On the Optimal Scheduling of Independent, Symmetric and Time-Sensitive Tasks

Fabio Iannello, *Student Member, IEEE*, and Osvaldo Simeone, *Member, IEEE*

***Abstract*—Consider a discrete-time system in which a *centralized controller* (CC) is tasked with assigning at each time interval (or *slot*) K resources (or *servers*) to K out of $M \geq K$ *nodes*. A node can execute a *task* when assigned to a server . The tasks are independently generated at each node by stochastically symmetric and memoryless random processes and stored in a finite-capacity *task queue*. The tasks are *time-sensitive* since there is a non-zero probability, within each slot, that a task expires before being scheduled. The scheduling problem is tackled with the aim of maximizing the number of tasks completed over time (or the *task-throughput*) under the assumption that the CC has no direct access to the state of the task queues. The scheduling decisions at the CC are based on the outcomes of previous scheduling commands, and on the known statistical properties of the task generation and expiration processes.**

Based on a Markovian modeling of the task generation and expiration processes, the CC scheduling problem is formulated as a partially observable Markov decision process (POMDP) that can be cast into the framework of restless multi-armed bandit (RMAB) problems. When the task queues are of capacity one, the optimality of a myopic (or greedy) policy is proved. The settings in this technical note provide a rare example where a RMAB problem can be explicitly solved.

***Index Terms*—Communication networks, Markov processes, queueing systems, stochastic optimal control.**

I. Introduction

The problem of scheduling concurrent tasks under resource constraints finds applications in a variety of fields including communication networks [1], distributed computing [2] and virtual machine scenarios [3]. In this technical note we consider a specific instance of this general problem in which a *centralized controller* (CC) is tasked with assigning at each time interval (or *slot*) K resources, referred to as *servers*, to K out of $M \geq K$ *nodes* as shown in Fig. 1. A server can complete a single task per slot and it can be assigned to one node per time interval. The tasks are generated at the M nodes by stochastically symmetric, independent and memoryless random processes. The tasks are stored by each node in a finite-capacity *task queue*, and they are *time-sensitive* in the sense that at each slot there is a non-zero probability that a task expires before being successfully completed. It is assumed that the CC has no direct access to the node queues, and thus it is not fully informed of their actual states. For instance, partial information is relevant in systems such as wireless sensor networks in which the nodes are physically separated by the CC, and their queues (which, e.g., collect data packets or energy) are either not accessible by the CC or could be accessed at some additional cost. Instead, the scheduling decisions at the CC are based on the outcomes of previous scheduling commands, and on the statistical knowledge of the task generation and expiration processes. If a server is assigned to a node with an empty queue, it remains idle for the whole slot. The purpose here is thus to pair servers to nodes so as to maximize the average number of successfully completed tasks within either a finite or infinite number of slots (*horizon*), which we refer to as *task-throughput*, or simply throughput.

Manuscript received November 29, 2011; revised July 05, 2012; accepted March 26, 2013. Date of publication April 18, 2013; date of current version August 15, 2013. Recommended by Associate Editor H. S. Chang.

F. Iannello is with the Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milan, 20133 Italy and also with the Center for Wireless Communication and Signal Processing Research, New Jersey Institute of Technology, Newark, NJ, 07102-1982 USA (e-mail: iannello@elet.polimi.it; fabio.iannello@njit.edu; fabio.iannello@gmail.com).

O. Simeone is with the Center for Wireless Communication and Signal Processing Research, New Jersey Institute of Technology, University Heights, Newark, NJ, 07102-1982 USA (e-mail: osvaldo.simeone@njit.edu).

Digital Object Identifier 10.1109/TAC.2013.2258791

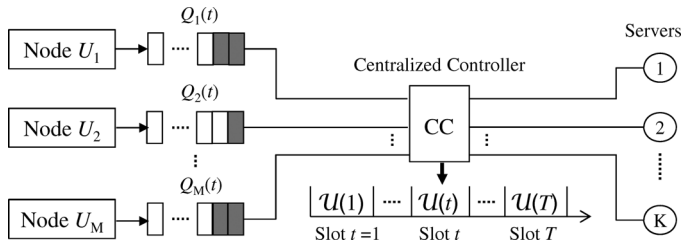

Fig. 1. Centralized controller (CC) assigns K resources (servers) to K out of M nodes to complete their tasks in each slot t in a limited resource scenario ($K \leq M$). The tasks of node U_i at slot t are stored in a task queue $Q_i(t)$.

A. Markov Formulation

We now introduce the stochastic model that describes the evolution of the task queues across slots. We consider task queues of capacity one (see [4] for a discussion with queues of arbitrary capacity), where $Q_i(t) \in \{0,1\}$ denotes the number of tasks in the queue of node U_i, for $i \in \{1,\ldots,M\}$. The stochastic evolution of queue $Q_i(t)$ is shown in Fig. 2 as a function of the scheduling decision $\mathcal{U}(t)$, which consists in the assignment at each slot t of the K servers to a subset $\mathcal{U}(t) \subseteq \{U_1,\ldots,U_M\}$ of K nodes, with $|\mathcal{U}(t)| = K$.

At each slot, node U_i can be either scheduled ($U_i \in \mathcal{U}(t)$) or not ($U_i \notin \mathcal{U}(t)$). If U_i is not scheduled (see Fig. 2(a)) and there is a task in its queue (i.e., $Q_i(t)=1$), then the queue will be empty in the next slot with probability (w.p.) $p_{10}^{(0)} = \Pr[Q_i(t+1)=0 \mid Q_i(t)=1, U_i \notin \mathcal{U}(t)]$, i.e., the current task expires while no new task enters the queue, whereas the queue will remain full w.p. $p_{11}^{(0)} = 1-p_{10}^{(0)}$. Instead, if node U_i is scheduled (see Fig. 2(b)) and $Q_i(t)=1$, its task is completed successfully and its queue in the next slot is either empty or full w.p. $p_{10}^{(1)} = \Pr[Q_i(t+1)=0 \mid Q_i(t)=1, U_i \in \mathcal{U}(t)]$ and $p_{11}^{(1)} = 1-p_{10}^{(1)}$, respectively. Probability $p_{11}^{(1)}$ accounts for the possible arrival of a new task. If $Q_i(t)=0$ the probabilities of receiving a new task when U_i is not scheduled and scheduled are $p_{01}^{(0)} = \Pr[Q_i(t+1)=1 \mid Q_i(t)=0, U_i \notin \mathcal{U}(t)]$ and $p_{01}^{(1)} = \Pr[Q_i(t+1)=1 \mid Q_i(t)=0, U_i \in \mathcal{U}(t)]$, respectively, while the probabilities of receiving no task are $p_{00}^{(0)} = 1-p_{01}^{(0)}$ and $p_{00}^{(1)} = 1-p_{01}^{(1)}$, respectively.

B. Related Work and Contributions

In this work we assume that the CC has no direct access to the state of the task queues $Q_1(t),\ldots,Q_M(t)$, while it knows the transition probabilities $p_{xy}^{(u)}$, with $x,y,u \in \{0,1\}$, and the outcomes of previously scheduled tasks. The scheduling problem is thus formalized as a partially observable Markov decision process (POMDP) [5], and then cast into a restless multi-armed bandit (RMAB) problem [6]. A RMAB is constituted by a set of arms (the queues in our model), a subset of which needs to be activated (or scheduled) in each slot by the controller.

To elaborate, we assume that the transition probabilities of the Markov chains in Fig. 2, the number of nodes M and servers K are such that

$$m = \frac{M}{K}, \text{ is integer, and} \tag{1a}$$

$$p_{11}^{(1)} \leq p_{01}^{(1)} \leq p_{01}^{(0)} \leq p_{11}^{(0)}. \tag{1b}$$

Assumption (1a) states that the ratio $m = M/K$ between the numbers M of nodes and K of servers is an integer, which generalizes the single-

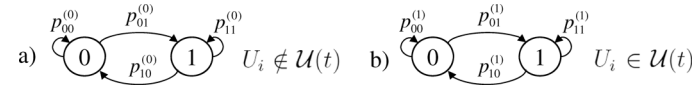

Fig. 2. Markov model for the evolution of the state of the task queue $Q_i(t) \in \{0,1\}$, when the node U_i: (a) is not scheduled in slot t (i.e., $U_i \notin \mathcal{U}(t)$); (b) is scheduled in slot t (i.e., $U_i \in \mathcal{U}(t)$).

server case ($K = 1$). Whereas, proving the results in this technical note for the case of non-integer m remains an open problem. Inequalities (1b) are motivated as follows[1]. The inequality $p_{11}^{(1)} \leq p_{01}^{(1)}$ imposes that the probability that a new task arrives when the task queue is full and the node is scheduled ($p_{11}^{(1)}$) is no larger than when the task queue is empty ($p_{01}^{(1)}$). This applies, e.g., when the arrival of a new task is independent on the queue's state and scheduling decisions (i.e., $p_{11}^{(1)} = p_{01}^{(1)}$), or when a new task is not accepted when the queue is full, i.e., $p_{11}^{(1)} = 0$. Inequality $p_{01}^{(1)} \leq p_{01}^{(0)}$ applies, e.g., when the task generation process does not depend on the queue's state and on the scheduling decisions, so that $p_{01}^{(1)} = p_{01}^{(0)}$, or when a new task cannot be accepted while the node is scheduled even if the queue is empty ($p_{01}^{(1)} = 0$). Inequality $p_{01}^{(0)} \leq p_{11}^{(0)}$ indicates that, when a node is not scheduled, the probability $p_{01}^{(0)}$ that its task queue is full in the next slot, given that it is currently empty, is smaller than the probability $p_{11}^{(0)}$ that the task queue is full in the next slot given that it is currently full. This applies, e.g., when the task generation and expiration processes are independent of each other.

1) Main Contributions: When the task queues are of capacity one, and under assumptions (1), we show that the *myopic policy* (MP) for the RMAB at hand is a round robin (RR) strategy that: *i*) re-numbers the nodes in a decreasing order according to the initial probability that their respective task queue is full; and *ii*) schedules the nodes periodically in group of K leveraging the initial ordering. The MP is proved to be throughput-optimal. Note that, the results in this technical note represent a rare case in which the optimal policy for a RMAB can be found explicitly [6].

2) Related Work: The work in this technical note is related to the works [7], [8], in which a similar RMAB problem is addressed. However, the main difference between our RMAB and the one in [7], [8] is the evolution of the arms across slots. In particular, in our RMAB, each arm evolves across a slot depending on the scheduling decision taken by the CC, while in [7], [8], the evolution of the arms is independent on the scheduling decision. The transition probabilities for the RMAB in [7], [8] are thus equivalent to setting $p_{01}^{(0)} = p_{01}^{(1)}$ and $p_{11}^{(0)} = p_{11}^{(1)}$ in the Markov chains of Fig. 2. For instance, our model applies to scenarios in which the arms are, e.g., data queues, where each arm draws a data packet from its queue only when scheduled. Instead, the model in [7], [8] applies to scenarios in which the arms are, e.g., communication channels, whose quality evolves across slots regardless whether they are selected for transmission or not.

In [7] it is shown that the MP is optimal for $p_{01}^{(0)} = p_{01}^{(1)} \leq p_{11}^{(0)} = p_{11}^{(1)}$ with $K = 1$, while [8] extends this result to an arbitrary K. The work [7] also proves that the MP in not generally optimal in the case $p_{01}^{(0)} = p_{01}^{(1)} \geq p_{11}^{(0)} = p_{11}^{(1)}$. We emphasize that neither our model nor the one in [7], [8] subsumes the other, and the results here and in [7], [8] are thus complementary.

Notation: Given a vector $\mathbf{x} = [x_1, \ldots, x_M]$ and a set $\mathcal{S} = \{i_1, \ldots, i_K\} \subseteq \{1, \ldots, M\}$ of cardinality $K \leq M$, we define vector $\mathbf{x}_{\mathcal{S}} = [x_{i_1}, \ldots, x_{i_K}]$. A scalar function $f(\mathbf{x})$ of $\mathbf{x}$ is also denoted as $f(x_1, \ldots, x_M)$ or as $f(x_1, \ldots, x_l, \mathbf{x}_{\{l+1,\ldots,M\}})$ for some $1 \leq l \leq M$, or similar notations depending on the context. Given a set $\mathcal{A}$ and a subset $\mathcal{B} \subseteq \mathcal{A}$, $\mathcal{B}^c$ represents the complement of $\mathcal{B}$ in $\mathcal{A}$.

[1]Assumption (1b) holds, e.g., in the relevant setting in which the task generation process is independent of the queue's state, the task expiration process and the scheduling command. To see this, let λ be the probability that a new task is generated in each slot and let μ be the probability that a task expires while the queue is full and the node is not scheduled. It immediately follows that $p_{01}^{(0)} = p_{01}^{(1)} = \lambda$, $p_{11}^{(0)} = \lambda + (1-\mu)(1-\lambda)$, and $p_{11}^{(1)} = \lambda$, and thus inequalities (1b) hold for any λ and μ.

II. Problem Formulation

Here we formalize the scheduling problem of Section I (see Fig. 1), in which the task generation and expiration processes are modeled, independently at each node, by the Markov models of Section I-A with queues of capacity one.

A. Problem Definition

The scheduling problem at the CC is addressed in a finite-horizon scenario over slots $t \in \{1, \ldots, T\}$. Let $\mathbf{Q}(t) = [Q_1(t), \ldots, Q_M(t)]$ be the vector collecting the states of the task queue at slot t. At slot $t = 1$, the CC is only aware of the initial probability distribution $\boldsymbol{\omega}(1) = [\omega_1(1), \ldots, \omega_M(1)]$ of $\mathbf{Q}(1)$, whose ith entry is $\omega_i(1) = \Pr[Q_i(1) = 1]$. Thus, the subset $\mathcal{U}(1)$ of $|\mathcal{U}(1)| = K$ nodes scheduled at slot $t = 1$ is chosen as a function of the initial distribution $\boldsymbol{\omega}(1)$ only. For any node $U_i \in \mathcal{U}(t)$ scheduled at slot t, an *observation* is made available to the CC at the end of the slot, while no observations are available for non-scheduled nodes $U_i \notin \mathcal{U}(t)$. Specifically, if $U_i \in \mathcal{U}(t)$ and $Q_i(t) = 1$, the task of U_i is served within slot t, and the CC observes that $Q_i(t) = 1$. Conversely, if $Q_i(t) = 0$ and $U_i \in \mathcal{U}(t)$, no tasks are completed and thus the CC observes that $Q_i(t) = 0$. We define $\mathcal{O}(t) = \{Q_i(t) : U_i \in \mathcal{U}(t)\}$ as the set of (new) observations made available at the CC at the end of slot t. At time t, the CC hence knows the history $\mathcal{H}(t)$ of all decisions and previous observations and the initial distribution $\boldsymbol{\omega}(1)$, where $\mathcal{H}(t) = \{\mathcal{U}(1), \ldots, \mathcal{U}(t-1), \mathcal{O}(1), \ldots, \mathcal{O}(t-1), \boldsymbol{\omega}(1)\}$, with $\mathcal{H}(1) = \{\boldsymbol{\omega}(1)\}$.

Since the CC has only partial information about the system state $\mathbf{Q}(t)$, through $\mathcal{O}(t)$, the scheduling problem at hand can be modeled as a POMDP. It is well-known that a sufficient statistics for taking decisions in such POMDP is given by the probability distribution of $\mathbf{Q}(t)$ conditioned on the history $\mathcal{H}(t)$ [9], referred to as *belief*, and represented by the vector $\boldsymbol{\omega}(t) = [\omega_1(t), \ldots, \omega_M(t)]$, with ith entry given by

$$\omega_i(t) = \Pr\left[Q_i(t) = 1 \mid \mathcal{H}(t)\right]. \tag{2}$$

Since the belief $\boldsymbol{\omega}(t)$ fully summarizes the entire history $\mathcal{H}(t)$ of past actions and observations [9], a *scheduling policy* $\pi = [\mathcal{U}^{\pi}(1), \ldots, \mathcal{U}^{\pi}(T)]$ is defined as a collection of functions $\mathcal{U}^{\pi}(t)$ that map the belief $\boldsymbol{\omega}(t)$ to a subset $\mathcal{U}(t)$ of $|\mathcal{U}(t)| = K$ nodes, i.e., $\mathcal{U}^{\pi}(t)$: $\boldsymbol{\omega}(t) \rightarrow \mathcal{U}(t)$. We will refer to $\mathcal{U}^{\pi}(t)$ as the subset of scheduled nodes, even though, strictly speaking, it is the mapping function defined above. The transition probabilities over the belief space are derived in Section II-B.

The *immediate reward* $R(\boldsymbol{\omega}, \mathcal{U})$, accrued by the CC when the belief vector is $\boldsymbol{\omega}$ and action $\mathcal{U}$ is taken, measures the average number of tasks completed within the current slot, and it is

$$R(\boldsymbol{\omega}, \mathcal{U}) = \sum_{U_i \in \mathcal{U}} \omega_i. \tag{3}$$

Notice that, since there are only K servers we have $R(\boldsymbol{\omega}, \mathcal{U}) \leq K$.

The *throughput* measures the average number of tasks completed over the slots $\{1, \ldots, T\}$ that, by exploiting (3) and under policy π, is given by

$$V_1^{\pi}(\boldsymbol{\omega}(1)) = \sum_{t=1}^{T} \beta^{t-1} \mathrm{E}^{\boldsymbol{\pi}}\left[R\left(\boldsymbol{\omega}(t), \mathcal{U}^{\boldsymbol{\pi}}(t)\right) \mid \boldsymbol{\omega}(1)\right]. \tag{4}$$

In (4), the expectation $\mathrm{E}^{\boldsymbol{\pi}}[\cdot|\boldsymbol{\omega}(1)]$, under policy π for initial belief $\boldsymbol{\omega}(1)$, is intended with respect to the distribution of the Markov process

$\boldsymbol{\omega}(t)$, as obtained from the transition probabilities to be derived in Section II-B. For generality, the definition (4) includes a discount factor $0 \leq \beta \leq 1$ [7], while the infinite horizon scenario (i.e., $T \to \infty$) will be discussed in Section III-C.

The goal is to find a policy $\pi^* = [\mathcal{U}^*(1), \ldots, \mathcal{U}^*(T)]$ that maximizes the throughput (4) so that

$$V_1^*(\boldsymbol{\omega}(1)) = V_1^{\pi^*}(\boldsymbol{\omega}(1)) = \max_{\pi} V_1^{\pi}(\boldsymbol{\omega}(1)), \quad \text{with} \quad \pi^* = \arg\max_{\pi} V_1^{\pi}(\boldsymbol{\omega}(1)). \tag{5}$$

B. Transition Probabilities

The belief transition probabilities, given decision $\mathcal{U}(t) = \mathcal{U}$ and $\boldsymbol{\omega}(t) = \boldsymbol{\omega} = [\omega_1, \ldots, \omega_M]$, are

$$p_{\boldsymbol{\omega}\boldsymbol{\omega}'}^{(\mathcal{U})} = \Pr\left[\boldsymbol{\omega}(t+1) = \boldsymbol{\omega}' \mid \boldsymbol{\omega}(t) = \boldsymbol{\omega}, \mathcal{U}(t) = \mathcal{U}\right] = \prod_{i=1}^{M} \Pr[\omega_i(t+1) = \omega_i' \mid \omega_i(t) = \omega_i, \mathcal{U}(t) = \mathcal{U}] \tag{6}$$

where $\boldsymbol{\omega}(t+1) = \boldsymbol{\omega}' = [\omega_1', \ldots, \omega_M']$, while the distribution of entry $\omega_i(t+1)$ is (see Fig. 2)

$$\Pr[\omega_i(t+1) = \omega_i' \mid \omega_i(t) = \omega_i, \mathcal{U}(t) = \mathcal{U}] = \begin{cases} \omega_i & \text{if } \omega_i' = p_{11}^{(1)} \text{ and } U_i \in \mathcal{U} \\ (1-\omega_i) & \text{if } \omega_i' = p_{01}^{(1)} \text{ and } U_i \in \mathcal{U} \\ 1 & \text{if } \omega_i' = \tau_0^{(1)}(\omega_i) \text{ and } U_i \notin \mathcal{U} \end{cases} \tag{7}$$

where we have defined the deterministic function

$$\tau_0^{(1)}(\omega) = \Pr[Q_i(t+1) = 1 \mid \omega_i(t) = \omega, U_i \notin \mathcal{U}(t)] = \omega p_{11}^{(0)} + (1-\omega) p_{01}^{(0)} = \omega\delta_0 + p_{01}^{(0)} \tag{8}$$

to indicate the next slot's belief when U_i is not scheduled ($U_i \notin \mathcal{U}(t)$), with $\delta_0 = p_{11}^{(0)} - p_{01}^{(0)} \geq 0$ due to inequalities (1b). Equation (8) follows from Fig. 2(a)), since the next slot's belief is either $p_{11}^{(0)}$ if $Q_i(t) = 1$ (w.p. ω) or $p_{01}^{(0)}$ if $Q_i(t) = 0$ (w.p. $(1-\omega)$).

Under assumptions (1b), it is easy to verify that function (8) satisfies the inequalities

$$p_{11}^{(1)} \leq p_{01}^{(1)} \leq \tau_0^{(1)}(\omega), \text{ for all } 0 \leq \omega \leq 1, \text{and} \tag{9}$$

$$\tau_0^{(1)}(\omega) \leq \tau_0^{(1)}(\omega'), \text{ for all } \omega \leq \omega' \text{ with } 0 \leq \omega,\ \omega' \leq 1. \tag{10}$$

The inequalities in (9) guarantee that the belief of a non-scheduled node is always larger than that of a scheduled one, while the inequality (10) says that the belief ordering of two non-scheduled nodes is maintained across a slot. These inequalities play a crucial role in the analysis below.

C. Optimality Equations

The dynamic programming (DP) formulation of problem (5) (see e.g., [10]) allows to express the throughput recursively over the horizon $\{t, \ldots, T\}$, under policy π and initial belief $\boldsymbol{\omega}$, as

$$V_t^{\pi}(\boldsymbol{\omega}) = \sum_{j=t}^{T} \beta^{j-t} \mathrm{E}^{\boldsymbol{\pi}}\left[R\left(\boldsymbol{\omega}(j), \mathcal{U}^{\boldsymbol{\pi}}(j)\right) \mid \boldsymbol{\omega}(t) = \boldsymbol{\omega}\right] = R\left(\boldsymbol{\omega}, \mathcal{U}^{\boldsymbol{\pi}}(t)\right) + \beta \sum_{\boldsymbol{\omega}'} p_{\boldsymbol{\omega}\boldsymbol{\omega}'}^{(\mathcal{U}^{\pi})} V_{t+1}^{\pi}(\boldsymbol{\omega}') \tag{11}$$

where $V_t^{\pi}(\cdot) = 0$ for $t > T$. The DP optimality conditions (or *Bellman equations*) are then expressed in terms of the *value function* $V_t^*(\boldsymbol{\omega}) = \max_{\pi} V_t^{\pi}(\boldsymbol{\omega})$, which represents the optimal throughput in the interval $\{t, \ldots, T\}$, and it is given by

$$V_t^*(\boldsymbol{\omega}) = \max_{\mathcal{U}(t) = \mathcal{U} \subseteq \{U_1, \ldots, U_M\}} \left\{ R(\boldsymbol{\omega}, \mathcal{U}) + \beta \sum_{\boldsymbol{\omega}'} p_{\boldsymbol{\omega}\boldsymbol{\omega}'}^{(\mathcal{U})} V_{t+1}^*(\boldsymbol{\omega}') \right\}. \tag{12}$$

Note that, since the nodes are stochastically equivalent, the value function (12) only depends on the numerical values of the entries of the belief vector $\boldsymbol{\omega}$ regardless of the way it is ordered. Finally, an *optimal policy* $\pi^* = [\mathcal{U}^*(1), \ldots, \mathcal{U}^*(T)]$ (see (5)) is such that $\mathcal{U}^*(t)$ attains the maximum in the condition (12) for $t \in \{1, 2, \ldots, T\}$.

III. Optimality of the Myopic Policy

We now define the myopic policy (MP) and show that, under assumptions (1), it is a round-robin (RR) policy that schedules the nodes periodically and that it is optimal for problem (5).

A. The Myopic Policy is Round-Robin

The MP $\pi^{MP} = \{\mathcal{U}^{MP}(1), \ldots, \mathcal{U}^{MP}(T)\}$, with throughput $V_t^{MP}(\cdot)$, is the greedy policy that schedules at each slot the K nodes with the largest beliefs so as to maximize the immediate reward (3), that is, we have

$$\mathcal{U}^{MP}(t) = \arg\max_{\mathcal{U}} R(\boldsymbol{\omega}(t), \mathcal{U}) = \arg\max_{\mathcal{U}} \sum_{U_i \in \mathcal{U}} \omega_i(t). \tag{13}$$

Proposition 1: Under assumptions (1), the MP (13), given an initial belief $\boldsymbol{\omega}'(1)$, is a RR policy that operates as follows: **1)** Sort vector $\boldsymbol{\omega}'(1)$ in a decreasing order to obtain $\boldsymbol{\omega}(1) = [\omega_1(1), \ldots, \omega_M(1)]$ such that $\omega_1(1) \geq \cdots \geq \omega_M(1)$. Re-number the nodes so that U_i has belief $\omega_i(1)$; **2)** Divide the nodes into m groups of K nodes each, so that the gth group $\mathcal{G}_g$, $g \in \{1, \ldots, m\}$, contains all nodes U_i such that $g = \lfloor (i-1)/K \rfloor + 1$, namely: $\mathcal{G}_1 = \{U_1, \ldots, U_K\}$, $\mathcal{G}_2 = \{U_{K+1}, \ldots, U_{2K}\}$, and so on; **3)** Schedule the groups in a RR fashion with period m slots, so that groups $\mathcal{G}_1, \ldots, \mathcal{G}_m, \mathcal{G}_1, \ldots$ are sequentially scheduled at slot $t = 1, \ldots, m, m+1, \ldots$ and so on.

Proof: According to (13), the first scheduled set is $\mathcal{U}^{MP}(1) = \mathcal{G}_1 = \{U_1, U_2, \ldots, U_K\}$. The beliefs are then updated through (7). Recalling (9), the scheduled nodes, in $\mathcal{G}_1$, have their belief updated to either $p_{11}^{(1)}$ or $p_{01}^{(1)}$, which are both smaller than the belief of any non-scheduled node in $\{U_1, \ldots, U_M\} \setminus \mathcal{G}_1$. Moreover, the ordering of the non-scheduled nodes' beliefs is preserved due to (10). Hence, the second scheduled group is $\mathcal{U}^{MP}(2) = \mathcal{G}_2$, the third is $\mathcal{U}^{MP}(3) = \mathcal{G}_3$, and so on. This proves that the MP, upon an initial ordering of the beliefs, is a RR policy. ■

We emphasize that the MP sorts the beliefs of the nodes only at the first slot in which it is operated, and then it keeps scheduling the groups of nodes according to their initial ordering, without requiring to recalculate the beliefs.

B. Optimality of the Myopic Policy

We now prove the optimality of the MP by showing that it satisfies the Bellman (12). To start with, let us consider a RR policy π^{RR} that operates according to steps **2)** and **3)** of Proposition 1 only (i.e., without re-ordering the nodes according to their initial belief), and let its throughput (11) be denoted by $V_t^{RR}(\boldsymbol{\omega})$. Note that, when the initial belief $\boldsymbol{\omega}$ is ordered so that $\omega_1 \geq \cdots \geq \omega_M$, then $V_t^{RR}(\boldsymbol{\omega}) = V_t^{MP}(\boldsymbol{\omega})$. Based on backward induction arguments similarly to [7], [8], the following lemma establishes a sufficient condition for the optimality of the MP.

Lemma 1: Assume that the MP is optimal at slot $t+1,\dots,T$, i.e., it satisfies (12). To show that the MP is optimal also at slot t it is sufficient to prove the inequality

$$V_t^{RR}(\boldsymbol{\omega}_{\mathcal{S}},\boldsymbol{\omega}_{\mathcal{S}^c}) \le V_t^{MP}(\boldsymbol{\omega}_{\mathcal{S}},\boldsymbol{\omega}_{\mathcal{S}^c}) = V_t^{RR}(\omega_1,\omega_2,\dots,\omega_M), \quad \text{for all } \omega_1 \ge \omega_2 \ge \cdots \ge \omega_M \tag{14}$$

and all sets $\mathcal{S} \subseteq \{1,\dots,M\}$ of K elements, with the elements in $\boldsymbol{\omega}_{\mathcal{S}^c}$ decreasingly ordered.

Proof: Since the MP is optimal from $t+1$ onward by assumption, it is sufficient to show that scheduling K nodes with arbitrary beliefs at slot t and then following the MP from slot $t+1$ on is no better than following the MP immediately at slot t. The performance of the former policy is given by the left-hand side (LHS) of (14). In fact $V_t^{RR}(\boldsymbol{\omega}_{\mathcal{S}},\boldsymbol{\omega}_{\mathcal{S}^c})$, for any set $\mathcal{S}$, represents the throughput of a policy that schedules the K nodes with beliefs $\boldsymbol{\omega}_{\mathcal{S}}$ at slot t, and then operates as the MP from $t+1$ onward, since beliefs in $\boldsymbol{\omega}_{\mathcal{S}^c}$ are decreasingly ordered. The MP's performance is instead given by the right-hand side (RHS) of (14). Note that, for $t=T$, it is immediate to verify that the MP is optimal. This concludes the proof. ■

Theorem 1: Under assumptions (1) the MP is optimal for problem (5), so that $\pi^{MP} = \pi^*$.

Proof: To start with, we first prove in Appendix A that the inequality

$$V_t^{RR}(\omega_1,\dots,\omega_j,y,x,\dots,\omega_M) \le V_t^{RR}(\omega_1,\dots,\omega_j,x,y,\dots,\omega_M) \tag{15}$$

holds for any $x \ge y$, with $0 \le j \le M-2$, and for all $t \in \{1,\dots,T\}$ and beliefs ω_k (not necessarily ordered), with $k \in \{1,\dots,M\}$. Inequality (15) for $j=0$ is intended as $V_t^{RR}(y,x,\dots,\omega_M) \le V_t^{RR}(x,y,\dots,\omega_M)$. If (15) holds, then inequality (14) is satisfied for all $\omega_1 \ge \cdots \ge \omega_M$ and all subsets $\mathcal{S} \subseteq \{1,\dots,M\}$ of K elements, with elements in $\boldsymbol{\omega}_{\mathcal{S}^c}$ decreasingly ordered. In fact, (15) states that the throughput of the RR policy never increases when, for any pair of adjacent nodes, the one with the smallest belief of the pair is scheduled first. Hence, by starting from the LHS of (14) (i.e., $V_t^{RR}(\boldsymbol{\omega}_{\mathcal{S}},\boldsymbol{\omega}_{\mathcal{S}^c})$) and by applying a convenient number of successive switchings between pair of adjacent elements of vector $[\boldsymbol{\omega}_{\mathcal{S}},\boldsymbol{\omega}_{\mathcal{S}^c}]$ to achieve $[\omega_1,\omega_2,\dots,\omega_M]$, we can obtain a cascade of inequalities as (15) (one for each switching), which guarantees that (14) holds. The successive switchings can be done, e.g., by using the bubble sort algorithm (see e.g., [11]), which repeatedly steps through the list to be sorted, comparing each pair of adjacent elements and swapping them if they are not decreasingly ordered. Accordingly, by Lemma 1 this is sufficient to prove that the MP is optimal, since inequality (14) holds for any arbitrary t. ■

C. Extension to the Infinite-Horizon Case

We now briefly describe the extension of problem (5) to the infinite-horizon case, for which the throughput under policy π and its optimal value are given by (see e.g., [7])

$$V^{\pi}(\boldsymbol{\omega}(1)) = \sum_{t=1}^{\infty} \beta^{t-1} \mathrm{E}^{\boldsymbol{\pi}}\left[R\left(\boldsymbol{\omega}(t),\mathcal{U}^{\pi}(t)\right) \mid \boldsymbol{\omega}(1)\right], \text{ and}$$
$$V^{*}(\boldsymbol{\omega}(1)) = \max_{\pi} V^{\pi}(\boldsymbol{\omega}(1)) \tag{16}$$

where the optimal policy is $\pi^* = \arg\max_{\pi} V^{\pi}(\boldsymbol{\omega}(1))$ and $0 \le \beta < 1$. From standard DP theory [10], the optimal policy π^* is *stationary*, so that the optimal decision $\mathcal{U}^*(t)$ is a function of the current state $\boldsymbol{\omega}(t)$ only, independently of slot t [10]. By following the same reasoning as in [7, Theorem 3], it can be shown that the optimality of the MP for the finite-horizon setting implies the optimality also for the infinite-horizon scenario, and similarly to [12] it can be proved that the MP, under certain conditions on the transitions probabilities $p_{xy}^{(u)}$, with $x,y,u \in \{0,1\}$, coincides with the Whittle index policy for the RMAB at hand (see [4] for full details). Moreover, by following [7, Theorem 4] it can be shown that the MP is optimal also for the undiscounted average reward criterion (i.e., $V_{avg}^{\pi}(\boldsymbol{\omega}(1)) = \lim_{T\to\infty}(1/T)\sum_{t=1}^{\infty}\mathrm{E}^{\boldsymbol{\pi}}\left[R\left(\boldsymbol{\omega}(t),\mathcal{U}^{\pi}(t)\right)|\boldsymbol{\omega}(1)\right]$).

IV. Conclusion

This technical note considers a centralized scheduling problem for independent, symmetric and time-sensitive tasks under resources constraints. The problem is to assign a finite number of resources to a larger number of *nodes* that may have tasks to be completed in their *task queue*. It is assumed that the *central controller* can not directly monitor the state of the queue of each node. Based on a Markovian modeling of the task generation and expiration processes, the scheduling problem is formulated as a partially observable Markov decision process (POMDP), and then cast into the framework of restless multi-armed bandit (RMAB) problems. Under the assumption that the task queues are of capacity one, a greedy, or *myopic policy* (MP), operating in the space of the a posteriori probabilities (beliefs) of the number of tasks in the queues, is proved to be optimal for both finite and infinite-horizon throughput criteria. The MP selects at each slot the nodes with the largest probability of having a task to be completed. It is shown that the MP is round-robin as it schedules the nodes periodically. When the task queues have capacities larger than one, the MP is generally suboptimal and finding optimal scheduling policies is still an open problem [4].

Overall, this technical note proposes a general framework for resource allocation that finds applications in several areas of current interest including communication networks and distributed computing.

Appendix
Proof of Theorem 1

The proof is divided into two steps. In the first step we derive the throughput of the RR policy in closed form, and then we show that inequality (15) holds.

As for the first step, the throughput for the RR policy (and thus of the MP) can be calculated as the sum of the contribution of each node separately (due to the round robin structure). To elaborate, let us focus on node U_i, with initial belief $\omega_i(1)$, and assume that $U_i \in \mathcal{G}_1$. Nodes in group $\mathcal{G}_1$ are scheduled at slots $t \in \{1+(j-1)m\}$, for $j \in \{1,2,\dots\}$. Let $r_j(\omega_i(1)) = \mathrm{E}^{\mathrm{RR}}[\omega_i(1+(j-1)m) \mid \omega_i(1)]$ be the average reward accrued by the CC from node U_i only, when scheduling it for the jth time at slot $t = 1+(j-1)m$ (see the RHS of (3)) (i.e., when operating the RR policy). At slot $t=1$ we have $r_1(\omega_i(1)) = \omega_i(1)$. To calculate $r_2(\omega_i(1))$ we first derive the average value of the belief (see (7)) after the slot of activity in $t=1$ as $\mathrm{E}^{\mathrm{RR}}[\omega_i(2) \mid \omega_i(1)] = \tau_1^{(1)}(\omega_i(1))$, where $\tau_1^{(1)} = \omega\delta_1 + p_{01}^{(1)}$ with $\delta_u = \left(p_{11}^{(u)} - p_{01}^{(u)}\right)$ (cf. (8)). We then account for the $(m-1)$ slots of passivity by exploiting (8), so that $r_2(\omega_i(1)) = \mathrm{E}^{\mathrm{RR}}[\omega_i(1+m) \mid \omega_i(1)] = \phi^{(1)}(\omega_i(t))$, where we have set $\phi^{(1)}(\omega) = \tau_0^{(m-1)}(\tau_1^{(1)}(\omega)) = \omega\alpha_m + \psi_m$ with $\alpha_m = \delta_1\delta_0^{m-1}$ and $\psi_m = p_{01}^{(1)}\delta_0^{m-1} + p_{01}^{(0)}(1-\delta_0^{m-1})/(1-\delta_0)$, and where $\tau_0^{(k)}(\omega) = \tau_0^{(1)}(\tau_0^{(k-1)}(\omega))$ indicates the belief of a node after k slots of passivity when the initial belief is ω (i.e., $\tau_0^{(k)}(\omega)$ is obtained recursively by applying $\tau_0^{(1)}(\omega)$ to itself k times). In general, we can obtain $r_j(\omega_i(1)) = \mathrm{E}^{\mathrm{RR}}[\omega_i(1+(j-1)m) \mid \omega_i(1)]$, for $j \ge 2$, by iterating the procedure above by applying $\phi^{(1)}(\omega)$ to itself $(j-1)$ times. After a little algebra we get $\phi^{(j-1)}(\omega) = \phi^{(1)}(\phi^{(j-2)}(\omega)) = \omega\alpha_m^{j-1} + \psi_m(1-\alpha_m^{j-1})/(1-\alpha_m)$, so that $r_j(\omega_i(1)) = \phi^{(j-1)}(\omega_i(1))$, where we set $\phi^{(0)}(\omega) = \omega$. The reasoning above can be applied when starting from any arbitrary slot t.

Finally, the total reward accrued by the CC from a node that is scheduled H times, when its belief at the first slot in which it is scheduled is ω, can be calculated by summing up the average reward $r_j(\cdot)$ during each slot in which the node is scheduled (see definition above), as

$$\begin{aligned}\theta^{(H)}(\omega) &= \sum_{j=1}^{H} \beta^{(j-1)m} r_j(\omega) \\ &= \frac{\psi_m}{1-\alpha_m}\left(\frac{1-\beta^{mH}}{1-\beta^m} - \frac{1-(\beta^m\alpha_m)^H}{1-\beta^m\alpha_m}\right) \\ &\quad + \frac{1-(\beta^m\alpha_m)^H}{1-\beta^m\alpha_m}\omega. \end{aligned} \tag{17}$$

Note that, for a node $U_i \in \mathcal{G}_g$, for $g \geq 1$ and with belief equal to ω at $t = 1$, the first slot in which the node is scheduled is $t = g$, and thus its belief at time $t = g$ becomes $\tau_0^{(g-1)}(\omega)$ (i.e., after $(g-1)$ slots of passivity while other groups are scheduled). Therefore, for a node $U_i \in \mathcal{G}_g$, with initial belief ω, the total contribution to the throughput is given by $\beta^{g-1}\theta^{(H)}\left(\tau_0^{(g-1)}(\omega)\right)$.

Let us now focus on the second step, i.e., proving the inequality (15). At $t = T$, it is easily seen to hold due to (3) and (11). We then need to show that (15) also holds at t. To do so, let us denote as $\mathcal{L}$ and $\mathcal{R}$ the RR policies whose throughputs are given by the LHS and RHS of (15) respectively. The differences between $\mathcal{L}$ and $\mathcal{R}$ are the positions of the nodes with belief x and y in the initial belief vectors. Therefore, some of the m groups created by the two policies might have different nodes (see the RR operations in Proposition 1). To simplify, we refer to the node with belief x (y) as node x (y). Let us assume that nodes x and y belong to groups $\mathcal{G}_{g'}$ and $\mathcal{G}_{g''}$ under policy $\mathcal{R}$, respectively, while they belong to groups $\mathcal{G}_{g''}$ and $\mathcal{G}_{g'}$ under policy $\mathcal{L}$, respectively, with $g'' \geq g'$, and g', $g'' \in \{1,\ldots,m\}$. If $g'' = g'$, then the two policies coincide and (15) holds with equality. If $g'' = g' + 1$ (nodes are adjacent but do not belong to the same group), the only difference between policies $\mathcal{L}$ and $\mathcal{R}$ is the scheduling order of nodes x and y.

To verify that inequality (15) holds, we need to prove that scheduling node y in group $\mathcal{G}_{g'}$ and node x in group $\mathcal{G}_{g''}$ is no better than doing the opposite for any $x \geq y$. To elaborate, let $H_x^{\mathcal{R}}(t) = H_y^{\mathcal{L}}(t)$ and $H_y^{\mathcal{R}}(t) = H_x^{\mathcal{L}}(t)$ be the number of times that node x (or y) is scheduled under policy $\mathcal{R}$ (or $\mathcal{L}$) and node y (or x) is scheduled under policy $\mathcal{L}$ (or $\mathcal{R}$) in the horizon $\{t, t+1, \ldots, T\}$, respectively. By recalling (17) and the discount factor β, the contribution generated by node x and y under policy $\mathcal{R}$ is $\beta^{g'-1}\theta^{\left(H_x^{\mathcal{R}}(t)\right)}(\tau_0^{(g'-1)}(x))$ and $\beta^{g''-1}\theta^{\left(H_y^{\mathcal{R}}(t)\right)}(\tau_0^{(g''-1)}(y))$ respectively, and similarly under policy $\mathcal{L}$ we have $\beta^{g''-1}\theta^{\left(H_x^{\mathcal{L}}(t)\right)}(\tau_0^{(g''-1)}(x))$ and $\beta^{g'-1}\theta^{\left(H_y^{\mathcal{L}}(t)\right)}(\tau_0^{(g'-1)}(y))$. Note that, in the argument of function $\theta^{(\cdot)}(\cdot)$, we have considered that the nodes in group $\mathcal{G}_{g'}$ are scheduled for the first time at slot $g'-1$, and thus the belief must be updated through function $\tau_0^{(g'-1)}(\cdot)$, and similarly for nodes in $\mathcal{G}_{g''}$ the first slot is $g''-1$. Moreover, the discount factor is $\beta^{g'-1}$ is common to all the nodes in group $\mathcal{G}_{g'}$, and so is $\beta^{g''-1}$ for group $\mathcal{G}_{g''}$.

By recalling that all the nodes, except x and y, are scheduled at the same slot under the two policies $\mathcal{R}$ and $\mathcal{L}$ (thus giving the same contribution to the throughput), the inequality (15) can thus be reduced to

$$\begin{aligned}&\beta^{g'-1}\theta^{\left(H_x^{\mathcal{R}}(t)\right)}(\tau_0^{(g'-1)}(x)) + \beta^{g''-1}\theta^{\left(H_y^{\mathcal{R}}(t)\right)}(\tau_0^{(g''-1)}(y)) \\ &\quad -\beta^{g''-1}\theta^{\left(H_x^{\mathcal{L}}(t)\right)}(\tau_0^{(g''-1)}(x)) - \beta^{g'-1}\theta^{\left(H_y^{\mathcal{L}}(t)\right)}(\tau_0^{(g'-1)}(y)) \geq 0\end{aligned}$$

which must hold for all admissible $H_x^{\mathcal{R}}(t) = H_y^{\mathcal{L}}(t)$ and $H_y^{\mathcal{R}}(t) = H_x^{\mathcal{L}}(t)$ and all $g'' \geq g'$, with g', $g'' \in \{1,\ldots,m\}$. There are two cases: **1)** $H_x^{\mathcal{R}}(t) = H_y^{\mathcal{L}}(t) = H_y^{\mathcal{R}}(t) = H_x^{\mathcal{L}}(t) = H \geq 1$, that is, nodes x and y are scheduled the same number of times within the horizon of interest under the two policies $\mathcal{R}$ and $\mathcal{L}$; **2)** $H_x^{\mathcal{R}}(t) = H_y^{\mathcal{L}}(t) = H$, and $H_y^{\mathcal{R}}(t) = H_x^{\mathcal{L}}(t) = H-1$, for $H \geq 1$, namely, node x (or y) is scheduled one time more than node y (or x) under policy $\mathcal{R}$ (or $\mathcal{L}$). By exploiting the RHS of (17), after a little algebra, one can verify that the inequality above holds in both cases, which concludes the proof of Theorem 1.

REFERENCES

[1] D. Bertsekas and R. G. Gallager, *Data Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[2] A. Benoit, L. Marchal, J.-F. Pineau, Y. Robert, and F. Vivien, "Scheduling concurrent bag-of-tasks applications on heterogeneous platforms," *IEEE Trans. Comput.*, vol. 59, no. 2, pp. 202–217, Feb. 2010.

[3] Y. Bai, C. Xu, and Z. Li, "Task-aware based co-scheduling for virtual machine system," in *Proc. ACM Symp. Appl. Comp.*, Sierre, Switzerland, Mar. 2010, pp. 181–188.

[4] F. Iannello, O. Simeone, and U. Spagnolini, On the Optimal Scheduling of Independent, Symmetric and Time-Sensitive Tasks [Online]. Available: http://arxiv.org/abs/1112.1229

[5] G. E. Monahan, "A survey of partially observable Markov decision processes: Theory, models, and algorithms," *Manag. Sci.*, vol. 28, no. 1, pp. 1–16, 1982.

[6] J. Gittins, K. Glazerbrook, and R. Weber, *Multi-armed Bandit Allocation Indices*. West Sussex, U.K.: Wiley, 2011.

[7] S. H. A. Ahmad, M. Liu, T. Javidi, Q. Zhao, and B. Krishnamachari, "Optimality of myopic sensing in multi-channel opportunistic access," *IEEE Trans. Inform. Theory*, vol. 55, no. 9, pp. 4040–4050, Sep. 2009.

[8] S. H. A. Ahmad and M. Liu, "Multi-channel opportunistic access: A case of restless bandits with multiple plays," in *Proc. 47th Annu. Allerton Conf. Commun., Control, Comput.*, Monticello, IL, Sep. 2009, pp. 1361–1368.

[9] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artif. Intell.*, vol. 101, pp. 99–134, May 1998.

[10] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Hoboken, NJ: Wiley, 2005.

[11] S. Ross, *Probability Models for Computer Science*. San Diego, CA: Academic Press, 2002.

[12] K. Liu and Q. Zhao, "Indexability of restless bandit problems and optimality of Whittle index for dynamic multichannel access," *IEEE Trans. Inform. Theory*, vol. 56, no. 11, pp. 5547–5567, Nov. 2010.