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 for the first time the *packet-level* scheduling problem for a network with RNs. This problem is shown to be not only NP-hard, but also very hard to approximate. To solve it, we propose an approximation with a performance guarantee, and a simple water-filling heuristic. 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) [3], [14], [15] 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.

978–1–4799–3360–0/14/$31.00 ©2014 IEEE

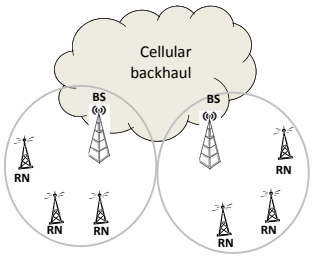

Fig. 1. Example of a network with RNs and their donor BSs

Rather, each RN is associated with a donor BS, to which it is connected through the OFDMA wireless link (see Figure 1). 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.

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. This algorithm has two important advantages compared to the user-level scheduling algorithms proposed in [15], [20], [23]. First, it allows different packets of the same user to have different priority. Second, it allows different packets of the same user to be transmitted using different MCSs.

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 certain 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) [11], 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. Section VI presents an extensive simulation study and Section VII concludes 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 [4], [7], [19]. In [4], the authors study the process of determining the cells that provide service to each mobile station. The potential benefit of global cell selection versus local SNR-based decision is studied. It is shown that the general case of the problem is hard to approximate. However, two algorithms are proposed and are shown to have a performance guarantee under a practical assumption. The authors of [19] also study the cell selection problem and focuses on heterogeneous systems. A new cell selection strategy is proposed to improve the efficiency.

In [7], two basic handover schemes for inter-cell interference coordination are considered, and a handover decision algorithm for improving cell edge throughput is proposed. It is shown that the proposed algorithm obtains a higher cell edge throughput compared to that obtained by the legacy SINR-based decision algorithm.

Papers in the second group include [15], [20], and [23]. In these papers, user-level admission control algorithms are proposed for OFDMA networks with relays. In [15], 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 [20], an algorithm for maximizing the total cell throughput while stabilizing user queues is proposed. In [23], 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 [14] and [18] relay strategies are compared. In [14], 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 [18], 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.

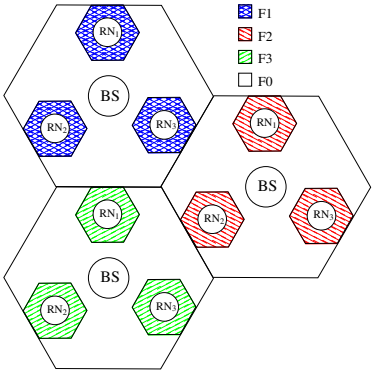

Fig. 2. The frequency reuse model considered in this paper

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 2 for $R = 3$ (frequency reuse aspects of this figure are discussed later on). A 10-ms LTE frame is divided into 10 1-ms subframes[1]. Each subframe contains a pool of scheduled blocks, to be assigned by the BS to waiting packets. The exact number of scheduled blocks in every subframe depends on the system bandwidth; it is 100, for example, in an LTE 20MHz system.

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 3(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 3(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.

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,

[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.

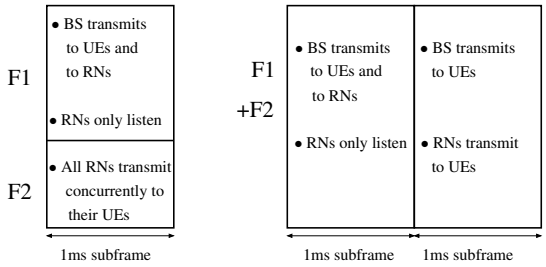

Fig. 3. An abstract structure of the LTE subframe (F1 and F2 are two orthogonal OFDMA subbands)

because it has an important advantage over distributed scheduling [10]: 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) 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 area* 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 area and R RN scheduling areas (Figure 3(a)). Thus, the scheduler has to allocate resources from $R + 1$ scheduling areas (pools) every 1ms. In the inband relaying model, the scheduler needs to make a scheduling decision every 2ms for two consecutive 1ms subframes (Figure 3(b)). In the first 1ms subframe, the scheduler allocates resources only from the BS scheduling area. In the second 1ms subframe, the scheduler allocates resources from the BS scheduling area and R

RN scheduling areas. Thus, in the inband relaying model, the scheduler has to make decisions for $R+2$ scheduling areas 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 area(s).

We adopt the profit-based scheduling model proposed in [9]. 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 [5], [13].

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

$p_{\textbf{packets}}$ - This profit value is defined as the packet transmission success probability. As a result, the sum of all profit values equals the expected number of successfully received packets, i.e., packet-level throughput.

$p_{\textbf{throughput}}$ - 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.

$p_{\textbf{energy}}$ - 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.

$p_{\textbf{delay}}$ - 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, $\sum p_{\text{delay}}$ is the expected number of packets scheduled in a given subframe and are likely to be delivered on time.

$p_{\textbf{pf}}$ - 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 [12] 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 area to another. Thus, the profit of a packet might also dynamically change.

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 area associated with this link. It depends on the length of the packet and the MCS, and it is what makes the scheduling problem for a BS with RNs intractable.

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 a specific model. However, these algorithms are easily adaptable to other frequency reuse models as well[2]. The considered model is shown in Figure 2 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 [16]. 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.

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.

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

Throughout the section, the following lemma will be used in order to reduce the number of transmission instances the scheduler considers for each data packet.

Lemma 1: If the scheduler is configured to use only links with an SINR>0dBm, then: (a) a packet can be transmitted either directly or through the RN with which

[2]In the extended version of this paper [8] we present two models. The one considered here is called there model-1, and the one not considered here is called model-2. The latter is relevant for inband relaying.

the user has the best SINR, and not through any other RN; (b) each packet can be associated with at most $(M^2 + M)$ transmission instances, where M is the number of MCSs. (SINR >0dBm is chosen because transmission success probability for SINR no greater than 0dBm is very low [5].)

Proof: The proof is omitted for lack of space. It can be found in the full version of this paper [8]. ■

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 area. 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[3]. 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 area 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 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.

Lemma 2: Problem 1 is NP-hard. Moreover, it admits no EPTAS[4].

Proof: The proof is omitted for lack of space. It can be found in the full version of this paper [8]. ■

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 [17]).

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 $K[i]$ 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 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: The proof is omitted for lack of space. It can be found in the full version of this paper [8]. ■

Many heuristics exist for d-MCKP [6], but they do not provide a known performance guarantee. In [17], a $(1 + \epsilon)$-approximation[5] 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 [22], a dynamic programming algorithm for solving d-KP (the

[3]The default RN is the RN for which the UE has an SINR>0dBm. By Lemma 1 there is only one such RN for each UE.

[4]An EPTAS (Efficient Polynomial-Time Approximation Scheme) is an algorithm which takes an instance of an optimization problem and a parameter ϵ>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.

[5]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 $\frac{p_{\text{opt}}}{\alpha}$.

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 [22], 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 ($D = 2$) 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. 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 $d_i \in \{2, \ldots, D\}$. 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.

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

Proof: The proof is omitted for lack of space. It can be found in the full version of this paper [8]. ■

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 algorithm are developed in the context of the model proposed in this paper. However, they can be easily adapted for other models 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: The proof is omitted for lack of space. It can be found in the full version of this paper [8]. ■

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 ($D = R + 1$, where R is the number of RNs) grows.

The dynamic programming algorithm for 2-MCKP can be used by Algorithm 1 in Step 2, and the dynamic programming algorithm presented in [11] 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 $O(D \cdot K[1] \cdot \max_{i \geq 2}\{K[i]\} \cdot n \cdot m)$.

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 [11]. 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 [11]:

Definition 3: The *efficiency* of a sparse d-MCKP configuration j of item i is $(p_i^j / (s_i^j[1] + s_i^j[2]))$, 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 $O(n \cdot m \cdot \log(n \cdot m))$.

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 I
SIMULATION NETWORK PARAMETERS

VI. 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 [21], [24], except that omni-directional antennas are considered instead of directional antennas. These parameters are summarized in Table I.

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 [5]. 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.

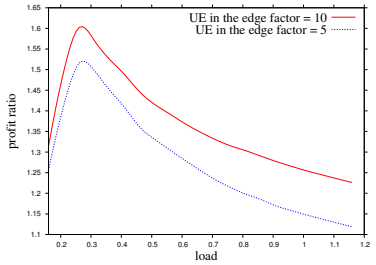

Fig. 4. The increase in performance from deploying 3 RNs for two UE distributions

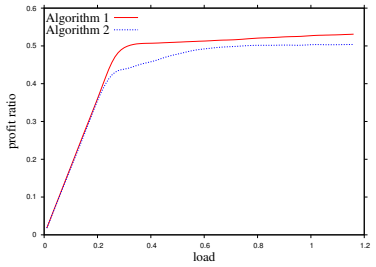

Fig. 5. The performance of Algorithm 1 and Algorithm 2 for 3 RNs

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: $\gamma_t(u) = p_t(u)/(\sum_{t' \in \mathcal{T}(t)} p_{t'}(u) + n_0 w)$, 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 [21]:

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

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:

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

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.

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. Throughout the simulations, we use the optimal dynamic programming algorithms for A_{MCKP} and $A_{\text{2-MCKP}}$, therefore Algorithm 1 is optimal.

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. We found that for the considered network parameters (Table I), 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 4 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 4, Algorithm 2 performs very close to Algorithm 1. Thus, the same curves shown in Figure 4 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 5, we reduce the distance for which a UE is considered as a cell-edge UE from 700 to 500. 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 4. 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 5 is smaller compared to the gain in the settings of Figure 4. This is expected, since more UEs can be reached directly by the BS.

Finally, we increase the number of RNs to 6. The results are omitted from this version due to lack of space and can be found in the full version of this paper [8].

VII. 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 proposed an algorithm with a performance guarantee and also developed a water-filling algorithm with simple implementation and low time complexity. We focused on a specific model and evaluated the performance of our algorithms for this model. Although the algorithms were presented in the context of this model, they can be easily adapted to other FFR models of an OFDMA wireless network with RNs. We used an extensive simulation study to compare the two algorithms. Our main conclusions are 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.

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

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

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

[6] 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.

[7] 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.

[8] R. Cohen and G. Grebla. Multi-dimensional OFDMA scheduling in a wireless network with relay nodes. www.cs.technion.ac.il/~rcohen/PAPERS/mult-OFDMA.pdf.

[9] 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.

[10] 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.

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

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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] B. Patt-Shamir and D. Rawitz. Vector bin packing with multiple-choice. *Discrete Applied Mathematics*, 160(10-11), 2012.

[18] 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.

[19] T. Qu, D. Xiao, and D. Yang. A novel cell selection method in heterogeneous LTE-advanced systems. *IC-BNMT*, Oct. 2010.

[20] 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.

[21] N. Tabia, A. Gondran, O. Baala, and A. Caminada. Interference model and evaluation in LTE networks. *(WMNC)*, Oct. 2011.

[22] H. Weingartner and D. Ness. Methods for the solution of the multidimensional 0/1 knapsack problem. *Operations Research*, 15:83–103, 1967.

[23] 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.

[24] O. Yilmaz, S. Hamalainen, and J. Hamalainen. System level analysis of vertical sectorization for 3GPP LTE. *ISWCS*, pages 453 –457, Sept. 2009.