Deadline Scheduling as Restless Bandits

Zhe Yu[†], Yunjian Xu[‡], and Lang Tong[†]

***Abstract*— The problem of stochastic deadline scheduling is considered. A constrained Markov decision process model is introduced in which jobs arrive randomly at a service center with stochastic job sizes, rewards, and completion deadlines. The service provider faces random processing costs, convex non-completion penalties, and a capacity constraint that limits the simultaneous processing of jobs. Formulated as a restless multi-armed bandit problem, the stochastic deadline scheduling problem is shown to be indexable. A closed-form expression of the Whittle's index is obtained for the case when the processing costs are constant. An upper bound on the gap-to-optimality of the Whittle's index policy is established, and the bound is shown to converge to zero as the job arrival rate and the the number of simultaneously available processors increase simultaneously.**

I. Introduction

The deadline scheduling problem, in its most generic setting, is the scheduling of jobs with different workloads and deadlines for completion. Typically, there are not enough servers to satisfy all the demand; the cost of processing may vary with time, and unfinished jobs by their deadlines incur a penalty.

In this paper, we are interested in the *stochastic deadline scheduling problem* where key parameters of the problem such as job arrivals, workloads, deadlines of completion, and processing costs are stochastic. In particular, we consider the problem of maximizing the average or discounted rewards over a finite or an infinite scheduling horizon.

A prototype application of such a problem is the charging of electric vehicles (EVs) at a charging service center [1], [2]. In such applications, EVs arrive at the service center randomly, each with its own charging demand and deadline for completion. The charging cost depends on the cost of electricity at the time of charging, and a penalty is imposed when the service provider is unable to fulfill the request. Similar applications include the scheduling of jobs at data centers [3], internet streaming [4], hospitals [5], and customer service centers [6].

The stochastic deadline scheduling problem is an instance of stochastic dynamic programming, for which obtaining the optimal solution is fundamentally intractable. However, practical applications often mandate that the processing schedule be constructed in real time. This means that, in general, one may have to sacrifice optimality in favor of approximate solutions that are scalable algorithmically and have performance close to that of the optimal scheduler. An important class of such algorithms is the so-called *index policies* [7] that attach an index to each unfinished job, rank them according to their indices, and assign available processors to the top ranked jobs. The index of each job is determined by the state of itself and is independent of the other jobs. Such policies offer scalable solutions if the ranking algorithm aligns with the objective of the scheduler and can be computed online.

Z. Yu[†] and L. Tong[†] are with the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY 14853, USA. Y. Xu[‡] is with the School of Engineering Systems and Design Pillar, Singapore University of Technology and Design, Singapore, 487372. Email: {zy73,lt35}@cornell.edu, yunjian_xu@sutd.edu.sg.
This work is supported in part by the National Science Foundation under Grant CNS-1248079.

A. Summary of results

In this paper, we formulate the stochastic deadline scheduling problem as a restless multi-armed bandit (RMAB) problem [8]. We examine the indexability of the problem and the performance of the Whittle's index policy. To this end, we first introduce a constrained Markov decision process (MDP) model with the objective of maximizing expected (discounted) profit subject to a constraint on the maximum number of jobs that can be processed simultaneously. The constructed MDP model captures the randomness in job arrivals, job sizes, deadlines, and processing costs.

Next, we reformulate the MDP as an RMAB problem with simultaneous plays [8] and establish the indexability of the formulated RMAB. The special structure of the problem, in particular, the pre-determined deadline and workload at the time of arrival, simplifies the computation of the Whittle's index. For the case with constant processing cost, we derive the Whittle's indexes in closed form.

The Whittle's index policy, unfortunately, is not optimal in general when the constraint on the number of processors that can be activated is strict. We obtain a bound on the gap-to-optimality for the Whittle's index policy and show that the gap approaches zero exponentially as the number of available processors and the job arrival rate increase simultaneously. This result provides a theoretical justification for using the Whittle's index policy as a baseline approach in applications where job arrivals are in the light traffic regime.

B. Related Work

The classical deadline scheduling problem is first considered by Liu and Layland [9] in a deterministic setting. For the single processor case, the results are quite complete. When all jobs can be finished on time, simple index algorithms (with linear complexity) such as the earliest deadline first (EDF) [9], [10] and the least laxity first (LLF) [11] achieve the same performance as the optimal off-line algorithm in the deterministic setting. There is also a substantial literature on deadline scheduling with multiple processors (for a survey, see [12]). It is shown in [13] that optimal online scheduling policies do not exist in general for the worst case performance measure.

The literature on deadline scheduling in the stochastic settings is less extensive. Panwar, Towsley and Wolf in [14] and [15] made early contributions in establishing the optimality of EDF in minimizing the unfinished work when there is a single processor, and the jobs are non-preemptive. The performance of EDF is quantified in the heavy traffic regime using a diffusion model in [16], [17], [18].

The multiprocessor stochastic deadline scheduling problem is less understood, primarily because the stochastic dynamic programming for such problems are intractable to solve in practice. A particularly relevant class of applications is scheduling in wireless networks where job (packets) arrival is stochastic, and packets sometimes have deadlines for delivery. In [19], the author analyzed the performance of the EDF policy for packets delivery in tree networks. Related problems of scheduling packets with deadlines in ad hoc networks are studied in [23].

The work closest to ours is [20] where the deadline scheduling problem is formulated as an RMAB problem and the indexability is established. There are, however, several important differences between [20] and this work. First, in [20], the arrivals are periodic or simultaneous. In our work, the arrival is random. Second, the unit job length is assumed in [20] while in our work the job length is stochastic. Finally, there is no analytical study about the performance of the Whittle's index policy for deadline scheduling problems before.

II. Problem Formulation

In this section, we introduce the stochastic deadline scheduling problem as a constrained MDP followed by an RMAB formulation.

A. Stochastic Deadline Scheduling as a Constrained MDP

We begin with a set of nominal assumptions in setting up the MDP formulation:

A1. The time is slotted, indexed by t.

A2. There are M processors available at all times. Each processor can only work on one job in a time slot, and each job can receive services from only one processor. A processor can be switched from one job to another without incurring switching cost.

A3. If a processor works on a job in time slot t, it receives a unit payment and incurs a time varying cost $c[t]$. Here we assume that $c[t]$ is a stationary Markov process with transition probability matrix $P=[P_{i,j}]$.

A4. If a job is not completed by its deadline, a penalty defined by a convex function of the amount of unfinished job is imposed on the scheduler at the end of the deadline.

A5. A newly arrived job will be randomly assigned to a position in a queue of size N waiting for processing. Here we assume that $N \gg 1$ and ignore the cases when there is no holding space for a newly arrived job.

A6. A job assigned to the ith position of the queue at time t reveals B_i—the total amount of job to be completed—and T_i—the deadline for completion. At $t+T_i$, the job is removed from the queue, regardless whether the job is completed. When the ith position is available, with probability $Q(T,B)$ a new job with deadline T and workload B arrives. With probability $Q(0,0)$, the position remains empty. The jobs arrived at different positions are statistically independent and identically distributed.

We now define the constrained MDP by defining the state, the action of the scheduler, reward, the state evolution, constraints, and the decision policy.

1) State Space: Consider first the state of the ith position in the queue. Let $T_i[t] \triangleq d_i - t$ be the lead time to deadline d_i, and $B_i[t]$ is the remaining job length, as illustrated in Figure 1.

The state of the ith position in the queue is defined as

$$S_i[t] \triangleq \begin{cases} (0,0) & \text{if no job waits at the } i\text{th position,} \\ (T_i[t], B_i[t]) & \text{otherwise,} \end{cases}$$

The processing cost $c[t]$ is an exogenous finite state Markov chain with transition probability matrix $P=[P_{i,j}]$.

The state of the MDP is defined by the queue states and the processing cost $c[t]$ as $S[t] \triangleq (c[t], S_1[t], \cdots, S_N[t]) \in \mathcal{S}$ and $\mathcal{S}$ the state space.

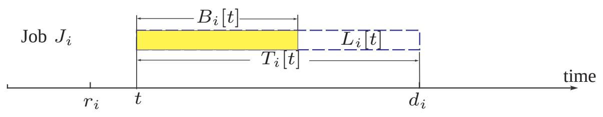

Fig. 1. An illustration for the position's state. r_i is the arrival time of a job at position i, d_i the deadline for completion, $B_i[t]$ the job length to be completed by d_i, $T_i[t]$ the lead time to deadline.

2) Action: The action of the scheduler in slot t is defined by the binary vector $\mathbf{a}[t] = (a_1[t], \cdots, a_N[t]) \in \{0,1\}^N$ where $a_i[t]=1$ means that a processor is assigned to work on the job at position i, for which the position is referred as *active*. The complement, $a_i[t]=0$, is when position i is *passive*, *i.e.*, no processor is assigned. For convenience, we allow a position without a job to be activated, in which case the processor assigned receives no reward and incurs no cost.

3) State Evolution: The evolution of the processing cost is according to the transition matrix P and independent of the actions taken by the scheduler.

The evolution of the queue state $S_i[t]$ depends on the scheduling action $a_i[t]$:

$$S_i[t+1] = \begin{cases} (T_i[t]-1, (B_i[t]-a_i[t])^+) & T_i[t] > 1, \\ (T,B) \text{ with prob. } Q(T,B) & T_i[t] \le 1. \end{cases} \tag{1}$$

where $b^+ = \max(b,0)$. Note that when $T_i[t] = 1$, the deadline is due in slot t and job in position i is removed. With probability $Q(T,B)$, some new job with lead time T and job size B arrives at the beginning of slot $t+1$.

4) Reward: For each job, the scheduler obtains one unit of reward if the job is processed for one time slot. At the job's deadline, *i.e.*, $T_i[t]=1$, the scheduler pays the penalty for the unfinished work. Let $F(B)$ be the convex penalty function of the amount B of the unfinished job, and $F(0)=0$. Denote the cost of processing at time t by $c[t]$. Thus the reward collected from job i at time t is given by

$$R_{a_i[t]}(S_i[t], c[t]) = \begin{cases} (1-c[t])a_i[t] & B_i[t] > 0, T_i[t] > 1 \\ (1-c[t])a_i[t] - F(B_i[t] - a_i[t]) & B_i[t] > 0, T_i[t] = 1 \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

5) Objective: Given the initial system state $S[0] = s$ and a policy π that maps each system state $S[t]$ to an action vector $\mathbf{a}[t]$, the expected discounted system reward is defined by

$$G_\pi(s) \triangleq \mathbb{E}_\pi \left(\sum_{t=0}^{\infty} \sum_{i=1}^{N} \beta^t R_{a_i[t]}(S_i[t], c[t]) \mid S[0] = s \right), \quad (3)$$

where $\mathbb{E}_\pi$ is the conditional expectation over the randomness in costs and job arrivals under a given scheduling policy π and $0 < \beta < 1$ the discount factor. The analysis can be extended to the average case [21].

6) Constrained MDP and Optimal Policies: We impose a constraint on the maximum number of processors that can be activated simultaneously. Specifically, $\sum_i^N a_i[t] \le M$. This constraint represents the processing capacity of the service provider. For the EV charging application, this assumption translates directly to the physical power limit imposed on the charging facility. Thus, the deadline scheduling problem can then be formulated as a constrained MDP.

$$G(s) = \sup_{\{\pi : \sum_i^N a_i^\pi[t] \le M, \forall t\}} G_\pi(s), \quad (4)$$

where $a_i^\pi[t]$ is the action sequence generated by policy π for position i. A policy π^* is optimal if $G_{\pi^*}(s) = G(s)$. Without loss of optimality, we will restrict our attention to stationary policies [22].

B. An RMAB Problem

Unfortunately, the MDP formulation does not result in a scalable optimal scheduling policy due to the fact that the state space grows exponentially with N.

Alternately, we seek to obtain an *index policy* [7] that scales linearly with N. We identify each position in the queue as an arm and formulate (4) as an RMAB problem. To this end, "playing" an arm is equivalent to assigning a processor to process the job (if there is one) at a location in the queue. The resulting multi-armed bandit problem is restless because the state of position i—the ith arm—evolves regardless whether arm i is active or passive. Note, however, that the evolution of the state of an arm is deterministic in nature.

A complication of casting (4) as an RMAB problem comes from the inequality constraint on the maximum number of simultaneously activated positions; the standard RMAB formulation imposes an equality constraint on the number of arms that can be activated. This can be circumvented by introducing M dummy arms and requiring that exactly M arms must be activated in each time slot. Specifically, each dummy arm always accrues a zero reward, and the state stays at $S_i = (0, 0)$. With the addition of dummy arms, the constraint on the maximum number of arms that can be activated in the original MDP problem can be transformed to an equality constraint. The reformulated RMAB problem has $N + M$ arms. We let $\{1, \cdots, N\}$ be the set of regular arms that generate reward (penalty) and $\{N + 1, \cdots, N + M\}$ be the set of dummy arms.

We define the extended state of each arm as $\tilde{S}_i[t] \triangleq (S_i[t], c[t])$, and denote the extended state space as $\mathscr{S}_i \triangleq \mathcal{S}_i \times \mathcal{S}_c$. The state transition of each arm and the associated rewards are inherited from (1-2) of the original MDP. The corresponding RMAB problem is defined by

$$\begin{array}{ll} \sup_\pi & \mathbb{E}_\pi \left\{ \sum_{t=0}^{\infty} \sum_{i=1}^{N+M} \beta^t R_{a_i[t]}(\tilde{S}_i[t]) \mid \tilde{S}_i[0] \right\} \\ \text{s.t.} & \sum_{i=1}^{N+M} a_i[t] = M, \quad \forall t. \end{array} \quad (5)$$

In (5), the arms are coupled by the processing cost and are not independent.

III. Whittle's Index Policy

To pursue the deadline scheduling problem as an RMAB, we need to establish the indexability of the RMAB.

A. Indexability

Consider the *ν-subsidized* single arm reward maximization problem [8] that looks for a policy π to activate/deactivate the arm to maximize the discounted accumulative reward:

$$V_i^\nu(s) = \sup_\pi \mathbb{E}_\pi \left(\sum_{t=0}^{\infty} \beta^t R^\nu_{a_i[t]}(\tilde{S}_i[t]) \mid \tilde{S}_i[0] = s \right), \quad (6)$$

where the subsidized reward is modified single arm reward (2) given by

$$R^\nu_{a_i[t]}(\tilde{S}_i[t]) = R_{a_i[t]}(\tilde{S}_i[t]) + \nu \mathbb{1}(a_i[t] = 0),$$

where $\mathbb{1}(\cdot)$ is the indicator function. In words, the *ν-subsidized* problem is a modification of the reward such that the scheduler receives a subsidy ν whenever the arm is passive.

Let $\mathcal{L}_a$ be an operator on V_i^ν defined by

$$(\mathcal{L}_a V_i^\nu)(s) \triangleq \mathbb{E}\left(V_i^\nu(\tilde{S}_i[t+1]) \middle| \tilde{S}_i[t] = s, a_i[t] = a \right).$$

The maximum discounted reward $V_i^\nu(\cdot)$ in (6) is determined by the Bellman equation

$$V_i^\nu(s) = \max_{a \in \{0,1\}} \left\{ R_a^\nu(s) + \beta(\mathcal{L}_a V_i^\nu)(s) \right\}. \quad (7)$$

Let $\mathscr{S}_i$ be the space of extended state of arm i and $\mathscr{S}_i(\nu)$ the set of states under which it is optimal to take the passive action in the ν-subsidy problem. The *indexability of the RMAB* is defined by the monotonicity of $\mathscr{S}_i(\nu)$ as subsidy level ν increases:

Definition 1 (Indexability [8]): Arm i is indexable if the set $\mathscr{S}_i(\nu)$ increases monotonically from $\emptyset$ to $\mathscr{S}_i$ as ν increases from $-\infty$ to $+\infty$. The RMAB problem is indexable if all arms are indexable.

We establish the indexability for the stochastic deadline scheduling problem.

Lemma 1 (Indexability): Each arm is indexable, and the RMAB problem (5) is indexable.

B. *Whittle's Index Policy*

Given the definition of indexability, the Whittle's index is defined as follows.

Definition 2 (Whittle's index [8]): If arm i is indexable, its Whittle's index $\nu_i(s)$ of state s is the infimum of the subsidy ν under which the passive action is optimal at state s, *i.e.*,

$$\nu_i(s) \triangleq \inf_\nu\{\nu : R_0(s) + \nu + \beta(\mathcal{L}_0 V_i^\nu)(s) \geq R_1(s) + \beta(\mathcal{L}_1 V_i^\nu)(s)\}.$$

Thus if arm i is indexable, any $\nu < \nu_i(s)$ makes activating arm i optimal. Likewise, any $\nu \geq \nu_i(s)$ makes it optimal to deactivate arm i.

We can compute the Whittle's index using a parametric programming method [23]. The special structure of the deadline problem, however, allows us to have a closed-form solution when the processing cost is constant.

Lemma 2: If $c[t] = c_0$ for all t, the Whittle's index of a regular arm $i \in \{1, \cdots, N\}$ is given by

$$\nu_i(T, B, c_0) = \begin{cases} 0 & \text{if } B = 0, \\ 1 - c_0 & \text{if } 1 \leq B \leq T-1, \\ 1 - c_0 + \beta^{T-1}[F(B-T+1) - F(B-T)] & \text{if } T \leq B. \end{cases} \quad (8)$$

The Whittle's index of a dummy arm is zero.

$$\nu_i(0, 0, c_0) = 0, \quad i \in \{N+1, \cdots, N+M\}.$$

In (8), when it is feasible to finish job i's request (*i.e.* its lead time is no less than its remaining processing time), job i's Whittle's index is simply the (per-unit) processing profit $1 - c_0$. When a non-completion penalty is inevitable, the index takes into account both the processing profit and the non-completion penalty. We note that the Whittle's index gives higher priority to jobs with less laxity. Here, the laxity of job i is defined as $L_i[t] \triangleq T_i[t] - B_i[t]$ (cf. Fig. 1).

Given the definition of Whittle's index, the Whittle's index policy for the deadline scheduling problem is stated as follows.

Definition 3 (Whittle's index policy): For the RMAB problem defined in (5), the Whittle's index policy sorts all arms by their Whittle's indices in a descending order and activates the first M arms.

Since the states of jobs and processing cost are finite, the Whittle's index can be computed off-line. In real-time scheduling, at the beginning of each time slot, the scheduler looks up the indices for each job and processes the ones with highest indices. When there is a tie, the scheduler breaks the tie randomly with a uniform distribution.

IV. Asymptotic optimality

We note that the Whittle's index policy is not optimal in general. Indeed, counterexamples show that there does not exist an optimal *index policy* that schedules jobs according to the order of jobs' indices that are computed based on jobs' current states [24]. In the following theorem, for the Whittle's index policy, we establish an upper bound on the gap-to-optimality as a function of M, the maximum number of available processors.

Theorem 1: Let $G(s)$ be the value function achieved by the optimal scheduler defined in (4) and $G_{\text{RMAB}}(s)$ be that by the Whittle's index policy, respectively. We have

$$G(s) - G_{\text{RMAB}}(s) \leq \frac{C}{1-\beta}\mathbb{E}[I[t]|I[t] > M]Pr(I[t] > M), \quad (9)$$

where $I[t]$ is the number of jobs ever in the queue within time $[t - \bar{T} + 1, t]$, $\bar{T}$ the maximum lead time of jobs, and C a constant determined by the processing cost and the penalty of non-completion.

Proof: The proof can be found in Appendix D of [24]. ■

When the traffic is heavy and the processing limit gets tighter, the gap-to-optimality is bounded by the event of the arrival exceeding the processing capacity. In finance area, the conditional expectation in the right hand side (RHS) of (9) is connected to the conditional value at risk (CVaR) [25]. CVaR measures the expected losses at some risk level and is extremely important in the risk management.

We now apply Theorem 1 to gain insights by considering special distributions (on the arrival process) of practical significance.

Proposition 1:
1. Suppose $I(t)$ follows a Poisson distribution with mean λ, then
$$G(s) - G_{\text{RMAB}}(s) \leq \frac{C}{1-\beta}\frac{\lambda^{M+1}e^{-\lambda}(M+1)}{(M+1-\lambda)M!}. \quad (10)$$
Specially, if $\lambda = M^\theta, \theta < 1$, the right hand side of (10) decreases supper exponentially to zero as M increases.
2. Suppose $I(t)$ follows a Binomial distribution with mean $Np < M$, then
$$G(s) - G_{\text{RMAB}}(s) \leq \frac{C}{1-\beta}\frac{N!M^2p^M(1-p)^{N-M+1}}{M!(N-M)!(M-Np)}. \quad (11)$$
Specially, if the processing capacity grows faster than the average arrival rate, *i.e.*, $M = \alpha N$ for $\alpha \in (p, 1)$, then the right hand side (RHS) of (11) decreases sub-exponentially to zero as M increases.

Proof: The proof can be found in Appendix E of [24]. ■

We note that Proposition 1 characterizes the asymptotic optimality of Whittle's index policy when the arrival of jobs and the processing capacity grow simultaneously while the overall system remains stable.

V. Numerical Results

In this section, results of numerical experiments are presented to compare the performance of the Whittle's index policy with other simple heuristic (index) policies, *i.e.*, EDF (earliest deadline first) [9] and LLF (least laxity first) [10].

If feasible, EDF processes M jobs with the earliest deadlines, and LLF processes M jobs with the least laxity. Both policies will fully utilize the processing capacity and activate M jobs as long as there are at least M unfinished jobs in the system. The Whittle's index policy, on the other hand, ranks

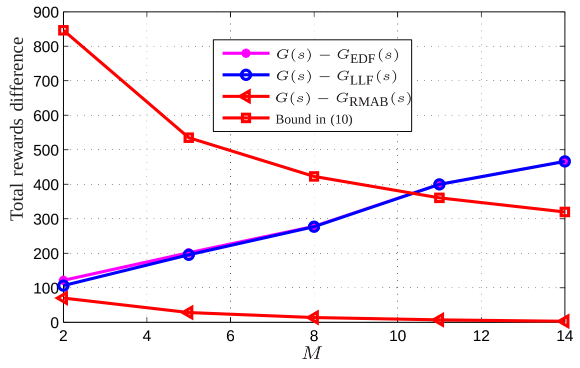

Fig. 2. Comparison of the total rewards achieved by three different index policies under dynamic processing cost: $Q(0,0) = 0.3$, $\bar{T} = 12$, $\bar{B} = 9$, $\beta = 0.999$, $F(B) = 0.2B^2$, $\theta = 0.999$, $N = 1000$.

all positions by the Whittle's index and activates the first M arms, and may put some (regular) positions idle (deactivated) when the processing cost is high.

In Figure 2, simulation results are presented to compare the performance achieved by various heuristic policies and to validate the theoretic results established in Proposition 1. The arrival sequence within $\bar{T}$ time slots follows a Poisson process with mean $M^{0.999}$. We fix the queue size $N = 1000$ and vary the processing capacity M as a parameter. The dynamic cost evolves according to a Markovian model that is trained using real-time electricity price signals from the California Independent System Operator (CAISO) (cf. Sections III and V of [26]). Each time slot of the constructed Markov chain lasts for 1 hour and the entire simulation horizon lasts for 300 days (with 24×300 time slots).

The EDF and LLF policies do not take into account the dynamics of processing costs, and their gap-to-optimality increases as both the job arrival rate and processing capacity grow. On the other hand, the gap between the total rewards achieved by the Whittle's index policy and the optimal policy quickly decreases to zero as the system scales.

VI. Conclusion

We considered the problem of large scale deadline scheduling—a problem that has wide applications in call centers, cloud computing, and EV charging. In such settings, it is essential to develop efficient on-line scheduling algorithms. To this end, the index policy proposed in this paper is attractive for its implementation simplicity, versatility in incorporating various operation uncertainties, and asymptotic optimality.

References

[1] Z. Yu, Y. Xu, and L. Tong, "Large scale charging of electric vehicles: A multi-armed bandit approach," in *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2015, pp. 389–395.

[2] Z. Yu, S. Chen, and L. Tong, "An intelligent energy management system for large-scale charging of electric vehicles," *CSEE Journal of Power and Energy Systems*, vol. 2, no. 1, pp. 47–53, 2016.

[3] J. Vilaplana, F. Solsona, I. Teixidó, J. Mateo, F. Abella, and J. Rius, "A queuing theory model for cloud computing," *The Journal of Supercomputing*, vol. 69, no. 1, pp. 492–507, 2014.

[4] B. B. Chen and P. V.-B. Primet, "Scheduling deadline-constrained bulk data transfers to minimize network congestion," in *Seventh IEEE International Symposium on Cluster Computing and the Grid (CCGrid'07)*. IEEE, 2007, pp. 410–417.

[5] J. Błażewicz, "Selected topics in scheduling theory," *North-Holland Mathematics Studies*, vol. 132, pp. 1–59, 1987.

[6] J. Dai and S. He, "Queues in service systems: Customer abandonment and diffusion approximations," *Tutorials in Operations Research, INFORMS: Hanover, MD*, pp. 36–59, 2011.

[7] J. C. Gittins, "Bandit Processes and Dynamic Allocation Indices," *Journal of the Royal Statistical Society*, vol. 41, no. 2, pp. 148–177, 1979.

[8] P. Whittle, "Restless bandits: Activity allocation in a changing world," *Journal of applied probability*, pp. 287–298, 1988.

[9] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *Journal of ACM*, vol. 20, pp. 46–61, 1973.

[10] M. Dertouzos, "Control robotics: the procedural control of physical processes," in *Proceedings of International Federation for Information Processing Congress*, 1974, pp. 807–813.

[11] A. Mok, "Fundamental design problmes of distributed systems for the hard real-time environment," Ph.D. dissertation, MIT, 1983.

[12] R. I. Davis and A. Burns, "A survey of hard real-time scheduling for multiprocessor systems," *ACM Computing Surveys*, vol. 43, no. 4, 2011.

[13] M. L. Dertouzos and A. K. Mok, "Multiprocessor online scheduling of hard-real-time tasks," *IEEE Transactions on Software Engineering*, vol. 5, pp. 1497–1506, 1989.

[14] S. S. Panwar, D. Towsley, and J. K. Wolf, "Optimal Scheduling Policies for a class of Queues with Customer Deadlines to the Beginning of Service," *Journal of Association for Computing Machinery*, vol. 35, no. 4, pp. 832–844, October 1988.

[15] D. Towsley and S. Panwar, "On the Optimality of Minimum Laxity and Earliest Deadline Scheduling for Real-Time Multiprocessors," in *Proceedings of IEEE Euromicro 90' Workshop on Real-Time*, Jun. 1990, pp. 17–24.

[16] J. Lehoczky, "Real-time queueing theory," in *Proceedings of 17th IEEE Real-Time Systems Symposium*, Dec. 1996, pp. 186 –195.

[17] B. Doytchinov, J. Lehoczky, and S. Shreve, "Real-time queues in heavy traffic with earliest-deadline-first queue discipline," *Annals of Applied Probability*, vol. 11, no. 2, pp. 332–378, 2011.

[18] L. Kruk, J. Lehoczky, K. Ramanan, and S. Shreve, "Heavy traffic analysis for EDF queues with reneging," *Annals of Applied Probability*, vol. 21, no. 2, pp. 484–545, 2011.

[19] P. P. Bhattacharya, L. Tassiulas, and A. Ephremides, "Optimal scheduling with deadline constraints in tree networks," *IEEE Transactions on Automatic Control*, vol. 42, no. 12, pp. 1703–1705, 1997.

[20] V. Raghunathan, V. Borkar, M. Cao, and P. R. Kumar, "Index policies for real-time multicast scheduling for wireless broadcast systems," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*. IEEE, 2008.

[21] P. K. Dutta, "What do discounted optima converge to?: A theory of discount rate asymptotics in economic models," *Journal of Economic Theory*, vol. 55, no. 1, pp. 64–94, 1991.

[22] E. Altman, *Constrained Markov decision processes*. CRC Press, 1999, vol. 7.

[23] J. Niño-Mora, "Characterization and computation of restless bandit marginal productivity indices," in *Proceedings of the 2nd international conference on Performance evaluation methodologies and tools*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007, p. 74.

[24] Z. Yu, Y. Xu, and L. Tong, "Deadline Scheduling as Restless Bandits," 2016, available on arXiv.

[25] R. T. Rockafellar and S. Uryasev, "Optimization of conditional value-at-risk," *Journal of risk*, vol. 2, pp. 21–42, 2000.

[26] S. Kwon, Y. Xu, and N. Gautam, "Meeting inelastic demand in systems with storage and renewable sources," *IEEE Tranactions on Smart Grid*, 2015.