Load-balancing for Multi-skilled Servers with Bernoulli Routing

Fernando Miguelez, Josu Doncel, Balakrishna Prabhu

Load-balancing for Multi-skilled Servers with Bernoulli Routing

Fernando Miguelez[a], Josu Doncel[a], Balakrishna J. Prabhu[b]

[a] University of the Basque Country, UPV/EHU, Leioa, Spain.
[b] LAAS-CNRS, Toulouse, France.

December 19, 2020

Abstract

We study the optimal Bernoulli routing in a multiclass queueing system with a dedicated server for each class as well as a common (or multi-skilled) server that can serve jobs of all classes. Jobs of each class arrive according to a Poisson process. Each server have a holding cost per customer and use the processor sharing discipline for service. The objective is to minimize the weighted mean holding cost.

First, we provide conditions under which classes send their traffic only to their dedicated server, only to the common server, or to both. A fixed point algorithm is given for the computation of the optimal solution. We then specialize to two classes and give explicit expressions for the optimal loads. Finally, we compare the cost of multiskilled server with that of only dedicated or all common servers. The theoretical results are complemented by numerical examples that illustrate the various structural results as well as the convergence of the fixed point algorithm.

1 Introduction

1.1 Motivation

We investigate the performance of a multiskilled queueing system formed by parallel servers with Processor Sharing (PS) queues. Jobs of different classes of customers arrive to the system following a Poisson process. There is one dedicated server for each class of customer and one multi-skilled server that can execute jobs of all classes. Furthermore, we assume that jobs are assigned to the servers according to the Bernoulli policy. Our goal is to find the optimal load balancing so as to minimize the weighted mean number of jobs in the system.

The main application of our model comes from wireless networks. Consider a region divided in different subregions. Each dedicated server models an antenna that provides service to a unique subregion and the multi-skilled server models a

central antenna that provides service to all the subregions. Using the results of this article, one can determine how the traffic of each subregion must be shared between the antenna of that region and the central one in order to minimize the performance of the system. This architecture has been previously considered by [1] in a different context where dedicated servers (or microcells in their model) can be switched on and off so as to minimize the weighted sum of the mean delay and the mean power consumption in the system.

1.2 Related Work

Load balancing has been widely investigated in different contexts. In data centers, for example, various policies depending upon the information available to the dispatcher have been proposed. In general, optimal policies for the typical performance measures such as mean processing times are not easy to determine albeit in some specific cases. For example, when no information on the state of the servers is available, the optimal Bernoulli routing policy was determined in [2] for mono-skilled servers only. For FCFS servers, a policy based on Sturm sequences [3] are known to be optimal. With more information on the server state, a number of heuristics such as Join the Shorter of d queues [4, 5] and Join the Shortest Queue [6] have been analyzed in the large server asymptotic case. In addition, there are various pull-based policies such as Join the Idle Queue [7] that are known to work well in practice. Another important routing policy is the Size Interval Task Assignment [8] where jobs of different sizes are executed in different servers and, therefore, the service requirement of incoming tasks need to be known. This policy has been further studied in [9] and the author in [10] presented a variation of this policy in which the size of jobs does not need to be known.

Load balancing has also been investigated for balancing energy costs in data centers using Energy Packet Networks model [11], whereas in [12] it is considered that data centers are located in different geographical zones. The above works are mostly concerned with mono-skilled or homogeneous servers.

In networks with multi-skilled agents or servers, skill-based routing policies have been proposed and investigated [13, 14]. These works are mainly oriented towards call-center architectures with Erlang-B or Erlang-C type of queues. An illustrative example is an overflow-type policy, where each incoming call has a list of agents ordered by priority, with highest priority given to mono-skilled ones, and is routed to the first available agent of this list. If no agent is available, the call can be queued or blocked depending on the architecture. These routing policies are usually difficult to analyze and the cited works are interested in approximations for the various performance measures for a given policy. In these models, obtaining the optimal policy analytically is not easy. We refer to [15] for a recent survey on multiskilled systems.

Multi-skilled queues appear also in the analysis of redundancy systems [16, 17] in which incoming requests can be sent simultaneously to a subset of queues. We do not investigate the redundancy aspect.

The network topology we consider makes our model different from [2] in which all servers can execute all type of tasks.

1.3 Contributions

The main contributions of the article are summarized as follows:

- We provide a necessary and sufficient condition for the stability of the system.
- We show that the optimization problem in terms of probabilities is equivalent to the optimization problem in terms of the loads of the servers.
- We fully characterize the optimal loads on the servers for two classes of customers. For more than two customers, we provide in Proposition 2 conditions under which each class of traffic satisfies one of the following: (i) it sends all its traffic to its dedicated server, (ii) it sends all its traffic to the multi-skilled server and (iii) it shares its traffic among the multi-skilled and its dedicated server.
- Using the result of Proposition 2, we present a fixed-point algorithm whose convergence ensures that the optimal loads on the servers are achieved. This algorithm starts with an initial condition of the set of servers (according to one of the three possible traffic sharing policies of Proposition 2) and its fixed point is given by the partition of the set of servers. Providing an analytical proof of this convergence on the partition of the set of servers seems to be an extremely difficult task. However, we illustrate the convergence of this algorithm using numerical experiments.
- We compare the performance of our model with the performance of two models. The first model consists of a system where all the servers are multi-skilled and we show the existence of a switching curve, i.e., when the arrival rate of one of the traffic increases, the model whose performance is better changes. The second model consists of a system with no sharing, that is, all the servers are dedicated or mono-skilled, and we provide conditions on the arrival rates such that the performance of the no-sharing model is larger than the performance of our model.
- We delve into the comparison of the aforementioned models using numerical experiments. First, we show the uniqueness of the switching curve when we compare our model with a system where all the servers are multi-skilled. We also observe that, in a system formed by servers with equal capacity and different (but not extremely large) holding costs, the region where our model outperforms the all-sharing system is very large.

1.4 Organization

In the next section, we describe the network model and define the optimization problem. Section 3 gives the stability condition and presents an equivalent

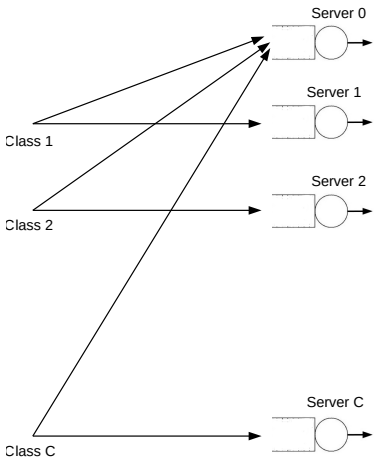

Figure 1: The model under study in this article.

problem in terms of loads on the servers. In Section 4, the main results on the structure of the optimal policy are provided for the model under consideration. We compare in Section 5 the performance of our model with the performance of models with other network topologies. We present our numerical experiments in Section 6. Finally, we discuss our main conclusions in Section 7.

2 Model Description

2.1 Notation

We consider a server farm with Processor Sharing (PS) queues and an input traffic of different classes. Let $\mathcal{K} = \{1, 2, \ldots, C\}$ be the set of classes. We assume that jobs of class $i \in \mathcal{K}$ arrive to the system according to a Poisson process and have generally distributed service times[1]. Let η_i be the traffic intensity of jobs of class i. The class of a job defines the set of servers that can be assigned to this job.

We consider a system with $C + 1$ servers. Let $\mathcal{S} = \{0, 1, \ldots, C\}$ be the set of servers. For a server $j \in \mathcal{S}$, we denote by r_j the capacity of Server j and by c_j its holding cost. We denote by $\mathcal{S}_i$ the set of servers that can execute jobs of class i and, for $A \subset \mathcal{K}$, $\mathcal{S}_A = \cup_{i \in A} \mathcal{S}_i$. For $j = 1, \ldots, C$, Server j executes jobs of class j, i.e., they are dedicated servers. On the other hand, Server 0 executes jobs of all the classes, i.e., it is a multiskilled server.

For $i = 1, \ldots, C$, we denote by p_i the probability that a class i job is executed in its dedicated server, i.e., Server i. For $j = 1, \ldots, C$, the load of Server j is

[1]Since our goal is to analyze the mean number of jobs and, in a M/G/1-PS queue, the mean number of jobs depends on the arrival rate and on the service time requirements only through the intensity, we do not specify the arrival rate of each class

defined as follows

$$\rho_j(\mathbf{p}) = \frac{\eta_j p_j}{r_j}, \tag{1}$$

whereas for Server 0 as

$$\rho_0(\mathbf{p}) = \frac{\sum_{j\in\mathcal{K}} \eta_j(1-p_j)}{r_0}. \tag{2}$$

2.2 Problem Formulation

For a given routing strategy $\mathbf{p} = (p_i)$, the mean number of jobs of server j is denoted by $\mathbb{E}[N_j(\mathbf{p})]$. In this article, we aim to find the routing matrix that minimizes the total cost of the system. More specifically, we analyze the following optimization problem:

$$\min_{\mathbf{p}} \quad \sum_{j\in\mathcal{S}} c_j \mathbb{E}[N_j(\mathbf{p})] \tag{PROB-OPT}$$

$$0 \le p_i \le 1, \text{ for all } i \in \mathcal{K}; \tag{3}$$

$$\eta_i p_i < r_i, \text{ for all } i \in \mathcal{K}\ ; \tag{4}$$

$$\sum_{i\in\mathcal{K}} \eta_i(1-p_i) < r_0. \tag{5}$$

The first constraint ensures that p_i's are probabilities. The second and third constraints ensure that all the servers are stable, that is, that the total incoming traffic into a server is smaller than its service capacity.

3 Preliminary Results

We first study the existence of a feasible solution of (PROB-OPT). This is the same as characterizing the conditions under which the system can be stabilized. In the following proposition, we provide this result.

Proposition 1 (Stability)**.** *The system under consideration can be stabilized if and only if*

$$r_0 + \sum_{i\in A} r_i > \sum_{i\in A} \eta_i, \ \forall A \subset \mathcal{K}. \tag{6}$$

Proof. See Appendix A. □

Taking into account that $\mathbb{E}[N_j(\mathbf{p})] = \frac{\rho_j}{1-\rho_j}$, we can reformulate (PROB-OPT) in terms of the loads on the servers as follows:

$$\min_{\rho} \quad \sum_{j\in\mathcal{S}} c_j \frac{\rho_j}{1-\rho_j} \tag{LOAD-OPT}$$

$$\sum_{i\in\mathcal{K}} \eta_i = r_0\rho_0 + \sum_{j\in\mathcal{K}} r_j\rho_j; \tag{7}$$

$$0 \le \rho_j < 1, \text{ for all } j \in \mathcal{S}; \tag{8}$$

$$\sum_{i\in A} \eta_i \le r_0\rho_0 + \sum_{j\in\mathcal{A}} r_j\rho_j, \ \forall A \subset \mathcal{K}. \tag{9}$$

We now show that the optimization problems we have considered so far are related. More precisely, we show that, if there are routing probabilities that satisfy (PROB-OPT), then it is possible to find loads that satisfy (LOAD-OPT).

Lemma 1. *Let* $\mathbf{p}$ *be a routing strategy that satisfies* (3)-(5)*. Then, for all* $j \in \mathcal{S}$*,* $\rho_j(\mathbf{p})$ *also satisfies the constraints of* (LOAD-OPT)*.*

Proof. First, we observe that, if (3)-(5) are satisfied, using (1) and (2), it follows that $0 \le \rho_j < 1$ for all $j \in \mathcal{S}$.

We now show that $\sum_{i\in\mathcal{K}} \eta_i = r_0\rho_0 + \sum_{j\in\mathcal{K}} r_j\rho_j$ in the following way:

$$r_0\rho_0 + \sum_{j\in\mathcal{K}} r_j\rho_j = \sum_{i\in\mathcal{K}} \eta_i(1-p_i) + \sum_{i\in\mathcal{K}} \eta_i p_i = \sum_{i\in\mathcal{K}} \eta_i,$$

where the first equality is given using (1) and (2).

Finally, we focus on the constraint $\sum_{i\in A} \eta_i \le r_0\rho_0 + \sum_{j\in\mathcal{A}} r_j\rho_j, \ \forall A \subset \mathcal{K}$. Using again (1) and (2), we have for all $A \subset \mathcal{K}$ that

$$r_0\rho_0 + \sum_{j\in A} r_j\rho_j = \sum_{i\in\mathcal{K}} \eta_i(1-p_i) + \sum_{i\in\mathcal{A}} \eta_i p_i \ge \sum_{i\in\mathcal{A}} \eta_i(1-p_i) + \sum_{i\in\mathcal{A}} \eta_i p_i = \sum_{i\in\mathcal{A}} \eta_i.$$

And the desired result follows. □

Note that (LOAD-OPT) is a convex problem with linear constraints and has an unique solution as long as the stability condition in Proposition 1 is verified. Moreover, from the above lemma, the solution of (PROB-OPT) can be obtained by optimizing directly over the loads. Then, the optimal routing probabilities can be determined later from (1), once the optimal load on each server is determined.

4 Analysis of the Solution of (LOAD-OPT)

Let $\delta_j = \sqrt{\frac{c_j/r_j}{c_0/r_0}}$ for all $j \in \mathcal{K}$. We denote by C_b the set of classes that route traffic to two servers, by C_0 the set of classes that routes all the traffic to Server 0 and by C_d the set of classes that send all the traffic to its dedicated server.

In the following proposition, we present the first result of this section. It gives the conditions under which a class of traffic belongs to C_b, C_0 or C_d.

Proposition 2. *Jobs of class i routes traffic to Server 0 if and only if*

$$\delta_i > \frac{1 - \frac{\eta_i}{r_i}}{1 - \rho_0^*},$$

and all the traffic of class i is routed to Server 0 if and only if

$$\delta_i \geq \frac{1}{1 - \rho_0^*},$$

where ρ_0^* is the optimal load at Server 0 and is given by

$$\rho_0^* = 1 - \frac{r_0 + \sum_{j \in C_b} r_j - \sum_{j \in C_b \cup C_0} \eta_j}{r_0 + \sum_{j \in C_b} \delta_j r_j}. \tag{10}$$

Besides, if $j \in C_d$ the optimal load of Server j is $\frac{\eta_j}{r_j}$, if $j \in C_b$ the optimal load of Server j is

$$\rho_j^* = 1 - \delta_j (1 - \rho_0^*) \tag{11}$$

and if $j \in C_0$ the optimal load of Server j is zero.

Proof. See Appendix B. □

The above result leads to this corollary which gives a simple sufficient condition to determine when a given class will not send all its traffic to the multi-skilled server.

Corollary 1. *Let $j \in \mathcal{S}$. If $\delta_j < 1$, then $j \notin C_0$.*

Proof. Since $\delta_j < 1$, we have that the condition $\delta_j < \frac{1}{1-\rho_0^*}$ is always satisfied and this implies that $j \notin C_0$ according to Proposition 2. □

From the above corollary, it follows another interesting property that says that, if $\delta_j < 1$ for all $j \in \mathcal{K}$, then $C_0 = \emptyset$.

The next result gives an ordering which can help identify classes that use both the dedicated and the multi-skilled server. This can be seen as a way to determine, for a given set of input parameters (arrival rate, server speeds, holding costs, etc.), the skills for which we need to train the multi-skilled servers in order for the system to be optimal.

Proposition 3. *Let $\delta_i \leq \delta_j$.*

(a) *If $i \in C_0$, then $j \in C_0$.*

(b) *If $i \in C_b \cup C_0$ and $\frac{\eta_i}{r_i} \leq \frac{\eta_j}{r_j}$, then $j \in C_b \cup C_0$.*

Proof. We first show (a). We consider that $i \in C_0$. Since $\delta_j \geq \delta_i$, it follows that $\delta_j \geq \delta_i \geq \frac{1}{1-\rho_0^*}$.

Therefore, from Proposition 2, $j \in C_0$.

We now show (b). We consider that $i \in C_b \cup C_0$. Since $\frac{\eta_i}{r_i} \leq \frac{\eta_j}{r_j}$ and $\delta_j \geq \delta_i$, it follows that

$$\delta_j \geq \delta_i \geq \frac{1-\frac{\eta_i}{r_i}}{1-\rho_0^*} \geq \frac{1-\frac{\eta_j}{r_j}}{1-\rho_0^*}.$$

Therefore, from Proposition 2, $j \in C_b \cup C_0$. □

We note that (b) of the above result can be stated as follows: if class j routes all the traffic to Server j, class i routes all its traffic to Server i when $\frac{\eta_i}{r_i} \leq \frac{\eta_j}{r_j}$ and $\delta_i \leq \delta_j$. In the following result, we show that, under similar conditions, the set of classes that send traffic to two servers can never be $\{i, j\}$.

Proposition 4. *If $\delta_i \leq \delta_j < 1$ and $\frac{\eta_i}{r_i} \leq \frac{\eta_j}{r_0+r_j}$, then C_b cannot be $\{i, j\}$.*

Proof. We assume that $C_b = \{i, j\}$. For this case, it follows from (10) that

$$\frac{1}{1-\rho_0^*} \geq \frac{r_0 + r_j\delta_j + r_i\delta_i}{r_0 + r_j + r_i - \eta_j - \eta_i},$$

where the above inequality is an equality if $C_0 = \emptyset$.

Since $\frac{\eta_i}{r_i} \leq \frac{\eta_j}{r_0+r_j}$, we have for class i that

$$\begin{aligned}
&\delta_i > \frac{1-\frac{\eta_i}{r_i}}{1-\rho_0^*} \geq \left(1-\tfrac{\eta_i}{r_i}\right)\frac{r_0 + r_j\delta_j + r_i\delta_i}{r_0 + r_j + r_i - \eta_j - \eta_i} \iff \\
&\delta_i(r_0 + r_j + r_i - \eta_j - \eta_i) > \\
&\qquad \left(1-\tfrac{\eta_i}{r_i}\right)(r_0 + r_j\delta_j + r_i\delta_i) \iff \\
&\delta_i(r_0 + r_j - \eta_j) > \left(1-\tfrac{\eta_i}{r_i}\right)(r_0 + r_j\delta_j) \iff \\
&\delta_i > \frac{1-\frac{\eta_i}{r_i}}{1-\frac{\eta_j}{r_0+r_j}}\frac{r_0 + r_j\delta_j}{r_0 + r_j} \geq \frac{r_0 + r_j\delta_j}{r_0 + r_j} \iff \\
&\delta_i > \delta_j + \frac{r_0(1-\delta_j)}{r_0 + r_j} > \delta_j,
\end{aligned}$$

which is in contradiction with $\delta_i \leq \delta_j$. □

Let T_i denote the sojourn time of jobs of class i. We now provide an interesting result related to the sojourn time of jobs.

Proposition 5. *If $\delta_i \leq \delta_j$. Then, $c_i\mathbb{E}[T_i] \leq c_j\mathbb{E}[T_j]$.*

Proof. We know that the sojourn time of jobs of class i and of class j follow an exponential distribution with rate $\frac{1}{r_i(1-\rho_i^*)}$ and $\frac{1}{r_j(1-\rho_j^*)}$ respectively. Therefore,

$$\begin{aligned} c_i\mathbb{E}(T_i) &= \frac{c_i}{r_i(1-\rho_i^*)} \\ &= \frac{c_i}{r_i\delta_i}\frac{1}{1-\rho_0^*} \\ &= \sqrt{\frac{c_0}{r_0}}\sqrt{\frac{c_i}{r_i}}\frac{1}{1-\rho_0^*} \\ &\leq \sqrt{\frac{c_0}{r_0}}\sqrt{\frac{c_j}{r_j}}\frac{1}{1-\rho_0^*} \\ &= c_j\mathbb{E}(T_j). \end{aligned}$$

And the desired result follows. □

4.1 Charaterization of the Solution of (LOAD-OPT) with $C = 2$

We now focus on the case $C = 2$. Throughout this article, we refer to this case as the M model. Without loss of generality, we assume that $\delta_1 \leq \delta_2$. The goal of this section is to fully characterize the solution of (LOAD-OPT) with $C = 2$.

We first note that, from Proposition 3, it can never be given the following cases: (i) $C_0 = \{1\}$ and $C_d = \{2\}$ and (ii) $C_0 = \{1\}$ and $C_b = \{2\}$. For the remaining cases, we have the following options:

1. $C_d = \{1, 2\}$. In this case, each class sends all its traffic to the dedicated server. Therefore, $\rho_i^* = \frac{\eta_i}{r_i}$ for $i = 1, 2$ and $\rho_0^* = 0$. According to Proposition 2 this occurs when

$$\delta_i \leq 1 - \tfrac{\eta_i}{r_i}, \quad i = 1, 2.$$

2. $C_d = \{1\}$ and $C_b = \{2\}$. In this case, all the traffic of class 1 is sent to Server 1 and the traffic of class 2 is sent to Server 0 and Server 2. As a result, $\rho_1^* = \frac{\eta_1}{r_1}$ and, from (10) and (11) we obtain that $\rho_2^* = 1-\delta_2\frac{r_0+r_2-\eta_2}{r_0+\delta_2 r_2}$ and $\rho_0^* = 1 - \frac{r_0+r_2-\eta_2}{r_0+\delta_2 r_2}$. According to Proposition 2, this case occurs when $\delta_1 \leq \frac{1-\frac{\eta_1}{r_1}}{1-\rho_0^*}$ and $\frac{1}{1-\rho_0^*} > \delta_2 > \frac{1-\frac{\eta_2}{r_2}}{1-\rho_0^*}$, i.e.,

$$\begin{aligned} \delta_1 \leq \left(1 - \tfrac{\eta_1}{r_1}\right)\frac{r_0+\delta_2 r_2}{r_0+r_2-\eta_2} \quad \text{and} \\ \frac{r_0+\delta_2 r_2}{r_0+r_2-\eta_2} > \delta_2 > \left(1 - \tfrac{\eta_2}{r_2}\right)\frac{r_0+\delta_2 r_2}{r_0+r_2-\eta_2}, \end{aligned}$$

which simplifying gives

$$\delta_1 \leq \left(1 - \tfrac{\eta_1}{r_1}\right)\frac{r_0+\delta_2 r_2}{r_0+r_2-\eta_2} \quad \text{and} \quad \frac{1}{1-\frac{\eta_2}{r_0}} > \delta_2 > 1 - \frac{\eta_2}{r_2}.$$

3. $C_d = \{1\}$ and $C_0 = \{2\}$. In this case, all the traffic of class 1 is sent to Server 1 and the traffic of class 2 is sent to Server 0. As a result, $\rho_1^* = \frac{\eta_1}{r_1}$, $\rho_0^* = \frac{\eta_2}{r_0}$ and $\rho_2^* = 0$. According to Proposition 2, this case occurs when $\delta_1 \leq \frac{1-\frac{\eta_1}{r_1}}{1-\rho_0^*}$ and $\delta_2 \geq \frac{1}{1-\rho_0^*}$, i.e.

$$\delta_1 \leq \frac{1-\frac{\eta_1}{r_1}}{1-\frac{\eta_2}{r_0}} \text{ and } \delta_2 \geq \frac{1}{1-\frac{\eta_2}{r_0}}.$$

4. $C_0 = \{1,2\}$. In this case, the traffic of both classes is sent to Server 0. Hence, $\rho_i^* = 0$ for $i = 1,2$ and from (10) that $\rho_0^* = 1 - \frac{r_0-\eta_1-\eta_2}{r_0} = \frac{\eta_1+\eta_2}{r_0}$. According to Proposition 2 and using that $\delta_1 \leq \delta_2$, this case occurs when $\delta_1 \geq \frac{1}{1-\rho_0^*}$, i.e.,

$$\delta_1 \geq \frac{r_0}{r_0-\eta_1-\eta_2}.$$

5. $C_b = \{1,2\}$. In this case, the traffic of class i is sent to Server 0 and Server i, for $i = 1,2$. From (10) and (11), it results that $\rho_0^* = 1 - \frac{r_0+r_1+r_2-\eta_1-\eta_2}{r_0+\delta_1 r_1+\delta_2 r_2}$ and, for $i = 1,2$, $\rho_i^* = 1 - \delta_i \frac{r_0+r_1+r_2-\eta_1-\eta_2}{r_0+\delta_1 r_1+\delta_2 r_2}$. Moreover, we conclude from Proposition 2 that this occurs when, for $i = 1,2$, $\frac{1}{1-\rho_0^*} > \delta_i > \frac{1-\frac{\eta_i}{r_i}}{1-\rho_0^*}$, which using that $\delta_2 \geq \delta_1$ gives

$$\delta_1 > \left(1-\tfrac{\eta_1}{r_1}\right)\frac{r_0+\delta_1 r_1+\delta_2 r_2}{r_0+r_1+r_2-\eta_1-\eta_2} \quad \text{and}$$
$$\frac{r_0+\delta_1 r_1+\delta_2 r_2}{r_0+r_1+r_2-\eta_1-\eta_2} > \delta_2 > \left(1-\tfrac{\eta_2}{r_2}\right)\frac{r_0+\delta_1 r_1+\delta_2 r_2}{r_0+r_1+r_2-\eta_1-\eta_2}.$$

We simplify the above expressions and we obtain

$$\delta_1 > \left(1-\tfrac{\eta_1}{r_1}\right)\frac{r_0+\delta_2 r_2}{r_0+r_2-\eta_2} \quad \text{and}$$
$$\frac{r_0+\delta_1 r_1}{r_0+r_1-\eta_1-\eta_2} > \delta_2 > \left(1-\tfrac{\eta_2}{r_2}\right)\frac{r_0+\delta_1 r_1}{r_0+r_1-\eta_1}.$$

6. $C_b = \{1\}$ and $C_d = \{2\}$. We observe that this case is symmetric to the case 2 (where $C_b = \{2\}$ and $C_d = \{1\}$) and using the same arguments, we get the following conditions

$$\frac{1}{1-\frac{\eta_1}{r_0}} > \delta_1 > 1-\frac{\eta_1}{r_1} \quad \text{and} \quad \left(1-\tfrac{\eta_2}{r_2}\right)\delta_2 \leq \frac{r_0+\delta_1 r_1}{r_0+r_1-\eta_1}.$$

7. $C_b = \{1\}$ and $C_0 = \{2\}$. In this case, the traffic of class 1 is sent to Server 0 and Server 1, whereas all the traffic of class 2 to Server 0. As a result, we have that $\rho_2^* = 0$ and from (10) and (11), we obtain that $\rho_0^* = 1 - \frac{r_0+r_1-\eta_1-\eta_2}{r_0+\delta_1 r_1}$ and $\rho_1^* = 1 - \delta_1 \frac{r_0+r_1-\eta_1-\eta_2}{r_0+\delta_1 r_1}$. According to Proposition 2,

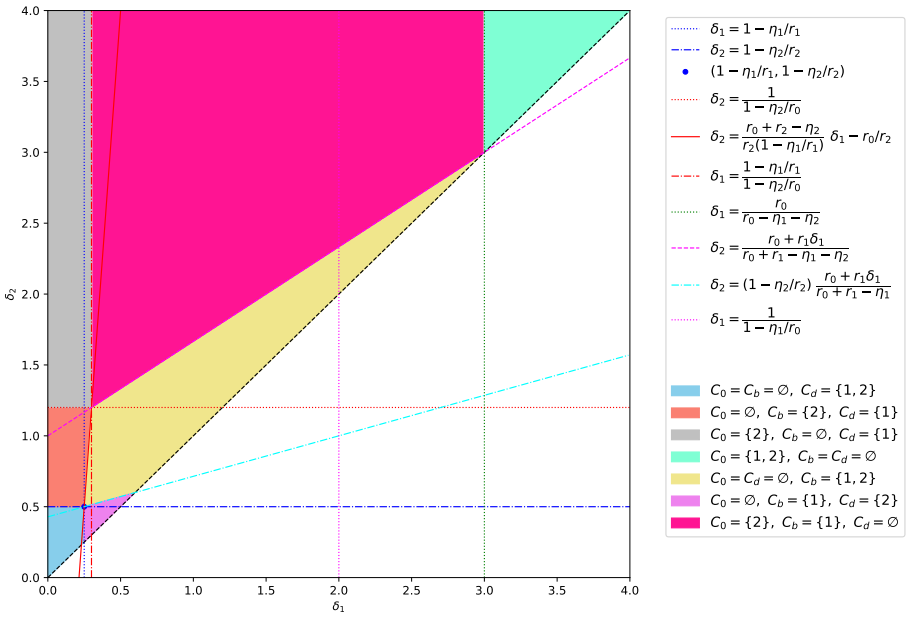

Figure 2: Conditions 1-7 determine a partition of the half-plane $0 \leq \delta_1 \leq \delta_2$.

we conclude that this case occurs when $\frac{1}{1-\rho_0^*} > \delta_1 > \frac{1-\frac{\eta_1}{r_1}}{1-\rho_0^*}$ and $\delta_2 \geq \frac{1}{1-\rho_0^*}$, i.e.,

$$\delta_2 \geq \frac{r_0 + \delta_1 r_1}{r_0 + r_1 - \eta_1 - \eta_2} > \delta_1 > \left(1 - \frac{\eta_1}{r_1}\right) \frac{r_0 + \delta_1 r_1}{r_0 + r_1 - \eta_1 - \eta_2},$$

which after some simplification results in

$$\delta_2 \geq \frac{r_0 + \delta_1 r_1}{r_0 + r_1 - \eta_1 - \eta_2} > \delta_1 > \frac{1 - \frac{\eta_1}{r_1}}{1 - \frac{\eta_2}{r_0}}.$$

The conditions described in items 1-7 split the feasible half-plane $0 \leq \delta_1 \leq \delta_2$ into, at most, 7 disjoint regions. A detailed example of such partition is shown in Figure 2.

4.2 Computation of the solution of (LOAD-OPT) with $C > 2$

As we saw before, the characterization of the solution of (LOAD-OPT) with $C = 2$ requires to distinguish seven different cases. This suggest that the characterization of the solution of (LOAD-OPT) with an arbitrary number of classes is to be out of reach. However, we provide a fixed-point algorithm using the result of Proposition 2. The pseudocode of this algorithm is shown in Algorithm 1.

Algorithm 1 Fixed-point Algorithm to compute the solution of (LOAD-OPT).

1: INITIALIZE traffic intensities: $\eta_1, \ldots, \eta_C$;
   capacity of the servers: $r_0, r_1, \ldots, r_C$;
   holding costs of the servers: $c_0, c_1, \ldots, c_C$;
   partition of $\mathcal{K}$: $\mathcal{C}_0$, $\mathcal{C}_d$ and $\mathcal{C}_b$
2: COMPUTE ρ_0 using $\mathcal{C}_0$, $\mathcal{C}_d$ and $\mathcal{C}_b$.
3: SET $D_0 = D_b = D_d = \emptyset$.
4: SET $\delta_j = \sqrt{\frac{c_j/r_j}{c_0/r_0}}$, $j = 1, \ldots, C$.
5: **while** $D_0 \neq \mathcal{C}_0$ or $D_b \neq \mathcal{C}_b$ or $D_d \neq \mathcal{C}_d$ **do**
6: SET $D_0 = \mathcal{C}_0$ and $D_b = \mathcal{C}_b$ and $D_d = \mathcal{C}_d$.
7: SET $\mathcal{C}_0 = \mathcal{C}_b = \mathcal{C}_d = \emptyset$.
8: **for all** $i \in \mathcal{K}$ **do**
9: **if** $\delta_i \geq \frac{1}{1-\rho_0}$ **then**
10: UPDATE $\mathcal{C}_0 = \mathcal{C}_0 \cup \{i\}$.
11: **else**
12: **if** $\delta_i > \frac{1-\frac{\eta_i}{r_i}}{1-\rho_0}$ **then**
13: UPDATE $\mathcal{C}_b = \mathcal{C}_b \cup \{i\}$.
14: **else**
15: UPDATE $\mathcal{C}_d = \mathcal{C}_d \cup \{i\}$.
16: **end if**
17: **end if**
18: **end for**
19: UPDATE ρ_0 using $\mathcal{C}_0$, $\mathcal{C}_d$ and $\mathcal{C}_b$.
20: **end while**
21: COMPUTE $\rho_1, \ldots, \rho_C$ using $\mathcal{C}_0$, $\mathcal{C}_d$ and $\mathcal{C}_b$.
22: **return** $\rho_0, \rho_1, \ldots, \rho_C$.

The main idea of this algorithm is that it starts from an initial partition $\mathcal{C}_0, \mathcal{C}_b$ and $\mathcal{C}_d$ that is used to compute ρ_0 (see Lines 2 and 19). This value of ρ_0 is then used to determine the set of classes that belong respectively to $\mathcal{C}_0$ (see Line 9-10), to $\mathcal{C}_b$ (see Line 12-13) and to $\mathcal{C}_d$ (see Line 14-15). The algorithm stops in the first iteration where $\mathcal{C}_0, \mathcal{C}_b$ and $\mathcal{C}_d$ do not change. When this occurs, according to Proposition 2, the optimal loads are obtained using (10) and (11) with the resulting partition of the algorithm. Unfortunately, we did not succeed is showing the convergence of this algorithm. However, as we will see in the numerical section, we study the converge of this algorithm and, in all the experiments we have carried out, the convergence is given in a very small number of steps.

We remark that this algorithm can be also used to analyze the economies when including a multi-skilled server into a system with C dedicated servers. For this purpose, we need to initiate the algorithm with an initial partition such that $\mathcal{C}_d = \mathcal{K}$ and $\mathcal{C}_0 = \mathcal{C}_b = \emptyset$ and with some values of $\eta_1, \ldots, \eta_C$ and $r_1, \ldots, r_C$ such that the system with only dedicated servers is stable. In that

case, the output of the algorithm will be one of the following possibilities: (i) the algorithm stops after the first iteration and (ii) the algorithm does not stop after the first iteration. In the former case, we can conclude that it is not beneficial to add a multi-skilled server, whereas in the latter one we can compare the cost at the initial state and the cost when the algorithm stops to compare the performance of both systems.

5 Performance Analysis

We now determine the scenarios in which it is profitable to either train or hire multi-skilled agents. For this, we compare the value of the objective function when there are only dedicated servers to that with also a multi-skilled one, as well as the case in which all the servers are multi-skilled and can serve all the jobs.

5.1 Comparison with All Full-Skilled Servers System

First, we compare the cost of the model with C dedicated servers and a single full-skilled server with the cost a system formed by $C+1$ servers with the same values of the holding costs and capacities as Server 0, but all the servers can serve jobs of all the classes. We call the latter model ASSAC (All Servers Serve All Classes).

Lemma 2. *Consider that $\eta_j \to 0$ for all $j > 1$ and $\frac{\delta_1}{1-\frac{\eta_1}{r_1}} < 1$. Then, the cost of the system with dedicated servers is δ_1^2 times smaller than the cost of the ASSAC model when η_1 is small enough.*

Proof. In the system with dedicated servers, when $\frac{\delta_1}{1-\frac{\eta_1}{r_1}} < 1$ and $\eta_j \to 0$ for all $j > 1$, all the jobs of class 1 are executed in Server 1 and the load of the rest of the servers is zero. Hence, the cost of this system is

$$\sum_{j\in\mathcal{S}} c_j \frac{\rho_j}{1-\rho_j} = c_1 \frac{\frac{\eta_1}{r_1}}{1-\frac{\eta_1}{r_1}}. \tag{12}$$

In the ASSAC model, the traffic is uniformly shared among all the servers and, therefore, the cost of this system when $\frac{\delta_1}{1-\frac{\eta_1}{r_1}} < 1$ and $\eta_j \to 0$ for all $j > 1$ is

$$\sum_{j\in\mathcal{S}} c_j \frac{\rho_j}{1-\rho_j} = c_0 \frac{\frac{\eta_1}{r_0}}{1-\frac{\eta_1}{(C+1)r_0}}. \tag{13}$$

When η_1 is small enough, (12) and (13) are approximately $\frac{c_1\eta_1}{r_1}$ and $\frac{c_0\eta_1}{r_0}$, respectively. And the desired result thus follows since ratio of the former and the latter is δ_1^2. □

From the above lemma, we have that the optimal cost of the system with dedicated servers is smaller than that of ASSAC in the considered regime.

Proposition 6. *Consider that $\eta_j \to 0$ for all $j > 1$ and $\frac{\delta_1}{1-\frac{\eta_1}{r_1}} < 1$. Then, the optimal cost of* (LOAD-OPT) *is smaller than the optimal cost of ASSAC when η_1 is small enough.*

We now show that the optimal cost of (LOAD-OPT) can be larger than that of ASSAC. The intuition behind this result is that the stability region of the ASSAC model is wider than the stability region for the model with one dedicated server. By taking the load close to the boundary of the stability region of the model with one dedicated server, the cost can be made to go infinity. For the ASSAC model, however, the availability of spare capacity means that the cost remains finite.

Proposition 7. *Consider that $\eta_j \to 0$ for all $j = 2, \ldots, C$ and $\eta_1 \to r_0 + r_1$. Then, the optimal cost of ASSAC is smaller than the optimal cost of* (LOAD-OPT)*.*

Proof. We first observe that the cost of ASSAC when $\eta_j \to 0$ for all $j > 1$ and and $\eta_1 \to r_0 + r_1$ is given by

$$c_0(C+1)\frac{\frac{r_0+r_1}{(C+1)r_0}}{1-\frac{r_0+r_1}{(C+1)r_0}} = c_0\frac{\frac{r_0+r_1}{r_0}}{1-\frac{r_0+r_1}{(C+1)r_0}},$$

which is clearly finite.

However, for the model with dedicated servers, class-1 jobs are served by Server 0 and Server 1, whose load tends to one when $\eta_1 \to r_0 + r_1$. Therefore, its cost tends to infinity. □

From the above propositions, it follows the existence of a switching curve when $\delta_1 < 1$. In Section 6, we study numerically this curve.

5.2 Comparison with No Sharing System

We consider a system formed by $C+1$ dedicated servers, but Server 0 can only serve jobs of class 1, whereas for $i \geq 2$ Server i can serve only jobs of class i. We call this model as system without sharing since jobs of different classes are not served in the same server. We compare the optimal cost of this system with the optimal cost of (LOAD-OPT).

In Proposition 7, we have shown that the optimal cost of (LOAD-OPT) tends to infinity when $\eta_1 \to r_0 + r_1$ and $\eta_j \to 0$ for $j = 2, \ldots, C$, whereas the optimal cost of ASSAC is finite. In the following result, we show that there is a regime where the optimal cost of the system without sharing is infinity, where the optimal cost of (LOAD-OPT) is finite. The intuition is similar here when we note that the stability region of the no-sharing model is included in that of the model with one shared server.

Proposition 8. *Consider that $\eta_j \to 0$ for all $j = 1, \ldots, C-1$ and $\eta_C \to r_C$. Then, the optimal cost of the system without sharing is larger than the optimal cost of* (LOAD-OPT)*.*

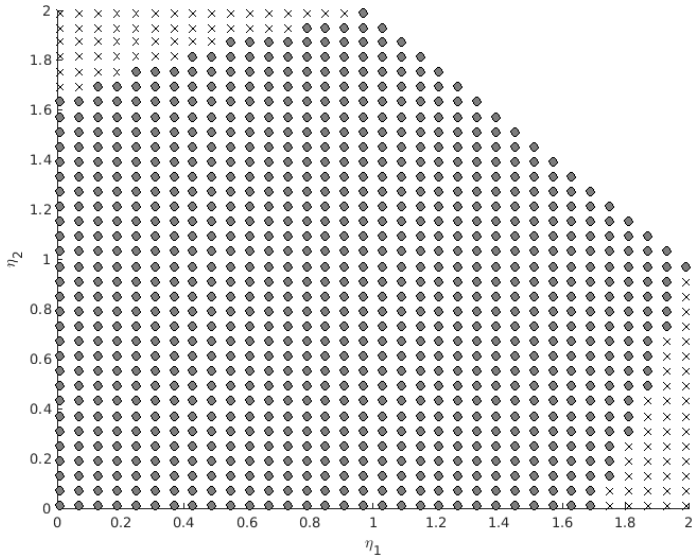

Figure 3: Comparison of optimal costs of Section 5.1 when $\delta_i < 1$ for $i = 1, 2$.

6 Numerical Experiments

In this section, we present the numerical experiments that complement the main theoretical findings of this article.

6.1 Comparison with ASSAC

We first focus on the performance comparison of Section 5.1, where we showed the existence of a switching curve when $\delta_1 < 1$. For $C = 2$, we analyze the value of the objetive function of the models under comparison in Section 5.1, which are the M model (i.e., the model we consider in Section 4.1) and the ASSAC model with three servers. In Fig. 3, we fix the values of the capacities and holding costs and we consider η_1 and η_2 such that both models are stable, that is, when $\eta_i < r_0 + r_i$, for $i = 1, 2$ and $\eta_1 + \eta_2 < r_1 + r_2 + r_0$. We set $r_1 = r_2 = r_0 = 1$ and $c_0 = 20$, $c_1 = 1$ and $c_2 = 2$. We represent with 'x' where the cost of the ASSAC model is smaller and with a filled 'o' the region where the value of the objective function of the M model is smaller. As it can be observed in Fig. 3, the switching curve is unique, that is, when we increase η_1 (or η_2) there is a single value where the model that outperforms changes. Another interesting conclusion of this experiment is that the region where the M model outperforms is very large.

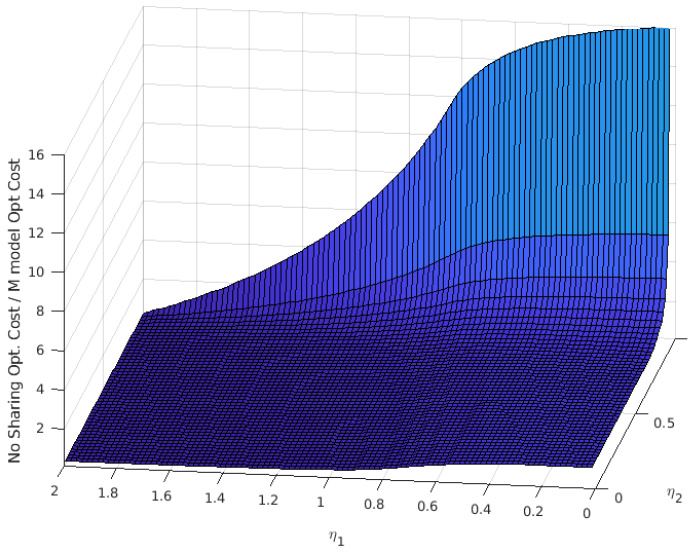

Figure 4: Ratio of the optimal costs under comparison in Section 5.2 ($\eta_1 \in (0, 2)$ and $\eta_2 \in (0, 1)$).

6.2 Comparison with no-sharing system

In the next set of experiments, we concentrate on the performance comparison of Section 5.2. We compare the value of the objective function of both models for the values of η_1 and η_2 such that both systems are stable, i.e., $\eta_1 < r_0 + r_1$ and $\eta_2 < r_2$ considering the same values of the parameters as in Fig. 3. As we said in Section 5.2, the stability region of the system without sharing is smaller than that of the M model. This implies that M model outperforms the system without sharing when $\eta_2 \to 1$. This phenomenon can be clearly observed in Fig. 4. We are also interested in comparing these models out of the boundary. For this purpose, we present in Fig. 5 a zoomed version of Fig. 4. From this illustration, we conclude that the performance of both models is very similar when $\eta_1 \in (0, 2)$ and $\eta_2 \in (0, 0.8)$.

6.3 The solution of (LOAD-OPT) for $C > 2$

We now study the Algorithm 1 since, as we said before, its convergence ensures that the solution of (LOAD-OPT) is obtained. We first consider a system with $C = 5$ classes of traffic and the values of the parameters presented in Table 1. We have chosen these parameters since the solution of (LOAD-OPT) for these values satisfies that $\mathcal{C}_d = \{3\}$, $\mathcal{C}_0 = \{4\}$ and $\mathcal{C}_b = \{1, 2, 5\}$, i.e., all the sets of the partition are non-empty.

We consider three different initial conditions: first, all the classes belong to C_d (see solid line in Figure 6); second, classes 1, 3 and 4 belong to C_d, whereas

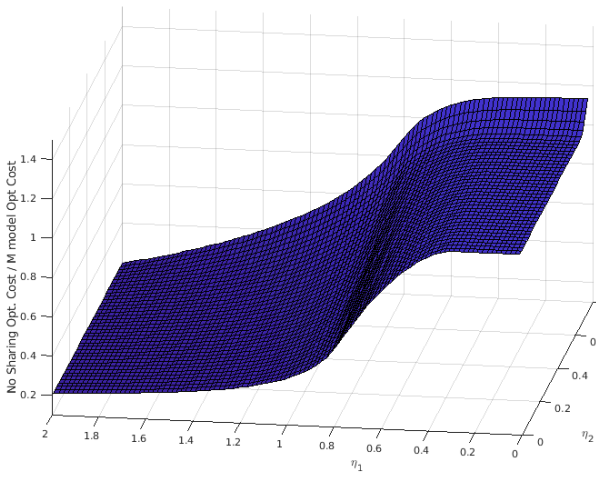

Figure 5: Ratio of the optimal costs under comparison in Section 5.2 ($\eta_1 \in (0, 2)$ and $\eta_2 \in (0, 0.8)$).

	$j=0$	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$
r_j	25	34	38	39	13	27
c_j	5	10	11	6	86	14
η_j		28	33	20	12	24

Table 1: Parameters of the system considered in Section 6.3.

classes 2 and 5 to C_b (see dashed line in Figure 6); and finally, classes 1, 2, 3 and 4 belong to C_d and class 5 to C_b (see dotted line in Figure 6).

We illustrate in Figure 6 the evolution of the loads of each server over the iterations of the algorithm. In the upper line of Figure 6 we show the loads of Server 0, Server 1 and Server 2, whereas in the bottom line the loads of Server 3, Server 4 and Server 5. We observe that, when the algorithm converges, the load of Server 4 is zero, which means that, for this case, $\rho_4^* = 0$. We also see that, for Server 3, the initial load in the scenario that is represented by the solid line (that is, the scenario where all the classes send all the traffic to its dedicated server) equals to the load when the algorithm converges. This means that, for class 3, we have that $\rho_3^* = \frac{\eta_3}{r_3}$.

It is important to remark that, as we can also observe in Figure 6, the algorithm converges to the same values for the three different initial partitions under consideration. We have also started the system with other initial partitions and the obtained results confirmed that the algorithm always converges. Another interesting property of this algorithm is that the number of iterations required to reach the convergence is very small. Indeed, when the initial partition of $\mathcal{K}$ is such that all the classes belong to C_d, the algorithm converges after 12 iterations. Moreover, for the rest of the cases, the algorithm converges for a less

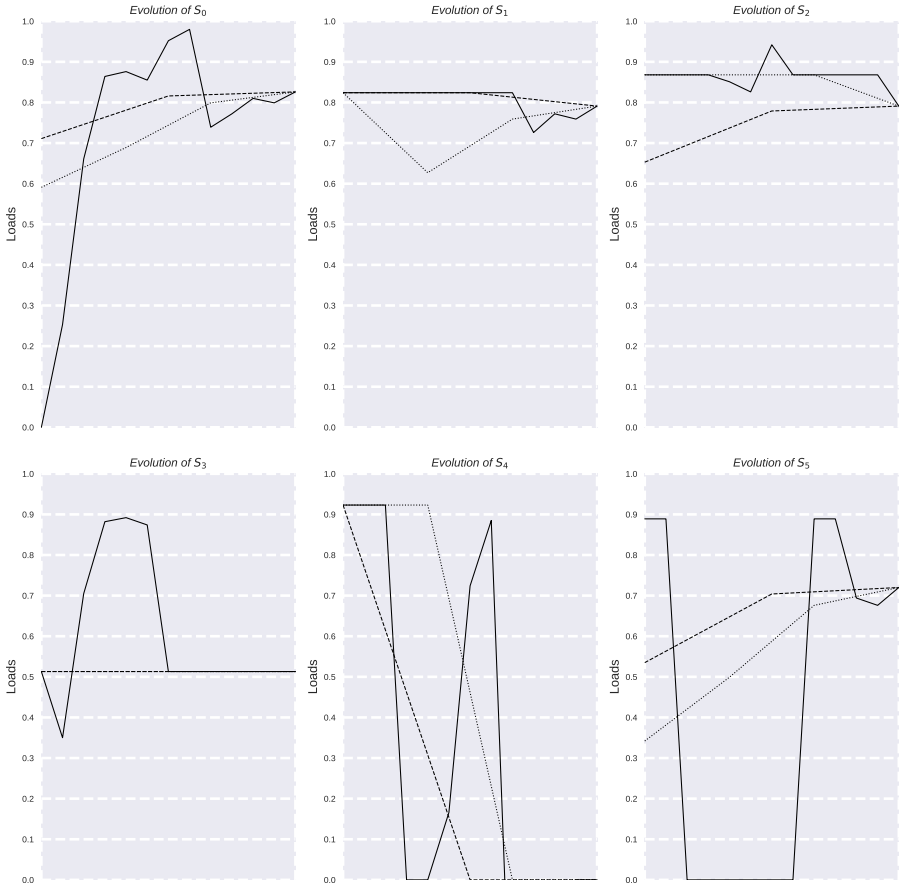

Figure 6: Convergence of the fixed-point algorithm presented in Section 4.2. The x-axis represents the iterations of the algorithm and the y-axis the load of each server.

number of iterations. When the initial partition of $\mathcal{K}$ is such that classes 1, 3 and 4 belong to C_d and classes 2 and 5 to C_b, it converges after 2 iterations and when the initial partition of $\mathcal{K}$ is such that classes 1, 2, 3 and 4 belong to C_d and class 5 to C_b, it converges after 3 iterations.

We now present further numerical work we have performed to analyze the convergence of Algorithm 1 for larger systems. For this set of experiments, we consider that the number of dedicated servers, C, varies from 10 to 200 with step 10. For each case we run our algorithm 10 times where, in each run, the parameters of the system are randomly chosen (but satisfying the stability condition); the results are depicted in Figure 7, where the blue bars represent the minimum number of iterations required for convergence and the yellow bars

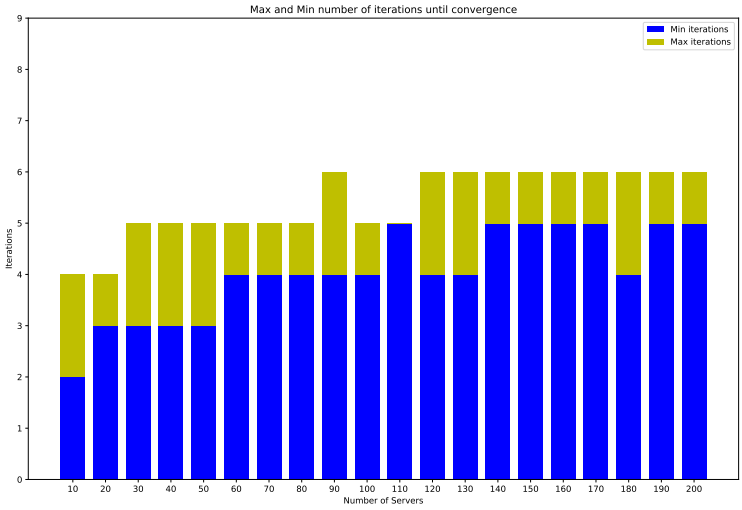

Figure 7: Minimum and maximum number of iterations until convergence for systems of different size.

the difference up to the maximum. The main conclusions of these experiments are twofold: first, we observe that the algorithm converges in all the cases; and second, that the number of iterations required to converge varies between 2 and 6 in all the cases. This means that the convergence of this algorithm is very fast even for large systems with 200 dedicated servers.

7 Conclusions

We study the optimal Bernoulli routing in a system with C dedicated servers and a single multi-skilled server. We first provide a necessary and sufficient condition for the stability of the system. We then reformulate this problem as a optimization problem in terms of the loads of the system and we show the equivalence of both problems. We provide structural properties on the solution of the derived problem, which allows us to fully characterize the optimal loads of the system when $C = 2$ and also to present a fixed point algorithm whose convergence ensure that the optimal loads are obtained. We compare the performance of this system with optimal loads with a system where all the servers are multi-skilled and also with a system where all the servers are dedicated. Finally, we explore numerically the convergence of the fixed point algorithm and show that, in all the considered cases, the algorithm converges in a very few number of steps.

For future work, we are interested in generalizing the results of this article to systems with a more complex topology. Besides, we think that an interesting extension of the performance analysis of this work would be to consider other popular load balancing policies such as Power of Two and Join the Shortest Queue.

References

[1] I. Taboada, S. Aalto, P. Lassila, F. Liberal, *Delay and energy-aware load balancing in ultra-dense heterogeneous 5G networks*, Transactions on Emerging Telecommunications Technologies 28 (9) (2017) e3170.

[2] E. Altman, U. Ayesta, B. J. Prabhu, *Load balancing in processor sharing systems*, Telecommunication Systems 47 (1-2) (2011) 35–48.

[3] B. Gaujal, E. Hyon, A. Jean-Marie, *Optimal Routing in Two Parallel Queues with Exponential Service Times*, Discrete Event Dynamic Systems 16 (1) (2006) 71–107. doi:10.1007/s10626-006-6179-3.

[4] M. Mitzenmacher, *The Power of Two Choices in Randomized Load Balancing*, IEEE Trans. Parallel Distrib. Syst. 12 (10) (2001) 10941104.

[5] N. D. Vvedenskaya, R. L. Dobrushin, F. I. Karpelevich, *Queueing system with selection of the shortest of two queues: An asymptotic approach*, Problems of Information Transmission 32 (1) (1996) 15–27.

[6] C. Graham, *Chaoticity on path space for a queueing network with selection of the shortest queue among several*, J. Appl. Probab. 37 (1) (2000) 198–211. doi:10.1239/jap/1014842277.

[7] Y. Lu, Q. Xie, G. Kliot, A. Geller, J. R. Larus, A. Greenberg, *Join-Idle-Queue: A Novel Load Balancing Algorithm for Dynamically Scalable Web Services*, Perform. Eval. 68 (11) (2011) 1056–1071.

[8] M. Harchol-Balter, M. E. Crovella, C. D. Murta, *On choosing a task assignment policy for a distributed server system*, Journal of Parallel and Distributed Computing 59 (2) (1999) 204–228.

[9] H. Feng, V. Misra, D. Rubenstein, *Optimal state-free, size-aware dispatching for heterogeneous M/G/-type systems*, Performance Evaluation 62 (1-4) (2005) 475–492.

[10] M. Harchol-Balter, *Task assignment with unknown duration*, in: Proceedings 20th IEEE International Conference on Distributed Computing Systems, IEEE, 2000, pp. 214–224.

[11] J.-M. Fourneau, *Modeling green data-centers and jobs balancing with energy packet networks and interrupted Poisson energy arrivals*, SN Computer Science 1 (1) (2020) 28.

[12] Z. Liu, M. Lin, A. Wierman, S. Low, L. L. H. Andrew, *Greening Geographical Load Balancing*, IEEE/ACM Transactions on Networking 23 (2) (2015) 657–671.

[13] G. Koole, A. Pot, J. Talim, *Routing heuristics for multi-skill call centers*, Vol. 2, 2003, pp. 1813–1816.

[14] R. B. Wallace, W. Whitt, *A Staffing Algorithm for Call Centers with Skill-Based Routing*, Manufacturing & Service Operations Management 7 (4) (2005) 276294.

[15] J. Chen, J. Do, P. Shi, *A survey on skill-based routing with applications to service operations management*, Queueing Systems 96 (2020) 53–82.

[16] K. Gardner, S. Zbarsky, S. Doroudi, M. Harchol-Balter, E. Hyytia, *Reducing Latency via Redundant Requests: Exact Analysis*, in: Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS 15, Association for Computing Machinery, New York, NY, USA, 2015, p. 347360.

[17] T. Bonald, C. Comte, F. Mathieu, *Performance of Balanced Fairness in Resource Pools: A Recursive Approach*, Proc. ACM Meas. Anal. Comput. Syst. 1 (2) (2017). doi:10.1145/3154500.

A Proof of Proposition 1

We first show that if there exists a subset $A \subset \mathcal{K}$ such that $\sum_{i\in A} \eta_i > r_0 + \sum_{i\in A} r_i$, then the system is not stable.

We know from (1) and (2) that

$$\begin{aligned} r_0\rho_0 + \sum_{i\in A} r_i\rho_i &= \sum_{i\in\mathcal{K}} \eta_i(1-p_i) + \sum_{i\in A} \eta_i(1-p_i) \\ &= \sum_{i\in A} \eta_i + \sum_{i\in\mathcal{K}\setminus A} \eta_i(1-p_i) \\ &\geq \sum_{i\in A} \eta_i \\ &> r_0 + \sum_{i\in A} r_i. \end{aligned}$$

Therefore, we have obtained that $r_0\rho_0 + \sum_{i\in A} r_i\rho_i > r_0 + \sum_{i\in A} r_i$, which requires that, at least, the load of one server is larger than one, i.e., that the system is not stable.

Let $\epsilon > 0$ small. We now show that, if (6) holds, then the system is stable. For this purpose, we define the following routing strategy: for all $i \in \mathcal{K}$ such that $\eta_i < r_i$, $p_i = 1 - \epsilon$ and for all $i \in \mathcal{K}$ such that $\eta_i \geq r_i$, $p_i = \frac{r_i}{\eta_i}(1-\epsilon)$. For

this choice, it is clear that $\rho_j < 1$ for all $j \in \mathcal{K}$. We now focus on Server 0 and we aim to show that

$$\sum_{i\in\mathcal{K}} \eta_i(1-p_i) < r_0.$$

We denote by $\mathcal{K}^*$ the set of classes such that $\eta_i \geq r_i$. Hence, the above expression is satisfied if and only if

$$\sum_{i\in\mathcal{K}\setminus\mathcal{K}^*} \eta_i\epsilon + \sum_{i\in\mathcal{K}^*} \eta_i(1-p_i) < r_0$$

Hence,

$$\begin{aligned}
&\sum_{i\in\mathcal{K}\setminus\mathcal{K}^*} \eta_i\epsilon + \sum_{i\in\mathcal{K}^*} \eta_i\left(1-\frac{r_i}{\eta_i}(1-\epsilon)\right) < r_0 \iff \\
&\sum_{i\in\mathcal{K}\setminus\mathcal{K}^*} \eta_i\epsilon + \sum_{i\in\mathcal{K}^*} (\eta_i - r_i(1-\epsilon)) < r_0 \iff \\
&\sum_{i\in\mathcal{K}\setminus\mathcal{K}^*} \eta_i\epsilon + \sum_{i\in\mathcal{K}^*} (\eta_i - r_i + r_i\epsilon) < r_0
\end{aligned}$$

We know from (6) that $\sum_{i\in\mathcal{K}^*}(\eta_i - r_i) < r_0$ and, therefore, the above inequality is satisfied if and only if

$$\epsilon\left(\sum_{i\in\mathcal{K}\setminus\mathcal{K}^*} \eta_i + \sum_{i\in\mathcal{K}^*} r_i\right) < r_0 - \sum_{i\in\mathcal{K}^*}(\eta_i - r_i).$$

In other words, the desired result follows if we choose $\epsilon > 0$ such that

$$\epsilon < \frac{r_0 - \sum_{i\in\mathcal{K}^*}(\eta_i - r_i)}{\left(\sum_{i\in\mathcal{K}\setminus\mathcal{K}^*} \eta_i + \sum_{i\in\mathcal{K}^*} r_i\right)}.$$

B Proof of Proposition 2

In the following result, we provide a property that will be useful to show the result of Proposition 2.

Lemma 3. *Let $\tilde{C} \subseteq \mathcal{K}$. Then,*

$$\sum_{i\in\tilde{C}} \eta_i = r_0\rho_0 + \sum_{j\in\tilde{C}} r_j\rho_j \iff C_b \cup C_0 \subseteq \tilde{C}.$$

Proof. To simplify the notation, we write $D = C_b \cup C_0$. If $D = \emptyset$, then $\rho_0 = 0$ and $\rho_j = \eta_j/r_j$ for $j = 1, 2, \ldots, C$, which implies clearly that $\sum_{i\in A} \eta_i = \sum_{j\in S_A} \rho_j r_j$ for all $A \subset \mathcal{K}$.

We now focus on the case $D \neq \emptyset$. We know that

$$\rho_j < \eta_j/r_j,\ \forall j \in D \qquad \text{and} \qquad \rho_j = \eta_j/r_j,\ \forall j \in \mathcal{K} \setminus D. \tag{14}$$

We now observe that $\mathcal{K} = D \bigcup (\mathcal{K} \setminus D)$ and therefore from (7)

$$\sum_{i \in D} \eta_i + \sum_{i \in \mathcal{K} \setminus D} \eta_i = r_0\rho_0 + \sum_{i \in D} r_i\rho_i + \sum_{i \in \mathcal{K} \setminus D} r_i\rho_i.$$

From $\rho_j = \eta_j/r_j,\ \forall j \in \mathcal{K} \setminus D$, it follows that

$$\sum_{i \in D} \eta_i + \sum_{i \in A} \eta_i = r_0\rho_0 + \sum_{j \in D} r_j\rho_j + \sum_{j \in A} r_j\rho_j, \quad \forall A \subseteq \mathcal{K} \setminus D. \tag{15}$$

Therefore, for any $\tilde{C} \subseteq \mathcal{K}$ such that $D \subseteq \tilde{C}$ the constraint (9) is satisfied as an equality. Besides, we now show that for any subset that does not contain D, the constraint (9) is satisfied as an inequality. For all $B \subset D$, (15) can be written as follows:

$$\sum_{i \in B} \eta_i + \sum_{i \in D \setminus B} \eta_i + \sum_{i \in A} \eta_i = r_0\rho_0 + \sum_{j \in B} r_j\rho_j + \sum_{j \in D \setminus B} r_j\rho_j + \sum_{j \in A} r_j\rho_j, \quad \forall A \subseteq \mathcal{K} \setminus D,$$

which, by $\rho_j < \eta_j/r_j\ \forall j \in D$, gives that

$$\begin{aligned}
\sum_{i \in B} \eta_i + \sum_{i \in A} \eta_i =& r_0\rho_0 + \sum_{j \in B} r_j\rho_j + \Big(\sum_{j \in D \setminus B} r_j\rho_j - \sum_{i \in D \setminus B} \eta_i \Big) \\
&+ \sum_{j \in A} r_j\rho_j \\
<& r_0\rho_0 + \sum_{j \in B} r_j\rho_j + \sum_{j \in A} r_j\rho_j, \quad \forall A \subseteq \mathcal{K} \setminus D.
\end{aligned}$$

And the desired result follows. □

We now prove the result of Proposition 2.

Proof. The Lagrangian corresponding to (LOAD-OPT) is

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\rho}, \boldsymbol{\nu}, \boldsymbol{\zeta}, \gamma, \boldsymbol{\xi}) =& \frac{c_0\rho_0}{1-\rho_0} + \sum_{j=1}^{C} \frac{c_j\rho_j}{1-\rho_j} + \sum_{j=0}^{C} \nu_j(-\rho_j) + \sum_{j=0}^{C} \zeta_j(\rho_j - 1) \\
&+ \gamma\Big(\sum_{i=1}^{C} \eta_i - r_0\rho_0 - \sum_{j=1}^{C} r_j\rho_j \Big) \\
&+ \sum_{\substack{A \subseteq \mathcal{K} \\ A \neq \emptyset}} \xi_A \Big(\sum_{i \in A} \eta_i - r_0\rho_0 - \sum_{j \in A} r_j\rho_j \Big)
\end{aligned}$$

Given that the optimization problem is convex, $\boldsymbol{\rho}^*, \boldsymbol{\nu}^*, \boldsymbol{\zeta}^*, \gamma^*, \boldsymbol{\xi}^*$ is a solution of (LOAD-OPT) if it satisfies Karush-Kuhn-Tucker conditions:

$$0 \leq \rho_j^* \leq 1, \quad \forall j = 0, 1, \ldots, C \tag{16}$$

$$\frac{c_0}{(1-\rho_0^*)^2} - \nu_0^* + \zeta_0^* - r_0\big(\gamma^* + \sum_{\substack{A \subset \mathcal{K} \\ A \neq \emptyset}} \xi_A^*\big) = 0 \tag{17}$$

$$\frac{c_j}{(1-\rho_j^*)^2} - \nu_j^* + \zeta_j^* - r_j\big(\gamma^* + \sum_{\substack{A \subset \mathcal{K} \\ j \in A}} \xi_A^*\big) = 0, \quad \forall j = 1, 2, \ldots, C \tag{18}$$

$$\nu_j^* \geq 0,\ \zeta_j^* \geq 0,\ \gamma^* \in \mathbb{R},\ \xi_A^* \geq 0, \quad \forall j = 0, 1, \ldots, C,\ \forall A \subset \mathcal{K}, A \neq \emptyset \tag{19}$$

$$\nu_j^* \rho_j^* = 0,\ \zeta_j^*(\rho_j^* - 1) = 0, \quad \forall j = 0, 1, \ldots, C \tag{20}$$

$$\sum_{i=1}^{C} \eta_i = r_0 \rho_0^* + \sum_{j=1}^{C} r_j \rho_j^* \tag{21}$$

$$\sum_{i \in A} \eta_i \leq r_0 \rho_0^* + \sum_{j \in A} r_j \rho_j^*, \quad \forall A \subset \mathcal{K}, A \neq \emptyset \tag{22}$$

$$\xi_A^*\Big(\sum_{i \in A} \eta_i - r_0 \rho_0^* - \sum_{j \in A} r_j \rho_j^*\Big) = 0, \quad \forall A \subset \mathcal{K}, A \neq \emptyset. \tag{23}$$

We observe that the objective function tends to infinity when $\rho_j \to 1$, which implies that $\rho_j^* < 1, \forall j = 0, 1, \ldots, C$ and, as a consequence of this and from (20), $\zeta_j^* = 0,\ \forall j = 0, 1, \ldots, C$. Furthermore, from Lemma 3 and (23), we conclude that $\forall A \in \mathcal{K}$ that does not contain $C_b \cup C_0$, its multiplier verifies that $\xi_A^* = 0$, because for those subsets the constraint (9) is satisfied as an inequality.

For Server 0, we know that $\rho_0^* = 0$ if $C_0 \cup C_b = \emptyset$ and $\rho_0 > 0$ otherwise. This clearly implies that $\nu_0^* \geq 0$ if $C_0 \cup C_b = \emptyset$ and $\nu_0^* = 0$ otherwise. For all $j \in \mathcal{K}$, we know that $\rho_j^* = \eta_j / r_j$ if $j \in C_d$, whereas $\rho_j^* < \eta_j / r_j$ otherwise. This clearly implies that $\nu_j^* = 0$ if $j \in C_d$. We also know that $\nu_j^* = 0$ if $j \in C_b$ because, in this case, $\rho_j^* > 0$, whereas if $j \in C_0$, we have that $\nu_j^* \geq 0$.

We first prove this result when $C_b \cup C_0 = \emptyset$. For this case, the load of Server 0 is zero and, thus, it is enough to show that $\delta_j < 1 - \eta_j / r_j$. From (17) and (18), we get that

$$c_0 - \nu_0^* - r_0\big(\gamma^* + \sum_{\substack{A \subset \mathcal{K} \\ A \neq \emptyset}} \xi_A^*\big) = 0 \tag{24}$$

$$\frac{c_j}{(1-\eta_j/r_j)^2} - r_j\big(\gamma^* + \sum_{\substack{A \subset \mathcal{K} \\ j \in A}} \xi_A^*\big) = 0, \quad \forall j = 1, 2, \ldots, C. \tag{25}$$

From (24) and since $\nu_0^* \geq 0$, it results that

$$\nu_0^* = c_0 - r_0\big(\gamma^* + \sum_{\substack{A \subset \mathcal{K} \\ A \neq \emptyset}} \xi_A^*\big) \geq 0 \iff \frac{c_0}{r_0} \geq \gamma^* + \sum_{\substack{A \subset \mathcal{K} \\ A \neq \emptyset}} \xi_A^*,$$

From (25), we obtain that

$$\gamma^* + \sum_{\substack{A\subset\mathcal{K}\\ j\in A}} \xi_A^* = \frac{c_j}{r_j}\frac{1}{(1-\eta_j/r_j)^2}, \quad \forall j = 1,2,\ldots,C.$$

Therefore,

$$\frac{c_j}{r_j}\frac{1}{(1-\eta_j/r_j)^2} = \gamma^* + \sum_{\substack{A\subset\mathcal{K}\\ j\in A}} \xi_A^* \leq \gamma^* + \sum_{\substack{A\subset\mathcal{K}\\ A\neq\emptyset}} \xi_A^* \leq \frac{c_0}{r_0},$$

which gives the desired condition, i.e., $\delta_j \leq 1-\eta_j/r_j, \quad \forall j = 1,2,\ldots,C.$

We focus on the case $C_0 \neq \emptyset$ or $C_b \neq \emptyset$. We note that (17) and (18) can be written as follows:

$$\frac{c_0}{(1-\rho_0^*)^2} - r_0\big(\gamma^* + \sum_{\substack{A\subset\mathcal{K}\\ C_b\cup C_0\subseteq A}} \xi_A^*\big) = 0 \tag{26}$$

$$c_j - \nu_j^* - r_j\big(\gamma^* + \sum_{\substack{A\subset\mathcal{K}\\ C_b\cup C_0\subseteq A}} \xi_A^*\big) = 0, \quad \forall j \in C_0 \tag{27}$$

$$\frac{c_j}{(1-\eta_j/r_j)^2} - r_j\big(\gamma^* + \sum_{\substack{A\subset\mathcal{K}\\ C_b\cup C_0\subseteq A\\ j\in A}} \xi_A^*\big) = 0, \quad \forall j \in C_d. \tag{28}$$

$$\frac{c_j}{(1-\rho_j^*)^2} - r_j\big(\gamma^* + \sum_{\substack{A\subset\mathcal{K}\\ C_b\cup C_0\subseteq A\\ j\in A}} \xi_A^*\big) = 0, \quad \forall j \in C_b. \tag{29}$$

We aim to show that, for all $j \in C_0$, $\delta_j \geq \frac{1}{1-\rho_0^*}$, and for all $j \in C_b$ $\frac{1}{1-\rho_0^*} > \delta_j > \frac{1-\eta_j/r_j}{1-\rho_0^*}$.

For the first condition, we observe that from (26) and (27), it follows that, for all $j \in C_0$,

$$\gamma^* + \sum_{\substack{A\subset\mathcal{K}\\ C_b\subseteq A}} \xi_A^* = \frac{c_0}{r_0}\frac{1}{(1-\rho_0^*)^2} = \frac{c_j}{r_j} - \frac{\nu_j^*}{r_j},$$

which, using that $\nu_j^* \geq 0$, gives that

$$\delta_j \geq \frac{1}{1-\rho_0^*}.$$

We now show the second condition, i.e., $\frac{1}{1-\rho_0^*} > \delta_j > \frac{1-\eta_j/r_j}{1-\rho_0^*}$ for all $j \in C_b$.

From (26) and (29), it follows that, for all $j \in C_b$,

$$\gamma^* + \sum_{\substack{A \subset \mathcal{K} \\ C_b \cup C_0 \subseteq A}} \xi_A^* = \frac{c_0}{r_0} \frac{1}{(1-\rho_0^*)^2} = \frac{c_j}{r_j} \frac{1}{(1-\rho_j^*)^2} \tag{30}$$

$$\iff 0 = \frac{c_j}{r_j} \frac{1}{(1-\rho_j^*)^2} - \frac{c_0}{r_0} \frac{1}{(1-\rho_0^*)^2} \tag{31}$$

$$\iff \delta_j = \frac{1-\rho_j^*}{1-\rho_0^*}, \tag{32}$$

which gives that

$$\rho_j^* = 1 - \delta_j(1-\rho_0^*),$$

as desired.

Using the last expression and that, for $j \in C_b$, $0 < \rho_j^* < \eta_j/r_j$, the desired result follows, i.e.,

$$\frac{1}{1-\rho_0^*} > \delta_j = \frac{1-\rho_j^*}{1-\rho_0^*} > \frac{1-\eta_j/r_j}{1-\rho_0^*}.$$

To finish, we compute the loads of all the servers. First, for $j \in C_d$, we have clearly that $\rho_j^* = \frac{\eta_j}{r_j}$. Besides, we use that for all $j \in C_b$, $\rho_j^* = 1 - \delta_j(1-\rho_0^*)$, and from the expression (7), it follows that

$$\sum_{i=1}^{C} \eta_i = r_0\rho_0^* + \sum_{j=1}^{C} r_j\rho_j^* \iff$$

$$\sum_{i=1}^{C} \eta_i = r_0\rho_0^* + \sum_{j \in C_b} r_j(1-\delta_j(1-\rho_0^*)) + \sum_{i \in C_d} \eta_i.$$

And rearranging both sides of the above expression, we obtain that

$$\rho_0^* = 1 - \frac{r_0 + \sum_{j \in C_b} r_j - \sum_{i \in C_b \cup C_0} \eta_i}{r_0 + \sum_{j \in C_b} \delta_j r_j}.$$

And the desired result follows. □