Title
Two workload properties for Brownian networks

Permalink
https://escholarship.org/uc/item/75v7x61b

Journal
Queueing Systems, 45(3)

ISSN
0257-0130

Authors
Bramson, M
Williams, R J

Publication Date
2003-11-01

Peer reviewed

Two Workload Properties for Brownian Networks

M. Bramson[*]
School of Mathematics
University of Minnesota
Minneapolis MN 55455
bramson@math.umn.edu

R. J. Williams[†]
Department of Mathematics
University of California, San Diego
La Jolla CA 92093-0112
williams@math.ucsd.edu

June 26, 2003

Abstract

As one approach to dynamic scheduling problems for open stochastic processing networks, J. M. Harrison has proposed the use of formal heavy traffic approximations known as Brownian networks. A key step in this approach is a reduction in dimension of a Brownian network, due to Harrison and Van Mieghem [21], in which the "queue length" process is replaced by a "workload" process. In this paper, we establish two properties of these workload processes. Firstly, we derive a formula for the dimension of such processes. For a given Brownian network, this dimension is unique. However, there are infinitely many possible choices for the workload process. Harrison [16] has proposed a "canonical" choice, which reduces the possibilities to a finite number. Our second result provides sufficient conditions for this canonical choice to be valid and for it to yield a non-negative workload process. The assumptions and proofs for our results involve only first-order model parameters.

Short title: Two Workload Properties.

Keywords and phrases: Dynamic scheduling, Brownian network, Brownian control problem, equivalent workload formulation, Leontief matrix, open stochastic processing network, unitary network, open multiclass queueing network.

AMS Subject Classification (2000): Primary: 60K25, 68M20, 90B36, Secondary: 90C05.

[*] Research supported in part by NSF Grant DMS-9971248.

[†] Research supported in part by NSF Grant DMS-0071408, a John Simon Guggenheim Fellowship, and a gift from the David and Holly Mendel Fund. During part of the period in which this research was conducted, R. J. Williams was supported by the Operations, Information and Technology Program of the Graduate School of Business, Stanford University.

1 Introduction

Open stochastic processing networks are used as models for complex manufacturing, telecommunications and computer systems (see e.g., [5, 40]). Some of these networks allow for flexible scheduling through dynamic (that is, state-dependent) alternate routing and sequencing, hereafter collectively referred to as dynamic scheduling. Usually, these networks cannot be analyzed exactly, and a challenging problem is to design dynamic scheduling policies that are simple to implement and whose performance is "good" in an appropriate sense.

In addressing this problem, it is natural to consider more tractable approximate models. Some authors have used fluid models, which are (formal) first-order (or law of large number) approximations to the original open stochastic processing networks. Asymptotic regimes in which these models are relevant include those of large initial queue lengths (see [31, 35] and references therein) and large numbers of buffers or servers (see [30] and references therein). Frequently, these models are used for analyzing "subcritical" systems where the load placed on the system is less than the capacity of the system to process that load.

While of considerable use, especially in showing stabilizability or controllability of an open stochastic processing network, fluid models generally capture only average statistics of the networks. On the other hand, diffusion models, which are (formal) second-order approximations, capture the effects of stochastic variability. Diffusion models are relevant in the heavy traffic (or critically loaded) regime where the load placed on the system is approximately balanced by the system's ability to process that load. In fact, Kelly and Laws [23] have argued that in this regime "important features of good control policies are displayed in sharpest relief".

In a series of successively more general papers, J. M. Harrison et al. [13, 14, 21, 16] have proposed and developed various aspects of a scheme for using Brownian networks (formal heavy traffic diffusion approximations) and associated control problems, called Brownian control problems (BCPs), as a tool for finding and analyzing the performance of "good" control policies for open stochastic processing networks. The steps involved in applying this approach may be summarized as follows.

(I) Formulate an open stochastic processing network control problem.

(II) Formulate a notion of heavy traffic. (There is no conventional notion of heavy traffic, since a general network model should allow alternate routing, and the nominal or average load on a server may then depend on the chosen control policy.)

(III) Formulate a Brownian control problem (BCP) as a formal heavy traffic approximation for the open

stochastic processing network control problem. Reduce the BCP to an equivalent workload formulation (EWF).
(IV) Analyze the BCP (or EWF) and "interpret" its solution by giving a proposed control policy for the original network.
(V) Investigate the performance of the policy proposed in (IV). In particular, determine whether it is asymptotically optimal (in the heavy traffic limit) and whether it achieves the same cost as the solution of the BCP.

A key element in this approach occurs in step (III), where the dimension of the Brownian network, on which the BCP is based, is frequently reduced, or at least not increased, by replacing the "queue length" process by a "workload" process. This reduction, which is due to Harrison and Van Mieghem [21], is the counterpart for controlled networks of the *state-space collapse* that has played a fundamental role in analyzing the performance of open multiclass queueing networks with fixed scheduling policies (cf., [7, 38]). Furthermore, in networks with alternate routing capabilities, a buffer may be served by more than one server and pooling of server resources may be possible. This feature may result in a reduction in dimension considerably beyond that which is achievable in Brownian networks associated with open multiclass queueing networks, see e.g., [15, 20, 39, 2, 3, 37, 33, 1]. It is a natural problem to determine conditions under which the dimension reduction is significant and to obtain other interesting properties of workload processes. This paper is a contribution in that direction.

Here, we adopt the fairly general modeling framework for heavily loaded open stochastic processing networks that is used in Harrison's canonical workload paper [16], although with some modifications of notation. (Harrison [18, 19] has recently proposed an even more general model framework with a broader notion of heavy traffic than that in [16]. Since relevant supporting equivalent workload theory has not yet been developed for the setting in [18, 19], we have chosen to focus on the situation of [16].) Our paper establishes two main results on workload processes for Brownian networks associated with heavily loaded open stochastic processing networks. Firstly, we derive a formula for the dimension of any workload process (see Theorem 6.1). This result is general, requiring no additional assumptions beyond those of an open stochastic processing network in heavy traffic. Corollary 6.2 to Theorem 6.1 yields the simple workload dimension formula (1) (see below), under a mild assumption. For a given Brownian network, according to the definition of Harrison and Van Mieghem [21], there are infinitely many possible choices for a workload process, although they all have the same dimension. In [16], Harrison proposed a "canonical" choice, which reduces the possibilities to a finite number. In general, for this choice to be valid, an additional condition is required beyond the assumptions specified in [16]. Our second result provides sufficient conditions for this canonical choice to be valid and to yield a non-negative workload process (see Theorem 7.3). The assumptions and proofs for our results involve only first-order model parameters for an open stochastic processing network.

Our original intent in Theorem 7.3 was to establish sufficient conditions for the workload process to be non-negative. In the course of our investigations, we discovered the need for an additional condition (beyond the conditions given in [16]) in order for Harrison's canonical choice of workload to be valid; it turned out that our conditions for non-negativity of the workload process also guaranteed validity of this choice. Since the completion of our work, Harrison [17] has submitted a correctional note to [16] in which he specifies an additional assumption, which, together with his other assumptions, implies that his canonical choice of workload process is valid. The conditions of our Theorem 7.3 imply that this additional assumption of Harrison [17] holds (see the Remark following the proof of Theorem 7.3). While our conditions in Theorem 7.3 are stronger than the additional condition of [17], our conditions also imply that the workload process is non-negative.

Although Harrison's model involves stochastic processes for describing an open stochastic processing network, the assumptions and proofs for our results only use structural data for the network and average statistics for the stochastic processes. To emphasize this, we will introduce a minimum of notation and structure. In particular, we will not review all of the details associated with steps (I)–(V) above, but rather refer the reader to the papers of Harrison et al. [13, 14, 21, 16]. To compensate for not introducing all of the stochastic structure associated with Harrison's approach, we will provide intuitive explanations and illustrative examples whenever possible. Nevertheless, for a full appreciation of the import of the results developed here, the reader will want at least to consult the work [16].

1.1 Organization of the Paper

Sections 2, 4 and 5 introduce the part of Harrison's [16] structure that is necessary for the statements and proofs of our main results, which are presented in Sections 6 and 7. Section 3 contains examples that are used to illustrate our results. Besides being a necessary prelude for our results, the material in Sections 2–5 provides a succinct introduction to some essential aspects of parts (I)–(III) of the scheme for approximating controlled stochastic processing networks. Although much of this material could be gleaned by careful study of the works [13, 16, 21, 20, 22], the presentation given here is intended to provide a helpful introduction to those works. As a guide to the reader, we now elaborate on the content of Sections 2–7.

In Section 2, we introduce the network structure and first-order (or average rate) parameters associated with Harrison's model of an open stochastic processing network. Since we do not introduce the primitive stochastic processes required for a full description of an open stochastic processing network, we motivate the first-order parameters by interpreting them as average statistics associated with such stochastic processes. An important feature of Harrison's model is that scheduling control is exercised through allocations of effort to various processing activities (also called activities), where a processing activity depletes the supplies in some buffers and augments those in others, with each activity utilizing the resources of a given set of servers.

To help orient the reader, in Section 3 we specify the network structure and first-order parameters that arise when Harrison's framework is used to describe four examples of open stochastic processing networks. These examples are also used later in the paper to illustrate our results. The first three examples discussed fall into the category of what we call *unitary networks* (see Section 3 for the definition). This category is defined for the first time in this paper and it includes, as special cases, open multiclass queueing networks and parallel server systems. Unitary networks with some mild non-degeneracy assumptions provide applications for our Theorem 7.3 on non-negativity of canonical workload (see Corollary 7.4). The fourth example, of a fork-join network, illustrates some of the features of Harrison's setup that are not encompassed by unitary networks.

In Section 4, we recall the notion of heavy traffic as defined by Harrison [16] in terms of a linear program. We also give an equivalent characterization of this condition which is sometimes useful. The heavy traffic assumption requires that there be a unique solution of the linear program. This solution determines average levels of effort to be allocated to the processing activities. An activity that has a strictly positive allocation in this sense is said to be "basic".

In Section 5, we briefly summarize the definition of a workload process as introduced by Harrison and Van Mieghem [21]. A workload process is obtained by applying a certain linear transformation to the "queue length" process appearing in a Brownian network. For a given Brownian network, this linear transformation is not uniquely determined, however, the dimension of its range is unique. In Sections 6 and 7, we state and prove two properties of workload processes.

In Theorem 6.1 of Section 6, we derive a formula for the dimension $\mathbf{L}$ of any workload process. This implies that, if each basic activity is performed by no more than one server, then

$$\mathbf{L} = \mathbf{I} + \mathbf{K} - \mathbf{B}, \tag{1}$$

where $\mathbf{I}$ is the number of buffers, $\mathbf{K}$ is the number of servers, and $\mathbf{B}$ is the number of basic activities for the open stochastic processing network (see Corollary 6.2).

In [16], Harrison proposed a "canonical" choice of the linear transformation (or workload matrix) used to define the workload process. Although Harrison's procedure does not always yield a unique workload matrix, it results in only finitely many possibilities. In fact, using his procedure, the rows of a canonical workload matrix are constructed from extremal optimal solutions of the dual to the linear program used to define heavy traffic. In Section 7, we give an example which shows that, in general, an additional condition, beyond the assumptions specified in [16], is needed for this canonical choice to be valid. Then, in Theorem 7.3 we give some sufficient conditions for Harrison's procedure to be valid and for the resulting workload process to be non-negative. As a key step in the proof, we establish sufficient conditions for all optimal solutions of the dual program to be non-negative (see Lemma 7.2).

The following notation is used throughout the paper. The n-dimensional Euclidean space is denoted by $\mathbb{R}^n$, $n \geq 1$. All vectors are taken as column vectors unless specifically indicated otherwise. All vector inequalities are to be interpreted componentwise. For a vector β, $\text{diag}(\beta)$ denotes the diagonal matrix with the entries of β on its diagonal. The transpose of a vector x or matrix D is indicated by x' or D', respectively, and the inverse of a non-singular square matrix D is denoted by D^{-1}.

2 Open Processing Networks — First-Order Data

Loosely speaking, an open processing network is a system that takes inputs of material of various kinds and uses various processing resources (hereafter called servers) to produce outputs of material of various (possibly different) kinds. Here, material is used as a generic substitute for a variety of entities that a system might process such as jobs, customers, packets, commodities, etc. In fact, material need not be discrete; for example, one may have continuous flows of material. Typically, there are constraints on the amount of material that a given server can process in a given time period. In addition, material may be processed by several servers, may be split up, or combined with other kinds of material, before a final output is produced. Control is exerted through allocations of effort for the processing of one or more kinds of material.

Deterministic (or average) models for describing such processing networks have a long history in economics and operations research. For example, the book "Activity Analysis of Production and Allocation", edited by T. C. Koopmans [25], provides an excellent summary of the early stages of development of such models. A prominent role is played there by the notion of a *processing activity,* which consumes certain kinds of material, produces certain (possibly different) kinds of material, and uses certain servers in the process. In a sense, Harrison's [16] model of an open stochastic processing network (which evolved from his earlier work [13] on sequencing problems for multiclass queueing networks) is a stochastic analogue of dynamic deterministic production models such as those first considered by Dantzig [11]. (See Harrison [19] for more on this connection.) An especially desirable feature of Harrison's model is that it is broad enough to include several familiar categories of processing networks as special cases, including open multiclass queueing networks (with sequencing control), processing facilities with alternate routing capabilities, and manufacturing plants in which multiple components may be combined to produce new components or in which components may be split up so that different parts undergo different types of processing and processed parts may be combined later on to produce a finished product (so-called fork-join networks).

We now summarize the network structure and first-order model parameters that feature in Harrison's description of an open stochastic processing network in heavy traffic. (In fact, Harrison's setup allows one to consider a sequence of networks in which the first-order parameters approach those satisfying a heavy traffic condition. Our results still apply to that setting, in which case our first-order data would be the limiting first-order data for the sequence of networks. To simplify the exposition, especially with regard to giving intuitive explanations of the meanings of various parameters, we have chosen to think of our first-order data as being associated with a single open stochastic processing network in heavy traffic, rather than as the limiting data for a sequence of networks approaching heavy traffic. As mentioned above, this does not affect the applicability of our results to the sequence of networks setting.) The first-order parameters capture the average behavior of stochastic processes appearing in Harrison's model and do not include the effects of statistical variability. However, these are the only parameters that will be needed for the derivation of our results. In fact, these are the same parameters that are used in Harrison's description of a "static planning problem" (see Definition 4.1 in Section 4 below), which is a static deterministic optimization problem involving average activity rates. Some examples of open stochastic processing networks and the associated first order data are given in Section 3. For assimilating the following, the reader is likely to find it helpful to consult those examples from time-to-time.

In Harrison's model of an open stochastic processing network, there are $\mathbf{I}$ infinite capacity *buffers* for holding material (one buffer for each class of material) awaiting processing, $\mathbf{K}$ (non-identical) *servers* for processing material, and $\mathbf{J}$ *processing activities.* Each of $\mathbf{I}$, $\mathbf{J}$, and $\mathbf{K}$ is assumed to be a strictly positive integer. Henceforth, processing activities will be referred to as *activities.* For each buffer i, material of class i may arrive to buffer i from outside the system and/or from the internal processing of material by the activities. We now introduce three parameters α, R and A, which describe average rates associated with the external arrival and internal processing of material in an open stochastic processing network.

A non-negative $\mathbf{I}$-dimensional vector α describes the average rates at which material arrives from outside the system. Specifically, for each class i, the average rate at which material of class i arrives to buffer i from outside the system is assumed to be given by the i^{th} coordinate α_i of the non-negative vector α. It is further assumed that $\alpha_i > 0$ for at least one i. This last assumption is consistent with the notion of an *open* stochastic processing network.

An $\mathbf{I} \times \mathbf{J}$ matrix R (with no a priori constraints), called the *input-output matrix,* specifies average rates at which activities can consume certain classes of material as inputs and produce the same or other classes of material as outputs. The interpretation of R is as follows. For $i = 1, \ldots, \mathbf{I}$ and $j = 1, \ldots, \mathbf{J}$, R_{ij} is the average amount of class i material consumed per unit of activity j, with a negative value being interpreted to mean that activity j is a net producer of material of class i. Since we have not imposed constraints on the entries in R, this allows for a very general structure. For example, for a given activity j, more than one class of material may be used as a net input (i.e., $R_{ij} > 0$ for more than one i), or may be generated as a net output (i.e., $R_{ij} < 0$ for more than one i). Note that we do not require a priori that a given activity j consume anything (i.e., we may have $R_{ij} \leq 0$ for all i), nor do we require that it produce anything (i.e., we may have $R_{ij} \geq 0$ for all i).

A non-negative $\mathbf{K} \times \mathbf{J}$ matrix A, called the *capacity consumption* matrix, specifies average rates at which server capacity is utilized by each of the activities. For $k = 1, \ldots, \mathbf{K}$ and $j = 1, \ldots, \mathbf{J}$, the entry A_{kj} of A can be interpreted as the average amount of server k's capacity that is consumed per unit of activity j.

An activity may use zero, one or more servers to perform the associated processing of material. Activities that do not use any server resources are implicitly allowed in Harrison's [16] setup. Such activities enable material to be consumed or produced without using the resources of any servers. One might think of such activities as discarding excess material from some buffers and importing material to others without using any processing resources. Mathematically, the inclusion of such activities amounts to the fact that the matrix A may have columns that contain all zeros. For an example of an application involving such an activity, see [19].

In applications where each activity is processed by exactly one server, i.e., where there is exactly one non-zero entry in each column of the matrix A, one can normalize and measure amounts of each activity j in units of capacity of the associated server k (where k is the unique value such that $A_{kj} \neq 0$). With this normalization, the matrix A contains only zeros and ones. However, in applications where some activity is processed either by no server or more than one server, such a normalization is not possible in general.

Servers have constraints on their processing abilities. As a matter of convention (or choice of units), it is assumed that each server has one unit of capacity available per unit of time. Control is exercised through the cumulative amounts of each activity that are undertaken up to time t for each $t \geq 0$. Idling of servers is allowed with such control.

The triple (R, A, α), described above, specifies the first-order data for Harrison's model. (One sometimes regards server capacity as an additional first-order model parameter, but since this has been normalized as in the above paragraph, it is not a variable parameter in this setting.) We re-emphasize here that the first-order data specifies the average statistics of the primitive stochastic processes in Harrison's model. We have intentionally not introduced information about the statistical distributions of the stochastic processes, beyond their first moments, as our results and proofs do not need this information. A control problem associated with the first-order data, called the *static planning problem,* is specified in Section 4 (see Definition 4.1), and this is used to define the notion of heavy traffic.

3 Examples

In this section, we specify the first-order data that arise from some examples of open stochastic processing networks. As in Section 2, in describing these examples, we do not specify the primitive stochastic processes for the networks, but rather just their average statistics which yield the first order data.

The first three examples below are all instances of what we call *unitary networks.* As far as we know, this term, which was suggested to us by J. M. Harrison, has not appeared previously in print. We define a *unitary network* to be an open stochastic processing network (in the sense of Harrison [16]) whose first-order data is unitary according to the following definition. (Those seeking a more detailed description of a unitary network may consult our prior work [8] which provides a detailed stochastic model for a class of unitary networks, where each activity consumes material from exactly one buffer, the processing of each activity is performed by exactly one server, and there is at most one activity associated with each buffer-server pair.)

Definition 3.1 (Unitary Data). *The first-order data (R, A, α) are unitary if the non-negative matrix A has at most one non-zero entry in each column and $R = (C - P')\text{diag}(\beta)$, where C is an $\mathbf{I} \times \mathbf{J}$ matrix of zeros and ones that contains at most one non-zero entry in each column, P' is the transpose of a non-negative $\mathbf{J} \times \mathbf{I}$ matrix P with all row sums less than or equal to one, and β is a $\mathbf{J}$-dimensional vector with strictly positive coordinates.*

An interpretation of unitary data in terms of the average dynamics of an open stochastic processing network is as follows. Each activity is associated with at most one buffer, from which it consumes material. This correspondence is prescribed by the matrix C, with activity j consuming material from buffer i if and only if $C_{ij} = 1$. Activity j can process material at an average rate of β_j units of material per unit of processing activity. An activity j satisfying $C_{ij} = 1$ for some i produces on average as many units of material as it consumes, provided material produced for immediate output is taken into account. An activity j satisfying $C_{ij} = 0$ for all i may produce material without needing to consume any material. The matrix P indicates, on average, what fraction of the material produced by a given activity is routed to each of the buffers for further processing. More precisely, on average, the fraction of material produced by activity j that is routed to buffer i is P_{ji}, and the fraction of material produced by activity j that immediately exits the system is $1 - \sum_{i=1}^{\mathbf{I}} P_{ji}$. (One may equally well interpret the entries of P as probabilities of routing material produced by a given activity to buffers in the network.) Each activity is performed by at most one server and this correspondence is described by the matrix A, with activity j being performed by server k if and only if $A_{kj} > 0$. An activity j that does not require the use of any server, i.e., such that $A_{kj} = 0$ for all k, can be thought of as an "outsourcing" activity that uses external resources for its execution.

Remark. The idea behind the term unitary is that one may think of material in such networks as consisting of indivisible units that cannot be split up or combined with other classes of material during processing. Note that our definition does allow material to be created from nothing (when a column of C contains all zeros) or to be processed without needing the use of any network resource (when a column of A contains all zeros).

In the first three examples given below, which are all examples of unitary networks, the matrices C and A do not have any columns containing all zeros. In particular, since each column of A has exactly one strictly positive entry, by measuring allocations of effort to activities as allocations of server capacity, we can assume that the positive entries are all ones and hence A contains only zeros and ones. In all three examples, server capacity is assumed to be measured in units of server time.

3.1 Open Multiclass Queueing Networks

An open multiclass queueing network is a unitary network with the following additional structure. There is a one-to-one correspondence between activities and buffers, and so the matrix C is the $\mathbf{I} \times \mathbf{I}$ identity matrix. It is also assumed that each activity is processed by exactly one server, which implies that each buffer is served by exactly one server. Thus, $\mathbf{I} = \mathbf{J} \geq \mathbf{K}$. (A special case is an open single class queueing network, where there is only one activity per server; then $\mathbf{I} = \mathbf{J} = \mathbf{K}$.) An open multiclass queueing network has an R matrix of the form $R = (I - P')\text{diag}(\beta)$, where β is a strictly positive $\mathbf{I}$-dimensional vector, I is the $\mathbf{I} \times \mathbf{I}$ identity matrix, and P' is the transpose of a non-negative matrix P with all row sums less than or equal to one. In addition, P is assumed to have spectral radius strictly less than one. This last property is equivalent to $P^n \to 0$ as $n \to \infty$, i.e., P can be regarded as a transition probability matrix for a transient Markov chain on $\mathbf{I}$ states; this property corresponds to the "open" aspect of the multiclass queueing network. The matrix A is a $\mathbf{K} \times \mathbf{I}$ matrix of zeros and ones with $A_{ki} = 1$ when server k performs the processing of activity i. Thus, the number 1 occurs exactly once in each column of A, but may appear several times in each row, since a server may process material from more than one buffer.

Open multiclass queueing networks are frequently used to describe the passage of customers or jobs through a network of servers or stations. In this case, the term material translates to customers or jobs, processing of material corresponds to service of a customer or job, and customers or jobs change class as they are routed from one buffer to another. The multiclass aspect is reflected in the fact that a server may be able to process customers or jobs from more than one buffer, i.e., of more than one class. Usually, P is interpreted as a probabilistic routing matrix such that P_{ji} is the probability that a customer or job from buffer j, after being processed by the server associated with that buffer, is routed next to buffer i.

A specific example of an open multiclass queueing network is depicted in Figure 1. This is the so-called criss-cross network. It has been studied by various authors, see e.g., [22, 10, 12, 34, 29, 32]. This network is arguably the first example where the Brownian network methodology was successfully applied to obtain a "good" control policy, cf. [22]. For this example, $\alpha = (\alpha_1, \alpha_2, 0)$,

$$R = \begin{bmatrix} \beta_1 & 0 & 0 \\ 0 & \beta_2 & 0 \\ -\beta_1 & 0 & \beta_3 \end{bmatrix}, \qquad A = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

3.2 Parallel Server Systems

The simplest unitary networks in which a buffer can be served by more than one server are the parallel server systems introduced in the heavy traffic context by Harrison and López [20]; see [1, 2, 3, 33, 36, 37, 39] for subsequent related work. (Other authors have studied similar systems where the model setup is slightly different; in those systems, the scheduling control, also called alternate routing control, is exercised before incoming material enters a buffer. Kelly and Laws [23] have summarized the literature on such systems up through 1993. For more recent work, see Kushner and Chen [28] and the references therein.) A primary feature of the parallel server model of Harrison and López [20] is that material is processed only once before it exits the system. In this model, each activity processes jobs from a unique buffer

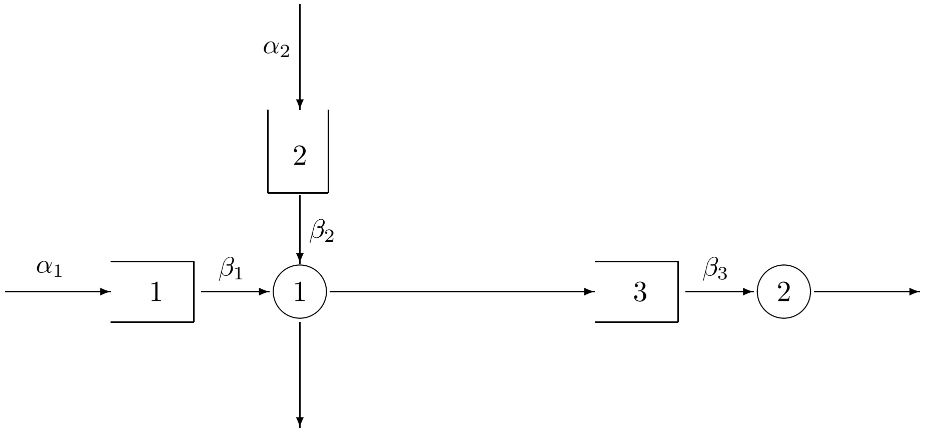

Figure 1: Criss-cross network [22]

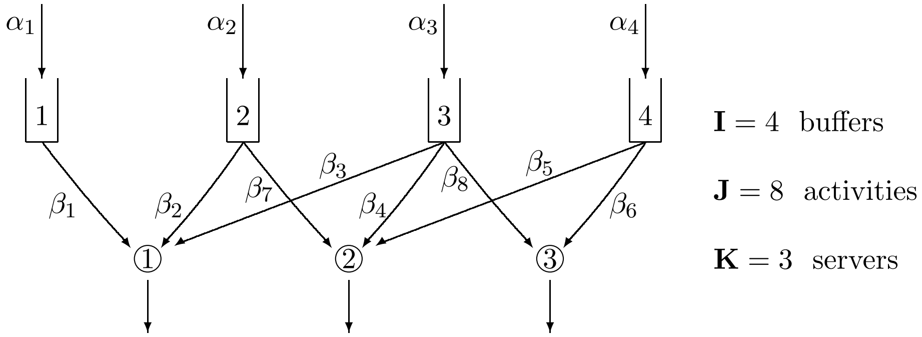

Figure 2: Example of a parallel server system [20]

and its processing is performed by a unique server. There is at most one activity associated with each buffer-server pair, which implies that there are a maximum of $\mathbf{I} \cdot \mathbf{K}$ possible activities. Parallel server systems are thus unitary networks with R matrices of the form $R = C\text{diag}(\beta)$, where, for each j, the j^{th} column of C has exactly one non-zero entry (with value equal to one) which occurs in the $i(j)^{th}$ row corresponding to the buffer $i(j)$ served by that activity, and β_j is the average rate at which activity j can consume material from buffer $i(j)$. The non-negative matrix A has exactly one strictly positive entry in each column (corresponding to the server that performs that activity). More precisely, for each j, if $k(j)$ is the server that performs activity j, then $A_{kj} = 1$ if $k = k(j)$ and $A_{kj} = 0$ if $k \neq k(j)$. It is further assumed that the rate of arrival of material to each buffer from outside the system is strictly positive, i.e., $\alpha_i > 0$ for $i = 1, \ldots, \mathbf{I}$. An example of a parallel server system is pictured in Figure 2. This example is taken from [20]. Here,

$$R = \begin{bmatrix} \beta_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \beta_2 & 0 & 0 & 0 & 0 & \beta_7 & 0 \\ 0 & 0 & \beta_3 & \beta_4 & 0 & 0 & 0 & \beta_8 \\ 0 & 0 & 0 & 0 & \beta_5 & \beta_6 & 0 & 0 \end{bmatrix}, \qquad A = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

3.3 A Job Shop

An example of a unitary network, that combines features of the two previous cases, is pictured in Figure 3. Prominent features of this example are that a server can serve more than one buffer, a buffer can be served by more than one server, and material may be routed to another buffer after processing at a server. (In Figure 3, the circle with a cross inside it represents the routing of material. The details of the routing mechanism are not apparent from the figure; they are described more fully below.) This example is taken from a paper by S. Kumar [27] where motivation is provided by a job shop application. For this network, $\alpha = (\alpha_1, \alpha_2, 0)$. It is assumed that material from buffer 2, after being processed at either server 1 or 2, is routed to buffer 3 with probability $\frac{1}{2}$, or exits the system with probability $\frac{1}{2}$. It is also assumed that material from all other buffers (besides buffer 2), exits the system after processing at one of the servers. Assuming that activity j can process material from the associated buffer at a rate of β_j units of material per unit of processing activity, we have

$$R = \begin{bmatrix} \beta_1 & 0 & 0 & \beta_4 & 0 & 0 \\ 0 & \beta_2 & 0 & 0 & \beta_5 & 0 \\ 0 & -\frac{1}{2}\beta_2 & \beta_3 & 0 & -\frac{1}{2}\beta_5 & \beta_6 \end{bmatrix}, \qquad A = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

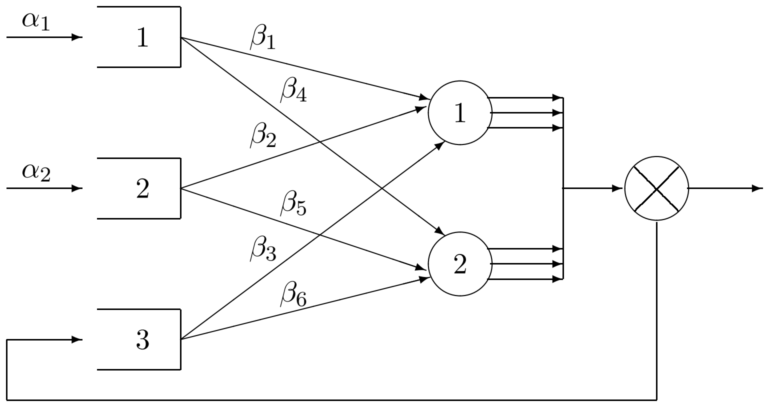

Figure 3: A job shop model [27]

Note that the routing protocol described above is based on the buffer from which material was consumed. However, Harrison's setup also allows routing based on the activity that processed material. For example, if one changes the model so that only material processed by activity 2 is to be routed to buffer 3 (still with probability $\frac{1}{2}$), then the matrix R needs to be modified by changing the entry R_{35} to a zero. In either case, one might view the routing protocol as arising from the need to "rework" material processed by certain activities, with probability $\frac{1}{2}$.

3.4 A Fork-Join Network

Figure 4 depicts an example of a so-called fork-join network. Such networks arise in modeling systems where material may be split up, or combined with other material, at various stages of processing. Common applications occur in manufacturing (see e.g., [9]); the term disassembly-assembly is sometimes used instead of fork-join in this context. (This model does not require synchronization, i.e., a joining activity need not (re)combine units of material that were formed by a forking activity at a previous stage of processing.) The network pictured in Figure 4 has three activities. The first consumes material from buffer 1 and requires simultaneous processing by servers 1 and 2. For concreteness, we think of this as splitting material into two parts with each part being processed by one of the servers. This is the forking aspect of the network. This occurs, for example, in the manufacture of clothing, where cloth is divided into pieces for separate processing and processed pieces are later assembled into a garment. We assume that activity 1 consumes material from buffer 1 at an average rate of β_1 units of material per unit of processing activity, and that this activity produces material for buffer 3 at an average rate of γ_1 units of material per unit of processing activity and for buffer 4 at an average rate of γ_2 units of material per unit of processing activity. The second activity consumes material from buffer 2 at a rate of

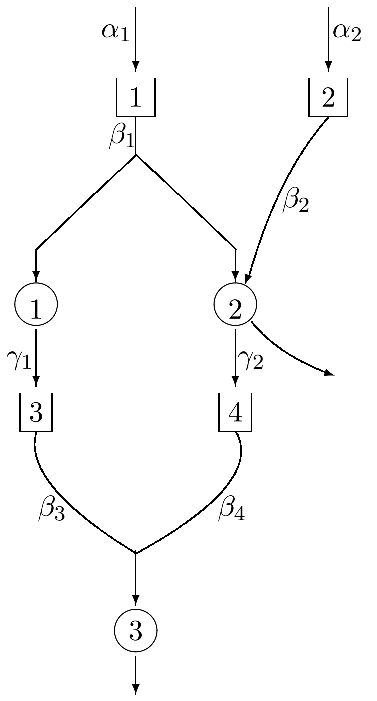

Figure 4: A fork-join network

β_2 units of material per unit of processing activity. This activity competes with activity 1 for service at server 2. Material processed by this activity exits the system after service at server 2. The third activity simultaneously consumes material from buffers 3 and 4 for processing by server 3. We interpret this as joining together two classes of material for processing. We assume that activity 3 consumes material from buffer j at an average rate of β_j units of material per unit of processing activity for $j = 3, 4$. Each of the β_j, $j = 1, \ldots, 4$, and γ_j, $j = 1, 2$, is assumed to be strictly positive. We assume that activity 1 utilizes the capacity of server 1 at a rate of one unit of capacity per unit of processing activity and of server 2 at a rate of $\frac{1}{2}$ unit of capacity per unit of processing activity. Activities 2 and 3 are assumed to utilize the capacity of servers 2 and 3, respectively, at rates of one unit of capacity per unit of processing activity. Then, we have

$$R = \begin{bmatrix} \beta_1 & 0 & 0 \\ 0 & \beta_2 & 0 \\ -\gamma_1 & 0 & \beta_3 \\ -\gamma_2 & 0 & \beta_4 \end{bmatrix}, \qquad A = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

For this example, $\alpha = (\alpha_1, \alpha_2, 0, 0)$. It is straightforward to see that this fork-join network is not a unitary network.

4 Heavy Traffic and a Linear Program

Harrison [16] has formulated the following deterministic static planning problem in terms of the first-order data (R, A, α) described in Section 2. Here $\mathbb{R}_+ = [0, \infty)$, $\mathbb{R}_+^{\mathbf{I}}$ is the non-negative $\mathbf{I}$-dimensional orthant and $\mathbf{1}$ is the $\mathbf{K}$-dimensional vector of all ones.

Definition 4.1 (Static Planning Problem) *Choose $\rho \in \mathbb{R}_+$ and $x \in \mathbb{R}_+^{\mathbf{J}}$ so that (ρ, x) is an optimal solution to the following linear program:*

$$\textit{minimize } \rho \quad \textit{subject to} \quad Rx = \alpha, \quad Ax \le \rho\mathbf{1} \quad \textit{and} \quad x \ge 0. \tag{LP}$$

Remark. The label (LP) reflects the fact that the static planning problem is a linear program. Later on, in Section 7, we consider the dual to this linear program.

For $j = 1, \ldots, \mathbf{J}$, the variable x_j in the static planning problem may be interpreted as the average number of units of activity j to be undertaken per unit time. The condition $Rx = \alpha$ implies that these allocations precisely balance the system on average, in the sense that the incoming rate of arrivals α is exactly balanced by the rate at which the system can process that load under the allocations given by x. If $Ax \le \mathbf{1}$, then the allocations do not exceed the capacity of the servers to perform this processing. Vectors x such that $Rx = \alpha$, $Ax \le \mathbf{1}$ and $x \ge 0$ are said to be *allowed allocations.* A solution (ρ, x) of the static planning problem with $\rho \le 1$ yields an allowed allocation x that also achieves the minimum possible value ρ for the maximum of the associated utilization rates of the $\mathbf{K}$ servers.

The following notion of heavy traffic was proposed by Harrison in [16].

Definition 4.2 (Heavy Traffic) *The first-order data (R, A, α) satisfy the heavy traffic condition if and only if*

(a) there is a unique optimal solution (ρ^*, x^*) of the linear program (LP), and

(b) this unique solution satisfies $\rho^* = 1$ and $Ax^* = \mathbf{1}$.

The following lemma provides an equivalent characterization of the heavy traffic assumption that is sometimes useful. Its proof is straightforward, but we include it here for completeness.

Lemma 4.3 *The first-order data (R, A, α) satisfy the heavy traffic condition if and only if the following two conditions hold:*

(i) there is a unique $\mathbf{J}$-dimensional vector x^* such that

$$Rx^* = \alpha, \qquad Ax^* \le \mathbf{1}, \qquad x^* \ge 0, \tag{2}$$

(ii) the unique solution x^* satisfies $Ax^* = \mathbf{1}$.

Proof. For the "only if" part of the lemma, suppose that the first-order data satisfy the heavy traffic condition. Then, for the optimal solution (ρ^*, x^*) of (LP), x^* satisfies (2). Suppose that $x^\dagger$ is another solution of (2). Then, $(Ax^\dagger)_k = 1$ for at least one k, for otherwise $Ax^\dagger \leq \rho\mathbf{1}$ for some $\rho < 1$, which violates the fact that the optimal value of the linear program is $\rho^* = 1$ (by (b) of the heavy traffic definition). It follows that $(1, x^\dagger)$ is also an optimal solution of (LP), since it is feasible and achieves the optimal value of the (LP). But then, by the uniqueness of the optimal solution of (LP) (by (a) of the heavy traffic definition), we must have $x^\dagger = x^*$. Thus, (i) of the lemma holds. Also, (ii) holds by property (b) of the heavy traffic definition.

For the "if" part of the lemma, suppose that (i) and (ii) of the lemma hold. Then, $(1, x^*)$ is a feasible solution of (LP) and there is no other feasible solution (ρ, x) with $\rho \leq 1$. Hence, $(1, x^*)$ is the unique optimal solution for (LP). Property (b) of the heavy traffic definition follows using (ii). □

The following assumption will be made implicitly throughout the remainder of this paper.

Assumption HT. *The first-order data (R, A, α) satisfy the heavy traffic condition.*

The activities j for which $x_j^* > 0$ are called *basic* activities and those for which $x_j^* = 0$ are called *non-basic* activities. Let $\mathbf{B}$ denote the number of basic activities. This quantity is of fundamental importance and figures prominently in the workload dimension formula given in Section 6 below. (Non-basic activities only come into play at the second order (or diffusion) level of the scheme (I)–(V).) Without loss of generality, we assume that the first $\mathbf{B}$ activities are the basic ones and the last $\mathbf{J} - \mathbf{B}$ are the non-basic ones. For later reference, we partition the matrices R and A:

$$R = [H \quad M] \qquad \text{and} \qquad A = [B \quad N] \tag{3}$$

where H, M, B, and N are $\mathbf{I} \times \mathbf{B}$, $\mathbf{I} \times (\mathbf{J} - \mathbf{B})$, $\mathbf{K} \times \mathbf{B}$ and $\mathbf{K} \times (\mathbf{J} - \mathbf{B})$ matrices, respectively.

Example. For open multiclass queueing networks, $R = (I - P')\text{diag}(\beta)$ is invertible and has a non-negative inverse, since β has strictly positive entries and P is non-negative with spectral radius strictly less than one. (See Lemma 6.2.1 of [4], for example.) In fact,

$$R^{-1} = \text{diag}(\beta)^{-1}(I - P')^{-1} = \text{diag}(\beta)^{-1}(I + P' + (P')^2 + \ldots),$$

where the power series in P' converges. If, for $\lambda = (I - P')^{-1}\alpha$, we have

$$\sum_{j:A_{kj}>0} \frac{\lambda_j}{\beta_j} = 1 \quad \text{for} \quad k = 1, \ldots, \mathbf{K}, \tag{4}$$

then Assumption HT holds with $x_j^* = \lambda_j/\beta_j$, $j = 1, \ldots, \mathbf{I}$. The condition (4) is the usual heavy traffic assumption for open multiclass queueing networks.

5 Workload Definition

Harrison [16] has described a procedure for formally approximating an open stochastic processing network, whose first-order data (R, A, α) satisfy Assumption HT, by a Brownian network. In [21], Harrison and Van Mieghem showed that a Brownian network can be reduced to an equivalent model of typically lower dimension by replacing the "queue length" process by a "workload" process. In essence, this is obtained by ignoring certain "reversible directions" for the queue length process.

More precisely, let

$$\mathcal{N} = \{\delta \in \mathbb{R}^{\mathbf{I}} : \delta = Rx,\ Ax = 0,\ x_N = 0\}, \tag{5}$$

where the variable x ranges over $\mathbb{R}^{\mathbf{J}}$, and x_N denotes the vector consisting of the components of x indexed by the non-basic activities $j = \mathbf{B} + 1, \ldots, \mathbf{J}$. Harrison and Van Mieghem [21] call $\mathcal{N}$ the set of *reversible displacements,* since, in a Brownian network, any two queue length vectors whose difference is an element of $\mathcal{N}$ are equivalent, because there is a control that can instantaneously exchange one of these queue length vectors for the other without incurring any idleness, and this control is reversible in the sense that its negative is a valid control that returns the queue length to its original state. (The condition that $x_N = 0$ in the definition of $\mathcal{N}$ is needed to ensure reversibility of the control, since in a Brownian network, controls associated with non-basic activities have to be non-increasing.)

Let $\mathcal{N}^{\perp}$ denote the orthogonal complement of $\mathcal{N}$ in $\mathbb{R}^{\mathbf{I}}$ and let $\mathbf{L}$ denote the dimension of $\mathcal{N}^{\perp}$. Harrison and Van Mieghem [21] define a *workload process* as follows. Fix a set of vectors in $\mathbb{R}^{\mathbf{I}}$ that forms a basis for $\mathcal{N}^{\perp}$ and let Λ be an $\mathbf{L} \times \mathbf{I}$ matrix whose rows are given by these basis vectors. Then, a workload process is obtained by applying the linear transformation Λ to the $\mathbf{I}$-dimensional queue length process in the Brownian network. Since the rows of Λ are orthogonal to $\mathcal{N}$, this is the same as applying Λ to the projection of the queue length process onto $\mathcal{N}^{\perp}$. In general, there are many choices for a basis for $\mathcal{N}^{\perp}$ and the workload process is far from unique. On the other hand, since $\mathcal{N}^{\perp}$ is fixed, the dimension $\mathbf{L}$ of the workload process is unique. In the following sections, any matrix Λ of the form described above will be called a *workload matrix,* and $\mathbf{L}$ will be called the *workload dimension.*

Example. Recall that for open multiclass queueing networks, $R = (I - P')\text{diag}(\beta)$ is invertible with a non-negative inverse. If, in addition, all of the $\mathbf{I}$ activities are basic (i.e., $\mathbf{B} = \mathbf{I}$), we have

$$\mathcal{N} = \{\delta \in \mathbb{R}^{\mathbf{I}} : AR^{-1}\delta = 0\}.$$

Then, $\mathcal{N}^{\perp}$ is generated by the rows of $AR^{-1} = A\,\text{diag}(\beta)^{-1}(I - P')^{-1}$. Under Assumption HT, each server processes at least one basic activity, and it follows that A has full rank. Consequently, $\mathcal{N}^{\perp}$ has dimension

$\mathbf{K}$ and one may set $\Lambda = AR^{-1}$, which is non-negative. This choice of Λ defines the workload process that has played a fundamental role in diffusion approximations to open multiclass queueing networks, see, for example, [13, 38]. (This process, which is sometimes called the *total workload* process, should be distinguished from the *immediate workload* process, which is obtained by applying $A\,\mathrm{diag}(\beta)^{-1}$ to the queue length process. Potential confusion arises from the fact that some authors shorten the term immediate workload to workload.)

Motivated in part by the concreteness of the workload properties for open multiclass queueing networks, in the next two sections we establish a general workload dimension formula and provide sufficient conditions for a canonical choice of Λ (proposed by Harrison [16]) to be valid and to yield a non-negative workload matrix. The latter property is appealing as it justifies use of the term workload, which one naturally tends to think of as a non-negative quantity.

6 Workload Dimension

In this section, we derive a formula for the workload dimension $\mathbf{L}$. This result is general in that it requires no additional assumptions beyond the heavy traffic assumption (Assumption HT) and the structure of the first order model parameters associated with an open stochastic processing network. As a corollary, we show that the workload dimension formula has the form (1) when each basic activity is processed by at most one server. For the special case when the workload dimension is one and the first-order data (R, A, α) are derived from a parallel server system (see Section 3), the result of the corollary was established by Harrison and López [20].

Before stating the main result of this section, we remind the reader that we take as given first-order data (R, A, α) satisfying the assumptions of Sections 2 and 4. In particular, the heavy traffic assumption (Assumption HT) holds. Recall the definition of $\mathbf{B}$, which is the number of basic activities, the definition of B, which is the matrix formed by deleting the columns from A that correspond to non-basic activities, and the decomposition (3) for the matrices R and A.

Theorem 6.1 *Suppose that the* $\mathbf{K} \times \mathbf{B}$ *matrix* B *has rank* $\overline{\mathbf{K}}$*, for some* $\overline{\mathbf{K}} \leq \mathbf{K}$*. Then, the workload dimension* $\mathbf{L}$ *is given by*

$$\mathbf{L} = \mathbf{I} + \overline{\mathbf{K}} - \mathbf{B}. \qquad (6)$$

Proof. By definition, $\mathbf{L}$ is equal to the dimension of $\mathcal{N}^{\perp} \subset \mathbb{R}^{\mathbf{I}}$, so it suffices to show that the dimension of $\mathcal{N}$ is $\mathbf{B} - \overline{\mathbf{K}}$. First, consider

$$\mathcal{B} = \{x_B \in \mathbb{R}^{\mathbf{B}} : Bx_B = 0\}, \qquad (7)$$

where x_B denotes the vector consisting of the components of $x \in \mathbb{R}^{\mathbf{J}}$ indexed by $j = 1, \ldots, \mathbf{B}$. Since the $\mathbf{K} \times \mathbf{B}$ matrix B has rank $\overline{\mathbf{K}}$, it follows that $\overline{\mathbf{K}} \leq \mathbf{B}$ and the null space $\mathcal{B}$ of B has dimension $\mathbf{B} - \overline{\mathbf{K}}$.

Now,

$$\mathcal{N} = \{Rx : Ax = 0,\, x_N = 0\} = \{Hx_B : x_B \in \mathcal{B}\}. \tag{8}$$

We claim that the range $\mathcal{N}$ of H on $\mathcal{B}$ has the same dimension as $\mathcal{B}$. Together with the above paragraph, this will show that $\mathcal{N}$ has dimension $\mathbf{B} - \overline{\mathbf{K}}$, which will complete the proof. Since H is a linear map, to show that $\mathcal{N}$ has the same dimension as $\mathcal{B}$, it is enough to show that the null space of H in $\mathcal{B}$ is trivial. So, suppose that $Hx_B = 0$ for some $x_B \in \mathcal{B}$. We also have $Bx_B = 0$. By setting $x_N = 0$, we can extend x_B to a $\mathbf{J}$-dimensional vector x such that

$$Rx = 0, \quad Ax = 0. \tag{9}$$

Since x^*, from the optimal solution of (LP), has strictly positive entries for its basic components, one can choose $\epsilon > 0$ such that all of the components of $\tilde{x} \equiv x^* + \epsilon x$ are non-negative. It is easy to see that

$$R\tilde{x} = \alpha, \quad A\tilde{x} = \mathbf{1}. \tag{10}$$

Thus, $(1, \tilde{x})$ is another optimal solution of (LP). However, by Assumption HT, such a solution is unique, and so we must have $\tilde{x} = x^*$, $x = 0$ and $x_B = 0$. □

Corollary 6.2 *Suppose that each column of the non-negative matrix B contains at most one strictly positive element. Then, B has rank $\mathbf{K}$ and the workload dimension $\mathbf{L}$ satisfies (1).*

Remark. In Section 7 (see Definition 7.1), we define the notion of a Leontief matrix. We note here that, since B is non-negative, the assumption on B in the above corollary is equivalent to the assumption that B is Leontief.

Proof of Corollary 6.2. By Theorem 6.1, it suffices to show that the (row) rank of B is $\mathbf{K}$. For this, suppose that for some $z \in \mathbb{R}^{\mathbf{K}}$, $z'B = 0'$, where $0'$ is the $\mathbf{B}$-dimensional row vector of all zeros. We wish to show that $z = 0$.

For each server k, there must be at least one $j \in \{1, \ldots, \mathbf{B}\}$ such that $B_{kj} > 0$. This follows from $Bx^*_B = \mathbf{1}$, where B has non-negative entries and $x^*_B = (x^*_1, \ldots, x^*_{\mathbf{B}})'$ is the $\mathbf{B}$-dimensional vector consisting of the first $\mathbf{B}$ components of the non-negative vector x^* appearing in Assumption HT.

For each k, let $j(k)$ be a basic activity such that $B_{kj} > 0$ (if there is more than one candidate, just pick one). Since B is non-negative, by the hypothesis of the corollary, the $j(k)^{th}$ column of B has only one non-zero entry, and that occurs in the k^{th} row. By considering the $j(k)^{th}$ coordinate of $z'B$, which takes the value 0, it follows that $z_k = 0$. Since k was arbitrary, it follows that $z = 0$. $\square$

Remark. Dimitris Bertsimas (oral communication) has noted the following interpretation of (1) when the rows of R are linearly independent. In that case, the standard form of (LP) has a constraint matrix with $\mathbf{I}+\mathbf{K}$ linearly independent rows and $\mathbf{I}+\mathbf{K}-(\mathbf{B}+1)$ is the *degree of degeneracy* of the basic solution (ρ^*, x^*) of the linear program (cf., [6], page 59). The quantity in the right side of (1) is then the degree of degeneracy plus one.

Examples. A simple way to satisfy the hypothesis of Corollary 6.2 is to require that the matrix A have at most one strictly positive entry in each column. This assumption is automatically satisfied for unitary networks. Thus, the workload dimension formula (1) holds for the first three examples in Section 3. In particular, for an open multiclass queueing network in which every activity is basic, $\mathbf{B} = \mathbf{I}$ and the workload dimension is equal to the number of servers $\mathbf{K}$. For the (non-unitary) fork-join network of Section 3 with $\alpha_1 = 1, \alpha_2 = \frac{1}{2}$, $\beta_1 = \beta_2 = \beta_3 = \beta_4 = \gamma_1 = \gamma_2 = 1$, Assumption HT holds with $x^* = (1, \frac{1}{2}, 1)$. The hypothesis of Corollary 6.2 is not satisfied, because $B = A$ has two non-zero entries in its first column. However, the rank of B is $\mathbf{K}$ and so by Theorem 6.1, $\mathbf{L} = 4$. On the other hand, if we remove buffer 2 and activity 2 from the fork-join network, redefine $A_{21} = 1$, and keep the above parameter settings, then Assumption HT holds with $x^* = (1, 1)$. However, $2 = \overline{\mathbf{K}} < \mathbf{K} = 3$ and $\mathbf{L} = 3$.

7 Non-Negative Canonical Workload

Recall from Section 5 that there are infinitely many possible choices for the workload matrix Λ. In [16], Harrison proposed a "canonical" choice, which reduces the possibilities to a finite number. In general, for this choice to be valid, an additional condition is required, beyond the assumptions specified in [16]. In this section, we provide sufficient conditions for this canonical choice to be valid, which moreover ensure that the workload matrix is non-negative. In particular, in Corollary 7.4, we show that it suffices to have a unitary network in which each basic activity is processed by exactly one server and each buffer is served by at least one basic activity. These assumptions automatically hold, for example, for parallel server systems.

This section is organized as follows. Firstly, in Section 7.1, we review Harrison's proposal for a canonical choice of workload matrix and provide an example to show that this choice is not always valid. We also point out the general assumption that is implicitly made in [16]. In Section 7.2, we present the main result of this section, Theorem 7.3, which gives general sufficient conditions for Harrison's canonical choice to be valid and for it to yield a non-negative workload matrix. Since Harrison's procedure uses extremal optimal solutions of the dual to the linear program (LP), as a preliminary to the proof of Theorem 7.3, we develop several properties of this dual program. The conditions of Theorem 7.3 are somewhat abstract. To facilitate its use, in Section 7.3, we show that the assumptions of Theorem 7.3 hold under certain natural conditions. In particular, these results imply the result for unitary networks cited in the paragraph immediately above. Finally, in Section 7.4, we give an example to show how the conclusion of Theorem 7.3 can fail if one of the assumptions is not satisfied.

7.1 Dual Program and Harrison's Canonical Workload Proposal

The following linear program is *dual* to the linear program (LP):

$$\text{maximize } y'\alpha \ \text{ subject to } \ y'R \le z'A, \quad z'\mathbf{1} = 1 \ \text{ and } \ z \ge 0. \tag{DP}$$

Linear programming theory tells us that the optimal value of the dual program (DP) equals that of the primal program (LP), i.e., it is one, and that the feasible region for the dual program, i.e., $\{(y,z) : y'R \le z'A,\ z'\mathbf{1} = 1,\ z \ge 0\}$, has only finitely many extreme points. Suppose that $(y^1, z^1), \ldots, (y^{\mathbf{N}}, z^{\mathbf{N}})$ are the extremal optimal solutions to the dual program (DP). That is, these are the extreme points of the feasible region for (DP) that achieve the optimal value of the (DP). Then, $(y^\ell)'\alpha = 1$ for $\ell = 1, \ldots, \mathbf{N}$. (A priori, since the dual feasible region need not be bounded, there might not be any extremal optimal solutions, in which case $\mathbf{N} = 0$.)

Harrison [16] proposed choosing a canonical workload matrix Λ as a matrix with rows given by a maximal linearly independent subset of the vectors $y^1, \ldots, y^{\mathbf{N}}$. The validity of Harrison's procedure hinges on the following representation which he claimed for the space $\mathcal{N}^\perp$ introduced in Section 5,

$$\mathcal{N}^\perp = \operatorname{span}\{y^1, \ldots, y^{\mathbf{N}}\}. \tag{11}$$

In this case, although the $\mathbf{L} \times \mathbf{I}$ matrix Λ need not be unique, there are only finitely many choices for it. Unfortunately, the representation (11), claimed in the statement of Proposition 3 of [16], does not always hold in the full generality of the setup in [16], as is shown by the counterexample below.

Example. Let $R = (1,1)'$, $A = (1)$ and $\alpha = (1,1)'$. These specify the first-order data for an open stochastic processing network with two buffers, one server and one activity, where the activity simultaneously consumes material from both buffers (cf., the fork-join example in Section 3). Assumption HT holds with $x^* = 1$ and the dual program (DP) reduces to

$$\text{maximize } y_1 + y_2 \ \text{ subject to } \ y_1 + y_2 \le 1 \ \text{ and } \ z_1 = 1.$$

The feasible region for this problem is a (two-dimensional) half-plane in $\mathbb{R}^3$ which has no extreme points. Thus, the right-hand side of (11) is the empty set. On the other hand, $\mathcal{N} = \{(0,0)\}$ and so $\mathcal{N}^\perp = \mathbb{R}^2$.

On examining the proof of Proposition 3 in [16], one finds that it implicitly assumes (through Proposition 1 of [16] and its use with small perturbations of λ in place of λ there), that the optimal value of the dual program (DP) (with $\alpha + \epsilon\delta$ in place of α for $\delta \in \text{span}\ \{y^1, \ldots, y^{\mathbf{N}}\}$, and $\epsilon \neq 0$ sufficiently small), is achieved at an extreme point of the feasible region for (DP). While this is automatically true if the feasible region for (DP) is bounded, it need not always hold in the generality of the model framework assumed in [16], as the example above shows.

7.2 Sufficient Conditions for Non-Negative Canonical Workload

In this subsection (cf., Theorem 7.3), we provide sufficient conditions under which (11) holds and $y^1, \ldots, y^{\mathbf{N}}$ are all non-negative. The first property ensures that Harrison's procedure for choosing a workload matrix is valid and the second ensures that the resulting matrix Λ is non-negative. The workload process, which is obtained by applying Λ to the non-negative queue length process, is then non-negative. This appealing property is consistent with the non-negativity of the workload process associated with open multiclass queueing networks (see the example in Section 5).

Key to the proof of the main result of this section is Lemma 7.2, which is given below. For the statement of the lemma, we need the following assumptions. For these, recall the definition of the matrix H appearing in the decomposition (3) of R.

Assumption BAB (Basic Activity for Each Buffer). *For each buffer i, there is at least one (basic) activity j such that $H_{ij} > 0$.*

Remark. The Assumption BAB is equivalent to the assumption that, for each buffer i, there is a *basic* activity j such that $R_{ij} > 0$.

Definition 7.1 *A matrix is Leontief if each column of the matrix contains at most one strictly positive element.*

Remark. The R and A matrices associated with unitary data (see Section 3 for the definition) are readily verified to satisfy the Leontief condition. Leontief matrices are a much studied and important sub-class of input-output matrices appearing in economic models. The definition above is the same as that used in Bertsimas and Tsitsiklis [6], page 195. We warn the reader wishing to consult other references on this subject that different authors may use the term Leontief for a slightly more restricted class of matrices, e.g., some authors require that each column of the matrix have exactly one strictly positive entry, and some require in addition that there be a vector $x \geq 0$ such that $Dx > 0$, where D is the matrix in question. We note also that when interpreting the entries in a Leontief matrix, rather than being concerned with consuming material at a given overall rate of α (cf., the interpretation of R in Section 2 and of (LP) in Section 4), authors are frequently concerned with producing material at a given rate α, and interpret a strictly positive entry in a Leontief matrix as a rate of production and a strictly negative entry as a rate of consumption. This essentially amounts to an interchange of the terms input and output in the interpretations given in Sections 2 and 4. For further details on the mathematical theory of Leontief matrices, we refer the interested reader to Chapter 9 of Berman and Plemmons [4], for square matrices, and to Koehler, Whinston and Wright [24], more generally.

In the following assumption, the *positive part* H^+ of the matrix H is defined by replacing all of the negative entries in H by zeros.

Assumption DCH (Decomposition of H). *The matrix H is Leontief and there is a non-negative $\mathbf{B} \times \mathbf{B}$ matrix Q that has spectral radius strictly less than one and that satisfies $H = H^+(I - Q)$, where I is the $\mathbf{B} \times \mathbf{B}$ identity matrix.*

Remark. The decomposition of H given in Assumption DCH may seem somewhat unnatural, but has the following consequences. The form assumed there is equivalent to $H^- = H^+Q$, where the $\mathbf{I} \times \mathbf{B}$ matrix $H^- = H^+ - H$. Since Q maps $\mathbb{R}^{\mathbf{B}}$ into $\mathbb{R}^{\mathbf{B}}$, one can iterate Q, which one cannot do for H^-. In particular, Q being non-negative and having spectral radius strictly less than one implies that $(I - Q)^{-1} = \sum_{i=0}^{\infty} Q^i$ exists and is non-negative, and $H^+ = H(I - Q)^{-1}$. These properties for Q and H will be used several times in the proof of Lemma 7.2. In Lemma 7.6 of the next subsection, we show that such Q exist for unitary networks which satisfy Assumption BAB. In that setting, Q is closely related to the matrix P' that is part of the specification of the R matrix for a unitary network.

Lemma 7.2 establishes various properties of the dual program (DP). In particular, it provides sufficient conditions under which all optimal solutions of (DP) are non-negative, and allows us to re-interpret these solutions as optimal solutions of another dual program (DP^+), which has a bounded feasible region. We remind the reader that we are assuming throughout that Assumption HT holds.

Lemma 7.2 *Suppose that Assumptions BAB and DCH hold. Then, the following four properties (i)–(iv) hold.*

(i) Any optimal solution (y, z) of the dual program (DP) is non-negative.

(ii) If each column of the matrix B has at least one strictly positive entry, then there is at least one optimal solution (y, z) of the dual program (DP) satisfying $y > 0$, $z > 0$.

(iii) The set $F \equiv \{(y,z) \in \mathrm{I\!R}^{\mathbf{I}} \times \mathrm{I\!R}^{\mathbf{K}} : y'R \leq z'A,\ z'\mathbf{1} = 1,\ y \geq 0,\ z \geq 0\}$ *is bounded.*

(iv) The dual program (DP) *and the linear program*

$$\text{maximize } y'\alpha \quad \text{subject to} \quad y'R \leq z'A, \quad z'\mathbf{1} = 1,\ y \geq 0 \quad \text{and} \quad z \geq 0, \qquad (DP^+)$$

have the same optimal values, the same sets of optimal solutions, and the same sets of extremal optimal solutions.

Proof of Lemma 7.2. For the proof of (i), let (y,z) be an optimal solution of the dual program (DP). To show that (y,z) is non-negative, it suffices to prove that $y \geq 0$, since $z \geq 0$ is one of the constraints of (DP). By complementary slackness, for components corresponding to basic activities, the inequality $y'R \leq z'A$ is an equality, and so

$$y'H = z'B. \tag{12}$$

Now, since Q is non-negative with spectral radius strictly less than one, $I - Q$ is invertible with non-negative inverse $(I-Q)^{-1}$ given by the convergent power series $\sum_{i=0}^{\infty} Q^i$. On postmultiplying (12) by $(I-Q)^{-1}$, we obtain

$$y'H^+ = z'B(I-Q)^{-1}, \tag{13}$$

where the right side is non-negative because z, B and $(I-Q)^{-1}$ are all non-negative. Fix $i \in \{1, \ldots, \mathbf{I}\}$. By Assumption BAB, there is at least one (basic) activity j such that $H_{ij} > 0$, and hence $H^+_{ij} > 0$. Choose such an activity and call it $j(i)$. Since H is Leontief, H^+ has at most one non-zero entry in each column and so the only non-zero entry in the $j(i)^{th}$ column of H^+ is the strictly positive entry in the i^{th} row. Thus, by (13), for $j = j(i)$ we have

$$0 \leq (z'B(I-Q)^{-1})_j = (y'H^+)_j = y_i H^+_{ij}, \tag{14}$$

and since $H^+_{ij} > 0$ for $j = j(i)$, we conclude that $y_i \geq 0$. Since i was arbitrary, we have $y \geq 0$.

We now prove (ii). Since there is an optimal solution of the linear program (LP) (by Assumption HT), there is at least one optimal solution of the dual program (DP). Since the solution of (LP) has all slack variables equal to zero (because $Ax^* = \mathbf{1}$), by strict complementary slackness (cf., [6], p. 192), there is an optimal solution (y,z) of the dual program (DP) with $z > 0$. Then, since $(I-Q)^{-1} \geq I$, we have from (14) that for each i, with $j = j(i)$,

$$y_i H^+_{ij} = (z'B(I-Q)^{-1})_j \geq (z'B)_j, \tag{15}$$

where $H^+_{ij} > 0$. Now, if we assume that each column of B has a strictly positive entry, then for each j, there is $k(j) \in \{1, \ldots, \mathbf{K}\}$ such that $B_{kj} > 0$ for $k = k(j)$. Since $z > 0$ and B is non-negative, it then follows that the right side of (15) is strictly positive and hence $y_i > 0$. Since i was arbitrary, $y > 0$.

For the proof of (iii), let (y, z) be a member of the set F, the feasible set for (DP^+). Then, $0 \leq z_k \leq 1$ for $k = 1, \ldots, \mathbf{K}$. By assumption, $y \geq 0$, and so it suffices to obtain a uniform upper bound for y. Applying $y'H \leq z'B$, $z \leq \mathbf{1}$ and the decomposition assumed for H, we have

$$y'H^+(I - Q) \leq z'B \leq \mathbf{1}'B. \tag{16}$$

On multiplying this inequality on the right by the non-negative matrix $(I - Q)^{-1}$, we obtain

$$y'H^+ \leq \mathbf{1}'B(I - Q)^{-1}. \tag{17}$$

Now, as in the proof of (i), for each $i \in \{1, \ldots, \mathbf{I}\}$, there is $j(i) \in \{1, \ldots, \mathbf{B}\}$ such that for $j = j(i)$, $H^+_{ij} > 0$ and so $y_i \leq (\mathbf{1}'B(I - Q)^{-1})_j / H^+_{ij}$. Since i was arbitrary, it follows that there is a uniform bound for y. Hence, the feasible set F for (DP^+) is bounded.

Finally, we prove (iv). Recall that (DP) has at least one optimal solution (y, z). By (i), for any such (y, z), $y \geq 0$, and so (y, z) is a feasible solution of (DP^+). Since feasible solutions of (DP^+) are also feasible for (DP), (y, z) is optimal for (DP^+) as well. So, the optimal values for (DP) and (DP^+) are the same. It follows that (y, z) is optimal for (DP) whenever it is optimal for (DP^+), and therefore the two sets of optimal solutions are the same. The extremal optimal solutions are also the same for the two problems, since in each case they are just the extreme points of the sets of optimal solutions. (An extremal optimal solution is, of course, an extreme point of the optimal solution set, whereas an optimal solution which is not an extreme point of the feasible set is a convex combination of two different feasible solutions which both must be optimal.) □

The following strengthening of part of the heavy traffic assumption (see (i) of Lemma 4.3), will be needed for the statement of Theorem 7.3 below, which is the main result of this section.

Assumption UX (Unique x). *Any solution x of*

$$Rx \geq \alpha, \quad Ax \leq \mathbf{1}, \quad x \geq 0, \tag{18}$$

satisfies $x = x^*$.

Theorem 7.3 gives sufficient conditions for Harrison's canonical choice of workload matrix to be valid and for the matrix to be non-negative.

Theorem 7.3 *Suppose that Assumptions BAB, DCH and UX hold. Let* $(y^1, z^1), \ldots, (y^{\mathbf{N}}, z^{\mathbf{N}})$ *be the extremal optimal solutions of (DP). Then,* $\mathcal{N}^\perp = span\{y^1, \ldots, y^{\mathbf{N}}\}$*, and any* $\mathbf{L} \times \mathbf{I}$ *matrix* Λ *whose rows are given by a set of* $\mathbf{L}$ *linearly independent vectors chosen from* $y^1, \ldots, y^{\mathbf{N}}$ *is a non-negative workload matrix.*

Proof. Let $\mathcal{M} = \text{span}\{y^1, \ldots, y^{\mathbf{N}}\}$. Suppose first that $\mathcal{N}^\perp = \mathcal{M}$. Since $\mathcal{N}^\perp$ has dimension $\mathbf{L}$, a maximal linearly independent subset of $y^1, \ldots, y^{\mathbf{N}}$ must have exactly $\mathbf{L}$ members. By part (i) of Lemma 7.2, all of the vectors in such a set are non-negative. So, by the definition given in Section 5, any $\mathbf{L} \times \mathbf{I}$ matrix Λ whose rows are given by such a maximal set of vectors is a non-negative workload matrix.

Now we turn to the proof that $\mathcal{N}^\perp = \mathcal{M}$. For this, we prove the equivalent statement that $\mathcal{N} = \mathcal{M}^\perp$, where $\mathcal{M}^\perp$ denotes the orthogonal complement of $\mathcal{M}$ in $\mathbb{R}^{\mathbf{I}}$.

First, we prove that $\mathcal{N} \subset \mathcal{M}^\perp$. For this, let $\delta \in \mathcal{N}$. Then, there is $x_B \in \mathbb{R}^{\mathbf{B}}$ such that $Hx_B = \delta$ and $Bx_B = 0$. Let (y, z) be an extremal optimal solution for (DP). By complementary slackness,

$$y'H = z'B \tag{19}$$

and so

$$y'\delta = y'Hx_B = z'Bx_B = 0. \tag{20}$$

Since $y^1, \ldots, y^{\mathbf{N}}$ are the y components of extremal optimal solutions for (DP), it follows that $\delta \in \mathcal{M}^\perp$.

To prove the reverse inclusion, $\mathcal{M}^\perp \subset \mathcal{N}$, fix $\delta \in \mathcal{M}^\perp$. By parts (i) and (iv) of Lemma 7.2, $(y^1, z^1), \ldots, (y^{\mathbf{N}}, z^{\mathbf{N}})$ are all non-negative and they are the extremal optimal solutions for (DP^+). Consider the following linear program obtained by modifying the objective function for (DP^+):

$$\text{maximize } y'(\alpha + \epsilon\delta) \text{ subject to } \quad y'R \le z'A, \quad z'\mathbf{1} = 1, \quad y \ge 0 \text{ and } z \ge 0, \tag{DP_ϵ^+}$$

for a given $\epsilon \in \mathbb{R}$. By (iii) of Lemma 7.2, the feasible region F for (DP^+), which is the same as that for (DP_ϵ^+), is bounded. Thus, the optimal value for (DP_ϵ^+) will be achieved at an extreme point of F. For those extreme points (y, z) of F that are not optimal for (DP^+) (i.e., that are not in the set $\{(y^1, z^1), \ldots, (y^{\mathbf{N}}, z^{\mathbf{N}}\})$, we have $y'\alpha < 1$. Since there are only finitely many such extreme points, it follows that there is $\tilde{\epsilon} > 0$ such that for each $\epsilon \in [-\tilde{\epsilon}, \tilde{\epsilon}]$, the optimal value of (DP_ϵ^+) is achieved at one of the extreme points $(y^1, z^1), \ldots, (y^{\mathbf{N}}, z^{\mathbf{N}})$. Since $\delta \in \mathcal{M}^\perp$, it follows that in each case the optimal value of (DP_ϵ^+) equals one.

The following linear program is dual to (DP_ϵ^+):

$$\text{minimize } \rho \text{ subject to } \quad Rx \ge \alpha + \epsilon\delta, \quad Ax \le \rho\mathbf{1} \text{ and } x \ge 0. \tag{LP_ϵ^+}$$

It follows from the duality theory of linear programming that, for each $\epsilon \in [-\tilde{\epsilon}, \tilde{\epsilon}]$, the optimal value of this program equals one. Let $(\rho, x) = (1, x^+)$ and $(\rho, x) = (1, x^-)$ be optimal solutions of (LP^+_ϵ) with $\epsilon = \tilde{\epsilon}$ and $\epsilon = -\tilde{\epsilon}$, respectively. These solutions satisfy

$$Rx^+ \geq \alpha + \tilde{\epsilon}\delta, \quad Ax^+ \leq \mathbf{1} \quad \text{and} \quad x^+ \geq 0, \tag{21}$$

$$Rx^- \geq \alpha - \tilde{\epsilon}\delta, \quad Ax^- \leq \mathbf{1} \quad \text{and} \quad x^- \geq 0. \tag{22}$$

Let $\bar{x} = (x^+ + x^-)/2$. Then,

$$R\bar{x} \geq \alpha, \quad A\bar{x} \leq \mathbf{1} \quad \text{and} \quad \bar{x} \geq 0.$$

By the assumption that (18) has the unique solution x^*, it follows that $\bar{x} = x^*$. Then, since $x^*_N = 0$, it follows from the non-negativity of x^+ and x^- that $x^+_N = x^-_N = 0$. Furthermore, since $Rx^* = \alpha$, it follows that the inequalities in (21) and (22) involving Rx^+ and Rx^- must be equalities. Similarly, since $Ax^* = \mathbf{1}$, we must have $Ax^+ = \mathbf{1}$ and $Ax^- = \mathbf{1}$. Thus, $x = (x^+ - x^-)/2\tilde{\epsilon}$ satisfies

$$Rx = \delta, \quad Ax = 0, \quad x_N = 0, \tag{23}$$

and so $\delta \in \mathcal{N}$, as desired. Thus we have shown that $\mathcal{M}^\perp \subset \mathcal{N}$. Since we have already proved the reverse inclusion, it follows that $\mathcal{N} = \mathcal{M}^\perp$. $\square$

Remark. Since this paper was first submitted for publication, Harrison submitted a correctional note [17] to [16] in which the following additional assumption was shown to imply that his canonical choice of workload matrix is valid, i.e., that (11) holds. Here, for ease of comparison, we use the same label, for this natural geometric assumption, as is used in [17].

Assumption 0. *For each* $\delta \in \mathbb{R}^{\mathbf{I}}$, $\alpha + \epsilon\delta \in \mathcal{V}$ *for all* ϵ *sufficiently small, where* $\mathcal{V} = \{Rx : x \in \mathbb{R}^{\mathbf{J}}_+\}$.

Since our Theorem 7.3 validates the canonical choice of workload matrix, it is not surprising that the conditions of that theorem imply that Assumption 0 holds. In fact, Assumption 0 follows from Assumptions BAB and DCH. For completeness, we provide a proof here. For this, we first employ Assumptions DCH and BAB to show that the mapping H from $\mathbb{R}^{\mathbf{B}}$ into $\mathbb{R}^{\mathbf{I}}$, defined by $x \to Hx$, is onto all of $\mathbb{R}^{\mathbf{I}}$. This, we do as follows. If $y \in \mathbb{R}^{\mathbf{I}}$ such that $y'H = 0$, then using the decomposition $H = H^+(I - Q)$ guaranteed by Assumption DCH, and multiplying on the right by $(I - Q)^{-1}$, we obtain $y'H^+ = 0$. By Assumption BAB and the Leontief property of H^+, which is inherited from H, we have that for each $i \in \{1, \ldots, \mathbf{I}\}$, there is an index $j(i)$ such that $H^+_{i,j(i)} > 0$ and $H^+_{k,j(i)} = 0$ for all $k \neq i$, and so it follows from $y'H^+ = 0$ that $y_i = 0$. Hence, H is onto $\mathbb{R}^{\mathbf{I}}$. It then follows that, for any $\delta \in \mathbb{R}^{\mathbf{I}}$, there is an $x_\delta \in \mathbb{R}^{\mathbf{B}}$ such that $Hx_\delta = \delta$. Since the components of x^*_B are strictly positive, being the basic components of x^*, it follows that there is an $\epsilon_0 > 0$ such that $x^\epsilon_B = x^*_B + \epsilon x_\delta$ has all components strictly positive provided $\epsilon \in [-\epsilon_0, \epsilon_0]$. Moreover, $Hx^\epsilon_B = \alpha + \epsilon\delta$. We can extend x^ϵ_B to a vector x^ϵ in $\mathbb{R}^{\mathbf{J}}$ by setting the additional $\mathbf{J} - \mathbf{B}$ components to zero, and then $Rx^\epsilon = \alpha + \epsilon\delta$. It follows that Assumption 0 holds.

An important collection of open stochastic processing networks which motivated the form of the conditions in Theorem 7.3 is the collection of unitary networks introduced in Section 3. In fact, we will prove the following corollary of Theorem 7.3 in the next subsection.

Corollary 7.4 *Suppose the first-order data (R, A, α) are unitary, the matrix B has exactly one strictly positive entry in each column, and Assumption BAB holds. Then, the conclusions of Theorem 7.3 hold.*

Examples. The assumptions of Corollary 7.4 are automatically satisfied by parallel server systems, since $\alpha > 0$ in such systems, which, in view of Assumption HT, implies that BAB holds. An open multiclass queueing network in which all activities are basic also satisfies the assumptions of Corollary 7.4. For the fork-join network of Section 3 with the parameters $\alpha_1 = 1, \alpha_2 = \frac{1}{2}, \beta_1 = \beta_2 = \beta_3 = \beta_4 = \gamma_1 = \gamma_2 = 1$, Assumption HT holds with $x^* = (1, \frac{1}{2}, 1)$, but the assumptions of Theorem 7.3 do not hold since H is not Leontief. In this case, $\mathcal{N} = \{0\}$ and so $\mathcal{N}^\perp = \mathbb{R}^4$. On the other hand, the optimal solutions of (DP) are given by

$$\Bigg\{(y,z) : y = \left(1 - \frac{1}{2}z_2, z_2, y_3, z_3 - y_3\right),\ z = (1 - z_2 - z_3, z_2, z_3), \qquad (24)$$
$$\text{where } z_2 \geq 0,\ z_3 \geq 0,\ z_2 + z_3 \leq 1,\ y_3 \in \mathbb{R}\Bigg\}.$$

Since y_3 is unrestricted, it is straightforward to see that this set has no extreme points and so there are no extremal optimal solutions. By choosing $y_3 < 0$, one obtains optimal solutions that fail to be non-negative. Thus, the conclusions of Theorem 7.3 also fail to be true for this example.

7.3 Sufficient Conditions for Assumptions DCH and UX to Hold

In this subsection we describe some natural conditions under which the assumptions of Theorem 7.3 hold. These conditions are of interest in their own right and will also allow us to prove Corollary 7.4.

Lemma 7.5 *Suppose that Assumptions BAB and DCH hold, and that B has a strictly positive entry in each column. Then, Assumption UX holds.*

The idea for the proof of this lemma is due to J. M. Harrison (private communication).

Proof. By part (ii) of Lemma 7.2, there is an optimal solution $(\tilde{y}, \tilde{z})$ of (DP) satisfying $\tilde{y} > 0$, $\tilde{z} > 0$. Suppose $\tilde{x}$ is a solution of (18) and $\tilde{x} \neq x^*$. Let $\gamma = R\tilde{x}$. If $\gamma = \alpha$, then by Assumption HT (see Lemma 4.3), $\tilde{x} = x^*$. So, $\gamma \geq \alpha$ and $\gamma \neq \alpha$. Since $\tilde{y} > 0$, it follows that $\tilde{y}'\gamma > \tilde{y}'\alpha = 1$.

Consider the linear program

$$\text{minimize } \rho \;\text{ subject to }\; Rx = \gamma, \;\; Ax \leq \rho\mathbf{1} \;\text{ and }\; x \geq 0. \tag{LP_γ}$$

Now, $(\rho, x) = (1, \tilde{x})$ is a feasible solution for this program. The following linear program is dual to (LP_γ),

$$\text{maximize } y'\gamma \;\text{ subject to }\; y'R \leq z'A, \;\; z'\mathbf{1} = 1 \;\text{ and }\; z \geq 0. \tag{DP_γ}$$

Since this has the same feasible set as (DP), $(\tilde{y}, \tilde{z})$ is feasible for (DP_γ). But the value of the dual objective at this point $(\tilde{y}, \tilde{z})$ is $\tilde{y}'\gamma > 1$, which exceeds the value of the primal objective at $(\rho, x) = (1, \tilde{x})$. This contradicts weak duality and so there cannot be any such solution $\tilde{x} \neq x^*$ of (18). □

Lemma 7.6 *Suppose that H is Leontief and Assumption BAB holds. Then, there is a non-negative $\mathbf{B} \times \mathbf{B}$ matrix Q such that*

$$H = H^+(I - Q), \tag{25}$$

where I is the $\mathbf{B} \times \mathbf{B}$ identity matrix. If, in addition, R is part of a set of unitary data (R, A, α), then one can choose this Q so that its spectral radius is strictly less than one. Hence, Assumption DCH holds.

Proof. For each buffer $i \in \mathbf{I}$, let $\mathcal{C}(i) = \{j : H_{ij} > 0\}$, the set of basic activities that consume material from buffer i. From the assumption that H is Leontief, we conclude that the sets $\mathcal{C}(i) : i = 1, \ldots, \mathbf{I}$, are disjoint. Since BAB holds, for each buffer i, we can choose a *basic* activity $b(i) \in \mathcal{C}(i)$; if there is more than one such activity, just choose one.

Define the $\mathbf{B} \times \mathbf{B}$ matrix Q as follows. For each basic activity l, since the $\mathcal{C}(i)$: $i = 1, \ldots, \mathbf{I}$, are disjoint, $l = b(i)$ for at most one i. If $l \neq b(i)$ for all i, set $Q_{lj} = 0$ for $j = 1, \ldots, \mathbf{B}$, and if $l = b(i)$ for some i, set $Q_{lj} = H^-_{ij}/H^+_{il}$ for that i and each $j = 1, \ldots, \mathbf{B}$, where $H^- = H^+ - H$. Thus, for $i = 1, \ldots, \mathbf{I}$ and $j = 1, \ldots, \mathbf{B}$,

$$(H^+Q)_{ij} = \sum_{l=1}^{\mathbf{B}} H^+_{il} Q_{lj} = \sum_{l \in \mathcal{C}(i)} H^+_{il} Q_{lj} = H^+_{ib(i)} Q_{b(i)j} = H^-_{ij},$$

where we have used the fact that if $l \in \mathcal{C}(i)$, then $l \neq b(m)$ for any $m \neq i$, because the sets $\mathcal{C}(m)$, $m = 1, \ldots, \mathbf{I}$ are disjoint. Thus,

$$H = H^+ - H^- = H^+ - H^+Q = H^+(I - Q), \tag{26}$$

and the desired decomposition (25) holds. (An interpretation of Q is that material produced by the basic activity j for buffer i, under H^-, is produced by j for the basic activity $b(i)$, under Q. That is, rather than relating basic activities to buffers, Q relates basic activities to basic activities. This is a mathematical device only, and it is not being suggested that material should be processed according to the dictates of Q.)

We now suppose, in addition, that R is associated with unitary data. Recall, in this case, $R = (C - P')\,\mathrm{diag}(\beta)$, where C is a $\mathbf{I} \times \mathbf{J}$ matrix of zeros and ones with at most one non-zero entry in each column, P' is the transpose of a $\mathbf{J} \times \mathbf{I}$ matrix P with all row sums less than or equal to one, and β is a $\mathbf{J}$-dimensional vector with strictly positive coordinates. Then, $H = (\tilde{C} - \tilde{P}')\,\mathrm{diag}(\tilde{\beta})$ where $\tilde{C}$ and $\tilde{P}'$ are obtained from C and P', respectively, by deleting the columns corresponding to non-basic activities, and $\tilde{\beta}$ is obtained from β by deleting the coordinates corresponding to non-basic activities. Since C is Leontief, this property is inherited by $\tilde{C}$, and hence also by $H^+ = \tilde{C}\,\mathrm{diag}(\tilde{\beta})$ and H. Since the column sums of P' are all less than or equal to one, the same is true for $\tilde{P}'$.

Let Q be defined as in the first part of this proof. One can check that, for any pair of basic activities (l,j), the (l,j) entry of Q equals zero if $l \neq b(i)$ for all i, and equals $\tilde{P}'_{ij}\tilde{\beta}_j/\tilde{\beta}_l$ if $l = b(i)$. (Recall that there is at most one i such that $l = b(i)$.) Therefore, the (l,j) entry of $\hat{Q} = \mathrm{diag}(\tilde{\beta})\,Q\,\mathrm{diag}(\tilde{\beta})^{-1}$ is zero if $l \neq b(i)$ for all i, and equals $\tilde{P}'_{ij}$ if $l = b(i)$. It follows that the column sums of the non-negative $\mathbf{B} \times \mathbf{B}$ matrix $\hat{Q}$ are all less than or equal to one. Hence, the spectral radius of Q, which is the same as that of $\hat{Q}$, is less than or equal to one.

We still need to show that the spectral radius of Q is *strictly* less than one. For this, we employ a proof by contradiction, and assume that the spectral radius of Q is equal to one. By the finite-dimensional Krein-Rutman (or general Perron-Frobenius) theorem, there is a right eigenvector x_B for Q corresponding to the eigenvalue one, where the eigenvector is not identically zero and has all non-negative entries (see Theorem 1.3.2 of [4]). Combining this with the decomposition $H = H^+(I - Q)$ obtained in the first part of the proof, we conclude that $Hx_B = 0$. Extend the $\mathbf{B}$-dimensional vector x_B to a $\mathbf{J}$-dimensional vector x by setting the first $\mathbf{B}$ entries of x to agree with those of x_B and the last $\mathbf{J} - \mathbf{B}$ entries to zero. Then, $Rx = 0$ and $x \geq 0$. Since the optimal solution $(1, x^*)$ of (LP) has $x^*_j > 0$ for $j = 1, \ldots, \mathbf{B}$, there is an $\epsilon > 0$ such that $\tilde{x} \equiv x^* - \epsilon x \geq 0$. Consequently, $R\tilde{x} = Rx^* = \alpha$; also, $A\tilde{x} = Ax^* - \epsilon Ax \leq \mathbf{1}$, since the entries in A are non-negative and $Ax^* = \mathbf{1}$. Thus, $\tilde{x}$ satisfies (2) with $\tilde{x}$ in place of x^* there. But, $\tilde{x} \neq x^*$, which violates Assumption HT (by Lemma 4.3), and so the desired contradiction is obtained.

Thus, the desired decomposition holds, where Q has spectral radius strictly less than one. Since H is Leontief, Assumption DCH follows. □

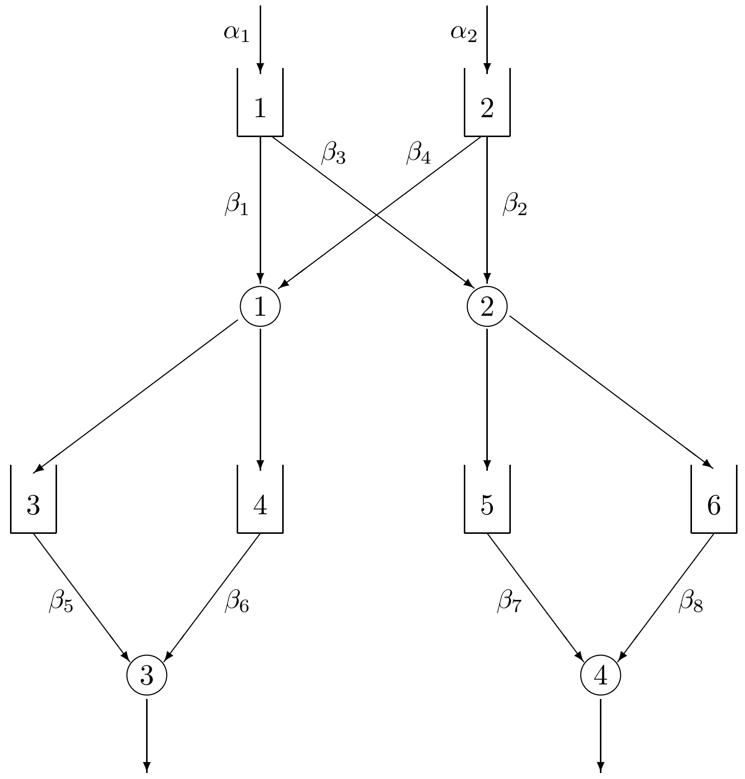

Figure 5: A unitary network with feedforward deterministic routing

Proof of Corollary 7.4. Since R is Leontief, by Lemmas 7.5 and 7.6, the assumptions of Theorem 7.3 hold. □

7.4 Example Where Assumption BAB Fails for a Unitary Network

The fork-join example following Corollary 7.4 shows that, if H is not Leontief, there may be optimal solutions of the dual program (DP) that fail to be non-negative. The following example illustrates how non-negativity can also fail to hold, even for a unitary network, if Assumption BAB is not satisfied.

Example. Consider the unitary network pictured in Figure 5. The network has an associated R matrix of the form $(C - P')\text{diag}(\beta)$, where the matrix P prescribes deterministic routing of material such that material processed at server 1 (or server 2) is routed in the same "direction" as the activity that processed it. In particular, $P_{14} = P_{25} = P_{36} = P_{43} = 1$, and all other P_{ji} are equal to 0. Assume that $\alpha_1 = \alpha_2 = 1$, $\beta_1 = \beta_2 = 1$, $\beta_3 = \beta_4 = \frac{1}{2}$, $\beta_5 = \beta_6 = \beta_7 = \beta_8 = 1$. There is a unique solution of (LP) given by $x_1^* = x_2^* = x_6^* = x_7^* = 1$, $x_3^* = x_4^* = x_5^* = x_8^* = 0$. The dual program (DP) has many optimal solutions, one of which is (y, z) with

$$y = \left(\frac{1}{2}, \frac{1}{2}, -\frac{1}{2}, 0, 0, -\frac{1}{2}\right), \quad z = \left(\frac{1}{2}, \frac{1}{2}, 0, 0\right).$$

Thus, this is an optimal solution of (DP) that is not non-negative. Of course, in this example, Assumption BAB does not hold, since there is no basic activity that consumes material from buffer 3 or buffer 6.

Acknowledgement. The authors thank J. M. Harrison for helpful discussions and, in particular, for drawing our attention to the notion of Leontief matrices and related literature. In addition, we are grateful to him for the idea for the proof of Lemma 7.5. We also thank D. Bertsimas for noting the interpretation of the workload formula described in the Remark following Corollary 6.2.

References

[1] B. Ata and S. Kumar, Heavy traffic analysis of open processing networks with complete resource pooling: asymptotic optimality of discrete review policies, preprint.

[2] 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.

[3] S. L. Bell and R. J. Williams, Dynamic scheduling of a multiserver system with complete resource pooling: asymptotic optimality of a threshold policy in heavy traffic, preprint.

[4] A. Berman and R. J. Plemmons, *Non-Negative Matrices in the Mathematical Sciences,* SIAM, Philadelphia, PA, 1994.

[5] D. Bertsekas and R. Gallagher, *Data Networks,* Prentice-Hall, Englewood Cliffs, NJ, 1992.

[6] D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization,* Athena Scientific, Belmont, MA, 1997.

[7] M. Bramson, State space collapse with application to heavy traffic limits for multiclass queueing networks, *Queueing Systems,* **30** (1998), 89–148.

[8] M. Bramson and R. J. Williams, On dynamic scheduling of stochastic networks in heavy traffic and some new results for the workload process, *Proceedings of the 39th IEEE Conference on Decision and Control,* December 2000, 516–521.

[9] J. A. Buzacott and J. G. Shantikumar, *Stochastic Analysis of Manufacturing Systems,* Prentice Hall, Englewood Cliffs, NJ, 1993.

[10] H. Chen, P. Yang and D. Yao, Control and scheduling in a two-station queueing network: optimal policies and heuristics, *Queueing Systems,* **18** (1994), 301–332.

[11] G. B. Dantzig, The programming of interdependent activities: mathematical model, in "Activity Analysis of Production and Allocation", T. C. Koopmans (ed.), John Wiley and Sons, New York, 1951, pp. 19–32.

[12] R. Hariharan, M. S. Moustafa and S. Stidham, Jr., Scheduling in a multi-class series of queues with deterministic service times, *Queueing Systems,* **24** (1996), 83–99.

[13] J. M. Harrison, Brownian models of queueing networks with heterogeneous customer populations, 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, pp. 147–186.

[14] J. M. Harrison, The BIGSTEP approach to flow management in stochastic processing networks, in F. P. Kelly, S. Zachary and I. Ziedins (eds.), *Stochastic Networks: Theory and Applications,* Oxford University Press, 1996, pp. 57–90.

[15] 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.

[16] J. M. Harrison, Brownian models of open processing networks: canonical representation of workload, *Annals of Applied Probability,* **10** (2000), 75–103.

[17] J. M. Harrison, Correction — Brownian models of open processing networks: canonical representation of workload, *Annals of Applied Probability,* **13** (2003), 390–393.

[18] J. M. Harrison, A broader view of Brownian networks, to appear in *Annals of Applied Probability,* **13** (2003).

[19] J. M. Harrison, Stochastic networks and activity analysis, to appear in *Analytic Methods in Applied Probability: In Memory of Fridrih Karpelevic,* Y. Suhov (ed.), American Mathematical Society, 2002.

[20] J. M. Harrison and M. J. López, Heavy traffic resource pooling in parallel-server systems, *Queueing Systems,* **33** (1999), 339–368.

[21] J. M. Harrison and J. A. Van Mieghem, Dynamic control of Brownian networks: state space collapse and equivalent workload formulations, *Ann. Appl. Prob.,* **7** (1997), 747–771.

[22] J. M. Harrison and L. Wein, Scheduling networks of queues: heavy traffic analysis of a simple open network, *Queueing Systems,* **5** (1989), 265–280.

[23] 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.

[24] G. J. Koehler, A. B. Whinston and G. P. Wright, *Optimization over Leontief Substitution Systems,* North Holland, Amsterdam, 1975.

[25] T. C. Koopmans, *Activity Analysis of Production and Allocation,* John Wiley and Sons, New York, 1951.

[26] S. Kumar, Two-server closed networks in heavy traffic: diffusion limits and asymptotic optimality, *Annals of Applied Probability,* **10** (2000), 930–961.

[27] S. Kumar, Scheduling open queueing networks with sufficiently flexible resources, *Proceedings of the 37th Allerton Conference,* Univ. of Illinois, September 1999.

[28] 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.

[29] H. J. Kushner and L. F. Martins, Heavy traffic analysis of a controlled multiclass queueing network via weak convergence methods. *SIAM J. Control and Optimization,* **34** (1996), 1781–1797.

[30] C. N. Laws and Y. C. Teh, Alternative routeing in fully-connected queueing networks, *Advances in Applied Probability,* **32** (2000), 962-982.

[31] C. Maglaras, Discrete-review policies for scheduling stochastic networks: Trajectory tracking and fluid-scale asymptotic optimality, *Annals of Applied Probability,* **10** (2000).

[32] C. Maglaras, Continuous-review tracking policies for dynamic control of stochastic networks, *Queueing Systems,* **43** (2003), 43–80.

[33] A. Mandelbaum and A. L. Stolyar, Scheduling flexible servers with convex delay costs: heavy-traffic optimality of the generalized $c\mu$-rule, to appear in *Operations Research.*

[34] L. F. Martins, S. E. Shreve and H. M. Soner, Heavy traffic convergence of a controlled, multi-class queueing system, *SIAM J. Control and Optimization,* **34** (1996), 2133–2171.

[35] S. P. Meyn, Sequencing and routing in multiclass queueing networks. Part I: feedback regulation, *SIAM J. Control and Optimization,* **40** (2001), 741-776.

[36] M. Squillante, C. H. Xia, D. Yao and L. Zhang, Threshold based priority policies for parallel server systems, preprint 2000.

[37] A. L. Stolyar, Maxweight scheduling in a generalized switch: state space collapse and workload minimization in heavy traffic, preprint.

[38] R. J. Williams, Diffusion approximations for open multiclass queueing networks: sufficient conditions involving state space collapse, *Queueing Systems*, **30** (1998), 27–88.

[39] R. J. Williams, On dynamic scheduling of a parallel server system with complete resource pooling, in *Analysis of Communication Networks: Call Centres, Traffic and Performance,* D. R. McDonald and S. R. E. Turner (eds.), Fields Institute Communications, Volume 28, American Mathematical Society, Providence, RI, 2000, pp. 49–71.

[40] D. D. Yao (ed.), *Stochastic Modeling and Analysis of Manufacturing Systems,* Springer-Verlag, New York, 1994.