Performance Paradox of Dynamic Bipartite Matching Models

Iratxe Iriondo, Josu Doncel

Performance Paradox of Dynamic Bipartite Matching Models

Iratxe Iriondo[1] and Josu Doncel[1][0000−0002−5552−9134]

University of the Basque Country, UPV/EHU. 48940 Leioa. Spain

Abstract. We analyze a system in which in each time slot one customer and one server arrive at the system according to a random process. Compatibilities between customers and servers are determined by a bipartite graph. An incoming customer (resp. server), if it finds a compatible server (resp. customer), they are matched and both leave the system. Otherwise, they are stored in a queue. We investigate the impact on the expected value of the unmatched customers and servers when we remove an edge from the compatibility graph. For a quasicomplete graph and a large family of matching policies, we provide necessary and sufficient conditions on the probability distribution of the arrivals such that a performance paradox occurs, i.e., such that removing an edge of the compatibility graph improves the performance of the system. This phenomenon can be seen as an analog of the Braess paradox in bipartite matching models.

Keywords: Bipartite matching models · Performance paradox· Markov chains

1 Introduction

We are interested in studying the performance of dynamic bipartite matching models. In these models, in each time slot, exactly one customer and one server arrive at the system. Compatible customer and server pairs are matched, in which case they leave the system. However, if upon arrival a customer (resp. server) does not find a compatible server (resp. customer) to be matched with, they wait in a queue.

To the best of our knowledge, the author in [10] is the first to study the dynamic bipartite matching model. In that work, the process of public housing in Boston is explored by considering that, upon the availability of a house, it is assigned to the longest family waiting for this kind of residence. The interest of [10] is the fraction of families having the same preferences that are assigned to a specific housing project, i.e., the matching rate. Later, an important contribution is given in [7], which introduces the First Come First Served infinite matching bipartite model. In this problem, a connected bipartite graph is defined, where nodes represent the class of incoming elements and the edges their compatibilities. As the compatibility graph is bipartite, the set of nodes can be separated into two parts: customer nodes and server nodes. Given the large number of

applications, for instance in call centers [8], many researchers have investigated this model. For instance, in [1], they characterize the necessary and sufficient conditions for the ergodicity of the Markov chain derived from this model and also show that the steady-state distribution has a product form expression. The authors in [2] consider other matching policies such as Last Come First Served, Random, or Priorities and study the stability condition for these cases. The bipartite matching models can be seen as a generalization of multi-skilled queueing networks in which customers and servers arrive randomly to the system (see [9] for a recent review of queueing systems with compatibilities).

There has been a recent interest of researchers in studying an alternative matching model in which the graph that describes the compatibilities is not bipartite. In this case, the arrivals of elements in the system are one by one. This variant is introduced by [11]. In [12], it is proven that the steady-state distribution of elements for this model under the First Come First Matched policy (which matches a customer with the oldest compatible server and servers with the oldest compatible customer) has a product-form expression. Using this result, the authors in [6] study the influence of adding an edge to the compatibility graph and conclude that, when flexibility in the compatibility graph increases and under the First Come First Matched policy, there exists a performance paradox in which the expected value of the number of unmatched items can increase. This result is generalized in [2] to greedy matching policies, which is a large family of matching policies that include First Come First Matched among others.

In this work, we address the following question: does the performance paradox of [6, 2] also occur in bipartite matching models? We consider a dynamic matching model with an arbitrary number of customer classes and two types of servers and a compatibility graph which consists of a quasicomplete graph. This means that all the customer and server pairs are compatible except for one. We consider a family of matching policies that prioritize previously unmatched items to match the incoming customer and server pairs. We first provide an analytical expression of the expected value of the total number of unmatched customers and servers in a matching model with a quasicomplete compatibility graph. Then, we remove one edge from the previously considered compatibility graph and we study the expected value of the total number of unmatched customers and servers. Using the derived expressions, we find a parametrized family of arrivals and, for this instance, we provide a necessary and sufficient condition for the existence of the performance paradox. We also show that the difference on the performance of both matching models when the performance paradox occurs is unbounded from above. This means that the degradation due to adding flexibility to a matching model can be arbitrarily large.

The rest of the article is organized as follows. We describe the model under study in section 2 as well as the assumptions we make. Then, in Section 3, we present the main results of our work concerning the performance paradox in bipartite matching models. Finally, the main conclusions and the future work are described in Section 4. For the sake a readability, some of the proofs have been reported to the Appendix.

2 Model Description

2.1 Bipartite Matching Models

We consider a system with multiple types of customers and servers in discrete time. In each time slot, one customer and one server arrive at the system. The set of customers classes is $\mathcal{C} = \{c_1, \ldots, c_m\}$ and the set of servers classes $\mathcal{S} = \{s_1, \ldots, s_n\}$. We denote by α_i (resp. by β_j) the probability that a customer of type c_i (resp. a server of type s_j) arrives at the system in a time slot. We assume independence of the arrivals of customers and servers. As a consequence, a customer of type c_i and a server of type s_j arrive at the system in a time slot with probability $\alpha_i\beta_j$. Also, $\sum_{i=1}^{m} \alpha_i = 1$ and $\sum_{i=1}^{n} \beta_i = 1$.

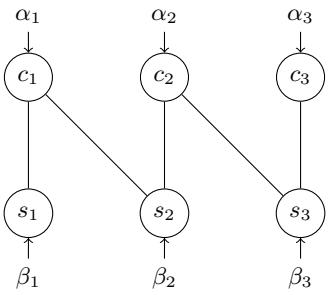

Fig. 1. A compatibility graph with three customer classes and three server classes.

We say that customers of type c_i are compatible with servers of type s_j when the customers of type c_i can be executed in a server of type s_j. Compatible customers and servers can be matched, in which case they disappear immediately; otherwise, they are stored in a queue. The compatibility between customers and servers is modeled as a bipartite graph $(\mathcal{C} \cup \mathcal{S}, \mathcal{E})$, where $\mathcal{E} \subset \mathcal{C} \times \mathcal{S}$ represents the set of compatible pairs of servers and customers. See Figure 1 for an example with $m = n = 3$.

The matching policy determines how compatible customers and servers are matched. An example of a matching policy is the First-Come-First-Matched discipline, in which an incoming customer is matched with the oldest compatible server (and, likewise, an incoming server is matched with the oldest compatible customer).

Example 1. Consider the compatibility graph depicted in Figure 1 and the First-Come-First-Served matching policy. The system is initially empty. Let us consider that a customer of type c_3 and a server of type s_2 arrive at the system. Since the arriving customer and server are not compatible, they are stored in a queue. If, in the next time slot, the incoming customer is of type c_2 and the incoming server is of type s_1, the server of type s_2 is matched with the customer of

type c_2 and the server of type s_1 is stored in the queue, leading to a situation in which there is one the customer of type c_3 and one server of type s_1 unmatched.

2.2 Performance Paradox

A bipartite matching model is formed by the triple $(G, (\boldsymbol{\alpha}, \boldsymbol{\beta}), \psi)$, where G is the compatibility graph, $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is the probability distribution of arrivals of customers and servers and ψ is the matching policy under consideration. For a given matching model, the number of unmatched customers and servers of the derived matching model is a Markov chain. This Markov chain will be denoted by $M(G, (\boldsymbol{\alpha}, \boldsymbol{\beta}), \psi)$ in the following. Let $\mathbb{E}[M(G, (\boldsymbol{\alpha}, \boldsymbol{\beta}), \psi)]$ be the expected value of the total number of unmatched customers and servers.

In this work, we study the impact on the mean number of customers when we remove an edge (c_i, s_j) from the compatibility graph. To this aim, we consider a compatibility graph $G - (c_i, s_j)$, which consists of the compatibility graph G without the edge (c_i, s_j). We denote by $M(G - (s_i, c_j), (\boldsymbol{\alpha}, \boldsymbol{\beta}), \psi)$ the Markov chain derived from this matching model and by $\mathbb{E}[M(G - (c_i, s_j), (\boldsymbol{\alpha}, \boldsymbol{\beta}), \psi)]$ the expected value of the mean number of customers and servers for this case.

We say that there exists a performance paradox in a matching model when

$$\mathbb{E}[M(G, (\boldsymbol{\alpha}, \boldsymbol{\beta}), \psi)] > \mathbb{E}[M(G - (c_i, s_j), (\boldsymbol{\alpha}, \boldsymbol{\beta}), \psi)].$$

From the above expression, we have that there exists a performance paradox in a matching model if adding an edge to the compatibility graph increases the expected value of unmatched customers and servers.

2.3 Assumptions

Let us present the following assumptions we make in this work.

Assumption 1 (Stability) *We assume that the arrivals satisfy the following condition:* $\forall C \subseteq \mathcal{C}\ \forall S \subseteq \mathcal{S}$

$$\sum_{c_i \in C} \alpha_i < \sum_{s_i \in S(C)} \beta_i \text{ and } \sum_{s_i \in S} \beta_i < \sum_{c_i \in C(S)} \alpha_i, \tag{1}$$

where $S(C)$ is the set of server types that are compatible with one of the customer types of C and $C(S)$ is the set of server types that are compatible with one of the customer types of S. According to [5], the above expression provide a necessary and sufficient condition for the stability of the matching model for the First-Come-First-Matched policy.

Assumption 2 (Compatibility graph) *We consider a compatibility graph which is a quasicomplete bipartite graph. This means that all but one of the customer and server pairs is compatible. We also assume that there is an arbitrary number of customer classes such that $m > 2$ and $n = 2$ server classes. Without loss of*

generality, we assume that customer class c_1 and server class s_2 are not compatible. This compatibility graph will be denoted as $K_{[2,m]} - (c_1, s_2)$. The stability condition of (1) *for this Markov chain is given by*

$$\alpha_1 < \beta_1. \tag{2}$$

In our performance paradox analysis, we assume that the edge we remove from the compatibility graph is (c_i, s_j) where $i \neq 1$ and $j \neq 2$. That is, the edges (c_1, s_2) and (c_i, s_j) do not have any node in common. Without loss of generality, we assume that $(c_i, s_j) = (c_m, s_1)$. This compatibility graph will be denoted as $K_{[2,m]} - \{(c_1, s_2), (c_m, s_1)\}$. The stability condition of (1) *for this case is given by*

$$\alpha_1 < \beta_1 \text{ and } \alpha_m < \beta_2. \tag{3}$$

Assumption 3 *We assume that the matching policy is such that, upon arrival of one customer and one server that are compatible, the matching priority is given to customers and servers that have not been previously matched (i.e. the incoming customers and servers are not matched even though they are compatible if there are other compatible customers and servers in the system). This family of matching policies includes First Come First Matched, and MaxWeight, which maximizes the number of matching at any time. This family of matching disciplines will be denoted as D.*

3 Performance Paradox Analysis

The main result of this work consists of providing a necessary and sufficient condition such that the performance paradox exists, i.e., such that

$$\mathbb{E}[M(K_{[2,m]} - (c_1, s_2), (\boldsymbol{\alpha}, \boldsymbol{\beta}), D)] > \mathbb{E}[M(K_{[2,m]} - \{(c_1, s_2), (c_m, s_1)\}, (\boldsymbol{\alpha}, \boldsymbol{\beta}), D)].$$

We first focus on $M(K_{[2,m]} - (c_1, s_2), (\boldsymbol{\alpha}, \boldsymbol{\beta}), D)$. In the following result, we characterize this Markov chain and we provide an expression of $\mathbb{E}[M(K_{[2,m]} - (c_1, s_2), (\boldsymbol{\alpha}, \boldsymbol{\beta}), D)]$. The proof of this result is reported in Appendix A.

Lemma 1. *The Markov chain $M(K_{[2,m]} - (c_1, s_2), (\boldsymbol{\alpha}, \boldsymbol{\beta}), D)$ is a birth-death process with birth probability equal to $\lambda_1 = \alpha_1\beta_2$ and death probability $\mu_1 = (1 - \alpha_1)\beta_1$. Therefore, if $\rho_1 = \lambda_1/\mu_1$, we have that $\rho_1 < 1$ and*

$$\mathbb{E}[M(K_{[2,m]} - (c_1, s_2), (\boldsymbol{\alpha}, \boldsymbol{\beta}), D)] = \frac{2\rho_1}{1 - \rho_1}.$$

We now focus on $M(K_{[2,m]} - \{(c_1, s_2), (c_m, s_1)\}, (\boldsymbol{\alpha}, \boldsymbol{\beta}), D)$. In the following result, we characterize this Markov chain and provide an analytical expression of $\mathbb{E}[M(K_{[2,m]} - \{(c_1, s_2), (c_m, s_1)\}, (\boldsymbol{\alpha}, \boldsymbol{\beta}), D)]$. The proof of this result can be found in Appendix B.

Lemma 2. *The Markov chain* $M(K_{[2,m]}-\{(c_1,s_2),(c_m,s_1)\},(\boldsymbol{\alpha},\boldsymbol{\beta}),D)$ *is formed by two birth-death processes which are connected by the state of the empty system. The birth probability and death probability of one of them are, respectively,* λ_1 *and* μ_1 *(which have been defined in Lemma 1), whereas for the other birth-death process the birth probability is* $\lambda_2 = \alpha_m\beta_1$ *and the death probability* $\mu_2 = (1-\alpha_m)\beta_2$*. Therefore, if* $\rho_1 = \lambda_1/\mu_1$ *and* $\rho_2 = \lambda_2/\mu_2$*, we have that* $\rho_1 < 1$ *and* $\rho_2 < 1$ *and*

$$\mathbb{E}[M(K_{[2,m]} - \{(c_1,s_2),(c_m,s_1)\},(\boldsymbol{\alpha},\boldsymbol{\beta}),D)] = \frac{2(1-\rho_1)(1-\rho_2)}{1-\rho_1\rho_2}\left(\frac{\rho_1^2}{(1-\rho_1)^1}+\frac{\rho_2^2}{(1-\rho_2)^1}\right).$$

From the above results, we conclude that there exists a performance paradox when

$$\frac{2(1-\rho_1)(1-\rho_2)}{1-\rho_1\rho_2}\left(\frac{\rho_1^2}{(1-\rho_1)^2}+\frac{\rho_2^2}{(1-\rho_2)^2}\right) > \frac{2\rho_1}{1-\rho_1}. \tag{4}$$

We now consider the following probability distribution for the arrivals of the customer types: $\alpha_1 = 0.45$, $\alpha_i = \frac{0.1}{m-2}$, for $i = 2,\ldots,m-1$ and $\alpha_m = 0.45$. For the arrivals of server types, we consider the following parametrized family of probability distributions: $\beta_1 = 0.5+\delta$ and $\beta_2 = 0.5-\delta$. We assume that $\delta \in (0, 0.05)$, which case (2) and (3) are satisfied, as it can be seen here:

$$\alpha_1 = 0.45 < 0.5+\delta = \beta_1 \text{ and } \alpha_3 = 0.45 < 0.5-\delta = \beta_2.$$

For these values and after some simplifications, we get that

$$\rho_1 = \frac{9}{11}\frac{0.5+\delta}{0.5-\delta}, \quad \rho_2 = \frac{9}{11}\frac{0.5-\delta}{0.5+\delta}.$$

It is easy to check that, when $\delta \in (0, 0.05)$, $\rho_1 < 1$ and $\rho_2 < 1$. From the above results, we have that

$$\frac{2(1-\rho_1)(1-\rho_2)}{1-\rho_1\rho_2}\left(\frac{\rho_1^2}{(1-\rho_1)^1}+\frac{\rho_2^2}{(1-\rho_2)^1}\right) = \frac{99}{10}\frac{1+400\delta^2}{1-400\delta^2}, \tag{5}$$

and

$$\frac{2\rho_1}{1-\rho_1} = \frac{9(1+2\delta)}{1-20\delta}. \tag{6}$$

Using the above formulas, we characterize the existence of a performance paradox in this matching model in the following theorem.

Theorem 1. *In the above matching model, there exists a performance paradox if and only if* $\delta \in (0.005, 0.05)$.

Proof. We aim to determine the values of δ such that (4) is satisfied. According to (5) and (6), we have that

$$\begin{aligned}
\frac{9(1+2\delta)}{1-20\delta} > \frac{99}{10}\frac{1+400\delta^2}{1-400\delta^2} &\iff 9(1+2\delta) > \frac{99}{10}\frac{1+400\delta^2}{1+20\delta} \\
&\iff 90(1+2\delta)(1+20\delta) > 99(1+400\delta^2) \\
&\iff (90+180\delta)(1+20\delta) > 99+39600\delta^2 \\
&\iff 90+180\delta+180\delta^2+3600\delta^2 > 99+39600\delta^2 \\
&\iff 3600\delta^2-1980\delta+9 < 0 \\
&\iff 3600\left(\delta-\frac{1}{20}\right)\left(\delta-\frac{1}{200}\right) < 0.
\end{aligned}$$

The last expression is only true when $\frac{1}{200} < \delta < \frac{1}{20}$, which proves that the desired result follows.

The intuition behind the above result is the following. When we remove the edge (c_m, s_1) of the compatibility graph, upon arrival of one customer of type c_1 and one server of type s_2, the former can be matched with a server of type s_1 and the latter with a customer of type c_m. We remark that this is not possible if the edge (c_m, s_1) would belong to the compatibility graph since they would be previously matched (and therefore, they could not be matched with a server of type s_1 and a customer of type c_m). A similar phenomenon has been also observed recently in non-bipartite matching models in [2, 6]. Our work shows that the performance paradox occurs also in bipartite matching models.

Finally, we show that the degradation due to the existence of the performance paradox can be arbitrarily large.

Theorem 2. *When $\delta \to 0.5^-$, then*

$$\mathbb{E}[M(K_{[2,m]} - (c_1, s_2), (\boldsymbol{\alpha}, \boldsymbol{\beta}), D)] - \mathbb{E}[M(K_{[2,m]} - \{(c_1, s_2), (c_m, s_1)\}, (\boldsymbol{\alpha}, \boldsymbol{\beta}), D)]$$

tends to infinity.

Proof. We note that, from (5) and (6), the desired result follows if we show that

$$\frac{9(1+2\delta)}{1-20\delta} - \frac{99}{10}\frac{1+400\delta^2}{1-400\delta^2},$$

tends to infinity when $\delta \to 0.05^-$.

From the above expression, using that $\delta = 0.05 - x$, we get the following:

$$\frac{9.1-18x}{20x} - \frac{198-3960x+39600x^2}{20x(20-200x^2)},$$

or equivalently

$$\frac{9.1-18x)(20-200x^2)-(198-3960x+39600x^2)}{20x(20-200x^2)}.$$

We simplify the last expression and it results:

$$\frac{-900x^2 + 445x - 4}{5x(20 - 200x)}.$$

We note that, when $x \to 0^+$, the numerator of the above ratio tends to -4, whereas the denominator to $+\infty$. This implies that the above ratio tends to infinity when $x \to 0^+$. And the desired result follows.

4 Conclusions and Future Work

We have studied the existence of a performance paradox in dynamic bipartite matching models. More precisely, we have considered a quasicomplete compatibility graph, an arbitrary number of customer classes and two server types, and we provide necessary and sufficient conditions on the arrivals of customers and servers such that the performance paradox exists. This work extends the performance paradox analysis of matching models with non-bipartite compatibility graphs of [2, 6] to bipartite matching models.

For future work, we are interested in analyzing the existence of the performance paradox in bipartite matching models with more complex compatibility graphs (for instance, more sparse compatibility graphs or with an arbitrary number of server classes). We would also like to analyze the existence of a performance paradox in other matching models such as in multigraphs [3] or in matching models with self-loops [4].

5 Acknowledgements

This research has been partially funded by the Department of Education of the Basque Government through the Consolidated Research Group MATHMODE (IT1456-22).

References

1. I. Adan and G. Weiss. Exact fcfs matching rates for two infinite multitype sequences. *Operations Research*, 60(2):475–489, 2012.
2. B. Ana, C. Arnaud, J. Doncel, and F. Jean-Michel. Performance Paradox of Dynamic Matching Models under Greedy Policies. working paper or preprint, June 2023.
3. J. Begeot, I. Marcovici, P. Moyal, and Y. Rahme. A general stochastic matching model on multigraphs. *arXiv preprint arXiv:2011.05169*, 2020.
4. A. Busic, A. Cadas, J. Doncel, and J.-M. Fourneau. Product form solution for the steady-state distribution of a markov chain associated with a general matching model with self-loops. In *European Workshop on Performance Engineering*, pages 71–85. Springer, 2022.
5. A. Bušić, V. Gupta, and J. Mairesse. Stability of the bipartite matching model. *Advances in Applied Probability*, 45(2):351–378, 2013.

6. A. Cadas, J. Doncel, J.-M. Fourneau, and A. Busic. Flexibility can hurt dynamic matching system performance. *ACM SIGMETRICS Performance Evaluation Review*, 49(3):37–42, 2022.
7. R. Caldentey, E. H. Kaplan, and G. Weiss. Fcfs infinite bipartite matching of servers and customers. *Advances in Applied Probability*, 41(3):695–730, 2009.
8. N. Gans, G. Koole, and A. Mandelbaum. Telephone call centers: Tutorial, review, and research prospects. *Manufacturing & Service Operations Management*, 5(2):79–141, 2003.
9. K. Gardner and R. Righter. Product forms for fcfs queueing models with arbitrary server-job compatibilities: an overview. *Queueing Systems*, 96(1):3–51, 2020.
10. E. H. Kaplan. *Managing the demand for public housing.* PhD thesis, Massachusetts Institute of Technology, 1984.
11. J. Mairesse, P. Moyal, et al. Stability of the stochastic matching model. *Journal of Applied Probability*, 53(4):1064–1077, 2016.
12. P. Moyal, A. Busic, and J. Mairesse. A product form for the general stochastic matching model. *arXiv preprint arXiv:1711.02620*, 2017.

A Proof of Lemma 1

We consider the Markov chain $M(K_{[2,m]} - (c_1, s_2), (\boldsymbol{\alpha}, \boldsymbol{\beta}), D)$. We assume that the system is empty. In each time slot, one customer and one server arrive at the system. If the incoming server is of type s_1, it is matched with the incoming customer since servers of class s_1 are compatible with all the customer classes. Likewise, if the incoming server is of type s_2 and the incoming customer of class c_i, with $i \neq 1$, they are matched. When the incoming server is of type s_2 and the incoming customer is of type c_1, they are not compatible, therefore they are stored in a queue. This occurs with probability $\alpha_1\beta_2$.

Assume now that there are k customers of type c_1 and k servers of type s_2 in the queue. With probability $\alpha_1\beta_2$, one customer of type c_1 and one server of type s_2 arrive at the next time slot. In this case, since the arriving elements are not compatible, the number of unmatched customers and servers gets increased by one. With probability $(1-\alpha_1)\beta_1$, one customer of type c_i, with $i \neq 1$, and one server of type s_1 arrive at the system, in which case, taking into account the matching policies under consideration (see Assumption 3), the incoming customer is matched with one server of type s_2 and the incoming server with one customer of type c_1. Therefore, the number of unmatched customers and servers decreases by one. In the rest of the cases, the incoming server and customer are matched, which implies that the number of unmatched customers and servers remains unchanged.

The above argument shows that the number of unmatched customers is a birth-death process in which the birth probability is $\lambda_1 = \alpha_1\beta_2$ and the death probability is $\mu_1 = (1-\alpha_1)\beta_1$. We now show that $\rho_1 = \lambda_1/\mu_1$ is smaller than one:

$$\frac{\alpha_1\beta_2}{(1-\alpha_1)\beta_1} < 1 \iff \alpha_1 < \beta_1,$$

and the last expression is true due to the stability condition (2).

As a result, the steady-state probability of being k customers of type c_1 unmatched in the system is $(1-\rho_1)\rho_1^k$. Therefore, we have clearly that the expected number of unmatched customers of class c_1 is $\frac{\rho_1}{1-\rho_1}$. Using the same reasoning, we derive that the expected value of the number of unmatched servers of class s_2 is $\frac{\rho_1}{1-\rho_1}$. As a consequence, the expected value of the total number of unmatched servers and customers is $\frac{2\rho_1}{1-\rho_1}$. And the desired result follows.

B Proof of Lemma 2

We consider the Markov chain $M(K_{[2,m]} - \{(c_1, s_2), (c_m, s_1)\}, (\boldsymbol{\alpha}, \boldsymbol{\beta}), D)$. One can use the same arguments as in Appendix A to conclude that two birth and death processes form the Markov chain; one with the same birth and death process as in Appendix A and the other with birth probability $\lambda_2 = \alpha_m\beta_1$ and death probability $\mu_2 = (1-\alpha_m)\beta_2$. Thus, if π_0 is the normalization constant, the steady-state probability of being k customers of class c_1 and k servers of class s_2 is $\pi_0\rho_1^k$, whereas the steady-state probability of being k customers of class c_1 and k servers of class s_2 is $\pi_0\rho_2^k$, where $\rho_2 = \lambda_2/\mu_2$. We know from Appendix A that ρ_1. We now show that $\rho_2 < 1$ as follows:

$$\rho_2 = \frac{\alpha_m\beta_1}{(1-\alpha_m)\beta_2} < 1 \iff \alpha_m < \beta_2,$$

and the last expression is true because of the stability condition (3). Therefore, we compute the value of the normalization constant as follows:

$$\pi_0 \left(1 + \sum_{i=1}^{\infty} \rho_1^i + \sum_{j=1}^{\infty} \rho_2^j \right) = 1 \iff \pi_0 = \frac{(1-\rho_1)(1-\rho_2)}{1-\rho_1\rho_2}.$$

From the above reasoning, we conclude that the expected value of unmatched customers is given by

$$\pi_0 \left(\sum_{i=1}^{\infty} i\rho_1^i + \sum_{j=1}^{\infty} j\rho_1^j \right) = \frac{(1-\rho_1)(1-\rho_2)}{1-\rho_1\rho_2} \left(\frac{\rho_1^2}{(1-\rho_1)^2} + \frac{\rho_2^2}{(1-\rho_2)^2} \right).$$

And the expected value of unmatched servers is also given by the above expression. Therefore, the desired result follows.