A Simple Heuristic for Load Balancing in Parallel Processing Networks with Highly Variable Service Time Distributions

Luz A. Caudillo-Fuentes
Department of Industrial and Operations Engineering
University of Michigan
1205 Beal Avenue, Ann Arbor, MI 48109-2117
fuentesl@umich.edu

David L. Kaufman
Department of Industrial Engineering
University of Pittsburgh
1048 Benedum Hall, Pittsburgh, PA 15261
dlk29@pitt.edu

Mark E. Lewis
School of Operations Research and Information Engineering
Cornell University
226 Rhodes Hall, Ithaca, NY 14853
mel47@cornell.edu

November 15, 2009

Abstract

Suppose that customers arrive to a service center (call center, web server, etc.) with two stations in accordance with independent Poisson processes. Service times at either station follow the same general distribution, are independent of each other and are independent of the arrival process. The system is charged station dependent holding costs at each station per customer per unit time. At any point in time, a decision-maker may decide to move, at a cost, some number of jobs in one queue to the other. The goals of this paper are twofold. First, we are interested in providing insights into this decision-making scenario. We do so, in the important case that the service time distribution is highly variable or simply has a heavy tail. Second, we propose that the savvy use of Markov decision processes can lead to easily implementable heuristics when features of the service time distribution can be captured by introducing multiple customer classes. To this end, we consider a two-station proxy for the original system, where the service times are assumed to be exponential, but of one of two classes with different rates. We prove structural results for this proxy and show that these results lead to heuristics that perform well.

1 Introduction

Suppose that customers arrive to a service center (call center, web server, etc.) with two stations in accordance with independent Poisson processes. Service times at either station follow the same general distribution, are independent of each other and are independent of the arrival process. The system is charged station dependent holding costs at each station per customer per unit time. At any point in time, a decision-maker may decide to move (or pass) some number of jobs in one queue to the other. It should be clear that the decision-maker's choice of the number of customers to move should depend on the number of customers at each station, the cost to move customers, the time elapsed since the service times of customers currently being processed by the server began and perhaps the number of future customers (s)he expects to arrive in the coming moments. With the exception of the elapsed service time information the control decisions seem ripe for an analysis via Markov decision processes (MDPs). Unfortunately, it is just that part of the state space that makes it uncountable and therefore intractable.

The goals of this paper are twofold. First, we are interested in providing insights into the above decision-making scenario. We do so, in the important case that the service time distribution is highly variable or simply has a heavy (non-exponential) tail. Second, we propose that the savvy use of Markov decision processes can lead to easily implementable heuristics when features of the service time distribution can be captured by introducing multiple customer classes.

The limitations of MDPs are well-known. As long as the state and action space descriptions (called the graph of the MDP) are multi-dimensional or consist of a large number of elements, solving the dynamic program quickly becomes intractable. In order to alleviate this problem there have been significant lines of research that study the structure of optimal policies in such areas as control of queues, manufacturing, transportation, inventory control and revenue management. For example, in the aforementioned model suppose the service time distribution is exponential. The state space is then two-dimensional. If the optimal policy can be described by a *monotone switching curve* the search for the optimal policy is reduced to finding the curve, rather than enumerating the state and action pairs throughout the decision space. Unfortunately, even in simple cases finding a structured optimal policy when the service time is generally distributed may be intractable. Consider an admission controlled $M/G/1$ queue that is used to model routing in a simple manufacturing system. If the service time distribution is exponential (so the system is an $M/M/1$), it is well-known that the optimal control policy is of *control limit* form. If the service times are generally distributed, then when a new customer arrives, the decision-maker must once again consider the time since the last service completion; the state space is uncountable. In this case, even reasonably sized discretizations of the time dimension lead to an intractable problem.

In reality, when the service time distribution is general, past experience gives the decision-maker significant information about its form. In this paper we present a heuristic that uses a multi-class queueing network

with exponentially distributed service times as a proxy for a problem with general service time distributions. In essence, for the original model with general distributions, services are classified into types as a way to record partial historical information that is useful for making control decisions. The proxy network problem has a tractable MDP solution with a control structure that is relevant, in a heuristic sense, to the intractable control problem faced in the original model. We focus our attention on the important case that the general distribution has a "heavy tail" (does not decay exponentially) or is highly variable. Intuitively, we are interested in systems where long service times provide significant useful information about the remaining service time distribution. We discuss our heuristic in the context of the new load-balancing model described above. Although it has its roots in service centers, it is also applicable to supply chain management and to transhipment models in transportation networks.

We should point out that the goal is to introduce a method for approximating the load balancing **decisions** made in a parallel processing network, not to approximate the service time distributions themselves. With an eye towards tractability and solutions that are easy to describe and implement, we restrict attention in the proxy model to a hyper-exponential (mixture of exponentials) service distribution with two classes. We find that the optimal control policies for the two-class proxy model, when translated in a smart way (as discussed in Section 5), indeed lead to policies that perform well in the original system. Alternatively, we conjecture that one could approximate service times with an Erlang distribution with k phases, and provide a similar analysis using a Markov decision process formulation. The difficulty would then be in translating that process into an implementable control policy. Moreover, the decision problem would be intractable for k of moderate size. The optimal control for the proxy model we propose is quite simple. More sophisticated MDP models would quickly lose this feature.

This paper makes several contributions. Of course, we describe a method for determining good control policies for the otherwise intractable load balancing problem. The employed proxy model, which is also new, most likely has applications outside of this context, and we find it interesting in its own right. For the proxy model, we show that the optimal control structure is characterized by a series of "do-not-move/move-up-to levels" and that these levels are monotone. Not only do these structural results provide insight, but they also aid in computation. In particular, since the state space of the proxy model has infinite dimension, computation is facilitated by truncation of the queue lengths. Truncation often leads to policies that are not monotone near the boundaries. However, we "smooth" the policies in accordance with the theoretical results, and we find that these smoothed policies perform better. Finally, performance was measured via simulation. We display the results of the numerical study, which show that our policies perform well as compared to some alternative heuristics.

The remainder of the paper is organized as follows. In Section 2 we discuss related literature. Section 3 contains some preliminary results, a further description of the original and proxy models and the optimal-

ity criteria. We present a Markov decision formulation of the proxy model and show several monotonicity results in Section 4. The description of our heuristic for controlling the original load balancing problem, including the relationship to the proxy model, is given in Section 5. Section 5 also contains an implementation of the heuristic and the numerical comparison to several alternative heuristics. The paper is concluded in Section 6.

2 Literature review

The theory (and drawbacks) of MDPs is well-documented. We refer the interested reader to the now classic text of Puterman [14]. The literature on the control of parallel processing networks is also abundant so we do not provide a complete review here. Instead the reader is pointed to the work of Shirazi et al. [18] and Wang and Morris [20] and the references therein. We focus on those papers with direct relevance to the current work. For a basic introduction to heavy-tailed distributions and their properties, see Sigman [19]. A discussion of several alternative definitions can be found in Heyde and Ku [8].

Paxson and Floyd [13] have found that for most of the traffic in the world wide web *session and connection* arrivals are modeled well using Poisson processes, but *packet* interarrivals are better described with heavy tailed distributions. This is further confirmed by Crovella et al. [3]. In particular, the hyper-exponential distribution has proved to be useful to approximate heavy-tailed distributions. Xu et al. [22] use such approximations to formulate generalized Petri nets in order to study the properties of distributed manufacturing systems. The hyper-exponential is one of the motivating factors of our two class Markov decision process formulation.

Harchol-Balter and Downey [6] compare the reassignment of processes to a different server at the time of birth vs. reassignment once the process has already started (preemptive migration) in order to balance CPU load in a network of stations. They obtain a preemptive reassignment strategy that is more effective than remote execution even when the memory transfer cost is high. Yum and Hua-Chun [23] develop an adaptive rule for balancing the load on a parallel queueing system, where some customers are required to wait for a particular server or set of servers. Their rule is a combination of a majority-vote rule (where votes are issued by switchers or routers) and a *join the biased queue* rule as presented by Yum and Schwartz [24]. Yum and Schwartz use this term to denote a rule similar to *join the shortest queue*, but a bias term is added to the queue lengths. This rule is robust to changes in the buffer sizes and input rates, and performs well according to the criteria of lower delay and lower blocking probability. Shimkin and Shwartz [17] study a system of queues that share an arrival process. Arriving customers are subject to admission and routing control. The purpose is to maximize income when there are holding costs and rewards for accepting customers. The arrival and service process parameters depend on the current state of the system. The authors prove the existence of a monotone optimal control policy.

Other research on systems with heavy-tail distributed service times includes Crovella and Harchol-Balter [2] who develop a policy that purposely operates the server hosts at different loads, and directs smaller tasks to the lighter-loaded hosts. Riska et al. [15] present an inexpensive technique for modeling load balancing policies on a cluster of servers conditioned on the fact that the service times of arriving tasks are drawn from heavy-tailed distributions. Their results provide exact information regarding the distribution of task sizes that compose the queue on each server. Beard and Frost [1] study a prioritization mechanism to alleviate overloads that result in blocking the access to service to all customers. Of course none of these studies include a Markov decision process formulation of an exponential model applied to the general model with heavy tailed distributed service times.

Our model is closely related to that in Down and Lewis [4]. Their work refers to a system of parallel queues, where the balancing decisions are taken at the times of arrivals or departures. They seek the optimal design and control policy for the system. There is also a close relation to the work of Lewis [10] where an M/M/1 queue is controlled by two "gatekeepers" that make the decisions of acceptance or rejection of a customer at two moments: the arrival and the moment prior to service. Another study related to the control of queueing systems with exponentially distributed service times can also be found in He and Neuts [7], who study policies that move a fixed amount of customers to control a system of two M/M/1 queues. Transfer of customers occurs when the difference of the queues reaches a critical level.

3 Preliminaries and Model Descriptions

In this section we discuss the formal definition of a parallel processing network with service times that follow a heavy-tailed distribution and the *proxy* model with exponential service times. Consider 2 parallel queues. Customers arrive to queue k according to independent Poisson processes of rate λ_k for $k = 1, 2$. The service processes of each queue are independent of each other and of both arrival streams. The n^{th} customer that is served by server k requires S_n^k time units of service where $\{S_n^k, n \geq 1,\ k = 1, 2\}$ are assumed to be i.i.d. and independent of the station to which the customer arrives. In the *general* model that motivates this study, the service times are assumed to follow a general distribution with finite mean. However, we are most interested in those service distributions that see a large proportion of short service times, but also see some very large service times; those that are highly variable. One such class of distributions is that with "heavy," non-exponential tails.

Definition 3.1 *A distribution function F, for random variable S, is said to be* ***heavy-tailed*** *if $\overline{F}(s) := 1 - F(s) = \mathbb{P}(S > s) > 0$, $s \geq 0$, and*

$$\lim_{s\to\infty} \mathbb{P}(S > s + \delta | S > s) = \lim_{s\to\infty} \frac{\overline{F}(s+\delta)}{\overline{F}(s)} = 1,\ \delta \geq 0. \tag{3.1}$$

Intuitively, if S follows a heavy-tailed distribution, then if S ever exceeds a large value, it is likely to exceed any larger value as well. Thus, while most times are short, a decision-maker that finds a customer whose service time is unusually long would not want to leave customers in queue behind it.

As an approximation to this model we consider a *proxy* model, where each arriving customer is of one of two classes. A customer's classification is not revealed until immediately prior to beginning service. Customers are of class j, $j = 1, 2$, with probability p_j and a class j customer requires an exponentially distributed amount of service with mean $1/\mu_j$. We assume that Class 1 are those with unusually long ("heavy") service times, while Class 2 corresponds to those with shorter ("standard") service times seen in the general model; that is, $1/\mu_1 \gg 1/\mu_2$. We will explain exactly how they are related to the general model when the heuristic is described more fully in Section 5.

In either model, let Π be the set of all non-anticipating policies. A policy $\pi \in \Pi$ prescribes how many customers to move from one queue to another, given the number of customers in each queue (the queue length processes), perhaps the amount of time each customer has been in service, and any other information that is required to make the (policy dependent) process Markovian. For example, in the proxy system the current "state" of the system includes the queue length processes and the classes of the customers currently in service at each queue.

There is a fixed cost for moving each customer of m units per customer. That is, if θ customers are moved, a cost of $m\theta$ is incurred. Customers currently in service (in either queue) cannot be moved; the control policy is assumed to be non-preemptive. The system also continuously incurs holding cost $h_k q_k$ per unit time that queue k contains q_k customers, including the one in service for $k = 1, 2$. Without loss of generality we assume that $h_1 \geq h_2$. We seek to find a strategy for load balancing under the infinite horizon expected discounted cost or the long-run average expected cost optimality criteria. Note here that the term "load balancing" is used somewhat loosely since the holding costs may cause the optimal policy to leave the distribution of the workload for each queue unbalanced. In some sense, perhaps "load distribution" would be more descriptive. However, having made this clarification, we will continue to refer to the control as balancing without further comment since it is common terminology.

For a fixed policy π, denote the set of decision epochs by $\mathbb{D} \equiv \{d_n, n \geq 0\}$ and the state at the n^{th} decision epoch by X_n. For example, if π depends only on the queue lengths, then $\mathbb{D}$ is the set of arrival times and service time completions. We assume that the time between decision epochs is bounded away from zero so that only a finite number of decisions can be made in a finite amount of time. That is, if the time between the n^{th} and $(n+1)^{st}$ decision epoch has distribution G_{n+1} then there exists $\delta > 0$ and $\epsilon > 0$ such that $1 - G_{n+1}(\delta) \geq \epsilon$ (cf. p. 532 of [14]). Let $Q^\pi(t) = \{Q_1^\pi(t), Q_2^\pi(t)\}$ be the queue length process, and let θ_n represent the balancing decision taken at decision epoch n, under π. Define the total discounted

expected cost up until time t as

$$v^{\pi}_{\beta,t}(x) = \mathbb{E}^{\pi}_{x}\Big(\sum_{n=0}^{N(t)} e^{-\beta d_n} c(X_n, \theta_n)\Big) + \int_0^t e^{-\beta u}\, \mathbb{E}^{\pi}_{x}[h_1 Q^{\pi}_1(u) + h_2 Q^{\pi}_2(u)]\mathrm{d}u,$$

where θ_n is the action taken at decision epoch n, $c(\cdot,\cdot)$ is the lump sum cost associated with moving customers from one queue to the other, $N(t)$ is the number of decision epochs in the first t time units, and the expectation of the system under policy π is conditioned on the initial state x. The criteria we are interested in are

$$v^{\pi}_{\beta}(x) = \lim_{t\to\infty} v^{\pi}_{\beta,t}(x), \qquad\qquad \varphi^{\pi}(x) = \limsup_{t\to\infty} \frac{v^{\pi}_{0,t}(x)}{t},$$

where $v^{\pi}_{\beta}(x)$ represents the infinite horizon $\beta-$discounted expected cost under π (the interchange of limit and expectation is justified by the monotone convergence theorem) and $\varphi^{\pi}(x)$ is called the long-run average expected cost starting in state x under policy π. The objective then is to find a policy π^* under each criterion such that $\gamma^{\pi^*}(x) \leq \gamma^{\pi}(x)$ for all states x and all policies $\pi \in \Pi$ for $\gamma = v_\beta, \varphi$. In the next section we provide results that simplify this search considerably for the proxy model. We view these results as interesting in their own right, but they are particularly useful in the implementation of our heuristic in the general model.

4 Optimal Control for the Proxy Model

For the proxy model all inter-arrival and service times are exponentially distributed, and the state may be described by a vector (I, y, i, j), where I represents the total number of customers in the system and y is the number of customers in queue 2 (including any customer in service). When $i\,(j) \in \{1, 2\}$ it represents the class of customer currently at server $1\,(2)$; $i\,(j) = 0$ means that queue 1 (2) is empty. If $x = (I, y, i, j)$, then the possible actions set is $A_x = \{-(y-1)^+, -(y-2), \ldots, I-y-2, (I-y-1)^+\}$. That is, for $\theta \in A_x$, $\theta = 0$ means that nothing will be moved while $\theta > 0$ means θ customers are moved from queue 1 to queue 2 and $\theta < 0$ means that $|\theta|$ are moved from queue 2 to queue 1. A customer that is currently in service cannot be moved. Let $\mathcal{W} := \{(I, y, i, j) \mid I - 1 \geq y \geq 1, i, j \in \{1, 2\}\}$ represent the set of states such that both servers have at least one customer to serve. Similarly, define $\mathcal{I}_1 := \{(I, y, 0, j) \mid I = y \geq 1\}$ and $\mathcal{I}_2 := \{(I, y, i, 0) \mid I \geq 1, y = 0\}$, where $\mathcal{I}_k$ represents the set of states where there are no customers to serve in queue $k = 1, 2$ while the other queue is non-empty (the $\mathcal{I}$ stands for "idle"). The state space $\mathbb{X}$ can now be written

$$\mathbb{X} := \mathcal{W} \cup \mathcal{I}_1 \cup \mathcal{I}_2 \cup \{(0, 0, 0, 0)\}.$$

We apply *uniformization* as described in Lippman [11], with uniformization constant $\Psi = \lambda_1 + \lambda_2 + 2\max\{\mu_1, \mu_2\}$. Without loss of generality assume $\Psi = 1$. This allows us to consider the discrete-time *equivalent* to the continuous proxy model already described. That is to say that the stationary optimal policies in the discrete-time case are the same as that in the continuous-time case. The infinite horizon discounted cost and the long-run average costs also coincide, but only up to a multiplicative constant. The cost function for each period includes holding and switching costs and is given by:

$$C((I, y, i, j), \theta) = \ |\theta| \, m + (I - y - \theta)h_1 + (y + \theta)h_2,$$

where (I, y, i, j) and θ denote the current state and action, respectively. Let the total expected cost of a load balancing policy π over the first t (discrete) decision epochs be defined

$$v^{\pi}_{\alpha,t}(x) := \mathbb{E}^{\pi}_{x} \sum_{n=0}^{t-1} \alpha^t C(X_n, \theta_n),$$

where X_n and θ_n represent the state and balancing decision at decision epoch n. Furthermore, define $v_{\alpha,t}(x) = \inf_{\pi \in \Pi} v^{\pi}_{\alpha,t}(x)$, where Π is the set of all non-anticipating policies; $v_{\alpha,t}(x)$ is the optimal cost-to-go for a t-horizon problem starting in state x, under discount factor α. In the case when $t = \infty$, we write v_α instead of $v_{\alpha,\infty}$. This defines the infinite horizon expected discounted cost criterion. As we are also interested in the average case, the average cost of a fixed policy in the discrete time model equals

$$\limsup_{n\to\infty} \frac{1}{n} v^{\pi}_{1,n}(x) = \limsup_{n\to\infty} \frac{1}{n} \mathbb{E}^{\pi}_{x} \sum_{t=1}^{n} C(X_t, \theta_t).$$

In the remainder of the section, we present several structural results for the finite horizon case. We then give a stability result and show that the structural results continue to hold in the infinite horizon discounted cost and average cost cases. The first result states that the basic features of an optimal policy in Lewis and Down [4] carry over to the current model.

Proposition 4.1 *Under the finite or infinite horizon discounted expected cost or the long-run average expected cost criterion, there exists an optimal policy that does not move customers from queue 2 to queue 1 (since $h_1 \geq h_2$), except possibly to avoid idling.*

Proof. The proof follows precisely in the same manner as that in Theorem 4.1 of Down and Lewis [4] and is omitted for brevity. ∎

Suppose $b = y + \theta$ for $\theta \in A_{(I,y,i,j)}$ (i.e., the number of customers in the low cost queue after performing the control action). Let $\mu_0 = 0$. Define $w_{\alpha,t}(I, y, i, j, b)$ as the cost-to-go, starting in state (I, y, i, j), for

moving up to amount b in period t, followed by optimal control in the remaining periods:

$$w_{\alpha,t}(I,y,i,j,b) = \begin{cases} m|b-y| + h_1(I-b) + h_2 b + U_{\alpha,t-1}(I,b,i,j), & \text{for } 1 \le y \le I-1, \\ m|b-y| + h_1(I-b) + h_2 b + p_1 U_{\alpha,t-1}(I,b,1,j) & \\ \quad + p_2 U_{\alpha,t-1}(I,b,2,j) & \text{for } y = I, i = 0, b < I, \\ h_2 I + U_{\alpha,t-1}(I,b,0,j) & \text{for } y = b = I, i = 0, \\ m|b-y| + h_1(I-b) + h_2 b + p_1 U_{\alpha,t-1}(I,b,i,1) & \\ \quad + p_2 U_{\alpha,t-1}(I,b,i,2) & \text{for } y = 0, j = 0, b > 1, \\ h_1 I + U_{\alpha,t-1}(I,b,i,0) & \text{for } y = b = 0, j = 0, \end{cases}$$

where for $(I,b,i,j) \in \mathcal{W}$

$$\begin{aligned} U_{\alpha,t}(I,b,i,j) = \alpha[&p_1\mu_i v_{\alpha,t}(I-1,b,1,j) + p_2\mu_i v_{\alpha,t}(I-1,b,2,j) + p_1\mu_j v_{\alpha,t}(I-1,b-1,i,1) \\ &+ p_2\mu_j v_{\alpha,t}(I-1,b-1,i,2) + \lambda_1 v_{\alpha,t}(I+1,b,i,j) + \lambda_2 v_{\alpha,t}(I+1,b+1,i,j) \\ &+ (1-\lambda_1-\lambda_2-\mu_i-\mu_j) v_{\alpha,t}(I,b,i,j)], \end{aligned}$$

for $(I,b,0,j) \in \mathcal{I}_1$ ($b = I$ in this case)

$$\begin{aligned} U_{\alpha,t}(I,I,0,j) = \alpha[&p_1\mu_j v_{\alpha,t}(I-1,I-1,0,1) + p_2\mu_j v_{\alpha,t}(I-1,I-1,0,2) \\ &+ \lambda_1(p_1 v_{\alpha,t}(I+1,I,1,j) + p_2 v_{\alpha,t}(I+1,I,2,j)) + \lambda_2 v_{\alpha,t}(I+1,I+1,0,j) \\ &+ (1-\lambda_1-\lambda_2-\mu_j) v_{\alpha,t}(I,I,i,j)], \end{aligned}$$

for $(I,b,i,0) \in \mathcal{I}_2$ ($b = 0$ in this case)

$$\begin{aligned} U_{\alpha,t}(I,0,i,j) = \alpha[&p_1\mu_i v_{\alpha,t}(I-1,0,1,0) + p_2\mu_i v_{\alpha,t}(I-1,0,2,0) \\ &+ \lambda_1 v_{\alpha,t}(I+1,0,i,0) + \lambda_2(p_1 v_{\alpha,t}(I+1,1,i,1) + p_2 v_{\alpha,t}(I+1,1,i,2)) \\ &+ (1-\lambda_1-\lambda_2-\mu_i) v_{\alpha,t}(I,0,i,0)], \end{aligned}$$

and for $I = b = 0$

$$\begin{aligned} U_{\alpha,t}(0,0,0,0) = \alpha[&\lambda_1(p_1 v_{\alpha,t}(1,0,1,0) + p_2 v_{\alpha,t}(1,0,2,0)) + \lambda_2(p_1 v_{\alpha,t}(1,1,0,1) + p_2 v_{\alpha,t}(1,1,0,2)) \\ &+ (1-\lambda_1-\lambda_2) v_{\alpha,t}(0,0,0,0)]. \end{aligned}$$

Let $A_{\mathcal{W}} := \{1,2,\ldots,I-1\}$. Similarly define $A_{\mathcal{I}_1} := \{1,2,\ldots I\}$, $A_{\mathcal{I}_2} := \{0,1,\ldots I-1\}$ and $A_{(0,0,i,j)} := \{0\}$. It is well-known that for $(I,y,i,j) \in \mathcal{W}$, $v_{\alpha,t}$ (note $v_{\alpha,0} = 0$) satisfies the following ***finite***

horizon optimality equations (FHOE)

$$v_{\alpha,t}(I,y,i,j) = \min_{b\in A_{\mathcal{K}}}\{w_{\alpha,t}(I,y,i,j,b)\}, \tag{4.1}$$

where $\mathcal{K} = \mathcal{W}, \mathcal{I}_1, \mathcal{I}_2,$ or $(0,0,0,0)$ depending on (I,y,i,j).

The next result states that there exists an optimal policy such that for each state there is a "do-not-move/ move-up-to" amount L. This means that if $y < L$ we move enough customers to have L customers in queue 2, and if $y \geq L$, we move no customers.

Proposition 4.2 *Suppose the current types at servers* 1 *and* 2 *are* i *and* j*, respectively. Then,*

1. *there exists a level* $L^t_{I,i,j} < I$ *such that for each* $t \geq 1$, $I \geq 2$ *and* $(I,y,i,j) \in \mathcal{W}\cup\mathcal{I}_2$*, the optimal policy is to bring the number of customers in queue 2 up to* $L^t_{I,i,j}$ *if* $y < L^t_{I,i,j}$ *and to move no customers if* $y \geq L^t_{I,i,j}$*, and*

2. $v_{\alpha,t}(I,y,i,j) - v_{\alpha,t}(I,y+1,i,j)$ *is non-decreasing in* y *(i.e.,* $v_{\alpha,t}$ *is convex in* y*) for all* i,j, $t \geq 0$, $I \geq 3$*, and* $y \leq I-3$.

Proof. By induction on t. Recall that $v_{\alpha,0}(\cdot) = 0$. So that Statement 2 holds trivially for $t = 0$. Consider $t = 1$ and assume $I \geq 2$. We have

$$w_{\alpha,1}(I,y,i,j,b) = m|b-y| + h_1(I-b) + h_2 b. \tag{4.2}$$

Since $(I,y,i,j) \in \mathcal{W}\cup\mathcal{I}_2$ queue 1 is non-empty. Recall from Proposition 4.1 that it is not optimal to move customers from queue 2 (the low-cost queue) to queue 1 (the high-cost queue) unless possibly if queue 1 is empty. That is, it suffices to consider $b \geq y$. When we restrict attention to the set $\{y, y+1, ..., I-1\}$, $w_{\alpha,1}$ is a linear function of b (since on this set $|b-y| = b-y$). Depending on the direction of the inequality $m - h_1 + h_2 \geq (\leq)\, 0$ the optimal action is either not to move customers or to move all of the customers to the low cost queue (except for the one currently receiving service at queue 1). That is to say either letting $L^1_{I,i,j} = 0$ or $I-1$ is optimal. This proves the first statement for $t = 1$.

Assume Statement 2 holds for $t-1$. To prove Statement 1 at time t recall

$$w_{\alpha,t}(I,y,i,j,b) = m|b-y| + h_1(I-b) + h_2 b + U_{\alpha,t-1}(I,b,i,j).$$

From Statement 2 at epoch $t-1$, and from the definition of $U_{\alpha,t}$, $U_{\alpha,t-1}(I,b,i,j)$ is a convex combination of convex functions. Thus, $w_{\alpha,t}(I,y,i,j,b)$ is convex in b. Let $L^t_{I,i,j}$ be the minimal (smallest) element of the set $\operatorname{argmin}_{b\in A_{\mathcal{W}}}\{w_{\alpha,t}(I,1,i,j,b)\}$. Note that by convexity $L^t_{I,i,j}$ is also in the $\operatorname{argmin}_{b\in\{y,y+1,...,I-1\}}\{w_{\alpha,t}(I,y,i,j,b)\}$

for all $1 \leq y \leq L^t_{I,i,j}$. The convexity of $w_{\alpha,t}$ together with Proposition 4.1 yield that it is not optimal to move customers for $y > L^t_{I,i,j}$. This proves Statement 1.

To complete the proof it remains to show that the preceding arguments imply that Statement 2 holds for time t and all (i,j). There are several cases to consider. Suppose for now that $(I,y,i,j) \in \mathcal{W}$.

Case 1: $L^t_{I,i,j} \leq y$. In this case, it is optimal not to move in the three states (I,y,i,j), $(I,y+1,i,j)$, $(I,y+2,i,j)$. Thus,

$$\begin{aligned} &v_{\alpha,t}(I,y,i,j) - 2v_{\alpha,t}(I,y+1,i,j) + v_{\alpha,t}(I,y+2,i,j) \\ &\quad = U_{\alpha,t-1}(I,y,i,j) - 2U_{\alpha,t-1}(I,y+1,i,j) + U_{\alpha,t-1}(I,y+2,i,j) \geq 0, \end{aligned}$$

where the inequality follows from the inductive hypothesis that the second statement holds at $t-1$.

Case 2: $y \leq L^t_{I,i,j} - 2$. The optimal action in the three states (I,y,i,j), $(I,y+1,i,j)$, $(I,y+2,i,j)$ is to allocate $I - L^t_{I,i,j}$ customers in queue 1 and $L^t_{I,i,j}$ in queue 2. Thus,

$$v_{\alpha,t}(I,y,i,j) - v_{\alpha,t}(I,y+1,i,j) = v_{\alpha,t}(I,y+1,i,j) - v_{\alpha,t}(I,y+2,i,j) = m.$$

Case 3: $y = L^t_{I,i,j} - 1$. First note $w_{\alpha,t}(I,y+1,i,j,y+2) \geq v_{\alpha,t}(I,y+1,i,j)$; moving one customer is a (potentially) suboptimal action for state $(I,y+1,i,j)$). Thus,

$$\begin{aligned} &v_{\alpha,t}(I,y,i,j) - v_{\alpha,t}(I,y+1,i,j) - [v_{\alpha,t}(I,y+1,i,j) - v_{\alpha,t}(I,y+2,i,j)] \\ &\quad \geq v_{\alpha,t}(I,y,i,j) - w_{\alpha,t}(I,y+1,i,j,y+1) - w_{\alpha,t}(I,y+1,i,j,y+2) + v_{\alpha,t}(I,y+2,i,j) \\ &\quad = m + U_{\alpha,t-1}(I,y+1,i,j) - U_{\alpha,t-1}(I,y+1,i,j) - m - U_{\alpha,t-1}(I,y+2,i,j) + U_{\alpha,t-1}(I,y+2,i,j) = 0, \end{aligned}$$

as desired.

Suppose $(I,y,i,j) \in \mathcal{I}_2$. Let $L^t_{I,i,0}$ be the minimal element of the set $\operatorname{argmin}_{b \in A_{\mathcal{I}_2}} \{w_{\alpha,t}(I,0,i,j,b)\}$. Thus, $v_{\alpha,t}(I,0,i,j) = w_t(I,0,i,j,L^t_{I,i,0})$. Assuming, $L^t_{I,i,2} \geq 2$

$$\begin{aligned} &v_{\alpha,t}(I,0,i,j) - v_{\alpha,t}(I,1,i,j) - [v_{\alpha,t}(I,1,i,j) - v_{\alpha,t}(I,2,i,j)] \\ &\quad \geq w_{\alpha,t}(I,0,i,j,L^t_{I,i,0}) - w_{\alpha,t}(I,1,i,j,L^t_{I,i,0}) - [w_{\alpha,t}(I,0,i,j,L^t_{I,i,2}) - w_{\alpha,t}(I,2,i,j,L^t_{I,i,2})] \\ &\quad = m - m = 0. \end{aligned}$$

Similarly for $L^t_{I,i,2} < 2$. Since for all $(I,y,i,j) \in \mathcal{I}_1 \cup \{(0,0,0,0)\}$, $y = I$ there is no convexity requirement on $\mathcal{I}_1 \cup \{(0,0,0,0)\}$. ∎

We remark that the previous result states the existence of an optimal "move-up-to" level for each (I, i, j). We next characterize these levels as monotone, non-decreasing in I. This result not only lends insight into the structure of the optimal policy, but it is also convenient both for implementation and to simplify its computation. Moreover, it is used to implement the load balancing heuristic presented in Section 5. Before proving the result, it is useful to recall the definition of submodularity:

Definition 4.3 *A function $g(j,k)$ is said to be **submodular** if and only if the difference $g(j,k) - g(j,k+1)$ is non-decreasing in j; that is, $g(j,k) - g(j,k+1) \leq g(j+1,k) - g(j+1,k+1)$.*

Proposition 4.4 *Let $I \geq 3$, $y \in \{0, ..., I-1\}$. Suppose the current types at servers 1 and 2 are i and j respectively. The following hold:*

1. *For $t \geq 1$ there exists optimal move-up-to levels $L^t_{I+1,i,j}$ and $L^t_{I,i,j}$ such that $L^t_{I+1,i,j} \geq L^t_{I,i,j}$.*
2. *$v_{\alpha,t}(I,y,i,j) - v_{\alpha,t}(I,y+1,i,j)$ is non-decreasing in I (i.e., $v_{\alpha,t}$ is submodular in (I,y)) for all $t \geq 0$ and $1 \leq y \leq I-3$.*

Proof. By induction on t. For $t = 0$ Statement 2 holds trivially since $v_{\alpha,0} = 0$. At $t = 1$ we have $w^1(I,y,i,j,b) = m|b-y| + h_1(I-b) + h_2 b$. As in the proof of Proposition 4.2 it is optimal either **(a)** not to move any customers, or **(b)** to move all the customers to the low cost queue (except for the one currently receiving service at queue 1). That is to say the optimal move up to level is $L^1_{I,i,j} = y$ or $I-1$ depending on the direction of the inequality $m - h_1 + h_2 \geq (\leq) 0$. Similarly, for state $(I+1, y, i, j)$, the optimal move up to level is $L^1_{I+1,i,j} = y$ or I (depending on the same inequality. Thus, $L^1_{I+1,i,j} \geq L^1_{I,i,j}$ as desired.

Assume now that Statement 1 holds for t and Statement 2 for $t-1$. There are 4 cases to consider to prove Statement 2 holds at time t. In each of the first three cases we take advantage of the fact that $w_{\alpha,t} \geq v_{\alpha,t}$.
Case 1: $y+1 < L^t_{I,i,j}$ and $y < L^t_{I+1,i,j}$. Then,

$$w_{\alpha,t}(I,y,i,j,L^t_{I,i,j}) - v_{\alpha,t}(I,y+1,i,j) - v_{\alpha,t}(I+1,y,i,j)$$
$$+ w_{\alpha,t}(I+1,y+1,i,j,L^t_{I+1,i,j}) = m - m = 0.$$

Case 2: $y+1 \geq L^t_{I,i,j}$ but $y < L^t_{I+1,i,j}$. Then,

$$w_{\alpha,t}(I,y,i,j,y+1) - v_{\alpha,t}(I,y+1,i,j) - v_{\alpha,t}(I+1,y,i,j)$$
$$+ w_{\alpha,t}(I+1,y+1,i,j,L^t_{I+1,i,j}) = m - m = 0.$$

Case 3: $y+1 \geq L^t_{I,i,j}$ and $y \geq L^t_{I+1,i,j}$. In this case we have

$$\begin{aligned} &w_{\alpha,t}(I,y,i,j,y) - v_{\alpha,t}(I,y+1,i,j) - v_{\alpha,t}(I+1,y,i,j) + w_{\alpha,t}(I+1,y+1,i,j,y+1) \\ &\quad = h_1 - h_2 - (h_1 - h_2) + U_{\alpha,t-1}(I,y,i,j) - U_{\alpha,t-1}(I,y+1,i,j) - U_{\alpha,t-1}(I+1,y,i,j) \\ &\qquad + U_{\alpha,t-1}(I+1,y+1,i,j). \end{aligned}$$

Each of preceding 3 cases imply submodularity for $v_{\alpha,t}$ since $w_{\alpha,t} \geq v_{\alpha,t}$ with the last one also using the inductive hypothesis ($U_{\alpha,t-1}$ is a linear combination of $v_{\alpha,t-1}$).

Case 4: $y+1 < L^t_{I,i,j}$ and $y \geq L^t_{I+1,y,i,j}$. Note that since $y+1 < L^t_{I,i,j}$ we have $y < L^t_{I,i,j} \leq L^t_{I+1,i,j} \leq y$, where the second inequality follows from the inductive assumption. Since Case 4 leads to a contradiction it cannot occur.

It remains to show that $L^{t+1}_{I+1,i,j} \geq L^{t+1}_{I,i,j}$. First note that if $L^{t+1}_{I,i,j} = 1$ the result holds trivially. Assume that $L^{t+1}_{I,i,j} \geq 2$. Note that the submodularity of $v_{\alpha,t}$ implies submodularity of $U_{\alpha,t}(\cdot,y,\cdot,\cdot)$ for $y \geq 2$. Suppose the result does not hold so that $L^{t+1}_{I+1,i,j} < L^{t+1}_{I,i,j}$. Fix $L^{t+1}_{I+1,i,j} < y \leq L^{t+1}_{I,i,j}$ so that the optimal action in $(I+1,y,i,j)$ is to do nothing, while the optimal action in state (I,y,i,j) is to move the number of customers in queue 2 to $L^{t+1}_{I,i,j}$. By using $L^{t+1}_{I,i,j}$ in state $(I+1,y,i,j)$, the optimality equations imply

$$\begin{aligned} v_{\alpha,t+1}(I+1,y,i,j) &= h_1(I+1-y) + h_2(y) + U_{\alpha,t}(I+1,y,i,j) \\ &< m(L^{t+1}_{I,i,j} - y) + h_1(I+1-L^{t+1}_{I,i,j}) \\ &\quad + h_2(L^{t+1}_{I,i,j}) + U_{\alpha,t}(I+1,L^{t+1}_{I,i,j},i,j). \end{aligned}$$

A little algebra yields

$$\begin{aligned} &m(L^t_{I,i,j} - y) + h_1(y - L^{t+1}_{I,i,j}) - h_2(L^t_{I,i,j} - y) \\ &\quad > U_{\alpha,t}(I+1,y,i,j) - U_{\alpha,t}(I+1,L^t_{I,i,j},i,j). \end{aligned} \tag{4.3}$$

Similarly (by considering the action "do nothing" in state (I,y,i,j))

$$\begin{aligned} &m(L^t_{I,i,j} - y) + h_1(y - L^{t+1}_{I,i,j}) - h_2(L^t_{I,i,j} - y) \\ &\quad < U_{\alpha,t}(I,y,i,j) - U_{\alpha,t}(I,L^t_{I,i,j},i,j). \end{aligned} \tag{4.4}$$

Combining (4.3) and (4.4) yields

$$U_{\alpha,t}(I,y,i,j) - U_{\alpha,t}(I,L^t_{I,i,j},i,j) - [U_{\alpha,t}(I+1,y,i,j) - U_{\alpha,t}(I+1,L^t_{I,i,j},i,j)] > 0,$$

which contradicts submodularity and the result is proven. ∎

4.1 The Infinite Horizon Discounted Cost and Average Cost Cases

In this section we note that the results from the previous section extend to the infinite horizon models. While the infinite horizon discounted cost case follows almost immediately, the average cost case is slightly more subtle and requires a stability result.

Proposition 4.5 *For the proxy model, under any stationary, non-idling policy the system is stable if and only if*

$$(\lambda_1 + \lambda_2)\left(\frac{p_1}{\mu_1} + \frac{p_2}{\mu_2}\right) < 2. \tag{4.5}$$

That is, there exists a stationary distribution.

Proof. To prove sufficiency we fix an arbitrary, stationary, non-idling policy π, find a *Lyapunov* function and apply *Foster's criterion* (cf. [9, Theorem 3.7]). This guarantees that all recurrent states are positive recurrent. To this end, consider the Markov chain induced by π. Denote this chain, with state space $\mathbb{X}$, by $\{X_n, n \geq 0\}$. Note that since π is non-idling, $(0,0,0,0)$ is accessible from every state in $\mathbb{X}$; any recurrent states must communicate with the distinguished state $(0,0,0,0)$. Denote the chain restricted to only those states that communicate with $(0,0,0,0)$ by $\{Z_n, n \geq 0\}$ and its state space by $\mathbb{X}^0$. Let $G = \{(I,y,i,j) \in \mathbb{X} | I \leq 1\}$. Let $\mu = \left(\frac{p_1}{\mu_1} + \frac{p_2}{\mu_2}\right)^{-1}$ and define

$$\mathcal{L}(I,y,i,j) \;\; = \;\; I/\mu + 1/\mu_i + 1/\mu_j, \;\; (I,y,i,j) \in \mathbb{X}.$$

For any action chosen and for $(I,y,i,j) \notin G$

$$\begin{aligned} \mathbb{E}\left[\mathcal{L}(Z_{n+1}) - \mathcal{L}(Z_n) | Z_n = (I,y,i,j)\right] &= (\lambda_1 + \lambda_2 - \mu_i - \mu_j)/\mu + \mu_i(p_1/\mu_1 + p_2/\mu_2 - 1/\mu_i) \\ &\quad + \mu_j(p_1/\mu_1 + p_2/\mu_2 - 1/\mu_j) \\ &= (\lambda_1 + \lambda_2)/\mu - 2. \end{aligned} \tag{4.6}$$

Since G is a finite subset of the irreducible set $\mathbb{X}^0$, we may now apply [9, Theorem 3.7] to $\{Z_n\}$ to get that all states in $\mathbb{X}^0$ are positive recurrent when the right-hand side of (4.6) is strictly negative: when (4.5) holds. Furthermore, since (4.6) also applies for states outside of $\mathbb{X}^0$, applying Proposition C.1.5 of [16] to $\{X_n\}$ yields that the expected time to reach $\mathbb{X}^0$ is finite. It follows that a stationary distribution exists (cf. [16, p. 294]).

To show necessity of the inequality, we note that [12, Theorem 11.5.1] implies that when $(\lambda_1 + \lambda_2)/\mu \geq 2$ the expected time to reach G from outside of G is infinite, and thus a stationary distribution cannot exist. ∎

Let U_α and w_α be the obvious infinite horizon analogues to $U_{\alpha,t}$ and $w_{\alpha,t}$, respectively. The following are called the *discounted cost optimality equations* (DCOE):

$$v_\alpha(I,y,i,j) = \min_{b \in A_{\mathcal{K}}} \{w_\alpha(I,y,i,j,b)\}, \tag{4.7}$$

where $\mathcal{K} = \mathcal{W}, \mathcal{I}_1, \mathcal{I}_2,$ or $(0,0,0,0)$ depending on (I,y,i,j). It is well-known that v_α satisfies the DCOE and a policy made up of actions that achieve the minimum on the right hand side of (4.7) is discounted cost optimal. Similarly, if we replace v_α in the definition of the DCOE by some function on $\mathbb{X}$, say ψ, and define U and w as the obvious average cost analogues to U_α and w_α, then the following are called the *average cost optimality equations* (ACOE):

$$g + \psi(I,y,i,j) = \min_{b \in A_{\mathcal{K}}} \{w(I,y,i,j,b)\}, \tag{4.8}$$

where $\mathcal{K} = \mathcal{W}, \mathcal{I}_1, \mathcal{I}_2,$ or $(0,0,0,0)$ depending on (I,y,i,j).

Proposition 4.6 *For the proxy model, the following hold:*

1. *For the discounted cost model*
   - *(a) The quantity $v_{\alpha,t}$ is non-decreasing in t and $\lim_{t\to\infty} v_{\alpha,t} = v_\alpha$.*
   - *(b) Any limit point of an optimal $t-$horizon policy is infinite horizon discounted cost optimal.*
   - *(c) In particular, the results of Propositions 4.2 and 4.4 hold in the infinite horizon discounted cost case.*
2. *For the average cost model, suppose $(\lambda_1+\lambda_2)\left(\frac{p_1}{\mu_1}+\frac{p_2}{\mu_2}\right) < 2$.*
   - *(a) The policy that moves customers only to avoid idling has finite average cost.*
   - *(b) The optimal average cost may be computed as $g = \lim_{\alpha\uparrow 1} v_\alpha(x)$ for any $x \in \mathbb{X}$.*
   - *(c) Any limit point of a $\alpha-$discounted cost optimal policy is average cost optimal.*
   - *(d) There exists a limit function, say ψ of $\psi_\alpha(x) = v_\alpha(x) - v_\alpha(0,0,0,0)$ for $x \in \mathbb{X}$ such that (g,ψ) satisfy the ACOE.*
   - *(e) In particular, the results of Propositions 4.2 and 4.4 hold in the average cost case.*

Proof. Since the state space is countable, the cost function is non-negative, and the action set in each state finite, the first two results in the discounted cost case follow from Proposition 4.3.1 of Sennott [16]. Taking limits in the value functions and along a subsequence in the policies yields the last discounted cost result.

In the average cost case condition **P1'** in Down and Lewis [4] holds for the non-idling policy described (see Example 3.3 in [4]). Applying Theorem 3.6 therein yields the first result in the average cost case.

Corollary 7.5.10 and Theorems 7.2.3 and 7.5.6 of Sennott [16] (collectively) yield Statements 2 (b), (c), and (d). Taking limits in the value functions and along a subsequence in the policies yields the last average cost result. ∎

5 The Load Balancing Heuristic and Numerics

In this section we present the *load balancing* (LB) heuristic for control of the original system with heavy-tailed distributed service times. The LB heuristic requires a mapping from (partial) state information of the original system to the state of the proxy model. For the original system the information of interest is the vector $(I(t), y(t), \eta_1(t), \eta_2(t))$, where $I(t)$ represents the total number of customers in the system at time t, $y(t)$ is the number of customers in the low cost queue (queue 2), and $\eta_k(t)$ denotes the time elapsed since the customer at station k began service. The classification of the customers as "standard" or "heavy" in service depends on a "trigger." The trigger, denoted τ, indicates when the service time $\eta_k(t)$ for the original system is deemed long enough to treat the customer at station k as a heavy type customer. We suggest a method of determining τ below. Define

$$z_k(\eta) = \begin{cases} 1 & \text{if } \eta > \tau, \\ 2 & \text{if } \eta \leq \tau. \end{cases}$$

The original system is observed continuously and controlled at times of arrivals, times of departures, and whenever a customer in service reaches τ units of time at the station. At such times, when the original system vector is $(I(t), y(t), \eta_1(t), \eta_2(t))$, the LB heuristic uses the optimal action for the proxy model with state $(I(t), y(t), z_1(\eta_1(t)), z_2(\eta_2(t)))$, i.e., it uses the move-up-to policy with level $L_{(I(t),z_1(\eta_1(t)),z_2(\eta_2(t)))}$.

The parameters for the proxy model should be chosen to resemble those of the original system. The arrival rates equal those of the original system. If a service time for the original system, S, with distribution $F(\cdot)$, has mean $\mathbb{E}[S] = 1/\mu$, then we suggest setting the average service time of the proxy model, unconditional on the customer type (class), also equal to $1/\mu$. That is, $p_1/\mu_1 + p_2/\mu_2 = 1/\mu$. Note that stability, condition (4.5), is guaranteed if $(\lambda_1 + \lambda_2)/\mu < 2$. We will consider one, of perhaps many, ways to determine p_1 and p_2. Let $p_2 = F(\tau)$ and $p_1 = 1 - F(\tau)$, where τ is predetermined. Under these settings, the service times conditioned on customer type are set to have the same means: $1/\mu_2 = \int_0^\tau s\mathrm{d}F(s)/p_2$ and $1/\mu_1 = \int_\tau^\infty s\mathrm{d}F(s)/p_1$. To determine τ, we suggest a quantile-based method. For a given probability a, define ϕ_a as the the a^{th} quantile: $\mathbb{P}(S \leq \phi_a) = a$. Define z as the probability that ϕ_a is reached given that the trigger τ has been reached: $z = \mathbb{P}(S > \phi_a | S > \tau)$. Given a and z, exogenously, τ can then be calculated from $F(\cdot)$. For the numerical study presented in Section 5.1, $a = 0.8$ and $z = 0.75$. Once τ is calculated, all of the parameters of the proxy model can be determined from the relationships above.

There is one last consideration to make regarding the implementation of the LB heuristic. The state space for the proxy model is (countably) infinite. Instead of solving this problem, the optimal policy is approximated using a truncated state space. That is, the policy is calculated for a system in which each queue has a maximum capacity of B customers. (For the numerical study it was assumed that arrivals to a full queue were lost, at no cost.) The truncated problem can be solved by well known algorithms, including policy iteration and value iteration; cf. [14].

One question remains: How can we apply a truncated policy to the original, non-truncated system? For example, suppose that the capacity $B = 35$ but the original system reaches $(I(t), y(t), z_1(\eta_1(t)), z_2(\eta_2(t))) = (41, 3, i, j)$. When $B = 35$, there are no proxy calculations for queue lengths $(q_1, q_2) = (38, 3)$. One possibility is to simply use the actions associated with $(q_1, q_2) = (\min\{I(t) - y(t), B\}, \min\{y(t), B\})$; for the example, $(q_1, q_2) = (35, 3)$. A downside of this approach is that truncation may result in a non-monotone policy. In short, the computed move-up-to levels increase in I except near the capacity limits, where the levels may suddenly drop off. As an alternative, we suggest a "smoothing" of move-up-to levels, in accordance with Propositions 4.2 and 4.4. Denote the optimal number of customers to move in state (I, y, i, j), for the truncated proxy model, as $\theta^*(I, y, i, j)$; let $\theta^*(I, y, i, j) = 0$ for $(I - y) > B$ or $y > B$. Define $b^+(I, y, i, j) = y + \theta^*(I, y, i, j)$, if $\theta^*(I, y, i, j) > 0$; $b^+(I, y, i, j) = 0$, otherwise. We first approximate move-up-to levels $\tilde{L}_{(I,i,j)} = \max_y b^+(I, y, i, j)$. These levels are not monotone in I; they decrease and are zero for $I > 2B$. So, in a second step we smooth the move-up-to levels to guarantee that they are monotone and positive for a large total number of customers. For (I, i, j) we choose $\hat{L}_{(I,i,j)} = \max_{\{\ell:0\le\ell\le I\}} \tilde{L}_{(\ell,i,j)}$. Then, in the original system we implement a move-up-to policy with level $\hat{L}_{(I(t),z_1(\eta_1(t)),z_2(\eta_2(t)))}$. This *smoothing* approach was compared to the above *no-smoothing* approach (that ignores move-up-to levels explicitly) in the numerical study, and the smoothing approach performed better.

5.1 Numerical Study

We tested the LB heuristic against four heuristics: *do nothing* (DN), *no idling* (NI), *join the shortest queue* (JSQ), and *modified join the shortest queue* (ModJSQ). The DN policy never moves customers. The NI policy moves exactly 1 customer, if available, from one queue to the other if and only if the other server is idle. The JSQ policy only moves customers at times of arrival, by moving an arriving customer to the other queue if the other queue is shorter. The ModJSQ policy moves new arrivals to the other queue if the other queue is accruing total holding costs at a lower rate. That is, a customer arriving to queue 1 (2) is moved to queue 2 (1) if $h_1 q_1 > h_2 q_2$ ($h_1 q_1 < h_2 q_2$).

In our experiment the service time S is distributed according to a *bounded* and *shifted* Pareto distribution. A standard (not bounded or shifted) Pareto distribution is a heavy-tailed, power-law distribution with two parameters, α and κ. It has support $[\kappa, \infty)$, and has infinite variance for $\alpha \le 2$. A bounded (but not shifted) Pareto distribution is similar to a standard Pareto distribution except that it has support $[\kappa, \kappa_2)$;

bounded above by κ_2. A bounded and shifted Pareto is a translation of the bounded Pareto to the origin. Its probability density function is

$$f(s) = \begin{cases} \frac{\alpha\,\kappa^\alpha}{1-(\kappa/\kappa_2)^\alpha}(s+\kappa)^{(-\alpha-1)} & \text{if } 0 \le s \le \kappa_2 - \kappa,\, 0 < \kappa \le \kappa_2, \\ 0 & \text{otherwise.} \end{cases}$$

Its mean is $\mathbb{E}[S] = \frac{\kappa_2^{-\alpha}\left(\kappa^\alpha \kappa_2 \alpha - \kappa\ \kappa_2^\alpha \alpha\right)}{((\kappa/\kappa_2)^\alpha - 1)(\alpha-1)} - K$ and its variance is

$$\text{Var}[S] = \frac{\kappa_2^{-2\alpha}\left(-\frac{((\kappa/\kappa_2)^\alpha - 1)\left(\kappa^2\kappa_2^\alpha - \kappa^\alpha\kappa_2^2\right)\alpha\kappa_2^\alpha}{\alpha-2} - \frac{\left(\kappa^\alpha\kappa_2\alpha - \kappa\ \kappa_2^\alpha\alpha\right)^2}{(\alpha-1)^2}\right)}{\left((\kappa/\kappa_2)^\alpha - 1\right)^2}.$$

The design of experiment is as follows: We fix $\kappa = 0.1$ and vary the utilization $\rho := 1/\mu = \mathbb{E}[S]$ and Var$[S]$, which in turn determine α and κ_2. We also fix $\lambda_1 = \lambda_2 = 1$, in which case the stability condition (4.5) becomes $\rho < 1$. We considered all combinations of $\rho \in \{0.5, 0.6, 0.7, 0.8, 0.85, 0.9, 0.95, 0.99\}$ and Var$[S] \in \{1, 3, 6, 9, 12\}$. As noted above, for LB $a = 0.8$ and $z = 0.75$, which, together with the distribution function, determine τ, p_1, and p_2. For the costs, h_2 is fixed at 1 and $h_1 \in \{1.25, 1.5, 2\}$ and $m \in \{0.75, 1.5, 2.5\}$. In total, the factorial consists of 360 combinations of parameter settings.

The policies were evaluated under the average cost criterion. For each policy and parameter setting, a simulation was developed consisting of 60 (consecutive) runs, plus an additional run at the the beginning as a warm-up period. Each run had a length of 100,000 time units. The optimal policy for the proxy model was determined using a truncated state space with capacity $B = 35$ customers per queue. Unless stated otherwise, LB refers to the smoothing approach.

Of the 360 cases, LB policy performs best in 298 (82.8%), the NI policy performs best in 29 (8.1%), the JSQ policy performs best in 12 (3.3%), and the ModJSQ policy performs best in 21 (5.8%). The DN policy has the highest costs in every case. In terms of costs, on average over the 360 cases, LB improves upon DN by 58.9%, NI improves upon DN by 55.8%, ModJSQ improves up DN by 53.1%, and JSQ improves upon DN by 51.9%. The standard deviations (s.d.) of these percent improvements upon DN are 6.4%, 6.1%, 6.3%, and 5.8% for LB, NI, ModJSQ, and JSQ, respectively. This implies that the use of a control policy to balance the load of the system is worthwhile.

The average reduction in total costs for LB over the alternative policies are 7.0% (s.d. 6.4%) for NI, 14.5% (s.d. 9.1%) for JSQ, and 12.3% (s.d. 9.3%) for ModJSQ. In the cases for which LB is the best policy, LB averages 7.4% (s.d. 5.3%) lower costs than the best alternative. On the other hand, when LB is not the best policy, it has 6.1% (s.d. 9.1%) higher costs than the best. An important observation to make is the fact that in most of the cases that LB is outperformed the utilization is high. When $\rho = .99$ is omitted LB is the best policy in 89.5% of the cases, and when both $\rho = .99$ and $\rho = .95$ are omitted, LB is the best policy in 93.7% of the cases. As it turns out, these high utilization are subject to appreciable simulation error.

We calculated 95% confidence intervals for the time average costs of a simulation run (assuming runs are independent) and found that we cannot be confident that the LB policy is actually outperformed in any of the cases with $\rho \geq 0.95$. Generally speaking, it can be problematic to simulate queues with truncated Pareto service time distributions; see [5]. Therefore, we compared simulated costs for the DN policy against the exact costs calculated using the Pollaczek-Khintchine formula. While the average absolute percent deviation between the simulated and exact DN costs is only 0.8% for $\rho \leq .9$, it is 12.0% for $\rho = .99$ and 2.2% for $\rho = .95$. In light of this, from here on we restrict attention to $\rho \leq .9$.

Averaged over the remaining 270 cases, LB reduces costs by 7.6% (s.d. 5.7%) over NI, 18.0% (s.d. 6.5%) over JSQ, and 15.4% (s.d. 5.5%) over ModJSQ. Table 1 displays the percent differences in costs for $\rho = .85, .9$; LB is not outperformed in **any** of these cases. The best alternative to LB is NI, and in fact NI is the only policy to outperform LB in the 270 cases – in 17 cases (6.3%); see Table 2. LB tends to be outperformed at the lower utilizations, when moving costs are higher, and the difference in holding cost rates between the queues is lower. As indicated in Table 1, for Var(S) $= 1$ the performances of LB and NI are closest when $h_1 = 1.25$ and $m = 2.5$; though the trend does not hold for all variances. Compared to NI, LB has lower costs by an average of 8.6% (s.d. 5.3%) when $\rho = .85$ and by 7.3% (s.d. 4.9%) when $\rho = .9$. For all 270 cases, where LB outperforms NI it does so by an average of 8.2% (s.d. 5.4%). On the other hand, when NI is better LB has only 0.8% (s.d. 0.6%) higher costs. This is true in general; the average difference in total costs between LB and the best policy is very small compared to the difference between the best policy and the other heuristics; see Table 3. In terms of moving costs, LB has higher moving costs than NI in 255 (94.4%) of the cases. The moving costs for LB are 56.1% (s.d. 47.4%) higher than those of NI. In the 15 cases where LB has lower moving costs, they are 2.6% (s.d. 2.5%) lower. The LB policy is more aggressive in moving customers.

We also ran simulations for the no-smoothing approach to the LB heuristic. On average, the smoothing approach outperformed the no-smoothing approach by 1.7% (s.d. 2.2%). To further test the effects of truncation, we ran simulations for the smoothing approach with queue capacity $B = 20$ and $\rho = .85, .9$. As compared to the smoothing approach when $B = 35$, the costs increased by 1.02% (s.d. 3.78%). (For $\rho = .85, .9$, the costs of the no-smoothing approach for $B = 35$ over the smoothing approach for $B = 35$ are higher by 0.7% (s.d. 1.2%).) So, truncation in the proxy model has an effect on the performance of the LB heuristic, and the smoothing approach does a better job than the no-smoothing approach at mitigating the effect.

In summary, LB (with smoothing) performs well. As indicated by the poor performance of DN, moving customers can greatly decrease holding costs. The simulation outputs for $\rho \geq .95$ are noisy and inconclusive; otherwise, LB outperforms ModJSQ and JSQ. The best alternative to LB is NI. The performances of LB and NI are closest when moving costs are high and the difference in holding cost rates is low. The NI policy can outperform LB when utilization is low, but LB is not outperformed by much and there are larger gains in the other cases. Finally, we should note that in choosing the parameters for the proxy model we only

considered $a = 0.8$ and $z = 0.75$. There is room for improvement by optimizing these settings. We leave such an exercise for future research.

6 Conclusion

In conclusion, we have introduced a new method for load balancing in the case for highly variable service distributions. The method introduced is robust to changes in the parameter settings even in the case where it is not adjusted to optimize the implementation. The most reasonable alternative to our heuristic appears to be a non-idling heuristic. In this case, the question is simply, is the consistency and savings worth the difficulty of implementing our heuristic. In many cases we believe more than 8.5% savings is worth the time to implement our heuristic.

At the same time, we have shown that the use of Markov decision processes can mitigate the challenges of a general service time distribution. We believe that the ideas described here can lead to insights for other queueing models. The example of admission controlled $M/G/1$ has already been alluded. Exactly the same intuition holds for service rate control in a $G/M/1$. Of course, these are just the building blocks for more sophisticated models. We note that an extension of the current work is to consider a larger network of queues, and we conjecture that the *two pairing* heuristics described in (Wu et al. [21] and Down and Lewis [4]) would be useful. We leave this for future research.

7 Acknowledgments

The authors would like to thank Douglas Down from McMaster University and anonymous referees for their helpful comments on previous versions of this paper.

This work was supported by the National Science Foundation under Grant Nos. CMMI-0540808 and CMMI-0826255. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The first author was also supported by the Mexican Council for Science and Technology (Consejo Nacional de Ciencia y Tecnologia).

ρ	h_1	m	NI Var[S]					JSQ Var[S]					ModJSQ Var[S]				
			1	3	6	9	12	1	3	6	9	12	1	3	6	9	12
0.85	1.25	0.75	4.52	4.46	4.73	7.84	4.17	11.56	11.13	11.13	12.16	9.46	13.44	11.36	10.78	10.55	5.85
		1.5	3.56	3.86	2.39	5.76	2.95	12.07	12.82	7.56	9.02	9.57	11.65	12.67	8.58	8.94	7.41
		2.5	0.90	1.64	2.85	2.41	2.86	10.98	10.51	10.47	7.06	9.63	13.79	11.00	12.60	6.86	8.14
	1.5	0.75	10.45	10.96	8.90	8.02	6.64	17.19	16.71	14.60	12.56	11.75	15.88	14.38	11.72	9.14	7.56
		1.5	7.78	10.20	7.38	8.17	4.61	17.28	16.63	14.08	14.03	10.22	14.63	14.01	11.23	11.32	8.99
		2.5	4.56	6.75	4.94	5.75	5.82	15.77	14.38	12.47	12.00	11.41	15.33	15.11	9.77	10.40	7.47
	2	0.75	21.05	20.31	15.85	14.50	13.94	27.59	24.88	20.11	19.93	18.87	21.89	18.97	15.73	10.77	7.11
		1.5	16.83	16.70	15.58	15.52	14.29	23.48	23.56	21.76	19.10	19.68	19.00	17.40	14.11	10.93	10.29
		2.5	13.59	16.20	11.76	11.09	10.65	23.36	22.57	18.30	17.28	15.36	16.89	16.94	11.76	10.03	7.40
0.9	1.25	0.75	4.96	4.66	3.55	1.77	3.10	10.63	9.01	9.47	6.57	6.45	11.33	8.49	7.41	4.95	7.42
		1.5	2.88	2.59	4.80	0.42	0.36	11.02	8.22	6.80	5.96	5.80	12.32	10.95	6.18	4.50	2.66
		2.5	2.24	2.64	2.25	1.18	1.41	10.80	9.77	7.13	6.41	5.60	12.61	8.50	9.67	7.40	5.49
	1.5	0.75	10.99	9.68	7.62	5.66	7.18	15.96	14.29	9.80	9.94	10.85	16.76	13.23	9.35	6.49	7.12
		1.5	9.18	8.10	6.98	2.80	4.95	15.78	12.56	11.56	8.13	8.61	14.75	12.68	7.98	8.39	8.12
		2.5	7.32	6.73	5.45	2.72	4.48	15.61	12.42	12.14	8.88	8.75	15.55	11.73	7.42	7.61	9.12
	2	0.75	20.93	15.98	13.21	8.92	9.88	25.69	20.07	17.09	13.01	14.80	19.86	14.75	8.35	4.22	0.99
		1.5	17.92	15.28	12.63	10.33	8.87	23.75	20.15	18.06	14.39	12.01	18.71	14.28	9.93	6.29	2.40
		2.5	15.65	13.17	11.26	6.58	7.49	22.26	18.37	13.78	11.96	12.25	16.49	11.04	7.33	1.86	1.60

Table 1: The percent decrease in total long-run average costs for LB compared to NI, JSQ, and ModJSQ.

	# of times optimal/total			
ρ	LB	NI	JSQ	ModJSQ
0.5 – 0.6	79/90	11/90	0/90	0/90
0.7 – 0.8	84/90	6/90	0/90	0/90
0.85 – 0.9	90/90	0/90	0/90	0/90
0.95	29/45	7/45	3/45	6/45
0.99	16/45	5/45	9/45	15/45

Table 2: The number of times each policy is optimal (amongst this group of heuristics).

ρ	LB	NI	JSQ	ModJSQ
0.5	0.1%	5.9%	30.5%	26.4%
0.6	0.1%	10.0%	29.9%	25.6%
0.7	0.1%	9.3%	24.2%	20.1%
0.8	0.0%	9.0%	19.9%	15.9%
0.85	0.0%	10.0%	18.3%	13.8%
0.9	0.0%	8.1%	14.4%	10.4%

Table 3: The average percent above optimal costs (amongst this group of heuristics).

References

[1] C. Beard and V. Frost. Prioritized resource allocation for stressed networks. *IEEE-ACM Transactions on networking*, 9(5):618–633, 2001. 4

[2] M. Crovell, M. Harchol-Barter, and C. Murta. Task assignment in a distributed system: improving performance by unbalancing load. *Performance Evaluation Review*, 26(1):268–269, June 1998. 4

[3] M. Crovella, M. Taqqu, and A. Bestavroz. Heavy tailed probability distributions in the world wide web. In R. Adler, R. Feldman, and M. Taqqu, editors, *A practical Guide to Heavy Tails: Statistical Techniques & Applications*, pages 3–26. Birkhäuser, 1998. 3

[4] D. Down and M. Lewis. Dynamic load balancing in parallel queueing systems: stability and optimal control. *The European Journal of Operational Research*, 168(2):509–519, January 2006. 4, 7, 14, 19

[5] D. Gross, J. F. Shortle, M. J. Fischer, and D. M. B. Masi. Difficultied in simulating queues with Pareto service. In *Proceedings of the 2002 Winter Simulation Conference*, pages 407–415, 2000. 18

[6] M. Harchol-Balter and A. Downey. Exploiting process lifetime distributions for dynamic load balancing. *ACM Transactions on Computer Systems*, 15(3):253–285, August 1997. 3

[7] Q. He and M. Neuts. Two m/m/1 queues with transfers of customers. *Queueing systems*, 42(4):377–400, 2002. 4

[8] C. Heyde and S. Ku. On the controversy over tailweight of distributions. *Operations Research Letters*, 32:399–408, 2004. 3

[9] V. Kulkarni. *Modeling, Analysis, Design, and Control of Stochastic Systems*. Springer-Verlag, New York, 1999. 13

[10] M. Lewis. Average optimal policies in a controlled queuing system with dual admission control. *Journal of Applied Probability*, 38:369–385, 2001. 4

[11] S. Lippman. Applying a new device in the optimization of exponential queueing systems. *Operations Research*, 23:687–710, 1975. 7

[12] S. Meyn and R. Tweedie. *Markov Chains and Stochastic Stability*. Springer–Verlag, New York, 1 edition, 1993. 13

[13] V. Paxson and S. Floyd. Wide-area traffic: The failure of Poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, 1995. 3

[14] M. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons Inc., New York, 1994. 3, 5, 16

[15] A. Riska, E. Smirni, and G. Ciardo. Analytic modeling of load balancing policies for tasks with heavy-tailed distributions. In *Proceedings of the Second International Workshop on Software and Performance – WOSP 2000*, pages 147–157, 2000. 4

[16] L. Sennott. *Stochastic Dynamic Programming and the Control of Queueing Systems*. John Wiley & Sons Inc., New York, 1999. 13, 14, 15

[17] N. Shimkin and A. Shwartz. Control of admission and routing in parallel queues operating in a random environment. In *Proceedings of the 28th Conference on Decision and Control*, volume 2, pages 1064–1065, Tampa, 1989. IEEE. 3

[18] B. Shirazi, A. Hurson, and K. Kavi. *Scheduling and Load Balancing in Parallel and Distributed Systems*. Wiley–IEEE Computer Society Press, 1 edition, 1995. 3

[19] K. Sigman. A primer on heavy-tailed distributions. *Queueing Systems*, 33(1–3):261–275, 1999. 3

[20] Y.-T. Wang and R. Morris. Load sharing in distributed systems. *IEEE Transactions on Computers*, 34:204–217, 1985. 3

[21] C.-H. Wu, D. Down, and M. Lewis. Heuristics for allocation of reconfigurable resources in a serial line with reliability considerations. *IIE Transactions*, 40(6):595–611, June 2008. 19

[22] T. Xu, A. Desrochers, and R. Graves. Hyperexponential-based network traffic model for distributed manufacturing. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 4, pages 3452–3457. IEEE, October 2003. 3

[23] T. Yum and L. Hua-chun. Adaptive load balancing for parallel queues with traffic constraints. *IEEE Transactions on Communications*, 32(12):1339–1342, December 1984. 3

[24] T.-S. Yum and M. Schwartz. The join-biased-queue rule and its application to routing in computer communication networks. *IEEE Transactions on Communications*, 29(4):505–511, April 1981. 3