Reducing Physical Layer Control Signaling Using Mobile-Assisted Scheduling

Reza Moosavi and Erik G. Larsson

Linköping University Post Print

N.B.: When citing this work, cite the original article.

©2011 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Reza Moosavi and Erik G. Larsson, Reducing Physical Layer Control Signaling Using Mobile-Assisted Scheduling, 2011, IEEE Transactions on Wireless Communications, 99, 1-12.
http://dx.doi.org/10.1109/TWC.2012.120312.120680
Postprint available at: Linköping University Electronic Press
http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-68071

Reducing Physical Layer Control Signaling Using Mobile-Assisted Scheduling

Reza Moosavi and Erik G. Larsson

***Abstract*—We present a scheme for reducing the part of the downlink signaling traffic in wireless multiple access systems that contains scheduling information. The theoretical basis of the scheme is that the scheduling decisions made by the base station are correlated with the CSI reports from the mobiles. This correlation can be exploited by the source coding scheme that is used to compress the scheduling maps before they are sent to the mobiles. In the proposed scheme, this idea is implemented by letting the mobiles make tentative scheduling decisions themselves, and then letting the base station transmit "agreement maps" instead of raw scheduling maps to the mobiles. The agreement maps have lower entropy and they require less resources to be transmitted than the original scheduling maps do. The improvement can be substantial. We also model the task of finding the optimal scheduling assignments according to the proposed scheme as a combinatorial optimization problem and present an efficient algorithm to find the optimal solution.**

***Index Terms*—Mobile-assisted scheduling; Control signaling; Resource allocation; Signaling overhead**

I. Introduction

Several state-of-the-art wireless multiple access systems exploit multiuser diversity by adaptive scheduling of users on the channel resources which are beneficial to them, given the current state of the fading radio channel. This opportunistic scheduling approach in combination with adaptive modulation and coding (AMC) and hybrid automated repeat request (HARQ) is essential for reaching the aggressive goals on peak throughput that are set for modern mobile broadband multiple access systems [2], [3]. The multiuser scheduling is typically performed by the base station, which given channel-state information (CSI) from the different users, makes a scheduling decision according to a predetermined scheduling algorithm. One price to pay for the opportunistic scheduling is that the scheduling decisions need to be communicated to the users over a control channel, which consumes resources that could otherwise have been used for payload data. This process is herein referred to as *signaling* of the scheduling assignments. Keeping the amount of transmitted control data low allows for more efficient usage of the channel resources. The transmission of scheduling information may consume a significant amount of channel resources in some situations [4]–[6]. Therefore any improvements in resource efficiency of the representation and transmission of scheduling information on the control channel are of interest.

The authors are with the Dept. of Electrical Engineering (ISY), Linköping University, Linköping, Sweden. Email:{reza,egl}@isy.liu.se. This work was supported in part by Ericsson, VINNOVA and the Excellence Center at Linköping-Lund in Information Technology (ELLIIT). E. Larsson was a Royal Swedish Academy of Sciences (KVA) Research Fellow supported by a grant from the Knut and Alice Wallenberg Foundation.

This paper is a comprehensive extension of our conference paper [1].

There are basically two approaches to lower the signaling overhead caused by the transmission of scheduling assignments. The first approach is to reduce the amount of control information that needs to be signaled to the users. A common technique is to aggregate a number of resources together into larger blocks and allocate each such block to one user. This technique is currently used in 3GPP Long Term Evolution (LTE). More specifically, the smallest possible scheduling granularity in LTE is 12 consecutive OFDM subcarriers in frequency and 14 consecutive OFDM symbols in time [7, Chapter 16]. In [8], the authors proposed a compression scheme to encode the control information. The compression scheme therein consists of a run-length encoder, followed by a universal variable-length code (UVLC). In [4] another technique to reduce the control signaling was proposed. The idea therein is to use semi-persistent scheduling decisions and to change the scheduling assignments only if the gain in throughput is larger than the loss due to the signaling caused by transmission of new assignments.

In the second approach, rather than reducing the amount of control information, the focus is on finding efficient ways to transmit the scheduling assignments. For instance, [9] proposed to use AMC for the transmission of control information based on a grouping of users into groups with similar channel qualities. In [10], a method for differential signaling of scheduling assignments was proposed. This idea is motivated by the fact that the scheduling information of different users are correlated. Since users with good channel conditions can in principle decode the information intended to users with poor channel conditions, one can exploit this correlation by differentially encoding the scheduling information. In order to recover the scheduling information, a user must then decode the scheduling information of some of the other users.

Contribution: We present a scheme that improves the performance (in the sense of resources required) of the control signaling in wireless multiple access systems. The idea is motivated by the fact that the *scheduling map* which describes how resources are allocated to different users, is correlated with the CSI. More precisely, each user in the cell measures her received signal-to-noise ratio (SNR) or signal-to-interference-plus-noise ratio (SINR) and reports it back to the base station. The base station uses the received reports from all users to make the scheduling decisions. The users know their own CSI, and this knowledge can be exploited to compress the scheduling maps more efficiently. One way of exploiting this correlation is the scheme that we propose in this paper. The specific contributions of our work are:

- We introduce *mobile-assisted scheduling* as a means for signaling of scheduling assignments with reduced

signaling overhead.
- We show that under ideal conditions, using entropy achieving compression, the signaling overhead according to the proposed scheme is about 20% less than that of conventional signaling.
- We formulate an optimization problem that finds the optimal assignments for both conventional LTE-like and for mobile-assisted scheduling schemes and introduce an efficient algorithm that finds near-optimal assignments.
- We compare the performance of the proposed scheme with that of the conventional approach in terms of resources required for the signaling of the scheduling information.

II. System Model

We consider the downlink of a system with inband control signaling, that is, the transmission of the control information takes place concurrently with the transmission of payload data and thus consumes some of the channel resources. We assume that the data to be transmitted is formatted into *frames*. We furthermore assume that in each frame, there are N_s *resource blocks* (resources, for short) that can be assigned to the users. Each such resource represents a subdivision of the time, frequency, code or spatial domain or any combination of those. Throughout the paper, we assume, for simplicity of the exposition, that no multiuser MIMO is used. The extension to the case with multiuser MIMO is however possible. We also introduce the following terminology:

- $C_n(k)$ is the L-level quantized "information rate" that user k can support, if she is scheduled in resource block n. More precisely, $C_n(k)$ is a function of the SNR or SINR for user k in resource block n, that reflects how much information that flows to this user in this resource block. We assume that this quantity is known at the base station.
- N_u denotes the number of "active" users in the cell and $\mathcal{U} = \{1, 2, \ldots, N_u\}$ is the set of all users that are scheduled for transmission in the frame.
- $\mathcal{S}$ denotes the resource block assignments for the users scheduled for transmission in the frame. Also $\mathcal{S}_n$ stands for the index of the user assigned to resource block n under the assignment $\mathcal{S}$.

In practice $C_n(k)$ may be obtained from the CSI reports of the individual users. This requires a control signaling mechanism in the uplink. The more precise CSI is available at the base station, the better scheduling decisions can be made. Therefore, there is always a trade-off between the quality of the CSI and the scheduling decisions, and better CSI requires more uplink overhead. This is true for all systems. While one can study this trade-off, we defer from venturing further into that topic here.

We use the following rule for the scheduling:

$$\mathcal{S}_n \in \left\{ \ell \mid \Phi_\ell(C_n(\ell)) = \max_k \Phi_k(C_n(k)) \right\}, \qquad (1)$$

where $\Phi_k(x)$, $k = 1, \ldots, N_u$ can be any arbitrary function. For instance by using $\Phi_k(C_n(k)) = C_n(k)$, for all k, we get the "maximum throughput" or "max C/I" scheduler. This scheduler is highly favorable for the users near the base station since those users have stronger channels on the average. As an another example, in order to achieve fairness among the users, one can normalize $C_n(k)$ by the average throughput of user k, resulting in the proportional fair scheduler [6], [11]. In mathematical terms, this is equivalent to using $\Phi_k(C_n(k)) = \frac{C_n(k)}{T_k}$, where T_k denotes the received throughput of user k averaged over a time window in the past. For notational brevity, we define the *scheduling metric* $R_n(k)$ for user k corresponding to resource block n as $R_n(k) \triangleq \Phi_k(C_n(k))$. Note that this value is available both at the base station and at the kth user, since we assume that the base station has knowledge of the quantized CSI. With the above definition, (1) can be equivalently represented by

$$\mathcal{S}_n \in \left\{ \ell \mid R_n(\ell) = \max_k R_n(k) \right\}. \qquad (2)$$

Note that since $R_n(k)$ is quantized, the maximum in (2) is in general not unique. The overall goal is to make scheduling decisions that can be communicated to the users with the smallest possible signaling overhead. If there is only one user attaining the maximum in (2), then clearly the solution is to schedule that user in resource block n. If there are more candidates, that is, if two or more users attain the maximum in (2), we would like to assign resource block n to one of the candidates in such a way that the overall signaling cost is minimized. Let $\mathfrak{S}_n$ denote the set of all user candidates for resource block n, and let $\mathfrak{S} \triangleq \{\mathfrak{S}_n\}_{n=1}^{N_S}$ be the set of all assignments satisfying (2). Finding the optimum assignment is a combinatorial optimization problem with $|\mathfrak{S}| \sim \mathcal{O}(N_u^{N_s})$ possible solutions. We will use two schemes for conveying the scheduling assignments to the users. The first scheme is based on the conventional *LTE-like* approach, see Section III. The second is our new scheme, see Section IV. We then present efficient algorithms for calculating the scheduling assignments in Section VI.

III. Conventional Approach for Signaling of Scheduling Assignments

In the conventional scheme, the scheduling decisions are transmitted to each user individually. This scheme does not exploit the correlation that exists among the scheduling information of different users. This correlation comes from the fact that no two users can be scheduled in the same resource block.[1] For example, if a certain user is scheduled in a specific resource block, then other users will not be scheduled in the same resource block. However, individual transmission of scheduling assignments gives the opportunity to choose the channel code and/or modulation format used for the transmission of scheduling information for each user separately and therefore many practical systems such as LTE employ this scheme [7, Section 16.4]. In this paper, we will also assume that the scheduling assignments are individually transmitted to the users. However, in contrast to what is

[1] This is true assuming that multiuser MIMO is not used. Multiuser MIMO is outside the scope of this paper.

done in LTE, we first compress the scheduling information intended to each user using run-length encoding, whenever doing so shortens the signaling message. Run-length encoding is not optimal in general and sometimes it can even be more costly to use run-length encoding than to send uncompressed raw scheduling decisions. However, it does not require any knowledge of the map statistics and was used for similar purposes, e.g. in [1], [5], [6].

Since there are N_s resources, the scheduling map is a vector of length N_s which is obtained from (2). For each scheduled user k, once the scheduling decision has been made, we first find the *binary bitmap* $\mathcal{B}_k$ which determines the resources that are assigned to her. We then compress each such binary bitmap individually using run-length encoding whenever that shortens the bit representation of the map. Let $N_{\text{map}}^{(k)}$ denote the number of bits required to represent the binary bitmap associated with the kth user. The total number of bits required in this case is $N_{\text{Conv}} = \sum_{k=1}^{N_u} \left(N_{\text{map}}^{(k)} + \beta_k + 1\right)$, where β_k is a fixed overhead associated with initialization (to be defined in more detail later) and 1 accounts for an extra flag bit required for indicating whether run-length encoding is used or not.

IV. Proposed Mobile-Assisted Scheduling Scheme

The basic idea of our proposed scheme is to exploit the CSI knowledge that each user has of her own channel to more efficiently compress the scheduling decisions. More precisely, since resource block n is assigned to the user with the highest scheduling metric $R_n(k)$ and since a user, say user k, knows her own supported throughput $C_n(k)$ (and hence her own scheduling metric $R_n(k)$), we can exploit this available information to convey the scheduling assignments intended for user k using less resources. This process is reminiscent of source coding with side information. Once the scheduling decisions have been made, the base station sends a compressed *agreement map* $\mathcal{M}_k$ (which is essentially an N_s bits long binary vector whose elements correspond to one of the N_s resource blocks) and a *threshold* τ_k to each user k. The agreement map and the threshold indicate the resources that have been assigned to the her as follows. User k could assume that all the resources on which her corresponding scheduling metric $R_n(k)$ is equal or greater than the threshold τ_k, are assigned to her. However, this assumption would not always be true (as we will see shortly) and hence the base station sends the agreement map $\mathcal{M}_k$ to indicate whether that assumption would lead to the correct result or not. We use the term *conflict* for the situations where the base station does not agree with the "proposal" of some user. For instance, there might be several users with the maximum metric for a specific resource block and since only one user can be scheduled in that resource block, the base station can only admit one of the proposals and has to refuse the others and thus a conflict occurs. As another example, assume that a user, say user k, has the highest scheduling metric on resource block n and assume that her corresponding metric $R_n(k)$ is smaller than the threshold τ_k. Since the user has the highest metric, the base station assigns resource block n to her. However this is not what the user expects (since $R_n(k)$ is smaller than the threshold) and a conflict occurs.

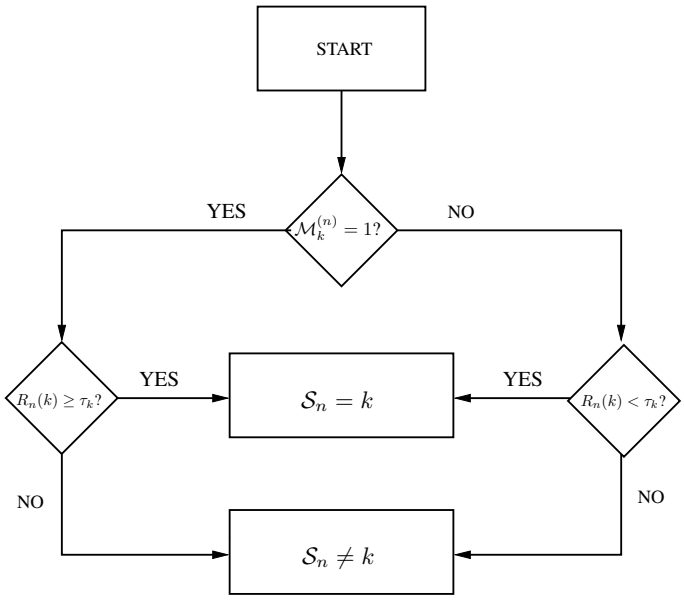


Fig. 1. Flowchart of the algorithm used by user k in order to determine whether resource block n is assigned to her or not, that is to determine whether $\mathcal{S}_n = k$. Here $\mathcal{M}_k^{(n)}$ denotes the nth element of the agreement map $\mathcal{M}_k$.

Upon the reception of the agreement map $\mathcal{M}_k$ and the threshold τ_k, user k uses the following algorithm to find out whether resource block n is assigned to her or not. If the corresponding field in the agreement map is "1", the user knows that the base station has agreed with her proposal, for the threshold given. Then she compares her metric $R_n(k)$ with the threshold and if it is greater than (or equal to) the threshold, she knows that she has been allocated that resource. Conversely, if $R_n(k)$ is less than the threshold, she concludes that the resource block has not been assigned to her. If, on the other hand, the corresponding field in the agreement map is "0", then the user knows that a conflict has occurred and that her proposal has not been accepted. In this case, if $R_n(k)$ is smaller than the threshold, she knows that the corresponding resource is assigned to her and vice versa. Fig. 1 summarizes this procedure.

Let $M_{\text{map}}^{(k)}$ be the number of bits required to represent the agreement map $\mathcal{M}_k$. The total number of bits required to represent the scheduling assignments with the proposed method is $N_{\text{MAS}} = \sum_{k=1}^{N_u} \left(M_{\text{map}}^{(k)} + \alpha_k + 1\right)$, where α_k is the number of bits required for initialization (to be defined later) and 1 accounts for the extra flag bit required to indicate whether run-length encoding is used or not.

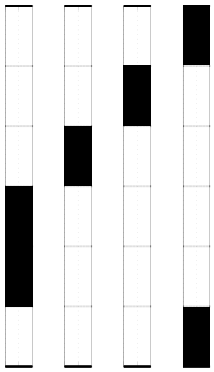
To illustrate how the proposed scheme works, we provide an example. Let there be $N_u = 4$ users, $N_s = 6$ resources, and $L = 5$ quantization levels. For simplicity, let us assume that the threshold is the same for all users, that is $\tau_k = \tau$ for $k = 1, \ldots, 4$. Table I illustrates the procedure for finding the agreement maps. Based on the scheduling metric from the users, the base station makes the scheduling decisions. In this example, the scheduling decisions are found directly for resource blocks 1, 2, 3 and 6, since on those blocks, there is only one user with the highest metric. However, for resource blocks 4 and 5, there is more than one user that the scheduler can select. Therefore, there are 4 possible scheduling maps in

Scheduling Metric

$R_n(1)$	$R_n(2)$	$R_n(3)$	$R_n(4)$
1	2	1	**5**
2	3	**5**	4
4	**5**	3	1
3	2	1	**3**
5	1	**5**	3
3	1	2	**5**

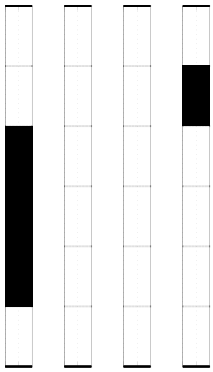
Sched. Map	Possible Agreement Maps $\tau = 4$	$\tau = 5$
U4	1 1 1 1	1 1 1 1
U3	1 1 1 0	1 1 1 1
U2	0 1 1 1	1 1 1 1
U1	0 1 1 1	0 1 1 1
U4	1 1 1 0	1 1 1 0
U1	1 1 0 1	1 1 0 1
U3	0 1 1 1	0 1 1 1
U4	1 1 1 1	1 1 1 1

Binary Maps



"Best" Agreement Maps

$\tau = 4$



$\tau = 5$

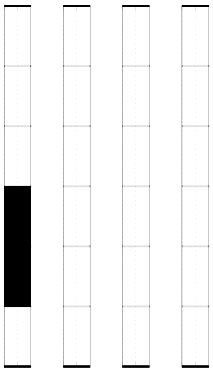


TABLE I
ILLUSTRATION OF THE PROPOSED MOBILE-ASSISTED SCHEDULING SCHEME.

this case, for each of which we may find the agreement maps with respect to a given set of thresholds. Table I also shows the possible agreement maps for two choices of thresholds: (i) when all thresholds are 4, that is $\tau = 4$ and (ii) when all thresholds are 5, corresponding to the case $\tau = 5$. The binary scheduling maps and also the "best" agreement maps (in the sense of requiring the minimum number of bits for representation), are also given for the two threshold sets. As we see, the agreement maps are less "noisy" compared to the binary maps and hence they can be compressed to a greater degree.

V. Theoretical Justification of the Proposed Scheme

We compare the conventional and the proposed schemes for compressing scheduling maps in terms of the entropy of their associated binary maps. Thereby the implicit assumption is the use of an entropy achieving compression scheme. This generally requires the maps to be infinitely long [12]. While this might not be practical due to short block lengths and tight delay requirements, it provides an insight into the ultimate improvement that we can achieve. For the analysis to follow, we assume that the scheduling metrics of each user, say user k, are independent of the scheduling metrics of the other users and have a uniform distribution over $[0, a_k]$. We assume without loss of generality that $a_1 \leq a_2 \leq \ldots \leq a_{N_u}$. Note that with these assumptions, the scheduling metrics of the users will be independent and non-identically (i.n.d.) distributed, which is useful for the situations where for instance the users are spread out in a cell with different distances to the base station.

To compute the entropy of the binary maps, we first need to find the probability of having a "1" in each position. Consider the event that the nth slot is assigned to the ith user, and denote this event by X_i. We would like to find $\Pr\{X_i\}$. For notational brevity, let $R_k = R_n(k)$, for $k = 1, 2, \ldots, N_u$ and let $I_{-i} \triangleq \{1, \ldots, i-1, i+1, \ldots, N_u\}$. Since we assign the nth slot to ith user if her metric is greater than the metrics of other users, we derive (3), with $a_0 = 0$.

Now let us compute the probability that the nth bit in the agreement map for the ith user is "1". First recall that we agree with a user either if (i) her scheduling metric is greater than the threshold τ_i and the slot has been assigned to her, or if (ii) her scheduling metric is less than the threshold and the slot has

$$p_i \triangleq \Pr\{X_i\} = \Pr\{R_i > R_k, \forall k \in I_{-i}\} = \int_{-\infty}^{\infty} p\left(R_i > R_k, \forall k \in I_{-i} | R_i\right) p_{R_i}(r)\mathrm{d}r$$
$$= \frac{1}{a_i}\int_0^{a_i} \prod_{k\in I_{-i}} p_{R_k}(r_k < r)\mathrm{d}r = \frac{1}{a_i}\int_0^{a_1} \prod_{k\in I_{-i}} p_{R_k}(r_k < r)\mathrm{d}r + \cdots + \frac{1}{a_i}\int_{a_{i-1}}^{a_i} \prod_{k\in I_{-i}} p_{R_k}(r_k < r)\mathrm{d}r$$
$$= \sum_{j=1}^{i} \frac{1}{N_u - j + 1} \cdot \frac{1}{\prod_{k=j}^{N_u} a_k} \left(a_j^{N_u-j+1} - a_{j-1}^{N_u-j+1}\right) \tag{3}$$

not been assigned to her. Note that this probability depends on the threshold τ_i and we would like to choose τ_i such that the entropy of the maps becomes as small as possible. Since the scheduling metric of user i lies in the interval $[0, a_i]$, we have $0 \le \tau_i \le a_i$. We thus assume for the coming discussions that $\tau_i \in [a_{i'-1}, a_{i'}]$, for some $i' \in \{1, \ldots, i\}$. Now we can write

$$q_i \triangleq \Pr\left\{\mathcal{M}_i^{(n)} = 1\right\} = \Pr\{R_i > \tau_i, X_i\} + \Pr\{R_i < \tau_i, X_i^c\} \tag{4}$$

where X_i, as before, is the event that the nth slot is assigned to the ith user. Each term in (4) can be written separately as (5) and (6) respectively, resulting in (7), where $\mathrm{sgn}(x)$ is the sign function with $\mathrm{sgn}(0) = 0$.

The entropy for a binary source is given by $H(p) = -p\log_2(p) - (1-p)\log_2(1-p)$ where p is the probability of observing a "1". For the proposed scheme the threshold can be optimized such that the entropy of the agreement map is made as small as possible. In special cases, we are able to use the property of $H(p)$ to find the optimal thresholds in closed form. More precisely, we know that $H(p)$ is a monotonically increasing function of p for $0 \le p \le 1/2$ and it is a monotonically decreasing function of p for $1/2 \le p \le 1$, which can be used to find the optimum thresholds in some cases, for example see case 1 below. We will solve this problem for the following two scenarios: (i) all a_i are equal to 1, corresponding to the case where the scheduling metrics are i.i.d. with a uniform distribution over $[0, 1]$, and (ii) the case with $a_i = i$ for $i = 1, \ldots, N_u$, which corresponds to the scenario where some users are much more likely than the others to be picked by the scheduler.

A. Case 1: i.i.d. scheduling metrics

In this case using (3), we see that $p_i = 1/N_u$ for $i = 1, \ldots, N_u$. This is natural, since in this case any user is equally likely to "win" the scheduling block n, due to the i.i.d. scheduling metric assumption. Also from (7), we see that $q_i = \tau_i + (1 - 2\tau_i^{N_u})/N_u$ for $i = 1, \ldots, N_u$. In this case, finding the optimum threshold is equivalent to maximizing q_i, which is done by taking the derivative of q_i with respect to τ_i and setting it to zero, resulting in $\tau_i^* = 2^{\frac{-1}{N_u-1}}$. We thus have $q_i^* = \frac{1}{N_u} + \left(1 - \frac{1}{N_u}\right) 2^{\frac{-1}{N_u-1}}$.

Fig. 2 illustrates the entropy of the binary maps and the best choice for the threshold τ_i^* as a function of the number of users respectively. As we can see the entropy of the agreement map is smaller than the entropy of the scheduling maps with the conventional approach. This means that after appropriate compression, the map should require less resources to be transmitted. As the number of users grows, the probability of assigning a given slot to a user decreases (cf. (3)). Therefore on the average, when there are many users in the cell, in order for a user to be scheduled her scheduling metric should be larger than in the case with a small number of users. Thus the optimum threshold should be an increasing function of the number of users. This explains the behavior of the optimum threshold curve in Fig. 2.

B. Case 2: i.n.d. scheduling metrics

In this case using (3) and (7) and after appropriate simplifications, we have

$$p_i = \sum_{j=1}^{i} \frac{(j-1)!}{(N_u - j + 1)N_u!}\left(j^{N_u-j+1} - (j-1)^{N_u-j+1}\right),$$

$$q_i = \frac{\tau_i}{i} + \frac{(i'-1)!\left(i'^{N_u-i'+1} + (i'-1)^{N_u-i'+1} - 2\tau_i^{N_u-i'+1}\right)}{(N_u - i' + 1)N_u!}$$
$$+ \sum_{j=1}^{i} \frac{(j-1)!\mathrm{sgn}(j-i')}{(N_u - j + 1)N_u!}\left(j^{N_u-j+1} - (j-1)^{N_u-j+1}\right),$$

with $i' = \lceil \tau_i \rceil$. Minimizing q_i with respect to τ_i can be done numerically in this case. Fig. 3 illustrates the normalized entropy sum of the binary maps $\frac{1}{N_u}\sum_{i=1}^{N_u} H(p_i)$ and that of the agreement maps $\frac{1}{N_u}\sum_{i=1}^{N_u} H(q_i^*)$, as a function of the number of users. As the results show, the agreement maps have lower entropy than the scheduling maps and hence they can be compressed more efficiently compared to the conventional approach in this case too.

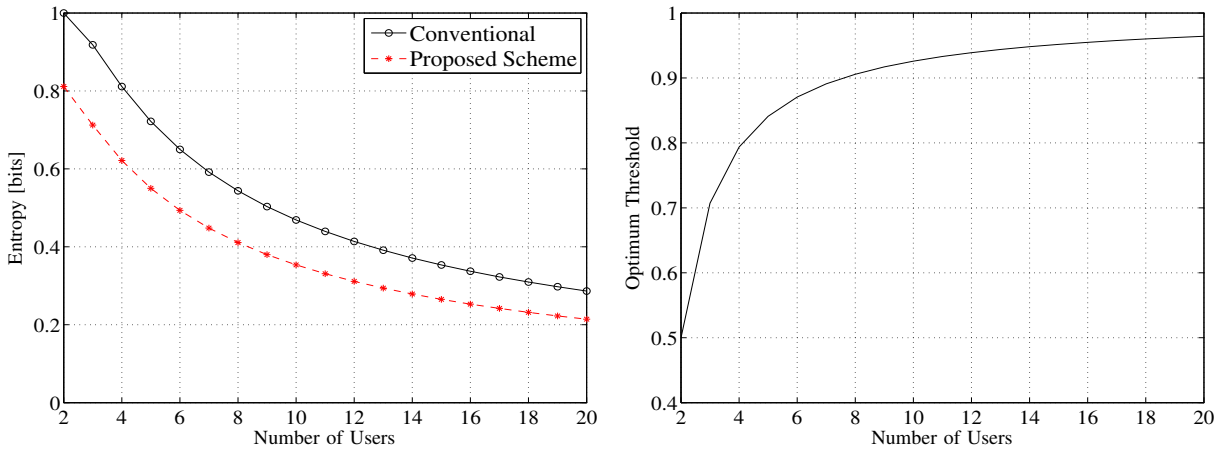
VI. Optimal Scheduling to Minimize Signaling Overhead

In this section, we formulate the optimization problem that finds the optimum scheduling assignments for both the conventional and the proposed schemes. Note that upon receiving a new set of CSI reports, this problem needs to be solved to obtain the new scheduling assignments. The formulation is inspired by the work in [5] which deals with finding the scheduling assignment that achieves the maximum system throughput according to the conventional approach (not according to the proposed mobile-assisted scheduling scheme). As an improvement to the work in [5], herein we consider the fact that run-length encoding does not always yield an

$$\Pr\{R_i > \tau_i, X_i\} = \int_{-\infty}^{\infty} p\left(R_i > \tau_i, X | R_i\right) p_{R_i}(r) \mathrm{d}r = \frac{1}{a_i} \int_{\tau_i}^{a_i} p\left(R_k < r, \forall k \in I_{-i}\right) \mathrm{d}r$$
$$= \frac{1}{a_i} \int_{\tau_i}^{a_i'} p\left(R_k < r, \forall k \in I_{-i}\right) \mathrm{d}r + \frac{1}{a_i} \int_{a_i'}^{a_{i'+1}} p\left(R_k < r, \forall k \in I_{-i}\right) \mathrm{d}r + \cdots + \frac{1}{a_i} \int_{a_{i-1}}^{a_i} p\left(R_k < r, \forall k \in I_{-i}\right) \mathrm{d}r$$
$$= \frac{1}{N_u - i' + 1} \cdot \frac{a_{i'}^{N_u - i' + 1} - \tau_i^{N_u - i' + 1}}{\prod_{k=i'}^{N_u} a_k} + \sum_{j=i'+1}^{i} \frac{1}{N_u - j + 1} \cdot \frac{1}{\prod_{k=j}^{N_u} a_k} \left(a_j^{N_u - j + 1} - a_{j-1}^{N_u - j + 1}\right), \tag{5}$$

$$\Pr\{R_i < \tau_i, X_i^c\} = \int_{-\infty}^{\infty} p\left(R_i < \tau_i, X_i^c | R_i\right) p_{R_i}(r) \mathrm{d}r = \frac{1}{a_i} \int_{0}^{\tau_i} p\left(R_k > r, \text{for some } k \in I_{-i}\right) \mathrm{d}r$$
$$= \frac{1}{a_i} \int_0^{\tau_i} \Big(1 - p(R_k < r, \forall k \in I_{-i})\Big) \mathrm{d}r = \frac{\tau_i}{a_i} - \frac{1}{a_i} \int_0^{a_1} p\left(R_k < r, \forall k \in I_{-i}\right)$$
$$- \frac{1}{a_i} \int_{a_1}^{a_2} p\left(R_k < r, \forall k \in I_{-i}\right) - \cdots - \frac{1}{a_i} \int_{a_{i'-1}}^{\tau_i} p\left(R_k < r, \forall k \in I_{-i}\right)$$
$$= \frac{\tau_i}{a_i} - \sum_{j=1}^{i'-1} \frac{1}{N_u - j + 1} \cdot \frac{a_j^{N_u - j + 1} - a_{j-1}^{N_u - j + 1}}{\prod_{k=j}^{N_u} a_k} - \frac{1}{N_u - i' + 1} \cdot \frac{\tau_i^{N_u - i' - 1} - a_{i'-1}^{N_u - i' + 1}}{\prod_{k=i'}^{N_u} a_k}. \tag{6}$$

$$q_i = \frac{\tau_i}{a_i} + \frac{a_{i'}^{N_u - i' + 1} + a_{i'-1}^{N_u - i' + 1} - 2\tau_i^{N_u - i' - 1}}{(N_u - i' + 1) \prod_{k=i'}^{N_u} a_k} + \sum_{j=1}^{i} \frac{\mathrm{sgn}(j - i')}{N_u - j + 1} \cdot \frac{a_j^{N_u - j + 1} - a_{j-1}^{N_u - j + 1}}{\prod_{k=j}^{N_u} a_k}. \tag{7}$$



(a) The entropy of the binary maps

(b) The optimum threshold

Fig. 2. The entropy of the binary maps and the optimum threshold as a function of the number of users for i.i.d. scheduling metrics.

effective compression. We take this into account via a "run-length indicator" vector $\boldsymbol{\rho}$, see Sections VI-A and VI-B. We also present efficient algorithms for finding the solutions and we give a computational complexity analysis of the schemes.

A. Optimum Scheduling Assignment According to the Conventional Scheme

As discussed in Section III, we first find a binary scheduling map $\mathcal{B}_k$ for each scheduled user k. This map is encoded by run-length encoding whenever this is effective. If run-length encoding is not used, so that the raw binary scheduling map $\mathcal{B}_k$ is sent to user k, then we need N_s bits to represent the scheduling map, that is $N_{\text{map}}^{(k)} = N_s$. Now suppose that run-length encoding is used to encode $\mathcal{B}_k$. In this case, a new entry in the run-length table for $\mathcal{B}_k$ is created when a switch from user k to another user (or conversely, a switch to user k from another user) is performed at some resource block n. Therefore, the required number of bits to represent the binary map $\mathcal{B}_k$ whenever run-length encoding is applied can be written as,

$$N_{\text{map}}^{(k)} = 1 + \lceil \log_2(N_s) \rceil + \sum_{n=2}^{N_s} f_{\text{Conv}}^k(\mathcal{S}_n, \mathcal{S}_{n-1}, n)$$

where $\sum_{n=2}^{N_s} f_{\text{Conv}}^k(\mathcal{S}_n, \mathcal{S}_{n-1}, n)$ is the number of bits added to the table when switching between users occurs, $\lceil \log_2(N_s) \rceil$ is the number of bits required to represent the length of the first run and 1 accounts for extra bit required to determine whether the first resource block is assigned to the user or not.[2]

[2]That is to determine whether the run-length table starts with "0" or "1".

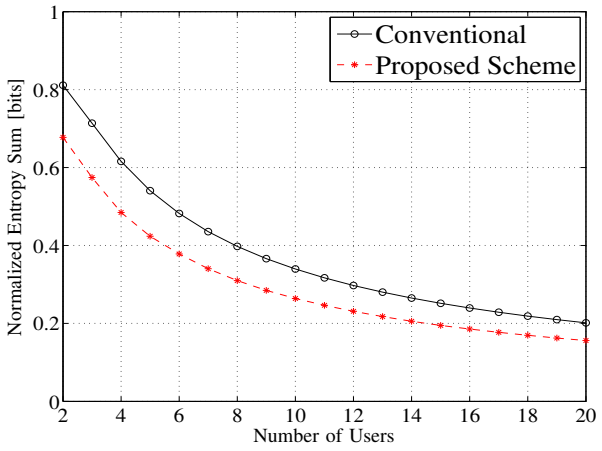


Fig. 3. The normalized entropy sum of the binary maps as a function of the number of users for i.n.d. scheduling metrics.

The switching occurs when (i) $\mathcal{S}_n \neq \mathcal{S}_{n-1}$ and (ii) $\mathcal{S}_n = k$ or $\mathcal{S}_{n-1} = k$. The first case corresponds to a switch to user k from another user whereas the latter case corresponds to a switch from user k to another user. If a switch occurs at position n, we must add $P_n = \lceil \log_2(N_s - n) \rceil$ bits to describe the length of the next active interval.[3] Thus

$$f^k_{\text{Conv}}(i,j,n) = \begin{cases} P_n, & i \neq j \cap (i = k \cup j = k) \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Now, we can finally express the number of bits required to represent the binary scheduling map $\mathcal{B}_k$ as

$$N^{(k)}_{\text{map}} = (1-\rho_k)N_s + \rho_k \left(1 + \lceil \log_2(N_s) \rceil + \sum_{n=2}^{N_s} f^k_{\text{Conv}}(\mathcal{S}_n, \mathcal{S}_{n-1}, n)\right) \tag{9}$$

where ρ_k is used to determine whether run-length encoding is used to encode the data or not. More precisely,

$$\rho_k = \begin{cases} 1, & \text{if run-length encoding is used,} \\ 0, & \text{if run-length encoding is not used.} \end{cases}$$

Therefore we can express the total number of bits required for signaling of the scheduling assignments according to the conventional scheme as

$$N_{\text{Conv}} = \sum_{k=1}^{N_u} \left(N^{(k)}_{\text{map}} + \beta_k + 1 \right).$$

The overhead β_k represents extra bits needed for initialization. Precise models for β_k can be defined for specific systems. For the numerical results to follow, we take $\beta_k = N_{\text{fec}}$, where N_{fec} represents the overhead associated with error detection/protection of the scheduling map.[4] We note that in practical systems, β_k may be different for different k. Now by defining the *run-length indicator* vector $\boldsymbol{\rho} = [\rho_1, \ldots, \rho_{N_u}]$, we can formulate the optimization problem as

$$\min_{\mathcal{S} \in \mathfrak{S}, \boldsymbol{\rho}} N_{\text{Conv}} \Leftrightarrow \min_{\mathcal{S} \in \mathfrak{S}, \boldsymbol{\rho}} \sum_{k=1}^{N_u} \left(N^{(k)}_{\text{map}} + \beta_k + 1 \right) \Leftrightarrow \min_{\mathcal{S} \in \mathfrak{S}, \boldsymbol{\rho}} \sum_{k=1}^{N_u} N^{(k)}_{\text{map}}, \tag{10}$$

where in the last step, we use the fact that the cumulative fixed overhead $\beta = \sum_{k=1}^{N_u} \left(\beta_k + 1\right)$ does not depend on the scheduling assignment and the run-length indicator vector.

This optimization problem can be decoupled into two sub-problems. The first problem finds the optimum scheduling assignments when the run-length indicator vector is known. This problem is a combinatorial problem with $|\mathfrak{S}|$ candidates which as discussed earlier, scales as $N_u^{N_s}$. We provide an efficient algorithm to solve this problem in Section VI-C which requires $\Gamma(N_u, N_s) = 2N_u^3(N_s - 1) + 1$ arithmetic operations.

The second problem concerns finding the optimum run-length indicator vector. This is also a combinatorial problem with 2^{N_u} candidates. The optimum solution can be found through exhaustive search. The complexity of finding the optimum solution is thus $2^{N_u}\Gamma(N_u, N_s) = \mathcal{O}(2^{N_u} N_u^3 N_s)$. We also provide an alternative heuristic algorithm which finds a sub-optimal solution as follows. We assume that run-length encoding is not used to compress the maps of any user, that is $\rho_k = 0$ for $k = 1, \ldots, N_u$. Then we change the run-length encoding indicators of the users one at a time and find the user for which changing the run-length encoding indicator results in the minimum representation cost. Let the ℓth user be that user. To find such a user, we need to search over N_u run-length indicator vectors. If flipping ρ_ℓ to one does not result in a lower binary representation, then we stop the search and assume that the optimum run-length indicator vector is a zero vector (indicating that we should not use run-length encoding for any of the users). Otherwise, if flipping ρ_ℓ to one results in a lower binary representation cost, we fix $\rho_\ell = 1$ and continue to find the second user for which switching to using run-length encoding yields the minimum representation cost (which requires a search over $N_u - 1$ run-length indicator vector candidates) and so on. We continue the search until either flipping the run-length indicator of the users does not decrease the binary representation cost further or all the N_u users are added to the list for which run-length encoding is used to compress the corresponding maps. The worst case computational complexity of finding the solution is thus

$$\sum_{k=0}^{Nu} (N_u - k)\Gamma(N_u, N_s) = \frac{N_u(N_u+1)}{2}\Gamma(N_u, N_s) = \mathcal{O}(N_u^5 N_s). \tag{11}$$

B. Optimum Scheduling Assignment According to Mobile-Assisted Scheduling Scheme

According to the proposed scheme, we send a compressed agreement map $\mathcal{M}_k$ along with a threshold τ_k to each scheduled user k. If run-length encoding is not used to compress $\mathcal{M}_k$, then $M^{(k)}_{\text{map}} = N_s$. Now assume that run-length is used to encode the agreement map $\mathcal{M}_k$. Then a new entry in the run-length table representation of $\mathcal{M}_k$ is created when a switch between "0" and "1" occurs at some resource block n. Based

[3]Since there are N_s resource blocks, the length of the next active interval cannot exceed $N_s - n$ and thus it can be represented by $\lceil \log_2(N_s - n) \rceil$ bits.

[4]In LTE, the control information of each user is protected by a cyclic redundancy check (CRC) of length 16 bits [13]. This CRC is also used to pinpoint the user to which the control data is intended. This is done by using a user specific CRC. Therefore, in the simulation results we assume that $N_{\text{fec}} = 16$.

on the values of $\mathcal{S}_{n-1}$, $\mathcal{S}_n$, $R_{n-1}(k)$ and $R_n(k)$, there are 16 possible cases for half of which a switch occurs. For instance, if both resource blocks $n-1$ and n are assigned to user k, that is if $\mathcal{S}_{n-1}=\mathcal{S}_n=k$, then a switch occurs if either of the following events occurs: (i) $R_{n-1}(k)\geq\tau_k$ but $R_n(k)<\tau_k$, or (ii) $R_{n-1}(k)<\tau_k$ while $R_n(k)\geq\tau_k$, where in the first case the base station disagrees with the proposal of kth user in resource block n and in the latter case, the base station disagrees with her proposal in resource block $n-1$. Table II summarizes all 16 different cases.

As before, when a switch occurs at resource block n, we append P_n bits to represent the next active interval. Based on Table II, we define function $f^k_{\text{MAS}}(i,j,c_1,c_2,t,n)$

$$f^k_{\text{MAS}}(i,j,c_1,c_2,t,n)=\begin{cases}0, & (A\cap B)\cup(\bar{A}\cap\bar{B})\\ P_n, & \text{otherwise}\end{cases} \tag{12}$$

where $A\triangleq\big(i=j=k\big)\cup\big(i\neq k,j\neq k\big)$ and $B\triangleq\big(c_1\geq t,c_2\geq t\big)\cup\big(c_1<t,c_2<t\big)$. We now can write the number of bits required to represent $\mathcal{M}_k$ in the case that run-length encoding is used as

$$M^{(k)}_{\text{map}}=1+\lceil\log_2(N_s)\rceil+\sum_{n=2}^{N_s}f^k_{\text{MAS}}\left(\mathcal{S}_n,\mathcal{S}_{n-1},R_{n-1}(k),R_n(k),\tau_k,n\right)$$

where τ_k is the threshold for user k. We may now express the number of bits required to represent the agreement map $\mathcal{M}_k$ as,

$$M^{(k)}_{\text{map}}=(1-\rho_k)N_s+\rho_k\Bigg(1+\lceil\log_2(N_s)\rceil+\sum_{n=2}^{N_s}f^k_{\text{MAS}}\left(\mathcal{S}_n,\mathcal{S}_{n-1},R_{n-1}(k),R_n(k),\tau_k,n\right)\Bigg) \tag{13}$$

where again ρ_k indicates whether run-length is used to encode $\mathcal{M}_k$ or not. Finally, we can write the total required number of bits for signaling of scheduling assignments as:

$$N_{\text{MAS}}=\sum_{k=1}^{N_u}\Big(M^{(k)}_{\text{map}}+\alpha_k+1\Big). \tag{14}$$

The signaling overhead α_k is $\alpha_k=N_{\text{fec}}+\lceil\log_2(L)\rceil$, where the first term represents the overhead associated with error protection/detection and the second term is the overhead needed to represent the threshold τ_k. We thus formulate the optimal scheduling assignment with mobile-assisted scheduling as

$$\begin{aligned}\min_{\mathcal{S}\in\mathfrak{S},\mathcal{T}\in\mathfrak{T},\boldsymbol{\rho}}N_{\text{MAS}}&\Leftrightarrow\min_{\mathcal{S}\in\mathfrak{S},\mathcal{T}\in\mathfrak{T},\boldsymbol{\rho}}\ \sum_{k=1}^{N_u}\Big(M^{(k)}_{\text{map}}+\alpha_k+1\Big)\\&\Leftrightarrow\min_{\mathcal{S}\in\mathfrak{S},\mathcal{T}\in\mathfrak{T},\boldsymbol{\rho}}\ \sum_{k\in\mathcal{U}}M^{(k)}_{\text{map}},\end{aligned} \tag{15}$$

where $\boldsymbol{\rho}$ is the run-length indicator vector, $\mathcal{T}\triangleq\{\tau_1,\tau_2,\ldots,\tau_{N_u}\}$ and $\mathfrak{T}$ denotes the set of all possible sets of thresholds. Note that again, we have ignored the constant term $\alpha=\sum_{k=1}^{N_u}(\alpha_k+1)$ that does not depend on the scheduling assignment $\mathcal{S}$, the threshold set $\mathcal{T}$ and the run-length indicator vector $\boldsymbol{\rho}$ in the last step.

This optimization problem can be split into three subproblems. The first task is to find the optimum assignments when the threshold set $\mathcal{T}$ and the run-length indicator vector $\boldsymbol{\rho}$ are known. This problem is essentially the same as finding the optimal assignments according to the conventional scheme when the run-length indicator vector is known, see Section VI-C. We give an efficient algorithm for solving this problem that requires $\Gamma(N_u,N_s)$ arithmetic operations.

$\mathcal{S}_{n-1}$	$\mathcal{S}_n$	$R_{n-1}(k)\geq\tau_k$	$R_n(k)\geq\tau_k$	Switch
k	k	yes	yes	$\times$
k	k	no	no	$\times$
k	k	yes	no	$\surd$
k	k	no	yes	$\surd$
$\neq k$	$\neq k$	yes	yes	$\times$
$\neq k$	$\neq k$	no	no	$\times$
$\neq k$	$\neq k$	yes	no	$\surd$
$\neq k$	$\neq k$	no	yes	$\surd$
k	$\neq k$	yes	yes	$\surd$
k	$\neq k$	no	no	$\surd$
k	$\neq k$	yes	no	$\times$
k	$\neq k$	no	yes	$\times$
$\neq k$	k	yes	yes	$\surd$
$\neq k$	k	no	no	$\surd$
$\neq k$	k	yes	no	$\times$
$\neq k$	k	no	yes	$\times$

TABLE II
THE 16 DIFFERENT POSSIBILITIES DURING RUN-LENGTH ENCODING ACCORDING TO MOBILE-ASSISTED SCHEDULING.

The second task is to find the optimum threshold set $\mathcal{T}$. Recall that if run-length encoding is not used to compress the agreement map $\mathcal{M}_k$, then we need N_s bits to represent the map regardless of the choice of the threshold τ_k. Therefore, the threshold for user k can be chosen arbitrarily whenever the run-length indicator vector ρ_k is zero. Alternatively, if run-length encoding is used to compress the agreement map $\mathcal{M}_k$, then there are L possible choices for τ_k. To find the optimum threshold, one needs to search over all L possible choices. Therefore, the total number of combinations of possible threshold sets when the run-length indicator vector is known is L^{N_ρ}, where N_ρ denotes the number of users for which run-length encoding is used (that is the number of ones in the run-length indicator vector $\boldsymbol{\rho}$).

Finally, the third task is to find the optimum run-length indicator vector. This is a combinatorial optimization problem with 2^{N_u} candidates. As discussed above, for each run-length indicator vector $\boldsymbol{\rho}$, one needs to solve a combinatorial problem with the computational complexity of $L^{N_\rho}\Gamma(N_u,N_s)$, where again N_ρ denotes the number of users for which run-length encoding is used. Therefore, the total complexity of finding the optimum solution of (15) is

$$\sum_{k=0}^{N_u}\binom{N_u}{k}L^k\Gamma(N_u,N_s)=(L+1)^{N_u}\Gamma(N_u,N_s)=\mathcal{O}(L^{N_u}N_u^3N_s) \tag{16}$$

where we use the binomial formula to derive the first equality:

$$(x+y)^N=\sum_{k=0}^{N}\binom{N}{k}x^{N-k}y^k.$$

As an alternative for finding the optimum solution, we

provide a heuristic suboptimal algorithm. To find the run-length indicator vector, we use the same suboptimal algorithm as in Section VI-A. For each run-length indicator vector, we find the threshold set using an *iterative local search* (ILS) procedure [14]. More precisely, given a run-length indicator vector $\boldsymbol{\rho}$, we first identify the users for whom run-length encoding should be used to compress their corresponding agreement maps, that is the users for which we would like to find the optimum threshold set. For those users, we pick an arbitrary threshold at random and compute the corresponding optimal scheduling assignment using the algorithm in Section VI-C. Let $\tau^{(0)}$ and $c^{(0)}$ be the threshold set and the corresponding binary representation cost of the optimal scheduling assignments (that is, the required number of bits to represent the scheduling assignments) respectively. We then find the optimal scheduling assignments for all threshold sets that are within a certain distance, say r, of $\tau^{(0)}$, which we refer to as the "neighboring" threshold sets of $\tau^{(0)}$. Now we find the "best" neighboring threshold set of $\tau^{(0)}$ (in the sense that the corresponding optimal scheduling assignments can be represented with the smallest possible number of bits). Let $\tau^{(1)}$ be the best neighboring threshold set of $\tau^{(0)}$ and let $c^{(1)}$ denote the corresponding binary representation of the optimal scheduling assignments. If $c^{(1)}$ is greater than $c^{(0)}$, that is, if all neighbors of $\tau^{(0)}$ require more bits for representation, we stop the algorithm and return $\tau^{(0)}$ as the suboptimal threshold set. Otherwise, we continue the algorithm to search among the neighboring threshold sets of $\tau^{(1)}$ and so on. Algorithm 1 illustrates these procedures.

Algorithm 1 Algorithm to find the threshold set according to the iterative local search (ILS) procedure.

1. Set $k = 0$;
2. Set $\tau^{(k)}$ = a randomly chosen threshold set;
3. Set $c^{(k)}$ = cost associated with optimal scheduling assignment with threshold set $\tau^{(k)}$;
4. Set $k = k + 1$;
5. Set $\tau^{(k)}$ = best neighboring threshold set of $\tau^{(k-1)}$;
6. Set $c^{(k)}$ = cost associated with optimal scheduling assignment with threshold set $\tau^{(k)}$;

if $c^{(k)} < c^{(k-1)}$ **then**
  Goto 4;
else
  return $\tau^{(k-1)}$;
end if

Note that the worst case computational complexity of finding the suboptimal threshold set is also L^{N_ρ}. However due to the specific structure of this problem, the ILS algorithm often does not visit all the candidate solutions and hence it has lower computational complexity. Therefore, the worst case computational complexity of the heuristic algorithm is

$$\sum_{k=0}^{N_u}(N_u-k)L^k\Gamma(N_u,N_s) = \frac{L(L^{N_u}-1)-N_u(L-1)}{(L-1)^2}\Gamma(N_u,N_s) = \mathcal{O}(L^{N_u-1}N_u^3N_s). \tag{17}$$

C. Efficient Algorithm to Solve The First Sub-Problem

Given the run-length indicator vector $\boldsymbol{\rho}$ and the threshold set $\mathcal{T}$, we would like to find the optimum scheduling assignment $\mathcal{S}$ according to (10) and (15) for the conventional and the proposed schemes respectively. Using (9) and (13) and neglecting the terms that do not depend on $\mathcal{S}$, we can rewrite them as,

$$\min_{\mathcal{S}\in\mathfrak{S}} \quad \sum_{k\in\mathcal{U}}\sum_{n=2}^{N_s}\rho_k f^k_{\text{Conv}}(\mathcal{S}_n,\mathcal{S}_{n-1},n) \tag{18}$$

for the conventional approach and

$$\min_{\mathcal{S}\in\mathfrak{S}} \quad \sum_{k\in\mathcal{U}}\sum_{n=2}^{N_s}\rho_k f^k_{\text{MAS}}(\mathcal{S}_n,\mathcal{S}_{n-1},R_{n-1}(k),R_n(k),\tau_k,n) \tag{19}$$

for the proposed scheme respectively. It is worth mentioning that in [5], an algorithm to solve problems with the same structure was given. The algorithm therein is based on dynamic programming [15] with N_u^2 states. However, we show that we can further simplify the algorithm and use dynamic programming with only N_u states.

The key observation is the fact that if two or more scheduling assignments $\mathcal{S}$ exist that yield the same allocation at some resource block n, then we need to only consider the one that requires the minimum representation cost for the following resource blocks. In other words, given that the scheduling assignment is known for resource block n, one can first solve for the optimum scheduling assignment for resource block n', $n' < n$ and then continue to find the optimum assignments for the following resources. The process is reminiscent of maximum-likelihood decoding of a convolutional code. More precisely, we obtain a trellis with N_u states and N_s instances in "time", see Fig. 4. Each state corresponds to $\mathcal{S}_n$ at time n. There are N_u branches emanating from each state. To take into account the fact that at each resource block n, we would like to schedule one of the users with the maximum scheduling metric, we associate a metric to each branch as follows: If $k \in \mathfrak{S}_n$ and $k' \in \mathfrak{S}_{n+1}$, that is if users k and k' are among the users with the maximum scheduling metric at resource blocks n and $n+1$ respectively, then the associated metric for the branch emanating from k going to k' is

$$\sum_{i=1}^{N_u}\rho_i f^i_{\text{conv}}(k,k',n)$$

in the conventional approach and

$$\sum_{i=1}^{N_u}\rho_i f^i_{\text{MAS}}(k,k',R_n(i),R_{n+1}(i+1),\tau_i,n)$$

in the proposed scheme. For other branches, we assign a very large value to make sure that they are not selected. As can be seen, we need at most $2N_u$ operations to compute the metric on the branches emanating from state k and going to state k', when users k and k' are among the users with the maximum scheduling metric at resource blocks n and $n+1$ respectively (assuming all ρ_i:s are not zero). For all other branches, we need only one operations. This means that the number of required operations for finding the trellis is

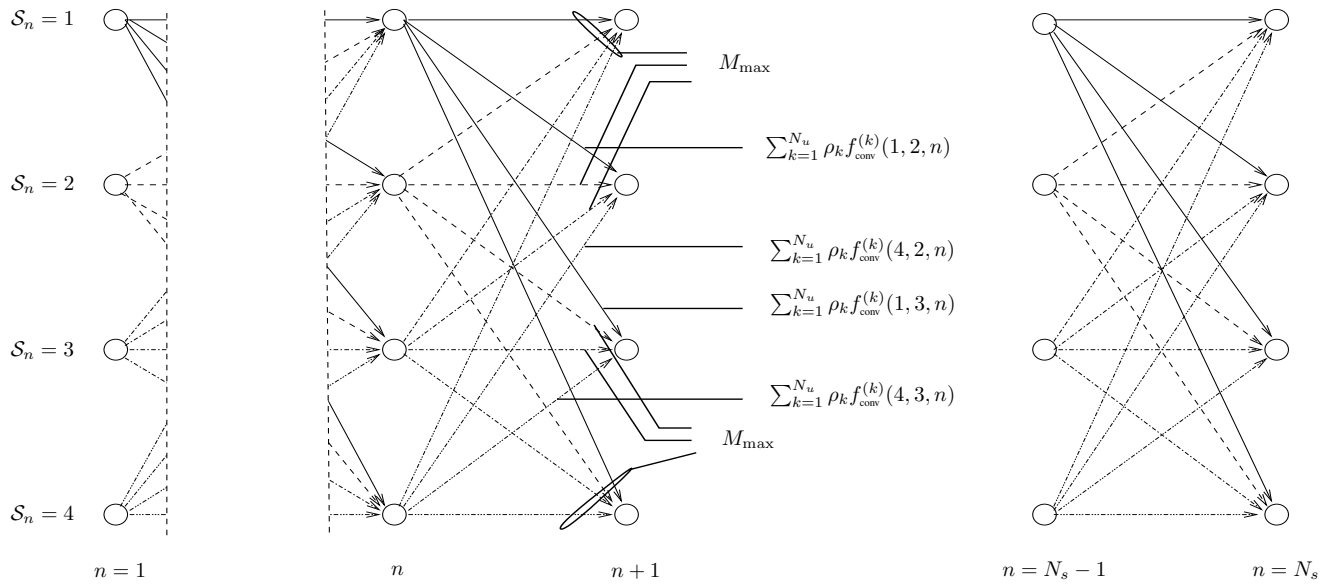


Fig. 4. Illustration of the trellis for the algorithm in Section VI-C for the case with four users ($\mathcal{U} = \{1, 2, 3, 4\}$). The assumption is that $\mathfrak{S}_n = \{1, 4\}$ and $\mathfrak{S}_{n+1} = \{2, 3\}$. The figure only shows excerpts of the trellis. Some randomly chosen branch metrics are also illustrated for the conventional scheme.

$2N_u^3(N_s - 1)$. Once the trellis has been found, a backtracing (Viterbi-like) algorithm is used to find the shortest path trough the trellis. Therefore, the total number of required operations is $\Gamma(N_u, N_s) \triangleq 2N_u^3(N_s - 1) + 1 = \mathcal{O}(N_u^3 N_s)$.

VII. Numerical Examples

To illustrate the performance gain of the proposed scheme, we present simulation results for an LTE-like OFDM system with 20 MHz bandwidth, 15 KHz subcarrier spacing, and 12 consecutive subcarriers per resource block resulting in $N_s = 110$ resource blocks (plus some guard bands). We assume a max C/I scheduler and we use three different channel models:

- (i) i.i.d. Rayleigh fading,
- (ii) i.i.d. with channels drawn according to the Vehicular A tapped-delay line model defined by the International Telecommunications Union (ITU) standard [16], and
- (iii) i.n.d. with channels drawn from either of the following three channel models with equal probability: (1) independent Rayleigh fading, (2) the ITU Vehicular A model, and (3) the ITU Vehicular B model [16].

The first two models represent the case with i.i.d. scheduling metrics while the third model represents the independent but not-identically distributed scheduling metrics. For the ILS algorithm, we assume $r = 1$. The results are averaged over 3200 channel realizations.

Fig. 5 and 6 show the number of bits required to represent the scheduling assignments for the schemes above as a function of the number of users for channel model (i). The number of quantization levels are $L = 2$ and $L = 4$ respectively. Here, a gain of roughly 16% is achieved for both cases. Fig. 7 and 8 illustrate the number of bits required to represent the scheduling assignments for the schemes above for channel models (ii) and (iii) with $L = 4$ respectively. We see that the proposed scheme reduces the signaling overhead by roughly 6% and 10% respectively. We observe that for

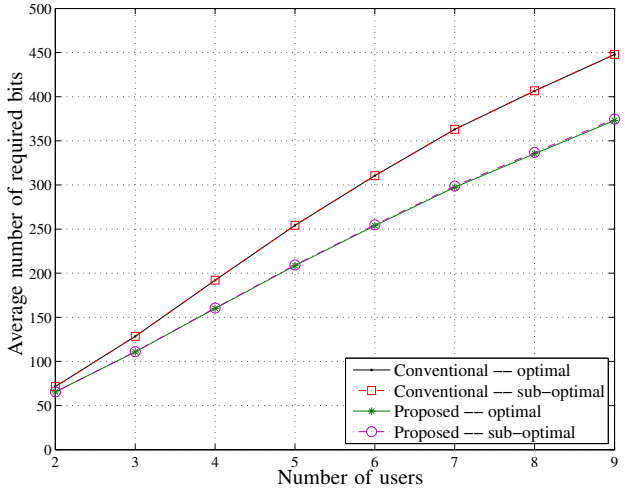


Fig. 5. Comparison of the two schemes in terms of the average number of bits required to represent the scheduling assignments, for an i.i.d. Rayleigh fading channel and $L = 2$ CSI quantization levels.

channel models (ii) and (iii), the signaling overhead is lower than that for channel model (i) (i.i.d. Rayleigh fading) for both the conventional scheme and the proposed mobile-assisted scheduling scheme. This is so because for channel model (i), the channels offer more multiuser diversity and hence the scheduling maps are more complex than they are for models (ii) and (iii). Therefore, run-length encoding is used less often with model (i) compared to with models (ii) and (iii). Note that, in all cases, the heuristic algorithms for finding the solutions perform nearly as well as the exhaustive search which always finds the global optimum.

As a measure of computational complexity, we count the number of operations required by each signaling scheme. As discussed earlier, to find the optimum solution according to the conventional scheme, the first sub-problem needs to be solved

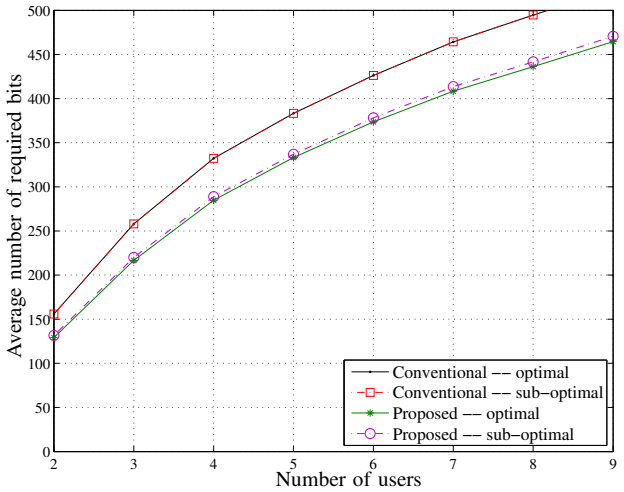


Fig. 6. Same as Fig. 5 but for $L = 4$.

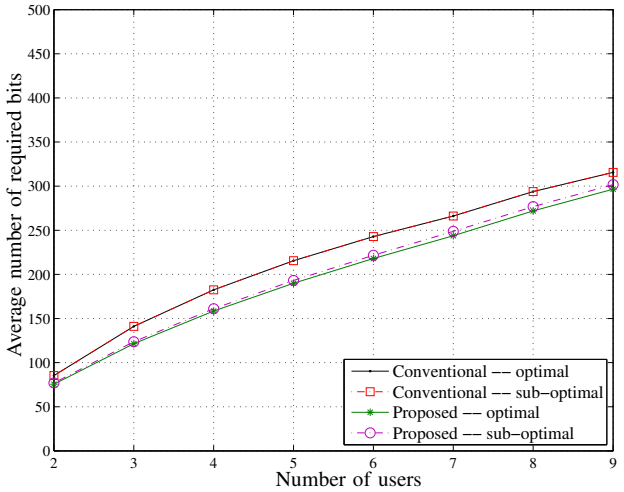


Fig. 7. Same as Fig. 6 but for an ITU-Vehicular A channel.

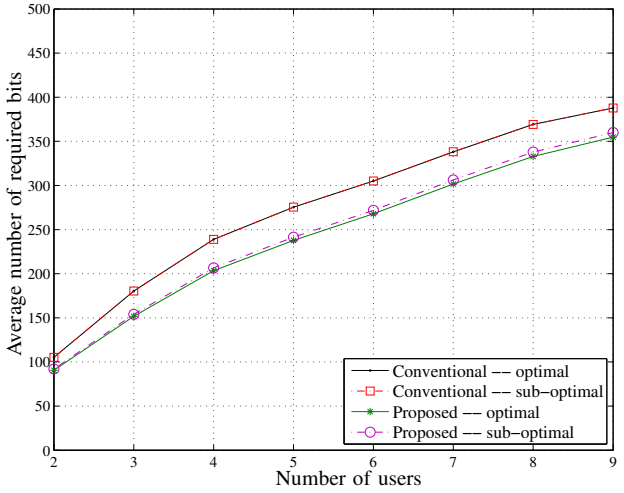


Fig. 8. Same as Fig. 6 but for i.n.d. channel model.

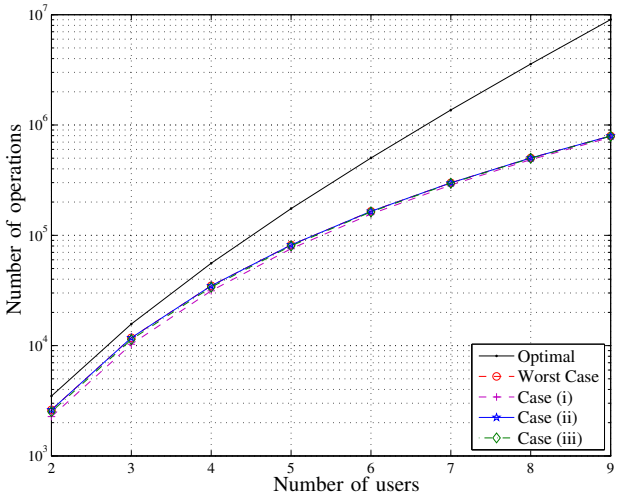


Fig. 9. Average number of required operations according to the conventional approach for different cases as a function of the number of users.

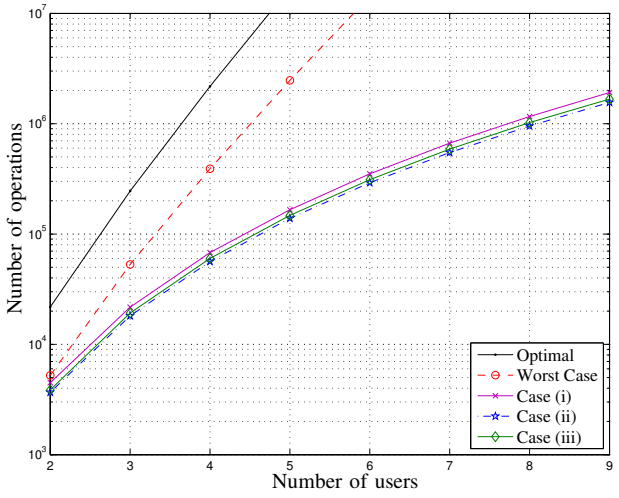


Fig. 10. Average number of required operations according to the proposed scheme for different cases as a function of the number of users.

for all the possible run-length-indicator vectors. That is, the complexity of finding the optimum solution is $2^{N_u}\Gamma(N_u, N_s)$. For the sub-optimal solution, the number of searches is upper bounded by $N_u(N_u+1)/2$ (see (11)). However, the number of searches might be smaller in practice. This can be seen from Fig. 9, where the average number of required operations is plotted for the conventional scheme. As we see, for channel models (ii) and (iii), the average value is the same as for the worst case scenario. This shows that the run-length encoding is used more often in these cases (as we discussed above for Fig. 7 and 8).

For the proposed scheme, finding the optimum solution requires that the first sub-problem is solved $(L+1)^{N_u}$ times (see (16)). The sub-optimal solution, on the other hand, has a worst-case computational complexity given by (17), which is also exponential in the number of users. However, the average complexity is much smaller, as can be seen from Fig. 10. Note that we assume $L = 4$, for the optimal and worst-case computational complexity curves. As we see, the average

complexity of the sub-optimal solution is not exponential in the number of users.

On a final note we again stress that the optimization problems need to be solved at the beginning of each frame. In LTE, the duration of a frame is 1 ms. Therefore, if for instance there are 9 active users in the cell, we need a processor that can operate at the speed of roughly 2 GFlops at the base station, in order for our schemes to work.

VIII. Future Work and Conclusion

We presented a new scheme for encoding of scheduling assignments in wireless multiple access systems. The scheme exploits the fact that the users are aware of their own channel conditions and hence expect to receive data in certain resource blocks. The process is reminiscent of compression with side information. The scheme can reduce the control signaling overhead by about 20%. We also provided an efficient algorithm to find the optimum scheduling assignments.

In the presented scheme, we did not consider the actual cost associated with the transmission of the scheduling assignments. Generally, in practical systems there will always be users that have poor channel conditions. The transmission of scheduling information to those users can be very costly in terms of radio resources needed. Taking the transmission cost into account, it is occasionally more beneficial to grant more of the scheduling proposals from the users who have poor channel conditions. For example, by simply agreeing to the entire proposal from a user with a poor channel, the base station would not need to send any scheduling (nor agreement) map at all to those users,[5] and thus the scheduling cost could be significantly reduced. However in doing so, the base station may need to compromise the throughput-optimality of the resulting scheduling assignments. The proposed scheme may be extended in this direction.

References

[1] R. Moosavi and E. G. Larsson, "Reducing downlink signaling traffic in wireless systems using mobile-assisted scheduling", *in Proc. of IEEE GLOBECOM*, pp. 1-5, Dec. 2010.

[2] A. J. Goldsmith and S. G. Chua, "Adaptive coded modulation for fading channels", *IEEE Trans. Commun.*, vol. 46, no. 5, pp. 595-602, May 1998.

[3] S. Lin, D. J. Costello and M. Miller, "Automatic-repeat-request error-control schemes," *IEEE Commun. Mag.*, vol. 22, pp. 5-16, Dec. 1984.

[4] J. Gross, H. F. Geerdes, H. Karl and A. Wolisz, "Performance analysis of dynamic OFDMA systems with inband signaling," *IEEE J. Select. Areas Commun.*, vol. 24, pp. 427-436, Mar. 2006.

[5] E. G. Larsson, "Optimal OFDMA downlink scheduling under a control signaling cost constraint", *IEEE Trans. Commun.*, vol. 58, pp. 2776-2781, Sep. 2010.

[6] R. Moosavi, J. Eriksson, E. G. Larsson, N. Wiberg, P. Frenger and F. Gunnarsson, "Comparison of strategies for signaling of scheduling assignments in wireless OFDMA," *IEEE Trans. Veh. Technol.*, vol. 59, pp. 4527-4542, Nov. 2010.

[7] E. Dahlman, S. Parkvall, J. Sköld and P. Beming, *3G Evolution HSPA and LTE for Mobile Broadband*, 2nd edition Academic Press, 2008.

[8] H. Nguyen, J. Brouet, V. Kumar and T. Lestable, "Compression of associated signaling for adaptive multicarrier systems," *in Proc. of IEEE VTC*, pp. 1916-1919, May 2004.

[9] M. Sternad, T. Svensson and M. Dottling, "Resource allocation and control signaling in the WINNER flexible MAC concept," *in Proc. of IEEE VTC*, pp. 1-5, Sep. 2008.

[10] R. Moosavi, J. Eriksson and E. G. Larsson, "Differential signaling of scheduling information in wireless multiple access systems," *in Proc. of IEEE GLOBECOM*, pp. 1-6, Dec. 2010.

[11] D. Stiliadis and A. Varma, "Efficient fair queueing algorithms for packet-switched networks," *IEEE/ACM Trans. Networking*, vol. 6, pp. 175-185, Apr. 1998.

[12] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd Edition A Wiley-Interscience Publication, 2005.

[13] 3GPP TS 36.212, Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding, Dec. 2009.

[14] R. K. Congram, C. N. Potts and S. L. Van de Velde, "An iterated dynasearch algorithm for the single machine total weighted tardiness scheduling problem," *Info. J. Computing*, Vol. 14, No. 1, pp. 52-67, 2002.

[15] D. Bertsekas, *Dynamic Programming and Optimal Control*, Athena Scientific Publication, 2007.

[16] ITU-R Recommendation M.1225, "Guidelines for evaluation of radio transmission technologies for IMT-2000," 1997.

[5]Except for a flag bit needed to indicate the acceptance of their proposals, and the associated value of the threshold.

Reza Moosavi received his B.Sc. in Electrical Engineering from Isfahan University of Technology, Isfahan, Iran in 2005 and his M.Sc. from Chalmers University of Technology, Göteborg, Sweden, in 2008. Since February 2009, he is a Ph.D student at the Communication Systems Division of the Department of Electrical Engineering, Linkping University, Sweden. His research interest include resource allocation and signaling protocols in cellular systems.

Erik G. Larsson received his Ph.D. degree from Uppsala University, Sweden, in 2002. Since 2007, he is Professor and Head of the Division for Communication Systems in the Department of Electrical Engineering (ISY) at Linköping University (LiU) in Linköping, Sweden. He has previously been Associate Professor (Docent) at the Royal Institute of Technology (KTH) in Stockholm, Sweden, and Assistant Professor at the University of Florida and the George Washington University, USA.

His main professional interests are within the areas of wireless communications and signal processing. He has published some 80 journal papers on these topics, he is co-author of the textbook *Space-Time Block Coding for Wireless Communications* (Cambridge Univ. Press, 2003) and he holds 10 patents on wireless technology.

He is Associate Editor for the *IEEE Transactions on Communications* and he has previously been Associate Editor for several other IEEE journals. He is a member of the IEEE Signal Processing Society SAM and SPCOM technical committees. He is active in conference organization, most recently as the Technical Chair of the Asilomar Conference on Signals, Systems and Computers 2012 and Technical Program co-chair of the International Symposium on Turbo Codes and Iterative Information Processing 2012.