

# Nearly-Optimal Scheduling of Users with Markovian Time-Varying Transmission Rates<sup>☆</sup>

Fabio Cecchi<sup>a,c</sup>, Peter Jacko<sup>b,c,\*</sup>

<sup>a</sup>*Eindhoven University of Technology, P.O. Box 513, 5600 MB, Eindhoven, The Netherlands*

<sup>b</sup>*Department of Management Science, Lancaster University Management School, Lancaster, LA1 4YX, UK*

<sup>c</sup>*BCAM — Basque Center for Applied Mathematics, Mazarredo 14, 48009 Bilbao (Basque Country), Spain*

---

## Abstract

We address the problem of developing a well-performing and implementable scheduler of users with wireless connections to the central controller, which arise in areas such as mobile data networks, heterogeneous networks, or vehicular communications systems. The main feature of such systems is that the quality of each user's channel is time-varying due to fading. The evolution of the channel over its quality states thus causes a time-varying transmission rate of each user. We consider Markovian channel dynamics, relaxing the common but unrealistic assumption of i.i.d. channels. We first focus on three-state channels and show that threshold policies (of giving higher priority to users with higher transmission rate) are not necessarily optimal. For the general case we design a scheduler which generalizes the recently proposed Potential Improvement (PI) scheduler, which gives priority to the users who are unlikely to improve their actual transmission rate soon by much. We propose two practical approximations of PI, whose performance is analyzed and compared to existing alternative schedulers in a variety of simulation scenarios. Our computational experiments indicate that the variant of PI, which only relies on the steady-state distribution of the channel, is robust and performs extremely well, and therefore we recommend its use for practical implementation.

*Keywords:* wireless networks, opportunistic scheduling, finite state Markov channel, non-i.i.d., Markov decision processes, stochastic scheduling, performance evaluation, stability

---

<sup>☆</sup>The authors are grateful to U. Ayesta for many fruitful discussions. An earlier and abridged version of this work appeared at ACM Sigmetrics 2013. This research was mostly done during the stay of F. Cecchi at BCAM during August-October 2012 funded by the BCAM Internship program. This research was partially supported by grant MTM2010-17405 (Ministerio de Ciencia e Innovación, Spain).

\*Corresponding author.

*Email addresses:* [f.cecchi@tue.nl](mailto:f.cecchi@tue.nl) (Fabio Cecchi), [p.jacko@lancaster.ac.uk](mailto:p.jacko@lancaster.ac.uk) (Peter Jacko)

## 1. Introduction

This paper is motivated by the need of designing an efficient scheduler for wireless systems such as mobile data networks (4G LTE, 5G), heterogeneous networks (HetNet), or vehicular communications systems. Such a scheduler must be capable of exploiting the system capacity to serve the heterogeneous demands of the users that are within the base station’s power range in such a way so as to optimize the system performance and user experience. We model such a system as a discrete-time multi-class queue with multiple preemptive servers, in which users of different classes arrive randomly and depart once their job is completed. Each user transmission is assumed as occurring through a dedicated fading channel, whose quality evolution is class-dependent with Markovian memory. Different classes of users may have different sets of accessible transmission rates associated with the modulation and coding schemes (MCSs) of the system. That is, each user’s channel is characterized by a set of accessible *channel quality conditions* (a.k.a. channel states). The user classes have heterogeneous waiting costs and mean job sizes. The objective is to (nearly-)optimize primarily the expected time-average waiting cost per user, but keeping in mind the trade-off with other performance objectives (e.g., fairness). Such a model covers both the downlink and synchronized uplink wireless systems.

Several schedulers have been proposed recently for such a *flow-level* scheduling problem based on ad-hoc arguments, simulation outcomes or approximate optimization, e.g., in Knopp and Humblet (1995); Borst (2005); Bonald (2004b); Aalto and Lassila (2010); Ayesta et al. (2010); Jacko (2011). The seminal work by Knopp and Humblet (1995) showed that system capacity can be improved by opportunistically serving users with maximal current transmission rate. Such a scheduler, known as the *Max Rate* scheduler in the wireless networks literature, is thus *naively opportunistic*: it is myopically throughput optimal (maximizing one-slot transmission rate) and simple to implement, but it ignores the possible future evolution and was shown to perform bad in the long-term. For instance, it may quickly become unstable (i.e., the number of waiting users explodes) as the load increases, while other schedulers may keep the system stable (Aalto and Lassila, 2010; Ayesta et al., 2010). It may also be extremely unfair to users who have their highest accessible transmission rates lower than the typical transmission rates of others. This scheduler is also known as the  $c\mu$ -rule in the stochastic scheduling literature, and we will adopt this name in this paper.

Gradient-based schedulers are the state-of-the-art in opportunistic scheduling, in particular, the Proportionally Fair (PF) scheduler patented in Chaponniere et al. (2002), was proposed to be implemented in the CDMA 1xEV-DO system of 3G cellular networks (Bender et al., 2000). PF maximizes the sum of logarithmic throughputs of the network, providing an improved fairness over the  $c\mu$ -rule (Kushner and Whiting, 2004). Borst (2005) analyzed flow-level stability of PF by approximating it by the *Relatively Best* (RB) scheduler, which gives priority to users according to their ratio of the current transmission rate to the mean accessible transmission rate. This scheduler is thus *fairly opportunistic*: it takes the channels’ transmission rate distribution into account, is not myopically throughput optimal, and guarantees a minimal throughput to the users with low accessible transmission rates. However, it is not maximally stable at flow-level (Aalto and Lassila, 2010).

The schedulers proposed in [Bonald \(2004b,a\)](#); [Aalto and Lassila \(2010\)](#); [Ayesta et al. \(2010\)](#); [Ayesta and Jacko \(2013\)](#), called the *Score Based* (SB), *Proportionally Best* (PB) and *Potential Improvement* (PI) schedulers, respectively, belong to the family of the *best-condition* schedulers<sup>1</sup>. A best-condition scheduler gives absolute priority to the users who are in their respective best accessible channel quality conditions over the others, hence ignoring the actual transmission rate associated with those best conditions. Such a scheduler is thus *smartly opportunistic*: it takes into account whether an improvement is possible, yet it still ignores the possible future evolution in detail. Overall, it turns out to perform well in the long-term and in heavily loaded systems, for being maximally stable ([Ayesta et al., 2013](#); [Kim et al., 2013](#)).

The PI scheduler was designed based on optimally solving a Lagrangian relaxation of a stochastic optimization problem ([Ayesta et al., 2010](#)). The other schedulers were either designed in an ad hoc way ([Bonald, 2004b](#); [Aalto and Lassila, 2010](#)), or based on solving an optimization problem under the time-scale separation assumption ([Aalto et al., 2011](#)). Fairness of best-condition schedulers has not been addressed adequately yet, only [Ayesta et al. \(2010\)](#) illustrated in one simulation scenario that PI maintained the average number of uncompleted jobs significantly more balanced than other schedulers.

Previous work on scheduling typically assumed that the channel evolution is independent and identically distributed (i.i.d.) from slot to slot, which greatly facilitates analysis. Considering a non-i.i.d. evolution of the channel quality condition is however important. It is known that channels do have a memory, although the precise evolution is usually not known explicitly and, moreover, it can change over time, so only easy-to-compute estimates can be available. For a broader discussion see, e.g., the literature on finite-state Markov modeling of fading channels ([Bai and Atiquzzaman, 2003](#); [Sadeghi et al., 2008](#); [Wang and Moayeri, 1995](#); [Zorzi et al., 1995](#); [Wang and Chang, 1996](#); [Wei et al., 2010](#); [Zheng et al., 2013](#)), autoregressive modeling for channel estimation ([Ghandour-Haidar et al., 2012](#)), Markov decision processes for adaptive control ([Chen et al., 2013](#)), and the references therein.

This paper provides insight into the complex nature of the fundamental questions (listed below) about scheduling under Markovian channels dynamics. We are still far from providing definite answers, but we aim at giving suggestions for practical implementation of schedulers and at pointing out the main issues for future research in this area.

*What is the structure of an optimal policy?* The structure of an optimal scheduler is extremely complicated and not necessarily intuitive. We show in an example that if there is a single user in the system it may be optimal to serve her in a certain channel quality condition and not to serve her when the condition is better. That is, an intuitive threshold policy is not always optimal. However it is always optimal to serve the single user in her best channel quality condition.

*Why are there no optimality results available?* Results of maximal stability have only been established. Even in the single-user system with three channel quality conditions it is

---

<sup>1</sup>[Ayesta et al. \(2013\)](#) used the term *best-rate* scheduler, but we found that term misleading, especially w.r.t. the Max Rate scheduler, therefore we suggest the term *best-condition* scheduler.

difficult to characterize when the optimal policy has the structure of a threshold policy. However, we give sufficient conditions that are likely to be satisfied in practice, which guarantee optimality of a threshold policy.

*How do the actions taken in the non-best channel quality conditions influence the performance?* Maximal stability only indicates what to do in the channel quality condition with the highest transmission rate. In the multi-user system, the condition for maximal stability is to give highest priority to the users in their best channel quality conditions. However, nothing has been established for the users which are not in their best conditions. Comparing in simulations the maximally stable schedulers, we observe that the actions taken in the non-best conditions may strongly influence the performance in light and medium traffic, while the differences are not significant in heavy traffic.

*(When) are the maximally stable schedulers preferable in practice?* There are cases, when an unstable scheduler results in a lower mean delay than a stable scheduler in the mid-term. For instance, when the mean job size of one class is significantly larger than of others, the  $c\mu$ -rule slowly accumulates long jobs in the system, while maximally stable schedulers maintain a large number of short jobs.

*Do the maximally stable schedulers perform well even in case of class-dependent waiting costs?* SB and PB were designed to minimize the number of uncompleted jobs (i.e., assumed unitary waiting costs), so it is not obvious how to extend them to the case with heterogeneous waiting costs. If their priority formula is multiplied by the waiting cost, these schedulers may no longer be maximally stable. In fact, our simulations show that they do not perform well, especially in heavier traffic. Only the PI scheduler remains maximally stable in case of heterogeneous waiting costs.

*How to resolve the trade-off between being naively opportunistic, smartly opportunistic, and prioritizing “short” jobs?* We propose a robust and reasonably simple-to-implement variant of the PI scheduler, which relies on the steady-state distribution of the Markov chain governing the evolution of the channel quality condition.

*How fair are the maximally stable schedulers?* We observe interesting fairness properties in simulations of this variant of the PI scheduler.

In [section 2](#) we formalize the system and the scheduling problem. An MDP approach is described in [section 3](#) in order to formulate a single-user optimization problem in which a price must be paid for service. This problem is addressed in [section 4](#), where we develop index policies and study the solvability of the problem by threshold policies. We solve the problem for channel evolution over three quality conditions and partially characterize the optimal solution in general. Some further examples and special cases are given in [section 5](#). Based on these results, we propose a new scheduler and two practical approximations in [section 6](#). Their performance is evaluated and compared with schedulers proposed in previous literature in [section 7](#). Finally, [section 8](#) concludes. The proofs and a detailed indexability analysis can be found in [Appendix A](#).

## 2. Problem Description

We consider a time-slotted system, which gives rise to a discrete-time job scheduling problem. Decisions are taken at the beginning of every time slot and are applied during the whole time slot. With an abuse of notation,  $t \in \mathcal{T} := \{0, 1, \dots\}$  will denote both the decision time (epoch) and the corresponding time slot. Each slot has a duration of  $\tau$  seconds.

### 2.1. Job-Channel-User Classes

Suppose that there are  $K$  classes of users, labeled  $k \in \mathcal{K} := \{1, 2, \dots, K\}$ . Each user of class  $k$  is uniquely associated with a job of class  $k$  which she requests to download, and with a dedicated wireless channel of class  $k$  through which the transmission connection is established. A class of users could, for instance, identify users located at the same physical properties (e.g. distance, obstacles), requesting to download the same type of file (e.g., video, mp3, web page, etc.), using the same connection technology, and having the same importance for the system operator. See [Table A.2](#) and [Table A.3](#) for examples we used in our simulations.

*User Arrivals.* For each class  $k \in \mathcal{K}$ , the number of class- $k$  users arriving to the system,  $A_k(t)$ , during the time slot  $t \in \mathcal{T}$ , creates an i.i.d. arrival process  $\{A_k(t)\}_{t \in \mathcal{T}}$  with generic element  $A_k$  and mean  $\lambda_k := \mathbb{E}_0[A_k] < \infty$ , where  $\mathbb{E}_0[\cdot]$  denotes the expectation conditional on information available at time epoch 0. The arrivals are assumed to be mutually independent across user classes.

*Job Sizes.* The (integer-valued) job/flow size  $b_k$  of class- $k$  user is measured in *bits* and is geometrically distributed with mean  $\mathbb{E}[b_k] < \infty$  for classes  $k \in \mathcal{K}$ . This assumption is the main limitation of existing models (including this paper), but to the best of our knowledge there has not been any attempt to analytically approach the case of non-geometric job sizes in the literature.

*Channel Quality Conditions.* For each user, the quality condition of her channel (or simply the *channel condition*) is varying from slot to slot, independently of all other users present in the system (including other users of the same class). For each class- $k$  user, the set of accessible quality conditions of the channel is finite and denoted by  $\mathcal{N}'_k := \{1, 2, \dots, N_k\}$ . The channel quality conditions typically correspond to modulation and coding schemes (MCSs) of the user's transmission technology. Moreover, the channel quality condition of a just arrived class- $k$  user is  $n \in \mathcal{N}'_k$  with probability  $q_{k,n} \geq 0$ , which satisfies  $\sum_{n \in \mathcal{N}'_k} q_{k,n} = 1$ . Note that this is independent of other users' channel quality conditions and of the slot it arrives at.

*Channel Quality Condition Evolution.* We assume that at each slot, the channel condition of every user in the system evolves according to a class-dependent (time-homogeneous) Markovian chain. Thus, for each user of class  $k \in \mathcal{K}$ , we can define a Markov chain with state space  $\mathcal{N}'_k$ .

We define by  $q_{k,n,m} := \mathbb{P}(Z_k(t+1) = m | Z_k(t) = n)$ , where  $Z_k(t)$  denote the channel condition of a class- $k$  user at time  $t$ . So,  $q_{k,n,m}$  represents the probability that the channel

of a class- $k$  user moves from condition  $n$  to condition  $m$  in one slot. The class- $k$  channel condition transition probability matrix is thus

$$\mathbf{Q}_k := \begin{matrix} & \begin{matrix} 1 & 2 & \dots & N_k \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ \vdots \\ N_k \end{matrix} & \begin{pmatrix} q_{k,1,1} & q_{k,1,2} & \dots & q_{k,1,N_k} \\ q_{k,2,1} & q_{k,2,2} & \dots & q_{k,2,N_k} \\ \vdots & \vdots & \ddots & \vdots \\ q_{k,N_k,1} & q_{k,N_k,2} & \dots & q_{k,N_k,N_k} \end{pmatrix} \end{matrix}$$

where  $\sum_{m \in \mathcal{N}'_k} q_{k,n,m} = 1$  for every condition  $n \in \mathcal{N}'_k$ . We emphasize that the channel condition transitions are independent across users.

*Transmission Rates.* Each channel condition corresponds to a particular MCS, which in turn determines the transmission rate while the channel is in this condition. When a class- $k$  user is in channel condition  $n \in \mathcal{N}'_k$ , she can receive data at transmission rate  $s_{k,n}$  bits per second. We assume that the higher the label of the channel condition, the higher the transmission rate, i.e.,  $0 \leq s_{k,1} < s_{k,2} < \dots < s_{k,N_k}$ .

To avoid trivial cases, we assume that each class  $k$  can be served, i.e.,  $s_{k,N_k} > 0$ , and that at least one class  $k$  is time-varying, i.e.,  $s_{k,1} < s_{k,N_k}$ .

*Waiting Costs.* For every user of class  $k$  the system operator accrues waiting cost  $c_k > 0$  at the end of every slot while her job is uncompleted.

## 2.2. Server

The server (e.g., a base station) has full knowledge of the above-defined parameters. At the beginning of every slot, the server observes the actual channel conditions of all the users present in the system, and decides which of them to serve during the slot. At every time, no more than  $C$  users can be served in parallel, i.e.  $C$  is the capacity of the server. We assume that the server is preemptive, that is, at every epoch it is permitted to interrupt the service of a user whose job is not yet completed. The server is also allowed to idle, and note that it is not work-conserving because of the time-varying transmission rate. Motivated by practical implementation, the observations of the processes defined above at epoch  $t$  always include arrivals at epoch  $t$ , while between two consecutive epochs, only service but no new arrivals occur.

## 2.3. Objectives

The aim is to identify scheduling policies that perform well with respect to the following objectives (or their combination):

- minimization of the expected average waiting cost per user;
- minimization of the expected time-average number of uncompleted jobs;
- maximization of some time-average fairness function across classes.

The approach we undertake in this paper is to minimize the expected average waiting cost per user, which as a special case covers also the second criterion. The performance of the suggested scheduler will be evaluated in relation to the fairness criterion in simulations.

### 3. MDP Approach

In this section we employ a Markov decision process (MDP) approach to design a well-grounded scheduling policy. Indeed, we extend the modeling framework introduced for the scheduling problem with i.i.d. channel condition evolution in [Ayesta et al. \(2010\)](#) based on restless bandits ([Whittle, 1988](#)).

[Whittle \(1988\)](#) suggested for the restless bandits an approach that leads to a per-user decomposition of the relaxed problem in which the constraint of the server's per-slot capacity is  $C$  *in expectation* rather than pointwise. This approach decomposes the original infinitely-dimensional problem into infinite number of parametric optimization problems of a single user of each class  $k$ . This approach of relaxation and decomposition was shown useful for designing a scheduler ([Ayesta et al., 2010](#)) that is well-performing, maximally stable and fluid-optimal under arbitrary arrivals in the special case of i.i.d. channels ([Ayesta et al., 2013](#)). The approach requires to analyze a single-user problem in which one pays price  $\nu$  for service. This parameter appears as the Lagrange multiplier from the Lagrangian relaxation, e.g., see [Ayesta et al. \(2010\)](#).

In order to admit an analytical approach, we introduce discounting of the waiting costs, with discount factor  $0 \leq \beta < 1$ . The results for the undiscounted case, which corresponds to the time-average criterion, will be obtained in the limit  $\beta \rightarrow 1$ .

We further need to characterize the departure probabilities. We denote by  $\mu_{k,n}$  the probability that the job requested is completed within the current time slot when the server is allocated to a class- $k$  user in channel condition  $n \in \mathcal{N}'_k$ . Since we consider jobs with geometric size, we can employ the results from [Ayesta et al. \(2010\)](#); [Jacko \(2011\)](#) that  $\mu_{k,n} = \min\{1, 1 - (1 - 1/\mathbb{E}_0[b_k])^{\tau s_{k,n}}\}$  which can be approximated, if jobs are sufficiently long (in particular, if  $\tau s_{k,n}/\mathbb{E}_0[b_k] \approx 0$ ) by

$$\mu_{k,n} \approx \tau s_{k,n} / \mathbb{E}_0[b_k]. \quad (1)$$

We remark that the departure probabilities are increasing in the channel condition, i.e.,  $0 \leq \mu_{k,1} < \dots < \mu_{k,N_k} \leq 1$ , because the transmission rates  $s_{k,n}$  are so.

#### 3.1. Job-Channel-User Restless Bandit Definition

At every time epoch, the generic class- $k$  user can be allocated with zero capacity of the base station or be one of the users served. We denote by  $\mathcal{A}_k$  the action space of user  $k$ , in particular  $\mathcal{A}_k := \{0, 1\}$  where the action 0 means not serving, while action 1 means serving.

Every job-channel-user of class  $k$  is characterized by the tuple

$$(\mathcal{N}_k, (\mathbf{W}_k^a)_{a \in \mathcal{A}}, (\mathbf{R}_k^a)_{a \in \mathcal{A}}, (\mathbf{P}_k^a)_{a \in \mathcal{A}}),$$

where

- $\mathcal{N}_k := \{0\} \cup \mathcal{N}'_k$  is the *state space* of a user, where state 0 indicates that the job is completed, and state  $n \in \mathcal{N}'_k$  indicates that the current channel condition is  $n$  and the job is uncompleted;
- $\mathbf{W}_k^a := (W_{k,n}^a)_{n \in \mathcal{N}_k}$ , where  $W_{k,n}^a$  is the expected one-slot capacity consumption, or *work* required by a user at state  $n$  if action  $a$  is selected at a time epoch. Specifically, for every state  $n \in \mathcal{N}_k$ ,

$$W_{k,n}^1 := 1, \quad W_{k,n}^0 := 0;$$

- $\mathbf{R}_k^a := (R_{k,n}^a)_{n \in \mathcal{N}_k}$ , where  $R_{k,n}^a$  is the expected one-slot *reward* earned by a user at state  $n$  if action  $a$  is selected at a time epoch. Specifically, for every state  $n \in \mathcal{N}'_k$ , it is the negative of the expected waiting cost,

$$R_{k,0}^a := 0, \quad R_{k,n}^1 := -c_k(1 - \mu_{k,n}), \quad R_{k,n}^0 := -c_k;$$

- $\mathbf{P}_k^a := (p_{k,n,m}^a)_{n,m \in \mathcal{N}_k}$ , where  $p_{k,n,m}^a$  is the probability for a user of evolving from state  $n$  to state  $m$  if action  $a$  is selected at a time epoch. The one-slot *transition probability matrices* are

$$\mathbf{P}_k^0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & q_{k,1,1} & \cdots & q_{k,1,N_k} \\ 0 & q_{k,2,1} & \cdots & q_{k,2,N_k} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & q_{k,N_k,1} & \cdots & q_{k,N_k,N_k} \end{pmatrix},$$

$$\mathbf{P}_k^1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \mu_{k,1} & \tilde{\mu}_{k,1}q_{k,1,1} & \cdots & \tilde{\mu}_{k,1}q_{k,1,N_k} \\ \mu_{k,2} & \tilde{\mu}_{k,2}q_{k,2,1} & \cdots & \tilde{\mu}_{k,2}q_{k,2,N_k} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{k,N_k} & \tilde{\mu}_{k,N_k}q_{k,N_k,1} & \cdots & \tilde{\mu}_{k,N_k}q_{k,N_k,N_k} \end{pmatrix},$$

where we have denoted by  $\tilde{\mu}_{k,n} := 1 - \mu_{k,n}$ .

The dynamics of user  $j$  of class- $k$  are captured by the *state process*  $X_j(\cdot)$  and the *action process*  $a_j(\cdot)$ , which correspond to state  $X_j(t) \in \mathcal{N}_k$  and action  $a_j(t) \in \mathcal{A}$  at all time epochs  $t \in \mathcal{T}$ . At time slot  $t$  the choice of action  $a_j(t)$  in user state  $X_j(t)$  entails the consumption of the allocated capacity (work), the gain of the reward and the evolution of the state to  $X_j(t+1) \in \mathcal{N}_k$ .

### 3.2. Restless Bandit Optimization Problem

We consider the following optimization problem for a job-channel-user defined as a restless bandit. Let  $\Pi_{X_k, a_k}$  be the space of all randomized, history dependent and non-anticipative policies, depending on the state-process  $X_k(\cdot)$  and deciding the action-process  $a_k(\cdot)$ .

Let  $\mathbb{E}_0^\pi$  denote the expectation over the future state process  $X_k(\cdot)$  and the action process  $a_k(\cdot)$ , conditioned on the initial state  $X_k(0)$  and on the policy  $\pi \in \Pi_{X_k, a_k}$ . For a given discount factor  $\beta$  and for every value of price  $\nu$  paid in every slot when serving, the aim is to find a policy minimizing the expected total discounted waiting cost over an infinite horizon,

$$\max_{\pi \in \Pi_{X_k, a_k}} \sum_{t=0}^{\infty} \beta^t \mathbb{E}_0^\pi \left[ R_{k, X_k(t)}^{a_k(t)} - \nu W_{k, X_k(t)}^{a_k(t)} \right]. \quad (2)$$

We remark that this optimization problem is for a single user and independent of all the others. Nevertheless, its solution will be the basis of our multi-user scheduling policy.

#### 4. Index-Based Solution

In this section we address the single-user subproblem (2) for a generic user, so we drop the user subscript  $k$  to simplify the notation.

##### 4.1. Index Values and Threshold Policies

Let us adapt to our setting the definition of *index values* and *indexability*, following [Jacko \(2010\)](#).

**Definition 1** (Indexability). *We say that problem (2) is indexable if there exist values  $\nu_n^* \in \mathbb{R} \cup \{-\infty, \infty\}$  for all  $n \in \mathcal{N}$  such that*

1. *it is optimal to serve the user in state  $n$  if  $\nu_n^* \geq \nu$ , and*
2. *it is optimal not to serve the user in state  $n$  if  $\nu_n^* \leq \nu$ .*

*Such values  $\nu_n^*$  are called the (Whittle) index values, and define an optimal index policy for the problem.*

As described in [Gittins \(1979\)](#); [Whittle \(1988\)](#); [Niño-Mora \(2007\)](#) the optimal solution can in some cases be found by means of index policies. Index values  $\nu_n^*$  are break-even values of the parameter  $\nu$  which define an optimal solution and they can be interpreted as the benefit which is obtained by serving a user in a certain state (dynamic shadow prices). It has been shown that for some non-trivial problems such index values may however not exist, i.e., indexability may not hold.

From the point of view of intuition and implementability, one is often interested in solving the problem by threshold policies.

**Definition 2** (Solvability by threshold policies). *We say that problem (2) is solvable by threshold policies if for any value of  $\nu$ , there exists a threshold state  $n(\nu)$  such that*

1. *it is optimal to serve the user in state  $n$  if  $n \geq n(\nu)$ , and*
2. *it is optimal not to serve the user in state  $n$  if  $n < n(\nu)$ .*

*Such policies are called threshold policies.*

The indexability property is much more general than solvability by threshold policies. Indeed, if the problem is indexable and the index values are non-decreasing in  $n$ , then it is solvable by threshold policies. However, an indexable problem is not solvable by threshold policies if the index values are not non-decreasing in  $n$ ; then the optimal solution may be counter-intuitive and significantly harder to identify.

The restless bandit problem and its index-based solution was introduced in Whittle (1988), generalizing the so-called Gittins index (Gittins and Jones, 1974). Whittle (1988) gave an intuitive definition of indices. An algorithm for computing index values and sufficient indexability conditions were introduced much later; see Niño-Mora (2007) for a survey. The algorithm that computes the index if a problem is indexable values is called *Adaptive-Greedy*, shortly the  $\mathcal{AG}$ -algorithm.

Indexability was established and index values were characterized in closed form in two important special cases in Ayesta et al. (2010); Jacko (2011). We state them below for completeness.

**Theorem 1** (Ayesta et al. (2010)). *If the channel condition evolves in an i.i.d. fashion, i.e.,  $q_{n,m} = q_m$  for each  $n \in \mathcal{N}'$ , then problem (2) is indexable and the index values are*

$$\nu_n^* = \frac{c\mu_n}{1 - \beta + \beta \sum_{m=n+1}^N q_m(\mu_m - \mu_n)}.$$

**Theorem 2** (Jacko (2011)). *If the channel evolves according to the Gilbert-Elliot model, i.e.,  $N = 2$ , then problem (2) is indexable and the index values are*

$$\nu_2^* = \frac{c\mu_2}{1 - \beta} \quad \nu_1^* = \frac{c\mu_1}{1 - \beta + \beta q_{1,2}^*(\mu_2 - \mu_1)},$$

where

$$q_{1,2}^* = \begin{cases} \frac{1}{\frac{\beta(1-\mu_2)}{q_2^{SS}} + \frac{1-\beta(1-\mu_2)}{q_{1,2}}} & \text{if } q_{1,2} > 0, \\ 0 & \text{if } q_{1,2} = 0, \end{cases}$$

and  $q_2^{SS}$  denotes the steady-state probability of channel condition 2, i.e.,

$$q_2^{SS} = \frac{q_{1,2}}{1 + q_{1,2} - q_{2,2}}.$$

Establishing indexability of our general model by the currently known approaches is likely to be technically extremely tedious, as these approaches require to establish certain monotonicity properties, see Niño-Mora (2007) for a survey. However, we believe in its validity based on the computational testing we have performed using the  $\mathcal{AG}$ -algorithm on many random problem instances. We formally conjecture it next.

**Conjecture 1.** *Problem (2) is indexable.*

In the rest of this section we thus focus only on characterization of index values, assuming that Conjecture 1 holds.

#### 4.2. Characterization of the Highest Index Value

It is important to understand which channel condition has the highest index value. The following theorem identifies the highest index value in closed form.

**Theorem 3.** Under [Conjecture 1](#), the index value  $\nu_N^* = \frac{c\mu_N}{1-\beta}$  and we have that  $\nu_N^* \geq \nu_n^*$  for every  $n \in \mathcal{N}'$ .

This result is thus an extension to the Markovian setting of the characterization of the highest index value in the i.i.d. and 2-conditions special cases stated above. What we can see is that the highest index value is *always* associated with the channel condition with the highest transmission rate,  $N$ , and, rather surprisingly, it *always* has a simple expression, which grows to  $+\infty$  as  $\beta \rightarrow 1$ . Note also that the proof of this theorem is valid for any matrix  $\mathbf{Q}$  without restrictions.

#### 4.3. Characterization of the Second Highest Index Value

In the rest of the section we focus on the undiscounted case ( $\beta = 1$ ), which is the most relevant in practice.

Let us state a conjecture concerning the second highest index value. Suppose that the steady-state distribution vector of matrix  $\mathbf{Q}$ , denoted by  $\mathbf{q}^{\text{SS}}$ , exists and is unique. Let for any  $m < N$ ,  $q_m^*$  be the weighted harmonic mean of  $q_N^{\text{SS}}$  (the  $N$ -th element of  $\mathbf{q}^{\text{SS}}$ ) and  $\bar{q}_m$ , (the expected weighted one-slot probability to reach channel condition  $N$  from the set of the remaining channel conditions), defined by

$$q_m^* := \frac{1}{\frac{1-\mu_N}{q_N^{\text{SS}}} + \frac{\mu_N}{\bar{q}_m}}, \quad \bar{q}_m := \sum_{h=1}^{N-1} q_{h,N} p_h^{(m)},$$

where the weights  $p_h^{(m)}$  are the elements of the steady-state probability vector  $\mathbf{p}^{(m)}$  (assumed to exist and be unique) of the  $(N-1) \times (N-1)$  matrix  $\mathbf{Q}^{(m)}$  created from  $\mathbf{Q}$  by merging column  $N$  with  $m$  and omitting row  $N$ ,

$$\mathbf{Q}^{(m)} := \begin{matrix} & 1 & \cdots & m & \cdots & N-1 \\ \begin{matrix} 1 \\ 2 \\ \vdots \\ N-1 \end{matrix} & \left( \begin{array}{cccccc} q_{1,1} & \cdots & q_{1,m} + q_{1,N} & \cdots & q_{1,N-1} \\ q_{2,1} & \cdots & q_{2,m} + q_{2,N} & \cdots & q_{2,N-1} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ q_{N-1,1} & \cdots & q_{N-1,m} + q_{N-1,N} & \cdots & q_{N-1,N-1} \end{array} \right) \end{matrix}.$$

**Conjecture 2.** Under [Conjecture 1](#), suppose that the steady-state distribution vector  $\mathbf{q}^{\text{SS}}$  exists and is unique with  $q_N^{\text{SS}} > 0$ , and that  $\mathbf{p}^{(m)}$  exists and is unique with  $\bar{q}_m > 0$  (hence,  $q_m^* > 0$ ) for all  $m < N$ . Then, the second highest index value in the undiscounted case ( $\beta = 1$ ) is of (every) state  $n$  which satisfies

$$n \in \arg \max_{m \in \mathcal{N}' \setminus \{N\}} \left\{ \frac{c\mu_m}{q_m^*(\mu_N - \mu_m)} \right\},$$

and the index value  $\nu_n^*$  of such state(s)  $n$  is the corresponding maximum (possibly  $+\infty$ ).

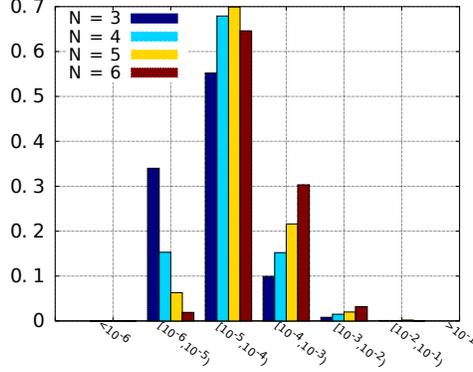


Figure 1: Histogram of the relative error between the second highest index computed by  $\mathcal{AG}$ -algorithm with  $\beta = 1 - 10^6$  and the formula given in [Conjecture 2](#). The distribution has been computed sampling 3000 random matrices  $\mathbf{Q}$  and vectors  $\mu$  (both using the uniform distribution and normalizing). The curves correspond to  $N = 3$  (navy blue),  $N = 4$  (light blue),  $N = 5$  (yellow) and  $N = 6$  (brown).

If this conjecture is true, then we can observe that the expression of the second highest index value and most likely also all the remaining index values might not admit a simple closed-form characterization, and thus might be prohibitively computationally expensive for practical implementation. In the following we give support for [Conjecture 2](#) by showing that it is true in the i.i.d. and in the 2-conditions Markovian special cases, and that it remains numerically precise in the general case as  $N$  grows. In the next subsection we will further prove its validity for the 3-conditions Markovian case.

**Theorem 4.** *[Conjecture 2](#) holds in the following cases:*

1. *If the channel condition evolves in an i.i.d. fashion, i.e.,  $q_{n,m} = q_m$  for each  $n \in \mathcal{N}'$ , then the second highest index value corresponds to state  $N - 1$  and equals*

$$\nu_{N-1}^* = \frac{c\mu_{N-1}}{q_N(\mu_N - \mu_{N-1})}.$$

2. *If the channel evolves according to the Gilbert-Elliot model, i.e.,  $N = 2$ , then the second highest index value corresponds to state  $N - 1 = 1$  and equals*

$$\nu_{N-1}^* = \frac{c\mu_{N-1}}{q_{N-1,N}^*(\mu_N - \mu_{N-1})},$$

where

$$q_{N-1,N}^* = \frac{1}{\frac{1-\mu_N}{q_N^{SS}} + \frac{\mu_N}{q_{N-1,N}}}.$$

[Conjecture 2](#) remains numerically precise in the general case with more than two channel conditions. In [Figure 1](#) we plot the histogram of the relative error between the second highest index computed by  $\mathcal{AG}$ -algorithm with  $\beta = 1 - 10^{-6}$  and the formula given in [Conjecture 2](#)

(which assumes  $\beta = 1$ ). Specifically, the vertical axis represents the probability that such an error belongs to intervals  $[10^{-i}, 10^{-i+1}]$ . We can observe that the precision of the formula is not very sensitive to change in  $N$ , and the relative error most often remains in the interval  $[10^{-6}, 10^{-3}]$ , which we believe is an acceptable numerical error given that the  $\mathcal{AG}$ -algorithm computes inverse matrices. The figure includes the relative error for  $N = 3$ , for which we prove validity of [Conjecture 2](#) in the next subsection.

#### 4.4. Index Values for Channel with Three Quality Conditions

Now we focus on problem (2) in the case  $N = 3$ . We will prove validity of [Conjecture 2](#) and derive closed-form expressions of the index values together with their useful approximations.

Let us first give a closed form characterization of the steady-state vector. Let

$$\begin{aligned} \alpha &= q_{2,1}q_{3,2} - q_{2,2}q_{3,1} + q_{3,1} - q_{1,1}q_{3,2} + q_{1,2}q_{3,1} + q_{3,2} \\ &\quad + q_{1,1}q_{2,2} - q_{1,2}q_{2,1} - q_{1,1} - q_{2,2} + 1. \end{aligned}$$

**Lemma 1.** *If the steady-state vector of matrix  $\mathbf{Q}$  with  $N = 3$  exists and is unique, then it is*

$$\mathbf{q}^{SS} = \frac{1}{\alpha} \begin{pmatrix} q_{2,1}q_{3,2} - q_{2,2}q_{3,1} + q_{3,1} \\ -q_{1,1}q_{3,2} + q_{1,2}q_{3,1} + q_{3,2} \\ q_{1,1}q_{2,2} - q_{1,2}q_{2,1} - q_{1,1} - q_{2,2} + 1 \end{pmatrix}.$$

If [Conjecture 1](#) holds, we already know (by [Theorem 3](#)) that the highest index value is the one associated with state 3. There are therefore two possibilities: the index value of state 2 is greater than that of state 1 (i.e., the problem is solvable by threshold policies), or vice versa.

**Theorem 5.** *[Conjecture 2](#) holds for problem (2) with  $N = 3$ . Thus, if  $\nu_2^* \geq \nu_1^*$  (i.e., it is solvable by threshold policies), then*

$$\nu_2^* = \frac{c\mu_2}{q_2^*(\mu_3 - \mu_2)},$$

where

$$q_2^* := \frac{1}{\frac{1-\mu_3}{q_3^{SS}} + \frac{\mu_3}{q_2}}, \quad \bar{q}_2 := q_{1,3}p_1^{(2)} + q_{2,3}p_2^{(2)},$$

where the weights  $p_1^{(2)} = \frac{q_{21}}{1-q_{11}+q_{21}}$  and  $p_2^{(2)} = \frac{1-q_{11}}{1-q_{11}+q_{21}}$ . If the problem is solvable by threshold policies after relabeling states 1 and 2, then these results hold as well (for the relabeled states).

We can further characterize the index value of state 1. Let us denote by

$$\begin{aligned} U &= (1 - \mu_2)(1 - \mu_3)\alpha + \mu_3(1 - \mu_2)(1 - q_{2,2} + q_{1,2}) \\ &\quad + \mu_2(1 - \mu_3)(1 - q_{3,3} + q_{1,3}) + \mu_2\mu_3, \end{aligned}$$

$$\begin{aligned} V &= (\mu_2 - \mu_1)[(-q_{1,1}q_{3,2} + q_{1,2}q_{3,1} + q_{3,2})(1 - \mu_3) + q_{1,2}\mu_3] \\ &\quad + (\mu_3 - \mu_1)[(q_{1,1}q_{2,2} - q_{1,2}q_{2,1} - q_{1,1} \\ &\quad - q_{2,2} + 1)(1 - \mu_2) + q_{1,3}\mu_2]. \end{aligned}$$

**Theorem 6.** If [Conjecture 1](#) holds for problem (2) with  $N = 3$  in the undiscounted case ( $\beta = 1$ ) and  $\nu_2^* \geq \nu_1^*$  (i.e., it is solvable by threshold policies), then the index value of state 1 is

$$\nu_1^* = \frac{c\mu_1 U}{V}. \quad (3)$$

Unfortunately, we have not been able to write this formula in a more intuitive form. However, an interesting approximation can be obtained for large jobs.

**Theorem 7.** Let us fix a bound  $M$  such that  $\mu_3 \leq M \leq 1$ , i.e., in view of (1) the expected job size is approximately at least  $\tau s_N/M$  bits. Then we have that the index value of state 1,

$$\nu_1^* = \frac{c(\mu_1 + \mathcal{O}(M^2))}{\sum_{m=2,3} q_m^{SS}(\mu_m - \mu_1) + \mathcal{O}(M^2)}, \quad (4)$$

and of state 2,

$$\nu_2^* = \frac{c\left(\frac{\mu_2}{q_3^{SS}} + \mathcal{O}(M^2)\right)}{\mu_3 - \mu_2}. \quad (5)$$

As a consequence, if  $M$  is small enough so that terms  $\mathcal{O}(M^2)$  can be neglected, then we have the following approximation for the index value of state 1,

$$\nu_1^* \approx \frac{c\mu_1}{\sum_{m=2,3} q_m^{SS}(\mu_m - \mu_1)}, \quad (6)$$

and of state 2,

$$\nu_2^* \approx \frac{c\mu_2}{q_3^{SS}(\mu_3 - \mu_2)}. \quad (7)$$

This characterization is nothing but the index value in the i.i.d. setting (cf. [Theorem 1](#)), where the steady-state distribution is employed while the underlying Markovian channel evolution is irrelevant. The precision of this approximation is excellent for large jobs, as showed in [Table 1](#) for channel condition 1. Both the absolute error and the relative error increase approximately linearly in  $M$ , i.e., decrease hyperbolically in job size.

Note that the larger the job, the smaller the parameter  $M$ , and the precision of this approximation could be interpreted as a sort of time-scale separation effect arising naturally in the solution: the *steady-state channel distribution* approximates well in which channel conditions the job will be served, whereas for shorter jobs the Markovian channel evolution may be more important indicating which channel condition is achieved first if starting from the current condition. However, note that this phenomenon differs from the time-scale separation as often simplistically assumed in other literature, which implies that the jobs realize the *time-average throughput*, see [Aalto et al. \(2011\)](#).

$M$	Absolute Error	Relative Error	$\varepsilon$
1	0.3880	14.08%	$+\infty$
0.5	0.1854	7.424%	0.16667
0.3	0.1273	4.498%	0.04286
0.1	0.0399	1.571%	0.00370
0.05	0.0237	0.828%	0.00088
0.01	0.0051	0.176%	0.00003
0.001	0.0005	0.017%	0.00000

Table 1: Mean absolute and relative errors of the approximation (6) in a sample of 2000 job-channel-user instances for each upper bound  $M \geq \mu_3$ .

#### 4.5. Solvability by Threshold Policies for Channel with Three Quality Conditions

We have given in the previous subsection formulae for computing the index values of a 3-conditions channel assuming they are solvable by threshold policies in the undiscounted case ( $\beta = 1$ ). We will give now two sufficient conditions for having such property satisfied and we will observe that they are satisfied in a large number of problem instances.

**Theorem 8.** *If [Conjecture 1](#) holds for problem (2) with  $N = 3$  in the undiscounted case ( $\beta = 1$ ), then we have that  $q_{13} \geq q_{23}$  implies that the index value of state 2 is greater than or equal to the index value of state 1, i.e., the problem is solvable by threshold policies.*

This fact seems quite evident, indeed  $q_{13} \geq q_{23}$  means that the one-slot probability to move to any better channel condition is surely higher if the user is in condition 1 than in condition 2. Observe that this inequality is satisfied for the i.i.d. special case, recovering again the result of solvability by threshold policies by [Ayesta et al. \(2010\)](#).

**Theorem 9.** *Let us denote by  $\Delta := \min\{\mu_3 - \mu_2, \mu_2 - \mu_1\}$ , let  $M$  be such that  $1 > M \geq \mu_3$  and denote by*

$$\varepsilon := \frac{M^2}{3(1 - M)}.$$

*If [Conjecture 1](#) holds for problem (2) with  $N = 3$  in the undiscounted case ( $\beta = 1$ ), then we have that  $\Delta \geq \varepsilon$  implies that the index value of state 2 is greater than or equal to the index value of state 1, i.e., the problem is solvable by threshold policies.*

In the last column of [Table 1](#) we show how small could  $\Delta \geq \varepsilon$  be given a range of upper bounds  $M$ . This condition seems to be really strong if  $M$  is small, which is the requirement to employ the approximation of  $\nu_1^*$  in the previous subsection. We emphasize that this is still quite a rough sufficient condition (see the proof). Finally, we remark that the counter-intuitive case that the index value of state 1 is greater than the index value of state 2 happens in our numerical testing with frequency of around 2.5% for  $\beta = 0.999$ .

## 5. Special Cases and Examples

We now complement the results of the previous section by some examples and special cases. All the results reported are for the undiscounted case ( $\beta = 1$ ).

### 5.1. Examples of General Channel

Interesting simplifications can be obtained when the channel dynamics is relatively sparse, as we illustrate in the following three examples.

*Example 1: Channel with Birth-Death Dynamics.* We consider the channel with  $N$  conditions and birth-death dynamics given by matrix

$$\mathbf{Q} = \begin{pmatrix} y+x & z & 0 & \dots & 0 \\ x & y & z & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & x & y & z \\ 0 & \dots & 0 & x & y+z \end{pmatrix},$$

with  $x, y, z > 0$ . Note that standard results on  $M/M/1/K$  system yield that

$$q_N^{\text{SS}} = \begin{cases} \frac{1 - \frac{x}{z}}{1 - \left(\frac{x}{z}\right)^N} & \text{if } x \neq z \\ \frac{1}{N} & \text{if } x = z. \end{cases} \quad (8)$$

Numerical testing with the AG-algorithm reveals that, although indexable, this kind of problem is not always solvable by threshold policies. For instance, if  $c = 1, N = 3, \mu_1 = 0.195, \mu_2 = 0.2, \mu_3 = 0.3$  and

$$\mathbf{Q} = \begin{pmatrix} 0.65 & 0.35 & 0 \\ 0.35 & 0.30 & 0.35 \\ 0 & 0.35 & 0.65 \end{pmatrix},$$

the AG-algorithm (with  $\beta = 1 - 10^{-8}$ ) gives index values  $\nu_1 = 14.80, \nu_2 = 13.02, \nu_3 = 30'000'000$ . Note that the departure probabilities in channel conditions 1 and 2 are very similar, but the potential improvement from condition 2 is much greater than from condition 1, resulting in a lower index value in condition 2.

The following proposition characterizes the second highest index value when it is associated with channel condition  $N - 1$ .

**Theorem 10.** *Under [Conjecture 2](#), if the second highest index value is associated with channel condition  $N - 1$ , then*

$$\nu_{N-1}^* = \frac{c\mu_{N-1}}{q_{N-1}^*(\mu_N - \mu_{N-1})}$$

where

$$q_{N-1}^* = \begin{cases} \frac{1 - \frac{x}{z}}{\left(1 - \left(\frac{x}{z}\right)^N\right) (1 - \mu_N) + \left(1 - \left(\frac{x}{z}\right)^{N-1}\right) \frac{\mu_N}{z}} & \text{if } x \neq z \\ \frac{1}{N(1 - \mu_N) + (N - 1) \frac{\mu_N}{z}} & \text{if } x = z. \end{cases}$$

*Example 2.* Suppose that channel condition  $N$  merged with  $N-1$  is absorbing (so,  $q_{N-1,N-1} + q_{N-1,N} = 1$ ) and accessible from all conditions  $h < N-1$ , then  $p_{N-1}^{(N-1)} = 1$  and  $p_h^{(N-1)} = 0$  for all  $h < N-1$ , hence  $\bar{q}_{N-1} = q_{N-1,N}$ , and therefore we can simplify

$$\frac{1}{q_{N-1}^*} = \frac{1 - \mu_N}{q_N^{\text{SS}}} + \frac{\mu_N}{q_{N-1,N}}.$$

*Example 3.* If the only possibility to reach channel condition  $N$  is from  $N-1$ , then we can simplify  $\bar{q}_{N-1} = q_{N-1,N} p_{N-1}^{(N-1)}$ .

## 5.2. Examples of Channel with Three Quality Conditions

First, we study two cases where matrix  $\mathbf{Q}$  is periodic.

*Example 4.* Let

$$\mathbf{Q} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \quad \text{i.e.,} \quad \mathbf{q}^{\text{SS}} = \frac{1}{3}(1, 1, 1).$$

We can prove that the problem is solvable by threshold policies and the index values are

$$\begin{aligned} \nu_2^* &= \frac{c\mu_2}{\frac{(\mu_3 - \mu_2)}{3(1 - \mu_3) + \mu_3}} = \frac{c\mu_2}{\frac{(\mu_3 - \mu_2)}{\frac{1-\mu_3}{q_3^{\text{SS}}} + \frac{\mu_3}{q_2}}}, \\ \nu_1^* &= \frac{c\mu_1}{\frac{(\mu_2 - \mu_1) + (\mu_3 - \mu_1)(1 - \mu_2)}{3(1 - \mu_2) + \mu_2 - \mu_3(1 - \mu_2)}}, \end{aligned}$$

where we have used that  $\bar{q}_2 := q_{1,3}p_1^{(2)} + q_{2,3}p_2^{(2)} = 0 \cdot 0 + 1 \cdot 1 = 1$ .

*Example 5.* Let

$$\mathbf{Q} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad \text{i.e.,} \quad \mathbf{q}^{\text{SS}} = \frac{1}{3}(1, 1, 1).$$

We can prove that the problem is solvable by threshold policies and the index values are

$$\begin{aligned} \nu_2^* &= \frac{c\mu_2}{\frac{(\mu_3 - \mu_2)}{3(1 - \mu_3) + 2\mu_3}} = \frac{c\mu_2}{\frac{(\mu_3 - \mu_2)}{\frac{1-\mu_3}{q_3^{\text{SS}}} + \frac{\mu_3}{q_2}}}, \\ \nu_1^* &= \frac{c\mu_1}{\frac{(\mu_2 - \mu_1)(1 - \mu_3) + (\mu_3 - \mu_1)}{3(1 - \mu_2) + 2\mu_2 - \mu_3(2 - \mu_2)}}, \end{aligned}$$

where we have used that  $\bar{q}_2 := q_{1,3}p_1^{(2)} + q_{2,3}p_2^{(2)} = 1 \cdot 1/2 + 0 \cdot 1/2 = 1/2$ .

Note that the formulae of  $\nu_2^*$  from Examples 1 and 2 are in accordance with [Theorem 5](#). Finally, we illustrate a case when the probability of staying in the same condition is  $1/2$ .

---

**Algorithm 1** Algorithmic scheme of PI\* scheduler

---

At every slot  $t$ ,  
 $C' \leftarrow$  Number of users with uncompleted jobs in their channel condition  $N_k$   
**if**  $C' \geq C$  **then**  
    Serve  $C$  users from among the users in their condition  $N_k$  (breaking ties randomly)  
**else**  
    Serve  $C'$  users in their condition  $N_k$   
    Serve  $C - C'$  users not in condition  $N_k$  with highest index value  $\nu_{k, X_k(t)}^*$  (breaking ties randomly)  
**end if**

---

*Example 6.* Let

$$\mathbf{Q} = \frac{1}{4} \begin{pmatrix} 2 & 2 & 0 \\ 1 & 2 & 1 \\ 0 & 2 & 2 \end{pmatrix}, \quad \text{i.e.,} \quad \mathbf{q}^{\text{SS}} = \frac{1}{4}(1, 2, 1).$$

We can prove that the problem is solvable by threshold policies and the index value is

$$\nu_2^* = \frac{c\mu_2}{\frac{(\mu_3 - \mu_2)}{4(1 - \mu_3) + 6\mu_3}} = \frac{c\mu_2}{\frac{(\mu_3 - \mu_2)}{\frac{1 - \mu_3}{q_3^{\text{SS}}} + \frac{\mu_3}{\bar{q}_2}}},$$

where  $\bar{q}_2 = q_{2,3}p_2^{(N)} = 1/4 \cdot 2/3 = 1/6$ .

## 6. Proposed Schedulers

Now we come back to the original multi-class problem with user arrivals, as described in [section 2](#). We set out to design feasible schedulers for the problem where it is allowed to serve up to  $C$  users in every slot. We are interested in the undiscounted case, which is essentially the case of optimization under the time-average criterion. We will do so motivated by the results obtained in [Theorem 3](#), [Theorem 5](#), [Theorem 6](#), [Theorem 7](#), and [Theorem 9](#).

We define the *Markovian Potential Improvement* (PI\*) scheduler, which is presented in [Algorithm 1](#). Index values are now interpreted as *priorities* for serving (the higher the index value, the higher the priority). However, as we have seen in the previous sections, the index values  $\nu_{k,n}^*$  are likely not to admit a simple closed-form characterization in the general setting, except for  $\nu_{k,N_k}^* = +\infty$ . We have only obtained a closed-form solution for  $N_k = 3$ . Therefore, we propose two approximations of the index values, which give rise to additional two new schedulers for general  $N_k$ .

First, we define the PI<sup>AG</sup> scheduler, which approximates  $\nu_{k,n}^*$  for  $n \in \mathcal{N} \setminus \{N\}$  by running the  $\mathcal{AG}$ -algorithm with  $\beta$  as close as possible to 1 while avoiding numerical instability problems. Note that this algorithm performs  $\mathcal{O}(N_k^3)$  elementary operations, and requires the knowledge of the matrix  $\mathbf{Q}_k$ . Moreover, this algorithm identifies whether threshold policies are optimal, and so these approximated index values may not necessarily be increasing in  $n$ .

Second, we define the  $\text{PI}^{\text{SS}}$  scheduler, which approximates  $\nu_{k,n}^*$  for  $n \in \mathcal{N} \setminus \{N\}$  by the formula

$$\frac{c_k \mu_{k,n}}{\sum_{m>n} q_{k,m}^{\text{SS}} (\mu_{k,m} - \mu_{k,n})}. \quad (9)$$

This approximation is based on conjecturing generalizability of [Theorem 7](#), which requires that  $\mu_{k,N_k} \leq M$ , where  $M$  is small enough so that terms  $\mathcal{O}(M^2)$  can be neglected. It is easy to prove that these approximated index values are increasing in  $n$  and that their computation requires  $\mathcal{O}(N_k)$  elementary operations (once the steady-state distribution  $\mathbf{q}_k^{\text{SS}}$  is known). The knowledge of the matrix  $\mathbf{Q}_k$  is not required; only the steady-state distribution is used, which may be significantly easier and more precise to estimate in practice.

We adopt the name of the potential improvement scheduler introduced in [Ayesta et al. \(2010\)](#), since [Jacko \(2011\)](#) for the 2-state channel and the previous section for the 3-state channel show that the index value is the ratio of the one-slot holding cost saving and the (weighted) potential improvement of the departure probability. This can be seen also as a way of optimally resolving the trade-off between opportunistic scheduling and short-jobs prioritization. We however note that yet another dimension (the Markovian evolution) comes into play and indicates that it may sometimes be better to neither be opportunistic nor give priority to a (myopically) shorter job. We can summarize the features of this rule as that priority is given to those users who are *unlikely to improve their actual transmission rate soon by much*.

The  $\text{PI}^*$  scheduler and both its approximations  $\text{PI}^{\text{AG}}$  and  $\text{PI}^{\text{SS}}$  reduce to a scheduler that is optimal if  $N_k = 1$  for all  $k$ , there is a single server ( $C = 1$ ) and arrivals are arbitrary ([Buyukkoc et al., 1985](#)). They all belong to the family of the best-condition schedulers, which have important stability properties in Markovian setting as shown in [Kim et al. \(2013\)](#).

**Theorem 11.** *In the single server case  $C = 1$ , if  $\mathbf{Q}_k$  is irreducible and aperiodic for every  $k$ , then the  $\text{PI}^*$  scheduler and both its approximations  $\text{PI}^{\text{AG}}$  and  $\text{PI}^{\text{SS}}$  are maximally stable under arbitrary arrivals.*

We believe that maximal stability is true even in the multi-server case and even if  $\mathbf{Q}_k$  is reducible or periodic. In fact, it is straightforward to argue that the stability region is upperbounded even in the case of generally distributed job sizes as follows.

For each class  $k$  we define  $B_k := \lceil b_k / s_{k,N_k} \rceil$ , the number of slots in the best channel condition ( $N_k$ ) needed to complete a job of size  $b_k$  bits. Thus,  $B_k$  is the minimum number of slots that a job of size  $b_k$  must spend in service in order to be completed. We denote by  $\mathbb{E}[B_k]$  (positive integer) the mean of this random variable of class  $k$ . We further define the traffic intensity of class  $k$  as  $\varrho_k := \lambda_k \mathbb{E}[B_k]$ .

**Theorem 12.** *If  $\varrho := \sum_{k \in \mathcal{K}} \varrho_k > C$ , then there is no scheduler that stabilizes the system.*

We believe that any system with  $\varrho := \sum_{k \in \mathcal{K}} \varrho_k < C$  is stabilizable by any best-condition scheduler if  $N_k$  is recurrent for every class  $k$ . We are, unfortunately, unable to conclude

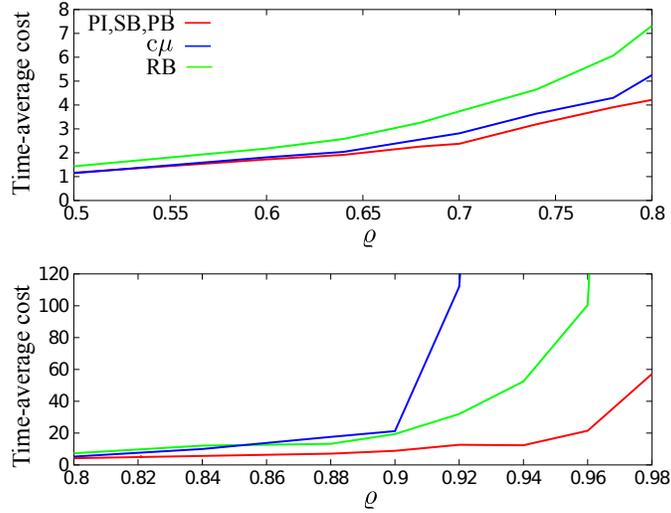


Figure 2: Scenario A: Time-average cost of PI\*, SB, PB (red), RB (green),  $c\mu$  (blue) as a function of  $\rho$ , computed from simulation over 330 sec.

anything with respect to (asymptotic) optimality of the proposed schedulers in systems with arrivals. In the next section we evaluate the performance of PI\* and compare it to existing schedulers proposed for this problem in the previous literature.

## 7. Experimental study

In this section we investigate the behavior and evaluate the performance of the PI\* scheduler and its suggested approximations in simulations. Due to limited space, we have selected two scenarios to present; more experimental results can be found in [Cecchi and Jacko \(2013\)](#). For comparison, we include several schedulers proposed in the previous literature. These schedulers are all priority-based, in the sense that the users served are the ones with highest values that can be interpreted as priorities. However, these alternative schedulers are based on priority values that are *not* Whittle indices, i.e., they have not been shown optimal in the single-user subproblem.

The schedulers we consider are based on the following priority values:

- the  $c\mu$  rule, i.e.  $\nu_{k,n}^{c\mu} = c_k \mu_{k,n}$ ;
- the **Relatively Best** (RB) rule, i.e.  $\nu_{k,n}^{\text{RB}} = \frac{c_k \mu_{k,n}}{\sum_{m=1}^{N_k} q_{k,m}^{\text{SS}} \mu_{k,m}}$ ;
- the **Proportionally Best** (PB) rule, i.e.  $\nu_{k,n}^{\text{PB}} = \frac{c_k \mu_{k,n}}{\mu_{k,N_k}}$ ;

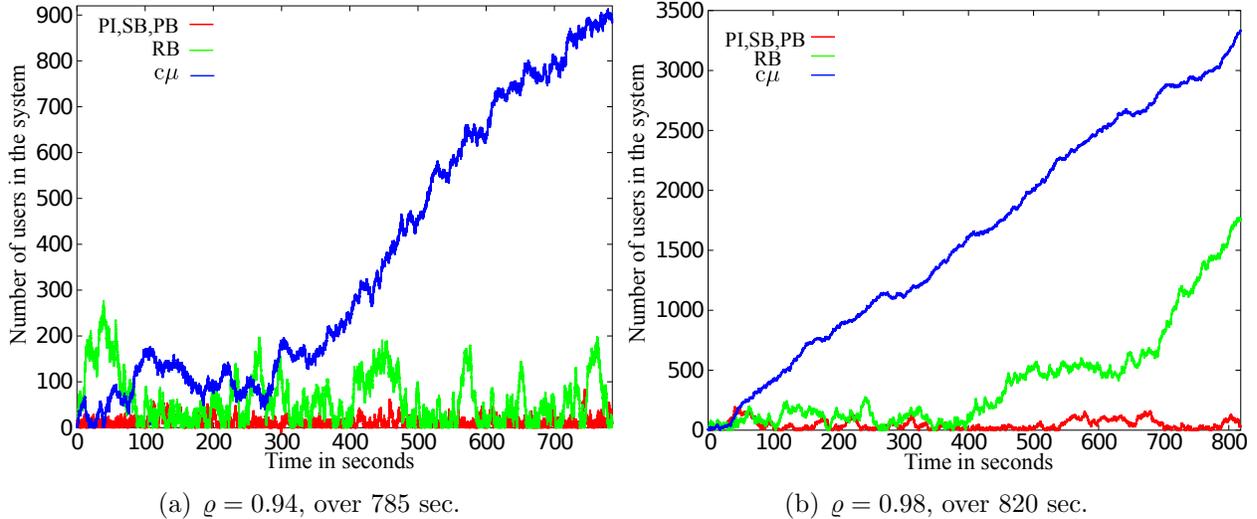


Figure 3: Scenario A: Evolution of the number of users in the system during simulation of  $PI^*$ , SB, PB (red), RB (green), and  $c\mu$  (blue).

- the **Score Based** (SB) rule, i.e.  $\nu_{k,n}^{SB} = c_k \sum_{m=1}^n q_{k,m}^{SS}$ .

We have adapted them to incorporate the waiting costs (the original definitions of RB, PB and SB are recovered with  $c_k = 1$ ). We break the ties randomly.

We focus on the case of capacity  $C = 1$  and  $K = 2$  classes. Each class  $k$  is characterized by a time homogeneous value  $\lambda_k \in [0, 1]$ , representing the probability that a new user belonging to class  $k$  enters the system during a slot. The restriction to Bernoulli arrivals is justified by the short slot duration in real systems, which is typically  $\epsilon = 1.67$ msec.

In order to analyze realistic scenarios, we consider transmission rates  $s_{k,n}$  actually employed in 4G LTE networks, see [Table A.2](#), which is adopted from [Sesia et al. \(2011\)](#). We further consider the following types of typical files downloaded in a wireless data network:

- short files (HTML web page, e-mail) with expected job size  $\mathbb{E}_0[b_k] = 0.5$ Mb (64kB)
- medium files (PDF document, image) with expected job size  $\mathbb{E}_0[b_k] = 5$ Mb (640kB)

For every class of users we select a subset of channel conditions from those defined in [Table A.2](#) and then determine the departure probabilities using (1).

Moreover, in every simulation we vary the value of  $\rho$  between 0.5 and 1, but for simplicity we always keep  $\rho_1 = \rho_2$ . In this way, the rate of arrivals of each class  $k$  is given by  $\lambda_k = \rho_k \mu_{k,N_k}$ . The channel condition of a user at the slot in which she arrives is determined by an equidistributed variable, i.e.,  $q_{k,n} = 1/N_k$ .

Note that the requirement  $c_k = c$  for all  $k$  implies that SB and PB are best condition schedulers. Such a property is guaranteed for the  $c\mu$  and RB rule only under the requirement that the values  $c_k \mu_{k,N_k}$  and  $\nu_{k,N_k}^{RB}$ , respectively, are the same for each user  $k$ . In contrast,

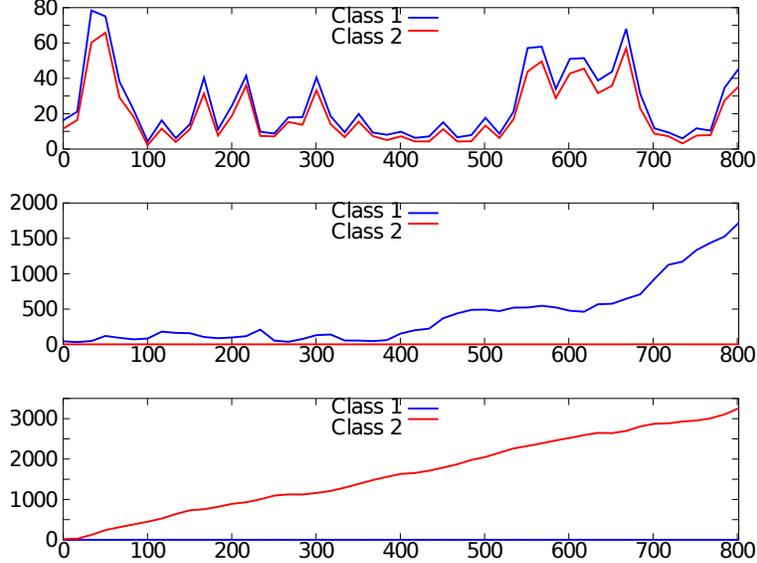


Figure 4: Scenario A: Evolution of the number of users of class 1 (blue) and class 2 (red) during simulation for  $\rho = 0.98$ . The values are averaged over intervals of 10000 slots (16.7 sec.). The rules plotted are PI, SB and PB (top), RB (middle),  $c\mu$  (bottom).

recall that the  $PI^*$  scheduler is a best-condition scheduler always (and therefore maximally stable). The parameters of both scenarios are summarized in [Table A.3](#).

### 7.1. Scenario A

In this scenario, jobs of each class are short. Waiting costs are  $c_1 = c_2 = 1$ , therefore the objective is to minimize the time-average number of users with uncompleted jobs in the system. The channel condition transition matrices are randomly generated, see [Table A.3](#). Class 1 can always be served with a higher transmission rate than class 2:  $s_{1,1} = 53.76\text{Mb/sec}$  while  $s_{2,3} = 33.6\text{Mb/sec}$ . Therefore, we have  $\mu_{1,1} \geq \mu_{2,3}$ , and so the  $c\mu$  rule always gives priority to class 1.

It can be checked that in this scenario the rules  $PI^*$ , SB and PB schedule equally. [Figure 2](#) shows the time-average waiting cost accrued by employing the different policies for varying  $\rho$ . The performance of every policy until  $\rho \leq 0.84$  is quite similar, even if the  $c\mu$  and the  $PI^*$  rules slightly outperform the RB rule. The  $c\mu$  rule becomes unstable between  $\rho = 0.92$  and  $\rho = 0.94$ . [Figure 3](#) indicates that the average increase of users in the system is about 1.2 per second for  $\rho = 0.94$  (note that the average number of arrivals per second is 75.8 for class 1 and 31.5 for class 2). The other rules are stable and the RB and  $PI^*$  rules yield a time-average cost of about 60 and 10, respectively. When  $\rho = 0.98$  (i.e., the average number of arrivals per second is increased to 79 for class 1 and 32.9 for class 2), RB becomes unstable, and the average increase in the number of users under this policy is about 2.1 users per second. This is still better than the average increase under the  $c\mu$  rule, which is of about 4.2 users per second. However, these policies are strongly outperformed by the  $PI^*$  rule: with  $\rho = 0.98$  this rule is stable and the time-average cost is less than 60. In [Figure 4](#) we present

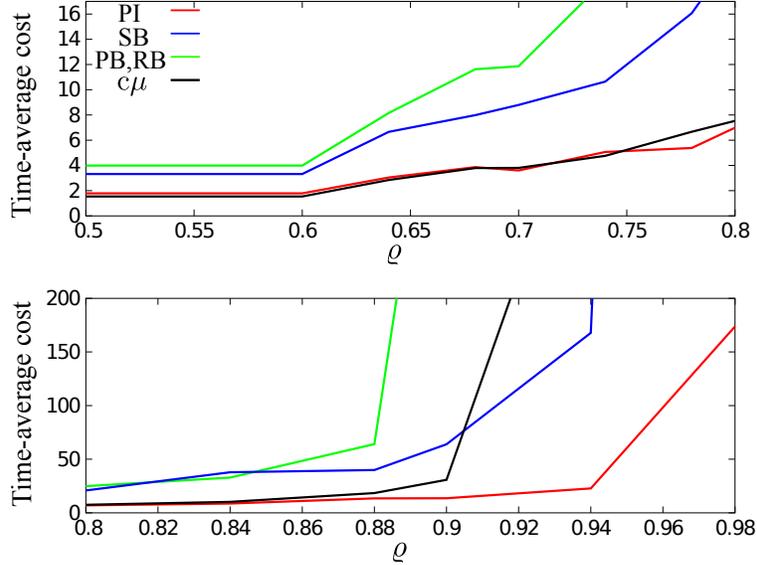


Figure 5: Scenario B: Time-average cost of PI\* (red), SB (blue), RB, PB (green),  $c\mu$  (black) as a function of  $\rho$ , computed from simulation over 330 sec.

the per-class evolution of the number of users in the system. It can be observed that the policies behave in a completely different way. The PI\* rule seems to be quite fair between the two classes, the  $c\mu$  rule gives priority to class 1 (so that class 2 accumulates) while on the contrary the RB favors class 2 (so that class 1 accumulates).

## 7.2. Scenario B

We believe this scenario to be a very realistic situation of two classes representing business and individual customers, respectively. Class 1 requires completion of a job ten times longer than class 2, and its waiting cost is twice the waiting cost of class 2. The remaining parameters are the same for the two classes, see Table A.3. The structure of the channel condition transition matrix is diagonally dominant, i.e. users are more likely to maintain their channel condition than to change it from one slot to another.

The PB and RB rules induce the same policy. Figure 5 shows that the rules  $c\mu$  and PI\* outperform the other policies for  $\rho \leq 0.9$ . All the policies are stable up to  $\rho = 0.9$ , however, the associated time-average costs are quite different. In Figure 6(a) we can see that for  $\rho = 0.94$  (arrivals 7.6 and 75.8 per second), the  $c\mu$  rule is unstable with the average increase of users about 0.3 per second. The other policies are still stable despite the fact that RB queues about 1300 users per slot. This level of users is reached and overcome by the  $c\mu$  rule only after about 1 hour. For  $\rho = 0.98$  (arrivals 8 and 79 per second) the RB rule is also unstable, see Figure 6(b). With this policy, the number of users in the system increases by almost 12 users per slot, which is considerably worse than the increase of 0.95 users per slot that results from the  $c\mu$  rule. The SB and PI\* rules are stable even though they queue respectively around 3000 and 200 users on average. It is insightful to reflect on the way in which the different policies deal with the two classes. Figure 7 elucidates that while SB and

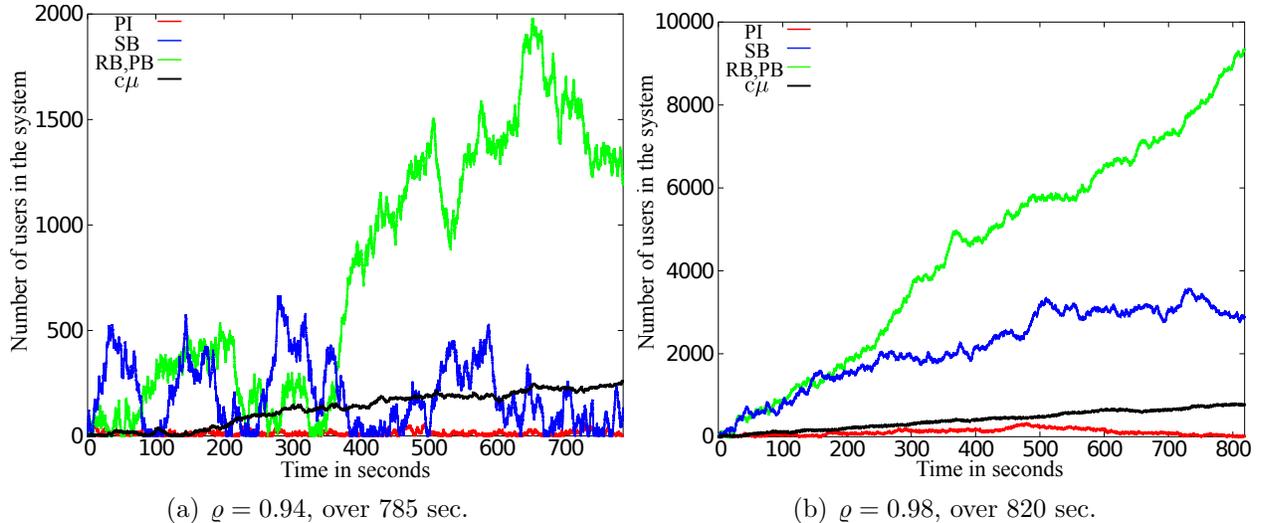


Figure 6: Scenario B: Evolution of the number of users in the system during simulation of PI\* (red), SB (blue), RB, PB (green),  $c\mu$  (black).

PB favour class 1 and the  $c\mu$  rule favours class 2 , the PI\* maintains a balance between the two classes.

## 8. Conclusion

The scheduling problem we are investigating is hard and remains far from being solved to optimality. However, it is motivated by an important practical problem, which explains why the research efforts' focus has been redirected to finding simpler, implementable schedulers. We follow these lines by designing and studying the PI\* scheduler. The introduction of more general and realistic features, like the Markovian evolution of the channel or the arbitrary number of channel conditions, prohibits obtaining closed, intuitive formulae, even for the PI\* priority/index values. In our opinion, an important role in the computation of such index values is played by  $q^*$ , which can loosely be interpreted as the probability of moving to a better condition soon. We elucidate that it is very hard to characterize it mainly due to the Markovian dynamics property. We have addressed the problem in detail for the 3-state case, where, if the jobs are sufficiently large, the PI<sup>SS</sup> scheduler seems to work as an efficient and effective scheduler. We believe that such an approximation is still reasonable for the general  $N$ -state case, though that should be investigated further. In particular, it would be useful to identify sufficient conditions that guarantee that problem (2) is solvable by threshold policies and test whether they are satisfied in practice.

Numerical simulations suggest that the PI\* scheduler works very well in a lot of different scenarios, backed up with the maximal stability property, which assures the system to be manageable whenever possible. Moreover, even though we have not included fairness as an optimization criterion in our MDP model, the PI\* scheduler surprisingly shows to be considerably fair between classes in all loads.

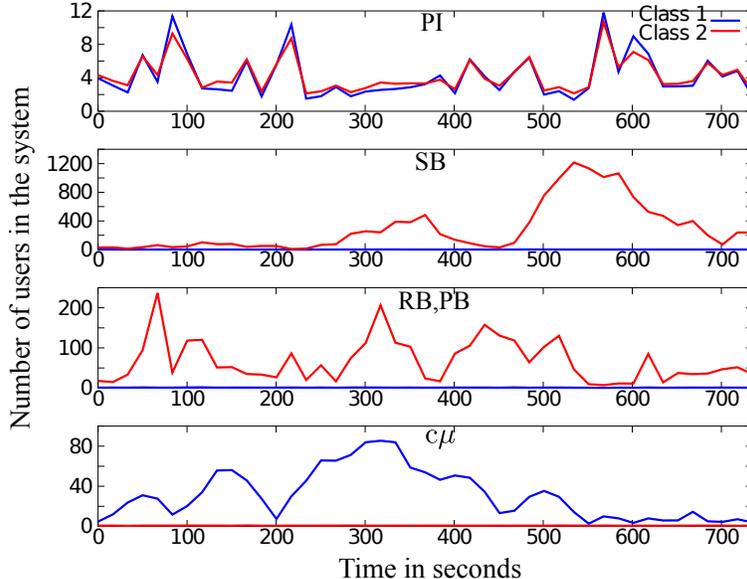


Figure 7: Scenario B: Evolution of the number of users of class 1 (blue) and class 2 (red) during simulation for  $\rho = 0.90$ . The values are averaged over intervals of 10000 slots of time (16.7 sec.).

It is not always possible in practice to know all the parameters characterizing the user classes. For instance, the transition probability matrix is likely to be unknown, but the steady-state distribution can be estimated from past observations at a desired precision. Our simulation experiments indicate that the variant of  $\text{PI}^*$  which only relies on the steady-state distribution of the channel is robust, performs extremely well, and therefore we suggest it to be used for practical implementation.

On the other hand, our simulations (many of them not reported here) indicate that in some specific situations it could be convenient to employ other policies instead of  $\text{PI}^*$ . If the system is relatively lightly loaded, the  $c\mu$  rule, performs sometimes better than the other schedulers in short-term, even if unstable. It would be interesting to identify the stability limit of the  $c\mu$  policy in the Markovian setting in order to alternate between  $c\mu$  and  $\text{PI}^*$ .

The main theoretical limitation of our model is its assumption of geometric job sizes. However, we believe that it is important to first understand this case, which is likely to be the simplest analytically. Future research should address the question of (in)sensitivity of our results to the job size distribution. Note also that the existing maximal stability results for best-condition schedulers (Ayesta et al., 2013; Kim et al., 2013) also rely on the geometric job size assumption.

Modulation	QPSK								16QAM				64QAM		
MCS	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Rate	4.2	6.72	8.4	11.256	16.8	21.84	25.2	26.88	33.6	44.688	50.4	53.76	67.2	75.6	80.64

Table A.2: Transmission rates in Mb/sec associated with the modulation and coding schemes (MCSs) in 4G LTE.

Scenario	MCS	Cost	Channel Condition	Transition Probability Matrix	Job Size
<b>A</b>	({12, 13, 15}, {1, 8, 9})	(1,1)	$\left( \begin{bmatrix} 0.4 & 0.21 & 0.39 \\ 0.48 & 0.5 & 0.02 \\ 0.26 & 0.3 & 0.44 \end{bmatrix} \right)$	$\left( \begin{bmatrix} 0.34 & 0.35 & 0.31 \\ 0.27 & 0.45 & 0.28 \\ 0.45 & 0.15 & 0.4 \end{bmatrix} \right)$	(short, short)
<b>B</b>	({12, 13, 15}, {12, 13, 15})	(2,1)	$\left( \begin{bmatrix} 0.6 & 0.3 & 0.1 \\ 0.25 & 0.5 & 0.25 \\ 0.1 & 0.3 & 0.6 \end{bmatrix} \right)$	$\left( \begin{bmatrix} 0.6 & 0.3 & 0.1 \\ 0.25 & 0.5 & 0.25 \\ 0.1 & 0.3 & 0.6 \end{bmatrix} \right)$	(medium, short)

Table A.3: Scenario parameters of class 1 and class 2.

## Appendix A. Indexability Analysis and Proofs

In the discussion below a crucial point is the resolution to optimality of the problem (2). We observe now that, since this is a standard MDP problem with finite action space and state space, there exists an optimal policy for this problem which is deterministic, stationary and independent of the initial state (Puterman, 2005, Chapter 6). Such an optimal policy can therefore be searched in the set  $\Pi_{X_k, a_k}$  which is the set of stationary policies reliant only on the process relative to user  $k$ .

Since there exists an optimal policy of (2) which is stationary, we could look for such an optimal merely by considering this kind of policies. Indeed the deterministic, stationary policies can be easily represented in terms of their *servicing set*  $\mathcal{S} \subseteq \mathcal{N}_k$ , more precisely, such a policy indicate to serve the user if she is in any state  $n \in \mathcal{S}$ , while not to serve her if  $n \notin \mathcal{S}$ . Hence, it is possible to rewrite the problem (2) in the form

$$\max_{\mathcal{S} \subseteq \mathcal{N}_k} \mathbb{E}_0^{\mathcal{S}} \left[ \sum_{t=0}^{\infty} \beta^t R_{k, X_k(t)}^{a_k(t)} \right] - \nu \mathbb{E}_0^{\mathcal{S}} \left[ \sum_{t=0}^{\infty} \beta^t W_{k, X_k(t)}^{a_k(t)} \right], \quad (\text{A.1})$$

Let us define  $\mathbb{R}_n^{\mathcal{S}} := \mathbb{E}_0^{\mathcal{S}} \left[ \sum_{t=0}^{\infty} \beta^t R_{k, X_k(t)}^{a_k(t)} \right]$  if the initial state is  $X_k(0) = n \in \mathcal{N}_k$ . Analogously we define  $\mathbb{W}_n^{\mathcal{S}}$ .

In the next paragraphs, it will turn out to be useful to have defined the quantity

$$\nu_n^{\mathcal{S}} := \frac{R_n^1 - R_n^0 + \beta \sum_{j \in \mathcal{N}} (p_{nj}^1 - p_{nj}^0) \mathbb{R}_j^{\mathcal{S}}}{W_n^1 - W_n^0 + \beta \sum_{j \in \mathcal{N}} (p_{nj}^1 - p_{nj}^0) \mathbb{W}_j^{\mathcal{S}}}.$$

It represents the rate between marginal reward and marginal work, where the marginal reward (work) is the difference of the expected reward earned (work required) by serving and not serving at the initial state  $n$  and employing policy  $\mathcal{S}$  afterwards.

*Proof of Theorem 3.* In order to prove this theorem, we follow the first step of the  $\mathcal{AG}$  algorithm (Niño-Mora, 2007, see). At the end of the first step the highest index value is assigned to the state  $n$  which leads to a greater value of  $\nu_n^\theta$ . It is straightforward that  $\nu_0^\theta = 0$ . Let's compute such a quantity for  $n \in \mathcal{N}'$ ,

$$\nu_n^\theta = \frac{\mu_n c - \mu_n \beta \left( \sum_{j \in \mathcal{N}'} q_{nj} \mathbb{R}_j^\theta - \mathbb{R}_0^\theta \right)}{1 - \mu_n \beta \left( \sum_{j \in \mathcal{N}'} q_{nj} \mathbb{W}_j^\theta - \mathbb{W}_0^\theta \right)}. \quad (\text{A.2})$$

It is necessary to compute the values of  $\mathbb{R}_j^\theta$  and  $\mathbb{W}_j^\theta$  for every  $j \in \mathcal{N}$ . It is straightforward that  $\mathbb{R}_0^\theta = \mathbb{W}_0^\theta = 0$ . Let us focus on the computation of total reward  $\mathbb{R}_j^\theta$  for  $j \in \mathcal{N}'$ ,

$$\mathbb{R}_j^\theta = \mathbb{E}_j^\theta \left[ \sum_{t=0}^{\infty} \beta^t R_{X(t)}^{a(t)} \right] = -c + \beta \sum_{i \in \mathcal{N}'} q_{ji} \mathbb{R}_i^\theta$$

It can be easily checked that a solution to this linear system is given by

$$\mathbb{R}_j^\theta = -\frac{c}{1-\beta}, \quad \forall j \in \mathcal{N}'.$$

For the total work we have to solve the system

$$\mathbb{W}_j^\theta = \mathbb{E}_j^\theta \left[ \sum_{t=0}^{\infty} \beta^t W_{X(t)}^{a(t)} \right] = \beta \sum_{i \in \mathcal{N}'} q_{ji} \mathbb{W}_i^\theta$$

and so the unique solution of this system is given by

$$\mathbb{W}_j^\theta = 0, \quad \forall j \in \mathcal{N}'.$$

By substituting these quantities in (A.2) it follows that

$$\nu_n^\theta = \frac{c\mu_n}{1-\beta}.$$

We recall that we have supposed that  $\mu_1 < \dots < \mu_N$ . So

$$\nu_N^\theta > \nu_n^\theta > \nu_0^\theta, \quad \forall n \in \mathcal{N}'.$$

We can conclude that

$$\nu_N^* = \nu_N^\theta = \frac{c\mu_N}{1-\beta}.$$

□

*Proof of Theorem 4. Proof of item 1:* It must be proved that  $\nu_{N-1}^{\{N\}} > \nu_n^{\{N\}} \forall n \in \{1, \dots, N-2\}$  and that  $q_{N-1}^* = q_N$ . We observe that

$$\bar{q}_m = \sum_{h=1}^{N-1} p_h^{(m)} q_{h,N} = \left( \sum_{h=1, h \neq m}^{N-1} q_h + (q_m + q_N) \right) q_N = q_N$$

and since  $q_N^{\text{SS}} = q_N$ , we have proved that  $q_n^* = q_N \quad \forall n \in \{1, \dots, N-1\}$ . The result follows from the assumption  $\mu_{N-1} > \mu_m$  for all  $m \in \{1, \dots, N-2\}$ .

*Proof of item 2:* It is straightforward observe that our results generalize this one, indeed  $\bar{q}_1^* = p_1^{(1)} q_{1,2} = q_{1,2}$ , since the  $1 \times 1$ -matrix reduces to (1).  $\square$

*Proof of Theorem 5.* The index value can be obtained emulating the  $\mathcal{AG}$ -algorithm and using a symbolic algebra software.  $\square$

*Proof of Theorem 6.* The index value can be obtained emulating the  $\mathcal{AG}$ -algorithm and using a symbolic algebra software.  $\square$

*Proof of Theorem 7.* It is straightforward to obtain that  $\mu_1 U = \mu_1 \alpha + (\mathcal{O}(M^2))$  and

$$\begin{aligned} V &= (\mu_2 - \mu_1)(q_{1,2}q_{3,1} - q_{1,1}q_{3,2} + q_{3,2}) \\ &\quad + (\mu_3 - \mu_1)(-q_{1,2}q_{2,1} + q_{1,1}q_{2,2} - q_{2,2} - q_{1,1} + 1) + (\mathcal{O}(M^2)). \end{aligned}$$

This gives the approximation for state 1, since  $q_2^{\text{SS}} = (q_{1,2}q_{3,1} - q_{1,1}q_{3,2} + q_{3,2})/\alpha$  and  $q_3^{\text{SS}} = (-q_{1,2}q_{2,1} + q_{1,1}q_{2,2} - q_{2,2} - q_{1,1} + 1)/\alpha$ .

By expanding the terms of  $\nu_2^*$  we can similarly obtain the approximation for state 2.  $\square$

Next we prove two easy-to-check theorems that guarantee the solvability of (2) with 3 states by threshold policies. In order to prove these theorems we have just to verify that the conditions given are sufficient to determine  $\nu_2^{\{3\}} \geq \nu_1^{\{3\}}$ . Such values arise from the second step of the  $\mathcal{AG}$ -algorithm, to be more precise these values are given by

$$\nu_n^{\{3\}} := \frac{R_n^1 - R_n^0 + \beta \sum_{j \in \mathcal{N}} (p_{nj}^1 - p_{nj}^0) \mathbb{R}_j^{\{3\}}}{W_n^1 - W_n^0 + \beta \sum_{j \in \mathcal{N}} (p_{nj}^1 - p_{nj}^0) \mathbb{W}_j^{\{3\}}}.$$

It is possible to rewrite these quantities in an equivalent simpler form, so that

$$\nu_2^{\{3\}} = \frac{c\mu_2}{\mu_3 - \mu_2} \left( \frac{1 - \mu_3}{q_3^{\text{SS}}} + \frac{\mu_3}{\bar{q}_2} \right) \quad (\text{A.3})$$

and

$$\nu_1^{\{3\}} = \frac{c\mu_1}{\mu_3 - \mu_1} \left( \frac{1 - \mu_3}{q_3^{\text{SS}}} + \frac{\mu_3}{\bar{q}_1} \right) \quad (\text{A.4})$$

Note that these expressions are the basis for [Conjecture 2](#).

*Proof of Theorem 8.* It can be checked that

$$\frac{1}{\bar{q}_2} \geq \frac{1}{\bar{q}_1} \Leftrightarrow q_{1,3} \geq q_{2,3}.$$

The theorem follow directly from this fact, indeed

$$\nu_2^{\{3\}} > \frac{c\mu_1}{\mu_3 - \mu_1} \left( \frac{1 - \mu_3}{q_3^{\text{SS}}} + \frac{\mu_3}{\bar{q}_2} \right)$$

$$\geq \frac{c\mu_1}{\mu_3 - \mu_1} \left( \frac{1 - \mu_3}{q_3^{\text{SS}}} + \frac{\mu_3}{\bar{q}_1} \right) = \nu_1^{\{3\}},$$

where the first inequality is due to the assumption  $\mu_2 > \mu_1$ .  $\square$

*Proof of Theorem 9.* The difference  $\nu_2^{\{3\}} - \nu_1^{\{3\}}$  can be expanded through the formulas (A.3) and (A.4), so that we can obtain this sufficient and necessary condition in order to guarantee the positivity of such a difference,

$$\begin{aligned} & \nu_2^{\{3\}} - \nu_1^{\{3\}} \geq 0 \Leftrightarrow \\ \Leftrightarrow & \frac{(1 - \mu_3)(\mu_2 - \mu_1)}{q_3^{\text{SS}}} \geq \frac{\mu_1(\mu_3 - \mu_2)\bar{q}_2 - \mu_2(\mu_3 - \mu_1)\bar{q}_1}{\bar{q}_1\bar{q}_2} \end{aligned} \quad (\text{A.5})$$

The left hand side of (A.5) can be roughly bounded by

$$\frac{(1 - \mu_3)(\mu_2 - \mu_1)}{q_3^{\text{SS}}} \geq (1 - \mu_3)(\mu_2 - \mu_1)$$

while the right hand side can be rewritten and bounded in the following way

$$\begin{aligned} & \frac{\mu_1(\mu_3 - \mu_2)\bar{q}_2 - \mu_2(\mu_3 - \mu_1)\bar{q}_1}{\bar{q}_1\bar{q}_2} \\ &= \frac{\mu_1(\mu_3 - \mu_2)(q_{2,1} + q_{2,3} + q_{1,2}) - \mu_2(\mu_3 - \mu_1)(q_{2,1} + q_{1,3} + q_{1,2})}{q_{1,3}q_{2,1} + q_{2,3}q_{1,2} + q_{2,3}q_{1,3}} \\ &\leq \frac{\mu_3(\mu_1 - \mu_2)(q_{1,2} + q_{2,1}) - \mu_3\mu_2q_{1,3} + \mu_3\mu_1q_{2,3} + \mu_1\mu_2(q_{1,3} - q_{2,3})}{3} \\ &\leq \frac{\mu_1q_{2,3}(\mu_3 - \mu_2) + \mu_1q_{1,3}(\mu_2 - \mu_3)}{3} \leq \frac{\mu_1(\mu_3 - \mu_2)}{3}. \end{aligned}$$

These inequalities provide us a sufficient condition for the difference  $\nu_2^{\{3\}} - \nu_1^{\{3\}}$  to be positive, indeed

$$\frac{\mu_1(\mu_3 - \mu_2)}{3} \leq (1 - \mu_3)(\mu_2 - \mu_1) \Rightarrow \nu_2^{\{3\}} - \nu_1^{\{3\}} \geq 0$$

So that if we suppose for each  $n, m = 1, 2, 3$  and  $n > m$ ,

$$\mu_n - \mu_m > \varepsilon \text{ and } \mu_3 \leq M$$

the sufficient condition is implied by this easy to be verified condition

$$\frac{3(1 - M)\varepsilon}{M^2} \geq 1$$

or equivalently by the hypothesis of the theorem

$$\varepsilon \geq \frac{M^2}{3(1 - M)}.$$

$\square$

*Proof of Theorem 10.* From [Conjecture 2](#) we have

$$\nu_{N-1}^* = \frac{c\mu_{N-1}}{q_{N-1}^*(\mu_N - \mu_{N-1})},$$

where

$$\frac{1}{q_{N-1}^*} = \frac{1 - \mu_N}{q_N^{\text{SS}}} + \frac{\mu_N}{\bar{q}_{N-1}}.$$

Let first  $x \neq z$ . The steady state probability vector of  $\mathbf{Q}$  is given by

$$\mathbf{q}^{\text{SS}} = C_N \left( 1, \frac{z}{x}, \frac{z^2}{x^2}, \dots, \frac{z^{N-1}}{x^{N-1}} \right)$$

where

$$C_N = \frac{1 - \frac{z}{x}}{1 - \frac{z^N}{x^N}} = \frac{x^{N-1}(x - z)}{x^N - z^N}$$

and in particular

$$q_N^{\text{SS}} = \frac{\frac{x}{z} - 1}{\left(\frac{x}{z}\right)^N - 1}.$$

Further, we have that

$$\bar{q}_m = q_{N-1,N} p_{N-1}^{(m)}.$$

The matrix where column  $N$  is merged with column  $N - 1$  has the same birth-death form as matrix  $\mathbf{Q}$ , therefore we have

$$p_{N-1}^{(N-1)} = C_{N-1} \frac{z^{N-2}}{x^{N-2}} = \frac{z^{N-2}(x - z)}{x^{N-1} - z^{N-1}},$$

and

$$\bar{q}_{N-1} = z \frac{\frac{x}{z} - 1}{\left(\frac{x}{z}\right)^{N-1} - 1}.$$

Similarly for  $x = z$ , we have  $q_n^{\text{SS}} = 1/N$ ,

$$p_{N-1}^{(N-1)} = \frac{1}{N - 1},$$

and

$$\bar{q}_{N-1} = q_{N-1,N} p_{N-1}^{(N-1)} = z \frac{1}{N - 1}.$$

□

## References

- Aalto, S., Lassila, P., 2010. Flow-level stability and performance of channel-aware priority-based schedulers, in: Proceeding of NGI 2010 (6th EURO-NF Conference on Next Generation Internet).
- Aalto, S., Penttinen, A., Lassila, P., Osti, P., 2011. On the optimal trade-off between SRPT and opportunistic scheduling, in: Proceedings of ACM Sigmetrics.
- Ayesta, U., Erausquin, M., Jacko, P., 2010. A modeling framework for optimizing the flow-level scheduling with time-varying channels. *Performance Evaluation* 67, 1014–1029.
- Ayesta, U., Erausquin, M., Jonckheere, M., Verloop, I.M., 2013. Scheduling in a random environment: Stability and asymptotic optimality. *IEEE/ACM Transactions on Networking* 21, 258–271.
- Ayesta, U., Jacko, P., 2013. Method for selecting a transmission channel within a time division multiple access (TDMA) communications system. EU Patent.
- Bai, H., Atiquzzaman, M., 2003. Error modeling schemes for fading channels in wireless communications: a survey. *IEEE Communications Surveys & Tutorials* 5, 2–9.
- Bender, P., Black, P., Grob, M., Padovani, R., Sindhushayana, N., Viterbi, A., 2000. CDMA/HDR: a bandwidth-efficient high-speed wireless data service for nomadic users. *IEEE Communications Magazine* 38, 70–77.
- Bonald, T., 2004a. Procédé de sélection de canal de transmission dans un protocole d'accès multiple à répartition dans le temps et système de communication mettant en oeuvre un tel procédé. EU Patent.
- Bonald, T., 2004b. A score-based opportunistic scheduler for fading radio channels, in: *Proceedings of European Wireless*, pp. 283–292.
- Borst, S., 2005. User-level performance of channel-aware scheduling algorithms in wireless data networks. *IEEE/ACM Transactions on Networking* 13, 636–647.
- Buyukkoc, C., Varaiya, P., Walrand, J., 1985. The  $c\mu$  rule revisited. *Advances in Applied Probability* 17, 237–238.
- Cecchi, F., Jacko, P., 2013. Scheduling of users with Markovian time-varying transmission rates. *SIGMETRICS Performance Evaluation Review* 41, 129–140.
- Chaponniere, E.F., Black, P.J., Holtzman, J.M., Tse, D.N.C., 2002. Transmitter directed code division multiple access system using path diversity to equitably maximize throughput. US Patent.

- Chen, C., Heath Jr., R.W., Bovik, A.C., de Veciana, G., 2013. A Markov decision model for adaptive scheduling of stored scalable videos. *IEEE Transactions on Circuits and Systems for Video Technology* 23, 1081–1095.
- Ghandour-Haidar, S., Ros, L., Brossier, J.-M., 2012. On the use of first-order autoregressive modeling for Rayleigh flat fading channel estimation with Kalman filter. *Signal Processing* 92, 601–606.
- Gittins, J.C., 1979. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society, Series B* 41, 148–177.
- Gittins, J.C., Jones, D.M., 1974. A dynamic allocation index for the sequential design of experiments, in: Gani, J. (Ed.), *Progress in Statistics*. North-Holland, Amsterdam, pp. 241–266.
- Jacko, P., 2010. Restless bandits approach to the job scheduling problem and its extensions, in: Piunovskiy, A.B. (Ed.), *Modern Trends in Controlled Stochastic Processes: Theory and Applications*. Luniver Press, United Kingdom, pp. 248–267.
- Jacko, P., 2011. Value of information in optimal flow-level scheduling of users with Markovian time-varying channels. *Performance Evaluation* 68, 1022–1036.
- Kim, J., Kim, B., Kim, J., Bae, Y.H., 2013. Stability of flow-level scheduling with Markovian time-varying channels. *Performance Evaluation* 70, 148–159.
- Knopp, R., Humblet, P., 1995. Information capacity and power control in single-cell multiuser communications, in: *Proceedings of IEEE International Conference on Communications*, pp. 331–335.
- Kushner, H., Whiting, P., 2004. Convergence of proportional-fair sharing algorithms under general conditions. *IEEE Transactions on Wireless Communications* 3, 1250–1259.
- Niño-Mora, J., 2007. Dynamic priority allocation via restless bandit marginal productivity indices. *TOP* 15, 161–198.
- Puterman, M.L., 2005. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., Hoboken, New Jersey.
- Sadeghi, P., Kennedy, R.A., Rapajic, P.B., Shams, R., 2008. Finite-state Markov modeling of fading channels—a survey of principles and applications. *IEEE Signal Processing Magazine* 25, 57–80.
- Sesia, S., Toufik, I., Baker, M., 2011. *LTE-The UMTS Long Term Evolution: From Theory to Practice*. Wiley.
- Wang, H.S., Chang, P.-C., 1996. On verifying the first-order Markovian assumption for a Rayleigh fading channel model. *IEEE Transactions on Vehicular Technology* 45, 353–357.

- Wang, H.S., Moayeri, N., 1995. Finite-state Markov channel—a useful model for radio communication channels. *IEEE Transactions on Vehicular Technology* 44, 163–171.
- Wei, Y., Yu, F.R., Song, M., 2010. Distributed optimal relay selection in wireless cooperative networks with finite-state Markov channels. *IEEE Transactions on Vehicular Technology* 59, 2149–2158.
- Whittle, P., 1988. Restless bandits: Activity allocation in a changing world. *A Celebration of Applied Probability*, J. Gani (Ed.), *Journal of Applied Probability* 25A, 287–298.
- Zheng, K., Liu, F., Lei, L., Lin, C., Jiang, Y., 2013. Stochastic performance analysis of a wireless finite-state Markov channel. *IEEE Transactions on Wireless Communications* 12, 782–793.
- Zorzi, M., Rao, R.R., Milstein, L.B., 1995. On the accuracy of a first-order Markov model for data transmission on fading channels, in: *Proceedings of Fourth IEEE International Conference on Universal Personal Communications (ICUPC)*, pp. 211–215.