Optimized Channel Utilization in Multi-Carrier Wireless Mobile Networks

Amrinder Arora, Fanchun Jin and Hyeong-Ah Choi

Dept. of Computer Science, George Washington University, Washington DC 20052, USA,
{amrinder,jinfc,hchoi}@gwu.edu

Abstract. This paper considers the problem of maximizing throughput in a multi-carrier wireless network that employs predictive link adaptation. The contributions of this paper are the following. We explicitly consider the time-penalty induced by the transmission-mode changes in wireless networks. We study the trade-offs between the channel-state prediction quality and the throughput. We provide sound offline problem analysis, with optimal or approximate solutions to the extent possible. We consider online version of the wireless channel scheduling problem, and provide a lower bound and propose algorithms that guarantee competitive ratios. We also extend our algorithms to accommodate the model when the actual channel capacity is different from as predicted earlier. Our results show that a modest consumption of resources for channel prediction and link adaptation may result in a significant throughput improvement, with only marginal gains through further enhancement of the prediction quality.
Keywords: Wireless networks, link adaptation, throughput maximization, offline optimal algorithm, online scheduling, competitive ratio, predictive scheduling.

1 Introduction

Channel-aware scheduling and link adaptation methods are widely considered to be crucial for realizing high data rates in wireless networks. Link Adaptation (LA), which loosely refers to changing transmission parameters, including modulation, coding rate, and power, over a link in response to changing channel conditions over time is considered to be a powerful means of achieving higher efficiency or throughput in wireless networks. The adaptation of the transmission parameters is performed according to the predicted future quality of the channel, also called as the channel-state (CS).

While it is desirable to adapt the transmission parameters according to the channel state information (CSI) to capture even small-scale variations, there are practical limitations to frequent link adaptation. Fast adaptation increases the number of mode-change messages transmitted over the channel, consuming bandwidth, and time resources [1]. While many aspects of scheduling transmissions over time-varying wireless channels have been studied (see, for example, [2–4] and the references therein), the penalty induced by LA has not been considered. Moreover, predicting the future channel quality may also consume significant amount of system resources (e.g., time, bandwidth and

power), since it may involve transmission of training-sequences, pilot tones, or feedback messages carrying the CSI [1]. Naturally, the additional cost and complication of predicting CS and LA increases with the desired prediction accuracy.

In this paper, we consider the scheduling problem in a multi-carrier wireless network that uses LA. The contributions of this paper are the following. We explicitly consider the time-penalty induced by the transmission-mode changes in wireless networks. We study the trade-offs between the channel-state prediction quality and the throughput. We provide sound offline problem analysis, with optimal or approximate solutions to the extent possible. We consider online version of the wireless channel scheduling problem, and provide a lower bound and propose algorithms that guarantee competitive ratios. We also extend our algorithms to accommodate the model when the actual channel capacity is different from as predicted earlier. Our results show that a modest consumption of resources for channel prediction and link adaptation may result in a significant throughput improvement, with only marginal gains through further enhancement of the prediction quality. *The results we obtain can provide meaningful guidelines in deciding what level of system resource consumption is justified for channel quality estimation and link adaptation.*

This paper is organized as follows. A background on link adaptation techniques is given next. System model and problem statement are given in Section 2. In Section 3, we present underlying scheduling algorithms that are used in the rest of the paper. Section 4 considers the case when multiple users share a single channel, and presents a dynamic programming algorithm for an offline optimal solution and a 2.598-competitive online algorithm. Section 5 considers the case when multiple users share multiple channels and presents an offline 2-optimal algorithm, a 4-competitive online algorithm, and an upper bound (shown to be near-optimal in simulations) to the offline optimal solution. In Section 6, we consider the model when channel estimates are imperfect (i..e, the estimate for a future channel is different from the actual one), and present our solution approach based on the results presented in Sections 4 and 5. Numerical results in imperfect channel estimates models are presented in Section 7. The paper concludes in Section 8.

1.1 Link Adaptation Techniques

Link Adaptation (LA) attempts to exploit the variations in the quality of the wireless channel by modifying a set of transmission parameters, such as the modulation, and coding modes. This is suitable for duplex communication, since the transmission parameters have to be adapted using some form of two-way transmission [5, 1].

The three components of link adaptation are: (i) estimation of the channel quality (ii) transmission parameter computation, and (iii) signaling (or blind detection) of the selected parameters.

In this paper, we focus on the link adaptation for downlink transmission of data over a channel from a base-station to a mobile terminal. In this context, the adaptation of the transmission parameters is done by the transmitter at the base station.

2 System Model and Problem Statement

We assume that, at the beginning of time slot t, the base station has perfect knowledge of the channel state in time slot t, and *perfect* or *imperfect* estimates of the channel state over timeslots $t+1, t+2, \cdots, t+h$. We will refer to h as the *look-ahead*. Given this information at the beginning of each time slot t , the base station (BS) decides to which mobile system (MS), on what channel, and at what data rate, it is going to transmit during time slot t. If a change is required in the receiving MS or the data rate (by varying the modulation rate, coding rate, etc.), then the next downlink transmission (in time-slot t) is used to notify the receiver of the new data rate, which may be confirmed by the MS through an ACK in the next uplink transmission. Thus, transmission at the new data rate (possibly to a new user) starts only after a delay of a full duplex transmission cycle. Let us assume that each time-slot, of duration δ, consists of a downlink transmission of $\delta/2$ followed by an uplink transmission of $\delta/2$. Thus, if the BS decides to change the transmission rate (or the MS) at the beginning of time slot $t\delta$, then transmission with the new rate starts at the beginning of time slot $(t+1)\delta$. The duration of each timeslot is assumed to be equivalent to this time-penalty δ. Hence, the earliest time to have a new data rate after timeslot t is $t+2$. The same penalty is assumed to be applied when a user is changed on the same channel.

We assume that there are m users, f channels, and n timeslots. The maximum rate estimation that can be used by user i on channel j in timeslot t is denoted by $c(i,j,t)$, and the rate assigned by the scheduler to user i on channel j in timeslot t is denoted by $x(i,j,t)$. Clearly, $x(i,j,t) \leq c(i,j,t)$ for each $1 \leq i \leq m$, $1 \leq j \leq f$, and $1 \leq t \leq n$.

The goal is to maximize $\sum_{i=1}^{m} \sum_{j=1}^{f} \sum_{t=1}^{n} x(i,j,t)$. The decision has to satisfy the following system constraints.

- A channel can be used for transmission to only one MS at a given time slot.
- The data transmission rate cannot exceed the time- and user-dependent maximum rate.
- Changing the data rate, or the intended MS, makes the next time slot useless in terms of data transmission.

We explore two versions of the scheduling problem in this paper: perfect and imperfect channel estimates models. At the beginning of timeslot t, the channel state over timeslots $t, t+1, \cdots, t+h$ are perfectly known in the perfect channel estimates model with h look-aheads, whereas only the channel state in t is perfectly known in the imperfect channel estimates model.

2.1 Perfect Channel Estimates Model

P1: *Given a channel status matrix $C = [c(i,j,t)]$ for $1 \leq i \leq m$, $1 \leq j \leq f$, and $1 \leq t \leq n$, the objective is to find schedule $X = [x(i,j,t)]$ that maximizes $\sum_{i=1}^{m} \sum_{j=1}^{f} \sum_{t=1}^{n} x(i,j,t)$ such that (i) $x(i,j,t) \leq c(i,j,t)$ for each $1 \leq i \leq m$, $1 \leq j \leq f$, and $1 \leq t \leq n$ and (ii) for any j, if $x(i,j,t) > 0$ and $x(i',j,t+1) > 0$, then $i = i'$ and $x(i,j,t) = x(i,j,t+1)$.*

For example, if $m = f = 1$ (i.e., single-user single-channel) and $C = [1, 3, 7, 8, 7, 15, 14]$, then $X_1 = [0, 0, 7, 7, 7, 0, 14]$ and $X_2 = [1, 0, 7, 7, 0, 14, 14]$ are two feasible solutions, where X_2 achieves more throughput.

2.2 Imperfect Channel Estimates Model

Let $c'_t(i, j, q)$, for $t+1 \leq q \leq t+h$, be the *estimate* of $c(i, j, q)$ measured in time slot t. (If $q > n$, assume $c'_t(i, j, q) = 0$.) Let $P_t^{i,j,q}(a, b)$ be the probability that $c(i, j, q) = b$, given that $c'_t(i, j, q) = a$. Consequently, $P_t^{i,j,q}(a, a)$ denotes the prediction accuracy for estimate a. At the beginning of each time slot t, the BS has the knowledge of $x(i, j, t-1)$, $c(i, j, t)$, $c'_t(i, j, q)$, and $P_t^{i,j,q}(a, b)$ for each user i.

P2: *Given a channel status matrices $C = [c(i, j, t)]$ and $C' = [c'_t(i, j, q)]$, and estimation error probability $P_t^{i,j,q}(a, b)$, for $1 \leq i \leq m$, $1 \leq j \leq f$, $1 \leq t \leq n$, and $t + 1 \leq q \leq t + h$, the objective is to maximize $\sum_{i=1}^{m} \sum_{j=1}^{f} \sum_{t=1}^{n} x(i, j, t)$ such that (i) $x(i, j, t) \leq c(i, j, t)$ for each $1 \leq i \leq m$, $1 \leq j \leq f$, and $1 \leq t \leq n$, and (ii) for any j, if $x(i, j, t) > 0$ and $x(i', j, t + 1) > 0$, then $i = i'$ and $x(i, j, t) = x(i, j, t + 1)$.*

2.3 Notations

Following notations are used consistently in this paper:

- m: Total number of users
- f: Total number of channels
- n: Total number of timeslots
- h: Lookahead for the online algorithm
- $c(i, j, t)$: Capacity for i-th user on j-th channel, during t-th timeslot
- $c'_t(i, j, q)$: Estimate of $c(i, j, q)$ done in timeslot t.
- $P_t^{i,j,q}(a, b)$: Prob. that actual capacity $c(i, j, q) = b$ given estimate $c'_t(i, j, q) = a$
- $P_t^{i,j,q}(a, a)$: Prob. that actual capacity $c(i, j, q) = a$ given estimate $c'_t(i, j, q) = a$
- $x(i, j, t)$: Allocation for i-th user on j-th channel, during t-th timeslot
- i, j, t: Indices for user, channel and timeslot respectively

3 Theoretical Foundations

To prepare solid theoretical foundations, we first of all consider the single-user single-channel version of the problem. Throughout this section, the notation $c(t)$ and $x(t)$ will be used instead of $c(i, j, t)$ and $x(i, j, t)$ for simplicity.

Firstly, we focus on finding a lower bound for the competitive ratio for a one-lookahead online algorithm for problem **P1**. Note that if the performance of an online algorithm $\mathcal{A}$ is c-times worse than the performance of an optimal offline algorithm, then algorithm $\mathcal{A}$ is said to be c-competitive.

3.1 Lower Bound

Theorem 1. *For the single-user single-channel version of problem* **P1**, *no one-lookahead algorithm can be better than* $3/2$*-competitive.*

Proof: Consider a deterministic 1-lookahead online algorithm $\mathcal{A}$ for single-user single-channel version of problem **P1**. For the purpose of this discussion, let $a(t)$ denote the output selected by algorithm $\mathcal{A}$ during timeslot t and let $x(t)$ denote the output selected by the constructed offline solution.

Consider input of $1, 2, c(3)$, where value of $c(3)$ is controlled by an adaptive online adversary. If algorithm $\mathcal{A}$ chooses $a(1) > 0$ in the first slot, adversary sets the value of $c(3) = a(1) - \epsilon$. In this case, the algorithm can only achieve a total allocation of $2a(1)$, while optimal total allocation is $3a(1) - 3\epsilon$. If algorithm chooses $a(1) = 0$ in the first slot, adversary sets the value of $c(3) = 1$. In this case, the algorithm can only achieve a total allocation of 2, while optimal total allocation is 3. Thus, no 1-lookahead online algorithm can achieve a total allocation of more than $2/3$ times that of optimal. ■

3.2 Wait-Dominate-Hold Algorithm

We use the Wait-Dominate-Hold Algorithm presented in [6]. It uses the usage of the previous time slot $x(t-1)$, the current capacity $c(t)$, and the next capacity $c(t+1)$. Let $m(t)$ denote the maximum possible current usage given $x(t-1)$ and $c(t)$ defined as: (a) $m(t) = 0$ if $x(t-1) > 0$ and $c(t) < x(t-1)$, (b) $m(t) = c(t)$ if $x(t-1) = 0$, and (c) $m(t) = x(t-1)$ if $c(t) \geq x(t-1) > 0$. An example usage of the Wait-Dominate-Hold algorithm is shown in Table 1. The precise set of rules for choosing $x(t)$ is as follows:

($\mathcal{R}1$) If $c(t+1) > 2m(t)$, set $x(t) = 0$.
($\mathcal{R}2$) If $m(t) \geq 2c(t+1)$, set $x(t) = m(t)$ (we note that $x(t+1) = 0$ will be assigned in the next slot schedule).
($\mathcal{R}3$) If $c(t+1)/2 \leq m(t) < 2c(t+1)$ and $x(t-1) > 0$, set $x(t) = m(t)$.
($\mathcal{R}4$) If $c(t+1)/2 \leq m(t) < 2c(t+1)$ and $x(t-1) = 0$, set $x(t) = \min\{m(t), c(t+1)\}$.

3.3 Wait-Dominate Algorithm

We generalize the WD algorithm presented in [6], by introducing a parameter ψ and modifying the algorithm definition as follows. We refer to the modified algorithm as WD(ψ). The modified WD algorithm uses the usage of the previous time slot $x(t-1)$, the current capacity $c(t)$, and the next capacity $c(t+1)$. Let $m(t)$ denote the maximum possible current usage given $x(t-1)$ and $c(t)$ defined as: (a) $m(t) = 0$ if $x(t-1) > 0$ and $c(t) < x(t-1)$, (b) $m(t) = c(t)$ if $x(t-1) = 0$, and (c) $m(t) = x(t-1)$ if $c(t) \geq x(t-1) > 0$.

The precise set of rules for choosing $x(t)$ is as follows:

($\mathcal{R}1$) If $c(t+1) > \psi\, m(t)$, set $x(t) = 0$.
($\mathcal{R}2$) If $\psi\ m(t) \geq c(t+1)$, set $x(t) = m(t)$ (we note that $x(t+1) = 0$ will be assigned in the next slot schedule).

An example usage of the Wait-Dominate algorithm using $\psi = 2$ is shown in Table 1. We observe that every non-zero usage from the WD algorithm is surrounded by 0 in preceding and succeeding timeslots.

Table 1. Example usage of Wait-Dominate-Hold and Wait-Dominate algorithms, using $\psi = 2$.

C:	23	23	7	5	15	31	62	3	7	7	15	17
X(WDH):	23	23	0	0	0	31	31	0	7	0	15	15
X(WD,$\psi = 2$):	23	0	7	0	0	31	0	0	7	0	15	0

Competitive Analysis

Theorem 2. *Wait-Dominate algorithm WD(ψ) is $\frac{\psi^3}{\psi^2-1}$-competitive.*

Proof: Define a block B_i as $[c(1), c(2), \ldots, c(k), c(k+1)]$, s.t.

$$c(i) > \psi\, c(i-1) \quad 1 < i \leq k$$
$$c(k+1) \leq \psi\, c(k)$$

Now, say we define a new capacity array C' as follows:

$$c'(i) = c(i) \ \ \forall 1 \leq i \leq k$$
$$c'(k+1) = \psi\, c(k)$$

We observe that the array C' as defined is strictly larger than the original capacity array C. Thus, OPT$(C) \leq$ OPT(C').

By using a simple geometric progression on alternate timeslots, we observe that OPT$(C') \leq c'(k+1)\psi^2/(\psi^2-1)$. Let $|X|$ denote the allocation achieved by WD algorithm. Then, to summarize:

$$\begin{aligned} |X| &= c(k) \\ \text{OPT}(C) &\leq \text{OPT}(C') \\ \text{OPT}(C') &\leq c'(k+1)\ \psi^2/(\psi^2-1) \\ &= c(k)\ \psi^3/(\psi^2-1) \\ \Rightarrow \text{OPT}(C) &\leq \psi^3/(\psi^2-1)|X| \end{aligned}$$

Since the allocation for the last timeslot of each block is always 0, the analysis for each block is independent. Thus, the result follows by extrapolating the result over the sum of all blocks. ■

Theorem 3. *Wait-Dominate algorithm WD(ψ) always achieves at least $\frac{\psi-1}{\psi^2}$ fraction of the total throughput.*

Proof: Proof is similar to that of Theorem 2 and is omitted for space constraints. ■

Using the theorems presented above, we can easily reach the following results.

Corollary 1. *WD algorithm can be tuned to yield competitive ratio of $3\sqrt{3}/2$, that is, 2.598.*

Proof: Using a value of $\psi = \sqrt{3}$ in Theorem 2 yields the desired result. ■

Corollary 2. *WD algorithm can be tuned to achieve $1/4$ of total capacity.*

Proof: Using a value of $\psi = 2$ in Theorem 3 yields the desired result. ■

4 Multiple User Single Channel

In this section, we consider the special case of problem **P1** in which there is only one channel. That is, multiple users share the same channel. We call that channel 1, and so, all capacity variables used in this section are of the form $c(i, 1, t)$. We begin by considering the offline version of the problem.

4.1 Dynamic Programming Optimal Algorithm

We propose the following algorithm $\mathcal{MUSC}$-$\mathcal{OPT}$ for multiple user, single channel offline version of the problem. The algorithm works as follows:

Let $A(t_1, t_2)$ ($t_1 < t_2$) denote the optimal solution from timeslot t_1 to t_2 both inclusive. The algorithm is based upon the following observation.

1. **Case I:** Only one user uses the channel from timeslot t_1 to t_2 without changing transmission rate. In this case, due to capacity constraints, the allocation must be the minimum capacity for that user over all time slots.
2. **Case II:** There exists timeslot k between timeslots t_1 and t_2 which has 0 transmission rate (this is the timeslot for which the users and/or transmission rates were changed). In this case, we can obtain the solution, assuming we have already calculated $A(t_1, k-1)$ and $A(k+1, t_2)$, where k may be any integer between t_1 and t_2 both inclusive.

$A(t_1, t_2)$ is the maximum of the solutions obtained from both cases. We can solve this problem with dynamic programming algorithm, and the formulation is:

$$A(t_1, t_2) = \max \begin{cases} (t_2 - t_1 + 1) * \max_{1 \le i \le m} \{\min_{t_1 \le k \le t_2} c(i, 1, k)\} & \text{(Case I)} \\ \max_{t_1 \le k \le t_2} \{A(t_1, k-1) + A(k+1, t_2)\} & \text{(Case II)} \end{cases} \quad (1)$$

For the special case, where $t_1 = t_2$, $A(t_1, t_2)$ is equal to the maximum capacity for the channel possible from all users at timeslot t_1. That is, $A(t_1, t_1) = \max_{1 \leq i \leq m} c(i, 1, t_1)$ for each i, $1 \leq i \leq m$, Also, for the sake of completeness of above recurrence, we define: $A(t, t-1) \stackrel{\text{def}}{=} 0$.

Theorem 4. *Algorithm $\mathcal{MUSC}$-$\mathcal{OPT}$ is optimal.*

Proof: (By induction on the length of time interval $t_2 - t_1 + 1$).

Base Step: If the length of time interval is 1, then the optimum solution is simply by choosing the user that has the maximum capacity for that timeslot. This is the solution returned by $\mathcal{MUSC}$-$\mathcal{OPT}$ in Case I.

Inductive Hypothesis: $\mathcal{MUSC}$-$\mathcal{OPT}$ is optimal for all time intervals of size up to n.

Induction Step: Say the optimum solution for a problem where length of time interval is n is OPT. As noted in the above observation, OPT contains either one time interval where capacity is 0, or it consists of a user using the channel capacity uniformly for all timeslots. $\mathcal{MUSC}$-$\mathcal{OPT}$ considers both the cases and returns the maximum. ■

4.2 Time Complexity Analysis

Theorem 5. *Algorithm $\mathcal{MUSC}$-$\mathcal{OPT}$ runs in $O(mn^3)$ time.*

Proof: The maximum range of variable k in the inductive step of the algorithm is from 1 to m, and the maximum range of variable k in the inductive step of the algorithm is from 1 to n, Therefore each element of array A can be computed in $O(mn)$. Since there are n^2 entries, the time complexity of algorithm is $O(mn^3)$. ■

Remark 1: While we have shown how the value $A(1, n)$ can be computed, we have skipped the details of how the optimum solution $x(i, 1, t)$ is stored or tracked. This can be accomplished by using a second matrix that stores the (timeslot of last 0, user number and the allocation) combination.

Remark 2: The time complexity of this algorithm is quite easily improved to $O(n^2(m+n))$ by using dynamic programming for calculating the Case I elements that are used in Equation (1) in a preliminary phase.

4.3 2.598-Competitive Online Algorithm

In the online version, we assume that all the channel capacities are not known ahead of time, and only a very limited lookahead is given. This version is a more practical approach, and we extend the results obtained from the offline version both for obtaining online algorithms and for competitive analysis.

Next we formally specify algorithm $\mathcal{MUSC}$-$\mathcal{ON}$ for online variation of multiple user single channel problem. The algorithm $\mathcal{MUSC}$-$\mathcal{ON}$ is a 1-lookahead algorithm based on the WD algorithm discussed in section 3.3. The precise definition for choosing $x(i, 1, t)$ is as follows:

1. For each timeslot, compute the maximum capacity over all users. Thus from the matrix $c(i,1,t)$, we compute a one-dimensional matrix $C = [c_{max}(t)]$, where $c_{max}(t) = \max_{1 \leq i \leq m} c(i,1,t)$.
2. Use the WD algorithm, to allocate based upon the one-dimensional matrix $C = [c_{max}(t)]$.

We remind the reader that WD algorithm has the property that each non-zero allocation is surrounded by 0 allocation in preceding and succeeding timeslots. Next we present the competitive analysis of the $\mathcal{MUSC}$-$\mathcal{ON}$ algorithm by comparing it to the performance of the optimal offline algorithm.

Theorem 6. *Algorithm $\mathcal{MUSC}$-$\mathcal{ON}$ is 2.598-competitive.*

Proof: Proof is largely based upon the known result for WD algorithm which is known to be 2.598-competitive. Say the optimal channel allocation matrix is given by X, and the total throughput is OPT. Since only one user can be active at one time,

$$OPT \leq \sum_{1 \leq t \leq n} \max_{1 \leq i \leq m} c(i,1,t) \tag{2}$$

Say the solution returned by $\mathcal{MUSC}$-$\mathcal{ON}$ is β. Since WD is 2.598-competitive, it implies that

$$\beta \geq \frac{1}{2.598} \sum_{1 \leq t \leq n} c_{max}(t) \tag{3}$$

From equations (2) and (3), we obtain that $\beta \geq OPT/2.598$. ■

Time Complexity for $\mathcal{MUSC}$-$\mathcal{ON}$ The optimal solution for each time slot can be computed $O(m)$ time, where m is the number of users. After that, decision of whether or not to use that timeslot can be made in constant time. The overall running time is thus $O(m)$ for each timeslot, and $O(mn)$ total for all timeslots.

5 Multiple User Multiple Channel

In this section, we consider the most general form of the problem that is presented in Section 2.1. We observe that in this case the we are given a total of fmn entries (one entry per user-channel combination for each of the n timeslots). Our goal is to find out the allocation for each of the entries such that the solution is feasible, and the total throughput is maximized.

We assume that each user is only allowed to use at most one channel in any given timeslot. Hence, for each of the timeslots, only $\min\{f,m\}$ entries can be non-zero, and from the total data set, at most $n \cdot \min\{f,m\}$ entries can be non-zero. Note that such a constraint can be due to a user's hardware constraint (e.g., each user is equipped with only one transceiver).

5.1 Maximum Throughput for One Timeslot

In this section, we consider the special case when there is only one timeslot and this provides a base step that is used repetitively in the remainder of this section.

If there is only one timeslot, then the problem is defined as: Given m users and f channels, with $c(i, j, 1)$ being the capacity for i-th user and j-th channel, we want to find an assignment that maximizes the throughput. An example capacity matrix for 6 users and 6 channels is shown in Table 2.

Table 2. Example capacity matrix for 6 users and 6 channels for 1 timeslot, with maximum matching solution for maximum throughput. An optimal solution is shown in bold.

	U1	U2	U3	U4	U5	U6
C1	3	8	3	7	4	**8**
C2	**11**	7	3	9	0	2
C3	3	2	7	1	**9**	7
C4	3	**9**	3	9	8	2
C5	7	5	**8**	3	4	8
C6	8	5	3	**12**	11	4

For a single timeslot, the problem of finding the maximum throughput is identical to finding the maximum weighted matching in the user-channel bipartite graph. Maximum weighted matching in a bipartite graph is a very well studied problem, and an optimal solution can be found in $O(m^{2.5})$ time [7, 8]. (Here, we assume that $m \geq f$ without loss of any generality in our system model.)

5.2 2-Approximation Offline Algorithm

We propose the following algorithm $\mathcal{MUMC}$-2 for multiple user, multiple channel offline version of the problem. The algorithm works in two stages:

- **Step 1:** Use maximum matching to calculate the optimal user-channel assignment for each timeslot, independent of preceding and succeeding timeslot. Say the output of 1st step is an array: $\alpha = [\alpha(1), \alpha(2), \alpha(3), \ldots, \alpha(n)]$.
- **Step 2:** Optimally select timeslots such that the total value is maximized and no adjacent timeslots are chosen. This can be done using a single pass dynamic programming algorithm, that is based upon the following recurrence. $\beta(k) = \max\{\beta(k-2) + \alpha(k), \beta(k-1)\}$.

We observe that the step 2 of this problem is different from the single user single channel offline problem. In the single user single channel problem, it is permissible for two adjacent timeslots to be both nonzero provided they are equal. However, for the step 2 of this problem, even if $\alpha(i) = \alpha(i+1)$, we still cannot allocate both of the slots, as the overall solution may not be feasible.

Lemma 1. *$\beta(n)$ is at least half of sum of $\alpha(t)$, i.e., $\beta(n) \geq \frac{1}{2}\sum_{t=1}^{n}\alpha(i)$.*

Proof: Consider two specific feasible solutions: (i) feasible solution in which only odd timeslots are chosen, and (ii) feasible solution in which only even timeslots are chosen.

Since $\beta(n)$ is optimal solution, it must be greater than both these feasible solutions. That is,

$$\beta(n) \geq \sum_{i \text{ is odd}} \alpha(i) \text{ and } \beta(n) \geq \sum_{i \text{ is even}} \alpha(i)$$

$$\Rightarrow 2\beta(n) \geq \sum_{i \text{ is odd}} \alpha(i) + \sum_{i \text{ is even}} \alpha(i) = \sum_{i=1}^{n} \alpha(i)$$

$$\Rightarrow \beta(n) \geq \frac{1}{2}\sum_{i=1}^{n} \alpha(i).$$

Theorem 7. *Algorithm $\mathcal{MUMC}$-2 is 2-Approximation.*

Proof: Say the optimum solution for the multiple user multiple channel is OPT. In that case, the OPT must be less than the sum of the optimum solutions obtained for each time slot. That is, $OPT \leq \sum_{1}^{n}\alpha(i)$. Using Lemma 5.2, $\beta(n) \geq \frac{1}{2}\sum_{1}^{n}\alpha(i)$. Hence, $\beta(n) \geq \frac{1}{2}OPT$. ■

Time Complexity Analysis In the first stage, we use maximum weighted matching algorithm which runs in $O(m^{2.5})$ time, for each timeslot. Thus, the total time of execution for the first stage is $O(nm^{2.5})$. The second stage runs in $O(n)$, resulting in an overall time complexity of $O(nm^{2.5}) + O(n)$, i.e., $O(nm^{2.5})$.

5.3 4-Competitive Online Algorithm

In this section, we study a 1-lookahead algorithm for scheduling in multiple user multiple channel network. This version is a more practical approach compared to the offline version, and we extend the results obtained from the offline version both for obtaining online algorithms and for competitive analysis.

The algorithm $\mathcal{MUMC}$-$\mathcal{ON}$ is specified as follows:

- **Step 1:** Calculate the maximum matching for each time slot independently. This step is the same as Step 1 for the offline 2-approximation algorithm $\mathcal{MUMC}$-2 . Again, we assume that the value of the maximum matching for the timeslot t is given by $\alpha(t)$.
- **Step 2:** Choose the timeslots to use using the Wait-Dominate (WD) algorithm, using the parameter $\psi = 2$. We observe that WD algorithm has the property that each non-empty allocation is surrounded by 0 allocation in preceding and succeeding timeslots.

Theorem 8. *Algorithm $\mathcal{MUMC}$-$\mathcal{ON}$ is 4-competitive.*

Proof: The proof is based on the proof for WD algorithm that is used in Step 2. Let the value of an optimal offline algorithm be given by OPT. Since the optimum value cannot exceed the sum of optimum values obtained independently for each timeslot, we have $OPT \leq \sum_{t=1}^{n} \alpha(t)$. Let the total throughput using the $\mathcal{MUMC}$-$\mathcal{ON}$ be β. Then using the fact that when using $\psi = 2$, WD algorithm always achieves at least $1/4$ of the total capacity, we have $\beta \geq \frac{1}{4} \sum_{t=1}^{n} \alpha(t)$. Combining these two equations, we obtain that $\beta \geq \frac{1}{4} OPT$. ■

5.4 Upper Bound for Performance Analysis

In section 5.3, we presented a 4-competitive online algorithm. However, the ratio is against the sum of maximum usage in each time slot taken separately, which provides a loose upper bound on the optimal. In fact, the sum of maximum possible usage in each time slot taken separately may be twice as much as the optimal. We denote that upper bound by U_1, where $U_1 = \sum_{t=1}^{n} \alpha(t)$.

From the analysis in Section 4.1, we observe that optimal throughput for multiple channels is no more than the sum of optimal solution $\mathcal{MUSC}$-$\mathcal{OPT}$ for each channel considered separately. We use that to propose the following alternate upper bound in which $OPT_{MUSC}(j)$ denotes the optimal throughput for channel j obtained using the $\mathcal{MUSC}$-$\mathcal{OPT}$ algorithm. We denote this upper bound by U_2, where $U_2 = \sum_{j=1}^{f} OPT_{MUSC}(j)$.

The overall upper bound is now given by $U = \min\{U_1, U_2\}$.

In our numerical analysis we observed that the channel based optimal upper bound (U_2) is almost always tighter than the upper bound achieved using the maximum matchings (U_1). In fact, simulation results in Section 7 show that U_2 is near-optimal.

6 Imperfect Channel Estimates

Both the online and offline versions of the problems considered before assume that the channel state is known deterministically. In reality, however the channel state for even the next timeslot may have a certain error probability. The channel state may change rapidly based upon the user's movements, or other network parameters, thus affecting the error probability for the predicted channel capacity.

In this section, we consider 1-lookahead for multiple user single channel and multiple user multiple channel versions of the problem, assuming imperfect channel estimation. Thus our 1-lookahead online algorithm should use the true capacity $c(i, j, t)$ for the current timeslot and the estimated capacity $c'(i, j, t+1)$ for the next timeslot. Since we are only considering 1-lookahead algorithms (i.e., $h = 1$), we simplify the notation used in Section 2.3. Henceforth, $c'(i, j, t+1)$ is the estimated capacity for i-th user on j-th channel during the $(t+1)$-th timeslot, as estimated before the beginning of timeslot t. Of course, the actual capacity in the $(t+1)$-th timeslot may be higher or lower than $c'(i, j, t+1)$.

6.1 Multiple User Single Channel

For incorporating the case of imperfect estimates for multiple user single channel, we modify WDH discussed in Section 3.2 as follows.

The modified algorithm is called *Weighted-Wait-Dominate-Hold* (WWDH) and accepts weight parameter (w), where w is the weight that the algorithm uses when considering the projected allocation in the next time slot $(t+1)$. The algorithm is defined as follows (Please note that for single channel case, j is always equal to 1).

1. For each user i, first form a tentative two-slot schedule using the same rules as WDH. That is, calculate $x(i,1,t)$ and $x'(i,1,t+1)$ using $x(i,1,t-1)$, $c(i,1,t)$ and $c'(i,1,t+1)$. Again, we note that $x'(i,1,t+1)$ is the projected allocation for the i-th user during the $(t+1)$-th timeslot.
2. Compute $x(i,1,t) + wx'(i,1,t+1)$.
3. Say the user i^* has the maximum value of $x(i,1,t) + wx'(i,1,t+1)$, i.e., $i^* = \arg\max_{1 \le i \le m}\{x(i,1,t) + wx'(i,1,t+1)\}$.
4. Set $x(i',1,t) = 0$, where $i' \neq i^*$, and keep $x(i^*,1,t)$ to be as calculated.

As noted earlier, the WDH algorithm, designed for single user single channel version of problem, allocates usage for the current timeslot keeping into consideration the "projected" allocation for the next timeslot. In the next timeslot, it may or may not use that projected allocation. The situation is even more marked for the multiple user single channel case, since there are m possible combinations that may change the anticipated decision. We interpret this increased likelihood of change to suggest that the next timeslot should have lesser weightage than the current timeslot.

6.2 Multiple User Multiple Channel

For the case of multiple user multiple channel, we propose best-fit heuristic algorithm based on WWDH for multiple user single channel. The essential idea of the algorithm is to iterate over channels from 1 to f, and using WWDH for each channel. All users that are allocated over a particular channel are considered to be unavailable over the next channels. Recall that each user is only allowed to use at most one channel in any given timeslot.

At each timeslot t, we use the following algorithm to get x with the sequence of channels 1 to f.

- For the first channel, i.e., if $j = 1$, we proceed by using WWDH for multiple user single channel.
- For the case of $j > 1$, if $x(i,k,t-1) > 0$ or $x(i,k,t) > 0$ for $1 \le k \le j-1$, $c(i,j,t)$ is forced to be 0. Then apply the algorithm WWDH for multiple user single channel to get $x(i,j,t)$.

7 Numerical Results

In this section, we present the empirical results for the algorithms presented in the preceding sections. Our goal here is to evaluate the algorithms by varying the number of users, channels and the prediction accuracy.

For calculating empirical results, we assume $1,000$ timeslots and channel capacity is randomly chosen from one of the 9 levels. We use the MCS capacity levels: $[8.8, 11.2, 14.8, 17.6, 22.4, 29.6, 44.8, 54.5, 59.2]$ (in kb/s) as outlined in the EGPRS modulation and coding schemes and peak data rates in [1].

For the case of algorithms for imperfect channel estimates, the following procedure is used to calculate the estimated capacity and true capacity values. The prediction accuracy is used to calculate the values $p(a, b)$ for all values of $b \neq a$, where $p(a, b)$ is the probability that actual capacity $c(i, j, t + 1)$ turns out to be equal to b, given the estimated value $c'(i, j, t + 1)$ was equal to a.

1. We assume that for any rate a, the prediction accuracy in capacity estimation for each user, each channel, and each time slot is the same, and is given by $p(a, a)$.
2. We first generate the value a by choosing a random value from the MCS list.
3. From $p(a, a)$, we first calculate $p(a, \bar{a}) = 1 - p(a, a)$.
4. We assume that the $p(a, b)$ is inversely proportional to the "distance" between a and b. Please note that the distance is between the indices in the MCS list. For example, the distance between 11.2 and 22.4 is 3. Using this inverse relationship and the fact that $\sum_{b \in \text{MCS}} p(a, b) = 1$, we can calculate $p(a, b)$ for all values of a and b.

We note that many results for the case of perfect estimation can be subsumed under the results for imperfect estimation, by setting the prediction accuracy probability to 1.

Next we plot throughput for different number of users, number of channels and different values of weight $w > 0$ (an input parameter to the WWDH algorithm), as well as different values of prediction accuracy. The throughput when $w = 0$ is obtained using a simple greedy algorithm without using the estimate $c'(i, j, t + 1)$ (i.e., 0-lookahead). The upper bound U used in all these figures is obtained by taking the sum of upper bound taken over each channel considered separately, as noted in Section 5.4.

In Figure 1, the throughput is plotted against the weight w used by WWDH, for 5 users and 5 channels, and for prediction accuracy varying from 0.5 to 1.0. The plot suggests that the weight of 0.5 maximizes the throughput for all values of accuracy. We observe that even for 1-lookahead with only 50% accuracy, we can still obtain a throughput that exceeds 75% of the upper bound. The results also confirm that the throughput is higher if the prediction accuracy for the channel is higher. This is also consistent for results shown in Figures 2, 3 and 4. Note that for a wide range of w values, the WWDH algorithm performs well, but for two extreme values (i.e., $w = 0$ or $w = 1$), the performance is lower. Again, this is also consistent for results shown in Figures 2, 3 and 4.

In Figures 2 and 3, the throughput is plotted against the weight w used by WWDH, for 5 channels where the number of users is 20 and 40 respectively. The prediction

accuracy varies from 0.5 to 1.0. The plot suggests that the optimal weight tends to decrease when the ratio m/f increases, though the throughput values are quite close for all weight values between 0.1 and 0.7. For $m = 20$, the optimal weight is observed to be 0.3, while for $m = 40$, the weight of 0.2 maximizes the throughput. We observe that for the case of $m = 20$, even with a 1-lookahead with only 50% accuracy, we can still obtain a throughput that exceeds 88% of the upper bound. Similar observations are made for $m = 40$.

Figure 1: Throughput vs. Weight
$m = 5, f = 5$

Figure 2: Throughput vs. Weight
$m = 20, f = 5$

Figure 3: Throughput vs. Weight
$m = 40, f = 5$

Figure 4: Throughput vs. Number of users
$w = 0.5, f = 5$

In Figure 4, we summarize our results. Even in case of single lookahead with only 50% prediction accuracy for 5 users, a throughput of 75% can be achieved by properly setting the weight parameter in WWDH algorithm algorithm. The achieved throughput for single lookahead with 50% prediction accuracy can be as high as 87% if the number of users is increased to 10.

8 Conclusions and Future Work

Channel-aware scheduling and link adaptation will play an important role in achieving the high data rates required by emerging fixed and mobile applications. In this paper, we have considered the problem of maximizing throughput in a multi-carrier wireless network that employs predictive link adaptation. The contributions are two-folds. Firstly, the time-penalty induced by the transmission mode change is incorporated in the scheduling. Secondly, the trade-offs between the channel state prediction quality and the throughput are studied.

We first considered the deterministic model in which the estimate for a future channel state is assumed to be accurate and developed several algorithms (offline and online) with provable bounds. We then extended the model to include the case when the channel estimates are imperfect and developed algorithms based on our deterministic solutions. Our simulation results show that a modest consumption of resources for channel prediction and link adaptation may result in a significant throughput improvement, with only marginal gains through further enhancement of the prediction quality. The results presented in this paper can provide meaningful guidelines in deciding what level of system resource consumption is justified for channel quality estimation and link adaptation.

We have not considered the quality of service requirements for different users in the scheduling problem. A natural extension of this work would be to study the effects of the channel prediction quality on the QoS requirements.

References

1. S. Catreux, V. Erceg, D. Gesbert, and R.W. Heath, "Adaptive modulation and MIMO coding for broadband wireless data networks," *IEEE Communications Magazine*, vol. 40, pp. 108–115, June 2002.
2. V. Tsibonis, L. Georgiadis, and L. Tassiulas, "Exploiting wireless channel state information for throughput maximization," in *Proceedings of IEEE Infocom '03*, April 2003, pp. 301–310.
3. S. Kulkarni and C. Rosenberg, "Opportunistic scheduling policies for wireless systems with short term fairness constraints," in *Proceedings of Globecom 2003*, December 1992, pp. 533–537.
4. X. Liu, E.K.P. Chong, and N. Shroff, "Opportunistic transmission scheduling with resource-sharing constraints in wireless networks," *IEEE JSAC*, vol. 19, no. 10, pp. 2053–2064, Oct. 2001.
5. T. Keller and L. Hanzo, "Adaptive multicarrier modulation: a convenient framework for time-frequency processing in wireless communications," *Proceedings of the IEEE*, vol. 88, pp. 1369–1394, August 1990.
6. A. Arora and H. A. Choi, "Channel aware scheduling in wireless networks," Tech. Rep. 002, The George Washington University, 2006.
7. J. E. Hopcroft and R. M. Karp., "An n 5/2 algorithm for maximum matchings in bipartite graphs," *SIAM Journal on Computing*, vol. 2, no. 4, pp. 225–231, 1973.
8. A. V. Goldberg and R. E. Tarjan, "A new approach to the maximum flow problem," *J. Assoc. Comput. Mach.*, vol. 35, pp. 921–940, 1988.