Throughput Optimization in High Speed Downlink Packet Access (HSDPA)

Tao Cui
Department of Electrical Engineering
California Institute of Technology
Pasadena, CA 91125, USA
Email: taocui@caltech.edu

Feng Lu, Anil Goteti, V. Sethuraman, S. P. Rao and P. Subrahmanya
Qualcomm Inc.
3165 Kifer Road
Santa Clara, CA 95051, USA
Email: {fenglu, agoteti, vigneshs, sprao, psubrahm}@qualcomm.com

Abstract— In this paper, we investigate throughput optimization in High Speed Downlink Packet Access (HSDPA). Specifically, we propose offline and online algorithms for adjusting the Channel Quality Indicator (CQI) used by the network to schedule data transmission. In the offline algorithm, a given target BLER is achieved by adjusting CQI based on ACK/NAK history. By sweeping through different target BLERs, we can find the throughput optimal BLER offline. This algorithm could be used not only to optimize throughput but also to enable fair resource allocation among mobile users in HSDPA. In the online algorithm, the CQI offset is adapted using an estimated short term throughput gradient without specifying a target BLER. An adaptive stepsize mechanism is proposed to track temporal variation of the environment. We investigate convergence behavior of both algorithms. Simulation results show that the proposed offline algorithm can achieve the given target BLER with good accuracy. Both algorithms yield up to 30% HSDPA throughput improvement over that with 10% target BLER.

I. INTRODUCTION

The success of 3rd generation wireless cellular networks is mainly based on efficient provisioning of the expected wide variety of services requiring different Quality of Service with respect to data rate, delay and error rate. In order to improve support for high data rate packet switched services, 3GPP has developed an evolution of UMTS based on WCDMA known as High Speed Downlink Packet Access (HSDPA) which was included in the Release 5 specifications [1]. HSDPA targets increased capacity, reduced round trip delay, and higher peak downlink (DL) data rates. Evolutions of HSDPA featuring data rates up to 84 Mbps are under development.

In HSDPA, the user equipment (UE) (also known as mobile station) monitors the quality of the downlink wireless channel and periodically reports this information to the base station (referred to here as NodeB) on the uplink. This feedback, called Channel Quality Indicator (CQI), is an indication of the highest data rate that the UE can reliably receive in the existing conditions on the downlink wireless channel. The frequency of reporting CQI is configured by the network, and is typically set to once every few milliseconds. Using the channel quality reports, the NodeB accordingly schedules data on the High Speed Physical Downlink Shared Channel (HS-PDSCH). The NodeB's selection of the transport block size (number of information bits per packet), number of channelization codes, modulation and resource allocation choices such as HS-PDSCH transmit power allocation are guided by the NodeB's interpretation of the reported CQI.

CQI reports are intended to accurately reflect the HS-PDSCH performance that the UE can support in the existing wireless channel conditions. It is recommended in [2] that, in static channel conditions, the UE report CQI such that it achieves a block error rate (BLER) close to 10% when scheduled data corresponding to the median reported CQI. In practice, the accuracy of CQI reports in reflecting HS-PDSCH performance is influenced by the wireless channel conditions such as the speed of the mobile user and the dispersive nature of the channel. Achieving a certain target BLER at a given scheduled data rate requires different average HS-DSCH SNR under different channel conditions. Also, the NodeB often uses different transport block sizes, number of codes and modulation, collectively referred to as the transport format resource combination (TFRC), to achieve similar data rates. The exact choice of TFRC that the NodeB uses affects the required HS-PDSCH SNR to achieve a certain target BLER. These variabilities may cause the actual BLER to deviate from the 10% target. Moreover, the 10% target BLER may not yield maximum throughput under all conditions of the wireless channel.

The cell throughput optimization in HSDPA can be considered a two part problem: one is code and power allocation across users, and the other is maximizing the link throughput for each user for a given resource allocation. In this paper, we focus on the link throughput optimization and consider throughput optimization through simple adjustments to the reported CQI. We propose offline and online algorithms for adjusting the CQI. In the offline algorithm, we first propose an adaptive algorithm to achieve a given target BLER using the stochastic gradient descent method, which adjusts the CQI offset adaptively based on the short term BLER obtained from the ACK/NACK history. By searching through different target BLERs, we can find the throughput optimal BLER offline. The proposed algorithm can be implemented at the UE as well as at the Node B. When applied at the Node B, in addition to achieving the target BLER, it can also save transmit power. This algorithm could be used not only to refine CQI-BLER alignment but also to enable fair resource allocation among mobile users in HSDPA. Standard stochastic approximation

(SA) algorithms typically require a decreasing stepsize [3]. We show the convergence of the offline algorithm with a constant stepsize.

In the online algorithm, we use a variation of the Kiefer-Wolfowitz algorithm [4] in SA, which does not need to specify a target BLER. The CQI offset is adapted gradually using an estimated short term throughput gradient. Unlike [4], the stepsize in the proposed algorithm does not decrease to zero. In addition, an adaptive stepsize mechanism is proposed to track temporal variation of the environment. With a constant step-size, we show that the proposed online algorithm converges to a small neighborhood of the local optimal solution. Our simulation results show that the proposed offline algorithm can achieve the given target BLER with good accuracy. Both throughput optimization algorithms are shown to improve the throughput by up to 30% in simulation. The throughput optimal BLER is calculated for popular channel path pro-files. In general, the throughput optimal BLER is not always 10% and depends on the channel path profile. For AWGN channels, it is about 10%, as is implied in [5]. Considering that the UE implementation in the simulation closely mirrors commercially shipping devices and already includes several receiver optimizations, the additional gain obtained through the algorithm is indicative of potential HSDPA throughput enhancement realizable in practice.

II. Link Throughput Expression in HSDPA

Throughput in HSDPA is the rate of transfer of information bits over the wireless channel in units of bits per subframe. For a given resource allocation across users, the network may schedule different TFRCs to a user depending on the perceived channel conditions of the user. We model the schedular using a 30 entry table of CQI vs TFRC such as [2, Table 7D]. For a given SNR g in dB, let $p_{cqi}(g)$ denote the probability that a new transmission with TFRC corresponding to CQI index cqi is scheduled, $b_{cqi}(g)$ be the first BLER, the BLER corresponding to the first transmission, when the TBS corresponding to cqi is scheduled and TBS_{cqi} denote the TBS corresponding to cqi. We assume that the second BLER (retransmission BLER) is zero. In the following, BLER denotes first BLER. Assuming that N subframes are to be received by UE, the average number of new transmissions with TBS_{cqi} is $p_{cqi}(g)N$. The average number of subframes required to send $p_{cqi}(g)N$ subframes is $p_{cqi}(g)(1+b_{cqi}(g))N$, assuming the second BLER is zero. Therefore, the average throughput at SNR g is

$$T(g)=\frac{\sum_{cqi=1}^{30}p_{cqi}(g)\mathrm{TBS}_{cqi}}{\sum_{cqi=1}^{30}p_{cqi}(g)(1+b_{cqi}(g))}. \quad (1)$$

We model the estimated CQI as a random variable with mean $m(g)=g+\kappa$, where κ is a fixed offset, and probability density function (pdf) $f(cqi,m(g))$. In practice, we find that the Gaussian distribution is a reasonable approximation of the real CQI distribution, i.e.,

$$f(cqi,m(g))=\frac{1}{\sqrt{2\pi\sigma^2(g)}}e^{-\frac{(cqi-m(g))^2}{\sigma^2(g)}}. \quad (2)$$

Given g, we can compute $p_{cqi}(g)$ as

$$p_{cqi}(g)=\begin{cases}\int_{cqi-0.5}^{+\infty}f(x,m(g))dx, & \text{if } cqi=30,\\\int_{-\infty}^{cqi+0.5}f(x,m(g))dx, & \text{if } cqi=1,\\\int_{cqi-0.5}^{cqi+0.5}f(x,m(g))dx, & \text{otherwise.}\end{cases} \quad (3)$$

When we apply a CQI offset Δ on κ or shift the distribution by Δ, (3) is modified to

$$p_{cqi}(\Delta)=\int_{cqi-0.5}^{cqi+0.5}f(x-\Delta,m(g))dx=\int_{cqi-0.5-\Delta}^{cqi+0.5-\Delta}f(x,m(g))dx, \quad (4)$$

for $cqi\neq 1,30$ and $T(g)$ in (1) will also be a function of Δ, i.e., $T(\Delta)$. In the following, for clarity in writing expressions such as $T(\Delta)$, we omit the dependence on g. In fading channels, the average throughput $T(\Delta)$ is an average of (1) over all fading states and it depends on the average SNR. Use of CQI adjustment mechanisms in commercial networks has been reported in the literature [5]. Application of a CQI offset Δ is one such mechanism.

The objective of this paper is to maximize the average throughput over Δ, i.e.,

$$\max_{\Delta}T(\Delta). \quad (5)$$

Note that when $b_{cqi}(g)$ is known for example by experiment and curve fitting. We can solve (5) numerically.

III. Offline Throughout Optimization

A. Algorithm

The algorithm compares the current short term BLER with the target BLER and updates the CQI offset according to

$$\Delta_{n+1}=\left[\Delta_n-\alpha\left(\hat{b}_n(\Delta_n)-b^*\right)\right]^{\mathcal{U}}, \quad (6)$$

where Δ_n is the CQI offset at the n-th iteration, $\hat{b}_n(\Delta_n)$ is the short term BLER at the n-th iteration, and $[\cdot]^{\mathcal{U}}$ denotes the mapping onto set $\mathcal{U}$ to limit the range of CQI offset. For example, we could choose $\mathcal{U}=[-2,2]$ to limit the CQI offset between -2 and 2. Though the mapping $[\cdot]^{\mathcal{U}}$ is not necessary to achieve convergence, we use it to avoid undesired large CQI offset.

The short term average BLER $\hat{b}(\Delta)$ is estimated using the ACK/NACK history within a sliding window of size w. Let t_n be the starting time of the n-th window for CQI offset update with $t_1=1$, and $X(t)$ and $Y(t)$ denote the status of the first transmission ACK and NACK in subframe t, respectively, where $X(t)=1$ if a first transmission ACK is received and $X(t)=0$ otherwise, and $Y(t)=1$ if a first transmission NACK is received and $Y(t)=0$ otherwise. After choosing t_n, t_{n+1} is chosen such that the number of first time transmissions between t_n+1 and t_{n+1} is w, i.e., $\sum_{t=t_n+1}^{t_{n+1}}(X(t)+Y(t))=w$. The short term first time BLER is then estimated by

$$\hat{b}_n(\Delta)=\frac{\sum_{t=t_n+1}^{t_{n+1}}Y(t)}{w}. \quad (7)$$

As the status of ACK/NACK is available at both Node B and UE, the CQI update algorithm can be implemented at both sides. At the UE, since unquantized (or raw) CQI is available, we could add Δ_n to the raw CQI, $\mathrm{CQI_{raw}}$, directly, i.e., the reported CQI is the quantization of $\Delta_n+\mathrm{CQI_{raw}}$. However, in high geometry, $\mathrm{CQI_{raw}}$ may be greater than 30. Because

the maximum CQI is 30, UE should report 30 even though the current channel condition could support a data rate (TBS) higher than that corresponding to CQI 30. In this case, we cannot achieve the target first BLER no matter what CQI offset is applied because $b_{cqi}(g) < b^*$, $\forall cqi \in \{1, \ldots, 30\}$ and $b(\Delta)$ in (8) is less than b^* for all Δ. This problem can be resolved by using the Node B algorithm. At Node B, only reported quantized CQI from UE is available. After obtaining CQI offset Δ_n, it performs quantization $Q(\Delta_n + \text{CQI}_{\text{reported}})$ and transmits TBS corresponding to the CQI after quantization, where $\text{CQI}_{\text{reported}}$ is the reported CQI from UE and $Q(\cdot)$ is the CQI quantization function which maps its input to an integer between 1 and 30. To get a fine control of the BLER and throughput and achieve the target BLER even at high SNR, we can also change the power of the HS-PDSCH channel by using the residual CQI, i.e., the HS-PDSCH channel power is reduced by $(\Delta + \text{CQI}_{\text{reported}}) - Q(\Delta + \text{CQI}_{\text{reported}})$. By applying this method at Node B, when $\Delta + \text{CQI}_{\text{reported}} > 30$, the HS-PDSCH power is reduced such that there exists a cqi such that $b_{cqi}(g) > b^*$. Thus, the target BLER can still be achieved. This algorithm also leads to power saving at Node B. From the system point of view, this means that Node B could support more users and the overall system's performance is improved.

The proposed algorithm in this subsection can be used to enable fair resource allocation among UEs by assigning a fixed target BLER for each UE. In this case, even though a UE may report a higher CQI, the proposed algorithm will add a negative CQI offset to the reported CQI such that the scheduled TBS achieves the target BLER. Thus, the Node B actually monitors whether the reported CQI from each UE matches the current channel condition the UE suffers.

B. Monotonicity of Average First BLER

To derive the algorithm (6), we first show that the first BLER is a monotonic function in the CQI offset Δ. The average first BLER can be written as

$$b(\Delta) = \sum_{cqi=1}^{30} p_{cqi}(\Delta) b_{cqi}(g). \tag{8}$$

Taking the derivative of $b(\Delta)$ with respect to Δ and taking into account the expression of $p_{cqi}(\Delta)$ in (3), we obtain

$$\frac{db(\Delta)}{d\Delta} = \sum_{cqi=2}^{30} f(cqi - 0.5 - \Delta, m(g))(b_{cqi}(g) - b_{cqi-1}(g)). \tag{9}$$

Due to CQI table design, a higher cqi corresponds to a higher TBS, which consists of higher modulation and coding rate. Typically, $b_{cqi}(g) > b_{cqi-1}(g)$, $\frac{db(\Delta)}{d\Delta} > 0$ and $b(\Delta)$ is a strictly increasing function in Δ. Thus, in this case, given throughput optimal CQI offset Δ^*, there exists a unique BLER b^* such that $b(\Delta^*) = b^*$. This result also holds in fading channels after averaging over all fading states. As $b(\Delta)$ is monotone, there is a one-to-one correspondence between BLER and CQI offset. Therefore, finding the throughput optimal BLER is equivalent to finding the optimal CQI offset. In the following, we develop adaptive algorithms to achieve a given target first BLER. The optimal throughput can then be found by searching all possible target BLERs.

C. Algorithm Derivation

In this subsection, we derive the adaptive algorithm in (6) that adjusts the CQI offset Δ to achieve the target BLER b^* by minimizing the mean-squared error (MSE) between $b(\Delta)$ and b^*, i.e., $f(\Delta) = |b(\Delta) - b^*|^2$. We need to solve

$$\min_{\Delta} |b(\Delta) - b^*|^2. \tag{10}$$

To solve (10), we use gradient descent method via

$$\Delta_{n+1} = \Delta_n - \tilde{\alpha} \frac{df(\Delta_n)}{d\Delta} = \Delta_n - 2\tilde{\alpha} \frac{db(\Delta_n)}{d\Delta}(b(\Delta) - b^*), \tag{11}$$

where $\tilde{\alpha} > 0$ is a stepsize. As $b_{cqi}(g)$ and σ_{cqi}^2 are generally unknown, it is hard to compute $\frac{db(\Delta_n)}{d\Delta}$ especially in fading channels where the BLER should be averaged over all the fading states. We replace $\frac{db(\Delta_n)}{d\Delta}$ with $\text{sign}\left(\frac{db(\Delta_n)}{d\Delta}\right) = 1$, where $\text{sign}(\cdot)$ denotes the sign of its input. We thus modify (11) to

$$\Delta_{n+1} = \Delta_n - \alpha(b(\Delta) - b^*)\text{sign}\left(\frac{db(\Delta_n)}{d\Delta}\right) = \Delta_n - \alpha(b(\Delta) - b^*), \tag{12}$$

where $\alpha > 0$ is a stepsize. We find that this modification works well in practice even though the convergence rate may be slower as compared to (11).

Exact implementation of (12) requires the knowledge of $b(\Delta)$, which is hard to obtain online. We propose replacing the long term BLER $b(\Delta)$ in (12) with the short term BLER $\hat{b}_n(\Delta_n)$, which gives (6).

D. Convergence Analysis

In this subsection, we analyze the algorithm (6) without considering the mapping $[\cdot]^{\mathcal{U}}$. The analysis can be extended to the case with $[\cdot]^{\mathcal{U}}$ as in [3]. We only give the results without proof due to the space limitation. Let $M_{\max}$ denote the maximum value of $\frac{db(\Delta)}{d\Delta}$ over all possible Δ. We can show that if $0 < \alpha < \frac{2}{M_{\max}}$, there exists a constant $0 < \eta < 1$ such that

$$|E\{\Delta_{n+1}\} - \Delta^*| < \eta|E\{\Delta_n\} - \Delta^*|. \tag{13}$$

Therefore, $E\{\Delta_n\}$ converges to Δ^* exponentially and a larger stepsize α indicates a faster convergence rate when $0 < \alpha < \frac{1}{M_{\max}}$. Second order analysis shows that $E\{(\Delta_n - \Delta^*)^2\}$ also converges, and its limiting point satisfies

$$\lim_{n \to +\infty} E\left\{|\Delta_n - \Delta^*|^2\right\} < \frac{\alpha}{(2 + \alpha\delta)4w\delta}. \tag{14}$$

where $\delta > 0$ depends on $\frac{db(\Delta)}{d\Delta}$.

The previous two results concern the convergence of Δ_n, which is the main focus of SA [3]. In our problem, we do not really care about whether Δ_n converges. We are only interested in whether the time average of $\hat{b}_n(\Delta_n)$ converges. By using the martingale inequality [6], we get

$$\lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} \hat{b}_i(\Delta_i) = b^* \text{ with probability 1.} \tag{15}$$

IV. ONLINE THROUGHPUT OPTIMIZATION

In this section, we develop an online throughput optimization algorithm, which does not need to specify a target BLER. The online algorithm works for any CQI table and any SNR.

A. Algorithm

To maximize the average throughput $T(\Delta)$ in (5), we consider using gradient descent method to find the maximum throughput and its corresponding CQI offset. We assume that g is fixed until the algorithm converges. Let Δ_n be the CQI offset at the n-th iteration and $\frac{dT(\Delta)}{d\Delta}$ be the derivative of $T(\Delta)$ with respect to Δ. By using gradient descent method, the CQI offset update is given by

$$\Delta_{n+1}=\Delta_n+\alpha'\frac{dT(\Delta_n)}{d\Delta}, \tag{16}$$

where $\alpha'>0$ is a stepsize. Using (16) directly requires the knowledge of both the exact derivative and the average throughput function. We propose to replacing (16) with

$$\Delta_{n+1}=\Delta_n+\alpha'\frac{\hat{T}(\Delta_n+\epsilon)-\hat{T}(\Delta_n)}{\epsilon}, \tag{17}$$

where $\hat{T}(\Delta_n)$ is an estimate of $T(\Delta_n)$. As in Section III-C, we use a window based algorithm to estimate the average throughput. Let $\mathrm{TBS}(t)$ be the TBS transmitted in subframe t and $R(t)$ and $F(t)$ denote the status of the ACK/NACK, where $R(t)=1$ if an ACK is received and $R(t)=0$ otherwise (including NTX subframes), and $F(t)=1$ if an ACK or NACK is received and $F(t)=0$ otherwise. Suppose that the CQI offset Δ is updated every w subframes and the average throughput $T(\Delta_n)$ is computed using the past w received subframes. At the beginning of the n-th iteration, we compute $T(\Delta_n)$ via

$$\hat{T}(\Delta_n)=\frac{\sum_{t=(n-1)w+1}^{nw}R(t)\mathrm{TBS}(t)}{\sum_{t=(n-1)w+1}^{nw}F(t)}. \tag{18}$$

From experiments, we find that the choice of α' is very important to the algorithm. We suggest choosing $\alpha'=\frac{\alpha}{\bar{T}_n}$, where $\bar{T}_n$ is the average throughput up to the n-th iteration, i.e., $\bar{T}_n=\frac{\sum_{i=1}^{n}\hat{T}(\Delta_i)}{n}$. Then, (17) become

$$\Delta_{n+1}=\left[\Delta_n+\left[\frac{\alpha}{\bar{T}_n}\frac{\hat{T}(\Delta_n+\epsilon)-\hat{T}(\Delta_n)}{\epsilon}\right]^{\mathcal{S}}\right]^{\mathcal{U}}, \tag{19}$$

where $[\cdot]^{\mathcal{A}}$ denotes the mapping onto set $\mathcal{A}$, which is used to reduce the effect of the random variation of the throughput on the CQI offset update algorithm. For example, we could choose $\mathcal{S}=[-0.1,0.1]$ and $\mathcal{U}=[-2,2]$ to limit the absolute CQI offset increment within 0.1 and the absolute CQI offset within 2. As in (6), the mapping $[\cdot]^{\mathcal{A}}$ is used to avoid undesired large CQI offset. The algorithm (19) differs from the Kiefer-Wolfowitz algorithm [4] in that (19) uses a fixed α and ϵ, while the one in [4] uses decreasing α and ϵ.

In fading channels, the throughput may change over time. To track the throughput variation, we suggest adapting α and ϵ in (19). Let $\hat{T}'(\Delta_n)=\frac{\hat{T}(\Delta_n+\epsilon)-\hat{T}(\Delta_n)}{\epsilon}$, $F_n=\frac{\partial\Delta_n}{\partial\alpha}$, and $P_n=\frac{\partial\Delta_n}{\partial\epsilon}$. Ignoring the mapping in (19), motivated by [7], we take the partial derivative of both sides with respect to α, which gives $F_0=0$ and

$$F_{n+1}=\left(1+\frac{\alpha}{\bar{T}_n}\hat{T}''(\Delta_n)\right)F_n+\frac{1}{\bar{T}_n}\hat{T}'(\Delta_n). \tag{20}$$

Let α_n denote α in the n-th iteration. We update α_n using another gradient method as

$$\alpha_{n+1}=\left[\alpha_n+\mu_1\frac{\partial T(\Delta_n)}{\partial\alpha}\right]^{\mathcal{V}_1}\approx\left[\alpha_n+\mu_1\hat{T}'(\Delta_n)F_n\right]^{\mathcal{V}_1}, \tag{21}$$

where we have replaced $\frac{dT(\Delta_n)}{d\Delta}$ by its short term average $\hat{T}'(\Delta_n)$, $\mu_1>0$ is a stepsize and $\mathcal{V}_1$ is a constraint set, e.g., $\mathcal{V}_1=[0.0001,0.5]$. Similarly, taking the partial derivative of both sides of (19) with respect to ϵ, we obtain $P_0=0$ and

$$P_{n+1}=P_n+\frac{\alpha}{\bar{T}_n}(1+P_n)\hat{T}''(\Delta_n). \tag{22}$$

Let ϵ_n denote the ϵ in the n-th iteration. By using the gradient method, we update ϵ_n as

$$\epsilon_{n+1}=\left[\epsilon_n+\mu_2\frac{\partial T(\Delta_n)}{\partial\alpha}\right]^{\mathcal{V}_2}\approx\left[\epsilon_n+\mu_2\hat{T}'(\Delta_n)P_n\right]^{\mathcal{V}_2}, \tag{23}$$

where $\mu_2>0$ is a stepsize and $\mathcal{V}_2$ is a constraint set, e.g., $\mathcal{V}_2=[-0.2,0.2]$. In both (20) and (22), we approximate the second order derivative $\hat{T}''(\Delta_n)$ by

$$\hat{T}''(\Delta_n)=\frac{\hat{T}'(\Delta_n)-\hat{T}'(\Delta_k)}{\Delta_n-\Delta_k}, \tag{24}$$

where $k<n$ is the first integer such that $|\Delta_n-\Delta_k|>\epsilon_n$. Equations (19), (21) and (23) constitute the online throughput maximization algorithm.

To implement the online algorithm, we initially choose $\Delta_1=0$ and ϵ_1 to be a small value to induce the gradient method, e.g., $\epsilon_1=-0.1$. In the n-th iteration, we first evaluate $\hat{T}(\Delta_n)$ using a window of size w and then switch to $\Delta_n+\epsilon_n$ to compute $\hat{T}(\Delta_n+\epsilon_n)$ using another window. α_n, ϵ_n, Δ_n, F_n and P_n are then updated using (19)-(23), respectively. To prevent throughput degradation, we may give a sign of ϵ_n in the direction of the gradient $\hat{T}'(\Delta_{n-1})$. Similar to the offline algorithm in Section III-C, the online algorithm can also be implemented at both Node B and UE.

We have used a simple approximation of the gradient. There are other alternatives such as parabolic interpolation, see e.g., [3], which may improve the gradient estimation but at the expense of higher complexity. The online algorithm can also be readily extended to maximize the average throughput per OVSF code.

B. Convergence Analysis

If we allow α_n and ϵ_n decrease as $n\to\infty$ and α_n and ϵ_n satisfy

$$\epsilon_n\to0,\quad\sum\alpha_n=\infty,\quad\sum\alpha_n\epsilon_n<\infty,\quad\sum\alpha_n^2\epsilon_n^{-2}<\infty, \tag{25}$$

e.g., $\alpha_n=n^{-1}$ and $\epsilon_n=n^{-1/3}$, from [4], Δ_n converges to Δ^* stochastically as $n\to\infty$.

In practical implementation, a constant stepsize is typically desired. Existing literature lacks the analysis of constant stepsize Kiefer-Wolfowitz algorithm. Assuming that $0<m\le-T''(\Delta)\le M$, i.e., $T(\Delta)$ is a concave function with bounded curvature, we can show that with fixed α and ϵ $E\{\Delta_n\}$ converges to a small neighbor around Δ^* determined by $|\epsilon|$. A smaller $|\epsilon|$ means that $E\{\Delta_n\}$ is closer to Δ^*. However, this is at the expense of a larger MSE. Let $N(w)$ be the upperbound on the variance of $\hat{T}(\Delta_n)$, which is a decreasing function in w. We obtain

$$E\left\{|\Delta_n-\Delta^*|^2\right\}\le\frac{\epsilon^2M^2(1-\alpha m)}{2m^2(2-\alpha m)}+\frac{2\alpha N(w)}{\epsilon^2m(2-\alpha m)}+\frac{\alpha\epsilon^2M^2}{4m(2-\alpha m)}. \tag{26}$$

TABLE I

ACHIEVED FIRST BLER USING THE OFFLINE ALGORITHM IN DIFFERENT CHANNELS AND GEOMETRIES, WHEN THE TARGET BLER IS SET TO 15%.

Geometry (dB)	AWGN	PA3	PB3	VA30	VA120
0	15.11%	15.04%	15.02%	15.13%	15.10%
5	15.12%	15.03%	15.02%	15.11%	15.09%
10	15.05%	15.03%	15.02%	15.09%	15.05%
15	15.08%	15.02%	15.01%	15.12%	15.07%

TABLE II

GAIN IN THROUGHPUT AT OPTIMAL BLER RELATIVE TO THROUGHPUT AT 10% BLER IN DIFFERENT CHANNELS AT GEOMETRY=10dB.

	AWGN	PA3	PB3	VA30	VA120
Optimal BLER	15%	15%	20%	90%	90%
Throughput gain	0.3%	4.4%	6.6%	32.6%	16.7%

From (26), we can see that for fixed α, $E\left\{|\Delta_n - \Delta^*|^2\right\}$ may increase dramatically by decreasing $|\epsilon|$.

If the first order derivative $T'(\Delta)$ is also bounded by P, i.e., $|T'(\Delta)| < P$ for all Δ, which is the case for HSDPA throughput optimization, we can further show that

$$|E\{T(\Delta_n)\} - T(\Delta^*)| \leq \frac{|\epsilon|MP}{2m}. \tag{27}$$

Therefore, the average throughput also converges to a neighborhood of the local maximum throughput $T(\Delta^*)$.

V. SIMULATION RESULTS

We evaluate the performance of proposed throughput optimization algorithms using a C++ HSDPA system simulator. The CQI mapping table for UE category 10 specified in [2] is used in all tests. A parameter, geometry G, is used in the simulations, which is defined as $G = \frac{I_{or}}{I_{oc}+N}$, where I_{or} is the signal power from Node B in the target cell, I_{oc} represents the signal power received from adjacent cells and N is the thermal noise power. We consider AWGN, Pedestrian 3kmph (PA3, PB3), and Vehicular A 30kmph and 120kmph (VA30, VA120) channels [8].

The offline algorithm depends on two parameters: window size w and stepsize α. For a fixed window size w, the larger the α, the larger the deviation of the CQI offset, the lower the throughput, and the faster the convergence. There is a tradeoff between the achievable throughput and the convergence rate, as expected from the analysis in Section III-D. We used $w = 20$ and $\alpha = 1$ to generate the results below. Table I compares the BLER achieved using the proposed offline algorithm in different channels when the target BLER is 15%. We can see that the proposed offline algorithm can achieve the target BLER with a high accuracy. In the online algorithm, the window size should be chosen to be small so as to track the channel variation when channel varies fast.

A target BLER of 10% is not always throughput optimal. The throughput optimal BLER depends on the channel profile. In Table II, we see that the throughput optimal BLER varies from 10% to 90% for different channel profiles. At 10dB geometry, the throughput optimal BLER is 15% in AWGN and PA3, 20% in PB3, and 90% in VA30 and VA120. In some channels, the throughput at the optimal BLER is about 30% higher than the throughput at 10% BLER. The 30% throughput

gain is significant in practical systems and it is achieved with low complexity. The network should base its choice of target BLER on channel variations.

A note on our assumptions: in our simulations, we send retransmissions at the same transmit power as first transmissions - i.e. there is no power back-off in retransmissions. The presence of a power back-off in retransmissions influences the throughput optimal BLER in different path profiles, as explained below. Retransmissions at lesser power are sufficient to recover the data in semi-static channels. In dynamic channels, the channel state may change significantly in the time between CQI measurement and the corresponding downlink transmission. For example, in Vehicular A 30km/h, the channel fading changes to an independent state in roughly 4ms, i.e. 2 subframes. When the network receives a CQI report and schedules data accordingly, the channel condition in which this data is received may be very different from that suggested by the CQI, and retransmissions may be necessary. Depending on the power level of retransmissions relative to the first transmission, relying more on retransmissions by operating at a higher first BLER may improve throughput.

VI. CONCLUSION

We have investigated throughput optimization in HSDPA using two adaptive outer loop algorithms. Both algorithms adjust the CQI offset to maximize the throughput. The offline algorithm used an adaptive algorithm to achieve a given target BLER using the stochastic gradient descent method based on the history of ACK/NACK. By searching through different target BLERs, the throughput optimal BLER can be found offline. The online algorithm used a variation of the Kiefer-Wolfowitz algorithm without specifying a target BLER. An adaptive stepsize mechanism was also proposed to make the algorithm robust to non-stationary condition. We have shown the convergence of both algorithms with a constant stepsize. Simulation results show that the proposed algorithms can achieve up to 30% throughput improvement over that with 10% target BLER. Interplay between the algorithms proposed here and other system level optimizations such as opportunistic scheduling may be an area for further study.

REFERENCES

[1] "3GPP, TR 25.858 version 5.0.0, Physical Layer Aspects of UTRA High Speed Downlink Packet Access ," Mar. 29, 2002.

[2] "3GPP, TR 25.214 version 7.0.0, Physical layer procedures (FDD)," Mar. 22, 2006.

[3] H. J. Kushner and G. G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*, 2nd ed. Springer-Verlag, 2003.

[4] J. Kiefer and J. Wolfowitz, "Stochastic estimation of the maximum of a regression function," *Annals of Mathematical Statistics*, vol. 23, no. 3, pp. 462–466, Sept. 1952.

[5] J. Derksen, R. Jansen, M. Maijala, and E. Westerberg, "HSDPA performance and evolution," *Ericsson Review*, vol. 3, pp. 117–120, 2006.

[6] L. Breiman, *Probability Theory*. Addison-Wesley, 1968. Republished (1991) in Classics of Mathematics. SIAM, Philadelphia.

[7] H. J. Kushner and J. Yang, "Analysis of adaptive step size SA algorithms for parameter tracking," *IEEE Trans. Automat. Contr.*, vol. 40, no. 8, pp. 1403–1410, Aug. 1995.

[8] "3GPP, TR 25.101 version 8.3.0, Technical specification group radio access network; user equipment (UE) radio transmission and reception (FDD)," June 6, 2008.