

Efficient Allocation of Periodic Feedback Channels in Broadband Wireless Networks

Reuven Cohen Guy Grebla
 Department of Computer Science
 Technion—Israel Institute of Technology
 Haifa 32000, Israel

Abstract—Advanced wireless technologies such as MIMO require each mobile station (MS) to send a lot of feedback to the base station. This periodic feedback consumes much of the uplink bandwidth. This expensive bandwidth is very often viewed as a major obstacle to the deployment of MIMO and other advanced closed-loop wireless technologies. This paper is the first to propose a framework for efficient allocation of periodic feedback channels to the nodes of a wireless network. Several relevant optimization problems are defined and efficient algorithms for solving them are presented. A scheme for deciding when the BS should invoke each algorithm is also proposed and shown through simulations to perform very well.

I. INTRODUCTION

In order to achieve high throughput in wireless networks, the transmitter needs to obtain up-to-date information about the channel quality observed by the receiver. To this end, advanced wireless standards require each mobile station (MS) to periodically transmit to the base station (BS) its Channel Quality Indicator (CQI). CQI is a measure of the downlink mobile channel, and is used by the BS to adapt the modulation and coding parameters to the channel status of the corresponding node. These measurements also play a major role in the BS's scheduling algorithm [5], [6].

When Multiple Input Multiple Output (MIMO) technology is incorporated into 4G wireless networks, the amount of feedback that must be transmitted from the MSs to the BS increases dramatically. In the MIMO closed-loop spatial multiplexing mode, for example, this feedback includes the Rank Indicator (RI), the Precoding Matrix Indicator (PMI), and the Channel Quality Indicator (CQI). The BS uses the PMI reports to determine how the precoding matrix should be configured for transmission. The RI reports indicate the number of MIMO transmission layers available to the reporting MS. All these indicators require a lot of expensive uplink bandwidth, mainly because they are sent periodically as long as there is transmission on the downlink channel. This expensive bandwidth is very often viewed as a major obstacle to the deployment of MIMO and other advanced closed-loop wireless technologies. Therefore, the uplink bandwidth to these indicators must be allocated very carefully, while achieving certain optimization objectives.

Our framework encompasses all common indicators, including CQI, RI and PMI. CQI feedbacks can be either wideband CQI, where the CQI is measured for the entire downlink channel bandwidth, or subband CQI, where each CQI is

measured over a subband. We do not distinguish between the various indicators and view them collectively as CSI (Channel Status Information) channels. Both 3GPP/LTE [1] and WiMax/802.16 [11] support periodic and aperiodic CSI feedback. While aperiodic CSI feedback requires the BS to send a signaling message each time it wants to receive a CSI report from an MS, periodic CSI feedback requires only one signaling message for the allocation of a CSI channel and one for its release. The allocation message indicates the location and periodicity of the CSI slots that comprise the allocated CSI channel. Once a CSI channel is allocated, the MS transmits CSI messages on the slots of this channel until it receives a deallocation message.

The contribution of this paper is threefold. It is, to the best of our knowledge, the first to present a formal framework for the allocation of periodic CSI channels. It also defines, again for the first time, several problems relevant to this framework and presents efficient algorithms for solving them. Finally, it presents a holistic scheme that indicates when the BS should invoke each of the proposed algorithms.

The framework proposed in this paper defines a profit/utility function for the allocation of a CSI channel to each MS. While the proposed framework and algorithms are general enough to address every profit function, we propose and discuss a specific function, for which the profit is equal to the expected number of packets transmitted to an MS using a correct CSI value due to the allocation of a CSI channel with a certain bandwidth.

Two commonly used BS scheduling models are proportional fair [21] and semi-persistent [13]. A proportional fair scheduler adjusts the instantaneous transmission rate to each user dynamically, even on the subframe granularity. A semi-persistent scheduler adjusts the instantaneous transmission rates less frequently; e.g., once every 10,000 subframes. While the framework presented in this paper is generic and can work with both scheduling schemes, to make the discussion more concrete, we present a specific profit function, which depends on the number of packets transmitted to each MS. Such a profit scheme is mostly suitable for semi-persistent schedulers.

The rest of this paper is organized as follows. In Section II, we discuss related work. In Section III, we show how to allocate slots to CSI channels using a complete binary tree, in order to guarantee an efficient collision-free allocation, and describe the considered CSI channel allocation model. Section IV is the core of the paper. It defines the CSI allocation problems and presents efficient algorithms for them. In Section V we

study the performance of the various algorithms and present a complete BS scheme for the allocation of CSI channels. Finally, Section VI concludes the paper.

II. RELATED WORK

Previous works have addressed aspects of the problem other than the one we address here. For example, with the exception of [16] and [23], previous works have not attempted to adjust the periodicity of the CQI reports to the specific needs of each MS. Rather, they have tried to reduce the cost of the CQI reports by: (i) not sending CQI reports if the channel condition has not significantly changed [7], [9], [10], [12], [19], [22]; (ii) sending a single CQI report to a group of MSs [15]; or (iii) sending a single CQI report for a subset of OFDM subchannels [19], [20]. All these works are orthogonal to the scheme and algorithms presented in this paper.

In [16], [23], the authors propose a CQI allocation scheme for 802.16. Their scheme views the CQI bandwidth as a “toy brick.” In contrast to these works, we represent the CSI bandwidth as a binary tree, which allows us to minimize the number of changes for allocating a CSI channel when the available CSI bandwidth is fragmented. We also allow different channels to have different profit functions and seek to optimize the total profit of the BS.

In [8], the authors address the OVFSF code assignment problem. While their work does not target the allocation of CSI channels, some of their results are relevant to us. In particular, the allocation framework proposed in this paper is based on a complete binary tree that is similar to the OVFSF tree used in [8]. However, OVFSF codes in [8] can only be assigned to a specific level in the tree whereas we allow each CSI channel to be associated with different levels and profit.

In [12], an adaptive CQI scheme is proposed, where a node reports the CQI value only if it has changed since the last report or if a timer expires. With the proposed scheme, battery capacity of the MS is conserved and uplink interference is reduced. While [12] also considers periodic CQI channels, it does not, in contrast to our scheme, (a) attempt to change the periodicity of the CQI reports; (b) address the case where the CQI bandwidth is insufficient for all the CQI channels.

In [22], the problem of getting too many CQI reports at the BS is studied. The goal of the proposed scheme is to reduce the number of these reports by careful selection of the specific OFDM subchannels for which such reports are required. In [10] a similar scheme is proposed, which also takes into account the QoS requirements of each MS. In [7], a new metric for the performance of CQI schemes is proposed and studied. It takes into account the total resources consumed by each CQI scheme. It is then used for comparing different, periodic and aperiodic, CQI schemes with different SNR values.

In [20], the authors propose to reduce the CQI bandwidth cost by reporting a single CQI value for a subset of sufficiently proximate OFDM subchannels. A hierarchical tree is used to create groups of subchannels. In [19], a similar hierarchical mechanism is used, but only CQI values with sufficient quality are reported. It is claimed that the proposed scheme can significantly reduce the CQI feedback overhead at the expense of

a little downlink performance degradation. In [15], proximate MSs are considered as a “CQI feedback group,” and only one representative node is asked to send a CQI report.

Our paper deals with the allocation of feedback channels, and not with how and when the nodes send feedback information. This important topic is addressed by many papers, some of which are mentioned in what follows.

In [18], the authors present an efficient method for calculating the PMI at the receiver. The method is based on maximizing the mutual information between the transmitted and received symbols with respect to the precoding matrix applied at the transmitter.

In [17], the authors present an efficient method for calculating the PMI, RI and CQI at the MS. To reduce the MS computational burden, the proposed method decomposes the problem into two separate steps: jointly evaluating the PMI and RI using a mutual information metric, and choosing the CQI value to achieve a given target block error ratio constraint.

In [3], the authors discuss the suitability of two options for the closed loop precoded MIMO transmission in LTE uplink. The first option is to use the same codebook of precoding matrices defined for LTE downlink, while the second option exploits the singular value decomposition of the channel matrix. Qualitative benefits of both solutions are discussed.

Finally, in [2] the authors give a brief overview of the LTE and LTE-advanced system downlink transmission and discuss different precoding matrix selection criteria. Following the analytical and numerical results, the authors conclude that the “minimum post-mean squared error” based criterion is a good candidate for precoding matrix selection at the receivers.

III. PRELIMINARIES

A. CSI channels

Decision-making schemes that might decide not to send certain CSI reports [7], [9], [10], [12], [19], [22], e.g., if the channel condition has not changed notably, cannot easily take advantage of the unused slots. This is because these slots are too short for regular packets and because the MS cannot rely on their availability. The approach taken by our paper is different in the sense that the BS allocates different bandwidth to different CSI channels in accordance with each channel’s individual profit function. Using the scheme we propose, the BS views the CSI bandwidth (i.e., the uplink bandwidth dedicated to the CSI channels) as a shared resource, to be dynamically allocated to the MSs. The BS can also adjust the size of this resource. For instance, when it realizes that there are not so many dynamic MSs in its cell, the BS can decrease the total CSI bandwidth and use it for other purposes. The CSI bandwidth is divided into several super-channels. A super-channel consists of one slot in every uplink frame (Figure 1a). Therefore, the number of such super-channels is equal to the number of CSI slots in every frame. Each super-channel is divided into multiple CSI channels, each of which uses only one slot every τ frames (Figure 1b). This paper presents algorithms for the division of a super-channel into multiple channels and for the allocation and deallocation of these CSI channels. To allocate a CSI channel, the BS sends to an MS a control message with the following parameters:

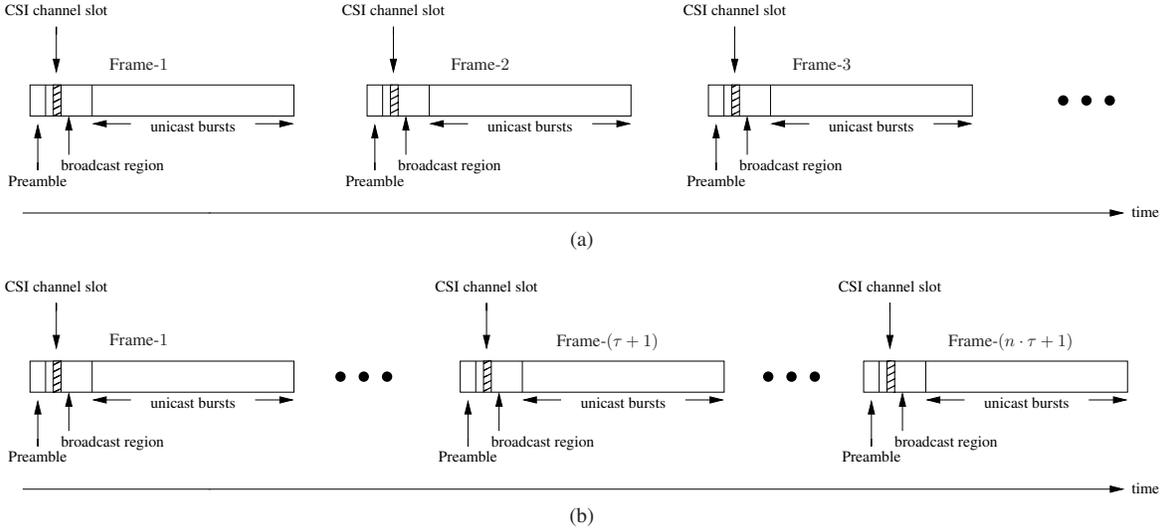


Fig. 1. (a) A CSI super-channel consists of the same slot in every uplink OFDMA frame; (b) a CSI channel consists of the same slot in every $\tau = 2^i$ frames

- (a) The sequence number of the first frame that contains a slot of this channel.
- (b) The number of frames τ between two consecutive slots of this channel.
- (c) The time during which this CSI channel is allocated to the MS. The BS can also allocate the channel with no expiration time, and then explicitly request it back.

A CSI channel C_j is denoted $\alpha_j | \tau_j$, where α_j is the sequence number of the first frame that contains a slot of this channel and τ_j is the periodicity of the slots. A smaller value of τ_j means more frequent CSI reports, which provide the BS with more accurate information about the channel state of the corresponding MS. However, if τ_j is too small, the BS is likely to receive too many identical CSI reports. Therefore, the optimal value of τ_j depends on the stability of the channel, which is affected by many factors such as MS mobility speed, physical obstacles, weather conditions, interference from other BSs/MSs or other wireless networks.

B. Power of 2 allocation

A power of 2 allocation is an allocation of CSI channels for which $\tau = 2^i$ holds for every channel, where i is an integer between 0 and C . Such an allocation is useful because it can prevent collisions between slots of two different CSI channels.

Definition 1: Two or more CSI channels are said to collide if they contain the same slot. In other words, a collision occurs between $\alpha_1 | \tau_1$ and $\alpha_2 | \tau_2$ if for some integers $x > 0$ and $y > 0$, $\alpha_1 + \tau_1 \cdot x = \alpha_2 + \tau_2 \cdot y$.

We now show how a power of 2 allocation can be performed when the bandwidth of each super-channel is maintained using a complete binary tree T_C whose height is C . We refer to such a tree as a *CSI allocation tree*. Then we shall see how such an allocation can be guaranteed to be collision-free. The leaves of T_C are in level 0, their parents are in level 1, and so on. We assign a label to every tree node in the following way. For a node in level l , the assigned label consists of $C - l$ digits from which the first $C - l - 1$ are the same as of the node's

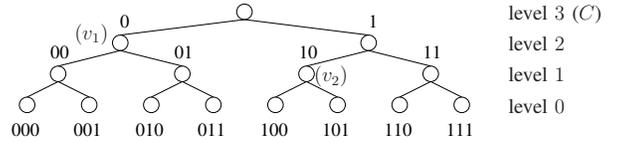


Fig. 2. An example of a labeled CSI allocation tree for a super-channel

parent and the last digit is set to 0 for a left child or to 1 for a right child. Figure 2 gives an example.

Let r be the reversed label of node v in the tree, and $d(r)$ be the decimal value of r . Then, node v is the root of a subtree whose height is l associated with the CSI slots $d(r) | 2^{C-l}$. For example, node v_1 in the tree of Figure 2 is the root of a subtree whose height is 2 associated with the CSI slots $0 | 2$, while node v_2 is the root of a subtree whose height is 1 associated with the slots $1 | 4$. We now prove that if each root-to-leaf path in the allocation tree has at most one allocated node, then the CSI channels represented by the tree do not collide. For instance, consider the two trees in Figure 3 and suppose that the black nodes indicate allocated slots. In both trees there is at most one allocated node on every root-to-leaf path. By the lemma below, this indicates that the CSI channels represented by the allocated tree nodes are collision-free. The fraction near every black node indicates the fraction of the super-channel bandwidth assigned to the corresponding CSI channel.

Lemma 1: Two nodes of a CSI allocation tree are on the same root-to-leaf path if and only if their corresponding slots collide.

Proof: Consider node v_1 in level l_1 and node v_2 in level l_2 of the tree. Without loss of generality, let $l_1 > l_2$. Recall that the corresponding CSI slots of v_1 and v_2 are $d(r_1) | 2^{C-l_1}$ and $d(r_2) | 2^{C-l_2}$, where r_1 and r_2 are the reverse labels of v_1 and v_2 respectively.

If v_1 and v_2 are on the same root-to-leaf path, the last $C - l_1$ digits of r_1 and r_2 are identical. Therefore, there exists an integer x such that $d(r_1) + 2^{C-l_1} \cdot x = d(r_2)$, implying that the corresponding CSI slots of v_1 and v_2 collide. If v_1 and v_2 are not on the same root-to-leaf path, the last $C - l_1$ digits

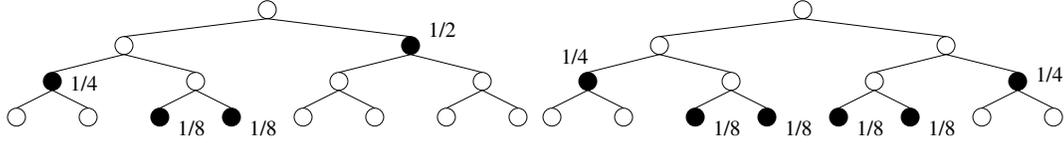


Fig. 3. Examples for two collision-free allocations

frame #	I	$I+1$	$I+2$	$I+3$	$I+4$	$I+5$	$I+6$	$I+7$	$I+8$	$I+9$	$I+10$	$I+11$	$I+12$	$I+13$	$I+14$	$I+15$
	1	2				2				2				2		

Fig. 4. Fragmentation of a CSI channel

of r_1 and r_2 are different. Therefore, for every two integers x and y , $d(r_1) + 2^{C-l_1} \cdot x \neq d(r_2) + 2^{C-l_2} \cdot y$ holds. ■

When 2 nodes not on the same root-to-leaf path are allocated to a single CSI channel, channel fragmentation occurs. For example, assigning the nodes 0|16 and 1|4 to the same CSI channel is translated to the allocation of slots shown in Figure 4. We can see that the slots of this channel are not uniformly distributed along the time axis. This is a suboptimal allocation because some of the slots are too close to previous slots and are not useful. In other words, the quality of a CSI channel is determined according to the maximum distance between two consecutive slots. For a given bandwidth, this distance is minimized if the slots are of equal distance from each other. For this reason, we do not allow channel fragmentation.

C. CSI Allocation Framework

Following the discussion above, we now describe our requirements from a CSI allocation framework:

(R1) Collisions and fragmentation of CSI channels are not allowed. Therefore, (a) a super-channel is divided into multiple CSI channels using a complete binary tree; (b) each CSI channel consists of at most one tree node, which is the root of a subtree; (c) subtrees allocated to different CSI channels are mutually disjoint.

(R2) For each tree level l and MS_j , a profit function $P_j(l)$ indicates the “profit of the system” from allocating this CSI channel to this MS.

While our framework is general enough to address every P_j function, throughout the paper we focus on the following specific one:

$$P_j(l) = \begin{cases} E_j \cdot 2^{(l_j^{\text{MAX}})} & \text{if } l > l_j^{\text{MAX}} \\ E_j \cdot 2^{(l)} & \text{Otherwise.} \end{cases} \quad (1)$$

We now show how this function guarantees that the profit is equal to the number of packets expected to be transmitted to MS_j using a correct CSI value. To this end, we show how the BS determines the profit function for each MS_j , namely, how it calculates E_j and l_j^{MAX} . The BS first estimates the dynamicity of the downlink channel of MS_j . This estimation is translated into a metric w_j , which indicates the average time window during which the CSI value of MS_j changes. The BS also calculates the average data packet rate r_j for

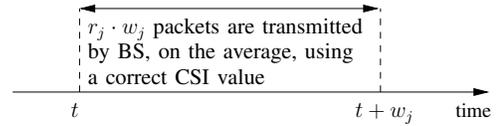


Fig. 5. Consecutive packets transmitted to MS_j using correct CSI value

MS_j , and sets $E_j \leftarrow w_j \cdot r_j$. Consequently, E_j is the average number of packets transmitted to MS_j using a correct CSI value (Figure 5).

For example, if a CSI channel is allocated to MS_j in level 0 (a leaf tree node), then the periodicity of this CSI channel is 2^C , and therefore $P_j(0) = E_j$ is the desired value. If we allocate to MS_j a CSI channel in a higher level, then E_j is multiplied by the number of CSI reports in a time window of 2^C subframes. When the time between two CSI reports becomes close to w_j , the BS is likely to receive from MS_j many identical CSI reports. Thus, there is an upper bound l_j^{MAX} on the level of the tree for which extra profit is obtained. This upper bound is the tree level where the CSI periodicity is w_j ; thus $l_j^{\text{MAX}} = C - \log w_j$.

The function $P_j(l)$ takes into account the number of packets for MS_j and not the amount of data transmitted to each user. However, as mentioned earlier, our framework supports any profit function. One can easily take into account the size of each packet by defining a new profit function $\tilde{P}_j(l)$, which equals $P_j(l)$ multiplied by the average packet size destined for MS_j . Thus, $\tilde{P}_j(l)$ would represent the amount of data transmitted to MS_j using a correct CSI value.

IV. ALGORITHMS FOR CSI ALLOCATION

A. Optimization Criterion

In this section we address the following problems related to the CSI allocation framework described in Section III:

- 1) How to allocate bandwidth to CSI channels when a tree (super-channel) is empty.
- 2) How to reallocate the bandwidth of a released CSI channel.
- 3) How to allocate a CSI channel to a new MS when the available CSI bandwidth is fragmented.
- 4) How to change the bandwidth of a CSI channel in order to take into account changes in the profit function of some MS(s), e.g., due to a new mobility pattern.

We first present algorithms for the various cases and then combine them into a scheme that indicates when each algorithm should be executed by the BS.

When a new MS enters the cell, the BS needs to determine its corresponding profit function. To this end, the BS allocates a basic (minimum bandwidth) CSI channel to every active MS. The bandwidth dedicated for the initial CSI channels is assumed to be sufficient for all active MSs. For example, the BS may have a binary tree whose height is $\lceil \log M \rceil$ for this basic allocation, where M is the maximum number of MSs that can be activated in the cell. Then, the initial CSI channels are allocated from the leaves of this tree. The initial CSI channel is used by the BS in order to determine the initial E_j value for MS_j , and to allocate a broader CSI channel when necessary. Since the BS can easily determine the expected number of packets transmitted by MS_j between two CSI reports, it can also determine E_j and $P_j(l)$. To simplify the discussion, if no CSI channel is allocated to MS_j (except the initial channel), we say that MS_j is allocated a tree node at level $l_j = -1$, and that $P_j(-1) = 0$.

In all the problems defined below, the optimization criteria is maximizing the total profit of the system; i.e., $\sum_{j=1}^n P_j(l_j)$, where MS_1, \dots, MS_n are the active MSs. As explained earlier, since our profit function is equal to the expected number of packets transmitted when the BS has a correct CSI value, maximizing the total profit is equivalent to maximizing the total number of packets transmitted by all active MSs using a correct CSI value. This is also equivalent to minimizing the expected number of packets transmitted by the BS when it has an incorrect CSI value.

B. CSI Allocation When the Tree Is Empty

We start with the basic problem, where we assume that the tree is empty and the goal is to find the best allocation for a given set of active MSs. This problem is referred to as CF-CSI-E (Collision Free CSI allocation in an Empty tree), and is formally defined as follows:

Problem 1 (CF-CSI-E):

Instance: The height of the allocation tree C and the profit function P_j for every active MS_j $1 \leq j \leq n$.

Objective: Find an allocation of CSI channels to the active MSs such that the total profit is maximized.

We now show that CF-CSI-E can be reduced to the Multiple Choice Multiple Knapsack Problem (MCKP) [14]. An MCKP instance is a set of m mutually disjoint classes N_1, \dots, N_m of items to be packed into a knapsack of capacity B . Each item $i \in N_j$ has a profit p_{ij} and a weight w_{ij} . The objective is to choose at most one item from each class such that the aggregated profit is maximized and the aggregated weight is not larger than B .

To reduce an instance of CF-CSI-E to an instance of MCKP, each CSI channel C_j is represented by a class N_j , and for each level i CSI subtree that can be allocated to C_j ($1 \leq i \leq l_j^{\max}$) there is an item $i \in N_j$. The knapsack capacity is set to $B = 2^C$. The weight of $i \in N_j$ is set to $w_{ij} = 2^i$ and the profit is set to $p_{ij} = P_j(i)$.

The above reduction gives rise to the following algorithm for CF-CSI-E:

Algorithm 1: (An algorithm for CF-CSI-E)

- 1) Reduce the CF-CSI-E instance to an MCKP instance as described above.
- 2) Run an algorithm, A_{MCKP} , that finds a solution to the MCKP instance.
- 3) Translate the solution returned by A_{MCKP} to a solution for CF-CSI-E, such that a CSI channel C_j is allocated a tree node in level i if item i in class N_j is chosen for the MCKP solution. ■

Lemma 2: If A_{MCKP} is an α -approximation to MCKP, Algorithm 1 is an α -approximation to CF-CSI-E.

MCKP has a simple 2-approximation greedy algorithm whose running time is $O(I \log I)$, where I is the total number of items [14]. Using linear selection, the running time can be improved to $O(I)$ [14]. It also has a pseudopolynomial time optimal dynamic programming algorithm whose running time is $O(B \cdot I)$ [14]. For practical instances, $B \leq 2^{10} = 1024$ holds for CF-CSI-E, because this allows a periodicity of up to 1 second. Thus, the running time for the optimal dynamic programming algorithm is $O(I)$. This algorithm is converted into an optimal polynomial time algorithm for CF-CSI-E.

The solution found by Algorithm 1 indicates only the tree level of each CSI channel and not the specific tree node. However, given such a solution, we use the following result, stated in [8] in the context of OVFS code assignment, to convert this information into a concrete allocation:

There exists a collision-free allocation of the tree nodes if and only if $\sum_{u \in V} 2^{l(u)} \leq 2^C$, where V is the set of all allocated nodes in the tree, $l(u)$ is the level of an allocated node u in the tree, and C is the height of the tree.

Figure 2 shows how a specific level l node is represented by a label consisting of $C-l$ digits. To obtain a concrete allocation, we sort the nodes in descending order of their level and for each node in level l , we find the smallest (in lexicographic order) label of $C-l$ digits that is still available. This process takes $O(|V| \log |V|)$ time, where V is the set of all allocated nodes in the tree.

The reduction of a CF-CSI-E instance to an MCKP instance in step 1 can be performed in $O(C \cdot n)$ time. If the running time of the MCKP algorithm used in step 2 is $T_{\text{MCKP}}(I, B)$, the total running time of Algorithm 1 is $O(C \cdot n + T_{\text{MCKP}}(C \cdot n, 2^C))$.

C. CSI Allocation with No Change to Previously Allocated CSI channels

We now define the second problem, referred to as CF-CSI-NC (Collision Free CSI allocation with No Change to previously allocated CSI channels). Here, some bandwidth of a super-channel tree becomes available following the release of a CSI channel when an active MS leaves the cell or becomes inactive. This bandwidth can be allocated by the BS to improve the total profit gained by the current active MSs.

Problem 2 (CF-CSI-NC):

Instance: The height of the allocation tree C , the profit function P_j for every active MS_j $1 \leq j \leq n$, and information about already allocated CSI channels.

Objective: Allocate the unused CSI bandwidth such that the gained profit is maximized.

Definition 2:

- (a) A free subtree in T is a subtree that contains only free nodes.
- (b) A free subtree is max-free if the subtree rooted at its parent is not free.

For example, Figure 6 shows 4 max-free subtrees (one of which is a leaf).

We now present an algorithm for CF-CSI-NC that is based on a reduction to the Multiple Choice Multiple Knapsack Problem (MC-MKP), which is an extension of MCKP to multiple knapsacks [14]. The instance of MC-MKP is a set of m mutually disjoint classes N_1, \dots, N_m of items and a set $\mathbb{B} = (B_1, \dots, B_{|\mathbb{B}|})$ of knapsack capacities. Each item $i \in N_j$ has a profit p_{ij} and a weight w_{ij} . The objective is to choose at most one item from each class and pack it in one of the knapsacks such that the total profit is maximized and the aggregated weight in each knapsack does not exceed its capacity.

As an example, we now define an MC-MKP instance and show its optimal solution. Our instance consists of two knapsacks with capacities $B_1 = 1$ and $B_2 = 2$, and 3 classes of items. Each class contains 2 items whose weights and profits are: $w_{11} = w_{12} = w_{13} = 1$, $p_{11} = p_{12} = 2$, $w_{21} = w_{22} = w_{23} = 2$, $p_{21} = p_{22} = 4$, $p_{13} = 3$, $p_{23} = 6$. An optimal solution for this instance is to pack item 1 from class N_1 in knapsack B_1 and item 2 from class N_3 in knapsack B_2 . This solution has a total profit of 8.

To reduce an instance of CF-CSI-NC to an instance of MC-MKP, each CSI channel C_j is represented by a class N_j , and for each level i subtree, which can be allocated to C_j ($1 \leq i \leq l_j^{\max}$), there is an item $i \in N_j$. Each max-free subtree of height h is represented by a knapsack of capacity 2^h in \mathbb{B} .

As an example of the reduction of CF-CSI-NC into MC-MKP, consider the CSI tree in Figure 6. This tree is translated into an MC-MKP instance that consists of 4 knapsacks whose capacities are 1, 2, 2, and 4. Suppose that there is one active MS with $E_j = 3$ and $l_j^{\max} = 2$. This MS is translated into a class with 3 items: item 1 has a profit of $E_j = 3$ and weight 1, item 2 has a profit of 6 and weight 2, and item 3 has a profit of 12 and weight 4.

The above reduction gives rise to the following algorithm for CF-CSI-NC:

Algorithm 2: (An algorithm for CF-CSI-NC)

- 1) Reduce the CF-CSI-NC instance to an MC-MKP instance as described above.
- 2) Run an algorithm, A_{MC-MKP} , that finds a solution for the MC-MKP instance.
- 3) Translate the solution returned by A_{MC-MKP} to a solution for CF-CSI-NC, such that a CSI channel C_j is allocated a tree node in level i of subtree z if item i in class N_j is packed in knapsack B_z of the MCKP solution. ■

Lemma 3: If A_{MC-MKP} is an α -approximation to MC-MKP, Algorithm 2 is an α -approximation to CF-CSI-NC.

In [4] it is shown that even without multiple choice, MC-MKP is hard to approximate in a fully polynomial time. We now present a 2-approximation greedy algorithm for MC-MKP. This algorithm combines the 2-approximation greedy algorithm for MCKP [14] and the 2-approximation algorithm for MKP [14].

Algorithm 3: (A 2-approximation greedy algorithm for MC-MKP)

- 1) For each class N_j with m items, create m new items, the first of which is the first item from N_j . For each item $i > 1$, the weight and profit are $w_{ij} - w_{(i-1)j}$ and $p_{ij} - p_{(i-1)j}$ respectively. From now on the algorithm relates to the new generated I items.
- 2) Sort the new items in decreasing order of their efficiencies (profit divided by weight).
- 3) Go over the knapsacks in increasing order of capacity. For each knapsack try to pack items in decreasing order of their efficiencies (only items whose weight is smaller than the current knapsack capacity are considered). The first item that does not fit into knapsack z is called the split item for z and denoted s_z .
- 4) Return the maximum between the items packed so far and the solution obtained by packing of s_z in knapsack z . ■

Lemma 4: Algorithm 3 is a 2-approximation to MC-MKP.

Proof: Consider the linear relaxation of a given MC-MKP instance. The total profit of the optimal solution for the linear relaxation is not smaller than that of the MC-MKP instance. Therefore, proving that Algorithm 3 is a 2-approximation with respect to the linear relaxation will complete the proof.

Let G be the total profit of the items packed at the end of step 3, S the total profit of the solution obtained by packing all the split items, and OPT the total profit of the optimal solution for the linear relaxation. Since the algorithm considers items in decreasing order of their efficiency, the items in G and fractions of the split items in S are the optimal solution to the linear relaxation, and therefore $S + G \geq OPT$ holds. The profit of the solution returned by Algorithm 3 in step 4 equals $\max\{S, G\}$, and therefore it is $\geq \frac{OPT}{2}$. ■

The running time of Algorithm 3 is $O(I \log I + K \cdot I)$, where I is the total number of items and K is the number of knapsacks. Using linear selection, the running time can be improved to $O(K \cdot I)$.

As an example of the execution of Algorithm 3, recall the MC-MKP instance considered earlier. Running the greedy algorithm on this instance first chooses item 1 of class 3 for knapsack 1, since this is the item with highest efficiency. Next, the algorithm proceeds to knapsack 2 and since all remaining items have an efficiency of 2, the profit of the returned solution is 7.

The solution found by Algorithm 3 is converted into a concrete allocation of CSI channels in the same way described earlier for Algorithm 1. However, this time sorting is performed for each max-free subtree (knapsack) separately.

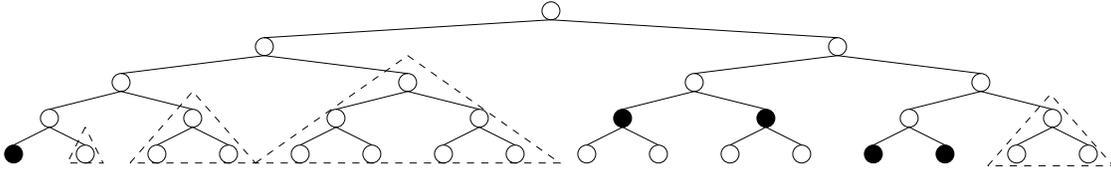


Fig. 6. A CSI tree with its 4 max-free subtrees (black nodes are occupied)

Since there are at most $O(n \cdot C)$ max-free subtrees, the CF-CSI-NC instance in step 1 can be reduced to an MC-MKP instance in $O(n \cdot C)$ time. If the running time of the MC-MKP algorithm used in step 2 is $T_{\text{MC-MKP}}$, the total running time of Algorithm 2 is $O(C \cdot n + T_{\text{MC-MKP}})$.

V. SIMULATION STUDY AND A COMPLETE BS SCHEME

We now present Monte Carlo simulation results of the various algorithms introduced in the paper. The goal of this section is twofold:

- To investigate the performance of these algorithms.
- Use the simulation results to develop a complete BS allocation scheme that indicates when the BS should invoke each algorithm.

Throughout this section we consider CSI allocation trees whose heights are $C = 10$ and $C = 8$. The average time window w_j between SINR changes is randomly selected between 32 and 1,024 subframes. Therefore, for each MS, $0 \leq l_j^{\text{MAX}} \leq 5$ holds. The average data packet rate r_j for each MS is uniformly chosen between 50 and 1,000 packets/second. For every MS $_j$, E_j is set to $r_j \cdot w_j$, and the profit function is as described in Eq. (1). An optimal pseudopolynomial time algorithm is used to solve the reduced MCKP instance in Algorithm 1 and a 2-approximation algorithm is used to solve the reduced MC-MKP instance in Algorithm 2.

A. The Performance of Algorithm 1 and Algorithm 2

We first compare Algorithm 1 to an algorithm that allocates only level-0 CSI channels (i.e., only tree leaves). As far as we know, this is the common scheme used today by BS. We consider up to 1,600 active MSs. For each number of MSs, we repeat the simulation 1,000 times with different seeds and average the results. In Figure 7 the x -axis indicates the number of active MSs (load) and the y -axis indicates the normalized profit obtained by Algorithm 1, i.e., the profit obtained by Algorithm 1 divided by the profit obtained by an algorithm that allocates only level-0 CSI channel to each MS.

As expected, when the number of MSs is small, allocating each of them a level-0 CSI channel leaves most of the allocation tree unused. Therefore, the normalized profit of Algorithm 1 is high. As the number of MSs increases, more of the tree can be used by allocating only level-0 nodes and the profit ratio decreases. Since a tree whose height is $C = 8$ has fewer leaves (bandwidth) than a tree with $C = 10$, level-0 allocation takes a bigger portion of the CSI tree. Therefore, the normalized profit is smaller for $C = 8$ than for $C = 10$.

Next, we compare the performance of Algorithm 1 with that of Algorithm 2. The total profit obtained by Algorithm 2 is

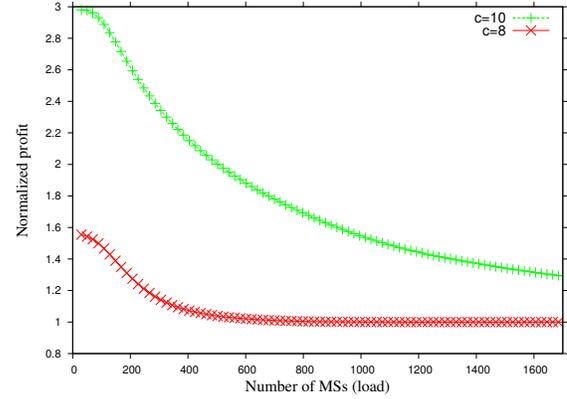


Fig. 7. Normalized profit of Algorithm 1 vs. the number of MSs (load)

expected to be smaller due to the fragmentation that might result because we do not allow this algorithm to delete already allocated CSI channels. For example, consider the CSI allocation tree in Figure 6 and assume that all MSs are allocated a CSI channel in their maximal level. Assume that a new MS whose maximal level is 3 becomes active. Since the height of the highest max-free subtree is 2, the new MS can be allocated a node in level ≤ 2 . In contrast, Algorithm 1 deletes the currently allocated CSI channels and returns an allocation where all MSs get their maximum level CSI channel, thereby obtaining a greater profit. If the new MS has a larger E_j value, the difference in profit is larger.

In the next trial, we start with an initial list of MSs and invoke Algorithm 1 to allocate them CSI channels. Then, we simulate 1,000 random events of adding or deleting randomly chosen MSs. Thus, the average load is proportional to the initial number of MSs. We maintain two separate CSI allocation trees. After each MS insertion or deletion, we invoke Algorithm 1 on the first tree and Algorithm 2 on the second. The results are shown in Figure 8, where the x -axis indicates the average number of initial MSs (load) and the y -axis indicates the total profit ratio between the tree maintained by Algorithm 1 and the tree maintained by Algorithm 2. Again, we present two curves: one for $C = 8$ and one for $C = 10$. When the number of MSs is small, there is enough CSI bandwidth to accommodate each arriving MS in its maximal level. Therefore, the profit ratio is very close to 1. As the number of MSs increases, Algorithm 2 is unable to allocate CSI channels at the optimal levels and the profit ratio increases. When the number of MSs increases further, both Algorithm 2 and Algorithm 1 are able to allocate CSI channels (at low levels) only to MSs whose E_j is high, and the profit ratio decreases back to 1.

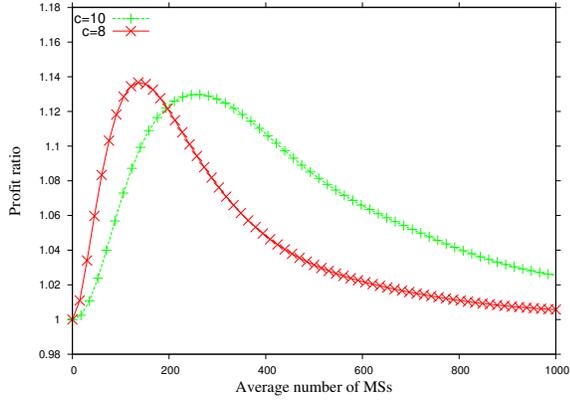


Fig. 8. Total profit of Algorithm 1 divided by total profit of Algorithm 2 vs. the number of MSs (load)

B. A Complete BS Scheme

We now combine Algorithm 1 and Algorithm 2 into a complete allocation scheme for the BS. An action is required from the BS in the following cases: (a) a new MS becomes active; (b) an active MS leaves the cell or becomes inactive; (c) the profit function of an active MS changes (e.g., due to a change in the user mobility speed). Algorithm 2 allows an increase in the profit without the overhead associated with the removal of existing CSI channels. However, Algorithm 2 is often unable to allocate a CSI channel not because the bandwidth is insufficient, but because it is fragmented. In such cases it might be more beneficial for the BS to clear the CSI allocation tree and invoke Algorithm 1. Thus, Algorithm 1 brings two important benefits to the scheduler. First, it serves as a benchmark for Algorithm 2, because it indicates the maximum total profit that can be obtained at every moment. Second, it can be occasionally invoked by the BS in order to replace the existing tree with a new one for the purpose of maximizing the profit.

All these considerations are combined into the complete BS scheme presented in Figure 9. The scheme is invoked when a new event is triggered at the BS. When a new MS becomes active or an active MS becomes inactive, the BS checks the ratio between the profit obtained by updating the current tree using Algorithm 2 and that obtained by building a new tree using Algorithm 1. If this ratio is smaller than a certain threshold t ($0 < t \leq 1$), then the new tree built by Algorithm 1 is used. Otherwise, the current tree is updated using Algorithm 2. This ensures that the obtained profit is never worse by a factor of t than the maximum possible. However, as t approaches 1, the number of CSI control (allocation and deallocation) messages sent to the MSs increases.

We evaluate the above scheme for a CSI allocation tree with $C = 10$. We set the average number of MSs to 250, which is where, as Figure 8 shows, the ratio between the profit obtained by Algorithm 1 and that obtained by Algorithm 2 is very high for $C = 10$. 1,000 random events are considered and averaged for each value of t .

Figure 10 shows the ratio between the profit achieved by the complete BS scheme and the (maximal) profit achieved

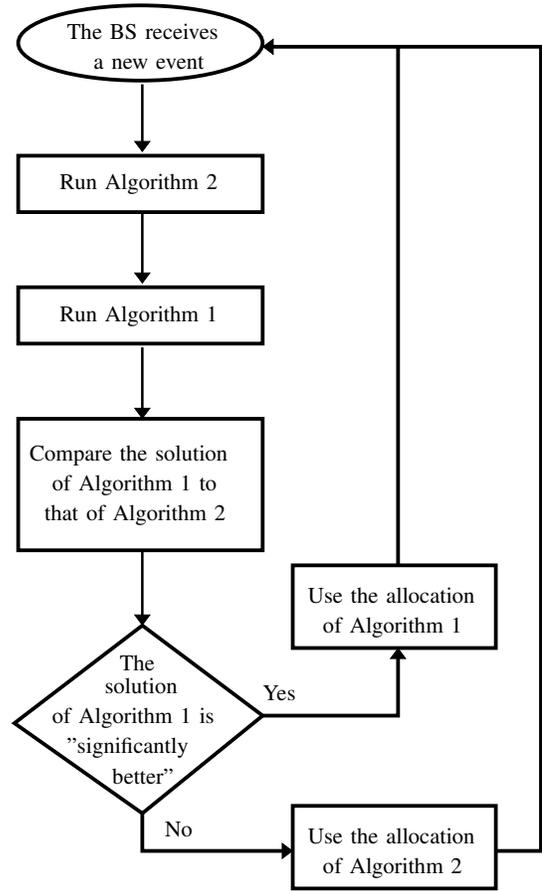


Fig. 9. The complete BS scheme (scheme 1)

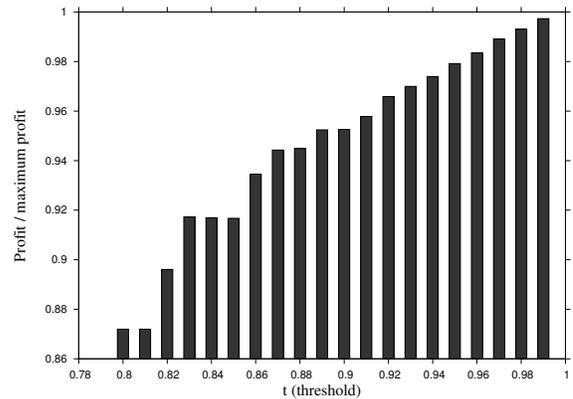


Fig. 10. The profit achieved by the proposed scheme divided by the maximum profit that can be achieved using Algorithm 1, as a function of the threshold t

by Algorithm 1 as a function of t . As expected, when the value of t increases, the profit of the scheme is closer to the optimal because the allocation of Algorithm 1 is used more often. To study the cost of using the allocation of Algorithm 1 more frequently, we average the number of changes per event for the proposed scheme as a function of t . The results are shown in Figure 11. We can see that a good tradeoff between efficiency and cost can be obtained for $0.84 \leq t \leq 0.94$.

Next, we enforce an upper bound of $t = 0.94$ and test the performance of the proposed scheme for different numbers

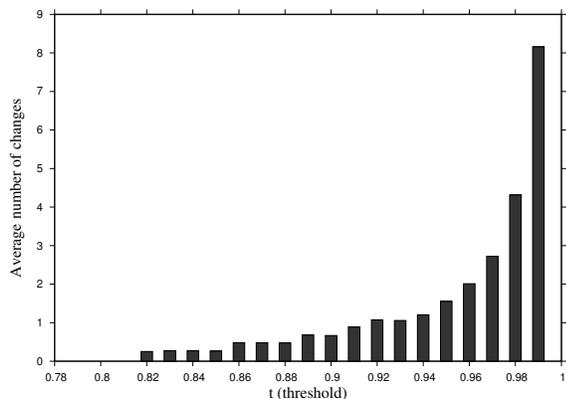


Fig. 11. The average number of changes per event of the proposed scheme as a function of the threshold t

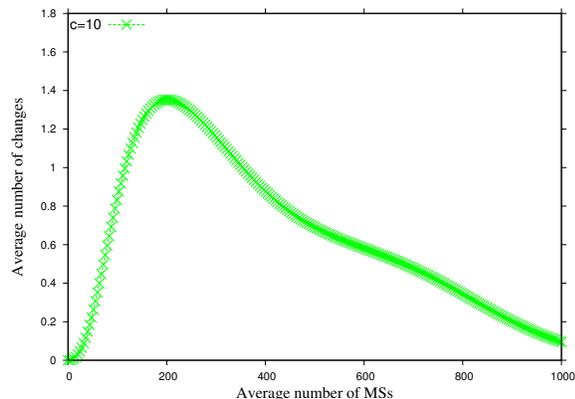


Fig. 12. The average number of changes per event of the proposed scheme as a function of the average number of MSs for $t = 0.94$

of active MSs. The results are shown in Figure 12, where the x -axis indicates the average number of MSs (load) and the y -axis indicates the average number of changes per event. The maximum number of changes per event occurs when the average number of MSs is ≈ 250 , which is expected because, as Figure 8 shows, this is where the maximum profit ratio between Algorithm 1 and Algorithm 2 is obtained.

We now show how to adapt the complete BS scheme to the case where the BS has limited CPU resources, and is unable to execute both Algorithm 1 and Algorithm 2 for each event. Two strategies can be employed: (a) use a more time efficient algorithm with a worst (or no) performance guarantee; (b) invoke Algorithm 1 less frequently by using a probabilistic algorithm that determines in advance the expected benefit from each execution. To demonstrate this idea, we observe that Algorithm 1 should be invoked more often when the allocation tree is “very fragmented,” and less often when the tree is “not very fragmented.”

Given an allocation tree, we define the *fragmentation index* as the number of leaves in the biggest max-free subtree divided by the total number of leaves in all max-free subtrees. For example, if there are 2^{C-1} max-free subtrees, and each of them is a leaf, then the fragmentation index is $\frac{1}{2^{C-1}}$. The adapted scheme computes the fragmentation index, and invokes Algorithm 1 with probability which is inverse proportional to this

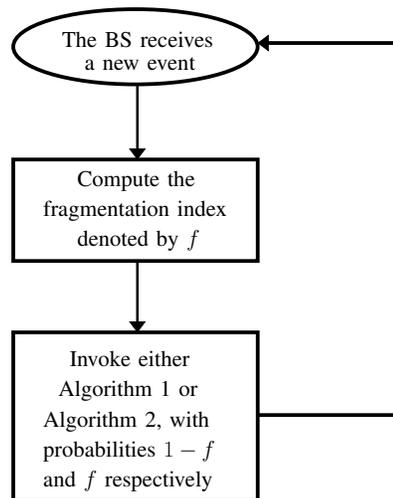


Fig. 13. The adapted BS scheme (scheme 2)

index. The modified scheme is shown in Figure 13, which now replaces Figure 9.

To evaluate the adapted scheme (scheme 2), we use a CSI allocation tree with $C = 10$. We start with an initial list of MSs and invoke Algorithm 1 to allocate their CSI channels. Then, we simulate 1,000 events of adding or deleting randomly chosen MSs. Thus, the average number of MSs equals the initial number. Figure 14 shows the normalized profit, i.e., the profit obtained using scheme 2 divided by the (maximal) profit obtained by invoking Algorithm 1 for each event, as a function of the average number of MSs. We see that scheme 2 achieves between 92% and 100% of the maximum profit. When the load is very small, the maximum bandwidth can be allocated to each CSI channel, and scheme 2 will thus obtain a profit very close to the maximum one. As the load increases, some channels do not get their maximum bandwidth and greater profit can be obtained using Algorithm 1 even when the tree is not very fragmented. Thus, we see that the normalized profit decreases. As the load increases further, we are more likely to have many pending MSs with high E_j values. Therefore, the tree is unlikely to be fragmented (in most cases only one node is free), and the profit ratio increases.

Next, we study the cost incurred by scheme 2 due to invoking Algorithm 1. To this end, we compute the average number of changes per event for scheme 2 as a function of the average number of MSs. The results are shown in Figure 15. As the average number of MSs increases, the tree becomes more fragmented, Algorithm 1 is invoked more frequently and the average number of changes increases as well. As the average number of MSs increases further, the tree is less fragmented and therefore Algorithm 1 is rarely invoked.

A comparison between Figure 15 and Figure 12 reveals that the average number of changes for scheme 2 is bigger than for scheme 1 when the load is light (less than 175 MSs in this concrete example). However, it is important to note that under light load, both Algorithm 1 and Algorithm 2 are very fast, because their running times are linear with the load. Therefore, scheme 1 can be invoked for light loads and scheme 2 for heavy loads.

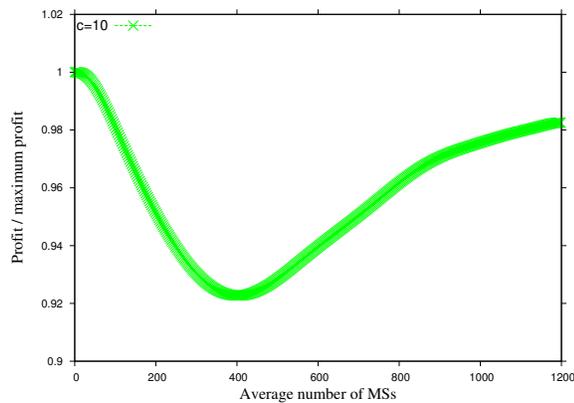


Fig. 14. The normalized profit achieved by scheme 2 vs. the average number of MSs (load)

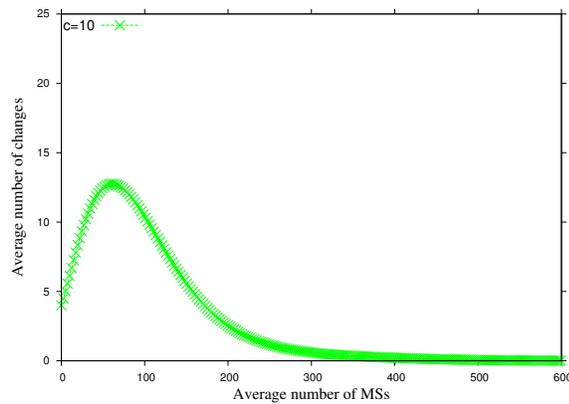


Fig. 15. The average number of changes per event of scheme 2 vs. the number of MSs (load)

VI. CONCLUSIONS

We presented a formal framework for the allocation of periodic CSI channels. In the proposed framework, the allocated bandwidth is maintained as a tree. Every MS is associated with a profit function that indicates the “profit of the system” from allocating a CSI channel of certain bandwidth to this MS. We defined two optimization problems for this framework, and proposed optimal polynomial-time algorithms for them. Our simulation study shows how the proposed algorithms can be combined into a unified scheme, to be invoked by the BS when a new event takes place.

One of the most important aspects of the proposed scheme is the definition of the profit function to be optimized by the BS. In this work, we used a function whose goal is to maximize the number of packets sent using the correct CSI value. We believe that other functions with other parameters should also be studied, and we leave this for future work.

REFERENCES

- [1] 3GPP. Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures (Release 10), 3GPP TS 36.213, June 2010.
- [2] Z. Bai, C. Spiegel, G. Bruck, P. Jung, M. Horvat, J. Berkmann, C. Drewes, and B. Gunzelmann. Closed loop transmission with precoding selection in LTE/LTE-Advanced system. *International Symposium on Applied Sciences in Biomedical and Communication Technologies*, pages 1–5, Nov. 2010.

- [3] G. Berardinelli, T. S. andrensen, P. Mogensen, and K. Pajukoski. SVD-based vs. release 8 codebooks for single user MIMO LTE-A uplink. *IEEE Vehicular Technology Conference*, pages 1–5, May. 2010.
- [4] C. Chekuri and S. Khanna. A PTAS for the multiple knapsack problem. *SODA*, pages 213–222, 2000.
- [5] R. Cohen, G. Grebla, and L. Katzir. Cross-layer hybrid FEC/ARQ reliable multicast with adaptive modulation and coding in broadband wireless networks. *IEEE/ACM Transactions on Networking*, 18(6):1908–1920, Dec. 2010.
- [6] R. Cohen and L. Katzir. Computational analysis and efficient algorithms for micro and macro OFDMA scheduling. *IEEE/ACM Transactions on Networking*, 18(1):15–26, Feb. 2010.
- [7] M. Döttling, B. Raaf, and J. Michel. Efficient channel quality feedback schemes for adaptive modulation and coding of packet data. *IEEE Vehicular Technology Conference*, 2:1243–1247, Sept. 2004.
- [8] T. Erlebach, R. Jacob, M. Mihalak, M. Nunkesser, G. Szabo, and P. Widmayer. An algorithmic view on OVFS code assignment. *Algorithmica*, 47(3):269–298, 2007.
- [9] N. Fukui. Study of channel quality feedback in UMTS HSPDA. *IEEE International Symposium on Personal, Indoor and Mobile Radio Communication Proceedings*, 1:336–340, Sept. 2003.
- [10] T. Hui, S. Qiaoyun, D. Kun, and L. Xingmin. A hybrid CQI feedback scheme for 4G wireless systems. *IEEE Vehicular Technology Conference*, pages 1–4, Apr. 2009.
- [11] Institute of Electrical and Electronics Engineers Inc. IEEE Draft Standard for Local and Metropolitan Area Networks – Part 16: Air Interface for Broadband Wireless Access Systems, June 2008.
- [12] S. Y. Jeon and D. H. Cho. Channel adaptive CQI reporting schemes for HSDPA systems. *IEEE Communications Letters*, 10:459–461, 2006.
- [13] D. Jiang, H. Wang, E. Malkamaki, and E. Tuomaala. Principle and performance of semi-persistent scheduling for VoIP in LTE system. *IEEE WiCom*, pages 2861–2864, 2007.
- [14] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer, 2004.
- [15] W. Lee and D. H. Cho. CQI feedback reduction based on spatial correlation in OFDMA system. *IEEE Vehicular Technology Conference*, pages 1–5, Sept. 2008.
- [16] W. Lilei, W. Xiaoyi, and X. Huimin. Strategies to improve the utilization ratio of CQI channels in IEEE 802.16 systems. *WiCOM '08*, pages 1–4.
- [17] S. Schwarz, C. Melführer, and M. Rupp. Calculation of the spatial preprocessing and link adaption feedback for 3GPP UMTS/LTE. *IEEE Conference on Wireless Advanced*, pages 1–6, Jun. 2010.
- [18] S. Schwarz, M. Wrulich, and M. Rupp. Mutual information based calculation of the precoding matrix indicator for 3GPP UMTS/LTE. *International ITG Workshop on Smart Antennas*, pages 52–58, Feb. 2010.
- [19] J. Su, B. Fan, K. Zheng, and W. Wang. A hierarchical selective CQI feedback scheme for 3GPP long-term evolution system. *IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications*, pages 5–8, Aug. 2007.
- [20] R. Vannithamby, G. Li, H. Yin, and S. Ahmadi. Proposal for IEEE 802.16m CQI feedback framework. May 2008.
- [21] C. Wengerter, J. Ohlhorst, and A. von Elbwart. Fairness and throughput analysis for generalized proportional fair frequency scheduling in OFDMA. *IEEE VTC*, 3:1903–1907, 2005.
- [22] X. Wu, J. Zhou, G. Li, and M. Wu. Low overhead CQI feedback in multi-carrier systems. *GLOBECOM*, pages 371–375, 2007.
- [23] W. Xiaoyi and W. Lilei. A new mathematical model for analyzing CQI channel allocation mechanism in IEEE 802.16 systems. *ICCSC*, pages 177–181, 2008.

PLACE
PHOTO
HERE

Reuven Cohen received the B.Sc., M.Sc. and Ph.D. degrees in Computer Science from the Technion - Israel Institute of Technology, completing his Ph.D. studies in 1991. From 1991 to 1993, he was with the IBM T.J. Watson Research Center, working on protocols for high speed networks. Since 1993, he has been a professor in the Department of Computer Science at the Technion. He has also been a consultant for numerous companies, mainly in the context of protocols and architectures for broadband access networks. Reuven Cohen has served as an

editor of the IEEE/ACM Transactions on Networking and the ACM/Kluwer Journal on Wireless Networks (WINET). He was the co-chair of the technical program committee of Infocom 2010 and headed the Israeli chapter of the IEEE Communications Society from 2002 to 2010.

PLACE
PHOTO
HERE

Guy Grebla received the B.A., M.A., and Ph.D. degrees in Computer Science from the Technion - Israel Institute of Technology, completing his Ph.D. studies in 2013. He is now a postdoctoral research scientist in Electrical Engineering department at Columbia University, New York, NY.