Operations Research

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

A Restless Bandit Model for Resource Allocation, Competition, and Reservation

Jing Fu, Bill Moran, Peter G. Taylor

http://pubsonline.informs.org/journal/opre

OPERATIONS RESEARCH
Articles in Advance, pp. 1–16
ISSN 0030-364X (print), ISSN 1526-5463 (online)

Methods

A Restless Bandit Model for Resource Allocation, Competition, and Reservation

Jing Fu,[a] Bill Moran,[b] Peter G. Taylor[a]

[a] School of Mathematics and Statistics, University of Melbourne, Melbourne, Victoria 3010, Australia; [b] Department of Electrical and Electronic Engineering, University of Melbourne, Melbourne, Victoria 3010, Australia

Contact: jing.fu@unimelb.edu.au, https://orcid.org/0000-0003-4615-8391 (JF); wmoran@unimelb.edu.au (BM); taylorpg@unimelb.edu.au (PGT)

Received: April 15, 2018
Revised: September 10, 2018; October 23, 2019
Accepted: June 26, 2020
Published Online in Articles in Advance: March 12, 2021

Subject Classifications: Dynamic programming - Applications: Dynamic Programming - Markov - Finite state
Area of Review: Stochastic Models

https://doi.org/10.1287/opre.2020.2066

Abstract. We study a resource allocation problem with varying requests and with resources of limited capacity shared by multiple requests. It is modeled as a set of heterogeneous restless multiarmed bandit problems (RMABPs) connected by constraints imposed by resource capacity. Following Whittle's relaxation idea and Weber and Weiss' asymptotic optimality proof, we propose a simple policy and prove it to be asymptotically optimal in a regime where both arrival rates and capacities increase. We provide a simple sufficient condition for asymptotic optimality of the policy and, in complete generality, propose a method that generates a set of candidate policies for which asymptotic optimality can be checked. The effectiveness of these results is demonstrated by numerical experiments. To the best of our knowledge, this is the first work providing asymptotic optimality results for such a resource allocation problem and such a combination of multiple RMABPs.

Funding: J. Fu's and P. Taylor's research is supported by the Australian Research Council (ARC) Centre of Excellence for the Mathematical and Statistical Frontiers (ACEMS) and ARC Laureate Fellowship [Grant FL130100039].
Supplemental Material: The e-companion is available at https://doi.org/10.1287/opre.2020.2066.

Keywords: restless bandits • resource sharing • Markov decision process

1. Introduction

1.1. Overview and Motivation

Modern technologies enable Internet resources such as routers, computing servers, and cables to be abstracted from the physical layer to a *virtual* layer, facilitating a quick response to demands for setting up communication networks or processing computing jobs. Virtual servers comprising different sets of physical resources are assigned to arriving customers who use these resources for a period of time and then return them to a pool when they depart.

Such networks are just particular examples of more general systems where users of different types arrive with a desire to be allocated resources of various kinds, to use these resources, and then return them. Users are often indifferent to the precise set of resources that they are allocated, they just require allocation of some resources that will enable them to accomplish the task at hand. In such circumstances, a network manager has the task of deciding whether an arriving customer should be admitted into the system and, if so, which set of resources should be assigned to satisfy their requirements.

In this paper, we describe and analyze a very general model for such systems. Specifically, we study a system in which J *resource pools*, each made up of finite numbers of *resource units* (RUs), await allocation to incoming requests of L different types. We refer to the number of RUs in a resource pool as its *capacity*. Each resource pool is potentially shared and *competed* for by many requests, but *reservation* of RUs for still-to-arrive requests is also allowed. When a request has been accommodated by a resource pool, an appropriate number of RUs of this type are occupied by the request until it leaves the system. The released RUs can be reused by other requests. A request is permitted to occupy RUs from more than one resource pool simultaneously. In this context, the number of requests of the same type that are accommodated by a group of resource pools varies according to a stochastic process, where the transition rates are affected by the resource allocation policy employed. Several such processes associated with the same resource pool are coupled by its capacity limitations.

By strategically assigning requests to appropriate combinations of RUs, we aim to maximize the long-run average revenue, defined as the difference between the long-run average reward earned by serving the requests and the long-run average cost incurred by using the resource pools. Such a resource allocation problem can be easily applied to a rich collection of classical models, such as loss networks in telecommunications, resource allocation for logistic systems, and job assignment in parallel computing.

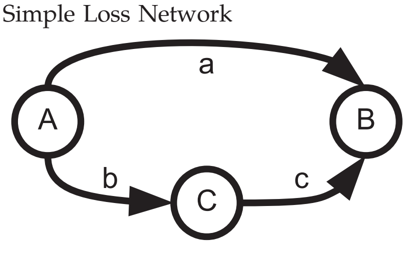
Kelly (1991) published a comprehensive analysis of *loss network* models with and without *alternative routing*. In the latter case, network traffic can be rerouted onto alternative paths when the original path fails or is full. In Kelly (1991), a list of alternative paths as choices of resource pools is given for each call/request. The alternative paths are selected in turn if preceding offered paths are unavailable. In contrast, the manager of a typical resource allocation problem described above is potentially able to change the priorities of paths dynamically. How this should be done is a key focus of this paper.

To illustrate the kind of problem of interest here, consider the simple loss network model shown in Figure 1. Links a, b, and c are abstracted as resource pools with capacities equal to 1, 3, and 3, respectively: link a consists of one channel as an RU, and links b and c each have three channels. Requests asking for a connection from A to B occupying one channel can be served by either path $\{a\}$ or $\{b, c\}$, but requests requiring two channels for each connection from A to B are able to be accommodated only by path $\{b, c\}$. We refer to the former and the latter as type I and type II requests, respectively. An arrival of a type I request results in one of the paths $\{a\}$ and $\{b, c\}$ being chosen by the optimizer depending on current traffic loads on the three links, where links b and c might be shared with existing type II requests. Occupied channels or RUs are released immediately and simultaneously when relevant requests are completed.

Resource allocation problems with small values of L and J, such as the previous example, can be modeled by Markov decision processes (MDPs) and solved through dynamic programming. However, in real-world applications, where L and J are large, resulting in high dimensionality of the state and action spaces, such an approach is often intractable.

In this paper, we use an analysis inspired by techniques applied to restless multiarmed bandit problems (RMABPs). A standard RMABP consists of parallel MDPs, each of which has available a binary action, to activate or not. The coupling occurs because only a limited number of the MDPs can be activated at the same time. Each of the MDPs, referred to as a *bandit process*, has its own individual state-dependent reward rates and transition probabilities when it is activated and when it is not.

Figure 1. Simple Loss Network



Attempts to solve the problem are faced with exponential growth in the size of the state space as the number of parallel bandit processes increases. This class of problems was described by Whittle (1988), who proposed a heuristic management policy that was shown to be asymptotically optimal under nontrivial extra conditions by Weber and Weiss (1990); this policy approaches optimality as the number of bandit processes tends to infinity and the number to be activated at each decision epoch increases in proportion. The policy, subsequently referred to as the *Whittle index policy*, always prioritizes bandit processes with higher state-dependent *indices* that intuitively represent marginal rewards earned by processes if they are selected. The Whittle indices can be computed independently for each bandit process: a process that imposes significantly reduced computational complexity. The Whittle index policy is scalable to a RMABP with a large number of bandit processes. Also, the asymptotic optimality property, if it is satisfied, guarantees a bounded performance degradation in a large-scale system and is appropriate for large problems where optimal solutions are intractable. The nontrivial extra conditions required by the asymptotic optimality proof in Weber and Weiss (1990) are related to proving the existence of a global attractor of a stochastic process.

RMABPs have been widely used in scheduling problems, such as channel detecting (Liu et al. 2012, Wang et al. 2019), job assignments in data centers (Fu et al. 2016), web crawling (Avrachenkov and Borkar 2016), target tracking (Krishnamurthy and Djonin 2007, Le Ny et al. 2010), and job admission control (Niño-Mora 2012, 2019). Here we treat the resource allocation problem described previously as a set of RMABPs coupled by linear inequalities involving random state and action variables.

1.2. Main Contributions

We propose a modified *index policy* that takes into account the capacity constraints of the problem. The index policy prioritizes combinations of RUs with the highest indices, each of which is a real number representing the marginal revenue of using its associated RUs. The policy is simple, scalable, and appropriate for a large-scale resource allocation problem.

Our analysis of asymptotic optimality of the index policy proceeds through a relaxed version of the problem and study of a global attractor of a stochastic process defined in EC.39 in the e-companion. We prove that the stochastic process will almost surely converge to a global attractor in the asymptotic regime regardless of its initial point, and hence the index policy is asymptotically optimal if and only if this global attractor coincides with an optimal solution of the resource allocation problem. Following ideas similar to those of Weber and Weiss (1990), optimality of the global attractor for the resource allocation problem can be deduced from its optimality for the relaxed problem, which can be analyzed with remarkably reduced computational complexity.

A sufficient condition for the global attractor and optimal solution to coincide is that the offered traffic for the entire system is *heavy*,and the resource pools in our system are *weakly coupled*. We rigorously define these concepts in Section 3.3. These results are enunciated in Theorems EC.1 and EC.2 and Corollary EC.1 in Appendix EC.9.3 in the e-companion.

When the abovementioned sufficient conditions are not satisfied, an asymptotically optimal index policy can still exist. In this case, we propose a method that can derive the parameters required by the asymptotically optimal policy. Although asymptotic optimality is not guaranteed, Theorem EC.2 in the e-companion provides a verifiable sufficient condition, less stringent than the one mentioned previously, to check asymptotic optimality of the index policy with adapted parameters. We numerically demonstrate the effectiveness of this method in Section 5.

The index policy exhibits remarkably reduced computational complexity, compared with conventional optimizers, and its potential asymptotic optimality is appropriate for large-scale systems where computational power is a scarce commodity. Furthermore, simulation studies indicate that an index policy can still be good in the prelimit regime. As mentioned earlier, our problem can be seen as a set of RMABPs coupled by the capacity constraints. When the capacities of all resource pools tend to infinity, the index policy reduces to the Whittle index policy because the links between RMABPs no longer exist.

To the best of our knowledge, no existing work has proved asymptotic optimality in resource allocation problems, where resource competition and reservation are potentially permitted, nor has there been a previous analysis of such a combination of multiple, different RMABPs, resulting in a much higher dimensionality of the state space.

The remainder of the paper is organized as follows. In Section 2, we give a detailed definition of the resource allocation problem. In Section 3, we obtain and analyze a relaxed version of the resource allocation problem by applying the Whittle relaxation technique that randomizes its action variables. Based on an optimal solution of the relaxed problem, in Section 4, we propose an algorithm to construct an index policy that is potentially optimal in the limiting regime mentioned previously. To demonstrate the effectiveness of the proposed policies, numerical results are provided in Section 5. In Section 6, we present conclusions.

1.3. Relation to the Literature

The classical MABP is an optimization problem in which only one bandit process (BP) among K BPs can be activated at any one time, whereas all the other $K-1$ BPs are *frozen*: an active BP randomly changes its state, whereas state transitions will not happen to the frozen BPs. In 1974, Gittins and Jones published the well-known *index theorem* for the MABP (Gittins and Jones 1974), and in 1979, Gittins (1979) proved the optimality of a simple *index policy*, subsequently referred to as the *Gittins index policy*. Under the Gittins index policy, an index value, referred to as the *Gittins index*, is associated with each state of each BP, and the BP with the largest index value is activated, whereas all the other BPs are frozen. More details about Gittins indices can be found in Gittins et al. (2011, chapter 2.12 and the references therein).

The optimality of the Gittins index policy for the conventional MABP fails for the general case where the $K-1$ BPs that are not selected can also change their states randomly; such a process is known as a RMABP Whittle (1988). The RMABP allows $M = 1, 2, \ldots, K$ BPs to be active simultaneously. In a similar vein to the Gittins index policy, Whittle assigned a state-dependent index value, referred to as the *Whittle index*, to each BP and always activated the M BPs with the highest indices. The Whittle indices are calculated from a *relaxed* version of the original RMABP obtained by randomizing the action variables. Whittle (1988) defined a property of a RMABP, referred to as *indexability*, under which the *Whittle index policy* exists. Whittle (1988) conjectured that the Whittle index policy, if it exists, is *asymptotically optimal*. Papadimitriou and Tsitsiklis (1999) proved that the optimization of RMABPs is PSPACE-hard in general; nonetheless, Weber and Weiss (1990) were able to establish asymptotic optimality of Whittle index policy under mild conditions.

Niño-Mora (2001) proposed a partial conservation law (PCL) for the optimization of RMABPs; this is an extension of the general conservation law (GCL) published in Bertsimas and Niño-Mora (1996). Later, Niño-Mora (2002) defined a group of problems that satisfies PCL-indexibility and proposed a new index policy that improved the Whittle index. The new index policy was proved to be optimal for problems with PCL-indexibility. PCL-indexibility implies (and is stronger than) Whittle indexibility. A detailed survey about the optimality of bandit problems can be found in Niño-Mora (2007).

Verloop (2016) proved the asymptotic optimality of the Whittle index policy in an extended version of an RMABP, where BPs randomly arrive and depart the system. She proposed an index policy that was not restricted to Whittle indexable models and numerically demonstrated its near optimality. Larrañaga et al. (2015) applied this extended RMABP to a queueing problem assuming convex, nondecreasing functions for both holding costs and measured values of people's impatience. More results on asymptotic optimality of index-like polices can be found in (Fu 2016, chapter IV).

Asymptotically optimal policies for cost-minimization problems in network systems using a fluid approximation technique have been considered in Bäuerle et al. (2000), Bäuerle (2002), Stolyar (2004), Nazarathy and Weiss (2009), and Bertsimas et al. (2015). The fluid approximation to the stochastic optimization problem can be much simpler than the original. A key problem here is to establish an appropriate fluid problem and translate its optimal solution to a policy amenable to the stochastic problem. Asymptotic optimality of the translated stochastic policy can be established if the fluid solution provides an upper/lower bound of the stochastic problem and the policy coincides with this bound asymptotically. The reader is referred to Meyn (2008) for a detailed description of fluid approximation across various models.

Although the fluid approximation technique helps with asymptotic analysis in a wide range of (cost-minimization) network problems, existing results cannot be directly applied to our problem, where the arrival and departure rates of requests are state-dependent and capacity violation over resource pools is strictly forbidden. Our system is always stable for any offered traffic because of the strict capacity constraints. In our case, the form of the corresponding fluid model remains unclear for generic policies. Even given the optimal solution of a well-established fluid model, the synthesis of an explicit policy in the stochastic model remains a challenge.

We adopt another approach, following the ideas of Whittle (1988) and Weber and Weiss (1990). Our asymptotic optimality is derived from an optimal solution of a relaxed version of the stochastic optimization problem. The relaxed problem is still a stochastic optimization problem with a discrete state space. We propose a policy based on intuition captured by the relaxed problem, of which the optimal solution provides a performance upper bound of the original problem. Then, we prove, under certain conditions, that this policy coincides with the upper bound asymptotically. The detailed analysis comprises the main content of the paper.

2. A Resource Allocation Problem

We use $\mathbb{N}_+$ and $\mathbb{N}_0$ to denote the sets of positive and nonnegative integers, respectively, and for any $N \in \mathbb{N}_+$, let $[N]$ represent the set $\{1, 2, \ldots, N\}$ with $[0] = \emptyset$. Let $\mathbb{R}$, $\mathbb{R}_+$, and $\mathbb{R}_0$ be the set of all, positive, and nonnegative reals, respectively.

2.1. System Model

Recall that there are L types of requests and J pools of RUs, all potentially different, with resource pool $j \in [J]$ having capacity C_j that can be dynamically allocated to and released by the L types of requests.

Each request comes with an associated list of candidate resource combinations. Specifically, requests from *request type* (RT) $\ell \in [L]$ can be accommodated by one of a set $\mathscr{P}_\ell$ of candidate *patterns*. One of these candidate patterns will be selected by a policy. Patterns are indexed by $i \in \mathbb{N}_+$. If a request is accommodated by pattern i, $w_{j,i}$ RUs of pool $j \in [J]$ are occupied until the request is completed and departs. We can thus identify pattern i with the *weight vector* $\boldsymbol{w}_i = (w_{j,i})$ that defines its requirement. Preemption or reallocation of requests are not allowed. A request is blocked if there is not enough capacity on any of its corresponding patterns. We might also want to block a request in other circumstances, if accepting it would be detrimental to future performance. In either case, we model the situation where a request is blocked by assigning it to the dummy pattern $d(\ell)$ with the weight vector set to $\mathbf{0}$.

It is possible for different RTs to be satisfied by the same pattern (this occurs, in particular with the dummy pattern). In such cases, we consider there to be multiple copies of each pattern, one for each RT that it can satisfy. This enables us to consider the sets $\mathscr{P}_\ell$ to be mutually exclusive; that is, $\mathscr{P}_{\ell_1} \cap \mathscr{P}_{\ell_2} = \emptyset$ for any $\ell_1 \neq \ell_2$. Given $|\mathscr{P}_\ell|$ patterns for each RT ℓ, we have in total $I = \sum_{\ell \in [L]} |\mathscr{P}_\ell|$ patterns associated with weight vectors $\boldsymbol{w}_i \in \mathbb{N}_0^J$, $i \in [I]$. For any pattern i, let $\ell(i)$ be the unique RT that is satisfied by that pattern.

Let $\mathcal{W}$ be a $J \times I$ matrix with entries $w_{j,i}$. We assume that there is no row and exactly L columns in $\mathcal{W}$ with all zero entries. Each of these zero columns corresponds to one of the dummy patterns $d(\ell)$ where requests of type $\ell \in [L]$ are blocked.

Requests of RT ℓ arrive at the system sequentially, following a Poisson process independent of the arrival processes of other request types, with rates λ_ℓ and the occupation times of the requests accommodated by pattern $i \in \mathscr{P}_\ell$ are exponentially distributed with parameter μ_i again independently of other random elements. Although there might be situations when it is reasonable to assume that the occupation time depends only on the request type ℓ, there might also be cases where the lifetime of a request depends on the resources accommodating it, which is why we allow the occupation time distribution to depend on i. The RUs used to accommodate a request are occupied and released at the same time. Neither the request nor the system knows the lifespan of a request until it is accomplished and departs the system.

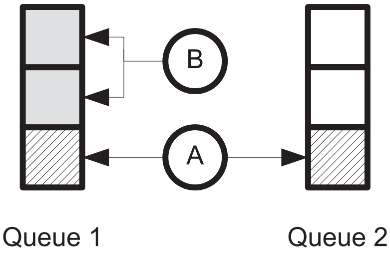
Because there are similarities between our problem and a parallel queueing model, we present a second example to clarify the similarities and differences. Consider two resource pools corresponding to two queues as illustrated in Figure 2, where both capacities are set to three; that is, $J = 2$ and $C_1 = C_2 = 3$. There are two types of requests: if a type 1 request is accommodated in the system, it will simultaneously occupy one RU of both pools; and a type 2 request can be accommodated by two RUs of either pool. In other words, $L = 2$, $\mathscr{P}_1 = \{1,2\}$, $\mathscr{P}_2 = \{3,4,5\}$, patterns 2 and 5 are dummy patterns with $\boldsymbol{w}_2 = \boldsymbol{w}_5 = \boldsymbol{0}$, $\boldsymbol{w}_1 = (1,1)$, $\boldsymbol{w}_3 = (2,0)$, $\boldsymbol{w}_4 = (0,2)$, and $I = 5$.

In this case, the number of occupied RUs in both resource pools may decrease or increase by one simultaneously or by two exclusively for an arrival or departure event. The transition rates are affected by the system controller: if the capacity constraints are not violated, there are two choices, resource pool 1 or 2, for accommodating a type 2 request. The task of a system manager is to find a policy for deciding which of these choices to take in order to maximize some long-term objective. Each choice will result in a parallel queueing model with dependencies between the sizes of queues, between the policy employed, and queue transition rates. As mentioned in Section 1, conventional optimization methods cannot be applied directly when L and J are large.

2.2. A Stochastic Optimization Problem

We focus here an explanation of the stochastic mechanism of the resource allocation problem.

Figure 2. Simple Parallel Queueing Model



An *instantiation* is generated in the memory of the system controller when a request of RT $\ell \in [L]$ is accommodated by a pattern $i \in \mathscr{P}_\ell$. Once the request departs the system, the associated instantiation will be removed from the controller's memory. As requests are accommodated and completed, the number of instantiations associated with each pattern forms a birth-and-death process, indicating the number of requests being served by this pattern. As mentioned in the second example, the birth-and-death processes for all patterns $i \in [I]$ are coupled by capacity constraints and affected by control decisions.

Let $N_i(t)$, $t \geq 0$, represent the number of instantiations for pattern i at time t. The process $N_i(t)$ has state space $\mathscr{N}_i$ that is a discrete, finite set of possible values. The finiteness of $\mathscr{N}_i$ derives from the finite capacities C_j. If $N_i(t)$ is known for all $i \in [I]$, the number of occupied RUs in pool $j \in [J]$ at time t is given by $S_j(t) = \sum_{i\in[I]} w_{j,i} N_i(t)$, which must be less than C_j. The vector $\boldsymbol{N}(t) = (N_i(t) : \ i \in [I])$ is the state variable of the entire system taking values in $\mathscr{N} := \prod_{i\in[I]} \mathscr{N}_i$, where $\prod$ represents Cartesian product. Because the state variables are further subject to capacity constraints to be discussed in Section 2.2.2, $\mathscr{N}$ is larger than necessary. With slightly abused notation, we still refer to $\mathscr{N}$ as the state space of the system.

2.2.1. Action Constraints.

We associate an action variable $a_i(\boldsymbol{n}) \in \{0,1\}$ with process $i \in [I]$ when the system is in state $\boldsymbol{n} \in \mathscr{N}$, and $\boldsymbol{a}(\boldsymbol{n}) = (a_i(\boldsymbol{n}) : \ i \in [I])$. The action variable $a_i(\boldsymbol{n})$ tells us what to do with a potential new request of type $\ell(i)$. If $a_i(\boldsymbol{n}) = 1$, then such a pattern will be allocated to pattern i. The *action constraint*,

$$\sum_{i\in\mathscr{P}_\ell} a_i(\boldsymbol{n}) = 1, \ \forall \ell \in [L], \ \forall \boldsymbol{n} \in \mathscr{N}, \tag{1}$$

ensures that exactly one pattern (which may be the dummy pattern $d(\ell)$) is selected for each RT ℓ and current state $\boldsymbol{n}$.

At any time t, we say that the arrival process for pattern i is *active* or *passive* according to whether $a_i(\boldsymbol{N}(t))$ is 1 or 0, respectively. The birth rate of process $i \in \mathscr{P}_\ell$, $\ell \in [L]$, is λ_ℓ if $a_i(\boldsymbol{N}(t)) = 1$; and zero otherwise. The death rate of process i is $\mu_i N_i(t)$. The time proportion that $a_{d(\ell)}(\boldsymbol{N}(t)) = 1$ is the *blocking probability* for requests of type ℓ.

2.2.2. Capacity Constraints.

To ensure feasibility of an allocation of a request of type $\ell(i)$ to pattern i when the state is $\boldsymbol{n}$, we need

$$\mathcal{W}(\boldsymbol{n} + \boldsymbol{e}_i) \leq \boldsymbol{C}, \tag{2}$$

where $\boldsymbol{e}_i$ is a vector with a one in the ith position and zeros everywhere else and $\boldsymbol{C} \in \mathbb{N}_+^J$ is a vector with entries C_j. In view of the action Constraint (1), a neat way to collect together the Constraints (2) for all $i \in \mathscr{P}_\ell$ is to write them in the form

$$\mathcal{W}(\boldsymbol{n} + \mathcal{E}_\ell \boldsymbol{a}(\boldsymbol{n})) \le \boldsymbol{C},\ \forall \boldsymbol{n} \in \mathcal{N}, \tag{3}$$

where $\mathcal{E}_\ell$ is a diagonal matrix of size I with entries $e_{\ell,i,i} = 1$ if $i \in \mathscr{P}_\ell$ and $e_{\ell,i,i} = 0$ if $i \in [I] \backslash \mathscr{P}_\ell$.

For two different request types ℓ_1 and ℓ_2, a constraint of the form

$$\mathcal{W}\big(\boldsymbol{n} + \mathcal{E}_{\ell_1}\boldsymbol{a}(\boldsymbol{n}) + \mathcal{E}_{\ell_2}\boldsymbol{a}(\boldsymbol{n})\big) \le \boldsymbol{C},\ \forall \boldsymbol{n} \in \mathcal{N}, \tag{4}$$

captures the idea that the action vector $\boldsymbol{a}(\boldsymbol{n})$ must be such that the allocation decisions for ℓ_1 and ℓ_2 ensure enough capacity to implement both of them when both requests arrive simultaneously while the state is $\boldsymbol{n}$. Another way to think about this is that, if a request of type ℓ_1 is allocated to a nondummy pattern i_1 when the state is $\boldsymbol{n}$, the decision for a request of type ℓ_2 when the state is $\boldsymbol{n}$ must satisfy Constraint (3) when the state is $\boldsymbol{n} + \boldsymbol{e}_{i_1}$. In particular, if there is not enough capacity to accommodate a request of type ℓ_2 when the state is $\boldsymbol{n} + \boldsymbol{e}_{i_1}$, then a request of type ℓ_2 must be allocated to the dummy pattern $d(\ell_2)$, when the state is $\boldsymbol{n}$. This can be viewed as giving priority to *reserving* resources for a type ℓ_1 request over a type ℓ_2 request when the state is $\boldsymbol{n}$. As we shall see below, the decision to do this will be made in order to optimize a long-term reward function.

Observing that $\sum_{\ell \in [L]} \mathcal{E}_\ell = I$, we see that the constraint

$$\mathcal{W}(\boldsymbol{n} + \boldsymbol{a}(\boldsymbol{n})) = \mathcal{W}\left(\boldsymbol{n} + \left(\sum_{\ell \in [L]} \mathcal{E}_\ell\right)\boldsymbol{a}(\boldsymbol{n})\right) \le \boldsymbol{C},\ \forall \boldsymbol{n} \in \mathcal{N}, \tag{5}$$

can be thought of as an extended version of (4). In (5), requests of all types are taken into account when the state is $\boldsymbol{n}$ and allocation decisions for some types are made in order to reserve resources for other types that turn out to be more profitable in the long run. In particular, resources are reserved for those request types ℓ which are allocated to nondummy patterns i at the expense of those types that are allocated to less profitable patterns or the corresponding dummy patterns. In this paper, all the results presented are based on the capacity Constraint (5).

From (5), there is an upper bound, $\min_{j \in [J]} \lceil C_j / w_{j,i} \rceil$, on the number of instantiations of pattern i, and this serves as a bounding state at which no further instantiation of this pattern can be added; that is, $\mathcal{N}_i = \{0, 1, \ldots, \min_{j \in [J]} \lceil C_j / w_{j,i} \rceil\}$ and $|\mathcal{N}_i| = \min_{j \in [J]} \lceil C_j / w_{j,i} \rceil + 1 < +\infty$. In this context, Equation (5) implies the condition

$$a_i(\boldsymbol{n}) = 0,\ \text{if } i \notin \{d(\ell) :\ \ell \in [L]\} \text{ and } n_i = |\mathcal{N}_i| - 1. \tag{6}$$

2.2.3. Objective. A *policy* ϕ is defined as a mapping $\mathcal{N} \to \mathscr{A}$ where $\mathscr{A} := \prod_{\ell \in [L]} \{0,1\}^{|\mathscr{P}_\ell|}$, determined by the action variables $\boldsymbol{a}(\boldsymbol{n})$ defined previously. When we are discussing a system operating with a given policy ϕ, we rewrite the action and state variables as $\boldsymbol{a}^\phi(\cdot)$ and $\boldsymbol{N}^\phi(t)$, respectively.

By serving a request of type $\ell \in [L]$ and occupying an RU of pool j for one unit of time, we gain expected reward r_ℓ and pay expected cost ε_j. The expected reward for a whole service is gained at the moment the service is completed. It corresponds to the situation where a request allocated to pattern i earns reward at rate $r_{\ell(i)}\mu_i$ for as long as it is in the system (so that the expected revenue per customer is $(r_{\ell(i)}\mu_i) \cdot (1/\mu_i) = r_{\ell(i)}$). The value of ϵ_j is the cost per unit time of using a unit of capacity from resource pool j in which case the expected cost of accommodating the request in pool j as part of pattern i is ϵ_j/μ_i. We seek a policy that maximizes the *revenue*: the difference between expected reward and cost, by efficiently using the limited amount of resources.

The objective is to maximize the long-run average rate of earning revenue, which exists because, for any policy ϕ, the process can be modeled by a finite-state Markov chain. Let $\boldsymbol{r} = (r_\ell :\ \ell \in [L])$ and $\boldsymbol{\varepsilon} = (\varepsilon_j :\ j \in [J])$. For all $\ell \in [L]$ and $i \in [I]$, define a $L \times I$ matrix $\mathcal{U}$ with entries $u_{\ell,i} := \mu_i \mathbb{1}_{i \in \mathscr{P}_\ell}$. By the Strong Law of Large Numbers for Continuous Time Markov Chains, see for example Serfozo (2009) (theorem 45 in Chapter 4), noting the subsequent discussion of the case where rewards are earned at jump times, the long-run average rate of earning revenue when the policy is ϕ is given by

$$R^\phi := \mathbb{E}_{\pi^\phi}[\boldsymbol{r}\mathcal{U} - \boldsymbol{\varepsilon}\mathcal{W}] = \sum_{i \in [I]} \sum_{n_i \in \mathcal{N}_i} \pi_i^\phi(n_i) \times \left(r_{\ell(i)}\mu_i - \sum_{j \in \mathcal{J}} w_{j,i}\varepsilon_j\right) n_i, \tag{7}$$

where $\pi_i^\phi(n_i)$ is the stationary probability that the state of process i is n_i when the policy is ϕ. Then we wish to find the policy ϕ that maximizes R^ϕ, that is we wish to find

$$R := \max_\phi R^\phi. \tag{8}$$

Define Φ to be the set of all policies with the constraints in (1) and (5) satisfied. Each policy in Φ is then a *feasible policy* for our resource allocation problem.

3. Whittle Relaxation

Our resource allocation problem with objective function defined by (8) and constraints given by (1) and (5) can be modeled as a set of RMABPs coupled by capacity constraints. We leave the specification of the RMABPs to Appendix EC.1 in the e-companion to this paper.

In this section, we provide a theoretical analysis of the resource allocation problem, following the idea of Whittle relaxation Whittle (1988). In the vein of a RMABP, we randomize the action variable $\boldsymbol{a}^{\phi}(\boldsymbol{n})$ so that its elements take values from $\{0,1\}$ with probabilities determined by the policy ϕ and relax Constraint (1) to require that

$$\lim_{t\to+\infty}\mathbb{E}\left[\sum_{i\in\mathcal{P}_\ell}a_i^{\phi}(\boldsymbol{N}^{\phi}(t))\right]=1,\ \forall\ell\in[L]. \tag{9}$$

Following similar ideas, we relax (5) into two equations:

$$\lim_{t\to+\infty}\mathbb{E}\big[\mathcal{W}\big(\boldsymbol{N}^{\phi}(t)+\boldsymbol{a}^{\phi}\big(\boldsymbol{N}^{\phi}(t)\big)\big)\big]\le\boldsymbol{C}, \tag{10}$$

and

$$\lim_{t\to+\infty}\mathbb{E}\Big[a_i^{\phi}\big(\boldsymbol{N}^{\phi}(t)\big)\ \mathbb{1}_{N_i^{\phi}(t)=|\mathcal{N}_i|-1}\Big]=0,\quad \forall i\in[I]\backslash\{d(\ell):\ \ell\in[L]\}. \tag{11}$$

Remark 1. Equation (10) is derived by taking expectations for both sides of Equation (5), and (11) is a consequence of (6), so constraints described by (10) and (11) are relaxed versions of the constraints described by (5). The justification for Equation (11) will be discussed in Appendix EC.9.1 in the e-companion, in conjunction with the physical meanings of all variables, when we increase the scale of the entire system. We refer to the problem with Objective (8), Constraints (9)–(11) and randomized control variables $\boldsymbol{a}^{\phi}(\boldsymbol{n})$, for all $\boldsymbol{n}\in\mathcal{N}$, as the *relaxed problem*.

A value a in $(0,1)$ can be interpreted as a randomisation between taking $a_i^{\phi}(\boldsymbol{n})=0$ and $a_i^{\phi}(\boldsymbol{n})=1$. Specifically we take $a_i^{\phi}(\boldsymbol{n})=1$ with probability a. We represent the set of policies that correspond to assigning such values $a\in(0,1)$ as $\tilde{\Phi}$. For $n_i\in\mathcal{N}_i$, $\phi\in\tilde{\Phi}$, $i\in[I]$, define

- $\alpha_i^{\phi}(n_i):=\lim_{t\to+\infty}\mathbb{E}[a_i^{\phi}(\boldsymbol{N}^{\phi}(t))\mid N_i^{\phi}(t)=n_i]$, which is the expectation with respect to the stationary distribution when policy ϕ is used, and the vector $\boldsymbol{\alpha}_i^{\phi}:=(\alpha_i^{\phi}(n_i):\ n_i\in\mathcal{N}_i)$;
- the stationary probability that $N_i^{\phi}(t)=n_i$ under policy ϕ to be π_{i,n_i}^{ϕ}, and the vector $\boldsymbol{\pi}_i^{\phi}:=(\pi_{i,n_i}^{\phi}:\ n_i\in\mathcal{N}_i)$.

Let $\Pi_n^{\phi}:=(\boldsymbol{\pi}_i^{\phi}\cdot(\mathcal{N}_i):i\in[I])^T$ and $\Pi_a^{\phi}:=(\boldsymbol{\pi}_i^{\phi}\cdot\boldsymbol{\alpha}_i^{\phi}:i\in[I])^T$, where $(\mathcal{N}_i)$ represents the vector $(0,1,\ldots,|\mathcal{N}_i|-1)$. The Lagrangian function for the optimization problem with Objective Function (8) and Constraints (9)–(11) is

$$\begin{aligned}g(\boldsymbol{\gamma},\boldsymbol{\nu},\boldsymbol{\eta}):=&\max_{\phi\in\tilde{\Phi}}(\boldsymbol{r}\mathcal{U}-\boldsymbol{\varepsilon}\mathcal{W})\Pi_n^{\phi}\\&-\sum_{\ell=1}^{L}\nu_\ell\left(\sum_{i\in\mathcal{P}_\ell}\boldsymbol{\pi}_i^{\phi}\cdot\boldsymbol{\alpha}_i^{\phi}-1\right)\\&-\boldsymbol{\gamma}\cdot\left(\mathcal{W}\left(\Pi_n^{\phi}+\Pi_a^{\phi}\right)-\boldsymbol{C}\right)\\&-\sum_{i\in[I]\backslash\{d(\ell):\ \ell\in[L]\}}\eta_i\pi_{i,|\mathcal{N}_i|-1}^{\phi}\alpha_i^{\phi}(|\mathcal{N}_i|-1),\end{aligned} \tag{12}$$

where $\boldsymbol{\nu}\in\mathbb{R}^L$, $\boldsymbol{\gamma}\in\mathbb{R}_0^J$, and $\boldsymbol{\eta}\in\mathbb{R}^{I-L}$ are Lagrange multiplier vectors for Constraints (9)–(11), respectively. In (12), the constraints no longer apply to variables $\boldsymbol{\alpha}_i^{\phi}$ ($i\in[I]$) but appear in the maximization as cost items weighted by their Lagrange multipliers. For $i\in[I]\backslash\{d(\ell):\ \ell\in[L]\}$, define functions

$$\begin{aligned}\Lambda_i(\phi,\boldsymbol{\gamma},\nu_{\ell(i)},\eta_i):=&\big(r_{\ell(i)}\mu_i-\boldsymbol{\varepsilon}\cdot\boldsymbol{w}_i\big)\boldsymbol{\pi}_i^{\phi}\cdot(\mathcal{N}_i)\\&-\nu_{\ell(i)}\boldsymbol{\pi}_i^{\phi}\cdot\boldsymbol{\alpha}_i^{\phi}-\boldsymbol{\gamma}\cdot\left(\boldsymbol{w}_i\left(\boldsymbol{\pi}_i^{\phi}\cdot(\mathcal{N}_i)+\boldsymbol{\pi}_i^{\phi}\cdot\boldsymbol{\alpha}_i^{\phi}\right)\right)\\&-\eta_i\pi_{i,|\mathcal{N}_i|-1}^{\phi}\alpha_i^{\phi}(|\mathcal{N}_i|-1),\end{aligned} \tag{13}$$

where we recall that $\boldsymbol{w}_i$ is the weight vector of pattern i given by the ith column vector of $\mathcal{W}$; similarly, for $\ell\in[L]$, $\boldsymbol{\gamma}\in\mathbb{R}_0^J$ and $\eta\in\mathbb{R}$, define $\Lambda_{d(\ell)}(\phi,\boldsymbol{\gamma},\nu_\ell,\eta):=-\nu_\ell\alpha_{d(\ell)}^{\phi}(n)$, where n is the only state in $\mathcal{N}_{d(\ell)}$. From Equation (12), for $\boldsymbol{\gamma}\in\mathbb{R}_0^J$, $\boldsymbol{\nu}\in\mathbb{R}^L$, and $\boldsymbol{\eta}\in\mathbb{R}^{I-L}$,

$$g(\boldsymbol{\gamma},\boldsymbol{\nu},\boldsymbol{\eta})=\max_{\phi\in\tilde{\Phi}}\sum_{i\in[I]}\Lambda_i\big(\phi,\boldsymbol{\gamma},\nu_{\ell(i)},\eta_i\big)+\sum_{\ell\in[L]}\nu_\ell+\boldsymbol{\gamma}\cdot\boldsymbol{C}, \tag{14}$$

where $\eta_{d(\ell)}$ ($\ell\in[L]$) are unconstrained real numbers that are used for notational convenience.

In the maximization problem on the right-hand side of (14), there is no constraint that restricts the value of one $\Lambda_i(\phi,\boldsymbol{\gamma},\nu_{\ell(i)},\eta_i)$ once the others are known. As a result, we can maximize the sum in (14) by maximizing each of the summands independently. We can thus write (14) as

$$g(\boldsymbol{\gamma},\boldsymbol{\nu},\boldsymbol{\eta})=\sum_{i\in[I]}\max_{\phi\in\tilde{\Phi}}\Lambda_i\big(\phi,\boldsymbol{\gamma},\nu_{\ell(i)},\eta_i\big)+\sum_{\ell\in[L]}\nu_\ell+\boldsymbol{\gamma}\cdot\boldsymbol{C}, \tag{15}$$

but with the maximum over $\phi\in\tilde{\Phi}$. Observe now that maximizing Λ_i over ϕ is equivalent to choosing $\boldsymbol{\alpha}_i^{\phi}(n_i)$ from $[0,1]^{|\mathcal{N}_i|}$, by interpreting $\alpha_{i,n}^{\phi}\in[0,1]$ as the probability that process i is activated under policy ϕ when it is in state n. Thus,

$$\begin{aligned}g(\boldsymbol{\gamma},\boldsymbol{\nu},\boldsymbol{\eta})=&\sum_{i\in[I]}\max_{\boldsymbol{\alpha}_i^{\phi}\in[0,1]^{|\mathcal{N}_i|}}\Lambda_i\big(\phi,\boldsymbol{\gamma},\nu_\ell,\eta_i\big)\\&+\sum_{\ell\in[L]}\nu_\ell+\boldsymbol{\gamma}\cdot\boldsymbol{C}.\end{aligned} \tag{16}$$

By slightly abusing notation, we refer to the policy ϕ determined by an action vector $\boldsymbol{\alpha}_i^{\phi}$ as the policy for pattern i, and define Φ_i as the set of all policies for pattern i.

Definition 1. The maximization of $\Lambda_i(\phi,\boldsymbol{\gamma},\nu_\ell,\eta_i)$ over $\boldsymbol{\alpha}_i^{\phi}\in[0,1]^{|\mathcal{N}_i|}$ is the *subproblem* for pattern $i\in[I]$.

For given $\boldsymbol{\gamma}$, $\boldsymbol{\nu}$ and $\boldsymbol{\eta}$, the subproblem for any pattern is an MDP, so that it can be numerically solved by dynamic programming. By solving the subproblems for all patterns $i\in[I]$, we obtain $g(\boldsymbol{\gamma},\boldsymbol{\nu},\boldsymbol{\eta})$. For any $\boldsymbol{\gamma}$, $\boldsymbol{\nu}$

and $\boldsymbol{\eta}$, the Lagrangian function $g(\boldsymbol{\gamma},\boldsymbol{\nu},\boldsymbol{\eta})$ is a performance upper bound for the primal problem described in (8)–(11), which is a relaxed version of the original resource allocation problem. Thus, there will be a nonnegative gap between this upper bound and the maximized performance of the original problem.

3.1. Analytical Solutions

Proposition 1. *For given $\boldsymbol{\nu}$ and $\boldsymbol{\gamma}$, there exists $\boldsymbol{E}\in\mathbb{R}^{I-L}$ such that, for any $\boldsymbol{\eta}>\boldsymbol{E}$, a policy of the subproblem for pattern i, referred to as $\bar{\varphi}\in\Phi_i$, determined by action vector $\boldsymbol{\alpha}_i^{\bar{\varphi}}\in[0,1]^{|\mathcal{N}_i|}$ is optimal for this subproblem, if, for $n\in\mathcal{N}_i$,*

$$\alpha_i^{\bar{\varphi}}(n)\begin{cases}=1 \quad \text{if} \quad 0<\lambda_\ell\left(r_\ell-\frac{1}{\mu_i}\sum\limits_{j\in\mathcal{J}_i}\varepsilon_j w_{j,i}\right) \\ \qquad -\left(1+\frac{\lambda_\ell}{\mu_i}\right)\sum\limits_{j\in\mathcal{J}_i}w_{j,i}\gamma_j-\nu_\ell \text{ and } n<|\mathcal{N}_i|-1, & (17)\\ \in[0,1] \quad \text{if} \quad 0=\lambda_\ell\left(r_\ell-\frac{1}{\mu_i}\sum\limits_{j\in\mathcal{J}_i}\varepsilon_j w_{j,i}\right) \\ \qquad -\left(1+\frac{\lambda_\ell}{\mu_i}\right)\sum\limits_{j\in\mathcal{J}_i}w_{j,i}\gamma_j-\nu_\ell \text{ and } n<|\mathcal{N}_i|-1, & (18)\\ =0 \quad \text{otherwise}, & (19)\end{cases}$$

where $\ell=\ell(i)$.

The proof will be given in Appendix EC.1 in the e-companion to this paper. In the maximization of $\Lambda_i(\phi,\boldsymbol{\gamma},\nu_{\ell(i)},\eta_i))$, the only term of Λ_i dependent on $\boldsymbol{\eta}$ is $-\eta_i\pi^{\phi}_{i,|\mathcal{N}_i|-1}\alpha_i^{\phi}(|\mathcal{N}_i|-1)$. The choice of a sufficiently large η_i guarantees that $\alpha_i^{\phi}(|\mathcal{N}_i|-1)$ is 0 for an optimal policy of the subproblem, so that Constraints (11) of the relaxed problem are satisfied. For convenience, in what follows we fix $\boldsymbol{\eta}$ to be one of these large values so that $\alpha_i^{\phi}(|\mathcal{N}_i|-1)=0$ for any optimal policy ϕ of the subproblem for pattern i. By slightly abusing notation, in all subsequent equations and discussions, we directly require $\alpha_i^{\phi}(|\mathcal{N}_i|-1)=0$ $(i\in[I]\backslash\{d(\ell):\ell\in[L]\})$ unless specified otherwise.

Remark 2. Recall that the action variables $\boldsymbol{\alpha}_i^{\phi}$ for any pattern $i\in[I]$ and policy $\phi\in\Phi_i$ are potentially state-dependent. However, the right-hand sides of Equations (17)–(19) are independent of the state variable n that appears on their left-hand side, provided that this is less than $|\mathcal{N}_i|-1$. This state-independence phenomenon is a consequence of the linearity of the reward and cost rates in the state variable, $N_i^{\phi}(t)$, for pattern $i\in[I]\backslash\{d(\ell):\ \ell\in[L]\}$: from our definition in Section 2, the reward and cost rates of process i in state $N_i^{\phi}(t)$ are $r_{\ell(i)}\mu_i N_i^{\phi}(t)$ and $\sum_{j\in\mathcal{J}_i}\varepsilon_j w_{j,i}N_i^{\phi}(t)$, respectively. A detailed analysis is provided in the proof of Proposition 1.

Using an argument similar to that in Whittle (1988), we can derive from (17)–(19) an abstracted *priority* for each *pattern-state pair* (PS pair) (i,n) with $n\in\mathcal{N}_i\backslash\{|\mathcal{N}_i|-1\}$ and $i\in[I]$; unlike in Whittle (1988), here, this priority is $(\boldsymbol{\gamma},\boldsymbol{\nu})$ dependent. The priority of a PS pair (i,n) with $n\in\mathcal{N}_i\backslash\{|\mathcal{N}_i|-1\}$ is determined by the *index*

$$\Xi_i(\boldsymbol{\gamma},\boldsymbol{\nu}):=\lambda_{\ell(i)}\left(r_{\ell(i)}-\frac{1}{\mu_i}\sum_{j\in\mathcal{J}_i}\varepsilon_j w_{j,i}\right)-\left(1+\frac{\lambda_{\ell(i)}}{\mu_i}\right)\sum_{j\in\mathcal{J}_i}w_{j,i}\gamma_j-\nu_{\ell(i)}, \tag{20}$$

and (17)–(19) can be characterized comparing $\Xi_i(\boldsymbol{\gamma},\boldsymbol{\nu})$ with 0. When there is strict inequality in the comparison (that is, the cases described in (17) and (19)), the value of $\alpha_i^{\phi}(n)$ is specified, because PS pairs (i,n) for all $n\in\mathcal{N}_i\backslash\{|\mathcal{N}_i|-1\}$ correspond to the same $\Xi_i(\boldsymbol{\gamma},\boldsymbol{\nu})$ value. However, there is still freedom to decide different values of $\alpha_i^{\phi}(n)$, when $\Xi_i(\boldsymbol{\gamma},\boldsymbol{\nu})=0$ (the case described in (18)). A detailed discussion about priorities of PS pairs corresponding to the same $\Xi_i(\boldsymbol{\gamma},\boldsymbol{\nu})$ will be provided in Section 3.2. By solving the subproblem of dummy pattern $d(\ell)$ $(\ell\in[L])$, which involves only one state $n\in\mathcal{N}_{d(\ell)}$, we obtain an optimal policy φ determined by

$$\alpha^{\varphi}_{d(\ell)}(n)\begin{cases}=1, & \text{if } 0<-\nu_\ell,\\ \in[0,1], & \text{if } 0=-\nu_\ell,\\ =0, & \text{otherwise.}\end{cases} \tag{21}$$

The priority of the state of a dummy pattern is then $\Xi_{d(\ell)}(\boldsymbol{\gamma},\boldsymbol{\nu})\equiv-\boldsymbol{\nu}$ for any $\boldsymbol{\gamma}$.

In addition, from Equation (19) in Proposition 1, for any given $\boldsymbol{\nu}\in\mathbb{R}^I$ and $\boldsymbol{\gamma}\in\mathbb{R}_0^J$, there exists $\boldsymbol{\eta}\in\mathbb{R}^{I-L}$ such that it is optimal to make states $|\mathcal{N}_i|-1$ passive (that is, $\alpha_i^{\bar{\varphi}}(|\mathcal{N}_i|-1)=0$) for all $i\in[I]\backslash\{d(\ell):\ \ell\in[L]\}$. Among all PS pairs (n,i) $(n\in\mathcal{N}_i,\ i\in[I])$, we assign, without loss of generality, the least priority to those PS pairs $(i,|\mathcal{N}_i|-1)$ for which $i\in[I]\backslash\{d(\ell):\ \ell\in[L]\}$.

The policy $\bar{\varphi}$ determined by (17)–(19) and (21) is optimal for the relaxed problem described by (8)–(11), if the given multipliers $\boldsymbol{\nu}$ and $\boldsymbol{\gamma}$ that appear in (17)–(19) and (21) satisfy the *complementary slackness conditions* of this relaxed problem, defined by complementary slackness:

$$\nu_\ell\left(\sum_{i\in\mathcal{P}_l}\boldsymbol{\pi}_i^{\phi}\cdot\boldsymbol{\alpha}_i^{\phi}-1\right)=0,\ \forall l\in[L], \tag{22}$$

and

$$\gamma_j\left(\boldsymbol{\omega}_j\cdot\left(\Pi_n^{\phi}+\Pi_a^{\phi}\right)-C_j\right)=0,\ \forall j\in[J], \tag{23}$$

where $\boldsymbol{\omega}_j=(w_{j,i}:\ i\in[I])$ is the jth row of matrix $\mathcal{W}$.

In this context, if resource pool $j\in[J]$ is very popular so that the capacity constraint corresponding to the jth row in (10) achieves equality, then γ_j is allowed to be positive, leading to a lower value of $\Xi_i(\boldsymbol{\gamma},\boldsymbol{\nu})$ than for $\gamma_j=0$. On the other hand, if resource pool $j\in[J]$ cannot be fully occupied and the jth capacity constraint in (10) is satisfied with a strict inequality, then the complementary slackness condition described in (23) forces γ_j to be zero. Following this mechanism, when resource pool $j \in [J]$ is overloaded and its priority is reduced by increasing γ_j, the offered traffic to this resource pool will be reduced in line with its priority.

If there exist multipliers $\boldsymbol{\nu}$, $\boldsymbol{\gamma}$ and a policy $\bar{\varphi}$ determined by (17)–(19), such that the complementary slackness Conditions (22) and (23) are satisfied by taking $\phi = \bar{\varphi}$, then, by the strong duality theorem, this policy $\bar{\varphi}$ is optimal for the relaxed problem; this observation, together with Theorem EC.1 in Appendix EC.9.3 in the e-companion, leads to the existence of an asymptotically optimal policy feasible for the original problem, derived with priorities of patterns induced by the descending order of $\Xi_i(\boldsymbol{\gamma}, \boldsymbol{\nu})$. More details about the analysis in the asymptotic regime will be provided in Appendix EC.9 in the e-companion. Here we focus on the nonasymptotic regime and specifically on the choice and computation of $\boldsymbol{\gamma}$ and $\boldsymbol{\nu}$.

3.2. Decomposable Capacity Constraints

In the general case, it is not clear whether the complementary slackness Conditions (22) and (23) can be satisfied and, even if they are, what the values of $\boldsymbol{\gamma}$ and $\boldsymbol{\nu}$ are. More important is the question of how the multipliers help with proposing an asymptotically optimal policy applicable to the original problem.

3.2.1. Priorities of PS Pairs. As described in Section 3.1, the priorities of PS pairs are determined by the descending order of $\Xi_i(\boldsymbol{\gamma}, \boldsymbol{\nu})$, with higher priorities given by higher values of $\Xi_i(\boldsymbol{\gamma}, \boldsymbol{\nu})$. It may happen that, because of different tie-breaking rules, the same $\boldsymbol{\gamma}$ and $\boldsymbol{\nu}$ lead to different priorities. For given $\boldsymbol{\gamma} \in \mathbb{R}_0^J$ and $\boldsymbol{\nu} \in \mathbb{R}^L$, let $\mathcal{O}(\boldsymbol{\gamma}, \boldsymbol{\nu})$ represent the set of all rankings of PS pairs compatible with the descending order of $\Xi_i(\boldsymbol{\gamma}, \boldsymbol{\nu})$ ($i \in [I]$). Also, for notational convenience, let $\mathcal{O}$ represent the set of all PS pair rankings.

To emphasize the priorities of these PS pairs, according to a given ranking $o \in \mathcal{O}$, we label all these pairs by their order $\iota^o \in [N]$ with $N := \sum_{i \in [I]} |\mathcal{N}_i|$ and $(i_{\iota^o}, n_{\iota^o})$ giving the pattern and the state of the ι^oth PS pair. We will omit the superscript o and use ι unless it is necessary to specify the underlying ranking. There exists one and only one $\ell \in [L]$ satisfying $i_\iota \in \mathcal{P}_\ell$ for any PS pair labeled by ι. Such an ℓ is denoted by ℓ_ι.

Algorithm 1. Priority-Style Policy for the Relaxed Problem

Input: a vector of nonnegative reals $\boldsymbol{\gamma} \in \mathbb{R}_0^J$ and a ranking of PS pairs $o \in \mathcal{O}$.

Output: a policy $\bar{\varphi}(o) \in \tilde{\Phi}$ determined by action variables $\boldsymbol{\alpha}_i^{\bar{\varphi}(o)} \in [0,1]^{|\mathcal{N}_i|}$ for all $i \in [I]$ and a vector of reals $\boldsymbol{\nu}(o, \boldsymbol{\gamma})$.

Function PriorityPolicy$o, \boldsymbol{\gamma}$:
  $\boldsymbol{\alpha}_i^{\bar{\varphi}} \leftarrow \mathbf{0}$ for all $i \in [I]$ /*Variables $\boldsymbol{\alpha}_i^{\bar{\varphi}}$ determine a policy $\bar{\varphi}$*/
  Initializing the list of candidate PS pairs as the list of all PS pairs;
  $\iota \leftarrow 0$ /*Iteration variable*/
  while $\iota < N$ **and** *the list of candidate PS pairs is not empty* **do**
    $\iota \leftarrow \iota + 1$;
    If *PS pair ι is not in the list of candidate PS pairs* **then**
      continue
    end
    $a_1 \leftarrow \inf\{\{\alpha_{i_\iota}^{\bar{\varphi}}(n_\iota) \in [0,1] \mid \sum_{i \in \mathcal{P}_{\ell_\iota}} \boldsymbol{\pi}_i^{\bar{\varphi}} \cdot \boldsymbol{\alpha}_i^{\bar{\varphi}} = 1.\} \cup \{1\}\}$; / *The maximal probability of activating PS pair ι such that * / / *the action constraint for RT ℓ_ι is not violated.* /
    $a_2 \leftarrow \inf\{\{\alpha_{i_\iota}^{\bar{\varphi}}(n_\iota) \in [0,1] \mid \exists j \in [J], \boldsymbol{\omega}_j \cdot (\Pi_n^{\bar{\varphi}} + \Pi_a^{\bar{\varphi}}) = C_j.\} \cup \{1\}\}$; / *The maximal probability of activating PS pair ι such that* / / * the capacity constraints are not violated.* /
    $\alpha_{i_\iota}^{\bar{\varphi}}(n_\iota) \leftarrow \min\{a_1, a_2\}$; / *Update $\alpha_{i_\iota}^{\bar{\varphi}}(n_\iota)$ with the maximal activating probability* / / *without violating any constraint. */
    if $\sum_{i \in \mathcal{P}_{\ell_\iota}} \boldsymbol{\pi}_i^{\bar{\varphi}} \cdot \boldsymbol{\alpha}_i^{\bar{\varphi}} = 1$ **then**
      / *If the action constraint achieves equality under policy $\bar{\varphi}$ */ /* determined by updated $\boldsymbol{\alpha}_i^{\bar{\varphi}}$, $i \in [I]$. */
      $\nu_{\ell_\iota}(o, \boldsymbol{\gamma}) \leftarrow \Xi_{i_\iota}(\boldsymbol{\gamma}, \mathbf{0})$
      remove all PS pairs $\iota' > \iota$ with $\ell_{\iota'} = \ell_\iota$ from the list of candidate PS pairs;
    else if $\exists j \in [J]$, $\boldsymbol{\omega}_j \cdot (\Pi_n^{\bar{\varphi}} + \Pi_a^{\bar{\varphi}}) = C_j$ **then**
      / *If a capacity constraint achieves equality under policy $\bar{\varphi}$ */ / *determined by updated $\boldsymbol{\alpha}_i^{\bar{\varphi}}$, $i \in [I]$. */
      remove all PS pairs $\iota' > \iota$ with $w_{j,i_{\iota'}} > 0$ from the list of candidate PS pairs;
    end
  end
  $\boldsymbol{\alpha}_i^{\bar{\varphi}(o)} \leftarrow \boldsymbol{\alpha}_i^{\bar{\varphi}}$ for all $i \in [I]$;
return

For any given ranking of PS pairs $o \in \mathcal{O}$, Algorithm 1 generates a policy $\bar{\varphi}(o)$ with priorities of PS pairs defined by o, such that (9)–(11) are satisfied: the policy $\bar{\varphi}(o)$ is feasible for the relaxed problem but not necessarily feasible for the original problem. The key idea for generating such a $\bar{\varphi}(o)$ is to initialize $\boldsymbol{\alpha}_i^{\bar{\varphi}(o)}$ to $\mathbf{0}$ for all $i \in [I]$, and sequentially activate the PS pairs according to their priorities defined by o until either a relaxed action or capacity constraint described in (9) and (10), respectively, achieves equality. In particular,

I. if a relaxed action constraint described in (9) achieves equality by activating PS pairs less than or equal to ι, then the multiplier ν_{ℓ_ι} is set to $\Xi_{i_\iota}(\boldsymbol{\gamma}, \mathbf{0})$, and all later PS pairs $\iota' > \iota$ with $\ell_{\iota'} = \ell_\iota$ are *disabled* from being activated and are removed from the *list of candidate pairs* awaiting later activation;

II. Similarly, if a relaxed capacity constraint described in (10) associated with resource pool $j \in [J]$ achieves equality by activating PS pairs less than or equal to ι, then all later PS pairs $\iota' > \iota$ with $w_{j,i_{\iota'}} > 0$ are disabled and removed from the list of candidate states.

Maintaining an iteratively updated list of candidate pairs in this way continues until all action constraints in (9) achieve equality: the policy $\bar{\varphi}(o)$ is determined by the resulting $\boldsymbol{\alpha}_i^{\bar{\varphi}(o)}$ ($i \in [I]$), and the multipliers $\boldsymbol{\nu}$ are updated as in I. The vector of these multipliers is denoted by $\boldsymbol{\nu}(o, \boldsymbol{\gamma})$ and is listed as an output of Algorithm 1. The PS pair labeled by ι satisfying the condition described in I) is called the *critical pair*, with the corresponding resource pool j referred to as the *critical pool* of PS pair ι, denoted by $j_\iota(o)$. From the description in I), there might be more than one resource pool for which the capacity constraints achieve equality simultaneously while activating PS pair ι; we choose one of them to be $j_\iota(o)$ and refer to this resource pool as the critical pool of ι. Let $\mathcal{I}(o)$ represent the set of all critical pairs with respect to the policy $\bar{\varphi}(o)$.

Lemma 1. *For any $o \in \mathcal{O}$ and $\iota, \iota' \in \mathcal{I}(o)$, if $\iota \neq \iota'$ then $i_\iota \neq i_{\iota'}$.*

Proof. Consider critical pairs $\iota, \iota' \in \mathcal{I}(o)$ with $\iota \neq \iota'$, and assume $\iota < \iota'$. Since ι is a critical pair, there is a critical resource pool j_ι, which is fully occupied. In this case, if $i_\iota = i_{\iota'}$, then pair ι' must require some resource units from pool j_ι and so $\alpha_{\iota'}^{\bar{\varphi}(o)} = 0$. PS pair ι' cannot be critical, which violates the condition $\iota' \in \mathcal{I}(o)$. Hence, $i_\iota \neq i_{\iota'}$. This proves the lemma.

Recall, for any ranking o, the policy $\bar{\varphi}(o)$ must satisfy the action and capacity Constraints (9)–(11). Also, because (9) holds, the complementary slackness conditions corresponding to the action Constraints (22) are satisfied by taking $\phi = \bar{\varphi}(o)$. However, the complementary slackness conditions corresponding to the capacity Constraints (23) and Equations (17)–(19) are not necessarily satisfied if we plug in $\phi = \bar{\varphi}(o)$ and $\boldsymbol{\gamma}$: the policy $\bar{\varphi}(o)$ is a heuristic policy applicable for the relaxed problem defined by (8)–(11) derived by intuitively prioritizing PS pairs according to their ranking $o \in \mathcal{O}$.

In Section 3.3 we shall define a particular class of resource allocation models, for which we can show the complementary slackness conditions are indeed satisfied.

Definition 2. The system said to be *decomposable* if there exist multipliers $\boldsymbol{\gamma} \in \mathbb{R}_0^J$, $\boldsymbol{\nu} \in \mathbb{R}^L$ and a ranking $o \in \mathcal{O}(\boldsymbol{\gamma}, \boldsymbol{\nu})$ such that $\boldsymbol{\nu} = \boldsymbol{\nu}(o, \boldsymbol{\gamma})$ and the complementary slackness Conditions (22) and (23) are satisfied by taking $\phi = \bar{\varphi}(o)$. In this case the optimal values of the dual variables are called *decomposable values*.

Recall that, in the general case, for $\boldsymbol{\gamma} \in \mathbb{R}_0^J$ and $\boldsymbol{\nu} \in \mathbb{R}^L$, even if $o \in \mathcal{O}(\boldsymbol{\gamma}, \boldsymbol{\nu})$, the policy $\bar{\varphi}(o)$ is not necessarily optimal (because it does not necessarily satisfy (17)–(19)). When the policy $\bar{\varphi}(o)$ is optimal for the relaxed problem, the ranking o can be used to construct an index policy applicable to the original problem (detailed steps are provided in Section 4). Theorem EC.1 (in Appendix EC.9.3 in the e-companion) then ensures that such an index policy is asymptotically optimal.

3.2.2. Derivation of the Pair Ranking.

We start with a proposition that shows how the values of the Lagrange multipliers $\boldsymbol{\nu}$ and $\boldsymbol{\gamma}$ can be derived from a knowledge of the critical pair and critical resource pool corresponding to a given order $o \in \mathcal{O}$.

Proposition 2. *For any given $\boldsymbol{\gamma}_0 \in \mathbb{R}_0^J$ and $o \in \mathcal{O}$, the linear equations*

$$\nu_{\ell_\iota}(o, \boldsymbol{\gamma}_0) = \Xi_{i_\iota}(\boldsymbol{\gamma}, \mathbf{0}), \ \forall \iota \in \mathcal{I}(o) \tag{24}$$

and

$$\gamma_j = 0, \ \forall j \notin \{ j_\iota(o) \in [J] \mid \iota \in \mathcal{I}(o) \} \tag{25}$$

have a unique solution $\boldsymbol{\gamma} \in \mathbb{R}^J$.

The proof of Proposition 2 will be given in Appendix EC.3 in the e-companion. For a ranking $o \in \mathcal{O}$, define a function $\mathcal{T}^o$ of $\boldsymbol{\gamma}_0 \in \mathbb{R}_0^J$ with respect to $o \in \mathcal{O}$: $\mathcal{T}^o(\boldsymbol{\gamma}_0) := \boldsymbol{\gamma}$, where $\boldsymbol{\gamma}$ is the unique solution of (24) and (25). Let $\mathcal{T}_j^o(\boldsymbol{\gamma}_0)$ represent the jth element of $\mathcal{T}^o(\boldsymbol{\gamma}_0)$.

Proposition 3. *If there exist $\boldsymbol{\gamma}_0 \in \mathbb{R}_0^J$ and $o \in \mathcal{O}(\boldsymbol{\gamma}_0, \mathbf{0})$ such that $\mathcal{T}^o(\boldsymbol{\gamma}_0) = \boldsymbol{\gamma}_0$, then $\boldsymbol{\gamma}_0$ is a vector of decomposable multipliers and the policy $\bar{\varphi}(o)$ based on ranking o is optimal for the relaxed problem defined by* (8)–(11).

The proof of Proposition 3 will be given in Appendix EC.4 in the e-companion. Recall that $\mathcal{I}(o)$ is the set of critical pairs with respect to the policy $\bar{\varphi}(o)$, $j_\iota(o)$ is the critical resource pool corresponding to critical pair $\iota \in \mathcal{I}(o)$ according to ranking o, and $\nu_{\ell_\iota}(o, \boldsymbol{\gamma}_0)$ is an output of Algorithm 1 when the inputs are o and $\boldsymbol{\gamma} = \boldsymbol{\gamma}_0$.

Remark 3. Proposition 3 provides a way of checking decomposability of $\boldsymbol{\gamma}_0$ and optimality of $\bar{\varphi}(o)$. By Proposition 3, any fixed point $\boldsymbol{\gamma}_0 \in \mathbb{R}_0^J$ of the function $\mathcal{T}^o$ with respect to a ranking $o \in \mathcal{O}(\boldsymbol{\gamma}_0, \mathbf{0})$ is a decomposable vector. The decomposability of $\boldsymbol{\gamma}_0$ can be checked without requiring knowledge of any $\boldsymbol{\nu} \in \mathbb{R}^L$. Also, we present the following corollary of Proposition 3.

Corollary 1. *For $\boldsymbol{\gamma}_0 \in \mathbb{R}_0^J$ and $o \in \mathcal{O}(\boldsymbol{\gamma}_0, \mathbf{0})$, if $\mathcal{T}^o(\boldsymbol{\gamma}_0) \neq \boldsymbol{\gamma}_0$, $\mathcal{T}^o(\boldsymbol{\gamma}_0) \in \mathbb{R}_0^J$ and $o \in \mathcal{O}(\mathcal{T}^o(\boldsymbol{\gamma}_0), \mathbf{0})$, then $\mathcal{T}^o(\mathcal{T}^o(\boldsymbol{\gamma}_0)) = \mathcal{T}^o(\boldsymbol{\gamma}_0)$.*

The hypothesis of Corollary 1 requires all components of $\mathcal{T}^o(\boldsymbol{\gamma}_0)$ to be nonnegative. This is not such an easy condition to satisfy. The proof of Corollary 1 will be given in Appendix EC.5 in the e-companion.

In this context, consider a given $\gamma_0 \in \mathbb{R}_0^J$ and a ranking $o \in \mathcal{O}(\gamma_0, \mathbf{0})$. If γ_0 is a fixed point of $\mathcal{T}^o$, then it is the vector of decomposable multipliers; if it is not but $\mathcal{T}^o(\gamma_0)$ is a nonnegative fixed point of $\mathcal{T}^o$, then $\mathcal{T}^o(\gamma_0)$ represents the decomposable multipliers. However, in both cases, we need to propose a specific γ_0; this requires prior knowledge of the multipliers, which is, in general, not available. Section 3.3 will discuss a special case where the decomposability is provable, and we have a method of deriving the decomposable multipliers. Here, to make a reasonably good choice of the Lagrangian multipliers in a general system, we embark on a *fixed point iteration method*.

Because Proposition 3 requires a fixed point γ of the function $\mathcal{T}^o$ with $o \in \mathcal{O}(\gamma, \mathbf{0})$, we need to iterate not only the value of γ but also the corresponding ranking o, which affects the function $\mathcal{T}^o$ and should be an element of $\mathcal{O}(\gamma, \mathbf{0})$. Following the idea of conventional fixed point interation methods, for $k \in \mathbb{N}_0$, let $\gamma_{k+1} = (\mathcal{T}^{o_k}(\gamma_k))^+$ with initial γ_0 and $o_0 \in \mathcal{O}(\gamma_0, \mathbf{0})$, where $(v)^+ := (\max\{0, v_i\} : i \in [N])$ for a vector $v \in \mathbb{R}^N$ $(N \in \mathbb{N}_+)$. Construct a ranking $o_{k+1} \in \mathcal{O}(\gamma_{k+1}, \mathbf{0})$ according to o_k: for any two different PS pairs (i, n) and (i', n') with $\Xi_i(\gamma_{k+1}, \mathbf{0}) = \Xi_{i'}(\gamma_{k+1}, \mathbf{0})$, (i, n) precedes (i', n') in the ranking o_{k+1} if and only if (i, n) precedes (i', n') in the ranking o_k. Here, the operation $(\cdot)^+$ is used to make all the elements of γ_{k+1} nonnegative, so that γ_{k+1} is feasible for the function $\mathcal{T}^{o_{k+1}}$. Thus, the ranking o_{k+1} inherits the tie-breaking rule used for o_k so that the difference between o_k and o_{k+1}, which must satisfy $o_k \in \mathcal{O}(\gamma_k, \mathbf{0})$ and $o_{k+1} \in \mathcal{O}(\gamma_{k+1}, \mathbf{0})$, is minimized. Corollary 1 can be used to check whether the γ_{k+1} is a fixed point of the function $\mathcal{T}^{o_k}$. Also, γ_{k+1} and o_{k+1} are uniquely determined by γ_k and o_k. We can consider (γ_k, o_k) as an entity which is an argument delivered to the function $\mathcal{T}^{o_k}(\gamma_k)$, and wish to find a fixed point in this sense.

In the general case, the function $\mathcal{T}^{o_k}(\gamma_k)$ is discontinuous in γ_k and the sequence $\{\gamma_k\}_{k=0}^\infty$ is heuristically generated with no proof of convergence to a fixed point. In fact, the choice of $\gamma_{k+1} = (\mathcal{T}^{o_k}(\gamma_k))^+$ may result in the sequence $\{\gamma_k\}_{k=0}^\infty$ being trapped in oscillations. To avoid this, with slight abuse of notation, we modify the iteration as $\gamma_{k+1} = (c\mathcal{T}^{o_k}(\gamma_k) + (1-c)\gamma_k)^+$ with a parameter $c \in [0, 1]$, which captures the effects of exploring the new point $\mathcal{T}^{o_k}(\gamma_k)$. Numerical examples of iterating γ_k will be provided in Section 5.

With an upper bound, $U \in \mathbb{N}_+$, we take $k^* := \arg\min_{k=1,2,\ldots,U} \|\gamma_{k-1} - \gamma_k\|$ and consider o_{k^*} as a reasonably good ranking of PS pairs. Such o_{k^*} is precomputable with computational complexity no worse than $O(U(N^2 + J^2))$, where N^2 and J^2 result from ordering the N pairs and solving the J linear equations, respectively. In Section 4, we show that an index policy feasible for the original problem can always be generated with such an o_{k^*}, and the implementation complexity is $O(I)$ in terms of computation and storage.

3.3. Weakly Coupled Constraints

Here, we discuss a sufficient condition under which the sequence $\{\gamma_k\}_{k=0}^\infty$ is provably convergent; and, in Section 5, when this condition fails, we show via numerical examples that the sequence might still converge.

Definition 3. Recall the matrix $\mathcal{W} = (w_{j,i})$ defined in Section 2.1. We say that row $j \in [J]$ is

1. A type 1 row if there is at most one $i \in [I]$ with $w_{j,i} > 0$;
2. A type 2 row if there is more than one $i \in [I]$ with $w_{j,i} > 0$.

That is, row j is a type 1 row if resource pool j is not shared by patterns of different types and is a type 2 row, otherwise. Denote by $\mathcal{J}_i = \{j \in [J] \mid w_{j,i} > 0\}$ the set of resource pools used by pattern i. We then define a condition.

3.3.1. Weak Coupling.

A system is weakly coupled if, for any pattern i, there is at most one $j \in \mathcal{J}_i$ with row j of $\mathcal{W}$ being a type 2 row.

This condition implies that there is at most one shared resource pool associated with each pattern. In a weakly coupled system, if pattern i_1 shares a resource pool j_{12} with pattern i_2 and pattern i_1 shares a resource pool j_{13} with pattern i_3, then $j_{12} = j_{13}$. A system where each of the patterns requires only one resource pool is clearly weakly coupled. In a weakly coupled system, dependencies between state variables of different patterns still exist, because each resource pool can be shared by requests of multiple RTs.

Definition 4. For a weakly coupled system define, for each $i \in [I] \backslash \{d(\ell) : \ell \in [L]\}$, $w_i^* = w_{j,i}$, where j is the only resource pool in $\mathcal{J}_i$ shared with other patterns, if there is one; or any member of the set $\arg\min_{j' \in \mathcal{J}_i} \frac{C_{j'}}{w_{j',i}}$, otherwise.

Definition 5. For a weakly coupled system define, for $\nu \in \mathbb{R}^L$, a set of PS rankings $\mathcal{O}^*(\nu) \subset \mathcal{O}$ such that, for any $o \in \mathcal{O}^*(\nu)$, PS pairs $\iota \in [N]$ are ranked according to the descending order of

$$\Xi_\iota^* = \begin{cases} \frac{\Xi_{i_\iota}(\mathbf{0},\mathbf{0}) - \nu_{\ell_\iota}}{w_{i_\iota}^*(1+\lambda_{\ell_\iota}/\mu_{i_\iota})}, & \text{if } \nexists \ell \in [L],\ i_\iota = d(\ell), \\ 0, & \text{otherwise.} \end{cases} \tag{26}$$

Proposition 4. *If the system is weakly coupled and there exists a ranking $o \in \mathcal{O}^*(\mathbf{0})$ satisfying $\nu(o, \mathbf{0}) = \mathbf{0}$, then the capacity constraints described in (10) are decomposable and the policy $\bar{\varphi}(o)$ is optimal for the relaxed problem defined by (8) and (9)–(11). In particular, there exist decomposable multipliers $\gamma \in \mathbb{R}_0^J$ satisfying, for $j \in [J]$,*

(i) *if there is a critical PS pair* $\iota \in \mathscr{I}(o)$ *with critical resource pool* $j = j_\iota(o)$, *and no* $j' \neq j$ *with* $j' \in \mathscr{J}_{i_\iota}$ *is critical for any other PS pair* $\iota' \in \mathscr{I}(o)$, *then*

$$\gamma_j = \frac{\Xi_{i_\iota}(\mathbf{0},\mathbf{0}) - \nu_{\ell_\iota}}{w_{j,i_\iota}(1 + \lambda_{\ell_\iota}/\mu_{i_\iota})}; \tag{27}$$

(ii) *if there are critical PS pairs* ι *and* ι' *in* $\mathscr{I}(o)$ *with critical resource pools* $j = j_\iota(o) \neq j_{\iota'}(o)$ *and* $j_{\iota'}(o) \in \mathscr{J}_{i_\iota}$, *then*

$$\gamma_j = \frac{w_{j_{\iota'}(o),i_\iota}}{w_{j,i_\iota}}\left(\frac{\Xi_{i_\iota}(\mathbf{0},\mathbf{0}) - \nu_{\ell_\iota}}{w_{j_{\iota'}(o),i_\iota}(1 + \lambda_{\ell_\iota}/\mu_{i_\iota})} - \frac{\Xi_{i_{\iota'}}(\mathbf{0},\mathbf{0}) - \nu_{\ell_{\iota'}}}{w_{j_{\iota'}(o),i_{\iota'}}(1 + \lambda_{\ell_{\iota'}}/\mu_{i_{\iota'}})}\right); \tag{28}$$

(iii) *otherwise*,

$$\gamma_j = 0. \tag{29}$$

The proof is given in Appendix EC.7 in the e-companion. From Lemma 1, for any critical PS pairs $\iota, \iota' \in \mathscr{I}(o)$ with $\iota \neq \iota'$, it follows that $i_\iota \neq i_{\iota'}$. If the system is weakly coupled, for any $j \in [J]$, there exist at most two different critical pairs $\iota \in \mathscr{I}(o)$ satisfying $j \in \mathscr{J}_{i_\iota}$. Also, in a weakly coupled system, for the second case stated in Proposition 4, if there are critical PS pairs ι and ι' in $\mathscr{I}(o)$ with critical resource pools $j = j_\iota(o) \neq j_{\iota'}(o)$ and $j_{\iota'}(o) \in \mathscr{J}_{i_\iota}$, then $j_\iota(o) \notin \mathscr{J}_{i_{\iota'}}$ because there is at most one resource pool in $\mathscr{J}_{i_\iota}$ shared with other patterns.

In Proposition 4, the assumption that the system is weakly coupled constrains the way in which resource pools are shared by different requests. The case where there is an $o \in \mathscr{O}^*(\mathbf{0})$ with $\boldsymbol{\nu}(o,\mathbf{0}) = \mathbf{0}$ will occur when the relaxed action Constraint (9) is satisfied with $\alpha_{d(\ell)}^{\bar{\varphi}(o)}(n) > 0$ for the only $n \in \mathscr{N}_{d(\ell)}$ and for all $\ell \in [L]$. To see this, the construction of the policy $\bar{\varphi}(o)$ guarantees that the resulting multipliers $\boldsymbol{\nu}(o,\mathbf{0})$ will be nonnegative, and therefore it follows from (21) that $\alpha_{d(\ell)}^{\bar{\varphi}(o)}(n) > 0$ only if $\nu_\ell(o,\mathbf{0}) = 0$. That is, having $\nu_\ell(o,\mathbf{0}) = 0$ is associated with there being a positive probability that the dummy pattern $d(\ell)$ is selected in the relaxed system. Furthermore, if there is a PS pair ι (for a nondummy pattern $i_\iota \in \mathscr{P}_\ell$) that satisfies the condition described in I, that is PS pair ι causes the relaxed action Constraint (9) to bite, Algorithm 1 will ensure that $\alpha_{i_{\iota'}}^{\bar{\varphi}(o)}(n_{\iota'}) = 0$ for all PS pairs ι' ranked lower than ι according to the order o. In particular, this will cause $\alpha_{d(\ell)}^{\bar{\varphi}(o)}(n) = 0$ for the only $n \in \mathscr{N}_{d(\ell)}$. When an inequality constraint "bites," it is satisfied with equality.

Therefore, if $\alpha_{d(\ell)}^{\bar{\varphi}(o)}(n) > 0$, it is because the relaxed capacity Constraints (10) bite before the relaxed action Constraints (9). If this is true for all ℓ, then the capacity constraints are biting for every request type, and therefore we refer to the case where there is an $o \in \mathscr{O}^*(\mathbf{0})$ with $\boldsymbol{\nu}(o,\mathbf{0}) = \mathbf{0}$ as a *heavy traffic* condition.

3.3.2. Heavy Traffic. The system is in heavy traffic if there is a ranking $o \in \mathscr{O}^*(\mathbf{0})$ such that $\boldsymbol{\nu}(o,\mathbf{0}) = \mathbf{0}$.

Remark 4. The property of being weakly coupled and in heavy traffic simplifies the analysis of the complementary slackness condition of the relaxed problem. In particular, the index related to a pattern, described in Equation (20), is affected only by the multipliers of resource pools $j \in [J]$ with $w_{j,i} > 0$. Weak coupling helps reduce the number of such multipliers γ_j, so that the index of a pattern is affected by at most one γ_j, which in turn affects other pattern indices. When the system is weakly coupled and in heavy traffic, we can analytically solve the I linear Equations (24) and (25) and derive the ϕ and $\boldsymbol{\gamma}$ that satisfy the complementary slackness condition described in Equations (22) and (23). A detailed discussion is provided in the proof of Proposition 4.

Proposition 4 guarantees the decomposability of a system when it is weakly coupled and in heavy traffic. The property of being weakly coupled and in heavy traffic is stronger than necessary for decomposability, but it is simple to check and is satisfied in a number of common resource allocation problems. Some examples of how to define such a system are given in Appendix EC.8 in the e-companion.

4. The Index Policy: Its Implementation in the Nonasymptotic Regime

In Section 3, we considered the relaxed problem with Constraints (9)–(11). Here, we return to the original problem with Constraints (1) and (5).

For each RT $\ell \in [L]$, we refer to a policy $\varphi \in \Phi$ as an *index policy* according to PS-pair ranking $o \in \mathscr{O}$, if it always prioritizes a candidate process in a PS pair with a ranking equal or higher than those of all the other candidate processes. This policy φ is applicable to the original problem while, the policy $\bar{\varphi}(o)$ proposed in Section 3.2.1 is not in general. The method of implementing such a φ is not unique; for instance, the computation of the ranking of the PS pairs can vary. Here we propose one possible implementation.

For $t > 0$, we maintain a sequence of I ordered PS pairs $(i, N_i^\varphi(t))$ $(i \in [I])$ that are associated with the I patterns, according to the given ranking o and the state vector $\boldsymbol{N}^\varphi(t)$: PS pair $(i, N_i^\varphi(t))$ is placed ahead of $(i', N_{i'}^\varphi(t))$ if and only if the former precedes the latter in the ranking o. Let $i_\sigma^o(\boldsymbol{N}^\varphi(t))$ $(\sigma \in [I])$ represent the pattern associated with the σth PS pair in this ordered sequence.

For a general ranking $o \in \mathscr{O}$, the variables $i_\sigma^o(\boldsymbol{N}^\varphi(t))$ are potentially updated at each state transition. Nonetheless, for the purpose of this paper, we mainly focus on the rankings $o \in \mathscr{O}(\boldsymbol{\gamma}, \boldsymbol{\nu})$ (for some $\boldsymbol{\gamma} \in \mathbb{R}_0^J$ and $\boldsymbol{\nu} \in \mathbb{R}^L$) that follow the descending order of $\Xi_i(\boldsymbol{\gamma}, \boldsymbol{\nu})$. In this case, the variables $i_\sigma^o(\boldsymbol{N}^\varphi(t))$ are updated only if a pattern $i \in [I] \backslash \{d(\ell) | \ell \in [L]\}$ transitions into or out of its boundary state $|\mathscr{N}_i| - 1$.

Under the index policy φ, we select L patterns to accept new arrivals of L types according to their orders in sequence $i^o_\sigma(\boldsymbol{N}^\varphi(t))$ ($\sigma \in [I]$). In particular, at a decision making epoch $t > 0$, we initialize $a^\varphi_i(\boldsymbol{N}^\varphi(t)) = 0$ for all $i \in [I]$ and a set of *available patterns* to be $[I]$. If, for $i = i^o_1(\boldsymbol{N}^\varphi(t))$, Constraints (5) will not be violated by setting $a^\varphi_i(\boldsymbol{N}^\varphi(t)) = 1$, then set $a^\varphi_i(\boldsymbol{N}^\varphi(t)) = 1$ and remove all patterns associated with request type $\ell(i)$ from the set of available patterns.

The other $L-1$ patterns are selected similarly and iteratively. That is, we look for the smallest $\sigma \in \{2, 3, \dots, I\}$ such that

- $i^o_\sigma(\boldsymbol{N}^\varphi(t))$ remains in the set of available patterns; and
- Capacity Constraints (5) will not be violated by setting $a^\varphi_i(\boldsymbol{N}^\varphi(t)) = 1$, where $i = i^o_\sigma(\boldsymbol{N}^\varphi(t))$.

If there is such a σ, set $a^\varphi_i(\boldsymbol{N}^\varphi(t)) = 1$ for $i = i^o_\sigma(\boldsymbol{N}^\varphi(t))$, remove all patterns associated with request type $\ell(i)$ from the set of available patterns, and continue selecting the remaining $L-2$ patterns in the same manner. When all the L patterns have been selected, we can stop. Detailed steps are provided in Algorithm 2, which has a computational complexity that is linear in I.

Algorithm 2. Implementing the Index Policy φ with respect to ranking o.

Input: a ranking of PS pairs $o \in \mathcal{O}$ and a given state $\boldsymbol{n} \in \mathcal{N}$.

Output: the action variables $\boldsymbol{a}^\varphi(\boldsymbol{n})$ under the index policy $\varphi \in \Phi$ with respect to ranking o when the system is in state $\boldsymbol{n}$.

Function IndexPolicy $o, \boldsymbol{n}$:
- $\boldsymbol{a}^\varphi(\boldsymbol{n}) \leftarrow \boldsymbol{0}$ /*Initializing the action variables */
- $\mathcal{P} \leftarrow [I]$ /* Initializing the set of available patterns */
- $\sigma \leftarrow 1$ /* Starting with the pattern with the highest priority */
- **while** $\mathcal{P} \neq \emptyset$ **do**
  - $i \leftarrow i^o_\sigma(\boldsymbol{n})$
  - **If** $i \in \mathcal{P}$ **and** Constraints (5) *are not violated by setting* $a^\varphi_i(\boldsymbol{n}) = 1$ *and* $\boldsymbol{N}^\varphi(t) = \boldsymbol{n}$ then
    - $a^\varphi_i(\boldsymbol{n}) \leftarrow 1$
    - Remove all patterns $i' \in \mathcal{P}$ with $\ell(i') = \ell(i)$ from $\mathcal{P}$
  - **end**
  - $\sigma \leftarrow \sigma + 1$
- **end**

return $a^\alpha(n)$

The performance of φ is mainly determined by the given order $o \in \mathcal{O}$. Based on later discussion of the asymptotic regime, if the policy $\bar{\varphi}(o)$ is optimal for the relaxed problem in the asymptotic regime, then φ is asymptotically optimal for the original problem. Even without the proved asymptotic optimality, the ranking o should ensure good performance of φ as it is always rational to prioritize patterns according to their potential profits. As long as there are reasonably good $\boldsymbol{\gamma}$ and $\boldsymbol{\nu}$ leading to a $o \in \mathcal{O}(\boldsymbol{\gamma}, \boldsymbol{\nu})$, which correctly reflects the potential profits of patterns, the performance degradation of $\bar{\varphi}(o)$ is likely to be limited for the relaxed problem and close to the optimal solution of the original problem; and the index policy φ derived from o is a promising choice for managing resources. For a given ranking o, if the policy $\bar{\varphi}(o)$ is optimal for the relaxed problem, then the index policy φ derived from o approaches optimality for the original problem as the capacity $\boldsymbol{C}$ and arrival rates $\boldsymbol{\lambda}$ tend to infinity proportional to a scaling parameter h; that is, the φ is *asymptotically optimal*. A rigorous discussion of asymptotic optimality, including the definition of h, is provided in Appendix EC.11 in the e-companion. We refer to the limiting case as $h \to \infty$ as the *asymptotic regime*.

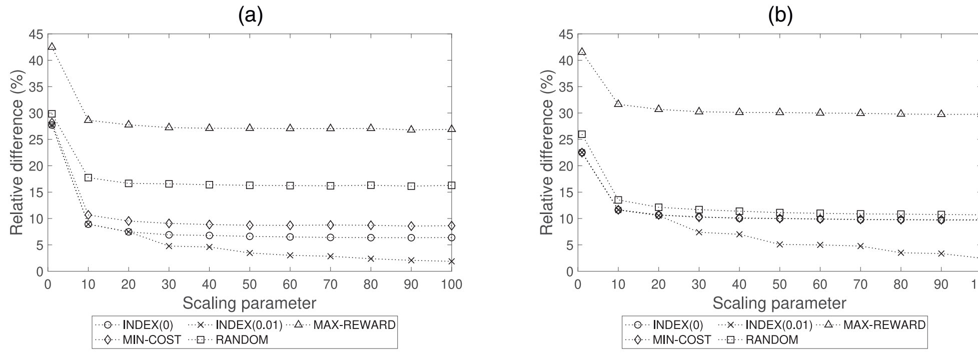
5. Numerical Results

We demonstrate via simulation the performance of the index policy φ as the capacities $\boldsymbol{C}$ and arrival rates $\boldsymbol{\lambda}$ are scaled by the scaling parameter h discussed in Appendix EC.9.2 in the e-companion. We concentrate on systems that are not weakly coupled or in heavy traffic. In this section, the confidence intervals of all the simulated average revenues at the 95% level based on the Student t distribution are maintained within ±3% of the observed mean.

Along with the fixed-point iteration method proposed in Section 3.2.2, we have been able to find systems that are not weakly coupled or in heavy traffic but are decomposable. Here, we provide two examples, where L and J are sampled uniformly from the sets $\{2, 3, 4, 5\}$ and $\{10, 11, \dots, 20\}$, respectively. As described in Appendix EC.9 in the e-companion, the asymptotic behavior of the stochastic process scaled by h under policy φ is further constrained by parameters $\boldsymbol{\epsilon} \in [0,1]^{J \times N}$. The numbers $\boldsymbol{\epsilon}$ are important in deriving the index policy φ as applied to the original problem that maximizes (8) subject to (1) and (5). Let $\epsilon_M = \max_{j \in [J], \iota \in [N]} \epsilon_{j,\iota}$. In Figures 3 and 4, we referred to an index policy φ with specific $\epsilon_M \in [0,1]$ as INDEX(ϵ_M).

We consider three baseline policies: two greedy policies that prioritize patterns with maximal reward rates and minimal cost rates and one policy randomly uniformly selecting an available pattern for each request type. We refer to the three policies as *Max-Reward*, *Min-Cost*, and *Random*. The Max-Reward and Min-Cost policies are in fact index policies with PS pairs ranked according to the descending order of their reward rates and the ascending order of their cost rates, respectively. The Random policy was proposed by Stolyar (2017) for a Virtual Machine replacement

Figure 3. Relative Difference of a Specific Policy to $R(o_{k^*})$ Against the Scaling Parameter of the System



Note. (a) Nonzero decomposable multipliers; and (b) zero decomposable multipliers.

problem, aiming to minimize the system blocking probabilities in the case with finite capacities. It is not a feasible policy of the original problem with capacity Constraints (5) because it does not reserve resource units for a specific pattern that is more profitable than the others. When there are not enough resource units in a pool to accommodate multiple request types that have chosen their patterns involving this pool, the Random policy will always assign the resource units to the request that arrives first.

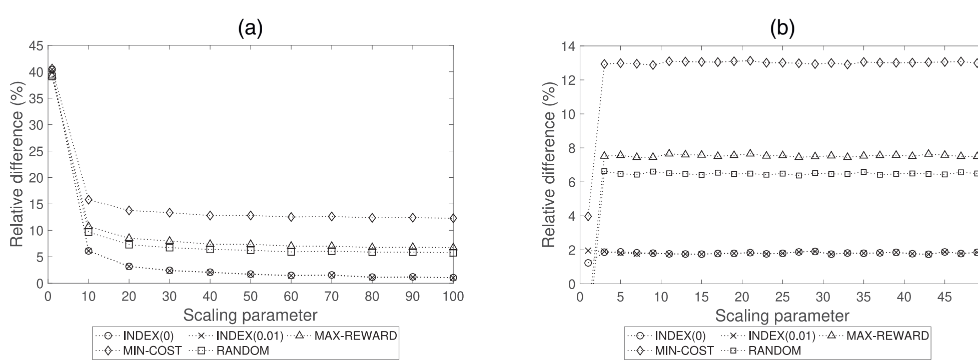
In Figure 3, we compare the performance of INDEX(0), INDEX(0.01), the baseline policies, and $\bar{\varphi}(o_{k^*})$, where o_{k^*} is the ranking of the multipliers γ_{k^*} resulting from the fixed point iteration method (described in Section 3.2.2) with parameter $c = 0.5$ and initial point $\gamma_0 = \mathbf{0}$. The system parameters are listed in Appendix EC.16 in the e-companion and are generated by pseudo-random functions. The discovered multipliers γ_{k^*} for simulations in Figure 3, (a) and (b), are $(269.555, 0, 0, 0, 0, 273.11, 0, 347.995, 0, 0, 0, 8.323 \times 10^{-7}, 9.726 \times 10^{-5}, 0)$ and $\mathbf{0}$, respectively, satisfying $\mathcal{T}^{o_{k^*}}(\gamma_{k^*}) = \gamma_{k^*}$ in the asymptotic regime. By Proposition 3, these γ_{k^*} are decomposable multipliers and, by Theorem EC.2 in the e-companion, the index policies derived from the rankings o_{k^*} are asymptotically optimal. Let $R(o) := \lim_{h\to+\infty} R^{\bar{\varphi}(o),h}$ $(o \in \mathscr{O})$ of which the existence is guaranteed in the proof of Theorem EC.1 in the e-companion. For the decomposable systems with $h < +\infty$ and $\bar{\varphi}(o_{k^*})$ optimal for the relaxed problem in the asymptotic regime, the asymptotic long-run average revenue, $R(o_{k^*})$, is no less than the optimum of the original problem: $R(o_{k^*})$ is an upper bound of $R^{\phi,h}$ for any $\phi \in \Phi$.

Figure 3 illustrates the relative difference of average revenues, $(R(o_{k^*}) - R^{\phi,h})/R(o_{k^*})$ for ϕ = INDEX(0), INDEX(0.01), Max-Reward, Min-Cost, and Random, against the scaling parameter h.

In this context, there are two aspects of performance evaluation presented in Figure 3. First, we see the performance of the index policies in the nonasymptotic regime by comparing their long-run average revenues with an upper bound on the optimum. In particular, Figure 3, (a) and (b), shows that INDEX(0.01) significantly outperforms INDEX(0) for large h: the small but positive ϵ does affect the performance of φ. The performance of INDEX(0.01) is close to the upper bound of the optimal solution with relative difference less than 5% for h greater than 50 in all three examples.

On the other hand, by comparing to $R(o_{k^*})$, a trend of coincidence between $R^{\text{INDEX}(0.01),h}$ and $R(o_{k^*})$ is observed in Figure 3 as h increases from 1 to 100, consistent with the proved asymptotic optimality of φ.

Figure 4. An Example with Non-Decomposable Multipliers



Notes. (a) Relative difference of a specific policy to $R(o_{k^*})$ against scaling parameter of the system. (b) Relative difference of a specific policy to $R(o_k)$ against k.

Recall that the examples presented in Figure 3 are not for systems with weak coupling or heavy traffic but the index policy φ is still proved to be asymptotically optimal here. Also, the performance of φ is close to the optimum without requiring extremely large h.

In Figure 4, we consider another example with multipliers that are not decomposable (that is, $\mathcal{T}^{o_{k^*}}(\gamma_{k^*}) \neq \gamma_{k^*}$). Similar to Figure 3, in Figure 4(a), we plot the relative difference of revenue of INDEX(0), INDEX(0.01), and the baseline policies to $R(o_{k^*})$ against the scaling parameter; whereas in Figure 4(b), fixing the scaling parameter $h = 50$, we illustrate curves of the relative differences, $(R(o_k) - R^{\phi,h})/R(o_k)$ (ϕ = INDEX(0), INDEX(0.01), Max-Reward, Min-Cost, Random), against the number of iterations k for the fixed point iteration method. The rankings o_k are potentially different as k varies, which influences $R(o_k)$. In Figure 4(a), the INDEX(0) and INDEX(0.01) are proposed based on the ranking o_{k^*}, whereas, with slightly abused notation, in Figure 4(b), INDEX(0) and INDEX(0.01) represent the index policies φ, which are derived from the rankings o_k associated with the varying k, with $\epsilon_M = 0$ and 0.01, respectively. The system parameters for the simulations in Figure 4 are listed in Appendix EC.16 in the e-companion.

Figure 4(a) can be read in a similar way to Figure 3 except that $R(o_{k^*})$ is not a proved upper bound for the average revenue for the original problem. Here, INDEX(0) and INDEX(0.01) perform similarly and numerically converge to $R(o_{k^*})$ as h increases, although the system is not necessarily decomposable. The convergence is consistent with Theorem EC.1 in the e-companion that generally holds without requiring decomposability. On the other hand, for each finite h (which corresponds to the nonasymptotic regime), INDEX(0) and INDEX(0.01) significantly outperform all the other baseline policies, although the system is not proved to be decomposable.

Figure 4(b) illustrates the performance trajectory as the iteration number k (the x axis) for the fixed point iteration method increases for a system with $h = 50$ (in the nonasymptotic regime). Recall that, for the simulations presented here, the average revenues of INDEX(0) and INDEX(0.01) and $R(o_k)$ are varying with k, whereas all of the baseline policies are independent of k. We observe a sharp jump on the curves between $k = 1$ and 5. This is caused by the initial setting, $\gamma_0 = \mathbf{0}$, which is not a good choice of multipliers. After several steps of the iteration method, the curves in Figure 4(b) become almost flat; that is, the values of $R(o_k)$, $R^{\mathrm{INDEX}(0)}$, and $R^{\mathrm{INDEX}(0.01)}$ become relatively stable for $k = 5$ to 50. Also, in Figure 4(b), after the performance becomes stable, INDEX(0) and INDEX(0.01) achieve clearly higher long-run average revenues than those of the baseline policies: given the poor setting at the beginning, the fixed point iteration method can still lead to a reasonably good ranking o_{k^*} and its associated multipliers γ_{k^*}.

6. Conclusions

We modeled a resource allocation problem, described by (8), (1), and (5), as a combination of various RMABPs coupled by limited capacities of the shared resource pools, which are shared, competed for, and reserved by requests. This presents us with an optimization problem for a stochastic system, aimed at maximizing the long-run average revenue by dynamically accommodating requests into appropriate resource pools.

Using the ideas of Whittle relaxation (Whittle 1988) and the asymptotic optimality proof of Weber and Weiss (1990), we proved the asymptotic optimality of an index policy (referred to as φ) if the capacity constraints are decomposable with multipliers $\gamma \in \mathbb{R}_0^J$ (Theorem EC.2 in the e-companion).

We proved a sufficient condition, described as the property of being weakly coupled and in heavy traffic, for the existence of such decomposable multipliers and the asymptotic optimality of policy φ (Corollary EC.1 in the e-companion). The property is not necessary but is easy to verify and covers a significant class of resource allocation problems. We have listed examples of systems with the property satisfied in Section 3.3.

In a general system, we proposed a fixed point method to fine tune the multipliers $\gamma \in \mathbb{R}_0^J$ and a ranking $o \in \mathcal{O}(\gamma, \mathbf{0})$. We proved that, if there exists a fixed point $\gamma \in \mathbb{R}_0^J$ of the function $\mathcal{T}^o$ satisfying $o \in \mathcal{O}(\gamma, \mathbf{0})$, then this γ is a vector of decomposable multipliers. We successfully discovered the decomposable multipliers in some situations without assuming weak coupling or heavy traffic by applying the fixed-point method. Also, in Section 5, we compared the index policy φ with different parameter $\boldsymbol{\epsilon}$ to baseline policies through simulations for systems that are not weakly coupled or in heavy traffic in the nonasymptotic regime. The index policy achieves clearly higher performance than the baseline policies. To the best of our knowledge, no existing work provides rigorous asymptotic optimality for a resource allocation problem where dynamic allocation, competition, and reservation are permitted.

Acknowledgments

References

Avrachenkov KE, Borkar VS (2016) Whittle index policy for crawling ephemeral content. *IEEE Trans. Control Network System* 5(1): 446–455.

Bäuerle N (2002) Optimal control of queueing networks: An approach via fluid models. *Adv. Appl. Probabilities* 34(2):313–328.
Bäuerle N (2000) Asymptotic optimality of tracking policies in stochastic networks. *Ann. Appl. Probab.* 1081(2-3):1065–1083.
Bertsimas D, Niño-Mora J (1996) Conservation laws, extended polymatroids and multiarmed bandit problems: A polyhedral approach to indexable systems. *Math. Oper. Res.* 21(2):257–306.
Bertsimas D, Nasrabadi E, Paschalidis IC (2015) Robust fluid processing networks.*IEEE Trans. Automated Controls* 60(3):715–728.
Cezik MT, L'Ecuyer P (2008) Staffing multiskill call centers via linear programming and simulation. *Management Sci.* 54(2):310–323.
Chen X, Han Z, Zhang H, Xue G, Xiao Y, Bennis M (2017) Wireless resource scheduling in virtualized radio access networks using stochastic learning. *IEEE Trans. Mobile Comput.* 17(4):961–974.
Coddington EA, Levinson N (1955) *Theory of Ordinary Differential Equations* (Tata McGraw-Hill Education, McGraw-Hill, New York).
Esposito F, Di Paola D, Matta I (2016) On distributed virtual network embedding with guarantees. *IEEE/ACM Trans. Networks* 24(1): 569–582.
Fikar C, Hirsch P (2017) Home healthcare routing and scheduling: A review. *Comput. Oper. Res.* 77:86–95.
Freidlin MI, Wentzell AD, Szücs J (2012) *Random Perturbations of Dynamical Systems* (Springer Science & Business Media, Berlin).
Fu J (2016) Energy-efficient heuristics for job assignment in server farms. PhD thesis, Department of Electronic Engineering, City University of Hong Kong, Hong Kong.
Fu J, Moran B, Guo J, Wong EWM, Zukerman M (2016) Asymptotically optimal job assignment for energy-efficient processor-sharing server farms. *IEEE J. Selected Areas Comm.* 34(12): 4008–4023.
Gittins JC (1979) Bandit processes and dynamic allocation indices. *J. Royal Statist. Soc. B* 41(2):148–177.
Gittins JC, Jones DM (1974) A dynamic allocation index for the sequential design of experiments. Gani J, ed. *Progress in Statistics* (North-Holland, Amsterdam), 241–266.
Gittins JC, Glazebrook K, Weber RR (2011) *Multi-Armed Bandit Allocation Indices*, 2nd ed. (Wiley, New York).
Kelly FP (1991) Loss networks. *Ann. Appl. Probab.* 1(3):319–378.
Krishnamurthy V, Djonin DV (2007) Structured threshold policies for dynamic sensor scheduling: A partially observed Markovdecision process approach. *IEEE Trans. Signal Processing* 55(10): 4938–4957.
Larrañaga M, Ayesta U, Verloop IM (2015) Asymptotically optimal index policies for an abandonment queue with convex holding cost. *Queueing Systems* 81(2-3):99–169.
Le Ny J, Feron E, Dahleh MA (2010) Scheduling continuous-time Kalman filters. *IEEE Trans. Automated Controls* 56(6):1381–1394.
Lieder A, Moeke D, Koole G, Stolletz R (2015) Task scheduling in long-term care facilities: A client-centered approach. *Oper. Res. Health Care* 6:11–17.
Liu H, Liu K, Zhao Q (2012) Learning in a changing world: Restless multiarmed bandit with unknown dynamics. *IEEE Trans. Inform. Theory* 59(3):1902–1916.
Meyn S (2008) *Control Techniques for Complex Networks* (Cambridge University Press, Cambridge, UK).
Nazarathy Y, Weiss G (2009) Near optimal control of queueing networks over a finite time horizon. *Ann. Oper. Res.* 170(1):233–249.
Niño-Mora J (2001) Restless bandits, partial conservation laws and indexability. *Adv. Appl. Probabilities* 33(1):76–98.
Niño-Mora J (2002) Dynamic allocation indices for restless projects and queueing admission control: A polyhedral approach. *Math. Programming* 93(3):361–413.
Niño-Mora J (2007) Dynamic priority allocation via restless bandit marginal productivity indices. *TOP* 15(2):161–198.
Niño-Mora J (2012) Admission and routing of soft real-time jobs to multiclusters: Design and comparison of index policies. *Comput. Oper. Res.* 39(12):3431–3444.
Niño-Mora J (2019) Resource allocation and routing in parallel multi-server queues with abandonments for cloud profit maximization. *Comput. Oper. Res.* 103:221–236.
Papadimitriou CH, Tsitsiklis JN (1999) The complexity of optimal queuing network control. *Math. Oper. Res.* 24(2):293–305.
Ross SM (1992) *Applied Probability Models with Optimization Applications* (Dover Publications, New York).
Serfozo R (2009) *Basics of Applied Stochastic Processes* (Springer Science & Business Media, Berlin).
Stolyar AL (2004) Maxweight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic. *Ann. Appl. Probabilities* 14(1):1–53.
Stolyar AL (2013) An infinite server system with general packing constraints. *Oper. Res.* 61(5):1200–1217.
Stolyar AL (2017) Large-scale heterogeneous service systems with general packing constraints. *Adv. Appl. Probabilities* 49(1):61–83.
Verloop IM (2016) Asymptotically optimal priority policies for indexable and non-indexable restless bandits. *Ann. Appl. Probabilities* 26(4):1947–1995.
Wallace RB, Whitt W (2004) Resource pooling and staffing in call centers with skill-based routing. *Oper. Res.* 7(4):276–294.
Wang J, Ren X, Mo Y, Shi L (2019) Whittle index policy for dynamic multi-channel allocation in remote state estimation. *IEEE Trans. Automatic Control* 99:1.
Weber RR, Weiss G (1990) On an index policy for restless bandits. *J. Appl. Probabilities* 3:637–648.
Wei X, Neely MJ (2017) Data center server provision: Distributed asynchronous control for coupled renewal systems. *IEEE/ACM Trans. Networks* 25(4):2180–2194.
Whittle P (1988) Restless bandits: Activity allocation in a changing world. *J. Appl. Probabilities* 25:287–298.

Jing Fu is a lecturer in the discipline of Electronic & Telecommunications Engineering, Royal Melbourne Institute of Technology (RMIT) University, since 2020. Her research interests now include energy-efficient networking/scheduling, resource allocation in large-scale networks, semi-Markov/Markov decision processes, restless multi-armed bandit problems, and stochastic optimization.

Bill Moran currently serves as Professor of Defence Technology at the University of Melbourne. He was elected to the Fellowship of the Australian Academy of Science in 1984. His main areas of research interest are in signal processing both theoretically and in applications to radar, waveform design and radar theory, sensor networks, and sensor management. He also works in various areas of mathematics including harmonic analysis, representation theory, and number theory.

Peter G. Taylor currently is a Redmond Barry Professor at University of Melbourne. His research interests lie in the fields of stochastic modeling and applied probability, with particular emphasis on applications in telecommunications, biological modeling, healthcare, economics, and disaster management. In 2013, he was awarded a Laureate Fellowship by the Australian Research Council, in 2017 the Ren Potts Medal by the Australian Society for Operations Research, in 2018 the George Szekeres Medal by the Australian Mathematical Society and in 2019 the ANZIAM Medal.