Abstract

Full title: Data-Based Models of Resource-Driven Activity Networks

Application number: 2014180

Principal investigators:
Israel: Avishai Mandelbaum (Technion)
USA: Mor Armony (New York University), Petar Momčilović (University of Florida)

Abstract:

The proposed research focuses on many-server service processing networks. Motivated by applications such as healthcare services, in which multiple types of resources (e.g. doctor, nurse, bed and equipment) are required for a service to be performed, we develop a resource-based activity-network framework, under which all entities play an equal role. In particular, customers, servers, equipment, etc. are all considered as resources, and activities are entities which take some resources as input and output a potentially different set of resources. This framework is natural in modeling service systems in which customers and servers have a symmetric role. One particular case of such symmetry is closed queueing networks, where both the customer and the server populations are finite.

In recent years, with advancements in information technology (e.g. RTLS, which stands for Real-Time-Location-Systems), large transaction-level data sets have become available, in which the location, time stamp, duration and participating resources of each transaction are accurately and usefully recorded. We have access to ample such data. It will support the development of algorithms that would automatically translate the data into our activity network model at any choice of aggregation level. In parallel, we plan to develop theory and tools to analyze such models in order to advise on design, staffing and control decisions for the original system. Our modeling framework encompasses a wide range of service networks such as closed and open queueing networks, fork-join networks, and networks that operate in multiple operational regimes.

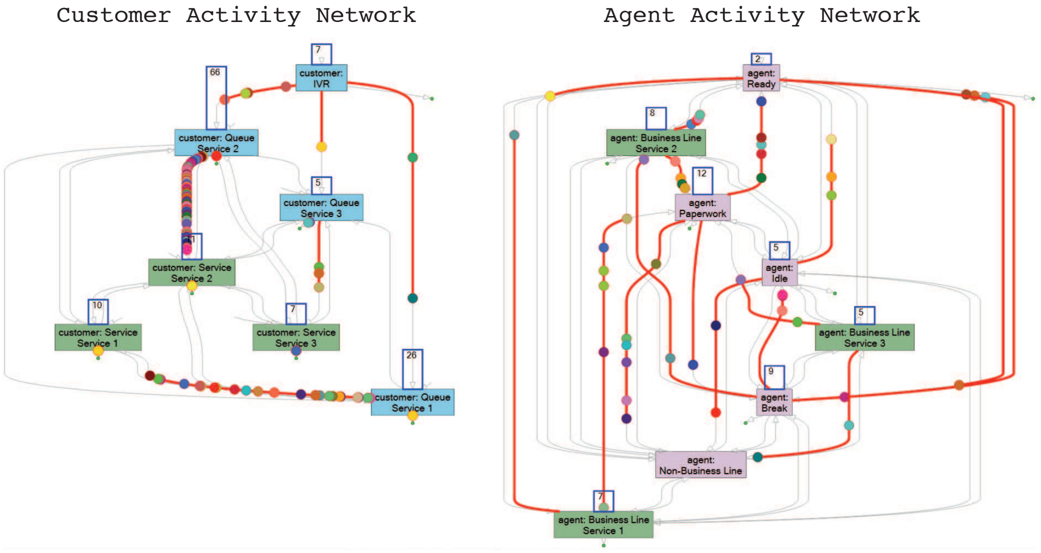
Research Plan

1. A brief description of the subject and the scientific and technological background

1.1. **Data Stories.** In the past, detailed (transaction-level) data collection in service systems was costly (e.g., via time studies). However, advances in information technology have enabled cost-effective collection of massive amounts of operational data. For example, real-time location systems (RTLS) are beginning to appear in hospitals and large outpatient clinics. This data availability is prompting development of novel approaches for analysis of service systems. In particular, the common top-down approach is replaced by the "bottom-up" alternative, where system-level behavior is extracted from low-level details recorded in massive amounts of raw data. While such amounts of data provide a detailed picture of the system, it is typically hard to make actionable conclusions from raw data. Therefore, it is of interest to develop a framework that can facilitate analysis and aggregation of raw data.

In Figure 1, we show two (time) snapshots of an activity network in a call center where customers and agents interact. The left graph shows the network for customers, while the right one for telephone agents. Rectangles in both graphs represent activities. Green rectangles correspond to activities that involve both customers and agents: the three rectangles on the left and the three rectangles on the right represent the same three activities. The remaining activities involve only customers (left) or agents (right). White vertical bars on top of activities represent the numbers of customers/agents involved in a particular activity at the given snapshot time. Directed edges between two activities indicate possible consumer/agent state transitions in the network. Individual customers and agents are shown with small discs. A disc moves between two activities while the corresponding customer/agent is engaged in the activity that corresponds to the egress node. The number of customers/agents engaged in an activity is given by the number of discs on its outgoing edges. An animation representing the time evolution of both networks can be found at CustomerServerNet. An additional animation, at ResourceNet, illustrates the same call center, but *both* customers and agents are shown in a *single* network: the latter is a data-based animation of a resource-driven activity network, which we now introduce.

In a call center, there exist two types of entities: customers and agents. In more complex service operations, multiple entities might exists, and all of them might need to be taken into account. For example, in a hospital, one might need to account not only for medical doctors and patients, but exam rooms, nurses and equipment as well. This is one of the main reasons to consider a unifying approach, based on resource symmetry, which allows for an arbitrary number of entities. In general, activity networks describing healthcare operations can have a daunting complexity. To demonstrate this point, in Figure 2, we show a (time) snapshot of a patient appointment network in a large, complex outpatient clinic that we are partnering with. Analogously to Figure 1, rectangles represent scheduled activities that involve patients, edges describe routes, and discs correspond to patients. We note that this is just a network for patients. Most of the shown activities require multiple resources. In order to understand the whole activity network, one needs to consider similar networks for



Customer Activity Network

Agent Activity Network

7
customer: IVR
66
customer: Queue Service 2
5
customer: Queue Service 3
11
customer: Service Service 2
10
customer: Service Service 1
7
customer: Service Service 3
26
customer: Queue Service 1
2
agent: Ready
8
agent: Business Line Service 2
12
agent: Paperwork
5
agent: Idle
5
agent: Business Line Service 3
9
agent: Break
agent: Non-Business Line
7
agent: Business Line Service 1

Figure 1. Two snapshots of an activity network in a call center. The left graph shows the network for customers, while the right one for agents; individual customers and agents are shown with discs on arcs. Both snapshots correspond to a call center where customers and agents interact. The snapshots are based on animations that can be viewed at CustomerServerNet and ResourceNet.

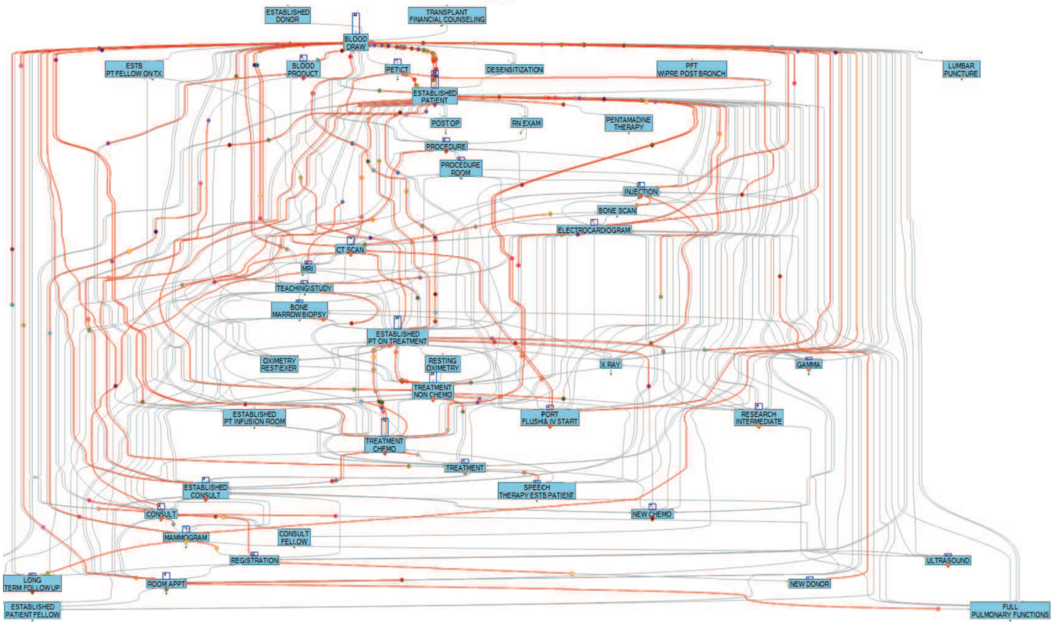


Figure 2. A snapshot of a patient appointment network in a complex outpatient clinic. Nodes (rectangles) correspond to scheduled activities, arcs describe patient routes. Patients are shown with discs on arcs.

all other entities in the system (medical doctors, nurses, equipment, rooms, etc.). Given the complexity of the overall network, we conclude that algorithmic tools for analysis of such networks are warranted. There exists an analogue of Figure 2 that corresponds to events that actually occurred in the clinic (data are obtained via about 1000 sensors of a real-time location (RTLS) system, at a resolution of 3 seconds).

Exploratory analyses of transaction-level data from a call center and a hospital can be found in [13] and [3], respectively. In [18], the authors use data to calibrate a queueing model (estimate its parameters). The approach in [7] is exactly the one advocated here: a symmetry between customers and servers; in their data-driven analysis, the authors first propose a queueing model that has features observed in data, and then they estimate parameters of a queueing model for customer-server interactions.

1.2. **Symmetry between customers and servers.** The data examples above underline a notion of symmetry or duality between servers and customers in queueing systems. For example, (i) server idleness may be interpreted as a server waiting for a customer, (ii) servers transferring between tasks may be thought of as “server networks”, and (iii) determining staffing levels is parallel to determining panel size. While, traditionally, the customer population in a queue is potentially large or infinite, the number of servers tends to be relatively small, so duality did not arise naturally. Moreover, conventionally, in heavy traffic, essentially all customers wait while servers are always busy [31, 32]. In contrast, in recent years, with the proliferation of information services (such as call centers, chat systems, server farms), a theory of many-server heavy traffic has been developed, in which significantly many servers wait for customers and similarly customers wait for servers [22, 19]. This symmetry has motivated us to come up with a unified framework of an *activity network*, in which customers and servers are considered simply as two different types of resources, and for an activity to occur the participation of a certain combination of resource types is required. (This framework is also natural to capture situations where more than one type of server is needed for a service to take place.) We next elaborate on some specific examples where the symmetry between customers and servers is apparent (in addition to models that are inherently symmetric, such as those considered in [47, 1, 39, 43]).

Offered load versus offered capacity. The concept of *offered load* has been discussed in the literature [51]. In a nutshell, offered load refers to the (average) number of customers in a system when capacity constraints are removed. Offered load has been used to advise on determining staffing levels [11, 51]. Analogously, one might look at a system that has unlimited demand and thus measure the number of busy servers [37, 20]. We refer to this measure as the system *offered capacity*, and comment that it may be used to help determine optimal panel sizes. In our unified framework, the notion of offered load refers to either offered load and offered capacity, which is consistent with treating both customers and servers as resources.

QED-regime characterization. The quality-and-efficiency driven (QED) regime (originally conceived as the Halfin-Whitt regime) is a many-server heavy-traffic asymptotic operational regime that has been extensively used to model customer contact centers and recently also healthcare systems. Traditionally, this regime has been characterized (within a specific mathematical setting) as the unique heavy-traffic regime in which the probability of a *customer being delayed* before service-start is strictly between 0 and 1. As it turns out, an equivalent characterization is given by the QED regime being the unique heavy traffic regime in which the probability that upon service completion the *server becomes idle* (waits for customers) is strictly between 0 and 1.

Asymptotic ASTA and Little's law. Little's law is the well known congestion law according to which, in steady-state, the mean queue length is equal to the arrival rate times the mean waiting time. In heavy-traffic, a similar relationship has been established with respect to corresponding transient performance measures, due to the snapshot principle [45, 46]. As was shown in [49], a similar relationship applies to the number of idle servers and their idle time; [49] also establishes a server analogue of the PASTA property. According to the PASTA property [52], customers who arrive to a system, according to a Poisson process experience, on average, the same system state as an overall long-term time-average. Analogously, [49] established an asymptotic server "ASTA" property, according to which, servers, upon service completion, experience on average the same system state as an overall long term time average, as long as service times are exponential. The results mentioned here relate to classical input-output theorems for queueing networks, e.g. see [48].

Skill-based routing and state-space collapse. In call centers, skill-based routing (SBR) refers to routing of customers of multiple types to servers of multiple skills. An alternate view of SBR could be the scheduling of servers of multiple skills to customers of multiple types. More generally, SBR is the process of matching between customers and servers. This symmetric point of view is apparent in skill-based routing schemes such as QIR [21] (queue and idleness ratio) and LISF [5] (last idle server first). Under the QIR policy, servers are assigned to serve a queue so as to maintain a certain ratio between the various queue lengths, while customers are assigned to server pools to maintain a certain ratio between the number of idle servers in the various server pools. In [21], the authors establish state-space collapse results with respect to this policy which show that, in the limit, the queues and the idleness processes indeed remain at their desired ratios. In a multi-server system, the LISF policy assigns the next customer to the server that has been idle the longest. This parallels a FIFO discipline for a queue of customers. In [5], it is shown that LISF maintains asymptotic fairness among servers, which is consistent with the view of FIFO as a fair policy towards customers [6].

Closed queueing systems. In standard open queueing systems, there exists an inherent asymmetry between servers and customers: customers spend a limited time in the system, while servers remain in the system forever. Hence, in order to develop a completely symmetric unifying approach, we have been naturally led to consider a closed model, where both customers and servers remain in the system forever. Note that this is without loss of generality since open models can be viewed as closed (the outside "world" can be thought of as a node in the network). Typically, closed queueing systems are harder to analyze than their open counterparts [50]. Our project focuses on developing tools for the analysis of closed systems. As demonstrated in [53], methods that do not take into account the repetitive nature of service (such as the Piecewise Stationary Approximation [33], often used to analyze time-varying queues) may lead to poor results in closed systems. Examples of heavy-traffic limit theorems, for closed queueing systems with a fixed number of servers, can be found in [36, 27, 28, 29, 38, 2]. Closed many-server systems with state-dependent drifts are explored in [41]. QED analyses of the machine repair model, a "closed" analogue of the standard many-server model, appear in [17, 42]. Additional closed many-server models were considered in [44, 8].

1.3. **Fluid Networks.** In fluid networks, the flow through the system is continuous and deterministic and work is infinitely divisible. Fluid networks are helpful in modeling the first order effects of processing networks and may be obtained as a limit of a sequence of queueing networks following functional strong law of large numbers. Fluid networks have been used extensively in the literature to study system stability [15, 16], to assist in capacity planning [30, 10], and establish state-space collapse results [12]. Fluid models have been shown to be a particularly meaningful modeling tool, which captures the predictable variability of systems [33, 53, 14].

While in conventional heavy traffic, fluid networks necessarily depend on their corresponding queueing network via the first moment of the relevant distribution, the situation is significantly different for the many-server heavy-traffic regime. Specifically, it has been established by [34, 35, 40] that, in heavy-traffic, the fluid limit depends on the queueing model through the entire service-time and time-to-abandon distribution. This type of limit, often referred to as measure-valued, has been useful in establishing results such as the accuracy of delay announcements in call-centers [4] and optimal scheduling of customers [9]; moreover, a measure-valued description is natural for capturing the level of details that exists in data-rich environments.

In the initial phase of this work, we are planning to use a fluid framework for our activity-network model. As will be demonstrated, this enables one to capture the key features of the network without compromising tractability. For example, while the fluid model does not capture specifics of routing schemes, it does capture the first order proportional split of traffic that is the result of routing. Our framework of fluid-activity-network allows one to assist in decisions that pertain to the design, planning and control of the system.

Our framework relates to and amply draws from stochastic processing networks (SPNs) [23, 24, 26] and their corresponding activity analysis [25]. However, there exist some fundamental differences, which we now explain. In SPNs, resources (servers) interact with customers (materials) via activities. Materials arrive to the network at specifies arrival rates and traverse the network according to some specified rules. Resource pools can be thought of as single server queues with high processing capacities – individual service times are negligibly short, as is typically assumed when considering the conventional heavy traffic regime (due to a time speedup). As a result, for example, non-bottleneck nodes can be removed from the network for analysis purposes. In contrast, our model is based on a many-server regime, implying that activity durations are characterized by proper probability distributions (there is no time speedup). As a consequence, all activities (nodes) in the network must be considered, since one cannot focus on bottleneck nodes only as it is the case in SPNs. Finally, in SPNs, constraints are due to finiteness of processing *capacities* of resources (servers). On the other hand, constraints in our models must be based on limited *amounts* (numbers) of resources (both servers and customers) in the network.

2. Objectives and significance of the research

Our research agenda has been motivated by the increasing availability of transaction-level data from service systems. The overall goal is to create data-driven models that can be used to improve operations of such systems. The specific items we plan to develop are:

- *A unified framework based on the symmetry of all entities in the system.* Earlier approaches focused typically on one type of entities in the system (e.g., customers or servers). In contrast, our framework considers all entities in the system simultaneously, since multiple entities can be needed to conduct a single activity. For example, in order for a medical exam to take a place, a medical provider, a patient, an exam room and some equipment must be all available simultaneously.
- *An algorithm for generating and analyzing data-based models.* Even service systems of a moderate size can generate vast amounts of transaction-level data. Thus, tools for automatically translating data into models are of interest. Our framework is particularly suitable for this task, since it is based on a "bottom-up" approach – transaction-level data describe the basic elements of our model (activities).
- *A methodology for addressing design/staffing/control problems.* The models developed will be particularly suitable for addressing such challenges. For example, our framework allows one to generalize the notion of offered load/capacity.

A detailed description of our methodology is provided next.

3. Comprehensive description of the methodology and plan of operation, including respective roles of the Israeli and American principal investigators

Plan of operation: All parts of the research agenda will be investigated jointly by researchers from the Israeli and American side. Mutual visits will facilitate this cooperation. Students working with the PIs, as well as the SEE Laboratory staff, will be expected to participate in research activities.

We now provide an overview of our research approach with some specific examples that illustrate the main ideas.

3.1. **Fluid Activity Network: The Static Model.** Consider a closed system comprised of n different resource pools and m different activities. Resources engage in activities. A unit of resource can be in several states (one at a time) – we use the notion of *sub-resource* to describe the resource-state pair. Let k be the number of sub-resources in the system. Any two units of a specific sub-resource are interchangeable. An $n \times k$ matrix R with values in $\{0, 1\}$ describes the relationship between resources and sub-resources. In particular, $R_{i,l} = 1$ whenever sub-resource l corresponds to resource i. Model primitives include also an n-vector b of resource amounts and an m-vector a of mean activity durations; the element a_j represents the mean duration of activity j. We define $A := \operatorname{diag}(a)$ for notational convenience. Initially, we assume that a unit of resource may correspond to a unit of one sub-resource at a time. We refer to this assumption as a one-to-one correspondence. We shall relax this assumption later (Example 4). Activities are described also by two nonnegative $k \times m$ matrices: an input (consumption) matrix C and an output (production) matrix P. The element $C_{l,j}$ defines the amount of sub-resources l required to be engaged in activity j. Similarly, $P_{l,j}$ specifies the amount of sub-resources l created upon completion of activity j. Given the one-to-one correspondence assumption, we require that $RC = RP$ (conservation of resources at each activity).

A plan x, an m-vector of activity levels, is feasible if

$$(1) \qquad RCAx \leq b, \quad (C - P)x = 0 \quad \text{and} \quad x \geq 0.$$

It is appropriate to think of x as a vector of rates at which activities are being conducted (thus the constraint $x \geq 0$). The vector CAx represents the total amount of sub-resources engaged in activities under x; $RCAx$ indicates the corresponding amount of required resources. The equality $(C-P)x = 0$ represents flow conservation constraints. The feasibility condition (1) is very similar to the corresponding condition for activity networks [25, (2.1)]. The fundamental difference is that the bound in (1) is in terms of resource counts (amounts) rather than processing rates as in [25]; in our framework, processing rates are assigned to activities, not sub-resources. We remark that additional constraints for x might be imposed due to system specific dynamics, e.g., routing as illustrated in Example 2.

Example 1 (Machine repair). Consider the standard machine repair model: the resource pools 1 and 2 correspond to repairmen and machines, respectively (there are b_1 repairmen and b_2 machines). There exist two activities ($1 =$ machine being repaired, $2 =$ machine working) and three sub-resources ($1 =$ repairman, $2 =$ broken machine, $3 =$ working machine) – see Figure 3. Let $1/\mu$ and $1/\lambda$ be the average repair and working times for a machine, respectively. One has

$$(2) \qquad R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \quad A = \begin{bmatrix} 1/\mu & 0 \\ 0 & 1/\lambda \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad P = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix},$$

since activity 1 consumes a broken machine and a repairmen (sub-resource 1) to produce a working machine and a repairmen; and activity 2 "creates" a broken machine (sub-resource 2) from a working machine (sub-resource 3). Therefore, in view of (1), a plan x with $x_1 = x_2 \geq 0$ is feasible if

$$\begin{bmatrix} 1/\mu & 0 \\ 1/\mu & 1/\lambda \end{bmatrix} x \leq b,$$

or equivalently $0 \leq x_1 = x_2 \leq \mu b_1 \wedge \frac{\lambda\mu}{\lambda+\mu} b_2$. □

Remark 1 (Open systems). An open system with exogenous arrivals and departures may be obtained as a limit of a sequence of closed systems. In particular, suppose that units of resource i arrive to the system at rate λ_i. Then, an activity exclusively involving resource i can be used to representing an external source. All sub-resources corresponding to resource i

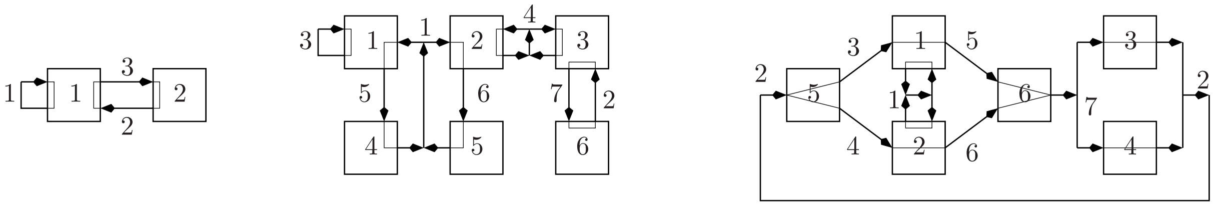


FIGURE 3. Graphical representations of the activity networks described in Example 1 (left), Example 3 (center), and Example 4 (right). Boxes represent activities, and arrows indicate flows of sub-resources between activities. Lines within boxes indicate correspondence between input and output sub-resources.

that leave the system are routed to this activity. Upon completing the activity (sub)-resources enter the system. The mean activity time is set to γ_i, while the amount of resource i is set to $\lambda_i\gamma_i$. Then, in the limit, as $\gamma_i \to \infty$, resource i enters the system with rate λ_i. For example, consider a fluid many-server open system: the number (amount) of servers (resource 1) is b, customers (resource 2) arrive at rate λ, and the mean service time is $1/\mu$ (activity 1). In order to create a closed system, activity 2 is introduced: its output is a customer that is ready to receive service (sub-resource 2), and its input is a customer that completes service (sub-resource 3). Then, $a = [1/\mu \quad \gamma]^\top$; R, C, P are as in (2); and (1) yields that x with $x_1 = x_2 \geq 0$ is feasible if

$$\begin{bmatrix} 1/\mu & 0 \\ 1/\mu & \gamma \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} b \\ \lambda\gamma \end{bmatrix}.$$

Letting $\gamma \to \infty$ results in $0 \leq x_1 \leq \lambda \wedge \mu b$, i.e., the total service rate cannot be higher than the arrival rate or the total processing capacity. □

Example 2 (Closed Jackson network). Consider a fluid version of a Jackson network with m stations. The number (amount) of servers at node i is b_i and the mean service time is a_i. There are $(m+1)$ (sub)-resources in the system: resources $1, \dots, m$ correspond to servers at different nodes, and the $(m+1)$st resource corresponds to customers. The amount of customers in the system is b_{m+1}. Matrix R is an identity matrix. There are m activities in the system – the ith activity represents service of customers at node i. Based on the preceding description, one has $C = P = [I \;\; e]^\top$, where e is a vector of ones and I is an identity matrix (activity i consumes/produces a type-i server and a customer). Routing of customers among the nodes is described by a stochastic matrix Π. This yields an additional constraint: $\Pi x = x$. Hence, a plan x is feasible if

$$a_i x_i \leq b_i, \;\; i = 1, \dots, m, \quad a^\top x \leq b_{m+1}, \quad x \geq 0, \quad \Pi x = x. \quad \square \tag{3}$$

The conditions in (1) characterize the set of feasible plans x. A number of optimization problems with respect to this region arise naturally. The most straightforward one is to maximize $v^\top x$ (subject to (1)), where v is an m-vector of value rates associated with the various activities; this assumes that b is given. For example, for the closed Jackson network of Example 2, solving this optimization problem with respect to any non-zero $v \geq 0$, will result in at least one binding constraint in (3), and if $a_i x_i = b_i$ node i is a bottleneck, while $a^\top x = b_{m+1}$ implies that no customer is waiting to receive service.

Offered load/rates. Another natural problem is that of determining offered load. As opposed to the traditional one-dimensional notion of offered load, in our context the offered load may be multi-dimensional, and is dependent on the subset of resources one wishes to focus on. For $\mathcal{S} \subseteq \{1, 2, \ldots, n\}$, finding the rate offered by the set $\mathcal{S}$ corresponds to finding the maximal activity rates that are feasible under the constraint that the capacities of the resources in $\mathcal{S}$ are given. Similarly, finding the load offered by $\mathcal{S}$ corresponds to finding the minimal resource levels that would permit the offered rate to be a feasible plan. To this end, given a vector of capacities b, define $b(\mathcal{S})$ by $b_i(\mathcal{S}) = b_i$, for $i \in \mathcal{S}$, and $b_i(\mathcal{S}) = \infty$, for $i \notin \mathcal{S}$. That is, $b(\mathcal{S})$ is obtained from b by removing any capacity bounds for resources that are outside the set $\mathcal{S}$. Let $\mathcal{X}(\mathcal{S})$ be the set of maximal m-vectors x such that (1) holds with b replaced with $b(\mathcal{S})$. That is $\mathcal{X}(\mathcal{S})$ is the set of offered rates. In addition, for $x \in \mathcal{X}(\mathcal{S})$, we introduce a set $\mathcal{B}(x, \mathcal{S})$ as a set of minimal n-vectors b' such that $b' = RCAy$ for some m-vector y with $(RCAy)_i = (RCAx)_i$, $i \in \mathcal{S}$. $\mathcal{B}(x, \mathcal{S})$ is then the offered load associated with offered rates x. One of our goals is to formulate a procedure for automatically calculating these offered rates and loads; subsequently, they will be used to advise on capacity planning and control.

Example 3 (Closed N-system). Consider an N-system with two customer pools (resources 1 and 2) and two server pools (resources 3 and 4). This example is depicted in Figure 3 as an activity network with 4 resources, 7 sub-resources and 6 activities. The first server pool (resource 3, sub-resource 3) can serve only the first pool of customers (resource 1) with unit rate – this corresponds to the activity 1. The second server pool (resource 4, sub-resource 4) can serve both classes of customers – activities 2 and 3 (the corresponding rates are 0.5 and 1). Upon a service completion, a customer enters an orbit. The mean time spent in the orbit depends on the customer-server pair. In particular, after activities 1, 2 and 3, a customer becomes sub-resource 5, 6 or 7, and enters activity 4, 5 or 6, respectively, with corresponding mean durations 1, 0.5 and 1. Finally, sub-resources 1 and 2 correspond to pool one and two customers that have completed their orbits, respectively. Thus, the system is described by $a = [1 \quad 0.5 \quad 1 \quad 1 \quad 0.5 \quad 1]^\top$,

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad P = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix};$$

$RC = RP$ holds. For the pair of server pools $(\{3,4\})$, we have

$$\mathcal{X}(\{3,4\}) = \mathrm{Conv}\langle \begin{bmatrix} b_3 \\ 0 \\ b_4 \\ b_3 \\ 0 \\ b_4 \end{bmatrix}, \begin{bmatrix} b_3 \\ \frac{1}{2}b_4 \\ 0 \\ b_3 \\ \frac{1}{2}b_4 \\ 0 \end{bmatrix} \rangle \quad \text{and} \quad \mathcal{B}(\begin{bmatrix} b_3 \\ \frac{1}{2}b_4 \\ 0 \\ b_3 \\ \frac{1}{2}b_4 \\ 0 \end{bmatrix}, \{3,4\}) = \{ \begin{bmatrix} 2b_3 + \frac{5}{4}b_4 \\ 0 \\ b_3 \\ b_4 \end{bmatrix} \},$$

where Conv denotes the convex hull. The set $\mathcal{X}(\{3,4\})$ is relatively simple due to the fact that no activity involves servers from the two pools. In the case when one considers the pair $\{2,4\}$, note that activity 3 involves both resources 2 and 4; while, resource 2 can engage in activity 3 only. Hence, one has

$$\mathcal{X}(\{2,4\}) = \mathrm{Conv}\langle \begin{bmatrix} \infty \\ \frac{1}{2}(b_4 - \frac{1}{2}b_2)^+ \\ \frac{1}{2}b_2 \wedge b_4 \\ \infty \\ \frac{1}{2}(b_4 - \frac{1}{2}b_2)^+ \\ \frac{1}{2}b_2 \wedge b_4 \end{bmatrix}, \begin{bmatrix} \infty \\ \frac{1}{2}b_4 \\ 0 \\ \infty \\ \frac{1}{2}b_4 \\ 0 \end{bmatrix} \rangle \quad \text{and} \quad \mathcal{B}(\begin{bmatrix} \infty \\ \frac{1}{2}b_4 \\ 0 \\ \infty \\ \frac{1}{2}b_4 \\ 0 \end{bmatrix}, \{2,4\}) = \{ \begin{bmatrix} \frac{5}{4}b_4 \\ 0 \\ 0 \\ b_4 \end{bmatrix} \}.$$

□

Remark 2 (Conventional heavy-traffic regime). Our framework stems from a many-server regime: many activities of the same type take place simultaneously (with non-negligible durations in general). In some cases, a system might involve some resources that indeed operate in a many-server regime (such as beds in a hospital ward), while others operate in conventional heavy-traffic (such as physicians in the same ward; see [3]); By considering a limit of a systems in a many-server regime, one can obtain a characterization of a system in which some or all activities/resources operate in the conventional heavy-traffic regime (or a single-server mode in general). In the conventional heavy-traffic regime, a "processing" capacity of a resource is large, but only one activity takes place at a given time (in the limit activity durations tend to 0). Results in [25] cover the case when all activities operate a single-server mode. In the mixed-regime case, for all sub-resources corresponding to resources not operating in the many server regime, we let the corresponding amounts (b_i's) decrease to 0; durations of all activities these sub-resources are involved in also decrease to 0 at the same rate.

For example, consider the setup described in Example 3. Now suppose that the second pool of servers (sub-resource 4) is replaced with a single server. In order to model this system, we let γb_4 be the amount of sub-resource 4; with $\gamma \to 0$. Durations of all activities that involve sub-resource 4 are also scaled by γ, leading to $a = [1 \quad 2\gamma \quad \gamma \quad 1 \quad 0.5 \quad 1]^\top$. Here, b_4 should be interpreted as the amount of sub-resource 4 capacity that is available per time unit; $a_2/\gamma = 2$ and $a_3/\gamma = 1$ stand for rates at which activities 2 and 3 consume the capacity of sub-resource 4. After letting $\gamma \to 0$, (1) implies $x_1 = x_4 \geq 0$, $x_2 = x_5 \geq 0$, $x_3 = x_6 \geq 0$ and $2x_1 + x_2/2 \leq b_1$, $x_3 \leq b_2$, $x_1 \leq b_3$, $2x_2 + x_3 \leq b_4$. □

A given unit of sub-resource can either be engaged in an activity or awaiting to be engaged in an activity. Recall that $(RCAx)_i$ units of resource i (out of b_i units) are engaged in activities in the network. Given a plan x that satisfies (1), we say that resource i is a bottleneck resource if the ith inequality in $RCAx \leq b$ is binding, i.e., $(RCAx)_i = b_i$ (recall the bottleneck resources for the Closed Jackson Network example above). In addition, all sub-resources that correspond to resource i (that is, a sub-resource j such that $R_{i,j} = 1$) are termed bottleneck sub-resources. Moreover, all activities that require bottleneck sub-resources are labeled as bottleneck activities. Let z be a k-vector describing the amount of sub-resources in the network. then $Rz = b$. Since CAx sub-resources are engaged in activities, $z \geq CAx$ and $(z - CAx)$ resources are awaiting to be to engaged in activities. Note that components of z are uniquely defined only for bottleneck sub-resources.

In the preceding, we considered only activities that preserve a one-to-one correspondence between resources and sub-resources. Next, we examine activity networks that include activities that do not preserve such a correspondence. In such networks, multiple units of different sub-resources can correspond to a single unit of multiple resource. To illustrate consider the following fork-join example.

Example 4 (Fork-join). Consider a variation of the earlier discussed machine-repair model. Recall that there are b_1 repairmen (resource 1, sub-resource 1) and b_2 machines (resource 2). A broken machine (sub-resource 2) is disassembled (activity 5) into two parts (sub-resources 3 and 4) that both need to be repaired separately (activities 1 and 2; these activities require two and a single repairman, respectively) – see Figure 3. Repaired parts (sub-resources 5 and 6) are assembled (activity 6) into working machines (sub-resource 7). Repaired machines work in one of two modes of operation (activities 3 and 4). The mean durations of the activities is given by $a = [1 \quad 2 \quad 1 \quad 2 \quad 0 \quad 0]^\top$; input-output relations for each activity are captured by

$$C = \begin{bmatrix} 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \quad \text{and} \quad P = \begin{bmatrix} 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Then, $(C - P)x = 0$ is equivalent to $x_1 = x_2 = x_3 + x_4 = x_5 = x_6$. At the same time, describing relations between (sub)-resources and resources with a binary matrix R is not sufficient.

In order to capture relationships between sub-resources within individual activities, we introduce a set of nonnegative $k \times k$ *parametric* matrices $\{T(i)\}$, where for each i this matrix describes how activity i transforms input sub-resources into output sub-resources. For an activity i and one of its input sub-resources j, let $\mathcal{O}(i, j)$ be the sets of activity i output sub-resources that correspond to the sub-resource j; let $\mathcal{I}(i)$ be the set of input sub-resources for activity i. For $l \notin \cup_{j \in \mathcal{I}(i)} \mathcal{O}(i, j)$, we set $T_{l,r}(i) = 1_{\{l=r\}}$; for $l \in \cup_{j \in \mathcal{I}(i)} \mathcal{O}(i, j)$, we let $T_{l,r}(i) = \beta^i_{l,r} \in [0, 1]$ if $l \in \mathcal{O}(i, r)$, and $T_{l,r}(i) = 0$ otherwise. The parameters $\{\beta^i_{l,r}\}$ satisfy, for all i, the following flow-conservation identity:

$$C_{:,i} = P^\top_{:,i} T(i); \tag{4}$$

If there exists a one-to-one correspondence between input and output sub-resources, then the corresponding β is set to 1. For example, for our fork-join example, one has

$$T(5) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \beta^5_{3,2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \beta^5_{4,2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad T(6) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix},$$

where $\beta^5_{3,2} + \beta^5_{4,2} = 1$ due to (4).

In the non one-to-one setting, the matrix R that relates resources and sub-resources satisfies, for all i,

$$T(i)\, R^\top = R^\top. \tag{5}$$

In general, both $\{T(i)\}$ and R in parametric form – this allows one to distinguish among different sub-resources that constitute sub-parts of a resource. In our fork-join example the matrix R has the following form:

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \beta^5_{3,2} & \beta^5_{4,2} & \beta^5_{3,2} & \beta^5_{4,2} & 1 \end{bmatrix};$$

the second row indicates that a "union" of units of sub-resources 3 and 4 (or 5 and 6) corresponds to a unit of resource 2.

Let $\mathcal{R}$ be the set of all possible values of R (under (4) and (5)), and let $\{R_l\}$ be the set of extreme points of $\mathcal{R}$. Then, $RCAx \le b$ in (1) should be replaced with

$$\bigvee_l R_l CAx \le b, \quad (C-P)x = 0 \quad \text{and} \quad x \ge 0, \tag{6}$$

where $\vee$ indicates that the inequality holds for all l. The first inequality ensures that a plan x does not require more resources than that are available, regardless of whether resources are decomposed into parts; this inequality can be rewritten as $\tilde{R}CAx \le \tilde{b}$ by eliminating redundant inequalities. In our fork-join example, $\mathcal{R}$ has two extreme points, and

$$\tilde{R}CAx = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} CAx \le \tilde{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_2 \end{bmatrix}$$

reduces to $4x_3 + 4x_4 \le b_1$, $3x_3 + 4x_4 \le b_2$. Finally, by setting $b_2 = \infty$ or $b_1 = \infty$ yields the offered rates

$$\mathcal{X}(\{1\}) = \frac{b_1}{4}\mathrm{Conv}\langle \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} \rangle \quad \text{and} \quad \mathcal{X}(\{2\}) = b_2\,\mathrm{Conv}\langle \frac{1}{4}\begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \frac{1}{3}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} \rangle,$$

respectively.

In the absence of one-to-one correspondence between sub-resources and resources, there exist additional constraints on the amount of sub-resources $z \geq CAx$ in the activity network. Informally, the number of parts that form a whole must match. Recall that the components of z that correspond to bottleneck sub-resources are uniquely defined: $z_i = (CAx)_i$. For $\beta^i_{l,r}$ and $\beta^i_{q,r}$ that are not identically equal to 0 or 1, vector z satisfies

$$I(\beta^i_{l,r})z = I(\beta^i_{q,r})z,$$

where $I(\beta)$ is a $n \times k$ binary matrix such that $I(\beta)_{i,j} = 1$ if the value of $R_{i,j}$ depends on the value of β, and 0 otherwise. In Example 4, this amounts to $z_3 + z_5 = z_4 + z_6$. □

Aggregation. We note that the activity networks described in Examples 1 and 4 can serve as models for the same physical process – repairmen repairing machines that break down. It is appropriate to think of activity 1 (respectively 2) in Example 1 as an activity that aggregates activities 1, 2, 5 and 6 (respectively 3 and 4) in Example 4. That is, activity 1 in Example 1 models the repair activity without going into details of the process. This implies that the repair process is fixed and cannot be altered. In the case when the repair process is found to be inadequate, its detailed structure is required (e.g. Example 4) in order to understand its limitations. For example, combining activities 3 and 4 in Example 4 into an activity 2 in Example 1 assumes that the routing of repaired machines to activities 3 and 4 is fixed. In a data-driven approach, the choice of a model is determined by the granularity of available data. However, one can also aggregate data if having a coarser model is desirable. Such models can be used to identify bottlenecks and understand interactions between functionally independent parts of activity networks. Our goal is to develop an algorithm for automatically generating system models from transaction-level data with a desired level of granularity.

3.2. **Fluid Activity Network: The Dynamic Model.** Our framework thus far was focused on a static fixed-point description, but did not explore how the system dynamics evolve over time. We now extend our framework to include an explicit time dimension. This extended framework is essential for modeling and analyzing systems that evolve in a time varying fashion, which is indeed very common. Let $G(t)$ and $X(t)$ be m-vectors: $G_i(t)$ is the distribution function of the ith activity (non-negative) duration (set $\bar{G}_i := 1 - G_i$), and $X_i(t)$ represents the amount (number) of activity i that starts in the time interval $[0, t]$; define $X(t) = 0$ for $t < 0$. Then, a_i is the first moment of G_i, and $X_i(\cdot)$ is non-decreasing. We focus on the total number of activities started rather than the instantaneous rate at which activities start, because $X(\cdot)$ can be discontinuous. Let $E(t)$ be a non-negative k-vector that represents the amount of sub-resources not engaged in activities at time $t = 0$. In addition, one needs to describe the state of the system at time $t = 0$. To this end, let $V(t)$ be an m-vector such that $V_i(t)$ represents the amount of i activity that is in progress at time 0 and are completed by time t. This implies that $V(\infty)$ describes the amount of activities in progress at time $t = 0$; set $\bar{V}(t) := V(\infty) - V(t)$ and, without loss of generality $V(0) = 0$. In addition, let e be a non-negative k-vector that represents the amount of sub-resources not engaged in activities just before time $t = 0$. The initial conditions obey $\tilde{R}(CV(\infty) + e) = \tilde{b}$.

Given that $X(t)$ activities are started in the time interval $[0,t]$, $G*X(t)$ of those activities end by time t; here

$$(7)\quad (G*X)_i(t) = G_i(0)X_i(t) + \int_0^t X_i(t-s)\,\mathrm{d}G_i(s) = G_i(t)X_i(0) + \int_0^t G_i(t-s)\,\mathrm{d}X_i(s).$$

Therefore, the total number of completed activities in $[0,t]$ is given by $(G*X+V)(t)$, while the amount of activities in progress is $(\bar{G}*X+\bar{V})(t)$. A dynamic plan $X(\cdot)$ is feasible if

$$(8)\ \tilde{R}C(\bar{G}*X+\bar{V})(t) \le \tilde{b}, \quad CX(t) \le P(G*X+V)(t)+e \quad \text{and} \quad X(\cdot) \text{ is non-decreasing.}$$

The second inequality states that the amount of consumed sub-resources cannot be higher than the amount of available sub-resources. The amount of sub-resources in the network not engaged in activities is given by $E(t) = P(G*X+V)(t) - CX(t) + e$, and hence the second inequality in (8) is equivalent to $E(t) \ge 0$.

Example 5 (Machine repair). We revisit Example 1. Suppose that repair and working times are exponentially distributed: $\bar{G}_1(t) = e^{-\mu t}$ and $\bar{G}_2(t) = e^{-\lambda t}$, $t \ge 0$. If $V_i(t) = 0$ (no activities are in progress at time $t=0$), then the initial conditions satisfy $e_1 = b_1$ and $e_2 + e_3 = b_2$. In view of (8), a non-decreasing X is a feasible dynamic plan provided that the following inequalities hold:

$$E_1(t) = \mu \int_0^t X_1(s)e^{-\mu(t-s)}\,\mathrm{d}s - X_1(t) + b_1 \ge 0,$$
$$E_2(t) = \lambda \int_0^t X_2(s)e^{-\lambda(t-s)}\,\mathrm{d}s - X_1(t) + e_2 \ge 0,$$
$$E_3(t) = \mu \int_0^t X_1(s)e^{-\mu(t-s)}\,\mathrm{d}s - X_2(t) + e_3 \ge 0.$$

The first two inequalities restrict the amount of activity 1 due to lack of repairmen and broken machines, respectively; the third inequality bounds the amount of activity 2 based on lack of working machines. Note that, for $t=0$, one has $X_1(0) \le b_1 \wedge e_2$ and $X_2(0) \le e_3$ as expected.

In this example, a dynamic plan that maximizes $X_2(T)$, for a fixed $T > 0$, is such that $E_1(t)E_2(t) = 0$ and $E_3(t) = 0$, for $t \in [0,T]$. This leads to (assuming $\lambda \ne \mu$)

$$X_2(t) = \mu \int_0^t X_1(s)e^{-\mu(t-s)}\,ds + e_3,$$
$$X_1(t) = \min\left\{\mu \int_0^t X_1(s)e^{-\mu(t-s)}\,\mathrm{d}s + b_1,\ \frac{\lambda\mu}{\lambda-\mu}\int_0^t X_1(s)\left(e^{-\mu(t-s)} - e^{-\lambda(t-s)}\right)\mathrm{d}s + b_2 - e_3e^{-\lambda t}\right\}.$$

In Figure 4, we plot $X_1(t)$ and $X_2(t)$ (along with their derivatives) that solve the preceding system of equations, under the following choice of parameters: $\lambda = 1$, $\mu = 2$, $b = [1\ 4.5]^\top$, $e = [b_1\ 0\ b_2]^\top$ (at time $t=0$ all machines are working). Given these parameters, sub-resource 1 (repairmen) is a bottleneck, limiting the maximum rate at which the two activities occur to 2 (see Example 1). □

Time varying behavior. In open networks (systems), time-varying performance can be induced by time-varying arrival rates. Since there are no external arrival streams of

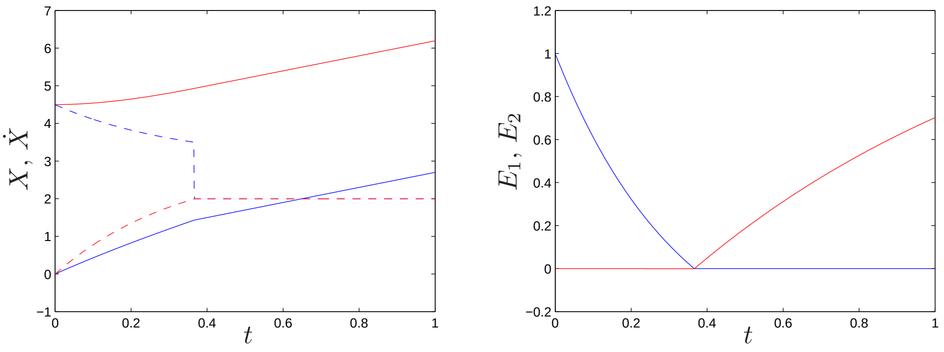


Figure 4. Performance functions for the system described in Example 5. Left: $X_1(t)$ (blue) and $X_2(t)$ (red) are shown with solid lines; the corresponding derivatives are shown with dashed lines. Right: $E_1(t)$ (blue) and $E_2(t)$ (red); $E_2(\infty) = 1.5$.

(sub)-resources in closed activity networks, we consider a system where activity durations are time-dependent. In particular, let G^τ be a vector of distribution functions of activity durations that start at time $t = \tau \geq 0$. Then, (7) generalizes to

$$(G^\tau * X)_i(t) = G_i^t(0)X_i(t) + \int_0^t X_i(t-s)\,\mathrm{d}G_i^{t-s}(s) = G_i^0(t)X_i(0) + \int_0^t G_i^s(t-s)\,\mathrm{d}X_i(s),$$

while (8) generalizes to

$$(9)\quad \tilde{R}C(\bar{G}^\tau * X + \bar{V})(t) \leq \tilde{b}, \quad CX(t) \leq P(G^\tau * X + V)(t) + e \quad \text{and} \quad X(\cdot) \text{ is non-decreasing,}$$

where $\bar{G}_i^\tau = 1 - G_i^\tau$; as before, the initial conditions obey $\tilde{R}(CV(\infty) + e) = \tilde{b}$. Similarly, $E(t) = P(G^\tau * X + V)(t) - CX(t) + e \geq 0$.

Due to the space constraint, we omit further details. We just remark that the concepts discussed earlier (offered load/rates, bottlenecks, fork-join) extend to cover this dynamic setting.

4. An account of available U.S. and Israeli resources, including all personnel and equipment relevant to the research

The research personnel includes the principal investigators, their graduate students, and possibly other research associates at the respective institutions. Standard computing, word processing and communication facilities are available to all investigators.

The Service Enterprise Engineering (SEE) Laboratory at the Technion will provide the infrastructure for data collection and analysis required in the present research. The laboratory personnel have an extensive background in transaction-level data analysis, as well as software development.

References

[1] I. Adan and G. Weiss. Exact FCFS matching rates for two infinite muti-type sequences. *Oper. Res.*, 60(2):475–489, 2012. 1.2

[2] J. Anselmi, B. D'Auria, and N. Walton. Closed queueing networks under congestion: Nonbottleneck independence and bottleneck convergence. *Math. Oper. Res.*, 38(3):469–491, 2013. 1.2

[3] M. Armony, S. Israelit, A. Mandelbaum, Y. Marmor, Y. Tseytlin, and G. Yom-Tov. Patient flow in hospitals: A data-based queueing-science perspective. Preprint. 1.1, 2

[4] M. Armony, N. Shimkin, and W. Whitt. The impact of delay announcements in many-server queues with abandonment. *Oper. Res.*, 57(1):66–81, 2009. 1.3

[5] R. Atar. Central limit theorem for a many-server queue with random service rates. *Ann. Appl. Probab.*, 18(4):1548–1568, 2008. 1.2

[6] B. Avi-Itzhak and H. Levy. On measuring fairness in queues. *Adv. Appl. Probab.*, 36(3):919–936, 2004. 1.2

[7] D. Azriel, P. Feigin, and A. Mandelbaum. Erlang-S: A data-based model of servers in queueing networks. Preprint. 1.1

[8] A. Bassamboo, S. Kumar, and R. Randhawa. Dynamics of new product introduction in closed rental systems. *Oper. Res.*, 57(6):1347–1359, 2009. 1.2

[9] A. Bassamboo and R. Randhawa. Scheduling homogeneous impatient customers. Preprint. 1.3

[10] A. Bassamboo, R. Randhawa, and A. Zeevi. Capacity sizing under parameter uncertainty: Safety staffing principles revisited. *Management Sci.*, 56(10):1668–1686, 2010. 1.3

[11] S. Borst, A. Mandelbaum, and M. Reiman. Dimensioning of large call centers. *Oper. Res.*, 52(1):17–34, 2004. 1.2

[12] M. Bramson. State space collapse with application to heavy traffic limits for multiclass queueing networks. *Queueing Syst. Theory Appl.*, 30:89–148, 1998. 1.3

[13] L. Brown, N. Gans, A. Mandelbaum, A. Sakov, H. Shen, S. Zeltyn, and L. Zhao. Statistical analysis of a telephone call center: A queueing-science perspective. *J. Amer. Stat. Assoc.*, 100:36–50, 2005. 1.1

[14] I. Cohen, A. Mandelbaum, and N. Zychlinski. Minimizing mortality in a mass casualty event: Fluid networks in support of modeling and staffing. *IIE Trans.*, 46(7):728–741, 2014. 1.3

[15] J. Dai. On positive harris recurrence of multiclass queueing networks: A unified approach via fluid limit models. *Ann. Appl. Probab.*, 5(1):49–77, 1995. 1.3

[16] J. Dai. A fluid-limit model criterion for instability of multiclass queueing networks. *Ann. Appl. Probab.*, 6(3):751–757, 1996. 1.3

[17] F. de Véricourt and O. Jennings. Dimensioning large-scale membership services. *Oper. Res.*, 56(1):173–187, 2008. 1.2

[18] J. Dong and W. Whitt. Stochastic grey-box modeling of queueing systems: Exploiting fitted birth-and-death processes. Preprint. 1.1

[19] O. Garnett, A. Mandelbaum, and M. Reiman. Designing a call center with impatient customers. *Manufacturing Service Oper. Management*, 4(3):208–227, 2002. 1.2

[20] Y. Guo, E. Lefeber, Y. Nazarathy, G. Weiss, and H. Zhang. Stability of multi-class queueing networks with infinite virtual queues. *Queueing Syst. Theory Appl.*, 76(3):309–342, 2014. 1.2

[21] I. Gurvich and W. Whitt. Queue-and-idleness-ratio controls in many-server service systems. *Math. Oper. Res.*, 34(2):363–396, 2009. 1.2

[22] S. Halfin and W. Whitt. Heavy-traffic limits for queues with many exponential servers. *Oper. Res.*, 29(3):567–588, 1981. 1.2

[23] J.M. Harrison. Brownian models of queueing networks with heterogeneous customer populations. In W. Fleming and P.L. Lions, editors, *Stochastic Differential Systems, Stochastic Control Theory and Applications*, volume 10, pages 147–186. Springer-Verlag, New York, 1988. 1.3

[24] J.M. Harrison. Brownian models of open processing networks: Canonical representation of workload. *Ann. Appl. Probab.*, 10(1):75–103, 2000. 1.3

[25] J.M. Harrison. Stochastic networks and activity analysis. In Yu. Suhov, editor, *Analytic Methods in Applied Probability. In Memory of Fridrih Karpelevich*, pages 53–76. American Mathematical Society, Providence, RI, 2002. 1.3, 3.1, 2

[26] J.M. Harrison. A broader view of brownian networks. *Ann. Appl. Probab.*, 13(3):1119–1150, 2003. 1.3

[27] J.M. Harrison and V. Nguyen. Some badly behaved closed queueing networks. In F.P. Kelly and R.J. Williams, editors, *Stochastic Networks*, volume 71 of *IMA Volumes in Mathematics and its Applications*, pages 117–124. Springer-Verlag, New York, 1995. 1.2

[28] J.M. Harrison and L. Wein. Scheduling networks of queues: Heavy traffic analysis of a two-station closed network. *Oper. Res.*, 38(6):1052–1064, 1990. 1.2

[29] J.M. Harrison and R. Williams. A multiclass closed queueing network with unconventional heavy traffic behavior. *Ann. Appl. Probab.*, 6(1):1–47, 1996. 1.2

[30] J.M. Harrison and A. Zeevi. A method for staffing large call centers based on stochastic fluid models. *Manufacturing Service Oper. Management*, 7(1):20–36, 2005. 1.3

[31] D. Iglehart and W. Whitt. Multiple channel queues in heavy traffic. I. *Adv. Appl. Probab.*, 2(1):150–177, 1970. 1.2

[32] D. Iglehart and W. Whitt. Multiple channel queues in heavy traffic. II: Sequances, networks, and batches. *Adv. Appl. Probab.*, 2(2):355–364, 1970. 1.2

[33] O. Jennings, A. Mandelbaum, W. Massey, and W. Whitt. Server staffing to meet time-varying demand. *Management Sci.*, 42(10):1383–1394, 1996. 1.2, 1.3

[34] W. Kang and K. Ramanan. Fluid limits of many-server queues with reneging. *Ann. Appl. Probab.*, 20(6):2204–2260, 2010. 1.3

[35] H. Kaspi and K. Ramanan. Law of large numbers limits for many-server queues. *Ann. Appl. Probab.*, 21(1):33–114, 2011. 1.3

[36] Y. Kogan, R. Lipster, and A. Smorodinskii. Gaussian diffusion approximation of closed Markov models of computer networks. *Problems Inform. Transmission*, 22(1):38–51, 1986. 1.2

[37] A. Kopzon, Y. Nazarathy, and G. Weiss. A push pull system with infinite supply of work. *Queueing Syst. Theory Appl.*, 62(1-2):75–111, 2009. 1.2

[38] S. Kumar. Two-server closed networks in heavy traffic: Diffusion limits and asymptotic optimality. *Ann. Appl. Probab.*, 10(3):930–961, 2000. 1.2

[39] X. Liu, Q. Gong, and V. Kulkarni. Diffusion models for double-ended queues with renewal arrival processes. *Stochastic Syst.*, to appear. 1.2

[40] Y. Liu and W. Whitt. The $\mathrm{G}_t/\mathrm{GI}/\mathrm{s}_t+\mathrm{GI}$ many-server fluid queue. *Queueing Syst. Theory Appl.*, 71(4):405–444, 2012. 1.3

[41] A. Mandelbaum and G. Pats. State-dependent queues: Approximations and applications. In F. Kelly and R. Williams, editors, *Stochastic Networks*, volume 71 of *IMA Volumes in Mathematics*, pages 239–282. Springer, New York, NY, 1995. 1.2

[42] P. Momčilović and A. Motaei. An analysis of a large-scale machine repair model. Preprint. 1.2

[43] G. Pang and A. Stolyar. A service system with on-demand agent invitations. Preprint. 1.2

[44] R. Randhawa and S. Kumar. Multiserver loss systems with subscribers. *Math. Oper. Res.*, 34(1):142–179, 2009. 1.2

[45] M. Reiman. The heavy traffic diffusion approximation for sojourn times in jackson networks. In R.L. Disney and T.J. Ott, editors, *Applied Probability – Computer Science, The Interface*, pages 409–422. Birkhauser, Boston, MA, 1982. 1.2

[46] M. Reiman. Open queueing networks in heavy traffic. *Math. Oper. Res.*, 9(3):441–458, 1984. 1.2

[47] C. René, E. Kaplan, and G. Weiss. FCFS infintie bipartite matching of servers and customers. *Adv. Appl. Probab.*, 41(3):695–730, 2009. 1.2

[48] K. Sevcik and I. Mitrani. The distribution of queuing network states at input and output instants. *J. ACM*, 28(2):358–371, 1981. 1.2

[49] A. Ward and M. Armony. Blind fair routing in large-scale service systems with heterogeneous customers and servers. *Oper. Res.*, 61(1):228–243, 2013. 1.2

[50] W. Whitt. Open and closed models for networks of queues. *AT&T Bell Lab. Techn. J.*, 63(9):1911–1979, 1984. 1.2

[51] W. Whitt. Offered load analysis for staffing. *Manufacturing Service Oper. Management*, 15(2):166–169, 2013. 1.2

[52] R. Wolff. Poisson arrivals see time averages. *Oper. Res.*, 30(2):223–231, 1982. 1.2

[53] G. Yom-Tov and A. Mandelbaum. Erlang-R: A time-varying queue with ReEntrant customers, in support of healthcare staffing. *Manufacturing Service Oper. Management*, to appear. 1.2, 1.3