Queueing Models and
Service Management
Vol. 2, No. 2, page 174-200, 2019

QMSM

Distributed Server Allocation for Content Delivery Networks

Sarath Pattathil, Vivek S. Borkar* and Gaurav Kasbekar

Department of Electrical Engineering
Indian Institute of Technology Bombay
Powai, Mumbai 400076, India.
(Received March 2018 ; accepted June 2019)

Abstract: We propose a dynamic formulation of file-sharing networks in terms of an average cost Markov decision process with constraints. By analyzing a Whittle-like relaxation thereof, we propose an index policy in the spirit of Whittle and compare it by simulations with other natural heuristics.

Keywords: Distributed control, Resource pooling, Whittle index.

1. Introduction

Recently, Content Delivery Networks (CDNs), which distribute content (*e.g.*, video and audio files, webpages) using a network of server clusters situated at multiple geographically distributed locations, have been extensively deployed in the Internet by content providers themselves (*e.g.*, Google) as well as by third-party CDNs that distribute content on behalf of multiple content providers (*e.g.*, Akamai's CDN distributes Netflix and Hulu content) (see Kurose and Ross [17], Leighton [21]). The delay incurred in downloading content to an end user is often significantly lower when a CDN is used compared to the case where all content is downloaded from a single centralized host, since the server clusters of a CDN are located close to end users [17], [21].

In this paper, we consider a server cluster which contains $M \geq 2$ servers and is part of a CDN. The server cluster stores N large file types (*e.g.*, videos). There is a high demand for each file type and therefore each file type is replicated across multiple servers within the cluster. Each file type is characterized by the average size of the file it stores. We do not maintain the identity of each individual file for every file type, but instead assume that the size of each file from any particular file type comes from a distribution. From now on, we refer to the file types as files for sake of brevity. Requests for the N files from end users or from smaller server clusters arrive at the server cluster from time to time. There are two approaches to serving the file requests (refer to Shah [29] and Shah and de Veciana [31]).

1. *Single Server Allocation*: Each file request is served by a single server [29, 31].
2. *Resource Pooling*: Each file request is simultaneously served by multiple

*Corresponding author
Email : borkar@ee.iitb.ac.in

servers, in particular, different chunks of the file are served by different servers in parallel [29, 31].

Resource pooling has been found to outperform single server allocation in prior studies (see Shah and de Veciana [29, 30, 31]). Hence in this paper, we assume that resource pooling is used. Also, we allow multiple files to be simultaneously downloaded from a given server. At any time instant, the sum of the rates at which a server j transmits different files is constrained to be at most μ_j. Requests for different files are stored in different queues, and there is a cost for storing a request in a queue. Let $\xi^{ij}(t)$ be the rate at which server j transmits file i at time t. We consider the problem of determining the rates $\xi^{ij}(t)$ for each i, j and t so as to minimize the expected time-averaged storage cost. We formulate this problem as a Markov Decision Process (MDP) (see Guo and Hérnandez-Lerma [14]). We show that this problem is Whittle-like indexable (see Whittle [36]) and use this result to propose a Whittle-like scheme [36] that can be implemented in a distributed manner[1]. We evaluate the performance of our scheme using simulations and show that it outperforms several natural heuristics for the problem such as Balanced Fair Allocation, Uniform Allocation, Weighted Allocation, Random Allocation and Max-Weight Allocation.

We now review related prior literature. In Shah and de Veciana [30], performance of Content Delivery Networks is evaluated in a static framework. This work also studies the tradeoffs between delay for each packet vs the energy used etc. The polymatroid structure of the service capacity in this model is exploited to get an expression for mean file transfer delay that is experienced by incoming file requests. Performance of dynamic server allocation strategies, such as random server allocation or allocation of least loaded server, are also explored. We use the model of [30] for CDN, but go a step further by looking at a fully dynamic optimization problem as an MDP.

In Shah and de Veciana [31], a centralized content delivery system with collocated servers is studied. Files are replicated in these servers and these serve as a pooled resource which cater to file requests. The article shows how dynamic server capacity allocation outperforms simple load balancing strategies such as those which assign the least loaded server, or assign the servers at random. The article also goes on to study file placement strategies that improve the utility of the system.

Several works including Leconte *et al.* [19, 20] and Moharir *et al.* [23] look at large-scale content delivery networks, focusing on placement of content in the servers. Of these, Leconte *et al.* [19] also studies the greedy method of server allocation and its efficiency under various regimes of server storage capacities, and under what content placement strategy it would be efficient. Zhou *et al.* [37] studies strategies for scheduling after the

[1]We use the phrase 'Whittle-like' instead of just Whittle because the scheme introduced in this paper, although in the same spirit of Whittle's original paper, is not exactly the same.

content placement stage, and proposes an algorithm, called the Fair Sharing with Bounded Degree (FSBD), for server allocation.

In Shah and de Veciana [32], multiclass queuing systems are studied with different arrival rates. The service rates are constrained to be in a symmetric polymatroid region. Large scale systems with a growing number of service classes are studied and several asymptotic results regarding fairness and mean delays are obtained.

Multi-server models are studied in Tsitsiklis and Xu [33] with each server connected to multiple file types and each file type stored in multiple servers, thereby creating a bipartite graph. This article focuses on the scaling regime where the number of servers goes to infinity. It is shown that even if the average degree $d_n << n :=$ the number of servers, an asymptotically vanishing queuing delay can be obtained. These results are based on a centralized scheduling strategy.

In Bonald and Comte [6], multi-server queues are studied with an arbitrary compatibility graph between jobs and servers. The paper designs a scheduling algorithm which achieves balanced fair sharing of the servers. Several parameters are analyzed using this policy by drawing a parallel between the state of the system at any time to that of a Whittle network.

However, none of the above papers [6, 19, 20, 23, 31, 32, 33] show Whittle indexability of the respective resource allocation problems they address. The work closest in spirit to ours is Larranaga *et al.* [18], which studies a Whittle indexability scheme for birth and death restless bandits. These model server allocation to queues, but it does not study the case when there are multiple servers storing the same file types as is the case in general content delivery networks. In the present work we take an alternative approach which considers a dynamic optimization or control problem that can be interpreted as a problem of scheduling restless bandits. We analyze it in the framework laid down by Whittle for deriving a heuristic index policy [36]. To the best of our knowledge, this paper is the first to show Whittle-like indexability of the server allocation problem in the setting of a CDN server cluster that uses resource pooling, with the objective of minimization of the expected time-averaged file request storage cost. The fact that this problem is Whittle-like indexable allows us to decouple the original average cost MDP, which is difficult to solve directly, into more tractable separate control problems for individual file types. The decoupling leads to an efficient algorithm based on computation of Whittle-like indices, which outperforms several natural heuristics for the problem. Our proof techniques broadly follow the general scheme of Agarwal *et al.* [1], albeit with some differences.

The Whittle index heuristic has been successfully applied to various resource allocation problems including crawling for ephemeral content [3], congestion control [4], UAV routing [26], sensor scheduling [25], routing in clusters [24], opportunistic scheduling [10], inventory routing [2], cloud computing [11] etc. General applications to resource

allocation problems can be found in Larranaga *et al.* [18]. Book length treatments of restless bandits can be found in Jacko [16] and Ruiz-Hernandez [28].

The rest of the paper is structured as follows. In section 2, we discuss our model and formulate the problem as a Markov Decision Process (MDP). Section 3 shows various structural properties of the value function of the MDP formulated in section 2. In section 4, we prove that the problem of server allocation in the resource pooling setting is in fact indexable and provide a scheme to compute this index. Section 5 discusses other heuristics for server allocation and presents numerical comparisons of the proposed index policy with other heuristics. We conclude the paper with a brief discussion in Section 6.

We conclude this section with a brief introduction to the Whittle index [36]. Let $X^i(t), t \geq 0, 1 \leq i \leq N,$ be N Markov chains, each with two modes of operations: active and passive, with associated transition kernels $p_1(\cdot|\cdot), p_0(\cdot|\cdot)$ respectively. Let $r_1^i(X^i(t)), r_0^i(X^i(t))$ be instantaneous rewards for the i^{th} chain in the respective modes with $r_1^i(\cdot) \geq r_0^i(\cdot)$. The goal is to schedule active/passive modes so as to maximize the total expected average reward

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_j E[r^j_{\nu^j(t)}(X^j(t))]$$

where $\nu^j(t) = 1$ if j th process is active at time t and 0 if not, under the constraint $\sum_j \nu^j(t) \leq M, \forall\, t$, i.e., at most M processes are active at each time instant. This hard constraint makes the problem difficult to solve (see Papadimitriou and Tsitsiklis [27]). So following Whittle, one relaxes the constraint to

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_j E[\nu^j(t)] \leq M.$$

(Whittle considers an equality constraint, but the logic here is analogous.) This makes it a problem with separable cost and constraints which, given the Lagrange multiplier λ, decouples into N individual problems with reward for passivity changed to $\lambda + r_0(\cdot)$. The problem is Whittle indexable if under optimal policy, the set of passive states increases *monotonically* from empty set to full state space as λ varies from $-\infty$ to $+\infty$. If so, the Whittle index for a given state can be defined as the value of λ for which both modes (active and passive) are equally desirable. The index policy is then to compute these for the current state profile, sort them in decreasing order, and render active the top M processes, the rest passive. The decoupling implies $O(N)$ growth of state space as opposed to the original problem, for which it is exponential in N. Further, the processes are coupled only through a simple index policy. The latter is known to be asymptotically optimal under certain conditions as $M, N \to \infty$ in constant ratio (refer to Weber and Weiss [34]). However, no convenient general analytic bound on optimality gap seems available.

2. Model and Problem Formulation

Consider a server cluster that contains multiple servers, each of which stores one or more files. We represent this system using a bipartite graph[2] $G=(F\cup S;E)$ where F is a set of N files, S is a set of M servers, E is the set of edges, and each edge $e\in E$ connecting a file $i\in F$ and server $j\in S$ implies that a copy of file i is replicated at server j (see Figure 1).

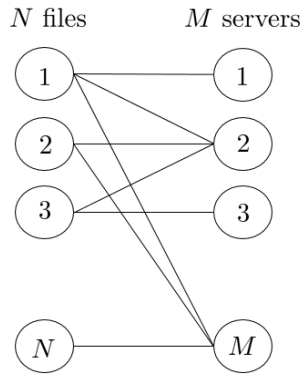

Figure 1. The model used in this paper. A link between file i and server j denotes that a copy of file i is stored in server j.

For $j\in S$, F_j denotes the set of files that are stored in server j. Similarly, for $i\in F$, S_i denotes the set of servers that store file i. Requests for file $i\in F$ arrive to the server cluster according to an independent Poisson process with rate Λ^i and are queued in a separate queue for each file type. We assume that the job (requested file) sizes have an exponential distribution. (For sake of simplicity, we assume their means to be identically equal to one. More general cases can be handled by suitable scaling of the $\xi^{ij}(\cdot)$'s defined below.). Let $\xi^{ij}(t)$ denote the rate at which server j transmits file type i at time t. Then the capacity constraint at each server can be expressed as

$$\sum_{i\in F_j}\xi^{ij}(t)\le\mu_j,\forall\, t\ge 0,\, j\in\{1,2,\cdots,M\}, \tag{1}$$

where μ_j is the maximum permissible rate of transmission from server j.

Let $f^i(x)$ be the cost for storing x jobs in the queue i. We assume $f^i(\cdot)$ to be an increasing strictly convex function (see Bertsekas [5]) for $i=1,2,\cdots,N.$ (We comment on the strict convexity assumption at the end of Section 3). Our aim is to minimize the long run average cost, given by

---

[2]Recall that a graph $G=(V,E)$ is said to be *bipartite* if its node set V can be partitioned into two sets F and S such that every edge in E is between a node in F and a node in S [35].

$$\limsup_{T\to\infty}\frac{1}{T}\int_0^T \mathbb{E}[\sum_{i=1}^N f^i(X^i(t))]dt,$$

where $X^i(t)$ is the length of queue i at time t.

This makes it a *continuous time Markov decision process* with the state process given by $\hat{X}(t)=[X^1(t),\cdots,X^N(t)],\ t\geq 0,$ taking values in the state space S^N where $S:=\{0,1,2,\cdots\}$ with control process $\xi(t):=\{\xi^{ij}(t)\}_{i\in F, j\in S_i},\ t\geq 0$, taking values in the compact control space $U:=\{u^{ij}, i\in F, j\in S_i : \sum_{i\in F_j} u^{ij}\leq \mu_j\ \forall\ j\}$. We shall consider as admissible control policies the $\{\xi^{ij}(\cdot)\}$ whereby one has the controlled Markov property, i.e., for $t\geq 0,\delta>0$,

$$P(\hat{X}(t+\delta)=y\mid \hat{X}(s),\xi(s),s\leq t)=q(y\mid x(t),\xi(t))\delta+o(\delta)$$

for a 'controlled rate matrix' $q=[[q(y\mid x,u)]],\ x,y\in S^N,\ u\in U$. A special case is that of the control policies wherein $\xi(t)$ is adapted to $\hat{X}(s),\ s\leq t$, for all $t\geq 0$. As usual, one has the important special subclasses of control policies, viz., stationary deterministic policy wherein $\xi(t)=v(\hat{X}(t))$ for a prescribed $v(\cdot): S^N\mapsto U,$ and stationary randomized policy wherein the conditional law of $\xi(t)$ given $\hat{X}(s),\ s\leq t,$ depends on $\hat{X}(t)$ alone.

Stability assumption: We assume there exists a stationary randomized policy under which the cost is finite (which in particular implies that the policy is stable in the sense that the corresponding Markov chain $\hat{X}(\cdot)$ is positive recurrent), and, in addition,

$$\sum_{j\in S_i}\mu_j>\Lambda^i,\ \forall\ i. \tag{2}$$

The stability assumption above ensures the existence of at least one stationary randomized policy under which the process is stable. Our assumption on the f^i's implies that $\lim_{x\to\infty} f^i(x)=\infty,\forall\ i$, implying in turn that the cost is *near monotone* [9] in the sense that it penalizes high values of the state $\|\hat{X}(t)\|$. In particular, an unstable control policy that leads to transience or null recurrence will lead to an infinite cost.

Note that $X^i(\cdot), 1\leq i\leq N,$ are in fact individual controlled Markov chains coupled through their controls that have to satisfy the constraint (1) that couples them. This forces us to view the combined process $\hat{X}(\cdot)$ as a single controlled Markov chain. The Whittle device we use below allows us to undo this for purposes of analysis via a clever heuristic. Specifically, the controlled rate matrix $Q^i(t), t\geq 0,$ of $X^i(\cdot)$ is given by for $z>0$,

$$Q^i(z+1\mid z,\xi^{ij}(t), j\in S_i)=\Lambda^i,$$
$$Q^i(z-1\mid z,\xi^{ij}(t), j\in S_i)=\sum_{j\in S_i}\xi^{ij}(t),$$

$$Q^i(z \mid z, \xi^{ij}(t), j \in S_i) = -(\Lambda^i + \sum_{j \in S_i} \xi^{ij}(t)),$$

for $z = 0$,

$$Q^i(1 \mid 0, \xi^{ij}(t), j \in S_i) = \Lambda^i,$$
$$Q^i(0 \mid 0, \xi^{ij}(t), j \in S_i) = -\Lambda^i.$$

Following the classic paper of Whittle [36], we relax the M instantaneous constraints (1) to M averaged constraint

$$\limsup_{T \to \infty} \frac{1}{T} \int_0^T \sum_{i \in F_j} \mathbb{E}[\xi^{ij}(t)] dt \leq \mu_j, \quad \forall j \in \{1, 2, \cdots, M\}, \tag{3}$$

where we assume that $0 \leq \xi^{ij}(t) \leq \mu_j \ \forall i, j, t$. Specifically, we have replaced the M hard constraints (1) that apply at *each* time instant by M *average* constraints which allow the violation of (1) from time to time, but requires it to hold only in an average sense. In particular, the left hand side of (3) can be viewed as another average cost functional. This makes it a classical constrained Markov decision process (refer to Borkar [9]). This has an equivalent formulation as a linear program on the space of measures, in terms of the so called *ergodic occupation measures* [9]. These measures are defined as probability measures on the product space $S^N \times U$ that are of the form

$$\Phi(dx, du) = \Phi_0(dx)\Phi_1(du \mid x),$$

where Φ_0 is the marginal on S^N which is required to be the stationary distribution of the Markov chain controlled by $\Phi_1(du \mid x)$, and the regular conditional law in the above decomposition is interpreted as a stationary randomized policy. The control problem can then be identified with the problem of minimizing the integral of the running cost $\hat{f}(\cdot, \cdots, \cdot) := \sum_i f^i(\cdot)$ w.r.t. this measure, a linear functional thereof, over the set of all ergodic occupation measures which turns out to be a closed convex set characterized by a set of linear equalities and inequalities. Specifically, one has

$$\text{Minimize} \quad \int \hat{f} d\Phi(dx, U)$$
$$\text{subject to:} \quad \Phi \geq 0, \Phi(S^N \times U) = 1,$$
$$\int \Phi(dx, du) \prod_i Q^i(y^i \mid x^i, u^{ij}, j \in S_i) = 0.$$

See [9] for details. This facilitates the use of standard tools of abstract convex optimization in this context. While we do not need the details thereof here, we do require one consequence of it, viz., that it allows one to consider an equivalent unconstrained average cost problem with cost

$$\lim_{T\to\infty}\frac{1}{T}\int_0^T\mathbb{E}\left[\sum_{i=1}^{N}f^i(X^i(t))+\sum_j\hat{\lambda}_j(\sum_{i\in F_j}\xi^{ij}(t)-\mu_j)\right]dt,$$

where $\hat{\lambda}_j \geq 0$ is the Lagrange multiplier associated with the j^{th} relaxed constraint

$$\lim_{T\to\infty}\frac{1}{T}\int_0^T\mathbb{E}[\sum_i\xi^{ij}(t)]\,dt \;\leq\; \mu_j.$$

(We replace the conventional '$lim\,sup_{T\to\infty}$' in analysis of average cost control by '$lim_{T\to\infty}$' by exploiting the fact that the results of [9] allow us to restrict to stationary randomized policies for which the $lim\,sup_{T\to\infty}$ above is in fact the $lim_{T\to\infty}$.) Since the cost is now separable in $X^i(\cdot)$'s, given the values of the Lagrange multipliers $\hat{\lambda}_j$, this optimization problem decouples into separate control problems for individual processes $X^i(\cdot)$, with the cost function for the i th process (file type) being given by

$$c^i(x^i,\hat{\lambda})=f^i(x^i)+\sum_{j\in S_i}\hat{\lambda}_j(\xi^{ij}(t)-\mu_j),$$

where $\hat{\lambda}=[\hat{\lambda}_1,\hat{\lambda}_2,\cdots\hat{\lambda}_M]$ is a vector containing all $\hat{\lambda}_{j'}s$. The average cost dynamic programming (DP) equation for this MDP for file type i is given by Guo and Hérnandez-Lerma [14]

$$\min_{\mu_j\geq\xi^{ij}\geq 0,\,j\in S_i}(c^i(x,\hat{\lambda})-\beta^i+\sum_y V^i_{\hat{\lambda}}(y)Q^i(y\,|\,x,\xi^{ij},j\in S_i))=0, \tag{4}$$

where

- β^i is the optimal cost for file type i,
- $V^i_{\hat{\lambda}}(\cdot)$ is the value function (sometimes called the 'relative value function').

In what follows, we drop the dependence of $V^i_{\hat{\lambda}}(\cdot)$ on i and $\hat{\lambda}$ for sake of notational simplicity and bring it back only when needed for the analysis. Substituting the values of Q^i back in the DP equation and dropping the superscript i (except from ξ^{ij}) for ease of notation, we have[3]

for $x>0$,

$$\min_{\mu_j\geq\xi^{ij}\geq 0,\,j\in S_i}(c(x,\hat{\lambda})-\beta+V(x+1)\Lambda+V(x-1)\sum_{j\in S_i}\xi^{ij}-V(x)(\Lambda+\sum_{j\in S_i}\xi^{ij}))=0, \tag{5}$$

equivalently,

$$\min_{\mu_j\geq\xi^{ij}\geq 0,\,j\in S_i}(f(x)+\sum_{j\in S_i}\hat{\lambda}_j(\xi^{ij}-\mu_j)-\beta+V(x+1)\Lambda$$
$$+V(x-1)\sum_{j\in S_i}\xi^{ij}(t)-V(x)(\Lambda+\sum_{j\in S_i}\xi^{ij}))=0. \tag{6}$$

[3]Note that when the queue of file type i is empty, no server needs to provide any service to that particular file type.

Adding $V(x)$ on both sides of equation (13), we get

$$V(x) = \min_{\mu_j \geq \xi^{ij} \geq 0, j \in S_i} (f(x) + \sum_{j \in S_i} \hat{\lambda}_j (\xi^{ij} - \mu_j) - \beta + V(x+1)\Lambda^i + V(x-1)\sum_{j \in S_i} \xi^{ij} + V(x)(1 - (\Lambda^i + \sum_{j \in S_i} \xi^{ij}))). \quad (7)$$

The equations for $x = 0$ can be written in a simiar fashion with appropriate modifications.

We now adapt the idea of uniformization to pass from a continuous time Markov chain to a discrete time Markov chain. If we scale all transition rates by a fixed multiplicative factor, it is tantamount to time scaling which will scale the average cost, but not affect the optimal policy. Hence without loss of generality, we can assume that the arrival and service rates are such that the coefficients of $V(\cdot)$ that appear in the right hand side of equation (7) are between some $\varepsilon > 0$ and 1 and can be interpreted as transition probabilities of a discrete time controlled Markov chain. Thus (7) is a dynamic programming equation for a discrete time Markov decision process with average cost. Note that the equation at best specifies V only up to an additive scalar, so for its well-posedness, in the least we need to add a qualification such as (say) $V(0) = 0$. We shall make this choice (which is by no means unique) and stay with it. See Borkar [8], Chapter VI, (in particular, Theorem 4.1, p. 87) for a complete treatment of well-posedness of (7). One only needs to verify the assumption therein of 'stability under local perturbation' which states that a stable stationary deterministic policy remains so if we change the control choice at exactly one state. This is immediate if each state has at most finitely many successors, as is the case here (see Lemma 1.1, p. 71, of [8]). We take the foregoing as given, suffice to say that the near-monotonicity of the cost and existence of a stable stationary randomized policy with finite cost by virtue of the 'Stability Assumption' above play a crucial role in establishing the DP equation.

As we are working with a fixed i, the control space is $U^i := \prod_{j \in S_i} [0, \mu_j]$ and a stationary deterministic policy corresponds to $\xi^{ij}(n) = \varphi(X(n))$ for a measurable $\varphi : S \mapsto U^i$, where $X(\cdot)$ is the corresponding controlled Markov chain, now in discrete time (We drop the superscript i for notational convenience.). We shall identify this policy with the map φ by a standard abuse of notation.

The expression which is to be minimized on the right hand side of (7) is linear in ξ^{ij}, $j \in S_i$ and each ξ^{ij} has the capacity constraint which restricts the values of ξ^{ij} to be $\leq \mu_j$, i.e., $\xi^{ij} \in [0, \mu_j]$. This, combined with the fact that the objective is linear, ensures that the minimum is attained at a corner where each server is either serving at full capacity or at zero capacity, i.e., at $\xi^{ij} = 0$ or $\xi^{ij} = \mu_j$ for all $j \in S_i$.

This achieves the first simplification in Whittle's program, viz., to decouple the original hard problem into N simpler problems. But unlike in the original Whittle case, where the decision was binary between active and passive modes, we have multiple decision variables, ξ^{ij} for each i. The foregoing shows that each one separately entails a binary

decision between 0 and μ_j respectively. Our approach to arriving at a Whittle-like policy is the most common one, viz., to first show the existence of an optimal threshold policy and then establish the monotonicity of the threshold in the Lagrange multiplier. Even the notion of a threshold does not make sense in a control space without a natural order, thus we need to reduce the problem to a situation where such is the case. This suggests that we apply the Whittle philosophy separately to each control variable in isolation, keeping the rest fixed at their respective capacities $\mu_{[\cdot]}$. We make this the basis for coming up with a Whittle-like index policy. Like the original Whittle scheme, this too is a heuristic, which we later compare with other natural heuristics empirically and find that it performs quite well in comparison. Our motivation for this specific choice and no other is as follows. In principle, we could fix any values of all but one control variable in order to reduce it to a single control variable case, but fixing the rest at maximum rate, which aids stability, puts the least onus on the flagged control variable vis-a-vis stability. To amplify this point, consider, e.g., the other extreme where we fix all other rates to zero. Then to ensure the existence of at least one stable stationary randomized policy for the decoupled problem, we would need a stronger restriction than the above 'Stability Assumption'. Observe in particular that we are now considering separate control problems associated not only for each process i separately, but for separate *pairs* of process $X^i(\cdot)$ and control $\xi^{ik}(\cdot)$ for a prescribed k, having fixed $\xi^{ij}(\cdot) \equiv \mu_j, \forall\ j \neq k.$ The sole variable being manipulated now takes values in an ordered set $[0, \mu_k]$ which facilitates search for an optimal threshold policy.

This also has the added bonus that all but one Lagrange multiplier drop out of each such DP equation, facilitating later the definition of Whittle-like index that would otherwise be quite messy.

We emphasize again that this is a heuristic policy just like the original Whittle case and need not be optimal. An optimal policy for the exact coupled problem will face the curse of dimensionality in a major way. To see this, suppose we use finite buffers of a constant size for each queue as an approximation and assume S_i, F_j are independent of i, j respectively, denoted simply as S_i, F_j, respectively. The state space for the original problem is the product of individual state spaces of the queues, which grows exponentially in $|S|$. In contrast, after decoupling the problem using Lagrange multipliers, it grows linearly in $|S|$. This is exactly the same problem which motivates the original Whittle index.

Since all other servers are serving at full rate, we have that

$$\xi^{ij}(t) = \mu_j, \quad \forall\ j \in S_i,\ j \neq k, \forall\ t \text{ .s.t. } X^i(t) > 0.$$

Let $\lambda_k = -\hat{\lambda}_k \mu_k$. We interpret λ_k as the marginal disutility of allowing server k to not serve when all other servers containing the file type are already serving at their full capacity. (This is only an interpretation of λ_k, in reality the servers may not always serve at full

capacity if the queue is not large enough.) This disutility plays the role of 'subsidy' in the original Whittle formulation which dealt with a reward maximization problem instead of cost minimization. On substituting $\xi^{ij} = \mu_j, \forall\, j \in S_i,\, j \neq k$, we have, for $x > 0$,

$$V(x) = f(x) - \beta + \min\left(\lambda_k + \sum_y p_1(y \mid x) V(y), \sum_y p_2(y \mid x) V(y) \right). \tag{8}$$

Here $p_1(\cdot \mid \cdot)$ is the transition probability when the server does not serve this file type and is given by (for $x > 0$)

$$\begin{aligned} p_1(x+1 \mid x) &= \Lambda, \\ p_1(x \mid x) &= 1 - (\Lambda + \sum_{j \in S_i, j \neq k} \mu_j), \\ p_1(x-1 \mid x) &= \sum_{j \in S_i, j \neq k} \mu_j, \end{aligned} \tag{9}$$

and $p_2(\cdot \mid \cdot)$ is the transition probability when the server serves this file type and is given by (for $x > 0$)

$$\begin{aligned} p_2(x+1 \mid x) &= \Lambda, \\ p_2(x \mid x) &= 1 - (\Lambda + \sum_{j \in S_i} \mu_j), \\ p_2(x-1 \mid x) &= \sum_{j \in S_i} \mu_j. \end{aligned} \tag{10}$$

For $x = 0$, the transition probabilities $p_1(\cdot \mid \cdot)$ and $p_2(\cdot \mid \cdot)$ are the same and are given by (for $i = 1, 2$)

$$\begin{aligned} p_i(1 \mid 0) &= \Lambda, \\ p_i(0 \mid 0) &= 1 - \Lambda. \end{aligned}$$

In the next section, we prove some structural properties of the value function.

3. Structural Properties of the Value Function

This section closely follows in spirit the approach of Agarwal *et al.* [1], Borkar [8], and Borkar *et al.* [11, 12], but with significantly different proofs.

Lemma 3.1 *$V(\cdot)$ is non-decreasing in the number of files.*

Proof. (Sketch) We use a 'pathwise coupling' argument. Consider initial conditions $x < x'$ in S and an optimal, therefore stable (i.e., positive recurrent) stationary deterministic policy $v(\cdot)$. Consider the controlled chains $X(n), X'(n), n \geq 0,$ as follows. We use the standard formulation of a controlled Markov chain as a dynamics driven by control and noise, i.e.,

$$X(n+1) = F(X(n), \xi(n), \zeta(n+1)),$$
$$X'(n+1) = F(X'(n), \xi(n), \zeta(n+1)),$$

with $X(0) = x, X'(0) = x'$, where $\{\xi_n\}$ is the control process, $\{\zeta(n)\}$ is i.i.d. noise uniform on $[0,1]$, and F is some measurable map. Note that the map F, the driving noise $\{\zeta(n)\}$, and the control sequence $\{\xi(n)\}$ is common across both. It is always possible to replicate the processes in law on a common probability space in this fashion. In addition, we choose $\xi(n) = v(X'(n)), \forall n$. This choice is optimal for $X'(\cdot)$, but not for $X(\cdot)$. In particular, $X'(\cdot)$ is a positive recurrent Markov chain and hits state 0 infinitely often with probability 1. Each time this happens, $X'(\cdot) - X(\cdot)$ drops by 1, hence

$$\tau := \min\{n \geq 0 : X'(n) = X(n)\} < \infty, \ a.s..$$

Note that by our construction,

- we have

$$X'(m) > X(m) \ \forall \ m < \tau, \tag{11}$$
$$= X(m) \ for \ m \geq \tau, \tag{12}$$

and

- for $n < \tau$, either $X'(m+1) - X(m+1) = X'(m) - X(m)$ or $X'(m+1) - X(m+1) = X'(m) - X(m) - 1$ and the latter case occurs only if $X(m) = X(m+1) = 0$ and $X'(m+1) = X'(m) - 1$.

For $x = X(m)$, respectively, $X'(m)$, (7) leads to

$$E\left[V(X'(m)) - V(X(m))\right] \geq E\left[V(X'(m+1)) - V(X(m+1))\right].$$

Iterating, we get for $T \geq 1$,

$$V(x') - V(x) \geq E\left[V(X'(\tau \wedge T)) - V(X(\tau \wedge T))\right].$$

Letting $T \to \infty$ and using Fatou's lemma, we have

$$V(x') - V(x) \geq E\left[V(X'(\tau)) - V(X(\tau))\right] = 0.$$

Lemma 3.2. *$V(\cdot)$ is strictly convex, strictly increasing, and has the property of increasing differences, i.e., for $z > 0$ and $x > y$*

$$V(x+z) - V(x) > V(y+z) - V(y).$$

Proof. The proof follows along similar lines as Lemma 6 in Borkar and Pattathil [12] and Theorem 4 in Agarwal *et al.* [1], but with several crucial differences. The argument uses induction. We embed the state space to the positive real line, $\mathbb{R}^+$. Take $x_1, x_2 \in S, x_2 > x_1 > 0$. Let $V_n(\cdot)$ denote the $\alpha-$discounted $n-$step problem (For $x < 0$, we define $V_n(x) = V_n(0)$). Let u be the optimal control for state x at time n. We have

$$V_n(x) = f(x) + \alpha V_{n-1}(x+1)\Lambda + \alpha V_{n-1}(x)(1-\Lambda-\sum_i \mu_i) + \alpha V_{n-1}(x-1)\sum_{j\neq k}\mu_j + (1-u)\lambda$$
$$+ \alpha(1-u)\mu_k V_{n-1}(x) + \alpha u \mu_k V_{n-1}(x-1). \tag{13}$$

We have that $V_0(x) \equiv f(x),$ which is strictly convex. Assume that V_{n-1} is convex. For x_1, x_2 as above, let $u_i, i=1,2$, be the minimizers for $x = x_i, i=1,2,$ respectively. in (13). Then

$$\begin{aligned} V_n(x_1) + V_n(x_2) &= f(x_1) + f(x_2) \\ &+ \alpha V_{n-1}(x_1)(1-\Lambda-\sum_j \mu_j) + \alpha V_{n-1}(x_2)(1-\Lambda-\sum_j \mu_j) \\ &+ \alpha V_{n-1}(x_1+1)\Lambda + \alpha V_{n-1}(x_2+1)\Lambda \\ &+ \alpha V_{n-1}(x_1-1)\sum_{j\neq k}\mu_j + \alpha V_{n-1}(x_2-1)\sum_{j\neq k}\mu_j \\ &+ (1-u_1)\lambda_k + \alpha(1-u_1)\mu_k V_{n-1}(x_1) + \alpha u_1 \mu_k V_{n-1}(x_1-1) \\ &+ (1-u_2)\lambda_k + \alpha(1-u_2)\mu_k V_{n-1}(x_2) + \alpha u_2 \mu_k V_{n-1}(x_2-1). \end{aligned}$$

Consider two separate cases depending on the values of u_1, u_2.

Case 1: $u_1 = u_2$

$$\begin{aligned} & V_n(x_1) + V_n(x_2) \\ & \geq^{*1} 2f\left(\frac{x_1+x_2}{2}\right) + 2\alpha V_{n-1}\left(\frac{x_1+x_2}{2}\right)(1-\Lambda-\sum_j \mu_j) \\ & \quad + 2\alpha V_{n-1}\left(\frac{x_1+x_2}{2}+1\right)\Lambda + 2\alpha V_{n-1}\left(\frac{x_1+x_2}{2}-1\right)\sum_{j\neq k}\mu_j \\ & \quad + 2\lambda_k\left(1-\frac{u_1+u_2}{2}\right) + 2\alpha\left(1-\frac{u_1+u_2}{2}\right)\mu_k V_{n-1}\left(\frac{x_1+x_2}{2}\right) \\ & \quad + 2\alpha\left(\frac{u_1+u_2}{2}\right)\mu_k V_{n-1}\left(\frac{x_1+x_2}{2}-1\right) \\ & \geq^{*2} 2f\left(\frac{x_1+x_2}{2}\right) + 2\alpha V_{n-1}\left(\frac{x_1+x_2}{2}\right)(1-\Lambda-\sum_j \mu_j) \\ & \quad + 2\alpha V_{n-1}\left(\frac{x_1+x_2}{2}+1\right)\Lambda + 2\alpha V_{n-1}\left(\frac{x_1+x_2}{2}-1\right)\sum_{j\neq k}\mu_j \\ & \quad + 2\lambda_k(1-u_3) + 2\alpha(1-u_3)\mu_k V_{n-1}\left(\frac{x_1+x_2}{2}\right) + 2\alpha u_3 \mu_k V_{n-1}\left(\frac{x_1+x_2}{2}-1\right) \\ & = 2V_n\left(\frac{x_1+x_2}{2}\right). \end{aligned}$$

Here u_3 is the optimal control when the state is $\frac{x_1+x_2}{2}$. Inequality *1 follows from the convexity of $f(\cdot)$ and $V_{n-1}(\cdot)$. Inequality *2 follows from the definition of the optimal control u_3.

Case 2: $u_1 \neq u_2$

Consider the case $u_2 = 0, u_1 = 1$ (The other case is similar)

$$
\begin{aligned}
&V_n(x_1) + V_n(x_2) \\
&\geq^{*1} 2f\left(\frac{x_1+x_2}{2}\right) + 2\alpha V_{n-1}\left(\frac{x_1+x_2}{2}\right)(1-\Lambda-\sum_j \mu_j) \\
&\quad + 2\alpha V_{n-1}\left(\frac{x_1+x_2}{2}+1\right)\Lambda + 2\alpha V_{n-1}\left(\frac{x_1+x_2}{2}-1\right)\times \\
&\quad \sum_{j\neq k}\mu_j + \lambda_k + \alpha\mu_k V_{n-1}(x_2) + \alpha\mu_k V_{n-1}(x_1-1) \\
&= 2f\left(\frac{x_1+x_2}{2}\right) + 2\lambda_k + 2\alpha V_{n-1}\left(\frac{x_1+x_2}{2}\right)(1-\Lambda-\sum_i \mu_i) \\
&\quad + 2\alpha V_{n-1}\left(\frac{x_1+x_2}{2}+1\right)\Lambda + 2\alpha V_{n-1}\left(\frac{x_1+x_2}{2}-1\right)\sum_{i\neq k}\mu_i \\
&\quad + 2\lambda_k\left(1-\frac{1}{2}\right) + 2\alpha\mu_k\left[\frac{1}{2}V_{n-1}(x_2) + \frac{1}{2}V_{n-1}(x_1-1)\right] \\
&\geq^{*2} 2f\left(\frac{x_1+x_2}{2}\right) + 2\alpha V_{n-1}\left(\frac{x_1+x_2}{2}\right)(1-\Lambda-\sum_i \mu_i) \\
&\quad + 2\alpha V_{n-1}\left(\frac{x_1+x_2}{2}+1\right)\Lambda + 2\alpha V_{n-1}\left(\frac{x_1+x_2}{2}-1\right)\sum_{i\neq k}\mu_i \\
&\quad + 2\lambda_k\left(1-\frac{1}{2}\right) + 2\alpha\mu_k\left[\frac{1}{2}V_{n-1}\left(\frac{x_1+x_2}{2}\right) + \frac{1}{2}V_{n-1}\left(\frac{x_1+x_2}{2}-1\right)\right] \\
&\geq^{*3} 2f\left(\frac{x_1+x_2}{2}\right) + 2\alpha V_{n-1}\left(\frac{x_1+x_2}{2}\right)(1-\Lambda-\sum_i \mu_i) \\
&\quad + 2\alpha V_{n-1}\left(\frac{x_1+x_2}{2}+1\right)\Lambda + 2\alpha V_{n-1}\left(\frac{x_1+x_2}{2}-1\right)\sum_{i\neq k}\mu_i \\
&\quad + 2\lambda_k(1-u_3) + 2\alpha(1-u_3)\mu_k V_{n-1}\left(\frac{x_1+x_2}{2}\right) + 2\alpha u_3\mu_k V_{n-1}\left(\frac{x_1+x_2}{2}-1\right) \\
&= 2V_n\left(\frac{x_1+x_2}{2}\right).
\end{aligned}
$$

Here u_3 is the optimal control when the state has $\frac{x_1+x_2}{2}$ files. Inequalities *1,*2 follow

from the convexity of $f(\cdot)$ and $V_{n-1}(\cdot)$ (we use the fact that convexity implies non-decreasing differences, i.e., $f(x+a)-f(x) \geq f(y+a)-f(y)$ for $x > y, a > 0$). Inequality *3 follows from the definition of the optimal control u_3.

Next consider the case where $x_1 > x_2 = 0$. We have

$$V_n(0) = f(0) + (1-u)\lambda_k + \alpha(1-\Lambda)V_{n-1}(0) + \alpha\Lambda V_{n-1}(1).$$

From this equation, we see that $u = 1$ if $\lambda_k > 0$ and $u = 0$ otherwise. We rearrange the above equation as

$$\begin{aligned} V_n(0) = {} & f(0) + (1-u)\lambda_k + \alpha(1-\Lambda-\sum_i \mu_i)V_{n-1}(0) \\ & + \alpha\sum_{i\neq k}\mu_i V_{n-1}(0) + \alpha\mu_k V_{n-1}(0) + \alpha\Lambda V_{n-1}(1). \end{aligned}$$

We have

$$\begin{aligned} & V_n(x_1) + V_n(0) \\ & = f(x_1) + f(0) + \alpha V_{n-1}(x_1)\left(1-\Lambda-\sum_i \mu_i\right) \\ & \quad + \alpha V_{n-1}(0)\left(1-\Lambda-\sum_i \mu_i\right) + \alpha V_{n-1}(x_1+1)\Lambda + \alpha V_{n-1}(1)\Lambda \\ & \quad + \alpha V_{n-1}(x_1-1)\sum_{i\neq k}\mu_i + \alpha V_{n-1}(0)\sum_{i\neq k}\mu_i \\ & \quad + (1-u_1)\lambda_k + (1-u_1)\alpha\mu_k V_{n-1}(x_1) \\ & \quad + \alpha u_1\mu_k V_{n-1}(x_1-1) + (1-u_2)\lambda_k + \alpha\mu_k V_{n-1}(0) \\ & \geq^{*1} 2V_n\left(\frac{x_1}{2}\right), \end{aligned}$$

where *1 is derived using convexity and by following similar arguments as in the case when $x_2 > 0$.

Therefore, by induction, we have that V_n is convex for all n. From equation (13), we see that V_n is the sum of a strictly convex function f and a convex function V_{n-1} when $x \geq 0$. This shows that V_n is in fact a strictly convex function for $x > 0$. (Note that $V_0 = f$, which is also strictly convex.) Letting $\tilde{V}_\alpha$ denote the value function of the infinite horizon α-discounted problem, we have $V_n \to \tilde{V}_\alpha$ pointwise by convergence of the value iteration algorithm. Since $V_n(x) - f(x),\ x \geq 0$ is convex for all n and convexity is preserved under pointwise convergence, $\tilde{V}_\alpha(x) - f(x), x \geq 0,$ is convex for all α. Letting $\bar{V}_\alpha(x) := \tilde{V}_\alpha(x) - \tilde{V}_\alpha(0)$, so will be $\bar{V}_\alpha - f$ for all α. By the vanishing discount argument of [1], $\bar{V}_\alpha \to$ the value function V of the average cost problem, pointwise. Thus $V - f$ is convex. Since f is strictly convex, it follows that $V(x),\ x \geq 0,$ is strictly convex. Strict convexity and non-decreasing property imply strict increase on $[0, \infty)$. Strict convexity also

implies strictly increasing differences. This proves the claim.

Lemma 3.3. *The optimal policy is a threshold policy, i.e.,* $\exists\, x^*$ *such that if* $x > x^*$, *the server serves at full capacity, otherwise the server does not serve this file type.*

Proof. In order to prove this, we show that the function

$$g(x) = \sum p_2(y \mid x)V(y) - \sum p_1(y \mid x)V(y)$$

is strictly decreasing. On simplifying this expression, we get

$$g(x) = \mu_k(V(x-1) - V(x)), \tag{14}$$

which is a strictly decreasing function in x by Lemma 3.2. Thus the minimizer in (16) changes from one to the other as this quantity crosses λ_k, while remaining fixed on either side thereof. This implies that the optimal policy is a threshold policy.

Note: We have made the assumption that the cost function f is strictly convex. We can relax this assumption to mere convexity and get analogous statements of Lemma 3.1 and 3.2, except that increasing will be replaced by non-decreasing. The only difference it makes is that the choice of threshold, and therefore of our Whittle-like index, may become non-unique over a closed interval wherever the value function has a linear patch. This can be disambiguated by using the convention that we use the smallest candidate value as the index, i.e., the smallest value of the state x for which it is equally desirable to be active or passive. It is easy to see that this is well defined and moreover, facilitates the ordinal comparisons in an unambiguous manner. Note that the scheduling policy depends only on such comparisons. Thus this does not cause any inconsistency and remains a plausible heuristic, though it is not clear how the performance get affected vis-a-vis the case when such ambiguities do not arise. That it still is a reasonable heuristic is supported by our simulations on a linear cost function reported below.

4. Whittle-like Indexability

We next prove a Whittle-like indexability result in the spirit of [36]. We use the phrase 'Whittle-like' because our problem formulation differs from that of [36], though it builds upon it.

Let π^ℓ denote the stationary probability distribution when the threshold is ℓ. That is, if the number of jobs is $\leq \ell$, then the server does not transmit, and if number of jobs is $> \ell$, then the server transmits at full rate. We have the following lemma.

Lemma 4.1. $\sum_{i=0}^{\ell} \pi^\ell(i)$ *is strictly increasing with* ℓ.

Proof. Let $\hat{\mu} = \sum_{j \in S_i; j \neq k} \mu_j$. The Markov chain formed with a threshold of ℓ is shown in

Figure 2. This is a time reversible Markov chain with stationary probabilities given by

$$\pi^{\ell}(i) = \pi^{\ell}(0)(\frac{\Lambda}{\hat{\mu}})^{i}, \quad \text{if } i \le \ell,$$

$$\pi^{\ell}(i) = \pi^{\ell}(0)(\frac{\Lambda}{\hat{\mu}})^{\ell}(\frac{\Lambda}{\mu_k + \hat{\mu}})^{i-\ell}, \quad \text{if } i > \ell,$$

where $\pi^{\ell}(0)$ is the stationary probability of state 0.

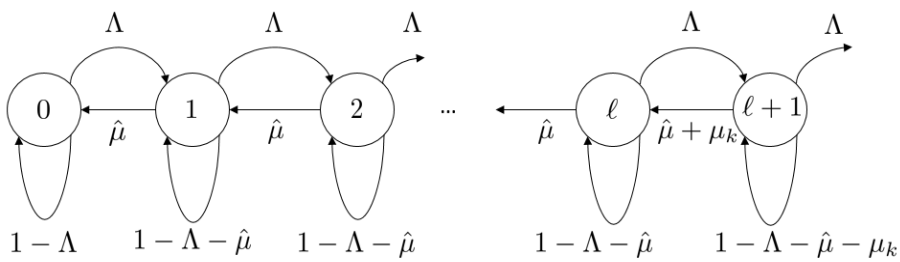

Figure 2. Markov Chain.

From this, we see that

$$\sum_{i=0}^{\ell}\pi^{\ell}(i) = \frac{\dfrac{(\frac{\Lambda}{\hat{\mu}})^{\ell+1}-1}{(\frac{\Lambda}{\hat{\mu}})-1}}{\dfrac{(\frac{\Lambda}{\hat{\mu}})^{\ell}-1}{(\frac{\Lambda}{\hat{\mu}})-1} + (\frac{\Lambda}{\hat{\mu}})^{\ell}(\frac{\hat{\mu}+\mu_k}{\hat{\mu}+\mu_k-\Lambda})},$$

which is a strictly increasing function of ℓ.

(Note that this formula holds when $\Lambda \neq \hat{\mu}$. If they are equal, we have $\sum_{i=0}^{\ell}\pi^{\ell}(i) = \dfrac{\ell+1}{\ell + \dfrac{\hat{\mu}+\mu_k}{\mu_k}}$ which is again an increasing function of ℓ.)

Theorem 4.1. *This problem is Whittle-like indexable in the sense that the set of passive states decreases monotonically from the whole state space to the empty set ϕ as $\lambda \to \infty$.*

Proof. The proof is along the lines of Theorem 1 in Borkar and Pattathil [12]. It has been reproduced for sake of completeness.

The optimal average cost of the problem is given by

$$\beta(\lambda) = \inf\{\sum_{i} f(i)\pi(i) + \lambda\sum_{i\in B}\pi(i)\},$$

where π is the stationary distribution and B is the set of passive states. The infimum

$\beta(\lambda)$ of this quantity affine in λ is over all admissible policies, which by Lemma 3.3 is the same as the infimum over all threshold policies. Hence $\beta(\cdot)$ is concave non-decreasing with slope <1. As a concave function of a scalar variable, it is differentiable except at countably many points and has right and left derivatives everywhere, which are non-increasing individually and across points of non-differentiability (i.e., at such points, the right derivative is less than or equal to the left derivative). By the envelope theorem (Theorem 1, Milgrom and Segal, [22]), the derivative of this function with respect to λ is given by

$$\sum_{i=0}^{x(\lambda)} \pi^{x(\lambda)}(i),$$

where $x(\lambda)$ is the optimal threshold under λ. In fact, since the threshold is discrete, it is seen that $\beta(\cdot)$ is piecewise linear with this derivative at points of differentiability, whereas at points of non-differentiability the two possible values thereof define the super-gradient. Since $\beta(\lambda)$ is a concave function, its derivative has to be a non-increasing function of λ, i.e.,

$$\sum_{i=0}^{x(\lambda)} \pi^{x(\lambda)}(i) \text{ is non-increasing with } \lambda.$$

But, from Lemma 4.1, we know that $\sum_{j=0}^{\ell} \pi^{\ell}(j)$ is a strictly increasing function of ℓ, where ℓ is the threshold. Then $x(\lambda)$ must be a strictly decreasing function of λ. The set of passive states for λ is given by $[0, x(\lambda)]$. It follows that the set of passive states monotonically decreases to ϕ as $\lambda \to \infty$. This implies Whittle-like indexability.

4.1. Proposed policy

We propose the following heuristic policy inspired by [36]:

Our decision epochs are the time instances when there is some change in the system, i.e., either an arrival or a departure occurs. For each server $j \in S$, the index computed for each file type connected to this server is known. The file type which has the smallest index is chosen and the server serves it at full rate. Each time there is either an arrival into a file type or there is a job completion, the new indices are sent to the server which then decides which queue to serve.

4.2. Computation of the Whittle-like index

The Whittle-like index $\lambda(x)$ when the number of jobs is x, is computed by the following linear system of equations and an iterative scheme. The scheme is for a fixed x, so we suppress the x-dependence of $\lambda(x)$ and denote it simply as λ. We denote the iterates as $\{\lambda^n\}$, which use the solution of the linear system as a subroutine at each update. We have used $V_\lambda(\cdot), \beta(\lambda)$ in place of $V(\cdot), \beta$ to make the λ-dependence ($\lambda(x)$ -

dependence to be precise) of V, β explicit as required by this part of analysis. The scheme is as follows.

For each $n \geq 0$, do the following

1. Given the current iterate λ^n, to solve

$$V_{\lambda^n}(y) = f(y) + \lambda^n + \sum_{y'} p_1(y' \mid y) V_{\lambda^n}(y) - \beta(\lambda^n), \text{ if } y \leq x, \tag{15}$$

$$V_{\lambda^n}(y) = f(y) + \sum_{y'} p_2(y' \mid y) V_{\lambda^n}(y) - \beta(\lambda^n), \text{ if } y > x, \tag{16}$$

$$V_{\lambda^n}(0) = 0. \tag{17}$$

This solves the Poisson equation for the given threshold policy with threshold x and the disutility parameter λ fixed at λ^n.

2. Update λ^n to λ^{n+1} according to the iteration

$$\lambda^{n+1} = \lambda^n + \eta(\sum_{x'} p_2(x' \mid x) V_{\lambda^n}(x') - \sum_{x'} p_1(x' \mid x) V_{\lambda^n}(x') - \lambda^n(x)). \tag{18}$$

Here η is a small step size (taken to be 0.01). This iteration makes an incremental correction to λ^n in the direction of decreasing the discrepancy between the returns for active and passive actions.

We analyze this scheme under the simplifying assumption that f is strictly convex. The proof of convexity of V shows that V will also be strictly convex, hence $x \mapsto V(x+z) - V(x)$ for $z > 0$ strictly increasing. In particular, the argument of Lemma 3.3 then shows that the Whittle-like index is uniquely defined for each x.

Theorem 4.2. *For each fixed x, $\lambda^n(x)$ converges to an $O(\eta)$ neighborhood of the Whittle-like index as $n \to \infty$.*

Proof. Since η is small, we can view (18) as an Euler scheme for approximate solution by discretization [13] of the ODE

$$\dot{\lambda}(t) = F(\lambda(t)) - \lambda(t),$$

where

$$F(\lambda)(y) := \sum_{y'} p_2(y' \mid y) V_{\lambda^n}(y) - \sum_{y'} p_1(y' \mid y) V_{\lambda^n}(y).$$

Equations (15)-(17) constitute a linear system of equations, hence $V_\lambda(x), \beta(\lambda)$ are linear in λ. Thus the above ODE is well-posed. Furthermore, this is a scalar ODE with equilibrium given by that value of λ for which

$$\lambda = \sum_{x'} p_2(x' \mid x) V_{\lambda^n}(x') - \sum_{x'} p_1(x' \mid x) V_{\lambda^n}(x).$$

i.e., the Whittle-like index at state x, unique as observed above. Above this value, the ODE

has a negative drift and below it, a positive drift. Thus it is a stable ODE (i.e., the trajectories do not blow up). As a stable scalar ODE, it converges to its equilibrium. Interpolate the iterates as $\bar{\lambda}(t) = \lambda(n)$ for $t = n\eta$ with linear interpolation on $[n\eta, (n+1)\eta]$, $\forall\, n$. Define $[t] := \sup\{n\eta : n\eta \le t < (n+1)\eta\}$. Then we have

$$\dot{\bar{\lambda}}(t) = F(\bar{\lambda}(t)) - \bar{\lambda}(t) + \upsilon(t), \quad a.e.,$$

where $\upsilon(t) := F(\bar{\lambda}([t])) - F(\bar{\lambda}(t))$. It is easy to check that given the boundedness of trajectories and linearity of F, $|\upsilon|$ is $O(\eta)$. The convergence of iteration (18) to a neighborhood of this equilibrium then follows from Theorem 1 of Hirsch [15] by standard arguments.

Remarks:

1. We have not imposed any restriction on the sign of $\lambda(x)$ though it is known a priori, because the stable dynamics above with a unique equilibrium automatically picks up the right $\lambda(x)$. The linear system (15), (16) and (17) is solved as a subroutine by a suitable linear system solver.
2. The convergence to a neighborhood of the desired limit rather than to the limit itself is due to the fact that we are using a constant stepsize, leading to non-vanishing discretization errors. See Butcher [13] for a detailed error analysis of Euler method in a much more general set-up. We can get exact convergence by using slowly decreasing stepsizes, i.e., stepsizes satisfying $a = a_n \to 0$ slowly enough so that $\sum_n a_n = \infty$. But a constant stepsize as above offers the advantage that it allows the iterates to track a slowly varying environment.

5. Simulations

In this section, we report simulations to compare the performance of the proposed Whittle-like index policy with other natural heuristic policies given as follows:

- **Balanced Fair Allocation:** This is a centralized scheme for allocating server capacities. See Bonald and Proutiere [7] for more details.
- **Uniform Allocation:** At each instant in time, each server splits its rate equally among all the files that it contains.
- **Weighted Allocation:** The server rates are split according to prescribed weights proportional to the arrival rates into the different file types.
- **Random Allocation:** The decision epochs are the same. At each instant, for each server, a file type is chosen randomly and the server serves this at full capacity.
- **Max-Weight Allocation:** Each server serves at full capacity that file type which has the most number of jobs at any given instant.

We first compare the Whittle-like policy with the unconstrained optimal scheme and the balanced fairness allocation scheme for the network shown in Figure 3. The results of the simulations are shown in Figure 4. The values of parameters are as follows: $\Lambda^1 = 0.2; f^1(x) = 13x; \Lambda^2 = 0.1; f^2(x) = 10x; \mu_1 = 0.2; \mu_2 = 0.2$. The unconstrained optimal value for the network in figure 3 is computed using the following set of iterative equations

$$V_{n+1}(x_1, x_2) = f^1(x_1) + f^2(x_2) - V_n(0,0) + \min_{i \in \{1,2,3,4\}} (\mathbb{E}^i[V_n(\cdot) \mid x_1, x_2]),$$
$$u_{n+1} = \underset{i}{argmin}(\mathbb{E}^i[V_n(\cdot) \mid x_1, x_2]).$$

Here, the control u denotes the following: $u = 1$ denotes server $1,2$ serve file 1; $u = 2$ denotes server 1 serves file 1 and server 2 serves file 2; $u = 3$ denotes server 1 serves file 2 and server 2 serves file 1; $u = 4$ denotes server $1,2$ serve file 2. The shorthand notation $\mathbb{E}^i[\cdot \mid \cdot]$ denotes the conditional expectation with respect to the transition probability under control $i \in \{1,2,3,4\}$.

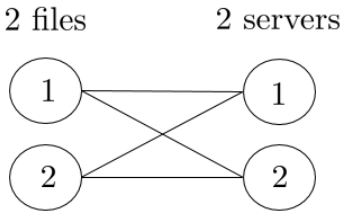

Figure 3. The Network that we use for simulations.

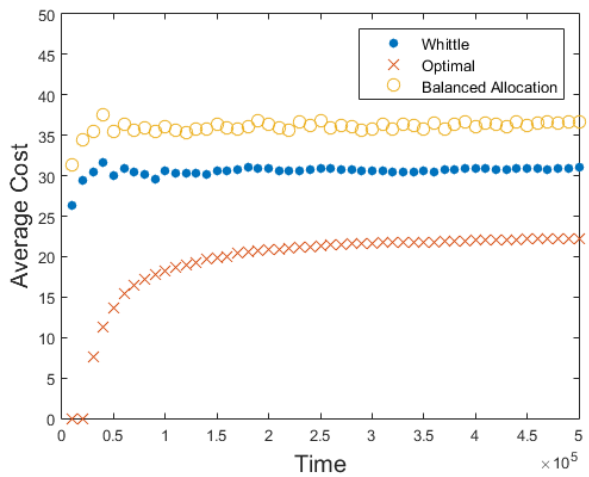

Figure 4. Comparison of Whittle-like policy, True optimal, and Balanced Fairness scheme.

The second network that we consider is shown in Figure 5. The parameters in this simulation are as follows: $\Lambda^1 = 0.1, f^1(x) = 10x, \Lambda^2 = 0.2, f^2(x) = 20x, \Lambda^3 = 0.1, f^3(x) = 10x, \mu_1 = 0.2, \mu_2 = 0.3.$

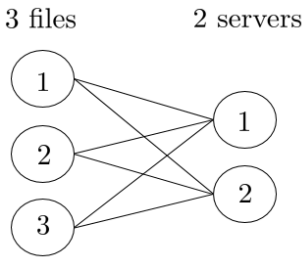

Figure 5. The Network that we use for simulations.

Figure 6 shows the Whittle-like indices assigned by file types 1 and 2 to server and Figure 7 shows the Whittle-like indices assigned by file type 1 to the two servers.

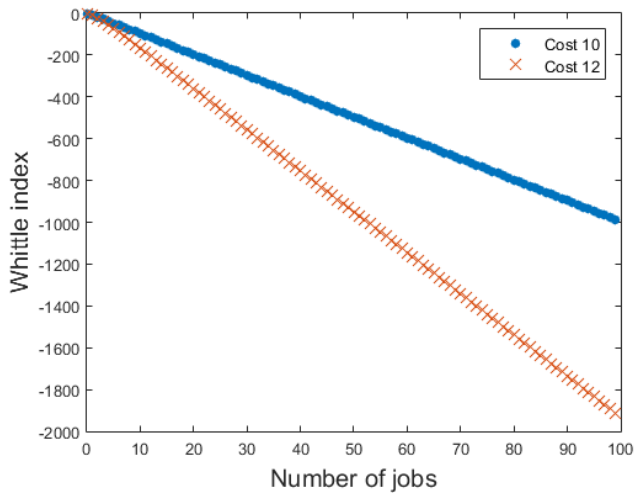

Figure 6. Whittle-like index assigned by different files to the same server.

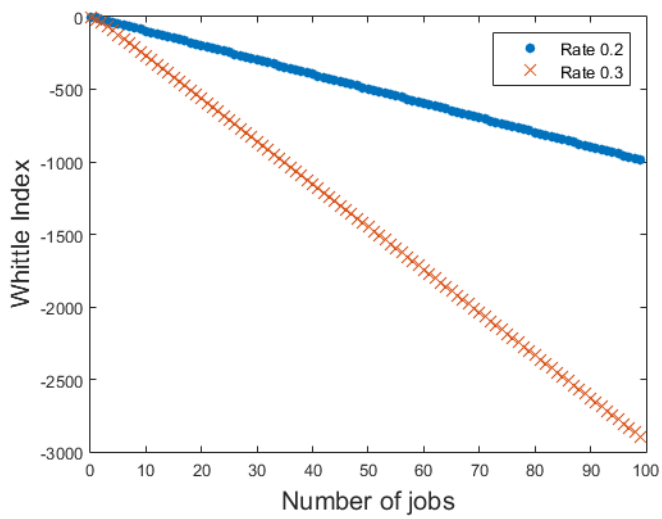

Figure 7. Whittle-like index assigned by the same file to different servers.

Figures 8 and 9 compare performance of the various methods that were described earlier in this section[4]. We can see that the Whittle-like index based policy performs better than the other methods of server allocation.

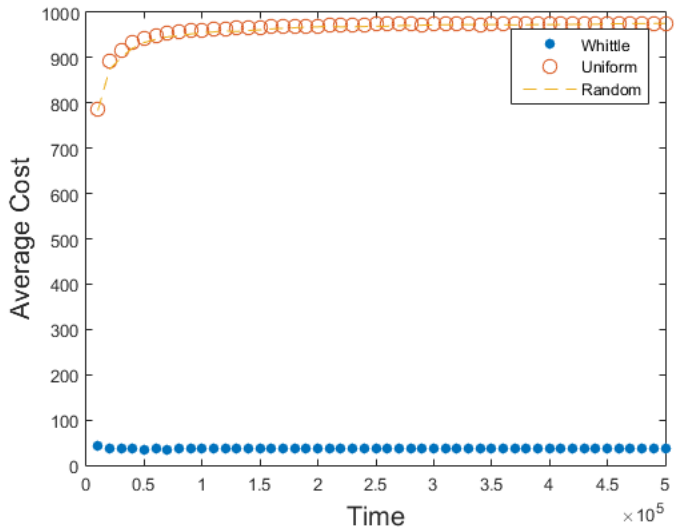

Figure 8. Comparison of Whittle-like policy with Uniform and Random policies.

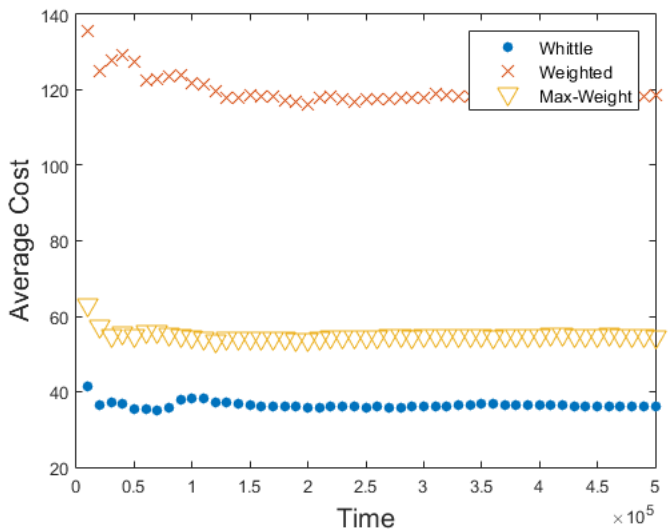

Figure 9. Comparison of Whittle-like policy with Weighted and Max Weight policies.

Figure 10 shows simulation results for the model with 10 file types and 10 servers such that file type i is stored in servers $i, i+1(\text{mod}10)$. $\Lambda^i = 0.2, f^i(x) = 15x, \mu_i = 0.2$ for

[4]We have separated these figures for better comparison. This is because the performance of the uniform and random allocation is much worse than the other policies.

$i = 1, 4, 7, 10$. $\Lambda^i = 0.3$, $f^i(x) = 20x$, $\mu_i = 0.3$ for $i = 2, 5, 8$. $\Lambda^i = 0.1$, $f^i(x) = 10x$, $\mu_i = 0.2$ for $i = 3, 6, 9$. Again, the Whittle-like policy shows a clear advantage.

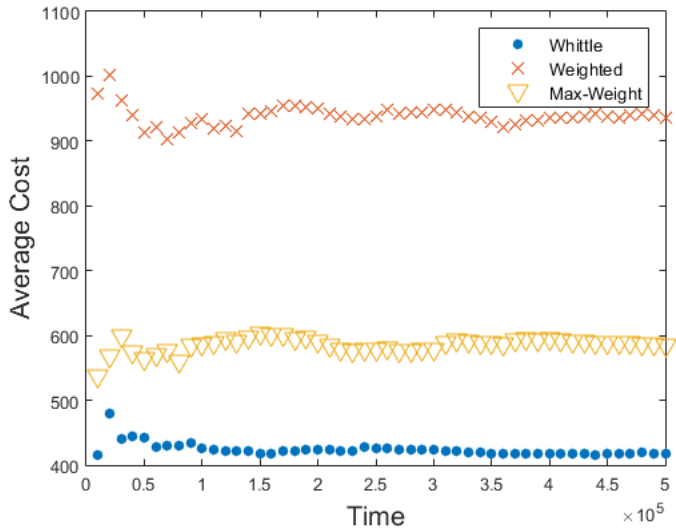

Figure 10. Comparison of Whittle-like policy with Weighted and Max Weight policies (10 file types and 10 servers).

6. Conclusions and Future Work

We have proved Whittle-like indexability of the server allocation problem in resource pooling networks. The allocation of servers using the Whittle-like scheme can be implemented in a distributed manner. The next step would be to extend this work to more general file types and possibly more complicated network topologies.

Acknowledgment

The work of Vivek S. Borkar was supported in part by a J. C. Bose Fellowship and a grant for 'Approximation of High Dimensional Optimization and Control Problems' from the Department of Science and Technology, Government of India.

References

[1] Agarwal, M., Borkar, V. S., & Karandikar, A. (2008). Structural properties of optimal transmission policies over a randomly varying channel. *IEEE Transactions on Automatic Control*, 53, 1476-1491.

[2] Archibald, T. W., Black, D. P., & Glazebrook, K. D. (2009). Indexability and index heuristics for a simple class of inventory routing problems. *Operations Research*, 57, 314-326, 2009.

[3] Avrachenkov, K., & Borkar, V. S. (2018). Whittle index policy for crawling ephemeral content. *IEEE Transactions on Control of Network Systems*, 5, 446-455.

[4] Avrachenkov, K., Borkar, V. S., & Pattathil, S. (2017). Controlling G-AIMD using Index Policy. The *56th IEEE Conference on Decision and Control,* Melbourne, December 12-15.

[5] Bertsekas, D. P. (1999). Nonlinear Programming. Belmont, Athena scientific.

[6] Bonald, T., & Comte, C. (2017). Balanced fair resource sharing in computer clusters. *Performance Evaluation,* 117, 70-83.

[7] Bonald, T., & Proutiere, A. (2003). Insensitive bandwidth sharing in data networks. *Queueing Systems,* 44, 69-100.

[8] Borkar V. S. (1991). *Topics in Controlled Markov Chains*, Pitman Research Notes in Mathematics, No. 240, Longman Scientific and Technical, Harlow, UK.

[9] Borkar V. S. (2002). Convex analytic methods in Markov decision processes'. *In Feinberg E. A., Shwartz A. (eds), Handbook of Markov Decision Processes*, Springer, Boston, MA.

[10] Borkar, V. S., Kasbekar, G. S., Pattathil, S., & Shetty, P. Y. (2018). Opportunistic scheduling in restless bandits. *IEEE Transactions on Control of Network Systems,* 5, 1952-1961.

[11] Borkar, V. S. , Ravikumar, K., & Saboo, K. (2017). An index policy for dynamic pricing in cloud computing under price commitments. *Applicationes Mathematicae*, 44, 215-245.

[12] Borkar, V. S., & Pattathil, S. (2017). Whittle indexability in egalitarian processor sharing systems. *Annals of Operations Research*, (available online at https://link.springer.com/content/pdf/10.1007/s10479-017-2622-0.pdf).

[13] Butcher, J. C. (2016). Numerical Methods for Ordinary Differential Equations (third edition), John Wiley & Sons, New York.

[14] Guo, X., & Hérnandez-Lerma, O. (2009). Continuous-Time Markov Decision Processes: Theory and Applications, Springer Verlag, Berlin-Heidelberg.

[15] Hirsch, M. W. (1989). Convergent activation dynamics in continuous time networks. *Neural Networks*, 2, 331-349.

[16] Jacko, P. (2010). Dynamic Priority Allocation in Restless Bandit Models, Lambert Academic Publishing.

[17] Kurose, J. F., & Ross, K. W. (2012). Computer Networking: A Top-Down Approach, Addison Wesley, Sixth Edition.

[18] Larranaga, M. , Ayesta, U., & Verloop, I. M. (2016). Dynamic control of birth-and-death restless bandits: Application to resource-allocation problems. *IEEE/ACM Transactions on Networking*, 24, 3812-3825.

[19] Leconte, M., Lelarge, M., & Massoulie, L. (2012). Bipartite graph structures for efficient balancing of heterogeneous loads. *Proceedings of ACM Sigmetrics/ Performance*, 41-52.

[20] Leconte, M., Lelarge, M., & Massoulie, L. (2015). Designing adaptive replication schemes in distributed content delivery networks. *Proceedings 27th IEEE International Teletraffic Congress (ITC 27),* 28-36.

[21] Leighton, T. (2009). Improving Performance on the Internet, *Communications of the ACM*, 52, 44-51.

[22] Milgrom, P., & Segal, I. (2002). Envelope theorems for arbitrary choice sets. *Econometrica*, 70, 583-601.

[23] Moharir, S., Ghaderi, J., Sanghavi, S., & Shakkottai, S. (2014). Serving content with unknown demand: the high-dimensional regime, *Proceedings of ACM Sigmetrics/ Performance,* 435-447.

[24] Ninõ-Mora, J. (2012). Admission and routing of soft real-time jobs to multi-clusters: Design and comparison of index policies. *Computers & Operations Research*, 39, 3431-3444.

[25] Nino-Mora, J., & Villar, S. S. (2011). Sensor scheduling for hunting elusive hiding targets via Whittle's restless bandit index policy. *The 5th International Conference on Network Games, Control and Optimization (NetGCooP 2011)*, Paris, Oct. 12-14.

[26] Ny, J. L., Dahleh, M., & Feron, E. (2008). Multi-UAV dynamic routing with partial observations using restless bandit allocation indices." *Proceedings of the American Control Conference, Seattle, June 11-13*, 4220-4225.

[27] Papadimitriou, C. H., & Tsitsiklis, J. N. (1999). The complexity of optimal queuing network control. *Mathematics of Operations Research*, 24, 293-305.

[28] Ruiz-Hernandez, D. (2008). Indexable Restless Bandits, VDM Verlag.

[29] Shah, V. (2015). *Centralized content delivery infrastructure exploiting resource pools: Performance models and asymptotics.* Ph.D. Dissertation, Department of Electrical and Computer Engineering, University of Texas at Austin, available at *https://repositories.lib.utexas.edu/handle/2152/31419*.

[30] Shah, V., & de Veciana, G. (2014). Performance evaluation and asymptotics for content delivery networks. *Proceedings IEEE INFOCOM, Atlanta, May 1-4,* 2607-2615.

[31] Shah, V., & de Veciana, G. (2015). High-performance centralized content delivery infrastructure: models and asymptotics. *IEEE / ACM Transactions on Networking,* 23, 1674-1687.

[32] Shah, V., & de Veciana, D. (2016). Impact of fairness and heterogeneity on delays in large-scale centralized content delivery systems. *Queuing Systems*, 83, 361-397.

[33] Tsitsiklis, J., & Xu, K. (2017). Flexible queuing architectures. *Operations Research,* 65, 1398-1413.

[34] Weber, R. R., & Weiss, G. (1990). On an index policy for restless bandits. *Journal of Applied Probability*, 27, 637-648.

[35] West, D. (2000). Introduction to Graph Theory. Prentice Hall, second edition.

[36] Whittle, P. (1988). Restless bandits: activity allocation in a changing world. *Journal of Applied Probability Vol. 25: A Celebration of Applied Probability*, 287-298.

[37] Zhou, Y., Fu, T., & Chiu, D. (2015). A unifying model and analysis of P2P VoD replication and scheduling. *IEEE / ACM Transactions on Networking,* 23, 1163-1175.