

# The Slice Is Served: Enforcing Radio Access Network Slicing in Virtualized 5G Systems

Salvatore D'Oro\*, Francesco Restuccia\*, Alessandro Talamonti<sup>†</sup>, and Tommaso Melodia\*

\*Department of Electrical and Computer Engineering, Northeastern University, Boston, USA,

Email: {s.doro, f.restuccia, t.melodia}@northeastern.edu.

<sup>†</sup>Politecnico di Milano, Milan, Italy, Email: alessandro.talamonti@mail.polimi.it.

**Abstract**—The notions of softwarization and virtualization of the radio access network (RAN) of next-generation (5G) wireless systems are ushering in a vision where applications and services are physically decoupled from devices and network infrastructure. This crucial aspect will ultimately enable the dynamic deployment of heterogeneous services by different network operators over the same physical infrastructure. RAN slicing is a form of 5G virtualization that allows network infrastructure owners to dynamically “slice” and “serve” their network resources (*i.e.*, spectrum, power, antennas, among others) to different mobile virtual network operators (MVNOs), according to their current needs. Once the slicing policy (*i.e.*, the percentage of resources assigned to each MVNO) has been computed, a major challenge is how to allocate spectrum resources to MVNOs in such a way that (i) the slicing policy defined by the network owner is enforced; and (ii) the interference among different MVNOs is minimized. In this article, we mathematically formalize the RAN slicing enforcement problem (RSEP) and demonstrate its NP-hardness. For this reason, we design three approximation algorithms that render the solution scalable as the RSEP increases in size. We extensively evaluate their performance through simulations and experiments on a testbed made up of 8 software-defined radio peripherals. Experimental results reveal that not only do our algorithms enforce the slicing policies, but can also double the total network throughput when intra-MVNO power control policies are used in conjunction.

**Index Terms**—Network Slicing, 5G, Radio Access Network (RAN), Interference Management.

## I. INTRODUCTION

Recent studies indicate that the demand for faster, lower-latency wireless cellular connection is growing exponentially each year. By 2023, Ericsson has predicted that around 3.5 billion cellular IoT connections will be active, and more than 1 billion 5G subscriptions will be activated worldwide [1]. It is now clear that existing one-size-fits-all resource allocation policies will not be able to sustain the sheer need for dynamic, effective and efficient radio access strategies, where network operators will need to make the very best use of the extremely limited spectrum bands available for commercial usage [2].

To address the above issues, radio access network (RAN) slicing has been recently welcomed as a promising approach by the academic and industrial communities alike [3–6]. RAN

slicing completely overturns the traditional model of single ownership of all network resources, and realizes a vision where physical network infrastructure is shared among multiple mobile virtual network operators (MVNOs), each one in charge of a separate “slice” of the network, which can be assigned/revoked by the network owner according to the current slicing policy. Although companies such as Amazon or Microsoft apply similar concepts in the context of cloud computing, *RAN slicing is an intrinsically different problem, since spectrum is a scarce resource for which over-provisioning is not possible.*

Once the RAN slices have been defined for each MVNO, a core problem is how to divide and allocate the spectrum resources, also called resource blocks (RBs), according to what prescribed by the slicing policy. In other words, if (for example) an MVNO has been assigned a slice of 15% of the spectrum resources, such MVNO should receive approximately 15% of the available RBs. Thus, the design and evaluation of *RAN slicing enforcement* algorithms is necessary to implement in practice the network owner (NO)’s slicing policy. Moreover, to be effective, RAN slicing enforcement algorithms must facilitate interference-mitigating strategies such as inter-base-station power control (IBSPC) [7, 8], MIMO [9], and coordinated multi-point (CoMP) [9–11] schemes such as Joint Transmission (JT) [12, 13]. However, since such technologies require tight cooperation and coordination among different base stations (BSs), *it becomes imperative to design effective and efficient slicing enforcement algorithms to guarantee that the same (or similar in time/frequency) RBs are assigned to the same MVNOs when BSs are close enough to interfere among themselves.*

To illustrate the above point, we consider the cellular network scenario depicted in Figure 1. Here, the network owner administers two BSs (assumed to be close enough to interfere with each other) and 16 RBs (*i.e.*, 4 frequency units during 4 time units). We consider the case where three MVNOs, namely M1, M2 and M3, have been assigned the following slice:  $M1 = 25\%$ ,  $M2 = 50\%$ ,  $M3 = 25\%$ , on both the BSs. Figure 1(a) shows an optimum slicing enforcement, represented as two *RB allocation matrices* (RBAM), where inter-MVNO interference is absent (*i.e.*, MVNOs control the same RBs at the two BSs). In this case, MVNOs have maximum flexibility and can easily mitigate interference between their mobile users (MUs) residing in the two BSs by using IBSPC. Conversely, the right side of Figure 1 shows sub-optimum RBAMs which cause

This work was supported in part by ONR under Grant 0014-16-1-2213 and Grant N00014-17-1-2046 and in part NSF under Grant CNS-1618727.

This paper has been accepted for publication in IEEE INFOCOM 2019. This is a preprint version of the accepted paper. Copyright (c) 2019 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

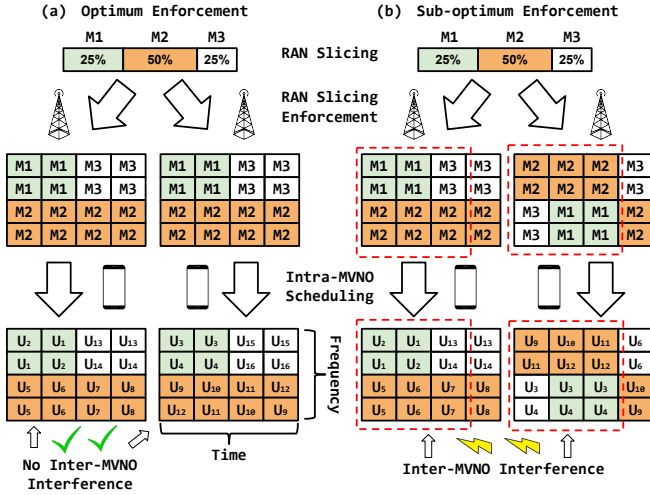


Fig. 1. Optimum and Sub-optimum RAN Slicing Enforcement.

inter-MVNO interference during 12 RBs.

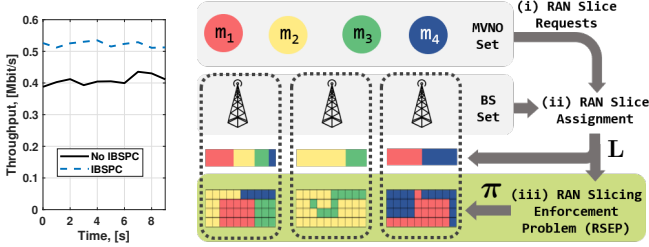


Fig. 2. Experimental Results, IBSPC vs No IBSPC.

Fig. 3. The RAN slicing architecture.

To further investigate the issue of severe inter-MVNO interference, we ran a series of experiments on the testbed to be discussed in detail in Section VII. In such experiments, the RBs are assigned randomly to MVNOs according to the percentage defined in the slicing policy. We performed channel estimation and then applied IBSPC to keep inter-cell interference below a fixed threshold. Figure 2, which depicts the total network throughput as a function of time, indicates that by using IBSPC, the network can increase its throughput by about 25%. *In Section VII, we show that an optimal slicing policy in conjunction with IBSPC can double the network throughput with respect to a random slicing enforcement algorithm.*

The issue of RAN resource allocation has attracted large research interest over the last few years [14–16]. However, the problem of physical-level allocation of spectrum resources to MVNOs still remains underexplored, as discussed in details in Section II (Related Work). This is not without a reason; the design of slicing enforcement algorithms presents the following unique challenges, which are substantially absent in traditional RAN resource allocation scenarios:

- 1) *Feasibility*: RBs need to be assigned to each MVNO according to the slicing policy. In other words, if an MVNO has been assigned 30% of spectrum resources by the network owner, then it should receive 30% of the total RBs available;

- 2) *Orthogonality*: at the same time, orthogonality among each RAN slice must be ensured. That is, each RB should be allocated to only one MVNO to avoid interference and other performance-degrading factors [5, 17, 18];
- 3) *Enabling of advanced 5G technologies*: the RB allocation should facilitate advanced signal processing and RF transmission technologies such as IBSPC, JC, CoMP and MIMO, that considerably improve network performance and have been envisioned to lead the way of 5G networks.

The objective of this work is to design, analyze and experimentally evaluate RAN slicing enforcement algorithms that address the three critical challenges mentioned above. Specifically, this paper makes the following contributions:

- We formulate the *RAN slicing enforcement problem* (RSEP) in Section IV, and show that its optimization version named *RSEP-QP* is NP-hard in Section V-A. To this end, we propose (i) an approximated version of RSEP-QP, named *RSEP-EQ*, in Section V-B; and (ii) a heuristic algorithm named *RSEP-MLF* in Section V-C;
- We prove in Section V-D that the execution time of both *RSEP-QP* and *RSEP-EP* can be drastically reduced (under specific conditions) by solving an equivalent problem, named the *aggregated RSEP*;
- We evaluate the performance of the proposed algorithms through extensive simulations (Section VI) and through experimental evaluation on a testbed composed of 8 USRP software-defined radio devices (Section VII). Results show that our approximation algorithms provide throughput close to the the optimal solution, and that by using IBSPC, our algorithms double the network throughput with respect to a baseline algorithm where slices are allocated in a random fashion.

## II. RELATED WORK

The issue of dividing RAN resources among a set of MVNOs has received significant interest from the research community over the last years; for excellent surveys on the topic, the reader is referred to [14, 15]. This problem (also referred to as *RAN slicing problem* [5, 16, 19]) has been addressed by designing solutions that use a variety of theoretical tools, ranging from optimization [20–23], auctions [24] and game theory [25–27].

The key limitation of the work above is that it does not show how to actually deploy RAN slices on top of the underlying physical network. For this reason, the enforcement of RAN slicing policies has attracted interest from the research community. Prior work [5, 6, 28–30] provides high-level orchestration platforms that virtualize the available resources to create pools of resources that are then shared and allocated among the MVNOs. Although this approach is efficient from a control and orchestration point of view, it might be ineffective in those scenarios where *fine-grained control of physical-layer resources is required*, for example, to enable IBSPC, CoMP and beamforming transmissions.

A few recent papers have focused on addressing the RAN slicing enforcement problem from a resource allocation perspective. Chang *et al.* [17] propose a partitioning algorithm that allocates the available RBs to each requesting MVNO by

simultaneously maximizing the percentage of satisfied MVNOs while allocating the minimum amount of RBs. Similarly, Han *et al.* [31] consider genetic algorithms to assign the available RBs to the MVNOs such that a long-term utility is maximized. However, [17, 31] analyze the problem considering a network with a single BS, and thus cannot be applied in multi-cell networks where MVNOs request different amounts of resources on different BSs. The authors in [18, 32] identify fine-grained RB management as a promising approach to guarantee orthogonality and reduce inter-MVNO interference, thus deploying highly-efficient 5G networks. However, [18] does not provide any algorithm to enforce slicing policies to maximize network efficiency, while [32] does not consider interference among BSs when allocating RBs.

To summarize, our work separates itself from existing literature on RAN resource allocation in the following aspects. We first provide a novel formulation of the RAN slicing enforcement problem (RSEP) that (i) satisfies MVNOs requests, (ii) enforces orthogonality by reducing inter-MVNO interference, and (iii) enables advanced communication techniques such as CoMP and coordinated beamforming by maximizing the number of simultaneous MVNO transmissions on different BSs. Then, we propose three algorithms that solve the RSEP with different optimality performance and computational complexity, and we show how they can be applied to virtualized LTE networks through simulations. Finally, through a real-world experimental testbed, we (i) demonstrate the feasibility and effectiveness of the proposed solutions; (ii) show that our model efficiently reduces inter-MVNO interference by (a) enforcing orthogonality among slices and (b) increasing network throughput by enabling intra-MVNO transmission strategies.

### III. SYSTEM MODEL AND PROBLEM OVERVIEW

We consider the RAN shown in Fig. 3, consisting of a set  $\mathcal{B} = \{1, 2, \dots, B\}$  of  $B$  base stations (BSs), each associated with a coverage area  $\rho_b$ ,  $b \in \mathcal{B}$ . Two BSs  $b$  and  $b'$  are *interfering* (or *adjacent*) with each other if  $\rho_b \cap \rho_{b'} \neq \emptyset$ . We define  $\mathbf{Y} = (y_{b,b'})_{b,b' \in \mathcal{B}}$  as a symmetric adjacency matrix such that  $y_{b,b} = 0$  for all  $b \in \mathcal{B}$ ,  $y_{b,b'} = 1$  if BSs  $b$  and  $b'$  interfere with each other, and  $y_{b,b'} = 0$  otherwise.

The RAN is administered by a NO, who periodically rents to a set  $\mathcal{M} = \{1, 2, \dots, M\}$  of  $M$  MVNOs virtual RAN slices built on top of the underlying physical network  $\mathcal{B}$ . For the sake of generality, we assume RAN slices are valid for  $T$  time slots. This way, slow-changing networks (*e.g.*, cellular networks in rural areas during nighttime) can be modeled with large values of  $T$ , while small values of  $T$  can be used to model fast-changing networks (*e.g.*, urban areas during daytime).

Similar to LTE, spectrum resources are represented as RBs, where each RB represents the minimum spatio-temporal scheduling unit [33]. Also, we consider an OFDMA channel access scheme (as in downlink LTE) where RBs are organized as a *resource grid*, and where  $N_{RB}$  and  $T$  represent the number of available subcarriers and temporal slots, respectively. Thus, the set of available resources at each BS is  $\mathcal{R}$ , with  $|\mathcal{R}| = N_{RB} \cdot T$ .

We explain the interaction between MVNOs and the NO with the help of Fig. 3. First, (i) MVNO's RAN slice requests are collected by the NO. Then, (ii) the NO generates a *slicing*

*profile*  $\mathbf{L} = (L_{m,b})_{m \in \mathcal{M}, b \in \mathcal{B}}$  where  $L_{m,b}$  represents the amount of resources that the NO should allocate to MVNO  $m \in \mathcal{M}$  on BS  $b$  in the time span  $0 \leq t \leq T$  (*i.e.*, *RAN Slice Assignment*); and (iii) computes a slicing enforcement policy  $\pi$  that allocates RBs among the MVNOs such that all MVNO's requests are satisfied (*i.e.*, *RAN Slice Enforcement*).

This paper addresses point (iii), since (ii) has been already extensively investigated [5, 14–16, 19–22, 24–26]. Specifically, we show how the NO can compute an efficient slicing enforcement policy  $\pi$  that satisfies the three requirements described in Section I, *i.e.*, feasibility, orthogonality and enabling of advanced technologies.

### IV. THE RAN SLICING ENFORCEMENT PROBLEM (RSEP)

For any given slicing profile  $\mathbf{L}$  and BS  $b$ , we identify the subset  $\mathcal{M}_b \subseteq \mathcal{M}$  of MVNOs that include BS  $b$  in their RAN slice as  $\mathcal{M}_b = \{m \in \mathcal{M} : L_{m,b} > 0\}$ .

Let  $x_{m,b,n,t} \in \{0, 1\}$  be the *RB allocation indicator* such that  $x_{m,b,n,t} = 1$  if RB  $(n, t) \in \mathcal{R}$  is allocated to MVNO  $m$ ,  $x_{m,b,n,t} = 0$  otherwise. Also, let  $\pi = (\pi_b)_{b \in \mathcal{B}}$  be the *slicing enforcement policy*, where  $\pi_b = (\pi_{m,b})_{m \in \mathcal{M}}$  and  $\pi_{m,b}$  represents the set of RBs on BS  $b$  that are allocated to MVNO  $m$ . In more detail, for any RB  $(n, t) \in \mathcal{R}$ , we have that  $(n, t) \in \pi_{m,b} \iff x_{m,b,n,t} = 1$ . Hence, the set  $\Pi$  of all feasible slicing enforcement policies  $\pi$  can be defined as:

$$\Pi = \{ \pi = (\pi_{m,b})_{m \in \mathcal{M}, b \in \mathcal{B}} : |\pi_{m,b}| = L_{m,b} \wedge \pi_{m,b} \cap \pi_{m',b} = \emptyset \forall m \neq m', m, m' \in \mathcal{M}, b \in \mathcal{B} \} \quad (1)$$

To properly formulate the RSEP, we now introduce the concept of linked RBs.

**Definition 1** (Linked RBs). A given RB  $(n, t)$  on the resource grid is *linked* to MVNO  $m$  on two interfering BS  $b$  and  $b'$  if and only if  $x_{m,b,n,t} = x_{m,b',n,t} = 1$  and  $y_{b,b'} = 1$ .

Linked RBs indicate those RBs that have been assigned to the same MVNO on adjacent BSs. Specifically, a linked RB allows the corresponding MVNO to simultaneously access a specific spectrum portion in the same time-slot from two different BSs.

The reason why this feature is particularly relevant to the RSEP is threefold: (i) MVNOs can use linked RBs to enable advanced transmission schemes (*e.g.*, power control, beamforming, MIMO and CoMP transmissions) among nearby BSs; (ii) as shown in Fig. 1, linked RBs can be used to deploy fully-orthogonal RAN slices with no inter-MVNO interference; and (iii) linked RBs do not generate inter-MVNO interference, thus avoiding any need for centralized coordination or distributed coordination among MVNOs.

It is clear that the maximization of the number of simultaneously linked RBs addresses the three issues identified in Section I. Thus, we focus our attention on this approach. By leveraging the concept of linked RBs, we define  $n_{b,b',m}$  as follows:

$$n_{b,b',m} = y_{b,b'} \cdot |\pi_{m,b} \cap \pi_{m,b'}|, \quad (2)$$

where (2) represents the number of linked RBs among interfering BSs. It is also worth noting that the relationship  $n_{b,b',m} = n_{b',b,m}$  always holds for all  $b, b' \in \mathcal{B}$  and  $m \in \mathcal{M}$ .

For each MVNO  $m \in \mathcal{M}$ , the total number  $N_m$  of linked RBs on the corresponding RAN slice under policy  $\mathbf{L}$  is

$$N_m = \frac{1}{2} \sum_{b \in \mathcal{B}} \sum_{b' \neq b} y_{b,b'} \cdot n_{b,b',m}, \quad (3)$$

where the  $1/2$  factor is introduced to avoid double counting the same RBs.

We formally define the RSEP as follows:

$$\underset{\pi \in \Pi}{\text{maximize}} \quad \sum_{m \in \mathcal{M}} N_m \quad (\text{RSEP})$$

In a nutshell, the objective in RSEP is to compute a feasible slicing enforcement policy  $\pi$  that maximizes the total number of linked RBs while guaranteeing that the computed policy does not violate the feasibility constraint  $\pi \in \Pi$ .

## V. ADDRESSING THE RSEP PROBLEM

Fig. 1 shows that the formulation in Problem RSEP is particularly well-suited for RAN slicing problems. This is because it satisfies MVNOs requirements in terms of number of obtained RBs, helps orthogonality among slices through the reduction of inter-MVNO interference, and enables coordination-based communications such as CoMP, JT and beamforming.

To solve Problem RSEP, we need to compute a slicing enforcement policy by exploring the feasible set  $\Pi$ . Given the formulation in Problem RSEP does not in itself provide any intuitions on how a solution can be computed. For this reason, we (i) reformulate Problem RSEP by using the RB allocation indicators introduced in Section IV; (ii) show that the problem is NP-hard; and (iii) present a number of algorithms to compute both optimal and sub-optimal solutions to Problem RSEP.

### A. Optimal Solution

By using the definition of the RB allocation indicator  $x_{m,b,n,t}$  and from (1), we have that (3) can be reformulated as

$$N_m = \frac{1}{2} \sum_{t=1}^T \sum_{n=1}^{N_{RB}} \sum_{b \in \mathcal{B}} \sum_{b' \neq b, b' \in \mathcal{B}} y_{b,b'} x_{m,b,n,t} x_{m,b',n,t}. \quad (4)$$

Let us consider the matrices  $\mathbf{B} = \mathbf{Y} \otimes \mathbf{I}_{N_{RB} \cdot T}$  and  $\mathbf{Q} = \mathbf{I}_M \otimes \mathbf{B}$ , where  $\otimes$  stands for Kronecker product and  $\mathbf{I}_k$  is the  $k \times k$  identity matrix. From (4), it can be easily shown that  $\sum_{m \in \mathcal{M}} N_m = \frac{1}{2} \mathbf{x}^\top \mathbf{Q} \mathbf{x}$ . Accordingly, Problem RSEP can be reformulated as

$$\underset{\mathbf{x}}{\text{maximize}} \quad \frac{1}{2} \mathbf{x}^\top \mathbf{Q} \mathbf{x} \quad (\text{RSEP-QP})$$

$$\text{subject to} \quad \sum_{t=1}^T \sum_{n=1}^{N_{RB}} x_{m,b,n,t} = L_{m,b}, \quad \forall b \in \mathcal{B}, \forall m \in \mathcal{M} \quad (\text{C1})$$

$$\sum_{m \in \mathcal{M}} x_{m,b,n,t} \leq 1, \quad \forall (n,t) \in \mathcal{R}, \forall b \in \mathcal{B} \quad (\text{C2})$$

$$x_{m,b,n,t} \in \{0, 1\}, \quad \forall (n,t) \in \mathcal{R}, \forall b \in \mathcal{B}, \forall m \in \mathcal{M} \quad (\text{C3})$$

where  $\mathbf{x} = (x_{m,b,n,t})_{m,b,n,t}$  can be represented as a  $M B N_{RB} T \times 1$  column array.

In Problem RSEP-QP, Constraint (C1) ensures that all MVNOs receive the assigned number of RBs, while Constraint (C2) guarantees that each RB is allocated to one MVNO only.

Finally, Constraint (C3) expresses the boolean nature of the RB allocation indicator. Problems RSEP and RSEP-QP are equivalent, as the latter is a reformulation of the former in terms of the RB allocation indicator. However, this new formulation shows that the RSEP can be modeled as a 0-1 (or binary) Quadratic Programming (QP) problem. We prove in Theorem 1 that Problem RSEP-QP is NP-Hard.

**Theorem 1.** *Problem RSEP-QP is NP-hard.*

*Proof:* It is sufficient to show that the matrix  $\mathbf{Q}$  is *indefinite*, *i.e.*, it admits both positive and negative eigenvalues. Indeed, it is well-known [34, 35] that even real-valued non-binary QP problems are NP-hard when  $\mathbf{Q}$  is indefinite.

From the definition of  $\mathbf{B}$  and  $\mathbf{Y}$ , matrix  $\mathbf{Q}$  has all zero entries in the main diagonal. Accordingly,  $\mathbf{Q}$  is a zero-diagonal (or hollow) symmetric matrix. Let  $\lambda$  be the set of eigenvalues of  $\mathbf{Q}$ . Notice that  $\sum_{\lambda_i \in \lambda} \lambda_i = \text{Tr}\{\mathbf{Q}\}$ , and  $\text{Tr}\{\mathbf{Q}\} = 0$  in our case. Thus, all the eigenvalues of  $\mathbf{Q}$  must sum up to zero, meaning that either all eigenvalues are equal to zero, or  $\mathbf{Q}$  has both positive and negative eigenvalues. Thanks to the symmetry of  $\mathbf{Q}$ , we can exclude the former case since it would imply that  $\mathbf{Q}$  is the zero-matrix (*i.e.*, there is no interference among BSs and  $y_{b,b'} = 0$  for all  $b, b' \in \mathcal{B}$ ). Therefore,  $\mathbf{Q}$  must have both positive and negative eigenvalues, *i.e.*,  $\mathbf{Q}$  is indefinite. This proves the theorem.  $\blacksquare$

Since Problem RSEP-QP is NP-hard, in Section V-B we leverage linear relaxation and the concept of equivalence to design a reduced-complexity solution to Problem RSEP-QP, while in Section V-C we design a heuristic algorithm that can compute a sub-optimal solution to Problem RSEP-QP with polynomial complexity.

### B. Approximated Solution

Let  $V = M \cdot B \cdot N_{RB} \cdot T$ , and let us consider the following transformed problem

$$\underset{\mathbf{x}}{\text{maximize}} \quad \frac{1}{2} \mathbf{x}^\top (\mathbf{Q} + 2\lambda \mathbf{I}_V) \mathbf{x} - \lambda \mathbf{e}^\top \mathbf{x} \quad (\text{RSEP-EQ})$$

$$\text{subject to} \quad (\text{C1}), (\text{C2})$$

$$0 \leq x_{m,b,n,t} \leq 1, \quad \forall (n,t) \in \mathcal{R}, \forall b \in \mathcal{B}, \forall m \in \mathcal{M} \quad (\text{C4})$$

where  $\lambda \in \mathbb{R}$  is a real-valued parameter whose relevance to Problem RSEP-EQ will be explained in Theorem 2, and  $\mathbf{e}^\top = (1, 1, \dots, 1)$ . The following theorem holds.

**Theorem 2.** *There exists  $\lambda \in \mathbb{R}$  such that Problem RSEP-EQ is equivalent to Problem RSEP-QP. Also, let  $z^*$  be the largest (positive) eigenvalue of  $\mathbf{Q}$ . For any  $\lambda \geq -z^*$ , Problem RSEP-EQ is a quadratic convex problem over the unit hypercube.*

*Proof:* Notice that  $\mathbf{Q}$  contains only 0-1 entries and  $x_{m,b,n,t} \leq 1$ , which implies that  $\mathbf{x}^\top \mathbf{Q} \mathbf{x}$  is always bounded and finite. Also,  $\mathbf{x}^\top \mathbf{Q} \mathbf{x}$  has continuous and bounded first-order derivatives over the unit hypercube, *i.e.*, it is Lipschitz-continuous in any open set that contains the unit hypercube. From [36, Th. 3.1], it must exist  $\lambda_0 \in \mathbb{R}$  such that  $\forall \lambda \geq \lambda_0$  Problems RSEP-EQ and RSEP-QP are equivalent. Intuitively, the utility function in Problem RSEP-EQ introduces the term  $\lambda \mathbf{x}^\top (\mathbf{e} - \mathbf{x})$  which generates a cost, or a penalty, proportional

to  $\lambda$  when constraint  $x_{m,b,n,t} \in \{0, 1\}$  is not satisfied. Accordingly, the binary constraint in Constraint (C3) can be dropped and relaxed with the unit hypercube constraint  $0 \leq x_{m,b,n,t} \leq 1$ . Recall that  $\mathbf{Q}$  matrix admits both negative and positive eigenvalues. Accordingly, let  $\mathbf{z}$  be the set of eigenvalues of  $\mathbf{Q}$  and  $z^* = \max\{z_1, z_2, \dots, z_{|\mathbf{z}|}\}$ , we can choose  $\lambda \geq z^*$  to show that the matrix  $\mathbf{Q} + 2\lambda\tilde{\mathbf{I}}_V$  is positive semi-definite. Thus, Problem RSEP-EQ is convex, which proves the Theorem. ■

**Remarks.** Theorem 2 shows that it is possible to relax the binary constraint of Problem RSEP-QP by replacing it with a penalty term. This produces an equivalent convex QP problem where binary variables  $x_{m,b,n,t}$  are replaced with continuous ones through a linear relaxation. In general, local and global solutions of convex quadratic maximization problems (and the corresponding concave quadratic minimization problems) lie on the vertices of the feasibility set [37], thus making Problem RSEP-EQ easier to solve when compared to Problem RSEP-QP. In some cases, Problem RSEP-EQ might still require exponential time with respect to the number of vertices. Approaches that restrict the search space to the vertices of the feasibility set, such as cutting plane and extreme point ranking methods [37], can be used to efficiently solve Problem RSEP-EQ.

### C. Heuristic Solution

Although Problem RSEP-EQ has lower complexity than Problem RSEP-QP, it is still hard to find a solution as the problem grows in size. Therefore, we design a polynomial solution to Problem RSEP-QP by using a heuristic approach. The key idea is to generate a solution that provides good performance while achieving low computational complexity. Given Problem RSEP-QP maximizes the number of shared RBs, we can allocate as many linked RBs as possible to those MVNOs that request the highest amount of RBs on multiple interfering BSs. Indeed, MVNOs that request the greatest number of resources on different interfering BSs are also expected to produce a high number of linked RBs. Accordingly, for each MVNO  $m$  we define the *linking index*  $l_m$  as

$$l_m = \sum_{b \in \mathcal{B}} \sum_{b' \neq b} \min\{L_{m,b}, L_{m,b'}\} y_{b,b'} \quad (5)$$

The linking index is used to sequentially allocate RBs to those MVNOs with the highest linking index. We refer to this procedure as the *Most Linked First* (MLF) procedure, which is illustrated in Algorithm 1 and works as follows:

- 1) we generate set  $\mathcal{M}^G = \mathcal{M}$  for all  $m, k \in \mathcal{M}^G$  s.t.  $m < k$ , then  $l_m \geq l_k$ ;
- 2) we start allocating RBs on all BSs in sequential order to the first MVNO in  $\mathcal{M}^G$ , i.e., the MVNO whose linking index  $l_m$  is the highest among all MVNOs in  $\mathcal{M}$ . When all RBs are allocated to the considered MVNO, say  $m'$ , we remove it from  $\mathcal{M}^G$  and we set  $l_{m'} = 0$ ;
- 3) if  $\mathcal{M}^G = \emptyset$ , we stop. Otherwise, we re-execute Step 2 until all MVNOs are assigned to the required RBs.

Line 4 requires to compute (5) which has complexity  $\mathcal{O}(MB^2)$ , while Line 5 has complexity  $\mathcal{O}(M \log M)$ . The while loop at Line 6 has complexity  $\mathcal{O}(N_{RB}BM)$ . Thus, the complexity of MLF is  $\mathcal{O}(C)$ , where  $C = \max\{MB^2, M \log M, N_{RB} \cdot B \cdot M\}$ .

---

### Algorithm 1 RSEP-MLF

---

- 1: Input  $\mathcal{B}; \mathcal{M}; \mathbf{Y}; \mathbf{L}$ ;
  - 2: Output A MLF RBs allocation  $\mathbf{x}^G = (x_{m,b,n,t}^G)_{m,b,n,t}$ ;
  - 3: Set  $x_{m,b,n,t}^G = 0$  for all  $m \in \mathcal{M}, b \in \mathcal{B}, (n, t) \in \mathcal{R}$ ;
  - 4: Compute the linking index  $\mathbf{l} = (l_m)_{m \in \mathcal{M}}$ ;
  - 5:  $\mathcal{M}^G \leftarrow$  Sort  $\mathcal{M}$  by  $l_m$  in decreasing order;
  - 6: **while**  $\mathcal{M}^G \neq \emptyset$  **do**
  - 7:     **for** each BS  $b \in \mathcal{B}$  **do**
  - 8:         Update  $x_{m,b,n,t}^G$  by allocating  $L_{\mathcal{M}^G(1),b}$  subsequent RBs to MVNO  $m$  on BS  $b$ ;
  - 9:     **end for**
  - 10:      $\mathcal{M}^G \leftarrow \mathcal{M}^G \setminus \{\mathcal{M}^G(1)\}$ ;
  - 11: **end while**
- 

### D. Speeding-up the execution of RSEP-QP and RSEP-EQ

Although Problems RSEP-QP and RSEP-EQ have exponential complexity, two intuitions help reduce their complexity by leveraging specific structural properties of the RSEP.

1) *Sparsity*: Let  $\mathbf{x}^{OPT}$  be an optimal solution to either Problem RSEP-QP or RSEP-EQ. If  $L_{m,b} = 0$  for a given MVNO  $m$  on BS  $b$ , then  $x_{m,b,n,t}^{OPT} = 0$  for all  $n$  and  $t$ . Furthermore, we notice that the complexity of many optimization problems strongly depends on the number of non-zero entries (i.e., the sparsity) of the  $\mathbf{Q}$  matrix, as pointed out in [38]. Thus we leverage the particular structure of our problem to reduce the complexity of the two problems by introducing two transformations. Specifically, let  $m'$  and  $b'$  such that  $L_{m',b'} = 0$ , for both RSEP-QP and RSEP-EQ we generate a reduced matrix  $\tilde{\mathbf{Q}}$  where we set  $Q_{m',b',n,t} = 0$  for all  $(n, t) \in \mathcal{R}$ . For RSEP-QP, it suffices to replace the  $\mathbf{Q}$  matrix with  $\tilde{\mathbf{Q}}$ . Instead, to keep the equivalence between RSEP-QP and RSEP-EQ, the objective function of RSEP-EQ should be reformulated as

$$\frac{1}{2} \mathbf{x}^\top (\tilde{\mathbf{Q}} + 2\lambda\tilde{\mathbf{I}}_V) \mathbf{x} - \lambda \quad (6)$$

where  $\tilde{\mathbf{I}}_V$  is the identity matrix where we set to zero those entries corresponding to the 2-tuple  $(m', b')$ .

Note that the two above transformations generate equivalent problems to RSEP-QP and RSEP-EQ and do not impact the optimality of the computed solutions. In fact, Constraint (C1) requires  $\sum_{t=1}^T \sum_{n=1}^{N_{RB}} x_{m',b',n,t} = 0$  when  $L_{m',b'} = 0$ . Since  $x_{m',b',n,t} \in \{0, 1\}$ , we have that  $x_{m',b',n,t} = 0$  for all  $n$  and  $t$  associated to the 2-tuple  $(m', b')$ . That is, at the optimal solution,  $x_{m',b',n,t} = 0$  independently of the value of  $q_{m',b',n,t}$ .

2) *RB Aggregation*: Let  $K = \text{GCD}(\mathbf{L})$  be the greatest common divisor (GCD) among all of the elements in the  $\mathbf{L}$  matrix. We show that Problems RSEP-QP and RSEP-EQ are equivalent to solve the same problems with a scaled RB grid, when given conditions on  $K$ ,  $T$  and  $N_{RB}$  are satisfied. Specifically, if  $K > 1$  and either the number  $N_{RB}$  of RBs or the number  $T$  of time slots are proportional to  $K$ , the available RBs can be aggregated in groups of  $K$  RBs so that each group can be seen as a single aggregated RB. We refer to such a property as *aggregability* of the RSEP. Definition follows below.

**Definition 2** (Aggregable RSEP). The RSEP is said to be *aggregable* if  $N_{RB} \pmod{K} = 0$  or  $T \pmod{K} = 0$ , where  $K = \text{GCD}(\mathbf{L}) > 1$  and  $A \pmod{B}$  is the  $A$  modulo  $B$  operator.

In the first case, we scale the number of RBs as  $\tilde{N}_{RB} = N_{RB}/K$ . In the second case, we scale the number of time slots as  $\tilde{T} = T/K$ . That is, for each BS  $b \in \mathcal{B}$ , the set  $\mathcal{R}_b$  of available RBs at  $b$  is replaced with an aggregated version of cardinality  $|\tilde{\mathcal{R}}_b| = N_{RB}T/K$  where  $K$  RBs are grouped together to create a single RB. We refer to this low-dimensional RSEP as the *aggregated RSEP*.

**Theorem 3.** *Let the RSEP be aggregable, it is possible to compute an optimal solution to the RSEP by solving the aggregated RSEP.*

*Proof:* Let  $Z = N_{RB}T$ ,  $K > 1$  be the GCD of  $\mathbf{L}$ ,  $P$  be the original RSEP problem and  $\tilde{P}$  be the aggregated RSEP with  $\tilde{T} = T/K$ . The proof for the case where we aggregate with respect to  $\tilde{N}_{RB} = N_{RB}/K$  follows the same steps. First, note that problem  $P$  is shift invariant with respect to the indexing of  $n$  and  $t$ . Our statement can easily be proven by simply noting that for any given optimal solution  $\mathbf{x}^*$ , the solution  $\mathbf{x}$  with  $x_{m,r,1,t} = x_{m,r,N_{RB},t}^*$  and  $x_{m,r,N_{RB},t} = x_{m,r,1,t}^*$  for all  $m$ ,  $r$  and  $t$  is clearly still optimal as it produces the same number of linked RBs as  $\mathbf{x}^*$ . In general, we can extend this result to any reshape procedure of the RB set  $\mathcal{R}$  that maintains the cardinality of  $\mathcal{R}$  constant and equal to  $Z$ . With this feature at hand, we will show that we can reduce the cardinality of  $\mathcal{R}$  by a factor  $K$  and still achieve equivalence and optimality.

Let  $\mathcal{X} \in \mathbb{R}^{N_{RB}\tilde{T} \times K}$  and  $\tilde{\mathcal{X}} \in \mathbb{R}^{N_{RB}\tilde{T} \times 1}$  be the feasibility sets of  $P$  and  $\tilde{P}$ , respectively. Also, let  $f_Z(\mathbf{x}) : \mathcal{X} \rightarrow \mathbb{N}$  and  $f_{Z/K}(\mathbf{x}) : \tilde{\mathcal{X}} \rightarrow \mathbb{N}$  be the objective functions of problem  $P$  and  $\tilde{P}$ , respectively. The optimal solution to  $P$  is denoted as  $\mathbf{x}^* \in \mathcal{X}$ , while the optimal solution to  $\tilde{P}$  is denoted as  $\tilde{\mathbf{x}}^* \in \tilde{\mathcal{X}}$ . Due to the optimality of  $\mathbf{x}^*$  and  $\tilde{\mathbf{x}}^*$ , we have that  $f_Z(\mathbf{x}^*) \geq f_Z(\mathbf{x})$  for all  $\mathbf{x} \in \mathcal{X}$ , and  $f_{Z/K}(\tilde{\mathbf{x}}^*) \geq f_{Z/K}(\tilde{\mathbf{x}})$  for all  $\tilde{\mathbf{x}} \in \tilde{\mathcal{X}}$ . Let us define  $\tilde{\mathbf{x}}_A^* \in \tilde{\mathcal{X}}$  be the solution to  $\tilde{P}$  generated by expanding the aggregated optimal solution  $\tilde{\mathbf{x}}^*$  to  $\tilde{P}$ . Let  $\mathcal{R} \in \mathbb{R}^{N_{RB}\tilde{T} \times K}$ ,  $\tilde{\mathbf{x}}_A^* = (\tilde{x}_{A_{m,b,\tau,k}^*})_{m,b,\tau,k}$  is generated by setting  $\tilde{x}_{A_{m,b,\tau,k}^*} = \tilde{x}_{m,b,\tau}^*$  for all  $k = 1, \dots, K$ ,  $m \in \mathcal{M}$  and  $b \in \mathcal{B}$ . Intuitively, we are replicating the matrix  $\tilde{\mathbf{x}}^*$  by adding  $K - 1$  rows whose entries are identical to those in  $\tilde{\mathbf{x}}^*$ .

We will now prove that  $P$  and  $\tilde{P}$  are equivalent by contradiction. Accordingly, we will negate our hypothesis and we will assume that the two problems are not equivalent, i.e.,  $f_Z(\mathbf{x}^*) > f_Z(\tilde{\mathbf{x}}_A^*)$ .

Let  $g(\mathbf{x}) : \mathcal{X} \rightarrow \mathbb{N}$  be defined as  $g(\mathbf{x}) = K^{-1}f_Z(\mathbf{x})$ . Intuitively, if we replace the objective function  $f_Z(\mathbf{x})$  of  $P$  with  $g(\mathbf{x})$ , we obtain the same problem where each linked RB gives a reward equal to  $K^{-1}f_Z(\mathbf{x})$  instead provides a unitary reward for each linked RB. By construction of  $\tilde{\mathbf{x}}_A^*$ , we have  $f_{Z/K}(\tilde{\mathbf{x}}^*) = K^{-1}f_Z(\tilde{\mathbf{x}}_A^*) = g(\tilde{\mathbf{x}}_A^*)$ . From the assumption  $f_Z(\mathbf{x}^*) > f_Z(\tilde{\mathbf{x}}_A^*)$ , we have that

$$g(\mathbf{x}^*) = K^{-1}f_Z(\mathbf{x}^*) > K^{-1}f_Z(\tilde{\mathbf{x}}_A^*) = g(\tilde{\mathbf{x}}_A^*) = f_{Z/K}(\tilde{\mathbf{x}}_A^*) \quad (7)$$

which states that  $g(\mathbf{x}^*) > f_{Z/K}(\tilde{\mathbf{x}}_A^*)$ .

To show that this last statement is a contradiction to our hypothesis (i.e.,  $\tilde{\mathbf{x}}^*$  is optimal for  $\tilde{P}$ ), we need to show that there always exist a mapping that transforms any solution in  $\mathcal{X}$  to an equivalent solution in  $\tilde{\mathcal{X}}$ . That is, we need to find a function  $h(\mathbf{x}) : \mathcal{X} \rightarrow \tilde{\mathcal{X}}$  such that  $h(\mathbf{x}) = \tilde{\mathbf{x}} \in \tilde{\mathcal{X}}$  that can be transformed into  $\tilde{\mathbf{x}}_A$  such that  $f_{Z/K}(\tilde{\mathbf{x}}) = K^{-1}f_Z(\tilde{\mathbf{x}}_A)$ .

In general, such a mapping is not unique, since any optimal solution in  $\mathcal{X}$  and  $\tilde{\mathcal{X}}$  is shift invariant. However, in Appendix A we present an easy mapping  $h(\mathbf{x}) = \tilde{\mathbf{x}}$  that, starting from an optimal solution  $\mathbf{x} \in \mathcal{X}$ , always generates an equivalent optimal solution  $\tilde{\mathbf{x}} \in \tilde{\mathcal{X}}$  such that  $f_{Z/K}(\tilde{\mathbf{x}}) = K^{-1}f_Z(\mathbf{x})$ .

The existence of the above mapping implies that  $\mathbf{x}_R^* = h(\mathbf{x}^*)$  satisfies  $f_{Z/K}(\mathbf{x}_R^*) = K^{-1}f_Z(\mathbf{x}^*) = g(\mathbf{x}^*)$ , which is clearly a contradiction. In fact, from (7) we have that  $f_{Z/K}(\mathbf{x}_R^*) = g(\mathbf{x}^*) > f_{Z/K}(\tilde{\mathbf{x}}^*)$ , which implies the existence of a solution  $\mathbf{x}_R^*$  that contradicts the optimality of  $\tilde{\mathbf{x}}^*$  over  $\tilde{\mathcal{X}}$ . It follows that  $f_{Z/K}(\tilde{\mathbf{x}}^*) = K^{-1}f_Z(\mathbf{x}^*)$  must hold. Hence, any solution  $\tilde{\mathbf{x}}^*$  to the aggregated RSEP can be expanded to obtain  $\tilde{\mathbf{x}}_A^*$  that is optimal for the original RSEP. This concludes the proof. ■

## VI. NUMERICAL ANALYSIS

We now assess the performance of the algorithms proposed in Section V. To this end, we simulate an LTE frequency division duplexing (FDD) system with 1.4 MHz channel bandwidth, which is divided into 72 subcarriers organized into  $N_{RB} = 6$  physical resource blocks (PRBs). Each PRB represents the minimum scheduling unit and consists of 12 subcarriers and 14 symbols. Time is divided into discrete time slots called *sub-frames*, the duration of each sub-frame equals the duration of one PRB and  $N_{SF} = 10$  sub-frames constitute a *frame*. Let  $N_F \in \mathbb{N}$  be the number of frames within the slicing enforcing window. It follows that  $T = N_F \cdot N_{SF}$ .

Unless stated otherwise, we assume that both the interference matrix  $\mathbf{Y}$  and the slicing profile matrix  $\mathbf{L}$  are generated at random at each simulation run. Results were averaged over 1000 independent simulation runs.

### A. Convergence Time Analysis

Fig. 4 shows the convergence time of RSEP-QP, RSEP-EQ and RSEP-MLF as a function of the number  $M$  of MVNOs when  $B = 5$  and  $N_F = 2$ . As expected, the convergence time increases as the number of MVNOs in the network increases. Moreover, the convergence time of RSEP-QP is considerably higher than the one of RSEP-EQ and RSEP-MLF. Fig. 4 also shows the impact of the sparsity and RB aggregation mechanisms in Sections V-D1 and V-D2 on the overall convergence time. As it can be observed, the techniques presented in Section V-D can effectively reduce the computation time of all the three problems. Moreover, we show the RB aggregation produces the best performance improvement in terms of convergence time.

*We point out that RSEP-QP requires approximately 100s to compute an optimal solution when  $M = 10$  and  $B = 5$ . On the contrary, RSEP-EQ only requires 1s while RSEP-MLF computes the solution in less than a millisecond.*

Interestingly, Fig. 4 reveals that the reduction in terms of convergence time brought by sparsity can not be appreciated in small-scale scenarios. For this reason, we have further investigated the impact of sparsity in large-scale networks and the obtained results are presented in Fig. 5. It is shown that sparsity can effectively reduce the computation time by several tens of seconds, and the gain increases as both  $M$  and  $B$  increase.

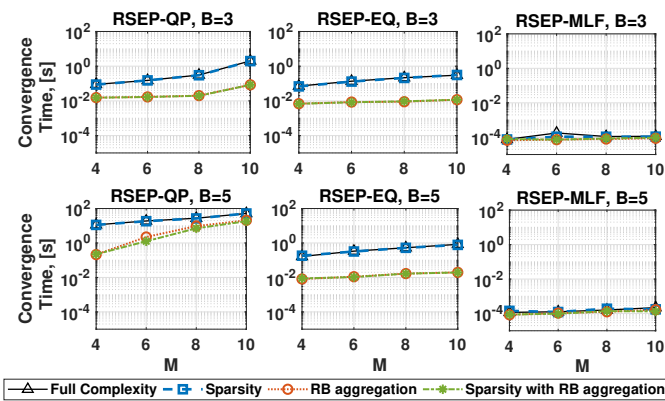


Fig. 4. Convergence time (s) of the three proposed solutions as a function of  $M$  considering different computational time reduction techniques.

### B. Optimality-gap Analysis

Another crucial aspect is the optimality-gap between RSEP-QP and RSEP-EQ/RSEP-MLF. Although Theorem 2 shows that (under some conditions) Problem RSEP-EQ is equivalent to Problem RSEP-QP, we can not guarantee that the solution computed by RSEP-EQ is a global optimum. Indeed, the solver might get stuck in one of the local maximizers, thus effectively preventing the computation of an actual global maximizer. Thus, in Fig. 6 we investigate the optimality-gap of RSEP-EQ and RSEP-MLF with respect to an optimal solution computed by RSEP-QP. In other words, the closer to zero is the optimality-gap, the closer to optimality the solution is.

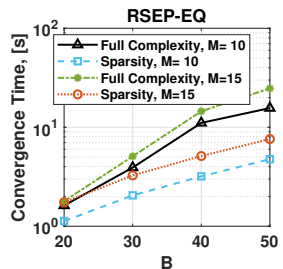


Fig. 5. Convergence time (s) of RSEP-EQ as a function of  $B$  considering different number  $M$  of MVNOs.

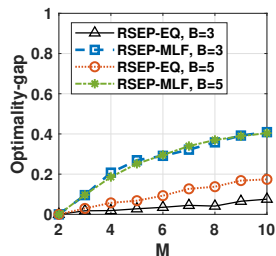


Fig. 6. Optimality-gap of RSEP-EQ and RSEP-MLF as a function of  $M$  considering different number  $B$  of BSs.

Fig. 6 shows that the optimality-gap increases as the number of MVNOs and BSs in the network increases. Intuitively, this is because the feasibility set increases as  $M$  and/or  $B$  increase. Given that local maximizers of RSEP-EQ lie on the vertices of the feasibility set, greater values of  $M$  and  $B$  produce a greater number of local maximizers, thus the probability of getting stuck in a local maximizer increases. Notice that although RSEP-MLF is negligibly affected by the number of BSs  $B$ , it achieves poor performance if compared to RSEP-EQ. Figs. 6 and 4 also indicate that RSEP-EQ represents an effective solution to the RSEP that is both near-optimal and computed in few seconds.

### C. Linked RBs Analysis

Fig. 7 shows the impact of  $M$  and  $B$  on the total number of linked RBs of the system when  $N_F = 10$  and  $T = 100$ . As

expected, RSEP-EQ always performs better than RSEP-MLF in terms of number of linked RBs. Moreover, Fig. 7 illustrates that larger values of  $B$  produce a greater number of linked RBs. Conversely, the number of linked RBs decrease as the number  $M$  of MVNOs increase. This is because, when more MVNOs include the same BS to their slices, it is harder to guarantee that all MVNOs will receive the corresponding amount of RBs joint with a large number of linked RBs.

## VII. EXPERIMENTAL ANALYSIS

In this section, we describe our testbed analysis and the results obtained through experimental evaluation. First, we describe in Section VII-A our experimental testbed and the scenario considered. Finally, we discuss the experimental results in Section VII-B.

### A. Experimental Testbed and Network Scenarios

To evaluate the performance of our algorithms, we have used an orthogonal frequency division multiple access (OFDMA) system as in LTE [39]. In OFDMA, time and frequency are divided into RBs, where each RB has time duration equal to the duration of an OFDMA frame, which spans  $K$  subcarriers split into  $\Delta T$  slots. Each of these slots are assigned to one MVNO according to the outcome of the slicing enforcement algorithm, who in turn can decide to assign subcarriers to MUs according to its own internal resource allocation and scheduling policy.

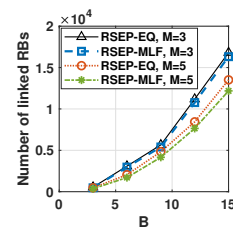


Fig. 7. Total number of linked RBs for RSEP-EQ and RSEP-MLF as a function of  $B$  for different values of  $M$ .

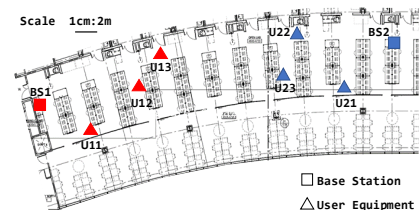


Fig. 8. Experimental Testbed.

We set up a testbed of 8 USRP software-defined radio for our experiments. Two USRP X310 act as BSs, 6 USRP N210 are used to implement MUs and all of them run GNU Radio. The USRP radios were deployed as shown in Figure 8, where we show that two groups of three N210s are associated with each BS. The BSs were synchronized in time and phase by using an *Octoclock* clock distributor by Ettus Research.

### B. Experimental Results

Our experiments were targeted to evaluate two critical performance parameters:

- 1) the effectiveness of our slicing enforcement algorithms in assigning the spectrum resources (i.e., the RBs) to the MVNOs according to the slicing policy;
- 2) the performance (i.e., total network throughput) of RSEP-QP as opposed to sub-optimal algorithms.

To address point 1), we consider a network with three MVNOs and 6 MUs, where each MU is associated with a different MVNO in each BS (e.g.,  $U_{11}$  is associated with MVNO

1 and BS1, while  $U_{23}$  with MVNO 3 and BS2). We consider the case where the two BSs enforce the following slicing policies on the three MVNOs, which change every  $T = 300$ s, by using the RSEP-QP algorithm.

- 1) BS1:  $\{70, 30, 0\}\%$ ;  $\{25, 25, 50\}\%$ ;  $\{30, 40, 30\}\%$ .
- 2) BS2:  $\{25, 25, 50\}\%$ ;  $\{30, 40, 30\}\%$ ;  $\{70, 30, 0\}\%$ .

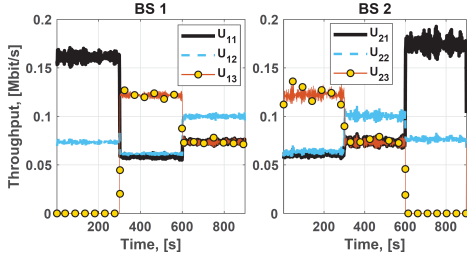


Fig. 9. Throughput per MU as a function of time and MUs.

Figure 9 depicts the throughput experienced by each MU as a function of time at BS1 and BS2, averaged over 10 repetitions. The sharp change in throughput corresponding to different slicing policies in Figure 9 indicates that our RSEP-QP slicing enforcement algorithm indeed assigns to each MVNO (and therefore, each MU) a number of RBs that is coherent to what expressed in the slicing policy.

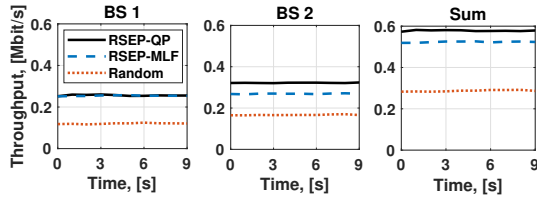


Fig. 10. Total network throughput for different slicing enforcement algorithms.

In addressing point 2), we consider the same network scenario as in 1). However, to remark the difference between the optimum and the sub-optimal algorithms, we consider 9 MVNOs instead of 3. The MVNOs have been assigned the following slicing policy at the two BSs:

- 1) BS1:  $\{7, 10, 18, 7, 3, 6, 23, 14, 12\}\%$ ;
- 2) BS2:  $\{5, 3, 16, 13, 10, 5, 30, 7, 11\}\%$ .

Furthermore, we consider the following IBSPC policy: when two MVNOs have matching RBs at the two BSs, we compute a power control policy that reduce the transmission power to keep the mutual interference below a fixed threshold.

Figure 10 shows the total network throughput as a function of time, for the RSEP-QP, RSEP-MLF and *Random* policies, averaged over 100 repetitions. The results in Figure 10 indicate that (i) as remarked in the Introduction, the throughput increase by using IBSPC in conjunction to an optimal slicing enforcement algorithm can help increase the total network throughput dramatically with respect to approaches that assign RBs to MVNOs in sub-optimal fashion. Our experiments confirm that RSEP-QP doubles the throughput with respect to the *Random* baseline, and that our RSEP-MLF algorithm approximates well the RSEP-QP algorithm. These results are a consequence of the fact that the RBs shared by the MVNOs are respectively 96,

79, and 13 in the case of RSEP-QP, RSEP-MLF and *Random*, therefore, the opportunity of IBSPC is significantly higher in the former two.

## VIII. CONCLUSIONS

In this paper, we have investigated the challenging and timely problem of radio access network (RAN) slicing enforcement in 5G networks. First, we have formulated the resource slicing enforcement problem (RSEP) and shown its NP-hardness. Then, we have proposed two approximation algorithms that render the problem tractable and scalable as the problem increases in complexity. Finally, we have evaluated the algorithms through extensive simulation and experimental analysis on a real-world testbed composed by 8 software-defined radios and GNU Radio. Results conclude that our algorithms are scalable and provide near-optimal performance. Moreover, our solutions effectively enforce RAN slicing policies by satisfying MVNOs requirements and by reducing inter-MVNO interference.

## REFERENCES

- [1] Ericsson Incorporated, “Ericsson Interim Mobility Report, June 2018,” <https://www.ericsson.com/assets/local/mobility-report/documents/2018/ericsson-mobility-report-june-2018.pdf>, 2018.
- [2] Federal Communications Commission (FCC), “Spectrum Crunch,” <https://www.fcc.gov/general/spectrum-crunch>.
- [3] A. Nakao, P. Du, Y. Kiriha, F. Granelli, A. A. Gebremariam, T. Taleb, and M. Bagaa, “End-to-end network slicing for 5G mobile networks,” *Journal of Information Processing*, vol. 25, pp. 153–163, 2017.
- [4] R. Kokku, R. Mahindra, H. Zhang, and S. Rangarajan, “CellSlice: Cellular wireless resource slicing for active RAN sharing,” in *Proc. of IEEE Intl. Conf. on Communication Systems and Networks (COMSNETS)*, 2013.
- [5] X. Foukas, M. K. Marina, and K. Kontovasilis, “Orion: RAN slicing for a flexible and cost-effective multi-service mobile network architecture,” in *Proc. of ACM Intl. Conf. on Mobile Computing and Networking (MobiCom)*, 2017.
- [6] P. Rost, C. Mannweiler, D. S. Michalopoulos, C. Sartori, V. Sciancalepore, N. Sastry, O. Holland, S. Tayade, B. Han, D. Bega *et al.*, “Network slicing to enable scalability and flexibility in 5G mobile networks,” *IEEE Communications magazine*, vol. 55, no. 5, pp. 72–79, 2017.
- [7] E. Hossain, M. Rasti, H. Tabassum, and A. Abdelnasser, “Evolution toward 5G multi-tier cellular wireless networks: An interference management perspective,” *IEEE Wireless Communications*, vol. 21, no. 3, pp. 118–127, 2014.
- [8] S. D’Oro, A. Zappone, S. Palazzo, and M. Lops, “A learning approach for low-complexity optimization of energy efficiency in multicarrier wireless networks,” *IEEE Transactions on Wireless Communications*, vol. 17, no. 5, pp. 3226–3241, 2018.
- [9] V. Jungnickel, K. Manolakis, W. Zirwas, B. Panzner, V. Braun, M. Losow, M. Sternad, R. Apelfrojd, and T. Svensson, “The role of small cells, coordinated multipoint, and massive MIMO in 5G,” *IEEE Communications Magazine*, vol. 52, no. 5, pp. 44–51, 2014.
- [10] R. Irmer, H. Droste, P. Marsch, M. Grieger, G. Fettweis, S. Brueck, H.-P. Mayer, L. Thiele, and V. Jungnickel, “Coordinated multipoint: Concepts, performance, and field trial results,” *IEEE Communications Magazine*, vol. 49, no. 2, pp. 102–111, 2011.
- [11] J. Lee, Y. Kim, H. Lee, B. L. Ng, D. Mazzaresse, J. Liu, W. Xiao, and Y. Zhou, “Coordinated multipoint transmission and reception in LTE-advanced systems,” *IEEE Communications Magazine*, vol. 50, no. 11, 2012.
- [12] D. Boviz and Y. E. Mghazli, “Fronthaul for 5G: Low bit-rate design enabling joint transmission and reception,” in *Proc. of IEEE Global Communications Conf. (GLOBECOM) – Workshops*, Dec 2016.
- [13] W. Nam, D. Bai, J. Lee, and I. Kang, “Advanced interference management for 5G cellular networks,” *IEEE Communications Magazine*, vol. 52, no. 5, pp. 52–60, 2014.
- [14] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, “Network slicing & softwarization: A survey on principles, enabling technologies & solutions,” *IEEE Communications Surveys & Tutorials*, 2018.



- [15] A. Kaloylos, "A survey and an analysis of network slicing in 5G networks," *IEEE Communications Standards Magazine*, vol. 2, no. 1, pp. 60–65, 2018.
- [16] K. Samdanis, S. Wright, A. Banchs, A. Capone, M. Ulema, and K. Obana, "5G network slicing - part 1: Concepts, principles, and architectures," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 70–71, May 2017.
- [17] C.-Y. Chang, N. Nikaen, and T. Spyropoulos, "Radio access network resource slicing for flexible service execution," in *Proc. of IEEE Intl. Conf. on Computer Communications (INFOCOM) – Workshops*, 2018.
- [18] O. Sallent, J. Perez-Romero, R. Ferrus, and R. Agusti, "On radio access network slicing from a radio resource management perspective," *IEEE Wireless Communications*, vol. 24, no. 5, pp. 166–174, 2017.
- [19] R. Ferrus, O. Sallent, J. Perez-Romero, and R. Agusti, "On 5g radio access network slicing: Radio interface protocol features and configuration," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 184–192, May 2018.
- [20] P. Zhao, H. Tian, S. Fan, and A. Paulraj, "Information prediction and dynamic programming based ran slicing for mobile edge computing," *IEEE Wireless Communications Letters*, 2018.
- [21] P. Caballero, A. Banchs, G. de Veciana, and X. Costa-Pérez, "Multi-tenant radio access network slicing: Statistical multiplexing of spatial loads," *IEEE/ACM Transactions on Networking*, vol. 25, no. 5, pp. 3044–3058, 2017.
- [22] A. A. Gebremariam, M. Chowdhury, M. Usman, A. Goldsmith, and F. Granelli, "SoftSLICE: Policy-based dynamic spectrum slicing in 5G cellular networks," in *Prof. of IEEE Intl. Conf. on Communications (ICC)*, May 2018.
- [23] P. L. Vo, M. N. H. Nguyen, T. A. Le, and N. H. Tran, "Slicing the edge: Resource allocation for RAN network slicing," *IEEE Wireless Communications Letters*, pp. 1–1, 2018.
- [24] P. Caballero, A. Banchs, G. de Veciana, and X. Costa-Prez, "Network slicing games: Enabling customization in multi-tenant networks," in *Proc. of IEEE Intl. Conf. on Computer Communications (INFOCOM)*, May 2017.
- [25] Y. Jia, H. Tian, S. Fan, P. Zhao, and K. Zhao, "Bankruptcy game based resource allocation algorithm for 5G cloud-RAN slicing," in *Proc. of IEEE Wireless Communications and Networking Conf. (WCNC)*, April 2018.
- [26] O. Narmanlioglu and E. Zeydan, "Learning in SDN-based multi-tenant cellular networks: A game-theoretic perspective," in *Proc. of IFIP/IEEE Intl. Symp. on Integrated Network and Service Management (IM)*, 2017.
- [27] S. D'Oro, F. Restuccia, T. Melodia, and S. Palazzo, "Low-complexity distributed radio access network slicing: Algorithms and experimental results," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, 2018.
- [28] A. Devlic, A. Hamidian, D. Liang, M. Eriksson, A. Consoli, and J. Lundstedt, "NESMO: Network slicing management and orchestration framework," in *Proc. of IEEE Intl. Conf. on Communications (ICC) – Workshops*, May 2017.
- [29] R. Ferrus, O. Sallent, J. Perez-Romero, and R. Agusti, "On 5G radio access network slicing: Radio interface protocol features and configuration," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 184–192, 2018.
- [30] A. Ksentini and N. Nikaen, "Toward enforcing network slicing on RAN: Flexibility and resources abstraction," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 102–108, 2017.
- [31] B. Han, J. Lianghai, and H. D. Schotten, "Slice as an evolutionary service: Genetic optimization for inter-slice resource management in 5g networks," *IEEE Access*, vol. 6, pp. 33 137–33 147, 2018.
- [32] R. Mahindra, M. A. Khojastepour, H. Zhang, and S. Rangarajan, "Radio access network sharing in cellular networks," in *Proc. of IEEE Intl. Conf. on Network Protocols (ICNP)*, 2013.
- [33] E. Dahlman, S. Parkvall, and J. Skold, *4G: LTE/LTE-advanced for mobile broadband*. Academic press, 2013.
- [34] P. M. Pardalos and S. A. Vavasis, "Quadratic programming with one negative eigenvalue is NP-hard," *Journal of Global Optimization*, vol. 1, no. 1, pp. 15–22, 1991.
- [35] S. Sahni, "Computationally related problems," *SIAM Journal on Computing*, vol. 3, no. 4, pp. 262–279, 1974.
- [36] F. Giannessi and F. Tardella, *Connections between Nonlinear Programming and Discrete Optimization*. Springer US, 1999, pp. 149–188.
- [37] C. A. Floudas and V. Visweswaran, "Quadratic optimization," in *Handbook of global optimization*. Springer, 1995, pp. 217–269.
- [38] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
- [39] D. Astély, E. Dahlman, A. Furuskär, Y. Jading, M. Lindström, and S. Parkvall, "LTE: the evolution of mobile broadband," *IEEE Communications magazine*, vol. 47, no. 4, 2009.

## IX. APPENDIX

### A. Aggregation map from $\mathcal{X}$ to $\tilde{\mathcal{X}}$

Let us consider the reshaped RB grid  $\mathcal{R} \in \mathbb{R}^{N_{RB}T \times 1}$  and let us define the RBAM  $\sigma = (\sigma_b)_{b \in \mathcal{B}}$  where  $\sigma_b(\mathbf{x}^*) = (\sigma_{b,\tau})_{\tau \in \mathcal{R}}$ . Henceforth,  $b$  and  $\tau$  will represent rows and columns of  $\sigma$ , respectively. For any given optimal solution  $\mathbf{x} \in \mathcal{X}$ , we build a map between each RB in  $\mathcal{R}$  and the MVNO that has been assigned with that RB on BS  $b$ . Let  $M_{b,\tau}(\mathbf{x}^*)$  be the MVNO that RB  $\tau$  has been assigned to, *i.e.*, the MVNO  $m$  such that  $x_{m,b,\tau} = 1$ . Accordingly, we set  $\sigma_{b,\tau} = M_{b,\tau}(\mathbf{x}^*)$ .

Let us now introduce some terminology for the sake of simplicity. Two columns  $\tau_1$  and  $\tau_2$  are said to be *coherently swapped* when all their corresponding entries  $\sigma_{b,\tau_1}$  are replaced with those of  $\sigma_{b,\tau_2}$  and *vice versa* for all  $b \in \mathcal{B}$ . Two columns are *partially swapped* when only a portion  $\hat{\mathcal{B}} \subset \mathcal{B}$  of entries is replaced among two columns. Two entries  $\sigma_{b_1,\tau}$  and  $\sigma_{b_2,\tau}$  are *linked* if  $M_{b_1,\tau}(\mathbf{x}) = M_{b_2,\tau}(\mathbf{x})$  and  $y_{b_1,b_2} = 1$ . Finally, we say that  $K$  adjacent entries  $\sigma_{b,\tau_1}, \dots, \sigma_{b,\tau_K}$  are *paired* if  $M_{b,\tau_1}(\mathbf{x}) = M_{b,\tau_2}(\mathbf{x}) = \dots = M_{b,\tau_K}(\mathbf{x})$ , they are said to be *unpaired* otherwise.

The algorithm works as follow. First, if any  $K$  columns of  $\sigma$  are identical, *i.e.*, all entries are paired, we remove them from  $\sigma$  and add one out of those  $K$  identical columns to an aggregated RBAM  $\sigma^A \in \mathbb{R}^{N_{RB}T/K \times 1}$ . Then, we take the following steps:

- 1) We select the row  $b_0$  of  $\sigma$  with the smallest number of distinct MVNOs and we move it to the lowest row;
- 2) We update  $\sigma$  by ordering row  $b_0$  in MVNO identifier order. This operation (i) creates ordered groups of  $K$  entries; and (ii) preserves the optimality of the solution as all columns are coherently swapped;
- 3) If all entries in the RBAM  $\sigma$  have been paired, we stop the algorithm;
- 4) If any  $K$  columns of  $\sigma$  are identical, we remove them from  $\sigma$  and we include one of them to  $\sigma^A$ ;
- 5) We select the row  $b_n$  (among the rows above  $b_0$ ) that shares the highest number of links with  $b_0$ , and we move it above  $b_0$ ;
- 6) If all the entries in  $b_n$  are paired, we go to 7); otherwise, we find  $K$  unpaired entries and we generate a partial swap of  $b_n$  and the upper rows such that i) the number of links remains the same<sup>1</sup>; and ii) the  $K$  entries are paired. *Since we are forcing the number of links to be the same, any partial swap generated in this step maintains the optimality of the solution. Although the partial swap might change the number of links per tenant, it does not changes the total number of links. Thus, the solution generated by the partial swap and the initial optimal solution are equivalent*;
- 7) We set  $b_0 = b_n$  and go to 3).

Upon termination, the algorithm creates the aggregated RBAM  $\sigma^A$  that is then transformed into a reshaped one  $\sigma^R$  by replicating its columns exactly  $K - 1$  times. All the entries in the reshaped RBAM  $\sigma^R$  are paired and the total number of links is equal to the original optimal RBAM  $\sigma$ . It is easy to

<sup>1</sup>Note that the initial unpaired solution is optimal, and any partial swap produces a number of links that is at most as high as that of the initial solution.

note that both RBAMs generate the same number of links, *i.e.*, the aggregation mapping generates an aggregated RBAM that is optimal for the RSEP.