaneously. The customer departs from the system, with all its copies, at the instant at which service of one of its copies is complete.

Figure 3 illustrates a possible state for our example. In it we display the list of customer types, in order of arrival, on the right side, and on the left side are the servers and their queues. The first customer, c^1 is of type c_1, and is currently being served simultaneously be all three servers. The second and third customers are of types c_2 and c_3 and queue up for servers s_2, s_3 respectively. The fourth and sixth customer, c^4, c^6 are again of type c_1 and queue up at all the three servers.

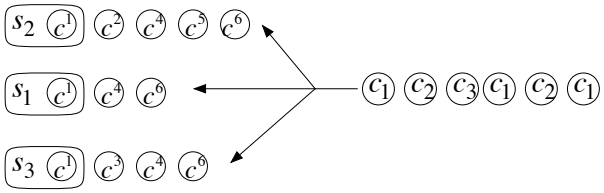

Figure 3: A current state with redundant queueing

Gardner et al. [5] have studied this system and defined the following process to describe it: $X^r(t) = (c^1, \ldots, c^L)$, where $c^1, \ldots, c^L$ are the types of all the customers in the system at time t, ordered by their arrival times, with c^1 the oldest. They have shown:

Theorem 2.5 (Gardner, Zbarsky, Doroudi,t Harchol-Balter, Hyytia and Scheller-Wolf [5]). *The process $X^r(t)$ is a continuous time discrete state Markov chain. It is ergodic if and only if (2.1) holds. Its stationary distribution is given, up to a normalizing constant, by:*

$$P^r(c^1, c^2, \ldots, c^L) \propto \prod_{\ell=1}^{L} \frac{\lambda_{c^\ell}}{\mu_{\mathcal{S}(\{c^1,\ldots,c^\ell\})}} \tag{2.5}$$

2.4 The FCFS parallel matching queue

Customers of various types arrive in independent Poisson streams of rates λ_{c_i} and queue up in order of arrival. Servers of various types arrive in independent Poisson streams of rates μ_{s_j}. An arriving server scans the queue of customers and matches with the longest waiting customer that is compatible with him, and the two leave the system immediately. If the server does not find a compatible customer in the queue he leaves immediately without a customer.

Figure 4 illustrates a possible history of this system, for our example. The figure shows a sequence of customers and servers

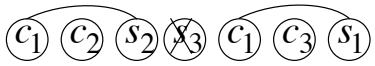

Figure 4: A partial history of the matching queue

specified by their types, ordered in the order of arrival from left to right. Customer of type c_1 arrived first, followed by a customer of type c_2. Next a server of type s_2 arrived and was immediately matched to the first customer and they departed together. Next a server of type s_3 arrived and left immediately without a match. This was followed by a customer of type c_1, then a customer of type c_3 and finally by a server of type s_1 that matched immediately with the third customer, of type c_1, and departed with him. At this point in time there was a queue of two customers, the earlier of type c_2, the later of type c_3.

We describe this system by the process $X^m(t) = (c^1, \ldots, c^L)$ where there are L customers in total, their types (random) are $c^1, \ldots, c^L$, ordered in order of arrival, with c^1 the longest waiting, and the time is t.

Theorem 2.6. *The process $X^m(t)$ is a Markov chain, it is ergodic if and only if (2.1) holds, and its stationary distribution is given up to a constant by:*

$$P^m(c^1, \ldots, c^L) \propto \prod_{\ell=1}^{L} \frac{\lambda_{c^\ell}}{\mu_{\mathcal{S}(\{c^1,\ldots,c^\ell\})}} \tag{2.6}$$

The proof of this theorem is identical to the proof of Theorem 2.5. It also follows directly from Theorem 4.1.

3 THE FCFS INFINITE BIPARTITE MATCHING MODEL

There are two independent doubly infinite series of customers $\ldots, c^{-2}, c^{-1}, c^0, c^1, c^2, \ldots$ drawn i.i.d. from C according to the probabilities α, and of servers $\ldots, s^{-2}, s^{-1}, s^0, s^1, s^2, \ldots$, drawn i.i.d. from $\mathcal{S}$ according to the probabilities β, and they are matched FCFS according to the compatibility graph $\mathcal{G}$.

Definition 3.1. We say that this system has complete resource pooling if the following equivalent conditions hold for any $S \subset \mathcal{S}$, $S \neq \emptyset, \mathcal{S}$ and $C \subset C$, $C \neq \emptyset, C$:

$$\alpha_C < \beta_{\mathcal{S}(C)}, \qquad \beta_S < \alpha_{C(S)}, \qquad \alpha_{\mathcal{U}(S)} < \beta_S. \tag{3.1}$$

What we mean by FCFS is that if s^n is matched with c^m, then there is no earlier $s^k \in \mathcal{S}(c^m)$ which is unmatched, and no earlier $c^\ell \in C(s^n)$ which is unmatched. Figure 5 illustrates FCFS infinite bipartite matching in our example, for a window of the sequences. In this figure one customer and one server were matched to an

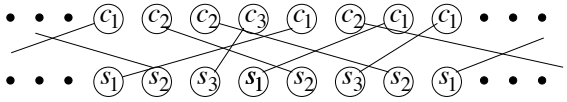

Figure 5: FCFS infinite bipartite matching

earlier (on the left of the window) customer and server, and one customer and one server were left unmatched in the end of the window, and are matched to a later (on the right of the window) customer and server. The following theorem is proved by Adan et al. [1]:

Theorem 3.2 (Adan, Busic, Mairesse and Weiss [1]). *If complete resource pooling (3.1) holds then almost surely there exists a FCFS matching of the two sequences and this matching is unique.*

We define the following transformation on the matched sequences:

Definition 3.3. For given matched sequences, the exchange transformation exchanges the position of each matched pair, so that if s^n was matched to c^m in the original system, then in the exchanged system we have $\tilde{c}^n$ matched to $\tilde{s}^m$. This defines a permutation of the original sequence $\ldots, c^{-2}, c^{-1}, c^0, c^1, c^2, \ldots$ to a new sequence $\ldots, \tilde{c}^{-2}, \tilde{c}^{-1}, \tilde{c}^0, \tilde{c}^1, \tilde{c}^2, \ldots$, and the original sequence $\ldots, s^{-2}, s^{-1}, s^0, s^1, s^2, \ldots$ to a new sequence $\ldots, \tilde{s}^{-2}, \tilde{s}^{-1}, \tilde{s}^0, \tilde{s}^1, \tilde{s}^2, \ldots$.

Figure 6 illustrates the exchanged sequences obtained by the exchange transformation from the illustration in Figure 5.

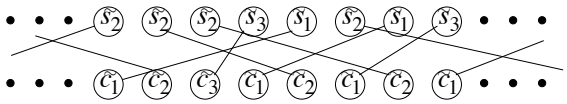

Figure 6: The exchange transformation applied to Figure 5

The following reversibility result is proved in [1]

Theorem 3.4 (Adan, Busic, Mairesse and Weiss [1]). *The exchanged sequences $\ldots, \tilde{c}^{-2}, \tilde{c}^{-1}, \tilde{c}^0, \tilde{c}^1, \tilde{c}^2, \ldots, \ldots, \tilde{s}^{-2}, \tilde{s}^{-1}, \tilde{s}^0, \tilde{s}^1, \tilde{s}^2, \ldots$, are independent and each is i.i.d. from C and from $\mathcal{S}$ according to α, β. Furthermore, the original matching is now the almost surely unique FCFS matching of the exchanged sequences in reversed time.*

Using the reversibility, it is easy to obtain stationary distributions for several Markov chains associated with this system. We consider making all the FCFS matches of s^ℓ, c^k for $k, \ell \leq n$, and define the process $X^\infty(n) = (c^{i_1}, \ldots, c^{i_L}, s^{j_1}, \ldots, s^{j_L})$, where customers in positions $i_1, \ldots, i_L$ and servers in positions $j_1, \ldots, j_L$ were left unmatched, and $c^{i_1}, \ldots, c^{i_L}, s^{j_1}, \ldots, s^{j_L}$ are the types of these unmatched customers and servers.

Theorem 3.5 (Adan, Busic, Mairesse and Weiss [1]). *The process $X^\infty(n)$ is a discrete time discrete state Markov chain. It is ergodic if and only if complete resource pooling (3.1) holds. Its stationary distribution is given, up to a normalizing constant, by:*

$$P^\infty(c^{i_1}, c^{i_2}, \ldots, c^{i_L}, s^{j_1}, \ldots, s^{j_L}) \propto \prod_{\ell=1}^{L} \frac{\alpha_{c^{i_\ell}}}{\beta_{\mathcal{S}(\{c^{i_1}, \ldots, c^{i_\ell}\})}} \times \prod_{\ell=1}^{L} \frac{\beta_{s^{j_\ell}}}{\alpha_{C(\{s^{j_1}, \ldots, s^{j_\ell}\})}} \tag{3.2}$$

4 RELATIONS BETWEEN THE THREE SERVICE MODELS

As we see from Theorems 2.5, 2.6 and Corollary 2.4, all three parallel service systems are associated with a Markov chain with the same stationary distribution. Furthermore this stationary distribution is similar to that of the infinite matching model. We now explore the relations between these models.

4.1 Equivalence of the redundancy service system and the matching queue

Note that although the redundancy system can have idle servers, and the matching queue cannot, the state of the redundancy system is completely determined by the sequence of customers in the system; servers are idle at a given time if there are no compatible customers in the system at that time. We will show that the matching and redundancy queues are sample-path equivalent in the sense that if we start them with the same customer state, and we couple the customer arrival processes in the two queues, and we couple potential service completions in the redundancy queue with service arrivals in the matching queue, then the sample paths for the state processes of the two systems will be identical, with probability one.

Theorem 4.1. *The redundancy service system and the matching queue and equivalent in the sense that the processes $X^r(t)$ and $X^m(t)$ are sample path equivalent. In particular this means that for each type of customer, the sojourn time in the system is the same for both models.*

Proof. Consider the situation at time t where the customers in the system, ordered in order of arrival, are of types $c^1, \ldots, c^L$, in each of the systems. Then if a customer of type c^* arrives he joins the queue as last in both systems, which remain identical. The only other thing that can happen is that one of the customers leaves. In the redundancy service system, the first customer is currently served by all servers in $\mathcal{S}(c^1)$ simultaneously, and will depart at

rate $\mu_{\mathcal{S}(c^1)}$. In the matching model, the first customer will depart if a server of type in $\mathcal{S}(c^1)$ arrives, which happens at rate $\mu_{\mathcal{S}(c^1)}$. Furthermore, in the redundancy service system, the ℓth customer is currently served by all servers in $\mathcal{S}(c^\ell)\backslash\mathcal{S}(\{c^1,\ldots,c^{\ell-1}\})$ simultaneously, if this set is non-empty, otherwise it is not being served. The rate of departure of the ℓth customer is then $\mu_{\mathcal{S}(c^\ell)\backslash\mathcal{S}(\{c^1,\ldots,c^{\ell-1}\})}$. In the matching system the ℓth customer will depart if a server of type in $\mathcal{S}(c^\ell)\backslash\mathcal{S}(\{c^1,\ldots,c^{\ell-1}\})$ arrives, which has the same rate $\mu_{\mathcal{S}(c^\ell)\backslash\mathcal{S}(\{c^1,\ldots,c^{\ell-1}\})}$.

So in the coupled system the first change from state $c^1,\ldots,c^L$ will be the same for both systems. This completes the proof. □

4.2 Comparing the FCFS-ALIS and the redundancy service systems

In contrast, the situation is different when we compare the FCFS-ALIS system with the redundancy system. We list some points for comparison:

- The process $X^q(t)|$busy and $X^r(t)$ have the same stationary distribution, but $X^r(t)$ includes all customers in the system, those waiting and those being served, while $X^q(t)$ only includes waiting customers, so there is an additional set of customers which are currently in service in the FCFS-ALIS system.
  It is in fact shown that the stationary distribution of the types of customers that are in service in the FCFS-ALIS system cannot be expressed in product form, even for the simple "N" compatibility graph; see [10].
- One can regard the FCFS-ALIS system also as a system in which customers split on arrival into several copies that queue up at all the compatible servers, similar to the redundancy queue. However, at the instant that service of one copy can start, only the longest idle server will start processing the job, and all other copies disappear, so there is no simultaneous processing.
- It is worth mentioning that the FCFS-ALIS system is equivalent to a system in which customers have full information about all the processing times in the system, and each arriving customer joins the compatible server with the shortest workload. This join the shortest workload policy (JSW) leads to the a Nash equilibrium determined by selfish customers.
- With the same set of customers $c^1,\ldots,c^L$, and the same set of idle servers $s^1,\ldots,s^K$ in the system, under FCFS-ALIS each busy server serves a different customer, while in the redundancy system different servers may serve the same customer simultaneously. Therefore, although the stationary distributions of $X^q(t)|$busy and $X^r(t)$ are the same, they are not sample path equivalent.
- Because all the processing times are exponentially distributed, there is no loss of processing time when a customer is served simultaneously by more than one server. In fact, if a set of servers are processing jobs, the next service completion will be at the same time whether they work on different customers or are processing the same customer simultaneously.
- If in the two systems there is the same set of customers (including waiting and in service), then the number of busy servers in the redundancy system is greater or equal to the number of busy servers in the FCFS-ALIS systems (equivalently, more idle servers under FCFS-ALIS, than in the redundancy system). This is because simultaneous service under the redundancy system is allowed.

The above considerations lead us to the following conjecture:

Conjecture 4.2. *The expected sojourn time under FCFS-ALIS is greater than under the redundancy system.*

A natural way to prove this conjecture is by coupling, however it is not immediately clear how to do it.

5 EMBEDDING IN THE INFINITE MATCHING MODEL

The similarity of the stationary distributions of the processes X^q, X^r, X^m and the infinite matching process X^∞, suggest that they may be more closely related. In this section we show how to embed the three service processes in the infinite matching model.

To do so we define a new infinite matching model that is very similar to the infinite matching model of Section 3 and [1, 2, 4]. It is also related to the model studied by [8].

We consider a single infinite sequence of customers and servers, which is generated as follows: each successive item in the list is a customer of type c_i with probability $\frac{\lambda_{c_i}}{\bar\lambda+\bar\mu}$, and it is a server of type s_j with probability $\frac{\mu_{s_j}}{\bar\lambda+\bar\mu}$, and successive items in the sequence are independent. The result is a sequence $\ldots,z^1,z^2,\ldots$, were each item z^n indicates either a type of customer or a type of server. We then perform FCFS matching of the customers and servers according to the compatibility graph $\mathcal{G}$, utilizing only matches of servers to earlier customers. This means in particular that a servers $z^n=s_j$, for which there is no earlier unmatched compatible customer, will remain unmatched. We call this the FCFS single stream interleaved infinite bipartite matching model, infinite interleaved matching model for short.

We can now define the the process $X^{b\infty}(n)=(c^1,\ldots,c^L)$ which records the ordered list of customers that are still unmatched after the n'th server has been matched to an earlier customer, or has been left unmatched, because no earlier match existed.

Theorem 5.1. *$X^{b\infty}(n)$ is a discrete time discrete state Markov chain, it is ergodic if and only if (2.1) holds, and its stationary distribution, up to a normalizing constant, is given by:*

$$P^{b\infty}(c^1,\ldots,c^L)\propto\prod_{\ell=1}^{L}\frac{\lambda_{c^\ell}}{\mu_{\mathcal{S}(\{c^1,\ldots,c^\ell\})}}. \tag{5.1}$$

The fraction of servers that remain unmatched is $1-\frac{\bar\lambda}{\bar\mu}$

Proof. It is seen immediately that the Markov chain $X^{b\infty}(n)$ is the jump chain of the process $X^m(t)$. Furthermore, the process $X^m(t)$ has jumps in which its state changes at the uniform times of a Poisson process of rate $\bar\lambda+\bar\mu$. The theorem follows. □

We now formulate two theorems for the interleaved infinite matching model. Their proofs are similar to the proof of Theorems 3.2, 3.4, given by Adan et al. [1] and will not be given here.

Theorem 5.2. *Let $\ldots,z^{-1},z^0,z^1,\ldots$ be a sequence of customer and server types defined as above. If (2.1) holds then almost surely there exists a interleaved matching of servers to cover all the customers, and this matching is unique.*

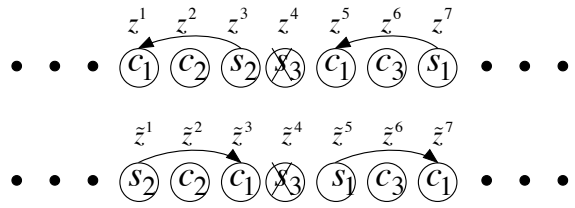

Figure 7: Interleaved matching and its reversal

We define an exchange transformation for the interleaved infinite matching model:

Definition 5.3. For the FCFS interleaved infinite matching model, if all matches were made on $\ldots, z^{-1}, z^0, z^1, \ldots$, we define the exchanged sequence $\ldots, \tilde{z}^{-1}, \tilde{z}^0, \tilde{z}^1, \ldots$ as follows: If $z^m = c_i$ was matched to $z^n = s_j$, where $m < n$, then in the exchanged sequence we will have $\tilde{z}^m = s_j$, $\tilde{z} = c_i$. If $z^n = s_j$ was unmatched, then $\tilde{z}^n = z^n$.

Figure 7 describes the state for a window in the doubly infinite interleaved sequence of customers and servers on the top panel. In it, $z^3 = s_2$ is matched with earlier $z^1 = c_1$, and z^7 is matched with z^5, while $z^2 = c_2$, $z^6 = c_3$ are not yet matched, and $z^4 = s_3$ will remain unmatched for ever. The state at 'time' 7, $X^{b\infty}(7) = (c_2, c_3)$, it is the embedding of the state of $X^m(t)$ as described in Figure 4. On the bottom panel of the Figure 7 we see the exchange transformation of the top panel, with the matchings in reversed time.

Theorem 5.4. *The sequence $\ldots, \tilde{z}^{-1}, \tilde{z}^0, \tilde{z}^1, \ldots$ obtained from the sequence $\ldots, z^{-1}, z^0, z^1, \ldots$ by the exchange transformation is an i.i.d. sequence. The unique matches for the new sequence, performed in reversed time result in exactly the same matches as in the original sequence.*

There are two important corollaries on properties of the match queue system which follow from these theorems:

Corollary 5.5. *The stationary distribution of the queues in the match queue model will remain the same if the arrival processes of customers and servers are quite general, as long as each successive arrivals is a customer of type c_i with probability $\frac{\lambda_{c_i}}{\bar{\lambda}+\bar{\mu}}$, and it is a server of type s_j with probability $\frac{\mu_{s_j}}{\bar{\lambda}+\bar{\mu}}$, independent of all other arrivals.*

Corollary 5.6. *The stationary sequence of types of customers departing from the match queue is i.i.d. with probabilities $\lambda_{c_i}/\bar{\lambda}$.*

Because the processes $X^m(t), x^r(t)$ are identical, we can also embed the redundancy system in the interleaved matching model. In particular, Corollary 5.6 holds for the redundancy service system. It implies that not only are customers of each type served FCFS by the system, but the system also equalizes service for the different types.

The process $X^q(t)$ can also be embedded in an infinite matching model, by considering the same sequences $\ldots, z^{-1}, z^0, z^1, \ldots$, but using a different matching mechanism: We now match each successive server $z^n = s_j$ to the earliest unmatched compatible customer $z^m = c_i$ where $m < k$ and k is the earliest position in the sequence with $k > n$, $z^k = s_j$. If no such match exists, the server z^n remains unmatched.

We define the process $X^{q\infty}(r)$ to describe the system after all possible matches that involve server z^k and customer z^ℓ for all $k, \ell \le r$ have been made. Then $X^{q\infty}(n) = (c^{i_1}, \ldots, c^{i_L}, s^{j_1}, \ldots, s^{j_K})$. Here $i_1, \ldots, i_k$ are the positions in the sequence of customers that are still unmatched and c_{i_ℓ} are their types, and $j_1, \ldots, j^K$ are positions in the sequence of servers that have not yet been matched, but may possibly be matched to a later customer in position $\ell > r$ in the sequence.

One can see that this process is the discrete time jump process of $X^q(t)$, and analogues of Theorems 5.1 and 5.2 hold.

APPENDIX: COMPLETION OF PROOFS

Proof of Theorem 2.3. The proof is by verifying that (2.3) satisfies partial balance. It is similar to the proof of Theorem 2.2 given in [3, 10], and to the proof of Theorem 2.5 given in [5].

We consider a state $x = (c^1, \ldots, c^L, s^1, \ldots, s^K)$. We list transitions in and out of the state x and their rates:

(i) Transition out of x due to arrival of type c_i, that joins the queue, rate λ_{c_i}, where $c_i \notin C(\{s^1, \ldots, s^K\})$
(ii) Transition out of x due to arrival of type c_i, that matches to one of the idle servers, at rate $\lambda_{C(\{s^1,\ldots,s^K\})}$.
(iii) Transition out of x due to completion of service, where server type s_j becomes idle, at rate: μ_{s_j}, for $s_j \notin S(\{c^1, \ldots, c^L\})$.
(iv) Transition out of x due to completion of service and start of service of a waiting customer, at rate: $\mu_{\mathcal{S}(\{c^1,\ldots,c^L\})}$
(v) Transition into state x due to arrival of c^L, at rate λ_{c^L}.
(vi) Transition into state x due to an arrival that matched with idle server s^* that was in position $k+1$, at rate: $\lambda_{C(s^*)\setminus C(\{s^1,\ldots,s^k\})}$, where $s^* \notin \mathcal{S}(\{c^1, \ldots, c^L\})$
(vii) Transition into state x due to a service completion, and server becoming idle, at rate μ_{s^K}.
(viii) Transition into state x due to a service completion, where a server is starting service of a customer c^* that was in position $\ell + 1$, at rate: $\mu_{\mathcal{S}(c^*)\setminus\mathcal{S}(\{c^1,\ldots,c^\ell\})}$.

We now show by substitution of the conjectured values from (2.3), that partial balance equations hold.

• Balance of (iv) with (v):

$$P^q(c^1, \ldots, c^L, s^1, \ldots, s^K) \times \mu_{\mathcal{S}(\{c^1,\ldots,c^L\})} = \prod_{\ell=1}^{L} \frac{\lambda_{c^\ell}}{\mu_{\mathcal{S}(\{c^1,\ldots,c^\ell\})}} \prod_{k=1}^{K} \frac{\mu_{s^k}}{\lambda_{C(\{s^1,\ldots,s^k\})}} \times \mu_{\mathcal{S}(\{c^1,\ldots,c^L\})};$$

$$P^q(c^1, \ldots, c^{L-1}, s^1, \ldots, s^K) \times \lambda_{c^L} = \prod_{\ell=1}^{L-1} \frac{\lambda_{c^\ell}}{\mu_{\mathcal{S}(\{c^1,\ldots,c^\ell\})}} \prod_{k=1}^{K} \frac{\mu_{s^k}}{\lambda_{C(\{s^1,\ldots,s^k\})}} \times \lambda_{c^L}.$$

• Balance of (ii) with (vii):

$$P^q(c^1, \ldots, c^L, s^1, \ldots, s^K) \times \lambda_{C(\{s^1,\ldots,s^K\})} = \prod_{\ell=1}^{L} \frac{\lambda_{c^\ell}}{\mu_{\mathcal{S}(\{c^1,\ldots,c^\ell\})}} \prod_{k=1}^{K} \frac{\mu_{s^k}}{\lambda_{C(\{s^1,\ldots,s^k\})}} \times \lambda_{C(\{s^1,\ldots,s^K\})};$$

$$P^q(c^1, \ldots, c^L, s_1, \ldots, s^{K-1}) \times \mu_{s^K} = \prod_{\ell=1}^{L} \frac{\lambda_{c^\ell}}{\mu_{\mathcal{S}(\{c^1,\ldots,c^\ell\})}} \prod_{k=1}^{K-1} \frac{\mu_{s^k}}{\lambda_{C(\{s^1,\ldots,s^k\})}} \times \mu_{s^K}.$$

• Balance of (i) with (viii):
For $c_i \notin C(\{s^1,\ldots,s^K\})$

$$P^q(c^1,\ldots,c^L,s^1,\ldots,s^K)\times\lambda_{c_i} = \prod_{\ell=1}^{L}\frac{\lambda_{c^\ell}}{\mu_{\mathcal{S}(\{c^1,\ldots,c^\ell\})}}\prod_{k=1}^{K}\frac{\mu_{s^k}}{\lambda_{C(\{s^1,\ldots,s^k\})}}\times\lambda_{c_i};$$

$$\sum_{\ell=0}^{L}P^q(c^1,\ldots c^\ell,c_i,c^{\ell+1},\ldots,c^L,s^1,\ldots,s^K) \times\mu_{\mathcal{S}(c_i)\setminus\mathcal{S}(\{c^1,\ldots,c^\ell\})} = = \sum_{\ell=0}^{L}\prod_{j=1}^{\ell}\frac{\lambda_{c^j}}{\mu_{\mathcal{S}(\{c^1,\ldots,c^j\})}}\times\frac{\lambda_{c_i}}{\mu_{\mathcal{S}(\{c_i,c^1,\ldots,c^\ell\})}} \times\prod_{j=\ell+1}^{L}\frac{\lambda_{c^j}}{\mu_{\mathcal{S}(\{c_i,c^1,\ldots,c^j\})}}\prod_{k=1}^{K}\frac{\mu_{s^k}}{\lambda_{C(\{s^1,\ldots,s^k\})}} \times\mu_{\mathcal{S}(c_i)\setminus\mathcal{S}(\{c^1,\ldots,c^\ell\})}.$$

To show that the two expressions do indeed balance, we need to show that:

$$\prod_{\ell=1}^{L}\frac{1}{\mu_{\mathcal{S}(\{c^1,\ldots,c^\ell\})}} = \sum_{\ell=0}^{L}\prod_{j=1}^{\ell}\frac{1}{\mu_{\mathcal{S}(\{c^1,\ldots,c^j\})}}\times\frac{1}{\mu_{\mathcal{S}(\{c_i,c^1,\ldots,c^\ell\})}} \times\prod_{j=\ell+1}^{L}\frac{1}{\mu_{\mathcal{S}(\{c_i,c^1,\ldots,c^j\})}}\times\mu_{\mathcal{S}(c_i)\setminus\mathcal{S}(\{c^1,\ldots,c^\ell\})} \tag{.2}$$

which follows by induction on L. For $L=1$:

$$\frac{1}{\mu_{\mathcal{S}(c_i)}}\frac{1}{\mu_{\mathcal{S}(c_i,c^1)}}\mu_{\mathcal{S}(c_i)} + \frac{1}{\mu_{\mathcal{S}(c^1)}}\frac{1}{\mu_{\mathcal{S}(\{c_i,c^1\})}}\mu_{\mathcal{S}(c_i)\setminus\mathcal{S}(c^1)}$$

$$= \frac{1}{\mu_{\mathcal{S}(\{c_i,c^1\})}}\frac{\mu_{\mathcal{S}(c^1)}+\mu_{\mathcal{S}(c_i)\setminus\mathcal{S}(c^1)}}{\mu_{\mathcal{S}(c_1)}} = \frac{1}{\mu_{\mathcal{S}(c^1)}},$$

and assuming that (.2) holds for $L-1$, we show that for L:

$$\sum_{\ell=0}^{L}\prod_{j=1}^{\ell}\frac{1}{\mu_{\mathcal{S}(\{c^1,\ldots,c^j\})}}\times\frac{1}{\mu_{\mathcal{S}(\{c_i,c^1,\ldots,c^\ell\})}} \times\prod_{j=\ell+1}^{L}\frac{1}{\mu_{\mathcal{S}(\{c_i,c^1,\ldots,c^j\})}}\times\mu_{\mathcal{S}(c_i)\setminus\mathcal{S}(\{c^1,\ldots,c^\ell\})}$$
$$= \sum_{\ell=0}^{L-1}\prod_{j=1}^{\ell}\frac{1}{\mu_{\mathcal{S}(\{c^1,\ldots,c^j\})}}\times\frac{1}{\mu_{\mathcal{S}(\{c_i,c^1,\ldots,c^\ell\})}} \times\prod_{j=\ell+1}^{L-1}\frac{1}{\mu_{\mathcal{S}(\{c_i,c^1,\ldots,c^j\})}}\frac{1}{\mu_{\mathcal{S}(\{c_i,c^1,\ldots,c^L\})}}\times\mu_{\mathcal{S}(c_i)\setminus\mathcal{S}(\{c^1,\ldots,c^\ell\})}$$
$$+\prod_{j=1}^{L}\frac{1}{\mu_{\mathcal{S}(\{c^1,\ldots,c^j\})}}\times\frac{1}{\mu_{\mathcal{S}(\{c_i,c^1,\ldots,c^L\})}}\times\mu_{\mathcal{S}(c_i)\setminus\mathcal{S}(\{c^1,\ldots,c^L\})}$$
$$= \prod_{j=1}^{L-1}\frac{1}{\mu_{\mathcal{S}(\{c^1,\ldots,c^j\})}}\times\frac{1}{\mu_{\mathcal{S}(\{c_i,c^1,\ldots,c^L\})}}\left(1+\frac{\mu_{\mathcal{S}(c_i)\setminus\mathcal{S}(\{c^1,\ldots,c^\ell\})}}{\mu_{\mathcal{S}(\{c^1,\ldots,c^L\})}}\right)$$
$$= \prod_{j=1}^{L}\frac{1}{\mu_{\mathcal{S}(\{c^1,\ldots,c^j\})}}$$

• Balance of (iii) with (vi):
For $s_j \notin \mathcal{S}(\{c^1,\ldots,c^L\})$

$$P^q(c^1,\ldots,c^L,s^1,\ldots,s^K)\times\mu_{s_j} = \prod_{\ell=1}^{L}\frac{\lambda_{c^\ell}}{\mu_{\mathcal{S}(\{c^1,\ldots,c^\ell\})}}\prod_{k=1}^{K}\frac{\mu_{s^k}}{\lambda_{C(\{s^1,\ldots,s^k\})}}\times\mu_{s_j};$$

$$\sum_{k=0}^{K}P^q(c^1,\ldots c^L,s^1,\ldots,s^k,s_j,s^{k+1},\ldots,s^K) \times\lambda_{C(s_j)\setminus C(\{s^1,\ldots,s^k\})} = \sum_{k=0}^{K}\prod_{\ell=1}^{L}\frac{\lambda_{c^\ell}}{\mu_{\mathcal{S}(\{c^1,\ldots,c^\ell\})}}\prod_{i=1}^{k}\frac{\mu_{s^i}}{\lambda_{C(\{s^1,\ldots,s^i\})}}\frac{\mu_{s_j}}{\lambda_{C(\{s_j,s^1,\ldots,s^k\})}} \prod_{i=k+1}^{K}\frac{\mu_{s^i}}{\lambda_{C(\{s_j,s^1,\ldots,s^i\})}}\times\lambda_{C(s_j)\setminus C(\{s^1,\ldots,s^k\})}.$$

To show that the two expressions do indeed balance, we need to show that:

$$\prod_{k=1}^{K}\frac{1}{\lambda_{C(\{s^1,\ldots,s^k\})}} = \sum_{k=0}^{K}\prod_{i=1}^{k}\frac{1}{\lambda_{C(\{s^1,\ldots,s^i\})}}\times\frac{1}{\lambda_{C(\{s_j,s^1,\ldots,s^k\})}} \times\prod_{i=k+1}^{K}\frac{1}{\lambda_{C(\{s_j,s^1,\ldots,s^i\})}}\times\lambda_{C(s_j)\setminus C(\{s^1,\ldots,s^k\})} \tag{.3}$$

The proof of (.3) is similar to the proof of (.2)

□

REFERENCES

[1] Ivo Adan, Ana Busic, Jean Mairesse, and Gideon Weiss. 2015. Reversibility and further properties of FCFS infinite bipartite matching. *arXiv preprint arXiv:1507.05939v2; Mathematics of Operations Research, to appear* (2015).

[2] Ivo Adan and Gideon Weiss. 2012. Exact FCFS matching rates for two infinite multitype sequences. *Operations Research* 60, 2 (2012), 475–489.

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

[4] René Caldentey, Edward H Kaplan, and Gideon Weiss. 2009. FCFS infinite bipartite matching of servers and customers. *Advances in Applied Probability* 41, 03 (2009), 695–730.

[5] Kristen Gardner, Samuel Zbarsky, Sherwin Doroudi, Mor Harchol-Balter, Esa Hyytiä, and Alan Scheller-Wolf. 2016. Queueing with redundant requests: exact analysis. *Queueing Systems* 83, 3-4 (2016), 227–259.

[6] Edward H Kaplan. 1984. *Managing the demand for public housing.* Ph.D. Dissertation. Massachusetts Institute of Technology.

[7] Edward H Kaplan. 1988. A public housing queue with reneging. *Decision Sciences* 19, 2 (1988), 383–391.

[8] Jean Mairesse and Pascal Moyal. 2016. Stability of the stochastic matching model. *Journal of Applied Probability* 53, 4 (2016), 1064–1077.

[9] Xuanming Su and Stefanos A Zenios. 2005. Patient choice in kidney allocation: A sequential stochastic assignment model. *Operations Research* 53, 3 (2005), 443–455.

[10] Jeremy Visschers, Ivo Adan, and Gideon Weiss. 2012. A product form solution to a system with multi-type jobs and multi-type servers. *Queueing Systems* 70, 3 (2012), 269–298.