WiMAX/OFDMA Burst Scheduling Algorithm to Maximize Scheduled Data

Ashraf Nusairat, *Member*, *IEEE*, and Xiang-Yang Li, *Senior Member*, *IEEE*

Abstract—OFDMA resource allocation algorithms manage the distribution and assignment of shared OFDMA resources among the users serviced by the basestation. The OFDMA resource allocation algorithms determine which users to schedule, how to allocate subcarriers to them, and how to determine the appropriate power levels for each user on each subcarrier. In WiMAX, the downlink (DL) TDD OFDMA subframe structure is a rectangular area of N subchannels $\times$ K time slots. Users are assigned rectangular bursts in the downlink subframe. The size of burst, in terms of number of subchannels and number of time slots, varies based on the user's channel quality and data to be transmitted for the assigned user. In this paper, we study the problem of assigning users to bursts in WiMAX TDD OFDMA system with the objective of maximizing downlink system throughput for the Partially Used subcarrier (PUSC) subchannalization permutation mode. Our main contributions in this paper are: 1) we propose different methods to assign bursts to users, 2) we prove that our *Best Channel* burst assignment method achieves throughput within a constant factor of the optimal, 3) through extensive simulations with real system parameters, we study the performance of the *Best Channel* burst assignment method. To the best of our knowledge, we are the first to study the problem of *DL Burst Assignment* in the downlink OFDMA subframe for PUSC subchannalization permutation mode taking user's channel quality into consideration in the assignment process.

Index Terms—WiMAX, OFDMA, wireless scheduling, burst scheduling, throughput maximization

1 Introduction

Mobile users are demanding increase in the services provided by wireless networks providers for day-today and entertainment activities. This demand requires an increase in network throughput to provide similar services provided by fixed networks. Advances in RF technology lead to an improved spectral efficiency which lead to the emerge of different 3G and 4G broadband wireless access networks such as UMTS, HSPA, CDMA2000 EVDO, WiMAX, and LTE with high data rates theoretically reaching up to 70 Mbps for Mobile-WiMAX. Those wireless broadband networks utilize different multiplexing techniques, like W-CDMA for UMTS, MC-CDMA for CDMA2000 EV-DO, and Orthogonal frequency-division multiple access (OFDMA) for WiMAX and LTE. The emerging 3G/4G broadband wireless networks enable wireless operators to offer wide range of more advanced broadband services while achieving greater network capacity.

The air interface in wireless networks is a shared resource, all users in the network content to use this resource; hence, a schedular is needed to resolve contentions. Scheduling-shared air-interface resources in wireless networks been studied widely in the research community [7], [8], [9], [10], [11], [12], [13], [14], [15], [18], [27], [28], [29], [30], [31], [32] and with the development in RF technology continues to be an area for further study. In the TDD OFDMA frame, the shared resources are the *subchannels* in the frequency domain and the *time slots* in the time domain. We will concentrate on the TDD Downlink OFDMA subframe adopted by WiMAX forum and propose algorithm to distribute the shared resources (*subchannels* and *time slots*) between users to maximize the transmitted data.

The TDD downlink OFDMA subframe can be logically viewed as a rectangle of N subchannels (in frequency domain) and K slots (in time domain). In the DL, users are allocated rectangular areas (*bursts*) consisting of n consecutive subchannels by k consecutive time slots in a given OFDMA frame. The problem of assigning the N subchannels by K slots to users with the objective of maximizing the data transmitted over a single downlink subframe is the *DL Burst Assignment* problem. We assume that the users to be scheduled in the frame are selected by a different entity (based on QoS requirements, priority or other criterion) and presented to the scheduler. The scheduler only attempts to allocate downlink bursts to the users to maximize the overall transferred data over the downlink subframe. In this paper, we mathematically formulate the downlink burst assignment as an optimization problem and propose different methods to assign bursts to users with the objective of maximizing throughput. One version of the burst assignment problems is equivalent to the Maximum Independent Set of Rectangles problem [17] which is NP-hard problem; hence, the burst assignment problem is NP-hard.

In the *DL Burst Assignment*, subset of existing users with pending downlink data needs to be scheduled in the downlink subframe with the objective of maximizing the scheduled data. The $N \times K$ subframe needs to be partitioned into bursts of n consecutive subchannels by k consecutive time slots. The size of data scheduled over the burst depends on the channel quality of the user assigned to the burst. In practice, different users could have different channel qualities over the same channel, and the

- A. Nusairat is with Motorola Solutions, Schaumburg, 485 Aberdeen St., Hoffman Estates, IL 60169. E-mail: nusaash@iit.edu.
- X.-Y. Li is with the Department of Computer Science, Illinois Institute of Technology, Room 229C, Stuart Building, 10 West 31st Street, Chicago, IL 60616. E-mail: xli@cs.iit.edu.

Manuscript received 30 Jan. 2011; revised 7 Sept. 2011; accepted 15 Sept. 2011; published online 8 Oct. 2011.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2011-01-0051.
Digital Object Identifier no. 10.1109/TMC.2011.211.

1536-1233/12/$31.00 © 2012 IEEE Published by the IEEE CS, CASS, ComSoc, IES, & SPS

same user may experience different channel qualities over different channels (multiuser diversity). The *DL Burst Assignment* is a challenging multidimensional problem in which the N subchannels by K time slots are to be assigned to subset of M users each with a demand d_m with the following constraints:

1. the allocated burst (n subchannels × k time slots) of each user must be rectangular,
2. the size of data transmitted over the allocated bursts must meet user's demand d_m,
3. allocated slots must be the minimum number of slots required to satisfy user's demand (WiMAX specifications allow data from one or several connections to be carried within a burst; in this paper, we assume that one burst contains data from one connection),
4. a user can have a maximum of one allocation/burst in a frame (WiMAX specifications allow more than one allocation per user; in this paper, we will constraint user's maximum number of allocations to one),
5. the total transmit power of all subchannels does not exceed the basestation's maximum transmission power. Users can be allocated to different subchannels based on the channel quality seen by the user; therefore, multiple bursts of different sizes could exist for a user in the downlink subframe, which adds more complexity to the burst assignment problem.

Recently, in [11], [27], [28], [29], [30], [31], [32], the problem of *DL Burst Assignment* has been studied. In [27], [28], [30], [31], the approaches 1) assume the burst's modulation and coding scheme (MCS) is assigned by a higher entity (i.e., QoS scheduler) and is passed to the burst allocator; the allocator is free to place the data region (burst) anywhere in the downlink subframe without taking the users' subchannel quality into consideration when the region is allocated, 2) assume multiple users are scheduled in a data region, 3) maximize number of allocated slots but not necessarily maximize the transmitted data because the burst's MCS is not the most efficient for all users in the burst. In [11], [32], the proposed burst assignment methods start by making a small allocation for all users, the allocations are initially distanced from each other to allow for expansion. Once initial allocations are done, the methods iterate over the users again and again to increase their rectangular allocation area. In [11], [32], all users are assigned at least one slot (given that number of users is less than $N \times K$) and the user's demand is not necessarily met. Compared with these previous approaches,

1. we consider the subchannel quality as a major factor in allocating a burst to the user which is a major contribution in this paper,
2. we maximize the frame throughput but not necessarily maximize number of allocated slots,
3. we assume one user is scheduled in one burst,
4. we do not guarantee the scheduling of all users in a frame but we guarantee to meet the demand of the scheduled users.

In our approach, users are selected for scheduling based on the amount of data to be transferred in a frame, then the best subchannel(s) with the available required number of slots is allocated to the user. The data scheduled over a burst are modulated using the appropriate MCS which depends on the users' subchannel(s) quality. Our main contributions in this paper are: 1) we propose different methods to assign bursts to users in the downlink subframe for PUSC subchannalization permutation mode taking user's channel quality into consideration (to the best of our knowledge, we are the first to study the problem of burst assignment for downlink OFDMA PUSC mode taking user's channel quality into consideration while assigning slots to users), 2) we prove that our *Best Channel* burst assignment method achieves throughput within a constant factor of the optimal, 3) through extensive simulations with real system parameters, we study the performance of the *Best Channel* burst assignment method.

The remainder of this paper is organized as follows: In Section 2, we give a quick introduction to OFDMA. Section 3 discusses and formulates the *DL Burst Assignment* problem. Our *DL Burst Assignment* methods are presented in Sections 4 and 5. In Section 6, we present and discuss our simulation results. Literature review is presented in Section 7. Section 8 concludes this paper.

2 OFDMA Overview

OFDMA is based on Orthogonal Frequency-Division Multiplexing (OFDM) a transmission scheme that partitions the available bandwidth into N orthogonal narrowband subcarriers [1], [2], [3], [4], [5], [6]. The subcarriers are placed very close to each other, which results in high spectral efficiency. By dividing the bandwidth into N subcarriers, a data stream with high bit rate will be divided into N parallel low bit rate substreams. This results in longer symbol duration for each substream which reduces the influence of multipath propagation and fading and resolves intersymbol interference (ISI). In OFDM, each data substream is mapped to individual data subcarrier and modulated using Phase Shift Keying (PSK) or Quadrature Amplitude Modulation (QAM) based on the target user subcarrier conditions.

2.1 OFDMA Based on OFDM

OFDMA is the multiuser version of OFDM digital modulation scheme and inherits all its advantages and disadvantages. Multiple access is achieved in OFDMA by allocating subsets of subcarriers to individual users. OFDMA employs multiuser diversity by dynamically allocating sets of subcarriers to the users based on the quality of the subcarrier as seen by the user, one subcarrier that is of low quality to one user can be of good quality to another user, and therefore can be allocated to that user.

In OFDMA, subcarriers are divided into groups of subcarriers, each group is named a subchannel, the subchannel is what gets allocated to a user at a given symbol. The subcarriers in a subchannel can be diverse or contiguous. In diversity permutation type, subcarriers are drawn pseudorandomly to form a subchannel. This provides frequency diversity and intercell interference averaging. The diversity permutations include downlink Fully Used subcarrier (FUSC), downlink PUSC and uplink PUSC and additional optional permutations. In this paper, we concentrate on downlink PUSC permutation type since it is mandatory to support in WiMAX systems.

3 WiMAX/OFDMA Burst Assignment Problem

3.1 Problem Description

In OFDMA system, the downlink subframe consists of N subchannels by K slots that can be allocated to users. The minimum allocation unit is one PUSC slot (all through this section, we will use the word "slot" to mean "PUSC slot" unless otherwise specified). When scheduling users, the scheduler assigns bursts to users with an objective of maximizing overall system throughput (other objectives could be considered, but we will concentrate on maximizing throughput).

The *DL Burst Assignment*, is the problem of assigning the N subchannels by K slots to users with the objective of maximizing the data transmitted over a single downlink subframe and is constraint by

1. the minimum unit of burst allocation is one downlink PUSC slot (1 subchannel × 2 times lots),
2. user's allocated bursts must be rectangular,
3. data transmitted over the allocated slots must meet user's demand (data to be transmitted to the user),
4. allocated slots must be the minimum number of slots required to satisfy user's demand (WiMAX specifications allow data from one or several connections can be carried within a burst, we assume that one burst contains data from one connection),
5. a user can have a maximum of one allocation in a frame (WiMAX specifications allow more than one allocation per user; in this paper, we will constraint user's maximum number of allocations to one),
6. based on subchannel quality, a user might not be assigned slots on subchannels with bad quality.

User's subchannel quality controls the number of slots to be allocated to meet the user's demand. A user can be allocated slots on one subchannel or multiple consecutive subchannels, and the quality of the subchannel or *consecutive subchannels* impacts the number of slots. The *consecutive subchannel*'s quality is determined by the quality of the worst subchannel in the set of the *consecutive subchannels*. User's might have bad subchannel quality on some subchannels; the user will not be assigned to those subchannels.

The *DL Burst Assignment*[1] problem is different from other assignment problems like general assignment [19], [20], bin packing [21], [22], [23] and rectangle tiling [24], [25], [26]. In the general assignment problem, multiple agents (subchannels) can't be assigned the same job (user) and jobs (users) can be assigned to any agent (subchannel), while in the burst assignment consecutive slots on consecutive subchannels can be assigned to the same user and users can't be assigned to subchannels with bad signal quality. In the bin packing problem, objects (users) can't be assigned to consecutive bins (subchannels) and can be assigned to a single bin (subchannel) with the objective of minimizing number of bins (subchannels); while in the burst assignment, users (objects) can be assigned to multiple consecutive slots on consecutive subchannels (bins) with the objective of maximizing the scheduled data. In the rectangle tiling problem, rectangles can be placed anywhere in the bounding rectangle, while in the burst assignment, bursts can't be assigned to any subchannel. For burst assignment problem, the shape of the rectangle is not fixed as long as the total data bits supported by this rectangle is larger than the demand, while for rectangle tiling problem, the shapes of the rectangles are fixed already.

TABLE 1
Notations Summary

Notation	Description
d_m	Demand of user m
r_m	Achieved rate of user m
R_m	Minimum rate requirement of user m
N	Total number of subchannels
K	Total number of slots
M	Total number of users
$SINR_{mn}$	SINR measured by user m over subchannel n
b_{mns}	Number of bits/slot that can be transmitted for user m over subchannel n for a given SINR s
P_{mn}	Required transmit power of subchannel n to transmit b_{mns} bits/slot for user m
P_n	Transmit power of subchannel n
c_{mn}	User m to subchannel n assignment indicator
$A_{k_c n_e k_d n_f}$	Rectangle defined by the two points (k_c, n_e) and (k_d, n_f) in the $N \times K$ domain
$A_{m k_c n_e k_d n_f}$	Rectangular allocation to user m over the rectangle $A_{k_c n_e k_d n_f}$
$W_{m k_c n_e k_d n_f}$	Number of bits that can be transmitted for user m over the allocation $A_{m k_c n_e k_d n_f}$
$X_{m k_c n_e k_d n_f}$	Binary variable indicating whether user m is assigned the rectangle $A_{m k_c n_e k_d n_e}$
I_{nk}	Binary variable indicating if slot k on subchannel n is covered by allocation $A_{m k_c n_e k_d n_f}$
U_{n_e}	Set of unique users assigned a time slot on subchannel n_e
S_{mn}	The SINR of user m over subchannel n
P_{mn}	Power required to achieve SINR S_{mn}
P_{n_e}	Transmit power of subchannel n_e
$b_{m n_e n_f}$	Number of bits/slot that can be transmitted for user m over subchannels $\{n_e, \ldots, n_f\}$ based on m's channel conditions (SINR) over $\{n_e, \ldots, n_f\}$

3.2 Problem Formulation

In this section, we formulate the problem of burst assignment to maximize scheduled data in WiMAX/OFDMA-based systems. A summary of notations is shown in Table 1.

Consider a basestation BS_i with maximum transmit power of P_0 serving M users operating at B channel bandwidth. The minimum rate requirement for users are denoted as $\{R_1, R_2, \ldots, R_m\}$. Let $\{r_1, r_2, \ldots, r_m\}$ be the achieved rate for the M users. Let d_m be the demand of each user m representing the size of data to be scheduled. Let N be the total number of available subchannels in the frequency domain. Let K be the number of available downlink PUSC slots in the time domain excluding control and broadcast slots.

We assume that another scheduling process decides the set of users M to be attempted for scheduling in the next frame. This process could decide which user to schedule

[1] *DL Burst Assignment* or *burst assignment* is used exchangeably in the rest of the paper unless otherwise specified.

based on QoS requirements such as delay, latency, and jitter. The *burst assignment* algorithm does not guarantee the scheduling of all M users; in a given frame, the *other scheduling process* presents a set of users to the *burst assignment* algorithm for scheduling with the objective of maximizing the frame throughput. In the case where not all users are scheduled in the current frame, the *other scheduling process* is responsible for adjusting its user selection method in the next frame to prioritize users based on their QoS requirements and avoid starvation of users with small data; the details of the *other scheduling process* are out of scope of this paper and can be a topic for future research.

We assume the basestation has full knowledge of the subchannel conditions for each user; the 802.16e standard provides the mechanism for the user devices to periodically report mean and standard deviation of received signal-to-interference-plus-noise ratio (SINR) and/or received signal strength (RSSI) to the basestation. The reports are sent over Channel Quality Indicator Channel (CQICH) in the control part of the UL subframe. 802.16e specifications allow for high flexibility in the periodicity of the SINR and/or RSSI measurements, the basestation controls the periodicity of the reports and therefore can implement an efficient algorithm to request SINR/RSSI from the end-user devices. If the periodicity of the SINR/RSSI reports introduces high overhead or the basestation is running out of CQICH resources to allocate to the end-user device to report SINR/RSSI, the basestation and the end-user device can implement proprietary procedures to piggyback the SINR/RSSI reports to the header of the MAC frame and not utilize the CQICH channel. It's worth noting that as part of the current subchannel quality measurement procedure defined by 802.16e specification, the user-device is measuring the qualities over the set of subcarriers and then it averages those measurements over all subchannels; therefore, a change will be required to average the measurements over each subchannel and report them per subchannel. It is worth noting that Mobile WiMAX channel coherence time (for low frequencies of 2.5 and 3.5 GHz) even at high speeds of 100 km/hr is larger than 3 msec [33], which is more than 10 times the symbol duration. Therefore, the two OFDMA symbols constructing the minimum DL PUSC slot can be treated as highly correlated; for speeds higher than 100 km/hr, better channel estimation algorithms may be needed.

Let $SINR_{mn}$ be the SINR measured by user m over subchannel n and let b_{mns} be the number of bits/slot that can be transmitted for user m over subchannel n for a given SINR s, $b_{mns} = \mathcal{F}(s)$, where $\mathcal{F}(.)$ is monotonically increasing function with respect to s.

In WiMAX, only one user can be assigned to a subchannel *at a given time slot* and one user could be assigned multiple continuous subchannels in a given time slot.

Let P_{mn} be the required transmit power of subchannel n to transmit b_{mns} bits/slot for user m. Let P_n be the transmit power of subchannel n, P_n will be the maximum transmit power required by all users allocated to subchannel n over all K slots in the scheduled frame. Let c_{mn} be the user-to-subchannel assignment indicator, for the mth user: $c_{mn} = 1$ if the nth subchannel is assigned to the mth user; otherwise, $c_{mn} = 0$. The required transmit power of subchannel n can then be expressed as $P_n = max\{P_{mn}\}, \forall m : c_{mn} = 1$.

The total transmit power of all subchannels must not exceed P_0, i.e., $\sum_{n=1}^{N} P_n \leq P_0$. In downlink, the power is limited by P_0 (maximum transmit power of basestation) and each subchannel n is assigned power p_n based on user m with the worst $SINR_{mn}$. The total power assigned to all subchannels must be less than p_0. Typically, users assigned to a subchannel have different target SINR based on user's RF conditions (RSSI, path loss, fading, interference), we assume that the basestation has knowledge of the channel gain and can estimate the target SINR of user m over subchannel n. For a certain target $SINR_{mn}$, the basestation knows the required transmit power at subchannel n so that user m gets the targeted $SINR_{mn}$. Different users on the same subchannel have different SINR and require different transmit powers, but a single subchannel can be allocated one transmit power value during a scheduled frame, and that value will be the maximum required to achieve the targeted SINR of the assigned users, with the condition that the total power assigned to all subchannels is less than P_0.

Let $A_{k_c n_e k_d n_f}$ be a rectangle defined by the two points (k_c, n_e) and (k_d, n_f) in the $N \times K$ domain representing the rectangular area over the contiguous subchannels $(n_e, \ldots, n_f)$ over the contiguous time slots $(k_c, \ldots, k_d)$, we have $k_c \leq k_d \leq K$, $n_e \leq n_f \leq N$. Let $A_{mk_c n_e k_d n_f}$ be defined as a rectangular allocation to user m over the rectangle $A_{k_c n_e k_d n_f}$. Each allocation $A_{mk_c n_e k_d n_f}$ has an associated weight of $W_{mk_c n_e k_d n_f}$ representing the number of bits that can be transmitted for user m over the allocation $A_{mk_c n_e k_d n_f}$. $W_{mk_c n_e k_d n_f}$ varies based on the b_{mns} of user m over all subchannels $n \in (n_e, \ldots, n_f)$, recall that b_{mns} is the achieved bits/slot for user m over subchannel n and SINR s.

$A_{mk_c n_e k_d n_f}$ can be considered a target allocation for user m *if and only if* the weight $W_{mk_c n_e k_d n_f} \geq d_m$ and any reduction of the rectangle $A_{mk_c n_e k_d n_f}$ in any direction (height or width) makes $W'_{mk_c n_e k_d n_f} < d_m$, where $W'_{mk_c n_e k_d n_f}$ is the weight of the reduced rectangle. In other words, we only consider the smallest rectangles that can fit d_m. Based on d_m, multiple $A_{mk_c n_e k_d n_f}$ allocations could exist for user m in the $N \times K$ domain, the scheduling algorithm will guarantee the assignment of only one of them to user m. Each rectangle $A_{k_c n_e k_d n_f}$ can only be assigned to one user in the scheduled frame, i.e., two allocations can't overlap, the scheduling algorithm will guarantee the assignment of only one user to a rectangle.

Each rectangle defined by the two points $(k_c, n_e), (k_d, n_f)$ in the $N \times K$ domain can only be assigned to one user in the scheduled frame, i.e., two allocations can't overlap. Let $X_{mk_c n_e k_d n_f}$ be a binary variable indicating whether user m is assigned the rectangle defined by the two points $(k_c, n_e), (k_d, n_f)$ (rectangle $A_{mk_c n_e k_d n_e}$)

$$X_{mk_c n_e k_d n_f} = \begin{cases} 1 & \text{if rectangle } \{(k_c, n_e), (k_d, n_f)\} \\ & \text{is assigned to user } m \\ 0 & \text{otherwise.} \end{cases}$$

A user can be allocated one rectangle ($\sum_m X_{mk_cn_ek_dn_f} \leq 1$) and allocated rectangles can't overlap ($\bigcap A_{mk_cn_ek_dn_f} = \Phi, \forall X_{mk_cn_ek_dn_f} = 1$).

Let I_{nk} be a binary variable indicating if slot k on subchannel n is covered by allocation $A_{mk_cn_ek_dn_f}$. Slot k on subchannel n is covered by allocation $A_{mk_cn_ek_dn_f}$ if $X_{mk_cn_ek_dn_f} = 1$ and $n_e \leq n \leq n_f$ and $k_c \leq k \leq k_d$

$$I_{nk} = \begin{cases} 1 & \begin{array}{l}\text{if rectangle } \{(k_c, n_e), (k_d, n_f)\} \\ \text{is assigned to a user and} \\ n_e \leq n \leq n_f \text{ and } k_c \leq k \leq k_d\end{array} \\ 0 & \text{otherwise.} \end{cases}$$

For the allocated rectangle not to overlap, I_{nk} should be less than 1; i.e., only one allocation is covering slot k on subchannel n. The condition for allocated rectangles not to overlap becomes: $I_{nk} \in \{0, 1\}, \forall n, k$.

Let U_{n_e} be the set of unique users assigned a time slot on subchannel n_e, which is defined as

$$U_{n_e} = \{m \mid X_{mk_cn_ek_dn_e} = 1, \forall m \in M, k_c \leq k_d \leq K\}.$$

Let S_{mn} be the SINR of user m over subchannel n and let the power required to achieve SINR S_{mn} be P_{mn}. Then, the subchannel power can then be defined as

$$P_{n_e} = MAX(P_{mn_e}), \quad \forall m \in U_{n_e}$$

and the total power is constrained by $\sum_{n_e} P_{n_e} \leq P_0$. The power assignment per subcarrier adds a third dimension to the scheduling problem. We will assume that the basestation transmit power P_0 is uniformly distributed among all subcarriers.

For allocation $A_{mk_cn_ek_dn_f}$ that expands multiple subchannels ($n_e < n_f$), there could exist multiple SINRs for a user. The basestation has to select *one* SINR as the target for the allocation in order to decide on the MCS to be used for the entire allocation. Recall that MCS (b_{mn}) is a function of SINR, $b_{mn} = \mathcal{F}(SINR_{mn})$.

Let $b_{mn_en_f}$ be the number of bits/slot that can be transmitted for user m over subchannels $\{n_e, \ldots, n_f\}$ based on m's channel conditions (SINR) over $\{n_e, \ldots, n_f\}$. Since there is only one MCS (bits/slot) applied for one user of allocation $A_{mk_cn_ek_dn_f}$, $b_{mn_en_f}$ can be viewed as the minimum MCS (bits/slot) that could be achieved over the individual subchannels $\{n_e, \ldots, n_f\}$. Hence, $b_{mn_en_f}$ can be expressed as

$$b_{mn_en_f} = MIN(b_{mn_e}, b_{mn_{e+1}}, \ldots, b_{mn_f}).$$

Now, $W_{mk_cn_ek_dn_f}$, which represents the number of bits that can be transmitted for user m over the allocated subchannels ($n_e \ldots n_f$), can be defined as

$$W_{mk_cn_ek_dn_f} = (k_d - k_c + 1) \cdot (n_f - n_e + 1) \cdot b_{mn_en_f}.$$

For a given allocation $A_{k_cn_ek_dn_f}$, there could be different weights $W_{mk_cn_ek_dn_f}$ when different users are assigned to it due to the difference in users' SINR over the subchannels $\{n_e, \ldots, n_f\}$.

The burst assignment is a maximization problem with the following objective:

$$Maximize \sum_m r_m$$

with the following constraints:

$$\begin{aligned}
&(1.1) \sum_{k_c}\sum_{k_d}\sum_{n_e}\sum_{n_f} X_{mk_cn_ek_dn_f} \leq 1 \\
&(1.2)\ (k_d - k_c + 1) \cdot (n_f - n_e + 1) \cdot b_{m,n_e,n_f} = W_{mk_cn_ek_dn_f} \\
&(1.3) \sum_{k_c}\sum_{k_d}\sum_{n_e}\sum_{n_f} X_{mk_cn_ek_dn_f} \cdot W_{mk_cn_ek_dn_f} = r_m \\
&(1.4)\ r_m \geq d_m \\
&(1.5)(k_d - k_c + 1) \cdot (n_f - n_e + 1) \cdot b_{m,n_e,n_f} \geq d_m \\
&(1.6)(k_d - k_c) \cdot (n_f - n_e + 1) \cdot b_{m,n_e,n_f} < d_m \\
&(1.7)(k_d - k_c + 1) \cdot (n_{f-1} - n_e) \cdot b_{m,n_e,n_{f-1}} < d_m \\
&(1.8)(k_d - k_c + 1) \cdot (n_f - n_{e-1}) \cdot b_{m,n_{e-1},n_f} < d_m, n_e > 1 \\
&(1.9) \sum_{m}\sum_{k_c}\sum_{k_d}\sum_{n_e}\sum_{n_f} X_{mk_cn_ek_dn_f} \leq 1 \\
&(1.10) I_{nk} \leq 1, \forall n \in [1, N], k \in [1, K] \\
&(1.11) \sum_{n_e} P_{n_e} \leq P_0, \\
&where: \\
&k_c \in [1, K], k_d \in [k_c, K], n_e \in [1, N], n_f \in [n_e, N].
\end{aligned} \tag{1}$$

Constraint (1.1) guarantees that an allocation is assigned to only one user. Constraints (1.2)-(1.4) guarantee that the allocated area (burst) meets the user's demand d_m. Constraints (1.5)-(1.8) guarantee that the allocated area is the smallest area that is equal to or greater than user's demand d_m. Constraint (1.9) guarantees at maximum one allocation is assigned to user m. Constraint (1.10) guarantees that the final allocated bursts do not overlap. Constraint (1.11) guarantees that the power allocated to the subchannels does not exceed the basestation's maximum transmit power.

Now, we can model the burst assignment problem as a conflict graph $G = (V, E)$ where V is the set of nodes v representing the conflicting/overlapping target allocations (bursts) $A_{mk_cn_ek_dn_f}$, and E is the set of edges connecting conflicting/overlapping bursts. A node $v \in V$ *iff* v overlaps with at least one node $u \in V$ where $u \neq v$. An edge (uv) exists between nodes u and v *iff* the bursts represented by u and v conflict/overlap. Each node v has a weight w equivalent to the rate of the target allocation represented by v. In this model, we assume nonconflicting allocations are not included in G since they can be assigned without conflict. The size of graph G is polynomial of N and K, in the worst case there can be up to $\frac{N(N+1)}{2} \times \frac{K(K+1)}{2} \times M$ conflicting nodes in G.

It is worth noting that more than one independent conflict graph could exist in the $N \times K$ area. If more than one conflict graph is created by the target allocations $A_{mk_cn_ek_dn_f}$, then the scheduling problem can be solved by independently solving the scheduling problem for each conflicting graph.

Finding the burst assignment that maximizes the scheduled data is equivalent to finding the maximum weighted independent set (MWIS) of the conflict graph G. The MWIS problem is NP-hard; therefore, our burst assignment is NP-hard. In the following sections, we propose different practical methods to assign bursts that maximize the scheduled data.

4 Burst Size Method

In this method, we assume all users have the same channel quality over all subchannels and therefore have the same SINR and MCS over all subchannels. We will then extend this method to deal with a more general case.

Let b_{all} denote the bit/slot that can be achieved by all users over any subchannel. In this case, a given allocation $A_{k_c n_e k_d n_f}$ has one weight given by: $W_{mk_c n_e k_d n_f} = (k_d - k_c) \cdot (n_f - n_e) \cdot b_{all}$ which is independent of the user assigned to the allocation and represents the area of allocation $A_{k_c n_e k_d n_f}$ (Burst Size).

The burst assignment problem is now the problem of rectangle tiling to maximize the overall tiled area. The assignment problem is NP-hard because the subset sum problem is NP-hard. Assume for the burst assignment problem, $\mathrm{N} = 1$. Then, this special burst assignment problem becomes a subset sum problem: we need to pick a set of users whose total data rates is maximized and is still less than $K \times b_{all}$. Hence, burst assignment is NP-hard.

We propose a method that schedules users based on their demand as follows:

1. For each user, m calculate the number of slots required to satisfy d_m, denote this as S_m: $S_m = \lceil \frac{d_m}{b_{all}} \rceil$.
2. Sort S_m in descending order.
3. Divide S_m into two sets:
    a. Set of users that require more than K slots ($S_m \geq K$), denote this set as S_{large}.
    b. Set of users that require less than K slots ($S_m < K$), denote this set as S_{small}.
4. Start assigning the users to slots as follows:
    a. Start assigning users from S_{large} in order. Assign each user to $\lceil \frac{S_m}{K} \rceil$ consecutive subchannels.
    b. Start assigning users from S_{small} in order using the first-fit decreasing (FFD) bin-packing method [21].

The *burst size* method is illustrated in Algorithm 1.

Algorithm 1. Burst Size Allocation Method
Input: Set of M users each with demand d_m, achieved bits/Slot of all users b_{all}, N subchannels, K slots
Output: Set of scheduled users

```
$S = Empty$ {Set of number of slots required to satisfy users' demand}
$S_{large} = Empty$ {Set of users that require more than $K$ slots}
$S_{small} = Empty$ {Set of users that require less than $K$ slots}
$M \leftarrow Sort(M)$ {Sort $M$ in descending order of demand $d_m$}
for $i = 1$ to $M$ do
    $S_i = \lceil \frac{d_i}{b_{all}} \rceil$
end for
for $i = 1$ to $M$ do
    if $S_i \geq K$ then
        $S_{large} \leftarrow S_i$
    else
        $S_{small} \leftarrow S_i$
    end if
end for
repeat
    $NumSlots = S_{large}\{1\}$ {The 1st element in the set, max # of required slots}
    $NumSubchan = \lceil \frac{NumSlots}{K} \rceil$
    if $NumSubchan$ Consecutive Subchannels Available then
        Assign user to the 1st $NumSubchan$ available consecutive subcannels
    end if
    $S_{large} = S_{large} \setminus S_{large}\{1\}$
until $S_{large}$ is empty
if Empty Subchannels Available then
    $bin\_packing\_FFD(S_{small}, EmptySubchannels)$
end if
```

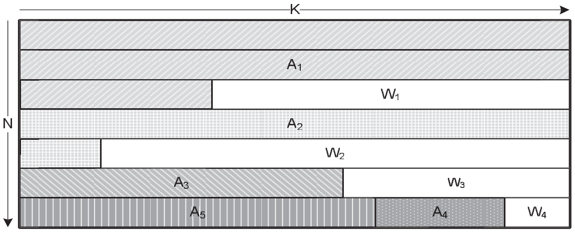

Fig. 1. Outcome example of burst size method.

In this method, number of slots required to satisfy d_m depends on the assumed achieved bits/slot b_{all}. Recall that the achieved bits/slot is upper bound by the supported modulation and coding scheme and it depends on the channel quality. In WiMAX, the best MCS is 64QAM5/6 (480 bits per downlink PUSC slot with MIMO-B) and defines the upper bound of achieved bits/slot (denote this upper bound by b_{max}), the worst MCS is QPSK (96 bits per downlink PUSC slot with MIMO-B) and defines the lower bound of achieved bits/slot (denote this lower bound by b_{min}). Hence, b_{all} is lower bound by b_{min} and upper bound by b_{max}.

In this method, if the assigned user's demand d_m requires more than one subchannel to be satisfied, there could be wasted slots in the last subchannel allocated to user m, denote this wasted slots as S_{w_m}. The wasted slots per assigned user can be calculated as

$$S_{w_m} = K \left\lceil \frac{S_m}{K} \right\rceil - \left\lceil \frac{d_m}{b_{all}} \right\rceil$$

and $0 \leq S_{w_m} \leq K - 1$. Fig. 1 shows an example of final allocation produced by Algorithm 1, the wasted slots in W_1 and W_2 can't be allocated to other users because allocations A_1 and A_2 must be rectangular in shape.

It's worth noting that the time complexity of Algorithm 1 is $O(MN)$.

Lemma 1. *The wasted slots of Algorithm 1 is $< \frac{1}{2}$ of the total number of slots in the $N \times K$ frame.*

Proof. If Algorithm 1 assigns all M users in the frame then if there are unassigned slots, those will not be considered wasted since there are no more users to assign to those empty slots.

Now, let's consider the two sets of users S_{large} and S_{small} when there are users left unassigned by the algorithm (not enough consecutive slots to assign to those users):

S_{large}. Recall that S_{large} is the sorted set of users with demands that require more than K slots (requires more than one subchannel in the frequency domain). The number of wasted slots for each user in this set is at maximum $K-1$ and is located in the last consecutive subchannel (W_2 in Fig. 1). The ratio of wasted slots to total number of slots in the allocated subchannels for a single user is at its maximum when the user is assigned two subchannels and only one slot is used from the second subchannel. In this case, the maximum ratio

$$\frac{K-1}{2K} = \frac{1}{2} - \frac{1}{2K} < \frac{1}{2}.$$

Therefore, the wasted slots in the subchannels allocated to the users in the S_{large} set is $< \frac{1}{2}$ of the total number of slots in the allocated subchannels.

S_{small}. Recall that S_{small} is the sorted set of users with demands that require less than K slots and the set of subchannels to be assigned to those users are the set of empty subchannels (all subchannels not assigned to users of S_{large}). For this set of users, the algorithm uses an efficient bin packing method to assign the users to subchannels. In this case, a user in this set will not be assigned to a single subchannel if there are not enough slots to meet its demand. The user's in this set either require more than $\frac{K}{2}$ slots or less than $\frac{K}{2}$ slots:

1. If a user requires *more* than $\frac{K}{2}$ slots and there are no subchannels available with the required number of slots, the user will not be assigned. In this case, all subchannels are *at least half full*, and the wasted slots in each subchannel are at maximum $\frac{1}{2}K-1$ (W_3 in Fig. 1).
2. If a user requires *less* than $\frac{K}{2}$ slots and there are no subchannels available with the required number of slots, the user will not be assigned. In this case, all subchannels are *more than half full*, and the wasted slots in each subchannel are at maximum $\frac{1}{2}K-1$ (W_4 in Fig. 1).

In this case, the maximum ratio of wasted to total number of slots in a subchannel

$$\frac{\frac{1}{2}K-1}{K} = \frac{1}{2} - \frac{1}{K} < \frac{1}{2}.$$

Therefore, the wasted slots in the subchannels allocated to the users in the S_{small} set is $< \frac{1}{2}$ of the total number of slots in the allocated subchannels.

Therefore, at the end of Algorithm 1 when all users are not assigned, the maximum wasted slots is $< \frac{1}{2}$ of total number of slots in the frame $N \times K$. □

Lemma 2. *Algorithm 1 achieves throughput* $\geq \frac{1}{2}\frac{b_{all}}{b_{max}}$ *of the optimal.*

Proof. From Lemma 1, we have the wasted slots at the end of the algorithm is $< \frac{1}{2}$ of total number of slots in the frame $N \times K$. Hence, the useful slots are $\geq \frac{1}{2}$ of total number of slots in the frame $N \times K$. The throughput of the algorithm is $\geq \frac{1}{2}NKb_{all}$, while the maximum throughput possible is NKb_{max}. Hence, the algorithm achieves a throughput $\geq \frac{1}{2}\frac{b_{all}}{b_{max}}$ of the optimal. □

5 Best Channel Method

In this section, we study a more general case where user's have different channel quality on different subchannels. Some users might have a good channel quality on subchannel n while others might have low or bad channel quality over the same subchannel. In this method, we will take advantage of this multiuser diversity and attempt assigning each user to slots on the subchannel (or set of subchannels) with the best quality.

We assume the BS has up to date information about the channel quality from all users. In this method, the BS maintains $\mathcal{B}$ sets of subchannels that the user can be assigned to, each set in $\mathcal{B}$ represents an MCS that the user can achieve if assigned to any of the subchannels in the set. The number of $\mathcal{B}$ sets is the total number of support MCS by the BS. Each MCS set in $\mathcal{B}$ includes the subchannels and the combination of consecutive subchannels with channel quality that allows the user to be assigned to it, for example, user m set 2 (for example, 64QAM3/4 MCS) may contain subchannels {(2),(4),(5), (4,5)} if the quality of subchannels 2, 4, and 5 allows the BS to assign the user 64QAM3/4 MCS.

Each set in $\mathcal{B}$ is sorted based on the number of slots needed to meet the user's demand. In a given set, this means sorting the set based on the number of consecutive subchannels that can be assigned to the user since the MCS is the same in each set, set 2 (64QAM3/4) in the previous example will be sorted as {(4,5), (1), (4), (5)}. If the user's demand d_m requires only one slot on a single subchannel in a set, then the set will only include single subchannels and will not include subsets of consecutive subchannels, in the previous example, if 1 slot is needed on set 2 (64QAM3/4) then the set will be {(1), (4), (5)}.

In this method, users are schedule to the first subchannel or subchannels from the $\mathcal{B}$ sets that has enough slots to meet the user's demand. If there are no available slots that meets the user's demand the user will be ignored and not scheduled. The *Best Channel* scheduling method is formally illustrated in Algorithm 2.

Algorithm 2. Best Channel Allocation Method
Input: N subchannels, K slots, set of M users each with demand d_m and MCS b_{mn} for each subchannel
Output: Set of scheduled users

```
$M \leftarrow Sort(M)$ {Sort $M$ in descending order of demand $d_m$}
for $m = 1$ to $M$ do
    for $n = 1$ to $N$ do
        Assign $n$ to the appropriate $\mathcal{B}[]$ set based on MCS ($b_{mn}$)
    end for
    for $i = 1$ to $MaxSupportedMCS$ do
        Update $\mathcal{B}[i]$ with consecutive subchannels if any
            and only include consecutive subchannels if $d_m$
            requires more than 1 slot on $\mathcal{B}[i]$
    end for
end for
```

```
for $m = 1$ to $M$ do
    for $i = 1$ to $MaxSupportedMCS$ do
        $scheduleUser(m, d_m, i)$ {try to schedule user $m$ on the 1st (set of) subchannels in $\mathcal{B}[i]$ with available slots that meets $d_m$}
        if User $m$ is $Scheduled$ then
            Update number of available slots of each subchannel user $m$ is scheduled on
            Exit For
        end if
    end for
end for
```

Lemma 3. *Users scheduled by Algorithm 1* $\subseteq$ *of the users scheduled by Algorithm 2.*

Proof. Both algorithms sort users in decreasing order of demand d_m and start scheduling users in order. We need to prove that each user scheduled by Algorithm 1 is also scheduled by Algorithm 2. Recall that Algorithm 1 assumes a fixed MCS b_{all} assigned to all users and users are scheduled based on the number of slots (area needed to meet the demand d_m), while Algorithm 2 uses the actual MCS (based on channel quality) to schedule users. Also, recall that b_{all} is the lowest MCS of all users that can be achieved at any subchannel. Algorithm 2 may use less slots because the actual MCS is used which is greater than b_{all}.

In Algorithm 2, user m is assigned to the best channel or set of consecutive subchannels that have slots available to satisfy d_m. User m will be assigned to either one subchannel n if $d_m \leq b_{mn}K$ or to y consecutive subchannels $(n, \ldots, n+y-1)$ if $b_{mn}K < d_m \leq b_{my}K$ where $y = \frac{d_m}{Kb_{my}}$ and b_{my} is the MCS of user m over the y consecutive subchannels. Since b_{mn} and b_{my} are at least b_{all}, the required slots for user m by Algorithm 2 are at most the same number of slots used by Algorithm 1. If Algorithm 2 assigns the same subchannels to user m as Algorithm 1 this indicates that the user was allocated to subchannels in B with the lowest MCS b_{all}, for the first user this means either the best MCS is b_{all} or there are not available slots at the subchannels in the $\mathcal{B}$ sets with MCS higher than b_{all}.

If the first user was scheduled by Algorithm 1 and its best MCS is $> b_{all}$ then the user will be assigned to a different subchannel and will use less slots. If the second user was scheduled by Algorithm 1 and its best MCS is $> b_{all}$ then the user will be assigned to a different subchannel and will use less slots, but that subchannel may not have the required slots (the first user is scheduled on it and not enough slots are available), then the second user will be scheduled on a different subchannel with MCS better than b_{all} if one is available and have available slots, but if none is available the user will be scheduled on a subchannel with the least MCS (b_{all}) and this subchannel is guaranteed to exist because the user was scheduled by Algorithm 1 on a subchannel with MCS = b_{all}. The same logic applies to all users who where scheduled by Algorithm 1.

Therefore, Algorithm 2 schedules all users who are scheduled by Algorithm 1 and at maximum uses the same number of slots used by Algorithm 1. □

TABLE 2
SINR to MCS Mapping

SINR	MCS	Bits/PUSC Slot
< 6	N/A	N/A
$[6-10[$	QPSK1/2	96
$[10-13[$	QPSK3/4	144
$[13-16[$	16QAM1/2	192
$[16-19[$	16QAM3/4	288
$[19-22[$	16QAM5/6	320
$[24-26[$	64QAM2/3	384
$[26-28[$	64QAM3/4	432
> 28	64QAM5/6	480

Lemma 4. *Algorithm 2 at least achieves throughput* $\geq \frac{1}{2}\frac{b_{all}}{b_{max}}$ *of the optimal.*

Proof. This follows from Lemma 3 which proved that every user scheduled by Algorithm 1 is also scheduled by Algorithm 2. Algorithm 1 achieves a throughput $\geq \frac{1}{2}\frac{b_{all}}{b_{max}}$ of the optimal. Hence, Algorithm 2 at least achieves a throughput $\geq \frac{1}{2}\frac{b_{all}}{b_{max}}$ of the optimal. □

It's worth noting that the time complexity of Algorithm 2 is $O(MN)$.

6 Performance Evaluation

In this section, we present our simulation methodology and environment along with our simulation results.

6.1 Simulation Methodology

We compared the *Sorted Best Channel* method, where users are sorted in descending order of data to be transferred, with a *Round Robin Best Channel* approach where the users are attempted to be scheduled in order.

In the *Sorted Best Channel* method, the users are sorted in descending order of data to be transferred, subchannels are selected based on the best subchannel method as described in Algorithm 2. In *Round Robin Best Channel*, users are not sorted but are attempted to be scheduled in order, the subchannels are selected based on the best subchannel method described in Algorithm 2 excluding the first step of sorting the users.

In both methods, at the start of simulation, we generate random[2] subchannel conditions for all users over all subchannels. We vary the subchannel conditions randomly from frame to frame as the simulation progresses, this represents changes in subchannel quality which changes the MCS assigned to the user's burst. We vary the subchannel conditions for random set of users each frame. The subchannel quality is then mapped to an MCS according to the mapping shown in Table 2.

At the start of a simulation, we randomly generate a large file between 1 and 2 MB for each user to transmit over the duration of one simulation. Each frame a random chunk of the file (between [50-1,000] bytes) is attempted to be scheduled. When a user is not scheduled in a frame, the same chunk is attempted for transmission in the next frame(s). If a user is scheduled in a frame and the file is not fully transferred, a new random chunk will be attempted for scheduling in the next frame(s).

[2] In all our simulations, we used uniform random distribution model.

TABLE 3
Simulation Configuration

Channel Bandwidth	5 MHz		10 MHz	
TDD Ratio	75/25	50/50	75/25	50/50
DL Unicast PUSC Slots	14	9	14	9
DL Subchannels	15		30	

In the *Sorted Best Channel* method, when a user is scheduled in the current frame and there is queued data, the user will be considered for scheduling in the next frame with all or subset of the queued data. In the next frame, the users will be sorted based on the data to be transferred; therefore, the users scheduled in the previous frame could be scheduled again in this frame and possibly in next frame(s). In *Round Robin Best Channel*, when a user is scheduled and there is queued data, the user will be pushed at the end of the users queue to be scheduled in next frame(s) based on its order in the queue.

In the *Sorted Best Channel* method when a user is not scheduled in the current frame, it will be considered for scheduling in next frame(s). In *Round Robin Best Channel*, if a user can't be scheduled in the current frame, the user will be skipped and will be considered for scheduling in the next frame before any other user who is scheduled in the current frame.

In both methods, after the users are scheduled using the best channel, if there are empty slots in the frame, then we attempt to fill them with nonscheduled users who maximize the overall transferred data in the frame.

In our simulations, we used full buffer model and error free transmission over the air (no HARQ burst retransmission). When retransmissions are required due to channel errors, the retransmission mechanism imposes certain constraints on the scheduler. If the retransmission mechanism is based on HARQ mode that requires retransmissions to be in the same burst location and same MCS as the original (first) transmission, then this puts more requirements on the *burst assignment* that we did not consider in our algorithm. In the case where retransmissions do not need to be in the same burst location and the same MCS, we assume the *other scheduling process* that selects users to be scheduled is responsible for retransmissions; from the *burst assignment* algorithm perspective, those retransmissions are new requests since the algorithm is stateless.

6.2 Simulation Environment

In our performance study, we simulated four different WiMAX system configurations based on channel bandwidth and downlink/uplink TDD ratio as listed in Table 3. For each subchannel, we used two downlink/uplink TDD ratios, one representing a small number of downlink slots (50/50 TDD) and the other representing a large number of downlink slots (75/25 TDD).

In our simulation, we assumed that the available downlink slots for user data (downlink Unicast PUSC slots) do not include the MAP and broadcast slots. The total number of time slots available in the WiMAX 5 msec frame is 48 slots, the first time slot is for preamble, another time slot is reserved for TTG, we assumed that MAP and broadcast consume six time slots. The number of time slots in the downlink subframe depends on the DL/UL TDD ratio of the system. In WiMAX, every two time slots over a subchannel in the downlink subframe constitute one downlink PUSC slot, Table 3 shows the number of DL PUSC slots and number of subchannels per channel bandwidth and downlink/uplink TDD ratio.

Our simulation configuration was as follows:

- 100 users were simulated each with a data of size [1-2] MB to be transmitted.
- We ran 100 simulations for each of the four different WiMAX system configurations listed in Table 3.
- Each simulation was 100 frames (0.5 second) long.
- Users' subchannel quality was randomly generated each frame for random set of users.
- In each simulation, we ran the *Sorted Best Channel*and *Round Robin Best Channel* methods and collected the following metrics per frame:
  - Size of data scheduled before filling empty slots.
  - Size of data scheduled after filling empty slots.
  - Number of scheduled users before filling empty slots.
  - Number of scheduled users after filling empty slots.
  - Number of wasted (unallocated) slots.
  - MCS Distribution.
  - Burst Overallocation (The difference between the burst capacity and the actual data scheduled in the burst).

6.3 Simulation Results

Due to the limited space, we will discuss the simulation results of two configurations. The configuration with the smallest number of downlink PUSC slots, 5 MHz, 50/50 TDD ratio (15 subchannels × 9 downlink PUSC slots), we will refer to this configuration as the *"Small Config."* In addition, the configuration with the maximum number of downlink PUSC slots, 10 MHz, 75/25 TDD ratio (30 subchannels × 14 downlink PUSC slots), we will refer to this configuration as the *"Large Config."* We averaged the metrics we collected for all the simulations of the two configurations. We also collected and averaged the metrics before the empty slots were filled (after the methods finish execution some slots might not be assigned) and after the empty slots are filled (the final schedule in the frame).

In the following sections, we discuss the results obtained for the *"Small Config"* and *"Large Config"* simulations.

6.3.1 Data Transferred

Our *Best Channel* algorithm attempts to maximize the data transferred per frame, to measure how the algorithm performs we calculated the average data transferred per frame for both the *"Small Config"* and the *"Large Config"*; Figs. 2 and 3 show the average data transferred per frame over the 100 simulations.

Both figures show that *Sorted Best Channel* achieves better average data transfer per frame ***before filling empty slots*** compared to the *Round Robin Best Channel*. This is due to the fact that *Sorted Best Channel* sorts, in descending order, users based on the size of the data to be transferred in the

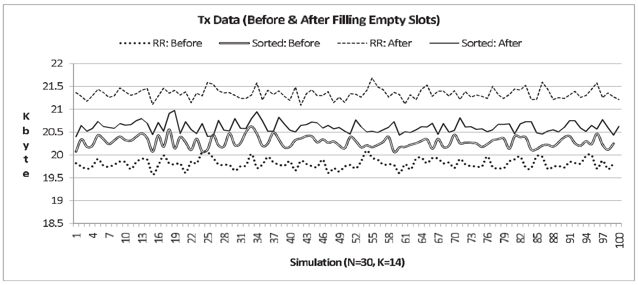

Fig. 2. Average KB transmitted per frame ($\mathrm{N} = 30, \mathrm{K} = 14$).

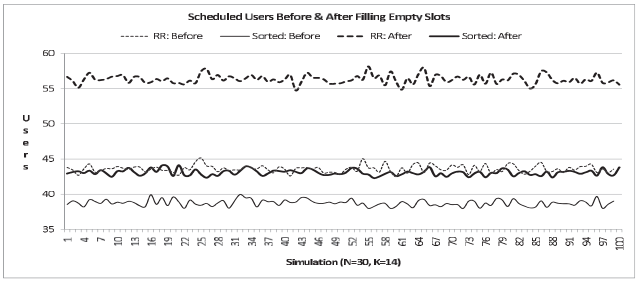

Fig. 4. Average scheduled users per frame ($\mathrm{N} = 30, \mathrm{K} = 14$).

frame; hence, the users with largest data are attempted to be scheduled first which increases the transferred data if they are successfully scheduled. Also, users with large data to be transferred in a frame have higher chance of being scheduled when sorted compared to the round robin users' scheduling.

After filling the empty slots with users' that maximize the overall frame transferred data, Fig. 2 shows that the *Round Robin Best Channel* achieves higher data than *Sorted Best Channel* for *"Large Config"* ($N = 30, K = 14$). This is because the number of empty slots after the completion of the *Round Robin Best Channel* scheduling of users is more than that of the *Sorted Best Channel* (almost six times) as shown in Fig. 6. The larger the number of empty slots, the more the probability of finding users with good MCS to schedule on those empty slots which increases the scheduled data. The *probability is more* because we have simulated full buffer configuration where at each frame there are large number of users with data to transfer. It is important to note that in full buffer configuration the round robin method should not end before filling all slots, but in our simulation we have the restriction that both methods will not schedule a user if there are not enough slots that form a rectangular area and fits the minimum user's data to be transferred in that frame, i.e., *no data fragmentation is allowed when both* Sorted Best Channel *and* Round Robin Best Channel *are scheduling users*. But when we start filling the empty slots with users to maximize the frame's throughput, we allow scheduling users with data less than 50 bytes which is the minimum scheduled data in our simulation.

For the *"Small Config"* ($N = 15, K = 9$), Fig. 3 shows that the *Sorted Best Channel* achieves higher data than *Round Robin Best Channel* before and after filling the empty slots. This is because the number of slots in the time domain is small and if the user requires large number of slots (e.g., >9) the user will be assigned consecutive subchannels with less slots in the time domain which decreases the number of empty slots compared to the *"Large Config"* where the same user with the same data may be assigned one subchannel and more slots in the time domain which may increase the number of empty slots. Also, *Sorted Best Channel* attempts to schedule users with highest data first, so those users will be assigned more than one consecutive subchannel if they have good SINR on all of them to satisfy their data demand which leads to less empty slots. While the *Round Robin Best Channel* attempts to schedule users in order, which means users with small data could be scheduled first and be assigned to best subchannels but because the data scheduled are small more empty slots are left in the subchannel and therefore less data transferred when compared with the *Sorted Best Channel* method. Fig. 6 shows that the number of empty slots after *Round Robin Best Channel* finishes is on average twice the number of the empty slots after *Sorted Best Channel* finishes, less number of empty slots contributes to less added data per frame after scheduling users in those empty slots. Also, it is important to note that the total number of PUSC slots in *"Small Config"* is much more less than that of *"Large Config"* but the load is the same for all simulations (number of users and minimum data to transfer per user per frame). For the same user who requires seven PUSC slots, in the *"Small Config"* the number of empty slots in the allocated subchannel (assuming one subchannel is allocated) is 2 which might not be enough to be allocated to other users, while in the *"Large Config"* the number of empty slots will be 7 which is probably enough to schedule other users.

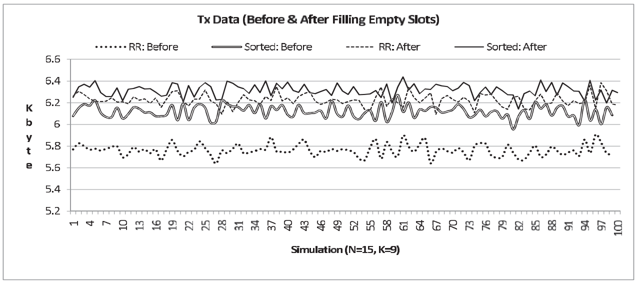

Fig. 3. Average KB transmitted per frame ($\mathrm{N} = 15, \mathrm{K} = 9$).

In summary, before filling empty slots, the *Sorted Best Channel* method makes better use of the available slots and schedules data higher than the *Round Robin Best Channel* method.

6.3.2 Scheduled Users

Another metric we analyzed in our simulation is the number of scheduled users, our algorithm does not attempt to maximize the number of scheduled users, but we wanted to get a feeling of how many users are scheduled per frame. Figs. 4 and 5 show the average number of scheduled users per frame.

In both *"Large Config"* and *"Small Config," Round Robin Best Channel* scheduled users more than *Sorted Best Channel* before filling empty slots and after filling the empty slots. *Round Robin Best Channel* schedules users in order without regard to the user's data; therefore, more users can be assigned to the same number of slots compared with the

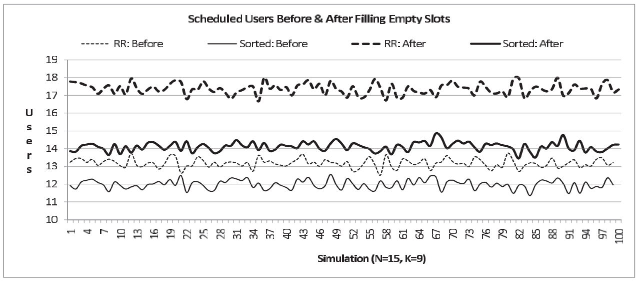

Fig. 5. Average scheduled users per frame ($N = 15, K = 9$).

case when users are sorted in descending order of data to be transferred.

In both *"Large Config"* and *"Small Config," Round Robin Best Channel* schedules users more than *Sorted Best Channel* after filling the empty slots because more empty slots are available when *Round Robin Best Channel* finishes compared to *Sorted Best Channel* as shown in Fig. 6, which in turn allows more users to be scheduled.

6.3.3 MCS Distribution

Both *Round Robin Best Channel* and *Sorted Best Channel* when attempting to schedule users try to find the best subchannel or combination of consecutive subchannels to meet the user's demand with the least number of slots. Both methods attempt to find the subchannel with the highest MCS, therefore the MCS distribution, defined as the percentage of slots allocated to each of the supported MCSs, is expected to be highly concentrated in the high MCS. Since we vary the subchannel conditions for random set of user's every frame, not all users will have subchannels with excellent channel quality and therefore will not be assigned high MCS (i.e., 64QAM) and could be assigned to subchannels with medium MCS (i.e., 16QAM) or low MCS (i.e., QPSK). Our simulations show that high percentage of assigned MCSs are in the high MCS range as shown in Fig. 7, almost 50 percent of the *"Large Config"* slots were allocated the highest MCS (64QAM5/6) in *Sorted Best Channel*, this is inline with what we expect.

In the *"Small Config,"* the MCS distribution shifted left one level, where only about 20 percent of slots are assigned 64QAM5/6. This is due to scheduling users in multiple consecutive subchannels to assign the required number of slots needed for the user's data, when this occurs, the assigned MCS is the lowest MCS over the consecutive subchannels. In the *"Small Config,"* the probability of assigning users to consecutive subchannels is higher than in the *"Large Config"* because less slots are available in the time domain per subchannel.

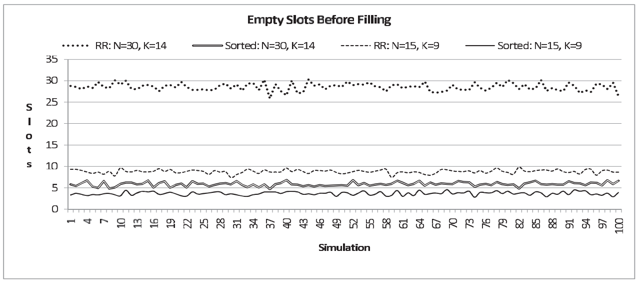

Fig. 6. Average empty slots per frame.

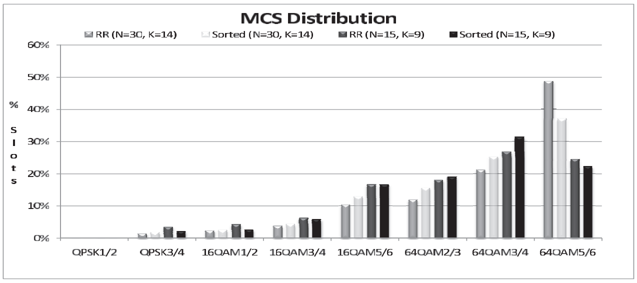

Fig. 7. MCS distribution.

6.3.4 Scheduled Throughput versus Maximum Throughput

The ratio of the scheduled throughput to the maximum theoretical frame throughput provides a measure of how efficient our scheduling algorithm in achieving its objective of maximizing frame throughput. The maximum frame throughput for the *"Small Config"* is 7.91 KB $(\frac{15\ subchannels \times 9\ PUSCSlots \times 480\ \text{bits/slot}}{(8\times 1{,}024)})$, and the maximum frame throughput for the *"Large Config"* is 24.61 KB $(\frac{30\ subchannels \times 14\ PUSCSlots \times 480\ \text{bits/slot}}{(8\times 1{,}024)})$.

For the *"Small Config"* before filling empty slots, the average scheduled data per frame for *Sorted Best Channel* is around 6.1 KB as shown in Fig. 3, this is about 77 percent of the maximum frame throughput. While for *Round Robin Best Channel*, the average scheduled data are 5.7 KB, which about 72 percent of the maximum frame throughput.

For the *"Large Config"* before filling empty slots, the average scheduled data per frame for *Sorted Best Channel* is around 21.25 KB as shown in Fig. 2, this is about 82 percent of the maximum frame throughput. While for *Round Robin Best Channel*, the average scheduled data are 19.75 KB, which about 80 percent of the maximum frame throughput.

The change in the throughput ratio between the *"Small Config"* and *"Large Config"* is due to the shift in the MCS distribution in the case of *"Small Config"* as we discussed in the previous section.

For *"Large Config,"* the *Sorted Best Channel* achieves 82 percent ratio compared to the maximum which is high ratio given the fact that we changed the subchannel quality for random set of users each frame; therefore, our *Best Channel* algorithm achieves high frame throughput.

6.4 Comparison with Other Methods

In this section, we compare our *Best Channel* algorithm performance with the performance of the algorithms studied in [27], [28]. In [27], [28], the users are attempted to be scheduled in order and are not sorted by the data size; therefore, we will use the *Round Robin Best Channel* simulation approach to compare our *Best Channel* algorithm with [27], [28].

In [27], the proposed algorithm allocates bursts into buckets, each bucket is one time slot wide and "number of subchannels" high and uses one MCS. The packets are allocated to buckets from top subchannel to the bottom

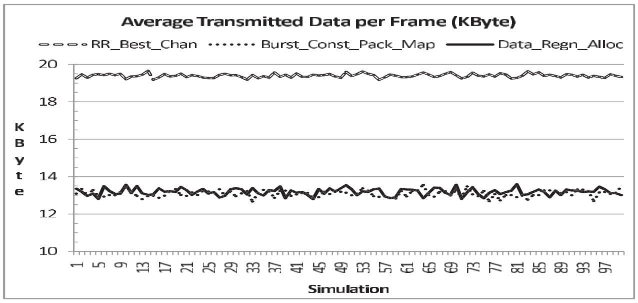

Fig. 8. Comparison with other methods: average transmitted data per frame.

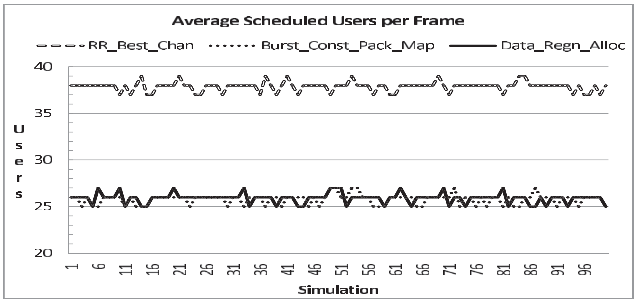

Fig. 9. Comparison with other methods: average scheduled users per frame.

subchannel. In [28], the proposed algorithm places data regions in a vertical raster starting from the bottom right slot of the downlink subframe. Each data region is modulated using one MCS and multiple users can be placed in a single data region. In both algorithms, the MCS (which directly impacts the transmitted data per slot) is determined in advance by task other than the proposed burst assignment algorithms.

We used the same simulation methodology and environment in the comparison except we did not fill the empty slots at the end of each simulation run and only simulated the *"Large Config."*

In Figs. 8 and 9, "RR_Best_Chan" represents our *Round Robin Best Channel*, "Burst_Const_Pack_Map" represents the algorithm proposed in [27], and "Data_Region_Alloc" represents the algorithm proposed in [28].

Fig. 8 shows the average scheduled data per frame for the three algorithms. The figure shows that the *Round Robin Best Channel* algorithms on average schedules 50 percent more data than the algorithms in [27] and [28]. This is due to the fact that *Round Robin Best Channel* algorithm tries to assign the user to the best subchannel(s) with available resources while the other two algorithms do not take the subchannel quality into consideration because the MCS is predetermined by another task; therefore, they assign the user to the next available subchannel; hence, *Round Robin Best Channel* algorithm requires less slots to schedule a user compared with the two algorithms. Therefore, more users can be scheduled in the frame by *Round Robin Best Channel* as shown in Fig. 9 which increases the frame transmitted data.

7 Literature Review

In [16], Wong et al. studied the margin-adaptive resource allocation problem (minimization of the overall transmission power under a data rate constraint) in multiuser OFDM system and formulated the problem of minimizing the overall transmit power by adaptively assigning subcarriers to the users along with the number of bits and power level to each subcarrier. The authors proposed an iterative subcarrier and power allocation algorithm to minimize the total transmit power given a set of fixed user data rates and bit error rate (BER) requirements. Given the instantaneous channel information, the algorithm obtains a suboptimal subcarrier allocation, and then single-user bit allocation is applied on the allocated subcarriers. Using this scheme, the overall required transmit power can be reduced by about 510 dB from the conventional OFDM without adaptive modulation.

In [10], Israeli et al. studied the problem of maximizing number of scheduled users in WiMAX system. Given a set of ordered users the authors proposed an algorithm to maximize the number of scheduled users maintaining the users' order. The authors defined for user i a Bounding Rectangle (BR_i) as a rectangle of at least area r_i (r_i is the weight assigned to user i) and at most does not include a subrectangle with area r_i. The MSS is allocated the Bounding Rectangle (some slots could be waisted if $BR_i > r_i$). The authors proposed an algorithm for Sequential Rectangle Placement (SRP) with proportional weights that divides the users into subranges, each subranges is further divided into sets and the users in each such set are placed on a separate set of rows. Our work is different as follows: 1) the authors try to maximize number of scheduled users, we try to maximize the scheduled data, 2) the algorithm does not take channel condition as seen by the user in the scheduling process.

In [11], Gutiérrez et al. formulated the joint scheduling and resource allocation problem and proposed an algorithm to maximize the spectral efficiency. The problem formulation has the limitation that all users are scheduled in every frame. To overcome this limitation, the authors proposed an iterative joint scheduling and resource allocation algorithm. The algorithm is divided into two phases: Service flow prioritization phase and Iterative Resource Allocation (IRA) phase. Our work is different than the work in [11] as follows: we concentrate on the user to subchannel assignment problem, we formulate the problem as IP problem and then we provide efficient algorithm to assign the users to subchannels taking into consideration the effective MCS a user can achieve over the allocated subchannels.

In [32], Galati Giordano et al. proposed two algorithms (Vector Allocation and Move Expand and Conquer (MEC)) for assigning users to bursts in the downlink subframe for the Advanced Modulation and Coding (AMC) permutation zone taking into consideration the subchannel quality. Vector Allocation algorithm does not guarantee rectangular allocation. In the Move Expand and Conquer algorithm after an initial allocation attempt, the solution is updated evaluating the most convenient small variations until the required demand is met or no variations are allowed. MEC schedules all users, since it starts by allocating one slot per user; therefore, it does not guarantee meeting the users demand in every frame and user's data could be fragmented.

Our approach differs from [32] in that 1) we study PUSC permutation zone, 2) we allocate the required slots to the user on the best subchannel or consecutive subchannels that meets user demand, 3) we do not guarantee the scheduling of all users in a frame but we guarantee to allocate the user's demand.

In [27], Ohseki et al. proposed an algorithm that allocates bursts into buckets, each bucket is one time slot wide and "number of subchannels" high. The packets are allocated to buckets from top subchannel to the bottom subchannel. Users with different PHY MODE (different MCS and repetition code) are placed in different buckets. At the end of scheduling, all buckets with the same PHY MODE are grouped together to construct one burst (similar to data region). This minimizes number of data regions, hence decreases control overhead.

In [28], Bacioccola et al. proposed simple data region allocation algorithm that allocates data regions from right to left and bottom to top of the downlink subframe. The algorithm places the data regions in a vertical raster starting from the bottom right slot of the downlink subframe without regard to the subchannel conditions. A data region can be split into a maximum of three smaller data regions each has an entry in the downlink MAP. Each data region is modulated using one MCS and multiple users can be placed in a single data region.

In [30], So-In et al. presented an algorithm that maximizes throughput and assigns bursts from right to left and bottom to top with the least width first vertically and the least height first horizontally. The algorithm sorts the allocations in descending order of size (number of required slots) and places the allocations in bursts in the frame without regard to the subchannel conditions.

In [31], Cicconetti et al. proposed Recursive Tiles and Stripes (RTS) algorithm to allocate data regions in the downlink subframe and pack HARQ bursts in those data regions with a profit associated with the HARQ bursts that RTS tries to maximize. The algorithm uses heuristic to solve the Subset-Sum Problem where each data region is an item and the HARQ bursts are the subitems and the algorithm tries to maximize the profit by selecting a subset of the subitems (HARQ bursts) to fit in the item (data region).

Our approach is different than [27], [28], [30], [31] in that they do not take channel quality into consideration while placing the bursts in the downlink subframe. They assume that the burst's MCS (data region MCS defined by the PHY MODE) assigned by a higher entity (i.e., QoS scheduler) and is passed to the burst allocator and the allocator is free to place the data region anywhere in the downlink subframe. We use a different approach in which we schedule at HARQ burst level and not at the data region level and the channel quality is the main factor for placing a burst in the subframe. Also, our approach maximizes the transmitted data in the subframe while the approaches in [27], [28], [30], [31] maximize number of allocated slots but not necessarily maximizes the transmitted data because the burst's MCS is not the most efficient. We compared our *Best Channel* algorithm performance with the performance of the algorithms studied in [27], [28] and presented the results in Section 6.4.

8 Conclusion

In this paper we studied the problem of maximizing the scheduled downlink data in the WiMAX/OFDMA wireless broadband systems. The user's data is allocated to bursts of variable size, the burst size varies based on the size of data to be transmitted and the user's subchannel(s) quality over the burst. We formulated the problem as maximization problem and proposed practical and efficient algorithm (*Best Channel*) to assign bursts in the PUSC downlink subframe while taking into consideration the user's channel quality. Users are scheduled based on the amount of data to be transferred in a frame and the best available subchannel(s) is allocated to the user. We proved that our algorithm (*Best Channel*) at least achieves throughput $\geq \frac{1}{2}\frac{b_{all}}{b_{max}}$ of the optimal. Through extensive simulations, we demonstrated that algorithm (*Best Channel*) achieved high frame throughput up to 80 percent of the maximum theoretical frame throughput. To the best of our knowledge, we are the first to study the problem of burst assignment in the downlink OFDMA subframe for PUSC subchannalization permutation mode taking user's channel quality into consideration in the assignment process.

Acknowledgments

The research of X.-Y. Li was partially supported by US National Science Foundation grants CNS-0832120 and CNS-1035894, the program for the Zhejiang Provincial Key Innovative Research Team, the program for the Zhejiang Provincial Overseas High-Level Talents (100 Talents Program), and the National Basic Research Program of China (973 Program) under grants 2010CB328100 and 2010CB334707.

References

[1] R. Chang, *Orthogonal Frequency Division Multiplexing,* US Patent 3488445, Jan. 1970.

[2] R. Chang, "Synthesis of Band-Limited Orthogonal Signals for Multichannel Data Transmission," *Bell System Technical J.,* vol. 45, pp. 1775-1796, Dec. 1966.

[3] T. Keller and L. Hanzo, "Adaptive Multicarrier Modulation: A Convenient Framework for Time-Frequency Processing in Wireless Communications," *Proc. IEEE,* vol. 88, no. 5, pp. 611-640, May 2000.

[4] B. Saltzberg, "Performance of an Efficient Parallel Data Transmission System," *IEEE Trans. Comm. Technology,* vol. 15, no. 6, pp. 805-811, Dec. 1967.

[5] S. Weinstein and P. Ebert, "Data Transmission by Frequency-Division Multiplexing Using the Discrete Fourier Transform," *IEEE Trans. Comm. Technology,* vol. 19, no. 5, pp. 628-634, Oct. 1971.

[6] Y.Z. William and W. Yiyan, "COFDM: An Overview," *IEEE Trans. Broadcasting,* vol. 41, no. 1, pp. 1-8, Mar. 1995.

[7] X.-Y. Li, A. Nusairat, Y. Wu, Y. Qi, J. Zhao, X. Chu, and Y. Liu, "Joint Throughput Optimization for Wireless Mesh Networks," *IEEE Trans. Mobile Computing,* vol. 8, no. 7, pp. 895-909, July 2009.

[8] W. Rhee and J.M. Cioffi, "Increase in Capacity of Multiuser OFDM System Using Dynamic Subchannel Allocation," *Proc. IEEE Vehicular Technology Conf.,* pp. 1085-1089, May 2000.

[9] Y.J. Zhang and K.B. Letaief, "Multiuser Adaptive Subcarrier-and-Bit Allocation with Adaptive Cell Selection for OFDM Systems," *IEEE Trans. Wireless Comm.,* vol. 3, no. 4, pp. 1566-1575, Sept. 2004.

[10] A. Israeli, D. Rawitz, and O. Sharon, "On the Complexity of Sequential Rectangle Placement in IEEE 802.16/WiMAX Systems," *Information and Computation,* vol. 206, no. 11, pp. 1334-1345, Nov. 2008.

[11] I. Gutiérrez, F. Bader, R. Aquilué, and J.L. Pijoan, "Contiguous Frequency-Time Resource Allocation and Scheduling for Wireless OFDMA Systems with QoS Support," *EURASIP J. Wireless Comm. and Networking,* vol. 2009, no. 11, pp. 1-12, Jan. 2009.

[12] S. Tang, X.-Y. Li, X. Wu, Y. Wu, X. Mao, P. Xu, and G. Chen, "Low Complexity Stable Link Scheduling for Maximizing Throughput in Wireless Networks," *Proc. Ann. IEEE Comm. Soc. Conf. Sensor, Mesh and Ad Hoc Comm. and Networks (SECON),* 2009.

[13] I.C. Wong, Z. Shen, B.L. Evans, and J.G. Andrews, "A Low Complexity Algorithm for Proportional Resource Allocation in OFDMA Systems," *Proc. IEEE Workshop Signal Processing Systems,* pp. 1-6, Oct. 2004.

[14] J. Jang and K.B. Lee, "Transmit Power Adaptation for Multiuser OFDM Systems," *IEEE J. Selected Areas in Comm.,* vol. 21, no. 2, pp. 171-178, Feb. 2003.

[15] Z. Shen, J.G. Andrews, and B.L. Evans, "Optimal Power Allocation in Multiuser OFDM Systems," *Proc. IEEE GlobeCom,* vol. 1, pp. 337-341, Dec. 2003.

[16] C.Y. Wong, R.S. Cheng, K.B. Letaief, and R.D. Murch, "Multiuser OFDM System with Adaptive Subcarrier, Bit, and Power Allocation," *IEEE J. Selected Areas in Comm.,* vol. 17, no. 10, pp. 1747-1758, Oct. 1999.

[17] P. Chalermsook and J. Chuzhoy, "Maximum Independent Set of Rectangles," *Proc. 19th Ann. ACM-SIAM Symp. Discrete Algorithms,* pp. 892-901, 2009.

[18] Y. Wang, W. Wang, X.-Y. Li, and W.-Z. Song, "Interference-Aware Joint Routing and TDMA Link Scheduling for Static Wireless Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 19, no. 12, pp. 1709-1726, Dec. 2008.

[19] L. Fleischer, M.X. Goemans, V.S. Mirrokni, and M. Sviridenko, "Tight Approximation Algorithms for Maximum General Assignment Problems," *Proc. 17th Ann. ACM-SIAM Symp. Discrete Algorithm (SODA '06),* pp. 611-620, 2006.

[20] R. Cohen, L. Katzir, and D. Raz, "An Efficient Approximation for the Generalized Assignment Problem," *Information Processing Letters,* vol. 100, no. 4, pp. 162-166, Nov. 2006.

[21] D.S. Johnson, "Fast Algorithms for Bin Packing," *J. Computer and System Sciences,* vol. 8, no. 3, pp. 272-314, June 1974.

[22] N. Karmarkar and R.M. Karp, "An Efficient Approximation Scheme for the One-Dimensional Bin-Packing Problem," *Proc. 23rd Ann. Symp. Foundations of Computer Science (FOCS '82),* 1982.

[23] B. Xia and Z. Tan, "Tighter Bounds of the First Fit Algorithm for the Bin-Packing Problem," *Discrete Applied Math.,* vol. 158, no. 15, pp. 1668-1675, Aug. 2010.

[24] E. Hopper and B.C.H. Turton, "An Empirical Investigation of Meta-Heuristic and Heuristic Algorithms for a 2D Packing Problem," *European J. Operational Research,* vol. 128, no. 1, pp. 34-57, Jan. 2001.

[25] K. Jansen and G. Zhang, "On Rectangle Packing: Maximizing Benefits," *Proc. 15th Ann. ACM-SIAM Symp. Discrete Algorithms (SODA '04),* pp. 204-213, 2004.

[26] Y.-L. Wu, W. Huang, S.-C. Lau, C.K. Wong, and G.H. Young, "An Effective Quasi-Human Based Heuristic for Solving the Rectangle Packing Problem," *European J. Operational Research,* vol. 141, no. 2, pp. 341-358, Sept. 2002.

[27] T. Ohseki, M. Morita, and T. Inoue, "Burst Construction and Packet Mapping Scheme for OFDMA Downlinks in IEEE 802.16 Systems," *Proc. IEEE GlobeCom,* pp. 4307-4311, 2007.

[28] A. Bacioccola, C. Cicconetti, L. Lenzini, E. Mingozzi, and A. Erta, "A Downlink Data Region Allocation Algorithm for IEEE 802.16e OFDMA," *Proc. Sixth Int'l Conf. Information, Comm. and Signal Processing,* pp. 1-5, 2007.

[29] C. Desset, E.B. de Lima Filho, and G. Lenoir, "WiMAX Downlink OFDMA Burst Placement for Optimized Receiver Duty-Cycling," *Proc. IEEE Int'l Conf. Comm.,* pp. 5149-5154, 2007.

[30] C. So-In, R. Jain, and A.-K. Tamini, "eOCSA: An algorithm for Burst Mapping with Strict QoS Requirements in IEEE 802.16e Mobile WiMAX Networks," *Proc. Second IFIP Conf. Wireless Days (WD),* pp. 1-5, Dec. 2009.

[31] C. Cicconetti, L. Lenzini, A. Lodi, S. Martello, E. Mingozzi, and M. Monaci, "Efficient Two-Dimensional Data Allocation in IEEE 802.16 OFDMA," *Proc. IEEE INFOCOM,* pp. 1-9, Mar. 2010.

[32] L. Galati Giordano, D. Lopez-Perez, L. Reggiani, L. Dossi, A. Juttner, and J. Zhang, "Two-Dimensional Radio Resource Allocation Algorithms with Contiguity Constraint for IEEE 802.16e Systems," *Proc. IEEE 20th Int'l Symp. Personal, Indoor and Mobile Radio Comm.,* pp. 2224-2229, 2009.

[33] J.G. Andrews, A. Ghosh, and R. Muhamed, *Fundamentals of WiMAX: Understanding Broadband Wireless Networking.* Prentice-Hall, Feb. 2007.

Ashraf Nusairat received the bachelor's degree in computer engineering from the Jordan University of Science and Technology in 1993, the MS degree in computer science from The University of Akron, Ohio, in 1999, and the PhD degree in computer science from the Illinois Institute of Technology in 2011. Currently, he works for Motorola Solutions as a wireless broadband systems architect. His research interests are in the areas of wireless broadband networks, wireless mesh networks, and wireless ad hoc networks. He is a member of the IEEE.

Xiang-Yang Li received the BEng degree in computer science and the bachelor's degree in business management from Tsinghua University, PR China, in 1995. He received the MS and PhD degrees in computer science from the University of Illinois at Urbana-Champaign in 2000 and 2001, respectively. He was a visiting professor of Microsoft Research Asia from May 2007 to August 2008. He has been an associate professor since 2006 and was an assistant professor of computer science at the Illinois Institute of Technology from 2000 to 2006. His research interests span wireless ad hoc and sensor networks, noncooperative computing, computational geometry, and algorithms. He was a guest editor of special issues for *ACM Mobile Networks and Applications* and the *IEEE Journal on Selected Areas in Communications*. He is a senior member of the IEEE.