Multi-Dimensional OFDMA Scheduling in a Wireless Network with Relay Nodes

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

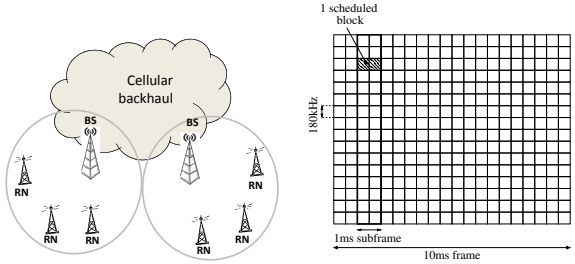
***Abstract*—LTE-advanced and other 4G cellular standards allow relay nodes (RNs) to be deployed as a substitute for base stations (BSs). Unlike a BS, an RN is not directly connected to the backbone. Rather, each RN is associated with a donor BS, to which it is connected through the OFDMA wireless link. A very important task in the operation of a wireless network is packet scheduling. In a network with RNs, such scheduling decisions must be made in each cell not only for the BS, but also for the RNs. Because the scheduler in a network with RNs must take into account the transmission resources of the BS and the RNs, it needs to find a feasible schedule that does not exceed the resources of a multi-dimensional resource pool. This makes the scheduling problem computationally harder than in a network without RNs. In this paper we define and study the *packet-level* scheduling problem for a network with RNs. This problem is not only NP-hard, but also admits no efficient polynomial-time approximation scheme. To solve it, we propose an efficient algorithm with a performance guarantee, and a simple water-filling heuristic. To the best of our knowledge, our algorithm is the first *packet-level* scheduling algorithm that provides a performance guarantee for a network with RNs. Using simulations, we evaluate our new algorithms and show that they perform very well.**

I. Introduction

The advent of sophisticated mobile devices and new applications has made spectral optimization crucial for wireless networks. New 4G technologies, such as LTE Advanced [2], employ OFDMA in their physical layer and use new concepts such as MIMO, CoMP and Relay Nodes (RNs) [4], [22], [23], [25], [26], [36] to increase the throughput.

Deploying long-range wireless networks with good coverage is a complex task, one that introduces a trade-off between cost and performance. One example of this trade-off is the desire to decrease the size of the cells in order to increase the network bandwidth available to every user. But decreasing cell size by adding more base stations (BSs) increases installation costs substantially, because the most expensive factor in the installation of a new BS is connecting it to the optical backbone.

To overcome this barrier, 4G cellular standards allow RNs to be deployed as a substitute for BSs. Unlike a BS, an RN is not directly connected to the backbone. Rather, each RN is associated with a donor BS, to which it is connected through the OFDMA wireless link (see Figure 1(a)). Each user equipment (UE) receives its data packets either directly from the BS, or indirectly over the BS→RN→UE route. The performance benefits from the deployment of RNs are three-fold: (a) increased network density; (b) increased network coverage; (c) increased network roll-out speed.



(a) (b)

Fig. 1. (a) Example of a network with RNs and their donor BSs; and (b) An abstract structure of the LTE frame

An important task in the operation of a wireless network is packet scheduling. This task comprises all real-time decisions that must be made by the BS before transmitting data on the downlink channel: which data packets to transmit during the next OFDMA subframe, which modulation and coding scheme (MCS) to use for each packet, whether to transmit a packet directly to the UE or via an RN, and so on. In a network with RNs, such scheduling decisions must be made for the RNs as well. In this paper we propose the first packet-level scheduling algorithm for such networks. While a user-level scheduling algorithm [26], [31], [37] is used for admission control, as part of the handover procedure, a packet-level scheduling algorithm is needed for deciding which packet to transmit and using what MCS.

Adding RNs to the network makes the scheduling problem computationally harder. Without RNs, the BS needs to decide which packets to transmit and which MCS to use for each transmitted packet. Each transmission of a packet using some MCS requires a cer-

tain amount of bandwidth in the next subframe and is associated with a certain utility function. The goal is to maximize the total profit without exceeding the total bandwidth. Therefore, without RNs, the scheduling problem is equivalent to the known NP-hard Multiple-Choice Knapsack Problem (MCKP) [18], to which excellent approximations, heuristics and dynamic programming algorithms exist.

In a network with RNs, the scheduler must also take into account the bandwidth available to each RN. Thus, each packet transmission now has a 2-dimensional size: the first dimension indicates the bandwidth resources required for the BS→RN transmission and the second indicates the bandwidth resources required for the RN →UE transmission. Thus, the scheduler must find a feasible schedule that does not exceed the resources of a *multi-dimensional resource pool*, whose number of dimensions depends on the number of RNs. This makes the scheduling problem in a network with RNs more similar to an extension of MCKP into multiple dimensions, a problem known as d-dimensional Multiple-Choice Knapsack (d-MCKP), which is computationally harder than MCKP. In order to solve this problem for a network with RNs, we transform it into a less general case of d-MCKP called sparse d-MCKP, and propose efficient algorithms to solve this new problem. One of our algorithms is proven to have a performance guarantee, and can also be optimal for realistic input size.

For ease of presentation, we explain the main concepts of our proposed algorithms for a BS with one omnidirectional antenna, although in many cellular networks that employ RNs the BS uses multiple directional antennas (also known as sectors). For such multi-sector networks, the algorithms proposed in this paper can be invoked independently for each sector, in which case the BS in each sector would also run an independent scheduler, for its directional antenna and for each RN in its sector. If this option is chosen, no changes are required to the proposed algorithms.

The rest of the paper is organized as follows. In Section II we discuss related work. In Section III we present our scheduling network. In Section IV, we define the new "OFDMA Scheduling with Relays and Dynamic MCS Selection" problem, which is the core of this paper. We show that it is NP-hard and equivalent to a special case of d-MCKP. In Section V we present efficient algorithms for solving this new problem. In Section VI we show how to adapt our algorithms to inband relaying. Section VII presents an extensive simulation study and Section VIII concludes the paper.

Table I summarizes the main abbreviations and acronyms used throughout the paper.

Acronym	Meaning
BS	Base Station
CQI	Channel Quality Indication
d-KP	d-dimensional Knapsack Problem
d-MCKP	d-dimensional Multiple-Choice Knapsack problem
GAP	Generalized Assignment Problem
MCKP	Multiple-Choice Knapsack Problem
MCS	Modulation and Coding Scheme
RN	Relay Node
SINR	Signal to Interference plus Noise Ratio
UE	User Equipment (the mobile host)

TABLE I
ABBREVIATIONS AND ACRONYMS USED IN THE PAPER

II. Related Work

Our paper is the first to propose **packet-level** scheduling algorithms for an OFDMA/LTE network with relay nodes (RNs). We therefore classify the papers described in this section into two groups. The first includes papers that propose packet-level scheduling for an OFDMA/LTE network without RNs. The second includes papers that address scheduling related issues in a network with RNs.

Papers belonging to the first group are [5], [10], [11], [30]. The most relevant work is probably our recent paper on joint scheduling in OFDMA networks [11]. Both that paper and this one propose a packet level scheduling algorithm to be employed by a scheduling logic at the BS once every OFDMA subframe. Both papers solve the most basic and most important scheduling question: which transmitter will transmit which packet and using what modulation and coding scheme (MCS). However, the two papers solve different problems. In [11], the scheduling decisions are made for multiple independent sectors. This problem is shown to be related to the theoretical Generalized Assignment Problem (GAP). In contrast, in this paper, scheduling decisions are made for a BS and its RNs. The resulting problem is shown to be more similar to the theoretical d-dimensional Knapsack Problem (d-KP) or d-MCKP problem. This is because every packet to be transmitted via an RN must be allocated resources from 2 pools, and these packets can be transmitted using any one of several possible MCSs. This makes for a different and computationally harder problem than GAP.

Papers in the second group include [26], [31], and [37]. In these papers, layer-2 user-level admission control algorithms are proposed for OFDMA networks with relays. In [26], the authors consider a cell with RNs, and assume that there is no direct wireless link between the BS and the UEs. The UEs are either delay sensitive or non-delay sensitive, and algorithms that select one of four possible transmission strategies for each UE are presented. In [31], an algorithm for maximizing the total cell throughput while stabilizing user queues is proposed. In [37], two algorithms for

utilizing spatial reuse are developed and are shown to improve the throughput. All these papers address the problem of deciding the transmission rates of the BS and RNs to each user. However, we distinguish between this "user-level admission control" problem, and our "packet-level RN scheduling" problem, mainly because in our model different packets of the same user might have different priority. Another important difference is that we allow different packets of the same user to be transmitted using different MCS, while in above-mentioned works the same MCS is determined for each user.

In [32], the throughput of a network with RNs is improved through adaptive frame segmentation and employing spatial reuse by RNs. In [14], the implementation aspects and constraints of the simplest network coding schemes for a two-way relay channel in LTE is considered.

In [7], [17], and [25], relay strategies are compared. In [25], the downlink performance for layer-3 and layer-1 relays is investigated. System-level simulations are used to demonstrate the impact of several relay conditions. In [29], the performance of several emerging half-duplex relay strategies in interference-limited cellular systems is analyzed. The performance of each strategy as a function of location, sectoring, and frequency reuse are compared with localized base station coordination. In [17], the performance of an infrastructure based multi-antenna relay network in the absence of a direct link is studied. As expected, their results show that the performance depends on the location of the RNs. Finally, in [7], the authors evaluate relay based heterogeneous deployment within the LTE-Advanced uplink framework. Different power control optimization strategies are proposed for 3GPP urban and suburban scenarios.

III. Network Model

A. Inband vs. Outband Relaying

We consider a cell with a BS at its center and R RNs, as shown in Figure 3(b) for $R = 3$ (frequency reuse aspects of this figure are discussed later on). Figure 1(b) shows a schematic structure of a 10-ms LTE frame, divided into 10 1-ms subframes[1].

Each RN is connected to its BS by an OFDMA wireless link, using either inband or outband relaying. In outband relaying, BS and RN transmissions use different subbands. Therefore, they can transmit simultaneously in each subframe, with no interference (Figure 2(a)). In inband relaying, however, the transmissions from the BS to the RNs or to the UEs are performed over the same subbands as those from the RNs to the UEs. Thus, simultaneous transmissions by the BS and RNs are not possible unless sufficient isolation in time or in space is ensured. Figure 2(b) assumes such isolation: in every two consecutive subframes, one is dedicated for transmissions from the BS and one for transmissions by the BS and RNs to UEs (isolation in time). The BS and the RNs can transmit together in every second subframe only if they are located far enough from each other (isolation in space). Otherwise, only the RNs can transmit in every second subframe.

[1]We are trying to abstract the problem in the most generic way. Therefore, we skip some of the LTE physical layer details that are not directly relevant to the description of the problem and algorithms.

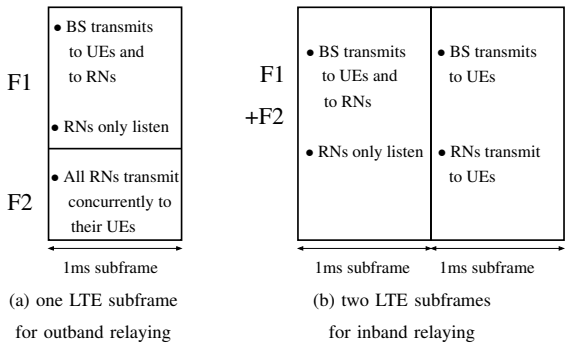


Fig. 2. An abstract structure of the LTE subframe (F1 and F2 are two orthogonal OFDMA subbands)

B. Our Scheduling Model

In an LTE network with RNs, one may distinguish between distributed and centralized scheduling. In distributed scheduling, each transmission entity, namely, a BS or an RN, autonomously decides what to transmit in every subframe. In centralized scheduling, all transmission decisions for the BS and the RNs are performed by the BS. In this paper we focus on centralized scheduling, because it has an important advantage over distributed scheduling [16]: the scheduler has a global view of the network resources and can optimize their usage. For instance, if an RN is overloaded, the BS can decide to transmit to some UEs directly, even if these UEs have better SINR with the busy RN than with the BS.

In the considered model, the BS receives periodic Channel Quality Indications (CQI) [12] from the UEs and RNs. Using these reports, the BS is able to estimate the SINR for transmissions from the BS to each UE or RN. The BS also receives CQI reports on the SINR between each UE and its RN. These reports are either transmitted directly by every UE to the BS, or forwarded by the RNs to the BS over an RN→BS control channel. The BS uses this information to make the following decisions: (a) which packet to transmit; (b) whether to transmit the packet directly or through an RN; (c) if the packet is not transmitted directly, through which RN to forward it; (d) which MCS (Modulation and Coding Scheme) to use for each transmission.

The scheduler determines how many scheduled blocks to allocate to every packet according to the chosen MCS. Some MCSs are more efficient, i.e., require fewer

scheduled blocks, but are less robust to transmission errors. Other MCSs are less efficient but more robust. Since there are several "pools" of scheduled blocks that the scheduler uses, a more formal discussion will require the following definitions:

Definition 1: A **scheduling zone** is a set of scheduled blocks to be assigned for transmission by the same transmission entity (a BS or an RN).

In the outband relaying model, the scheduler needs to make a scheduling decision every 1ms subframe for 1 BS scheduling zone and R RN scheduling zones (Figure 2(a)). Thus, the scheduler has to allocate resources from $R+1$ scheduling zones (pools) every 1ms. In the inband relaying model, the scheduler needs to make a scheduling decision every 2ms for two consecutive 1ms subframes (Figure 2(b)). In the first 1ms subframe, the scheduler allocates resources only from the BS scheduling zone. In the second 1ms subframe, the scheduler allocates resources from the BS scheduling zone and R RN scheduling zones. Thus, in the inband relaying model, the scheduler has to make decisions for $R+2$ scheduling zones every 2ms.

Definition 2: A **transmission instance** of a packet is a triple [packet, path, MCSs], where path is either BS→UE or BS→RN$_i$→UE, and MCSs is a list that indicates the MCS to be used for the transmission of the packet over each link along the path (1 link if the path is BS→UE; 2 links if it is BS→RN$_i$→UE). Each transmission instance requires allocation of scheduled blocks from the corresponding scheduling zone(s).

We adopt the profit-based scheduling model proposed in [13]. Thus, each transmission instance of a data packet at time t is associated with a profit and a cost. The profit depends on the following parameters: (a) how important it is to the application that the packet be delivered at t; (b) the probability that this packet will be successfully received by the UE. This probability can be computed by the BS by taking into account (i) the SINR on each wireless link (BS→UE or BS→RN$_i$ and RN$_i$→UE); (ii) the length of the packet; and (iii) the MCS used for transmitting this packet [6], [24].

We now give examples of concrete profit values whose aim is to optimize either the throughput, energy, delay, or fairness.

$\boldsymbol{p}_{\textbf{packets}}$ - This profit value is defined as the packet transmission success probability. As a result, the profit sum of all the packets transmitted in a given subframe is equal to the expected number of successfully received packets, which is the packet-level throughput.

$\boldsymbol{p}_{\textbf{throughput}}$ - This profit value is defined as p_{packets} multiplied by the length of the packet. As a result, the sum of all profit values of all transmitted packets equals the expected number of successfully received bits, i.e., bit-level throughput.

$\boldsymbol{p}_{\textbf{energy}}$ - This profit value is defined as $p_{\text{throughput}}$ divided by the transmission energy cost. As a result, the sum of all profit values of all transmitted packets equals the expected number of bits transmitted per energy unit, namely, the transmission energy utilization.

$\boldsymbol{p}_{\textbf{delay}}$ - For each packet, this profit value indicates the probability that this packet will be delivered on time if it is transmitted during the next subframe. As a result, the sum of the profit is equal to the expected number of packets scheduled in a given subframe that will be delivered on time.

$\boldsymbol{p}_{\textbf{pf}}$ - For each user, the most urgent packet destined for this user is assigned a profit value of $\log(p_{\text{throughput}})$. The profit for all remaining packets is set to zero. It is shown in [19] that an allocation that maximizes $\sum \log R_u$, where R_u is the rate of user u, is proportional fair. As a result, a proportional fair allocation is one that maximizes $\sum p_{\text{pf}}$.

The success probability for transmitting a given packet varies from one scheduling zone to another. Thus, the profit associated with a packet transmission can be different for different scheduling zones. Moreover, the profit of a given packet in a given scheduling zone might also change between two consecutive 1ms subframes, e.g., due to user mobility.

While the profit of a packet is a scalar, the cost is a vector that has one or more dimensions: one for each link over which the packet is scheduled. The cost on each link is equal to the number of scheduled blocks required for transmitting the packet in the scheduling zone associated with this link, which depends on the user SINR in that zone, on the length of the packet, and on the chosen MCS. The fact that the cost of a packet is a vector is what makes the scheduling problem much more difficult than the NP-hard Multiple-Choice Knapsack Problem (MCKP), for which very good approximations exist.

C. Frequency Reuse Models

In addition to the decision whether to use inband or outband relaying, the frequency reuse model must also be decided upon. In order to describe our algorithms in a specific context, we focus on two models. However, these algorithms are easily adaptable to other frequency reuse models as well. The first model, called model-1, is shown in Figure 3(a) and is relevant for outband relaying. Here, bandwidth is partitioned into $N+1$ subbands: F0, F1, F2 and F3 ($N=3$ in this figure). The BS in every cell uses subband F0 (i.e., the BSs work using frequency reuse 1), while all the RNs in every cell use either F1, F2 or F3. This guarantees that close RNs in neighboring cells use different subbands. This combination of reuse-1 by the BSs and reuse $1/3$ by the RNs can be viewed as an implementation of FFR (Fractional Frequency Reuse), which is very common in networks with no RNs [9],

[21], [27], [35]. Since outband relaying is considered for this model, the BSs and RNs use different orthogonal subbands. Thus, the BSs transmit using high power, and they can reach the cell-edge UEs with no interference from/to their RNs.

The second model, called model-2, is shown in Figure 3(b) and is relevant for inband relaying. This model employs a complete reuse-1, where every BS and every RN uses all subbands. To avoid interference with their RNs, the BSs transmit using lower power than in model-1. But this power is sufficient to allow each BS to reach its RNs with good SINR. The transmission power of the RNs is strong enough to reach their cell edge UEs, but not so strong so as to interfere with other RNs in the same or adjacent cell.

We emphasize that this paper does not claim that the considered frequency reuse model is the best for an LTE network with RNs. The decision about which model to use depends on many factors and regulations that are beyond the scope of this paper. We use model-1 and model-2 because we believe that they are general enough for presenting our ideas and algorithms in a concrete context.

IV. The Scheduling Problem

This section is divided into two subsections. In the first subsection, we define the scheduling problem in OFDMA networks with RNs and show hardness results. In the second subsection, we define a **new theoretical problem** called sparse d-MCKP and show that it is equivalent to our OFDMA scheduling problem.

A. Preliminaries

Theoretically, each user can receive its packets from the BS or from any RN. However, as proven below, a user may have a reasonable SINR from at most one RN. Therefore, the scheduler transmits a packet to a user either directly from the BS or through the RN with which the user has the best SINR. This RN is referred to as the **default RN** of the considered user.

Lemma 1: (a) The user may have an SINR>0dB only from the default RN (SINR >0dB is chosen because transmission success probability for SINR no greater than 0dB is very low [6]); (b) each packet can be associated with at most (M^2+M) transmission instances, where M is the number of MCSs.

Proof: The SINR of a UE for a packet received from a BS or an RN is equal to $\frac{S}{I+N_0}$, where S is the received power at this UE from the transmitter, I is the interference power of other simultaneous transmissions, and N_0 is the noise power. In both model-1 and model-2, all RNs in the same cell transmit using the same subbands. Therefore, when calculating the SINR of a UE for RN_k, $S = P_k$ and $I \geq \sum_{j \neq k} P_j$, where P_j is the power of the signal received by this UE from RN_j. This implies that for each UE there can be only one RN for which the SINR > 1. This RN is the one for which the received power at this UE is the highest, since in the expression of the SINR for any other RN this power is considered as interference. To prove (b), note that by (a) a packet is either transmitted directly, or through a specific RN, say RN_i. In the first case, the MCS to be used on the BS→UE link is one of the M possible MCSs. In the latter case, there are M MCSs for the link BS→RN_i, and M for the link RN_i→UE. Thus, there are at most M^2 possible combinations, and the total number of transmission instances is (M^2+M). ∎

We now define the "OFDMA Scheduling with Relays and Dynamic MCS Selection" problem, which is the core of this paper.

Problem 1 (OFDMA Scheduling with Relays and Dynamic MCS Selection)

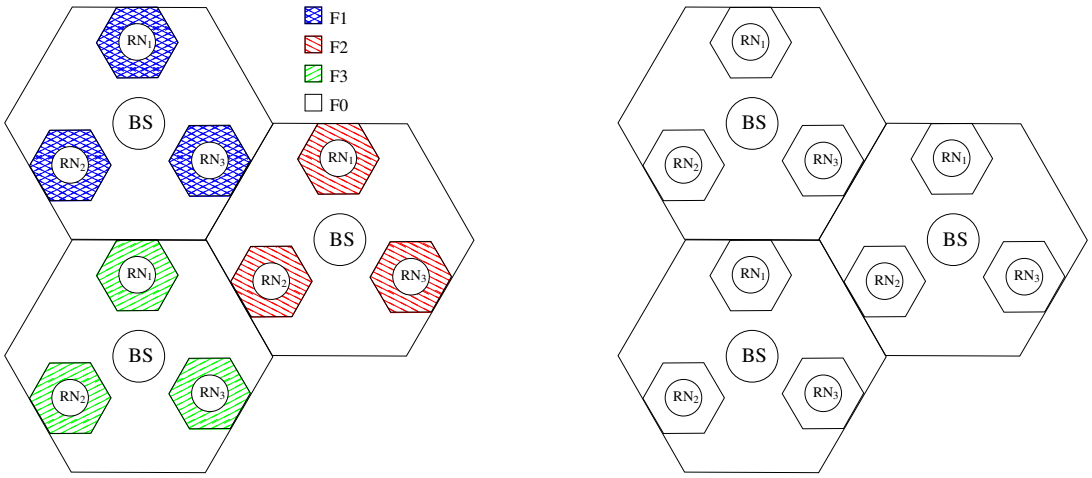
Instance: The scheduler is given the number of scheduled blocks to be allocated in each scheduling zone. For each packet_i, the scheduler determines the RN with which the UE has the best SINR, say RN_j. It then considers at most $(M + M^2)$ transmission instances for transmitting this packet to the UE. M instances are for the direct BS→UE transmission and M^2 for transmissions through RN_j, where M is the number of MCSs. Each transmission instance is associated with a profit and with a 2-dimensional size: one that indicates the number of scheduled blocks for the transmission by the BS, and one that indicates the number of scheduled blocks for the transmission by the default RN. The latter is 0 if the packet is transmitted over the BS→UE path.

Objective: Find a feasible schedule that maximizes the total profit. A feasible schedule is one for which the number of scheduled blocks available in each scheduling zone is not exceeded. ∎

As an example, consider a BS that has 3 packets waiting for transmission: packet_1, packet_2 and packet_3 to UE_1, UE_2 and UE_3 respectively. Suppose that the default RNs for these UEs are RN_1, RN_2 and RN_3 respectively. Examples for two possible schedules are:

- *(schedule 1)* packet_1 is transmitted using MCS-1 to RN_1 and then using MCS-2 to UE_1; packet_2 is transmitted using MCS-1 to RN_2 and then using MCS-1 to UE_2; packet_3 is transmitted using MCS-3 directly to UE_3;
- *(schedule 2)* packet_1 is transmitted using MCS-1 directly to UE_1; packet_2 is transmitted using MCS-2 to RN_2 and then using MCS-1 to UE_2; packet_3 is not transmitted (it might either be transmitted during one of the next subframes or dropped by the BS due to lack of bandwidth).

Technically, there are (M^2+M) different ways to transmit each packet. Thus, the total number of different



(a) model-1 (for outband relaying)
different subbands are used

(b) model-2 (for inband relaying)
only one subband is used everywhere

Fig. 3. The frequency reuse models considered in this paper

schedules are $(M^2 + M + 1)^3$. The "+1" covers the case where the packet is not transmitted during this schedule. Obviously, the number of possible schedules grows exponentially with the number of packets.

We start by showing hardness results for Problem 1 using a reduction from the NP-hard two-dimensional Knapsack Problem (2-KP) [18]. An instance for 2-KP is a set of n items and a 2-dimensional knapsack. Each item i has a profit $p_i \geq 0$ and a 2-dimensional size: $s_i[1]$ and $s_i[2]$. The knapsack's 2-dimensional size is $[K_1, K_2]$ where K_1 and K_2 are integers > 0. The objective is to find a feasible set of items with a maximum profit. A feasible set of items is a set for which the total size of the selected items in each dimension d is at most K_d.

When we compare a transmission instance in Problem 1 to an item of 2-KP, we see clear similarity: both are associated with a scalar profit and with a 2-dimensional cost. However, there is also one difference: in Problem 1 the 2nd element of the 2-dimensional cost refers to one of R different scheduling zones (i.e., R "knapsacks"), while in 2-KP it refers to the same knapsack. For example, the scheduler may have the following two transmission instances for two different packets:

1) A transmission instance [packet$_1$, BS$\rightarrow$RN$_1\rightarrow$UE$_1$, MCS-1]. This instance has a cost >0 in the scheduling zones of the BS and RN$_1$.

2) A transmission instance [packet$_2$, BS$\rightarrow$RN$_2\rightarrow$UE$_2$, MCS-2]. This instance has a cost >0 in the scheduling zones of the BS and RN$_2$.

This implies that while we can use 2-KP to show that Problem 1 is NP-hard, we cannot simply run an algorithm for 2-KP to solve Problem 1. Note also that 2-KP is not a special case of Problem 1 with $R = 1$. This is because each packet in Problem 1 can have more than one transmission instance from which at most one is selected, while in 2-KP each item has only one configuration.

Lemma 2: Problem 1 is NP-hard. Moreover, it is unlikely that it has an EPTAS[2].

Proof: The proof appears in the Appendix. ∎

B. d-MCKP vs. Sparse d-MCKP

Our algorithms for Problem 1 are presented in Section V. They first transform an instance of Problem 1 into an instance of another well-known theoretical problem, called d-dimensional Multiple-Choice Knapsack (d-MCKP [28]). This problem differs from 2-KP in two important ways: (a) each item has several configurations, from which at most one can be chosen for the solution; (b) the size of each item is a D dimensional vector, where D is an integer > 0 (i.e., $D = 2$ does not necessarily hold). These differences make d-MCKP more similar to our problem, and allow a solution for d-MCKP to be transformed into a solution to Problem 1.

An instance of d-MCKP consists of a D-dimensional knapsack and a set of n items, each with m or fewer D-dimensional configurations. Each configuration j of item i has a D-dimensional vector size $s_i^j \in (\mathbb{N}^+)^D$, in which the dth dimension $s_i^j[d]$ is an integer ≥ 0. Each configuration j of item i has profit $p_i^j \geq 0$. The size of the D-dimensional knapsack is also a vector, $[K[1], \ldots, K[D]]$, where $K[d]$ is an integer > 0. The objective is to find a feasible set of configurations such that the profit is maximized. A feasible set of configurations is a set

[2]An EPTAS (Efficient Polynomial-Time Approximation Scheme) is an algorithm which takes an instance of an optimization problem and a parameter $\epsilon>0$ and, in time $O(f(1/\epsilon)\cdot n^c)$, where n is the problem size and $c>0$ is a constant, produces a solution that is within a factor $1+\epsilon$ of being optimal.

for which the total size of the selected configurations in each dimension d is at most $K[d]$ and at most one configuration of each item is selected. It is important to note that despite their similarity, d-MCKP and Problem 1 are different because a configuration of d-MCKP may have a size > 0 in *each of the D-dimensions*, whereas a configuration (transmission instance) in Problem 1 may have a size > 0 in *at most two* dimensions: that of the BS and that of one RN. We take advantage of this difference in order to develop efficient algorithms for Problem 1.

Lemma 3: Any algorithm for d-MCKP can be transformed into an algorithm for Problem 1 with the same running time and performance guarantees.

Proof: A transformation similar to that presented in the proof of Lemma 2 can be used to transform an instance of Problem 1 into an instance of d-MCKP in linear time. ∎

Let p_{opt} be the total profit of the optimal solution and $\alpha \geq 1$. An **α-approximation** returns a solution whose profit is at least $\frac{p_{\text{opt}}}{\alpha}$.

Many heuristics exist for d-MCKP [3], [8], but they do not provide a known performance guarantee. In [28], a $(1+\epsilon)$-approximation[3] for d-MCKP is given for $\epsilon \geq 0$. However, this algorithm is impractical for Problem 1 for two reasons: (a) it requires solving a linear program, which is impractical for a BS that needs to solve Problem 1 once every 1ms; (b) its running time becomes impractical for large values of D. In [34], a dynamic programming algorithm for solving d-KP (the d-dimensional Knapsack Problem) is presented. This problem is similar to d-MCKP except that each item has only one configuration. Using similar ideas to those in [34], a dynamic programming for d-MCKP can be devised. It returns an optimal solution, but its running time renders it impractical when the number of RNs grows. However, we later show that it can be invoked as a procedure on small d-MCKP instances ($D = 2$ and $D = 3$) to solve Problem 1.

A closer look at Problem 1 reveals an important difference between it and d-MCKP: in Problem 1 each item has at most two size dimensions while in d-MCKP there are D. This difference allows us to define a new theoretical problem called "sparse d-MCKP," which will be shown to be more equivalent to Problem 1 than d-MCKP.

Definition 3: An instance of **sparse d-MCKP** consists of a D-dimensional knapsack and a set of n items, each with at most m configurations. Each configuration j of item i has a profit $p_i^j \geq 0$ and a 2-dimensional size $s_i^j[1]$ and $s_i^j[2]$, where $s_i^j[1]$ is the size of this configuration in the 1st dimension and $s_i^j[2]$ is the size of this configuration in some other dimension d_i, where $d_i \in \{2, \ldots, D\}$.

[3]Let p_{opt} be the total profit of the optimal solution and $\alpha \geq 1$. An α-approximation returns a solution whose profit is at least $\frac{p_{\text{opt}}}{\alpha}$.

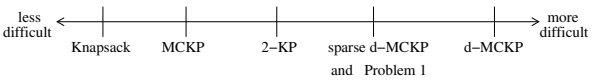


Fig. 4. Comparative difficulty of the various problems related to this paper

In addition, $s_i^j[2] > 0$ implies that $s_i^j[1] > 0$ must hold. The size of the D-dimensional knapsack is a vector, $[K[1], \ldots, K[D]]$, where each component is an integer > 0. The objective is to find a feasible set of configurations, with at most one configuration for each item, such that the profit is maximized. A feasible set of configurations is a set for which the total size of the selected configurations in each dimension does not exceed the knapsack size.

Since an instance to the sparse d-MCKP is also an instance to d-MCKP, d-MCKP is computationally more difficult than sparse d-MCKP. However, from the proof of Lemma 2 it follows that solving sparse d-MCKP is at least as computationally difficult as solving 2-KP. Figure 4 illustrates the comparative difficulty of sparse d-MCKP and related problems.

Lemma 4: Problem 1 is equivalent to sparse d-MCKP.

Proof: We prove this by showing how to transform an instance of Problem 1 into an instance of sparse d-MCKP in polynomial time such that a solution for the transformed instance will also solve the instance of Problem 1. The other direction, namely, transforming an instance of sparse d-MCKP into an instance of Problem 1, can be proven in a similar way.

By Lemma 1, each transmission instance of a packet is associated with the path BS→UE or BS→RN$_i$→UE, where UE is the destination of the packet and RN$_i$ is the default RN for this UE. Using the same transformation as in the proof of Lemma 3, we create an instance of d-MCKP for which all configurations of the same item have size > 0 in the first dimension, size ≥ 0 in the default RN dimension, and size 0 in every other dimension. Note that if the size in the default RN dimension is > 0, so is the size in the first (BS) dimension. The resulting d-MCKP instance is now transformed into a sparse d-MCKP instance in the following way. Each D-dimensional d-MCKP configuration j of item i with a vector size $\hat{s}_i^j[1], \ldots, \hat{s}_i^j[D]$ is transformed into a sparse d-MCKP configuration in which d_i is set to the dimension $d > 1$ whose size > 0. We set $s_i^j[1] = \hat{s}_i^j[1]$ and $s_i^j[2] = \hat{s}_i^j[d_i]$. ∎

V. Scheduling Algorithms

This section is divided into two subsections. In the first subsection we present a pseudo-polynomial time algorithm, which uses algorithms for MCKP and 2-MCKP as procedures, and prove that this algorithm returns an approximation for Problem 1. In the second subsection we present a water-filling algorithm for Problem 1. This

algorithm does not have a performance guarantee, but has a better running time and is simpler to implement. Both algorithms are first developed in the context of model-1 described in Section III-C. However, as we show in Section VI, they can be easily adapted for model-2 as well.

A. A Pseudo-Polynomial Time Algorithm

We now propose a pseudo-polynomial time algorithm, which transforms any α-approximation algorithm for 2-MCKP ($A_{\text{2-MCKP}}$) and any β-approximation algorithm for MCKP (A_{MCKP}) into an $(\alpha \cdot \beta)$-approximation algorithm for sparse d-MCKP. The algorithm divides the items into $D-1$ disjoint sets and solves an instance of 2-MCKP for each set separately. Then, an MCKP (which is equivalent to 1-MCKP) instance is generated, in which an item configuration corresponds to a solution for a 2-MCKP instance. The MCKP instance is solved and all corresponding item configurations are returned as a solution.

Algorithm 1: (An $(\alpha \cdot \beta)$-approximation algorithm for sparse d-MCKP)

1) Divide the items into $D-1$ disjoint sets according to their d_i dimension ($d_i \in \{2, \ldots, D\}$). The set corresponding to d_i is denoted $M[d_i]$.
2) For $d = 2 \ldots D$:

   For $k = 0 \ldots K[1]$ run $A_{\text{2-MCKP}}$ on $M[d]$ with knapsack size $[k, K[d]]$. Let SOL_k^d be the returned solution.
3) Create a new MCKP instance as follows:
   - The knapsack size is $K[1]$.
   - Each $M[d]$ is transformed into an MCKP item with $K[1]+1$ configurations. The size of configuration j ($j \in \{0, \ldots, K[1]\}$) is the total size in the 1st dimension of SOL_j^d and its profit is the total profit of this solution. Thus, in the resulting MCKP instance, the total number of items is $(D-1)$ and each item has $(K[1]+1)$ configurations.
4) Run A_{MCKP} to solve the MCKP instance. Each configuration in the solution corresponds to a subset of the configurations given in the original sparse d-MCKP instance. Return the union of all those subsets. ∎

Lemma 5: If $A_{\text{2-MCKP}}$ is an α-approximation for 2-MCKP and A_{MCKP} is a β-approximation for MCKP, Algorithm 1 is an $(\alpha \cdot \beta)$-approximation for sparse d-MCKP.

Proof: Let OPT be an optimal solution for the sparse d-MCKP instance. The chosen configurations in OPT can be partitioned into $D-1$ sets according to their d_i dimension. For each set i, $i = 2 \ldots D$, let s_i be the total size in the first dimension of all configurations in this set, and let p_i be their total profit. In Step 2 of Algorithm 1, when the inner loop reaches $k = s_i$ (because $s_i \leq K[1]$) and the outer loop reaches $d = i$ (i.e., the items in $M[i]$ are considered), $A_{\text{2-MCKP}}$ is invoked on a 2-MCKP instance with item set $M[i]$ (all configurations of these items from the sparse d-MCKP instance are considered) and a knapsack size $[s_i, K[i]]$. Therefore, in the MCKP instance created in Step 3 of the algorithm, the item transformed from $M[i]$ has a configuration whose size in the first dimension is at most s_i and its profit is $\geq \frac{p_i}{\alpha}$, where α is the approximation ratio of $A_{\text{2-MCKP}}$.

We now show that the solution returned in Step 4 of Algorithm 1 is an $(\alpha \cdot \beta)$-approximation. First, note that the solution returned by Algorithm 1 is feasible, because (a) the knapsack size of the MCKP instance created in Step 3 is $K[1]$ and thus the total size in the first dimension does not exceed the capacity; (b) for any other dimension ($d > 1$), the MCKP instance created in Step 3 contains at most one item whose size in this dimension in one or more configurations is > 0 but does not exceed the knapsack size. Let $P(\text{OPT})$ be the total profit of OPT (the optimal solution for the sparse d-MCKP instance). The profit of the optimal solution to the MCKP problem returned in Step 4 is at least $\sum_i \frac{p_i}{\alpha} = \frac{P(\text{OPT})}{\alpha}$. Since in Step 4 A_{MCKP} is invoked, its output returns a β-approximation and thus its total profit is at least $\frac{P(\text{OPT})}{\alpha \cdot \beta}$. Finally, in Step 4, the union of all original configurations has the same total profit. Thus, the approximation ratio of the solution for the sparse d-MCKP instance holds. ∎

We now analyze the running time of Algorithm 1, which depends on the running time of the procedures it uses in Step 2 and Step 4 . Let $T(A_{\text{2-MCKP}}, n, m, k[1], k[2])$ be the running time of $A_{\text{2-MCKP}}$ on a 2-MCKP instance with n items, each with at most m configurations, and a 2-dimensional knapsack size $[k[1], k[2]]$. Algorithm 1 invokes $A_{\text{2-MCKP}}$ $((D-1) \cdot (K[1]+1))$ times. Let $T(A_{\text{MCKP}}, n, m, K[1])$ be the running time of A_{MCKP} on an MCKP instance with n items, each with at most m configurations and a knapsack size $K[1]$. The MCKP in Step 3 has $(D-1)$ items, each with $(K[1]+1)$ configurations and a knapsack size $K[1]$. The total running time of Algorithm 1 is therefore $O(D \cdot K[1] \cdot T(A_{\text{2-MCKP}}, n, m, K[1], \max_{i \geq 2}\{K[i]\}) + T(A_{\text{MCKP}}, D, K[1], K[1]))$, where n is the number of items, each with at most m configurations, in the sparse d-MCKP instance. This running time remains practical even when D (which is no more than $R+2$, where R is the number of RNs).

The dynamic programming algorithm for 2-MCKP can be used by Algorithm 1 in Step 2, and the dynamic programming algorithm presented in [18] can be used

by Algorithm 1 in Step 4. In this case both A_{MCKP} and $A_{\text{2-MCKP}}$ are optimal and thus Algorithm 1 is an optimal algorithm whose running time is $O(D \cdot (K[1])^2 \cdot \max_{i \geq 2}\{K[i]\} \cdot n \cdot m)$, where n is the number of items (i.e., the number of packets waiting for transmission in Problem 1) and m is the maximum number of item configurations (i.e., the maximum number of transmission instances for a packet in Problem 1). By Lemma 1, $m \leq (M^2 + M)$.

An additional improvement can be applied when the dynamic programming algorithm for 2-MCKP is used by Algorithm 1. We can avoid the loop in Step 2 and instead generate the required $K[1]+1$ solutions for each $M[d]$ set by running the dynamic programming algorithm for knapsack size $[K[1], K[d]]$, and then finding the solution using the corresponding entry in the dynamic programming array. This reduces the time complexity of the algorithm to $O(D \cdot K[1] \cdot \max_{i \geq 2}\{K[i]\} \cdot n \cdot m)$.

In practical systems, $\max_i\{K[i]\}$ is very small. For example, in a 20 Mhz LTE system, $\max_i K[i] \leq 100$. Consequently, in such systems the running time of Algorithm 1 is practical for a BS that needs to run it every 1ms.

B. A Water-Filling Algorithm

We now present a new polynomial time algorithm for sparse d-MCKP, which is based on the heuristic for d-KP presented in [18]. Unlike Algorithm 1, the new algorithm does not have a theoretical performance guarantee. However, it is simple to implement and its running time is better than that of Algorithm 1 when the latter uses the dynamic programming algorithms as its sub-procedures. To describe the new algorithm we need the following definition [18]:

Definition 4: The **efficiency** of a sparse d-MCKP configuration j of item i is $\frac{p_i^j}{s_i^j[1]+s_i^j[2]}$, where p_i^j is the profit of the corresponding configuration, and $s_i^j[1]$ and $s_i^j[2]$ are its 2-dimensional size.

The algorithm first sorts the configurations in decreasing order of their efficiency, and then considers them for the solution in this order. Each configuration is added to the final schedule if: (a) no previous configuration for the corresponding item is already in the solution; and (b) the resource pool in each dimension is not exceeded.

Algorithm 2: (A water-filling algorithm for sparse d-MCKP)

1) Compute the efficiency for configuration j of item i for each pair (i, j), $i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, m\}$.
2) Sort all the configurations of all items in decreasing order of efficiency.
3) Go over the configurations list from the most efficient to the least efficient; add each configuration to the solution if (a) its item has not been selected yet (in previous configurations); (b) it does not exceed the resource pool in any dimension.
4) Return the resulting schedule. ∎

Given that $D \leq n$, sorting the configurations dominates the running time of this algorithm, and its time complexity is $O(n \cdot m \cdot \log(n \cdot m))$.

In Section VII we show that in practical cases the performance of Algorithm 2 is very good. However, this algorithm provides no theoretical performance guarantee. As an example, consider the following sparse d-MCKP instance. The number of dimensions is $D = 2$, the number of items is $n = 2 \cdot M$, each with one configuration ($m = 1$); the weights and profits of items $2 \cdot i$ and $2 \cdot i - 1$ ($i = 1, \ldots, M$) are $s_{2i}^1[1] = s_{2i}^1[2] = s_{2i-1}^1[1] = 1$, $s_{2i-1}^1[2] = M$, $p_{2i}^1 = 1$, $p_{2i-1}^1 = M/2$. The size of the 2-dimensional knapsack is $K[1] = M$, $K[2] = M^2$. The efficiency of all items whose index is an even number is $1/2$; the efficiency of items whose index is an odd number is $M/(2(M+1))$. Algorithm 2 will obtain an aggregated profit of M while the optimal solution is to pack all items whose index is an odd number, thereby obtaining a total profit of $M/2 \cdot M = M^2/2$. Thus, the worst-case performance ratio of Algorithm 2 is unbounded.

VI. Adapting Our Algorithms to Model-2

We have shown that Problem 1 is equivalent to sparse d-MCKP for model-1. But this is not the case for model-2 since here there are two BS scheduling zones (Figure 2(b)). Thus, in each configuration of each item (packet), there are at most 3 dimensions whose size can be larger than 0: two correspond to the scheduling zones of the two BSs (the first and second dimensions) and one to the default RN's scheduling zone. We can solve such instances by applying a small change to Algorithm 1, as follows:

- In Step 1 of Algorithm 1, the items are divided according to the dimension of their default RN's scheduling zone. Thus, $D-2$ item sets are created: $M[d]$, for $d \in \{3, \ldots, D\}$.
- In Step 2 we loop over all pairs (k_1, k_2) for $k_1 \in \{0, \ldots K[1]\}$ and $k_2 \in \{0, \ldots K[2]\}$. In each iteration, a 3-MCKP instance, whose item set is $M[d]$ and knapsack size is $[k_1, k_2]$, is solved using an algorithm for 3-MCKP, $A_{\text{3-MCKP}}$.
- In Step 3 we create an instance of 2-MCKP instead of MCKP. Here, the knapsack size is $[K[1], K[2]]$ and each item has $(K[1]+1) \cdot (K[2]+1)$ configurations. Each configuration in Step 3 is created while taking into account two dimensions (the two that correspond to the BS scheduling zones).

- We run $A_{\text{2-MCKP}}$ to solve the 2-MCKP instance in Step 4.

Lemma 6: If $A_{\text{3-MCKP}}$ is an α-approximation for 3-MCKP and $A_{\text{2-MCKP}}$ is a β-approximation for 2-MCKP, the adapted version of Algorithm 1 is an $(\alpha \cdot \beta)$-approximation for sparse d-MCKP in model-2.

Proof: The proof is similar to that of Lemma 5. ∎

We now analyze the time complexity of the algorithm. Let $T(A_{\text{3-MCKP}}, n, m, k[1], k[2], k[3])$ be the running time of $A_{\text{3-MCKP}}$ on a 3-MCKP instance with n items, each with at most m configurations, and knapsack size $[k[1], k[2], k[3]]$. The time complexity of the algorithm is $O(D \cdot K[1] \cdot K[2] \cdot T(A_{\text{3-MCKP}}, n, m, K[1], K[2], \max_{i\geq 3}\{K[i]\}))$, since 3-MCKP is solved instead of 2-MCKP. When the dynamic programming algorithm is used as a procedure for solving 3-MCKP in Step 2, an additional improvement can be made, similar to the one made for Algorithm 1, which reduces the time complexity by a factor of $K[1] \cdot K[2]$.

Next, we explain how to adapt Algorithm 2 for model-2. For sparse d-MCKP under model-2, an item i has at most 3 non-zero dimensions. However, since in each configuration there are at most 2 dimensions whose size can be larger than 0, we can run Algorithm 2, except that $D+2$ dimensions should be considered. It is easy to see that the running time and correctness analysis remain valid.

VII. Simulation Study

In this section we present Monte-Carlo simulation results for the algorithms proposed in the paper. The purpose of this section is three-fold: (a) to compare the performance of Algorithm 1 and Algorithm 2; (b) to study the impact of various network parameters on the performance of our algorithms; and (c) to study the performance gain from using RNs.

A. Network Model

We consider a hexagonal network cell and its 2-hop neighboring cells (total of 19 cells). Scheduling is performed in this cell, while the surrounding cells are considered for the calculations of the SINR experienced by each receiver. Our interference model and parameters are based on the 3GPP specifications [1] and on the work presented in [33], [38], except that omni-directional antennas are considered instead of directional antennas. These parameters are summarized in Table II.

The system bandwidth is 20Mhz; thus, there are 100 scheduled blocks in every 1ms subframe. The average size of each data packet is 3.5 scheduled blocks if it is transmitted using [QPSK, 1/2], which is the most robust MCS out of the 7 MCSs considered in this study. For each MCS and link (BS→RN, RN→UE and BS→UE), the success probability of a transmitted packet is determined from the corresponding SINR value at the receiver using data taken from [6]. Our utility function in this section aims at maximizing the number of successfully delivered packets. Thus, the profit from transmitting a packet to a user using a particular MCS is taken as the probability that the packet is successfully received over the BS→UE or the BS→RN→UE links. The cost of transmitting a packet is equal to the number of scheduled blocks used in each link, which depends on the length of the packet and the chosen MCS for each link. This cost is rounded up to the nearest integer.

Parameter	Value	Parameter	Value
network layout	19 BSs	UE/RN height	1.5m
system frequency	1,500MHz	TX power	39dBm
BS antenna height	20m	TX ant. gain	18.9dBi
inter-site distance	1,700m	RN power	30dBm
num. of MCSs	7	system bw.	20Mhz

TABLE II
Simulation network parameters

B. Interference Model

We start by describing how the SINR of each user is calculated. Let $p_t(u)$ be the power received by UE u from transmitter t, where t is either a BS or an RN. In addition, let $\mathcal{T}(t)$ be the set of transmitters, other than t, that transmit over the same subband used by t. The SINR experienced by u is defined by:

$$\gamma_t(u) = \frac{p_t(u)}{\sum_{t' \in \mathcal{T}(t)} p_{t'}(u) + n_0 w},$$

where w is the system bandwidth (20 MHz), n_0 is the thermal noise over the bandwidth w, and $p_t(u)$ is the end power given by the following equation [33]:

$$p_t(u) = p_t - \text{PL}_t(u) + g_t(\text{dBm}).$$

In this equation, p_t is the dBm power of antenna t, and g_t is the gain of this antenna. $\text{PL}_t(u)$ is the path loss, estimated using the Hata propagation model. It is calculated using the following equation [15]:

$$\text{PL}_t(u) = 69.55 + 26.16\log_{10}(f_0) - 13.82\log_{10}(z_t) - a(z_u) + (44.9 - 6.55\log_{10}(z_u))\log_{10}(d_t(u)),$$

where $f_0 = 1{,}500$MHz is the transmission frequency, z_t is the height (meters) of t's antenna, z_u is the height (meters) of user u, $d_t(u)$ is the distance (kilometers) between u and the antenna of t, and $a(z_u) = 0.8 + (1.1 \cdot \log_{10}(f_0) - 0.7) \cdot z_u - 1.56\log_{10}(f_0)$ is a function that fits a small or medium sized city.

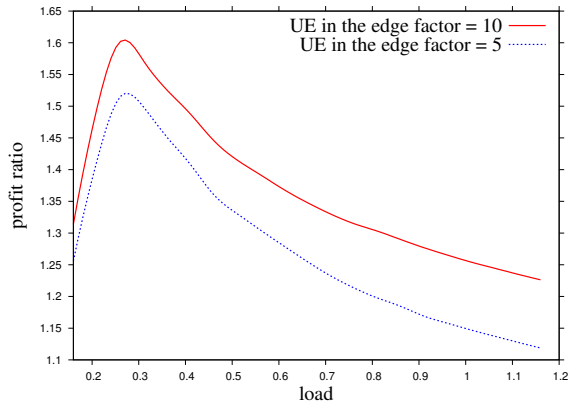


Fig. 5. The increase in performance from deploying 3 RNs in model-1 for two UE distributions

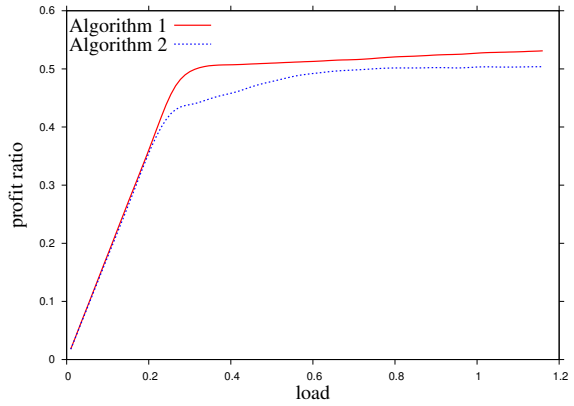


Fig. 6. The performance of Algorithm 1 and Algorithm 2 in model-1 for 3 RNs

C. Simulation Results

To draw one point on each of the graphs presented in this section, we generate 100 random instances with different seeds and average their results.

We start by evaluating the performance gain from adding RNs. Throughout this section, the **normalized load** is defined as the number of scheduled blocks required to transmit all pending packets, if they are all transmitted directly by the BS using the most efficient MCS, divided by the total number of scheduled blocks in a subframe over all subbands. Since the number of pending packets is unlimited, the load can be greater than 1. Throughout the simulations, we use the optimal dynamic programming algorithms for A_{MCKP}, $A_{\text{2-MCKP}}$, and $A_{\text{3-MCKP}}$ in Algorithm 1, therefore Algorithm 1 is optimal.

We first consider model-1. The number of scheduled blocks in the reuse-1 subband at the BS is set to 55 and the number of scheduled blocks in the reuse-1/3 subband available to each RN is set to 15. Since $55+3\cdot 15 = 100$, this reuse scheme utilizes all available subbands. Also, since $55 \geq 3 \cdot 15$, the BS has enough pending packets in every subframe to fill up the next 1ms subframe of 3 RNs. We found that for the considered network

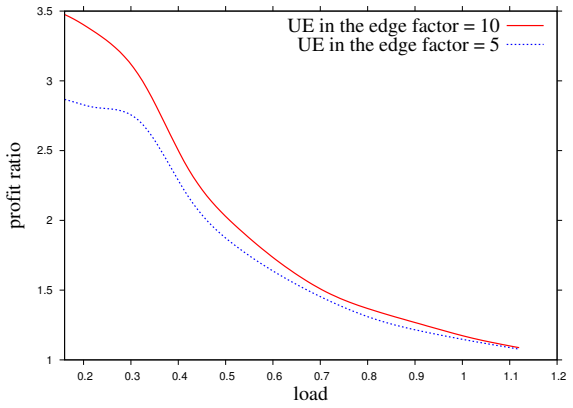


Fig. 7. The increase in performance from deploying 3 RNs in model-2 for two UE distributions

parameters (Table II), placing the RNs at a distance of 500 meters from the BS results in a reasonable SINR for a BS→RN transmission and a reasonable SINR for RN to cell-edge UE transmissions.

To see the benefit from adding RNs to a network, we compare the performance to that of a network that employs the same FFR scheme but does not employ RNs. For a fair comparison, similar parameters are used with and without RNs. Specifically, the same number of scheduled blocks for reuse-1/3 and reuse-1 subbands is considered. Under these parameters, the maximum number of packets that can be scheduled in a subframe with no RNs is 70 (15 in the reuse-1/3 subband and 55 in the reuse-1 subband), and the normalized load is calculated according to this number. A UE is viewed as a cell-edge UE if its distance from the BS is more than 700 meters, and its distance from some RN is shorter than the distance of this RN from the cell edge. In this case, the SINR for direct BS→UE transmission is very low. This allows us to simulate practical scenarios where RNs are placed in areas where many UEs have a poor SINR for direct transmission by the BS.

Figure 5 shows the performance gain when 3 RNs are placed in every cell. The y-axis shows the ratio between the total profit obtained by Algorithm 1 for a network with 3 RNs and the total profit obtained without RNs. The latter is determined by a dynamic programming algorithm that obtains an optimal solution. The x-axis in this figure is the normalized load as defined earlier. The figure shows 2 curves: in the lower curve a UE is 5 times more likely to be a cell-edge UE than to be uniformly located in the cell; in the upper curve this ratio increases to 10. As expected, the increase in performance is greater when there are more cell-edge UEs. In addition, we can see that with RNs the performance of the network increases by up to 60%. For small loads, the increase is small since there are not many pending packets and they can be scheduled in the reuse-1/3 subband of the BS when no RNs are used. But, as the load increases, there

is not enough reuse-1/3 bandwidth to accommodate all these packets. When these packets are transmitted using the BS reuse-1 bandwidth, they acquire a small profit due to a poor SINR. With RNs, however, these packets can be transmitted with good SINR through the RNs. As the load increases further, there are more UEs closer to the BS; these UEs do not require the assistance of the RNs and thus the performance gain decreases.

For the parameters used for Figure 5, Algorithm 2 performs very close to Algorithm 1. Thus, the same curves shown in Figure 5 for Algorithm 1 also represent Algorithm 2. The reason for this is that with this set of parameters, the efficiency of the transmission configurations that use the RNs is very high, which makes them attractive for selection by Algorithm 2.

In Figure 6, we reduce the distance for which a UE is considered as a cell-edge UE from 700 to 500 meters. A UE is now 5 times more likely to be a cell-edge UE than to be uniformly located within the cell. Other than that, we use the same parameters as for Figure 5. The y-axis shows the ratio between the total profit obtained by the algorithm (Algorithm 1 or Algorithm 2) and the maximum profit obtained if all packets are transmitted directly by the BS using the most efficient MCS. The x-axis in this figure is the normalized load as defined earlier.

This time, Algorithm 1 exhibits better performance than Algorithm 2 for high loads. This is because for such loads more cell-edge UEs have a reasonable SINR for the direct BS transmissions. Therefore, such configurations have a higher efficiency. Algorithm 2 is more likely to choose direct BS→UE configurations, and it obtains a smaller profit.

The performance gain due to the addition of RNs in the setting of Figure 6 is smaller compared to the gain in the settings of Figure 5. This is expected, since more UEs can be reached directly by the BS.

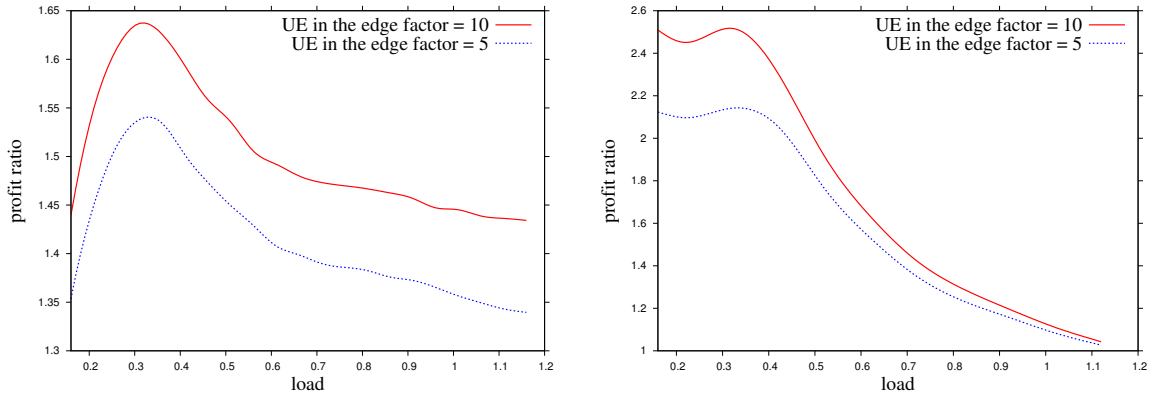
We now consider model-2. Recall that the decision about whether model-1 or model-2 should be used depends on many factors and regulations that are beyond the scope of this paper. However, we compare the performance of our algorithms in two different models to show that they are generic and they work well in different models. Since the interferences are different for these two models, some network parameters, such as the location of the RNs, are determined separately for each model.

As we did for model-1, we start with 3 RNs. All the parameters remain the same as in model-1, except that now, because reuse-1 is employed by the BS and RNs, each BS and each RN has 100 scheduled blocks. In addition, the distance of the RNs from the BS increases to 700 meters. The RNs are required to be closer to the cell edge in order to have a reasonable SINR for the transmissions from the RNs to cell-edge UEs. Figure 7 shows the performance gain when 3 RNs are placed in every cell. The y-axis shows the ratio between the total profit obtained by Algorithm 1 for a network with 3 RNs and the total profit obtained when RNs are not used. The latter is determined by a dynamic programming algorithm that obtains an optimal solution. The x-axis in this figure is the normalized load as defined earlier. Since model-2 schedules 2 consecutive subframes together (see Figure 2(b)), in each subframe we set the number of packets for the cell with no RNs to be half of the number of packets when the cell has RNs. This is necessary in order to fairly evaluate how the addition of RNs affects performance.

We can see in Figure 7 that for low loads, the addition of RNs results in much higher profit gain compared to Figure 5. This is because when reuse-1 is employed, cell-edge UEs experience significant interference from neighboring cells. Their SINR for direct BS transmission is so low that they cannot be reached without the assistance of an RN. When the load increases, some UEs are closer to the BS and thus can receive their packets directly. Therefore, the profit ratio decreases.

The next scenario for model-2 is obtained by reducing the distance for which a UE is considered as a cell-edge from 700 to 500 meters. In contrast to model-1, it turns out that even in this case the performance of Algorithm 2 is very close to the optimal solution. This is because the RNs and BS use the same subband, and they interfere with each other. Consequently, a packet can have a reasonable SINR for direct BS→UE transmission or for BS→RN→UE transmission, but not for both. This leads to a good transmission success probability and to good efficiency only for one configuration, which is likely to be selected by Algorithm 2.

Finally, we increase the number of RNs to 6. For model-1 we set the number of scheduled blocks in the reuse-1 subband to 70 and the number of scheduled blocks in the reuse-1/3 subband of each RN to 10. These parameters are chosen such that there are sufficient scheduled blocks for the BS to transmit enough packets to each RN simultaneously in the same subframe. In model-1 the RN distance is 500 and in model-2 it is 700. In both models a UE is considered as a cell-edge if its distance is $\geq$ 700 meters from the BS, and its distance from the RN is not more than the distance of the RN from the cell edge. The results are shown in Figure 8(a) for model-1 and Figure 8(b) for model-2. While increasing the number of RNs allows better spatial reuse, it also increases the interference experienced by each RN. Therefore, with 6 RNs each RN uses a more robust and less efficient MCS for its transmissions, and the performance gain is similar to that experienced for 3 RNs.



(a) model-1 (b) model-2

Fig. 8. The increase in performance from deploying 6 RNs for two UE distributions

VIII. Conclusions

We defined the scheduling problem for an OFDMA cell with relay nodes (RNs) as a new optimization problem called sparse d-MCKP and proved it is NP-hard. We developed an algorithm with a performance guarantee for solving this problem and a water-filling algorithm with a lower time complexity. We focused on a specific frequency reuse model and evaluated the performance of the two algorithms for this model. While the algorithms were presented in this specific context, they can be easily adapted to other reuse models of OFDMA networks with RNs. We used an extensive simulation study to evaluate the two algorithms. Our main conclusion is that our water-filling heuristic is usually as efficient as our approximation, even if the latter is implemented such that its results are optimal. We also showed that increasing the network throughput with RNs is not a trivial task, and it depends on the location of the RNs and the UEs, and on the number of scheduled blocks available to each RN.

References

[1] 3GPP. Evolved Universal Terrestrial Radio Access E-UTRA; Further Advancements for E-UTRA Physical Layer Aspects, TR 36.814.

[2] 3GPP. Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 10), TS 36.211.

[3] M. M. Akbar, M. S. Rahman, M. Kaykobad, E. Manning, and G. Shoja. Solving the multidimensional multiple-choice knapsack problem by constructing convex hulls. *Computers & Operations Research*, 33(5):1259 – 1273, 2006.

[4] I. Akyildiz, D. Gutierrez-Estevez, and E. Reyes. The evolution to 4G cellular systems: LTE-Advanced. *Phy. Comm.*, 3(4), 2010.

[5] D. Amzallag, R. Bar-Yehuda, D. Raz, and G. Scalosub. Cell selection in 4G cellular networks. *IEEE INFOCOM*, 2008.

[6] K. Balachandran et al. Design and analysis of an IEEE 802.16e-based OFDMA communication system. *BLTJ*, 11(4), 2007.

[7] Ö. Bulakci, S. Redana, B. Raaf, and J. Hämäläinen. Impact of power control optimization on the system performance of relay based LTE-advanced heterogeneous networks. *Journal of Communications and Networks*, 13(4):345–359, 2011.

[8] N. Cherfi and M. Hifi. A column generation method for the multiple-choice multi-dimensional knapsack problem. *Computational Optimization and Applications*, 46:51–73, 2010.

[9] C.-S. Chiu and C.-C. Huang. Improving inter-sector handover user throughput by using partial reuse and softer handover in 3GPP LTE downlink. *ICACT*, 1:463 –467, feb. 2008.

[10] H.-H. Choi, J. B. Lim, H. Hwang, and K. Jang. Optimal handover decision algorithm for throughput enhancement in cooperative cellular networks. *IEEE VTC Fall*, pages 1–5, 2010.

[11] R. Cohen and G. Grebla. Joint scheduling and fast cell selection in OFDMA wireless networks. Accepted to IEEE/ACM Trans. Netw.

[12] R. Cohen and G. Grebla. Efficient allocation of CQI channels in broadband wireless networks. *IEEE INFOCOM*, pages 96 –100, April 2011.

[13] R. Cohen and L. Katzir. A generic quantitative approach to the scheduling of synchronous packets in a shared uplink wireless channel. *IEEE/ACM Trans. Netw.*, 15(4):932–943, Aug. 2007.

[14] H. Hamdoun, P. Loskot, T. O'Farrell, and J. He. Practical network coding for two way relay channels in LTE networks. In *VTC Spring*, pages 1–5. IEEE, 2011.

[15] M. Hata. Empirical formula for propagation loss in land mobile radio services. *IEEE Transactions on Vehicular Technology*, 29(3):317–325, Aug. 1980.

[16] F. Huang, J. Geng, G. Wei, Y. Wang, and D. Yang. Performance analysis of distributed and centralized scheduling in two-hop relaying cellular system. *IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1337 –1341, Sept. 2009.

[17] H. Katiyar and R. Bhattacharjee. Outage performance of multi-antenna relay cooperation in the absence of direct link. *IEEE Communications Letters*, 15(4):398–400, 2011.

[18] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer, 2004.

[19] F. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8(1):33–37, 1997.

[20] A. Kulik and H. Shachnai. There is no EPTAS for two-dimensional knapsack. *Inf. Process. Lett*, 110(16):707–710, 2010.

[21] Y. M. Kwon, O. K. Lee, J. Y. Lee, and M. Y. Chung. Power control for soft fractional frequency reuse in OFDMA system. *ICCSA*, 6018:63–71, 2010.

[22] D. Lee, H. Seo, B. Clerckx, E. Hardouin, D. Mazzarese, S. Nagata, and K. Sayana. Coordinated multipoint transmission and reception in LTE-advanced: deployment scenarios and operational challenges. *IEEE Communications Magazine*, 50(2):148–155, 2012.

[23] L. Liu, R. Chen, S. Geirhofer, K. Sayana, Z. Shi, and Y. Zhou. Downlink MIMO in LTE-advanced: SU-MIMO vs. MU-MIMO. *IEEE Communications Magazine*, 50(2):140–147, 2012.

[24] V. Mhatre and C. Rosenberg. The impact of link layer model on

the capacity of a random ad hoc network. In *IEEE International Symposium on Information Theory*, pages 1688 –1692, July 2006.
[25] S. Nagata, Y. Yan, X. Gao, A. Li, H. Kayama, T. Abe, and T. Nakamura. Investigation on system performance of L1/L3 relays in LTE-advanced downlink. *IEEE VTC*, 2011.
[26] D. W. K. Ng, E. S. Lo, and R. Schober. Dynamic resource allocation in MIMO-OFDMA systems with full-duplex and hybrid relaying. *IEEE Trans. on Communic.*, 60(5):1291–1304, 2012.
[27] T. D. Novlan, J. G. Andrews, I. Sohn, R. K. Ganti, and A. Ghosh. Comparison of fractional frequency reuse approaches in the OFDMA cellular downlink. *IEEE GLOBECOM*, 2010.
[28] B. Patt-Shamir and D. Rawitz. Vector bin packing with multiple-choice. *Discrete Applied Mathematics*, 160(10-11), 2012.
[29] S. W. Peters, A. Y. Panah, K. T. Truong, and R. W. H. Jr. Relay architectures for 3GPP LTE-advanced. *EURASIP J. Wireless Comm. and Networking*, 2009.
[30] T. Qu, D. Xiao, and D. Yang. A novel cell selection method in heterogeneous LTE-advanced systems. *IC-BNMT*, Oct. 2010.
[31] M. Salem, A. Adinoyi, M. Rahman, H. Yanikomeroglu, D. Falconer, and Y.-D. Kim. Fairness-aware radio resource management in downlink ofdma cellular relay networks. *IEEE Transactions on Wireless Communications*, 9(5):1628–1639, 2010.
[32] K. Sundaresan and S. Rangarajan. Adaptive resource scheduling in wireless OFDMA relay networks. In A. G. Greenberg and K. Sohraby, editors, *INFOCOM*, pages 1080–1088. IEEE, 2012.
[33] N. Tabia, A. Gondran, O. Baala, and A. Caminada. Interference model and evaluation in LTE networks. *(WMNC)*, Oct. 2011.
[34] H. Weingartner and D. Ness. Methods for the solution of the multidimensional 0/1 knapsack problem. *Operations Research*, 15:83–103, 1967.
[35] W. Wu and T. Sakurai. Capacity of reuse partitioning schemes for OFDMA wireless data networks. *IEEE International Symposium on Indoor and Mobile Radio Communications*, pages 2240 – 2244, Sept. 2009.
[36] X. Xue, J. hong Zhao, and H. Qu. Inter-cell interference coordination scheme based on CoMP. pages 33 –36, Feb. 2012.
[37] Z. Yang, Q. Zhang, and Z. Niu. Throughput improvement by joint relay selection and link scheduling in relay-assisted cellular networks. *IEEE Transactions on Vehicular Technology*, 61(6):2824 –2835, July 2012.
[38] O. Yilmaz, S. Hamalainen, and J. Hamalainen. System level analysis of vertical sectorization for 3GPP LTE. *ISWCS*, pages 453 –457, Sept. 2009.

APPENDIX

Proof of Lemma 2

In [18] it is shown that 2-KP is NP-hard. In [20] it is shown that it is also unlikely to have an EPTAS. We first show how to transform an instance of 2-KP into an instance of Problem 1 in polynomial time such that a solution for the transformed instance will also solve the 2-KP instance. We use a network with a single RN (i.e., $R = 1$). Without loss of generality, let the knapsack size be K for each dimension. The first dimension is transformed into a BS scheduling zone with $K \cdot (n+1) + n$ scheduled blocks, where n is the number of items in the 2-KP instance. The 2nd dimension is transformed into an RN scheduling zone with K scheduled blocks. Every 2-KP item i is transformed into a transmission instance of a data packet whose profit is set to p_i. The path and MCSs for this transmission instance are determined as follows:

- If $s_i[1] > 0$ and $s_i[2] > 0$, the path for the transmission instance is BS→RN→UE. The data packet size and MCSs are chosen such that $(n+1) \cdot s_i[1]$ scheduled blocks are required in the BS scheduling zone and $s_i[2]$ scheduled blocks are required in the RN scheduling zone.
- If $s_i[1] > 0$ and $s_i[2] = 0$, the path is BS→UE. The data packet size and MCS are chosen such that $(n+1) \cdot s_i[1]$ scheduled blocks are required in the BS scheduling zone.
- If $s_i[1] = 0$ and $s_i[2] > 0$, the path for the transmission instance is BS→RN→UE. The data packet size and MCSs are chosen such that one scheduled block is required in the BS scheduling zone and $s_i[2]$ scheduled blocks are required in the RN scheduling zone.

Note that this is a valid input to Problem 1, since each packet is either transmitted only in the BS scheduling zone or in the scheduling zones of the BS and the RN. The above transformation can be performed in polynomial time.

Given an instance of 2-KP and a solution to the corresponding transformed instance of Problem 1, we can easily convert this solution to a solution for the original 2-KP problem. In fact, the selected set of items is also an optimal solution for 2-KP. This is because the only change in the item sizes is made in the first dimension, and at most n items whose size in this dimension is 1 are selected. Each other item i has a size of at least $z_i \cdot (n+1)$, for some integer $z_i > 0$.

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.

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.