4-2022

Joint capacity allocation and job assignment under uncertainty

Peng WANG

Yun Fong LIM
Singapore Management University, yflim@smu.edu.sg

Gar Goei LOKE

manuscript

Joint Capacity Allocation and Job Assignment under Uncertainty

Peng Wang
NUS Business School, National University of Singapore, Singapore 119245, peng.wang@nus.edu.sg

Yun Fong Lim
Lee Kong Chian School of Business, Singapore Management University, Singapore 178899, yflim@smu.edu.sg

Gar Goei Loke
Rotterdam School of Management, Erasmus University, Burgemeester Oudlaan 50, 3062PA Rotterdam, The Netherlands, loke@rsm.nl

In this paper, we consider the multi-period joint capacity allocation and job assignment problem. The goal of the planner is to simultaneously decide on allocating resources across the J different supply nodes, and assigning of jobs of I different demand origins to these J nodes, so as to maximize the reward for matching or minimize the cost of failure to match. We furthermore consider three features: (i) supply is replenishable after random time, (ii) demand is random; and (iii) demand can wait and need not be fully fulfilled immediately. Such problems emerge in many service management settings such as ride-sharing fleet re-positioning, and patient management in healthcare. We introduce a *distributive decision rule*, which decides on the proportion of jobs to be served by each of the supply nodes. We borrow ideas from the pipeline queues framework (Bandi and Loke 2018), which cannot be directly applied to our setting, and hence requires the development of new reformulation techniques. Our model has a convex reformulation and can be solved by a sequence of linear programs, in practice. We test our model against state-of-the-art models that focus solely on the capacity allocation or job assignment decisions, in the setting of nurse scheduling and patient overflow respectively. Our model performs strongly against the benchmarks, recording $1-15\%$ reductions in costs, and shorter computation times. Our model opens the door to consider new problems in platform operations and online services where the planner is able to influence the supply of services or resources partially.

Key words: Programming, Convex optimization, Resource allocation
History: May 18, 2022

1. Introduction

Job assignment, also termed matching, has long been studied as a core area of Operations Management (Karp et al. 1990, Reeves and Sweigart 1982, Riedel 1999). In these problems, the planner manages a network where there are I different types of jobs, that can potentially be fulfilled by J different types of resources. The goal of the planner is to assign these jobs to the resources and in so doing, minimize the costs incurred or maximize the revenue accrued in the assignment. We illustrate this in the schematic in Figure 1. In time, many variations of this problem have been

considered, for example, whether there is uncertainty in the demand or the availability of the resource; whether the problem is posed in a multi-period setting and there is roll-over of resources or unmet demand; whether the jobs are heterogenous, or can be classified into types wherein they are homogeneous; whether multiple resources of different types are required to fulfill the different demands; whether there is full knowledge of the uncertainties or the planner needs to learn about the uncertainties as they continue to make assignment decisions, etc.

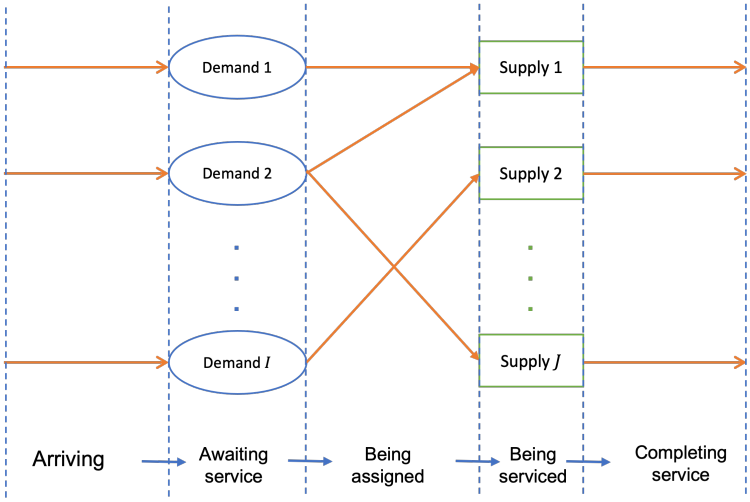


Figure 1 Schematic of the joint capacity allocation and job assignment problem

In recent times, the job assignment problem has seen new applications in service provision based business models that work within the confines of an online platform. Such developments are driven by the growth of the sharing economy, and changes in the patterns of logistics and customer demands in the post COVID-19 world. Examples of such operations include last-mile delivery (Qi et al. 2018), vehicle routing in ride-sharing (Spivey and Powell 2004) and distributed systems (Ghosh et al. 2017), to name a few. They bring new challenges to the traditional job assignment problem. First, they are large-scale, involving a large number of customers and transactions at any point in time. Second, the online setting provides an abundance of data from which patterns of demand can be inferred, and at extremely granular levels.

More critically, in the setting of online service operations, we are seeing instances where planners can influence the availability of resources. Such influence can be indirect, for example, incentivizing drivers to move towards a locality with greater demand on ride-sharing platforms, or direct, such as the physical repositioning of vehicles in a car rental operation. In this paper, we are specifically interested to consider cases where the planner can make a direct decision on the allocation of resources. In terms of our schematic in Figure 1, this is captured by the direct control and allocation of resources amongst the J supply nodes.

We term these *joint capacity allocation and job assignment problems*. Here, the planner first makes a here-and-now decision on how much resources of each type is to be made available for each supply node $j = 1, \ldots, J$, incurring some cost in the process, then observes how the uncertainty unfolds, before deciding on the job assignment recourse of assigning jobs of type $i = 1, \ldots, I$ to resource of type $j = 1, \ldots, J$, under the limitations of the resource allocation earlier decided.

In the literature, the context where the capacity allocation decisions are made only once, even if job assignment occurs over multiple time stages, has been considered, such as the facility location problem or the resource scheduling problem (*e.g.* Gupta and Wang 2008). In the context of online services, however, the planner is sometimes able to dynamically allocate resources, so as to direct resources to local shortages and in anticipation of the demand. Such multi-period setting will induce *endogeneity* into the problem – where each allocation decisions itself affects the nature of the uncertainty in the service provision.

In this paper, we are particularly interested in examining the joint capacity allocation and job assignment context. As is common within such contexts, we consider the situation where there is both demand uncertainty and resource availability uncertainty. Specifically, in the latter, we assume that these resources are replenishable, as is common of services, but take some random but finite amount of time to complete service. Here, our model allows jobs to wait until they are assigned / served as is common of service provision. We limit our attention to situations where one resource is sufficient to fulfill one job, whose type is limited by the type of the job. We are also interested to consider this problem in the context of a discrete-time finite-window setting, as it most closely resembles the operating context in practice. Our proposed model will be relevant for both the online platform setting, and also traditional settings that share similar traits and characteristics, such as healthcare operations management. However, we exclude the following settings, where the reward or cost of job assignment is unknown, such as online advertising where the value of each match is unknown and sufficient exploration is required; where the resources are not renewable and are exhausted, such as advanced booking; or where the jobs or resources are highly heterogeneous, such as personalized services. We are interested to consider such assumptions as they are commonly seen in many other literature, which we shall imminently discuss in the motivating examples.

Motivating examples

Before proceeding, we list three examples posed in this setting to motivate the relevance of our context and the degree of applicability of our proposed model.

We first raise a traditional example in the area of staff scheduling, such as nurse scheduling in an emergency department (ED), as described in Chan et al. (2021). Here, nurses are allocated to different areas of the ED to serve patients in that area. In this context, a job corresponds to

a patient, whereas a resource corresponds to a nurse. Jobs and resources here are classified by the particular areas in the ED. We can see that the resource here is replenishable, however its availability is uncertain, due to the random time required for nurses to serve each patient. Demand is also uncertain. In this paper, the authors suggest that the planner could flexibly re-allocate nurses among the different areas at the start of each work shift, in order to react to local shortages in each of the areas. These form the capacity allocation decisions, while the job assignment decisions are then dependent on how the nurses are actually assigned to different patients.

We next raise an example in the area of car rentals (also bicycle sharing, such as in Shu et al. 2013). In car rentals, the job corresponds to the customer and the resource corresponds to the car. Jobs and resources are classified according to geographical localities. If there is demand but no cars in any nearby locality, then the planner incurs lost sales. In the absence of intervention, this happens frequently – there is usually imbalance in vehicle numbers after periods of peak demand. As such, there is incentive for the planner to reposition the cars, *as and when the need arises*. The repositioning amounts to the capacity allocation decision, whereas the job assignment decision relates to how accessible is each locality to the geographical location of where the demand arises. Notice that there is demand and supply uncertainty, because the planner does not *a priori* know where the customers intend to go, and even if they do, the actual time taken to reach; in other words, the time taken for the resource to be replenished depends on the traffic situation.

Our third example is in the area of cloud computing. In this case, the jobs are actual computational jobs that require work. The resources is the allocated computing bandwidths for each of the computing clusters. The planner not just needs to assign the jobs to the clusters, but also potentially increase the bandwidth of the clusters during times of shortages, which amount to the capacity allocation decision. Once again, the time it takes for the computing task to complete is uncertain, as is the demand for computing tasks of different types.

1.1. Literature review

In this subsection, we broadly examine the literature in capacity allocation and job assignment. To the best of our abilities, we are unable to find any works that simultaneously handle both capacity allocation and job assignment decisions. Hence, we do not review the literature specific to capacity allocation and job assignment, but but from the methodological perspective, so as to identify core challenges in trying to solve the joint capacity allocation and job assignment problem.

Queueing: One of the most popular approach to the job assignment and capacity allocation problem is to model it as a queueing network. Under this approach, the matching of demand to resources is modeled as routing decisions in a network of queues and servers. In such a context, demand arrivals and service times are modelled as stochastic and their distributions are assumed to be known.

For example, Armony and Ward (2010) investigate which agent should handle a new arrival when there are multiple agents available in a call center. To minimize customer waiting time subject to a fair division of the workload among agents with different skill levels, the authors propose a threshold policy to determine server priorities based on the total number of customers in the system. To arrive at an optimization formulation, there are two common approaches seen in the literature. The first is using fluid models (Dai and Meyn 1995). For example, Puha and Ward (2019) study a multi-class many-server network with impatient customers, and use a restricted fluid model to approximately solve a scheduling control problem. Chan et al. (2021) examine dynamic allocation of servers in a multi-class setting. By assuming Poisson arrivals and exponential service times, they construct a deterministic discrete- and finite-time fluid control approximation. They show that their model is asymptotically optimal. However, the transient and state-dependent nature of our problem renders the steady-state setting of fluid models less applicable. Moreover, it is not always easy to find the fluid model or to prove its convergence or stability conditions, for more complex network structures. The second is approximate dynamic programming (ADP) approach. For example, Martonosi (2011) investigates whether dynamically reassigning servers to parallel queues in response to queue imbalances can reduce average waiting times. Dai and Shi (2019) study the inpatient overflow problem in a hospital to decide whether and when a patient should be assigned to a non-primary ward when their primary ward is fully occupied. The authors model the problem as a Markov decision process in a multi-class, multi-pool parallel-server queuing system. They employ an ADP approach to solve the model. They demonstrate that their ADP algorithm is remarkably effective in finding good overflow policies via numerical experiments in realistic hospital settings. However, the large state space of our problem makes it hard for an ADP approach to balance between tractablility and veracity. The dependence of ADP on the choice of basis functions also makes it difficult to extend beyond the specific applications the basis functions are defined for.

Stochastic Programming: Another stream of literature employs a stochastic programming framework to solve the job assignment and capacity allocation problems. Lyu et al. (2019) address an online ride-matching problem using convex optimization techniques. They consider multiple objectives including the revenue, pick-up distance, and service quality. They prove that their policy can achieve the closest solution to any pre-determined multi-objective target. Özkan and Ward (2020) study dynamic matching policies for real-time ride sharing. They propose policies based on a linear program that accounts for time-varying arrival rates of customers and drivers, who are willing to wait. When pricing affects customer and driver arrival rates with time-homogeneous parameters, they show that their policy leads to fully utilized drivers under mild conditions. Jaillet and Lu (2014) examine an online stochastic bipartite matching problem for advertising. Based on

the solution of linear programs of maximum-flow problems, they show that their online algorithm is computationally efficient with better bounds. He et al. (2019) develop a data-driven robust framework to study a patient scheduling problem in an emergency department. They balance the patients' door-to-provider time and the length of stay. Correia et al. (2018) consider a system with multiple hubs facing stochastic demands in a multi-period setting. They propose a model to decide on the locations of the hubs and their capacities.

However, it may be difficult to formulate a model if the decisions changes the distribution of the uncertainty. While it may still be possible to conduct Sample Average Approximations (SAA) to evaluate the expectation, one requires an exponential number of data samples. Lacking which, the performance can be suboptimal (as illustrated in Zhou et al. 2022).

Learning: There are works applying the learning approach to the matching problem, such as Dai and Gluzman (2021), Johari et al. (2021), etc. The learning literature is less relevant to our work as in our problem setting, we assume that the decision-maker possesses some data to estimate the demand and supply distributions, and moreover, the assignment cost is known.

1.2. Our approach

The literature is extremely scant on formulations that solve the multi-period joint capacity allocation and job assignment problem. The challenges arise due to the multi-period nature of the problem, the interactions between allocation and assignment decisions with the uncertainty, and its large scale nature leading to tractability concerns. To this end, we appeal to the Pipeline Queues methodology seen in Zhou et al. (2022), and more originally in Bandi and Loke (2018). The methodology lies within the stream of Satisficing (Brown and Sim 2008, Lam et al. 2013, Jaillet et al. 2022), in the broad area of robust optimization. Traditionally, the approach to multi-period problems in Robust Optimization has been to define decision-rules on the primal uncertainty in a time-independent nature. The approach in Pipeline Queue, however, is to define the uncertainty in an endogenous fashion to allow the uncertainty to change given the decisions in the previous time period. This can yield strong performance, as illustrated in Zhou et al. (2022).

1.3. Contributions

We propose a discrete-time finite-window model for solving a general class of multi-period joint capacity allocation and job assignment problems arising in the replenishable supply context with both demand and resource availability uncertainty. Our formulation has the following benefits:

a) It proposes a general framework for the joint capacity allocation and job assignment problem in the multi-period setting. To the best of our knowledge, such a framework has yet to be

examined by the literature today, despite it becoming increasingly relevant. Moreover, our decision policy is an adaptive policy that is state-dependent.

b) It exhibits superior performance: despite being posed in the joint setting, when applied to only capacity allocation or job assignment, it still outperforms models designed for these settings. Here, we nominate the state-of-the-art models, Chan et al. (2021) for the capacity allocation setting (Section 4.1), and Dai and Shi (2019) for the job assignment setting (Section 4.2). Our model, when restricted to these settings, out-performs both models. Moreover, the two reference models utilize different solution techniques, namely, fluid models and approximate dynamic programming respectively. Hence, our experiments also illustrate the superiority of our approach over these paradigms.

c) It is tractable. Specifically, it can be solved via a sequence of convex programs with polynomial number of constraints in time and nodes in the network (Theorem 2).

It is critical at this point to mention that the multi-period joint capacity allocation and job assignment problem cannot be directly solved using the Pipeline Queues framework, as discussed in Section 2.4, and numerically illustrated in Appendix B.3. As such, this paper also contributes to the growing theory and methodology in the technique of Pipeline Queues. Specifically, in Theorem 1, we employ a new technique, not seen in previous works on Pipeline Queues, in order to address the new challenges faced in the joint capacity allocation and job assignment setting. While our proposed model might visually appear similar to linear decision rules often seen in the Robust Optimization literature, we specifically discuss their differences in Section 2.4.

Organization of the paper

We first formulate the problem in Section 2. To make the problem tractable, we propose a decision criteria to handle the uncertainty, and present our full model in Section 3. Section 4 conducts numerical experiments by comparing our approach against other methods in the literature. Finally, Section 5 provides some concluding remarks. For brevity, all proofs have been omitted from the main text and are instead presented in Appendix A.

Notation. We adopt the convention that $\inf \emptyset = +\infty$, where $\emptyset$ is the empty set. We use $\text{Bin}(n, p)$ to refer to the Binomial random variable of n independent trials, each with probability p.

2. Problem formulation

We present our formulation for a joint capacity-allocation and job-assignment problem. Specifically, we model the flow of jobs from arrival, to awaiting service, to being assigned, to experiencing service and to service completion, as progression through a bipartite network, as presented in Figure 2.

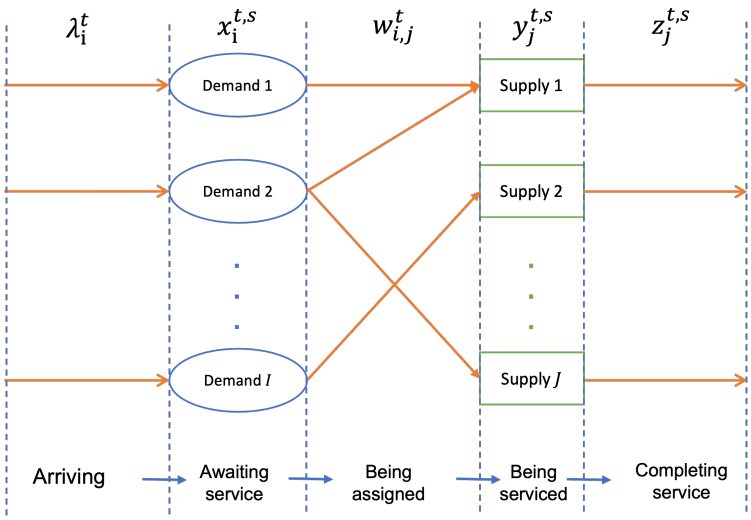


Figure 2 A bipartite graph that illustrates the flow of jobs through the system

Here, we assume that jobs of different types arrive at the system, and that they each require a resource to be completed. The type of job would then determine the type of resource that can be used to complete it. From this, we motivate a bipartite graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ of vertices $\mathcal{N}$ and edges $\mathcal{E}$. The vertices $\mathcal{N}$ can be subdivided into the demand nodes $i \in \mathcal{I} = \{1, 2, \ldots, I\}$, representing the collection of identical type i jobs arriving into the system, and supply nodes $j \in \mathcal{J} = \{1, 2, \ldots, J\}$, representing the collection of identical type j resources that can be utilized. As such, whether jobs of type i can be fulfilled by resources of type j, is represented by the edge $(i, j) \in \mathcal{E}$. Here, we specifically assume that each job only requires one resource to be fulfilled. Moreover, the resources are replenishable, in other words, once the jobs are fulfilled, the resource is made available again to fulfill other jobs.

This structure is general, but adequate at addressing problems in our context. For example, in the nurse scheduling problem, each demand node $i \in \mathcal{I}$ corresponds to a particular type of medical condition and each job corresponds to a patient with that condition. Each supply node $j \in \mathcal{J}$ corresponds to the services provided by nurses in a specific area of the ED. Here, the capacities of the supply nodes are given by the number of nurses allocated to the area. The edges (i, j) represent the patients with conditions i that can be served by nurses in area j. Hence, the decisions include determining the optimal allocation of nurses across the different specific areas j, and $\boldsymbol{w}_{i,j}$ the number of patients with condition i that are to be assigned to nurses in area j, where $\boldsymbol{w}_{i,j}$ is constrained to be 0 only if the edge (i, j) is not in the graph. As we can see in this system, the patients are allowed to wait, though this comes at some cost, and service can take a random amount of time, but depends on the specific area j which decides the type of treatment provided to the patient. In Section 4.1, we study precisely this problem setting.

Sequence of events

Consider a planning horizon with T time periods. Let $\mathcal{T} := \{1, \ldots, T\}$ and $\mathcal{T}_0 := \{0\} \cup \mathcal{T}$. We describe the sequence of events in each time period, as illustrated in Figure 3.

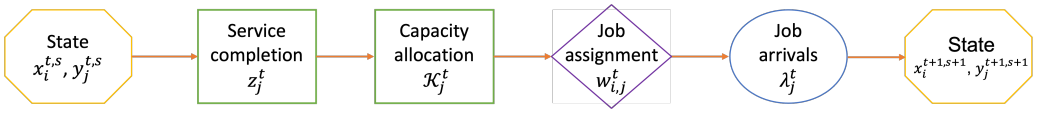


Figure 3 The sequence of events in period t

At the start of each period t, amongst all of the jobs at present being processed in the supply nodes, some number of them, z_j^t, completes their service, and subsequently leaves the system. The resources that are assigned to these jobs now become available and are temporarily idling. At this point, in the most general setting, the decision-maker decides on a *capacity allocation* $\mathcal{K}_j^t$ for each of the supply nodes of type j resources. This number represents the number of resources of type j that are present at time t, be it idling or in the midst of already serving a job. Or in other words, this decision variable can also be seen as a reallocation of idling resources across different supply nodes. If a purely job assignment problem is considered without capacity allocation, then $\mathcal{K}_j^t$ are just fixed constants. Based on this new capacity allocation, the decision-maker also decides on a *job assignment* of jobs of type i to resource of type j. This is denoted by the decision variable $w_{i,j}^t$, which is the number of jobs of type i to be fulfilled by resources of type j. At this point, a job assignment cost $a_{i,j}^t$ is incurred per job. In the model, this is represented by a $w_{i,j}^t$ flow from demand node i into supply node j. These jobs will commence service by available resources at their respective supply nodes by the end of the time period, and will take some number of time periods to be processed. Thereafter, a random λ_i^t number of new jobs of type i arrives at the demand nodes. These jobs are assumed to join a queue with infinite buffer, along with all the other yet-to-be-assigned jobs of the same type. The collection of all these jobs is loosely referred to as the queue in demand node i for type i jobs. At this point, they all incur a waiting cost (per period) b_i^t, that depends on the time period t and the job type i. This completes the sequence of events for time period t.

In summary, within each time period, the decision-maker always has to make two decisions, the capacity allocation decisions $\mathcal{K}_j^t$ (optional) and the job assignment decisions $w_{i,j}^t$. Two types of uncertainties, namely the arrivals λ_i^t and the service completions z_j^t, will unfold some time or another. The decision-maker incurs two costs, job assignment costs $a_{i,j}^t$ and waiting costs b_i^t, in this process, which they intend to minimize over the planning horizon by optimizing their decisions. This describes the optimization problem the decision-maker hopes to solve.

We make the following assumption on the uncertainties.

Assumption 1

a) The random variables of the number of arrivals $\lambda_i^t \sim \Lambda_i^t$ are drawn independently and identically from some general demand distribution Λ_i^t, that is exogenous to the system. Moreover, the demand distribution is assumed to be finite, and independent of each other, in both type i and time t. Specifically, Λ_i^t and $\Lambda_{i'}^{t'}$ are independent for any $(i,t) \neq (i',t')$.

b) Service completions is assumed to obey some finite and discrete general service time distribution S_j^t, which depends only on the resource type j that is servicing the job and the time t at which the resource was first assigned to the job. Moreover, each job is assumed to have an independent and identical service time drawn from this distribution.

We take some time here to justify these assumptions. First, we assume finite-ness in both of these distributions, because it is natural to do so in the real-world context, and because we are working within finite time horizons. Additionally, finite-ness guarantees that bounded moment generating functions exists for these distributions, which will be critical for our model later (see Proposition 3). Second, the independence and identical assumption is a reflection of the assumed homogeneity in the jobs. In large scale systems, each job is unable to influence the system unilaterally, and small dependent effects have minimal impact on the eventual outcome. In particular, independence will also play a critical role in the reformulation of the model later. In the language of queueing theory, we are considering a $GI/GI/\cdot$ system.

Remark 1. One may question why the service time distribution does not involve the type of job i. If this is desired, then the supply node j may be split into multiple supply nodes (i,j), receiving jobs only of type i and utilizing resource type j, with different service time distributions imposed on them. As such, Assumption 1b) is without loss of generalization.

2.1. Discrete-time model

We first begin by describing a discrete time dynamics for the aforementioned sequence of events. Let x_i^t denote the total jobs of type i awaiting assignment at demand node i at the end of period t, for $i \in \mathcal{I}$, $t \in \mathcal{T}_0$. Note that x_i^t can be interpreted as the queue length at demand node i. Let y_j^t denote the number of jobs that are currently being processed by resource of type j at supply node j by the end of period t, for $j \in \mathcal{J}$, $t \in \mathcal{T}_0$. Here, x_i^0 and y_j^0 are understood to be the initial conditions of the system. The dynamics at demand node i can be expressed as follows:

$$x_i^t = x_i^{t-1} + \lambda_i^t - \sum_{j:(i,j)\in\mathcal{E}} w_{i,j}^t, \quad t \in \mathcal{T}, \tag{1}$$

in other words, the number of jobs awaiting assignment at the end of time period t would be the jobs awaiting assignment from the previous time period, minus those that have been assigned, and then added those that have just arrived.

As before, we let the number of jobs that complete their service at the end of period t at supply node j be z_j^t, for $j \in \mathcal{J}$, $t \in \mathcal{T}$. Thus, the dynamics at supply node j can be expressed as follows:

$$y_j^t = y_j^{t-1} + \sum_{i:(i,j)\in\mathcal{E}} w_{i,j}^t - z_j^t, \quad t \in \mathcal{T}. \tag{2}$$

Similarly, this means that the number of jobs being serviced by resource of type j at the end of time period t would make up those that were already serviced in the previous time period, less those that have completed service and added those that have recently been assigned to resource type j.

Although the above formulation is simple, there are two challenges here that make it difficult to proceed. The first challenge is the state-dependent nature of the random variable z_j^t. In general, z_j^t depends on y_j^t (more strictly, y_j^{t-1}), as the more jobs there are, the more jobs there will be which are completing service on average. Moreover, linking z_j^t to the service time distribution S_j^t is difficult, especially for any general distribution S_j^t, and especially in the transient finite time horizon setting we are considering. This arises because service completion is linked to the time that each job has been served by resource of type j, and this is not tracked in the formulation (2). As an example, if most of the jobs at supply node j have been served for a long duration, then z_j^t is likely to be large; however, if most of these jobs have only been served for a short duration, then z_j^t is likely to be small.

The second challenge relates to the nature of how $w_{i,j}^t$ is defined. Here, we had not been specific about it for good reasons. For example, should the policy on $w_{i,j}^t := w_{i,j}^t(x_i^{t-1}, y_j^{t-1})$ be a function of the last observed state of the system, as one would expect in a Markov Decision Process setting? If so, then to the best of our knowledge, we do not know any tractable ways for solving the above formulation under large networks (*i.e.* I and J both large), large state spaces (*i.e.* x_i^t and y_j^t both numerically large) and long time horizons (T large). On the other hand, other options, such as solving for a static $w_{i,j}^t$, *i.e.* where $w_{i,j}^t$ only depends on initial data x_i^0 and y_j^0 and distributional information about Λ_i^t and S_j^t, is needlessly myopic and do not result in good solutions, as we will illustrate later in Section B.3.

In the next two subsections, we shall in turn address these two challenges in our modified framework of the above dynamics (1) and (2), to arrive at our proposed model.

2.2. Discrete-time model with present delay s

To overcome the first difficulty, we need to keep track of how long each job spends at each node (as motivated by the approach introduced in Bandi and Loke 2018). To this end, we introduce the discrete time model with present delay s. Let M represent the largest time that any job will await

assignment or be serviced by any resource. Denote $\mathcal{M} = \{1, \ldots, M\}$ and $\mathcal{M}_0 = \{0\} \cup \mathcal{M}$. For all subsequent dynamics, it is assumed that the job leaves the system if M is exceeded.

Let $x_i^{t,s}$ denote the number of jobs that have spent s periods at demand node i at the start of period t, for $i \in \mathcal{I}$, $t \in \mathcal{T}_0$, $s \in \mathcal{M}_0$. The index s, which keeps track of how long a job has spent at a node, is termed the *present delay*. Similarly, let $y_j^{t,s}$ denote the number of jobs that have been served for s periods by resources of type j at supply node j at the start of period t, for $j \in \mathcal{J}$, $t \in \mathcal{T}_0$, $s \in \mathcal{M}_0$. As before, $x_i^{0,s}$ and $y_j^{0,s}$ are understood as the initial conditions. Finally, we re-defined the decision variables as $w_{i,j}^{t,s}$, to denote the number of jobs of type i that awaited assignment for s periods at demand node i, and are now at period t, assigned to be fulfilled by resource of type j at supply node j, for $(i,j) \in \mathcal{E}$, $t \in \mathcal{T}$, $s \in \mathcal{M}$.

The dynamics at demand node i can be expressed as

$$x_i^{t,0} = \lambda_i^t, \tag{3}$$

$$x_i^{t,s} = x_i^{t-1,s-1} - \sum_{j:(i,j)\in\mathcal{E}} w_{i,j}^{t,s}, \qquad s \in \mathcal{M}, \tag{4}$$

for $i \in \mathcal{I}$, $t \in \mathcal{T}$. For the purposes of brevity, we write our dynamics here in (4), without abandonment. However, our framework can be easily generalized to the case where jobs abandon the queue with a probability that depends on the current time period t and present delay s (see Appendix C for more details). Here, the dynamics differ from (1), because all inflow into demand node i corresponds to the collection of type i jobs that have spent 0 amount of time (*i.e.* yet to spend 1 period of time) in the system at the end of period t, which is by definition $x_i^{t,0}$. In other words, the inflow components split from the outflow components, represented in (4).

To proceed and define the dynamics on the supply nodes, one needs to be first clear about the service time distribution.

Assumption 1b') *We assume that at any time period $t \in \mathcal{T}_0$, each job that is serviced by resource of type $j \in \mathcal{J}$ has an identical and independent probability of $1 - q_j^{t,s}$ of completing service, if it has already been serviced for $s-1 \in \mathcal{M}$ periods. In other words, if S_j^t is the random variable of the service time of the job, which started service at time t by resource of type j, then*

$$\mathbb{P}\left[S_j^{t-s} \geq s \,\middle|\, S_j^{t-s} \geq s-1\right] = q_j^{t,s}.$$

Proposition 1 *Any general service time distribution that obeys Assumption 1b) can be represented in the form of Assumption 1b'),* i.e. *it is without loss of generalization that Assumption 1b') can be assumed.*

The above Proposition and Assumption justify the use of the Binomial distribution to count the number of completed jobs in each time period: the number of jobs that are serviced for s time periods by resource of type j which are completed at time t follows a Binomial distribution $\text{Bin}\left(y_j^{t-1,s-1}, 1-q_j^{t,s}\right)$. We call $q_j^{t,s}$ the *survival probability*, and this induces the dynamics at supply node j:

$$y_j^{t,0} = \sum_{i:(i,j)\in\mathcal{E}} \sum_{s\in\mathcal{M}_0} w_{i,j}^{t,s}, \tag{5}$$

$$y_j^{t,s} \sim \text{Bin}\left(y_j^{t-1,s-1}, q_j^{t,s}\right), \qquad s\in\mathcal{M}, \tag{6}$$

for $j\in\mathcal{J}$, $t\in\mathcal{T}$. Similar to the dynamics for the demand nodes, the dynamics here splits into the inflows (5) and outflows (6). A corollary of (6) is the simplified expression:

$$y_j^{t,s} \sim \begin{cases} \text{Bin}(y_j^{t-s,0}, p_j^{t,s}), & 0\le s<t, \\ \text{Bin}(y_j^{0,s-t}, p_j^{t,s}), & t\le s\le M, \end{cases} \qquad j\in\mathcal{J}, t\in\mathcal{T},$$

where $p_j^{t,s} = \prod_{\tau=0}^{\min\{t,s\}-1} q_j^{t-\tau,s-\tau}$. Notice that $p_j^{t,s} = \mathbb{P}[S_j^{t-s}\ge s]$. Hence, we call $p_j^{t,s}$ the *cumulative survival probability*.

At this point, the dynamics are well-defined, and we have addressed the issue about the service completions. It leaves now to address the matter of the assignment decisions, $w_{i,j}^{t,s}$.

2.3. Distributive discrete time model with present delay s

In Bandi and Loke (2018)'s original framework for Pipeline Queues, the flows out of queues are static variables and the flows out of servers can be made to be decision rules. In our setting, the former corresponds to $w_{i,j}^{t,s}$, while there are no flows out of the supply nodes that require any form of decision. This means that implementing the model in 2.2 would correspond with a direct application of the Pipeline Queues framework. However, as discussed earlier, having static $w_{i,j}^{t,s}$ is needlessly myopic. As such, we propose a *distributive decision rule* as follows:

$$w_{i,j}^{t,s} = x_i^{t-1,s-1}\frac{\alpha_{i,j}^{t,s}}{\beta_i^{t-1,s-1}}, \qquad \forall(i,j)\in\mathcal{E}, t\in\mathcal{T}, s\in\mathcal{M}, \tag{7}$$

where $\alpha_{i,j}^{t,s}$ and $\beta_i^{t,s}$ are *scaling factors*. We call them scaling factors because $\alpha_{i,j}^{t,s}/\beta_i^{t-1,s-1} = w_{i,j}^{t,s}/x_i^{t-1,s-1}$.

A similar decision rule is proposed in Bandi and Loke (2018), however, limitations on the structure of the network does not immediately allow such a formulation to be tractable.[1] As we shall

[1] In the original Pipeline Queues framework, the queues, which receive flows arising from such decision rules, are constrained to only receive one such source. Moreover, a further assumption is required to assume independence in the reformulation process.

see, this change is highly non-trivial, because independence is partially lost in this process. This eventually leads to the new methodology proposed in Theorem 1 later.

With decision rule (7), the decision variable $w_{i,j}^{t,s}$ is replaced by two decision variables $\alpha_{i,j}^{t,s}$ and $\beta_i^{t,s}$. This affords us some degrees of freedom – the number of decision variables between $w_{i,j}^{t,s}$ and $\alpha_{i,j}^{t,s}$ are one-to-one, hence, we have the freedom of freely deciding on the values for $\beta_i^{t,s}$. With (7), the earlier dynamics (4) becomes

$$x_i^{t,s} = x_i^{t-1,s-1}\left(1 - \sum_{j:(i,j)\in\mathcal{E}} \frac{\alpha_{i,j}^{t,s}}{\beta_i^{t-1,s-1}}\right), \quad i\in\mathcal{I}, t\in\mathcal{T}, s\in\mathcal{M}. \tag{8}$$

At this point, we exercise our degree of freedom and define

$$\beta_i^{t,s} = \beta_i^{t-1,s-1} - \sum_{j:(i,j)\in\mathcal{E}} \alpha_{i,j}^{t,s}, \quad i\in\mathcal{I}, t\in\mathcal{T}, s\in\mathcal{M}, \tag{9}$$

so that

$$x_i^{t,s} = x_i^{t-1,s-1}\frac{\beta_i^{t,s}}{\beta_i^{t-1,s-1}}, \quad i\in\mathcal{I}, t\in\mathcal{T}, s\in\mathcal{M}. \tag{10}$$

This leads to the interpretation of $\beta_i^{t,s}$ as the scaling factor corresponding to the proportion of $x_i^{t,s}$ that the decision-maker chooses to not assign at period $t-1$ and carries over to period t to be decided later. Under this lens, (9) can be viewed as preserving the flow balance at demand node i. For consistency, we must also impose $\beta_i^{t-1,s-1} \geq \beta_i^{t,s}$ to ensure the monotonicity as implied by (4). At the boundaries, we declare $\beta_i^{0,s} = x_i^{0,s}$, $i\in\mathcal{I}$, $s\in\mathcal{M}_0$, and set $\beta_i^{t,0} = \mathbb{E}[\lambda_i^t]$, $i\in\mathcal{I}$, $t\in\mathcal{T}$, where in practice, the empirical mean $\beta_i^{t,0} = \hat{\mathbb{E}}[\lambda_i^t]$ will be used.

Lastly, (5) is rewritten as

$$y_j^{t,0} = \sum_{i:(i,j)\in\mathcal{E}} \sum_{s\in\mathcal{M}} x_i^{t-1,s-1}\frac{\alpha_{i,j}^{t,s}}{\beta_i^{t-1,s-1}}, \qquad j\in\mathcal{J}, t\in\mathcal{T}, \tag{11}$$

under the new variables. Table 1 summarizes the notation used in our model.

2.4. Differences between our approach and other works

Comparing with linear decision rule. The distributive decision rule (7) ensures that the assignment decisions will always vary with the uncertainty. Unlike decision rules often considered in the setting of robust optimization (*e.g.* in the case of linear decision rules for affine factor models for the uncertainty, Bertsimas et al. 2010, Bertsimas and Goyal 2012), in general, the distributive decision rule is not a function of the uncertainty, but the state of the system (this is explained in depth in Jaillet et al. 2021). Though in the current form, the decision variables can be reformulated as linear functions of the uncertainties, we would like to stress a few key differences. Firstly, our model can be generalized to the case where the jobs in the demand nodes are allowed to abandon the queue.

Table 1 List of parameters and variables

		Parameters
$\mathcal{T}$	:	Set of time periods $\{0, 1, \ldots, T\}$
$\mathcal{I}$	:	Set of demand nodes $\{1, \ldots, I\}$
$\mathcal{J}$	:	Set of supply nodes $\{1, \ldots, J\}$
$\mathcal{K}_j^t$	:	Number of resources (capacity) of supply node j in period t
$\tilde{\lambda}_i^t$	:	Number of arrivals at demand node i at the start of period t (a random variable)
$q_j^{t,s}$	:	Survival probability of each job at supply node j that has been served for s periods at the start of period t
$p_j^{t,s}$	:	Cumulative survival probability, corresponding to $q_j^{t,s}$
		State and decision variables
$x_i^{t,s}$	:	Number of jobs that have spent s periods at demand node i at the start of period t
$y_j^{t,s}$	:	Number of jobs that have been served for s periods at supply node j at the start of period t
$\alpha_{i,j}^{t,s}$	:	Scaling factor corresponding to the proportion of $x_i^{t,s}$ that arrives at supply node j from demand node i at the start of period t
$\beta_i^{t,s}$	:	Scaling factor corresponding to the proportion of $x_i^{t,s}$ that is carried over to period t at demand node i

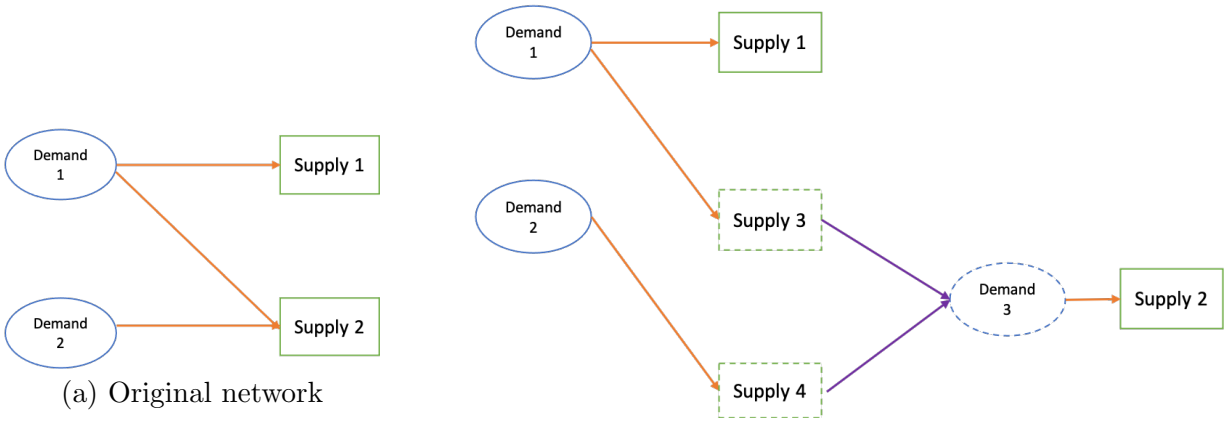
(See Appendix C.) Under this assumption, $x_i^{t,s}$ is a binomial random variable, which can not be reformulated as a linear function of the uncertainties. Secondly, directly applying linear decision rule in our problem will make the model intractable. Without introducing the scaling factors but directly defining $w_{i,j}^{t,s} = x_i^{t-1,s-1}\alpha_{i,j}^{t,s}$, the eventual formulation would not be convex, especially in Theorem 1 later.

Comparing with pipeline queue framework. While Bandi and Loke (2018) introduced the proportional decision rule, its application is quite different. In Bandi and Loke (2018), the decision rule is implemented on the flows from servers to queues, which are from supply to demand nodes in our context. Instead, they propose static decisions for flows from queues to servers, akin to Equation (4). Note that if either the static decisions for flows from queues to servers are changed to the proportional decision rule, or multiple flows modelled by the proportional decision rule are allowed to arrive at a single node, then, to the best of our knowledge, the P-Queue model fails to be tractable. The reasons for this is technical and we would not belabour into them here. Please see Remark 3 after the proof of Theorem 1 in Appendix A for more details.

However, in our setting, both of these situations occur. It is therefore, important to note that the fact that we can arrive at a tractable formulation despite the occurrences of two assumption-violating requirements that will lead to intractability in the original P-Queue model, indicates that our methodology here leverages upon the specific context of our problem. This is also why, to this effect, we require new methodology to conduct reformulations of the entropic value-at-risk operator, seen in Theorem 1.

If we must obey the restrictions posed by the P-Queue framework when formulating a model for the joint capacity allocation and job assignment problem, then a number of dummy nodes would need to be created in the network. For example, Figure 4(a) shows a simple network with two demand nodes and two supply nodes, and its reformulation in 4(b) if the restrictions of the P-Queue framework are to be strictly adhered to.

Such formulations amount to unnecessary approximations – they increase the complexity of the problem and, more severely, create a minor time lag in dynamics of the system, as a result of discretization of time and the need for jobs to remain for at least one period in the dummy nodes.



(a) Original network

(b) P-Queue reformulation

Figure 4 A queuing network and its P-Queue reformulation

Constraints

In our model, we consider four types of constraints as follows. These constraints are evaluated at the end of the time period t after all the dynamics has been resolved. As such, these constraints are applied for all $t \in \mathcal{T}$. For each of these constraints, as many copies of the constraint with different parameters can be added as the context requires.

$$\text{Waiting Cost}: \quad \sum_{s\in\mathcal{M}_0} b_i^{t,s} x_i^{t,s} \le C_i^t \sum_{s\in\mathcal{M}_0} x_i^{t,s}, \qquad i\in\mathcal{I},\ \ t\in\mathcal{T}, \tag{12}$$

$$\text{Assignment Cost}: \quad \sum_{i\in\mathcal{I}} \sum_{j:(i,j)\in\mathcal{E}} \sum_{s\in\mathcal{M}} a_{i,j}^{t,s}\ x_i^{t-1,s-1} \frac{\alpha_{i,j}^{t,s}}{\beta_i^{t-1,s-1}} \le A^t, \qquad t\in\mathcal{T}, \tag{13}$$

$$\text{Capacity}: \quad \sum_{s\in\mathcal{M}_0} y_j^{t,s} \le \mathcal{K}_j^t, \qquad j\in\mathcal{J},\ \ t\in\mathcal{T}, \tag{14}$$

$$\text{Allocation Constraint}: \quad \sum_{j\in\mathcal{J}} \kappa_j^t \mathcal{K}_j^t \le B^t, \qquad t\in\mathcal{T}. \tag{15}$$

The first constraint limits the (average) waiting cost at demand node i to a certain target C_i^t. Recall that $b_i^{t,s}$ represents the waiting cost of a job of type i that has waited for s periods without assignment by the end of period t. Notice that with the introduction of the index s, the definition of this cost is now more general. A common choice for $b_i^{t,s}$ would be $b_i^{t,s} \equiv s$, which would then recover the average waiting cost constraint in the traditional queueing sense. Thus, this constraint can be viewed as constraints on the service level.

The second constraint is essentially $\sum_{i\in\mathcal{I}} \sum_{j:(i,j)\in\mathcal{E}} \sum_{s\in\mathcal{M}} a_{i,j}^{t,s} w_{i,j}^{t,s}$, and ensures that the total cost of performing all the job assignments in period t is within a certain target A^t. Recall that $a_{i,j}^{t,s}$ represents the cost of assigning a job of type i that has waited for s periods to resource of type j over the course of period t. Such a form would be useful for describing the cost of wrong or poor assignment, for example in the context of Dai and Shi (2019), where the decision-maker has the option of sending patients of a particular type to wards of a wrong type, thereby incurring costs, but relieving the pressure in the queue.

The third constraint simply requires that the number of jobs that are serviced by resource of type j does not exceed the capacity allocated by the decision-maker to resource j in supply node j. The fourth constraint imposes a global capacity budget, B^t on the capacity allocation decisions of the decision-maker. Here, we assume that budgeting for resource of type j at the start of time period t incurs a cost of κ_j^t per unit of resource. If capacity allocation decisions are not required, then the fourth constraint can be removed and the right-hand side of the third constraint simply becomes a constant.

3. Decision criteria and reformulation

To complete the description of our model, we need to define an objective. One may consider choosing one of the four constraints, in the previous section, as the optimization objective, or some combination of them, such as the total costs, waiting and assignment included. The challenge of doing so, is that the constraints are a function, albeit linear, of the state variables $x_i^{t,s}$ and $y_j^{t,s}$, which are random variables. There are some options here, such as Stochastic Programming or Chance Programming. Both of these options are not necessarily simple, and to the best of our knowledge, we do not know how to make them tractable given the complex dynamics defined in the earlier section.

Instead, we appeal to the Satisficing approach (Brown and Sim 2008, Lam et al. 2013, Jaillet et al. 2022), which can be seen as lying in the intersection of distributionally robust optimization and stochastic programming. Consider the following version of constraints (12)–(14), wrapped within the entropic value at risk operator, $k \log \mathbb{E} \exp(\cdot/k\theta)$ for some parameters $k, \theta > 0$:

$$\text{Waiting Cost}: \qquad k \log \mathbb{E} \exp\left(\left(\sum_{s\in\mathcal{M}_0} (b_i^{t,s} - C_i^t)\, x_i^{t,s} \right) \Big/ k\theta_{1,i}^t \right) \le 0, \quad i \in \mathcal{I}, t \in \mathcal{T}, \tag{16}$$

$$\text{Assignment Cost}: \quad k\log\mathbb{E}\exp\left(\left(\sum_{i\in\mathcal{I}}\sum_{j:(i,j)\in\mathcal{E}}\sum_{s\in\mathcal{M}} a_{i,j}^{t,s}\, x_i^{t-1,s-1}\frac{\alpha_{i,j}^{t,s}}{\beta_i^{t-1,s-1}} - A^t\right)\Big/ k\theta_2^t\right)\le 0,\ t\in\mathcal{T}, \tag{17}$$

$$\text{Capacity}: \quad k\log\mathbb{E}\exp\left(\left(\sum_{s\in\mathcal{M}_0} y_j^{t,s} - \mathcal{K}_j^t\right)\Big/ k\theta_{3,j}^t\right)\le 0, \quad j\in\mathcal{J}, t\in\mathcal{T}, \tag{18}$$

Informally, the expressions that replace $\cdot$ in the operator $k\log\mathbb{E}\exp(\cdot/k\theta)$ are constrained to be smaller than 0. Hence, any violation is penalized more than proportionately by the exp function. As such, the above constraints can be viewed as a relaxation of their respective counterparts, because they can now be violated, but where the frequency and magnitude of the violations are constrained to be small. This is best represented by the following Proposition, which is a traditional result in the satisficing literature. Note that the allocation constraint (15) does not involve any uncertain variables and thus does not require reformulation.

Proposition 2 *If k satisfies $k\log\mathbb{E}\exp(X/k\theta)\le 0$, then for any $\phi>0$, $\mathbb{P}(X\ge\phi)\le\exp(-\phi/k\theta)$.*

Here, ϕ represents the amount of violation, and the Proposition states that the probability of violating by at least ϕ, decays exponentially where the steepness of the decay is controlled by the parameters k and θ. The smaller these two parameters are, the sharper the guarantees against constraint violation, $\mathbb{P}(X\ge\phi)$. In our formulation (16) - (18), k is the global parameter and θ is the idiosyncratic parameter that applies to each constraint. In this vein, k is referred to as the *risk level* in the literature, and each θ is understood to calibrate the level of risk aversion for each constraint. For "hard" constraints (*e.g.* the capacity constraints), which we do not wish to see violated ever, we can set $\theta_{3,j}^t$ to be small; conversely, for "soft" constraints (*e.g.* waiting cost and assignment cost constraints), where some small degree of violation can be tolerated, we can choose larger values for $\theta_{1,i}^t$ and θ_2^t.

As smaller k leads to sharper guarantees, the decision criterion, therefore, is to minimize the risk level k. Notice that the functional $k\log\mathbb{E}\exp(\cdot/k\theta)$ is monotone in k, with limiting behaviour that as $k\to\infty$, $k\log\mathbb{E}\exp(\cdot/k\theta)\to\frac{1}{\theta}\mathbb{E}[\cdot]$, in other words, when the risk level is infinitely large, then we recover the solution in expectation, and as $k\to 0$, then $k\log\mathbb{E}\exp(\cdot/k\theta)\to\operatorname{esssup}\{\cdot/\theta\}$, which is the fully robust case. This leads to our proposed Joint Capacity Allocation and Job Assignment (J-CAJA) model:

$$\begin{aligned}
\text{minimize} \quad & k && && \text{(J-CAJA)}\\
\text{subject to} \quad & k\log\mathbb{E}\exp\left(\left(\sum_{s\in\mathcal{M}_0}\left(b_i^{t,s}-C_i^t\right)x_i^{t,s}\right)\Big/ k\theta_{1,i}^t\right)\le 0, && i\in\mathcal{I}, t\in\mathcal{T},\\
& k\log\mathbb{E}\exp\left(\left(\sum_{i\in\mathcal{I}}\sum_{j:(i,j)\in\mathcal{E}}\sum_{s\in\mathcal{M}} a_{i,j}^{t,s}\, x_i^{t-1,s-1}\frac{\alpha_{i,j}^{t,s}}{\beta_i^{t-1,s-1}} - A^t\right)\Big/ k\theta_2^t\right)\le 0, && t\in\mathcal{T},
\end{aligned}$$

$$
\begin{aligned}
& k \log \mathbb{E} \exp\left(\left(\sum_{s \in \mathcal{M}_0} y_j^{t,s} - \mathcal{K}_j^t \right) \Big/ k\theta_{3,j}^t \right) \leq 0, && j \in \mathcal{J}, t \in \mathcal{T}, \\
& \sum_{j \in \mathcal{J}} \kappa_j^t \mathcal{K}_j^t \leq B^t, && t \in \mathcal{T}, \\
\text{auxiliary} \quad & \beta_i^{t-1,s-1} \leq \beta_i^{t,s} + \sum_{j:(i,j)\in\mathcal{E}} \alpha_{i,j}^{t,s}, && i \in \mathcal{I}, t \in \mathcal{T}, s \in \mathcal{M}, \\
& \beta_i^{t-1,s-1} \geq \beta_i^{t,s}, && i \in \mathcal{I}, t \in \mathcal{T}, s \in \mathcal{M}, \\
\text{dynamics} \quad & x_i^{t,0} = \lambda_i^t, && i \in \mathcal{I}, t \in \mathcal{T}, \\
& x_i^{t,s} = x_i^{t-1,s-1} \frac{\beta_i^{t,s}}{\beta_i^{t-1,s-1}}, && i \in \mathcal{I}, t \in \mathcal{T}, s \in \mathcal{M}, \\
& y_j^{t,0} = \sum_{i:(i,j)\in\mathcal{E}} \sum_{s \in \mathcal{M}} x_i^{t-1,s-1} \frac{\alpha_{i,j}^{t,s}}{\beta_i^{t-1,s-1}}, && j \in \mathcal{J}, t \in \mathcal{T}, \\
& y_j^{t,s} \sim \mathrm{Bin}(y_j^{t-1,s-1}, q_j^{t-1,s-1}), && j \in \mathcal{J}, t \in \mathcal{T}, s \in \mathcal{M}, \\
\text{decisions} \quad & \alpha_{i,j}^{t,s} \geq 0, && (i,j) \in \mathcal{E}, t \in \mathcal{T}, s \in \mathcal{M}, \\
& \beta_i^{t,s} \geq 0, && i \in \mathcal{I}, t \in \mathcal{T}, s \in \mathcal{M}, \\
& \mathcal{K}_j^t \geq 0, && j \in \mathcal{J}, t \in \mathcal{T}.
\end{aligned} \tag{19}
$$

We replace the flow balance constraint (9) with Inequality (19) to ensure that the formulation is convex. While it is known in the literature that the entropic constraints, namely (16), (17), and (18), are convex, they are not yet in a form that can be easily computed. Fortunately, the following sequence of results will reformulate them into convex combinations of the decision variables. Before beginning, for convenience, define $g_i^t(\omega) := \log \mathbb{E}\left[\exp(\omega \lambda_i^t)\right]$ as the log-moment generating function of λ_i^t, for $i \in \mathcal{I}$, $t \in \mathcal{T}$.

Proposition 3 *Suppose Λ_i^t obeys Assumption 1a). Then, $g_i^t(\omega)$ is convex.*

Proposition 4 (Reformulating Waiting Cost Constraints)

1. *For a given period t, the jobs $x_i^{t,s}$, $i \in \mathcal{I}$, $s \in \mathcal{M}_0$ are independent.*
2. *Let $\tilde{b}_i^{t,s} = b_i^{t,s} - C_i^t$. The waiting cost constraints can be reformulated as follows:*

$$
k \log \mathbb{E} \exp\left[\left(\sum_{s \in \mathcal{M}_0} \tilde{b}_i^{t,s} x_i^{t,s} \right) \Big/ k\theta_{1,i}^t \right] = k \sum_{s=0}^{t-1} g_i^{t-s} \left(\frac{\tilde{b}_i^{t,s} \beta_i^{t,s}}{\beta_i^{t-s,0} k\theta_{1,i}^t} \right) + \frac{1}{\theta_{1,i}^t} \sum_{s=t}^{M} \tilde{b}_i^{t,s} \beta_i^{t,s}. \tag{20}
$$

In particular, this expression is jointly convex in k and the decisions $\beta_i^{t,s}$.

In the above result, the two terms represent the contributions from the arrivals (under the first summand, $s < t$, meaning these terms originate from jobs that arrived after the start of the modelling time $t = 0$) and those jobs already in the initial conditions (conversely, $s \geq t$ in the second summand). These terms, in particular, those whose expressions involve the risk level k, can

be viewed as risk-averse corrections of the expected number of jobs. For more details on this, one is referred to the discussion in Zhou et al. (2022). The assignment cost constraints (17) can be reformulated in a similar manner as follows, and will result in similarly two contributions from the arrivals and the initial jobs.

Proposition 5 (Reformulating Assignment Cost Constraints) *The assignment cost constraints can be reformulated as follows:*

$$\begin{aligned} & k \log \mathbb{E} \exp \left(\left(\sum_{i \in \mathcal{I}} \sum_{j:(i,j) \in \mathcal{E}} \sum_{s \in \mathcal{M}} a_{i,j}^{t,s} \; x_i^{t-1,s-1} \frac{\alpha_{i,j}^{t,s}}{\beta_i^{t-1,s-1}} - A^t \right) \Big/ k\theta_2^t \right) \\ =& k \sum_{s=1}^{t-1} \sum_{i \in \mathcal{I}} g_i^{t-s} \left(\sum_{j:(i,j) \in \mathcal{E}} \frac{a_{i,j}^{t,s} \alpha_{i,j}^{t,s}}{\beta_i^{t-s,0} k\theta_2^t} \right) + \frac{1}{\theta_2^t} \sum_{s=t}^{M} \sum_{(i,j) \in \mathcal{E}} a_{i,j}^{t,s} \; \alpha_{i,j}^{t,s} - \frac{A^t}{\theta_2^t} \, . \end{aligned} \tag{21}$$

The right hand side of (21) is jointly convex in k and $\alpha_{i,j}^{t,s}$, $(i,j) \in \mathcal{E}, t \in \mathcal{T}, s \geq 0$.

The reformulation of the capacity constraints (18) is not straightforward. In particular, unlike Proposition 4 and 5, given t and j, $y_j^{t,s}$ are not independent across s. This is because jobs arriving in the same cohort at demand node i may be assigned to the same supply node j at different time periods. As a result, the jobs assigned to resource of type j in supply node j are a mixture of dependent random variables and would not 'split' under the expectation. This is where our formulation deviates from that in Section 2.2, which is aligned with Pipeline Queues. The presence of multiple flows arising from the distributive decision rule being supplied to the same supply node is causing the loss of independence, and which is skirted around by graph structure limitations in the original Pipeline Queues paper.

THEOREM 1 **(Reformulating Capacity Constraints)**. *The capacity constraints can be reformulated as follows:*

$$\begin{aligned} & k \log \mathbb{E} \exp \left[\left(\sum_{s \in \mathcal{M}_0} y_j^{t,s} - \mathcal{K}_j^t \right) \Big/ k\theta_{3,j}^t \right] \\ =& \frac{1}{\theta_{3,j}^t} \sum_{s=t}^{M} r_j^{t,s}(k\theta_{3,j}^t) y_j^{0,s-t} + k \sum_{i:(i,j) \in \mathcal{E}} \sum_{t'=1}^{t-1} g_i^{t'} \left(\sum_{\tau=1}^{t-t'} \frac{r_j^{t,t-t'-\tau}(k\theta_{3,j}^t) \alpha_{i,j}^{t'+\tau,\tau}}{k\theta_{3,j}^t \beta_i^{t',0}} \right) \\ & + \frac{1}{\theta_{3,j}^t} \sum_{i:(i,j) \in \mathcal{E}} \sum_{s=0}^{t-1} \sum_{\tau \geq t-s} r_j^{t,s}(k\theta_{3,j}^t) \alpha_{i,j}^{t-s,\tau} - \frac{\mathcal{K}_j^t}{\theta_{3,j}^t}, \end{aligned} \tag{22}$$

where $r_j^{t,s}(h) = h \log\left(1 - p_j^{t,s} + p_j^{t,s} \exp\left(1/h\right)\right)$. Moreover, the right hand side of (22) is convex in $\alpha_{i,j}^{t,s}$, $(i,j) \in \mathcal{E}, t \in \mathcal{T}, s \in \mathcal{M}_0$.

In Theorem 1, the three terms in (22) namely represent the contributions arising from the jobs initially already being serviced by resource of type j at the start $t = 0$, the jobs that are assigned to resource of type j that had arrived after the start, and the jobs that are assigned to resource of type j but were initially awaiting assignment at the start. In particular, the presence of the summation within the log moment generating function $g_i^{t'}$ in the second term is where the complexity arising from the lack of independence takes shape. More details are described in the proof.

THEOREM 2. *Problem (J-CAJA) can be solved by solving a sequence of convex programs.*

Theorem 2 is critical for tractability. Propositions 4 and 5 and Theorem 1 show that every entropic constraint in (18)-(17) can be reformulated into just one convex constraint. In other words, Problem (J-CAJA) reduces to a convex program with $O(IJT^2)$ constraints. This is interesting for a few reasons. First, it does not grow exponentially in the length of the planning horizon T, and thus avoids the curse of dimensionality. Second, the complexity does not depend on the capacity of the demand and the supply nodes, which usually cannot be achieved in Dynamic Programming. The above two factors make our approach scalable.

Approximating the log-moment generating function term

Note that the functional form $k \cdot g_i^{t-s}(\zeta\beta_i^{t,s}/k)$ for some constant ζ (or $k \cdot g_i^{t-s}(\zeta'\alpha_{i,j}^{t,s}/k)$, where ζ' is a constant) appears in (20), (21) and (22). In case the distribution of Λ_i^t is unknown, or the decision-maker chooses to adopt a data-driven approach, one can estimate this function using a data set. Suppose we are given N data samples $\{\lambda_{i,l}^t\}_{l=1}^N$ of the arrival distribution Λ_i^t , we evaluate the log-moment generating function term using *sample average approximation* (SAA):

$$k \cdot g_i^{t-s}(\zeta\beta_i^{t,s}/k) \approx k \log \left[\sum_{l=1}^{N} \exp\left(\zeta \frac{\beta_i^{t,s}}{k} \lambda_{i,l}^{t-s} \right) \Big/ N \right]. \tag{23}$$

Although the right hand side of (23) is jointly convex in k and $\beta_i^{t,s}$, it is still not easy to optimize $\beta_i^{t,s}$. We propose a successive cutting-plane algorithm to approximate this expression as follows. Without loss of generality, assume $\lambda_{i,1}^{t-s} \leq \ldots \leq \lambda_{i,N}^{t-s}$. We can first approximate the right hand side of (23) by its two asymptotes, $\zeta\lambda_{i,1}^{t-s}\beta_i^{t,s}$ and $\zeta\lambda_{i,N}^{t-s}\beta_i^{t,s}$. Subsequently, we declare a threshold for the accuracy of the estimation of this expression. Whenever the problem is feasible and the threshold is not met, a cutting plane is generated at the present solution of the problem. We continue this procedure until either the threshold is met or the problem becomes infeasible. See Jaillet et al. (2021) for the details.

4. Numerical comparisons

In this section, we illustrate the model (J-CAJA) on two applications, (i) for the nurse allocation problem, which only involves capacity allocation, specifically, the context in Chan et al. (2021), in Section 4.1; and (ii) for the inpatient overflow problem, which only involves job assignment, specifically, the context in Dai and Shi (2019), in Section 4.2. In both of these applications, we compare our model against the proposed state-of-the-art benchmarks. Further details of the comparison studies are also furbished in Appendix B, including how the simulations are ran, how the benchmark models are solved and how the parameters for our model are fixed.

Our choice to perform the numerical study in this particular way is due to our inability to find models in the literature that perform joint capacity allocation and job assignment in a multi-period setting. Instead, we shall illustrate that our model performs strongly against two models catered to either the capacity allocation problem or the job assignment problem, despite our model being constructed to solve the more general problem. It is unclear if either of the benchmark models can be trivially extended to the joint setting that our model considers.

Also important to the discussion is the consequence of using static, as opposed to adaptive decisions, for the job assignment decisions. For brevity, we have deferred this to Appendix B.3. This examination can be interpreted as the comparison of our proposed model against the original Pipeline Queue framework of Bandi and Loke (2018).

4.1. Application I: Nurse allocation problem

Overcrowding in emergency departments (ED) is a perennial problem for hospitals. Overcrowding and long delays can have serious ramifications for the morbidity of patients (Hoot and Aronsky 2008). As such, the allocation of resources in an ED is critical in ensuring positive patient outcomes. There has been greater scrutiny recently in the allocation and scheduling of nurses in EDs. Nurses are the primary managers and caretakers of patients in the ED (Green 2010). Hence, their unavailability can be a major contributor to delays experienced by patients. There are also recent attempts to explore if senior nurses may reduce the burden and utilization of doctors in EDs (Laurant et al. 2018).

In this application, we study the question of nurse allocation and shift scheduling. We assume that there are two areas in the ED, and that there is a fixed capacity of nurses to be scheduled in shifts across these two areas. Each nurse can only serve in one area during a shift. In a planning horizon with periods $0, 1, \ldots, T$, the decision-maker can reallocate the nurses at the end of shifts. Specifically, this means reallocation can happen at periods $t = l\tau$, for $l = 1, 2, \ldots$, where τ is the length of a shift. The goal is to minimize the expected total waiting cost of all of the patients in

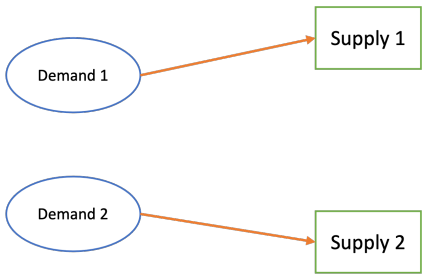


Figure 5 A bipartite network with two dedicated demand-supply dyads

the demand nodes over the planning horizon. This two-dedicated-servers system is shown in Figure 5 with $I = J = 2$. There are a total of $B^t \equiv B$ nurses to allocate between the two supply nodes.

To solve this problem, Chan et al. (2021) consider a discrete-time fluid model with Poisson arrivals and exponential service times, and construct a deterministic fluid approximation for the problem. Given the capacity of the supply nodes, the dynamics between two consecutive decision epochs $l\tau$ and $(l+1)\tau$ can be described by an ordinary differential equation that is solved analytically, which would allow them to determine the system's state at the start of period $(l+1)\tau$ as a function of the state and the reallocation policy at the l-th shift. The authors then solve the finite horizon deterministic problem using a dynamic programming approach.

We consider a total capacity of $B = 60$ resources to be allocated between two supply nodes. This is because the time required to solve the benchmark model, which results in a dynamic program, can be long, as it also depends on B. Deviations of the optimal policy about a mean value is not large, and any smaller choice of B would make comparisons meaningless. We set $T = 12$ and $\tau = 3$. Hence, reallocation occurs 4 times. Our formulation is represented in (24). To ensure comparability with the benchmark model, we adopted actual waiting costs (25), as opposed to the average cost formulation seen in (J-CAJA). Also, constraints are added to model shifts.

$$\min_{\boldsymbol{\alpha},\boldsymbol{\beta},\mathcal{K}} \quad k \tag{24}$$

$$\text{subject to} \quad k \log \mathbb{E} \exp\left(\left(\sum_{s \in \mathcal{M}_0} x_i^{t,s} - C_i^t \right) \Big/ k\theta_{1,i}^t \right) \le 0, \quad i \in \mathcal{I}, t \in \mathcal{T}, \tag{25}$$

$$k \log \mathbb{E} \exp\left(\left(\sum_{s \in \mathcal{M}_0} y_j^{t,s} - \mathcal{K}_j^t \right) \Big/ k\theta_{3,j}^t \right) \le 0, \quad j \in \mathcal{J}, t \in \mathcal{T},$$

$$\sum_{j \in \mathcal{J}} \mathcal{K}_j^t \le B, \quad t \in \mathcal{T},$$

$$\mathcal{K}_j^{3l+1} = \mathcal{K}_j^{3l+2} = \mathcal{K}_j^{3l+3}, \quad j \in \mathcal{J}, l = 0, \ldots, 3.$$

For the purposes of this simulation, we consider Poisson arrivals, which is as assumed by the benchmark model. Here, the mean arrival rate at each demand node is time-homogeneous at 10 per period. We consider this setting under different service time distributions. In all cases, we fix

the mean service times at supply nodes 1 and 2 to 3.3 and 2.5 periods respectively, for all the service time distributions we experimented on. As the benchmark model only utilizes information about the mean service time, the proposed policy by the benchmark model is the same for all the service time distributions. Specifically, we consider 4 service time distributions, as plotted in Figure 6. These distributions represent (i) the assumed Geometric distribution by the benchmark model, (ii) a more right-skewed distribution, which is approximately half-normal, (iii) a more left-skewed distribution, which is approximately log-normal, and finally (iv) a non-continuous and non-monotone example in the form of the two-point distribution.

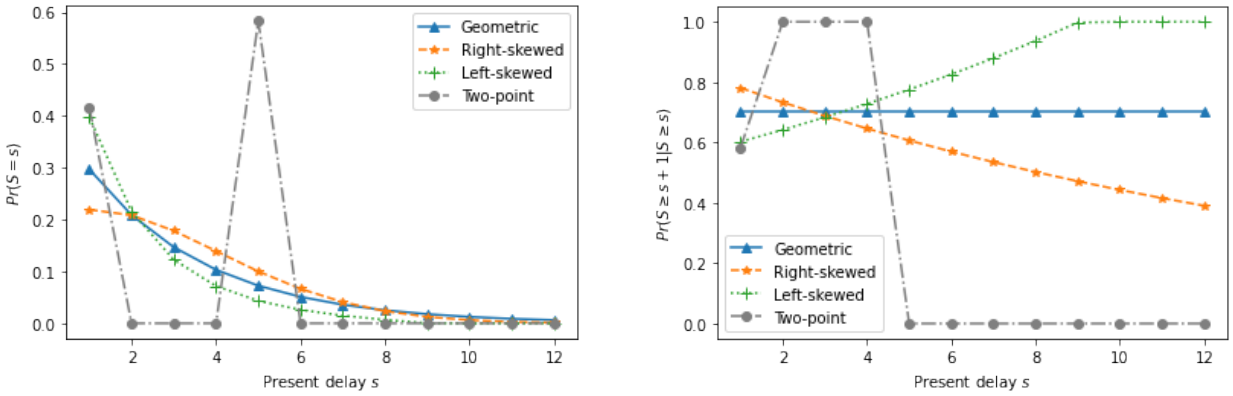


Figure 6 Service (left) and survival (right) probability distributions of job type 1

To solve for the optimal decisions of our model (J-CAJA), we solved for $\mathcal{K}_1^t$ and $\mathcal{K}_2^t$, constraining them to remain the same within shifts, *i.e.* $\mathcal{K}_j^t = \mathcal{K}_j^{t+1}$, $\forall t \not\equiv -1 \pmod{\tau}$. The job assignment decisions are trivial in this case, and are not used. At the start of each shift, only the most recent capacity decisions are implemented, and the model is then evolved till the end of the shift, before we re-solve the model using the new state, in such a rolling horizon fashion. To perform the comparison, we ran 100 simulations. In each simulation, we use the same sample paths for both models, to maintain comparability. More details of the simulations can be found in Appendix B.1.

In Table 2, we present the performance of the policies obtained from our model (J-CAJA) against the benchmark model, in terms of the waiting cost. Here, we reflected both the average waiting cost over the 100 simulations, as well as the 90^{th} percentiles, for both of the queues. The total waiting time across both queues is also computed, which is the objective of the benchmark model. Additionally, in Figure 7, we plot a histogram of the performance differences between our model and the benchmark model for each of the respective cases in Table 2.

From Table 2, we see that our model proposes policies with shorter total waiting times for all the cases, than the benchmark, including the geometric case, which is the distribution assumed in

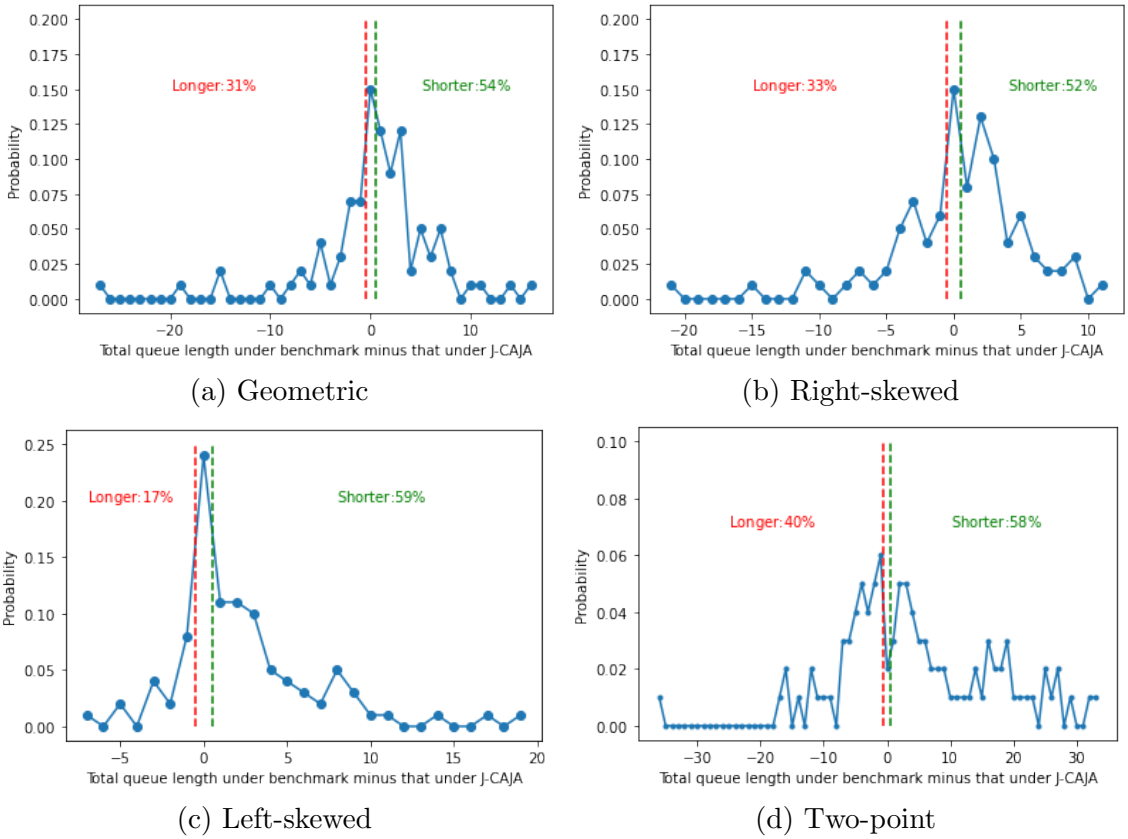
Table 2 Comparison of (J-CAJA) against the reference literature under capacity allocation only setting

	Waiting cost	Geometric		Right-skewed	
		J-CAJA	**Benchmark**	**J-CAJA**	**Benchmark**
Queue 1	**Average**	22.05 (−1%)	22.21	28.48 (−5%)	29.93
	90^{th} **percentile**	55 (−7%)	59	66 (−20%)	83
Queue 2	**Average**	10.06 (−3%)	10.41	12.57 (+10%)	11.47
	90^{th} **percentile**	30 (+11%)	27	40 (+29%)	31
Total	**Average**	32.11 (−2%)	32.62	41.05 (−1%)	41.40
	90^{th} **percentile**	77 (−4%)	80	98 (−6%)	104
	Waiting cost	**Left-skewed**		**Two-point**	
		J-CAJA	**Benchmark**	**J-CAJA**	**Benchmark**
Queue 1	**Average**	9.80 (−16%)	11.66	31.65 (−25%)	41.94
	90^{th} **percentile**	26 (−0%)	26	68 (−24%)	89
Queue 2	**Average**	4.61 (−9%)	5.07	18.73 (+46%)	12.83
	90^{th} **percentile**	10 (−38%)	14	44 (+29%)	34
Total	**Average**	14.41 (−14%)	16.73	50.38 (−8%)	54.77
	90^{th} **percentile**	31 (−9%)	34	101 (−14%)	118

Percentage improvements against the benchmark represented in brackets; negative values signify improvement

the benchmark model. This is surprising, since in this case, the likelihood of completing service in every time period is the same, hence, tracking present delay in our model does not confer us an advantage. Thus, we interpret these results as saying that our model, though not exactly modelling the queue dynamics, but being posed in the transient setting, grants us a very slight edge over models, which very precisely model the queue-server system, but is retrofitted from the steady-state to the transient setting. The histogram showing the individualized comparisons amongst the 100 simulations in Figure 7, show this more clearly – our model performs marginally better than the benchmark model in a good number of situations. The trade-off is that there are some situations where our model performs very much worse than the benchmark. Upon closer inspection, we realize this is because our model is more comfortable with allowing queues to build up in the shorter queue, Queue 2 (as seen from the higher 90^{th} waiting times). In exchange, it is able to achieve reductions in the longer Queue 1, and this is better on the overall. This behaviour might also be related to how our model is a risk-based model – it is assessing that there are greater risk-pooling effects to be gained.

Going back to Table 2, we now examine how the service time distribution affects the results. Recall that the benchmark policy does not change, as the mean service time remains fixed. Hence, changes can be interpreted as the degree in which our model reacts to the service time distributions. Most notably, our performance seems to be stronger for the case of the left-skewed distribution



(a) Geometric (b) Right-skewed

(c) Left-skewed (d) Two-point

Figure 7 Histogram of performance differences between (J-CAJA) and the reference literature for different service time distributions

(14% reductions in waiting times), than for the right-skewed distribution (only 1% reductions in waiting times). The Geometric distribution can be thought of as lying in between these two cases. The left-skewed distribution approximated log-normal service times. In this sense, we can imagine the left-skewed distribution as being long-tailed. We posit that our model works well in this case because of the foundations of our model being rooted in robust optimization. Hence, it has a greater capacity to deal with large variations in the service time. In contrast, the benchmark model, in assuming only mean service times are more likely to suffer from the occasional long-tailed service time. The converse is seen for the right-skewed distribution, which is adapted from the half-normal distribution. In this case, we know that the decay rate in probability of extremely long service times is very quick. Hence, the relative effect of robustness in this case is diminished, and correspondingly we see smaller reductions.

The case of the two-point distribution is interesting, in the sense that our model achieves modest improvements over the benchmark case (around 8% reductions in waiting times), but feels that there should be a wholly different configuration of the optimal decisions, which leads to Queue 1 being drastically shorter than the benchmark case, and Queue 2 being drastically longer.

In summary, what we see here is that our model, when compared to the state-of-the-art, can hold its ground, or perform significantly better in some cases, in the capacity allocation problem, *even though our model can handle joint capacity allocation and job assignment problems*. In particular, our gains are much more pronounced in situations with higher variability, due to our robust approach. As a final minor comment, we would just like to point out a minor deficiency in the solution of the benchmark model. In our simulations, we had noticed that in the first time period, there is a large degeneracy in the optimal solution proposed by the benchmark model, precisely because it is a deterministic approximation, initiated at zero initial state, and the traffic intensity is less than 1. For example, in the above setting, the benchmark model proposes $\mathcal{K}_1^1$ to be anything between 20 to 42, which the model evaluates as having the same objective value. However, if one does not carefully choose the correct decision here (say choosing $\mathcal{K}_1^1 = 20$), then performance of the benchmark model can deteriorate by a further 5%. We do not know how more often such degeneracies may occur, and this can hinder practical implementation.

4.2. Application II: Inpatient overflow problem

In this application, we again return to the question of overcrowding in EDs. Another common operational problem associated with overcrowding is the process of ward allocation after the patient has received critical care. Often, patients wait in the ED area before allocation to a ward corresponding to the medical specialty of their ailment, such as the surgical wards or the cardiology wards. This is called the boarding time. Therefore queues can build if there is a lack of beds in the corresponding wards. There is evidence that points to its effect on both patient outcomes (Singer et al. 2011) and efficiency (Pines et al. 2011). In such situations, it has been proposed to ward patients in non-primary wards that have vacancies, that is, wards that do not belong to the primary specialty of the patient. This is termed as overflow. While this reduces the boarding time of the patients, there are inefficiencies associated with assigning patients to the wrong specialty, such the travel time of physicians, nurses who are inexperienced in provision of care for patients of another specialty, and coordination of equipment and medicine (Rabin et al. 2012). Thus, this has set up an important trade-off between boarding time and its associated costs to the ED, and the overflow and its inefficiencies.

Here, we study precisely this problem. For simplicity, we consider a hospital with two wards for patients corresponding to two different health specialties. Predominantly, patients should be warded at their primary wards. However, it is possible to ward patients at a non-primary ward, incur some wrong matching costs, but potentially alleviate overcrowding, that is, in exchange for shorter overall boarding time. Here, wrong matching costs can include the inefficiencies of moving

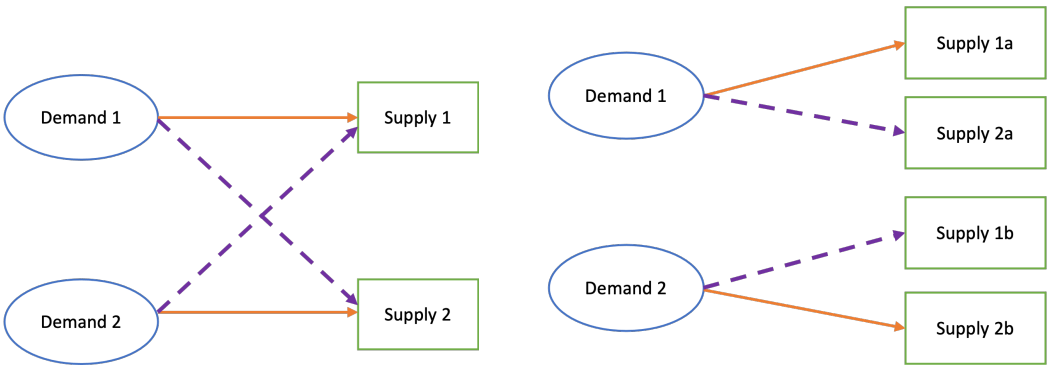


Figure 8 Network structures considered in the reference literature (left) and our proposed adapted model (right)

staff and equipment between the wards. The total number of bed in each ward is known and fixed, hence there are no capacity allocation decisions. We illustrate this in Figure 8 (left).

Here, the goal of the decision-maker is to decide, at every time period, the optimal job assignment decisions from the demand nodes to supply nodes, in order to minimize the total cost of assignment and cost of waiting. Both of these costs correspond to the assignment cost constraint and the waiting cost constraint in (J-CAJA), hence we can easily adapt (J-CAJA) to this problem context. Specifically, here we let $a_{i,j}^{t,s} = 0$ for all t, s if $i = j$, meaning that the right type of assignment incurs no cost, and $a_{i,j}^{t,s} > 0$ for all t, s if $i \neq j$. In this study, we have set $a_{i,j}^{t,s} = 2$ for all t, s and $i \neq j$ and the waiting costs as $b_i^{t,s} = 1$ for all t, s, i. Our formulation is represented in (26). Note that there are no capacity decisions, so $\mathcal{K}_j$ are constants for all $j \in \mathcal{J}$.

In the benchmark literature Dai and Shi (2019), the authors consider a more complicated setting with two timescales, where job completions (discharges) are decided on the longer timescale, and job assignment decisions (warding) are executed on the shorter timescale. While this can be easily handled by our model, which tracks both time indices t and s, we avoid doing so in this application, as it induces complexity in the solution of the benchmark model, and might reduce salience in the differences in performance of the two approaches. Instead, we assume that job assignment decisions are executed on the same timescale as job completions. The benchmark model can accept this, using the same techniques of approximate dynamic programming (ADP). Our argument is that if the benchmark model is sub-optimal under just one timescale, it cannot be better when a second more precise timescale is added, as the errors in the longer timescale easily compound into the shorter timescale. Indeed, the value functions at the shorter timescale are recursively defined in the solution methodology of the reference literature. On the other hand, (J-CAJA) treats all time periods equally and avoids this. The last difference between our model and the reference literature is that the latter adopts a long run average cost formulation. However, in reality, these decisions are implemented in the transient setting. Nonetheless, we can still use the value functions provided by the reference model to seek the optimal job assignment solution.

$$
\begin{aligned}
\min_{\boldsymbol{\alpha},\boldsymbol{\beta}} \quad & k && && (26)\\
\text{subject to} \quad & k\log\mathbb{E}\exp\left(\left(\sum_{s\in\mathcal{M}_0} x_i^{t,s} - C_i^t\right)\Big/ k\theta_{1,i}^t\right) \le 0, && i\in\mathcal{I}, t\in\mathcal{T},\\
& k\log\mathbb{E}\exp\left(\left(\sum_{i\in\mathcal{I}}\sum_{s\in\mathcal{M}} 2x_i^{t-1,s-1}\frac{\alpha_{i,2-i}^{t,s}}{\beta_i^{t-1,s-1}} - A^t\right)\Big/ k\theta_2^t\right) \le 0, && t\in\mathcal{T},\\
& k\log\mathbb{E}\exp\left(\left(\sum_{s\in\mathcal{M}_0} y_j^{t,s} - \mathcal{K}_j\right)\Big/ k\theta_{3,j}^t\right) \le 0, && j\in\mathcal{J}, t\in\mathcal{T}.
\end{aligned}
$$

In this experiment we perform 100 simulations, each up to time period $T = 10$. We assume that arrivals follow a Poisson distribution and that the arrival rate is $\lambda_i = 3$ for $i \in \mathcal{I}$. Capacity is fixed at $\mathcal{K}_j = 10$, for $j \in \mathcal{J}$. However, the service rates are different for the two supply nodes: $\mu_1 = 0.25$, and $\mu_2 = 0.45$. That is, if there is no cross assignment from a demand node to its non-primary supply node, the utilization rate of the two supply nodes will be more than, and less than 1 respectively.

We plot the results of our comparison in Table 3. Here, we reflect the total costs and break them down to their assignment and waiting cost components. Similar to the experiment in Section 4.1, we vary the service time distribution using the same distributions as before in Figure 6.

Table 3 Comparison of (J-CAJA) against the reference literature under job assignment only setting

	Cost	**Geometric**		**Right-skewed**	
		J-CAJA	**Benchmark**	**J-CAJA**	**Benchmark**
Assignment	**Average**	9.30 (−22%)	11.98	9.44 (−31%)	13.74
	90$^{\text{th}}$ percentile	18 (−18%)	22	18 (−25%)	24
Waiting	**Average**	30.89 (+8%)	28.63	36.79 (+12%)	32.91
	90$^{\text{th}}$ percentile	83 (+8%)	77	87 (+6%)	82
Total	**Average**	40.19 (−1%)	40.61	46.23 (−1%)	46.66
	90$^{\text{th}}$ percentile	91 (−1%)	92	97 (+8%)	90
	Cost	**Left-skewed**		**Two-point**	
		J-CAJA	**Benchmark**	**J-CAJA**	**Benchmark**
Assignment	**Average**	1.42 (−43%)	2.50	9.56 (−33%)	14.28
	90$^{\text{th}}$ percentile	4 (−33%)	3	18 (−25%)	24
Waiting	**Average**	4.24 (+1%)	4.18	39.50 (+11%)	35.62
	90$^{\text{th}}$ percentile	13 (+30%)	10	84 (+5%)	80
Total	**Average**	5.66 (−15%)	6.68	49.06 (−2%)	49.90
	90$^{\text{th}}$ percentile	18 (+0%)	18	93 (+1%)	92

Percentage improvements against the benchmark represented in brackets; negative values signify improvement

From Table 3, we can see that in the case of Geometric service times, our model only attains modest improvements of 1% over the benchmark. While this appears small, the reference literature does state in their own experiments that their method gets to within 2% of the true optimal. Similar to the case in Section 4.1, the benchmark model cannot handle non-Geometric service times. In the right-skewed and the two-point distribution cases, their results keep pace with ours, but in the left-skewed case, the gap opens to some 15%. It is for the same reason as before, that is, the longer tailed distribution better allows robustness in our model to kick in, that we see this pattern.

What is also interesting is that our model seems to better favour longer waiting times, in exchange for fewer wrong assignments. In this experiment, the cost of making a wrong assignment is twice the cost of a job remaining one time period in the demand node. The fact that our model is willing to run the larger risk of waiting, may be a reflection of its belief in the ability of the adaptive assignment decisions in reacting to any future time build-ups in queue lengths. Indeed, assigning jobs wrongly incurs cost immediately, whereas betting on shorter waiting times is stochastic and to some degree, our model is able to handle that.

Wrong specification. A closer examination of the reference literature reveals that when a job from a particular demand node is wrongly assigned to another supply node, its job completion time now follows the distribution of the new supply node. In the context of hospital ward assignment, this is akin to saying that if a heart patient were to be assigned to the orthopaedic department, that their length-of-stay would follow orthopaedic patients, as opposed to heart patients.

What might be more realistic in reality might be the kind of network in Figure 8(right), where we have constructed dummy supply nodes to represent the jobs of type i, being served by supply of type j. In this case, we can easily just make the service time distributions of dummy supply nodes $(1,a)$ and $(1,b)$, and $(2,a)$ and $(2,b)$ the same. We do however, need to ensure that capacity constraints on the original supply nodes are observed. This can be done by constraining the capacities of the dummy supply nodes: $\mathcal{K}_{1,a}+\mathcal{K}_{2,a}=\mathcal{K}_{1,b}+\mathcal{K}_{2,b}=10$. Notice that in this formulation, the capacities of the dummy supply nodes enter into the decision variables and thus we arrive at the joint capacity allocation and job assignment setting. It is not clear, to the best of our knowledge, how the reference literature can be applied to this setting.

In Table 4, we illustrate the results when the true job completion times depend only on the demand type, which are the earlier given service rates, but where the benchmark model wrongly assumes that job completion follows the distribution of the type of the supply node. In other words, the true dynamics is modelled by Figure 8(right), and that which we apply (J-CAJA) to solve, but the benchmark is only able to execute Figure 8(left).

It is clear from Table 4, that such model mis-specification can have a large impact on the quality of the decisions. What used to be a margin of 1% improvement of our model over the benchmark

Table 4 Comparison when service completion depends on demand type, but benchmark wrongly assumes dependence on supply type

	Cost	Geometric	
		J-CAJA	**Benchmark**
Assignment	**Average**	8.08 (−16%)	9.58
	90$^{\text{th}}$ percentile	16 (−11%)	18
Queueing	**Average**	53.14 (−5%)	55.75
	90$^{\text{th}}$ percentile	125 (−4%)	130
Total	**Average**	61.22 (−6%)	65.33
	90$^{\text{th}}$ percentile	138 (−8%)	150

has grown to 6%. In particular, both the assignment costs and queueing costs have decreased. We dive deeper to examine how has the differences in policies have led to this. In Figure 9, we plot the assignment decisions adopted by our model and the benchmark over the 100 simulations over the time horizon of $T = 10$. Positive values indicate an assignment of jobs from Team 1 to Team 2, and negative values indicates the reverse.

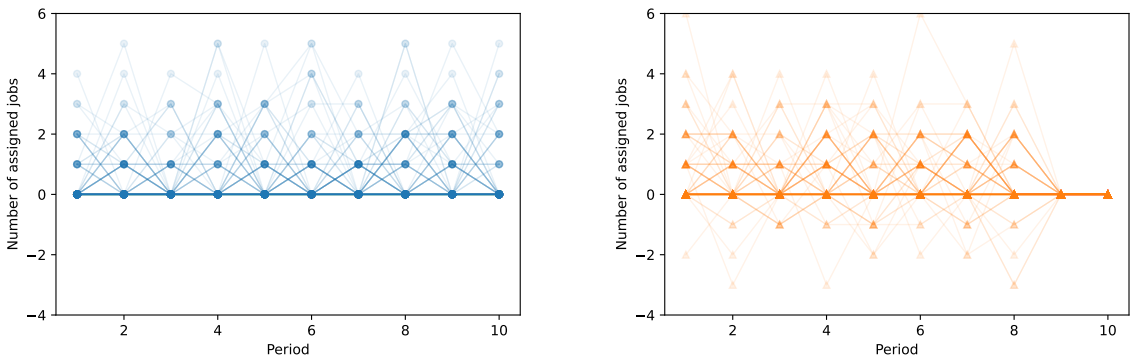


Figure 9 Job assignment from Demand 1 to Supply 2 for the reference literature (left) and our proposed adapted model (right) under 100 simulations

As we can see, the benchmark model only assigns jobs from the supply node with slower completion rates to the other. It wrongly believes that doing so would somehow increase the innate completion rate of the job, thus increasing the rapidity of the job being cleared from the system. Unfortunately, this is not the case. Moreover, the added cost of doing so, is that these longer service time jobs choke up the capacity in the faster supply node. As such, queueing costs also begin to grow correspondingly. Thus, we end up with a situation where the benchmark model also incurs higher queueing costs than our model.

This simple extension illustrates the flexibility of having a joint capacity allocation and job assignment model, where here we have creatively used the capacity decision variables as a means for differentiating job completion by demand type, instead of service type. In the models in the extent literature, without such flexibility, it may become necessary to make critical assumptions that could be violated in reality, and lead to low quality decisions.

5. Conclusion

In this paper, we address the multi-period joint capacity allocation and job assignment problem under uncertainty. Our model is tractable, provides an adaptive policy for job assignment, and is illustrated to perform well in tests against state-of-the-art models. We also introduce new techniques that advance the nascent literature in the Pipeline Queues paradigm.

In some contexts, such as in cloud computing networks, a growing question about pricing has emerged, for example, how to adequately charge the customer for the speed of completing a particular task within the cloud computing network, or whether it is possible to dynamically change the price of the service as a function of the current congestion in clusters. Our model might provide potential avenues for answering this question because we have an optimization formulation where the dual variables, representing shadow prices, would be able to reveal potential insights. Both of these are potential areas for future work and we intend to engage in them moving forward.

References

Armony, Mor, Amy R Ward. 2010. Fair dynamic routing in large-scale heterogeneous-server systems. *Operations Research* **58**(3) 624–637.

Bandi, Chaithanya, Gar Goei Loke. 2018. Exploiting hidden convexity for optimal flow control in queueing networks. *Extracted from SSRN 3190874* .

Bertsimas, Dimitris, Vineet Goyal. 2012. On the power and limitations of affine policies in two-stage adaptive optimization. *Mathematical programming* **134**(2) 491–531.

Bertsimas, Dimitris, Dan A Iancu, Pablo A Parrilo. 2010. Optimality of affine policies in multistage robust optimization. *Mathematics of Operations Research* **35**(2) 363–394.

Brown, David B., Melvyn Sim. 2008. Satisficing measures for analysis of risky positions. *Management Science* **55**(1) 71–84.

Chan, Carri W, Michael Huang, Vahid Sarhangian. 2021. Dynamic server assignment in multiclass queues with shifts, with applications to nurse staffing in emergency departments. *Operations Research* **69**(6) 1936–1959.

Correia, Isabel, Stefan Nickel, Francisco Saldanha-da Gama. 2018. A stochastic multi-period capacitated multiple allocation hub location problem: Formulation and inequalities. *Omega* **74** 122–134.

Dai, Jim G., Mark Gluzman. 2021. Queueing network controls via deep reinforcement learning. *Stochastic Systems. Forthcoming* .

Dai, Jim G, Sean P Meyn. 1995. Stability and convergence of moments for multiclass queueing networks via fluid limit models. *IEEE Transactions on Automatic Control* **40**(11) 1889–1904.

Dai, Jim G., Pengyi Shi. 2019. Inpatient overflow: An approximate dynamic programming approach. *Manufacturing & Service Operations Management* **21**(4) 894–911.

Ghosh, Supriyo, Pradeep Varakantham, Yossiri Adulyasak, Patrick Jaillet. 2017. Dynamic repositioning to reduce lost demand in bike sharing systems. *Journal of Artificial Intelligence Research* **58** 387–430.

Green, Linda V. 2010. Using queueing theory to alleviate emergency department overcrowding. *Wiley Encyclopedia of Operations Research and Management Science* .

Gupta, Diwakar, Lei Wang. 2008. Revenue management for a primary-care clinic in the presence of patient choice. *Operations Research* **56**(3) 576–592.

He, Shuangchi, Melvyn Sim, Meilin Zhang. 2019. Data-driven patient scheduling in emergency departments: A hybrid robust-stochastic approach. *Management Science* **65**(9) 4123–4140.

Hoot, Nathan R, Dominik Aronsky. 2008. Systematic review of emergency department crowding: causes, effects, and solutions. *Annals of emergency medicine* **52**(2) 126–136.

Jaillet, Patrick, Sanjay Dominik Jena, Tsan Sheng Ng, Melvyn Sim. 2022. Satisficing models under uncertainty. *INFORMS Journal on Optimization. Forthcoming* .

Jaillet, Patrick, Gar Goei Loke, Melvyn Sim. 2021. Strategic manpower planning under uncertainty. *Operations Research. Forthcoming* .

Jaillet, Patrick, Xin Lu. 2014. Online stochastic matching: New algorithms with better bounds. *Mathematics of Operations Research* **39**(3) 624–646.

Johari, Ramesh, Vijay Kamble, Yash Kanoria. 2021. Matching while learning. *Operations Research* **69**(2) 655–681.

Karp, Richard M, Umesh V Vazirani, Vijay V Vazirani. 1990. An optimal algorithm for on-line bipartite matching. *Proceedings of the twenty-second annual ACM symposium on Theory of computing*. 352–358.

Lam, Shao-Wei, Tsan Sheng Ng, Melvyn Sim, Jin-Hwa Song. 2013. Multiple objectives satisficing under uncertainty. *Operations Research* **61**(1) 214–227.

Laurant, Miranda, Mieke van der Biezen, Nancy Wijers, Kanokwaroon Watananirun, Evangelos Kontopantelis, Anneke JAH van Vught. 2018. Nurses as substitutes for doctors in primary care. *Cochrane Database of Systematic Reviews* (7).

Lyu, Guodong, Wang Chi Cheung, Chung-Piaw Teo, Hai Wang. 2019. Multi-objective online ride-matching. *Available at SSRN 3356823* .

Martonosi, Susan E. 2011. Dynamic server allocation at parallel queues. *IIE Transactions* **43**(12) 863–877.

Özkan, Erhun, Amy R Ward. 2020. Dynamic matching for real-time ride sharing. *Stochastic Systems* **10**(1) 29–70.

Pines, Jesse M, Robert J Batt, Joshua A Hilton, Christian Terwiesch. 2011. The financial consequences of lost demand and reducing boarding in hospital emergency departments. *Annals of emergency medicine* **58**(4) 331–340.

Puha, Amber L., Amy R. Ward. 2019. Scheduling an overloaded multiclass many-server queue with impatient customers. *Operations Research & Management Science in the Age of Analytics*. INFORMS, 189–217.

Qi, Wei, Lefei Li, Sheng Liu, Zuo-Jun Max Shen. 2018. Shared mobility for last-mile delivery: Design, operational prescriptions, and environmental impact. *Manufacturing & Service Operations Management* **20**(4) 737–751.

Rabin, Elaine, Keith Kocher, Mark McClelland, Jesse Pines, Ula Hwang, Niels Rathlev, Brent Asplin, N Seth Trueger, Ellen Weber. 2012. Solutions to emergency department boardingand crowding are underused and may need to be legislated. *Health Affairs* **31**(8) 1757–1766.

Reeves, Gary R., James R. Sweigart. 1982. Multiperiod resource allocation with variable technology. *Management Science* **28**(12) 1441–1449.

Riedel, Marco. 1999. Online matching for scheduling problems. *Annual Symposium on Theoretical Aspects of Computer Science*. Springer, 571–580.

Shu, Jia, Mabel C. Chou, Qizhang Liu, Chung-Piaw Teo, I-Lin Wang. 2013. Models for effective deployment and redistribution of bicycles within public bicycle-sharing systems. *Operations Research* **61**(6) 1346–1359.

Singer, Adam J, Henry C Thode Jr, Peter Viccellio, Jesse M Pines. 2011. The association between length of emergency department boarding and mortality. *Academic Emergency Medicine* **18**(12) 1324–1329.

Spivey, Michael Z, Warren B Powell. 2004. The dynamic assignment problem. *Transportation science* **38**(4) 399–419.

Whitt, Ward. 2005. Two fluid approximations for multi-server queues with abandonments. *Operations Research Letters* **33**(4) 363–372.

Zhang, Jiheng. 2013. Fluid models of many-server queues with abandonment. *Queueing Systems* **73**(2) 147–193.

Zhou, Minglong, Gar Goei Loke, Chaithanya Bandi, Zi Qiang Glen Liau, Wilson Wang. 2022. Intraday scheduling with patient re-entries and variability in behaviours. *Manufacturing & Service Operations Management* **24**(1) 561–579.

Appendices

A. Proofs

In this Appendix, we present the proofs that were omitted from the main body of text.

A.1. Proof of Proposition 1.

From one side, assuming S_j^t follows a general distribution, we can calculate $q_j^{t,s}$ by:

$$\begin{aligned} q_j^{t,s} &= \mathbb{P}\big[S_j^t \geq s \,\big|\, S_j^t \geq s-1\big] \\ &= \mathbb{P}\big[S_j^t \geq s, S_j^t \geq s-1\big] / \mathbb{P}\big[S_j^t \geq s-1\big] \\ &= \mathbb{P}\big[S_j^t \geq s\big] / \mathbb{P}\big[S_j^t \geq s-1\big] \end{aligned}$$

From the other side, given $q_j^{t,s}$ for any t and s, we can derive the distribution of S_j^t by:

$$\begin{aligned} \mathbb{P}\big[S_j^t \geq s\big] &= \prod_{\tau=1}^{s} \mathbb{P}\big[S_j^t \geq \tau\big] / \mathbb{P}\big[S_j^t \geq \tau-1\big] \\ &= \prod_{\tau=1}^{s} q_j^{t+\tau,\tau} \end{aligned}$$

□

A.2. Proof of Proposition 2.

From $k \log \mathbb{E} \exp\left(X/k\theta\right) \leq 0$, we get that $\mathbb{E}\left[\exp(X/k\theta)\right] \leq 1$. By Markov's inequality, we have

$$\begin{aligned} \mathbb{P}[X \geq \phi] &= \mathbb{P}\big[\exp(X/k\theta) \geq \exp(\phi/k\theta)\big] \\ &\leq \mathbb{E}\left[\exp(X/k\theta)\right] / \exp(\phi/k\theta) \\ &\leq \exp(-\phi/k\theta). \end{aligned}$$

□

A.3. Proof of Proposition 3.

This is a simple consequence of Hölder's inequality. □

A.4. Proof of Proposition 4.

1. For a given t, we have:

$$x_i^{t,s} = \begin{cases} x_i^{t-1,s-1}\dfrac{\beta_i^{t,s}}{\beta_i^{t-1,s-1}} = \ldots = \lambda_i^{t-s}\dfrac{\beta_i^{t,s}}{\beta_i^{t-s,0}} & s < t \\ x_i^{t-1,s-1}\dfrac{\beta_i^{t,s}}{\beta_i^{t-1,s-1}} = \ldots = x_i^{0,s-t}\dfrac{\beta_i^{t,s}}{\beta_i^{0,s-t}} & s \geq t \end{cases}$$

   As we assume the demand arrivals are independent for all t and $i \in \mathcal{I}$, $x_i^{0,s}$ are constants for all s and $i \in \mathcal{I}$, and all $\beta_i^{t,s}$ are decision variables, we can conclude that the $x_i^{t,s}$'s are independent.

2. As $x_i^{t,s}$ are independent across all s, we get that:

$$\begin{aligned} k\log\mathbb{E}\exp\left[\left(\sum_s \tilde{b}_i^{t,s}x_i^{t,s}\right)\Big/ k\theta_{1,i}^t\right] &= \sum_s k\log\mathbb{E}\exp\left(\tilde{b}_i^{t,s}x_i^{t,s}\Big/ k\theta_{1,i}^t\right) \\ &= \sum_{s=0}^{t-1} k\log\mathbb{E}\exp\left(\frac{\tilde{b}_i^{t,s}\beta_i^{t,s}}{\beta_i^{t-s,0}k\theta_{1,i}^t}\lambda_i^{t-s}\right) + \sum_{s\geq t}\frac{\tilde{b}_i^{t,s}\beta_i^{t,s}}{\beta_i^{0,s-t}\theta_{1,i}^t}x_i^{0,s-t} \\ &= k\sum_{s=0}^{t-1} g_i^{t-s}\left(\frac{\tilde{b}_i^{t,s}\beta_i^{t,s}}{\beta_i^{t-s,0}k\theta_{1,i}^t}\right) + \sum_{s\geq t}\frac{\tilde{b}_i^{t,s}\beta_i^{t,s}}{\theta_{1,i}^t} \end{aligned}$$

   To complete the proof, we show that the right hand side of (20) is *jointly* convex in the decision variables k and $\beta_i^{t,s}$, $i \in \mathcal{I}$, $t \in \mathcal{T}$, $s \geq 0$. Recall that $\beta_i^{t-s,0}$ is not part of the decision variables. Therefore, the function $k \cdot g_i^{t-s}\left(\tilde{b}_i^{t,s}\beta_i^{t,s}/\beta_i^{t-s,0}k\theta_{1,i}^t\right)$ in the right hand side of (20) can be expressed as $k \cdot g_i^{t-s}(\zeta\beta_i^{t,s}/k)$, where $\zeta = \tilde{b}_i^{t,s}/\beta_i^{t-s,0}\theta_{1,i}^t$ is a constant. As $g_i^{t-s}(\cdot)$ is a convex function by Proposition 3, the expression $k \cdot g_i^{t-s}(\zeta\beta_i^{t,s}/k)$ is a perspective function and thus is well-known to be jointly convex in $k > 0$ and $\beta_i^{t,s}$.

□

Remark 2. We prove a stronger version of this Proposition that allows for abandonment in Appendix C.

A.5. Proof of Proposition 5.

The proof is analogous to Proposition 4 and is omitted for brevity. □

A.6. Proof of Theorem 1.

We prove a stronger result of Theorem 1, as is shown in Theorem 1' below, where we assume there is abandonment in the demand nodes. The detailed dynamics of this generalization is presented in Appendix C, and involves only a change in the definition of the dynamics for the state variables in the demand nodes, x.

Proof of Theorem 1, given that Theorem 1' holds. If there is no abandonment in the demand nodes, $f_i^{t,s}=1$ for all $i\in\mathcal{I}, t\in\mathcal{T}$ and $s\in\mathcal{M}$, and thus $\rho(\epsilon, f_i^{t,s})=\epsilon$ for any ϵ. Moreover, for the optimal solution, the equality holds for all of the epigraph constraints on $\eta_{i,j}^{t,s}$. For convenience, we denote $r_j^{t,s}(k\theta_{3,j}^t)=r_j^{t,s}$ for all $s\in\mathcal{M}$ and $j\in\mathcal{J}$, as its argument is a fixed constant for a given fixed k. In this situation, each of the terms that appear in the final expression of Theorem 1' has the following simplification:

For $1\le t'\le t-1$,

$$\begin{aligned}
\eta_i^{t-t'+1,1} =&\eta_{i,j}^{t-t'+2,2}+r_j^{t,t'-1}\alpha_{i,j}^{t-t'+1,1}\\
=&\eta_{i,j}^{t-t'+3,3}+r_j^{t,t'-2}\alpha_{i,j}^{t-t'+2,2}+r_j^{t,t'-1}\alpha_{i,j}^{t-t'+1,1}\\
=&\dots\\
=&\eta_{i,j}^{t-1,t'-1}+\sum_{\tau=1}^{t'-2} r_j^{t,t'-\tau}\alpha_{i,j}^{t-t'+\tau,\tau}\\
=&r_j^{t,0}\alpha_{i,j}^{t,t'}+r_j^{t,1}\alpha_{i,j}^{t-1,t'-1}+\sum_{\tau=1}^{t'-2} r_j^{t,t'-\tau}\alpha_{i,j}^{t-t'+\tau,\tau}\\
=&\sum_{\tau=1}^{t'} r_j^{t,t'-\tau}\alpha_{i,j}^{t-t'+\tau,\tau}
\end{aligned}$$

So we have

$$k\sum_{i:(i,j)\in\mathcal{E}}\sum_{t'=1}^{t-1} g_i^{t-t'}(\eta_i^{t-t'+1,1}/k\theta_{3,j}^t\beta_i^{t-t',0}) = k\sum_{i:(i,j)\in\mathcal{E}}\sum_{t'=1}^{t-1} g_i^{t'}\left(\sum_{\tau=1}^{t-t'}\frac{r_j^{t,t-t'-\tau}(k\theta_{3,j}^t)\alpha_{i,j}^{t'+\tau,\tau}}{k\theta_{3,j}^t\beta_i^{t',0}}\right).$$

Similarly, for $t'>t$,

$$\begin{aligned}
\eta_{i,j}^{1,t'-t+1} =&\eta_{i,j}^{2,t'-t+2}+r_j^{t,t-1}\alpha_{i,j}^{1,t'-t+1}\\
=&\eta_{i,j}^{3,t'-t+3}+r_j^{t,t-2}\alpha_{i,j}^{2,t'-t+2}+r_j^{t,t-1}\alpha_{i,j}^{1,t'-t+1}\\
=&\dots\\
=&\eta_{i,j}^{t-1-(t'-M)^+,t'-1-(t'-M)^+}+\sum_{\tau=1}^{t-2-(t'-M)^+} r_j^{t,t-\tau}\alpha_{i,j}^{\tau,t'-t+\tau}\\
=&\sum_{\tau=1}^{t-(t'-M)^+} r_j^{t,t-\tau}\alpha_{i,j}^{\tau,t'-t+\tau}
\end{aligned}\tag{27}$$

We have:

$$\begin{aligned}
\sum_{t'=t}^{t-1+M}\eta_{i,j}^{1,t'-t+1} &= \sum_{t'=t}^{t-1+M}\sum_{\tau=1}^{t-(t'-M)^+} r_j^{t,t-\tau}\alpha_{i,j}^{\tau,t'-t+\tau}\\
&= \sum_{t'=t}^{t-1+M}\sum_{\tau'=t'-t+1}^{t'-(t'-M)^+} r_j^{t,t'-\tau'}\alpha_{i,j}^{\tau'-t'+t,\tau'}
\end{aligned}$$

$$=\sum_{s=0}^{t-1}\sum_{\tau=t-s}^{M} r_j^{t,s}\alpha_{i,j}^{t-s,\tau}$$

We see that Equation (22) is recovered. □

Theorem 1' (Reformulating Capacity Constraints with Abandonment) *The capacity constraints can be reformulated as follows:*

$$\begin{aligned} & k\log\mathbb{E}\exp\left[\left(\sum_{s\in\mathcal{M}_0} y_j^{t,s}-\mathcal{K}_j^t\right)\Big/ k\theta_{3,j}^t\right] \qquad (28)\\ =&\frac{1}{\theta_{3,j}^t}\sum_{s=t}^{M} r_j^{t,s}(k\theta_{3,j}^t)y_j^{0,s-t}+k\sum_{i:(i,j)\in\mathcal{E}}\sum_{t'=1}^{t-1} g_i^{t-t'}(\eta_i^{t-t'+1,1}/k\theta_{3,j}^t\beta_i^{t-t',0})\\ &+\frac{1}{\theta_{3,j}^t}\sum_{i:(i,j)\in\mathcal{E}}\sum_{t'=t}^{t-1+M}\eta_{i,j}^{1,t'-t+1}-\frac{\mathcal{K}_j^t}{\theta_{3,j}^t}, \end{aligned}$$

where the constraints for η are defined more precisely in each of the different regions:

(i) When $1\le t'<t$,

$$\begin{aligned} &k\theta_{3,j}^t\beta_i^{t-1,t'-1}\rho(r_j^{t,0}\alpha_{i,j}^{t,t'}/k\theta_{3,j}^t\beta_i^{t-1,t'-1},f_i^{t-1,t'-1})+r_j^{t,1}\alpha_{i,j}^{t-1,t'-1}\le\eta_{i,j}^{t-1,t'-1}\\ &k\theta_{3,j}^t\beta_i^{t-t'+\tau,\tau}\rho(\eta_{i,j}^{t-t'+\tau+1,\tau+1}/k\theta_{3,j}^t\beta_i^{t-t'+\tau,\tau},f_i^{t-t'+\tau,\tau})+r_j^{t,t'-\tau}\alpha_{i,j}^{t-t'+\tau,\tau}\le\eta_{i,j}^{t-t'+\tau,\tau},\\ &\tau=t'-2,\dots,1 \end{aligned}$$

(ii) When $t\le t'\le M$,

$$\begin{aligned} &k\theta_{3,j}^t\beta_i^{t-1,t'-1}\rho(r_j^{t,0}\alpha_{i,j}^{t,t'}/k\theta_{3,j}^t\beta_i^{t-1,t'-1},f_i^{t-1,t'-1})+r_j^{t,1}\alpha_{i,j}^{t-1,t'-1}\le\eta_{i,j}^{t-1,t'-1}\\ &k\theta_{3,j}^t\beta_i^{\tau,t'-t+\tau}\rho(\eta_{i,j}^{\tau+1,t'-t+\tau+1}/k\theta_{3,j}^t\beta_i^{\tau,t'-t+\tau},f_i^{\tau,t'-t+\tau})+r_j^{t,t-\tau}\alpha_{i,j}^{\tau,t'-t+\tau}\le\eta_{i,j}^{\tau,t'-t+\tau},\\ &\tau=t-2,\dots,1 \end{aligned}$$

(iii) When $M+1\le t'\le M+t-1$,

$$\begin{aligned} &k\theta_{3,j}^t\beta_i^{t-t'+M-1,M-1}\rho\left(\frac{r_j^{t,t'-M}\alpha_{i,j}^{t-t'+M,M}}{k\theta_{3,j}^t\beta_i^{t-t'+M-1,M-1}},f_i^{t-t'+M-1,M-1}\right)+r_j^{t,t'-M+1}\alpha_{i,j}^{t-t'+M-1,M-1}\le\eta_{i,j}^{t-t'+M-1,M-1}\\ &k\theta_{3,j}^t\beta_i^{\tau,t'-t+\tau}\rho(\eta_{i,j}^{\tau+1,t'-t+\tau+1}/k\theta_{3,j}^t\beta_i^{\tau,t'-t+\tau},f_i^{\tau,t'-t+\tau})+r_j^{t,t-\tau}\alpha_{i,j}^{\tau,t'-t+\tau}\le\eta_{i,j}^{\tau,t'-t+\tau},\\ &\tau=M+t-t'-2,\dots,1, \end{aligned}$$

where $\rho(x,p)=\log(1-p+pe^x)$.

Proof of Theorem 1'. The left hand side of (28) can be rewritten as follows:

$$\begin{aligned} & k\log\mathbb{E}\exp\left[\left(\sum_{s\in\mathcal{M}_0} y_j^{t,s}-\mathcal{K}_j^t\right)\Big/ k\theta_{3,j}^t\right]\\ =&k\log\mathbb{E}\exp\left[\sum_{s=0}^{t-1} y_j^{t,s}\Big/ k\theta_{3,j}^t\right]+k\log\mathbb{E}\exp\left[\sum_{s=t}^{M} y_j^{t,s}\Big/ k\theta_{3,j}^t\right]-\mathcal{K}_j^t/\theta_{3,j}^t. \qquad (29) \end{aligned}$$

Notice that if $s \geq t$, $y_j^{t,s}$ represents the jobs that have longer service times than the modelling time. In other words, they were already in the system at the start of period 0, and are a fraction of the initial state of the system, $y_j^{0,s-t}$. Thus, they are independent from the other terms, *i.e.*, $\sum_{s=0}^{t-1} y_j^{t,s}$ and $\sum_{s=t}^{M} y_j^{t,s}$ are independent.

For the second term in (29), we have:

$$\begin{aligned} k \log \mathbb{E} \exp \left[\sum_{s=t}^{M} y_j^{t,s} \Big/ k\theta_{3,j}^t \right] &= k \log \mathbb{E} \exp \left[\sum_{s=t}^{M} \text{Bin}\left(y_j^{0,s-t}, p_j^{t,s}\right) \Big/ k\theta_{3,j}^t \right] \\ &= \sum_{s=t}^{M} k \log \mathbb{E} \exp \left[\text{Bin}(y_j^{0,s-t}, p_j^{t,s}) / k\theta_{3,j}^t \right] \\ &= \sum_{s=t}^{M} \frac{r_j^{t,s}(k\theta_{3,j}^t)}{\theta_{3,j}^t} y_j^{0,s-t} \end{aligned}$$

For the first term in (29) we have:

$$\begin{aligned} & k \log \mathbb{E} \exp \left[\sum_{s=0}^{t-1} y_j^{t,s} \Big/ k\theta_{3,j}^t \right] \\ &= k \log \mathbb{E} \left[\mathbb{E} \left[\exp \left[\left(\sum_{s=0}^{t-1} \text{Bin}(y_j^{t-s,0}, p_j^{t,s}) \right) \Big/ k\theta_{3,j}^t \right] \Big| \, y_j^{t-s,0}, s = 1, \ldots, t-1 \right] \right] \\ &= k \log \mathbb{E} \left[\prod_{s=0}^{t-1} \mathbb{E} \left[\exp \left(\text{Bin}(y_j^{t-s,0}, p_j^{t,s}) / k\theta_{3,j}^t \right) \big| \, y_j^{t-s,0}, s = 1, \ldots, t-1 \right] \right] \qquad (30) \\ &= k \log \mathbb{E}_{\left\{ y_j^{t',0} \right\}_{t'=1}^{t}} \left[\prod_{s=0}^{t-1} \exp \left(\frac{r_j^{t,s}}{k\theta_{3,j}^t} y_j^{t-s,0} \right) \right] \\ &= k \log \mathbb{E}_{\left\{ x_i^{t',s} \right\}^*} \left[\exp \left[\sum_{s=0}^{t-1} \frac{r_j^{t,s}}{k\theta_{3,j}^t} y_j^{t-s,0} \right] \right] \qquad (31) \\ &= k \log \mathbb{E}_{\left\{ x_i^{t',s} \right\}^*} \left[\exp \left[\sum_{s=0}^{t-1} \frac{r_j^{t,s}}{k\theta_{3,j}^t} \sum_{\tau=1}^{M} x_i^{t-s-1,\tau-1} \frac{\alpha_{i,j}^{t-s,\tau}}{\beta_i^{t-s-1,\tau-1}} \right] \right] \qquad (32) \end{aligned}$$

where $\{x_i^{t',s}\}_* = \{x_i^{t',s} | 1 \leq t' \leq t, 0 \leq s \leq M\}$. In Equation (31), we have shifted the expectation from the inflows of the supply nodes y, to the state variables of the demand nodes x, since by definition, the inflows are a linear combination of the demand state variables. It suffices now to only consider the dynamics over the demand state variables. At this point, it is not yet possible to evaluate the expectation over the double summations (on indices s and τ). This is because some of the state variables x are independent. For example, given any $s \leq t-2, \tau \geq 2$, we have $x_i^{t-s-1,\tau-1} = \text{Bin}\left(\frac{\beta_i^{t-s-1,\tau-1}}{\beta_i^{t-s-2,\tau-2}} x_i^{t-s-2,\tau-2}, f_i^{t-s-1,\tau-1} \right)$. Thus, by definition $x_i^{t-s-1,\tau-1}$ and $x_i^{t-s-2,\tau-2}$, both of which appears in the double summation, are dependent. In order to proceed, we need to exploit the structure inherent in the definition of the dynamics of the state variables x.

$$
\begin{aligned}
& k\log\mathbb{E}\left[\exp\left[\sum_{s=0}^{t-1}\frac{r_j^{t,s}}{k\theta_{3,j}^t}\sum_{\tau=1}^{M}x_i^{t-s-1,\tau-1}\frac{\alpha_{i,j}^{t-s,\tau}}{\beta_i^{t-s-1,\tau-1}}\right]\right] \\
&= k\log\mathbb{E}\left[\exp\left[\sum_{t'=1}^{t-1}\sum_{\tau=1}^{t'}\frac{r_j^{t,t'-\tau}}{k\theta_{3,j}^t}x_i^{t-t'+\tau-1,\tau-1}\frac{\alpha_{i,j}^{t-t'+\tau,\tau}}{\beta_i^{t-t'+\tau-1,\tau-1}}\right.\right. \\
&\quad +\sum_{t'=t}^{M}\sum_{\tau=t'-t+1}^{t'}\frac{r_j^{t,t'-\tau}}{k\theta_{3,j}^t}x_i^{t-t'+\tau-1,\tau-1}\frac{\alpha_{i,j}^{t-t'+\tau,\tau}}{\beta_i^{t-t'+\tau-1,\tau-1}} \\
&\quad \left.\left.+\sum_{t'=M+1}^{t-1+M}\sum_{\tau=t'-t+1}^{M}\frac{r_j^{t,t'-\tau}}{k\theta_{3,j}^t}x_i^{t-t'+\tau-1,\tau-1}\frac{\alpha_{i,j}^{t-t'+\tau,\tau}}{\beta_i^{t-t'+\tau-1,\tau-1}}\right]\right] \\
&= \sum_{t'=1}^{t-1}k\log\mathbb{E}\left[\exp\left[\sum_{\tau=1}^{t'}\frac{r_j^{t,t'-\tau}}{k\theta_{3,j}^t}x_i^{t-t'+\tau-1,\tau-1}\frac{\alpha_{i,j}^{t-t'+\tau,\tau}}{\beta_i^{t-t'+\tau-1,\tau-1}}\right]\right] \\
&\quad +\sum_{t'=t}^{M}k\log\mathbb{E}\left[\exp\left[\sum_{\tau=t'-t+1}^{t'}\frac{r_j^{t,t'-\tau}}{k\theta_{3,j}^t}x_i^{t-t'+\tau-1,\tau-1}\frac{\alpha_{i,j}^{t-t'+\tau,\tau}}{\beta_i^{t-t'+\tau-1,\tau-1}}\right]\right] \\
&\quad +\sum_{t'=M+1}^{t-1+M}k\log\mathbb{E}\left[\exp\left[\sum_{\tau=t'-t+1}^{M}\frac{r_j^{t,t'-\tau}}{k\theta_{3,j}^t}x_i^{t-t'+\tau-1,\tau-1}\frac{\alpha_{i,j}^{t-t'+\tau,\tau}}{\beta_i^{t-t'+\tau-1,\tau-1}}\right]\right],
\end{aligned}
\tag{33}
$$

Here, we regroup terms in the double summation, according to the index of t', which represents the cohort. In other words, for a given t', all of the terms in the inner summations over τ originate from the same inflow, $x_i^{t-t',0}$ (which happens for the first of the summations), or from the same initial conditions $x_i^{0,t'-t}$ (which happens for the second and third summations). Later, we shall see that the evaluations do indeed reduce to these terms. This is evidenced by the appearance of τ in both of the time indices, the model time index and the present delay index. As such, both indices move simultaneously within the inner summation, as it does in the defined dynamics on the state variables.

Because of this, each cohort defined by t' is also independent of each other, since each cohort is now a random variable function that depends only on either the inflows at period $t-t'$, namely $\lambda^{t-t'}$, which are assumed to be independent, or the initial conditions, namely $x_i^{0,t'-t}$, and are therefore also independent. Thus, linearity across the entropic operator holds under independence, which legitimizes Equation (33).

It leaves now to separately evaluate each of these summations. Here, we show the detailed step-by-step evaluation for the first case when $t' < t$, which corresponds to the first summation. The other cases are analogous and their derivations are omitted for brevity; only the final result is presented.

$$
k\log\mathbb{E}\left[\exp\left[\sum_{\tau=1}^{t'}\frac{r_j^{t,t'-\tau}}{k\theta_{3,j}^t}x_i^{t-t'+\tau-1,\tau-1}\frac{\alpha_{i,j}^{t-t'+\tau,\tau}}{\beta_i^{t-t'+\tau-1,\tau-1}}\right]\right]
$$

$$
\begin{aligned}
=&k\log\mathbb{E}\left[\mathbb{E}\left[\exp\left[\sum_{\tau=1}^{t'}\frac{r_j^{t,t'-\tau}}{k\theta_{3,j}^t}x_i^{t-t'+\tau-1,\tau-1}\frac{\alpha_{i,j}^{t-t'+\tau,\tau}}{\beta_i^{t-t'+\tau-1,\tau-1}}\right]\middle| x_i^{t-t'+\tau-1,\tau-1},\tau=1,\ldots,t'-1\right]\right]\\
=&k\log\mathbb{E}\left[\exp\left[\sum_{\tau=1}^{t'-1}\frac{r_j^{t,t'-\tau}}{k\theta_{3,j}^t}x_i^{t-t'+\tau-1,\tau-1}\frac{\alpha_{i,j}^{t-t'+\tau,\tau}}{\beta_i^{t-t'+\tau-1,\tau-1}}\right]\right. \qquad (34)\\
&\left.\times\mathbb{E}\left[\exp\left[\frac{r_j^{t,0}}{k\theta_{3,j}^t}x_i^{t-1,t'-1}\frac{\alpha_{i,j}^{t,t'}}{\beta_i^{t-1,t'-1}}\right]\middle| x_i^{t-t'+\tau-1,\tau-1},\tau=1,\ldots,t'-1\right]\right]\\
=&k\log\mathbb{E}\left[\exp\left[\sum_{\tau=1}^{t'-1}\frac{r_j^{t,t'-\tau}}{k\theta_{3,j}^t}x_i^{t-t'+\tau-1,\tau-1}\frac{\alpha_{i,j}^{t-t'+\tau,\tau}}{\beta_i^{t-t'+\tau-1,\tau-1}}\right]\right.\\
&\left.\times\ \mathbb{E}\left[\exp\left[\frac{r_j^{t,0}}{k\theta_{3,j}^t}\frac{\alpha_{i,j}^{t,t'}}{\beta_i^{t-1,t'-1}}\mathrm{Bin}(x_i^{t-2,t'-2},\frac{\beta_i^{t-1,t'-1}}{\beta_i^{t-2,t'-2}},f_i^{t-1,t'-1})\right]\right]\right]\\
=&k\log\mathbb{E}\left[\exp\left[\sum_{\tau=1}^{t'-1}\frac{r_j^{t,t'-\tau}}{k\theta_{3,j}^t}x_i^{t-t'+\tau-1,\tau-1}\frac{\alpha_{i,j}^{t-t'+\tau,\tau}}{\beta_i^{t-t'+\tau-1,\tau-1}}\right]\right.\\
&\left.\times\exp\left[\frac{\beta_i^{t-1,t'-1}}{\beta_i^{t-2,t'-2}}x_i^{t-2,t'-2}\rho(r_j^{t,0}\alpha_{i,j}^{t,t'}/k\theta_{3,j}^t\beta_i^{t-1,t'-1},f_i^{t-1,t'-1})\right]\right] \qquad (35)
\end{aligned}
$$

Equation (34) holds because all other x's are constants within the inner conditional expectation, and thus can be moved out of the inner expectation. Note that for any given constant l, the constraint represented by the RHS of Equation (35) bounded above by l is equivalent to:

$$
\begin{cases}
k\log\mathbb{E}\left[\exp\left[\sum_{\tau=1}^{t'-2}\frac{r_j^{t,t'-\tau}}{k\theta_{3,j}^t}x_i^{t-t'+\tau-1,\tau-1}\frac{\alpha_{i,j}^{t-t'+\tau,\tau}}{\beta_i^{t-t'+\tau-1,\tau-1}}\right]\exp\left[\frac{\eta_i^{t-1,t'-1}}{k\theta_{i,j}^t\beta_i^{t-2,t'-2}}x_i^{t-2,t'-2}\right]\right]\le l\\
k\theta_{3,j}^t\beta_i^{t-1,t'-1}\rho(r_j^{t,0}\alpha_{i,j}^{t,t'}/k\theta_{3,j}^t\beta_i^{t-1,t'-1},f_i^{t-1,t'-1})+r_j^{t,1}\alpha_{i,j}^{t-1,t'-1}\le\eta_{i,j}^{t-1,t'-1}
\end{cases}. \qquad (36)
$$

Here, we have performed the reformulation by replacing the LHS of the lower constraint of (36) with $\eta_{i,j}^{t-1,t'-1}$ using its epigraph formulation. This is made possible only because of the fact that the lower constraint is jointly convex in both $\beta_{i,j}^{t-1,t'-1}$ and $\eta_{i,j}^{t-1,t'-1}$. From here, we can proceed with the reformulation by examining the LHS of the upper constraint.

$$
\begin{aligned}
&k\log\mathbb{E}\left[\exp\left[\sum_{\tau=1}^{t'-2}\frac{r_j^{t,t'-\tau}}{k\theta_{3,j}^t}x_i^{t-t'+\tau-1,\tau-1}\frac{\alpha_{i,j}^{t-t'+\tau,\tau}}{\beta_i^{t-t'+\tau-1,\tau-1}}\right]\exp\left[\frac{\eta_i^{t-1,t'-1}}{k\theta_{i,j}^t\beta_i^{t-2,t'-2}}x_i^{t-2,t'-2}\right]\right]\\
=&k\log\mathbb{E}\left[\exp\left[\sum_{\tau=1}^{t'-2}\frac{r_j^{t,t'-\tau}}{k\theta_{3,j}^t}x_i^{t-t'+\tau-1,\tau-1}\frac{\alpha_{i,j}^{t-t'+\tau,\tau}}{\beta_i^{t-t'+\tau-1,\tau-1}}\right]\right.\\
&\left.\times\mathbb{E}\left[\exp\left[\frac{\eta_i^{t-1,t'-1}}{k\theta_{i,j}^t\beta_i^{t-2,t'-2}}x_i^{t-2,t'-2}\right]\middle| x_i^{t-t'+\tau-1,\tau-1},\tau=1,\ldots,t'-2\right]\right]
\end{aligned}
$$

$$
\begin{aligned}
=&k\log\mathbb{E}\left[\exp\left[\sum_{\tau=1}^{t'-2}\frac{r_j^{t,t'-\tau}}{k\theta_{3,j}^t}x_i^{t-t'+\tau-1,\tau-1}\frac{\alpha_{i,j}^{t-t'+\tau,\tau}}{\beta_i^{t-t'+\tau-1,\tau-1}}\right]\right.\\
&\left.\times\mathbb{E}\left[\exp\left[\frac{\eta_i^{t-1,t'-1}}{k\theta_{i,j}^t\beta_i^{t-2,t'-2}}\mathrm{Bin}(x_i^{t-3,t'-3}\frac{\beta_i^{t-2,t'-2}}{\beta_i^{t-3,t'-3}},f_i^{t-2,t'-2})\right]\right]\right]\\
=&k\log\mathbb{E}\left[\exp\left[\sum_{\tau=1}^{t'-2}\frac{r_j^{t,t'-\tau}}{k\theta_{3,j}^t}x_i^{t-t'+\tau-1,\tau-1}\frac{\alpha_{i,j}^{t-t'+\tau,\tau}}{\beta_i^{t-t'+\tau-1,\tau-1}}\right]\right.\\
&\left.\times\exp\left[\frac{\beta_i^{t-2,t'-2}}{\beta_i^{t-3,t'-3}}x_i^{t-3,t'-3}\rho\left(\frac{\eta_i^{t-1,t'-1}}{k\theta_{i,j}^t\beta_i^{t-2,t'-2}},f_i^{t-2,t'-2}\right)\right]\right].
\end{aligned}
\tag{37}
$$

Once again, for any given constant l, the constraint represented by the RHS of Equation (37) being bounded above by l is equivalent to the following:

$$
\begin{cases}
k\log\mathbb{E}\left[\exp\left[\sum_{\tau=1}^{t'-3}\frac{r_j^{t,t'-\tau}}{k\theta_{3,j}^t}x_i^{t-t'+\tau-1,\tau-1}\frac{\alpha_{i,j}^{t-t'+\tau,\tau}}{\beta_i^{t-t'+\tau-1,\tau-1}}\right]\exp\left[\frac{\eta_i^{t-2,t'-2}}{k\theta_{i,j}^t\beta_i^{t-3,t'-3}}x_i^{t-3,t'-3}\right]\right]\leq l\\
k\theta_{3,j}^t\beta_i^{t-2,t'-2}\rho\left(\frac{\eta_i^{t-1,t'-1}}{k\theta_{i,j}^t\beta_i^{t-2,t'-2}},f_i^{t-2,t'-2}\right)+r_j^{t,2}\alpha_{i,j}^{t-2,t'-2}\leq\eta_{i,j}^{t-2,t'-2}
\end{cases}.
\tag{38}
$$

Similar to before, the validity of this reformulation rests upon the convexity of the lower constraint of (38). At this point, we can see that the LHS of the upper constraint of (38) is of the same structure as the upper constraint of (36). Thus, one can proceed along in the same fashion till the index of τ reaches 1. At each point, a perspective constraint of the form similar to the lower constraint of (38) is produced. In eventuality, the remaining term within the entropic operator gives:

$$
\begin{aligned}
&k\log\mathbb{E}\left[\exp\left[x_i^{t-t',0}\eta_i^{t-t'+1,1}/k\theta_{3,j}^t\beta_i^{t-t',0}\right]\right]\\
=&k\log\mathbb{E}\left[\exp\left[\lambda_i^{t-t'}\eta_i^{t-t'+1,1}/k\theta_{3,j}^t\beta_i^{t-t',0}\right]\right]\\
=&kg_i^{t-t'}(\eta_i^{t-t'+1,1}/k\theta_{3,j}^t\beta_i^{t-t',0})
\end{aligned}
\tag{39}
$$

where we had compiled all of these epigraph constraints in totality:

$$
\begin{aligned}
&k\theta_{3,j}^t\beta_i^{t-1,t'-1}\rho(r_j^{t,0}\alpha_{i,j}^{t,t'}/k\theta_{3,j}^t\beta_i^{t-1,t'-1},f_i^{t-1,t'-1})+r_j^{t,1}\alpha_{i,j}^{t-1,t'-1}\leq\eta_{i,j}^{t-1,t'-1}\\
&k\theta_{3,j}^t\beta_i^{t-t'+\tau,\tau}\rho(\eta_{i,j}^{t-t'+\tau+1,\tau+1}/k\theta_{3,j}^t\beta_i^{t-t'+\tau,\tau},f_i^{t-t'+\tau,\tau})+r_j^{t,t'-\tau}\alpha_{i,j}^{t-t'+\tau,\tau}\leq\eta_{i,j}^{t-t'+\tau,\tau},\\
&\tau=t'-2,\ldots,1
\end{aligned}
$$

For brevity, we will not belabour on the proofs for the second (the case when $t\leq t'\leq M$) and the third (the case when $M<t'<t-1+M$) summations. We simply state the results below.

For the second summation where $t \leq t' \leq M$, for any given constant l,

$$k \log \mathbb{E}\left[\exp\left[\sum_{\tau=t'-t+1}^{t'} \frac{r_j^{t,t'-\tau}}{k\theta_{3,j}^t} x_i^{t-t'+\tau-1,\tau-1} \frac{\alpha_{i,j}^{t-t'+\tau,\tau}}{\beta_i^{t-t'+\tau-1,\tau-1}}\right]\right] \leq l$$

can be reformulated as

$$\begin{aligned}
&\eta_{i,j}^{1,t'-t+1}/\theta_{3,j}^t \leq l \\
&k\theta_{3,j}^t \beta_i^{t-1,t'-1} \rho\left(\frac{r_j^{t,0}\alpha_{i,j}^{t,t'}}{k\theta_{3,j}^t \beta_i^{t-1,t'-1}}, f_i^{t-1,t'-1}\right) + r_j^{t,1}\alpha_{i,j}^{t-1,t'-1} \leq \eta_{i,j}^{t-1,t'-1} \\
&k\theta_{3,j}^t \beta_i^{\tau,t'-t+\tau} \rho\left(\frac{\eta_{i,j}^{\tau+1,t'-t+\tau+1}}{k\theta_{3,j}^t \beta_i^{\tau,t'-t+\tau}}, f_i^{\tau,t'-t+\tau}\right) + r_j^{t,t-\tau}\alpha_{i,j}^{\tau,t'-t+\tau} \leq \eta_{i,j}^{\tau,t'-t+\tau}, \qquad \tau = t-2,\ldots,1.
\end{aligned}$$

And for the third summation where $M < t' \leq t-1+M$, for any given constant l,

$$k \log \mathbb{E}\left[\exp\left[\sum_{\tau=t'-t+1}^{M} \frac{r_j^{t,t'-\tau}}{k\theta_{3,j}^t} x_i^{t-t'+\tau-1,\tau-1} \frac{\alpha_{i,j}^{t-t'+\tau,\tau}}{\beta_i^{t-t'+\tau-1,\tau-1}}\right]\right]$$

can be reformulated as

$$\begin{aligned}
&\eta_{i,j}^{1,t'-t+1}/\theta_{3,j}^t \leq l \\
&k\theta_{3,j}^t \beta_i^{t-t'+M-1,M-1} \rho\left(\frac{r_j^{t,t'-M}\alpha_{i,j}^{t-t'+M,M}}{k\theta_{3,j}^t \beta_i^{t-t'+M-1,M-1}}, f_i^{t-t'+M-1,M-1}\right) + r_j^{t,t'-M+1}\alpha_{i,j}^{t-t'+M-1,M-1} \leq \eta_{i,j}^{t-t'+M-1,M-1} \\
&k\theta_{3,j}^t \beta_i^{\tau,t'-t+\tau} \rho\left(\frac{\eta_{i,j}^{\tau+1,t'-t+\tau+1}}{k\theta_{3,j}^t \beta_i^{\tau,t'-t+\tau}}, f_i^{\tau,t'-t+\tau}\right) + r_j^{t,t-\tau}\alpha_{i,j}^{\tau,t'-t+\tau} \leq \eta_{i,j}^{\tau,t'-t+\tau}, \qquad \tau = M-2,\ldots,1.
\end{aligned}$$

Overall, we have:

$$\begin{aligned}
&k \log \mathbb{E}_{\{\lambda_i^{t'}\}_{t'=1}^t}\left[\exp\left[\sum_{s=0}^{t-1} \frac{r_j^{t,s}}{k\theta_{3,j}^t} y_j^{t-s,0}\right]\right] \\
=&\sum_{t'=1}^{t-1} k g_i^{t-t'}(\eta_i^{t-t'+1,1}/k\theta_{3,j}^t \beta_i^{t-t',0}) + \frac{1}{\theta_{3,j}^t} \sum_{t'=t}^{t-1+M} \eta_{i,j}^{1,t'-t+1}
\end{aligned}$$

□

Remark 3. The above proof works for our situation, but not in general for the setting in Bandi and Loke (2018) for two reasons. First, in the original setting of Bandi and Loke (2018), there may be multiple layers of servers and queues or even loops from demand arrival to service completion in the problem structure; here, the longest chain of nodes is 2. This is critical in the reformulation of Equation (32), which benefits from both statistical independence in the resultant structure, and the fact that the boundary conditions are the arrivals, which lead to the $g_i^{t-t'}$ terms. In contrast, in Bandi and Loke (2018), it is the static decision variables that restricts the longest stochastic

chain of nodes to 2, even though that is not true for the overall network. There are consequences to using these static decision rules in general, though we do illustrate it in our setting in Appendix B.3. Second, jobs from the same cohort of arrivals may arrive at different periods in the servers. If this were done in Bandi and Loke (2018), the distributive decision rule might become intractable. It is also the solution method introduced after Equation (33) that is able to resolve this problem, at least within this context of the joint capacity allocation and job assignment setting.

A.7. Proof of Theorem 2.

Note that each constraint in Propositions 4 and 5 and Theorem 1 is monotone in k. Therefore, we can solve Problem (J-CAJA) using interval bisection on k, where each sub-problem evaluates the feasibility of the constraints under a given k. By Propositions 4 and 5 and Theorem 1, these constraints are jointly convex in the decision variables $\alpha_{i,j}^{t,s}$ and $\beta_i^{t,s}$. □

B. Details of Comparison Studies

In this Appendix, we park further details about the comparison studies that we did not have the chance to mention in the main body of the paper. These will include details on how we formulated the benchmark policies and how the simulations were conducted in an apple-to-apple fashion.

B.1. Application I: Nurse allocation problem

The following discussion is devoted to the details of the comparison against the benchmark model of Chan et al. (2021) presented in Section 4.1, in the purely capacity allocation setting.

Problem definition. The continuous-time capacity allocation problem with finite horizon is stated as Problem 2 in Section 3.2 of Chan et al. (2021). The planning horizon is a finite number T. There are N shifts, each with time duration τ. Thus, $T = N\tau$. The service capacity is allocated among I dedicated queues. The total capacity is $\mathcal{K}$. At time points $t = n\tau + 1$ where $n = 0, \ldots, N-1$, by observing the state of the system $z^t = (z_1^t, \ldots, z_I^t)$, with z_i^t denoting the number of customers in queue i, $i = 1, \ldots, I$, the planner makes the capacity allocation decision $\mathcal{K}^t = (\mathcal{K}_1^t, \ldots, \mathcal{K}_I^t)$. $\sum_{i=1}^{I} \mathcal{K}_i^t \leq \mathcal{K}$. The goal is to minimize the total queuing cost over time T.

In our simulation, we set $T = 12$, $\tau = 3$, $N = 4$, $\mathcal{K} = 60$ and $I = 2$. The arrival rates are balanced $\lambda_1 = \lambda_2 = 10$, but with non-homogeneous service rates $\mu_1 = 1/3.3 = 0.3$ and $\mu_2 = 1/2.5 = 0.4$.

Solving for the benchmark. The benchmark is solved via a dynamic programming (DP) approach, where the transitions of the states are approximated by a fluid model (see Chan et al. (2021)). Denote the set of optimal policies at time point $n\tau$ observing the state x, as $\phi^k(x)$, from which the eventual decision $\mathcal{K}[n]$ is picked. Notice that there can be degeneracy, *i.e.*, that the set of optimal policy does not contain a singleton. This is especially observed during zero initial conditions. In the case of degeneracy, we choose the capacity allocation proportionally closest to the inverse of the service rates of the supply nodes. These optimal policies are computed at the beginning of periods $n\tau$, $n = 0, 1, 2, 3$, after observing the number of customers in the I queues. Notice that when I or $\mathcal{K}$ is large, it becomes time-consuming to solve the model.

Solving for our model. First, we explain how our model is applied to this context, specifically, how the constraints are specified. At the end of period $3n$, $n = 0, 1, 2, 3$, we observe the state of the system to determine the initial condition of our approach: $x_i^{3n,s}$, $y_i^{3n,s}$, $i = 1, 2$, $s = 0, \ldots, M$ and solve our model for a planning horizon starting from period $3n + 1$ to period T. We use the non-normalized version of the queueing cost constraint: $k \log \mathbb{E} \exp\left(\left(\sum_s b_i^{t,s} x_i^{t,s} - C_i^t\right) \Big/ k\theta_{1,i}^t\right) \leq 0$, $\forall i \in \mathcal{I}$, $\forall t > 0$, where $b_i^{t,0} = 0$, $b_i^{t,s} = 1$ for $s \geq 1$, and $C_i^t = 3$, $\forall i \in \mathcal{I}$. Linear constraints $\mathcal{K}_i^{3n'+1} = \mathcal{K}_i^{3n'+2} = \mathcal{K}_i^{3n'+3}$ for $n' \geq n$ are added to ensure that the capacity remains constant over the same shift. Each time

we solve our model, it returns both the capacity allocation decisions and the distributive job assignment decisions. However, since the servers and queues are dedicated, the job assignment decisions are meaningless and we only adopt the capacity decision $\mathcal{K}_i^{3n+1}$ as the capacity of queue i, $\forall i \in \mathcal{I}$, for the following shift.

Simulation setup. We assume empty queues and servers at the beginning of period 1 (end of period 0). Customers are served on a first-come-first-serve basis. The total completion time for each job is random and modelled by the assumed service time distribution, which we vary in our analysis. In moving from continuous to discrete time, we assume that new jobs arrive at the end of each period. This is to preserve the situation where the number of jobs is not known before the decisions, *i.e.* committing to a shift allocation. However, the new jobs do not occur queuing cost within this period. Let x_i^t denote the total numbers of jobs that are waiting in queue i at the beginning of period t. Since each job in the queue incurs the same waiting cost per period no matter how much time it has waited in the queue, we can save the hassle of tracking the present delay in the queues, and set $x_i^{t,0} = x_i^t$ and $x_i^{t,s} = 0,\ \forall s > 0$ in our model. Let $z_i^t \equiv x_i^t + \sum_{s\geq 0} y_i^{t,s}$ denote the total number of jobs in queue i and supply node i at the beginning of period t. $\boldsymbol{z}^t = (z_1^t, \ldots, z_I^t)$ will be used as state of the system at the beginning of period t for the benchmark model.

We adopt the following specific sequence of events in the simulations:

1. At the beginning of period $3n+1$, $n = 0, 1, 2, 3$, the total number of jobs in the servers and queues are observed. At this point, the shift decisions are made – from $\mathcal{K}_i^{3n+1} \in \phi^n(\boldsymbol{z}^{3n})$ in the case of the benchmark, and from the solution of (J-CAJA) in our case. Capacity is then fixed for the duration of the shift: $\mathcal{K}_i^{3n+1} = \mathcal{K}_i^{3n+2} = \mathcal{K}_i^{3n+3}$, $\forall i \in \mathcal{I}$.
2. At period $t = 3n+m$, $m = 1, 2, 3$, the jobs in the supply nodes are serviced. The transition can be modelled by the binomial distribution, that is, $y_i^{t+1,s+1} \sim \text{Bin}(y_i^{t,s}, q_i^{t+1,s+1}), s \geq 0$. The jobs in the queues and new arriving jobs are assigned to the supply nodes: $y_i^{t+1,0} = \min(\mathcal{K}_i^{t+1} - \sum_{s>0} y_i^{t,s}, x_i^t + \lambda_i^t)$. We can also calculate the queue length at the beginning of period $t+1$ by $x_i^{t+1} = x_i^t + \lambda_i^t - y_i^{t+1,0}$.
3. The total queueing cost in each round of simulation is calculated as $\sum_{i=1}^{I}\sum_{t=1}^{T} x_i^t$.

B.2. Application II: Inpatient overflow problem

The following discussion is devoted to the details of the comparison against the benchmark model of Dai and Shi (2019) presented in Section 4.2, in the purely job assignment setting.

Problem definition. The discrete-time job assignment problem is stated in Section 3.1 of Dai and Shi (2019). There are I classes of jobs and J server pools, where $I = J$. Each class of jobs corresponds to one primary server pool, where it bears no cost when assigning a job to its primary server pool.

However, if a job of type i is assigned to server of type j, $(i \neq j)$, it costs $a_{i,j}$. Moreover, if a job of type i is not assigned to any server, it bears a cost of b_i per period. The number of jobs that are assigned from queue i to supply node j at period t is denoted as $w_{i,j}^t$. The goal is to minimize the overall cost: $\sum_{t=1}^{T}\left(\sum_i b_i x_i^t + \sum_{i\neq j} a_{i,j} w_{i,j}^t\right)$.

Solving for the benchmark. Dai and Shi (2019) introduces a infinite-horizon approximate dynamic programming approach (ADP) to solve the problem. They use a set of basis functions to approximate the value function at each state when conducting value iteration. The basis functions include the queue lengths of each type, the squares of the queue lengths, and the waiting cost of a corresponding single-pool system. The value iteration stops when the parameters of the basis functions converge. In the reference literature, due to the large state-space of the ADP formulation, the authors proposed a sampling method to approximate the value function on the fly. In our simulations, we solve exhaustively for the whole state-space, so that our results will reflect the best possible outcome of the ADP approach altogether, ignoring tractability concerns. Furthermore, we focus on the one timescale problem by assuming there is only one checkpoint in each day.

Solving for our model. In the reference literature, the objective is to minimize the sum of both the assignment costs and the queueing costs, whereas these costs are modelled separately as different constraints in (J-CAJA). To resolve this, we let the targets (right hand sides of the constraints) be decision variables and then constrain their sum, which thus represents the total costs, under a larger global target. $\sum_{t=1}^{T}\left(A^t + \sum_{i\in\mathcal{I}} C_i^t\right) \leq D$. This way, our model will deliberate between the trade-offs of the assignment and queueing costs, as opposed to it being specified by the decision-maker as targets. When solving for our model, we recover the proportions for the assignment $\boldsymbol{\alpha}$. As such, the final assignment decisions are state-dependent and may very well be non-integral. We discuss in the simulation setup, how we dealt with this.

Simulation setup. We compare different approaches for a planning horizon $T = 10$. We set up a warm-up periods of 50 beforehand to get the initial condition.The warm-up periods are set so that the state of the system will reach a steady state which benefits the performance of the benchmark policy derived from the infinite-horzion DP. Moreover, in these warm-up periods, we adopt the benchmark policy.

Here is the sequence of events in the simulations:

1. The number of jobs and the service times of the job in each type server are generated from their respective distributions. Note that we generate the service times of each job at all the supply nodes. Each job can be denoted as a $I+1$-dimensional vector: $(t, \tau_1, \ldots, \tau_I)$, where t is the arriving time and τ_i is the service time of the job in supply node i.

2. At the start of each period t, we observe the state of the system. Similar to the simulation setup in Section B.1, we assume that $z_i^t \equiv x_i^t + \sum_{s\geq 0} y_i^{t,s}$ denote the total number of jobs in queue i and supply node i. $\boldsymbol{z}^t = (z_1^t, \ldots, z_I^t)$ is the state of the system at the beginning of period t for the benchmark model. The job assignment decision can be obtained from the optimal policy $\phi(\boldsymbol{z}^t)$. We set the assignment cost targets and waiting cost targets as decision variables, and set an overall target for the sum of these decision variables. The number of job assignment from demand node i to supply node j at this period can be derived by taking the integer part of the solution, that is, $w_{i,j}^t = \lfloor \sum_{s\geq 0} x_i^{t,s} \alpha_{i,j}^{t+1,s+1} / \alpha_{i,j}^{t,s} \rfloor$.
3. Each job in the supply node is denoted as a two-dimentional vector (τ_i, σ_i), where tau_i is the total service time to complete the job and σ_i is the current service time. Each newly assigned job in a supply node can be denoted as $(\tau_i, 0)$. The current service time of a job in a supply node will also be updated: $\sigma_i \leftarrow \sigma_i + 1$. If $\sigma_i \geq \tau_i$, this job is completed and leaves the system. The new jobs join the queues. The cost of this period is recorded: $\sum_i b_i x_i^t + \sum_{i\neq j} a_{i,j} w_{i,j}^t$.
4. The total waiting costs and assignment costs in each round of simulation is summed and recorded.

To deal with the misspecification case where the service time depends on the job, we make the following modifications to the simulation.

1. Each job only needs to be denoted by two-dimensional vector (t, τ), where t is the arriving time and τ is the service time of the job depending on its type.
2. When calculating z_i^t, all the jobs in queue i, supply nodes ia and ib have to be counted.
3. The jobs from queue 1 orginally flowing into supply node 1 now flows to supply node 1a. The jobs from queue 2 orginally flowing into supply node 2 now flows to supply node 2b. The sum of jobs in supply nodes ia and ib can not exceed the capacity of supply node i.

B.3. Comparison against static decision rule in joint capacity allocation and job assignment setting

In this subsection, we study the joint job assignment and capacity allocation setting; and specifically examine the comparison between static and dynamic models. As explained in Section 2.4, the Pipeline Queues framework, if directly applied to this problem, would lead to a static formulation. This comparative study explains why this is a bad idea. We consider the simple network in Figure 8(left), under Poisson arrivals and Exponential service times.

Derivation of static model. Here, we first present the derivation of the static policy model under the Pipeline Queue paradigm. The static model is obtained by using $w_{i,j}^{t,s}$ as the decision variables directly, as opposed to transforming them into a decision rule on the state $\boldsymbol{x}$ as in (J-CAJA). In

such a case, we need to add auxiliary constraints to ensure that no more jobs are assigned out of the demand nodes than there are actual jobs in the nodes. This is not guaranteed in the static case, but is guaranteed by design in the adaptive case. We call this the 'no-underflow constraint':

$$-x_i^{t,s} + \sum_{j:(i,j)\in\mathcal{E}} w_{i,j}^{t,s} \leq 0 \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, s \geq 0$$

If $t \leq s$, this constraint can be reformulated as a linear constraint by writing $x_i^{t,s} = x_i^{0,s-t} - \sum_{j:(i,j)\in\mathcal{E}} \sum_{t'=0}^{t-1} w_{i,j}^{t',s-t+t'}$. However, if $t > s$, $x_i^{t,s}$ is a random variable. We can transform it as an entropy constraint, that is:

$$k \log \mathbb{E} \exp \left[\left(\sum_{j:(i,j)\in\mathcal{E}} w_{i,j}^{t,s} - x_i^{t,s} \right) \Big/ k\theta_{4,i}^{t,s} \right] \leq 0, \tag{40}$$

which has reformulation

$$k \log \mathbb{E} \exp \left[\left(\sum_{j} w_{i,j}^{t,s} - x_i^{t,s} \right) \Big/ k\theta_{3,i}^{t,s} \right] = \sum_{j} \sum_{t'=0}^{s} w_{i,j}^{t-s+t',t'} + k g_i^{t-s} \left(-\frac{1}{k\theta_{3,i}^{t,s}} \right).$$

We can see that the constraint is linear in the decision variables $w_{i,j}^{t,s}$. Similarly, the capacity constraints and the queuing cost constraints can be reformulated to linear expressions in $w_{i,j}^{t,s}$.

The overall optimization model for the static decision rule can be written as follows:

$$\begin{aligned}
\text{minimize} \quad & k \\
\text{subject to} \quad & k \log \mathbb{E} \exp \left(\left(\sum_{s\geq 0} y_j^{t,s} - \mathcal{K}_j^t \right) \Big/ k\theta_{3,j}^t \right) \leq 0, && j \in \mathcal{J}, t \in \mathcal{T}, \\
& k \log \mathbb{E} \exp \left(\left(\sum_{s\geq 0} \left(b_i^{t,s} - C_i^t \right) x_i^{t,s} \right) \Big/ k\theta_{1,i}^t \right) \leq 0, && i \in \mathcal{I}, t \in \mathcal{T}, \\
& k \log \mathbb{E} \exp \left(\left(\sum_{i\in\mathcal{I}} \sum_{j:(i,j)\in\mathcal{E}} \sum_{s\geq 0} a_{i,j}^{t,s} \, w_{i,j}^{t,s} - A^t \right) \Big/ k\theta_2^t \right) \leq 0, && t \in \mathcal{T}, \\
& k \log \mathbb{E} \exp \left[\left(\sum_{j:(i,j)\in\mathcal{E}} w_{i,j}^{t,s} - x_i^{t,s} \right) \Big/ k\theta_{4,i}^{t,s} \right] \leq 0, && i \in \mathcal{I}, t \in \mathcal{T}, \\
& x_i^{t,0} = \lambda_i^t, && i \in \mathcal{I}, t \in \mathcal{T} \setminus \{0\}, \\
& x_i^{t+1,s+1} = x_i^{t,s} - \sum_{j:(i,j)\in\mathcal{E}} w_{i,j}^{t,s}, && i \in \mathcal{I}, t \in \mathcal{T}, s \geq 0, \\
& y_j^{t+1,0} = \sum_{i:(i,j)\in\mathcal{E}} \sum_{s\geq} w_{i,j}^{t,s}, && j \in \mathcal{J}, t \in \mathcal{T}, \\
& y_j^{t+1,s+1} \sim \text{Bin}(y_j^{t,s}, q_j^{t,s}), && j \in \mathcal{J}, t \in \mathcal{T}, \\
& w_{i,j}^{t,s} \geq 0, && (i,j) \in \mathcal{E}, t \in \mathcal{T}, s \geq 0.
\end{aligned} \tag{41}$$

Benchmarks. It is important to note that for the static decision rule, we added the 'no-underflow constraints', which is reformulated into the entropic form (40). This depends on the tightening

parameter $\theta_{4,i}^{t,s}$, which represents how tightly do we want to enforce that this constraint is not violated. As such, there are actually an infinite family of static models, yielding different solutions for different choice of $\theta_{4,i}^{t,s}$. In this comparison, we present two versions of the static model. In the first case, we let $\theta_{4,i}^{t,s} \equiv 1$, which we call the *vanilla static decision rule*. This might lead to violations in the sense that more jobs are assigned that there are available. Thus, in the second case, we tighten this to $\theta_{4,i}^{t,s} \equiv 0.1$ and furthermore, round down $w_{i,j}^{t,s}$, if necessary, to avoid assigning more jobs than there is available. We call this the *adjusted static decision rule*.

Parameter specifications. We keep service times geometric, but change the arrival pattern to $\Lambda^t \sim (15-n)+2n\beta(3,3)$, in other words, its mean is 15, and its support is $[15-n, 15+n]$. The parameter n here, adjusts the variance in the arrivals; if $n=0$, the arrival is deterministic, and if $n=15$, the arrival recovers the scaled beta distribution we had previously used.

Results. We first set $n=2$. In Table 5, we compare the performance of the distributive decision rule against both versions of the static model (vanilla and adjusted static decision rules). Here, we compute, in simulation, the probability, median and 90^{th} percentile of waiting cost constraint and auxiliary constraint violation. While the probabilities of waiting cost constraint violation are 35.8% and 47.8% under the distributive and the vanilla decision rules respectively, the degree of violation is on very different scales. Specifically, the median and 90^{th} percentile of the degree of violation of the waiting cost constraint under the distributive decision rule are 0.137 and 0.296 respectively, which are 30 times smaller than that under the vanilla static decision rule. Similar observation can be seen on the assignment cost constraints.

Table 5 Comparing the distributive and the static decision rules

Constraint type	**Metric**	**Distributive**	**Vanilla static**	**Adjusted static**
Auxiliary constraints	$\mathbb{P}$[violation]	–	62.7%	–
	Median violation	–	0.853	–
	90^{th} percentile	–	1.208	–
Waiting cost constraints	$\mathbb{P}$[violation]	35.8%	47.8%	88.7%
	Median violation	0.137	4.286	7.412
	90^{th} percentile	0.296	8.973	14.727
Assignment cost constraints	$\mathbb{P}$[violation]	27.6%	42.7%	90.1%
	Median violation	0.178	5.008	6.677
	90^{th} percentile	0.734	12.888	13.835

Moreover, under the distributive decision rule, the auxiliary constraint is always satisfied by design. In contrast, we see that the vanilla static decision rule violates the auxiliary constraint

and assigns more jobs than there are available 62.7% of the time. The adjusted static decision rule overcomes this issue by ensuring that the auxiliary constraint is always satisfied (in probability). However, the performance drops: the probability of violating the waiting cost constraint and the degree of violation grow in comparison to the vanilla static model, and are much larger than that under the distributive decision rule.

In summary, the vanilla static decision rule suffers from issues of feasibility. The adjusted static decision rule overcomes this issue but pays a penalty in terms of performance. In contrast, our distributive decision rule not only satisfies the auxiliary constraint, but also achieves a large probability of meeting the waiting cost target.

From here, we allow n to vary and plot the resultant risk levels k Figure 10. From Figure 10, we see that for all models, the risk level scales up with variability in demand n. Moreover, both static models experience much higher risk levels k than our proposed fully adaptive model. The core reason for this is because it is increasingly difficult to prevent constraint violation the longer the modelling time with purely static decisions. Also, explained earlier, the tighter constraint requirement for the Adjusted static model results in a higher risk level k than its Vanilla static counterpart.

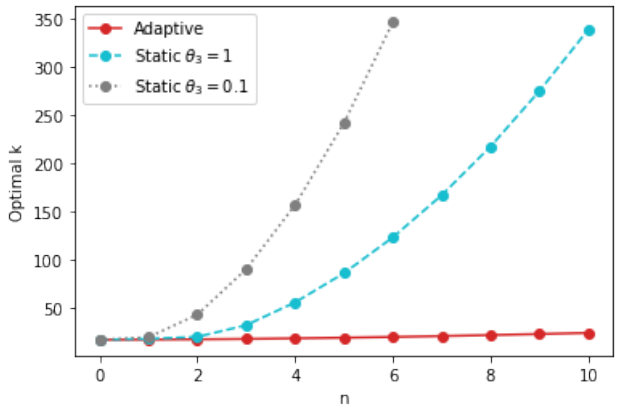


Figure 10 Optimal value of k for adaptive and static decisions

C. Generalizing to abandonment

Modelling abandonment as a probability of departure given the amount of time that the job has currently waited for, s, has previously been seen in the literature (such as Zhang 2013, Whitt 2005). Here, we adopt similar assumption except under the discrete time context, which leads to the following dynamics:

$$x_i^{t,s} \sim \text{Bin}\left(x_i^{t-1,s-1}\frac{\beta_i^{t,s}}{\beta_i^{t-1,s-1}}, f_i^{t,s}\right), \quad i\in\mathcal{I}, t\in\mathcal{T}, s\in\mathcal{M}, \tag{42}$$

where we assume that at period $t-1$, the non-assigned jobs in demand node i which has waited for $s-1$ periods have a probability of $1-f_i^{t,s}$ to abandon the queue. Proposition 4' reformulates the waiting cost constraints under the assumption of abandonment as convex constraints. The reformulation of assignment cost constraints under abandonment can be done similarly.

Proposition 4'

1. *For a given period t, the jobs $x_i^{t,s}$, $i\in\mathcal{I}$, $s\in\mathcal{M}_0$ are independent.*
2. *Let $\tilde{b}_i^{t,s} = b_i^{t,s} - C_i^t$. The waiting cost constraints can be reformulated as follows:*

$$\begin{aligned} &k\log\mathbb{E}\exp\left[\left(\sum_{s\in\mathcal{M}_0}\tilde{b}_i^{t,s}x_i^{t,s}\right)\Big/ k\theta_{1,i}^t\right] \\ &= kg_i^t(\tilde{b}_i^{t,0}/k\theta_{1,i}^t) + k\sum_{s=1}^{t-1} g_i^{t-s}(\xi_i^{t-s+1,1}/k\theta_{1,i}^t\beta_i^{t-s,0}) + \frac{1}{\theta_{1,i}^t}\sum_{s\geq t}\xi_i^{1,s-t+1}, \end{aligned} \tag{43}$$

where

$$\begin{aligned} &k\theta_{1,i}^t\beta_i^{t,s}\rho(\tilde{b}_i^{t,s}/k\theta_{1,i}^t, f_i^{t,s}) \leq \xi_i^{t,s} \\ &k\theta_{1,i}^t\beta_i^{t-\tau,s-\tau}\rho(\xi_i^{t-\tau+1,s-\tau+1}/k\theta_{1,i}^t\beta_i^{t-\tau,s-\tau}, f_i^{t-\tau,s-\tau}) \leq \xi_i^{t-\tau,s-\tau}, \tau = 1,\ldots,\min(t-1,s-1) \end{aligned}$$

In particular, this expression is jointly convex in k, decisions $\beta_i^{t,s}$, and auxiliary variables $\xi_i^{t,s}$.

Proof of Proposition 4'. [adapted from Jaillet et al. (2021)]

1. We prove this by induction on t. If $t=0$, $x_i^{0,s}$ are constants. If for t, the jobs $x_i^{t,s}$, $i\in\mathcal{I}$, $s\in\mathcal{M}_0$ are independent, consider $E\left[\mathbb{1}_{\{x_i^{t+1,s}\leq m\}}\mathbb{1}_{\{x_{i'}^{t+1,s'}\leq n\}}\right]$. If $s=0$, $x_i^{t+1,s} = \lambda_i^{t+1}$ is the demand arrivals and thus it is trivially true. Suppose now $s\geq 1$, and if $i=i'$, $s\neq s'$.

$$\begin{aligned} \mathbb{E}\left[\mathbb{1}_{\{x_i^{t+1,s}\leq m\}}\mathbb{1}_{\{x_{i'}^{t,s'}\leq n\}}\right] &= \mathbb{E}\left[\mathbb{E}\left[\mathbb{1}_{\{x_i^{t+1,s}\leq m\}}\mathbb{1}_{\{x_{i'}^{t,s'}\leq n\}}\middle|\mathbb{1}_{\{x_i^{t,s-1}\leq m'\}}\right]\right] \\ &= \mathbb{E}\left[\mathbb{1}_{\{x_{i'}^{t,s'}\leq n\}}\mathbb{E}\left[\mathbb{1}_{\{x_i^{t+1,s}\leq m\}}\middle|\mathbb{1}_{\{x_i^{t,s-1}\leq m'\}}\right]\right] \qquad (44) \\ &= \mathbb{E}\left[\mathbb{1}_{\{x_{i'}^{t,s'}\leq n\}}\right]\mathbb{E}\left[\mathbb{E}\left[\mathbb{1}_{\{x_i^{t+1,s}\leq m\}}\middle|\mathbb{1}_{\{x_i^{t,s-1}\leq m'\}}\right]\right] \qquad (45) \\ &= \mathbb{E}\left[\mathbb{1}_{\{x_{i'}^{t,s'}\leq n\}}\right]\mathbb{E}\left[\mathbb{1}_{\{x_i^{t+1,s}\leq m\}}\right] \end{aligned}$$

Equation (44) follows because of the independence between $x_{i'}^{t,s'}$ and $x_i^{t,s-1}$ as is assumed in the induction hypothesis. Equation (45) follows because $\mathbb{E}\left[\mathbb{1}_{\{x_i^{t+1,s}\le m\}}\middle|\mathbb{1}_{\{x_i^{t,s-1}\le m'\}}\right]$ is just a function of $x_i^{t,s-1}$ due to the dynamics defined in (42). Thus independence again allows the splitting of expectations. Now we perform the next step. Again, similar logic applies to $s=0$; Otherwise,

$$\begin{aligned}
\mathbb{E}\left[\mathbb{1}_{\{x_i^{t+1,s}\le m\}}\mathbb{1}_{\{x_{i'}^{t+1,s'}\le n\}}\right] &= \mathbb{E}\left[\mathbb{E}\left[\mathbb{1}_{\{x_i^{t+1,s}\le m\}}\mathbb{1}_{\{x_{i'}^{t+1,s'}\le n\}}\middle|\mathbb{1}_{\{x_i^{t,s-1}\le m'\}}\right]\right] \\
&= \mathbb{E}\left[\mathbb{1}_{\{x_{i'}^{t+1,s'}\le n\}}\mathbb{E}\left[\mathbb{1}_{\{x_i^{t+1,s}\le m\}}\middle|\mathbb{1}_{\{x_i^{t,s-1}\le m'\}}\right]\right] \qquad (46)\\
&= \mathbb{E}\left[\mathbb{1}_{\{x_{i'}^{t+1,s'}\le n\}}\right]\mathbb{E}\left[\mathbb{E}\left[\mathbb{1}_{\{x_i^{t+1,s}\le m\}}\middle|\mathbb{1}_{\{x_i^{t,s-1}\le m'\}}\right]\right] \qquad (47)\\
&= \mathbb{E}\left[\mathbb{1}_{\{x_{i'}^{t+1,s'}\le n\}}\right]\mathbb{E}\left[\mathbb{1}_{\{x_i^{t+1,s}\le m\}}\right],
\end{aligned}$$

where (46) follows because of the independence between $x_i^{t,s-1}$ and $x_{i'}^{t+1,s'}$, as proven in the previous step, and similarly for (47).

2.

$$\begin{aligned}
&k\log\mathbb{E}\exp\left(\left(\sum_{s\in\mathcal{M}_0}\tilde{b}_i^{t,s}x_i^{t,s}\right)\Big/k\theta_{1,i}^t\right)\\
=&\sum_{s=0}^{t-1}k\log\mathbb{E}\exp\left(\tilde{b}_i^{t,s}x_i^{t,s}\Big/k\theta_{1,i}^t\right)+\sum_{s\ge t}k\log\mathbb{E}\exp\left(\tilde{b}_i^{t,s}x_i^{t,s}\Big/k\theta_{1,i}^t\right) \qquad (48)
\end{aligned}$$

Equality (48) is because $x_i^{t,s}$ is independent for different s. For $s=0$,

$$\begin{aligned}
&k\log\mathbb{E}\exp\left(\tilde{b}_i^{t,s}x_i^{t,s}\Big/k\theta_{1,i}^t\right)\\
=&kg_i^t(\tilde{b}_i^{t,0}/k\theta_{1,i}^t)
\end{aligned}$$

Recall that $\rho(x,p)=\log(1-p+pe^x)$. For $0<s<t$,

$$\begin{aligned}
&k\log\mathbb{E}\exp\left(\tilde{b}_i^{t,s}x_i^{t,s}\Big/k\theta_{1,i}^t\right)\\
=&k\log\mathbb{E}\exp\left(\frac{\tilde{b}_i^{t,s}}{k\theta_{1,i}^t}\mathrm{Bin}\left(x_i^{t-1,s-1}\frac{\beta_i^{t,s}}{\beta_i^{t-1,s-1}},f_i^{t,s}\right)\right)\\
=&k\log\mathbb{E}\exp\left(x_i^{t-1,s-1}\frac{\beta_i^{t,s}}{\beta_i^{t-1,s-1}}\rho(\tilde{b}_i^{t,s}/k\theta_{1,i}^t,f_i^{t,s})\right) \qquad (49)
\end{aligned}$$

Similar to the proof of Theorem 1', the RHS of Equation (49) being bounded above by l is equivalent to the following:

$$\begin{cases}
k\log\mathbb{E}\exp\left(\frac{x_i^{t-1,s-1}}{k\theta_{1,i}^t}\frac{\xi_i^{t,s}}{\beta_i^{t-1,s-1}}\right)\le l\\
k\theta_{1,i}^t\beta_i^{t,s}\rho(\tilde{b}_i^{t,s}/k\theta_{1,i}^t,f_i^{t,s})\le\xi_i^{t,s}
\end{cases} \qquad (50)$$

We can proceed with the reformulation by examining the LHS of the upper constraint of (50)

$$\begin{aligned} & k\log\mathbb{E}\exp\left(\frac{x_i^{t-1,s-1}}{k\theta_{1,i}^t}\frac{\xi_i^{t,s}}{\beta_i^{t-1,s-1}}\right) \\ =& k\log\mathbb{E}\exp\left(x_i^{t-2,s-2}\frac{\beta_i^{t-1,s-1}}{\beta_i^{t-2,s-2}}\rho(\xi_i^{t,s}/k\theta_{1,i}^t\beta_i^{t-1,s-1}, f_i^{t-1,s-1})\right) \end{aligned} \tag{51}$$

where the RHS of Equation (51) being bounded above by l is equivalent to the following:

$$\begin{cases} k\log\mathbb{E}\exp\left(\frac{x_i^{t-2,s-2}}{k\theta_{1,i}^t}\frac{\xi_i^{t-1,s-1}}{\beta_i^{t-2,s-2}}\right)\leq l \\ k\theta_{1,i}^t\beta_i^{t-1,s-1}\rho(\xi_i^{t,s}/k\theta_{1,i}^t\beta_i^{t-1,s-1}, f_i^{t-1,s-1})\leq \xi_i^{t-1,s-1} \end{cases}$$

In eventuality, the remaining term within the entropic operator gives:

$$k\log\mathbb{E}\exp\left(x_i^{t-s,0}\frac{\xi_i^{t-s+1,1}}{k\theta_{1,i}^t\beta_i^{t-s,0}}\right) \tag{52}$$

$$=k\log\mathbb{E}\exp\left(\lambda_i^{t-s}\frac{\xi_i^{t-s+1,1}}{k\theta_{1,i}^t\beta_i^{t-s,0}}\right) \tag{53}$$

$$=kg_i^{t-s}(\xi_i^{t-s+1,1}/k\theta_{1,i}^t\beta_i^{t-s,0}), \tag{54}$$

where we had compiled all of these epigraph constraints in totality:

$$\begin{aligned} & k\theta_{1,i}^t\beta_i^{t,s}\rho(\tilde{b}_i^{t,s}/k\theta_{1,i}^t, f_i^{t,s})\leq \xi_i^{t,s} \\ & k\theta_{1,i}^t\beta_i^{t-\tau,s-\tau}\rho(\xi_i^{t-\tau+1,s-\tau+1}/k\theta_{1,i}^t\beta_i^{t-\tau,s-\tau}, f_i^{t-\tau,s-\tau})\leq \xi_i^{t-\tau,s-\tau}, \tau=1,\ldots,s-1 \end{aligned}$$

Similarly, for $s\geq t$, $k\log\mathbb{E}\exp\left(\tilde{b}_i^{t,s}x_i^{t,s}\Big/k\theta_{1,i}^t\right)\leq l$ is equilavent to:

$$\begin{aligned} & \frac{1}{\theta_{1,i}^t}\xi_i^{1,s-t+1}\leq l \\ & k\theta_{1,i}^t\beta_i^{t,s}\rho(\tilde{b}_i^{t,s}/k\theta_{1,i}^t, f_i^{t,s})\leq \xi_i^{t,s} \\ & k\theta_{1,i}^t\beta_i^{t-\tau,s-\tau}\rho(\xi_i^{t-\tau+1,s-\tau+1}/k\theta_{1,i}^t\beta_i^{t-\tau,s-\tau}, f_i^{t-\tau,s-\tau})\leq \xi_i^{t-\tau,s-\tau}, \tau=1,\ldots,t-1, \end{aligned}$$

where we set $\beta_i^{0,s-t}=x_i^{0,s-t}$.

□