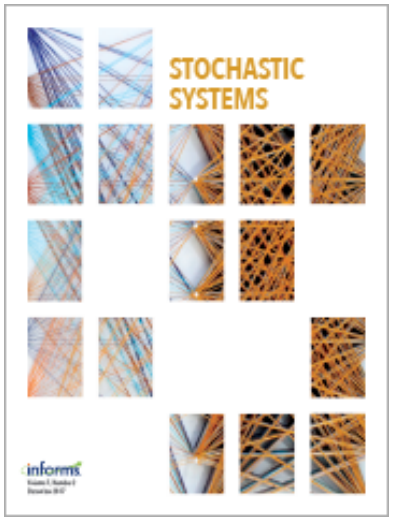

Stochastic Systems

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Dynamic Scheduling for Parallel Server Systems in Heavy Traffic: Graphical Structure, Decoupled Workload Matrix and some Sufficient Conditions for Solvability of the Brownian Control Problem

V. Pesic, R. J. Williamshttp://www.math.ucsd.edu/~williams

To cite this article:
V. Pesic, R. J. Williamshttp://www.math.ucsd.edu/~williams (2016) Dynamic Scheduling for Parallel Server Systems in Heavy Traffic: Graphical Structure, Decoupled Workload Matrix and some Sufficient Conditions for Solvability of the Brownian Control Problem. Stochastic Systems 6(1):26-89. https://doi.org/10.1287/14-SSY163

Stochastic Systems
2016, Vol. 6, No. 1, 26–89
DOI: 10.1214/14-SSY163

DYNAMIC SCHEDULING FOR PARALLEL SERVER SYSTEMS IN HEAVY TRAFFIC: GRAPHICAL STRUCTURE, DECOUPLED WORKLOAD MATRIX AND SOME SUFFICIENT CONDITIONS FOR SOLVABILITY OF THE BROWNIAN CONTROL PROBLEM*

By V. Pesic† and R. J. Williams‡

XR Trading† and University of California, San Diego‡

We consider a dynamic scheduling problem for parallel server systems. J. M. Harrison has proposed a scheme for using diffusion control problems to approximately solve such control problems for heavily loaded systems. This approach has been very successfully used in the special case when the diffusion control problem can be reduced to an equivalent one for a one-dimensional workload process. However, it remains a challenging open problem to make substantial progress on using Harrison's scheme when the workload process is more than one-dimensional. Here we present some new structural results concerning the diffusion control problem for parallel server systems with arbitrary workload dimension. Specifically, we prove that a certain server-buffer graph associated with a parallel server system is a forest of trees. We then exploit this graphical structure to prove that there exists a matrix, used to define the workload process, that has a block diagonal-like structure. An important feature of this matrix is that, except when the workload is one-dimensional, this matrix is frequently different from a choice of workload matrix proposed by Harrison. We demonstrate that our workload matrix simplifies the structure of the control problem for the workload process by proving that when the original diffusion control problem has linear holding costs, the equivalent workload formulation also has a linear cost function. We also use this simplification to give sufficient conditions for a certain least control process to be an optimal control for the diffusion control problem with linear holding costs. Under these conditions, we propose a continuous review threshold-type control policy for the original parallel server system that exploits pooling of servers within trees in the server-buffer graph and uses non-basic activities connecting different trees in a critical manner. We call this partial pooling. We conjecture that this threshold policy is asymptotically optimal in the heavy traffic limit. We illustrate the solution of the diffusion control problem and our proposed threshold control policy for a three-buffer, three-server example.

Received November 2014.

*Research supported in part by NSF grants DMS-0906535 and DMS-1206772.

MSC 2010 subject classifications: Primary 60K25, 68M20, 90B36; secondary 60J70.

Keywords and phrases: Stochastic networks, dynamic control, resource pooling, heavy traffic, Brownian Control Problems, state space collapse, threshold policies.

CONTENTS

1. Introduction. In this paper, we consider a problem of dynamic scheduling for parallel server systems. Such systems arise as stochastic models for "one pass" processing in a variety of applications in operations management, including manufacturing, computer systems and customer service centers. An important feature of these systems is that servers can have overlapping capabilities that allow for flexible scheduling of jobs to available servers. Indeed, parallel server systems constitute an important subclass of more general stochastic processing networks in which one can study the implications of server flexibility without the additional complications of feedback.

The structure of the parallel server systems considered in this paper is described in detail in Section 2 below. Briefly, in these systems, multiple types of jobs are processed using a bank of heterogeneous servers operating in parallel. Jobs awaiting service are stored in buffers according to their type and jobs exit the system after receiving service. Jobs from a given buffer may be processed by one of several different servers and servers may process more than one type of job. The system manager seeks to minimize holding costs by dynamically scheduling waiting jobs to available servers. In general, it is a challenging problem to design "good" control policies for these systems.

1.1. *Heavy traffic approach.* With the exception of a few special cases, dynamic scheduling problems for parallel server systems cannot be solved exactly, and it is natural to resort to more tractable approximations. Here we consider one class of such approximations proposed by J. M. Harrison [14, 17] called Brownian Control Problems (BCPs). For these approximations, stochastic processes describing queue-lengths in the original network are approximated by diffusion processes under a rescaling of time and space. The limiting regime for the approximation is one in which the number of servers is kept fixed while the nominal load on the system approaches the processing capacity of the system. This is often referred to as the conventional heavy traffic regime. Harrison in fact proposes diffusion approximations for more general stochastic processing networks than the parallel server systems considered here. However, there are a variety of open problems associated with using Harrison's scheme and by studying the subclass of parallel

server systems, which are interesting in their own right, we are able to produce some new results related to these open problems.

The steps in using Harrison's [14, 17] scheme may be described as follows. (This outline is the same for parallel server systems and for more general stochastic processing networks.)

(I) Formulate a control problem for the original stochastic processing network model.
(II) Formulate a notion of heavy traffic (when there is flexibility in assigning jobs from a buffer to more than one server, the load on the servers depends on the scheduling of jobs and so one needs to specify what is meant by heavy traffic).
(III) Formulate a formal diffusion approximation (Brownian Control Problem) for the original control problem.
(IV) Solve the Brownian Control Problem (BCP).
(V) "Interpret" the solution of the BCP by proposing a policy for the original network.
(VI) Investigate the performance of the policy proposed in (V). In particular, determine whether it is asymptotically optimal (in the heavy traffic limit).

For stochastic processing networks that include the parallel server systems considered here, steps (I)–(III) have been well developed in works of Harrison et al. [14, 17, 19]. For unitary networks, which include parallel server systems, Budhiraja and Ghosh [11] have proved convergence of the value function for the original system to that for the BCP, thereby justifying removal of the word "formal" in step (III). (Kushner and Chen [23] also proved such convergence for a certain family of parallel server systems.) For step (IV), a substantial simplification of the Brownian Control Problem was obtained by Harrison and Van Mieghem [19] to a so-called Equivalent Workload Formulation (EWF). In this, the Brownian queue-length process in the BCP is replaced by a Brownian workload process in the EWF. This can result in a substantial reduction in dimension of the state space for the diffusion control problem. It also connects with notions of resource pooling introduced by Kelly and Laws [22].

In the special case when the Brownian workload process is one-dimensional, there have been quite a few works executing some or all of steps (IV)–(VI) for various kinds of stochastic processing networks and especially for parallel server systems [1, 6, 7, 13, 16, 18, 29, 32, 34]. Beyond one-dimensional workload, there are a few examples of multi-class queueing networks (which are not parallel server systems) in which the Brown-

ian workload dimension is low for which steps (IV)–(V), and occasionally (VI), have been executed (see e.g., [10, 20, 22, 26, 27, 30, 33]). Also, several methods for generically approximating solutions of the BCP have been proposed for various stochastic processing networks, see e.g., [11, 15, 24, 25, 28]. However, these generic methods typically do not take advantage of elegant structure revealed through solution of the EWF and BCP. Consequently, it has remained an outstanding open problem to take advantage of the EWF dimension reduction to execute steps (IV)–(VI) when the Brownian workload is not one-dimensional, and to develop interpretations in step (V) that preserve elegant structure revealed through solution of the EWF and BCP. In particular, for parallel server systems, this has been an open problem for Brownian workload dimensions above one.

In this paper, towards breaking the aforementioned "dimension barrier", we provide some sufficient conditions for resolving steps (IV)–(V) for parallel server systems with arbitrary Brownian workload dimension. In subsequent work, we plan to address (VI) under these conditions.

To set the scene for describing our contributions, the next subsection reviews the literature related to steps (IV)–(VI) when the Brownian workload is one-dimensional. After that, in §1.3, we summarize the main contributions of this paper. We follow that with subsections describing the organization of this paper and notation and terminology.

1.2. *Prior work for one-dimensional workload.* Here we shall emphasize work for parallel server systems, and will mention related work for more general stochastic processing networks in passing. Henceforth, we shall just use the term "workload" in place of "Brownian workload".

When the workload dimension is one, for suitable holding costs (including those that are linear and certain convex costs), the EWF and BCP can be solved explicitly, and the optimal solution of the EWF is the least control solution. This situation of one-dimensional workload is often refered to as one of *complete resource pooling* (CRP), because, in the solution of the BCP, the efforts of the individual servers can be efficiently combined to act as a single pooled resource or "superserver". Even in this case where the EWF and BCP diffusion control problems can be solved, so that step (IV) is resolved, there is still the challenge of executing steps (V)–(VI).

Harrison and López [18] initiated study of parallel server systems when the workload is one-dimensional. For linear holding costs, they addressed steps (IV) and (V) by giving an explicit solution for the BCP and using this solution to propose a discrete review policy for use in the original parallel server system. In the subsequent work [16], Harrison proved asymptotic optimality of such a discrete review policy for the case of two servers and

two buffers with linear holding costs and special distributional assumptions when the workload is one-dimensional, thus completing step (VI) for this example.

Another important aspect of [18] was that Harrison and López gave an elegant characterization of the situation when the workload dimension is one. In particular, they proved that one-dimensional workload for parallel server systems is equivalent to connectedness of a certain server-buffer graph in which the servers and buffers are nodes and undirected edges between the nodes are given by basic activities. (A basic activity is an activity that plays a role in the solution of a first order (or fluid) approximation to the parallel server scheduling problem.) We note that although the characterization in [18] was developed in the context of linear holding costs, the proof does not use the form of the cost and so this result is true for a more general cost structure such as that in Harrison and Van Mieghem [19].

Williams [34] subsequently showed that the server-buffer graph identified in [18] is necessarily a tree and proposed a continuous review dynamic threshold policy based on this structure. Bell and Williams [6, 7] proved asymptotic optimality of such a dynamic threshold policy for linear holding costs when the workload is one-dimensional, thus completing the final step (VI) for parallel server systems with linear holding costs.

For parallel server systems with certain strictly convex holding costs and one-dimensional workload, Stolyar [32] and Mandelbaum and Stolyar [29] proved asymptotic optimality of max-weight and generalized $c\mu$-type policies, respectively. Stolyar's work was in discrete time and that of Mandelbaum and Stolyar was in continuous time. Also, the work of Stolyar was for generalized switches which are somewhat more general than parallel server systems. Going beyond one-pass systems, Ata and Kumar [1] and Dai and Lin [13] considered stochastic processing networks with feedback when the workload is one-dimensional. Ata and Kumar proved asymptotic optimality of a discrete review policy for unitary networks with linear holding costs. For more general stochastic processing networks with quadratic holding costs, Dai and Lin proved asymptotic optimality of so-called maximum pressure policies. An attractive feature of the works [13, 29, 32] is that they do not require knowledge of the arrival rates for execution of the policies. The policies used there do not cover the linear holding cost case. However, as outlined in [29, 32] and shown in [13], the policies can be used to give asymptotically ϵ-optimal policies in the linear holding cost case.

1.3. *Beyond one-dimensional workload: Contributions of this paper.* In a sense, this paper is a generalization of the seminal work of Harrison and

López [18] for parallel server systems with one-dimensional workload to the situation where the workload dimension is more than one.

Our first result generalizes the graphical result of [18] to prove that the server-buffer graph introduced in [18] for parallel server systems is for arbitrary workload dimension a forest of trees, where the number of trees equals the workload dimension.

Our second result relates to the linear transformation used to define workload from queue-length. In general, the matrix used to define workload from queue-length in the reduction of the BCP to the EWF is not unique. When the workload is one-dimensional, it is effectively unique, being unique up to a scalar multiple. However, for higher dimensional workloads, there are generally infinitely many choices of workload matrix. The choice of workload matrix is important as it affects the analytic tractability of the EWF and hence the BCP. In [17], Harrison proposed a method for choosing the workload matrix which reduced the possible choices to a finite set. His choice is in terms of the extremal optimal solutions of the dual of the linear program used to define heavy traffic. This choice was very useful in the case of one-dimensional workload. However, we argue in this paper that, at least for the parallel server systems considered here, when the workload dimension exceeds one, there is a better choice of workload matrix, which in a certain sense uses a localized version of Harrison's choice.

More precisely, for our second result, we prove that a workload matrix can be chosen that has a natural block diagonal-like structure (in which each block is a single row). This choice of workload matrix is naturally suggested by the "forest of trees" structure of the server-buffer graph revealed in our first result. Indeed, the rows of the workload matrix are obtained from the solutions of the duals of the linear programs associated with each of the individual trees. We call this a *decoupled workload matrix* because each queue-length component affects just one workload component and the workload matrix partitions or decouples the queue-length components coming from different trees. We further show that for our choice of workload matrix, when the BCP has linear holding costs, the *cost function for the EWF is a linear function of the workload* (with positive coefficients). This represents a significant simplification since in general the EWF cost function is convex and piecewise linear when the BCP has linear holding costs. In Section 6.4, we use a simple example to illustrate the fact that in general the EWF has a more straightforward structure under our decoupled workload matrix than under the choice proposed in [17].

For our third result, assuming linear holding costs for the BCP, we exploit the simplification of the EWF afforded by our decoupled workload matrix to

provide sufficient conditions for explicit solvability of the EWF and the BCP. More precisely, we provide sufficient conditions for a least control to be a solution of the EWF. We focus on the situation where the trees in the server-buffer graph are connected in a fairly minimal way by non-basic activities that are relatively expensive to use. (Non-basic activities are activities that do not play a role in solving a first order (or fluid) approximation to the parallel server scheduling problem, but which can play an important role in solving the second order (BCP) diffusion approximation.) In an operations management application, an example of this situation would arise when in each tree there is a single server who is cross-trained to occasionally serve jobs from a buffer in another tree, although at a higher overall cost than for its usual activities. We further show how the set of controls in the EWF can be reduced to yield a Reduced Equivalent Workload Formulation (REWF) and that a least control process, whose existence is guaranteed by results of Yang [35], is an optimal control for the REWF. Under our conditions, we propose a continuous review threshold-type control policy for the original parallel server system which we conjecture is asymptotically optimal in the heavy traffic limit. This control policy takes advantage of pooling of servers within trees, which we call *partial pooling*. Thus we provide sufficient conditions for execution of steps (IV)–(V) of Harrison's scheme, where this was made possible by use of our decoupled workload matrix. The resolution of these steps is not known in general using the choice of workload matrix in [17].

To illustrate our theoretical developments, we consider a three-buffer, three-server parallel server system with linear holding costs for which the workload is two-dimensional. The reader curious about the application of our results may wish to consult this example before reading about the general case. For this specific example we solve the REWF (and hence the EWF and BCP), and we describe the threshold control policy in detail. We prove asymptotic optimality of this policy for this example in a separate work [31]. To our knowledge, this example is the first instance of a proved asymptotically optimal policy for a parallel server system with more than one-dimensional workload and critical use of non-basic activities in the conventional heavy traffic regime. (We note that Ata and Van Mieghem [3] propose selective use of a non-basic activity based on a large deviations analysis of the approximating diffusion control problem in a parallel server example with two-dimensional queue-length and workload; however, they do not address the issue of asymptotic optimality. Also, in the different context of the many server or Halfin-Whitt scaling limit, Atar, Mandelbaum and Shaiket [4] made use of non-basic activities under a complete resource pooling condition.)

1.4. *Organization of the paper.* The paper is organized as follows. In Section 2, we describe the model of a parallel server system considered here. As we will need various elements from steps (II)–(III) of Harrison's scheme to provide context for our results, in Sections 3 and 4 we summarize these steps, as well as the EWF reduction provided in [19]. In Section 3, we introduce a sequence of parallel server systems and describe what it means for the sequence to approach heavy traffic. In Section 4, we describe the Brownian Control Problem and the Equivalent Workload Formulation associated with the sequence of parallel server systems. In this section we also introduce a further reduction, showing how to reduce the Equivalent Workload Formulation to a simpler control problem called the Reduced Equivalent Workload Formulation (REWF). Apart from the reduction to the REWF, readers familiar with steps (II)–(III) of Harrison's scheme may wish to skim Sections 2–4 and proceed rather quickly to Section 5 where the main results begin.

In Section 5, we prove our first main result on the "forest of trees" structure of the server-buffer graph. In Section 6 we prove our second main result. We prove the existence of a decoupled workload matrix and describe the associated control matrix. The results up to this point do not depend on the form of the holding cost function in the BCP. Later in Section 6, when the BCP has linear holding costs and our decoupled workload matrix is used, we prove that the cost function for the EWF is a linear function of workload. In Section 7, using the simplification of the EWF provided by our decoupled workload matrix when the BCP has linear holding costs, we give some sufficient conditions under which the least control process is a solution of the REWF. Under these conditions, we solve the BCP. In Section 8, we interpret for the original parallel server system the solution of the BCP obtained in Section 7 and we propose a control policy which we conjecture is asymptotically optimal in the heavy traffic limit. In Section 9, we give a three-buffer, three-server example of a parallel server system. For this example we carry out the steps from Sections 2-7 and we describe the control policy proposed in Section 8. In Section 10, we summarize some directions for further research.

1.5. *Notation and terminology.* The set of non-negative integers is denoted by $\mathbf{N}$ and the value $+\infty$ is denoted by ∞. We let $\mathbf{R}_+$ denote $[0, \infty)$. The m-dimensional ($m \geq 1$) Euclidean space is denoted by $\mathbf{R}^m$ and the m dimensional positive orthant is denoted by $\mathbf{R}^m_+ = \{x \in \mathbf{R}^m : x_i \geq 0 \text{ for } i = 1, \ldots, m\}$. Let $|x|$ denote the norm on $\mathbf{R}^m$ given by $|x| = \left(\sum_i x_i^2\right)^{1/2}$. Let $\{e_1, \ldots, e_m\}$ be the standard basis for $\mathbf{R}^m$. A sum over an empty index set

is defined to be zero. Vectors in $\mathbf{R}^m$ should be treated as column vectors unless indicated otherwise, inequalities between vectors should be interpreted componentwise, the transpose of a vector b will be denoted by b', the diagonal matrix with the entries of a vector b on its diagonal will be denoted by $diag(b)$, and the dot product of two vectors b and c in $\mathbf{R}^m$ will be denoted by $b \cdot c$.

For a matrix A, the i^{th} column of A will be denoted by A^i.

For each positive integer m, let $\mathbf{D}^m$ be the space of "Skorokhod paths" in $\mathbf{R}^m$ having time domain $\mathbf{R}_+$, i.e., $\mathbf{D}^m$ is the set of all functions $\omega : \mathbf{R}_+ \longrightarrow \mathbf{R}^m$ that are right continuous on $\mathbf{R}_+$ and have finite left limits on $(0, \infty)$. Let $\mathbf{D}^m_+ = \{\omega \in \mathbf{D}^m : \omega(0) \geq 0\}$. The member of $\mathbf{D}^m$ that stays at the origin in $\mathbf{R}^m$ for all time will be denoted by $\mathbf{0}$. For $\omega \in \mathbf{D}^m$,

$$||\omega||_t = \sup_{s\in[0,t]} |\omega(s)|, \text{ for each } t \geq 0. \tag{1.1}$$

Consider $\mathbf{D}^m$ to be endowed with the usual Skorokhod J_1-topology. Let $\mathcal{M}^m$ denote the Borel σ-algebra on $\mathbf{D}^m$ associated with the J_1-topology. For a non-negative integer m, given a probability space $(\Omega, \mathcal{F}, \mathbf{P})$, a m-dimensional stochastic process defined on this space is a collection $X = \{X(t) : t \in \mathbf{R}_+\}$ of measurable functions $X(t) : \Omega \to \mathbf{R}^m$ where Ω has the σ-algebra $\mathcal{F}$ and $\mathbf{R}^m$ has the Borel σ-algebra. Such a process X will be said to be non-decreasing, if each of its components is non-decreasing $\mathbf{P}$-a.s. All of the continuous-time stochastic processes in this paper are assumed to have sample paths in $\mathbf{D}^m$ for some $m \geq 1$. (We shall frequently use the term process in place of stochastic process.)

Suppose that $\{W^n\}_{n=1}^\infty$ is a sequence of processes with sample paths in $\mathbf{D}^m$ for some $m \geq 1$. Then we say that $\{W^n\}_{n=1}^\infty$ is tight if and only if the probability measures induced by the W^n on $(\mathbf{D}^m, \mathcal{M}^m)$ form a tight sequence, i.e., they form a weakly relatively compact sequence in the space of probability measures on $(\mathbf{D}^m, \mathcal{M}^m)$. The notation $W^n \Longrightarrow W$ as $n \to \infty$, where W is a process with sample paths in $\mathbf{D}^m$, will mean that the probability measures induced by the W^n on $(\mathbf{D}^m, \mathcal{M}^m)$ converge weakly to the probability measure on $(\mathbf{D}^m, \mathcal{M}^m)$ induced by W. If, for each n, W^n and W are defined on the same probability space, we write $W^n \to W$ uniformly on compact time intervals in probability (u.o.c. in prob.) as $n \to \infty$, if $\mathbf{P}(||W^n - W||_t \geq \epsilon) \to 0$ as $n \to \infty$ for each $\epsilon > 0$ and $t \geq 0$. In particular, if W is a continuous deterministic process and $W^n \Longrightarrow W$, then $W^n \to W$ u.o.c. in probability. This result is implicitly used several times in the proofs below to combine statements involving convergence in distribution to deterministic processes.

A filtered probability space is a quadruple $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}, \mathbf{P})$ where $(\Omega, \mathcal{F}, \mathbf{P})$ is a probability space and $\{\mathcal{F}_t\}$ is a filtration, i.e., a family of sub-σ-algebras of the σ-algebra $\mathcal{F}$ indexed by $t \in \mathbf{R}_+$ such that $\mathcal{F}_s \subset \mathcal{F}_t$ whenever $0 \leq s \leq t < \infty$. An m-dimensional process $X = \{X(t) : t \in \mathbf{R}_+\}$ defined on such a filtered probability space is said to be adapted if for each $t \geq 0$ the function $X(t) : \Omega \to \mathbf{R}^m$ is measurable when Ω has the σ-algebra $\mathcal{F}_t$ and $\mathbf{R}^m$ has its Borel σ-algebra. Given a probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}, \mathbf{P})$, a vector $\theta \in \mathbf{R}^m$, an $m \times m$ symmetric, strictly positive definite matrix Σ, an $\{\mathcal{F}_t\}$-Brownian motion with statistics (θ, Σ) starting at the origin, is an m-dimensional process on $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}, \mathbf{P})$ such that the following hold under $\mathbf{P}$:

(a) X is an m-dimensional Brownian motion with continuous sample paths such that $X(0) = 0$ $\mathbf{P}$-a.s.,
(b) $\{X_i(t) - \theta_i t, \mathcal{F}_t, t \geq 0\}$ is a martingale for $i = 1, \ldots, m$,
(c) $\{(X_i(t) - \theta_i t)(X_j(t) - \theta_j t) - \Sigma_{ij} t, \mathcal{F}_t, t \geq 0\}$ is a martingale for $i, j = 1, \ldots, m$.

In this definition, the filtration $\{\mathcal{F}_t\}$ may be larger than the one generated by X; however for each $t \geq 0$, under $\mathbf{P}$, the σ-algebra $\mathcal{F}_t$ is independent of the increments of X from t onward. The parameter θ is called the drift of the Brownian motion X and Σ is called the covariance matrix of X.

2. Parallel server system.

In this section we describe our model of a parallel server system. In the following section, we introduce the notion of heavy traffic for a sequence of such systems. The setup in this section and the next one is similar to that used in Bell and Williams [7].

2.1. *System structure.*

We consider a parallel server system (see Figure 1) consisting of a positive, finite number $\mathbb{I}$ of infinite capacity buffers (job classes) for holding jobs awaiting service, indexed by $i = 1, \ldots, \mathbb{I}$, and a positive, finite number $\mathbb{K}$ of (non-identical) servers working in parallel indexed by $k = 1, \ldots, \mathbb{K}$. Customers arrive to each of the buffers from outside the system. Arrivals to buffer i are called class i jobs. Jobs within each buffer are ordered according to their arrival times with the job that arrived the longest time ago being at the head of the line. Each job that enters the system requires a single service by a server before it leaves the system. Service of a given job class i by a given server k is called a processing activity. A single server k may be capable of processing several different job classes and a single job class i may be capable of being processed by one of several servers. To describe the available processing activities, it is assumed that there are a positive, finite number $\mathbb{J}$ of processing activities, indexed by

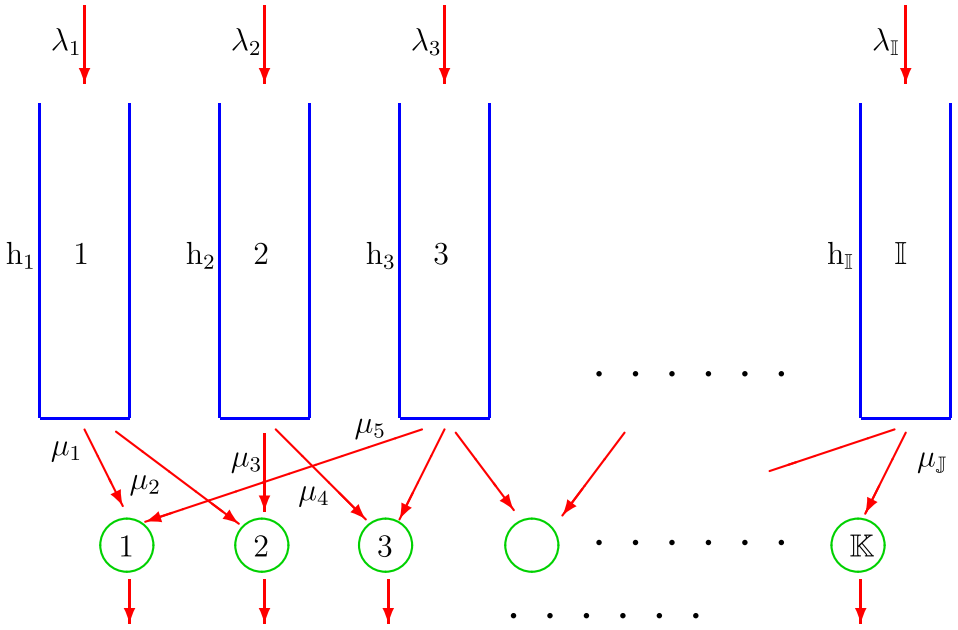

FIG 1. *Parallel server system.*

$j = 1, \ldots, \mathbb{J}$, where $\mathbb{J} \leq \mathbb{I} \cdot \mathbb{K}$. Each activity j, serves a single buffer $i(j)$ and is performed by a single server $k(j)$. The relations between the activities and buffers, and activities and servers, are specified by two deterministic matrices C, A where C is an $\mathbb{I} \times \mathbb{J}$ matrix with

$$C_{ij} = \begin{cases} 1 & \text{if activity } j \text{ processes class } i, \\ 0 & \text{otherwise,} \end{cases} \tag{2.1}$$

and A is a $\mathbb{K} \times \mathbb{J}$ matrix with

$$A_{kj} = \begin{cases} 1 & \text{if server } k \text{ performs activity } j, \\ 0 & \text{otherwise.} \end{cases} \tag{2.2}$$

Each activity j has exactly one class $i(j)$ and one server $k(j)$ associated with it, and so each column of C and each column of A contains the number one exactly once. We assume that each job class is capable of being served by at least one processing activity and each server is capable of performing at least one processing activity, and so each row of C and each row of A contains the number one at least once. Once a job starts being served at a server it remains there until its service is complete, even if its service is

suspended for some time. A server may not start on a new job of class i until it has finished any class i job that it is working on or that it has in suspension. When taking a job from a buffer, a server always takes the job at the head of the line. A server may not work unless it has a job to work on. It is assumed that the system is initially empty. For later use, we let $\mathcal{I} = \{1, \ldots, \mathbb{I}\}, \mathcal{J} = \{1, \ldots, \mathbb{J}\}$ and $\mathcal{K} = \{1, \ldots, \mathbb{K}\}$.

2.2. *Stochastic primitives.* All random variables and stochastic processes in our parallel server model are defined on a complete probability space $(\Omega, \mathcal{F}, \mathbf{P})$. The expectation operator under $\mathbf{P}$ is denoted by $\mathbf{E}$. For each buffer $i \in \mathcal{I}$, there is a sequence of strictly positive, independent and identically distributed (i.i.d.) random variables $\{u_i(l), l = 1, 2 \ldots\}$, with mean $\lambda_i^{-1} \in (0, \infty)$ and squared coefficient of variation (variance divided by the square of the mean) $a_i^2 \in [0, \infty)$. The random variable $u_i(l)$ represents the interarrival time between the $(l-1)^{th}$ and l^{th} customer to buffer i; by convention, the "0^{th} arrival" occurs at time zero. Let

$$\zeta_i(n) = \sum_{l=1}^{n} u_i(l), \quad n = 1, 2, \ldots, \tag{2.3}$$

and define

$$E_i(t) = \sup\{n \geq 0 : \zeta_i(n) \leq t\} \qquad \text{for all} \quad t \geq 0. \tag{2.4}$$

Then $E_i(t)$ is the number of arrivals to buffer i that have occurred in $[0, t]$, and λ_i is the long run arrival rate to buffer i. For each activity $j \in \mathcal{J}$, there is a sequence of strictly positive i.i.d. random variables $\{v_j(l), l = 1, 2, \ldots\}$ with mean $\mu_j^{-1} \in (0, \infty)$ and squared coefficient of variation $b_j^2 \in [0, \infty)$. The random variable $v_j(l)$ is the amount of service time required by the l^{th} job processed by activity j, and μ_j is the long run rate at which activity j could process its associated class of job $i(j)$ if the associated server $k(j)$ worked continuously and exclusively on this class. For $j \in \mathcal{J}$, let $\eta_j(0) = 0$,

$$\eta_j(n) = \sum_{l=1}^{n} v_j(l), \qquad n = 1, 2, \ldots, \tag{2.5}$$

and

$$S_j(t) = \sup\{n \geq 0 : \eta_j(n) \leq t\} \qquad \text{for all} \quad t \geq 0. \tag{2.6}$$

Then $S_j(t)$ is the number of jobs that activity j could process up to time t if the server $k(j)$ worked continuously and exclusively on class $i(j)$ jobs. The interarrival time sequences $\{u_i(l) : l = 1, 2, \ldots\}, i \in \mathcal{I}$, and service time sequences $\{v_j(l), l = 1, 2, \ldots\}, j \in \mathcal{J}$, are all assumed to be mutually independent.

2.3. *Scheduling control.* The system is controlled by specifying how each server is to allocate its time to its processing activities. The setup described here is fairly general. It allows for dynamic sequencing and alternate routing of jobs. For example, if server k performs more than one activity (i.e., $A_{kj} \neq 0$ for more than one j), then once service of a job is complete, server k can make a sequencing decision, i.e., which activity to perform next. If a given job class i can be processed by more than one activity (i.e., $C_{ij} \neq 0$ for more than one j), then class i may be serviced by one of a collection of servers and so simple alternate routing capabilities are encompassed here.

Formally, scheduling control is exerted through specification of a $\mathbb{J}$-dimensional stochastic process, $T = \{T(t), t \geq 0\}$ where

$$T(t) = (T_1(t), \ldots, T_{\mathbb{J}}(t))' \qquad \text{for} \quad t \geq 0, \tag{2.7}$$

and $T_j(t)$ is the cumulative amount of time devoted to activity j by server $k(j)$ in the time interval $[0, t]$. The control process T must satisfy certain natural constraints that go along with its interpretation (see (2.11)–(2.15)) below. For each $t \geq 0$, let

$$I(t) = \mathbf{1}t - AT(t), \tag{2.8}$$

where $\mathbf{1}$ is the $\mathbb{K}$-dimensional vector of ones. Then for each $k \in \mathcal{K}$, $I_k(t)$ is the cumulative amount of time that server k has been idle up to time t. The (cumulative) idle-time process, $I(\cdot)$, is continuous and non-decreasing in all of its components. This implies that T is Lipschitz continuous with Lipschitz constant equal to one. For each j, $S_j(T_j(t))$ is the number of jobs processed by activity j in the time interval $[0, t]$. For each $i \in \mathcal{I}$, let

$$Q_i(t) = E_i(t) - \sum_{j=1}^{\mathbb{J}} C_{ij} S_j(T_j(t)), \tag{2.9}$$

which we write in the vector form (with a slight abuse of notation for $S(T(t))$) as

$$Q(t) = E(t) - CS(T(t)). \tag{2.10}$$

Then $Q_i(t)$ is interpreted as the number of class i jobs that are either in queue or in the process of being served at time t. The following properties are assumed for any scheduling control T with associated queue-length Q and idle-time process I. For each $i \in \mathcal{I}$, $j \in \mathcal{J}$, $k \in \mathcal{K}$,

$$T_j(t) \in \mathcal{F} \quad \text{for each} \quad t \geq 0, \tag{2.11}$$

(2.12) T_j is Lipschitz continuous with a Lipschitz constant of one,

(2.13) T_j is non-decreasing and $T_j(0) = 0$,

(2.14) I_k is continuous, non-decreasing, and $I_k(0) = 0$,

(2.15) $Q_i(t) \geq 0$ for all $t \geq 0$.

For later reference, we collect here the queueing system equations satisfied by Q and I:

$$Q(t) = E(t) - CS(T(t)), \quad t \geq 0, \tag{2.16}$$

$$I(t) = \mathbf{1}t - AT(t), \quad t \geq 0, \tag{2.17}$$

where T, Q and I satisfy the properties (2.11)–(2.15). In addition to the properties mentioned above one might expect that T should satisfy some additional non-anticipating property. Even though this is a reasonable assumption to make, we have not restricted T a priori in this way. However, the policy that we propose in Section 8 is non-anticipating.

The cost function we shall use involves holding costs associated with the expense of holding jobs of each class in the system until they have completed service. We defer the precise description of this cost function to the next section, since it is formulated in terms of normalized queue-lengths, where the normalization is in diffusion scale. Indeed, in the next section, we describe the sequence of parallel server systems to be used in formulating the notion of heavy traffic asymptotic optimality.

3. Sequence of systems, heavy traffic and the cost function. For the parallel server system described in the last section, the problem of finding a control policy that minimizes a cost associated with holding jobs in the system is notoriously difficult. One possible means for discriminating between policies is to look for policies that outperform others in some asymptotic regime. Here we regard the parallel server system as a member of a sequence of systems indexed by r that is approaching heavy traffic (this notion is defined below). In this asymptotic regime, the queue-length process is normalized with diffusive scaling – this corresponds to viewing the system over long intervals of time of order r^2 (where r will tend to infinity in the asymptotic limit) and regarding a single job as only having a small contribution to the overall cost of storage, where this is quantified to be of order $1/r$.

3.1. *Sequence of systems.* Consider a sequence of parallel server systems indexed by r, where r tends to infinity through a sequence of values in $[1, \infty)$. The r^{th} system has the same basic structure as described in Section 2, except that the arrival and service rates and scheduling control are allowed to vary

with r. We denote this dependence on r by appending a superscript r to all of the relevant quantities. We assume that the interarrival and service times are given for each $r \geq 1$, $i \in \mathcal{I}, j \in \mathcal{J}$, by

$$u_i^r(l) = \frac{1}{\lambda_i^r}\check{u}_i(l), \qquad v_j^r(l) = \frac{1}{\mu_j^r}\check{v}_j(l), \quad \text{for} \quad l = 1, 2, \ldots, \tag{3.1}$$

where $\check{u}_i(l), \check{v}_i(l)$, are independent of r, with mean 1 and squared coefficient of variation a_i^2, respectively b_i^2. The sequences $\{\check{u}_i(l), l = 1, 2, \ldots\}, \{\check{v}_j(l),\ l = 1, 2 \ldots\}$ are mutually independent sequences of i.i.d. random variables. This setup is convenient for allowing the sequence of systems to approach the heavy traffic limit by simply changing arrival and service rates while keeping the underlying sources of variability $\check{u}_i(l), \check{v}_j(l)$ unaffected. We make the following assumption about the first order parameters for our sequence of systems.

ASSUMPTION 3.1. *There are vectors $\lambda \in \mathbf{R}_+^{\mathbb{I}}, \mu \in \mathbf{R}_+^{\mathbb{J}}$ such that*

(i) $\lambda_i > 0$ for all $i \in \mathcal{I}$, $\mu_j > 0$ for all $j \in \mathcal{J}$,
(ii) $\lambda^r \to \lambda$ and $\mu^r \to \mu$, as $r \to \infty$.

3.2. *Heavy traffic and fluid model.* In [17], Harrison proposed a notion of heavy traffic for stochastic processing networks with scheduling control. Given the parameters λ, μ from Assumption 3.1, for our sequence of parallel server systems, his notion is the same as Assumption 3.2 below. Henceforth, we let $R = C\mathit{diag}(\mu)$.

ASSUMPTION 3.2. *There is a unique optimal solution (ρ^*, x^*) of the linear program:*

$$\textit{minimize}\, \rho \quad \textit{subject to}\, Rx = \lambda, \quad Ax \leq \rho\mathbf{1} \quad \textit{and} \quad x \geq 0. \tag{3.2}$$

Moreover, this solution is such that $\rho^* = 1$ and $Ax^* = \mathbf{1}$.

A fluid model solution (with zero initial condition) is a triple of continuous deterministic functions $(\bar{Q}, \bar{T}, \bar{I})$ defined on $[0, \infty)$, where $\bar{Q}$ takes values in $\mathbf{R}_+^{\mathbb{I}}$, $\bar{T}$ takes values in $\mathbf{R}_+^{\mathbb{J}}$ and $\bar{I}$ takes values in $\mathbf{R}_+^{\mathbb{K}}$, such that

$$\bar{Q}(t) = \lambda t - R\bar{T}(t), \quad t \geq 0, \tag{3.3}$$

$$\bar{I}(t) = \mathbf{1}t - A\bar{T}(t), \quad t \geq 0, \tag{3.4}$$

and for all i, j, k,

$$\bar{T}_j \text{ is Lipschitz continuous with a Lipschitz constant of one,} \tag{3.5}$$

(3.6) $\bar{T}_j$ is non-decreasing, and $\bar{T}_j(0) = 0$,

(3.7) $\bar{I}_k$ is continuous, non-decreasing, and $\bar{I}_k(0) = 0$,

(3.8) $\bar{Q}_i(t) \geq 0$ for all $t \geq 0$.

A continuous function $\bar{T} : [0, \infty) \to \mathbf{R}_+^{\mathbb{J}}$ such that (3.3)–(3.8) hold is called a fluid control. The system is said to be balanced under $\bar{T}$ if the associated $\bar{Q}$ is constant in time. In this case, since the system starts empty, that means that $\bar{Q} \equiv 0$. The system is said to incur no idleness under $\bar{T}$ if $\bar{I} \equiv 0$, i.e., $A\bar{T} = \mathbf{1}t$ for all $t \geq 0$.

DEFINITION 3.1. *The fluid model is said to be in heavy traffic if the following two conditions hold:*

i) there is a unique fluid control $\bar{T}^*$ under which the fluid system is balanced, and
ii) under $\bar{T}^*$, the fluid system incurs no idleness.

In [34], Williams proved the following.

PROPOSITION 3.1. *The fluid model is in heavy traffic if and only if Assumption 3.2 holds.*

We impose the following heavy traffic assumption on our sequence of parallel server systems, henceforth.

ASSUMPTION 3.3. *(Heavy Traffic) For the sequence of parallel server systems defined in Section 3.1 satisfying Assumption 3.1, assume that Assumption 3.2 holds and that there is a vector $\theta \in \mathbf{R}^{\mathbb{I}}$ such that*

$$r(\lambda^r - R^r x^*) \to \theta, \quad \text{as} \quad r \to \infty, \tag{3.9}$$

where $R^r = C\text{diag}(\mu^r)$

Activities j for which $x_j^* > 0$ in Assumption 3.2 are called *basic.* Activities j for which $x_j^* = 0$ in Assumption 3.2 are called *non-basic.* Let $\mathbb{B}$ and $\mathbb{N}$ denote the number of basic and non-basic activities, respectively. It is assumed, without any loss of generality, that the first $\mathbb{B}$ activities are basic and the last $\mathbb{N}$ activities are non-basic. We let $\mathcal{B} = \{1, \ldots, \mathbb{B}\}$ and $\mathcal{N} = \{\mathbb{B} + 1, \ldots, \mathbb{J}\}$. Let

$$R = [H, J], \qquad A = [B, N], \tag{3.10}$$

be partitions of R, A according to basic and non-basic activities. Let K be the $(\mathbb{K} + \mathbb{N}) \times \mathbb{J}$ dimensional matrix

$$K = \begin{pmatrix} B & N \\ 0 & -I \end{pmatrix}, \tag{3.11}$$

where $-I$ is the negative of the $\mathbb{N}\times\mathbb{N}$ identity matrix. Later we will implicitly use the following property of K.

LEMMA 3.1. *Let K be given by (3.11), then $range(K) = \mathbf{R}^{\mathbb{K}+\mathbb{N}}$.*

PROOF. It is enough to show that for $l = 1, \ldots, \mathbb{K}+\mathbb{N}$, $e_l \in range(K)$. For $l \le \mathbb{K}$, since each server serves at least one basic activity, there exists a $j \in \mathbb{B}$ such that $l = k(j)$. Then by (2.2), (3.10) and (3.11), $K^j = e_l$ and so $e_l \in range(K)$. For $\mathbb{K} < l \le \mathbb{K}+\mathbb{N}$ and $j = l - \mathbb{K} + \mathbb{B}$, $j \in \mathcal{N}$ and by (2.2), (3.10) and (3.11), $K^j_{k(j)} = 1$ and $K^j_l = -1$. Since we already showed that $e_{k(j)} \in range(K)$, it follows that $e_{k(j)} - K^j = e_l \in range(K)$. □

3.3. *Diffusion scaling and cost function.* For a fixed r, and a scheduling control T^r, the associated queue-length and idle-time processes are given by equations (2.16) and (2.17) in Section 2, where the superscript r is appended to E, S, Q, I, and T there. The diffusion scaled queue-length and idle-time processes are defined by

$$\hat{Q}^r(t) = r^{-1}Q^r(r^2t), \qquad \hat{I}^r(t) = r^{-1}I^r(r^2t), \quad t \ge 0. \tag{3.12}$$

For a control T^r, and its associated diffusion scaled queue-length process, $\hat{Q}^r$, we define the expected cumulative discounted holding cost as follows:

$$\hat{J}^r(T^r) = \mathbf{E}\left(\int_0^\infty e^{-\gamma t} f(\hat{Q}^r(t))dt\right), \tag{3.13}$$

where $\gamma > 0$ is a fixed constant (discount factor) and $f : \mathbb{R}^{\mathbb{I}}_+ \to \mathbb{R}_+$ is a continuous function. We shall be especially interested in the case when f is linear and strictly positive except at the origin, i.e., $f(q) = h \cdot q$ for $q \in \mathbb{R}^{\mathbb{I}}_+$ where $h = (h_1, \ldots, h_{\mathbb{I}})'$, $h_i > 0$ for all $i \in \mathcal{I}$. To write equations for $\hat{Q}^r, \hat{I}^r$, it is convenient to consider centered diffusion scaled versions $\hat{E}^r, \hat{S}^r$ of the primitive processes E^r, S^r:

$$\hat{E}^r(t) = r^{-1}(E^r(r^2t) - \lambda^r r^2 t), \quad t \ge 0, \tag{3.14}$$

$$\hat{S}^r(t) = r^{-1}(S^r(r^2t) - \mu^r r^2 t), \quad t \ge 0, \tag{3.15}$$

and a deviation process $\hat{Y}^r$ that measures normalized deviations of server time allocations from the nominal allocations given by x^*:

$$\hat{Y}^r(t) = r^{-1}(x^* r^2 t - T^r(r^2 t)), \quad t \ge 0. \tag{3.16}$$

Also, we define the fluid scaled allocation process $\bar{T}^r$,

$$\bar{T}^r(t) = r^{-2}T^r(r^2t), \quad t \geq 0. \tag{3.17}$$

Let

$$\hat{U}^r = \begin{pmatrix} \hat{I}^r \\ -\hat{Y}^r_{\mathbb{N}} \end{pmatrix}, \tag{3.18}$$

where $\hat{Y}^r_{\mathbb{N}}$ consists of the components of $\hat{Y}^r$ that are indexed by non-basic activities. Upon substituting the above into the equations for Q^r, I^r, we obtain for $t \geq 0$:

$$\hat{Q}^r(t) = \hat{X}^r(t) + R^r\hat{Y}^r(t), \tag{3.19}$$

$$\hat{U}^r(t) = K\hat{Y}^r(t), \tag{3.20}$$

$$\hat{X}^r(t) = \hat{E}^r(t) - C\hat{S}^r(\bar{T}^r(t)) + r(\lambda^r - R^r x^*)t, \tag{3.21}$$

where $\hat{U}^r_k$ is continuous, non-decreasing and $\hat{U}^r_k(0) = 0$, for $k = 1, \ldots, \mathbb{K}+\mathbb{N}$, and $\hat{Q}^r_i(t) \geq 0$ for all $t \geq 0$ and $i = 1, \ldots, \mathbb{I}$. Combining (3.1) and Assumption 3.1 with the mutual independence of the stochastic primitive sequences of i.i.d. random variables $\{\check{u}_i(l)\}_{l=1}^{\infty}, i \in \mathcal{I}, \{\check{v}_j(l)\}_{l=1}^{\infty}, j \in \mathcal{J}$, we may deduce from the functional central limit theorem for renewal processes that

$$(\hat{E}^r, \hat{S}^r) \Rightarrow (\tilde{E}, \tilde{S}), \quad \text{as} \quad r \to \infty, \tag{3.22}$$

where $\tilde{E}, \tilde{S}$ are independent, $\tilde{E}$ is an $\mathbb{I}$-dimensional driftless Brownian motion that starts from the origin and has a diagonal covariance matrix whose i^{th} diagonal entry is $\lambda_i a_i^2$, and $\tilde{S}$ is a $\mathbb{J}$-dimensional driftless Brownian motion that starts from the origin and has a diagonal covariance matrix whose j^{th} diagonal entry is $\mu_j b_j^2$.

4. Brownian Control Problem and Equivalent Workload Formulation. In this section, following the method proposed by Harrison et al. [17, 19], we formulate a Brownian Control Problem (BCP) and its Equivalent Workload Formulation (EWF) as formal approximations to the control problem for the sequence of parallel server systems. We also show that the EWF can sometimes be further reduced, by the removal of some redundant controls, to a Reduced Equivalent Workload Formulation or REWF. Henceforth, we let $\bar{T}^*(t) = x^*t$, for all $t \geq 0$.

4.1. *Brownian Control Problem (BCP).*

DEFINITION 4.1. *(Admissible control for the BCP) An admissible control for the BCP is a $\mathbb{J}$-dimensional, adapted process $\tilde{Y} = \{\tilde{Y}(t), t \geq 0\}$*

defined on some filtered probability space $(\tilde{\Omega}, \tilde{\mathcal{F}}, \{\tilde{\mathcal{F}}_t\}, \tilde{\mathbf{P}})$ *which supports* $\mathbb{I}$*-dimensional adapted processes* $\tilde{Q}$ *and* $\tilde{X}$*, such that the following three properties hold under* $\tilde{\mathbf{P}}$*:*

(i) $\tilde{Q}(t) = \tilde{X}(t) + R\tilde{Y}(t) \in \mathbf{R}_+^{\mathbb{I}}$ *for all* $t \geq 0$, $\tilde{\mathbf{P}}$*-a.s.,*

(ii) $\tilde{U} \equiv \{K\tilde{Y}(t), t \geq 0\}$ *is non-decreasing and* $\tilde{U}(0) \geq 0$, $\tilde{\mathbf{P}}$*-a.s.,*

(iii) $\tilde{X}$ *is an* $\mathbb{I}$*-dimensional* $\{\tilde{\mathcal{F}}_t\}$*-Brownian motion starting at the origin, with drift* θ *and diagonal covariance matrix* Σ *whose* i^{th} *diagonal entry is equal to* $\lambda_i a_i^2 + \sum_{j=1}^{\mathbb{J}} C_{ij}\mu_j b_j^2 x_j^*$ *for* $i \in \mathcal{I}$*.*

We call $\tilde{Q}$ *the state process,* $(\tilde{Q}, \tilde{U})$ *the extended state processes and* $\tilde{X}$ *the Brownian motion, for the Brownian Control Problem under the control* $\tilde{Y}$*.*

REMARK 4.1. *Note that the filtered probability space with Brownian motion* $\tilde{X}$ *upon which* $\tilde{Y}$ *is defined is part of the specification of* $\tilde{Y}$*. In other words, this is a weak formulation of the control problem. In the subsequent text, when we refer to an admissible control* $\tilde{Y}$*, it will be implicit that this also carries with it a filtered probability space and a Brownian motion. The first* $\mathbb{K}$ *components of* $\tilde{U}$ *will sometimes be denoted by* $\tilde{I}$ *as they correspond to diffusion analogues of idle-time.*

DEFINITION 4.2. *(Brownian Control Problem-BCP) Determine the optimal value*

$$\tilde{J}^* = \inf_{\tilde{Y}} \tilde{J}(\tilde{Y}) \quad \text{where} \quad \tilde{J}(\tilde{Y}) \equiv \tilde{\mathbf{E}}\left(\int_0^\infty e^{-\gamma t} f(\tilde{Q}(t))dt\right), \tag{4.1}$$

where the infimum is taken over all admissible controls for the BCP and $\tilde{\mathbf{E}}$ *denotes expectation under the probability* $\tilde{P}$ *associated with* $\tilde{Y}$*. An admissible control* $\tilde{Y}^*$ *that achieves the infimum in (4.1) is called an optimal control for the BCP.*

The Brownian motion $\tilde{X}$ appearing in the BCP is the formal limit in distribution of $\hat{X}^r$ of (3.21). The functional central limit theorems for the independent renewal processes E^r, S^r and a time change theorem (together with the assumption that $\bar{T}^r \Rightarrow \bar{T}^*$), are used to derive the covariance matrix for this Brownian motion. The control process $\tilde{Y}$ in the BCP is a formal limit of the deviation processes $\hat{Y}^r$, where convergence of $\hat{Y}^r(0)$ to $\tilde{Y}(0)$ is not required. The initial condition for $\tilde{U}$ is relaxed from that in the prelimit to allow for the possibility of an initial jump in the queue-length process in the BCP. In fact, for the optimal solutions of the BCPs that we will solve here, such a jump will not occur and then the BCPs are equivalent to ones in which $\tilde{U}(0) = 0$.

4.2. *Equivalent Workload Formulation (EWF).* Harrison and Van Mieghem [19] showed that one can reduce the dimensionality of the Brownian Control Problem to that of an Equivalent Workload Formulation (EWF). We summarize the relevant theory below.

DEFINITION 4.3. *(Space of reversible displacements) Let*

$$\mathcal{R} = \{\delta \in \mathbf{R}^{\mathbb{I}} : \ \delta = Rx \text{ and } Kx = 0, x \in \mathbf{R}^{\mathbb{J}}\}. \tag{4.2}$$

One can think of the vector space $\mathcal{R}$ as follows. Given $\tilde{q} \in \mathbf{R}^{\mathbb{I}}_+, \tilde{q} > 0$, for any $\delta \in \mathcal{R}$ such that $\tilde{q} + \delta > 0$, using allowed controls in the BCP it is possible to instantaneously move the "queue-length" from $\tilde{q}$ to $\tilde{q} + \delta$ without incurring any idleness nor using any non-basic activities. Since $\mathcal{R}$ is a vector space such changes are reversible. The idea of the Equivalent Workload Formulation is to focus on the non-reversible displacements of the queue-length, i.e., those in $\mathcal{R}^{\perp}$.

DEFINITION 4.4. *(Workload dimension, workload matrix) Let $\mathbb{L}$ be the dimension of $\mathcal{R}^{\perp}$. Then $\mathbb{L}$ is called the workload dimension. Let M be any $\mathbb{L} \times \mathbb{I}$ dimensional matrix whose rows span $\mathcal{R}^{\perp}$. Then M is called a workload matrix.*

A simple formula for the workload dimension was established by Bramson and Williams [12] for Brownian Control Problems associated with general stochastic processing networks. When applied to parallel server systems, that result yields the following.

PROPOSITION 4.1. *The workload dimension $\mathbb{L} = \mathbb{I} + \mathbb{K} - \mathbb{B}$.*

The following result was proved by Harrison and Van Mieghem [19]. It is key to the reduction of the BCP to the EWF.

PROPOSITION 4.2. *Let M be an arbitrary workload matrix. Then there exists an $\mathbb{L} \times (\mathbb{K} + \mathbb{N})$ matrix G such that*

$$MR = GK. \tag{4.3}$$

The choice of G is usually not unique. We refer to G as a control matrix associated with M. For the remainder of this subsection and in the next, we fix a choice for M and an associated G. Let $\mathcal{W} = M\mathbf{R}^{\mathbb{I}}_+$ and define

$$g(w) = \inf\{f(q) : Mq = w, q \in \mathbf{R}^{\mathbb{I}}_+\}, \ w \in \mathcal{W}. \tag{4.4}$$

We will focus on the situation where the following assumption holds.

ASSUMPTION 4.1. *The mapping g given by (4.4) is a well defined, continuous function from $\mathcal{W}$ into $\mathbf{R}_+$ and the infimum in (4.4) is attained for each $w \in \mathcal{W}$. Moreover, there exists a continuous mapping $\phi : \mathcal{W} \to \mathbf{R}_+^{\mathbb{I}}$ such that for $w \in \mathcal{W}$, $\phi(w) \in \{q \in \mathbf{R}_+^{\mathbb{I}} : Mq = w, f(q) = g(w)\}$.*

It is known that Assumption 4.1 is satisfied if f is linear and for each $w \in \mathcal{W}$ the set $\{q \in \mathbf{R}_+^{\mathbb{I}} : Mq = w\}$ is compact, see [9]. Later, in Theorem 6.1, we show that Assumption 4.1 holds for our choice of decoupled workload matrix, M. Indeed, we shall prove that for this choice of workload matrix, g is a linear map when f is linear. This is quite a simplification since in general one can at most expect g to be a convex, piecewise linear function when f is linear.

DEFINITION 4.5. *(Admissible control for the EWF). An admissible control for the EWF is a $(\mathbb{K}+\mathbb{N})$-dimensional adapted process $\tilde{U}$ defined on some probability space $(\Lambda, \mathcal{E}, \{\mathcal{E}_t\}, \mathbf{Q})$, which supports $\mathbb{L}$-dimensional adapted processes $\tilde{W}$ and $\tilde{\xi}$, such that the following properties hold under $\mathbf{Q}$:*

(i) $\tilde{W}(t) = \tilde{\xi}(t) + G\tilde{U}(t) \in \mathcal{W}$ *for all* $t \geq 0$, $\mathbf{Q}$*-a.s.,*
(ii) $\tilde{U}$ *is non-decreasing,* $\tilde{U}(0) \geq 0$, $\mathbf{Q}$*-a.s.,*
(iii) $\tilde{\xi}$ *is an* $\mathbb{L}$*-dimensional* $\{\mathcal{E}_t\}$*-Brownian motion starting at the origin, with drift* $M\theta$ *and covariance matrix* $M\Sigma M'$.

We call $\tilde{W}$ the state process with Brownian motion $\tilde{\xi}$ for the EWF under the control $\tilde{U}$. We let $\mathcal{A}$ denote the set of admissible controls for the EWF. Note that this depends on the (fixed) choices of M and G.

REMARK 4.2. *Since range$(K) = \mathbf{R}^{\mathbb{K}+\mathbb{N}}$ (see Lemma 3.1), we do not need to add a constraint on the range of $\tilde{U}$. Note that the filtered probability space with Brownian motion $\tilde{\xi}$ upon which the control process $\tilde{U}$ is defined is part of the specification of $\tilde{U}$.*

DEFINITION 4.6. *(Equivalent Workload Formulation-EWF) Determine the optimal value*

$$\check{J}^* = \inf_{\tilde{U}} \check{J}(\tilde{U}) \quad \text{where} \quad \check{J}(\tilde{U}) = \mathbf{E}\left(\int_0^\infty e^{-\gamma t} g(\tilde{W}(t))dt\right), \tag{4.5}$$

where the infimum is taken over all admissible controls for the EWF and $\mathbf{E}$ denotes expectation under the probability $\mathbf{Q}$ associated with $\tilde{U}$. An admissible control that achieves the infimum in (4.5) is called an optimal control for the EWF.

REMARK 4.3. *Nominally, the values of* $\check{J}(\tilde{U})$ *and* $\check{J}^*$ *depend on the choice of* M *and* G*. However, since* M *and* G *are fixed, to lighten the notation, we have not explicitly indicated them here. Furthermore, as the following theorem shows, the value of* $\check{J}^*$ *is equal to* $\tilde{J}^*$*, and so in fact, this value is the same, regardless of the choice of* M *and* G*. Despite this, as we shall see, for some choices of* M *and* G*, it can be easier to see how to solve the EWF.*

The following theorem shows the equivalence of the BCP to the EWF. For a similar formulation of the BCP and the EWF this equivalence was proved by Harrison and Van Mieghem in [19]. For the formulation used here the equivalence can be proved in the same way as in Harrison and Williams [21].

THEOREM 4.1. *Suppose that Assumption 4.1 holds. The optimal value* $\tilde{J}^*$ *for the BCP is equal to the optimal value* $\check{J}^*$ *for the EWF.*

4.3. *Reduced EWF (REWF).* In this subsection we show how the EWF may sometimes be further reduced by reducing the matrix G. Let $\mathbb{P} = \mathbb{K}+\mathbb{N}$. Given an $\mathbb{L} \times \mathbb{D}$ matrix H for $1 \leq \mathbb{D} \leq \mathbb{P}$, the cone generated by the matrix H is defined as follows,

$$\mathcal{C}(H) = \left\{ v \in \mathbf{R}^{\mathbb{L}} : v = Hu, u \in \mathbf{R}^{\mathbb{D}}_{+} \right\} = \left\{ \sum_{j=1}^{\mathbb{D}} u_j H^j,\ u_j \in [0, \infty) \right\} = H\mathbf{R}^{\mathbb{D}}_{+},$$

where H^j denotes the j^{th} column of H.

LEMMA 4.1. *Suppose* $\hat{G}$ *is an* $\mathbb{L} \times \mathbb{D}$ *matrix where* $1 \leq \mathbb{D} \leq \mathbb{P}$ *and* $\mathcal{C}(G) = \mathcal{C}(\hat{G})$*. Let* $\tilde{U}$ *be an admissible control for the EWF. Then there exists a* $\mathbb{D}$*-dimensional adapted process* $\hat{U}$ *defined on the same filtered probability space as* $\tilde{U}$ *such that:*

(i) $G\tilde{U} = \hat{G}\hat{U}$*,*
(ii) $\hat{U}$ *is non-decreasing with* $\hat{U}(0) \geq 0$ *almost surely.*

PROOF. Let G^j be the j^{th} column of G. Then, since $\mathcal{C}(G) = \mathcal{C}(\hat{G})$ there exists $u^j \in \mathbf{R}^{\mathbb{D}}_{+}$ such that $G^j = \hat{G}u^j$. Let

$$\hat{U} = u^1\tilde{U}_1 + \cdots + u^{\mathbb{P}}\tilde{U}_{\mathbb{P}}. \tag{4.6}$$

It is straight forward to verify the properties $(i), (ii)$ and that $\hat{U}$ is an adapted process. □

DEFINITION 4.7. *(Admissible control for the REWF) For some* $1 \le \mathbb{D} \le \mathbb{P}$*, let* $\hat{G}$ *be an* $\mathbb{L} \times \mathbb{D}$ *matrix such that* $\mathcal{C}(\hat{G}) = \mathcal{C}(G)$*. An admissible control for the REWF associated with* $\hat{G}$ *is a* $\mathbb{D}$*-dimensional adapted process* $\hat{U} = \{\hat{U}(t), t \ge 0\}$ *defined on some filtered probability space* $(\Lambda, \mathcal{E}, \{\mathcal{E}_t\}, \mathbf{Q})$*, which supports* $\mathbb{L}$*-dimensional adapted processes* $\hat{W}$ *and* $\hat{\xi}$*, such that the following properties hold under* $\mathbf{Q}$*:*

(i) $\hat{W}(t) = \hat{\xi}(t) + \hat{G}\hat{U}(t) \in \mathcal{W}$ *for all* $t \ge 0$*,* $\mathbf{Q}$*-a.s.,*
(ii) $\hat{U}$ *is non-decreasing with* $\hat{U}(0) \ge 0$*,* $\mathbf{Q}$*-a.s.,*
(iii) $\hat{\xi}$ *is an* $\mathbb{L}$*-dimensional* $\{\mathcal{E}_t\}$*-Brownian motion starting at the origin, with drift* $M\theta$ *and covariance matrix* $M\Sigma M'$*.*

We let $\hat{\mathcal{A}}(\hat{G})$ *denote the set of admissible controls for the REWF associated with* $\hat{G}$*.*

DEFINITION 4.8. *(REWF) Suppose that* $\hat{G}$ *is an* $\mathbb{L} \times \mathbb{D}$ *matrix for some* $1 \le \mathbb{D} \le \mathbb{P}$ *such that* $\mathcal{C}(G) = \mathcal{C}(\hat{G})$*. The REWF associated with* $\hat{G}$ *is to determine the optimal value*

$$\hat{J}^{*,\hat{G}} = \inf_{\hat{U}} \hat{J}^{\hat{G}}(\hat{U}) \quad \text{where} \quad \hat{J}^{\hat{G}}(\hat{U}) = \mathbf{E}\left(\int_0^\infty e^{-\gamma t} g(\hat{W}(t))dt\right), \tag{4.7}$$

where the infimum is taken over all $\hat{U} \in \hat{\mathcal{A}}(\hat{G})$ *and* $\mathbf{E}$ *denotes expectation under the probability* $\mathbf{Q}$ *associated with* $\hat{U}$*. An admissible control* $\hat{U}$ *that achieves the infimum in (4.7) is called an optimal control for the REWF associated with* $\hat{G}$*.*

Each $\mathbb{L} \times \mathbb{D}$ matrix $\hat{G}$, with the property that $\mathcal{C}(G) = \mathcal{C}(\hat{G})$, gives rise to an REWF. The following lemma shows that all REWFs are equivalent to the EWF.

THEOREM 4.2. *For some* $1 \le \mathbb{D} \le \mathbb{P}$*, let* $\hat{G}$ *be an* $\mathbb{L} \times \mathbb{D}$ *matrix such that* $\mathcal{C}(G) = \mathcal{C}(\hat{G})$*. Then, the optimal value* $\hat{J}^{*,\hat{G}}$ *of the REWF associated with* $\hat{G}$ *is the same as the optimal value* $\check{J}^*$ *of the EWF associated with* G*.*

PROOF. By Lemma 4.1 for each $\tilde{U} \in \mathcal{A}$ there exists a $\hat{U} \in \hat{\mathcal{A}}(\hat{G})$ defined on the same filtered probability space as $\tilde{U}$ such that $\tilde{G}\tilde{U} = \hat{G}\hat{U}$. Then with $\tilde{\xi} = \hat{\xi}$, $\tilde{W} = \tilde{\xi} + G\tilde{U} = \hat{\xi} + \hat{G}\hat{U} = \hat{W}$, and therefore, $\check{J}(\tilde{U}) = \hat{J}^{\hat{G}}(\hat{U})$. Also, by a similar proof to that for Lemma 4.1 (by switching the roles of G and $\hat{G}$), for each $\hat{U} \in \hat{\mathcal{A}}(\hat{G})$ there exists a $\tilde{U} \in \mathcal{A}$ defined on the same probability space as $\hat{U}$ such that $\hat{G}\hat{U} = G\tilde{U}$ and again $\hat{J}^{\hat{G}}(\hat{U}) = \check{J}(\tilde{U})$. It follows that $\check{J}^* = \hat{J}^{*,\hat{G}}$. □

4.4. *Harrison's choice of workload matrix and the dual program.* Harrison [17] provided an alternative description of $\mathcal{R}^{\perp}$ and proposed a choice for M and G, as follows.

DEFINITION 4.9. *(Dual Program DP)*

$$\text{maximize } y \cdot \lambda \text{ subject to } y'R \leq z'A,\ z \cdot \mathbf{1} = 1 \text{ and } z \geq 0. \tag{4.8}$$

PROPOSITION 4.3. *Let* $\{(y^1, z^1), \ldots, (y^{\tilde{\mathbb{L}}}, z^{\tilde{\mathbb{L}}})\}$ *be the set of extremal optimal solutions of the dual program. Let* $\{(y^1, z^1), \ldots, (y^{\mathbb{L}}, z^{\mathbb{L}})\}$ *be such that* $\{y^1, \ldots, y^{\mathbb{L}}\}$ *is a maximal linearly independent subset of* $\{y^1, \ldots, y^{\tilde{\mathbb{L}}}\}$. *Then,*

$$\mathcal{R}^{\perp} = \text{span}\{y^1, \ldots, y^{\mathbb{L}}\}. \tag{4.9}$$

Proposition 4.3 suggests a choice for a workload matrix M. In particular, fix $\{y^1, \ldots, y^{\mathbb{L}}\}$ and let

$$M = \begin{pmatrix} y^1 \\ \vdots \\ y^{\mathbb{L}} \end{pmatrix}, \tag{4.10}$$

where we abuse notation and we think of $y^1, \ldots, y^{\mathbb{L}}$ as row vectors. Following Harrison [17], let

$$G = [\Pi \ \ \Pi N - MJ], \tag{4.11}$$

where

$$\Pi = \begin{pmatrix} z^1 \\ \vdots \\ z^{\mathbb{L}} \end{pmatrix}, \tag{4.12}$$

and vectors $z^1, \ldots, z^{\mathbb{L}}$ are viewed as row vectors that accompany $y^1, \ldots, y^{\mathbb{L}}$ as in Proposition 4.3. Then, the matrix G given by (4.11) satisfies the relation $MR = GK$ in Proposition 4.2. Moreover, from the dual program it follows that $G \geq 0$.

5. Structure of the server-buffer graph.

5.1. *Server-buffer graph.*

DEFINITION 5.1. *(Server-buffer graph $\mathcal{G}$) The graph $\mathcal{G}$ in which servers and buffers form the nodes, and undirected edges between the nodes are given by basic activities, is called the server-buffer graph.*

The following theorem was established in two stages. In [18], Harrison and López proved the equivalence of (i)–(iii) below and subsequently Williams [34] showed that (i)–(iii) are equivalent to (iv).

THEOREM 5.1. *The following conditions are equivalent:*

(i) the dual program (4.8) has a unique solution (y^*, z^*),
(ii) the number of basic activities $\mathbb{B} = \mathbb{I} + \mathbb{K} - 1$,
(iii) all servers communicate via basic activities,
(iv) the graph $\mathcal{G}$ *is a tree.*

REMARK 5.1. *It follows from Proposition 2 in [18] that for the solution in (i),* $y^* > 0$, $z^* > 0$, $y^*H = z^*B$ *and* $y^*J < z^*N$.

A parallel server system that satisfies any of the equivalent conditions (i)–(iv) of Theorem 5.1 is said to satisfy the complete resource pooling (CRP) condition. From the results of Bramson and Williams [12] embodied in Proposition 4.1, we know that in general $\mathbb{L} = \mathbb{I} + \mathbb{K} - \mathbb{B}$, and so Theorem 5.1 characterizes the situation when $\mathbb{L} = 1$. In the next subsection, we state and prove a result that generalizes part (iv) of Theorem 5.1 to $\mathbb{L} > 1$.

5.2. *Forest of trees structure.* Our first main result is the following.

THEOREM 5.2. *The workload dimension* $\mathbb{L}$ *is equal to the number of connected components in the server-buffer graph* $\mathcal{G}$. *Indeed, the server-buffer graph* $\mathcal{G}$ *is a forest of* $\mathbb{L}$ *trees.*

To prove Theorem 5.2, we introduce the following enumeration scheme, which we shall use henceforth. Recall that $\mathcal{I} = \{1, ..., \mathbb{I}\}$, $\mathcal{K} = \{1, ..., \mathbb{K}\}$ and $\mathcal{B} = \{1, ..., \mathbb{B}\}$. Let $\mathbb{M}$ be the number of connected components in $\mathcal{G}$ denoted by $\mathcal{T}_1, \ldots, \mathcal{T}_{\mathbb{M}}$. For each $m \in \mathcal{M} = \{1, \ldots, \mathbb{M}\}$ let $\mathcal{I}_m$ be the subset of $\mathcal{I}$ that indexes the buffers in $\mathcal{T}_m$. Similarly, for each $m \in \mathcal{M}$ let $\mathcal{K}_m$, respectively $\mathcal{B}_m$, be the subset of $\mathcal{K}$, respectively $\mathcal{B}$, that indexes the servers, respectively the basic activities, in $\mathcal{T}_m$. The cardinalities of $\mathcal{I}_m$, $\mathcal{K}_m$ and $\mathcal{B}_m$ are denoted by $\mathbb{I}_m$, $\mathbb{K}_m$ and $\mathbb{B}_m$, respectively. Note that $\mathbb{B}_m$ is the number of edges in $\mathcal{T}_m$. The set $\mathcal{N} = \{\mathbb{B} + 1, ..., \mathbb{J}\}$ of non-basic activities, has cardinality $\mathbb{N}$. For each $m \in \mathcal{M}$, let $\mathcal{N}^{m,c}$ be the set of non-basic activities that consume material from buffers in $\mathcal{T}_m$ and let $\mathcal{N}^{m,p}$ be the set of non-basic activities that are processed by servers in $\mathcal{T}_m$. For each $m, m' \in \mathcal{M}$ let $\mathcal{N}^m_{m'}$ be the set of non-basic activities that consume material from buffers in $\mathcal{T}_m$ and that are processed by servers in $\mathcal{T}_{m'}$. Note that $\mathcal{N}^m_{m'} = \mathcal{N}^{m,c} \cap \mathcal{N}^{m',p}$. Let $\mathbb{N}^{m,c}$,

$\mathbb{N}^{m',p}$ and $\mathbb{N}^m_{m'}$ be the cardinalities of $\mathcal{N}^{m,c}$, $\mathcal{N}^{m',p}$, and $\mathcal{N}^m_{m'}$, respectively. Then for any $m, m' \in \mathcal{M}$,

$$\mathbb{N}^{m,c} = \sum_{l=1}^{\mathbb{M}} \mathbb{N}^m_l \qquad \text{and} \qquad \mathbb{N}^{m',p} = \sum_{l=1}^{\mathbb{M}} \mathbb{N}^l_{m'}.$$

We can and do choose the enumeration of buffers, servers, basic and non-basic activities so that the following properties hold.

Convention 5.1.

i) If buffer $i \in \mathcal{I}_m$ and buffer $i' \in \mathcal{I}_{m'}$ where $m < m'$, then $i < i'$,
ii) if server $k \in \mathcal{K}_m$ and server $k' \in \mathcal{K}_{m'}$ where $m < m'$, then $k < k'$,
iii) if i and i' are distinct buffers such that $i < i'$ and if j and j' are basic activities such that $i = i(j)$ and $i' = i(j')$, then $j < j'$,
iv) if $j \in \mathcal{N}^{m,c}$ and $j' \in \mathcal{N}^{m',c}$ where $m < m'$, then $j < j'$,
v) if $j' \in \mathcal{N}^m_{m'}$ and $j'' \in \mathcal{N}^m_{m''}$ where $m' < m''$, then $j' < j''$.

This convention induces the following partitions of R and A,

$$R = \begin{pmatrix} H^1 & 0 & \dots & 0 & J^1 & 0 & \dots & 0 \\ 0 & H^2 & \ddots & \vdots & 0 & J^2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & H^{\mathbb{M}} & 0 & \dots & 0 & J^{\mathbb{M}} \end{pmatrix}, \tag{5.1}$$

$$A = \begin{pmatrix} B^1 & 0 & \dots & 0 & N^1_1 & 0 & \dots & 0 & \dots & N^{\mathbb{M}}_1 & 0 & \dots & 0 \\ 0 & B^2 & \ddots & \vdots & 0 & N^1_2 & \ddots & \vdots & \dots & 0 & N^{\mathbb{M}}_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots & \ddots & \ddots & 0 & \dots & \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & B^{\mathbb{M}} & 0 & \dots & 0 & N^1_{\mathbb{M}} & \dots & 0 & \dots & 0 & N^{\mathbb{M}}_{\mathbb{M}} \end{pmatrix}, \tag{5.2}$$

where H^m is the $\mathbb{I}_m \times \mathbb{B}_m$ matrix of rates at which basic activities in $\mathcal{T}_m$ consume material from buffers in $\mathcal{T}_m$. The $\mathbb{I}_m \times \mathbb{N}^{m,c}$ matrix J^m is the matrix of average processing rates for non-basic activities in $\mathcal{N}^{m,c}$ that consume material from buffers in $\mathcal{T}_m$. The $\mathbb{K}_m \times \mathbb{B}_m$ matrix B^m has a solitary one in its j^{th} column in the row corresponding to the server that processes the j^{th} basic activity in $\mathcal{T}_m$. The matrix $N^m_{m'}$ is a $\mathbb{K}_{m'} \times \mathbb{N}^m_{m'}$ matrix of zeros and ones that signals which servers in $\mathcal{T}_{m'}$ process jobs from buffers in $\mathcal{T}_m$ using non-basic activities, i.e., it has a solitary one in its j^{th} column for the j^{th} non-basic activity that consumes material from a buffer in $\mathcal{T}_m$ and is processed by a server in $\mathcal{T}_{m'}$. The row that contains the one corresponds

to the server that performs the processing of that activity. For each m, the enumeration induces a partition of J^m:

$$J^m = \left[J_1^m, \ldots, J_{\mathbb{M}}^m\right],$$

where $J_{m'}^m$ is the $\mathbb{I}_m \times \mathbb{N}_{m'}^m$ matrix of consumption rates for non-basic activities in $\mathcal{N}_{m'}^m$. In the following proof, depending on the context, a vector x sometimes denotes a row vector and at other times it denotes a column vector; whichever is meant will be clear from the context.

PROOF OF THEOREM 5.2. Since each $\mathcal{T}_m$ is connected and $\mathbb{I}_m + \mathbb{K}_m$ is the number of nodes in $\mathcal{T}_m$, it follows immediately that

$$\mathbb{B}_m \geq \mathbb{I}_m + \mathbb{K}_m - 1, \quad \text{for} \quad m = 1, \ldots, \mathbb{M}. \tag{5.3}$$

Summing over all connected components, we obtain that

$$\mathbb{B} = \sum_{m=1}^{\mathbb{M}} \mathbb{B}_m \geq \sum_{m=1}^{\mathbb{M}} (\mathbb{I}_m + \mathbb{K}_m - 1) = \mathbb{I} + \mathbb{K} - \mathbb{M}. \tag{5.4}$$

Rearranging the terms in the last equation and using Proposition 4.1 we obtain that

$$\mathbb{M} \geq \mathbb{I} + \mathbb{K} - \mathbb{B} = \mathbb{L}. \tag{5.5}$$

Hence the workload dimension $\mathbb{L}$ is less than or equal to the number $\mathbb{M}$ of connected components in $\mathcal{G}$. To establish that $\mathbb{M} = \mathbb{L}$, it is enough to show that $\mathbb{B}_m = \mathbb{I}_m + \mathbb{K}_m - 1$ for $1 \leq m \leq \mathbb{M}$. For an argument by contradiction, suppose that for some index m^*,

$$\mathbb{B}_{m^*} > \mathbb{I}_{m^*} + \mathbb{K}_{m^*} - 1. \tag{5.6}$$

From the heavy traffic Assumption 3.2 we have that $x^* = (x_{\mathbb{B}}^*, 0_{\mathbb{N}})$, where $x_{\mathbb{B}}^*$ is a positive $\mathbb{B}$-dimensional vector of nominal rate allocations for basic activities and $0_{\mathbb{N}}$ is an $\mathbb{N}$-dimensional vector of zeros whose entries are nominal allocations for non-basic activities. With a slight abuse of notation, we write

$$x_{\mathbb{B}}^* = (x_1^*, ..., x_{\mathbb{M}}^*), \tag{5.7}$$

where x_m^* is a $\mathbb{B}_m$-dimensional vector of allocations for the basic activities in $\mathcal{B}_m$, $m = 1, \ldots, \mathbb{M}$. Similarly, we write

$$\lambda = (\lambda_1, \ldots, \lambda_{\mathbb{M}}), \tag{5.8}$$

where λ_m is an $\mathbb{I}_m$-dimensional vector of average arrival rates for the buffers in $\mathcal{I}_m$ and

$$\mathbf{1}_{\mathbb{K}} = (\mathbf{1}_1, \ldots, \mathbf{1}_{\mathbb{M}}), \tag{5.9}$$

where $\mathbf{1}_m$ is a $\mathbb{K}_m$-dimensional vector of ones. From the heavy traffic Assumption 3.2 and (5.1)–(5.2),

$$\lambda_m = H^m x_m^* \quad \text{and} \quad \mathbf{1}_m = B^m x_m^* \quad \text{for} \quad m = 1, \ldots, \mathbb{M}, \tag{5.10}$$

and in particular $\lambda_{m^*} = H^{m^*} x_{m^*}^*$ and $\mathbf{1}_{m^*} = B^{m^*} x_{m^*}^*$. There are two cases to consider.

Case I: Suppose that $\mathbb{I}_{m^*} + \mathbb{K}_{m^*} < \mathbb{B}_{m^*}$.

Let

$$P^{m^*} = \begin{pmatrix} H^{m^*} \\ B^{m^*} \end{pmatrix}. \tag{5.11}$$

Then P^{m^*} is an $(\mathbb{I}_{m^*} + \mathbb{K}_{m^*}) \times \mathbb{B}_{m^*}$ matrix and the null space of P^{m^*} is nontrivial. Thus, there exists a non-zero $\mathbb{B}_{m^*}$-dimensional vector v such that $P^{m^*} v = 0$. Since all of the components of $x_{m^*}^*$ are strictly greater than zero, there exists a $\delta > 0$ such that all of the components of $x_{m^*}^* + \delta v$ are strictly greater than zero. Define a $\mathbb{J}$-dimensional vector $\hat{v}$ as follows,

$$\hat{v} = (0_1, \ldots, 0_{m^*-1}, v, 0_{m^*+1}, \ldots, 0_{\mathbb{M}}, 0_{\mathbb{N}})', \tag{5.12}$$

where 0_m is the $\mathbb{B}_m$-dimensional zero vector for $m \neq m^*$ and $0_{\mathbb{N}}$ is the $\mathbb{N}$-dimensional zero vector. Then

$$\begin{pmatrix} R \\ A \end{pmatrix} \delta \hat{v} = 0. \tag{5.13}$$

The first $\mathbb{B}$ components of $x^* + \delta \hat{v}$ are strictly greater than zero, the last $\mathbb{N}$ components are identically zero, and

$$R(x^* + \delta \hat{v}) = \lambda, \qquad A(x^* + \delta \hat{v}) = \mathbf{1}_{\mathbb{K}}. \tag{5.14}$$

This violates the uniqueness part of the heavy traffic Assumption 3.2.

Case II: Suppose that $\mathbb{I}_{m^*} + \mathbb{K}_{m^*} = \mathbb{B}_{m^*}$.

Let P^{m^*} be as in *Case I*. There are two subcases to consider.

(i) The null space of P^{m^*} is not $\{0\}$. Then we can repeat the argument from *Case I*.

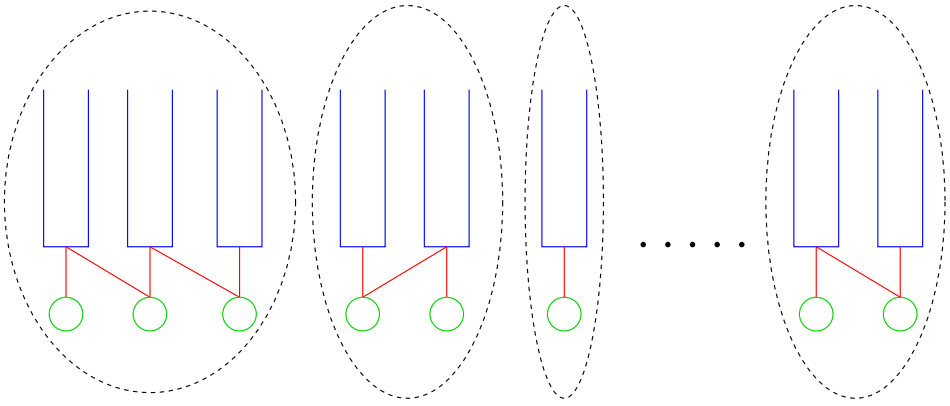

FIG 2. *Example of a server-buffer graph $\mathcal{G}$ where only basic activities are shown. Trees $\mathcal{T}_1, \ldots, \mathcal{T}_{\mathbb{L}}$ encircled by dashed lines are enumerated from left to right.*

(ii) The null space of P^{m^*} is $\{0\}$. Then P^{m^*} is invertible and there exists a $\mathbb{B}_{m^*}$-dimensional vector u such that

$$P^{m^*} u = \begin{pmatrix} 0_{m^*} \\ -\mathbf{1}_{m^*} \end{pmatrix}, \tag{5.15}$$

where 0_{m^*} is the $\mathbb{I}_{m^*}$-dimensional zero vector and $\mathbf{1}_{m^*}$ is the $\mathbb{K}_{m^*}$-dimensional vector of ones. As in *Case I*, there is a $\delta \in (0,1)$ such that all of the components of $x^*_{m^*} + \delta u$ are strictly positive and in the same manner as in (5.12), we can extend u to a $\mathbb{J}$-dimensional vector $\hat{u}$ satisfying

$$R(x^* + \delta \hat{u}) = \lambda \quad \text{and} \quad A(x^* + \delta \hat{u}) = \mathbf{1} - \delta \hat{\mathbf{1}}_{m^*}, \tag{5.16}$$

where $(\hat{\mathbf{1}}_{m^*})_k = 1$ if $k \in \mathcal{K}_{m^*}$ and $(\hat{\mathbf{1}}_{m^*})_k = 0$ otherwise. This violates the heavy traffic Assumption 3.2. Thus, by contradiction we have that,

$$\mathbb{B}_m = \mathbb{I}_m + \mathbb{K}_m - 1, \text{ for all } m, \tag{5.17}$$

as desired. Furthermore, as observed by Williams [34], since the connected graph $\mathcal{T}_m$ with $\mathbb{I}_m + \mathbb{K}_m$ nodes has exactly $\mathbb{I}_m + \mathbb{K}_m - 1$ edges, it must be a tree (cf. Theorem 3.1 in [8]). □

From this point on, the trees in $\mathcal{G}$ will be denoted by $\mathcal{T}_1, \ldots, \mathcal{T}_{\mathbb{L}}$, and we let $\mathcal{L} = \{1, \ldots, \mathbb{L}\}$.

6. Decoupled workload matrix and the cost function in the EWF. In this section we prove that one can choose the workload matrix M for parallel server systems to have a simple block diagonal-like structure related to the forest of trees structure of the server-buffer graph. We call this a *decoupled workload matrix* because each queue-length component affects just one workload component and the workload matrix partitions or decouples the queue-length components coming from different trees. We exploit the structure of this decoupled workload matrix to prove that when the BCP cost function f is linear, the cost function g for the EWF is also linear. Indeed, we give an explicit form for g and for the associated continuous selection function ϕ. We end the section by giving an example to illustrate the point that when the workload dimension is more than one, our decoupled workload matrix and the associated EWF typically has a simpler structure than for the choice of workload matrix proposed by Harrison [17].

6.1. *Decoupled workload matrix and associated control matrix.* We have seen in Section 4 that Harrison's choice for the workload matrix is obtained by finding the extremal optimal solutions of the dual program for the entire system. For our choice, for each $l \in \mathcal{L}$, let $\tilde{\mathcal{T}}_l$ be the graph obtained by adding the non-basic activities in $\mathcal{N}_l^l$ to the tree $\mathcal{T}_l$. Then $\tilde{\mathcal{T}}_l$ consists of servers and buffers in $\mathcal{T}_l$ and all activities that both consume material from buffers in $\mathcal{T}_l$ and are processed by servers in $\mathcal{T}_l$. We treat each graph $\tilde{\mathcal{T}}_l$, $l \in \mathcal{L}$, in isolation as a parallel server system with a one-dimensional workload. By collecting these workload vectors, we construct a workload matrix for the whole system. In contrast to the situation for Harrison's choice of workload matrix, the control matrix G that goes with our decoupled workload matrix is not necessarily non-negative. We now describe our choice in detail. In the following, depending on the context, a vector x sometimes denotes a row vector and at other times it denotes a column vector; whichever is meant will be clear from the context.

For each $l \in \mathcal{L}$, let λ_l be specified as in (5.8), let x_l^* denote the $(\mathbb{B}_l + \mathbb{N}_l^l)$-dimensional vector consisting of the components of x^* indexed by the activities in $\tilde{\mathcal{T}}_l$, and let

$$R^l = [H^l, J_l^l], \quad A^l = [B^l, N_l^l]. \tag{6.1}$$

Then (λ_l, R^l, A^l) can be viewed as specifying "first order parameters" for a parallel server system with network structure given by $\tilde{\mathcal{T}}_l$. As noted in [12], the notions of heavy traffic and workload dimension and matrix only depend on such first order parameters. In particular, we have the following. For this, recall that we are assuming that Assumption 3.3 holds, which includes Assumptions 3.1 and 3.2.

LEMMA 6.1. *For $l \in \mathcal{L}$, $\tilde{\mathcal{T}}_l$ is in heavy traffic, i.e., $(x_l^*, 1)$ is the unique solution of the linear program associated with (λ_l, R^l, A^l):*

$$\text{(6.2)} \qquad \textit{minimize}\, \rho \textit{ subject to } R^l x_l = \lambda_l,\ A^l x_l \le \rho \mathbf{1}_l \textit{ and } x_l \ge 0.$$

PROOF. Fix $l \in \mathcal{L}$. From the block diagonal structure of (5.1)–(5.2), the definition of x^*, and the fact that non-basic components of x^* are zero, it follows that $(x_l^*, 1)$ is a feasible solution of the linear program (6.2) associated with (λ_l, R^l, A^l) and it satisfies $A^l x_l^* = \mathbf{1}_l$. Suppose that $(x_l^\dagger, \rho^\dagger)$ is an optimal solution of the linear program associated with (λ_l, R^l, A^l) and $x_l^\dagger \ne x_l^*$. Then, $R^l x_l^\dagger = \lambda_l$, $A^l x_l^\dagger \le \rho^\dagger \mathbf{1}_l \le \mathbf{1}_l$. Let $x^\dagger$ be the vector obtained from x^* by replacing the components indexed by activities in $\tilde{\mathcal{T}}_l$ by the components of $x_l^\dagger$ associated with those activities. Then, by (5.1)–(5.2), $Rx^\dagger = \lambda, Ax^\dagger \le \mathbf{1}$ and $x^\dagger \ne x^*$. Thus, $(x^\dagger, 1)$ is a feasible solution of the linear program (3.2). By Assumption 3.2, the optimal value of that program is one, and by the assumed uniqueness of its optimal solution, we must have $x^\dagger = x^*$ and hence $x_l^\dagger = x_l^*$, $\rho^\dagger = 1$. □

By Theorem 5.1, when viewed in isolation, $\tilde{\mathcal{T}}_l$ corresponds to a parallel server system with a one-dimensional workload, where a workload matrix (vector) for $\tilde{\mathcal{T}}_l$ can be obtained from the unique solution of the following dual program.

DEFINITION 6.1.

$$\text{(6.3)} \quad \textit{maximize } \tilde{y}^l \cdot \lambda_l \textit{ subject to } \tilde{y}^l R^l \le \tilde{z}^l A^l, \tilde{z}^l \cdot \mathbf{1}_l = 1 \textit{ and } \tilde{z}^l \ge 0.$$

Let (y^l, z^l) be the (optimal) solution of the above dual program for $\tilde{\mathcal{T}}_l$. Then y^l is the choice of workload matrix (vector) for $\tilde{\mathcal{T}}_l$ proposed in [17]. It follows that $y^l \in \mathcal{R}_l^\perp$ where

$$\text{(6.4)} \qquad \mathcal{R}_l = \{\delta_l \in \mathbf{R}^{\mathbb{I}_l} : \delta_l = R^l x_l, K^l x_l = 0, x_l \in \mathbf{R}^{\mathbb{B}_l + \mathbb{N}_l^l}\},$$

is the space of reversible displacements for $\tilde{\mathcal{T}}_l$,

$$\text{(6.5)} \qquad K^l = \begin{pmatrix} B^l & N_l^l \\ 0 & -I^l \end{pmatrix},$$

and I^l is the $\mathbb{N}_l^l \times \mathbb{N}_l^l$ identity matrix. Recall that by Proposition 2 in [18], $y^l > 0$ and $z^l > 0$. Let $\hat{y}^l$ be the $\mathbb{I}$-dimensional vector:

$$\text{(6.6)} \qquad \hat{y}^l = (0_1, .., 0_{l-1}, y^l, 0_{l+1}, .., 0_{\mathbb{L}}),$$

which is the augmentation of y^l, where $0_{l'}$ is the $\mathbb{I}_{l'}$-dimensional vector of zeros for $l' \neq l$. Let M be the $\mathbb{L} \times \mathbb{I}$ matrix with rows given by $\hat{y}^l$, $l \in \mathcal{L}$. Then M has a block diagonal-like structure:

$$M = \begin{pmatrix} \hat{y}^1 \\ \hat{y}^2 \\ \vdots \\ \hat{y}^{\mathbb{L}-1} \\ \hat{y}^{\mathbb{L}} \end{pmatrix} = \begin{pmatrix} y^1 & 0 & \dots & \dots & 0 \\ 0 & y^2 & 0 & \dots & \vdots \\ \vdots & 0 & \ddots & 0 & \vdots \\ \vdots & \dots & 0 & y^{\mathbb{L}-1} & 0 \\ 0 & \dots & \dots & 0 & y^{\mathbb{L}} \end{pmatrix}. \tag{6.7}$$

The following lemma shows that M is a valid choice for a workload matrix for the entire parallel server system.

LEMMA 6.2. *For each $l \in \mathcal{L}$, $\hat{y}^l \in \mathcal{R}^\perp$. Furthermore $\hat{y}^1, \dots, \hat{y}^{\mathbb{L}}$ are linearly independent and form a basis for $\mathcal{R}^\perp$.*

PROOF. Let $w \in \mathcal{R}$ be arbitrary. Then $w = (w_1, ..., w_{\mathbb{L}})$, where each w_l is $\mathbb{I}_l$-dimensional. We would like to show that $\hat{y}^l \cdot w = 0$. Since $w \in \mathcal{R}$ there exists an $x \in \mathbf{R}^{\mathbb{J}}$ such that $w = Rx$ and $Kx = 0$. By (5.1)–(5.2) and the fact that the non-basic entries in x are zero, it follows that $w_l = R^l x_l$ and $K^l x_l = 0$. Therefore $w_l \in \mathcal{R}_l$ and by the assumption on y^l, $y^l \cdot w_l = 0$. Hence $\hat{y}^l \cdot w = 0$ and since $w \in \mathcal{R}$ was arbitrary $\hat{y}^l \in \mathcal{R}^\perp$. By (6.6), $\hat{y}^1, ..., \hat{y}^{\mathbb{L}}$ are orthogonal and hence linearly independent. The result then follows since $\mathbb{L}$ linearly independent vectors in the $\mathbb{L}$-dimensional vector space $\mathcal{R}^\perp$ constitute a basis for $\mathcal{R}^\perp$. □

For the choice of M in (6.7), $\tilde{Q}$ as in the BCP and $\tilde{W} = M\tilde{Q}$, for each $l \in \mathcal{L}$, $\tilde{W}_l$ is a sum of diffusion "queue-lengths" associated with buffers in $\tilde{\mathcal{T}}_l$. In this sense, $\tilde{W}_l$ represents the diffusion workload of $\tilde{\mathcal{T}}_l$. This interpretation is not generally available for Harrison's proposal for workload as the l^{th} component of his diffusion workload often involves diffusion queue-lengths for buffers from more than one tree in $\mathcal{G}$ (see Section 6.4 for an example).

LEMMA 6.3. *With the above choice of M, $\mathcal{W} = M\mathbf{R}_+^{\mathbb{I}} = \mathbf{R}_+^{\mathbb{L}}$.*

PROOF. For $l \in \mathcal{L}$, by Proposition 2 in [18], $y^l > 0$. The conclusion is immediate by the form of M in equation (6.7). □

For the workload matrix M described above we find a control matrix G that will satisfy $MR = GK$. For this, for each $l \in \mathcal{L}$, we augment the

$\mathbb{K}_l$-dimensional vector z_l to a $\mathbb{K}$-dimensional vector:

$$\hat{z}^l = (0_1, .., 0_{l-1}, z^l, 0_{l+1}, .., 0_{\mathbb{L}}), \tag{6.8}$$

where 0_l is the $\mathbb{K}_l$-dimensional zero vector. Let

$$\Pi = \begin{pmatrix} \hat{z}^1 \\ \hat{z}^2 \\ \vdots \\ \hat{z}^{\mathbb{L}-1} \\ \hat{z}^{\mathbb{L}} \end{pmatrix} = \begin{pmatrix} z^1 & 0 & \dots & \dots & 0 \\ 0 & z^2 & 0 & \dots & \vdots \\ \vdots & 0 & \ddots & 0 & \vdots \\ \vdots & \dots & 0 & z^{\mathbb{L}-1} & 0 \\ 0 & \dots & \dots & 0 & z^{\mathbb{L}} \end{pmatrix} \tag{6.9}$$

LEMMA 6.4. *Let M, Π be given by (6.7) and (6.9). Define an $\mathbb{L} \times \mathbb{P}$ matrix G by*

$$G = [\Pi \;\; \Pi N - MJ]\,. \tag{6.10}$$

Then G is a valid choice for a control matrix, i.e., the relation $MR = GK$ is satisfied.

PROOF. For $l \in \mathcal{L}$, since (y^l, z^l) is a unique extremal optimal solution of the dual linear program for the subnetwork $\tilde{\mathcal{T}}_l$, we have that that $y^l H^l = z^l B^l$. By the form of M, Π, H and B, it follows that $\Pi B = MH$, and therefore

$$\begin{aligned} GK &= [\Pi \;\; \Pi N - MJ] \begin{pmatrix} B & N \\ 0 & -I \end{pmatrix} = [\Pi B \;\; \Pi N - \Pi N + MJ] \\ &= [MH \;\; MJ] = M\,[H \;\; J] = MR. \end{aligned} \tag{6.11}$$

□

We proceed to describe G more explicitly. First we compute MJ:

$$MJ = \begin{pmatrix} y^1 J^1 & 0 & \dots & 0 \\ 0 & y^2 J^2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & y^{\mathbb{L}} J^{\mathbb{L}} \end{pmatrix}$$

$$= \begin{pmatrix} y^1 J^1_1 & \dots & y^1 J^1_{\mathbb{L}} & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\ 0 & \dots & 0 & y^2 J^2_1 & \dots & y^2 J^2_{\mathbb{L}} & \dots & \vdots & \dots & \vdots \\ \vdots & \dots & \vdots & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\ 0 & \dots & 0 & \vdots & \dots & \vdots & \dots & y^{\mathbb{L}} J^{\mathbb{L}}_1 & \dots & y^{\mathbb{L}} J^{\mathbb{L}}_{\mathbb{L}} \end{pmatrix},$$

where $y^l J^l$ is an $\mathbb{N}^{l,c}$-dimensional vector and $y^l J^l_m$ is an $\mathbb{N}^l_m$-dimensional vector. Then we compute ΠN:

$$\Pi N = \begin{pmatrix} z^1 N_1^1 & 0 & \dots & 0 & \dots & z^1 N_1^{\mathbb{L}} & 0 & \dots & 0 \\ 0 & z^2 N_2^1 & \ddots & \vdots & \dots & 0 & z^2 N_2^{\mathbb{L}} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 & \dots & \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & z^{\mathbb{L}} N_{\mathbb{L}}^1 & \dots & 0 & \dots & 0 & z^{\mathbb{L}} N_{\mathbb{L}}^{\mathbb{L}} \end{pmatrix},$$

where $z^l N_l^m$ is an $\mathbb{N}_l^m$-dimensional vector. For the above choices of M and Π, it is not true in general that $\Pi N - MJ \geq 0$. More specifically, G has the following form:

$$G = [\Upsilon_0, \Upsilon_1, \Upsilon_2, \dots, \Upsilon_{\mathbb{L}}], \tag{6.12}$$

where

$$\Upsilon_0 = \Pi = \begin{pmatrix} z^1 & 0 & \dots & \dots & 0 \\ 0 & z^2 & 0 & \dots & \vdots \\ \vdots & 0 & \ddots & 0 & \vdots \\ \vdots & \dots & 0 & z^{\mathbb{L}-1} & 0 \\ 0 & \dots & \dots & 0 & z^{\mathbb{L}} \end{pmatrix}, \tag{6.13}$$

and for $l \in \mathcal{L}$, Υ_l is given by the following

$$\begin{pmatrix} z^1 N_1^l & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & z^2 N_2^l & 0 & \dots & \dots & \dots & \vdots \\ \vdots & 0 & \ddots & 0 & 0 & \dots & 0 \\ -y^l J_1^l & -y^l J_2^l & \dots & z^l N_l^l - y^l J_l^l & -y^l J_{l+1}^l & \dots & -y^l J_{\mathbb{L}}^l \\ 0 & 0 & 0 & 0 & z^{l+1} N_{l+1}^l & 0 & 0 \\ \vdots & \dots & \dots & \dots & 0 & \ddots & \vdots \\ 0 & \dots & \dots & \dots & \dots & 0 & z^{\mathbb{L}} N_{\mathbb{L}}^l \end{pmatrix}. \tag{6.14}$$

For $l, m \in \mathcal{L}$, each column of N_m^l and J_m^l has one positive entry and all other entries are equal to zero. Recall that for each $l \in \mathcal{L}$, $y^l > 0$, $z^l > 0$ and $y^l J_l^l < z^l N_l^l$ (see Remark 5.1). It follows that for each $l, m \in \mathcal{L}$, $z^m N_m^l > 0$, $-y^l J_m^l < 0$ and $z^l N_l^l - y^l J_l^l > 0$.

As we will see in Section 9, in practice it may be computationally convenient to choose a slightly different workload and control matrix according to the following lemma.

LEMMA 6.5. *For any $c \in \mathbf{R}_+^{\mathbb{L}}$ such that $c^l > 0$ for $l = 1, \ldots, \mathbb{L}$, let*

$$M^c = \begin{pmatrix} c^1 y^1 & 0 & \ldots & \ldots & 0 \\ 0 & c^2 y^2 & 0 & \ldots & \vdots \\ \vdots & 0 & \ddots & 0 & \vdots \\ \vdots & \ldots & 0 & c^{\mathbb{L}-1} y^{\mathbb{L}-1} & 0 \\ 0 & \ldots & \ldots & 0 & c^{\mathbb{L}} y^{\mathbb{L}} \end{pmatrix},$$

$$\Pi^c = \begin{pmatrix} c^1 z^1 & 0 & \ldots & \ldots & 0 \\ 0 & c^2 z^2 & 0 & \ldots & \vdots \\ \vdots & 0 & \ddots & 0 & \vdots \\ \vdots & \ldots & 0 & c^{\mathbb{L}-1} z^{\mathbb{L}-1} & 0 \\ 0 & \ldots & \ldots & 0 & c^{\mathbb{L}} z^{\mathbb{L}} \end{pmatrix},$$

and

$$G^c = [\Pi^c \;\; \Pi^c N - M^c J]. \tag{6.15}$$

Then, M^c and G^c are valid choices for a workload and an associated control matrix.

PROOF. Since $c^l > 0$ for each l, the rows of M^c constitute a basis for $\mathcal{R}^\perp$ by Lemma 6.2. By a straight forward computation as in Lemma 6.4 we see that $M^c R = G^c K$. □

Note that, if c is the $\mathbb{L}$-dimensional vector of ones, then $M^c = M$ and $G^c = G$. In the next two subections the matrices M and G are chosen as described in this subsection.

6.2. *Columns of G and components of the control $\tilde{U}$.* The control process $\tilde{U}$ in the EWF is $\mathbb{P}$-dimensional. The first $\mathbb{K}$ components of $\tilde{U}$ are Brownian model analogues of server idle-times. The last $\mathbb{N}$ components of $\tilde{U}$ are Brownian model analogues of allocations to non-basic activities. An increase in $\tilde{U}_j$ moves the system in the direction G^j, where G^j is the j^{th} column of G. For this, there are three cases to consider:

(i) increasing $\tilde{U}_k$ for $k \in \mathcal{K}$, increases the workload $\tilde{W}_l$ for $\tilde{\mathcal{T}}_l$ when server k belongs to $\tilde{\mathcal{T}}_l$ and has no effect on other components of the workload;
(ii) increasing $\tilde{U}_{j'}$ for $j' = j - \mathbb{B} + \mathbb{K}$ where $j \in \mathcal{N}_l^l$, $l \in \mathcal{L}$, corresponds to using a non-basic activity in $\tilde{\mathcal{T}}_l$ and this increases the workload $\tilde{W}_l$ for $\tilde{\mathcal{T}}_l$ and has no effect on other components of the workload;

(iii) increasing $\tilde{U}_{j'}$ for $j' = j - \mathbb{B} + \mathbb{K}$ where $j \in \mathcal{N}_m^l$, $m \neq l$, $m, l \in \mathcal{L}$, corresponds to use of a non-basic activity that connects two different components $\tilde{\mathcal{T}}_l$ and $\tilde{\mathcal{T}}_m$, and this will decrease the workload $\tilde{W}_l$ for $\tilde{\mathcal{T}}_l$ and increase the workload $\tilde{W}_m$ for $\tilde{\mathcal{T}}_m$. This has to do with the fact that j consumes material from a buffer in $\tilde{\mathcal{T}}_l$ and is processed by a server in $\tilde{\mathcal{T}}_m$.

6.3. *Cost function in the EWF when the BCP has linear holding cost.* In this subsection, we assume that the cost function f in the BCP is linear. More precisely, we assume the following.

ASSUMPTION 6.1. *Assume that the cost function $f : \mathbb{R}_+^{\mathbb{I}} \to \mathbb{R}_+$ is given by*

$$f(q) = h \cdot q \quad \text{for all } q \in \mathbb{R}_+^{\mathbb{I}}, \tag{6.16}$$

where $h = (h_1, \ldots, h_{\mathbb{I}})'$ satisfies $h_i > 0$ for $i = 1, \ldots, \mathbb{I}$.

Recall that for $w \in \mathbf{R}_+^{\mathbb{L}}$,

$$g(w) = \min\{h \cdot q : Mq = w, q \in \mathbf{R}_+^{\mathbb{I}}\}. \tag{6.17}$$

THEOREM 6.1. *Under Assumption 6.1, the cost function g is linear. In particular, for $w \in \mathbf{R}_+^{\mathbb{L}}$,*

$$g(w) = \kappa \cdot w, \tag{6.18}$$

where

$$\kappa_l \equiv \min_{i \in \mathcal{I}_l} \left(\frac{h_i}{\hat{y}_i^l} \right), \quad l \in \mathcal{L}. \tag{6.19}$$

Moreover, a continuous selection function ϕ associated with g is given by $\phi(w) = q^*(w)$ for $w \in \mathbf{R}_+^{\mathbb{L}}$, where, for each $l \in \mathcal{L}$,

$$q_{i_l^*}^*(w) = \frac{w_l}{\hat{y}_{i_l^*}^l} \quad \text{and} \quad q_i^*(w) = 0 \quad \text{for} \quad i \in \mathcal{I}_l \setminus \{i_l^*\}, \tag{6.20}$$

and $i_l^* \in \mathcal{I}_l$ is chosen such that $\kappa_l = h_{i_l^*}/\hat{y}_{i_l^*}^l$.

PROOF. Fix $w \in \mathbf{R}_+^{\mathbb{L}}$. Recall the special block diagonal-like structure of the workload matrix M shown in (6.7). If $Mq = w$ for $q \in \mathbf{R}_+^{\mathbb{I}}$, then for each $l \in \{1, \ldots, \mathbb{L}\}$,

$$w_l = \hat{y}^l \cdot q, \tag{6.21}$$

and

$$h \cdot q = \sum_{l=1}^{\mathbb{L}} \sum_{i \in \mathcal{I}_l} h_i q_i = \sum_{l=1}^{\mathbb{L}} \sum_{i \in \mathcal{I}_l} \left(\frac{h_i}{\hat{y}_i^l} \right) \hat{y}_i^l q_i \tag{6.22}$$

$$\geq \sum_{l=1}^{\mathbb{L}} \kappa_l \left(\sum_{i \in \mathcal{I}_l} \hat{y}_i^l q_i \right) = \sum_{l=1}^{\mathbb{L}} \kappa_l w_l. \tag{6.23}$$

If $q^*(w)$ is as in (6.20), then $Mq^*(w) = w$ and $h \cdot q^*(w) = \kappa \cdot w$. This completes the proof. □

REMARK 6.1. *If M^c and G^c are used in place of M and G, for each l, i_l^* can be kept the same, but the values of κ and q^* need to be adjusted for the fact that y^l is replaced by $c^l y^l$ in (6.19) and (6.21).*

In view of the form of the cost function g and the observations made in Section 6.2, we see that increasing $\tilde{U}_k$ for $k \in \mathcal{K}$ or increasing $\tilde{U}_{j'}$ for $j' = j - \mathbb{B} + \mathbb{K}$ where $j \in \mathcal{N}_l^l$, $l \in \mathcal{L}$, will cause an increase in the cost function $g(\tilde{W})$. As a consequence, we will find that for an optimal solution of the EWF, we do not need to use the non-basic activities $j \in \mathcal{N}_l^l$ within any $\tilde{\mathcal{T}}_l$, $l \in \mathcal{L}$, and we only need to use the Brownian model analogues $\tilde{U}_k, k \in \mathcal{K}$, of server idle-time to keep the workloads $\tilde{W}_l$, $l \in \mathcal{L}$, non-negative, although, under the cost structure of the next section, we will only do this as a last resort. Indeed, we will sometimes be able to use a non-basic activity $j \in \mathcal{N}_m^l$, $m \neq l$, $m, l \in \mathcal{L}$, connecting two different components $\tilde{\mathcal{T}}_l$ and $\tilde{\mathcal{T}}_m$, to keep the workload $\tilde{W}_m$ for $\tilde{\mathcal{T}}_m$ non-negative while simultaneously reducing the workload $\tilde{W}_l$ in $\tilde{\mathcal{T}}_l$; in this case, we do not need to use $\tilde{U}_k$ for servers k in $\tilde{\mathcal{T}}_m$ to keep the workload $\tilde{W}_m$ non-negative. Because of the above, it will turn out that to describe an optimal control for the EWF, we only need to focus on the server-buffer graph, with its basic activities within trees, augmented by the non-basic activities that connect different trees. Accordingly, henceforth we will speak of the workload associated with a tree $\mathcal{T}_l$ rather than $\tilde{\mathcal{T}}_l$.

6.4. *Our decoupled workload matrix versus Harrison's choice.* In this subsection, for a simple example, we explicitly compute our decoupled workload matrix and associated control matrix, and compare these to Harrison's choices proposed in [17]. We consider a parallel server system consisting of two buffers, two servers and three activities (see Figure 3). First order parameters are as follows:

$$\lambda_1 = \mu_1 > 0, \quad \lambda_2 = \mu_2 > 0, \quad R = \begin{pmatrix} \mu_1 & 0 & 0 \\ 0 & \mu_2 & \mu_3 \end{pmatrix}, \quad A = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

Assumption 3.2 is satisfied with $x^* = (1, 1, 0)$. Activities 1 and 2 are basic and activity 3 is non-basic. The server-buffer graph $\mathcal{G}$ consists of two trees, $\mathcal{T}_1, \mathcal{T}_2$ and by Theorem 5.2 the workload is 2-dimensional. Note that $\tilde{\mathcal{T}}_l = \mathcal{T}_l$ are trees for $l = 1, 2$. Harrison's workload matrix is obtained by finding extremal solutions of the dual program:

$$\begin{aligned} &\textit{maximize} && y_1\lambda_1 + y_2\lambda_2 \\ &\textit{subject to} && y_1\lambda_1 \le z_1, \quad y_2\lambda_2 \le z_2, \quad y_2\mu_3 \le z_1, \\ &&& z_1 + z_2 = 1, \quad z_1,\, z_2 \ge 0. \end{aligned}$$

It is well known that the value of the above dual program equals the value of the primal program in Assumption 3.2, which equals one. Using the fact that $\lambda_1 = \mu_1$, $\lambda_2 = \mu_2$, the extremal solutions of this dual program are seen to be (y^1, z^1) and (y^2, z^2) where

$$\begin{aligned} y^1 &= (1/\lambda_1, 0)\,, \ z^1 = (1, 0), \\ y^2 &= (\mu_3/(\lambda_1(\mu_3 + \lambda_2)), 1/(\mu_3 + \lambda_2))\,, \\ z^2 &= (\mu_3/(\mu_3 + \lambda_2), \lambda_2/(\mu_3 + \lambda_2))\,, \end{aligned} \tag{6.24}$$

The following is the workload matrix in Harrison's sense given by equation (4.10):

$$M = \begin{pmatrix} 1/\lambda_1 & 0 \\ \mu_3/(\lambda_1(\mu_3 + \lambda_2)) & 1/(\mu_3 + \lambda_2) \end{pmatrix}, \tag{6.25}$$

with associated matrices,

$$\Pi = \begin{pmatrix} 1 & 0 \\ \mu_3/(\mu_3 + \lambda_2) & \lambda_2/(\mu_3 + \lambda_2) \end{pmatrix}, \quad N = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad J = \begin{pmatrix} 0 \\ \mu_3 \end{pmatrix},$$

so that

$$G = [\Pi \ \Pi N - MJ] = \begin{pmatrix} 1 & 0 & 1 \\ \mu_3/(\mu_3 + \lambda_2) & \lambda_2/(\mu_3 + \lambda_2) & 0 \end{pmatrix}.$$

We now proceed to compute our decoupled workload matrix and the associated control matrix. Each tree is considered as an isolated parallel server system. We solve systems of inequalities associated with the dual program (6.3) for each tree:

$$\begin{aligned} \mathcal{T}_1 : &\quad y_1\lambda_1 = 1, \quad y_1\mu_1 \le z_1, \quad z_1 = 1; \\ \mathcal{T}_2 : &\quad y_2\lambda_2 = 1, \quad y_2\mu_2 \le z_2, \quad z_2 = 1. \end{aligned}$$

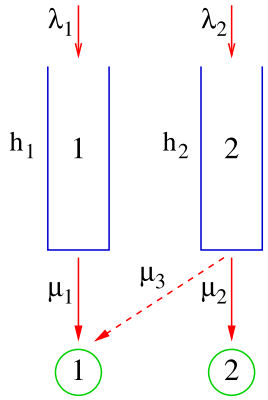

FIG 3. *Two "trees" connected via a single non-basic activity.*

This yields

$$\hat{y}^1 = (1/\lambda_1, 0)\,, \quad \hat{z}^1 = (1,0)$$
$$\hat{y}^2 = (0, 1/\lambda_2)\,, \quad \hat{z}^2 = (0,1).$$

Using (6.7), (6.9) and Lemma 6.4 we choose

$$M = \begin{pmatrix} 1/\lambda_1 & 0 \\ 0 & 1/\lambda_2 \end{pmatrix}, \qquad G = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & -\mu_3/\lambda_2 \end{pmatrix}. \tag{6.26}$$

The third column of the control matrix G has a positive first entry and a negative second entry. As discussed in Section 6.2, this is because activity 3 is non-basic, it is processed in $\mathcal{T}_1$ and it consumes material from buffer 2 in $\mathcal{T}_2$. Intuitively, we expect that use of activity 3 will reduce the workload of $\mathcal{T}_2$ and increase the workload of $\mathcal{T}_1$.

In contrast, Harrison's M matrix is not diagonal and all of the components of the control directions are non-negative. The second component of Harrison's workload does not represent the workload of tree $\mathcal{T}_2$ (i.e., the workload associated with buffer 2), but it is a sum of scaled workloads for the two trees. Accordingly, the state space $\mathcal{W}$ for Harrison's workload process will be a wedge contained within the non-negative quadrant of two-dimensional space, where the rays defining the wedge are given by the columns of his choice for the matrix M. Thus, even in this simple example, the state space for our choice of workload matrix is somewhat more straightforward than for Harrison's choice.

7. Solution of the BCP for certain graph and cost structure. Henceforth, for the remainder of this paper, we assume that Assumption 6.1 holds, i.e., the cost function in the BCP is linear.

In this section, we prove that under a certain minimal connectedness assumption on the extended server-buffer graph $\mathcal{H}$ defined below, and a certain monotonicity assumption on the cost function g, a least control is optimal for the REWF, which enables us to solve the EWF and the BCP. Henceforth, we let M and G denote a choice of M^c and G^c for some $c > 0$ as in Lemma 6.5; for convenience we suppress the superscript c. When we mention the EWF, we mean the EWF corresponding to these matrices M and G.

7.1. *Graph structure.*

DEFINITION 7.1. *(Extended server-buffer graph with external non-basic activities) Let $\mathcal{H}$ be the graph in which servers and buffers form the nodes and undirected edges between nodes are given by basic activities plus the non-basic activities that connect distinct trees in $\mathcal{G}$.*

The set of non-basic activities connecting distinct trees in $\mathcal{G}$ is given by $\bigcup_{l \neq m} \mathcal{N}_m^l$ and we denote it by $\mathcal{N}^{ext}$, where *ext* is mnemonic for external. We call the activities in $\mathcal{N}^{ext}$ external non-basic activities and we let $\mathbb{N}^{ext}$ denote the cardinality of $\mathcal{N}^{ext}$. An edge in $\mathcal{H}$ is either a basic activity or a non-basic activity in $\mathcal{N}^{ext}$ that consumes material from a buffer in some tree $\mathcal{T}_l$ and that is processed by a server in some other tree $\mathcal{T}_m$, $m \neq l$. Note that, $\mathcal{N}^{ext}$ includes all non-basic activities except those that are wholly confined to individual trees $\mathcal{T}_l$ for $l \in \mathcal{L}$. We now consider parallel server systems where the trees in $\mathcal{H}$ are connected via external non-basic activities in a fairly minimal way. We assume the following.

ASSUMPTION 7.1.

(i) For each $l \in \mathcal{L}$, $\mathbb{N}^{l,p} \leq \mathbb{N}_l^l + 1$.
(ii) $\mathcal{H}$ is connected.

REMARK 7.1. *Part (i) of Assumption 7.1 means that for each $l \in \mathcal{L}$ there is at most one external non-basic activity that is processed by a server in $\mathcal{T}_l$. Part (i) implies that $\mathbb{N}^{ext} \leq \mathbb{L}$. On the other hand, part (ii) implies that $\mathbb{N}^{ext} \geq \mathbb{L} - 1$. Together parts (i) and (ii) imply that $\mathbb{N}^{ext}$ equals $\mathbb{L} - 1$ or $\mathbb{L}$. If $\mathbb{N}^{ext} = \mathbb{L} - 1$, the graph $\mathcal{H}$ is a tree, otherwise $\mathbb{N}^{ext} = \mathbb{L}$ and $\mathcal{H}$ contains a loop (see [8]). If (i) holds, but not (ii), then $\mathcal{H}$ consists of two or more connected components each of which satisfies parts (i) and (ii) of Assumption 7.1 and each component can be treated separately by the methods described below.*

For a non-basic activity $j \in \mathcal{N}^{ext}$ let $t^c(j), t^p(j)$ be such that j consumes material from a buffer in $\mathcal{T}_{t^c(j)}$ and is processed by a server in $\mathcal{T}_{t^p(j)}$. For each $l \in \mathcal{L}$, let $a^c(l)$ be the subset of $\mathcal{N}^{ext}$ consisting of non-basic activities that consume material from buffers in $\mathcal{T}_l$, and let $a^p(l)$ be the subset of $\mathcal{N}^{ext}$ consisting of non-basic activities that are processed by servers in $\mathcal{T}_l$. By part (i) of Assumption 7.1, for each $l \in \mathcal{L}$, the set $a^p(l)$ consists of at most one element, but it may be empty. If $\mathbb{N}^{ext} = \mathbb{L}$, then $a^p(l)$ is not empty for all $l \in \mathcal{L}$. Otherwise, $\mathbb{N}^{ext} = \mathbb{L} - 1$ and there is exactly one index $l = l^p$ for which $a^p(l)$ is empty. For notational convenience, we henceforth adopt the following convention about the enumeration of trees.

CONVENTION 7.1. *If $\mathbb{N}^{ext} = \mathbb{L} - 1$, then $l^p = \mathbb{L}$, i.e., there is no external non-basic activity that is processed in $\mathcal{T}_{\mathbb{L}}$.*

Note that Convention 7.1 does not violate our enumeration of servers, buffers, basic and non-basic activities specified in Convention 5.1; it can be thought of as a further refinement of that enumeration. When the set $a^p(l)$ is not empty, we refer to $a^p(l)$ as the external non-basic activity processed in $\mathcal{T}_l$. On the other hand, for a fixed $l \in \mathcal{L}$, the set $a^c(l)$ may include several non-basic activities or it may be empty. Also, $a^c(l)$ may be empty for several $l \in \mathcal{L}$.

7.2. *Control matrix $\tilde{G}$ and cost assumption.* We will solve the EWF by solving the REWF associated to a reduced control matrix $\tilde{G}$ chosen as follows. For this, for each $l \in \mathcal{L}$, let i_l^* be as in Theorem 6.1 and let k_l^* be a server that can serve buffer i_l^* via a basic activity. Also, recall the convention about the enumeration of trees that we adopted in Section 5.2 and how components of $\tilde{U}$ are associated with Brownian analogues of server idle-times and allocations to non-basic activities as described in Section 6.2.

DEFINITION 7.2. *Let $\tilde{G}$ be the $\mathbb{L} \times \mathbb{L}$ matrix defined as follows. There are two cases.*

Case I: $\mathbb{N}^{ext} = \mathbb{L}$.

For $l = 1, \ldots, \mathbb{L}$, the l^{th} column of $\tilde{G}$, $\tilde{G}^l$ is given by the column of G that corresponds to the component of $\tilde{U}$ associated with the non-basic activity $a^p(l) \in \mathcal{N}^{ext}$, i.e., the external non-basic activity that is processed in $\mathcal{T}_l$.

Case II: $\mathbb{N}^{ext} = \mathbb{L} - 1$.

For $l = 1, \ldots, \mathbb{L} - 1$, the l^{th} column of $\tilde{G}$, $\tilde{G}^l$ is given by the column of G that corresponds to the component of $\tilde{U}$ associated with the non-basic

activity $a^p(l) \in \mathcal{N}^{ext}$, i.e., the non-basic activity that is processed in $\mathcal{T}_l$, and the $\mathbb{L}^{th}$ column of $\tilde{G}$ is given by the column of G that corresponds to the component of $\tilde{U}$ associated with the idle-time of server $k^*_{\mathbb{L}}$.

REMARK 7.2. *The matrix $\tilde{G}$ is obtained by deleting some columns of G and reordering the remaining columns. If $\mathbb{N}^{ext} = \mathbb{L}$, for each l, since $\tilde{G}^l$ corresponds to $a^p(l)$, $\tilde{G}^l$ has a positive l^{th} entry, and since $a^p(l)$ consumes material from a buffer in $\mathcal{T}_{t^c(a^p(l))}$, $\tilde{G}^l$ has a negative entry in the position with index $t^c(a^p(l))$; all other entries of $\tilde{G}^l$ are zero. If $\mathbb{N}^{ext} = \mathbb{L}-1$, the first $\mathbb{L}-1$ columns of $\tilde{G}$ have the form just described, while the $\mathbb{L}^{th}$ column has a positive $\mathbb{L}^{th}$ entry and all other entries are equal to zero. These observations follow from (6.13), (6.14) for $\Upsilon_{t^c(a^p(l))}$, and the fact that $z_{\mathbb{L}}$ is associated with the servers in the tree $\mathcal{T}_{\mathbb{L}}$. Thus,*

$$\tilde{G} = \tilde{G}^+ + \tilde{G}^-, \tag{7.1}$$

where $\tilde{G}^+$ is a diagonal matrix with positive diagonal entries and $\tilde{G}^-$ is a matrix whose non-zero entries are off-diagonal and non-positive. Moreover, each column of $\tilde{G}$ and $\tilde{G}^-$ has at most one negative entry.

Henceforth we make the following assumption about the cost function g. Recall the definition of the vector $\kappa > 0$ from Theorem 6.1 where it is shown that $g(w) = \kappa \cdot w$.

ASSUMPTION 7.2. *For each $l \in \mathcal{L}$, $\kappa \cdot \tilde{G}^l > 0$.*

This assumption corresponds to the situation where the external non-basic activities are expensive activities that should only be used when other alternatives are not available to reduce cost. Indeed, when workloads are positive, an external non-basic activity can be used to reduce workload in the tree that the activity consumes from, but this will result in an accompanying increase in workload in the tree that the activity is processed in, and under the above assumption, the magnitude of the reduction in cost in the EWF coming from one tree will be less in magnitude than the accompanying increase in cost coming from the other tree. On the other hand, if the workload in the tree where the non-basic activity is processed is zero, either server idling or use of the non-basic activity will be needed to counter negative Brownian excursions so as to keep the workload non-negative in the EWF; in this case, use of the non-basic activity results in a net reduction in cost due to consumption from another tree, whereas idling a server will not result in any cost reduction. Thus, under the above assumption, for an

optimal control for the EWF, an external non-basic activity will only be used when the workload in the tree where the server is located is zero, but then it will be used rather than idling a server, provided there is work to be done in the tree that it consumes from. In Section 9, we give an explicit example where this assumption is satisfied.

The proof of the following theorem is postponed until Section 7.4.

THEOREM 7.1. *Let $\tilde{G}$ be as in Definition 7.2. Then $\mathcal{C}(G) = \mathcal{C}(\tilde{G})$.*

With the choice of matrix $\hat{G} = \tilde{G}$ for the REWF, in view of Theorem 4.2, we can find an optimal control for the EWF from an optimal control for the REWF by setting components of the control for the EWF that correspond to columns of G that are not columns of $\tilde{G}$ to be identically zero. This is more formally described in the following lemma.

THEOREM 7.2. *Suppose $\hat{U}^*$ is an optimal control for the REWF associated with $\tilde{G}$, i.e., $\hat{J}^{*,\tilde{G}} = \hat{J}^{\tilde{G}}(\hat{U}^*)$. Then an optimal control $\tilde{U}^*$ for the EWF is given by setting*

$$\tilde{U}^*_j = \hat{U}^*_l \text{ if } G^j = \tilde{G}^l \text{ for some } l, \text{ and } \tilde{U}^*_j = 0 \text{ if } G^j \text{ is not a column of } \tilde{G}.$$

PROOF. By Theorem 4.2, and the optimality of $\hat{U}^*$ for the REWF, $\check{J}^* = \hat{J}^{*,\tilde{G}}(\hat{U}^*)$. It is easy to check that $\tilde{U}$ is an admissible control for the EWF and $G\tilde{U}^* = \tilde{G}\hat{U}^*$. It follows that, $\check{J}(\tilde{U}^*) = \hat{J}^{\tilde{G}}(\hat{U}^*) = \check{J}^*$. □

7.3. *Solution of the REWF and BCP.* In this subsection, we show that the REWF has a least control, that the least control is optimal for the REWF and we give equations characterizing the control. We express a solution of the BCP in terms of the optimal solution of the REWF. To simplify notation throughout this subsection we suppress $\tilde{G}$ in $\hat{\mathcal{A}}^{\tilde{G}}$ and $\hat{J}^{\tilde{G}}$ and we denote them simply by $\hat{\mathcal{A}}$ and $\hat{J}$ respectively.

DEFINITION 7.3. *Let $\mathcal{V}$ index the possible filtered probability spaces with associated Brownian motions on which admissible controls for the REWF associated to $\tilde{G}$ can be defined. For each $\upsilon \in \mathcal{V}$, let $(\Lambda^\upsilon, \mathcal{E}^\upsilon, \{\mathcal{E}^\upsilon_t\}, \mathbf{Q}^\upsilon)$ denote the filtered probability space, $\hat{\xi}^\upsilon$ denote the associated $\{\mathcal{E}^\upsilon_t\}$-Brownian motion, $\hat{\mathcal{A}}^\upsilon$ denote the set of admissible controls, and*

$$\hat{J}^{\upsilon,*} = \inf_{\hat{U} \in \hat{\mathcal{A}}^\upsilon} \hat{J}(\hat{U}). \tag{7.2}$$

Given $\hat{\mathcal{A}}^\upsilon$, a control process $\hat{U}^* \in \hat{\mathcal{A}}^\upsilon$ is called a least control process in $\hat{\mathcal{A}}^\upsilon$ if for each $l \in \{1, \ldots, \mathbb{L}\}$, $\hat{U}^*_l \leq \hat{U}_l$ for all $\hat{U} \in \hat{\mathcal{A}}^\upsilon$.

REMARK 7.3. *In view of Definition 7.3 we have that* $\hat{\mathcal{A}} = \bigcup_{\upsilon \in \mathcal{V}} \hat{\mathcal{A}}^{\upsilon}$ *and* $\hat{J}^* = \inf_{\upsilon \in \mathcal{V}} \hat{J}^{\upsilon,*}$.

Next we will use results of Yang [35] to show that for each $\upsilon \in \mathcal{V}$, there is a (unique) least control process $\hat{U}^{\upsilon,*} \in \hat{\mathcal{A}}^{\upsilon}$. Then we will show that $\hat{J}^{\upsilon,*} = \hat{J}(\hat{U}^{\upsilon,*})$ and that this value does not depend on υ. This then implies that $\hat{J}^* = \hat{J}^{\upsilon,*}$. To use the results of Yang, we need the following definition and lemma.

DEFINITION 7.4. *(Stiemke matrix) An* $m \times n$ *matrix* D *is a Stiemke matrix if there exists an* $x \in \mathbf{R}^n_+$ *such that* $Dx > 0$.

LEMMA 7.1. $\tilde{G}$ *is a Stiemke matrix with exactly one positive element in each row.*

PROOF. Since G has the form (6.10) and Π is a Stiemke matrix, G is a Stiemke matrix. Thus, since the cone $\mathcal{C}(G) = \mathcal{C}(\tilde{G})$, $\tilde{G}$ is a Stiemke matrix. By construction, diagonal elements of $\tilde{G}$ are positive and off-diagonal elements of $\tilde{G}$ are non-positive, hence each row of $\tilde{G}$ has exactly one positive element. □

THEOREM 7.3. *For each* $\upsilon \in \mathcal{V}$, *there is a unique least control process,* $\hat{U}^{\upsilon,*}$, *in* $\hat{\mathcal{A}}^{\upsilon}$. *This satisfies*

$$\hat{J}(\hat{U}^{\upsilon,*}) = \hat{J}^{\upsilon,*} = \hat{J}^*. \tag{7.3}$$

PROOF. Fix $\upsilon \in \mathcal{V}$. It follows from the results in Theorems 1 and 4 of Yang [35] that there is a continuous function $\Phi : \mathbf{D}^m_+ \mapsto \mathbf{D}^m_+$ such that

$$\hat{U}^{\upsilon,*} = \Phi(\hat{\xi}^{\upsilon}) \tag{7.4}$$

is the unique least control process in $\hat{\mathcal{A}}^{\upsilon}$. Indeed, it follows from Lemma 11 in [35] that since $\hat{\xi}^{\upsilon}$ has continuous paths, so does $\hat{U}^{\upsilon,*}$. Suppose $\hat{U}$ is any element of $\hat{\mathcal{A}}^{\upsilon}$. Let $\hat{W}$ be the associated workload:

$$\hat{W}(t) = \hat{\xi}^{\upsilon}(t) + \tilde{G}\hat{U}(t) \quad \text{for } t \geq 0.$$

Let $\hat{W}^{\upsilon,*}$ be the workload process associated to the least control $\hat{U}^{\upsilon,*}$. By the minimality of $\hat{U}^{\upsilon,*}$,

$$\hat{U}_l(t) \geq \hat{U}^{\upsilon,*}_l(t), \quad \text{for all } t \geq 0 \text{ and } l \in \mathcal{L}. \tag{7.5}$$

Then, for $t \geq 0$,

$$\begin{aligned}\hat{W}(t) &= \hat{\xi}^{\upsilon}(t) + \tilde{G}\hat{U}^{\upsilon,*}(t) + \tilde{G}\left(\hat{U}(t) - \hat{U}^{\upsilon,*}(t)\right) \\ &= \hat{W}^{\upsilon,*}(t) + \tilde{G}(\hat{U}(t) - \hat{U}^{\upsilon,*}(t)),\end{aligned}$$

and by Theorem 6.1,

$$\begin{aligned} g(\hat{W}(t)) &= \kappa \cdot \hat{W}(t) \\ &= \kappa \cdot \left(\hat{W}^{\upsilon,*}(t) + \tilde{G}(\hat{U}(t) - \hat{U}^{\upsilon,*}(t))\right) \\ &\geq \kappa \cdot \hat{W}^{\upsilon,*}(t) \\ &= g(\hat{W}^{\upsilon,*}(t)), \end{aligned} \tag{7.6}$$

where we have used Assumption 7.2 and (7.5) for the last inequality. It follows that,

$$\begin{aligned} \hat{J}(\hat{U}) &= \mathbf{E}\left[\int_0^\infty e^{-\gamma t} g(\hat{W}(t))dt\right] \\ &\geq \mathbf{E}\left[\int_0^\infty e^{-\gamma t} g(\hat{W}^{\upsilon,*}(t))dt\right] = \hat{J}(\hat{U}^{\upsilon,*}). \end{aligned} \tag{7.7}$$

Since $\hat{U} \in \hat{\mathcal{A}}^{\upsilon}$ was arbitrary, $\hat{J}(\hat{U}^{\upsilon,*}) = \hat{J}^{\upsilon,*}$. Since υ was arbitrary, this holds for all $\upsilon \in \mathcal{V}$.

Now, for each $\upsilon \in \mathcal{V}$, $\hat{U}^{\upsilon,*} = \Phi(\hat{\xi}^{\upsilon})$ and the corresponding state process $\hat{W}^{\upsilon,*} = \Psi(\hat{\xi}^{\upsilon})$ where $\Psi(x) = x + \tilde{G}\Phi(x)$ for $x \in \mathbf{D}_+^m$. Thus, the law of $\hat{W}^{\upsilon,*}$ is uniquely determined by that of the Brownian motion $\hat{\xi}^{\upsilon}$, and this is the same for each $\upsilon \in \mathcal{V}$. Since the value of $\hat{J}(\hat{U}^{\upsilon,*})$ just depends on the distribution of $\hat{W}^{\upsilon,*}$, it follows that this is the same for all $\upsilon \in \mathcal{V}$, and hence it is the value of $\hat{J}^*$. □

In view of Theorem 7.3, to specify an optimal solution of the EWF, we may fix $\upsilon \in \mathcal{V}$ and use the optimal control associated with the setup indexed by υ. Henceforth we suppress the subscript υ from $\hat{\xi}^{\upsilon}$, $\hat{U}^{\upsilon,*}$ and $\hat{W}^{\upsilon,*}$. We now describe $\hat{U}^*$ and $\hat{W}^*$ more explicitly below; in the case when $\mathbb{N}^{ext} = \mathbb{L}-1$ this can be used to explicitly construct $\hat{U}^*$ from $\hat{\xi}$. For $l \in \mathcal{L}$, consider the positive constants $\beta_l = \tilde{G}_l^l$ and $\alpha_l^j = -\tilde{G}_l^{t^p(j)}$ for each $j \in a^c(l)$. With the convention that the sum over an empty set is zero, we have for each $l \in \mathcal{L}$ and $t \geq 0$,

$$\hat{W}_l^*(t) = \hat{\xi}_l(t) - \sum_{j \in a^c(l)} \alpha_l^j \hat{U}^*_{t^p(j)}(t) + \beta_l \hat{U}_l^*(t), \tag{7.8}$$

where by the minimality of $\hat{U}^*$ (see equation (5.3), the remark following it, and Lemma 9 in [35]),

$$\hat{U}_l^*(t) = \left(-\frac{1}{\beta_l} \inf_{0 \le s \le t} \left\{ \hat{\xi}_l(s) - \sum_{j \in a^c(l)} \alpha_l^j \hat{U}_{t^p(j)}^*(s) \right\} \right) \vee 0. \tag{7.9}$$

Indeed, $\beta_l \hat{U}_l^*$ is the solution of the one-dimensional Skorokhod problem for $\hat{\xi}_l - \sum_{j \in a^c(l)} \alpha_l^j \hat{U}_{t^p(j)}^*$. This is the minimal process that can be added to $\hat{\xi}_l - \sum_{j \in a^c(l)} \alpha_l^j \hat{U}_{t^p(j)}^*$ to keep $\hat{W}_l^*$ non-negative and accordingly, $\hat{U}_l^*$ can only increase when $\hat{W}_l^*$ is zero. Note that (7.9) provides an endogenous description of $\hat{U}^*$. In the case when $\mathbb{N}^{ext} = \mathbb{L} - 1$, this can be solved explicitly. When $\mathbb{N}^{ext} = \mathbb{L}$, even though an explicit form is not possible, the result of Yang [35] provides existence of a solution.

Given the above solution of the REWF (and hence of the EWF by Theorem 7.2), one can specify the Brownian queue-length and idle-time processes for the BCP that accompany it as follows. We do this for the original M and G; if one uses M^c and G^c in their place, one has to multiply the y's in the formulas below by the appropriate components of c. There are two cases to consider.

Case I: Suppose that $\mathbb{N}^{ext} = \mathbb{L} - 1$. Then for $l \in \mathcal{L}$, $k \in \mathcal{K}$, and $t \ge 0$,

$$\tilde{Q}_{i_l^*}^*(t) = \frac{\hat{W}_l^*(t)}{y_{i_l^*}^l}, \qquad \tilde{Q}_i^* \equiv 0 \text{ for } i \in \mathcal{I}_l \setminus \{i_l^*\}, \tag{7.10}$$

$$\tilde{I}_{k_{\mathbb{L}}^*}^*(t) = \hat{U}_{\mathbb{L}}^*(t), \qquad \tilde{I}_k^* \equiv 0 \text{ for } k \ne k_{\mathbb{L}}^*. \tag{7.11}$$

Case II: Suppose that $\mathbb{N}^{ext} = \mathbb{L}$. Then for $l \in \mathcal{L}$, $k \in \mathcal{K}$ and $t \ge 0$,

$$\tilde{Q}_{i_l^*}^*(t) = \frac{\hat{W}_l^*(t)}{y_{i_l^*}^l}, \qquad \tilde{Q}_i^* \equiv 0 \text{ for } i \in \mathcal{I}_l \setminus \{i_l^*\}, \qquad \tilde{I}_k^* \equiv 0 \text{ for } k \in \mathcal{K}. \tag{7.12}$$

We note in particular that in *Case II*, there is no Brownian idle-time. An associated control process $\tilde{Y}^*$ can be defined by following the proof of Theorem 4.1 (see Theorem 5.2 of [21]).

7.4. *Proof of Theorem 7.1.* We shall use the following in the proof of Theorem 7.1.

DEFINITION 7.5. *For $l \in \mathcal{L}$ fixed, let $\pi_1 = l$. We inductively define $\pi_2, \ldots, \pi_\iota$, where $\iota \leq \mathbb{L}$, as follows. If $\mathbb{N}^{ext} = \mathbb{L}$, then $\pi_{j+1} = t^c(a^p(\pi_j))$ for $j = 1, \ldots, \iota - 1$ and ι is the first index such that $t^c(a^p(\pi_\iota)) = \pi_j$ for some $j \leq \iota - 1$. If $\mathbb{N}^{ext} = \mathbb{L} - 1$, then assuming that $\pi_1, \ldots, \pi_j$ have been defined, if $a^p(\pi_j)$ is defined, let $\pi_{j+1} = t^c(a^p(\pi_j))$, or if $a^p(\pi_j)$ is not defined, set $\iota = j$ and the induction procedure stops.*

REMARK 7.4. *If $\mathbb{N}^{ext} = \mathbb{L}$, then $a^p(\pi_j)$ exists for all $j \leq \iota$ and ι is well defined since $\mathcal{H}$ contains a loop. For example, if $\iota = \mathbb{L}$ and $t^c(a^p(\pi_\iota)) = \pi_1$, then $\mathcal{H}$ is a "circle of trees" connected via non-basic activities. If $\mathbb{N}^{ext} = \mathbb{L} - 1$, then by Convention 7.1, $\pi_\iota = \mathbb{L}$. The sequence $\pi_1, \ldots, \pi_\iota$ consists of distinct entries and depends on the choice of $l \in \mathcal{L}$ and the original enumeration of trees.*

Proof of Theorem 7.1. We need to show that each column of G is in $\mathcal{C}(\tilde{G})$. The matrix $\tilde{G}$ includes all of the columns of G which correspond to non-basic activities in $\mathcal{N}^{ext}$. By (6.12)–(6.14), each column of G that does not correspond to a non-basic activity in $\mathcal{N}^{ext}$ is a positive constant times e_l, for some l, where e_l is the l^{th} vector in the standard basis for $\mathbf{R}_+^{\mathbb{L}}$. Therefore, we need to show that for each l, $e_l \in \mathcal{C}(\tilde{G})$. It is enough to show that some positive constant times e_l is in $\mathcal{C}(\tilde{G})$. There are two cases to consider.

Case I: $\mathbb{N}^{ext} = \mathbb{L} - 1$. Fix $l \in \mathcal{L}$. Let $\pi_1, \ldots, \pi_\iota$ be as in Definition 7.5. For each $j = 1, \ldots, \iota - 1$. $\tilde{G}^{\pi_j}$ corresponds to the external non-basic activity $a^p(\pi_j)$ which is processed in $\mathcal{T}_{\pi_j}$ and consumes from $\mathcal{T}_{\pi_{j+1}}$; accordingly $\tilde{G}^{\pi_j}_{\pi_j} > 0$, $\tilde{G}^{\pi_j}_{\pi_{j+1}} < 0$ and all other entries of $\tilde{G}^{\pi_j}$ are zero (see Remark 7.2). Furthermore, since $\pi_\iota = \mathbb{L}$, $\tilde{G}^{\pi_\iota}_{\pi_\iota} > 0$ and all other entries of $\tilde{G}^{\pi_\iota}$ are zero. It follows by successive cancellation that there are positive constants $d_1, \ldots, d_\iota$ such that

$$d_1\tilde{G}^{\pi_1} + d_2\tilde{G}^{\pi_2} + \cdots + d_\iota\tilde{G}^{\pi_\iota},$$

has a positive l^{th} entry and all other entries are zero. Hence, this vector is a positive constant times e_l and is in $C(\tilde{G})$.

Case II: $\mathbb{N}^{ext} = \mathbb{L}$. Fix $l \in \mathcal{L}$ and let $\pi_1, \ldots, \pi_\iota$ be as in Definition 7.5. As in *Case I*, for $j = 1, \ldots, \iota$, $\tilde{G}^{\pi_j}_{\pi_j} > 0$, $\tilde{G}^{\pi_j}_{\pi_{j+1}} < 0$, and all other entries of $\tilde{G}^{\pi_j}$ are zero, where we define $\pi_{\iota+1} = t^c(a^p(\pi_\iota))$. Let $1 \leq \iota' \leq \iota - 1$ be such that $\pi_{\iota'} = t^c(a^p(\pi_\iota))$. It follows that there are positive constants $d_{\iota'}, \ldots, d_\iota$ such that the entry of

$$u = d_{\iota'}\tilde{G}^{\pi_{\iota'}} + d_{\iota'+1}\tilde{G}^{\pi_{\iota'+1}} + \cdots + d_\iota\tilde{G}^{\pi_\iota}$$

with index $\pi_{\iota'}$ is $d_{\iota'}\tilde{G}^{\pi_{\iota'}}_{\pi_{\iota'}} + d_{\iota}\tilde{G}^{\pi_{\iota}}_{\pi_{\iota'}}$ and all other entries are zero. Then,

$$\sum_{j=\iota'}^{\iota} \kappa \cdot \tilde{G}^{\pi_j} d_j = \kappa \cdot u = \kappa_{\pi_{\iota'}} \left(d_{\iota'}\tilde{G}^{\pi_{\iota'}}_{\pi_{\iota'}} + d_{\iota}\tilde{G}^{\pi_{\iota}}_{\pi_{\iota'}} \right),$$

where the left side is (strictly) positive by Assumption 7.2 and the positivity of the d_j's. Since $\kappa_{\pi_{\iota'}} > 0$, it follows that so too is $d_{\iota'}\tilde{G}^{\pi_{\iota'}}_{\pi_{\iota'}} + d_{\iota}\tilde{G}^{\iota}_{\pi_{\iota'}}$ and the vector u is a positive constant times $e_{\pi_{\iota'}}$. Then, in a similar manner to that in *Case I*, it follows by successive cancellation that there are positive constants $c_1, \ldots, c_{\iota'}$ such that

$$c_1\tilde{G}^{\pi_1} + c_2\tilde{G}^{\pi_2} + \cdots + c_{\iota'-1}\tilde{G}^{\pi_{\iota'-1}} + c_{\iota'}u,$$

has a positive l^{th} entry and all other entries are zero. Hence, this vector is a positive constant times e_l and is in $C(\tilde{G})$. □

8. Proposed interpretation of the optimal solution of the BCP. In this section we describe a proposed interpretation of the solution obtained in the last section for the REWF, EWF and the BCP. Recall that we assume that Assumption 6.1 holds, i.e., the cost function in the BCP is linear.

As in Section 3, here we consider a sequence of parallel server systems indexed by $r \in [1, \infty)$, where in particular as $r \to \infty$ the first order parameters in the r^{th} system satisfy the heavy traffic Assumption 3.1; we also assume that Assumptions 7.1 and 7.2 are satisfied. Here M and G are fixed as in Section 7 (as before, if M^c, G^c are used with $c \neq \mathbf{1}$, one has to replace y^l by $c^l y^l$ in the following). As shown in Section 7, under these assumptions, one can solve the REWF exactly and hence the EWF and BCP. Nevertheless, interpreting this solution for the original parallel server system is a challenging problem. In Section 8.1 we outline a proposed interpretation and in Section 8.2 we expand on some details of this.

8.1. *Overall description of the control policy.* We introduce a notion of (nominal) workload for the r^{th} parallel server system: $W^r = MQ^r$. The solution of the (R)EWF and of the associated BCP suggest that the bulk of the work in each tree $\mathcal{T}_l$ should be kept in the cheapest buffer i^*_l. Each tree $\mathcal{T}_l$ can be viewed as a parallel server system with one-dimensional workload W^r_l. When the workload of each tree in the r^{th} system is significantly greater than zero, each tree should be controlled using only basic activities in the tree and operating under a dynamic threshold-type policy as described in [7] for the case of one-dimensional workload. This should achieve a pooling of servers in each tree to approximately minimize the workload in each tree and

to push the bulk of the work in the tree into the cheapest buffer, which is buffer i_l^* for tree $\mathcal{T}_l$. When the workload for a tree is below a small threshold, there are two cases to consider: (i) for $\mathbb{N}^{ext} = \mathbb{L}$, when the workload W_l^r of $\mathcal{T}_l$ in the r^{th} system is close to zero, it should be kept non-negative by switching the server s_l^* that can process the external non-basic activity $a^p(l)$ over to processing that activity; (ii) for $\mathbb{N}^{ext} = \mathbb{L} - 1$, for $l = 1, \ldots, \mathbb{L} - 1$, when the workload W_l^r of $\mathcal{T}_l$ in the r^{th} system is close to zero, it should be kept non-negative in a similar manner to when $\mathbb{N}^{ext} = \mathbb{L}$, except that for the workload $W_{\mathbb{L}}^r$ of $\mathcal{T}_{\mathbb{L}}$, this should be kept non-negative by allowing the server $k_{\mathbb{L}}^*$ to idle only when there are no jobs for that server to process. We elaborate on this control policy, which is a dynamic threshold-type policy, in greater detail below.

8.2. *Threshold policy.* We first consider the case where $\mathbb{N}^{ext} = \mathbb{L}$. For each $l \in \{1, \ldots, \mathbb{L}\}$, let s_l^* be the server in $\mathcal{T}_l$ that performs the external non-basic activity $a^p(l)$ and let b_l^* be the buffer in $\mathcal{T}_{t^c(l)}$ that activity $a^p(l)$ consumes from. Recall that $a^c(l)$ is the set of external non-basic activities that consume from buffers in $\mathcal{T}_l$. Let $b^\dagger(a^c(l))$ be the set of buffers that have material consumed by activities in $a^c(l)$. Each buffer will have a threshold placed on it. Let $L_i^r \geq 0$ denote the size of the threshold placed on buffer i in the r^{th} system. For each l, define

$$D^{l,r} = \{q \in \mathbf{R}_+^{\mathbb{I}} : q_i \leq L_i^r \text{ for all } i \in \mathcal{I}_l\}.$$

When $Q^r \in D^{l,r}$, the workload in tree $\mathcal{T}_l$ is considered to be small and service of the external non-basic activity $a^p(l)$ will be initiated. The dynamic control policy is described as follows; it depends on the the current value of the queue-length Q^r.

Fix $l \in \mathcal{L}$. For the servers in tree $\mathcal{T}_l$, there are two cases to consider.

Case I: $Q^r \notin D^{l,r}$. The workload $W_l^r \geq \min(y_i^l L_i^r : i \in \mathcal{I}_l)$. The servers in $\mathcal{T}_l$ use a threshold-type control policy that only involves the use of basic activities. This policy was outlined in [34] and elaborated on in [7]. We summarize it for our context below. A key to the description of this policy is a hierarchical structure of the server-buffer tree $\mathcal{T}_l$ and an associated protocol for the dynamic allocation of buffer priorities at each server. Visualizing the tree as growing downwards from its root, this protocol is described in an iterative manner, working from the bottom of the tree up towards the root where the root of $\mathcal{T}_l$ is the server k_l^* that serves buffer i_l^* via a basic activity. For the following description, buffers that link one level of servers to the next highest level of servers are called transition buffers. Buffers that are served

by a single server are called terminal buffers. Note that since $\mathcal{T}_l$ is a tree, with the exception of k_l^*, there is exactly one transition buffer immediately above each server. However, unless a given server is at the lowest level, there may be one or more transition buffers immediately below this server.

Consider a server at the lowest level. This server gives the lowest priority to the buffer that is immediately above it in $\mathcal{T}_l$ (there will always be such a buffer unless the server is at the root of the tree). This buffer is also served by a server in the next level of servers up in the tree and is therefore a transition buffer. At this lowest level, the highest priority is given to terminal buffers. The priority ranking for these buffers is not so important. For concreteness we rank them in such a way that the lower numbered ones have higher priority over the higher numbered ones, see [7] for details. Next, we look at a server in the next level of servers in the tree. This server may serve several transition buffers immediately below it and unless it is the root of the tree it also serves a transition buffer immediately above it. Also, it may serve several terminal buffers. The highest priority is given to transition buffers immediately below the server. If there are several such buffers, they are ranked in such a way that the lower numbered ones receive higher priority over the higher numbered ones. However, if the number of jobs for a transition buffer associated with such an activity is at or below the threshold for this buffer, service of that activity is suspended. The next priority is given to terminal buffers immediately below the server. Again the lower numbered ones receive higher priority over the higher numbered ones. The lowest priority is given to the transition buffer immediately above the server in $\mathcal{T}_l$. This procedure is repeated until the root of the tree is reached. The server at the root behaves as do other servers except that it gives the lowest priority to the cheapest buffer i_l^*. If the number of jobs in the cheapest buffer i_l^* and in terminal buffers for server k_l^* equals zero, server k_l^* starts serving its transition buffers. If two or more servers simultaneously attempt to serve a particular transition buffer, a tie breaking rule is used to determine which server takes a job first, see [7]. A server will idle if it has no more jobs to serve.

There is one exception to the above protocol. This relates to when buffers in $b^\dagger(a^c(l))$ are being served. These are the buffers in $\mathcal{T}_l$ that can be processed by external non-basic activities. If such a buffer is a terminal buffer that is *not* the cheapest buffer i_l^* in $\mathcal{T}_l$, then it needs special treatment to ensure that any non-basic activity serving it will have enough jobs to serve. When the number of jobs in such a buffer i is at or below the level L_i^r, any service of this buffer by any server in $\mathcal{T}_l$ is suspended until the queue-length of this buffer is strictly greater than L_i^r.

Case II: $Q^r \in D^{l,r}$. As long as $Q^r \in D^{l,r}$, server s_l^* tries to perform activity $a^p(l)$, which processes jobs from b_l^* in tree $\mathcal{T}_{t^c(l)}$. If there are no jobs in the buffer b_l^* for server s_l^* to serve, then s_l^* idles until there is a job in buffer b_l^* for it to serve, or until Q^r exits $D^{l,r}$. All other servers in $\mathcal{T}_l$ operate under the protocol described in *Case I*.

The policy when $\mathbb{N}^{ext} = \mathbb{L} - 1$ is the same as that described above with one exception. In this case, there is no external non-basic activity that is processed in $\mathcal{T}_{\mathbb{L}}$ and so all servers in $\mathcal{T}_{\mathbb{L}}$ operate as in *Case I*.

REMARK 8.1. *Readers may wonder why we allow the server s_l^* to idle in Case II, when there still might be a small amount of work for it to process in the tree $\mathcal{T}_l$. We do this because of experience we gathered from our proof of asymptotic optimality of our policy for the example given in Section 9 and our exploration of other examples. Our idling convention facilitates our proof of the heavy traffic limit for the queue-lengths of buffers in $\mathcal{T}_l$. In particular, it makes it feasible to track the deviations of transition buffers from their thresholds. In fact, this forced idling of server s_l^* will only occur when both the workload in $\mathcal{T}_l$ is small and there are no jobs in buffer b_l^*. We expect this event to occur so rarely that the amount of unnecessary idletime incurred by server s_l^* due to this rule will be negligible in the diffusion limit. (For the example in Section 9, we prove this in [31].) Indeed, we conjecture that the policy described above has the same heavy traffic behavior as the policy in which Case II is amended to allow server s_l^* to revert to serving jobs in $\mathcal{T}_l$ when $Q^r \in D^{l,r}$ and there are no jobs in buffer b_l^* for s_l^* to serve. However, it will be more difficult to prove the asymptotic optimality of this amended policy in general, as it is difficult to obtain a detailed (excursion level) apriori estimate of how much time is spent in this modified activity. Nevertheless, practitioners wishing to use the general form of our policy may consider making this amendment as it may lead to slightly enhanced performance in the prelimit.*

For the case $\mathbb{N}^{ext} = \mathbb{L}$, we anticipate that for asymptotic optimality of our control policy, the thresholds should be such that for each i, as $r \to \infty$, $L_i^r/r \to 0$ and $L_i^r \to \infty$. The precise size of the thresholds should be suitably chosen depending on higher order moment assumptions placed on the primitive arrival and service processes. In particular, we conjecture that as in [7], these thresholds can be chosen to be of the order of $\log r$ as $r \to \infty$ under the following exponential moment assumptions. With finiteness of fewer moments, the thresholds would likely need to be bigger (up to $o(r)$). Thresholds for the case $\mathbb{N}^{ext} = \mathbb{L} - 1$ should be set similarly, except that for buffers in $\mathcal{T}_{\mathbb{L}}$ that are terminal buffers that are either not served by external

non-basic activites or that correspond to the cheapest buffer in the tree, a threshold of zero can be set. This is because thresholds on these buffers are only used to define $D^{\mathbb{L},r}$ and this set is not used to specify the policy in this case when there is no external non-basic activity processed in $\mathcal{T}_{\mathbb{L}}$.

ASSUMPTION 8.1. *For $i \in \mathcal{I}$, $j \in \mathcal{J}$, for all $m \geq 1$, let $u_i(m) = \frac{1}{\lambda_i}\check{u}_i(m)$ and $v_j(m) = \frac{1}{\mu_j}\check{v}_i(m)$. Assume that there is an open neighborhood $\mathcal{O}$ of $0 \in \mathbf{R}$ such that for all $\ell \in \mathcal{O}$,*

$$\Lambda_i^a(\ell) \equiv \log \mathbf{E}\left[e^{\ell u_i(1)}\right] < \infty \qquad \text{for} \quad i \in \mathcal{I}, \tag{8.1}$$

and

$$\Lambda_j^s(\ell) \equiv \log \mathbf{E}\left[e^{\ell v_j(1)}\right] < \infty \qquad \text{for} \quad j \in \mathcal{J}. \tag{8.2}$$

Assuming this, we conjecture that the above policy is asymptotically optimal in the following sense.

CONJECTURE 8.1. *Let $\{T^r\}$ be any sequence of scheduling controls (one for each member of the sequence of parallel server systems) and let $\{T^{r,*}\}$ denote the sequence of controls associated with our threshold policy for suitable thresholds $L_i^r, i \in \mathcal{I}$, chosen to be of order $\log r$ as $r \to \infty$. Then*

$$\liminf_{r\to\infty} \hat{J}^r(T^r) \geq J^* = \lim_{r\to\infty} \hat{J}^r(T^{r,*}) \tag{8.3}$$

and $J^* < \infty$ is the optimal value of the BCP.

In the next section, we illustrate our proposed policy for a specific example. In a separate work [31], we prove asymptotic optimality of the policy for this example.

Our proposed policy provides a simple, threshold-based, feedback control interpretation of the solution of the Brownian Control Problem. However, this policy is only one possible asymptotically optimal policy. There are likely many other policies that are asymptotically optimal. In particular, asymptotically optimal discrete review policies likely can be constructed along the lines of the general BIGSTEP scheme proposed by Harrison [15], which was shown to be asymptotically optimal under a complete resource pooling assumption in [1, 16]. Indeed, under discrete review, the transition between *Cases I and II* could be approximately detected and our policy for *Case I* could likely be replaced by a discrete review scheme similar to that used by Ata and Kumar [1], and our policy for *Case II* could be approximated by a discrete review policy in which the appropriate non-basic activity is used.

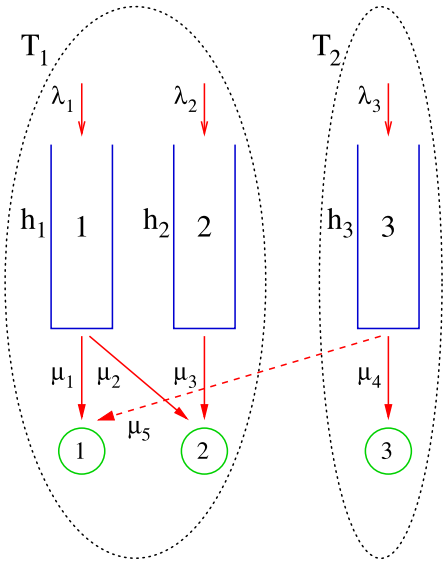

FIG 4. *Graph $\mathcal{H}$ for the example in Section 9. Here* T_l *stands for the tree* $\mathcal{T}_l$, $l = 1, 2$.

9. Illustrative example.

9.1. *Description and first order data.* We consider a sequence of parallel server systems indexed by $r \in [1, \infty)$ whereeach system has 3 buffers, 3 servers and 5 activities (see Figure 4). The first order parameters for the r^{th} member of the sequence has

$$C = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}, \quad A = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix},$$

(9.1)

$$\lambda^r = (\lambda_1^r, \lambda_2^r, \lambda_3^r), \quad \mu^r = (\mu_1^r, \mu_2^r, \mu_3^r, \mu_4^r, \mu_5^r),$$

such that the following holds.

ASSUMPTION 9.1. *There are constant vectors* $\lambda = (\lambda_1, \lambda_2, \lambda_3), \mu = (\mu_1, \mu_2, \mu_3, \mu_4, \mu_5), (\theta_1, \theta_2, \theta_3)$ *such that as* $r \to \infty$,

(i) $\lambda_i^r \to \lambda_i > 0$ *for* $i = 1, 2, 3$,
(ii) $\mu_j^r \to \mu_j > 0$ *for* $j = 1, \ldots, 5$,
where $\lambda_1 > \mu_1$, $\lambda_2 < \mu_3$, $\lambda_3 = \mu_4$, $\frac{\lambda_1 - \mu_1}{\mu_2} = 1 - \frac{\lambda_2}{\mu_3}$,
(iii) $r\mu_2^r \left(\frac{\lambda_1^r - \mu_1^r}{\mu_2^r} - \frac{\lambda_1 - \mu_1}{\mu_2} \right) \to \theta_1$,

(iv) $r\mu_3^r\left(\frac{\lambda_2^r}{\mu_3^r}-\frac{\lambda_2}{\mu_3}\right)\to\theta_2$,
(v) $r(\lambda_3^r-\mu_4^r)\to\theta_3$.

Assumptions 3.2, 3.3 are satisfied with

$$x^*=\left(1,\frac{\lambda_1-\mu_1}{\mu_2},\frac{\lambda_2}{\mu_3},1,0\right). \tag{9.2}$$

9.2. *Brownian Control Problem.* Assuming linear holding costs as in Assumption 6.1, the Brownian Control Problem associated with the above parameters is as follows.

DEFINITION 9.1. *(Brownian Control Problem).*

$$\textit{minimize}\quad \mathbf{E}\left(\int_0^\infty e^{-\gamma t}h\cdot\tilde{Q}(t)dt\right) \tag{9.3}$$

using a 5-dimensional adapted control process $\tilde{Y}=(\tilde{Y}_1,\tilde{Y}_2,\tilde{Y}_3,\tilde{Y}_4,\tilde{Y}_5)$ *defined on some filtered probability space* $(\tilde{\Omega},\tilde{\mathcal{F}},\{\tilde{\mathcal{F}}_t\},\tilde{\mathbf{P}})$ *that supports 3-dimensional adapted processes* $\tilde{Q}$ *and* $\tilde{X}$ *such that* $\tilde{\mathbf{P}}$*-a.s. for all* $t\ge 0$*:*

$$\tilde{Q}_1(t)=\tilde{X}_1(t)+\mu_1\tilde{Y}_1(t)+\mu_2\tilde{Y}_2(t)\ge 0, \tag{9.4}$$
$$\tilde{Q}_2(t)=\tilde{X}_2(t)+\mu_3\tilde{Y}_3(t)\ge 0, \tag{9.5}$$
$$\tilde{Q}_3(t)=\tilde{X}_3(t)+\mu_4\tilde{Y}_4(t)+\mu_5\tilde{Y}_5(t)\ge 0, \tag{9.6}$$
$$\tilde{I}_1=\tilde{Y}_1+\tilde{Y}_5 \textit{ is non-decreasing},\quad \tilde{I}_1(0)\ge 0, \tag{9.7}$$
$$\tilde{I}_2=\tilde{Y}_2+\tilde{Y}_3 \textit{ is non-decreasing},\quad \tilde{I}_2(0)\ge 0, \tag{9.8}$$
$$\tilde{I}_3=\tilde{Y}_4 \textit{ is non-decreasing},\quad \tilde{I}_3(0)\ge 0, \tag{9.9}$$
$$\tilde{Y}_5 \textit{ is non-increasing},\quad \tilde{Y}_4(0)\le 0, \tag{9.10}$$

where $\tilde{X}$ *is a 3-dimensional* $\{\tilde{\mathcal{F}}_t\}$*-Brownian motion starting at the origin with drift* $\theta=(\theta_1,\theta_2,\theta_3)$ *and with diagonal covariance matrix* Σ *whose* i^{th} *diagonal entry is equal to* $\lambda_i a_i^2+\sum_{j=1}^5 C_{ij}\mu_j b_j^2 x_j^*$ *for* $i=1,2,3$.

9.3. *Solution of the REWF and BCP.* According to Theorem 5.2, since the server-buffer graph $\mathcal{G}$ consists of two trees $\mathcal{T}_1$ and $\mathcal{T}_2$, the workload is two-dimensional. There are four basic activities and a single (external) non-basic activity. The extended server-buffer graph $\mathcal{H}$ (see Figure 4) satisfies Assumption 7.1. Following the steps outlined in Sections 4–6, we proceed to compute a workload matrix M and control matrix G that goes with it. For

this, we solve the dual programs for $\mathcal{T}_1$ and $\mathcal{T}_2$. The unique solution of the dual program for $\mathcal{T}_1$ is given by

$$(y_1^1, y_2^1) = \left(\frac{1}{\mu_1+\mu_2}, \frac{\mu_2}{\mu_3(\mu_1+\mu_2)}\right), \qquad (z_1^1, z_2^1) = \left(\frac{\mu_1}{\mu_1+\mu_2}, \frac{\mu_2}{\mu_1+\mu_2}\right),$$

and the unique solution of the dual program for $\mathcal{T}_2$ is given by

$$y_1^2 = \frac{1}{\mu_4}, \qquad z_1^2 = 1. \tag{9.11}$$

By Lemma 6.5 we can multiply (y^1, z^1) by $\mu_1+\mu_2$ to obtain a workload matrix

$$M = \begin{pmatrix} 1 & \mu_2/\mu_3 & 0 \\ 0 & 0 & 1/\mu_4 \end{pmatrix}, \tag{9.12}$$

with corresponding control matrix

$$G = \begin{pmatrix} \mu_1 & \mu_2 & 0 & \mu_1 \\ 0 & 0 & 1 & -\mu_5/\mu_4 \end{pmatrix}, \tag{9.13}$$

and (κ_1, κ_2) is such that $\kappa_1 = \min(h_1, h_2\mu_3/\mu_2)$, $\kappa_2 = h_3\mu_4$. By Definition 7.2 the control matrix $\tilde{G}$ that goes with the REWF (see Definitions 4.7-4.8) is

$$\tilde{G} = \begin{pmatrix} \mu_1 & 0 \\ -\mu_5/\mu_4 & 1 \end{pmatrix}. \tag{9.14}$$

Given $\tilde{X}$ as in Definition 9.1, let $\hat{\xi} = M\tilde{X}$. Then, $\hat{\xi}$ is a 2-dimensional Brownian motion with drift $M\theta$ and covariance matrix $M\Sigma M'$. In terms of $\hat{\xi}$ the workload $\hat{W}$ in the REWF is given by

$$\begin{aligned} \hat{W}_1(t) &= \hat{\xi}_1(t) + \mu_1\hat{U}_1(t), \\ \hat{W}_2(t) &= \hat{\xi}_2(t) + \hat{U}_2(t) - (\mu_5/\mu_4)\hat{U}_1(t). \end{aligned}$$

The correspondence with controls in the BCP is

$$\hat{U}_1 = -\tilde{Y}_5, \qquad \hat{U}_2 = \tilde{I}_3.$$

Henceforth, we assume that the parameters satisfy the following.

ASSUMPTION 9.2.

(i) $h_1\mu_2 > h_2\mu_3$,
(ii) $h_2\mu_1\mu_3 > h_3\mu_2\mu_5$.

The first assumption (i) assumes that buffer 2 is the "cheapest" in the tree $\mathcal{T}_1$. Part (ii) of Assumption 9.2 corresponds to Assumption 7.2, which implies that activity 5 is an expensive activity in the sense that in the optimal solution of the BCP, it will only be used to reduce workload in tree $\mathcal{T}_2$ when there is no work in $\mathcal{T}_1$.

The assumptions of Section 7.2 are satisfied and therefore by Section 7.3, the solution of the REWF and BCP are given by the following. For all $t \geq 0$,

$$\hat{W}_1^*(t) = \hat{\xi}_1(t) + \mu_1 \tilde{U}_1^*(t), \quad \hat{U}_1^*(t) = \left(-\frac{1}{\mu_1} \inf_{0 \leq s \leq t} \{\hat{\xi}_1(s)\} \right) \vee 0, \tag{9.15}$$

$$\hat{W}_2^*(t) = \hat{\xi}_2(t) + \hat{U}_2^*(t) - (\mu_5/\mu_4)\hat{U}_1^*(t), \tag{9.16}$$

$$\hat{U}_2^*(t) = \left(-\inf_{0 \leq s \leq t} \{\hat{\xi}_2(s) - (\mu_5/\mu_4)\hat{U}_1^*(s)\} \right) \vee 0, \tag{9.17}$$

$$\tilde{Q}_1^*(t) = 0, \quad \tilde{Q}_2^*(t) = (\mu_3/\mu_2)\hat{W}_1^*(t), \quad \tilde{Q}_3^*(t) = \mu_4 \hat{W}_2^*(t), \tag{9.18}$$

$$\tilde{I}_1^*(t) = \tilde{Y}_1^*(t) + \tilde{Y}_5^*(t) = 0, \ \tilde{I}_2^*(t) = \tilde{Y}_2^*(t) + \tilde{Y}_3^*(t) = 0, \tag{9.19}$$

$$\tilde{Y}_4^*(t) = \tilde{I}_3^*(t) = \hat{U}_2^*(t), \ \tilde{Y}_5^*(t) = -\hat{U}_1^*(t), \tag{9.20}$$

$$\tilde{Y}_1^*(t) = -\tilde{Y}_5^*(t) = \hat{U}_1^*(t), \tag{9.21}$$

$$\tilde{Y}_2^*(t) = -\mu_2^{-1}(\tilde{X}_1(t) + \mu_1 \tilde{Y}_1^*(t)), \tag{9.22}$$

$$\tilde{Y}_3^*(t) = \mu_3^{-1}(\tilde{Q}_2^*(t) - \tilde{X}_2(t)), \tag{9.23}$$

and the associated minimum cost is

$$\tilde{J}^* = \mathbf{E}\left(\int_0^\infty e^{-\gamma t} \left((h_2\mu_3/\mu_2)\hat{W}_1^*(t) + h_3\mu_4 \hat{W}_2^*(t) \right) dt \right). \tag{9.24}$$

In this instance, the optimal control $\tilde{Y}^*$ for the BCP is uniquely determined by the optimal control $\hat{U}^*$ for the REWF. In general, this need not be the case (see [19]). In the example treated here, when $\hat{W}_1^*$ hits zero, $\hat{U}_1^*$ increases by a minimal amount in order to keep $\hat{W}_1^*$ non-negative. When $\hat{W}_2^*$ hits zero, $\hat{U}_2^*$ increases by a minimal amount in order to keep $\hat{W}_2^*$ non-negative (see Figure 5 for a depiction of the optimal control directions).

In contrast to the simple structure of the REWF obtained using our decoupled workload matrix, the workload and control matrices proposed by Harrison for this example are given by the following (after premultiplication by a diagonal matrix with positive diagonal entries):

$$M = \begin{pmatrix} 1 & \mu_2/\mu_3 & 0 \\ 1 & \mu_2/\mu_3 & \mu_1/\mu_5 \end{pmatrix}, \tag{9.25}$$

$$G = \begin{pmatrix} \mu_1 & \mu_2 & 0 & \mu_1 \\ \mu_1 & \mu_2 & \mu_4\mu_1/\mu_5 & 0 \end{pmatrix}. \tag{9.26}$$

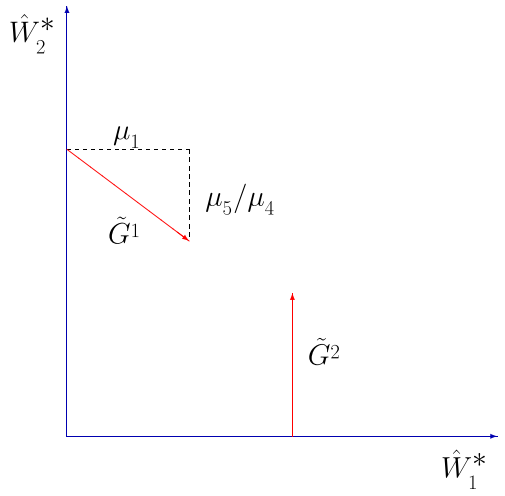

FIG 5. *For the optimal control $\hat{U}^*$ for the REWF, $\hat{U}_1^*$ is the amount of pushing done in the direction $\tilde{G}^1$ and $\hat{U}_2^*$ is the amount of pushing done in the direction $\tilde{G}^2$.*

With this choice for M and G, the state space for the workload in the EWF will be a wedge in the non-negative quadrant of two-dimensional space and all of the control directions have non-negative components. The results of Yang [35] do not apply directly here as that paper requires the state space to be the non-negative orthant, not a subset of it. Furthermore, the cost function for the EWF is not so simple to derive with the above choice of M and G. This is to be compared with the linear function of workload (with positive coefficients) resulting from our decoupled workload matrix.

9.4. *Threshold policy.* The policy which follows is a special instance of the policy proposed in Section 8 for $\mathbb{N}^{ext} = \mathbb{L} - 1$. Recall that the terms class and buffer are interchangeable. The root of the first tree is server 2 and the root of the second tree is server 3. The policy is described using three regions in the two-dimensional state space for (Q_1^r, Q_2^r) that are induced by thresholds on buffers 1 and 2 (see Figure 6).

For each r, let

$$L_1^r = \lfloor \tilde{C} \log r \rfloor + 1,\ L_2^r = \lfloor (8\mu_3 L_1^r)/\lambda_1 \rfloor + 1 \text{ and } L_3^r = 0$$

where $\tilde{C}$ is a sufficiently large positive constant. The policy for servers in tree $\mathcal{T}_1$ when $Q^r \in \mathcal{D}^{1,r}$, is similar to that used in [6]. Indeed, the threshold L_1^r, on the transition buffer 1 in $\mathcal{T}_1$, is of the same form as that in [7]. Here we need an additional threshold L_2^r on buffer 2, which combined with the threshold on buffer 1 helps us to detect when the workload in $\mathcal{T}_1$ is small (of

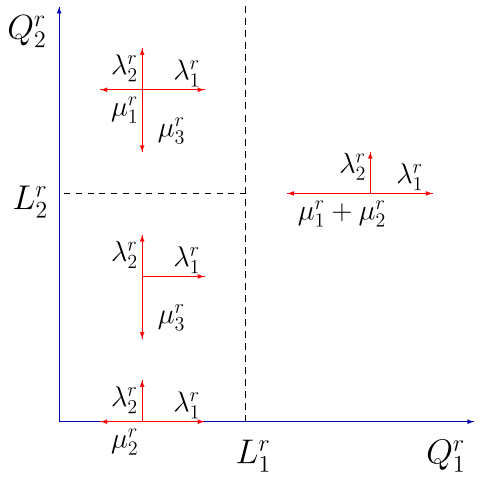

FIG 6. *Depiction of thresholds and related transition rates for* (Q_1^r, Q_2^r) *under the threshold policy.*

order $\log r$). Note that both thresholds are of order $\log r$. The threshold L_3^r can be chosen to be zero because it is a terminal buffer that is the cheapest buffer in $\mathcal{T}_2$, where $\mathbb{N}^{ext} = 2 - 1 = 1$.

In the r^{th} system, the dynamic threshold policy operates as follows:

(i) Server 3 operates whenever possible. In other words server 3 is never idle when there are jobs in buffer 3 or at server 3.

(ii) When the number of class 1 jobs is above the threshold L_1^r, server 1 and server 2 both process class 1 jobs. In particular, as soon as the number of class 1 jobs reaches level $L_1^r + 1$ from below, server 2 suspends any work on class 2 jobs and shifts service to class 1 jobs. As soon as the number of class 1 jobs reaches level L_1^r from above, server 2 suspends any work on class 1 jobs and shifts service to class 2 jobs provided that buffer 2 is non-empty; if there are no class 2 jobs to be served, server 2 continues serving class 1 jobs until a new arrival of a class 2 job occurs.

(iii) When the number of class 1 jobs is at or below the threshold L_1^r and the number of class 2 jobs is above the threshold L_2^r, server 1 works on class 1 jobs and server 2 works on class 2 jobs.

(iv) When the number of class 1 jobs is at or below the threshold L_1^r and the number of class 2 jobs is at or below the threshold L_2^r, server 1 works on class 3 jobs provided that there are jobs in buffer 3 for it to serve and server 2 works on class 2 jobs provided that buffer 2 is

not empty. In this regime, if there are no jobs in buffer 3 that server 1 can serve, then server 1 idles and, if buffer 2 becomes empty, server 2 works on class 1 jobs.

In [31], it is proved under Assumption 8.1 that this policy is asymptotically optimal in the heavy traffic limit for this example for all sufficiently large constants $\tilde{C}$. As pointed out in Section 8, the size (in terms of order relative to r) of the non-zero thresholds depends on the moment assumptions imposed on the interarrival and service times. A relaxation of the exponential moment assumptions would require larger thresholds, which must still be $o(r)$ as $r \to \infty$.

As we remarked in Section 8, the forced idling of server 1 in case (iv), when there are no jobs in buffer 3 for it to serve, is done to facilitate our proof of asymptotic optimality. Indeed, our proof shows that this amount of idling is negligible in diffusion scale in the heavy traffic limit. It is reasonable to conjecture that, if instead of idling in this case, server 1 serves jobs from buffer 1, then the resulting policy will also be asymptotically optimal. However, proving this would require considerably more delicate estimates than are used in our current proof in [31].

The previous paragraph reminds us that the policy we have proposed is only one asymptotically optimal policy. It is likely that there are many other policies that are asymptotically optimal. However, as in the case of complete resource pooling [13], the maximum pressure policy is *not* asymptotically optimal for this example with linear holding costs. This can be seen from an analysis due to Ata and Lin [2] who studied the heavy traffic behavior of certain stochastic processing networks operating under a maximum pressure policy. When specialized to the example treated in this section, their results show that under a maximum pressure policy, the diffusion limit of the queue-length process has no degenerate components. This can be used to show that the maximum pressure policy is not asymptotically optimal.

The reader may wonder what happens if some of the parameters in the example of this section are changed. We give a brief discussion of some possibilities (a)–(c) here.

(a) If the inequality in Assumption 9.2(i) is reversed, then buffer 1 will be the cheapest (or equal cheapest) buffer in the tree $\mathcal{T}_1$. We expect that the policy outlined in Section 8 will again be asymptotically optimal in this case. The main change in the policy from that described above is that the buffer priorities for server 2 are reversed: server 2 gives highest priority to buffer 2 and only serves buffer 1 when buffer 2 is empty.

(b) If the inequality in Assumption 9.2(ii) is reversed, then the non-basic activity is no longer "expensive". In this case, it may be expedient to use the non-basic activity even when the workload in $\mathcal{T}_1$ is not small. We do not know how to solve the EWF in this case, as the solution may involve a free boundary on which the non-basic activity is used.

(c) If $\lambda_1 = \mu_1$ and $\lambda_2 = \mu_3$, then the graph $\mathcal{H}$ will have three trees (with one buffer and one server per tree) connected by two non-basic activities, activities 2 and 5. Assumption 7.1 on the graphical structure will hold with $\mathbb{N}^{ext} = \mathbb{L} - 1$. Provided that Assumption 7.2 on the cost is satisfied, i.e., it is expensive to use the non-basic activities, then the EWF can be solved and our proposed policy from Section 8 can be applied. In a sense, this example is easier to treat than the example in Section 9 because there is no pooling within trees, and the workload and queue-length processes have the same dimension.

10. Further research. There are several directions for further research that are suggested by the simplified structure for the EWF revealed through use of our decoupled workload matrix. We describe three of these below.

The first direction relates to relaxing the assumptions about the cost and graph structure assumptions in Section 7. The cost assumption ensures that the optimal control process $\hat{U}^*$ will only increase when some component of the workload process reaches zero. It would be very interesting to relax this cost assumption. However, it is expected that in general the solution of the EWF will involve a (challenging) free boundary problem in this case. The graph structure assumption ensures that for each workload component, there is just one control direction (i.e., column of $\tilde{G}$) for the REWF that can be used to increase the workload component (and so prevent it from becoming negative when it is zero). The latter ensures that there is just one control direction associated with each boundary face of the orthant and that there exists a least control. If one generalized to allow more than one control direction on each boundary face, a more complex control problem results. It would be interesting to find good sufficient conditions for solvability of such problems which would allow for a more general structure of non-basic activities between trees.

It is natural to consider other cost functions besides linear holding costs. In particular, it is natural to consider convex cost functions of the form $f(q) = \sum_{i=1}^{\mathbb{I}} c_i q_i^{\alpha}$ for some $\alpha > 1$, $c_i > 0, i = 1, \ldots, \mathbb{I}$. It seems likely that some sufficient conditions for solvability of the EWF can be obtained in this case and that judicious use could be made of the maxweight, generalized $c\mu$ or maximum pressure policies of [13, 29, 32] within trees for some asymptotically optimal policies.

In this paper, we have seen that for parallel server systems with more than one-dimensional workload, there can be advantages to choosing a different workload matrix from that proposed by [17]. It would be interesting to explore whether a simplified structure of the EWF (and REWF) can be obtained for more general stochastic processing networks with flexible servers and feedback, by choosing a workload matrix distinct from that proposed in [17].

REFERENCES

[1] B. Ata and S. Kumar. Heavy traffic analysis of open processing networks with complete resource pooling: Asymptotic optimality of discrete review policies. *Annals of Applied Probability*, **15** (2005), 331–391. MR2115046

[2] B. Ata and W. Lin. Heavy traffic analysis of maximum pressure policies for stochastic processing with multiple bottlenecks, *Queueing Systems*, **59** (2008), 191–235. MR2439295

[3] B. Ata and J. A. Van Mieghem. The value of dynamic resource pooling: Should a service network be integrated or product-focused? *Management Science*, **55** (2009), 115–131.

[4] R. Atar, A. Mandelbaum and G. Shaikhet. Queueing systems with many servers: Null controllability in heavy traffic. *Annals of Applied Probability*, **16** (2006), 1764–1804. MR2288704

[5] R. Atar and I. Gurvich, Scheduling parallel servers in the nondegenerate slowdown diffusion regime: Asymptotic optimality results, *Annals of Applied Probability*, **24** (2014), 760–810. MR3178497

[6] S. L. Bell and R. J. Williams. Dynamic scheduling of a system with two parallel servers in heavy traffic with resource pooling: Asymptotic optimality of a threshold policy. *Annals of Applied Probability*, **11** (2001), 608–649. MR1865018

[7] S. L. Bell and R. J. Williams. Dynamic scheduling of a parallel server system in heavy traffic with complete resource pooling: Asymptotic optimality of a threshold policy. *Electronic J. of Probability*, **10** (2005), 1044–1115. MR2164040

[8] C. Berge. *Graphs*, North Hollans, Amsterdam, 1985. MR0809587

[9] V. Bohm. On the continuity of the optimal policy set for linear programs. *SIAM J. Appl. Math.*, **28** (1975), 303–306. MR0371390

[10] A. Budhiraja and A. P. Ghosh. A large deviations approach to asymptotically optimal control of crisscross network in heavy traffic. *Annals of Applied Probability*, **15** (2005), 1887–1935. MR2152248

[11] A. Budhiraja and A. Ghosh, Controlled stochastic networks in heavy traffic: Convergence of value functions, *Annals of Applied Probability*, **22** (2012), 734–791. MR2953568

[12] M. Bramson and R. J. Williams. Two workload properties for Brownian networks. *Queueing Systems*, **45** (2003), 191–221. MR2024178

[13] J. G. Dai and W. Lin. Asymptotic optimality of maximum pressure policies in stochastic processing networks, *Annals of Applied Probability*, **18** (2008), 2239–2299. MR2473656

[14] J. M. Harrison. Brownian models of queueing networks with heterogeneous customer population. In *Stochastic Differential Systems, Stochastic Control Theory and Their Applications*, IMA Volume **10**, W. Fleming and P. L. Lions (eds.), Springer Verlag, New York, 1988, 147–186. MR0934722

[15] J. M. Harrison. The BIGSTEP approach to flow management in stochastic processing networks. In *Stochastic Networks: Theory and Applications*, F. P. Kelly, S. Zachary and I. Ziedins (eds.), Oxford University Press, 1996, 57–90.
[16] J. M. Harrison. Heavy traffic analysis of a system with parallel servers: Asymptotic optimality of discrete review policies. *Annals of Applied Probability*, **8** (1998), 822–848. MR1627791
[17] J. M. Harrison. Brownian models of open processing networks: Canonical representation of workload. *Annals of Applied Probability*, **10** (2000), 75–103. Correction: **13** (2003), 390–393. MR1765204
[18] J. M. Harrison and M. J. López. Heavy traffic resource pooling in parallel server systems. *Queueing Systems*, **33** (1999), 339–368. MR1742575
[19] J. M. Harrison and J. A. Van Mieghem. Dynamic control of Brownian networks: State space collapse and Equivalent Workload Formulations. *Annals of Applied Probability*, **7** (1997), 747–771. MR1459269
[20] J. M. Harrison and L. M. Wein. Scheduling networks of queues: Heavy traffic analysis of a simple open network. *Queueing Systems*, **5** (1989), 265–280. MR1030470
[21] J. M. Harrison and R. J. Williams. Workload reduction of a generalized Brownian network. *Annals of Applied Probability*, **15** (2005), 2255–2295. MR2187295
[22] F. P. Kelly and C. N. Laws. Dynamic routing in open queueing networks: Brownian models, cut constraints and resource pooling. *Queueing Systems*, **13** (1993), 47–86. MR1218844
[23] H. J. Kushner and Y. N. Chen, Optimal control of assignment of jobs to processors under heavy traffic, *Stochastics and Stochastics Reports*, **68** (2000), 177–228. MR1746180
[24] H. J. Kushner and P. Dupuis, *Numerical Methods for Stochastic Control Problems in Continuous Time*, Springer-Verlag, New York, 1992. MR1217486
[25] H. J. Kushner and L. F. Martins, Numerical methods for stochastic singular control problems, *SIAM J. Control and Optimization*, **29** (1991), 1443–1475. MR1132190
[26] C. N. Laws. Resource pooling in queueing networks with dynamic routing. *Advances in Applied Probability*, **24** (1992), 699–726. MR1174386
[27] C. N. Laws and G. M. Louth. Dynamic sequencing of a four station queueing network. *Probab. Engrg. Inform. Sci.*, **4** (1990), 131–156.
[28] C. Maglaras. Continuous-review tracking policies for dynamic control of stochastic networks. *Queueing Systems*, **43** (2003), 43–80. MR1957806
[29] A. Mandelbaum and A. L. Stolyar. Scheduling flexible servers with convex delay costs: Heavy-traffic optimality of the generalized $c\mu$-rule. *Operations Research* **52** (2004), 836–855. MR2104141
[30] L. F. Martins, S. E. Shreve and H. M. Soner. Heavy traffic convergence of a controlled, multiclass queueing system. *SIAM J. Control Optim.* **34** (1996), 2133–2171. MR1416504
[31] V. Pesic and R. J. Williams, Dynamic scheduling of a parallel server system with partial pooling: Heavy traffic analysis of a three-buffer, three-server system, in preparation.
[32] A. L. Stolyar. MaxWeight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic. *Annals of Applied Probability*, **14** (2004), 1–53. MR2023015
[33] L. Wein. Scheduling networks of queues: Heavy traffic analysis of a multistation network with controlable inputs. *Operations Research*, **40** (1992), 312–334.
[34] R. J. Williams. On dynamic scheduling of a parallel server system with complete resource pooling. In *Analysis of Communication Networks: Call Centers, Traffic and*

Performance, D. R. McDonald and S. R. E. Turner (eds.), American Mathematical Society, Providence, RI, 2000, 49–71. MR1788708

[35] P. Yang. Least controls for a class of constrained linear stochastic systems. *Mathematics of Operations Research*, **10** (1993), 275–291. MR1250119

V. Pesic
XR Trading
550 West Jackson Blvd, Suite 1000
Chicago, IL 60661, USA
E-mail: vlad.pesic@xrtrading.com

R. J. Williams
Department of Mathematics
University of California, San Diego
9500 Gilman Drive
La Jolla, CA 92093-0112, USA
E-mail: williams@math.ucsd.edu
URL: http://www.math.ucsd.edu/∼williams