Heuristic Approach of Optimal Code Allocation in High Speed Downlink Packet Access Networks

Hussein Al-Zubaidy, Jerome Talim, Ioannis Lambadaris[1]
SCE-Carleton University
1125 Colonel By Drive, Ottawa, ON, K1S 5B6 Canada
Email: {hussein, jtalim, ioannis.lambadaris}@sce.carleton.ca

Abstract— **In this paper, we use the Markov Decision Process (MDP) technique to find the optimal code allocation policy in High-Speed Downlink Packet Access (HSDPA) networks. A discrete stochastic dynamic programming model for the HSDPA downlink scheduler is presented. The model then is solved numerically using value iteration. The system performance when using the resulted optimal policy as compared to Round Robin (RR) is studied using simulation. The behaviour of the value function was observed then used to develop a heuristic scheduling policy. The devised heuristic policy performs very close to the optimal policy. It has much less computational complexity which makes it easy to deploy and with only slight reduction in performance compared to the optimal policy.**

I. Introduction

The rapid development of wireless technology enables the implementation of services which are so far available only on IP based networks. Each service has its own requirements, in terms of bandwidth (Web browsing service for instance), or Quality of Service (QoS) for real-time applications. The new 3G systems (e.g., HSDPA) were designed to have an IP-based infrastructure that enables the reuse of the available IP resources and technologies and in order to reduce the cost [1]. Nevertheless, the added packet switching capability introduced new challenges that have to be dealt with.

One of the challenges is to meet the QoS requirements of the offered services. Wireless links are subject to time- and location-dependent signal attenuation, fading and interference, which will result in bursty errors and time varying channel capacities. Therefore, the direct application of the available wireline QoS techniques is impractical. Furthermore, it is extremely difficult to provide hard (absolute) QoS guarantees and only soft QoS (Differentiated services) can be provided [2]. Packet scheduling is one of the most important QoS control approaches for wireless communications [3]. The scheduling algorithms in wireless systems should take into consideration the variation in channel characteristics, make use of the user diversity to maximize throughput, and aim at providing all users with a fair access to the network.

Scheduling in HSDPA systems involves not only *Transmission Time Intervals (TTI) allocation* but also *codes allocation*. On the downlink, HSDPA uses Code and Time Division Multiplexing (CDM/TDM) and has 15 codes to be alocated per TTI. Most of the available work in scheduler design (e.g. [2]and [4]) is based on intuition and creativity of the designers. This approach can be described as a *procedural approach*. This, most likely, will result in a suboptimal algorithm at the best, that performs well in some scenarios and poor in the others. Another observation is the lack of work on schedulers that dynamically allocate codes as well as TTI for the users in the system.

This work presents a novel approach for scheduling. An analytic model, using stochastic dynamic programming is built to represent the HSDPA scheduler with some realistic assumptions to the rest of the system components. This model is a simplifying abstraction of the real scheduler which estimates system behaviour under different conditions and describes the role of various system components in these behaviours. This model can be solved numerically to obtain the optimal code allocation policy for some given *objective function* in a straight forward manner.

The rest of the paper is organized as follows; section 2 describes the problem. In section 3 we introduce the model. Section 4 presents a two user case study. In section 5, a heuristic policy is presented. In section 6, the performance of the heuristic policy is studied. Conclusions are given in section 7.

II. Problem Definition and Conceptualization

Third generation release R'5 [1], also called High-Speed Downlink Packet Access (HSDPA), is an IP-based network that can offer users a high speed asymmetric radio link with downlink peak bit rate up to 14.4 Mbps [10].

A. HSDPA Downlink Scheduler Abstraction

The HSDPA downlink channel uses a mix of Time Division Multiplexing and Code Division Multiplexing:

- Time is slotted into fixed length 2 ms TTIs.
- During each TTI, there are 15 available codes that may be allocated to one or more users.

During one TTI, the channel capacity associated to one single user depends on the number of allocated codes and on the channel condition. This is mainly due to the fact that HSDPA uses AMC to adapt the transmission rate to the current channel conditions. A mobile user with good channel conditions will experience higher data rate than the other users.

[1]This work was supported by Mathematics of Information Technology and Complex Systems (MITACS) and Natural Sciences and Engineering Research Council of Canada (NSERC).

The diagram in Figure 1 depicts a conceptual realization of the HSDPA downlink scheduler. Different users have

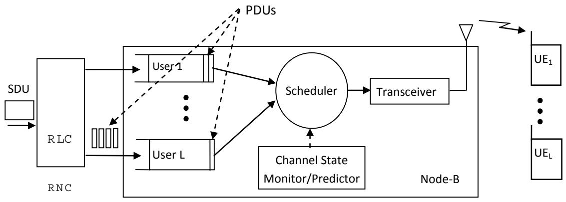


Fig. 1. HSDPA scheduler model (downlink).

separate buffers in the base station (Node-B according to 3GPP), and they are competing for the system resources. Channel state monitor/predictor is necessary to monitor current channel conditions of each user and predict his channel state during the next TTI. This information will then be used to adapt the transmission rate to the expected channel conditions. The arrived Service Data Units (SDU) are assumed to be segmented by the Radio Link Control (RLC) into u_i fixed size Protocol Data Units (PDU) before delivering them to Node-B. The PDUs then will be classified and inserted into the proper buffers awaiting transmission to the intended user. RNC is the Radio Network Controller unit which implements the RLC protocol.

B. HSDPA Downlink Channel Model

The wireless channel for the HSDPA system is modelled as a Finite-State Markov Channel (FSMC) following [7]. This is done by partitioning the signal to noise ratio (SNR) into finite number of intervals, each representing a state in a Markov Chain. Assuming that the fading is slow enough that the channel states for consecutive time epochs are neighbouring states, then the model will be reduced into a discrete time birth and death process, as shown in Figure 2.

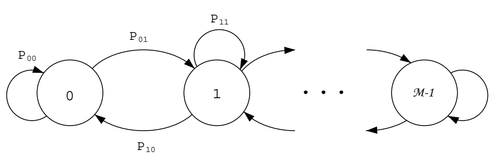


Fig. 2. FSMC model for HSDPA downlink channel.

Depending on the expected SNR, different modulation and error-correcting coding rates can be dynamically selected from a set of Modulation and Coding Schemes (MCS) [5]. The higher the order of the MCS selected the higher the transmission rate. The SNR is mapped directly into MCS and hence into data rates. In light of this, the states in our channel model will equivalently represent data rate levels rather than SNR.

III. Model Description

In this section, we use the approach we presented in [6] to find the optimal code allocation policy for the HSDPA downlink scheduler. We present a general model for this system, based on Markov Decision Process (MDP), and suggest a reward function that achieve the objective function.

To describe a system as a MDP model, the states, actions, rewards and transition probabilities have to be defined first. In our proposed model, time is slotted in constant intervals of size Δt. Let T denote the set of decision epochs of the system, and $T = \{1, 2, \ldots\}$. At time $t \in T$, we define $\boldsymbol{s}(t)$ and $\boldsymbol{a}(\boldsymbol{s})$ as the system state and the action taken at that state. HSDPA downlink scheduler is modelled by the 5-tuple $(T, S, A, P_{\boldsymbol{ss'}}(\boldsymbol{a}), R(\boldsymbol{s}, \boldsymbol{a}))$, where S and A are the state and action spaces, $P_{\boldsymbol{ss'}}(\boldsymbol{a}) = Pr(\boldsymbol{s}(t+1) = \boldsymbol{s'} | \boldsymbol{s}(t) = \boldsymbol{s}, \boldsymbol{a}(\boldsymbol{s}) = \boldsymbol{a})$ is the state transition probability, and $R(\boldsymbol{s}, \boldsymbol{a})$ is the immediate reward when at state $\boldsymbol{s}$ and taking action $\boldsymbol{a}$.

A. Basic Assumptions

There are L active users in the cell. A user $i \in I = \{1, 2, \ldots, L\}$ is allocated a buffer of size B_i. For the sake of simplicity, we will assume that $B_i = B$ for all $i \in I$. Error free transmission is assumed for eliminating the need for retransmission queue. SDUs arrive at the RNC during the current TTI will be segmented by RLC into a fixed number of PDUs (u_i) and delivered to Node-B to be inserted into their respected buffer at the beginning of the next TTI.
For each user $i \in I$ and slot $t \in T$, we define:

- $y_i(t)$ the number of scheduled PDUs,
- $x_i(t) \in \mathcal{X} = \{0, 1, 2, ..., B\}$ the queue size,
- $z_i(t) \in \{0, u_i\}$ the number of arriving PDUs.

The SDUs destined to user i arrives at the RNC during one TTI according to the Bernoulli distribution with parameter q_i. Arrivals are assumed to be independent of the system state and of each other. PDU size is chosen to be equal to the minimum Transport Format and Resource Combination (TFRC) for one code (i.e., one code is needed to transmit one PDU when the channel is in state 1). The scheduler can assign the available 15 codes as chunks of c codes at a time to active users in the system. The chunk size c must divide the total number of codes (15); therefore, $c \in \{1, 3, 5, 15\}$. For example, choosing $c = 5$ means that the policy can assign 0, 5, 10, or 15 of the available 15 codes to any user at any given TTI.

B. FSMC State Space

The channel state of user i during slot t is denoted by $\gamma_i(t)$; and its associated channel state space is the set $\mathcal{M} = \{0, 1, \ldots, M-1\}$, where M is the total number of available channel states. $\mathcal{M}$ constitutes a subset of the available MCS set recommended by 3GPP. The elements of $\mathcal{M}$ were ordered in a way such that $\gamma_i(t)$ is directly proportional to the number of PDUs that can be transmitted by user i in one TTI. This ordering is necessary to reduce computational complexity. Furthermore, we assume that user i channel can handle up to $\gamma_i(t)$ PDUs per code, i.e., a $\gamma_i(t) = 2$ means that at time t, user i can transmit two PDUs using one code and up to 30 PDUs when using all the 15 codes. The Markov transition probability $P_{\gamma_i \gamma_i'}$ is known and can be calculated for any mobile environment with Rayleigh fading channel [7].

C. State and Action Sets

The system state $\boldsymbol{s}(t) \in S$ is a vector comprised of multiple state variables representing the queue sizes and the channel states for the L users. In other word,

$$\boldsymbol{s}(t) = (x_1(t), x_2(t), \ldots, x_L(t), \gamma_1(t), \gamma_2(t), \ldots, \gamma_L(t)) \quad (1)$$

and, $S = \{\mathcal{X} \times \mathcal{M}\}^L$ is finite, due to the assumption of finite buffers size and channel states.

The action space A is the set of all possible actions. The action $\boldsymbol{a}(\boldsymbol{s}) \in A$ is taken when in state $\boldsymbol{s}$. The action taken at each slot corresponds to the number of codes allocated to each user. Let $D = \{0, 1, \ldots, 15/c\}$ be the action space for a single user, where c is the code chunk size (the minimum number of codes that can be allocated at any given time). Let $a_i(\boldsymbol{s}) \in D$ be the number of code chunks allocated to user i when in state $\boldsymbol{s}$. Then the number of codes allocated to user i is $a_i(t)c$. In this case, $\boldsymbol{a}(\boldsymbol{s})$ will be the collection of code allocation to all users, that is

$$\boldsymbol{a}(\boldsymbol{s}) = (a_1(\boldsymbol{s}), a_2(\boldsymbol{s}), \ldots, a_L(\boldsymbol{s})) \quad (2)$$

subject to

$$\sum_{i=1}^{L} a_i(\boldsymbol{s}) \leq \frac{15}{c}, \quad and \quad a_i(\boldsymbol{s}) \leq \left\lceil \frac{x_i(t)}{\gamma_i(t)c} \right\rceil$$

The first constraint means that the policy can not allocate more than the available 15 codes at each time slot. The second makes the policy conserving by allocating no more codes to user i than that required to empty its buffer.

D. Reward Function

In this subsection, we describe the reward function used to determine the optimal allocation policy. As stated previously, the objective is to maximize the throughput while maintaining fairness between active users. Let the *fairness factor*, denoted by σ, be a parameter that reflects the significance of fairness in the optimal policy. Define $\bar{x}$ as the average instantaneous size of the L queues in the system at time t, i.e., $\bar{x} = \frac{1}{L}\sum_{i=1}^{L} x_i$, (we suppressed the time index to simplify notation). The reward function $R(\boldsymbol{s}, \boldsymbol{a})$ will have two components corresponding to the two objectives and it is given by

$$R(\boldsymbol{s}, \boldsymbol{a}) = \sum_{i=1}^{L} a_i \gamma_i c - \sigma \sum_{i=1}^{L} (B - \bar{x}) \, \mathbf{1}_{\{x_i = B\}} \quad (3)$$

where $\mathbf{1}_{\{\cdot\}}$ is the indicator function. The positive term of the reward maximizes the cell throughput. If the reward is composed of this part only, then the policy will always favour the users with good channel conditions. Therefore the users with less favourable channels will starve. That is why we introduced the second term, which guarantees some level of fairness and reduces dropping probability. Lower σ will result in a policy that favours cell throughput over fairness, while higher σ has the opposite effect. Overall, $R(\boldsymbol{s}, \boldsymbol{a})$ will produce a policy that maximizes cell throughput for a given σ.

E. Transition Probability function

$P_{\boldsymbol{ss}'}(\boldsymbol{a})$ denotes the probability that choosing an action $\boldsymbol{a}$ at time t when in state $\boldsymbol{s}$ will lead to state $\boldsymbol{s}'$ at time $t+1$. Using (1) and (2), $P_{\boldsymbol{ss}'}(\boldsymbol{a})$ can be stated as follows

$$\begin{aligned} P_{\boldsymbol{ss}'}(\boldsymbol{a}) &= Pr(\boldsymbol{s}(t+1) = \boldsymbol{s}' | \boldsymbol{s}(t) = \boldsymbol{s}, \boldsymbol{a}(t) = \boldsymbol{a}) \\ &= Pr(x_1', \ldots, x_L', \gamma_1', \ldots, \gamma_L' | x_1, \ldots, x_L, \\ &\qquad \gamma_1, \ldots, \gamma_L, a_1, \ldots, a_L) \end{aligned} \quad (4)$$

The evolution of the queue size (x_i) is given by

$$\begin{aligned} x_i' &= \min\left([x_i - y_i]^+ + z_i', B\right) \\ &= \min\left([x_i - a_i \gamma_i c]^+ + z_i', B\right) \end{aligned} \quad (5)$$

where, z_i' is the arrival to queue i at $t+1$, $[e]^+$ equals e if $e \geq 0$ and 0 otherwise. The channel state γ_i depends only on the previous channel state, that is $Pr(\gamma_i' | \boldsymbol{s}) = Pr(\gamma_i' | \gamma_i) = P_{\gamma_i \gamma_i'}$. Accordingly, we can write (4) as follows

$$P_{\boldsymbol{ss}'}(\boldsymbol{a}) = \prod_{i=1}^{L} \left(P_{x_i x_i'}(\gamma_i, a_i) \, P_{\gamma_i \gamma_i'}\right) \quad (6)$$

where $P_{\gamma_i \gamma_i'}$ is the Markov transition probability of the FSMC. Define $W1$ and $W2$ as follows

$$\begin{aligned} W1 &= [x_i - a_i \gamma_i c]^+ + u_i \\ W2 &= [x_i - a_i \gamma_i c]^+ \end{aligned}$$

We derived $P_{x_i x_i'}(\gamma_i, a_i)$ using (5) and the law of total probability, and arrived at the following expression

$$P_{x_i x_i'}(\gamma_i, a_i) = \begin{cases} 1 & \text{if } x_i' = x_i = B \ \& \ a_i \gamma_i = 0, \\ q_i & \text{if } x_i' = x_i = B \ \& \ 0 < a_i \gamma_i \leq u_i, \\ q_i & \text{if } x_i' = B \ \& \ x_i < B \ \& \ W1 \geq B, \\ q_i & \text{if } x_i' < B \ \& \ x_i' = W1, \\ 1 - q_i & \text{if } x_i' < B \ \& \ x_i' = W2, \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

The first three cases in (7) corresponds to the boundary state, while the remaining cases correspond to the non-boundary states. For complete derivation of the state transition probability see [8].

F. Value Function

In this paper, we investigate an infinite-horizon MDP. We use the total expected discounted reward optimality criterion with discount factor λ, where $0 < \lambda < 1$, in attempt to find the policy π among all policies, that maximize the *value function* $V^\pi(s)$. The following optimality equation is used to characterize the optimal policy [9]

$$V^*(\boldsymbol{s}) = \max_{\boldsymbol{a} \in A} \left[R(\boldsymbol{s}, \boldsymbol{a}) + \lambda \sum_{\boldsymbol{s}' \in S} P_{\boldsymbol{ss}'}(\boldsymbol{a}) V^*(\boldsymbol{s}')\right] \quad (8)$$

where $V^*(\boldsymbol{s})$ is the maximal discounted value function (i.e., $V^*(\boldsymbol{s}) = \sup_\pi V^\pi(\boldsymbol{s})$), attained when applying the optimal policy π^*.

Value iteration (also known as successive approximation) is used to solve this model numerically. The first step is to define $V_0(\boldsymbol{s})$ to be any arbitrary bounded function. Then run the following recursive equation for $n > 0$

$$V_n(\boldsymbol{s}) = \max_{\boldsymbol{a}\in A}[R(\boldsymbol{s},\boldsymbol{a}) + \lambda \sum_{\boldsymbol{s}'\in S} P_{\boldsymbol{ss}'}(\boldsymbol{a})V_{n-1}(\boldsymbol{s}')]$$

V_n converges to V^* as $n \to \infty$ [11]. For a given $\epsilon > 0$, the algorithm can be stopped after n iteration, providing the following

$$\|V_{n+1} - V_n\| < \epsilon(1-\lambda)/2\lambda \tag{9}$$

where $\|v\| = \sup_{s\in S}|v(s)|$. If (9) holds, then $\|V_{n+1} - V^*\| < \epsilon/2$, according to [9].

Using results from the discounted case we can generalize for the infinite horizon average reward using results from [11].

IV. Case Study: Two Users with Two-State Channel

The approach presented earlier was used to model the case when there are two users (i.e., $L=2$) sharing the same cell. The channel is modelled as a two-state FSMC with transition probability matrix

$$\begin{bmatrix} 1-\alpha_i & \alpha_i \\ \beta_i & 1-\beta_i \end{bmatrix} \tag{10}$$

The two user case will simplify the resultant policy and makes it easy to visualize, evaluate, and to deduct conclusions for the optimal policy. It also serves as a verification for the proposed approach, since it may be possible to verify the results for such a case intuitively. The obtained results can then be generalized to more complex cases.

User i is said to be *connected* when $\gamma_i=1$ with probability $P(\gamma_i = 1) = \alpha_i/(\alpha_i + \beta_i)$, and *not connected* ($\gamma_i = 0$) with probability $P(\gamma_i=1)=\beta_i/(\alpha_i + \beta_i)$.

The remaining parameters were chosen as follows: $B=25$, $\sigma=0.5$, $\lambda=0.95$, $\epsilon=0.1$, and c=3, 5 or 15. The action space depends on the value of c. For example, if $c = 5$ then there are *four* possible actions for each user (i.e., $D=\{0, 1, 2, 3\}$) and $A = \{(0,0), (0,1), (0,2), (0,3), (1,0), (1,1), (1,2), (2,0), (2,1), (3,0)\}$, where $\boldsymbol{a} = (a_1, a_2)$ corresponds to a_1c codes assigned to user1 and a_2c codes assigned to user 2. Similarly, when $c = 15$ then there are *two* possible actions per user (i.e., $D = \{0, 1\}$) and when $c = 3$ then there are *six* possible actions per user (i.e., $D=\{0, 1, 2, 3, 4, 5\}$).

The model is solved using value iteration to determine the optimal scheduling policy. The effect of the channel quality and arrival probability on the behaviour of the optimal policy was studied. Figure 3 provide general structure of the optimal policy for c =15 and 3 (see [6] for the case $c = 5$).

The optimal policy for two symmetrical users with the same channel characteristics ($\alpha_i = \beta_i = p$) for all $0 \le p \le 1$ and with $P(z_i = 5) = 0.5$ for all $i \in \{1, 2\}$ is shown in sub-figures 3(a) and 3(d). Only the case when the two users have $\gamma_i=1$ is shown here, since the two users are competing for the system resources. The other three cases when one or both of them has $\gamma=0$ resulted in a policy that assigns all the codes (required) to the connected user and nothing to the other. The optimal policy in this case can be described as follows: *divide the codes between the connected users in proportion to their queue length*. When $c=15$, the action space will be reduced to $A=\{(0, 0), (0, 1), (1, 0)\}$ and the policy will be equivalent to *serve the longest queue first (LQF)*, which makes intuitive sense and matches with the findings in [12] for a case similar to the $c=15$ case.

The effect of the channel quality on the optimal policy structure when $\gamma_1=\gamma_2=1$ is shown in sub-figures 3(b) and 3(e). When $P(\gamma_1=1) > P(\gamma_2=1)$ the policy favours user 2 which is less likely to have $\gamma_2=1$ compared to user 1. The bias in favour of user 2 is depicted in sub-figures 3(b) and 3(e) by a larger dark area, which corresponds to optimal action (0,1) and (0,5) respectively, compared to sub-figures 3(a) and 3(d). We noticed that this bias increases as the difference between $P(\gamma_1=1)$ and $P(\gamma_2=1)$ increases. The reason is that using an LQF in this situation will result in uncontrollable growth in user 2 queue. User 2 will start experiencing unfairness in terms of higher delay and more drops. Hence, more resources have to be assigned to the user with the worst channel to avoid that result. The resource sharing in this case will be governed by the difference $\Delta P_\gamma = P(\gamma_1=1) - P(\gamma_2=1)$ as well as their relative queue length.

The arrival probability has similar effect on the optimal policy structure. The relative increase in one of the users arrival probability will result in more traffic inserted in that user's buffer and it will require more resources to keep the queue length stable and achieve fairness between the two users.

Sub-figures 3(c) and 3(f) shows the optimal policy when $P(z_1=5)=0.8$ and $P(z_2=5)=0.5$ and both users have the same channel quality. The policy shifts in favour of the user with higher arrival probability (user 1 in this case). The shift is proportional to the difference $\Delta P_z = P(z_1=u) - P(z_2=u)$.

V. Near-optimal Heuristic Scheduling Policy

The optimal policy allocates the codes in proportion to the *weighted* queue length of the connected users. We devised a heuristic approach for code allocation in HSDPA system that works in the three cases introduced earlier (i.e., c =15, 5 or 3) and takes into account the channel quality and load variations. It can also be used (with little modifications) for any value of c. The suggested heuristic policy tries to mimic the behaviour of the optimal policy studied in IV.

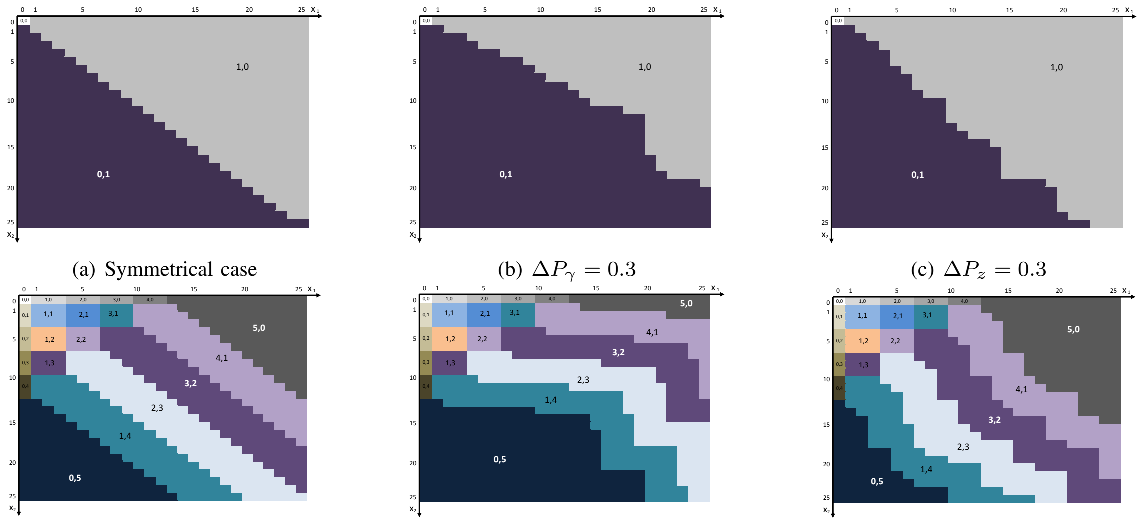
The weight (w_i) is a function of the difference of the two channel qualities and that of the arrival probabilities:

$$w_1 = f([-\Delta P_\gamma]^+, [-\Delta P_z]^+) \tag{11}$$

$$w_2 = f([\Delta P_\gamma]^+, [\Delta P_z]^+) \tag{12}$$

A. The Heuristic Policy for $c = 15$

In this case, the optimal policy is a switch over policy as depicted in Figure 3. We can identify three regions which correspond to the three possible actions: (0,0), (1,0) and (0,1). The heuristic policy is a weighted LQF and it assigns codes to users according to the following rules:



(a) Symmetrical case (b) $\Delta P_\gamma = 0.3$ (c) $\Delta P_z = 0.3$

(d) Symmetrical case (e) $\Delta P_\gamma = 0.3$ (f) $\Delta P_z = 0.3$

Fig. 3. The optimal policy for two user case for $c = 15$ and 3 respectively

- Rule1: when there is only one connected user then assign all the needed codes to that user,
- Rule2: when both users are not connected (i.e., $\gamma_1 = \gamma_2 = 0$) then no codes will be allocated to any user,
- Rule3: when the two users are connected allocate code chunks according to (13) below

$$\boldsymbol{a}(t) = \begin{cases} (1,0) & \text{if } w_1 x_1 > w_2 x_2, \\ (0,1) & \text{if } w_1 x_1 \leq w_2 x_2 \end{cases} \tag{13}$$

B. The Heuristic Policy for $c = 5$

The optimal policy defines ten each of which is characterized by an optimal action as shown in section IV. However, only four of these regions are of interest. They lie within the area where the demand exceeds the available resources. Based on this observation, the heuristic policy partitions the state space into four major regions that corresponds to the actions (3,0), (2,1), (1,2), and (0,3). The heuristic policy defines two additional regions corresponds to actions (0,0) and (1,1) to make the policy conservative. The same heuristic policy in V-A above will apply here except for Rule3 which will be modified as follows:

- Rule3: when the two users are connected, *if* $x_1 + x_2 < 15$ *then* allocate codes to the two users in proportion to their queue length, *else* allocate the code chunks as follows

$$\boldsymbol{a}(t) = \begin{cases} (3,0) & \text{if } w_1 x_1 > w_2 x_2 + 10, \\ (2,1) & \text{if } w_2 x_2 < w_1 x_1 \leq w_2 x_2 + 10, \\ (1,2) & \text{if } w_2 x_2 - 10 \leq w_1 x_1 \leq w_2 x_2, \\ (0,3) & \text{if } w_1 x_1 < w_2 x_2 - 10, \end{cases} \tag{14}$$

C. The Heuristic Policy for $c = 3$

In this case, there are 21 different regions in the state space as shown in Figure 3. The heuristics used in V-A and V-B can be extended to this case. Again only Rule3 need to be modified as shown below

- Rule3: when the two users are connected, *if* $x_1 + x_2 < 15$ *then* allocate codes to the two users in proportion to their queue length, *else* allocate the code chunks as follows

$$\boldsymbol{a}(t) = \begin{cases} (5,0) & \text{if } w_1 x_1 > w_2 x_2 + 12, \\ (4,1) & \text{if } w_2 x_2 + 6 < w_1 x_1 \leq w_2 x_2 + 12, \\ (3,2) & \text{if } w_2 x_2 < w_1 x_1 \leq w_2 x_2 + 6, \\ (2,3) & \text{if } w_2 x_2 - 6 < w_1 x_1 \leq w_2 x_2, \\ (1,4) & \text{if } w_2 x_2 - 12 < w_1 x_1 \leq w_2 x_2 - 6, \\ (0,5) & \text{if } w_1 x_1 \leq w_2 x_2 - 12, \end{cases} \tag{15}$$

D. Weight Function and Other Considerations

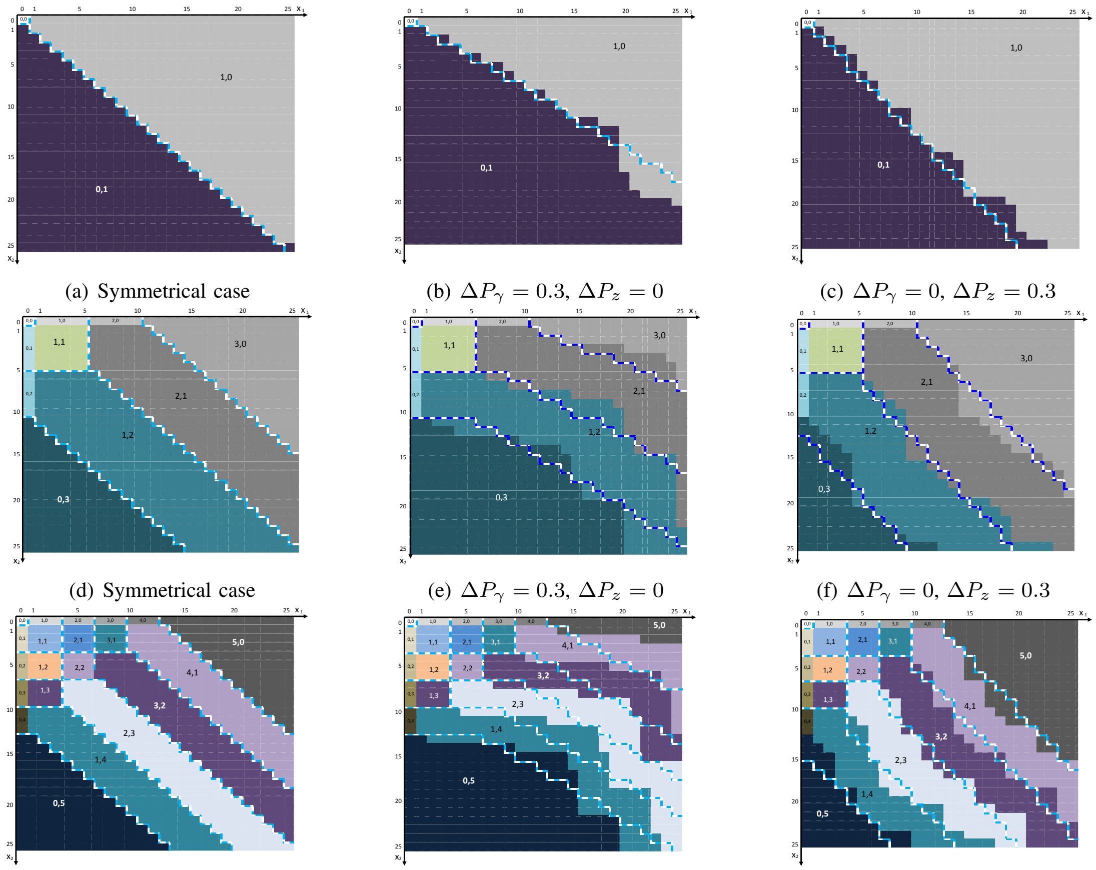
We observed the behaviour of the optimal policy by running a range of scenarios. We noticed that the intermediate regions (e.g., the regions corresponds to actions (1,2) and (2,1) in '$c = 5$' case) has almost a constant width that equals $2c$ in all the scenarios that have been studied. We also noticed that the optimal policy is monotonic and a_1 (respectively a_2) is increasing in x_1 (respectively x_2). It is also apparent from the studied scenarios that $f()$ is increasing in $|\Delta P_\gamma|$ and decreasing in $|\Delta P_z|$. Following these observations, we approximated w_1 and w_2 as follows

$$\hat{w}_1 = 1 + 1.5[-\Delta P_\gamma]^+ - 0.7[-\Delta P_z]^+ \tag{16}$$

$$\hat{w}_2 = 1 + 1.5[\Delta P_\gamma]^+ - 0.7[\Delta P_z]^+ \tag{17}$$

The ratio w_1/w_2 represents the slop of the switchover line between the different areas in the policy. When $\Delta P_\gamma = 0$ and $\Delta P_z = 0$ then $\hat{w}_1/\hat{w}_2 = 1$ and the policy for the three cases will look exactly like the ones in sub-figures 3(a) and 3(d). The suggested heuristic policy can be modified to accommodate classes. This is done by adding a multiplicative parameter to the weight in (16) to implement differentiated services.

Figure 4 show the heuristic policy (the dotted line) superimposed on the optimal policy from section IV for different loading and channel quality conditions. From these graphs, it is fair to say that there is a reasonable convergence between the heuristic policy and the optimal policy.



(a) Symmetrical case (b) $\Delta P_\gamma = 0.3$, $\Delta P_z = 0$ (c) $\Delta P_\gamma = 0$, $\Delta P_z = 0.3$

(d) Symmetrical case (e) $\Delta P_\gamma = 0.3$, $\Delta P_z = 0$ (f) $\Delta P_\gamma = 0$, $\Delta P_z = 0.3$

(g) Symmetrical case (h) $\Delta P_\gamma = 0.3$, $\Delta P_z = 0$ (i) $\Delta P_\gamma = 0$, $\Delta P_z = 0.3$

Fig. 4. The heuristic policy (dotted line) in comparison to the optimal policy for $c = 15, 5$ and 3 respectively

We also noticed that the effect of σ is minimal in this case. This is mainly due to the two-states channel model (connected or not connected). When connected, both users will have the same data rate and serving either one will result in the same reward. However, it is expected that σ will have a prominent role when using FSMC model with more than two states.

VI. Performance Evaluation of The Suggested Heuristic Policy

The performance of the optimal policy and the devised heuristic policy was studied using simulation. The Round Robin fair queueing is used as a baseline. All the assumptions made before is also used in the simulation for consistency. The buffers sizes used in this part is $B_1 = B_2 = 50$.

The system throughput when applying the heuristic policy, RR and the optimal policy is shown in Figure 5. The channel model parameters was chosen such that $P(\gamma_1 = 1) = 0.84$ and $P(\gamma_2 = 1) = 0.5$. Figure 5 shows that the suggested heuristic policy performs very close to the optimal policy. It also shows that RR performance converges to that of the optimal policy in case of light loading (e.g., $\rho = 0.5$). However, it performs up to 30% worse than the optimal policy in heavy load conditions. Where $\rho = \sum_i P_{z_i} u_i / r^\pi$ is the offered load and r^π is the measured system capacity under the policy π.

Queueing delay performance is shown in Figure 6. Figures 7 and 8 show the average queue lengths of both users for the suggested heuristic policy in comparison with that of RR and the optimal policy. From those graphs, the following conclusions were deducted:

- The proposed heuristic policy performance is very close to that of the optimal policy.
- The optimal policy provides the smallest difference in queueing delay between the two users, which means higher fairness level. The heuristic policy provides a comparable performance to that of the optimal policy, while the round robin has the worst fairness and delay performance.
- The performance of the RR policy is highly dependent on the loading conditions. The results obtained proved that RR has poor performance in wireless channel.

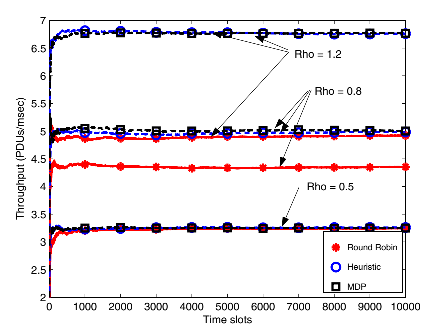


Fig. 5. System throughput for different loading conditions.

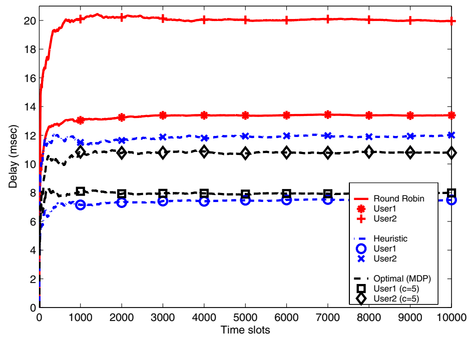


Fig. 6. Queueing delay performance, $P(\gamma_1 = 1) = 0.84$, $P(\gamma_2 = 1) = 0.5$, $q_1 = 0.8$, $q_2 = 0.5$ and $u = 10$.

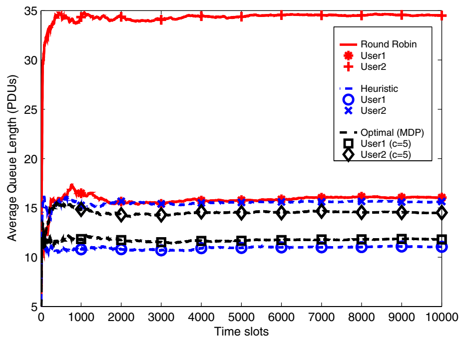


Fig. 7. Queue length, $\rho = 0.75$, $P(\gamma_1 = 1) = 0.84$, $P(\gamma_2 = 1) = 0.5$, $q_1 = 0.5$, $q_2 = 0.5$ and $u = 10$.

The reason why RR performs so poorly in wireless environment is that it does not take into account the channel quality variation, while the optimal policy tracks this variation very closely.

A. Computational Complexity

The approach used is to run the value iteration for a system with small B, to reduce the computation time, then use this model to study the structure of the optimal policy for different channel conditions and loading scenarios. The obtained information is then used to build a heuristic policy that can be expanded to larger buffers sizes. The same approach can be used in the case when more than two users are involved.

The suggested heuristic approach trades performance for simplicity. However, the small performance loss is acceptable price to pay for the huge reduction in computation time. The heuristic policy has deterministic polynomial complexity with constant time complexity, i.e., $O(1)$. On the other hand, the calculation of the optimal policy has an exponential time complexity in B with $O(B^L)$ per one iteration, where L is the number of active users in the system, and is intractable for very large B. The number of iteration required depends on how fast the policy converges, which in turn depends on many other parameters, such as ϵ, λ, and c. Studying the exact complexity for this problem is beyond the scope of this paper.

VII. Conclusion

An MDP model for the scheduling problem in 3G-HSDPA wireless system was developed. Value iteration was used to solve for the optimal scheduling policy for a system with two users and two-states Finite State Markov Channel model. The

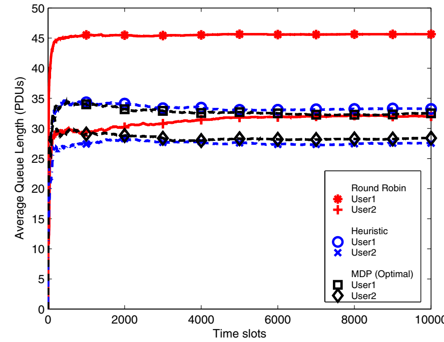


Fig. 8. Queue length, $\rho = 1.1$, $P(\gamma_1 = 1) = 0.6$, $P(\gamma_2 = 1) = 0.6$, $q_1 = 0.8$, $q_2 = 0.5$ and $u = 10$.

policy structure was obtained for different policy granularities. The study showed that the optimal policy can be described as *share the codes in proportion to the weighted queue length of the connected users.*

We developed a heuristic approach to obtain a near-optimal policy. It has a reduced constant time complexity ($O(1)$) as compared to the exponential time complexity needed in the determination of the optimal policy. The performance of the resulted heuristic policy matches very closely to the optimal policy. The results also proved that RR is undesirable in HSDPA system due to the poor performance and lack of fairness if deployed in such environment. The suggested heuristic policy can be extended to the case with more than two active users. It also can be easily adapted to accommodate more than one class of service.

References

[1] 3GPP, *High Speed Downlink Packet Access (HSDPA): Overall Description (Release 5)*. 3GPP Technical specification, TS 25.308, V5.7.0, Dec 2004.

[2] H. Wei, R. Izmailov, *Channel-Aware Soft Bandwidth Guarantee Scheduling for Wireless Packet Access*. IEEE Wireless Communications and Networking Conference (WCNC 04), March 2004.

[3] H. R. Shao, C. Shen, D. Gu, J. Z and P. Orlik, *Dynamic Resource Control for High-Speed Downlink Packet Access Wireless Channel*. IEEE Transaction on Vehicular Technology, vol. 44, no. 1, Feb. 1995.

[4] T. Bonald, *A Score-Based Opportunistic Scheduler for Fading Radio Channels*. Proceeding of the European Wireless Conference, Barcelona, March 2004.

[5] T. E. Kolding, F. Frederiksen and P. E. Mogensen, *Performance Aspects of WCDMA Systems with HSDPA*. IEEE 56th VTC, vol. 1, pp. 47781, 2002.

[6] H. Al-Zubaidy, J. Talim and I. Lambadaris, *Optimal Scheduling Policy Determination for High Speed Downlink Packet Access*. to appear in The The IEEE International Conference on Communications (ICC 2007), Glasgow, Scotland, from 24-28 June 2007.

[7] H. S. Wang and N. Moayeri, *Finite-State Markov Channel–A Useful Model for Radio Communication Channels*. IEEE Transactions on Vehicular Technology, vol. 44, pp. 163171, Feb. 1995.

[8] H. Al-Zubaidy, J. Talim and I. Lambadaris, *Optimal Scheduling in High Speed Downlink Packet Access Networks*. Technical Report no. SCE-06-16, System and Computer Engineering, Carleton University, Nov. 2006.

[9] M. Puterman, *Markov Decision Process: Discrete Stochastic Dynamic Programming*. USA: John wiley & Sons Inc., 1994.

[10] H. Holma and A. Toskala, *WCDMA for UMTS, Radio Access for Third Generation Mobile Communication*, 3rd ed. USA: John wiley & Sons Inc., 2004.

[11] S. M. Ross, *Introduction to Stochastic Dynamic Programming*. USA: Academic Press, 1983.

[12] L. Tassiulas and A. Ephremides, *Dynamic server allocation to parallel queues with randomly varying connectivity*. IEEE Transactions on Information Theory, Vol. 39, Issue 2, Page(s):466 - 478, Mar. 1993.