Dual Degree Dissertation

Stochastic Control for Energy Efficient Resource Alloaction in Wireless Networks

submitted in partial fulfillment of the requirements
for the degree of

Bachelor of Technology
and
Master of Technology
(under the Dual Degree Programme)

by

Abhijeet Bhorkar
Roll No: 01D07014

under the guidance of

Prof. Abhay Karandikar

Department of Electrical Engineering
Indian Institute of Technology
Bombay
July, 2006

Acknowledgments

With a deep sense of gratitude, I would like to thank my adviser, Prof. Abhay Karandikar. He always inspired me to produce upto the potential and has been constant source of encouragement. I enjoyed freedom both in thoughts and research direction while working under him. I am glad to work to him, for he is not only a good researcher, teacher and sharp thinker, but also and most importantly a kind person.

My sincere thanks are due to Prof. Vivek S. Borkar, without whom this thesis would not have been possible. Under him, I was exposed to rigorous research and able to appreciate the works of great researchers.

I would never forget the company of my fellow lab mates of Infonet lab. Hemant Rath has been really cooperative. Discussions with Nitin Salodkar helped me to shape up my thesis. I would like to thank Ashutosh Gore for technical discussions and editing of thesis and paper submissions.

I would like to thank my friends Nitesh Dixit, Chirag Jain, Omkar Kulkarni, Sumit Laad, Nishant Singh for their constant support and making stay at IIT a memorable one.

I thank my parents, my grandparents and my sister.

Abhijeet Bhorkar
June 29, 2006

Abstract

The central theme of the thesis is to design stochastic transmission control algorithms to achieve target Quality of Service (QoS) requirements while considering energy efficiency of the wireless communication system. Towards this we address the problem of scheduling algorithm to provide QoS guarantees like minimum rate, fairness, average and absolute delay, while minimizing the average power required for the transmission. We first consider multiuser single cell system. We propose a centralized power optimal scheduling algorithm, based on stochastic approximation for uplink with minimum rate and fairness constraints. We next formulate the problem for point to point wireless link with finite buffer at the transmitter to provide delay constraints. This problem falls within the framework of constrained Markov Decision Problem. We adopt learning methods like reinforcement learning to design an online algorithm to provide delay guarantees.

Contents

List of Figures

Chapter 1

Introduction

The ubiquitous deployment of wireless networks is not too far! With a century long history of wireless communication, the research in the wireless communication has taken long strides rendering implementable reliable wireless solutions and a cheap alternative to wired networks. Wireless Local Area Networks (WLANs), Fixed Broadband Wireless Access (WiMax) and cellular technologies [1], speak the success stories of wireless communication. However, significant challenges still persist for the wireless networks to become an acceptable solution for widespread deployment. In this thesis, we address some of the unresolved issues. In particular we consider a subset of resource allocation problems which arise in the area of wireless communication networks.

A system capability is highly dependent on the proper utilization of its resources. Similarly for wireless systems, efficiency is dependent upon its physical resources like energy, time and bandwidth. This mandates the upper layer in network stack to also incorporate the effect of physical layer, crumbling the layered architecture. This $cross - layer$ viewpoint has a potential to increase the system efficiency tremendously. In this thesis, we are mainly concerned with the interaction of Media Access Control Layer (MAC) with the physical layer and do not consider the upper layer effects.

The resource allocation problems in networks deals with system level issues like scheduling, flow control, admission control. to provide acceptable services to the users. These Quality of Services (QoS) can be minimum rate guarantee, fairness, average or absolute delay. These allocation problems are handled either by a centralized controller or by distributed algorithm. Examples of networks using centralized scheduler include networks using standards like IEEE 802.16 [1], while distributed networks include sensor networks and networks using standards like IEEE 802.11 [1]. A general class of networks, multi-hop networks, deals with transmission, whereby intermediate nodes can forward the packets towards the destination. The resource allocation problem is further exacerbated by the changing topology, We would be mainly concerned with the single hop networks which forms a basic sub-unit in multi-hop networks.

The solution methods for wireless networks differ significantly from the wired networks because of unique characteristics of wireless networks. Firstly, wireless medium is inherently broadcast medium, which results in inter channel interference issues. This factor comes into picture in multi-hop or cellular networks. Multi-hop networks is beyond the scope of this thesis and we consider Time Division Multiple Access (TDMA) single hop networks. Hence the interference issue does not arise in our formulation . Secondly, randomly varying wireless channel adds another dimension of complexity for managing resources. [1]

In a wireless medium, the channel conditions of mobile users are time varying. This means that the receiver receives the signal with varying power. The fluctuation in the wireless link is attributed to is three independent phenomenon: Path loss, slow log normal shadowing and fast multi-path-fading. Path loss depends on the distance between transmitter and receiver. Slow log normal shadowing is due to diffraction effects and fast fading is due to multi-path reception. In our thesis, we mainly focus on the the time varying channel due to fast fading.

We consider discrete time model for modeling fading. We assume block fading, which means that the channel remains constant during the duration of one block of symbols transmitted during one time slot. We assume no inter-symbol interference (ISI).

Let $\tau(n)$ be the transmitted signal during slot n, The transmitted signal is affected by fading and Gaussian additive noise. Let z_n is the Additive white Gaussian Noise (AWGN) with spectral density $\frac{N_0}{2}$. Then the received signal $r(n)$ is given by,

$$r(n) = \sqrt{x(n)}\tau(n) + z(n), \tag{1.1}$$

where $x(n)$ is the channel gain during n^{th} slot. The expected capacity for a fading channel is given by,

$$C = \mathbf{E}W\log(1 + x(n)\frac{P}{WN_0})^2, \tag{1.2}$$

where W is the channel bandwidth and P is the transmission power.

The wireless nodes rely on limited battery power. Thus, power consumption and power management is imperative in wireless networks. The central theme of the thesis is minimization of average transmission power required for the system to achieve acceptable service for each user. Minimizing average transmission power leads to minimization of overall energy required for the transmission.

The multiuser fading environment yields an important concept of *multiuser* diversity. Multiuser diversity is attributed to the fact that different users perceive difference channel condition and that at any instant there is at least one user with relatively good channel condition. Multiuser diversity can be exploited by “Opportunistic scheduling” that is, for improving throughput, a user with good channel condition must be given chance to transmit. *Opportunistic Scheduling* is a technique to intelligently exploit the channel variation and increase the capacity. By schedul-

[1] Interference and fading can be analyzed simultaneously by modeling interference by a random variable and considering joint random variable for interference and fading

[2] In the subsequent chapters we would not explicitly mention the dependency on W and N_0 and will be concerned with the form log(1+xP)

ing the user which has the best channel condition the overall system throughput is maximized [2] [3]. Methods to artificially introduce fast channel variations or increasing multiuser diversity and thereby maximizing throughput have been proposed in [4]. As also argued in [5], we can utilize the multiuser diversity in the channel along with power control at the transmitter to further increase the throughput. Analogously power control can be used to minimize the system power consumption, a scare resource. The power variation across slots would be the basic means in minimizing the system power requirement.

1.1 Organization of Thesis

The thesis is organized as follows.

In Chapter 2, we first discuss the generalized system model, which we will use for the rest of the thesis. We then formulate the power minimization problem for multiuser system, with rate constraints. We give a formal proof for the optimal solution. We use stochastic approximation method to find Lagrangian associated with the optimal solution. We then prove the conditions required for the optimality and the stability of the stochastic approximation algorithm used. Finally we apply the optimal solution as an extension to the multi-hop networks and multiple channel system. We assume that the transmitter can transmit at continuous rates.

In Chapter 3, we focus on the fairness issues. We take time as the resource to be shared. A scheduler is a long term fair if it can achieve the required fairness criterion in a long run of time. For a short term fair scheduler, it should achieve the objective within a given finite window of time slots. We first consider power optimal long term temporal fairness. We apply the same idea for the short term fairness. Finally we define a novel "Throughput short fair scheduler".

In Chapter 4, we consider a single user system. We first present average delay constrained power optimal solution for finite buffer discrete channel formulation. This method is not scalable enough for large buffer and continuous channel, We, therefore formulate the problem using function approximation method, to address the scalability issue.

In Chapter 5, we deal with devising scheduling policies for real application like video. Here again we use reinforcement learning methods, specifically 'Q learning'. We formulate the problem as minimizing power subject to distortion constraint and absolute delay constraint. We conclude the thesis with summery of results and point future research direction.

In Appendix A, mathematical preliminaries are presented. Appendix B discusses Markov Decision Processes (MDP) and reinforcement learning methods for solving MDPs. We then extend the temporal difference learning algorithm to deal with average cost continuous state space with multiple policies with function approximation.

Chapter 2

Power Optimal Opportunistic Scheduling -Minimum Rate Guarantees

2.1 Introduction

Wireless channels exhibit time varying fading characteristics, which vary from user to user. This multiuser diversity can be exploited by *opportunistically* scheduling the user with the best channel condition. Multiuser diversity has been explored in the pioneering work of Knopp and Humblet [5], where the problem of maximizing the information capacity of the uplink in a single cell environment under an average power constraint has been addressed. Opportunistic scheduling, however, introduces the issue of fairness among users. Proportional fairness in multiuser diversity has been investigated in [6].

In wireless systems, battery and transmission power constraints mandate conservative energy expenditure during transmission. Hence, resource allocation policies have to optimize energy resources subject to Quality of Service (QoS) constraints like minimum rate, delay and fairness. Most of the research work in energy efficient scheduling has focused on a point to point wireless link scenario. [7] provides an overview of energy efficient scheduling under delay constraints.

In this chapter, we consider energy optimal opportunistic control strategies for a multiuser TDMA system subject to minimum rate constraints. Moreover, we propose a stochastic approximation based online algorithm and argue that this method achieves optimality. We also extend the algorithm to consider "temporal" long term fairness as well as short term fairness[1]. [2, 3, 8] have investigated opportunistic scheduling under various types of fairness constraints. However, they do not consider variation in transmission power for energy efficient scheduling.

In [9], the authors have considered an interference-based joint scheduling and power allocation

[1] By temporal fairness, we mean that each user has access to certain number of time slots.

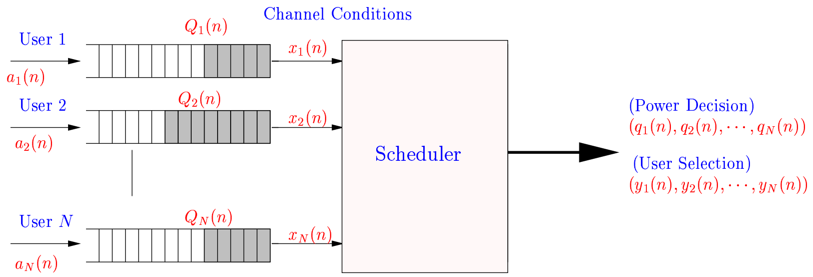


Figure 2.1: Single hop system model

scheme for a multicellular environment. Though their problem formulation is mathematically similar to ours, issues such as the convergence, optimality and stability of the iterative algorithm have not been addressed. Moreover, we have validated our algorithm for independent and identically distributed (i.i.d.) as well as Markovian channel fading.

Bit error rate constrained, power optimal solution for the uplink Code Division Multiple Access (CDMA) system with time varying interference is considered in [10]. However the problem considered is single user power optimal scheduling with varying channel conditions. Our work is different, as we consider multiuser optimal policy for TDMA system. Our work has similarity with [5] where the dual of similar problem is considered, which is maximizing capacity, subject to power constraint. However, the problem considered is from the information theoretic point of view. We consider the dual problem and use a different approach to derive our results. We suggest a stochastic approximation based algorithm to achieve power minimization and prove its optimality.

In Section 2.2, we describe our system model and derive the optimal scheduling policy. In Section 2.3, we describe the stochastic approximation method used to solve the joint Opportunistic and power optimal scheduling problem. In Section 2.4 we provide a detailed proof verifying all the assumptions required for the convergence of the stochastic approximation.

2.2 Optimal Scheduling

In this section we describe the generalized system model, which will be used in this chapter and Chapter 3. Consider a multiuser TDMA system with the base station as the centralized scheduler as shown in Figure 2.2. Time is divided into slots of equal duration. The channel is time varying with slow fading[2]. The channel state at the beginning of slot n is denoted by the vector $(x_1(n), x_2(n), \cdots, x_N(n))$, where $x_i(n)$ denotes the channel gain for user i at slot n and N is the number of users. We assume that the channel state (channel gain) changes only at slot boundaries and perfect channel state information (CSI).

The channel state process $(x_1(n), x_2(n), \cdots, x_N(n))$ is assumed to be $\mathbb{R}^d-$valued and ergodic

[2] Fading is constant over a slot duration.

with marginal distribution ν, where $N \geq d \geq 1$. Channel gains experienced by different users are independent and identically distributed (i.i.d.). The channel state evolution with time can be either i.i.d. or Markovian. The rate requirement of each user is known apriori at the base station.

In any given time slot, only one user is allowed to transmit. The scheduler determines the user who can transmit and its transmission power subject to that user's rate constraint. For each user i, we associate an indicator function $y_i(n)$ which is 1 if user i is scheduled at time slot n, otherwise it is 0. Let $q(n)$ be the actual transmission power of the scheduled user at time slot n. Let $a_i(n)$ be the number of arrivals and $Q_i(n)$ be the queue length at time slot n. Let C_i be the time-average minimum rate requirement for user i and U_i be the utility function for user i, which is increasing and concave in channel gain x_i and power q. We assume $U_i(q, x_i) = \log(1 + qx_i)$, which is equivalent to the information theoretic capacity bound. Our objective is to minimize average power subject to average rate constraints, which can be expressed as:

$$\min \limsup_{M\to\infty} \frac{1}{MN} \sum_{n=1}^{M} \sum_{i=1}^{N} q(n) y_i(n),$$

$$\text{s.t.} \liminf_{M\to\infty} \frac{1}{M} \sum_{n=1}^{M} U_i(y_i(n) q_i(n), x_i(n)) \geq C_i \ \forall i. \tag{2.1}$$

Due to ergodicity, we focus on the transmission policy for any slot and do not explicitly state the dependence of the channel state process on time n. Let $\mathbf{x} = (x_1, \cdots, x_N)$ denote the channel state vector. $A = (\mathbf{e_1}, \cdots \mathbf{e_N})$, where $\mathbf{e_i}$ denotes the unit vector in the i^{th} coordinate direction. Let $\mathbf{y} = (y_1, \cdots, y_N)$ be the vector of indicator random variables. Note that only one of the random variables y_i will be 1 in a given time slot. Let p be the conditional law of $(q, \mathbf{y})$ given $\mathbf{x}$, which can be decomposed as $p_1(dq|\mathbf{y}, \mathbf{x}) p_2(\mathbf{y}|\mathbf{x})$. Thus, we can write the optimization problem (2.1) as:

$$\begin{aligned} \min \int \nu(dx_1, \cdots, dx_N) \sum_{\mathbf{y}\in A} \int_{[0,\infty)} p_1(dq|\mathbf{y}, \mathbf{x}) p_2(\mathbf{y}|\mathbf{x}) q, \\ \text{s.t.} \int \nu(dx_1, \cdots, dx_N) \sum_{\mathbf{y}\in A} \int_{[0,\infty)} p_1(dq|\mathbf{y}, \mathbf{x}) p_2(\mathbf{y}|\mathbf{x}) \\ \log(1 + q y_i x_i) \quad \geq \quad C_i \ \ \forall i, \quad q \geq 0. \end{aligned} \tag{2.2}$$

Proposition 2.1 *The optimal policy is to select user k and transmission power q^*, where*

$$\begin{aligned} k &= \arg\min_i \left\{ \left(\lambda_i - \frac{1}{x_i}\right)^+ - \lambda_i \Bigg[\right. \\ & \left. \log\left(1 + \left(\lambda_i - \frac{1}{x_i}\right)^+ x_i\right) - C_i \Bigg] \right\}, \qquad (2.3) \\ q^* &= \left(\lambda_k - \frac{1}{x_k}\right)^+, \qquad (2.4) \end{aligned}$$

and λ_i is the Lagrange multiplier associated with the rate constraint for user i.

Proof: The Lagrangian associated with (2.2) is

$$\begin{aligned} f(p_1, p_2, \boldsymbol{\lambda}) \triangleq \int \nu(dx_1, \cdots, dx_N) \sum_{\mathbf{y}\in A} \int_{[0,\infty)} p_1(dq|\mathbf{y}, \mathbf{x}) \\ p_2(\mathbf{y}|\mathbf{x}) \left(q - \sum_i \lambda_i \left[\log\left(1 + qy_ix_i\right) - C_i\right]\right) \qquad (2.5) \end{aligned}$$

where $\boldsymbol{\lambda}= (\lambda_1, \cdots, \lambda_N)$. Therefore, the optimization problem decomposes into: minimize with respect to (w.r.t.) $p_1(q|\mathbf{x}, \mathbf{y})$ and then minimize w.r.t. $p_2(\mathbf{y}|\mathbf{x})$. Note that the cost function $f(p_1, p_2, \boldsymbol{\lambda})$ is linear in the joint probability distribution when the marginal distribution of $\mathbf{x}$ is fixed and the minimization is over the conditional distributions. The set of probability distributions with a fixed marginal is a closed convex set with extreme points corresponding to those distributions for which the conditional distributions are point masses [11]. Thus for each $\mathbf{x}$, we minimize over q and $\mathbf{y}$. The Lagrangian (2.5) is strictly convex w.r.t. q and $\mathbf{y}$ and hence the minimizer is unique. Since joint minimization over q and $\mathbf{y}$ can be done in any order, we minimize first with respect to q and then w.r.t. $\mathbf{y}$. Thus we first minimize (2.5) w.r.t. q for a fixed i which corresponds to $\mathbf{y} = \mathbf{e}_i$. The reduced single user min-max problem is:

$$\max_{\lambda_i} \min_q \mathcal{L}(\lambda_i, q) \qquad (2.6)$$

where $\mathcal{L}(\lambda_i, q) = q - \lambda_i(\log(1 + qx_i) - C_i)$. Denote the optimal q for $\mathbf{y} = \mathbf{e}_i$ by q_i. To minimize (2.6) w.r.t. q, we differentiate $\mathcal{L}(\lambda_i, q)$ w.r.t. q,

$$\frac{\partial \mathcal{L}}{\partial q} = 1 - \lambda_i \left(\frac{x_i}{1 + qx_i}\right), \qquad (2.7)$$

leading, by the Kuhn-Tucker Theorem [12], to

$$q_i = \left(\lambda_i - \frac{1}{x_i}\right)^+. \qquad (2.8)$$

Minimizing (2.5) w.r.t. $\mathbf{y}$ yields the optimal policy,

$$\begin{aligned} k &= \arg\min_i \left\{ q_i - \lambda_i \left[\log(1+q_i x_i) - C_i\right] \right\} \\ &= \arg\min_i \left\{ \left(\lambda_i - \frac{1}{x_i}\right)^+ - \lambda_i \left[\right.\right. \\ &\qquad \left.\left. \log\left(1 + \left(\lambda_i - \frac{1}{x_i}\right)^+ x_i\right) - C_i \right]\right\}. \end{aligned} \tag{2.9}$$

The optimal policy is to schedule user k which satisfies (2.9). The scheduled user will transmit with power $q^* = q_k$ as given in (2.8). ■

2.3 Stochastic Approximation based Online Algorithm

In this section, we focus on the on-line optimal policy to estimate parameters $\boldsymbol{\lambda}$ of the algorithm. We use stochastic approximation to implement the policy. The policy and the update equation involved in the algorithm are low in complexity The stochastic approximation algorithm guarantees almost sure (a.s.) convergence to the optimal solution, if certain properties of the update equation and the objective functions are satisfied. We prove that these properties are satisfied in our case and thus the algorithm converges to optimal $\boldsymbol{\lambda}$ with probability (w.p.) 1. We outline informal idea of the stochastic approximation scheme used. We provide formal proof of the optimality of the stochastic approximation algorithm in Section 2.4. Stochastic approximation can be used to determine the optimum solution for a perturbed function (in our case perturbation is channel fading). After minimizing (2.5) over $(q, \mathbf{y})$ in Section 2.2, we maximize over $\boldsymbol{\lambda}$ to obtain the optimal solution. The stochastic gradient ascent scheme for maximization over $\boldsymbol{\lambda}$ is given by,

$$\begin{aligned} \lambda_i(n+1) &= \Gamma\left(\lambda_i(n) - \alpha(n) \left[y_i(n) \log\left(1 + \right.\right.\right. \\ &\qquad \left.\left.\left. \left(\lambda_i - \frac{1}{x_i(n)}\right)^+ x_i(n)\right)\right] - C_i \right) \forall i \end{aligned} \tag{2.10}$$

where[3]:

1. $y_i(n) = I((q_i^* - \lambda_i \left[\log(1+q_i^* x_i) - C_i\right]$. Note that $y_i(n) \leq (q_j^* - \lambda_j[\log(1+q_j^* x_j) - C_j]), j \neq i)$.

2. $\alpha(n)$ is a positive scalar sequence satisfying [13],

$$\sum_n \alpha(n) = \infty, \quad \sum_n \alpha(n)^2 < \infty,$$

[3] $I(a \leq b) = 1$ if $a \leq b$, $= 0$ otherwise.

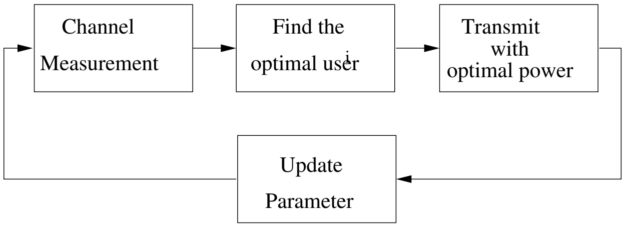


Figure 2.2: Block diagram for on-line policy

3. $\Gamma(\cdot)$ is the projection to the set $[0, L]$ where $L \geq 0$ is a very large but finite number, i.e., $\Gamma(x) = \max(0, \min(x, L))$.

4. We take $\alpha(n) = \frac{l}{n}$, where l, the initial learning rate, is a small constant.

Note that we have assumed the transmission power q to be unconstrained. However, if we impose a constraint $q \leq q_{max}$ for a prescribed $q_{max} < \infty$, then we can replace q^* by $\hat{q}^* = q^* \wedge q_{max}$[4]. In [2, 3, 8], the authors have used stochastic approximation algorithm, but the convergence proof is not discussed. Moreover the technical proof in these algorithms is simple because of the differentiable functions involved. We now sketch the proof of convergence for the stochastic approximation scheme as outlined in (2.10). The details are discussed in Section 2.4 We consider the channel state process to be i.i.d. across slots. The proof of convergence for the Markovian model is along similar lines.

Let $\tilde{y}_i(n) = y_i(n)$ with $\lambda_i(n)$ replaced by λ_i and $E_s[\ \cdot\ ]$ denote the stationary expectation. Rewrite iteration (2.10) as,

$$\lambda_i(n+1) = \Gamma\left(\lambda_i(n) - \alpha(n)\left[h_i\left(\boldsymbol{\lambda}(n)\right) + M_i(n+1)\right]\right),$$

where,

$$\begin{aligned} h_i(\boldsymbol{\lambda}(n)) &= E_s\Bigg[\tilde{y}_i(n)\Bigg(\log\left(1 + \left(\lambda_i - \frac{1}{x_i(n)}\right)^+ x_i(n)\right) \\ &\quad -C_i\Bigg)\Bigg] \,|_{\lambda_i = \lambda_i(n)} \\ M_i(n+1) &= y_i(n)\log\left(1 + \left(\lambda_i(n) - \frac{1}{x_i(n)}\right)^+ x_i(n)\right) \\ &\quad -C_i - h_i\left(\boldsymbol{\lambda}(n)\right). \end{aligned}$$

This iteration will converge w.p. 1 to an invariant set of the differential equation (Ref. [13])

$$\dot{\boldsymbol{\lambda}}(t) = h(\boldsymbol{\lambda}(t)) + \mathbf{z}(t), \tag{2.11}$$

where $h(\cdot) = [h_1(\cdot), \cdots, h_N(\cdots)]$ and $\mathbf{z}(t) = [z_1(t), \cdots, z_N(t)]$ is the boundary correction term due

[4] $a \wedge b = \min(a, b)$

to the projection operator Γ [13]. Note that $h_i(\boldsymbol{\lambda}) \in \partial F(\boldsymbol{\lambda})$, where,

$$\begin{aligned} F(\boldsymbol{\lambda}) &= E_s\Bigg[\min_i \Big\{\Big(\lambda_i - \frac{1}{x_i(n)}\Big)^+ - \lambda_i\Big(\log\Big(\\ &\qquad 1 + \Big(\lambda_i - \frac{1}{x_i(n)}\Big)^+ x_i(n)\Big) - C_i\Big)\Big\}\Bigg] \end{aligned} \tag{2.12}$$

is the point-wise minimum of a family of affine functions of $\boldsymbol{\lambda}$ and is a strictly concave function of $\boldsymbol{\lambda}$. ∂F denotes its superdifferential. This can be verified by invoking the recent extensions of the envelope theorem [14] [15]. Thus the ordinary differential equation (2.11) may be viewed as the differential inclusion

$$\dot{\boldsymbol{\lambda}}(t) \in \partial F(\boldsymbol{\lambda}(t)) + \mathbf{z}(t).$$

Technical details for this are given in Lemma 2.3. This is a supergradient ascent scheme for a strictly concave function and thus will converge to its unique maximum on the constraint set. If L is sufficiently large, this will be the desired vector of Lagrange multipliers by the saddle point theorem [12]. Thus, the iterates (2.10) converge almost surely to the Lagrange multipliers.

2.4 Proofs

In this section we give technical details for the convergence of the stochastic approximation, considered in the previous section.

$F(\boldsymbol{\lambda}) \overset{\Delta}{=} \min_{p_1,p_2} f(p_1, p_2, \boldsymbol{\lambda})$, Let $D_x F$ denote the partial differentiation of F w.r.t. x.

The differential inclusion of F at λ is given by $\partial^+ F(\boldsymbol{\lambda})$, $\partial^+ F$ is upper semicontinuous [16]. From the existence of optimal solution for (2.2), we must have, stationary point $0 \in \partial^+ F$.

2.4.1 Existence of optimal stochastic approximation algorithm

Lemma 2.1

$$F(\boldsymbol{\lambda}) \textit{ is concave.}$$

Proof: F is affine in $\boldsymbol{\lambda}$ and F is the point-wise minimum of a family of affine functions of $\boldsymbol{\lambda}$. Hence F is concave.

Lemma 2.2 *The stochastic approximation scheme for the maximization of function $F(\boldsymbol{\lambda})$ is given by,*

$$\boldsymbol{\lambda}(n+1) = \Gamma(\boldsymbol{\lambda}(n) + \alpha(n)[-\tilde{h}(\boldsymbol{\lambda}(n)) + M_{n+1}]), \ \tilde{h} \in \partial^+ F.$$

Proof: F is concave function, $F : \{0, \mathbb{R}^+\} \to \mathbb{R}$. A concave function is continuous in the interior of the domain. Hence F is continuous over $(0, \infty)$. By stochastic subgradient descent [17], Lemma 2.2 is proved. ▮

Lemma 2.3

$$\dot{\boldsymbol{\lambda}}(t) \in \partial F(\boldsymbol{\lambda}(t)) + \mathbf{z}(t). \tag{2.13}$$

Proof: $f(., p_1, p_2)$ is affine continuous and differentiable in $\boldsymbol{\lambda}$ and continuous in p_1, p_2. Hence the following properties are satisfied.

1. $f(\boldsymbol{\lambda}, p_1, p_2)$ is differentiable at $\boldsymbol{\lambda}$ uniformly in p_1, p_2.

2. $p_1 \rightarrow D_{\boldsymbol{\lambda}} f(\boldsymbol{\lambda}, p_1, p_2)$ is continuous.

3. $p_2 \rightarrow D_{\boldsymbol{\lambda}} f(\boldsymbol{\lambda}, p_1, p_2)$ is continuous.

4. $\boldsymbol{\lambda} \rightarrow f(\boldsymbol{\lambda}, p_1, p_2)$ is lower semicontinuous.

By [14], if above conditions are satisfied, then $\partial^+ F(\boldsymbol{\lambda}) = \bar{co} Y(\boldsymbol{\lambda})$,[5] where,

$$\begin{aligned} Y(\boldsymbol{\lambda}) : &= D_{\boldsymbol{\lambda}} f(\boldsymbol{\lambda}, p_1, p_2) : \{p_1, p_2 \in [0,1], F^*(\boldsymbol{\lambda}) = f(\boldsymbol{\lambda}, p_1, p_2).\} \\ p_1 &= \text{ Dirac function at } \left(\lambda_i - \frac{1}{x_i}\right)^+ \text{ given } i, \\ \{p_2\}_i &= P(y_i = 1) \\ &= P\left(I\left(\left(q_i^* - \lambda_i \left[\log(1 + q_i^* x_i) - C_i\right]\right) \geq \left(q_j^* - \lambda_j \left[\log(1 + q_j^* x_j) - C_j\right]\right), j \neq i\right)\right). \end{aligned}$$

Let $||\;||$ denote the Euclidean norm on $\mathbb{R}^N$. Let $H : \mathbb{R}^N \rightarrow \mathbb{R}^N$ denote a set-valued map. Stochastic approximation of the following form,

$$\boldsymbol{\lambda}(n+1) = \boldsymbol{\lambda}(n) + \alpha(n)\left[\bar{h}(n) + M(n+1)\right], \quad \bar{h}(n) \in H \tag{2.14}$$

characterizes a stochastic inclusion limit $\dot{\boldsymbol{\lambda}}(t) \in H(\boldsymbol{\lambda}(t))$ if,

1. H is upper semicontinuous.

2. $H(\boldsymbol{\lambda})$ is convex and and compact.

3. For some $K > 0$ and for all $\boldsymbol{\lambda}$

$$\sup_{\tilde{h} \in \hat{h}(\boldsymbol{\lambda})} ||\tilde{h}|| < K(1 + ||\boldsymbol{\lambda}||),$$

Set valued map $\partial^+ F$ satisfies the above properties by (A.18), (2.16).

[5] $\bar{co} Y$ denotes the compact convex hull of Y

Now we have to show $h(\boldsymbol{\lambda}) \in Y(\boldsymbol{\lambda})$. Differentiating f w.r.t. λ_i and substituting optimal values q^*, y^* we get,

$$\begin{aligned} \frac{\{\partial f\}_i}{\partial \lambda_i}\bigg|_{q^*,y^*} &= -[\int \nu(dx)\left(y_i(n)\text{log}(1+(\lambda_i(n)-\frac{1}{x_i})^+x_i)\right) - C_i]. \\ &\Rightarrow \; h(\boldsymbol{\lambda}) \in \bar{co}Y(\boldsymbol{\lambda}). \end{aligned}$$

$$\begin{aligned} &\int [\nu(dx)y_i(n)\text{log}(1+(\lambda_i(n)-\frac{1}{x_i})^+x_i) - C_i] \\ &= \; [\int \nu(dx)\left(y_i(n)\text{log}(1+(\lambda_i(n)-\frac{1}{x_i})^+x_i)\right)] \\ &\leq \; [\int [\nu(dx)\left(\text{log}(\lambda_i(n)x_i)\right)] \\ &\leq \; \int \nu(dx)\text{log}(\lambda_i(n)) + \int \nu(dx)\text{log}(x_i) \\ &< \; \hat{K}(1+\lambda_i) \text{ for some } \hat{K} \end{aligned} \tag{2.15}$$

From (2.15) we get,

$$||h|| < K(1+||\boldsymbol{\lambda}||). \tag{2.16}$$

By [17], taking set valued map H as $\partial^+ F$, stochastic approximation (2.13) satisfies a differential inclusion limit $\dot{\boldsymbol{\lambda}}(t) \in \partial^+ F$. Thus from Lemma 2.1, Lemma 2.2 is proved. ▮

2.4.2 Stability

Lemma 2.4 $\boldsymbol{\lambda}(t)$ *converges surely to a unique globally asymptotically stable equilibrium point.*

Proof: Let $\tilde{F}= -F$. Thus $\tilde{F}$ is the function to be minimized. Consider a continuous Lyapunov function $V = \tilde{F}(\boldsymbol{\lambda}) - \tilde{F}(\boldsymbol{\lambda^*})$. $V(\boldsymbol{\lambda}^*) = 0$, where $\boldsymbol{\lambda}^*$ is the optimal point. $V(\boldsymbol{\lambda}) \geq 0$ as λ^* is the optimal minimum point. For the non smooth Lyapunov function we use the Dini derivative D^+ and now the condition for stability is,

$$\langle \phi, D^+V(x) \rangle \leq 0, \quad \phi \in -\partial\tilde{F}. \tag{2.17}$$

But, $D^+V(x) \in \partial\tilde{F}$. Thus (2.17) is satisfied. As the minimum of F is unique, $\boldsymbol{\lambda}(t)$ converges surely converges to optimal point and is stable. ▮

The boundedness of the algorithm can be proved by assuming $\lambda_i \in [0, L] \; \forall \; i, L \geq 0$. The following theorem states precisely the boundedness of the iterates. Let $c_0 = 0$. Whenever $\boldsymbol{\lambda}$ is outside $[0, L]^N$, then $c_{n+1} = c_n + 1$, else c_{n+1}=c_n.

Theorem 2.1 *If* $\boldsymbol{\lambda}^* \in [0, L]^N$ *and all the assumption for Lemma 2.14 hold, then* $\lim_{n\to\infty} c_n < \infty$.

This theorem from [18] implies that the projection is required only a finite number of times.

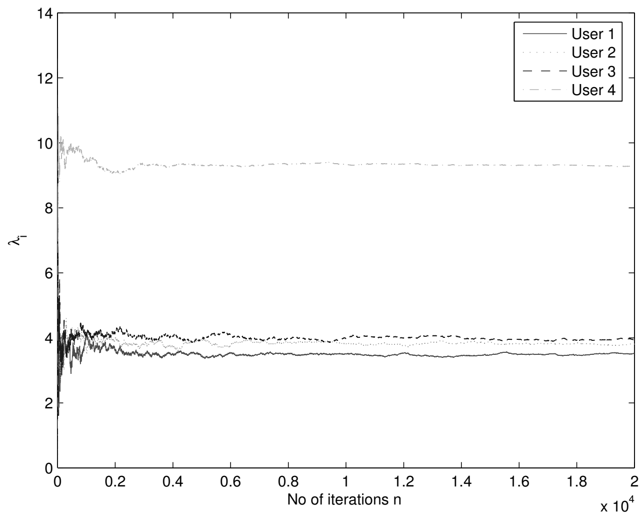


Figure 2.3: Convergence for i.i.d. channel

2.4.3 Convergence of Stochastic Approximation Algorithm

We demonstrate the convergence of $\lambda_i(n)$ via simulations. Consider a single hop network of 4 wireless users with 1 base station. We assume a Rayleigh fading channel, whose probability density function is given by $\mu(x) = \frac{1}{\gamma}e^{\frac{-x}{\gamma}}$, where $\gamma > 0$. We first show the convergence for channel model with i.i.d. Rayleigh fading. To make sense of absolute numbers, we take $\mathbf{C} = (0.6, 0.8, 0.7, 0.2)$, $\boldsymbol{\lambda}(0) = (1, 1, 1, 1)$ and $\gamma = (1, 1, 0.9, 0.3)$. Figure 2.3 shows the convergence for i.i.d. channel. The average power vector to achieve the desired rates over 50 independent runs is, **P**=(0.7557, 1.0925, 1.0546, 0.9612).

For simulation purposes, we next model the more general case of Markovian channel fading to demonstrate the correctness of our algorithm. Assume that the channel gain for user i obeys the auto-regressive equation,

$$x_i(n + 1) = \alpha x_i(n) + (1 - \alpha)g_i(n), \tag{2.18}$$

where noise $g_i(n)$ is Gaussian with variance σ and correlation coefficient α. We take $\alpha = 0.3$, $\mathbf{C} = (0.6, 0.8, 0.7, 0.2)$, $\boldsymbol{\lambda}(0) = (1, 1, 1, 1)$ and $\sigma = (1, 1, 0.9, 0.3)$. Figure 2.4 shows a particular trajectory for $\lambda_i(n)$. Our results demonstrate that the Lagrange multipliers converge for all users within 3000 iterations. The average power required is given by $(1.3677, 1.9966, 1.8971, 1.6322)$. From the absolute numbers it can be inferred that average power required for the Markovian channel is greater than for i.i.d. channel case, as expected.

Remark 2.1 *In wireless data transfer applications, the duration of transfer is of the order of seconds, while the slot duration is of the order of microseconds. Hence, even if there is non-optimality for the initial slots, convergence will occur much before the actual completion of data transfer.*

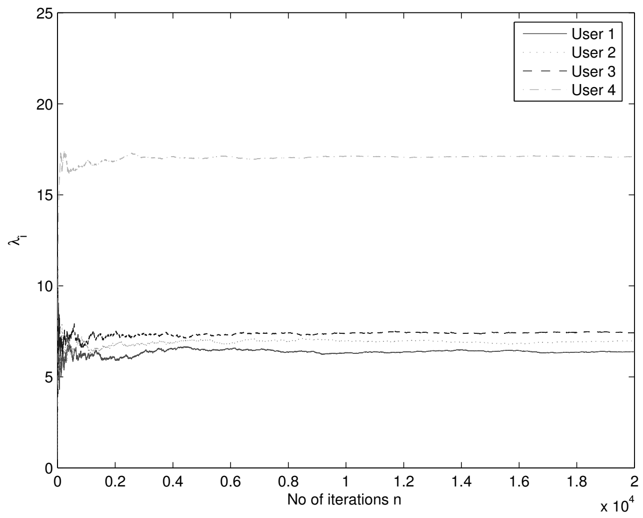


Figure 2.4: Convergence for Markovian channel

Remark 2.2 *In practical scenarios, we may not want actual convergence to take place, or we may only like to be within the neighborhood of the optimal solution. In [19], lock-in phenomenon for stochastic approximation algorithm has been considered. If the iterate $\boldsymbol{\lambda}(n)$ is within the domain of attraction (the iterate has begun to converge), then there exists a finite number of iterations for the iterate to be within a finite distance from the convergence point λ^*. A probabilistic lower bound is given for the occurrence of this "nearness" within finite number of iterations.*

2.4.4 Performance of Energy Optimal Opportunistic Scheduling

We compare our scheduling policy with the round robin power scheme. Consider a symmetric system, i.e., all users have the same channel conditions and minimum rate constraints. In the round robin power scheme, we transmit with optimal power for each user in a round robin manner. The scheme reduces to power-optimal single user scheme with minimum rate guarantee dependent on N. Thus we determine the power p such that

$$\int \log(1 + p(x)x)\mu(x)\ dx = NC \tag{2.19}$$

is satisfied. In our simulations, we assume $C = 0.6$, $\gamma = 0.7$. The results, shown in Figure 2.5, demonstrate that as the number of users increases, the ratio of average transmission power of the optimal policy to that of the round robin policy increases, but the marginal increase per user decreases. The gain obtained from variable power scheme increases with number of users, which is due to multiuser diversity.

We use (2.19) to calculate P numerically for a given C, for constant power opportunistic case. In Figure 2.6, we compare the opportunistic scheme and the optimal scheme in terms the ratio of

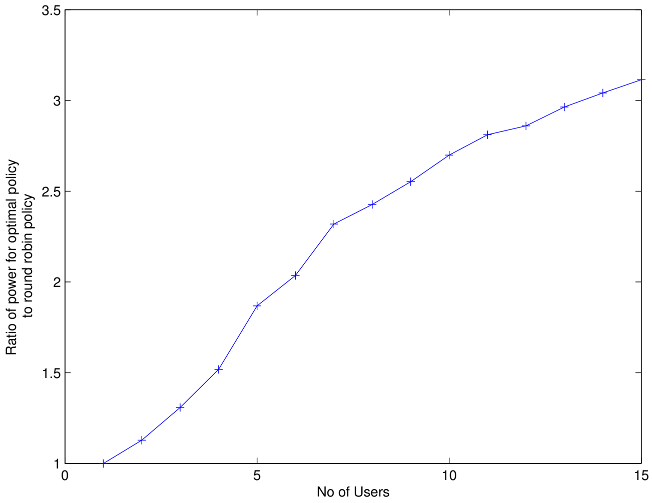


Figure 2.5: Gain of the optimal policy over round robin policy

average power required to satisfy a specific set of channel condition and rate constraints for two cases.

The simulation results show an interesting result. For low γ values the gain is larger than when the γ is large. Also as the number of users increases, the gain obtained by the optimal solution deceases. The decrease can be attributed again to the multiuser diversity. As the number of users increase, or the channel fading is fast, the opportunistic constant power scheduling performs as good as the optimal solution.

2.5 Extensions

2.5.1 Extension to Multiple Channels

In this section, we extend the result for single channel case to independent multiple channel multiuser system. We first notice an important fact that a channel can be considered to be a sub-slot of a TDMA slot. The idea is explained in Figure 2.7. Let there be M number of channels. Now the new system can be considered to be made up of M sub-slots with the channel condition known. Thus combination of (2.3) and stochastic approximation algorithm over M time slots would give a sub-optimal solution for multiple channels as:

$$\begin{aligned} k &= \arg_k \min \left(\left(\lambda_k - \frac{1}{x_k} \right)^+ - \lambda_k \left[\log(1 + \left(\lambda_k - \frac{1}{x_k} \right)^+ x_k) - C_i \right] \right). \\ q^* &= \frac{1}{M} \left(\lambda_k - \frac{1}{x_k} \right)^+, \end{aligned} \tag{2.20}$$

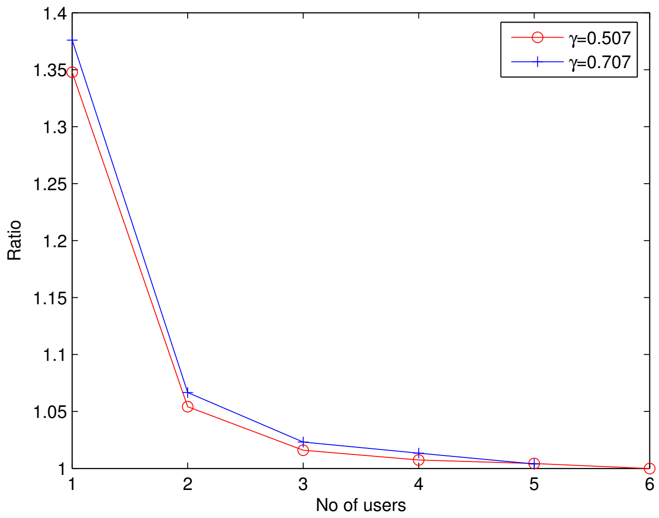


Figure 2.6: Gain of the optimal policy over opportunistic scheduling with constant power

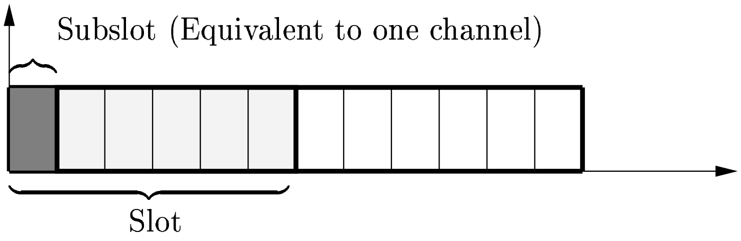
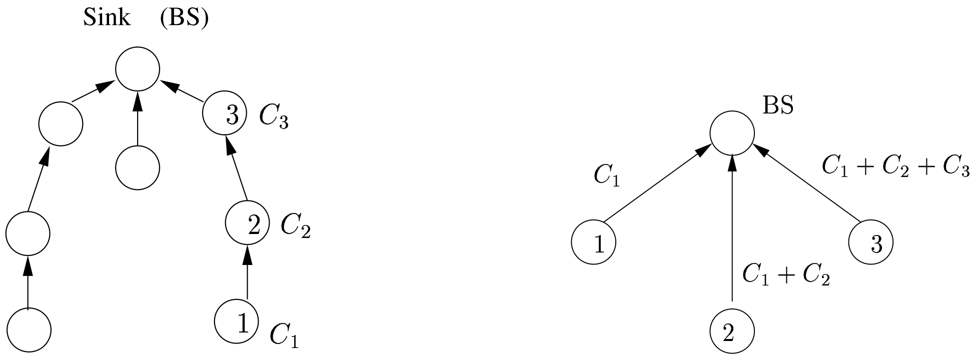


Figure 2.7: Equivalence between TDMA and Multichannel schemes

where $\frac{1}{M}$ takes into account the decrease in effective resource, in the equivalent slot based system. If we take an example of Orthogonal Frequency Division Multiplexing (OFDMA), M would denote the number of subcarriers. The algorithm would perform update equation for each channel during a given slot. The algorithm is sub-optimal because the algorithm does not take into knowledge of channel gains of all the channels simultaneously.

2.5.2 Multi-hop Networks

In this section we would extend the result to the single sink Multi-hop TDMA network. The network we examine is shown in Figure 2.8(a). It consists of mobile nodes as sources and base station as a single sink for all the sources. We say that the nodes are in transmission range if the transmission is received without any error and in interference range if one transmission does not allow other transmissions in the same time slot. We assume that mobile stations placed as shown in Figure 2.8(a), are in the interference range. The situation is realistic in accordance with the current IEEE 802.16 based Mesh network [20]. However for the simplicity we assume that the routing matrix is predefined and is static. Thus we have a problem of scheduling the nodes energy efficiently so that the rate constrained of all the nodes are satisfied. Here the effective rate requirements would be sum of individual rate requirements along that route as shown in Figure 2.8(a). We can now convert the multi-hop scenario into single hop. The effective link rates are given in Figure 2.8(b). Thus now the users are scheduled with the new rate constraints according to the policy (2.3).



(a) Multi-hop transmission scheme

(b) Single hop transmission

Figure 2.8: Equivalence between single hop and Multi-hop transmission.

2.6 Conclusions and Discussions

In this chapter, we have obtained a power optimal opportunistic scheme for multiuser TDMA system with minimum rate constraints for individual users. We have proposed a stochastic approximation based technique, obtained an online optimal scheduling algorithm and argued the theoretical convergence of the stochastic approximation policy. The results are extended to multiple channel and multi-hop scenario.

Appendix

2.7 Boundedness of Iterates

For proving the boundedness of the iterates $\boldsymbol{\lambda}$ we use a variation of the method adopted for proving the boundedness of a linear stochastic approximation algorithm in [21]. We impose following assumption for proving the boundedness.

Assumption 2.1 $x(n) \in (0, \infty)$

This assumption is imposed for the boundedness of the function $\hat{h}(\boldsymbol{\lambda})$ defined after the assumptions are mentioned. The assumption is valid for most of the pdfs used in modeling the channel.

Assumption 2.2 $\lambda_i(n) > 0$

This assumption is also imposed for the boundedness of the function $\hat{h}(\boldsymbol{\lambda})$. For the linear programming problem considered, the constrained is satisfied at the boundary. This means λ_i^* is > 0, whenever $C_i > 0$. In the stochastic approximation algorithm, we abide by this constraint by setting $\lambda_i = \alpha, \alpha > 0, \alpha \rightarrow 0$, whenever $\lambda_i \leq \alpha$.

Assumption 2.3 $h_i(\boldsymbol{\lambda}) - C_i \neq 0$ *for* $\boldsymbol{\lambda} \neq \boldsymbol{\lambda}^*$

Lemma 2.5 $\left|\frac{\log\left(1+(\lambda_i - \frac{1}{x_i})^+\right)}{\lambda_i}\right|$ *is bounded.*

Proof: It can be easily proved using Assumption 2.1, 2.2.

Consider the following iteration to update $\boldsymbol{\lambda}$,

$$\boldsymbol{\lambda}(n+1) = \boldsymbol{\lambda}(n) + \alpha(n)\left(h(\boldsymbol{\lambda}) + M(n+1)\right) \tag{2.21}$$

We study the asymptotic behavior of the iterates $\boldsymbol{\lambda}$. Consider the subsequence $< \boldsymbol{\lambda}(n_j) >$, where the sequence of integers n_j are defined as,

$$n_0 = 0, \quad n_{j+1} = \min\left\{ n > n_j \left| \sum_{l=n_j}^{n-1} \alpha(l) > T \right.\right\},$$

where $T > 0$. We define a sequence,

$$\hat{\lambda}_i^j(n) = \frac{\lambda_i(n)}{\max\left(1, |\lambda_i(n)|\right)}, \quad n \geq n_j,$$

where $|\quad|$ denotes the max norm. The new iterations $\lambda^j(n)$, are given by,

$$\begin{aligned} \hat{\lambda}_i^j(n+1) &= \hat{\lambda}_i^j(n) + \alpha(n)\left(\frac{(h_i(\boldsymbol{\lambda}^j(n))}{\lambda_i^j(n)} \frac{\lambda_i^j(n)}{\max(1, |\boldsymbol{\lambda}^j(n)|)}\right) + \alpha(n)\frac{M^j(n+1)}{\max\left(1, |\boldsymbol{\lambda}^j(n)|\right)} && (2.22) \\ &= \hat{\lambda}_i^j(n) + \alpha(n)\left(\hat{h}_i(\boldsymbol{\lambda}^j(n)) + \hat{M}(n+1)\right) && (2.23) \end{aligned}$$

Define a stopping time $\tau_j^1(C) = \min\left\{n \geq n_j : |\hat{\boldsymbol{\lambda}}^j(n) \geq K|\right\}$ The explicit dependence of K is not important in the further analysis, hence we suppress the notation. We first note that the value of $\mathbf{E}[|\hat{\boldsymbol{\lambda}}^j(n)|]$ is bounded. Let $b_j = \frac{1}{\max\left(1, |\hat{\boldsymbol{\lambda}}^j(n)|\right)}$. Following lemma states that, as long as the iterates are bounded, the perturbation noise $\hat{M}(n+1)$ also remain negligible.

Lemma 2.6 *There exits a constant* C*, such that,*

$$\mathbf{E}\left[\max_{n_j \leq n \leq \tau_j^1 \wedge n_{j+1}} \left|\sum_{l=n_j}^{n} \alpha(l)\hat{M}(n+1)\right|^2\right] \leq C \sum_{l=n_j}^{n_{j+1}-1} \alpha(l)^2 \tag{2.24}$$

Proof: If $\tau_j^1 \geq n_{j+1}$, the above inequality is trivially satisfied. Hence we consider only for $n > n_j$. As $M(n+1)$ is a martingale sequence,

$$\sum_{l=n_j}^{n\wedge\tau_j^1} \alpha(l) b_j M(l+1), \tag{2.25}$$

is also a martingale and we can use the Doob's inequality,

$$\mathbf{E}\left[\max_{n_j \leq n \leq \tau_j^1 \wedge n_{j+1}} \left|\sum_{l=n_j}^{n^{\tau_j^1}} \alpha(l)\hat{M}(n+1)\right|^2\right], \tag{2.26}$$

$$\leq C_1 \mathbf{E}\left[\sum_{l=n_j}^{n^{\tau_j^1}} \left|\alpha(l)\hat{M}(n+1)\right|^2\right], \tag{2.27}$$

$$\leq C_2 \sum_{l=n_j}^{n_{j+1}} \alpha(l)^2 \mathbf{E}\left[I\{l \leq \tau_j^1]\} \left|\hat{M}(n+1)\right|^2\right], \tag{2.28}$$

$$\leq C_3 \sum_{l=n_j}^{n_{j+1}} \alpha(l)^2 \tag{2.29}$$

Next we consider an approximate deterministic iteration given by,

$$\boldsymbol{\lambda}^j(n+1) = \boldsymbol{\lambda}^j(n) + \alpha(n)\left(h(\boldsymbol{\lambda})\right), \tag{2.30}$$

$$\boldsymbol{\lambda}^j(n_j) = \hat{\boldsymbol{\lambda}}^j(n_j) \tag{2.31}$$

Define a new stopping time $\tau_j^2(\delta)$ as,

$$\tau_j^2(\delta) = \min\{n \geq n_j : \left|\hat{\boldsymbol{\lambda}}^j(n) - \boldsymbol{\lambda}^j(n)\right|\} \tag{2.32}$$

From Lemma 2.6 and boundedness of $\hat{h}(\boldsymbol{\lambda})$, we obtain,

$$\sup_j \max_{n_j \leq n} \left|\boldsymbol{\lambda}^j(n)\right| \leq C, \tag{2.33}$$

for some constant C. Hence $\tau_j^1(C+\delta) \geq \tau_j^2(\delta)$. That is by the time $\boldsymbol{\lambda}^j(n)$ gets out of the ball with radius $C+\delta$, $\hat{\boldsymbol{\lambda}}^j(n)$ must be deviated from $\boldsymbol{\lambda}^j(n)$ by at most δ since $\boldsymbol{\lambda}^j(n)$ is completely inside the ball with radius C.

Lemma 2.7

$$\lim_j \max_{n_j \leq n \leq n_{n+1}} \left|\hat{\boldsymbol{\lambda}}^j(n) - \boldsymbol{\lambda}^j(n)\right| = 0, \ \textit{w.p. 1} \tag{2.34}$$

Proof: Since $\hat{h}$ is bounded, for $n \geq n_j$,

$$\left|\hat{\boldsymbol{\lambda}}^j(n+1) - \boldsymbol{\lambda}^j(n+1)\right| \leq C \sum_{l=n_j}^{n} \alpha(l) \left|\hat{\boldsymbol{\lambda}}^j(l) - \boldsymbol{\lambda}^j(l)\right| + \left|\sum_{l=n_j}^{n} \alpha(l)\hat{M}(l+1)\right| \tag{2.35}$$

Using Gronwall inequality,

$$\max_{n_j \leq n \leq n_{j+1} \wedge \tau_j^1} \left|\hat{\boldsymbol{\lambda}}^j(n+1) - \boldsymbol{\lambda}^j(n+1)\right| \leq e^{CT} \max_{n_j \leq n \leq n_{j+1} \wedge \tau_j^1} \left|\sum_{l=n_j}^{n} \alpha(l)\hat{M}(l+1)\right| \tag{2.36}$$

Using Chebyshev inequality,

$$\mathrm{P}\left(\max_{n_j \leq n \leq n_{j+1} \wedge \tau_j^1} \left|\hat{\boldsymbol{\lambda}}^j(n+1) - \boldsymbol{\lambda}^j(n+1)\right| \geq \delta\right) \leq \frac{C_1}{\delta^2} \sum_{l=n_j}^{n_{j+1}-1} \alpha(l)^2 \tag{2.37}$$

Since $\tau_j^1 \geq \tau_j^2$, LHS is actually $\mathrm{P}(\tau_j^2 \leq n_{j+1})$. Therefore,

$$\mathrm{P}\left(\max_{n_j \leq n \leq n_{j+1} \wedge \tau_j^1} \left|\hat{\boldsymbol{\lambda}}^j(n+1) - \boldsymbol{\lambda}^j(n+1)\right| \geq \delta\right) \leq \frac{C_1}{\delta^2} \sum_{l=n_j}^{n_{j+1}-1} \alpha(l)^2 \tag{2.38}$$

Using Borel Cantelli Lemma and using the property $\sum_n \alpha(n)^2$, we prove the lemma.

Lemma 2.8 *If for $\delta > 0$,*

$$\boldsymbol{\lambda}'\hat{h}\boldsymbol{\lambda} \geq \delta \left|\boldsymbol{\lambda}\right|^2 \tag{2.39}$$

Then, for small $a > 0$,

$$\left|1 - a\hat{h}\boldsymbol{\lambda}\right| \leq (1 - \frac{1}{2}a\delta) \left|\boldsymbol{\lambda}\right| \tag{2.40}$$

Lemma 2.9

$$\sup_n \left|\boldsymbol{\lambda}(n)\right| < \infty, \ \textit{w.p. 1}, \tag{2.41}$$

hence the iterates are bounded.

Proof: Since $\hat{h}(\boldsymbol{\lambda})$ is bounded, for large n_j , for some constant C.

$$\left|\boldsymbol{\lambda}^j(n+1)\right| \leq \left(1 - \frac{1}{2}\alpha(n)\delta\right) \left|\boldsymbol{\lambda}^j(n)\right| + \alpha(n)\frac{C}{\max(1, |\boldsymbol{\lambda}(n_j))},\tag{2.42}$$

Using the inequality $1 - x \leq e^{(-x)}$, we get.

$$\left|\boldsymbol{\lambda}^j(n+1)\right| \leq e\left(-\frac{1}{2}\sum_{l=n_j}^{n} \alpha(l)\delta\right) \left|\boldsymbol{\lambda}^j(n)\right| + \sum_{l=n_j}^{n} \alpha(l)\frac{C}{\max(1, |\boldsymbol{\lambda}(n_j)|)},\tag{2.43}$$

From Lemma 2.7, for $\alpha_j \to 0$,

$$\frac{|\boldsymbol{\lambda}(n_{j+1}|)}{\max(1, |\boldsymbol{\lambda}(n_j)|)} \leq e^{(-\frac{1}{2}\delta T)} \frac{|\boldsymbol{\lambda}(n_j)|}{\max(1, |\boldsymbol{\lambda}(n_j)|)} + T \frac{C}{\max(1, |\boldsymbol{\lambda}(n_j)|)} + \alpha_j \tag{2.44}$$

$$\Rightarrow \quad |\boldsymbol{\lambda}(n_{j+1})| \leq \left(e^{(-\frac{1}{2}\delta T)} + \alpha_j\right) |\boldsymbol{\lambda}_{n(j)}| + CT + \alpha_j. \tag{2.45}$$

$$\begin{aligned}
\sup_n |\boldsymbol{\lambda}(n)| &= \sup_j \max\left(1, |\boldsymbol{\lambda}(n_j)|\right) \max_{n_j \leq n \leq n_{j+1}} \left|\hat{\boldsymbol{\lambda}}^j(n)\right| \\
&\leq \sup_j \max(1, |\boldsymbol{\lambda}(n_j)|) \left(\max_{n_j \leq n \leq n_{j+1}} \left|\hat{\boldsymbol{\lambda}}^j(n)\right| + \max_{n_j \leq n \leq n_{j+1}} \left|\boldsymbol{\lambda}^j(n) - \hat{\boldsymbol{\lambda}}^j(n)\right|\right) \\
&< \infty
\end{aligned}$$

Chapter 3

Power Optimal Opportunistic Scheduling - Fairness Guarantees

In this chapter, we introduce fairness while considering the power optimization. In Chapter 2 we have restricted ourselves to satisfy minimum rate guarantee of individual users and not considered relative chance of transmission among users. In this chapter, we consider energy optimal fairness in the long term as well as short term.

3.1 Scheduling and Fairness issues

A comprehensive work in [22] considers wireless packet fair schedulers and extends the work of wireline fair queuing to wireless. In particular, fairness is assured by making lagging flows to catch-up leading flows over long run in binary channel model.

The Proportional Fair (PF) Scheduler of the Qualcomm High Data Rate (HDR) system [23] deals with conflict between fully exploiting the channel (by selecting the user with the highest current rate) and being fair. The PF algorithm can be shown to be maximizing the logarithmic utility functions for the users in asymptotic sense. In a time slot, the user j is selected for transmission if,

$$j = \arg\max_i \left(\frac{r_i(n)}{[R_i(n)]^\alpha} \right) \tag{3.1}$$

$$R_i(M) = \frac{1}{M} \sum_{n=1}^{M} r_i(n) y_i(n) \tag{3.2}$$

where $R_i(M)$ is the time average of rates for user i over time M, $y_i(n)$[1] is the indicator function for user i and $r_i(n)$ is the rate of user i an the slot n. α is the fair exponent, for opportunistic scheduling $\alpha = 0$, for proportional fair $\alpha = 1$. Variants of proportional fair schedulers like relatively

[1] $y_i(n) = 1$ if user i is scheduled at slot n otherwise 0

fair scheduler have been proposed. Modified Weighted Delay First (M-LWDF) strategy [24] serves the user j such that,

$$j = \arg\max_i \left(\frac{\gamma_i W_i(n)}{c_i(n)} \right) \tag{3.3}$$

where W_i is the head of the line delay for user i, $\gamma_i = \delta_i/R_i$, $\delta_i > 0$, R_i is the average rate requirement for user i, $c_i(n)$ is the power required per unit of transmission. The policy is able to satisfy minimum rate requirement in the long term and probabilistic delay constraints even under discrete rate scheduling.

In [25], the author points out that the proportional fair scheduler is unstable by giving some examples for which stable scheduler exits. A modified fair rule in [26] called exponential rule is able to provide stability, if there exits any policy which does so. The exponential rule selects the user j if,

$$j = \arg\max_i \gamma_i r_i(n) \exp \left(\frac{\delta_i Q_i(n)}{\beta + \left[\delta_i \overline{Q_i(n)} \right]^\eta} \right) \tag{3.4}$$

where $Q_i(n)$ is the queue length at time slot n, $\overline{Q_i(n)}$ is the average queue length, δ_i, β_i, η are positive constants. A larger weighted latency of one of the users results in a very large exponent, overriding the channel conditions and leading to the large latency user getting priority. On the other hand, for small weighted latency differences, the exponential term is close to 1 and the policy becomes proportional fair rule.

In [27], the authors compared the guaranteed supportable arrival rates for delay constrained traffic for opportunistic and TDM schedulers in Rayleigh's fading channel conditions. Using large deviation theory, it is shown that TDM scheduler performs better when the number of users exceeds a threshold level which depends on the channel parameters. There is a trade off between total system throughput and fairness and QoS guarantee among users. Maximizing throughput strategy can lead to unfairness among users. Hence a compromise between throughput and fairness has to be reached.

In [8], the authors consider the problem of throughput maximization with deterministic and probabilistic long term fairness constraints. The throughput maximization and fairness achievement are decoupled into two sub problems and solved as different entities. A adaptive iterative algorithm is suggested for finding control weight vectors determined by current fairness achieved among users. These weights are then used for maximizing the throughput.

In [2], an optimal index policy is derived for long term fairness in terms of bandwidth allocation and maximizing throughput considering the probability distribution of instantaneous user rates.

In [3], the authors study scheduling policies under Short Term Fairness (STF) constraints. Short term fairness reduces the inter scheduling delays at the cost of throughput. Using special case of window size of $M = N$ and $M = \infty$, where N are the number of users, the STF constrained policy assigns $\phi_i M$ number of time slots to a user i in any scheduling frame of window M and maximizes the system throughput under these constraints where ϕ_i is the weight assigned to the user i such that $\sum \phi_i \leq 1$. A heuristic policy that schedules the users that maximize the system throughput, while trying to provide the required STF guarantees is suggested. A heuristic policy for obtaining for a general M has been suggested. It has been proved that such allocation in opportunistic regime gives more throughput than scheduling non-opportunistically.

$$\mathbf{E}(r_i(n)y_i(n)) \geq \phi_i \mathbf{E}(r_i(n)) \tag{3.5}$$

As apposed to the view of throughput fairness, alternative idea of temporal fairness is more suitable for wireless opportunistic schedulers. We define the temporal fairness as fairness in the the number of media access or the number slots to the users. Providing throughput fairness results in the degradation of the other users with weak channel conditions. On the other hand, the power optimal scheduling considered in the Chapter 2 results in starvation of strong users in oder to satisfy the rate guarantees of week users.

Hence in this chapter, we propose a scheme that minimizes power while guaranteeing minimum rate and long term fairness to each user.

3.2 Temporal Fairness

3.2.1 Long Term Fairness

Our objective is to opportunistically schedule the user with the best channel condition such that rate guarantees and temporal fairness are achieved and average transmission power is minimized.

Let ϕ_i be the proportion of *temporal bandwidth* allocated to user i and $\boldsymbol{\phi} = [\phi_1 \phi_2 \cdots \phi_N]$

Thus, ϕ_i represents the fraction of the time slots allocated to user i. Our objective is to minimize average power subject to rate and fairness constraints. Our optimization problem is the same as that of (2.1) with the following additional constraint

$$\liminf_{M \to \infty} \frac{1}{M} \sum_{n=1}^{M} \mathbf{E}\ y_i(n) \geq \phi_i \quad \forall i. \tag{3.6}$$

Using the ergodicity assumption from Section 2.2, the Lagrangian with the fairness constraint is:

$$\mathcal{L}(p_1, p_2,\ \boldsymbol{\lambda}\ ) \triangleq \int \nu(dx_1, \cdots, dx_N) \sum_{y \in A} \int_{[0,\infty)} p_1(dq|y,x) \\ p_2(y|x) \left(q - \sum_i \lambda_i \left[\log(1 + q y_i x_i) \ \ - C_i\right] + \sum_i \lambda'_i (y_i - \phi_i) \right),$$

where λ'_i is the Lagrange multiplier associated with the constraint (3.6), $\boldsymbol{\lambda}$ is the vector $(\lambda_1, \cdots, \lambda_N, \lambda'_1, \cdots, \lambda'_N)$. Following the approach adopted in Section 2.2, we obtain the optimal policy as: Select user k and transmission power q^* where

$$\begin{aligned} k &= \arg\min_i \Big\{ \Big(\lambda_i - \frac{1}{x_i}\Big)^+ - \lambda_i \Big[\log\Big(\\ &\quad 1 + \Big(\lambda_i - \frac{1}{x_i}\Big)^+ x_i \Big) - C_i \Big] + \lambda_i'(1-\phi_i) \Big\} \qquad (3.7) \\ q^* &= \Big(\lambda_k - \frac{1}{x_k}\Big)^+. \qquad (3.8) \end{aligned}$$

Using the stochastic approximation algorithm from Section 2.2, the Lagrange multiplier update equations can be written as

$$\begin{aligned} \lambda_i(n+1) &= \Big[\lambda_i(n) - a(n) \Big[y_i(n) \log\Big(\\ &\quad 1 + \Big(\lambda_i - \frac{1}{x_i(n)}\Big)^+ x_i(n) \Big) \Big] - C_i \Big]^+ \\ \lambda_i'(n+1) &= [\lambda_i'(n) - a(n)(y_i(n) - \phi_i)]^+ \quad \forall i \qquad (3.9) \end{aligned}$$

The optimality of the above scheme can be proved in a manner similar to that in Section 2.2.

3.2.2 Short Term Fairness

In Section 3.2.1, we have considered long term fairness. Long term fairness guarantees average proportional time share. However, one of the problems associated with long term fairness is starvation or Head of Line (HOL) blocking. There exist conditions when a user may not get a chance to transmit for some period of time even after being assured a minimum rate guarantee. Thus, it is important to consider a short term fair scheduler.

We consider a window of size $M \geq N$ slots. In a short term fair scheduler, we allocate time share equal to $\phi_i M$[2] to user i over this window and say that the scheduler is short term fair over the window M. The case $M \to \infty$ is same as the long term fairness. We first discuss the case when $M = N$. Let $\mathcal{A}$ be the set of users, i.e., user $k \in \mathcal{A}$. For $M = N$, we can allocate a maximum of one slot per user. We first select the user from the set $\mathcal{A}$ which is optimal for that time slot from (3.7)[3]. Let k be the optimal user. We remove user k from the list: $\mathcal{A} = \mathcal{A} \setminus \{k\}$. We repeat the above process on modified $\mathcal{A}$. We call this policy as *elimination policy.* The algorithm for general M is explained below.

3.2.3 Performance Results

We run the simulation over 20000 time slots. We take rate vector $\mathbf{C} = (0.6, 0.8, 0.7, 0.2)$, $\boldsymbol{\gamma} = (1, 1, 0.9, 0.3)$, $l = 10$ and $\boldsymbol{\lambda}(0) = (1, 1, 1, 1)$. In Figure 3.1 we have shown a snapshot of a particular trajectory for $\lambda_i(n)$. The results demonstrate that the λs converge for all users. The average power

[2] We assume that $\phi_i M$ is an integer $\forall$ i.

[3] In the modified algorithm, the set of users is $\mathcal{A}$.

Algorithm 1 Temporal Short Term Fair Scheduling

```
Slot vector $\mathbf{v}= M(\phi_1, \phi_2, \cdots, \phi_N)$
$\mathcal{A} = \{1, 2, \cdots, N\}$
$i = 1$
for $i \leq M$ do
    for each $j \in \mathcal{A}$ do
        Choose optimal $k$ using (3.7).
        Transmit with power $q^*$
        $\{\mathbf{v}\}_k = \{\mathbf{v}\}_k - 1$
        if $\{\mathbf{v}\}_k \leq 0$ then
            $\mathcal{A} = \mathcal{A} \setminus \{k\}$
        end if
    end for
end for
```

vector to achieve the desired rates over 10 independent runs is, $\mathbf{P} = (0.7557, 1.0925, 1.0546, 0.9612)$. In our simulations, we assume that the channel gains are Markovian across slots, as in (2.18). To make sense of absolute numbers, we assume $\mathbf{C} = (0.6, 0.8, 0.7, 0.2)$ and $\boldsymbol{\phi} = (0.3, 0.4, 0.2, 0.1)$. We implement Algorithm 1 and plot the average power required with increasing window size, as shown in Figure 3.2. The power required is a decreasing function of window size. The actual fairness achieved by the long term and short term temporal fair algorithm are plotted in Figure 3.3. It may be noted that in short term fair scheduler more emphasis is given to providing temporal fairness, but in the process, the actual rate obtained may deviate from the desired rates. Thus there is trade off between window size and actual rates achieved.

3.3 Throughput Short Term Fair scheduling

Here again we consider a window length of M slots. We say that the short term fairness is satisfied if we satisfy the rate constraints for the duration of length M. or the overall throughput in M time slots is MTC_i, where T is the slot length and C_i is the individual rate constraint. Thus the new short term fair algorithm will transmit exactly MTC_i for user i, so that the overall power is minimized. The problem can be formulated as,

$$\min \sum_{n=1}^{M} \sum_{i=1}^{N} q(n) y_i(n) \tag{3.10}$$

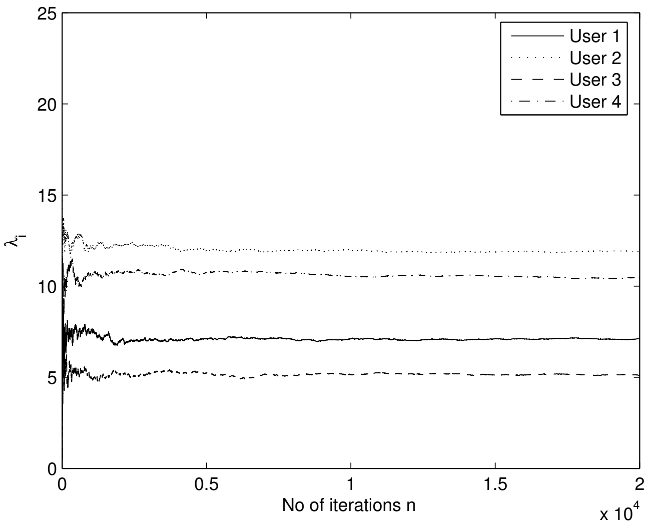


Figure 3.1: Trajectory of $\lambda_i(n)$

$$
\begin{aligned}
\sum_{n=1}^{M}\sum_{i=1}^{N} U_i(y_i(n)q_i(n)x_i(n)) &\geq MTC_i \quad \forall i,\\
q(n) &\geq 0,\\
\sum_{i=1}^{N} y_i(n) &\leq 1 \quad \forall n.
\end{aligned}
\tag{3.11}
$$

We assume concave power-rate relationship, taking U_i as standard Shannon capacity as in Section 2.2. We consider a finite horizon problem, as opposed to infinite horizon problem considered in Section 2.2, here it would be be desirable to wait for better channel conditions before transmission and keep track of the amount of data to be transmitted by each user. The state of the system is given by, $\{\mathbf{r}(\mathbf{n}), \mathbf{x}(n)\}$, where $\{\mathbf{r}(n)\}_i$ denotes the residual amount to be transmitted by user i at slot n. At $n = 0, \{\mathbf{r}(\mathbf{n})_i = MTC_i\}$. A dynamic programing formulation of (3.11) is expressed as (cf. Chapter B):,

$$
\mathbf{J}(n, \mathbf{r}(n), \mathbf{x}(n)) = \min\left(q(n) + \bar{\mathbf{J}}(n+1, \mathbf{r}(n+1))\right), \tag{3.12}
$$

$$
= \min_{y_i(n),q(n)} \sum_{i}^{N} \frac{1}{x_i}\left(e^{(u_i(n)y_i(n))} - 1\right) + \bar{\mathbf{J}}(n+1, \mathbf{r}(n+1)), \tag{3.13}
$$

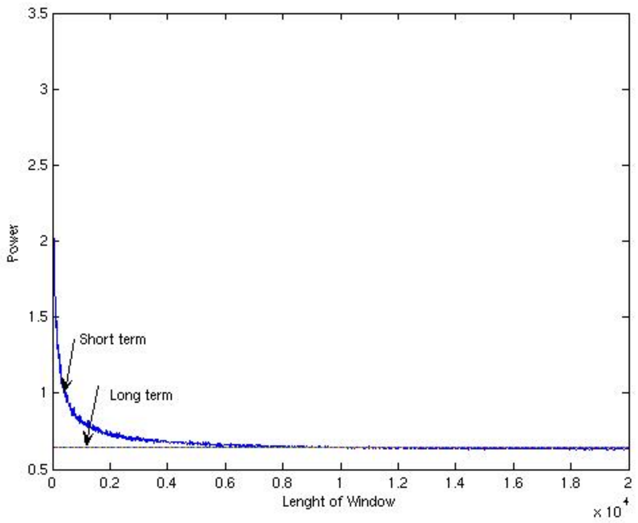


Figure 3.2: Power required for the short term fairness compared to long term fairness

where $u_i(n)$ is the amount of data transmitted by user i in time slot n and the expected future cost of decision is given by,

$$\begin{aligned}\bar{\mathbf{J}}(n+1,\mathbf{r}(n+1)) &= \mathbf{E}\left(\mathbf{J}(n+1,\mathbf{r}(n+1),\mathbf{x}(n+1))\right) && (3.14)\\ &= \mathbf{E}\left(\mathbf{J}(n+1,\mathbf{r}(n+1),\mathbf{x}(n+1))\right) && (3.15)\end{aligned}$$

The stopping condition has to be imposed so that by $M+1$th time slot, MTC_i data has been transmitted. This condition can stated as terminal cost,

$$\bar{\mathbf{J}}(M+1,\mathbf{r}(M+1)) = \infty. \tag{3.16}$$

The dynamic programing problem (3.13) does not have a closed form expression. It can be calculated numerically by assuming the discrete channel state. However, the state space of this problem is large and increases exponentially as the number of users increases. A heuristic policy is needed to deal with intensive computation.

Heuristic policy

We use optimal policy (2.3) for the heuristic short term fair algorithm. On a long run of time slots, the short term fair algorithm works on a finite window size of M. We would simulate optimal stochastic approximation algorithm in the background. Thus at the beginning of each window M, we would use the current values of λ obtained using stochastic approximation algorithm (2.10). For $M-N$ number of slots, we schedule the users with the optimal power values. We also keep the track of the amount of data transmitted in these time slots by each user as well as update

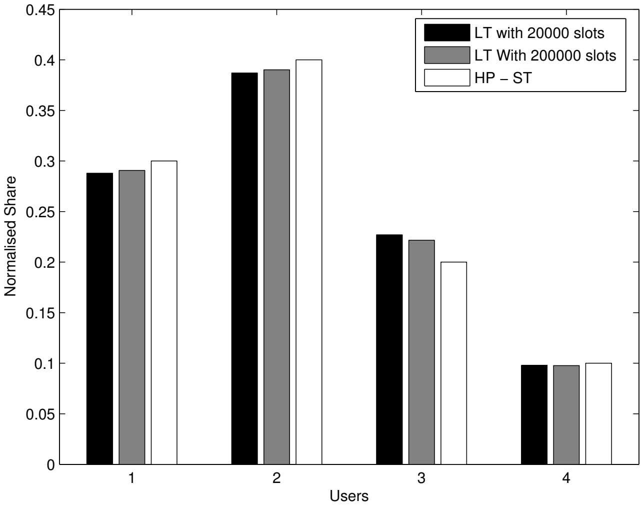


Figure 3.3: Fair achieved by short term fair scheduler

the parameters λ_i. Now for the N time slots, we use the elimination policy, i.e. schedule the user which is best according to the criterion (2.3), but transit with power, so that the overall throughput requirement for M times slot, for that particular user is met. Now eliminate this user and perform the same algorithm on rest of the users. Thus last N slots are used to satisfy the throughput requirements.

3.4 Conclusions and Discussions

In this chapter, we have extended power-optimal rate constrained scheduling to incorporate temporal long-term fair scheduling. We utilize this concept of long term fair scheduling to devise a heuristic based short term fair algorithm. We compare the performance of the short term fair and long term fair algorithms with respect to power consumption and fairness achieved. We also state a throughput fair power optimal control policy. based on Markov Decision formulation and suggest a heuristic policy for obtaining throughput fairness.

Chapter 4

Power Optimal Opportunistic Scheduling -Average Delay Guarantees

4.1 Introduction

Efficient use of limited resources for providing Quality of Services (QoS) is an important issue in the design of wireless networks. Power efficient transmission on the uplink has much significance, since it impacts the battery life of the consumer mobile device and thus the overall efficiency of the scarce resources. Hence methods for wireless transmission should be designed so that throughput is maximized with minimum power consumption. However along with maximizing throughput, the scheduler must also take into account the delay requirements of the individual users. The opportunistic scheduling dealt in Chapter 2, does not consider the effect of the queue lengths and arrivals, and bounds on the delay experienced by the user remain unspecified. Hence, the individual delay requirements of the users should be emphasized while designing the scheduling algorithm.

Wireless networks ought to be designed to cater to the needs of heterogeneous traffic like, video, voice and file transfers. For file transfer like applications, the performance measure would be average delay constraint. On the other hand, real time applications like video and voice have to conform to the absolute delay guarantees. We consider these cases separately. First we, discuss the average delay problem. In Chapter 5, we address the problem for strict delay constraint.

According to the classic result of Shannan capacity, for the AWGN channel the transmission rate is a concave increasing function of transmitted power. This means that the marginal utility of transmitting with higher power actually decreases. Only sufficient power needs to be transmitted so that delay requirements are met. Thus the power efficient transmission comes with a trade-off with the delay. In [28], the authors have considered problem of optimizing average power, subject to average delay constraint for AWGN channel. They showed that the optimal stochastic scheduler can be expressed as convex linear combination of deterministic schedulers. They used a well known fact that a optimal policy can be obtained by randomization of deterministic policies. In [29], dual problem of maximizing throughput given the delay constraint and power constraint in an AWGN channel is considered. However [28, 29] do not consider the effect of fading in devising scheduling strategies.

In the time varying fading environment, delaying transmission of packets during “bad channels” and waiting for “good channels” states is the basic methodology used in the power minimization. The scheduler transmits with more power and rate during ”good channel” states so that power is most efficiently used.

In the early work [30], transmission policies are derived for the average power minimization with average delay constraints in a time varying channel. However assuming a linear relationship between power and rate, [30] has showed that the gains in power are possible, even if the power delay relationship is not convex by exploiting the channel conditions.

In [31, 32], the authors have investigated the issues in minimizing power for time varying channels. The problem is formulated as a discrete time constrained Markov Decision Process (MDP). Various structural properties of the policies involved are analyzed. The convexity of the capacity curve is used in concluding the convex relationship between power and delay. An asymptotic quantitative expression for the Power-Delay curve is also derived. In a more recent work [33], the authors have derived a closed form expression for minimizing average delay given the average power constraint.

Most of the work discussed above present structural properties of the optimal policies. The actual scheduling algorithm to achieve the optimal trade-off is never considered, which is important for the protocol and implementation design. In this work we develop, an online adaptive deterministic algorithm. We first formulate the problem in Section 4.3. We then discuss the case when the state space (queue-length and channel state) and the action space (transmission rates) are discretized. We formulate, a post-decision discrete state space based Markov Decision problem, to decrease the storage space.

4.2 System Model

Consider a point to point slotted TDMA system consisting of single transmitter as scheduler and a receiver with a wireless link. The wireless link is characterized by the fading and additive white Gaussian noise. The channel state at the beginning of slot n is denoted by x_n. where x_n denotes the channel gain for the user. The channel process x_n is assumed to take values in finite set X. We assume that the channel state is constant during a slot and change only at slot boundaries. We also assume that there is perfect channel state information (CSI). We assume that the transmitter has a finite buffer of length B.

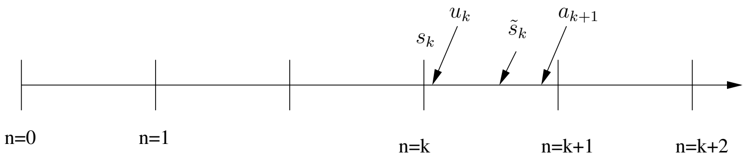


Figure 4.1: System Model

The unit of arrival and transmission can be number of bits or number of packets per slot. The arrival process a_n at the transmitter is Markovian. We assume that arrival occurs at the end of the slot n or at $(n+1)^-$. Let Q_n denote the queue length at the transmitter at slot n^+. Since the queue length is bounded by B and if $Q_n + a_{n+1} \geq B$, then the extra units of data are dropped.

In every time slot [1], the scheduler determines the number of units u_n to transmit and its corresponding power at time n^+, depending on the state of the system at time n. We assume that u_n is discrete and takes values in a finite set U. This condition is equivalent to considering, discrete rates, which is an important constraints in the design of a practical scheduler. We also restrict ourselves to policies such that $u_n \leq Q_n$ is satisfied. Let the set of policies satisfying this constraint be denoted by $\mathcal{U}_{cd} \in \mathcal{U}_d$, where $\mathcal{U}_d$ is the set of all deterministic policies.

In actual systems, the arrivals occur uniformly over the slot. Thus our approach is conservative wherein average delay is about half of the slot more than the actual average delay.

Data enters into the queue and get buffered. The buffer evolution at the transmitter is given as,

$$\begin{aligned} q_{n+1} &= \min\{\max\{Q_n - u_n, 0\} + a_{n+1}, B\} \\ &= \min\{Q_n - u_n + a_{n+1}, B\} \text{ for } u_n \in \mathcal{U}_d. \end{aligned} \tag{4.1}$$

4.3 Problem Formulation

Power minimization problem is states mathematically as follows. Let P_n denote the power required for the transmission at slot n. The sample path dependent average power P is,

$$P = \limsup_{N\to\infty} \frac{1}{N}\sum_{n=1}^{N} P_n, \tag{4.2}$$

The average delay constraint is not tractable from the instantaneous samples. We convert the average delay constraint into average queue length constraint using Little's theorem. Let $\bar{Q}$ be the average queue length corresponding to the average delay constraint $\bar{D}$ and average arrival rate $\bar{a}$

[1] We use time and slot interchangeably

[2]. By Little's theorem,

$$\bar{Q} = \bar{a}\bar{D}$$

In the calculation of the effective average arrival rate $\bar{a}$, we also consider the effect of data dropped due to buffer overflow. We define average buffer overflow as,

$$\epsilon = \limsup_{N\to\infty} \frac{\sum_{n=0}^{N} \max(0, Q_n - u_n + a_{n+1} - B)}{\sum_{n=1}^{N} a_n}. \tag{4.3}$$

ϵ can be considered as equivalent to probability of packet drop.

If a_{avg} is the actual arrival rate, then effective arrival rate, after dropping of packets is given by,

$$\begin{aligned} \bar{a} &= a_{avg}\,(1-\epsilon) \\ \Rightarrow \bar{Q} &\overset{\Delta}{=} a_{avg}(1-\epsilon)\bar{D}. \end{aligned}$$

The average queue length Q_{avg} of online-algorithm is given by,

$$Q_{avg} = \limsup_{N\to\infty} \frac{1}{N}\sum_{n=1}^{N} Q_n.$$

We present here a constrained optimization problem. Our objective is to minimize the power for the point to point transmission, subject to delay constraint $\bar{D}$ and drop probability constraints $\bar{\epsilon}$.

$$\text{Minimize} \quad P \tag{4.4}$$

$$\text{subject to} \quad Q_{avg} - \bar{Q} \leq 0, \tag{4.5}$$

$$\epsilon - \bar{\epsilon} \leq 0. \tag{4.6}$$

Optimization problem (4.4) can be formulated as Constrained Markov Decision Problem (CMDP) and solved using Lagrangian technique described in Appendix B.4 with state at time n: $s_n = (Q_n, x_n)$. The Lagrangian for the above optimization problem is given by,

$$\mathcal{L}(\lambda_1, \lambda_2) = P + \lambda_1 Q_{avg} + \lambda_2\epsilon - \lambda_1\bar{Q} - \lambda_2\bar{\epsilon}, \tag{4.7}$$

where $\boldsymbol{\lambda} = \{\lambda_1 \geq 0, \lambda_2 \geq 0\}$ are the Lagrange multipliers.

The immediate cost of the CMDP is given by,

$$c_n = P_n + \lambda_1\left(Q_n - \bar{D}(a_{n+1} - d_n)\right) + \lambda_2\left(d_n - \bar{\epsilon}a_{n+1}\right), \tag{4.8}$$

[2] We will denote the constraint by '$^-$'

where,

$$d_n = \max\left\{0, Q_n - u_n + a_{n+1} - B\right\}. \tag{4.9}$$

For a given $\boldsymbol{\lambda}$ If there exists an optimal admissible, policy which minimizes 4.4, then there exists vector h called as difference cost, such that,

$$h^*(s) = \min_{u \in U} \mathbf{E}\left\{c(s,u) + h^*(\bar{s}) - h^*(s^0)\right\}, \tag{4.10}$$

for some $s^0 \in [0, B] \times X$ and $\bar{s}$ is the next state of the simulated process. (4.10) is the average cost relative value Bellman equation. For the solving average cost problem, we use relative value iteration (RVI) algorithm for find the optimal. Here w.r.t. a reference state s^0 we perform the following iteration,

$$h_{n+1}(s) = \max_{u \in U}\left\{c(s,u) + \sum_{\bar{s}} P(\bar{s}|s,u)h_n(\bar{s}) - h_n(s^0),\right\} \tag{4.11}$$

, where $P : S \times U \to S$ is the transition matrix. The proof of convergence of above algorithm is presented in [34].

We now discuss the optimality of the dual problem 4.7. d_n is a convex function of policy u_n, hence c_n is a convex function of policy u_n. We can apply Theorem B.1, so that the optimal average cost for the CMDP is given by,

$$\begin{aligned} h^* &= \sup_{\boldsymbol{\lambda}} \inf_{u \in U} \{h^{\boldsymbol{\lambda}}(u)\} \\ &= \inf_{u \in U} \{h^{\boldsymbol{\lambda}^*}(u)\}, \end{aligned}$$

where $\boldsymbol{\lambda}^*$ is the optimal Lagrange multiplier.

We concentrate on determining $\inf_{u \in U}\{h^{\boldsymbol{\lambda}}(u)\}$, where $\boldsymbol{\lambda}$ is constant. The maximization over $\boldsymbol{\lambda}$ will involve, use of multiple time scale stochastic approximation and is considered later in the chapter.

4.4 Online Algorithm

The overview of the algorithm strategy is explained in this section. We first convert CMDP 4.4 into learning framework using post decision based formulation. The motivation of using post decision formulation is given in 4.5. Thus we first perform, RVI based on post decision state (cf. Section 4.5) done on a faster time scale. while the Lagrange multipliers $\boldsymbol{\lambda}$ are updated at a slower time scale. The assumptions and the algorithm is stated in Section 4.5. In Appendix 4.8 the convergence of the post-decision based learning algorithm for CMDP is presented.

4.5 Post-Decision state Formulation

The Post-Decision state is briefly described in Appendix B.3. In our case it is possible to recognize the exact “virtual state”, a state immediately after a decision is made, called *post decision state*. Using a pre-decision method, we take the decision that takes into account the impact of future. Thus moving into the state before the decision is taken makes the state s_{n+1}, a random variable. A simulation based learning algorithm, equivalent to the value iteration is not possible because the expectation operator is inside the *min* operator in RVI (4.11). To circumvent this problem, if we consider a state after the decision is taken, we can interchange the **E** and *min* operators in (4.11). Recursion built around the post-decision state variable provides us with the direct control over the structure of the value function, which can be exploited in the design of algorithms equivalent to the value iteration [3]. Thus the algorithms built using the post-decision variable reduce the space complexity.

The state variables and value function are indexed with superscript “$\sim$”, to denote the post-decision state variable. We now define function S^d, which maps the pre-decision state to the post decision state and S^p which maps post decision state to the pre-decision state. With these definitions, following relations are easy to obtain.

$$\begin{aligned}
\tilde{s}_n &= S^d(s_n, u_n) \\
\tilde{Q}_n &= Q_n - u_n \\
\tilde{x}_n &= x_n \\
s_n &= S^p(\tilde{s}_n, a_{n+1})
\end{aligned}$$

We will now sketch the value iteration algorithm based on the post- decision state.
Define:

$$\tilde{h}_{n+1}(\tilde{s}_n) = \mathbf{E}[h_n(s_{n+1})|\tilde{s}], \tag{4.12}$$

Using (4.12) and Relative Value Iteration (4.11), we can write (4.10) using the post-decision state variable as,

$$\tilde{h}_{n+1}(\tilde{s}) = \mathbf{E}\left[\min_{u \in \mathcal{U}_d} \left\{ c(s, \boldsymbol{\lambda}, u) + \tilde{h}_{n+1}(\tilde{\dot{s}}|\tilde{s}_n) - \tilde{h}(\tilde{s}^0) \right\}\right] \tag{4.13}$$

which leads to following synchronous post decision based relative value iterations with respect to

[3] Otherwise we can also use Q-learning algorithm.

some arbitrary state s^0:

$$\begin{aligned}
\tilde{h}_{n+1}(\tilde{s}) &= \tilde{h}_n(\tilde{s}) + \alpha(n)\Bigg[\min_{u\in\mathcal{U}_d}\Big\{c_n(s_n,\boldsymbol{\lambda},u) \qquad (4.14)\\
&\qquad +\tilde{h}_n(\dot{\tilde{s}})\Big\} - \tilde{h}_n(\tilde{s}^0) - \tilde{h}_n(\tilde{s})\Bigg] \quad \forall\tilde{s}\\
\tilde{h}_0(\tilde{s}) &= 0 \quad \forall\tilde{s}, \qquad (4.15)
\end{aligned}$$

where $\alpha(n)$ are the step sizes (specified in 4.19) An online version of the RVI should use asynchronous updates in which, updates occur only to that component of value function corresponding to the state of actual visit. Let

$$\nu(\tilde{s},n) = \sum_{m=0}^{n} I\{\tilde{s} = \tilde{s}_m\}, \quad \forall\tilde{s}\in S, \tag{4.16}$$

where $I(\cdot)$ is an indicator function.

$$\begin{aligned}
\tilde{h}_{n+1}(\tilde{s}) &= \tilde{h}_n(\tilde{s}) + \alpha(\nu(\tilde{s},n))I\{\tilde{s}_n=\tilde{s}\}\Bigg[\min_{u\in\mathcal{U}_d}\Big\{c(s,\boldsymbol{\lambda},u) \qquad (4.17)\\
&\qquad +\tilde{h}_n(\dot{\tilde{s}})\Big\} - \tilde{h}_n(\tilde{s}_0) - \tilde{h}_n((\tilde{s})\Bigg] \quad \forall\tilde{s}
\end{aligned}$$

For the asynchronous algorithm we need following additional assumptions on the step sizes for the convergence as stated in [35].

Assumption 4.1 *If $[z]$ denotes the integer part of z, then for $x\in(0,1)$,*

$$\sup_k \frac{\alpha([xk])}{\alpha(k)} < \infty \tag{4.18}$$

$$\frac{\sum_{m=0}^{[yk]}\alpha(m)}{\sum_{m=0}^{k}\alpha(m)} \to 1 \textit{ uniformly in } y\in[x,1]$$

Assumption 4.2 *There exists $\Delta > 0$ such that,*

$$\liminf_{n\to\infty}\frac{\nu(\tilde{s},n)}{n+1} \geq \Delta \ a.s.$$

Also, for all $x\geq 0$ and

$$N(t,x) = \min\{m\geq n : \sum_{k=n}^{m}\alpha(k)\leq x\},$$

the limit

$$\lim_{n\to\infty} \frac{\sum_{k=\nu(n,\tilde{s})}^{\nu(N(n,x),\tilde{s})} \alpha(k)}{\sum_{k=\nu(t,\tilde{s})}^{\nu\left(N(n,x),\tilde{\tilde{s}}\right)} \alpha(k)}$$

exists for all $\tilde{s}, \tilde{\tilde{s}}$

The assumptions imply that all the states are updated often in and evenly manner.

Taking a cue from Q-learning (cf. Appendix B.5.2), let us call RVI 4.17 based on Post decision state as "Post-learning" algorithm.

The next step would be to optimize with respect to $\boldsymbol{\lambda}$, to obtain $\boldsymbol{\lambda}^*$ and thus the optimal policy. Assume that we have obtained optimal policy for given $\boldsymbol{\lambda}$. Using two time scale stochastic approximation [36], RVI algorithm to sees $\boldsymbol{\lambda}$ as a constant. Thus we first perform primal minimization on a faster time scale, while the dual maximization is performed on a slower time scale. The $\boldsymbol{\lambda}_n$ iterations are performed on a slower time scale, This effect is obtained if averaging step sizes $\alpha(n)$ and $\beta(n)$, satisfy,

$$\begin{aligned}
&\sum_n \alpha(n) = \infty, \quad \sum_n \beta(n) = \infty, \\
&\sum_n \alpha(n)^2 + \sum_n \beta(n)^2 < \infty, \\
&\lim_{n\to\infty} \frac{\beta(n)}{\alpha(n)} \to 0.
\end{aligned} \tag{4.19}$$

As the boundedness of $\boldsymbol{\lambda}$ is not easy to ensure, we project the iterates $\boldsymbol{\lambda}_n$ into the interval $[0, K_1] \times [0, K_2]$, using projection functions Γ_1 and Γ_2 for λ_1 and λ_2 respectively, where K_1, K_2 are chosen so that so that $\boldsymbol{\lambda}^* \in [0, K_1] \times [0, K_2]$ and the $\boldsymbol{\lambda}$ iterates are given by,

$$\lambda_{1_{n+1}} = \Gamma_1[\lambda_{1_n} + \beta(n)\left(Q_n - \bar{D}(a_n - d_n)\right)] \tag{4.20}$$

$$\lambda_{2_{n+1}} = \Gamma_2[\lambda_{2_n} + \beta(n)\left(d_n - \bar{\epsilon} a_n\right)] \tag{4.21}$$

4.6 Simulation Results

In this section we perform experimental study of the performance of the proposed algorithm. The transmission power required at the nth slot is given by,

$$P_n = \frac{N_0 W}{x_n}\left(e^{u_n/W} - 1\right),$$

where N_0 is the spectral density AWG noise of the channel and W is the spectrum bandwidth of the wireless link. We consider transmission over a wireless channel of bandwidth $W = 500$ KHz and noise variance bandwidth product, $N_0 W = 0.39$. The fading is modeled using Rayleigh channel, whose probability density function is given by $\mu(x) = \frac{1}{\gamma}e^{\frac{-x}{\gamma}}$, where $\gamma > 0$. We assume channel is i.i.d. across slots. and is modeled with parameter $\gamma = 1$ We discretize the channel into equal probability regions as shown in Figure 4.2.

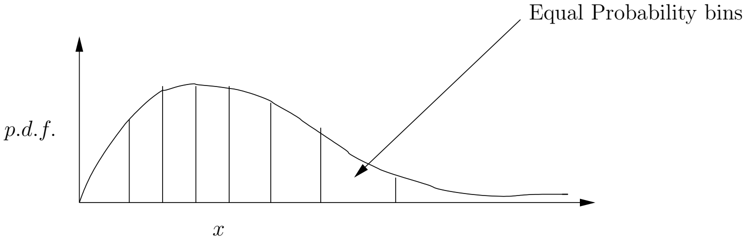


Figure 4.2: Discretization of Rayleigh channel

We quantify data in terms of packets transmitted. We assume packet of size of 1 Kb. We particularly consider discrete rate transmission with rates $u = \{0, 200, 400, 800\}$ Kbps. We assume slot length of 10ms. These rates corresponds to different modulation schemes, selected depending on the state of the system. We assume Poisson arrival of packets in each time slot. For the simulation purpose, it is required that the algorithm should explore actions at all states. This is achieved by running soft-max policy during initial time slots and thereafter apply the greedy algorithm.

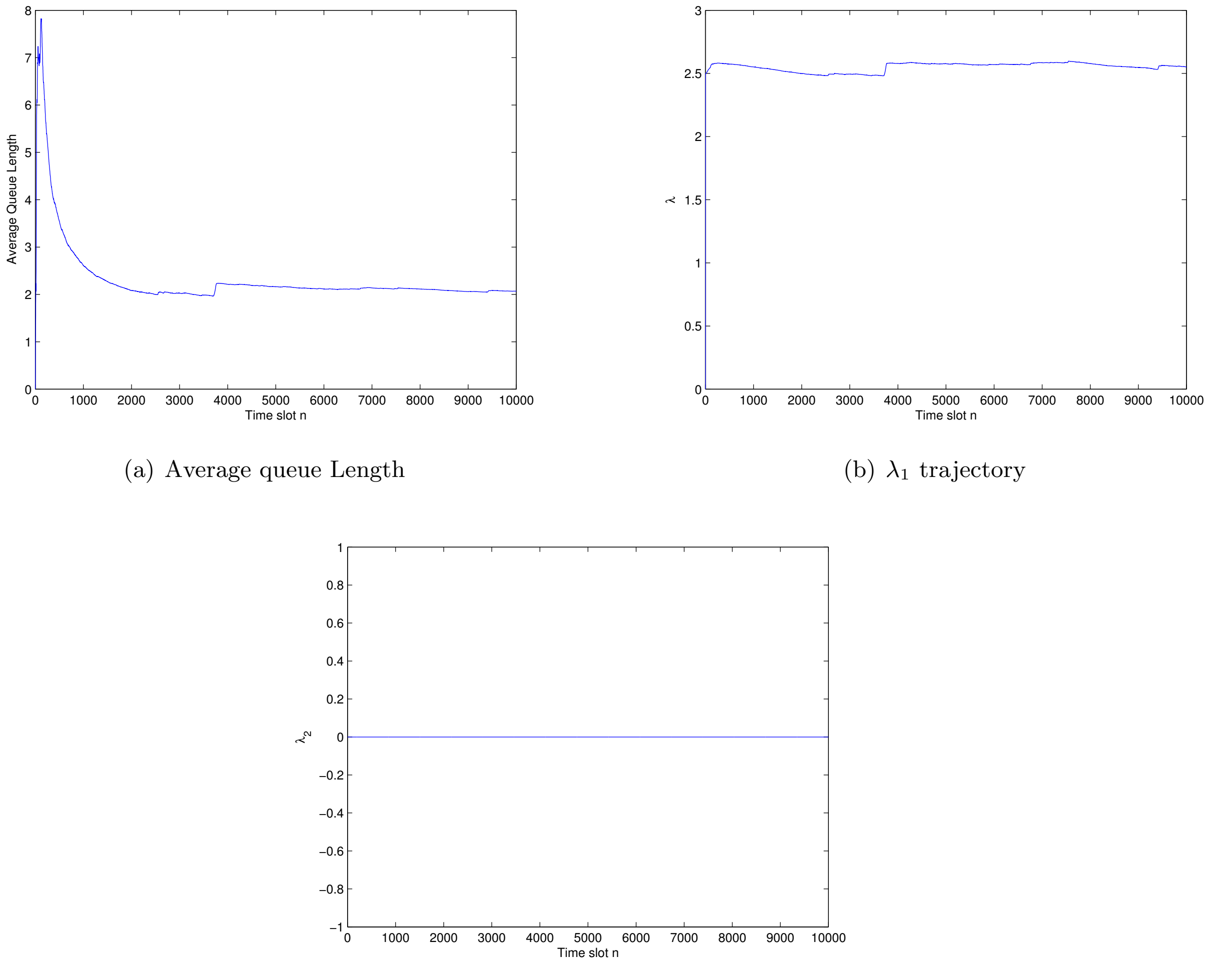
First we consider a buffer of large size $B = 100$ Kb, and average arrival rate of 1 packet. The system helps us to analyze the power delay curve without taking into consideration the effect of data drop. Figure 4.5 shows the plot of power-delay curve. As the constant on the delay increases the average transmission power required decreases. The plot also demonstrates the convex characteristics of the power-delay curve.

The second experiment demonstrates the effect of weak and tight delay constraints under a finite buffer assumption. We take the buffer of size $B = 50$Kb and average arrival rate $\lambda = 2$ packets. For tight delay constraint we assume that $\bar{D} = 1$ packet and $\bar{\epsilon} = 0.1$. Figure 4.3 shows the plots for the tight delay constraint. In this case the convergence occurs faster within 2000 slots.

For weak delay constraint we assume that $\bar{D} = 100$ packets and $\bar{\epsilon} = 0.1$. Figure 4.4 shows the plots for the tight delay constraint. Here the convergence takes nearly 6000 slots for actual convergence.

For tight delay constraint, Figure 4.3(c) shows that $\lambda_2 = 0$ for all iterations, which means that there are few/no data drop events. The algorithm has succeeded in maintaining the queue length small and nearly satisfying the delay constraint. For weak constraint, we have the opposite effect. The constraint of $\bar{D} = 100$ packets, would never be satisfied with $B = 100$ Kb. The algorithm now tries to keep the buffer drop below the specified constraint. In this case, λ_1 will remain close to 0, demonstrating that the the delay constraint is satisfied with equality.

Remark 4.1 *In the experiments, it is observed, for the decreasing step sizes $(\alpha(n), \beta(n))$ con-*



(a) Average queue Length

(b) λ_1 trajectory

(c) λ_2 trajectory

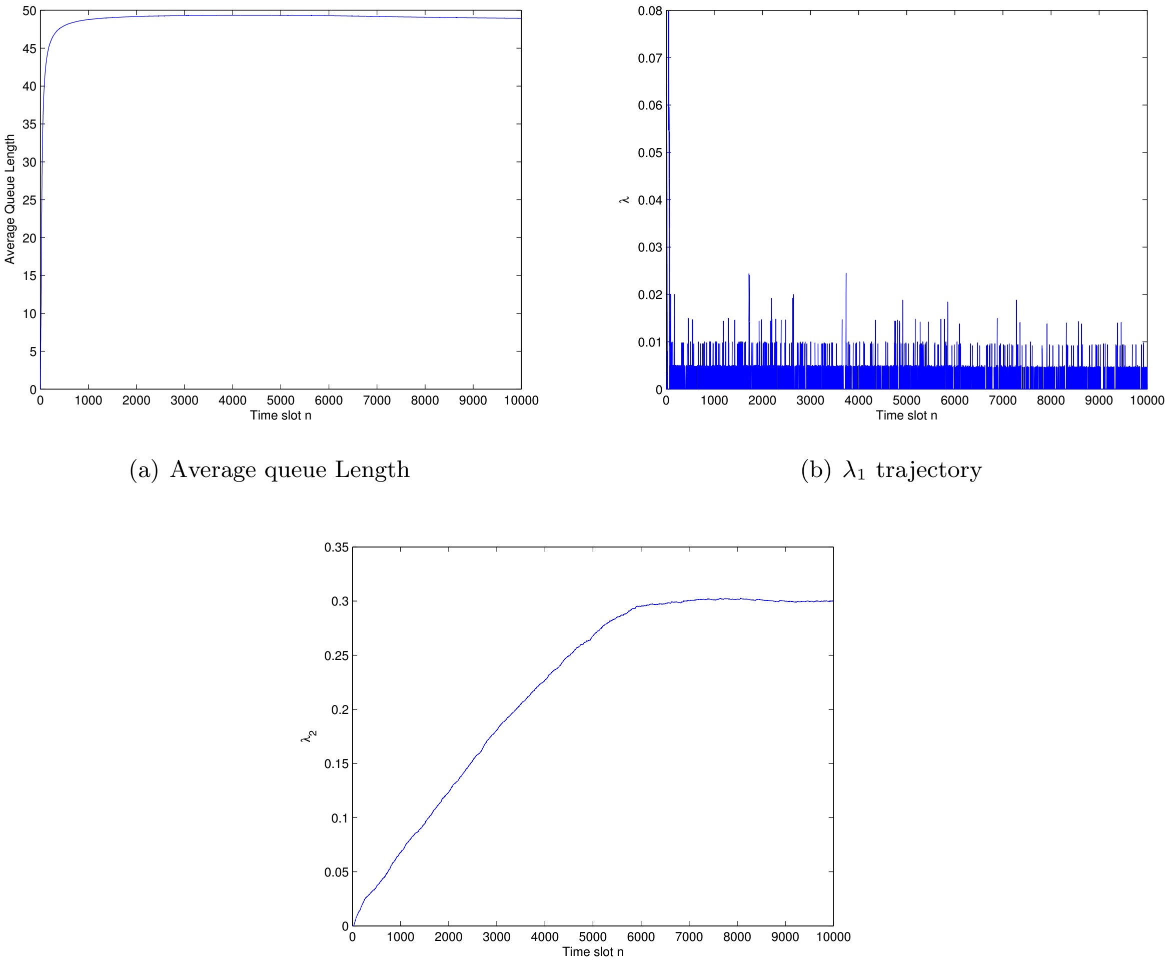
Figure 4.3: Convergence plots for $\bar{D} = 1$ and $\epsilon = 0.1$ (finite state space)

straints are not satisfied. One possible reason would be value estimation rate is faster than state exploration rate. In order to speed up the algorithm and to allow for rapid state exploration, in the experiments, we have used constant sequences e.g. $\alpha = 0.01 \leq 1$, $\beta = 0.001 \leq 1$. *In the literature [37] convergence of ODE for decreasing step sequences and for constant bounded sequences is given.*

4.7 Scheduling for Continuous Channel and Large Buffer size

In the previous sections, we have assumed discretized channel, and used finite state space formulation for the Markov Decision problem. This formulation faces two noticeable deficiencies.

- The underlying channel is not actually discrete, but continuous
- As the buffer size is increased, the time and space complexity increase exponentially. For



(a) Average queue Length

(b) λ_1 trajectory

(c) λ_2 trajectory

Figure 4.4: Convergence plots for $\bar{D} = 100$ and $\epsilon = 0.1$ (finite state space)

the learning based algorithms with very large state space, it becomes impossible to learn for each state.

We use function approximation based technique within actor-critic framework (cf. Appendix B.8) for avoiding these difficulties. We assume queue length to be continuous for analysis. If we further assume that the arrival and departure occur in multiples of the fixed amount fluid then the analysis will also hold for the discrete arrival and discrete rate scenario. Unconstrained improved temporal difference method for solving average cost MDP is discussed in Appendix B.6. In this section, we state the function approximation based algorithm for solving constrained MDP without the convergence proof. We use multiple time scale stochastic approximation method for solving the constrained MDP. The unconstrained algorithm is solved on a faster time scale, while the Lagrange multiplier are updated on a slower time scale. The details about the convergence of

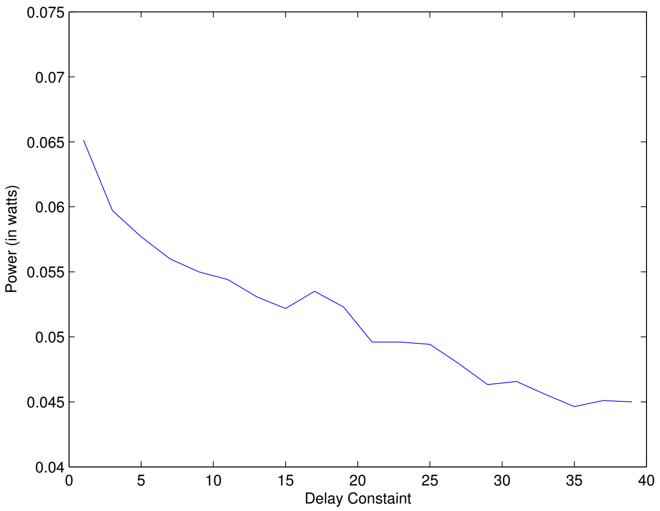


Figure 4.5: Power Delay curve with finite state space

the unconstrained problem is given in Appendix B.6. For the constrained multiple time scale the slower scale analysis follows in a similar way as discussed in Section 4.8 and not stated to avoid repetition. Let the difference value function be given by,

$$h(x) = \sum_{i=1}^{K} f_i(x) r_i, \tag{4.22}$$

where $f = [f_1, f_2, \cdots f_K]$, are the feature vectors and $r = [r_1, r_2, \cdots r_K]$ are the corresponding weights. The weight update is given in Algorithm 2.

4.7.1 Simulation Results

The simulation assumptions are the same as discussed in Section 4.6. except that we assume continuous Rayleigh channel distribution . We take parameter $\Lambda = 0.99$ and the feature vectors as, $f = [1, Q, x, Qx]$

First we consider a buffer of large size $B = 100$ Kb, and average arrival rate of 1 packet. The system helps us to analyze the power delay curve without taking into consideration the effect of data drop. Figure 4.7.1 shows the plot of power-delay curve. As the constraint on the delay increases the average transmission power required decreases. The plot also demonstrates the convex characteristics of the power-delay curve.

The second experiment demonstrates the effect of weak and tight delay constraints under a finite buffer assumption. We take the buffer of size $B = 50$ Kb and average arrival rate $\lambda = 2$. For tight delay constraint we assume that $\bar{D} = 1$ and $\bar{\epsilon} = 0.1$. Figure 4.7 shows the plots for the tight delay constraint. For weak delay constraint we assume that $\bar{D} = 100$ and $\bar{\epsilon} = 0.1$. Figure 4.8

Algorithm 2 Algorithm for weight updates

$$\begin{aligned}
r_{n+1} &= r_n + \alpha(n)\bar{B}_n \sum_{k=0}^{n} \left(\sum_{m=0}^{k} \Lambda^{k-m} f(s_m) \right) e_n(s_k, s_{k+1}) && (4.23) \\
e_n(s_n, s_{n+1}) &= c(s_n, s_{n+1}) - \phi_n + \alpha(n) \left(f(s_{n+1}) - f(s_n) r_n \right), \forall k, n \\
\phi_{n+1} &= \phi_n + \beta(n) \left(c(x_n, x_{n+1}) - \phi_n \right) && (4.24) \\
\bar{B}_{n+1} &= \bar{B}_n - \frac{\bar{B}_n f(s_{n+1}) f'(s_{n+1}) \bar{B}_n}{1 + f(s_{n+1})' \bar{B}_n f(s_{n+1})} \\
Q(s_n, u_n) &= r(s_n, u_n) + V(s_{n+1}) \\
\lambda_{1_{n+1}} &= \Gamma_1 [\lambda_{1_n} + \gamma(n) \left(Q_n - \bar{D}(a_n - d_n) \right)] && (4.25) \\
\lambda_{2_{n+1}} &= \Gamma_2 [\lambda_{2_n} + \gamma(n) \left(d_n - \bar{\epsilon} a_n \right)] && (4.26)
\end{aligned}$$

where sequences $\alpha(n)$ and $\gamma(n)$, satisfy,

$$\begin{aligned}
&\sum_n \alpha(n) = \infty, \quad \sum_n \beta(n) = \infty, \quad \sum_n \gamma(n) = \infty \\
&\sum_n \alpha(n)^2 + \sum_n \beta(n)^2 + \sum_n \gamma(n)^2 < \infty \\
&\lim_{n \to \infty} \frac{\gamma(n)}{\alpha(n)} \to 0 \\
&\beta(n) = c\alpha(n) \text{ for some constant } 0 < c \leq 1
\end{aligned}$$

shows the plots for the tight delay constraint.

In Appendix B.6, we proved the existence of fixed point for Algorithm 2. However nothing can be said about the uniqueness and optimality of Algorithm 2. For weak delay constraint, Figure 4.8(a) shows that average queue length Q_{avg} actually decreases, rather than being near the buffer size. This behavior is due to non exploration of policies during simulation or because of the non uniqueness in the algorithm is yet to be researched.

4.8 Conclusions and Discussions

In this chapter, we have designed an online power optimal scheduling algorithm with average delay constraint and finite buffer. We have dealt with finite state space and continuous state space separately.

For the finite state space, we have used post decision based formulation of Markov Decision Process. We first proved the convergence of the algorithm using multi time scale stochastic approximation. We showed the convexity of the power delay curve, a well known fact, experimentally.

For the continuous channel and large buffer size, we represented value function using function approximation. We used temporal difference algorithm for value function improvement and soft-

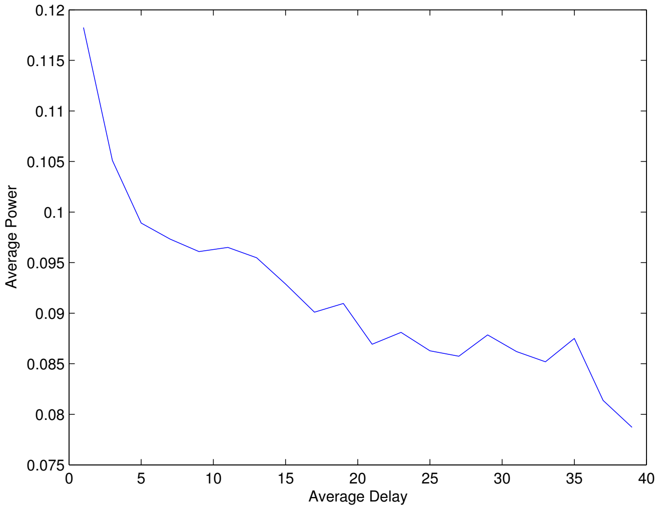


Figure 4.6: Power Delay curve with continuous state space

max for policy improvement. We proved existence of fixed point. however we were not able to able to prove the exact convergence. The experiments showed that the delay constraints are met without equality. This may be because of non-exploration or non-uniqueness of the algorithm. The issue is further required to be studied.

Here, it should be noted that, our aim here is to highlight the issues in the development of algorithm for delay constrained power optimal scheduling and thus considered only point to point wireless link. The algorithm developed can be extended to multiple users but, with increased complexity (state space).

Appendix

Convergence of the Post-learning Algorithm

We recourse to two time scale ODE analysis of stochastic approximation algorithms [36].

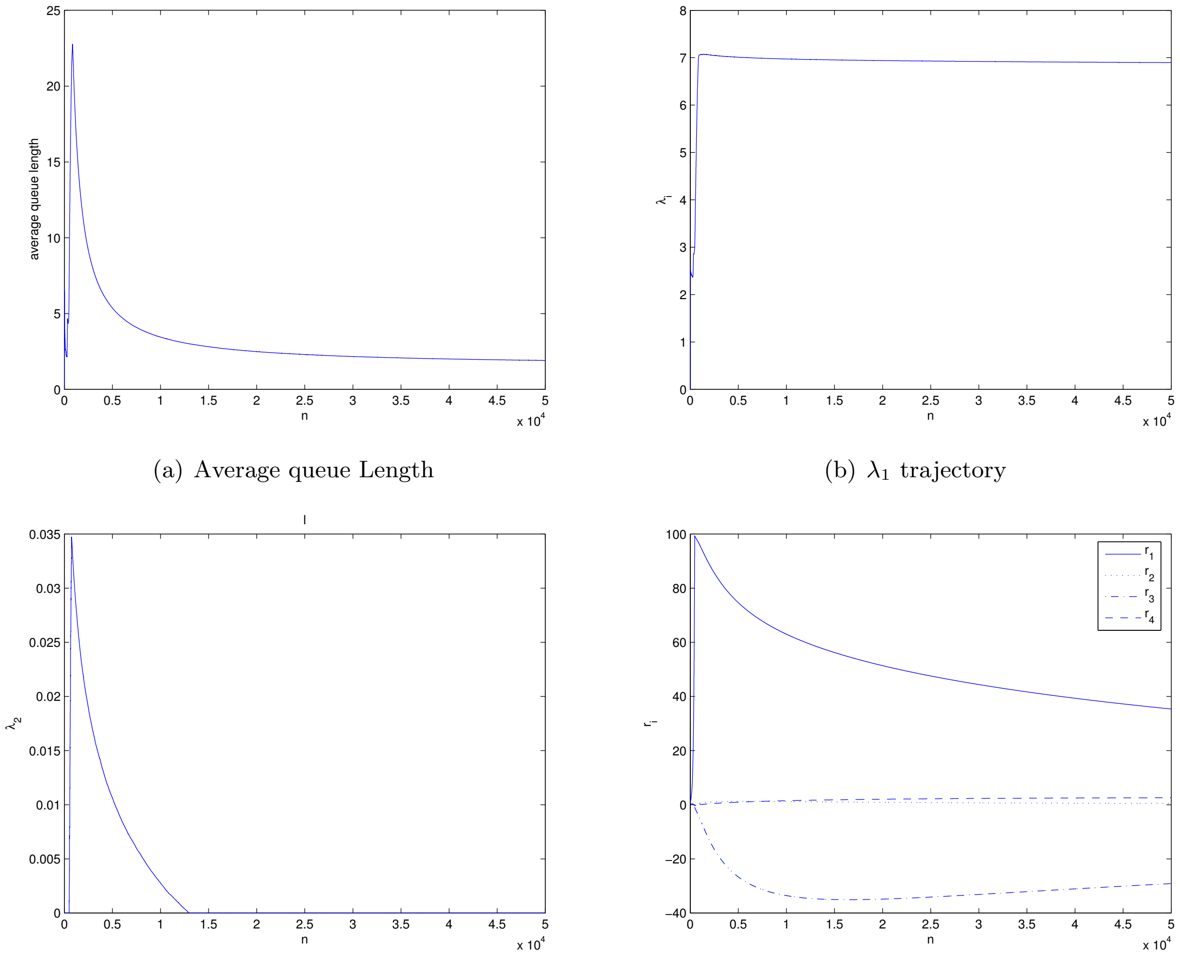
Assumption 4.3 *Iterates $\tilde{h}_n$ are bounded.*

We rewrite the RVI Algorithm 4.11 as,

$$\tilde{h}_{n+1} = \tilde{h}_n + \alpha(\nu(\tilde{s}, n)) \left(T(\tilde{h}_n) - \tilde{h}_n(\tilde{s}_0)e - \tilde{h}_n + M_{n+1} \right), \tag{4.27}$$

where e is a $|S| \times 1$ vector with all entries 1. The map $T : |S| \times 2 \rightarrow |S| \times 1$ is defined by,

$$T\tilde{h}(\tilde{s}, \boldsymbol{\lambda}) = \sum_{\dot{s}} P(\tilde{\dot{s}}|\tilde{s}) \min_u \left\{ c(s, \boldsymbol{\lambda}, u) + \tilde{h}(\tilde{\dot{s}}) \right\} - \tilde{h}_n(\tilde{s}_0)e$$



(a) Average queue Length

(b) λ_1 trajectory

(c) λ_2 trajectory

(d) r trajectory

Figure 4.7: Convergence plots for $\bar{D} = 1$ and $\epsilon = 0.1$ (continuous state space)

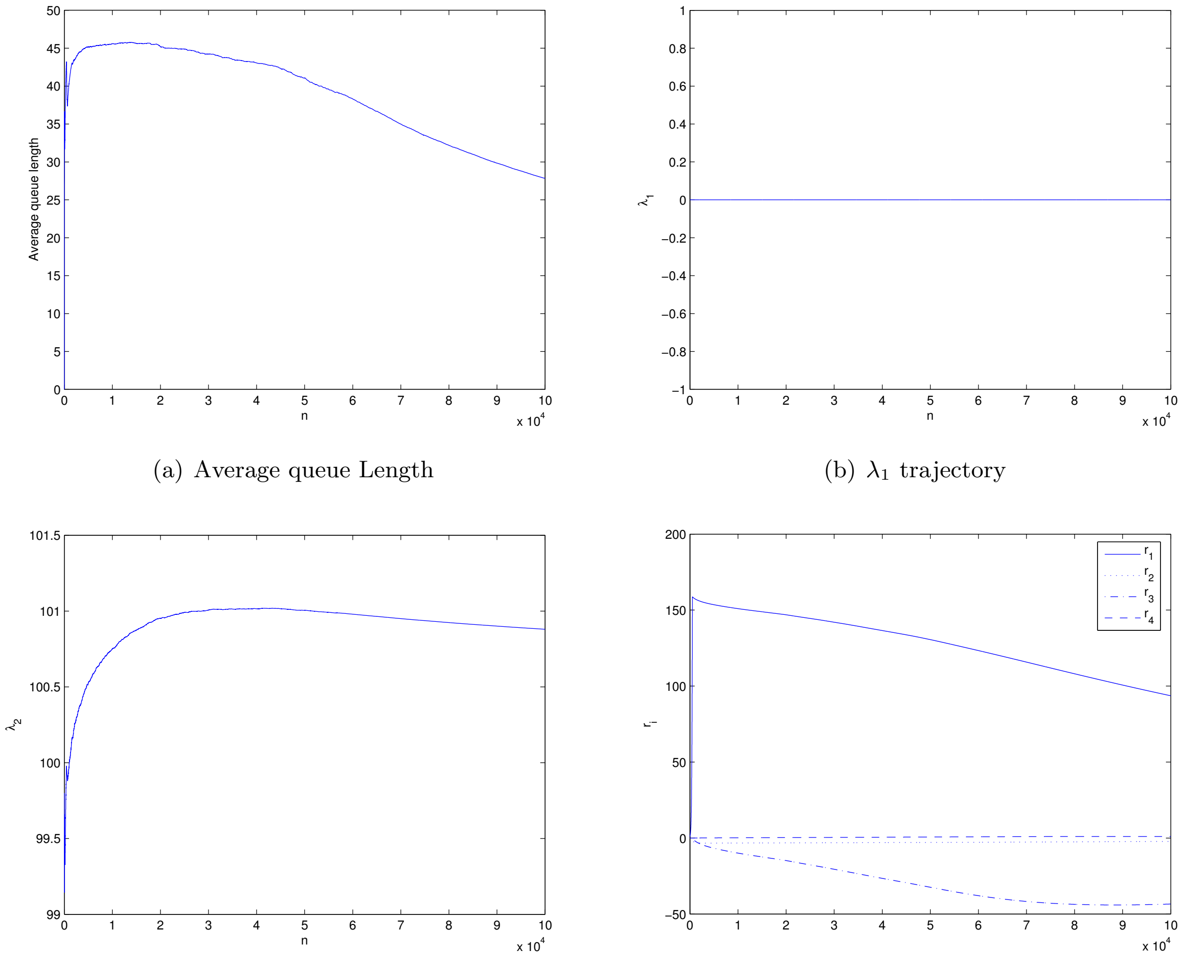
and the Martingale sequence M_{n+1} is given by,

$$M_{n+1}(s) = \min_{u}\{c(s, \boldsymbol{\lambda}, u) + \tilde{h}_n(\dot{s}) - T\tilde{h}_n(\tilde{s})\}$$

Let $\mathcal{F}_n \triangleq \sigma(\tilde{h}_n, M_n), t \geq 0$, denote the increasing family of sigma fields. We observe that,

1. $\mathbf{E}[M_{n+1}|\mathcal{F}_n] = 0$.
2. $\mathbf{E}[||M_{n+1}||^2|\mathcal{F}_n] \leq C(1 + ||\tilde{h}_n||^2)$, for some constant $C \geq 0$. This can be easily proved by noting that, $\mathbf{E}\,||h_n|| \leq C_1 \left|\left|\tilde{h}_n\right|\right|$ for some constant $C_1 \geq 0$.

These relations are the basic assumptions in converting the Iterations 4.11 to ODE form. By the theory of two time scale stochastic approximation, we can treat $\boldsymbol{\lambda}$ as a constant in the basic RVI



(a) Average queue Length

(b) λ_1 trajectory

(c) λ_2 trajectory

(d) r trajectory

Figure 4.8: Convergence plots for $\bar{D} = 100$ and $\epsilon = 0.1$ (continuous state space)

algorithm It can be shown that, iterations track following ODE's asymptotically [35],

$$\dot{\tilde{h}}(t) = T(\tilde{h}(t), \boldsymbol{\lambda}) - \tilde{h}(t). \tag{4.28}$$

We note that the RHS of the above ODE is Lipschitz continuous and hence the ODE has a unique solution. The deterministic sequence of policies for a given $\boldsymbol{\lambda}$ is equivalent to some randomized policy $\mu^{\boldsymbol{\lambda}}(s|u)$. Let $\pi^{\mu}(s)$ be the steady state probability for state s by employing policy and thus $\mu^{\boldsymbol{\lambda}}(s|u)$

$$\begin{aligned} X^{\boldsymbol{\lambda}} &\triangleq \sum_{s} \pi^{\mu}(s) x \\ \epsilon^{\boldsymbol{\lambda}} &\triangleq \sum_{s} \pi^{\mu}(s) \epsilon(s) \end{aligned}$$

The Lagrange multipliers track the ODE,

$$\begin{aligned}\dot{\lambda}_1(t) &= x(\boldsymbol{\lambda}(t)) - \bar{X}, && (4.29)\\ \dot{\lambda}_2(t) &= \epsilon(\boldsymbol{\lambda}(t)) - \bar{\epsilon}, && (4.30)\end{aligned}$$

Let $\tilde{h}^{\boldsymbol{\lambda}}$ denote the value function for given $\boldsymbol{\lambda}$. We now fix $\boldsymbol{\lambda}$ and prove convergence results for Equation 4.28.

Lemma 4.1 *Equation 4.28 has a unique equilibrium point at $\tilde{h}$.*

Proof: Average cost Bellman's equation involving post decision state, has solutions of the form $\tilde{h} = \tilde{h}^* + ce$, where c is a constant and unique h^*, such that $h^* = \phi$, where ϕ is the average cost. Thus $T(\tilde{h}^*) = \tilde{h}^*$ and $\tilde{h}^*$ is the equilibrium point of Equation 4.28.

Lemma 4.2 *$\tilde{h}^*$ is the globally asymptotically stable equilibrium point for Equation 4.28.*

Proof: The proof can be given in an analogous way as in [38].

The conversion from recursive iteration to ODE 4.28 requires boundedness of iterates $\tilde{h}_n$. We initially have assumed this fact. Now we prove that indeed iterates $\tilde{h}_n$ are bounded.

Lemma 4.3 *The iterates $\tilde{h}_n$ remain bounded a.s.*

Proof: We use results from [37]. Consider a function,

$$T^s\tilde{h}(\tilde{s}) = \sum_{\dot{s}} P(\dot{s}|\tilde{s})\tilde{h}(\tilde{\dot{s}}). \tag{4.31}$$

Then

$$\lim_{r\to\infty} \frac{T(r\tilde{h}, \lambda)}{r} = T^s \tag{4.32}$$

and the ODE,

$$\dot{\tilde{h}}(t) = T^s(\tilde{h}) - \tilde{h}, \tag{4.33}$$

has the origin as globally asymptotically stable equilibrium. Thus the results of [37] apply and we conclude that the iterates are bounded.

Lemma 4.4

$$\left|\left|\tilde{h}_n - \tilde{h}(\lambda^n)\right|\right| \to 0. \tag{4.34}$$

Proof: $\tilde{h}$ is piecewise linear and concave decreasing function of $\boldsymbol{\lambda}$. Hence the function $\tilde{h}$ is continuous function of $\boldsymbol{\lambda}$. By [36] the result follows.

Theorem 4.1 *The iterates $\{\tilde{h}_n, \boldsymbol{\lambda}_n\} \to \{\tilde{h}^*, \boldsymbol{\lambda}^*\}$ a.s.*

Proof: Let $G(\boldsymbol{\lambda}) = \mathbf{E}_{\mu}^{\boldsymbol{\lambda}}[c_n]$ Now for maximizing with respect to $\boldsymbol{\lambda}$, the equivalent gradient scheme is given by,

$$\dot{\boldsymbol{\lambda}}(t) = \nabla G(\boldsymbol{\lambda}(t)) \tag{4.35}$$

(4.29),(4.30) are equivalent to (4.35) as G is perfectly differentiable. By considering $\nabla G(\boldsymbol{\lambda}(t))$ as Lyapunov function and noting that we note,

$$-\left|\nabla G(\boldsymbol{\lambda}(t))\right|^2 < 0. \tag{4.36}$$

Thus the iterates $\boldsymbol{\lambda}_n$ converges almost surely to the maximum of G. Hence $\tilde{h}^{\boldsymbol{\lambda}}$ converges to optimal $\tilde{h}^*$.

Proposition 4.1 *If there exists an admissible policy then, the Lagrange multipliers satisfy the relation,* $\lambda_1\lambda_2 = 0$*, (except possibly at one set of constraints).*

Proof: This relationship implies that, both the constraints cannot remain active simultaneously. Let $\hat{Q} = a_{avg}\bar{D}$. Let q denote the queue length. First constraint will keep average queue length $Q \leq \hat{Q}$.

1. Case 1: $\hat{Q} < B$ For an admissible policy $Q < \hat{Q}$. Consider $|Q - Q_{avg}|$ as the Lyapunov function. There exists a ball B^{δ} at Q, such that $P(Q - Q_{avg} > \delta) < \theta, \quad \theta > 0$, implying $\epsilon < \theta$. Hence the second constraint is satisfied without equality. i.e. $\lambda_2 = 0$.

2. Case 2: $\hat{Q} \geq B$ For an admissible policy $Q_{avg} < \bar{D}(1 - \bar{\epsilon})$, previous argument works and $\lambda_2 = 0$. For $Q_{avg} > \bar{D}(1 - \bar{\epsilon})$, $\lambda_1 = 0$. However the arguments fails if $Q_{avg} = \bar{D}(1 - \bar{\epsilon})$.

Chapter 5

Energy Efficient Video Transmission over Wireless Networks

In the previous chapters, our main objective has been to provide Quality of Service (QoS) guarantees like minimum rate, average delay and fairness as well minimize the power requirement. In this chapter, we present the performance of video application. In particular we consider, power minimization with a long term distortion requirement and absolute delay constraint and thus defers from our previous exposition (cf. Section 4).

Shannon's theory states that source coding and channel coding can be separately dealt with. The underlying impractical assumption is that, source coding has infinite symbol length. Hence joint source and channel coding has major impact in effective use of resources. Joint source and channel coding has been an important research area, particularly for video transmission in wireless networks. In this chapter, we develop on-line joint source and channel coding scheme which exploits channel variations using power control in video transmission. Most of the work in the joint coding has not considered the power adaptation and exploited the channel variations. We use reinforcement learning framework to design an online joint coding scheme for power minimization in point to point video transmission. In our work, the terms source coding and channel coding defers from the standard nomenclature for these terms. We would use term source coding as the quantization steps used for quantizing the video and the channel coding for the transmission rate.

The optimization problem, that we consider can be formulated as minimizing power under the long term distortion constraint and absolute delay constraint to deliver a sequence of frames. The channel feedback is considered for adapting the rate of the transmission. This problem can be modeled as a Markov Decision Problem and can be solved using reinforcement learning techniques for on-line implementation. We use Q-learning algorithm for learning the environment and producing decisions for every Macro Block (MB) resulting in encoding parameters and the rate of transmission for each MB.

5.1 Introduction

In this chapter, we intend to consider the interaction of video compression, resulting distortion, transmitter power and rate adaptation. The goal is to efficiently utilize transmission energy while meeting the delay and video quality constraints imposed by a video streaming application.

The traditional approach of rate control for MPEG-4 video based [39] on second-order rate-distortion model. The focus of the rate adaptation algorithm in [40] is on the video characteristics and not on the channel over which the bit stream is transmitted.

In wireless networks, due to the user's mobility, the channel behavior is inherently time-varying, with periods of good channel alternating with periods of high error rates. Hence, error resilience and error concealment become extremely important. A complimentary approach in [41] is to adapt the behavior of the video encoder to the conditions of the channel. Usually, the behavior of the video encoder and decoder is adapted to cope with the effects of the lossy and time-varying channel. Apart from distortion, transmission of real time video mandates that the absolute delay requirement are also met [42].

In [43], the authors formulate the problem as finite horizon Markov Decision problem and assume that channel transition matrix is apriori known to the transmitter. The Macro Block (MB) level parameters (quantization and transmission rate) are obtained by minimizing the power while meeting constraint on the average distortion to transmit the video sequence over wireless channel and the absolute delay constraint.

We frame the power optimization problem as a constrained finite horizon Markov Decision Process (cf. Appendix B). MDP assumes the underlying transition model known to the transmitter. Hence a model free approach for the channel (environment) is required for the on-line implementation of the joint coding algorithm. The transmitter should be able to learn the channel behaviors and adapt to the changes to determine optimal solution based on the history of decisions and channel conditions. Reinforcement learning methods are suited for the model free optimization of MDP. We use Q-learning method modified suitably for the finite horizon MDP.

In Section 5.2, we describe the Q-learning algorithm for finite horizon MDP. In Section 5.3, we present system model and state the problem formulation in detail. Section 5.5 demonstrates the simulation details and results.

5.2 Finite Horizon Q learning

We consider finite state space and action space, N horizon MDP. The expected cost for the finite horizon MDP, is given by,

$$V = \mathbf{E}\sum_{n=0}^{N} c(s_n, u_n), \tag{5.2}$$

where s_n is the state and policy u_n is the action at horizon n. Q learning algorithm discussed in Appendix B.5.2 works for infinite horizon problems which involve stationary policies. Finite

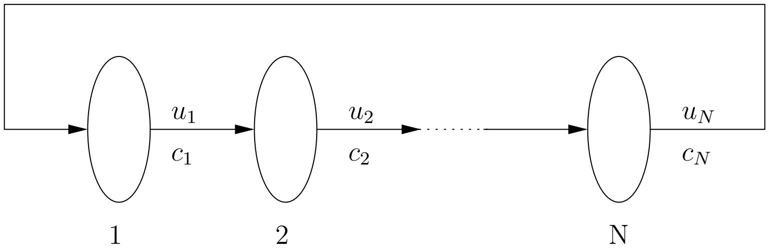


Figure 5.1: Finite horizon Q learning

Algorithm 3 Algorithm for Finite Horizon average cost Q learning

$$
\begin{aligned}
Q_{n+1}(s,u) &= Q_n(s,u) + \alpha_n(s,u)e_n \\
\phi_{n+1} &= \phi_n + \beta_n e'_n, \\
e_n &= \begin{cases} c_n - \phi_n + max_b Q_n(y_n,b) - Q_n(s,u) \\ \text{if } (s,u) = (s_n,u_n), s_n \in S_i, i < N \\ c_n - \phi_n \\ \text{if } (s,u) = (s_n,u_n), s_n \in S_N \\ 0 \text{ otherwise} \end{cases}, \\
e'_n &= \begin{cases} c_n - \phi_n + max_b Q_n(y_n,b) - Q_n(s,u) \\ \text{if } s = s_n, s_n \in S_i, i < N \\ c_n - \phi_n \\ \text{if } s = s_n, s_n \in S_N \\ 0 \text{ otherwise} \end{cases}, \qquad (5.1)
\end{aligned}
$$

$$\sum \alpha_n = \infty, \sum \alpha_n^2 < \infty \ , \sum \beta_n = \infty, \ \sum \beta_n^2 < \infty.$$

horizon problems may have non-stationary policies and the direct application of Q learning is not possible. A simple technique, which converts finite horizon problem into infinite horizon is presented in [44]. Figure 5.1 shows that, if we artificially introduce a loop from the horizon N to horizon 0, then we get a infinite horizon version of the considered finite horizon problem, with the assumption that, from state $s \in S_N$[1] at horizon N, the transition can occur to any state $\bar{s} \in S_0$ at horizon 0. The resulting Q learning algorithm for average cost finite horizon MDP is described in Algorithm 3.

5.3 Problem Formulation

We have adopted the system model from [43]. A block diagram of the system is shown in Figure 5.2. The encoder encodes the incoming video data stream according to MPEG-4 standard.

[1] Since finite horizon the policy is non-stationary it is dependent on the horizon, S_i denote the state space for horizon i.

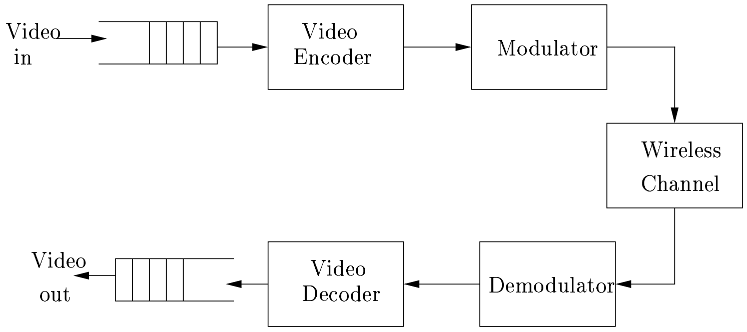


Figure 5.2: System Block Diagram

This encoded video is to be transmitted over wireless channel which is a time-varying and unreliable channel. Here a specific example of a varying wireless channel with frequency non-selective block fading modeled as finite state Markov channel is considered. We assume a slotted system with each slot of length T_c. The channel value remains constant during the slot and it changes at the slot boundary. Our objective is to minimize the expected power per frame under the constraints of distortion and delay. Important assumptions in the implementation of the algorithm are:

1 We assume that the transmitted data is received without any error.

2 There is no sudden change of scene in the video. Thus the rate distortion curve for each frame do not change significantly.

3 A video sequence is large enough so that algorithm can converge. A typical video sequence would of 10 seconds or 300 frames. For real time applications like conferencing, the length of sequence is even larger.

5.3.1 Distortion Constraints

We assume the video is encoded as a sequence of IPPPP$\cdots$ i.e. every I frame occurring at the start of GOP (Group of Pictures). The Rate-Distortion (R-D) model for the P frame in the GOP is represented mathematically as,

$$\frac{R_p}{M_p} = a_1 Q_p^{-1} + a_2 Q_p^{-2} \tag{5.3}$$

$$R_p(D_p) = \ln\left(\frac{1}{\alpha D_p}\right) \tag{5.4}$$

where,

- Q_p is the quantization level used for the current frame p.

- M_p is the Mean Absolute Difference (MAD), computed using motion-compensate residual for the luminance component.

- a_1, a_2 are first- and the second-order coefficients.

- R_p is bit rate for frame p.

- D_p is distortion for frame p.

- α is constant of proportionality.

Let q_k be the quantization parameter of kth MB and u_k be its corresponding transmission rate. q_k takes values from a finite set $\mathcal{Q}$, u_k from a finite set of rates $\mathcal{R}$. and channel state x_k from X. Let the total number of bits in each MB be B_k. Let each frame consists of M number of MBs.

5.3.2 Delay Constraints

Each frame enters the queue at the transmitter with a constant rate. Let T_{MB} be the inter arrival time between MBs at the queue of the transmitter. Let the delay experienced by the kth the MB is given by δ_k, which is expressed as,

$$\delta_k = \omega_k + \frac{B_k}{u_k} \tag{5.5}$$

where,

$$\omega_k = (\delta_{k-1} - T_{MB})^+ \tag{5.6}$$

is the waiting time for each MB. ω_k is the additional time the MB must wait for the preceding MB to finish its transmission. For real time application the MB entering at time slot n, must be decoded at the receiver by the time $nT_c + T$, where T is the end to end delay experienced by each MB. If we remove the encoding and decoding delays from T, which is $T_{max} = T - (M+1)T_{MB}$ [43].

5.3.3 Energy Considerations

Let $P(x_n, u_k)$ be the power required for the transmission at rate u_k and fading coefficient x_n, given by,

$$P(x_n, u_k) = \frac{N_0 W}{x_n}\left(e^{u_k/W} - 1\right), \tag{5.7}$$

where N_0 is the power spectral density AWGN channel and W is the spectrum bandwidth of the wireless link. If B_k are the number of bits of MB k, then the number of slots required for the transmission is given by, $L_k = \lceil \frac{B_k}{u_k T_c} \rceil$. Figure 5.3 explains the concept of slot and number of slots required for kth MB.

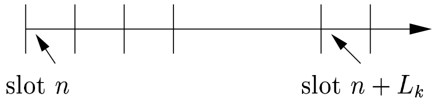


Figure 5.3: kth MB transmission at nth slot

Energy required for transmission of kth MB, starting the transmission at slot n is obtained as,

$$E_k = \mathbf{E}\left\{ \sum_{l=n}^{L_k+n-1} P(x_l, u_k)T_c \,|x_n \right\} \tag{5.8}$$

Hence the total energy required for the transmission of whole frame is given by $\sum_k^M E_k$.

5.3.4 Optimization Model

We now formulate the energy minimization problem as,

$$\begin{aligned} &\min_{u_k,q_k} \mathbf{E}\sum_k^M E_k \\ \text{such that } &\frac{1}{M}\mathbf{E}\sum_k D_k \leq \frac{1}{M}D_{max} \qquad (5.9) \\ &\delta_k \leq T_{max}, \forall k, \qquad (5.10) \end{aligned}$$

where D_{max} is the average distortion constraint per frame and $\mathbf{E}\left\{\sum_k D_k\right\}$ is the expected distortion per frame. For the constrained optimization problem (5.10), we consider relaxed unconstrained optimization problem as,

$$\begin{aligned} &\min_{u_k,q_k} \mathbf{E}\left\{ \sum_k \left[E_k + \lambda D_k\right] \right\}, \\ &\text{such that } \delta_k \leq T_{max}, \forall k \end{aligned} \tag{5.11}$$

where and λ is Lagrange multiplier.

(5.11) optimization problem is M horizon Markov Decision Problem. The state for kth MB is given by $s_k = (\omega_k, x_k)$. and action $a_k = (q_k, u_k)$. For introducing the absolute delay, we restrict the set of feasible policies to, $\mathcal{U}(s_k) = \left\{ q_k \in \mathcal{Q}, u_k \in \left\{ \frac{B_k}{u_k} \leq T_{max} \right\} \right\}$ with per stage cost incurred is given by,

$$c(s_k, u_k) = E_k + \lambda D_k. \tag{5.12}$$

Bellman's Equation with cost to go function V is given by,

$$V_k(s_k) = \min_{\mathcal{U}(s_k)} \mathbf{E}\left\{c(s_k, u_k) + V_{k+1}(s_{k+1})\right\}. \tag{5.13}$$

Finite horizon Bellman Equation (5.13) requires explicit knowledge of channel transition matrix for determining optimal solution. Hence we take resource to learning methods for solving (5.13). We first convert the finite horizon problem into approximate infinite horizon problem. discussed in Section 5.2. We then apply finite horizon Q learning Algorithm 3. In calculation of the energy E_k, the algorithm must also learn about the expected energy for transmission of L_k duration of slots given channel condition x_k. Thus we are required to find, $E\sum_0^{L_k}\frac{1}{x_k} \quad \forall L_k, x_0 \in X$ This is done by averaging samples for every L_k and $x \in X$. The update for the Lagrange multiplier is performed at a slower time scale as [36],

$$\lambda_{n+1} = \lambda_n + \gamma_n\left(D_n - \frac{D_{max}}{M}\right),\ \gamma_n = o(\alpha_n), \tag{5.14}$$

5.4 Implementation Details

We now discuss implementation level details for the algorithm. The algorithm works on two levels - frame level and MB level.

5.4.1 Frame Level Calculations

At frame level, the rate-distortion model given in (5.3) is used to calculate the frame level encoding parameters. The algorithm first calculates the bit rates R for given set of quantization parameter. The R-D model gives corresponding distortion values for set of R values.

5.4.2 MB Level Calculations

Here the minimization problem stated in (5.11) is solved. From the frame level algorithm, the distortion and the rate for different quantization levels are obtained for that particular frame. The values of quantization parameters for each MB q_k and rate of transmission for each MB u_k are obtained based on a state s_k.

5.4.3 Updating the Model Parameters

The model parameters are updated by window method using least square approximation. The source coding parameters Q_i and the actual bits taken by the frame R_i, for past n frames called as window are stored. The model parameters are calculated as,

$$a_1 = \frac{\sum_{i=1}^{n} Q_i R_i - a_2 Q_i^{-1}}{n} \tag{5.15}$$

$$a_2 = \frac{n\sum_{i=1}^{n} R_i - \left(\sum_{i=1}^{n} Q_i^{-1}\right)\left(\sum_{i=1}^{n} Q_i R_i\right)}{n\sum_{i=1}^{n} Q_i^{-2} - \left(\sum_{i=1}^{n} Q_i^{-1}\right)^2} \tag{5.16}$$

Q_i is approximated to be the average of all the quantizations used for each frame i, in the window.

5.5 Simulations Results

The video sequence is encoded with the MPEG-4 implementation provided by MoMuSys. The possible quantization parameters are given by $\mathcal{Q} = \{4, 8, 16, 20, 24, 31\}$. The channel bandwidth is given by $W = 500$ KHz. We consider the AWGN channel of variance $N_0W = 0.39$. The Time required for encoding each MB is considered as $T_{MB} = 0.7$ ms. We assume that the fading transitions occur every T_c seconds where $T_c = 0.1$ ms

The fading channel model is represented by a two-state Markov chain with state space $X = \{0.9, 0.1\}$. The transition probability matrix is considered to be symmetric as follows:

$$\begin{bmatrix} p & 1-p \\ 1-p & p \end{bmatrix} \tag{5.17}$$

Let $p = 0.7$ and we consider $T_{max} = 100$ ms. The transmission rate is chosen from $\mathcal{R} \in \{100, 200\}$ Kbps. We perform experiments using foreman video sequence of duration 10 sec, with 100 MB per frame. In Figure 5.7 expected power required for transmission per frame vs. the expected distortion per frame is plotted. It shows the convex relationship between average transmission power and average distortion. We now let $D_{max} = 2000$. Figure 5.5 and 5.5 show particular snapshot of the original video sequence and transmitted video sequence respectively. Figure 5.7 shows the expected power required for transmission per frame and Figure 5.6 shows the expected Peak Signal to Noise Ratio (PSNR) per frame. There is a sudden drop in the PSNR at frame number 250 because there is a sudden scene change in the foreman video sequence.

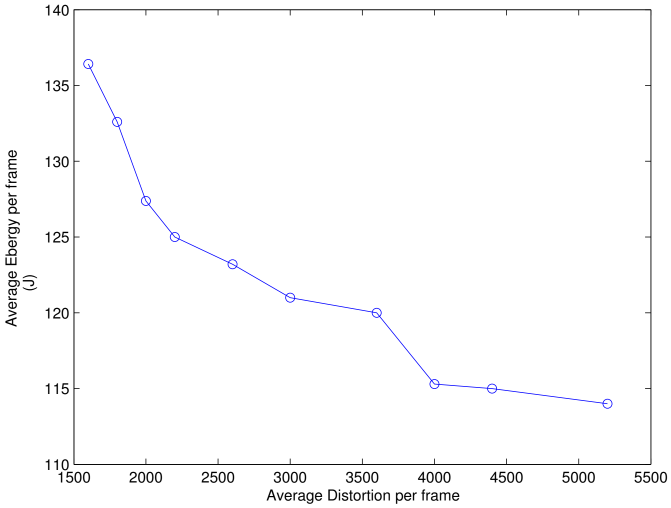
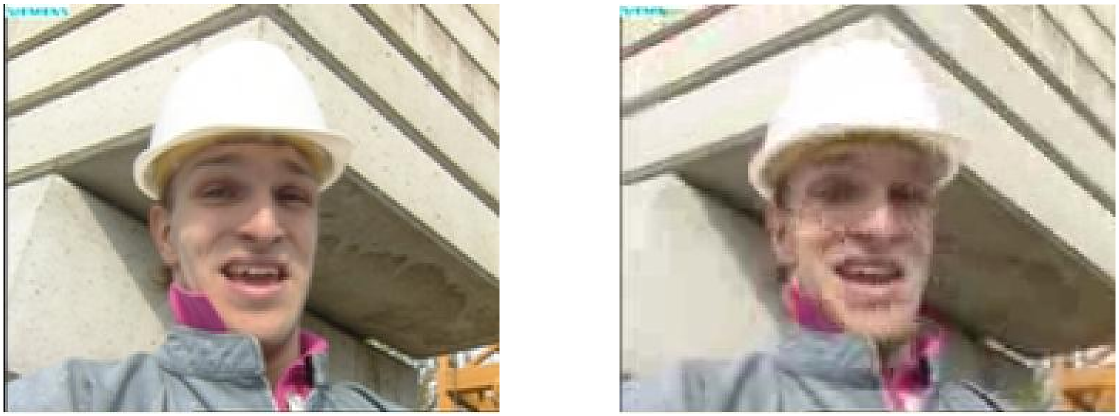


Figure 5.4: Power-Distortion curve



(a) Transmitted Foreman frame

(b) Received Foreman frame

Figure 5.5: Transmission of Foreman Frame

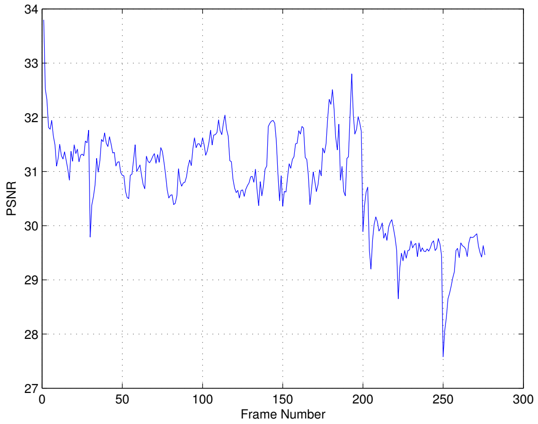


Figure 5.6: Output PSNR Graph

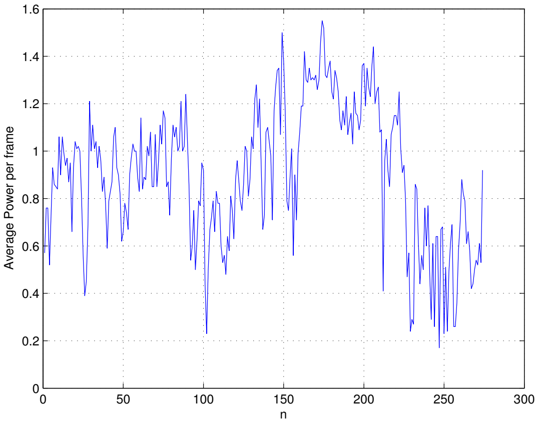


Figure 5.7: Expected Power for $D_{max} = 2000$

5.6 Conclusions and Discussions

The optimization problem discussed in these paper deals with the important issue of distortion experienced in wireless video constrained to energy of transmission and delay in the reception. We have shown the convex relationship between power and distortion experimentally.

Chapter 6

Conclusions

In this thesis, we have explored the issues related to resource allocation in wireless networks. Exploitation of the channel variations using "opportunistic scheduling" has been instrumental in efficiently utilizing the scarce resources. We have developed algorithms for providing QoS like minimum rate, fairness and average delay guarantees, while minimizing the average power for transmission. Towards this, we have exploited the convex relationship between transmission rate (capacity of channel) and power for developing optimal scheduling algorithms under various constraints. We posed the resulting problems in the framework of constrained convex optimization and used stochastic control techniques in designing online solutions to these problems.

We have first considered the minimum rate constrained multiuser scheduling problem and provided stochastic approximation based optimal algorithm. It has been shown that solution for rate constrained multiuser system has the form of memoryless 'water filling'. The performance of the optimal algorithm is compared with the round-robin scheduling scheme. We have also addressed the issue of power optimal temporal short term fair and long term fairness.

We have then considered average delay constrained power optimal algorithm for finite buffer. We have formulated the problem as a average cost Markov Decision Problem. We have presented solution for finite state space version of this problem using 'Post learning'. We have used function approximation technique for solving the continuous channel state and large buffer space version of the problem. This problem has been formulated for point to point link setting. However this method can be easily extended to multiple user scenario. We would like to remark that the function approximation algorithm convergence without using multiple time scale is non-trivial but we have provided the existence of fixed point using function approximation for the average cost MDP.

Finally, we have considered the problem of video transmission over wireless channel. Specifically, we have addressed the issue of minimizing power subject to distortion and delay constraints. We have argued how Q learning algorithm can be used for minimizing power in a finite horizon MDP setting. Here we have demonstrated the convex nature of the power-distortion curve experimentally.

Resource allocation in wireless networks is a complex problem with many facets. In this thesis, we have addressed only few aspects of this problem. There are many interesting problems yet to be solved in this area. Few refinements and modifications to the problems considered are discussed here. While providing minimum rate and fairness guarantees, we have assumed that the scheduler can transmit with arbitrary rates. More practical case of discrete rate scheduler needs to be investigated.

Among various notions, we have used quantifiable temporal fairness as a fairness measure. The idea of fairness in fading environment not yet completely resolved. It is not clear what definition of fairness is more appropriate in this context. Further investigation is required in this direction.

In practice, both average rate and delay constraint may have to be satisfied simultaneously. A joint multiuser delay and rate constrained problem can be investigated. For a broader class of Ad-hoc Networks the network layer issues like routing also play an important role in optimal scheduling policies. Thus joint routing and scheduling algorithms for wireless Ad-hoc networks is an important area of further investigation. Performance of these algorithms with multiple channels particularly Multiple Input Multiple Output(MIMO) systems can be studied. Approaches to these problems may include off-line optimal solutions with the assumption of entire traffic and channel information, on-line model-based solutions and heuristic algorithms. Heuristic algorithms play an important role in real-time scheduling problems because the optimal scheduling problems are exponentially complex or NP-complete and simplicity is a desirable feature.

From the stochastic control theory point of view, function approximation seems to be a potential technique in solving large stage space optimization problems. However, convergence of function approximation with multiple policies is still an unresolved issue. A significant research is required in developing function approximation based provably convergent algorithms.

The online solution presented for video transmission is rather restrictive and imposes constraints on the smoothness of video frames, an impractical situation. The problem of developing an online algorithm for joint source and data rate adaptation suited to all scenarios of video sequences needs further investigation. In this study, we have assumed that a packet is made of one MB. A variable packet length power optimal online scheduling could be a potential area for future work.

Appendix A

Review of Stochastic Approximation and Related Concepts

Mathematical preliminaries and relevant results from the adaptive learning and stochastic approximation literature [37, 45, 18] are presented in this appendix.

A.1 Stochastic Approximation

In this section we will state key results in stochastic approximation theory. Let $H(.,.) : \mathbb{R} \times \mathbb{R}^{\mathbb{N}} \to \mathbb{R}$ be a Lipschitz continuous function with respect to first argument uniformly in second. Let $h(.) : \mathbb{R} \to \mathbb{R}$ is Lipschitz continuous, such that for scalar K,

$$||h(x) - h(y)|| \leq K \, ||x - y|| \tag{A.1}$$

Consider the stochastic approximation of the form,

$$\begin{aligned} x(n+1) &= x(n) + a(n) \left(H(x(n), \gamma(n)) \right), \quad n \geq 0, \\ x(n+1) &= x(n) + a(n) \left(h(x(n)) + M(n+1) \right), \\ \mathbf{E}[H(x, \gamma)] &= h(x), \end{aligned} \tag{A.2}$$

The 'martingale difference sequence' is given by,

$$M(n+1) = \mathbf{H}(x(n), \gamma(n)) - h(x(n)), \tag{A.3}$$

where $x(n) \in \mathbb{R}$. $\gamma(n) \in \mathbb{R}^{\mathbb{N}}$ are i.i.d. or Markovian random variable. We assume that the step sizes satisfy,

$$\sum_{n=0}^{\infty} a(n) = \infty, \quad \sum_{n=0}^{\infty} a(n)^2 < \infty \tag{A.4}$$

Equation A.2 can be viewed as the discretization of ordinary differential equation (ODE),

$$\dot{x}(t) = h(x(t)), \tag{A.5}$$

with the noisy measurement of $h(x)$. As the function $h(x)$ is Lipschitz, ODE A.5 has a unique solution. Let the ODE posses unique asymptotically stable equilibrium point x^*. The fact that step sizes have infinite sum ensures that the algorithm does not converge to a point other than x^*. If $x(n)$ remain bounded then the martingale sum $\sum_{n=0}^{\infty} a(n)M(n+1)$ converges with probability one (w.p. 1) and iterates track the ODE to converge to x^*.

If each component of the iterate $x(n)$ is not updated simultaneously, then the resulting version is called as asynchronous stochastic algorithm, given by,

$$x_i(n+1) = x_i(n) + a(\nu(n,i))I(n,i)[h(x(n)) + M(n+1)], \tag{A.6}$$

where $\nu(n,i) = \sum_{k=0}^{n} I(k,i)$, is the number of times component i gets updated up to time n. The asynchronous version of the above algorithm is shown to track the scaled o.d.e. $\bar{x}(t) = \frac{1}{\epsilon}h(x(t))$, $\epsilon \in \mathbb{R}$. [35] For notational simplicity the scaling term ϵ is not states explicitly. The two time scale version of stochastic algorithm is given by,

$$x(n+1) = x(n) + a(n)\left[h(x(n), y(n)) + M_1(n+1)\right], \tag{A.7}$$

$$y(n+1) = y(n) + b(n)\left[g(x(n), y(n)) + M_2(n+1)\right], \tag{A.8}$$

where,

$$\sum_{n=0}^{\infty} b(n) = \infty, \quad \sum_{n=0}^{\infty} b(n)^2 < \infty, \quad b(n) = o(a(n)), \tag{A.9}$$

Here $x(n)$ update occur at a faster time scale, while $y(n)$ updates occur at a slower time scale. Thus for iterates $x(n)$, $y(n)$ can be viewed as constant and for $y(n)$, $x(n)$ to have attained equilibrium value. Thus iteration A.7 track the o.d.e., $\dot{x}(t) = h(x(t), \hat{y})$, where $\hat{y}$ act as a constant. Let it has globally asymptotically stable equilibrium, $\lambda(\hat{y})$. Thus $x(n)$ track $\lambda(y(n))$ and A.8 tracks $\dot{y}(t) = g(\lambda(y(t), y(t))$. If both the o.d.e. posses globally asymptotically stable equilibrium, then $(x(n), y(n)) \rightarrow (\lambda(y^*), y^*)$ a.s.

Theorem A.1 *(Martingale convergence theorem.) Let $\{\zeta(n) : n = 0, 1, \cdots\}$ be a martingale and there exists a positive scalar L such that $\mathbf{E}\left[\zeta(n)^2\right] \leq L \ \forall n$, then there exists a random variable ζ such that,*

$$\lim_{n\rightarrow\infty} \zeta(n) = \zeta \tag{A.10}$$

Theorem A.2 *(Cauchy-Schwartz inequality.) If x and y are elements of complex space over which inner product $< >$ is defined then,*

$$|\langle x, y\rangle|^2 \leq \langle x, x\rangle . \langle y, y\rangle \tag{A.11}$$

Theorem A.3 *(Kolmogorov-Doob Inequality.) Let $X0, X1, \cdots$ be a martingale sequence. Then, for any a,*

$$\Pr\left[\max_{0\leq i\leq n} X(i) \leq a\right] \leq \frac{\mathbf{E}[|X(n)|]}{a} \tag{A.12}$$

Theorem A.4 *(Gronwall's Lemma.) If, for $t_0 \leq t \leq t_1$, $f(t) \geq 0$ and $g(t) \geq 0$ are continuous functions such that the inequality,*

$$f(t) \leq K + L\int_{t_0}^{t} f(s)g(s)ds, \tag{A.13}$$

holds for some constants $K \geq 0$ and $L \geq 0$, then,

$$f(t) \leq K\exp\left(L\int_{t_0}^{t} g(s)ds\right) \tag{A.14}$$

Theorem A.5 *(Borel Cantelli Lemma.) Let (Σ_n) be a sequence of events in some probability space. If the sum of the probabilities of the Σ_n is finite*

$$\sum_{n=1}^{\infty} \Pr(\Sigma_n) < \infty, \tag{A.15}$$

then ,

$$\Pr\left(\limsup_{n\to\infty} \Sigma_n\right) = 0. \tag{A.16}$$

Theorem A.6 *Sherman-Morrison formula Let A be invertible matrix A, u column vector and v be row vector.*

$$(A+uv)^{-1} = A^{-1} - \frac{A^{-1}uvA^{-1}}{1+vA^{-1}u}. \tag{A.17}$$

Definition A.1 *(Upper Semi-continuous Map.) Set valued map $\hat{h}$ is upper semi continuous (u.s.c.) if, for any neighborhood $N(\hat{h}(x))$ of $\hat{h}(x)$, there exits a neighborhood $N(x)$ of x such that that $\hat{h}(N(x)) \subset N(\hat{h}(x)$.*

Definition A.2 *(Superdifferential.) The superdifferential $\partial^+ f^*$: of function $f : \mathbb{R} \to \mathbb{R}$ is given by, $\partial^+ f^*$ is compact, convex, nonempty set of all Γ satisfying,*

$$f^*(z) \quad \leq \quad f^*(x) + \langle \Gamma, z-x \rangle, \tag{A.18}$$

where $x, z \in \mathbb{R}$. If The smallest closed convex set containing the set $\mathcal{X}$ is denoted by $\bar{co}(\mathcal{X})$. The differential inclusion of f at z is also given by

$$\partial^+ f^*(z) \quad = \quad \bigcap_{\delta>0} \bar{co}\left(\bigcup_{\hat{z}} \in N_\delta(z)\partial^+ f^*(\hat{z})\right).$$

Appendix B

Reinforcement Learning for Average Cost Markov Decision Processes

Markov Decision Process (MDP) is a general model for a large class of multistage, decision making problems with uncertainty. The underlying probabilistic transitions, occur according to a the Markov chain. Dynamic Programming (DP) is a technique for solving the MDP based optimization problems. Various methods in DP include value iteration and policy iteration. However, these methods require exact structure of the transition law. In recent years, reinforcement learning has emerged as a popular paradigm for simulation based algorithms and providing "near optimal" solution to the Markov Decision Problems. Reinforcement learning algorithms learn good policies, in an *environment*, where the environment incurs a cost, for the action taken by the *agent*. Depending upon the feedback obtained from the environment, the agent changes its action which would lead to a better strategy. The legacy effect of "curse of dimensionality" in large state space Markov Decision Processes, is also present in the reinforcement learning methods. An approximate form called functional approximation is suggested for tackling the large state space problem. In functional approximation method, the value function is approximated as a linear or non-linear combination of basis vectors called feature vectors, thereby providing an implementable method for large state space. In this chapter we first discuss Markov decision problems involving average cost. We then take a look at various reinforcement learning methods, like learning using post decision state, Q learning, temporal difference TD(Λ) learning. We then focus on the Markov decision processes with average cost involving general state space and finite action space. We use function approximation method involving a variation of the standard temporal difference as suggested in [46]. We further extend this algorithm for the average cost optimal problems involving multiple policies. In this chapter, both finite state space and general state space are referred.

B.1 Control using Markov Decision Processes

Markov Decision Processes form a basic framework for dynamically controlling systems, which evolve in a stochastic way. In Markov decision processes or controlled Markov chains, current decisions are influenced by previous decisions. MDP involve sequential optimization problems, observed over a infinite or finite duration of horizon N. They involve different performance criterion, like infinite horizon, discounted cost, infinite horizon average cost. Detailed explanation about MDP is given in [47], [34, 48]. In this exposition, we would concentrate on average cost infinite horizon Markov Decision Processes.

Section B.2 deals with finite state and action space. Section B.7 considers generalized state and finite action space. The assumptions in the generalized state space are rather involved and only relevant assumptions are presented.

Consider a discrete time stochastic process, specified by the tuple, $\{S, U, P, u, c\}$, where,

- S indicates for finite state space $S = \{1, 2 \cdots l\}$. For generalized case (infinite dimensional continuous space) we assume, S is compact Borel space of "states" and is $\mathbb{R}$ valued. The Borel space S is Polish (complete and separable). Let $\mathcal{B}(S)$ be the σ algebra of the space S.

- U indicates the finite space of actions. $U = \{u_1, u_2, \cdots u_a\}$. and a probability space $\{U, \mathcal{B}(U), \mathcal{P}_u\}$

- P indicates the conditional law $P(y|s,u)$ $y \in \mathcal{B}(S)$, the probability of moving from state s to state y, under the policy u. For general state space, $P(.|s,u) : S \to \mathcal{B}(S)$ is a measurable function called transition kernel (transition matrix for finite state space) and $\odot(p(.|s,u)) = 1$, where $\odot(p(.|s,u)) = \sum_{y \in S} p(.|s,u)$ for finite state space and $\odot(p(.|x,u)) = \int_{y \in S} p(dy|s,u)$ for general state space. For a measurable function f on S, the transition kernel act as a operator, (similarly the transition matrix acts on the vector)

$$Pf(s) = \int f(y) P(dy|s,u). \tag{B.1}$$

  The nth step transition kernel is given by,

$$P^n(z|s,u) = \int P(z|y,u) P^{n-1}(dy|s,a). \tag{B.2}$$

  The transition kernel induces (an invariant measure) π under Assumption B.1,

$$P\pi(s) = \pi(s). \tag{B.3}$$

- $c : \mathcal{U} \to \mathbb{R}$, is the immediate cost incurred, under policy u and at state s, such that $(s,u) \in \mathcal{U}$, $s \in S$, $u \in U$. For general state space, we assume c to be continuous bounded function of state s.

The history space, $\mathcal{H}_n$ at time n, is defined as,

$$\mathcal{H}_n = \mathcal{H}_{n-1} \times S \times \mathcal{U} \tag{B.4}$$

Thus a sample history at time n, is given by, $h_n = \{s_0, u_0, s_1, u_1, \cdots, s_n, u_n\}$, where the state evolves as a sequence $\{s_n\}, n = 1, 2, \cdots$, under the action sequence $\{u_n\}, n = 1, 2 \cdots$.

A policy μ is a sequence of actions $\{u_n\}$. The policies $\boldsymbol{\mu}$ under which the process s_n is Markov are called Markov policies. A generalized class of Markov policies are randomized policies $\mu_n(.|h_n) : U \to \mathcal{P}_u$. $\mu_n(.|h_n)$ represents condition law where the action $u_n \in U$ is selected, with the distribution μ_n. Thus policy $\boldsymbol{\mu}$ represents a sequence of conditional laws, μ_n. Markov policies such that $\mu_n = \mu$, are called as stationary policies. Stationary policies depend on the current state and are given by $\mu = [\mu(u|i)]_{u \in U, i \in S}$. If the policy $\mu(.|i) i \in S$ has a Dirac function, then the policy is called as non randomized or deterministic. Let U_S denote the set of all stationary policies. A policy is admissible if it satisfies the constraints imposed by the structure of the problem. This may include constraints such as a policy or action that cannot be used for a particular state. (Note that these constraints may also include the constraints introduced by the optimization problem. We discuss such Constrained Markov Decision Process (CMDP) in Section B.4).

B.2 Markov Decision Process

The expected average cost over infinite horizon incurred by a policy μ is given by,

$$V_\mu = \limsup_{N \to \infty} \mathbf{E} \left[\frac{1}{N} \sum_{n=0}^{N-1} c(s_n, u_n) \right] \tag{B.5}$$

Corresponding path wise average cost is given by,

$$V_\mu = \limsup_{N \to \infty} \frac{1}{N} \sum_{n=0}^{N-1} c(s_n, u_n) \tag{B.6}$$

Assumption B.1 *Every stationary policy results in an irreducible Markov chain*

Assumption B.2 *The transition costs $c(s_n, u_n)$ are bounded.*

Remark B.1 *In the general state space, these assumptions are required to be stated in more technical terms, as the irreducibility in general state space is not the same as in finite state space.*

Definition B.1 *(Irreducibility measure.) A probability measure ψ is called an irreducible measure* *for a Markov chain, if for any given point $s \in S$, and given any set $A \in \mathcal{B}(U)$.*

$$\psi(A) > 0 \Rightarrow P^n(s, A), \text{ for some } n \text{ .}$$

Definition B.2 *(Irreducible Markov Chain.) A Markov chain s_n is called ψ-irreducible if there is an irreducibility measure on $\mathcal{B}(S)$ such that,*

$$\psi(A) = 0 \Rightarrow \psi(L_A) = 0$$

with,

$$L_A = \{s \in S | P^n(s, A) > 0 \text{ for some } k\}.$$

If a Markov chain is ψ-irreducible, then there is a measure on $\mathcal{B}(S)$ such that, starting at any point $s \in S$, the chain can reach any "ψ-large" set $A \in \mathcal{B}(S)$ with non-zero probability.

Definition B.3 *(Geometric Ergodicity of Markov chains) For the set S, there is "geometric drift" towards A, that is, for some function L and some $\beta > 0$*

$$P(s, dy)L(y) \leq (1 - \beta)L(s) + \mathbb{1}_A(s), \tag{B.7}$$

where $\mathbb{1}_A(s)$ is a indicator function and is 1 if $s \in A$, otherwise 0. This implies that there exists limiting probability measure π, a constant $R < \infty$ and some uniform rate $\zeta < 1$ such that,

$$\sup_{|f| \leq L} \left| \int P^n(s, dy) f(y) - \int (\pi) f(y) \right| \leq RL(s)\zeta^n \tag{B.8}$$

The optimal average cost is obtained by minimizing over all possible policies ans is expressed as,

$$J^* = \inf_{\mu \in U_S} J_\mu \tag{B.9}$$

For finite state space, if there exists an optimal admissible policy μ^*, which satisfies (B.9), then there exists a scalar $\phi*$ and vector h such that,

$$\phi^* + h(s) = \min_{u \in U} \{c(s, u) + P(y|s, u)h(s)\} \tag{B.10}$$

The vector h is unique up to additive constant. $h(s) - h(y)$ represents the difference in the total cost from starting from s instead of y. ϕ^* uniquely specifies the optimal average cost. The differential cost h can be specified uniquely by letting $h(s^0) = \phi^*$ for arbitrary $s^0 \in S$, The average cost optimal equation with unique h^* is given by,

$$h^*(s) = \min_{u \in U} \{c(s, u) + P(y|s, u)h^*(y) - h^*(s^0)\} \quad s, y \in S \tag{B.11}$$

Further an optimal stationary policy μ must satisfy,

$$\mu \in \arg\min u \in U\{c(s,.) + P(y|s,.)h(y)\} \tag{B.12}$$

For a given stationary randomized policy μ, the corresponding average cost is given by,

$$\phi_\mu + h_\mu(s) = \sum_{u\in U} \{c(s,u) + P(y|s,u)h_\mu(y)\} \tag{B.13}$$

B.2.1 Value Iteration Algorithm

Value iteration algorithm provides an iterative method for determining the optimal value function.

For the average cost problem, we use relative value iteration (RVI) algorithm. Here we arbitrarily choose a reference state s^0 and perform the following iteration,

$$h_{n+1}(s) = \max_{u\in U} \left\{ c(s,u) + \sum_y P(y|x,u)h_n(y) - h_n(s^0) \right\} \tag{B.14}$$

The proof of the convergence of above algorithm is presented in [34].

B.3 Post Decision State based formulation

Consider the scenario where the state transition be expressed as,

$$s_{n+1} = f(s_n, u_n, w_{n+1}), \tag{B.15}$$

where $f : S \times U \times W \to S$, $w_n \in W$ is a disturbance which induces probabilistic transitions into the system and determines the transition matrix (transition kernel for continuous state space) We have viewed our system evolution, where transitions take place, after the disturbance w_n has occurred. Here the state of the system, is defined after the new disturbance occurs, but before the action (decision) is taken. Hence we call such a state s_n as the pre-decision variable. It is possible to define the state after the decision take place. Such a state $\tilde{s}_n$ is termed as “post decision state variable” and the history of the information, decision and states is given by:

$$\mathcal{H}_n = \{s_0, u_0, s_1, u_1, \tilde{s}_1, s_2, u_2, \cdots, \tilde{s}_{n-1}, s_n\}.$$

We utilize “post decision” based approach in reducing the complexity of algorithms in Chapter 4.

B.4 Constrained Markov Decision Processes

Consider a constrained optimization problem,

$$\text{minimize} \quad \limsup_{N\to\infty} \mathbf{E}\left[\frac{1}{N}\sum_{n=0}^{N} c(s_n, u_n)\right] \tag{B.16}$$

$$\text{subject to} \quad \limsup_{N\to\infty} \frac{1}{N}\sum_{n=0}^{N} \; X_i(s_n, u_n) \leq \bar{X}_i, \;\; i = 1\cdots K \tag{B.17}$$

Following the standard Lagrangian approach for solving the constrained Markov Decision Problem [49] the corresponding unconstrained problem of minimizing the Lagrangian is given by,

$$V^{\lambda,\mu} = \limsup_{N\to\infty} \frac{1}{N}\left[\sum_{n=0}^{N} c(s_n, u_n) + \sum_{i=1}^{K} \lambda_i X_i(s_n, u_n)\right] \tag{B.18}$$

We directly state the following theorem for the constrained MDPs from [49].

Theorem B.1 *If the cost function is convex in s, then, there exists an optimal stationary randomized policy μ^*, which needs at most K randomizations, i.e. policy is deterministic for all except one $s \in S$ and for the later state policy randomizes between K policies. There exist Lagrange multipliers $\boldsymbol{\lambda}$, such that $\lambda_i \geq 0$,$\forall i \leq K$, such that μ^* minimizes the cost $V^{\lambda^*}(\mu)$. Furthermore the following "saddle point condition hold".*

$$V^{\lambda} = \inf_{\mu\in U_S} \sup_{\lambda\geq 0}\{V^{\lambda}(\mu)\} = \sup_{\lambda\geq 0} \inf_{\mu\in U_S}\{V^{\lambda}(\mu)\}. \tag{B.19}$$

Also if $\lambda \geq 0$ is the optimal Lagrange multiplier, then the necessary condition for average cost optimality is given by,

$$h(s) = \min_{u}\left\{c(s,u) + \sum_{i}^{K} \lambda_i X(s,u) - \phi + \sum_{\bar{s}} p(\bar{s}|s,u)h(\bar{s})\right\}, \quad \forall s \in S. \tag{B.20}$$

B.5 Simulation based Learning Algorithms for MDP

Implementing value iteration (cf. Section B.2.1) requires the agent to know the expected rewards and the transition matrices. Simulation based learning approach has been developed to perform prediction and optimization, where the agent learns expected rewards and the transition matrices implicitly by trying various possible actions to arrive at the optimal strategies asymptotically. Learning based methods can be further sub classified into algorithms which use actor-critic framework [50] like Temporal difference learning and others like Q learning.

B.5.1 Temporal Difference Learning

Temporal Difference (TD) learning is forms actor part of "Actor-Critic" framework. Actor evaluates the the policy specified by the critic. Critic evaluates the value function for a given policy based on the reward and transitions occurring for the policy. Temporal difference method, improves the value estimation based on the observed and estimated value functions.

For average cost MDP, define Nth step temporal difference at iteration n by,

$$\delta_n^N = \sum_{i=n}^{n+N} c(s_i, u_i) + h_{n+N}(s_{n+1}) - h_n(s^0) \tag{B.21}$$

TD(Λ) method gives weightages to all the temporal differences δ_n^N,depending on the parameter $\Lambda \in [0, 1]$ and finds an effective δ_n as,

$$\delta_n = (1 - \Lambda) \sum_{i=0}^{\infty} \Lambda^i \delta_n^i \tag{B.22}$$

It can be shown that, the effective δ_n is expressed as,

$$\delta_n = \delta_1 z_n, \tag{B.23}$$

where, z_n is called as eligibility trace. Eligibility traces are calculated iteratively as,

$$z_n(s) = \begin{cases} \alpha\Lambda z_{n-1}(s) & \text{if } s \neq s_n \\ \alpha\Lambda z_{n-1}(s) + 1 & \text{if } s = s_n \end{cases}$$

The temporal learning rule becomes,

$$J(s_n) \rightarrow J(s_n) + \gamma_n \delta_n \delta_n^1 \tag{B.24}$$

Depending on the changed value function, a policy improvement is made by the actor.We would discuss about policy improvement methods in Section B.8.

Two timescale version for value iteration

Actor critic based learning algorithms require separating the rate (time scale) at which the policy and the value function updating to be performed on different time scale for ease for convergence proofs. Value function evaluation (critic) is performed on a faster time scale while the policy improvement (actor) is performed on the slower time scale. While evaluating a policy μ the value function observes the policy as a constant. The intuition is that the the value function for a current policy is evaluated, which is improved by changing the policy at a slower rate.

B.5.2 Q learning

For policy μ $Q : S \times U \rightarrow \mathbb{R}$ function is defined as,

$$Q_\mu(s,u) = \left[c(s,u) + \mathbf{E}_{s' \in S} h_\mu(s'|s) - Q(s^0,u)\right], \tag{B.25}$$

The policy improvement algorithm using Q function is given by,

$$u' = \arg\max_u Q(s,u) \tag{B.26}$$

In the learning environment Q function can be evaluated as,

$$Q(s_n,u_n) \rightarrow Q(s_n,u_n) + \gamma_n \left[c_n + \alpha \max_{u'} Q(s_{n+1},b) - Q(s_n,u_n) - Q(s^0,u_n)\right] \tag{B.27}$$

B.6 Function Approximation

For finite state and action space, we can represent value function using table based entry for each state, as done in previous section. Clearly this method is not scalable. For large state space, it is likely that the current visited state is never visited before. Function approximation uses learning done for other states for approximating the value function for the current unvisited state. The problem of large state space and action space is addressed by function approximation methods, where the value function is represented by a set of basis vectors called as feature vectors.

The issues involve in the function approximation because of the stochastic nature of the function approximation and is not just regression analysis. Although function approximation is used various forms, convergence proof only linear function approximation are known [51], where value function $h(x) : S \rightarrow \mathbb{R}$ is approximated by linear combination of feature vectors $f_i(x)$ $i \in 1 \cdots K$ and $r_i \in \mathbb{R}$ as,

$$h(x) = \sum_i^K f_i(x) r_i \tag{B.28}$$

Function approximation basically involves projection of the function $h(x)$ onto the space spanned by feature vectors. Important issues like need for online-sampling for convergence and necessity of the weighted projection rather that Euclidean projection are discussed in [51].

In learning algorithms involving function approximation, multiple policies introduces further complexities. Greedy policy improvement removes the linearity in the value function. Further the projection operator Π, which maps $h(x)$ onto the space of feature vectors is not non-expansive under max-norm, a property used for convergence for table based learning, which complicates the issue of convergence. The only known method of convergence is to use multiple time scale actor critic framework [52]. Analysis of general state space and action space for average cost problems is presented in [52].

In this dissertation we are mainly concerned with function approximation for continuous state space and finite action space. The convergence proof for the continuous state space is not possible, without using policy function approximation. The policy space would be compact set of randomized policies and thus the analysis of the [52] is applicable. However the algorithm has slow convergence properties.

For infinite state space and finite action space we propose a modification of the algorithm discussed in Appendix B.8.1. We present a finite state space and action state space functional approximation based reinforcement learning algorithm in Appendix B.8.1. The algorithm in B.8.1 is based on the two time scale approach discussed in Section B.5.1 suggested in [21]. For finite state space the convergence can be proved, however with the modification suggested for the infinite state, we can prove certain convergence properties and not the exact convergence. Although exact convergence is not proved. For the sake of completeness we prove the convergence properties for the finite state space and finite action space algorithm in Appendix B.8.1.

B.7 Function Approximation with single policy for Average Cost Problem

We consider the following Poisson equation for the average cost problem.

$$h(s) = \min_{u} \mathbf{E} \left[c(s,u) - \phi^* + \int P(dy|s,u)h(y) \right], \tag{B.29}$$

where ϕ^* is the average cost. $h(s)$ is non-unique up to an additive constant. It is well known that the following definition for some state $s_0 \in S$ leads to unique average difference cost $h(s)$

$$h(s^0) = \phi^*,$$

with the property,

$$\int \pi(ds)h(s) = 0$$

The Poisson equation B.29 can be written in terms of dynamic programming operator T as,

$$Th = \min_{u}[c(s,u) - \phi^* + \int p(s,u,dy)h(y)], \tag{B.30}$$

Stationary randomized policies, forms a convex polytope in which deterministic policies are present at the corners. Deterministic policies improvement algorithms are not continuous functions of value function and result in difficulty in proving the convergence of the learning algorithm. On the other hand randomized policies will be continuous functions of value functions. We describe an simulation based actor critic learning algorithm for for continuous state space and comment on its convergence properties using continuity of randomized policies. We suggest an algorithm soft-max policy based algorithm, where actor and critic work at the same time scales, as apposed to the algorithm in Section B.8.1, which work on different time scales. We prove the existence of fixed point for this algorithm.

Let μ_a be a stationary randomized policy suggested by the policy improvement actor algorithm. Let $\pi_a(x)$ be the invariant distribution of state space using μ_a policy. The subscript a suggest that the policy is improved by the actor.

B.7.1 Value Function Evaluation (Critic)

Let μ_a be the policy (randomized or pure) which is to be improved using actor algorithm. We state and derive properties for the critic using a fixed policy μ_a. We consider approximated function difference value function $\bar{h} : S \to \mathbb{R}$,

$$\bar{h}(s) = \sum_i^K f_i(s) r_{ia},$$

where $f_i : S \to \mathcal{R}$ is a continuous function over the space S. Let T_{μ_a} denote the operator for the fixed policy μ_a.

$$T_{\mu_a}\bar{h}(s) = \left[\int p(s, \mu_a, dy) c(s, \mu_a, y) - \phi^* + \bar{h}(y)\right]^1$$

$r_a = (r_{1a}, \cdots, r_{Ka})$ are the scalars which also depend the actor. Subscript a shows this explicit dependency. For notational simplicity we do not show the explicit dependency of a on r. Let $f = [f_1, f_2, \cdots f_K]$. The value function $\bar{h}(x)$ can be considered as a projection Π of $h(x)$ on the space spanned by $f(x)$. Let $|| \, ||_\pi$ denote the weighted norm with respect to the invariant distribution π. The weighted norm $||g(s)||_\pi$ is defined by $\int g(s)\pi(ds)g(s)$. The projection Π of $h(x)$ is given by,

$$\Pi h_{\mu_a} = \arg\min_{f'r} \left|\left|h - f(x)'r\right|\right|_{\pi_{\mu_a}} \tag{B.31}$$

where weighted norm is with respect to the invariant distribution π_{μ_a}.

Assumption B.3 *1.* $\mathbf{E}\left[f_i^2(x)\right] < \infty$

2. f_i *are linearly independent*

We use improved $\Lambda - LSPE$ (Least square policy evaluation method), for value function calculation for a given policy. It is an improved version of the classical temporal difference method and as argued in [46], it has fastest convergence rate in the family of Temporal Difference (TD) learning algorithms. In the $\Lambda - LSPE$ method the weight r_n is expressed by,

$$\begin{aligned}
\bar{r}_n &= \arg\min_r \sum_{m=0}^{n} (f(s_m)'r - f(s_m)'r_n - \sum_{k=m}^{n} (\alpha\Lambda)^{k-m} d_n(s_m, s_{m+1}))^2. \\
r_{n+1} &= r_n + \beta_n(\bar{r}_n - r_n) && \text{(B.32)} \\
d_n(s_m, s_{m+1}) &= c(s_m, s_{m+1}) - \phi_n + (f(s_{m+1}) - f(s_m)r_n, \forall k, n && \text{(B.33)} \\
\phi_{n+1} &= \phi_n + \gamma_n \left(c(s_n, s_{n+1}) - \phi_n\right) && \text{(B.34)}
\end{aligned}$$

The approximate value iteration using temporal difference can be written in as,

$$\bar{J}_{k+1} = \Pi_{u_a} T_{\mu_a} J_k$$

We consider following minimization for Λ-LSPE.

$$\bar{r}_n = \arg\min_r \sum_{m=0}^{n} (f(s_m)'r - f(s_m)'r_n - \sum_{k=m}^{n} (\alpha\Lambda)^{k-m} d_n(s_k, s_{k+1}))^2. \qquad \text{(B.35)}$$

By setting the gradient of the above minimization function to zero, we obtain the following iterative gradient scheme.

$$\begin{aligned}
r_{n+1} &= r_n + \beta_n \left(\sum_{m=0}^{n} f(s_m) f(s_m)'\right)^{-1} \sum_{k=0}^{n} \left(\sum_{m=0}^{k} (\alpha\Lambda)^{k-m} f(s_m)\right) d_n(s_m, s_{m+1}) \\
&= r_n + \left(\sum_{m=0}^{n} f(s_m) f(s_m)'\right)^{-1} (A_n r_n + b_n) \\
&= r_n + \left(\sum_{m=0}^{n} \frac{f(s_m) f(s_m)'}{n+1}\right)^{-1} (\frac{A_n}{n+1} r_n + \frac{b_n}{n+1}) \quad \forall t, \\
&= r_n + \bar{B}_n(\frac{A_n}{n+1} r_n + \frac{b_n}{n+1}) \quad \forall n, && \text{(B.36)} \\
\phi_{n+1} &= \phi_n + \gamma_n(c(s_n, s_{n+1}) - \phi_n) && \text{(B.37)}
\end{aligned}$$

where,

$$
\begin{aligned}
A_n &= \sum_{k=0}^{n} z_k(f(s_{m+1})' - f(s_m)')), \\
b_n &= \sum_{k=0}^{n} z_k c(s_k, s_{k+1}), \\
z_k &= \sum_{m=0}^{k} (\alpha\Lambda)^{k-m} f(s_m) \\
\bar{B}_n &= \left(\sum_{m=0}^{n} \frac{f(s_m)f(s_m)'}{n+1}\right)^{-1}
\end{aligned}
\tag{B.38}
$$

Using Sherman-Morrison formula $\bar{B}_n$ is computed recursively as (Ref. A.6),

$$
\bar{B}_n = \bar{B}_n - \frac{\bar{B}_n f(s_n) f(s_n)' \bar{B}_n}{1 + f(s_n)\bar{B}_n f(s_n)}
\tag{B.39}
$$

Assumption B.4 *The step sizes β_n and γ_n satisfy,*

1

$$
\sum_{n=0}^{\infty} \gamma_n = \infty, \quad \sum_{n=0}^{\infty} \gamma_n^2 < \infty
\tag{B.40}
$$

2 There exists a positive scalar c such that the sequence β_n satisfies $\gamma_n = c\beta_n, \quad \forall n$.

Note that in the iterative policy we do not show decreasing value of the step size β_n. We now state basic lemmas, in order to present the iterate in a convenient form for analysis. Let $\nu_n(s)$ be number of visits to state s up to time n.

Lemma B.1 *Let $N^\epsilon(s) \in S$ be the ϵ neighborhood of s,*

$$
\lim_{\epsilon\to 0} \lim_{n\to\infty} \frac{\nu_n(N^\epsilon(s))}{(n+1)} = \lim_{\epsilon\to 0} \int_\epsilon \pi(ds) = \pi(s) \ \textit{w. p. 1.}
\tag{B.41}
$$

Proof: Geometric Ergodicity of Markov chains implies lemma B.1.

Lemma B.2

$$\begin{aligned}
B &= \mathbf{E}\left[\lim_{n\to\infty}\left(\sum_{m=0}^{n}\frac{f(s_m)f(s_m)'}{n+1}\right)\right] = (\int_S f(s)\pi(ds)f(s)') \\
\mathbf{E}[f(s_n)f(s_{n+m})'] &= \int_S \pi(ds)f(s)\int_S P^m(dy|s)f(y)' \\
A &= \mathbf{E}\left[\lim_{n\to\infty}\frac{A_n}{n+1}\right] \\
&= \int_S \pi(ds)f(s)\left\{\sum_{m=0}^{\infty}\int_S \Lambda)^m P^{m+1}(dy|s)f(y)'\right. \qquad (B.42) \\
&\quad \left. -\sum_{m=0}^{\infty}\int_S \Lambda^m P^m(dy|s)f(y)'\right\} \qquad (B.43) \\
\left|A-\frac{A_n}{n+1}\right| &< C\zeta^n \ \mathit{for}\ C>0\in\mathbb{R} \qquad (B.44) \\
b &= \mathbf{E}\left[\lim_{n\to\infty}\frac{b_n}{n+1}\right] \\
\left|b-\frac{b_n}{n+1}\right| &< C\zeta^n \ \mathit{for}\ C>0\in\mathbb{R} \qquad (B.45) \\
&= \int_S \pi(ds)f(s)\sum_{m=0}^{\infty}\int_S \Lambda)^m P^m(dy|s)\bar{c}(y), \qquad (B.46)
\end{aligned}$$

where $\bar{c} = \int_S P(dy|s)c(s,y)$.

Proof: Notational changes in [53]. Convergence with geometric rate is proved using geometric ergodicity assumption of Markov chain (cf. Definition B.3).

We impose following assumption involving inequality on the weighted norm $||\ ||_\pi$:

Assumption B.5 *For* $J(x): S \to R$.

$$||PJ||_\pi < ||J||_\pi \qquad (B.47)$$

This assumption can be imposed by letting r *such that* $r \in \mathbb{R}^K$, $f'r \neq 1$ *constant*

Iterate B.36 can be written in the form, $r_{n+1} = r_n + \beta_n(h_n + M_n)$, where,

$$\begin{aligned}
h_n &= \bar{B}(Ar_n + b - \phi_n) \qquad (B.48) \\
M_n &= (\bar{B}_n A_n - \bar{B}A)r_n + \bar{B}_n b_n - \bar{B}b + \phi^* - \phi_n \qquad (B.49)
\end{aligned}$$

B.7.2 Proof of convergence for $\Lambda \in [0,1)$

Lemma B.3 *Eigen values of the matrix* $I + (\bar{B})A$ *are less than 1*

Proof: The proof reduces to showing eigen values of the matrix

$$(1-\Lambda)\left\{\left(\int \pi(ds)f(s)f(s)'\right)^{-1}\left(\int \pi(dx)f(x)\sum_{m=0}^{\infty}\Lambda^m\int P^{m+1}(dy|s)f(y)'\right)\right\} \tag{B.50}$$

are less than unity. Assume $\pi(ds)$ is a differentiable function, $\pi(ds)=p(s)ds$. Let $a(s)$ be the eigen vector and β be the corresponding eigen value. Thus,

$$\begin{aligned}\left\{\left(\int \pi(ds)f(s)f(s)'\right)^{-1}\left(\int \pi(ds)f(s)\underbrace{(1-\Lambda)\sum_{m=0}^{\infty}\Lambda^m\alpha^{m+1}\left(\int P^{m+1}(dy|s)f(y)'a(y)\right)}_{L(s)}\right)\right\} &= \beta a(s),\\ \left\{\left(\int \pi(ds)f(s)f(s)'\right)^{-1}\left(\int \pi(dz)f(z)L(z)\right)\right\}a(s) &= \beta a(s).\end{aligned}$$

$$\text{Let } W=\sqrt{p(s)}f(s)'. \tag{B.51}$$

Left-multiply both sides by W, to obtain,

$$W\left\{\left(\int \pi(ds)f(s)f(s)'\right)^{-1}\left(\int \pi(ds)f(s)L(s)\right)\right\}=\beta Wa(s). \tag{B.52}$$

Take the Euclidean norm on both sides. RHS of the above equation becomes,

$$\begin{aligned}||\beta Wa(s)|| &= |\beta|\,||Wa(s)|| &\text{(B.53)}\\ &= |\beta|\sqrt{\int \sqrt{p(s)}f(s)'z(s)f(s)'a(s)\sqrt{p(s)}dx} &\text{(B.54)}\\ &= |\beta|\,\big|\big|f(s)'a(s)\big|\big|_\pi &\text{(B.55)}\end{aligned}$$

LHS is given by,

$$\begin{aligned}
&\left\|\sqrt{p(s)}f(s)'\left\{\left(\int p(y)f(y)f(y)'dy\right)^{-1}\left(\int p(z)f(z)L(z)dz\right)\right\}\right\| \\
&= \left\|\sqrt{p(s)}f(s)'\left\{\left(\int p(y)f(y)f(y)'dy\right)^{-1}\left(\int \sqrt{p(z)}f(z)dz\sqrt{p(z)}L(z)\right)\right\}\right\| \\
&= \left\|\int\left\{\underbrace{\sqrt{p(s)}f(s)'\Big(\int p(y)f(y)f(y)'dy\Big)^{-1}\sqrt{p(z)}f(z)}_{K(x,z)\in\ l_2}\right\}\underbrace{\sqrt{p(z)}L(z)dz}_{\in\ l_2}\right\| \\
&\leq \left\|\sqrt{\int\left(\sqrt{p(s)}f(s)'\Big(\int p(y)f(y)f(y)'dy\Big)^{-1}\sqrt{p(z)}f(z)dz\right)^2}\sqrt{\int\left(\sqrt{p(z)}L(z)dz)\right)^2}\right\| \\
&\qquad \text{(by Cauchy-Schwartz inequality)} \\
&= \left\|\sqrt{p(s)}f(s)'\Big(\int p(y)f(y)f(y)'dy\Big)^{-1}\sqrt{p(z)}f(z)\right\|\ \left\|\sqrt{p(z)}L(z)\right\|
\end{aligned}$$

$\left\|\sqrt{p(s)}f(s)'\left(\int p(y)f(y)f(y)'dy\right)^{-1}\sqrt{p(z)}f(z)\right\| = 1$

$$\Rightarrow \text{LHS} \leq \left\|\sqrt{p(s)}L(s)\right\| \tag{B.56}$$

$$\begin{aligned}
\left\|\left(\sqrt{p(s)}L(s)\right)\right\| &= \left\|L(s)\right\|_\pi \\
&= \left\|(1-\Lambda)\sum_{m=0}^{\infty}\Lambda^m\left(\int P^{m+1}(dy|s)f(y)'a\right)\right\|_\pi \\
&\leq (1-\Lambda)\sum_{m=0}^{\infty}\Lambda^m\left\|\left(\int P^{m+1}(dy|x)f(y)'a(y)\right)\right\|_\pi \\
&< (1-\Lambda)\sum_{m=0}^{\infty}\Lambda^m\left\|f(s)'a(s))\right\|_\pi \\
&< (1-\Lambda)\sum_{m=0}^{\infty}\Lambda^m\left\|f(s)'a(s)\right\|_\pi \\
&< \left\|f(s)'a\right\|_\pi
\end{aligned}$$

As $I + \bar{B}A$ is negative definite, $\bar{B}A$ is also negative semidefinite. Consider the deterministic algorithm for the above stochastic approximation,

$$r_{n+1} = r_n + \beta_n\bar{B}(Ar_n + b) \tag{B.57}$$

$$\phi_{n+1} = \phi_n + \gamma_n(\bar{c} - \phi_n), \tag{B.58}$$

where $\bar{c} = \bar{c}(s) = E[c(s_n, s_{n+1})|s_n = s]$. Let $z = E[z_n] = \frac{\int \phi(ds)\pi(s)}{1-\Lambda}$ Let

$$\theta_n = \begin{bmatrix} \phi_n \\ r_n \end{bmatrix}$$

(B.57) and (B.58) can be written in the matrix form as,

$$\theta_{n+1} = \theta_n + \beta_n \left(C\theta_n + d \right), \tag{B.59}$$

where,

$$C = \begin{bmatrix} -c & 0\cdots \\ -\bar{B}z & A \end{bmatrix}$$

$$d = \begin{bmatrix} -c\phi^* \\ -\bar{B}b \end{bmatrix}$$

Let L be a diagonal matrix with first element $l > 0$ and all other diagonal elements 1. Consider a modified iteration with $\bar{\theta}_n$=$L^{\frac{1}{2}}\theta_n$, Hence,

$$\bar{\theta}_{n+1} = \bar{\theta}_n + \beta_n \left(L^{\frac{1}{2}} C L^{-\frac{1}{2}} + L^{\frac{1}{2}} b \right). \tag{B.60}$$

Lemma B.4 *LC is negative definite*

Proof: Let

$$\theta = \begin{bmatrix} \phi \\ r \end{bmatrix}$$

.

$$\theta LC\theta' = -lc\phi^2 - \frac{1}{1-\Lambda} \left|\left| Yf'r \right|\right| + r'Ar, \tag{B.61}$$

where operator Y is given by,

$$Y = \left(\int p(y)f(y)f(y)'dy \right)^{-1} \frac{\phi(s)p(s)}{1-\Lambda} \tag{B.62}$$

$||Y||$ is finite because of compact state space and bounded feature vectors. Hence

$$|Yr| \leq C_1 \left|\left| r \right|\right|, \text{ for some constant } C_1 \tag{B.63}$$

As matrix A is negative definite,

$$r'Ar \leq -C_2 \left|\left| r \right|\right|^2, \text{ for some constant } C_2 \tag{B.64}$$

$$\theta LC\theta' \leq -lc\phi^2 + C_1 \left|\left| r \right|\right| - C_2 \left|\left| r \right|\right|^2, \tag{B.65}$$

for some large l, $\theta LC\theta < 0$. Hence LC is negative definite. As LC is negative definite, $L^{\frac{1}{2}}CL^{-1\frac{1}{2}}$ also negative definite.

Theorem B.2 *For policy* μ

$$r_n \rightarrow r_\mu^*, \quad \phi_n \rightarrow \phi_\mu^* \text{ w. p. 1} \tag{B.66}$$

Proof: From Lemma B.2, C and d converges geometrically. From [45] and Lemma B.4, $\bar{\theta}_n$ converges to the $\bar{\theta}^*$, if θ_n converges to θ^*. and $\theta^* = L^{-\frac{1}{2}}\bar{\theta}^*$. Thus $r_n \rightarrow r_\mu^*, \quad \phi_n \rightarrow \phi^*\mu$ w. p. 1.

B.8 Policy Improvement

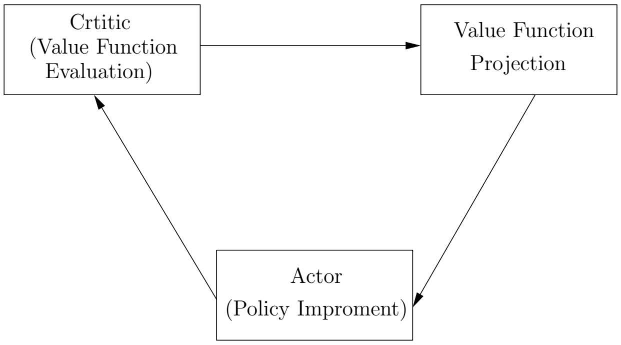


Figure B.1: Actor-Critic Framework

We have considered a constant policy evaluation. The single policy iteration with compact state space has unique fixed point for average cost Markov Decision Problems. But the existence of fixed point can't be guaranteed while using a greedy policy B.67 improvement for the function approximation.

$$u'_a = \arg\max_{u_a}\{c(s_n, u_a) - \phi_n + f(s_{n+1})r_a\} \tag{B.67}$$

We use following non-greedy solution for the policy exploration.

$$
\begin{aligned}
r_{n+1} &= r_n + \beta_n \bar{B}_n \sum_{k=0}^{n} \left(\sum_{m=0}^{k} \Lambda^{k-m} f(s_m) \right) d_n(s_k, s_{k+1}) && \text{(B.68)} \\
\phi_{n+1} &= \phi_n + \gamma_n (c(x_n, x_{n+1}) - \phi_n) \\
\bar{B}_{n+1} &= \bar{B}_n - \frac{\bar{B}_n f(s_{n+1}) f(s_{n+1})' \bar{B}_n}{1 + f(s_{n+1})' \bar{B}_n f(s_{n+1})} \\
Q(s_n, u_n) &= r(s_n, u_n) + V(s_{n+1}) \\
\mu_{\phi r}^{\delta}(s_n, u_a) &= \frac{Q(s_n, u_a)/\delta}{\sum_{u'_a} \exp Q(s_n, u'_a)/\delta} && \text{(B.69)}
\end{aligned}
$$

B.8.1 Convergence Properties

The continuity of the agent's action selection strategy is used in proving the existence of fixed points for the algorithm. We do not show the optimality and uniqueness of the convergence of the proposed algorithm. The randomized policy $\mu^{\delta}(s,u)$ achieves two important requirements for the implementation of the algorithm. It makes the policy a continuous function of the value function along with the exploration as well. Let $\pi_{\mu^{\delta}}$ be the stationary invariant distribution attained by following the randomized policy $\mu^{\delta}_{\phi r}(x,u)$.

Assumption B.6 *r_n is bounded.*

The transition kernel P_{μ} is a continuous function of any randomized policy μ and thus invariant distribution π_{μ} is also continuous function of policy μ. Similarly average cost ϕ_{μ} is a continuous function of μ. Hence the vector $\theta.$ is continuous function of μ. Thus r_u is a continuous function of μ. Since μ forms a convex polytope $\mathcal{C}$, $R = \{r_{\mu} | r_{\mu} \in \mathcal{C}\}$. Let $R_{max} = \max ||r||$.

Lemma B.5 *(B.68) has a fixed point.*

Proof: We consider the scaled iteration B.60. Let $T\theta_{\mu} = \bar{\theta}_{\mu} + \gamma C\bar{\theta}_{\mu} + d$. With respect to some norm $|| \ ||_w$,

$$\begin{aligned} ||T\theta_{\mu}||_w &\leq \left|\left|\bar{\theta}_{\mu}\right|\right|_w + \left|\left|\gamma C\bar{\theta}_{\mu}\right|\right|_w + ||d||_w \\ &\leq (1+\gamma)\,||r||_w + D \end{aligned}$$

Thus T is a continuous function of r_{μ} over a compact space,

$$\mathcal{R} = \left\{ r \middle| \, ||r||_w \leq (1+\gamma)R_{max} + D \right\}$$

Hence by Brouwer's fixed point theorem, the lemma is proved.

A Look at Finite State space and Finite Action Space Function Approximation

We do not deal with finite state space function approximation further. Hence for the completeness, in this section, the sketch of the proof of convergence for the finite state, finite policy function approximation is presented.

From ODE analysis for the stochastic approximation, iterations B.68 track the ODE,

$$\dot{r}(t) = YO\left(T^{\Lambda} f' r(t) - f' r(t)\right), \tag{B.73}$$

where Operator O is,

$$O = \pi_{\mu(x)_{\delta}} \phi(x)$$

Algorithm 4 Convergent form of Actor Critic Algorithm for finite state and action space

$$r_{n+1} = r_n + \beta_n \bar{B}_n \left(A_n \theta_n + b_n \right) \tag{B.70}$$

$$\phi_{n+1} = \phi_n + \gamma_n (c(s,a) - \phi_n)$$

$$Q_{n+1}(s,u) = Q_n(s,u) + \epsilon_n \left(c(s,u) - \phi_n + f(N(s,u))' r_n - f(s)' r_n \right), \tag{B.71}$$

where $N(s,u)$ gives the simulated next state by taking action a from state x.

$$\mu_\delta(s_n, u_a) = \frac{Q(s_n,u_a)/\delta}{\sum_{u'_a} \exp Q(s_n,u'_a)/\delta} \tag{B.72}$$

and acts on vector v, $Ov = \int \pi(x)_{\mu(x)_\delta} \phi(x) v(x) dx$, and T^Λ operator is,

$$T^\Lambda(h_n(s)) = (1-\Lambda) \sum_{m=0}^{\infty} \Lambda^m \mathbf{E} \left[\sum_{n=0}^{m} c(s_n, s_{n+1}) - \phi_n + \alpha^{m+1} h_n(s_{m+1}) \,|s_0 = s \right]. \tag{B.74}$$

For Algorithm 4 the actor critic algorithm in [21] is modified to include the function approximation in the value function evaluation. In Algorithm 4 value function is evaluated at faster time scale, while the policy improvement by the actor is done at a slower time scale. Except for the ODE related to the value function evaluation, convergence proof in [21] can be extended to prove the convergence of Algorithm 4.

Bibliography

[1] W. Lee, *Wireless and Cellular Telecommunications.* 3rd edition: McGraw-Hill, 2005.

[2] X. Liu, E. Chong, and N. B. Shroff, "Opportunistic Transmission Scheduling with Resource-Sharing Constraints in Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 10, pp. 2053–2065, 2001.

[3] S. Kulkarni and C. Rosenberg, "Opportunistic Scheduling Policies for Wireless Systems with Short Term Fairness Constraints," in *IEEE GLOBECOM*, December 2003.

[4] P. Viswanath, D. N. C. Tse, and R. Laroia, "Opportunistic Beamforming using Dumb Antennas," *IEEE Transactions on Information Theory*, vol. 48, pp. 1277–1294, June 2002.

[5] R. Knopp and P. A. Humblet, "Information Capacity and Power Control in Single-cell Multiuser Communications," in *IEEE International Conference on Communications*, (Seattle, USA), June 1995.

[6] E. F. Chaponniere, P. Black, J. M. Holtzman, and D. Tse, "Transmitter directed Multiple Receiver System using Path Diversity to Equitably Maximize Throughput." U.S. Patent No. 6449490, September 10, 2002.

[7] R. Berry and E. M. Yeh, "Cross-Layer Wireless Resource Allocation," *IEEE Signal Processing Magzine*, vol. 21, pp. 59–68, September 2004.

[8] Y. Liu and E. Knightly, "Opportunistic Fair Scheduler over Multiple Wireless Channel," in *IEEE INFOCOM*, vol. 3, pp. 1106–1115, 2003.

[9] X. Liu, E. Chong, and N. Shroff, "Joint Scheduling and Power-Allocation for Interference Management in Wireless Networks," in *IEEE Vehicular Technology Conference Fall 2002*, pp. 757–766, 2002.

[10] J. Rulnic and N. Bambos, "Mobile Power Management for Wireless Communication Networks," *Wireless Networks*, vol. 3, pp. 3–14, 1997.

[11] V. S. Borkar, "On White Noise Representations in Stochastic Realization Theory," *SIAM Journal on Control and Optimization*, vol. 31, pp. 1093–1102, 1993.

[12] D. P. Bertsekas, *Nonlinear Programming.* Belmont, Mass., 2nd edition: Athena Scientific, 1999.

[13] H. Kushner and G. Yin, *Stochastic Approximation Algorithms and Applications.* New York: Springer-Verlag, 1997.

[14] M. Bardi and I. C. Dolcetta, *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations.* Boston: Birkhauser, 1997.

[15] P. Milgrom and I. Segal, "Envelop Theorems for Arbitrary Choice Sets," *Econometrica*, vol. 70, pp. 583–601, 2002.

[16] A. Jean and C. Arrigo, *Differential Inclusions* . Berlin: Springer-Verlag, 1984.

[17] V. Borkar, "On White Noise representations in Stochastic Realization Theory," *SIAM Journal on Control and Optimization*, vol. 31, pp. 1093–1102, 1993.

[18] B. Delyon, "General Results on the Convergence of Stochastic Algorithms," *IEEE Transactions on Automatic Control*, vol. 41, no. 9, pp. 1245–1256, 1996.

[19] V. S. Borkar, "On the Lock-in Probability of Stochastic Approximation," *Combinatorics, Probability and Computing*, vol. 11, no. 1, pp. 11–20, 2002.

[20] A. Ghosh, D. R. Wolter, J. G. Andrews, and R. Chen, "Broadband wireless access with WiMax/802.16: current performance benchmarks and future potential," *IEEE Communications Magazine*, vol. 43, pp. 129 – 136, February 2005.

[21] V. Konda and V. S. Borkar, "Actor-critic type learning algorithms for Markov Decision Processes," *SIAM Journal on Control and Optimization*, vol. 38, pp. 94–123, 2000.

[22] V. Bharghavan, S. Lu, and T. Nandagopal, "Fair scheduling in wireless networks: Issues and approaches," *IEEE Personal Communications*, pp. 44–53, 1999.

[23] J. M. Holtzman, "Asymptotic Properties of Proportional-Fair Sharing Algorithms," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 3, pp. 33–37, 2001.

[24] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting, and R. Vijayakumar, "CDMA data QoS Scheduling on the Forward link with Variable Channel Conditions," tech. rep., Bell Labs Technical Memorandum, April 2000.

[25] M. Andrews, "Instability of the Proportional Fair Scheduling Algorithm for HDR," *IEEE Transactions on Wireless Communications*, vol. 3, pp. 1422–1426, September 2003.

[26] S. Shakkottai and A. Stolyar, "Scheduling for Multiple flows sharing a Time-varying Channel: the Exponential Rule," *AMS Translations Series 2*, vol. 207, 2002.

[27] A. Eryillmaz and R. Srikant, "Scheduling with QoS Constrains over Rayleigh Fading Channel," in *IEEE Conference on Decision and Control*, vol. 4, pp. 3447– 3452, December 2004.

[28] D. Rajan and A. Sabharwal, "Transmission Policies for Bursty traffic sourses on Wireless Channel," in *35 th Annual Conference on Information Sceinces and System*, (Baltimore), March 2001.

[29] D. Rajan and A. Sabharwal, "Delay and Rate Contrained Transmission Policies over Wireless Channels," in *IEEE GLOBECOM*, November 2001.

[30] E. Collins and R. L. Cruz, " Transmission Policies for Time Varying Channels with Average Delay Constraints," in *Allerton Conference on Communications, Contro, and Computing*, (Monticello,IL), 1999.

[31] R. Berry and R. Gallager, "Communication over Fading Channels with Delay Constraints," *IEEE Transation on Information Theory*, vol. 48, pp. 1135–1149, May 2002.

[32] R. Berry, "Power and Delay Trade offs in Fading Channels." PhD Thesis, Massachusetts Institute of Technology, June 2000.

[33] M. Goyal, A. Kumar, and V. Sharma, "Power Constrained and Delay Optimal Policies for Scheduling Transmission over a Fading Channel," in *IEEE INFOCOM*, April 2003.

[34] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Volumes I and II.* Belmont, Mass., 2nd edition: Athena Scientific, 1995.

[35] V. S. Borkar, "Asynchronous Stochastic Approximation," *SIAM Journal on Control and Optimization*, vol. 36, pp. 840–851, 1998.

[36] V. S. Borkar, "Stochastic Approximation with two time scales," *Systems and Control Letters*, vol. 29, pp. 291–294, 1996.

[37] V. S. Borkar and S. Meyn, "The ODE method for Convergence of Stochastic Approximation and Reinforcement Learning," *SIAM Journal on Control Optimization*, vol. 38, no. 2, pp. 447–469, 2000.

[38] J. Abounadi, D. Bertsekas, and V. S. Borkar, "Learning algorithms for Markov Decision Processes," *SIAM Journal on Control and Optimization*, vol. 40, pp. 681–698, 2001.

[39] I. J. S. W. 11, " Information technology Coding of audio-visual objects, Part 1: Systems, Part 2: Visual, Part 3: Audio." FCD 14496, December 1998.

[40] H. Lee, T. Chiang, and Y. Zhang, "Scalable Rate Control for MPEG-4 Video," *IEEE Transaction On Circuits And Systems For Video Technology*, vol. 10, pp. 878–894, September 2000.

[41] Y. Eisenberg, C. E. Luna, T. N. Pappas, R. Berry, and A. K. Katsaggelos, "Joint Source Coding and Transmission Power Managemant for Energy Efficient Wireless Video Communication," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, pp. 411–424, June 2002.

[42] C.-Y. Hsu, A. Ortega, and M. Khansari, "Rate Control for Robust Video Transmission over Burst-Error Wireless Channels," *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 756–773, May 1999.

[43] C. E. Luna, Y. Eisenberg, R. Berry, T. N. Pappas, and A. K. Katsaggelos, "Joint Source Coding and Data Rate Adaptation for Energy Efficient Wireless Video Streaming," *IEEE Journal on Selected Areas in Communication*, vol. 21, pp. 1710–1720, December 2003.

[44] F. Garcia and S. M. Ndiaye, "A learning rate analysis of reinforcement learning algorithms in finite-horizon," in *Proceedings of 15th International Conference on Machine Learning*, pp. 215–223, Morgan Kaufmann, San Francisco, CA, 1998.

[45] A. Benv'eniste, M. M'etivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximations.* Spring-Verlag, 1990.

[46] D. P. Bertsekas, V. S. Borkar, and A. Nedic, *Handbook of Learning and Approximate Dynamic Programming, Improved Temporal Difference methods with Linear Function Approximation.* Piscataway, NJ.: Wiley-IEEE Press, 2004.

[47] A. Arapostathis, V. S. Borkar, E. Fernandez-Gaucherand, M. K. Ghosh, and S. Markus, "Discrete time Controlled Markov processes with average cost criterion - a survey," *SIAM Journal of Control and Optimization*, vol. 31, pp. 282–344, March 1993.

[48] S. Meyn and R. Tweedie, *Markov Chains and Stochastic Stability.* London: Springer-Verlag, 1993.

[49] Eitan Altman, *Constrained Markov Decision Processes.* Chapman and Hall/CRC, 1999.

[50] V. S. Borkar, "An Actor-Critic algorithm for Constrained Markov Decision Processes," *Systems and Control Letters* , vol. 54, pp. 207–213, 2005.

[51] J. N. Tsitsiklis and B. V. Roy, "An analysis of temporal-difference learning with function approximation," Tech. Rep. LIDS-P-2322, MIT, 1996.

[52] V. Konda and J. Tsitsiklis, " Actor-Critic Algorithms," *In Advances in Neural Information Processing Systems NIPS-12. MIT Press.*, 2000.

[53] A. Nedic and D. P. Bertsekas, "Least Squares Policy Evaluation Algorithms with Linear Function Approximation," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 13, pp. 79–110, 2003.