Coordinated 5G Network Slicing: How Constructive Interference Can Boost Network Throughput

Salvatore D'Oro, *Member, IEEE*, Leonardo Bonati, *Student Member, IEEE*,
Francesco Restuccia, *Member, IEEE, ACM*,
and Tommaso Melodia, *Fellow, IEEE, Senior Member, ACM*

***Abstract*— Radio access network (RAN) slicing is a virtualization technology that partitions radio resources into multiple autonomous virtual networks. Since RAN slicing can be tailored to provide diverse performance requirements, it will be pivotal to achieve the high-throughput and low-latency communications that next-generation (5G) systems have long yearned for. To this end, effective RAN slicing algorithms must (i) partition radio resources so as to leverage coordination among multiple base stations and thus boost network throughput; and (ii) reduce interference across different slices to guarantee slice isolation and avoid performance degradation. *The ultimate goal of this paper is to design RAN slicing algorithms that address the above two requirements.* First, we show that the RAN slicing problem can be formulated as a 0-1 Quadratic Programming problem, and we prove its NP-hardness. Second, we propose an optimal solution for small-scale 5G network deployments, and we present three approximation algorithms to make the optimization problem tractable when the network size increases. We first analyze the performance of our algorithms through simulations, and then demonstrate their performance through experiments on a standard-compliant LTE testbed with 2 base stations and 6 smartphones. Our results show that not only do our algorithms efficiently partition RAN resources, but also improve network throughput by 27% and increase by 2× the signal-to-interference-plus-noise ratio.**

***Index Terms*— Network slicing, 5G, radio access network (RAN), interference management.**

I. Introduction

THE sheer number of mobile subscriptions worldwide—predicted to be around 8.9 billions by the end of 2025 [1]—will generate amounts of traffic that currently commercially-available wireless infrastructures and spectrum bands are not able to support [2]. Critically, traditional one-size-fits-all resource allocation policies will not enable *dynamic, effective and efficient* radio access strategies, which motivates the already increasing demand for novel solutions to design and deploy faster, lower-latency wireless cellular connections [3], [4].

To address the above issues, radio access network (RAN) slicing has been recently welcomed as a promising approach by the academia and industry alike [5]–[15]. This technology allows multiple mobile virtual network operators (MVNOs) to share the same physical infrastructure—ultimately realizing a game-changing vision that completely overturns the traditional model of single ownership of all network resources. Although a similar concept is already widely applied in the context of cloud computing by companies such as Amazon and Microsoft [16], *RAN slicing is an intrinsically different problem as: (i) Spectrum is a scarce resource for which over-provisioning is not possible [17], and (ii) interference jeopardizes isolation across slices belonging to different MVNOs, thus resulting in performance degradation if not handled properly* [10], [18].

As shown in Fig. 1, in RAN slicing applications each MVNO controls a separate "slice" of the network. Slices can be assigned/revoked by the Infrastructure Provider (IP) which determines the slices to be admitted to the system and how many resources each slice should receive. Once RAN slicing policies have been defined, a key problem is how to allocate the spectrum resource blocks (RBs) as prescribed by the slicing policy. This problem, also referred to as the RAN slicing enforcement problem (RSEP) [10], ensures that if an MVNO has been assigned a slice of 15% of the spectrum resources, such MVNO receives approximately 15% of the available RBs.

The design and evaluation of *RAN slicing enforcement* algorithms is paramount to implement in practice the slicing policy of the IP. Moreover, to be effective, RAN slicing enforcement algorithms must facilitate interference-mitigating strategies such as inter-base station power control (IBSPC) [10], [19], [20], MIMO [21], and coordinated multi-point (CoMP) [21]–[23] schemes such as Joint Transmission (JT) [24], [25]. Therefore, *it becomes imperative to design effective and efficient slicing enforcement algorithms assigning the same (or similar in time/frequency) RBs to the same MVNOs when BSs interfere among themselves.*

To illustrate the above point, we consider the cellular network scenario depicted in Fig. 1. Here, the IP 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: $\mathrm{M1} = 25\%, \mathrm{M2} = 50\%, \mathrm{M3} = 25\%$, on both the BSs. Fig. 1a shows an optimum slicing enforcement, represented as two *RB allocation matrices* (RBAMs), 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 cellular users

Manuscript received January 24, 2020; revised September 18, 2020; accepted April 6, 2021; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor T. Spyropoulos. This work was supported in part by the ONR under Grant N00014-19-1-2409 and Grant N00014-20-1-2132 and in part by the NSF under Grant CNS-1618727. *(Corresponding author: Salvatore D'Oro.)*

The authors are with the Institute for the Wireless Internet of Things, Northeastern University, Boston, MA 02115 USA (e-mail: s.doro@northeastern.edu; l.bonati@northeastern.edu; f.restuccia@northeastern.edu; t.melodia@northeastern.edu).

Digital Object Identifier 10.1109/TNET.2021.3073272

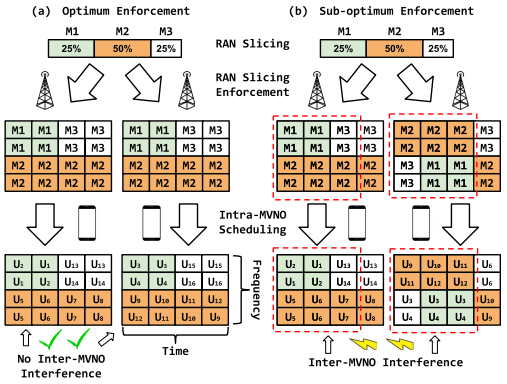


Fig. 1. Optimum and sub-optimum RAN slicing enforcement.

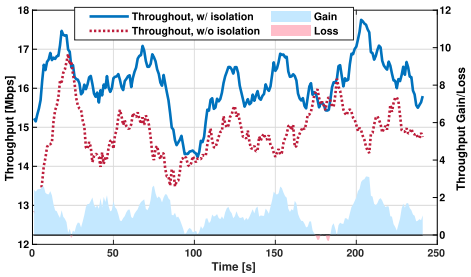


Fig. 2. Impact of coordination-based slicing on network throughput. Lines represents throughput measurements, shaded areas indicate gains and losses.

residing in the two BSs by using IBSPC. Conversely, Fig. 1b shows sub-optimum RBAMs causing inter-MVNO interference during 12 RBs, which will likely result in performance degradation due to poor interference management.

To further demonstrate the negative impact of inter-MVNO interference, we ran a series of experiments on the LTE-compliant testbed described in Section VIII. Similarly to Fig. 1, in such experiments we deploy two LTE base stations and instantiate two RAN slices controlled by MVNOs M_1 and M_2, respectively. Each slice is assigned with 50% of the available RBs and serves a set of cellular users (i.e., commercial LTE smartphones). In Fig. 2, we report the network throughput of the network. Specifically, we compare measured throughput with (RBs are allocated to minimize inter-MVNO interference as in Fig. 1a) and without (Fig. 1b) slice isolation. It is easy to notice that slice isolation considerably improves network throughput and provides a throughput gain up to 3 Mbps. *In Section VIII, we show how our algorithms considerably improve network throughput with respect to slicing enforcement algorithms that do not enforce isolation across slices, such as the one in Fig. 1b.*

Although the problem of RAN slicing has attracted large interest [7]–[9], [11]–[15], [26]–[29], only few works have tackled the issue of physical-level allocation of spectrum resources to MVNOs [10]. 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) *Enabling of Advanced 5G Technologies:* 5G systems will heavily rely upon advanced signal processing and RF transmission technologies such as IBSPC, JC, CoMP and MIMO. This techniques considerably improve network performance, but require coordination among BSs in proximity. For this reason, the RB allocation should facilitate and foster such coordination;
2) *Isolation:* As demonstrated in Fig. 2, to increase efficiency, 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 [7], [30], [31];
3) *Contract Compliance:* MVNOs stipulate contracts with the IP to obtain control over the amount of resources specified by the slicing policy. In other words, the RSEP must guarantee that if an MVNO has been assigned 30% of spectrum resources and it is paying to get them, then it should also receive 30% of the total RBs available.

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

- We formulate the *RAN slicing enforcement problem* (RSEP), and show that it is NP-hard. Therefore, we propose approximation and heuristic solutions tailored for different network scales, optimality and timing requirements;
- We show via simulations that the computation time of the proposed algorithms can be as low as few hundreds of microseconds without considerably impacting the overall efficiency of the computed solutions. We also show that by enforcing slice orthogonality, the proposed approach reduces inter-MVNO interference, thus effectively doubling the overall signal-to-interference-plus-noise ratio (SINR) of the network if compared to traditional RAN slicing enforcement approaches;
- We demonstrate the effectiveness of the proposed algorithms through experimental evaluation on a LTE-compliant testbed composed of 2 LTE base stations and 6 commercial off-the-shelf (COTS) users. Results show that our approach outperforms other slicing techniques that do not enforce isolation across RAN slices. Specifically, our algorithms compute slicing strategies that lead to SINR and throughput improvements up to 27%.

The remainder of this paper is organized as follows. Section II surveys the literature on the topic. The considered RAN model is illustrated in Section III; Section IV introduces the RSEP problem, and Section V presents optimal, approximated and heuristic solutions to the RSEP. Section VI presents two effective complexity reduction techniques to further speed-up the computation of enforcement strategies. The performance of the proposed algorithms are assessed numerically and experimentally in Sections VII and VIII, respectively. Final remarks are given in Section IX, which concludes the article.

II. Related Work

The problem of determining how many resources each RAN slice should receive, also known as RAN slicing [7], [28], [32], has received significant interest from the research community over the last years; for excellent surveys on recent work on the topic the reader may refer to [26], [27]. Theoretical tools, ranging from optimization [11], [33]–[36], auctions [37], game theory [38]–[40] and artificial intelligence [13], [41] have been proposed. However, such work does not address how to

actually deploy RAN slices on top of the underlying physical network.

This key aspect has stimulated the research community to research the enforcement of RAN slicing policies. Prior work [7], [8], [32], [42], [43] virtualizes the available resources to create "pools" that are then shared and allocated among the MVNOs. This approach, however, may be ineffective in scenarios where *fine-grained control of physical-layer resources is required*, for example, to enable IBSPC, CoMP and beamforming.

Recent work has focused on addressing the RAN slicing enforcement problem from a resource allocation perspective. In [9], Mancuso *et al.* present a stochastic model to predict the impact of different enforcement policies on the overall performance of a sliced cell. Chang *et al.* [30] 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.* [44] consider genetic algorithms to assign the available RBs to the MVNOs such that a long-term utility is maximized. However, [9], [30], [44] 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 [31], [45] identify fine-grained RB management as a promising approach to guarantee orthogonality and reduce inter-MVNO interference, thus deploying highly-efficient 5G networks. However, [31] does not provide any algorithm to enforce slicing policies to maximize network efficiency, while [45] does not consider interference among BSs when allocating RBs.

In our prior work [10], we have proposed algorithms to (i) satisfy MVNOs requests; (ii) enforce orthogonality by reducing inter-MVNO interference across multiple BSs, and (iii) enable advanced coordination-based communication techniques. In this paper we ameliorate [10] by (i) presenting an improved heuristic algorithm that can compute a solution to the RSEP in few milliseconds while achieving a small optimality gap; (ii) investigating the impact of different enforcement policies on the interference of the network, showing that the proposed approach improves the SINR of the system by 2 times; and (iii) implementing our algorithms, and demonstrate their effectiveness, on a standard-compliant LTE testbed with 2 base stations and 6 cellular users. We show that not only our approach improves the overall throughput of the network by 27%, but it can be seamlessly integrated in standard 5G systems.

III. System Model and RAN Slicing Overview

We consider the RAN shown in Fig. 3, consisting of a set $\mathcal{B} = \{1, 2, \ldots, B\}$ of B base stations (BSs) associated with a *coverage area* ρ_b, $b \in \mathcal{B}$. Two BSs b and b' are *interfering* (or *adjacent*) if $\rho_b \cap \rho'_b \neq \emptyset$, *i.e.,* their coverage areas overlap. We define $\mathbf{Y} = (y_{b,b'})_{b,b' \in \mathcal{B}}$ as a symmetric adjacency matrix where $y_{b,b} = 0$ 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. An illustrative example of the adjacency matrix is shown in Fig. 3.

The RAN is administered by an IP, who periodically rents virtual RAN slices built on top of the underlying physical network $\mathcal{B}$ to a set $\mathcal{M} = \{1, 2, \ldots, M\}$ of M MVNOs.

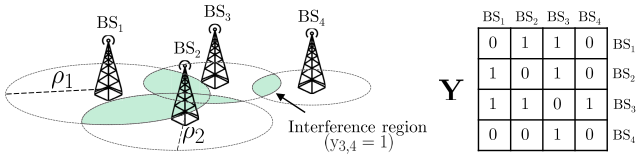


Fig. 3. An illustrative example with 4 BSs and their corresponding adjacency matrix $\mathbf{Y}$. Green areas show interference (or adjacency) regions where coverage areas overlap.

TABLE I
Summary of Notation

Variable	Description
$\mathcal{B}$	Set of base stations (BSs)
ρ_b	Coverage area of base station $b \in \mathcal{B}$
$\mathcal{M}$	Set of Mobile Virtual Network Operators (MVNOs)
$\mathbf{Y}$	Adjacency matrix
$y_{b,b'}$	Adjacency indicator
N_{RB}	Number of available subcarriers
T	Number of temporal slots within the slicing window
$\mathcal{R}$	Set of the available resources at each BS, $\lvert\mathcal{R}\rvert = N_{RB} \cdot T$
$\mathbf{L}$	RAN Slicing profile
$\mathcal{M}_b$	Set of MVNOs including BS b in their RAN slice
$x_{m,b,n,t}$	RB allocation indicator
$\boldsymbol{\pi}$	Slicing enforcement policy
Π	Set of all feasible slicing enforcement policies $\boldsymbol{\pi}$

Without loss of generality, we assume RAN slices are valid for T consecutive time slots [10]. We can utilize the T parameter to model different scenarios with different temporal scales. As an example, large T values can be used to model environments with slow-varying dynamics (*e.g.,* cellular networks in rural areas during nighttime), and small T values are used to model environments with fast-varying dynamics (*e.g.,* urban areas during daytime) requiring frequent update of RAN slice enforcement policies to adapt to mobility and channel dynamics.

In line with 5G NR and LTE standards, we assume that spectrum resources are represented as RBs, where each RB corresponds to the minimum scheduling unit [46]. We also consider an OFDMA channel access scheme where RBs are organized as in a time-frequency *resource grid* with N_{RB} subcarriers and T temporal slots. Thus, the set of available resources at each BS is $\mathcal{R}$, with $|\mathcal{R}| = N_{RB} \cdot T$, where each RB in $\mathcal{R}$ can be represented as a 2-tuple (n, t) with $n = 1, 2, \ldots, N_{RB}$ and $t = 1, 2, \ldots, T$. We assume that all BSs share the same resource grid structure, the case where this assumption is relaxed is considered in [47].

The interaction between MVNOs and the IP can be summarized as illustrated in Fig. 4. First, (i) MVNOs' *RAN slice requests* are collected by the IP. Then, (ii) the IP determines which requests should be admitted to the system, and 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 IP 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 $\boldsymbol{\pi}$ that allocates RBs among the MVNOs such that all admitted requests are satisfied (*i.e., RAN Slice Enforcement Problem*).

Problem (ii) has been already extensively investigated in the literature [7], [26]–[28], [32]–[35], [37]–[39]. For this reason, in this paper we instead address point (iii) by showing how the IP can compute an efficient slicing enforcement policy $\boldsymbol{\pi}$ that satisfies the three requirements described in Section I, *i.e.,*

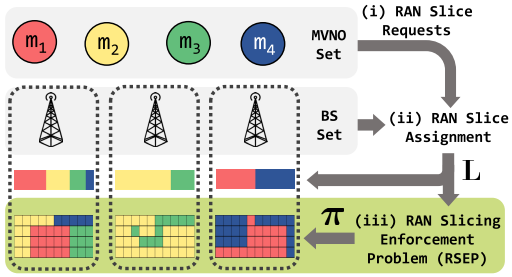


Fig. 4. The RAN slicing architecture.

enabling of advanced 5G technologies, isolation and contract compliance.

In this paper, We tackle the problem from an IP's point of view which has no access to mobile users' location, demanded traffic and channel conditions. Thus, we consider the case where MVNOs submit slice requests that reflect the current state of the network and, for privacy and business concerns, do not share the above information with the IP. Accordingly, MVNOs requests a target number of RBs that guarantees satisfaction of service-level agreements (SLAs) and meet desired Key Performance Indicator (KPI) metrics.

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 $\boldsymbol{\pi} = (\boldsymbol{\pi}_b)_{b\in\mathcal{B}}$ be the *slicing enforcement policy*, where $\boldsymbol{\pi}_b = (\boldsymbol{\pi}_{m,b})_{m\in\mathcal{M}}$ and $\boldsymbol{\pi}_{m,b}$ 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 $(n,t) \in \boldsymbol{\pi}_{m,b} \iff x_{m,b,n,t} = 1$. Hence, the set Π of all feasible slicing enforcement policies $\boldsymbol{\pi}$ can be defined as:

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

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 or more BSs. This is relevant because (i) linked RBs enable 5G advanced transmission schemes (*e.g.,* distributed beamforming, MIMO, CoMP transmissions and power control) among nearby BSs; (ii) as shown in Fig. 1, linked RBs can be used to deploy fully-orthogonal RAN slices that do not interfere with each other, and hence (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, for each MVNO m we define the number of linked RBs associated to interfering BSs b and b', i.e., $y_{b,b'} = 1$, as follows:

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

where 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 slicing profile $\mathbf{L}$ is

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

where the $1/2$ factor is introduced to avoid counting the same RBs twice, and $n_{b,b',m}$ is defined in (2).

With (2) and (3) at hand, we can formally define the RSEP as follows:

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

In a nutshell, the objective of Problem RSEP is to compute a feasible slicing enforcement policy $\boldsymbol{\pi}$ that maximizes the total number of linked RBs while guaranteeing that such policy does not violate the feasibility constraint $\boldsymbol{\pi} \in \Pi$. Moreover, 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 5G communications such as CoMP, JT and beamforming. In Sections VII and VIII, we will demonstrate how increasing the number of linked RBs of the system improves key performance metrics such as throughput and SINR.

V. Addressing the RSEP Problem

To solve Problem RSEP, we need to compute a slicing enforcement policy by exploring the feasible set Π searching for a solution that maximizes the number of linked RBs. However, the formulation in Problem RSEP does not in itself provide any intuitions on how a solution can be computed. For this reason, now we: (i) Reformulate Problem RSEP by using the RB allocation indicators introduced in Section IV; (ii) show that the resulting problem is NP-hard, and (iii) present a number of algorithms to address and solve Problem RSEP.

A. Optimal Solution

By using the definition of the RB allocation indicator $x_{m,b,n,t} \in \{0,1\}$ and from (1), (3) can be reformulated as

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

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

$$\underset{\boldsymbol{x}}{\text{maximize}} \ \frac{1}{2}\mathbf{x}^\top \mathbf{Q}\mathbf{x} \tag{RSEP-QP}$$

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

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

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

where $\mathbf{x} = (x_{m,b,n,t})_{m,b,n,t}$ is a $MBN_{RB}T \times 1$ 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. We point out that problems RSEP and RSEP-QP are equivalent. Indeed, the latter is a reformulation of the former in terms of the RB allocation indicator. However, we have been able to show that the RSEP can be modeled as a 0-1 (or binary) Quadratic Programming (QP) problem. Thus, in Theorem 1 we prove that Problem RSEP-QP—and thus Problem RSEP—is NP-Hard.

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

Proof: To prove the NP-hardness of Problem RSEP-QP, 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 [48], [49] 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 $\boldsymbol{\lambda}$ be the set of eigenvalues of $\mathbf{Q}$. Notice that $\sum_{\lambda_i \in \boldsymbol{\lambda}} \lambda_i = \mathrm{Tr}\{\mathbf{Q}\}$, and $\mathrm{Tr}\{\mathbf{Q}\} = 0$ 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. ∎

The indefiniteness of the $\mathbf{Q}$ matrix prevents the application of well-established results on quadratic functions where positive/negative definiteness guarantees the existence of a unique global solution a priori.

To find an optimal solution to Problem RSEP-QP it is possible to apply spatial Branch and Bound (sB&B) techniques for non-linear non-convex problems. These algorithms have been shown to globally solve these class of problems by iteratively generating convex relaxations whose accuracy is refined at each iterations [50]. Although sB&B makes it possible to globally solve Problem RSEP-QP, its complexity is still too high to be effectively employed in real-world 5G network deployments where the number of base stations and users is extremely large.

The objective of the following two subsections is to address the above issues and design algorithms that can compute effective RAN slice enforcing policies with low computational complexity.

B. Approximated Solution

Let $V = M \cdot B \cdot N_{RB} \cdot T$, and let us consider the following transformed version of Problem RSEP-QP

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

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

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

where $\lambda \in \mathbb{R}$ is a real-valued parameter whose relevance to Problem RSEP-EQ will be explained in Theorem 2, and $\boldsymbol{e}^\top = (1, 1, \ldots, 1)$. 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: Intuitively, the utility function in Problem RSEP-EQ introduces the term $\lambda\mathbf{x}^\top(\boldsymbol{e} - \mathbf{x})$ which generates a cost, or a penalty, proportional to λ 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 $0 \leq x_{m,b,n,t} \leq 1$. 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 [51, Th. 3.1], it must exist $\lambda_0 \in \mathbb{R}$ such that Problems RSEP-EQ and RSEP-QP are equivalent for all $\lambda \geq \lambda_0$. In Theorem 1 we have proved that the $\mathbf{Q}$ matrix admits both negative and positive eigenvalues. Accordingly, let $\mathbf{z}$ be the set of eigenvalues of $\mathbf{Q}$ and $z^* = \max\{z_1, z_2, \ldots, z_{|\mathbf{z}|}\}$. It is possible to show that if $\lambda \geq z^*$, then the matrix $\mathbf{Q} + 2\lambda\mathbf{I}_V$ is positive semi-definite. Thus, Problem RSEP-EQ is convex if $\lambda \geq z^*$, which proves the theorem. ∎

Remarks: Theorem 2 shows that we can relax the binary constraint of Problem RSEP-QP with a penalty term λ. When λ is large enough, RSEP-EQ and RSEP-QP are equivalent and produce the same solutions. Otherwise, equivalence does not hold and solutions computed by RSEP-EQ might substantially deviate from the optimal ones computed by RSEP-QP.

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 [52]. Since the vertex space is considerably smaller than the complete feasibility set Π considered in Problem RSEP-QP, Problem RSEP-EQ is easier to solve when compared to Problem RSEP-QP. Specifically, approaches such as cutting plane and extreme point ranking methods [52] can be used to efficiently solve Problem RSEP-EQ.

C. Heuristic Solution

Although Problem RSEP-EQ has lower complexity than Problem RSEP-QP, in the worst case it still requires exponential time with respect to the number of vertices, which spurred us to design polynomial-time algorithms.

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

Algorithm 1 RSEP-MLF

1: `Input` $\mathcal{B}; \mathcal{M}; \mathbf{Y}; \mathbf{L}$;
2: `Output` A MLF RBs allocation $\mathbf{x}^G = (x^G_{m,b,n,t})_{m,b,n,t}$;
3: `Set` $x^G_{m,b,n,t} = 0$ 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 $x^G_{m,b,n,t}$ 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**

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

$$l_m = \sum_{b \in \mathcal{B}} \sum_{b' \in \mathcal{B} \setminus \{b\}} \min\{L_{m,b}, L_{m,b'}\} y_{b,b'} \tag{5}$$

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 $\mathcal{M}^G = \mathcal{M}$ in ascending order of l_m. Specifically, for any $m, k \in \mathcal{M}^G$, $m < k$ if $l_m \geq l_k$;
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 $C = \max\{MB^2, M \log M, N_{RB} \cdot B \cdot M\}$.

D. Improved RSEP-MLF

The greedy algorithm RSEP-MLF enjoys fast convergence time at the price of sub-optimal performance [10]. Thus, we design a novel heuristic that reduces the gap between RSEP-QP and RSEP-MLF while keeping the computational complexity as low as possible. To this end, we present RSEP-IMLF, which improves upon RSEP-MLF by iteratively adjusting the RB allocation strategy to increase the number of linked RBs.

First, note that the RSEP is shift-invariant with respect to the indexing of the RB (n) and temporal slot (t). This is because, for any given 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 equivalent to $\mathbf{x}^*$ as it produces the same number of linked RBs as $\mathbf{x}^*$. In general, we can extend this result to any reshape procedure that maintains the cardinality of $\mathcal{R}$ equal to $N_{RB} \cdot T$. We show this in Fig. 5, where the original RB grid (left) contains $N_{RB} \cdot T = 12$ RBs and has

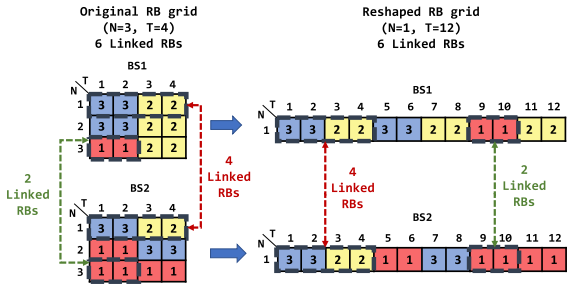


Fig. 5. An illustrative example of a RB grid reshaping with $B = 2$ BSs, $M = 3$ MVNOs and 6 linked RBs. The original RB grid is shown on the left, while the reshaped grid is shown on the right.

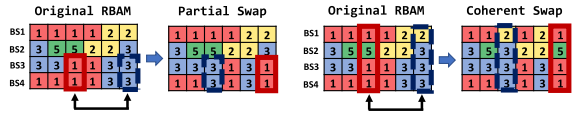


Fig. 6. An illustrative example of a RB allocation matrix (RBAM) and swapping procedures with $B = 4$ BSs and $M = 5$ MVNOs.

6 linked RBs. Fig. 5 shows that by reshaping the resource grid into a row vector does not change the amount of linked RBs.

Another important notion is the concept of *RB allocation matrix* (RBAM). Let us consider the reshaped RB grid $\mathcal{R} \in \mathbb{R}^{N_{RB}T \times 1}$, the RBAM is represented by the matrix $\boldsymbol{\sigma} = (\boldsymbol{\sigma}_b)_{b \in \mathcal{B}}$ where $\boldsymbol{\sigma}_b(\mathbf{x}^*) = (\sigma_{b,\tau})_{\tau \in \mathcal{R}} : \mathcal{X} \rightarrow \mathcal{R}$. Henceforth, b and τ will represent rows and columns of $\boldsymbol{\sigma}$, respectively. For any slicing enforcement solution $\mathbf{x} \in \mathcal{X}$, the RBAM builds 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 τ has been assigned to, *i.e.,* the MVNO m such that $x_{m,b,\tau} = 1$. Accordingly, we set $\sigma_{b,\tau} = M_{b,\tau}(\mathbf{x}^*)$. An example of a possible RBAM with $B = 4$ BSs and $M = 5$ MVNOs is shown in Fig. 6.

We are now ready to introduce the concept of *swapping*. Specifically, we say that two columns τ_1 and τ_2 of the RBAM are *coherently swapped* when all their corresponding entries σ_{b,τ_1} are replaced with those of σ_{b,τ_2} and *vice versa*. On the contrary, two columns are *partially swapped* when only a portion of entries is replaced among two columns. An example of a coherent swap is shown in the right side of Fig. 6, where the third and sixth columns are swapped. Instead, the left side shows a partial swap where only the two bottom elements of the columns of the RBAM are swapped.

By leveraging the concepts of reshaping, RBAM and swapping, we can finally develop an improved version of RSEP-MLF (i.e., RSEP-IMLF) which works as follows:

1) Compute a slicing enforcement solution $\mathbf{x} \in \mathcal{X}$ via RSEP-MLF and derive the corresponding RBAM $\boldsymbol{\sigma}$. Define $\mathcal{B}^* = \mathcal{B}$;
2) Select the row $b_0 \in \mathcal{B}^*$ in $\boldsymbol{\sigma}$ with the smallest number of distinct MVNOs and remove it from $\mathcal{B}^*$, *i.e.,* $\mathcal{B}^* = \mathcal{B}^* \setminus \{b_0\}$;
3) Pick the row $b^* \in \mathcal{B}^*$ that shares the highest number of linked RBs with b_0;
4) Select at random two columns τ_1 and τ_2. Perform a single-row partial swap on the RBAM $\boldsymbol{\sigma}$ by swapping the two elements (b_0, τ_1) and (b_0, τ_2). If the partial swap has improved the number of linked RBs, we update the RBAM accordingly. This step is repeated at most I_S times, where $I_S > 0$ is a parameter specifying the

Algorithm 2 RSEP-IMLF

```
Input B; M; Y; L;
Output A RBs allocation x^G = (x^G_{m,b,n,t})_{m,b,n,t};
Set B* ← B;
x* ← A MLF RB allocation computed through Algorithm 1;
σ ← The RBAM for x*;
b_0 ← The row of σ with the smallest number of distinct MVNOs;
while |B*| ≠ 0 do
    b* ← The row that shares the highest number of linked RBs with b_0;
    while i ≤ I_S do
        (τ_1, τ_2) ← Selects two columns at random;
        σ* ← A copy of σ with elements (b_0, τ_1) and (b_0, τ_2) swapped;
        if number of linked RBs has improved then
            σ ← σ*;
        end if
        i ← i + 1;
    end while
    B* ← B* \ {b_0};
end while
```

maximum number of trials RSEP-IMLF tries to improve upon the current slicing enforcement strategy;

5) If $\mathcal{B}^* = \emptyset$, we stop. Otherwise we re-execute Step 2).

The rationale behind RSEP-IMLF is to compute a sub-optimal solution fast, and then iteratively try to increment the total number of linked RBs by testing a limited amount of swapping combinations. As discussed in Section V-C, the complexity of Step 1 is $\mathcal{O}(C)$, where $C = \max\{MB^2, M\log M, N_{RB}\cdot B\cdot M\}$. Step 2 is executed once, and its complexity is $\mathcal{O}(B)$, while the complexity of Step 3 is $\mathcal{O}(BI_S)$. Accordingly, the overall complexity of RSEP-IMLF is $\mathcal{O}(C+B+BI_S) = \mathcal{O}(C+BI_S)$, meaning that RSEP-IMLF contributes to the overall complexity of RSEP-MLF with an additional linear complexity term.

E. Fairness Aspects

As shown in (RSEP-QP), we aim at maximizing the number of linked RBs of the system without considering how these RBs are distributed across the different slices. On the one hand, this makes it possible to assign RBs in a way that reduces interference and maximizes inter-slice isolation. On the other hand, different MVNOs can be assigned with different number of linked RBs, thus resulting in unfair linked RBs distribution. Although this problem is out of the scope of this paper, we believe that the problem is extremely challenging and it is worth of investigation. For this reason, here we discuss two different approaches that might effectively solve the above problem and generate more fair enforcement policies. The simplest approach would be to introduce a new constraint guaranteeing that a minimum number N_{min} of linked RBs is allocated to each MVNO (*i.e.*, $N_m \geq N_{min}$). From (4), this approach would result in a quadratic constraint that might make the problem unfeasible if N_{min} is too large. Another approach, would be to adapt the objective function of the RSEP problem via a α-fairness utility. If compared to the previous approach, this formulation would avoid unfeasibility of the problem, but would introduce a strong non-linearity in the objective function that would eventually result in higher computational complexity.

VI. Speeding-Up 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.

A. Sparsity

Let $\mathbf{x}^{\mathrm{OPT}}$ 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 $x^{\mathrm{OPT}}_{m,b,n,t} = 0$ 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 [53]. Thus, we reduce the complexity of the two problems by inducing sparsity through two transformations. 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 $\tilde{\mathbf{Q}}$. To keep equivalence between RSEP-QP and RSEP-EQ, the objective function of RSEP-EQ is rewritten as

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

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 $q_{m',b',n,t}$.

B. RB Aggregation

Let $K = \mathrm{GCD}(\mathbf{L})$ 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 resources can be aggregated in groups of K RBs, and each of such groups can be seen as a single aggregated RB. We refer to such a property as *aggregability* of the RSEP, whose definition is as follows.

Definition 2 (Aggregable RSEP): The RSEP is said to be *aggregable* if $N_{RB} \pmod K = 0$ or $T \pmod K = 0$, where $K = \mathrm{GCD}(\mathbf{L}) > 1$ 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. From Section V-D, recall that problem P is shift invariant with respect to the indexing of n and t. 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 $\tilde{P}$ in the RBAM space, respectively. Also, let $f_Z(\mathbf{x}) : \mathcal{X} \to \mathbb{N}$ and $f_{Z/K}(\mathbf{x}) : \tilde{\mathcal{X}} \to \mathbb{N}$ 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 $\tilde{\mathbf{x}}^*_A \in \mathcal{X}$ be the solution to P generated by expanding the aggregated optimal solution $\tilde{\mathbf{x}}^*$ to $\tilde{P}$. Let $\mathcal{R} \in \mathbb{R}^{N_{RB}\tilde{T}\times K}$, the expanded solution $\tilde{\mathbf{x}}^*_A = (\tilde{x}^*_{A_{m,b,\tau,k}})_{m,b,\tau,k}$ 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 $g(\mathbf{x}) : \mathcal{X} \to \mathbb{N}$ be defined as $g(\mathbf{x}) = K^{-1}f(\mathbf{x})$. Intuitively, if we replace the objective function $f(\mathbf{x})$ of P with $g(\mathbf{x})$, we obtain the same problem where each linked RB gives a reward equal to K^{-1} ($f(\mathbf{x})$ 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

$$\begin{aligned} g(\mathbf{x}^*) &= K^{-1}f_Z(\mathbf{x}^*) \\ &> K^{-1}f_Z(\tilde{\mathbf{x}}^*_A)g(\tilde{\mathbf{x}}^*_A) = f_{Z/K}(\tilde{\mathbf{x}}^*) \end{aligned} \tag{7}$$

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

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 exists 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 $h(\mathbf{x}) : \mathcal{X} \to \tilde{\mathcal{X}}$ such that $h(\mathbf{x}) = \hat{\mathbf{x}} \in \tilde{\mathcal{X}}$ that can be transformed into $\hat{\mathbf{x}}_A$ such that $f_{Z/K}(\hat{\mathbf{x}}) = K^{-1}f_Z(\hat{\mathbf{x}}_A)$.

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 $h(\mathbf{x}) = \tilde{\mathbf{x}}$ algorithm 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. ■

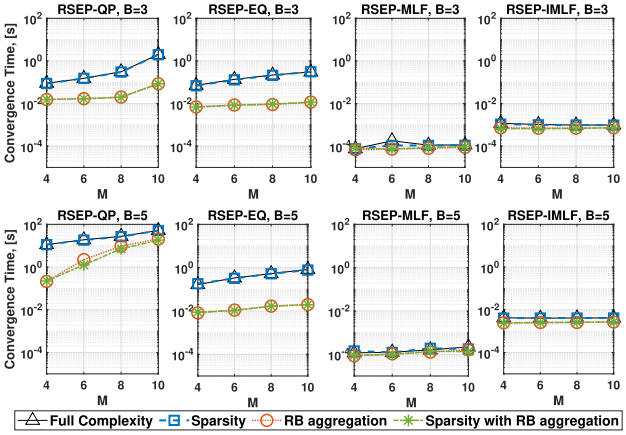


Fig. 7. Convergence time (s) of the three proposed solutions as a function of M considering different computational time reduction techniques.

VII. 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 6 Physical Resource Blocks (PRBs). Each PRB consists of 12 subcarriers and 7 symbols. Time is divided into discrete time slots called *sub-frames*. Each sub-frame is formed of two PRBs, lasts 1 ms, and is the minimum scheduling unit in LTE. Groups of $N_{SF} = 10$ sub-frames constitute a *frame*.

In our analysis, each RB corresponds to one sub-frame, therefore we consider a total of $N_{RB} = 6$ RBs per time slot. 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 $\mathbf{L} = (L_{m,b})_{m\in\mathcal{M},b\in\mathcal{B}}$ defined in Section III are generated at random at each simulation run.

In order to evaluate the benefits of the proposed approach, in the following of this section we compare our algorithms with slice-unaware schemes that do not leverage information on network topology and interference to instantiate RAN slices. We refer to this method as the *w/o isolation* case where RBs are assigned to requesting MVNOs in a round-robin fashion with complexity $\mathcal{O}(1)$.

The simulator is implemented in MATLAB and is interfaced with IBM CPLEX optimization toolbox. Specifically, CPLEX is used to solve RSEP-QP and RSEP-EQ, while the two heuristics RSEP-MLF and RSEP-IMLF have been implemented in MATLAB only. Results were averaged over 1000 independent simulation runs.

A. Convergence Time Analysis

Fig. 7 shows the convergence time of the four algorithms presented in Section V as a function of the number M of MVNOs when $N_F = 2$. As expected, the algorithm with the slowest convergence time is the optimal algorithm RSEP-QP, while the fastest algorithm is RSEP-MLF. Interestingly enough the convergence time of both RSEP-QP and RSEP-EQ increases as the number of MVNOs in the network grows, a behavior that is not exhibited by the two heuristic algorithms RSEP-MLF and RSEP-IMLF whose convergence time only slightly increases as a function of M.

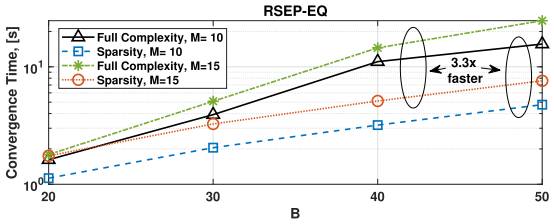


Fig. 8. Convergence time (s) of RSEP-EQ as a function of B considering different number M of MVNOs.

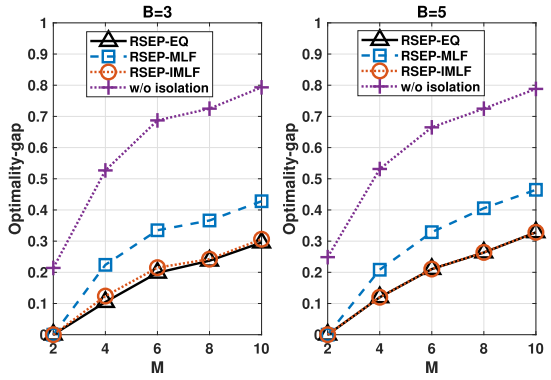


Fig. 9. Optimality-gap of RSEP-EQ, RSEP-MLF and RSEP-IMLF as a function of M considering different number B of BSs.

Fig. 7 also shows the impact of the sparsity and RB aggregation mechanisms in Sections VI-A and VI-B on the overall convergence time. It can be observed that the techniques presented in Section VI can effectively reduce the computation time of all the four algorithms. Moreover, we show that RB aggregation is the technique that 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* $B = 5$ *and RSEP-EQ only requires* $1s$*. On the contrary, RSEP-IMLF computes a solution within few milliseconds, while RSEP-MLF computes the solution in less than a millisecond.*

It is worth to point out that Fig. 7 reveals how the reduction in terms of convergence time brought by sparsity cannot 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. 8. Our results show that sparsity can effectively reduce the computation time by several tens of seconds, and the gain increases as both M and B increase. From Fig. 8 we can conclude that sparsity is a complexity reduction technique that best performs in large scale network deployments.

B. Optimality-Gap Analysis

Another crucial aspect is the optimality-gap between the optimal solution computed by RSEP-QP and those computed through RSEP-EQ/RSEP-MLF algorithms. Although Theorem 2 shows that (under some conditions) Problem RSEP-EQ is equivalent to Problem RSEP-QP, we cannot 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.

For this reason, in Fig. 9 we investigate the Optimality-gap of RSEP-EQ, RSEP-MLF and RSEP-IMLF with respect to an optimal solution computed by RSEP-QP. This performance metric is defined as one minus the ratio between the utility function achieved by any of the aforementioned approximation and heuristic algorithms and that achieved by RSEP-QP. The closer to zero is the gap, the closer to optimality is the solution computed by approximation and heuristic algorithms.

Fig. 9 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 as well. Notice that although RSEP-MLF is negligibly affected by the number of BSs B, it achieves poor performance if compared to RSEP-EQ. It is worth to mention that RSEP-IMLF is perhaps the most efficient algorithm which effectively trades-off between optimality and computational complexity. Indeed, Figs. 9 and 7 show that RSEP-IMLF can compute RAN slice enforcement strategies that achieve the same performance as RSEP-EQ in the order of few milliseconds. Furthermore, the w/o isolation case employs a round-robin scheme that, although being fast, results in very high gap.

C. Linked RBs and SINR Analysis

Fig. 10 shows the impact of M on the percentage of linked RBs of the system when $N_F = 10$, $B = 5$ and $T = 100$. As expected, RSEP-EQ and RSEP-IMLF always perform better than RSEP-MLF in terms of number of linked RBs. Moreover, Fig. 10 illustrates that the number of linked RBs decreases as the number M of MVNOs increases. 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 jointly with a large number of linked RBs.

As demonstrated in Fig. 10, and if compared to the traditional approach where inter-slice isolation is not enforced, our approach increases the percentage of RBs that can be used to perform coordination-based transmissions. A major question, however, is whether or not the enforcement strategies presented in this paper can actually bring performance gains in terms of throughput and interference mitigation when applied to real-world 5G networks. To answer such an interesting question, at each simulation run we have generated a random network topology with B BSs and M independent user sets each consisting of 10 cellular users randomly deployed within the simulated area. In other words, we assume that each MVNO requests a RAN slice to serve 10 cellular users. Channel gain coefficients between BSs and cellular users are computed through the well-established free-space path loss model.

Recall that the slicing profile $\mathbf{L}$ is randomly generated at each simulation run. For each $\mathbf{L}$ we allocate RBs to MVNOs by running different RAN slicing enforcement algorithms. Then, for each MVNO we compute the optimal downlink transmission policy that maximizes the rate of the system [54] by determining (i) which user should be scheduled in each RB, (ii) how much power to allocate to each transmission, and (iii) whether or not a user should be served by multiple neighbouring BSs through CoMP transmissions.

Our results are reported in Fig. 11, where we show the average SINR for different RAN slicing enforcement algorithms

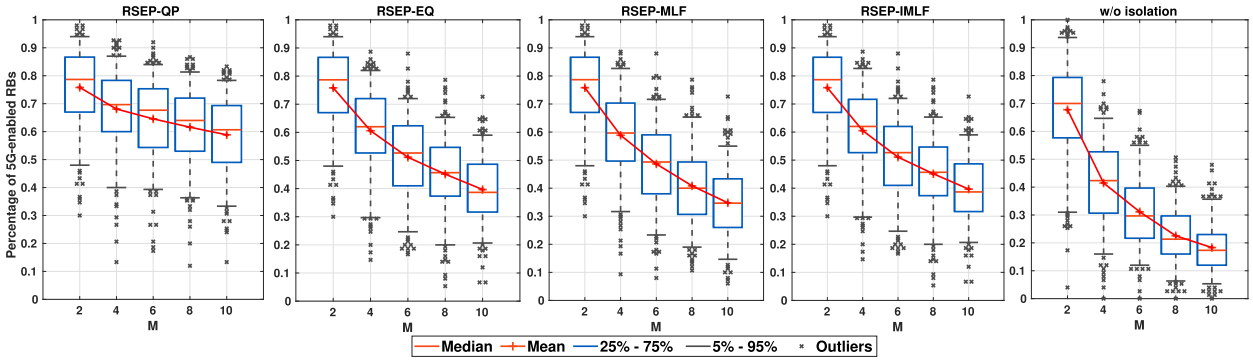


Fig. 10. Percentage of linked RBs as a function of M and different RAN slicing enforcement policies ($B = 5$).

as a function of the number of MVNOs and BSs. In general, Fig. 11 shows that our approach always improves the SINR of cellular users by providing gains up to 2 times and an effective SINR gain up to 10 dB. Despite the optimality ratio decreases when large values of B and M are considered, Fig. 11 shows that cellular users still experience higher SINR values if compared to traditional RAN slicing algorithms where isolation across slices is not enforced. This results show the effectiveness of our approach even in the case of sub-optimal RAN slice enforcement policies.

D. Bandwidth and Time-Scale Analysis

We now investigate the impact of different bandwidth configurations and time-scale requirements. Specifically, we consider the case of a resource grid with a 20 MHz bandwidth, $N_{RB} = 100$ RBs and $T = 5$s. In this configuration, we let $M = 5$ MVNOs change the slicing profile $\mathbf{L}$ at a slower frequency if compared to the case considered in previous sections. Since we have already demonstrated that complexity reduction techniques presented in Section VI effectively reduce computation times, we will present results obtained by applying both sparsity and RB aggregation techniques to our approach. Also, given that the considered scenario presents a very high number of RBs, we investigate the impact of slice requests profiles on the complexity of the problem. Specifically, we consider 5 different cases where MVNOs are allowed to submit requests where the number of requested RBs must be a multiple of $\xi \in \{2, 5, 10, 20, 50\}$, which is equivalent to submitting requests with an instantaneous minimum bandwidth equal to $0.4, 1, 2, 4, 10$ MHz for each subframe. We refer to ξ as the minimum RB request block size. Indeed, the minimum request requirement does not hold if the MVNO is not willing to request any RB on a particular BS.

In Fig. 12, we show the convergence time and optimality-gap of our algorithms for different values of ξ. As expected, large values of ξ facilitate RB aggregation (see Section VI-B) and considerably reduce the computation time of all solutions, especially for RSEP-QP which experiences a reduction in the computation time by a factor $1000\times$ when moving from $\xi = 2$ to $\xi = 50$. Similar considerations apply to the optimality-gap as well which decreases as the value of ξ decreases as well. These results show that aggregation of RBs not only reduces the complexity of the problem, but it also results in more efficient solutions.

VIII. Experimental Evaluation

The objective of this section is to experimentally demonstrate that the benefits of our approach are not restricted to

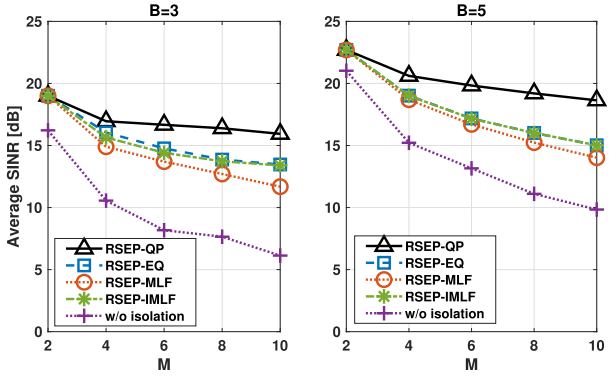


Fig. 11. Average SINR achieved by the proposed algorithms as a function of M considering different number B of BSs.

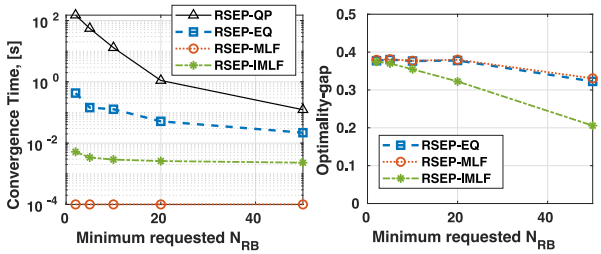


Fig. 12. Convergence time (s) and optimality-gap of our algorithms as a function of the minimum RB request block size ξ.

simulation scenarios only, but they also apply to real cellular network deployments. For this reason, in Sections VIII-A and VIII-B we first describe the testbed and the scenario considered in our experiments. Then, we discuss the obtained results in Section VIII-C.

A. Experimental Setup

To demonstrate the superior performance of our algorithms, we have instantiated a standard-compliant LTE cellular network on the Arena testbed [55] located in the main campus of Northeastern University, Boston, MA, USA. Arena is an experimental software-defined radio (SDR) testbed whose goal is to facilitate prototyping and performance evaluation of algorithms and protocols for sub-6GHz applications in a real-world wireless environment. Arena consists of Ettus Research USRPs N210 and X310 SDRs whose antennas are connected via 100 feet long SMA cables and are hanging from the ceiling of a 2240 square feet office space. Antennas have a 3 dBi gain while USRPs have a 30 dB maximum transmission

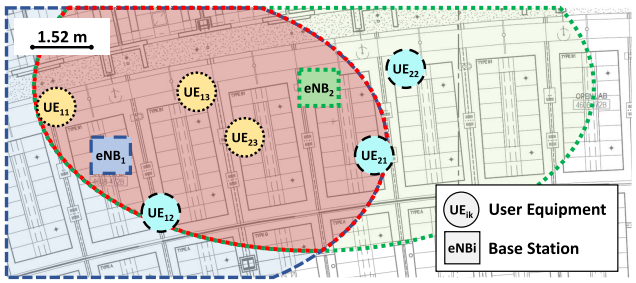


Fig. 13. Experimental setup.

gain and are controlled via software by GPU-enabled high-computational power Dell EMC PowerEdge R340 servers.

To deploy a standard-compliant LTE network, we leveraged the srsLTE [56] open-source software which offer LTE-compliant base station (eNB) and UE protocol stack implementations, as well as an Evolved Packet Core application. In this section we discuss implementation details and report results obtained through our srsLTE-based prototype. However, we would like to remark that our solutions can be seamlessly ported with minimal changes to other open-source software platforms such as OpenAirInterface (OAI) [57].

We deployed two standard-compliant eNBs on Arena USRPs X310 serving 6 COTS UEs (Xiaomi Redmi Go). The deployed LTE network is shown in Fig. 13, where UE_{xy} is served by eNB_x, with $x \in \{1, 2\}, y \in \{1, 2, 3\}$.

We deploy the LTE network in Frequency Division Duplex (FDD) mode in the LTE Band 7 with a bandwidth equal to 10MHz and 50 RBs. As shown in Fig. 13, we consider 2 MVNOs serving UE_{11}, UE_{13}, UE_{23} (slice 1) and UE_{12}, UE_{21}, UE_{22} (slice 2), respectively. The two eNBs are deployed approximately 6 meters apart and their coverage areas partially overlap. For example, UE_{13} and UE_{23}—which are both associated to slice 1—experience severe interference due to the proximity to adjacent eNBs.

This setup is particularly well-suited to showcase the performance gains brought by interference reduction of our approach.

B. Integrating RAN Slicing in srsLTE

Although srsLTE provides functionalities to deploy an LTE cellular network with few lines of code only, it does not directly provide RAN slicing functionalities required by our algorithms. For this reason, we have extended srsLTE functionalities by integrating RAN slicing mechanisms at each eNB. An overview of the extended framework is shown in Fig. 14. Blocks highlighted in green indicate the extended software components to integrate RAN slicing functionalities within the srsLTE framework.

In each experiment, the IP receives RAN slice requests generated by a set of MVNOs. Each RAN slice request specifies which eNBs should be included in the slice, and the number of RBs that should be assigned to the slice on each eNB. Upon reception of these requests, the IP executes one of the RAN slicing enforcement algorithms proposed in this article to assign the available RBs to the requesting MVNOs such that the number of linked RBs is maximized. The solution of the algorithm is then converted into a set of B configuration files (*i.e.*, the `config.txt` files in Fig. 14). Each configuration file is associated to individual eNBs and specifies which RBs should be assigned to each slice. These files are then

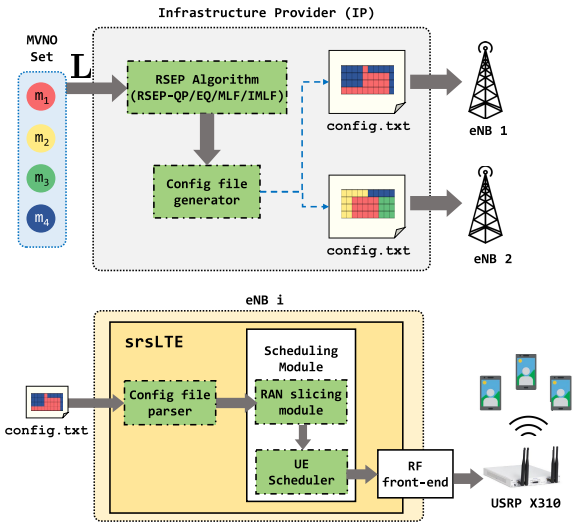


Fig. 14. Prototype overview and integration with srsLTE. Green blocks indicate the extended software components.

dispatched to the corresponding eNB and processed by a *config file parser module*. RAN slicing enforcement information is then fed to a *RAN slicing module* that instantiates RAN slices and exclusively assigns RBs to each of them according to the configuration specified in the `config.txt` file. Finally, RBs assigned to each RAN slice are used by individual MVNOs to schedule UE downlink transmissions performed by Arena USRPs. This is achieved by assigning each UE to one slice only. This association is implemented by assigning a unique identifier (i.e., an integer number) to each slice and associating the unique international mobile subscriber identity (IMSI) of each UE to a slice identifier. This way, the *UE scheduler* module can schedule UEs belonging to a specific slice on RBs that have been assigned to that slice only.

C. Experimental Results

We consider the case where two MVNOs lease eNB resources (*i.e.*, RBs) to instantiate RAN slices. Our experiments aim at evaluating two critical performance parameters, *i.e.*, network throughput and SINR experienced by UEs. To showcase the effectiveness of our algorithms, we compared the optimal RSEP-QP method presented in Section V-A, with the traditional one (*i.e.*, *w/o isolation*) in which RAN slices are instantiated without leveraging network topology information and without enforcing slice isolation.

We ran 10 experiments on the testbed presented in Section VIII-A. At each experiment run we generate a random slicing profile $\mathbf{L}$ in MATLAB. To compensate for inaccurate synchronization of our experimental equipment, slicing profile $\mathbf{L}$ are generated with $K = 9$, which we have experienced to be a large enough value to ensure that most RBs are synchronized across adjacent BSs.

Mobile users perform a 2-minute long speed-test (which ensures that transmission buffers are constantly backlogged with downlink packets and slices are always active) and report both throughput and SINR measurements.

To provide a fair comparison between different approaches, for each $\mathbf{L}$ we compute RAN slicing enforcement policies by using the method presented in this paper (*i.e.*, RSEP-QP) and traditional ones (*i.e.*, *w/o isolation*). Also, to avoid

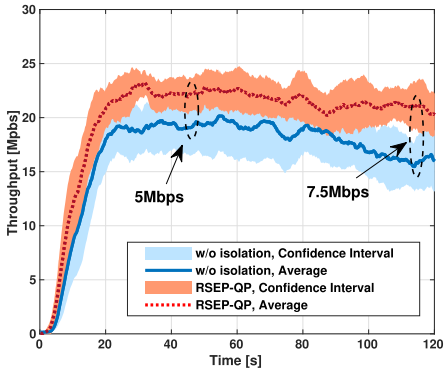
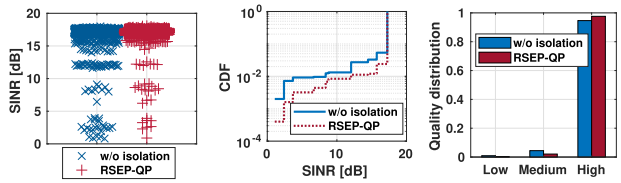


Fig. 15. Experimental throughput comparison.



(a) SINR measurements. (b) CDF. (c) SINR Quality.

Fig. 16. Experimental SINR analysis.

time-varying performance degradation introduced by Internet connectivity, which would result in an unfair comparison between the two methods, the speed-test server is locally hosted on the Arena testbed.

The average network throughput over the 10 experiments is reported in Fig. 15. Our results clearly show that our approach (*i.e.*, RSEP-QP) outperforms traditional interference-agnostic approaches and increases throughput by approximately 27% (approximately 5Mbps gain) with peak throughput gains up to 7.5Mbps.

In Fig. 16 we analyze SINR measurements reported by UEs under the two considered methods (Fig. 16a). The Cumulative Distribution Function (CDF) of the SINR shown in Fig. 16b clearly demonstrates that traditional approaches are subject to poor SINR performance due to high interference across heterogeneous RAN slices. Our approach, instead, effectively reduces such interference and improves the SINR experienced by UEs. This can be easily noticed in Fig. 16c where we show the ratio of users reporting Low ($\text{SINR} \leq 5\text{dB}$), Medium ($5\text{dB} < \text{SINR} \leq 17\text{dB}$) and High ($\text{SINR} > 17\text{dB}$) SINR. Fig. 16c shows that our approach results in a larger portion of UEs reporting higher SINR if compared to traditional approach which, instead, shows higher percentage of UEs reporting low and medium SINR levels.

IX. Conclusion

In this article, 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 three approximation and heuristic algorithms that render the problem tractable and scalable as the problem increases in complexity. Finally, we have evaluated the algorithms through simulations, and demonstrated their effectiveness through experimental analysis on a testbed composed by 2 LTE base stations and 6 cellular users. Results conclude that our algorithms are scalable and provide near-optimal performance. Moreover, our solutions

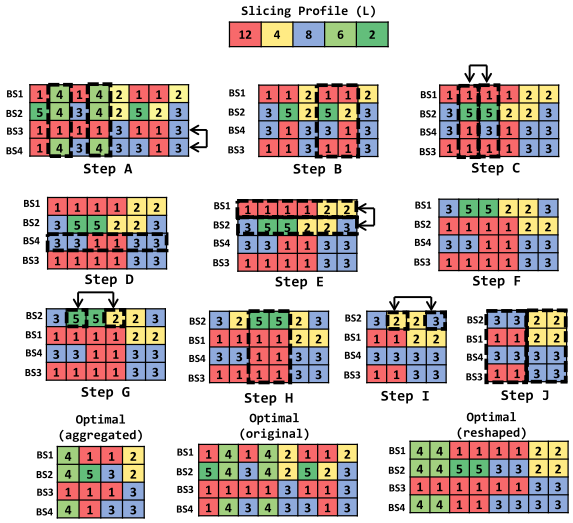


Fig. 17. An example of the reshaping algorithm in Appendix A.

effectively enforce RAN slicing policies by satisfying MVNOs requirements, reducing inter-MVNO interference, and providing throughput and SINR gains up to 27% and 100%, respectively.

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 consider the RB allocation matrix (RBAM) $\boldsymbol{\sigma} = (\boldsymbol{\sigma}_b)_{b \in \mathcal{B}}$ where $\boldsymbol{\sigma}_b(\mathbf{x}^*) = (\sigma_{b,\tau})_{\tau \in \mathcal{R}} : \mathcal{X} \rightarrow \mathcal{R}$. Similarly to what done in Section V-D, b and τ will represent rows and columns of $\boldsymbol{\sigma}$, respectively. Let $M_{b,\tau}(\mathbf{x}^*)$ be the MVNO that RB τ has been assigned to, *i.e.,* the MVNO m such that $x_{m,b,\tau} = 1$. Accordingly, we set $\sigma_{b,\tau} = M_{b,\tau}(\mathbf{x}^*)$.

Fig. 17 depicts an example that will help us explain how we can map any optimal solution $\mathbf{x}^* \in \mathcal{X}$ to the RSEP to an aggregated solution $\tilde{\mathbf{x}}^* \in \tilde{\mathcal{X}}$. Let us represent in Step A the RBAM corresponding to the optimal solution of the RSEP when $B = 4$ interfering BSs are deployed and $M = 5$ tenants request a different amount of RBs on each BS. Furthermore, we consider $N_{RB} = 2$ and $T = 4$. We consider the slicing profile $\mathbf{L}$ in Fig. 17 where GCD is $K = 2$. Recall that two columns τ_1 and τ_2 are said to be *coherently swapped* when all their corresponding entries σ_{b,τ_1} are replaced with those of σ_{b,τ_2} and *vice versa*. Similarly, two columns are *partially swapped* when only a portion 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 $\sigma_{b,\tau_1}, \ldots, \sigma_{b,\tau_K}$ are *paired* if $M_{b,\tau_1}(\mathbf{x}) = M_{b,\tau_2}(\mathbf{x}) = \cdots = M_{b,\tau_M}(\mathbf{x})$, they are said to be *unpaired* otherwise.

1 Our mapping algorithm works as follows. First, if any K columns of $\boldsymbol{\sigma}$ are identical (see Step A, where columns 2 and 4 are identical), we remove them from $\boldsymbol{\sigma}$ (see Step B) and add them to the aggregated RBAM (see the first column in the bottom-left RAMB). Then, we take the following steps:

1) We select the row b_0 in $\boldsymbol{\sigma}$ with the smallest number of distinct MVNOs and we move it to the lowest row (see Steps A and B where we swap rows 3 and 4);
2) We update $\boldsymbol{\sigma}$ by ordering row b_0 in MVNO identifier order (see Steps B and C, where to order row b_0, we coherently swap column 3 with 5, and then 4 with 5).

This operation (i) creates ordered groups of K entries (see Step C); and (ii) preserves the optimality of the solution as all columns are coherently swapped;

3) If all entries in the RBAM have been paired, we stop the algorithm;
4) If any K columns of $\boldsymbol{\sigma}$ are identical, we remove them from $\boldsymbol{\sigma}$ (as done in Steps H and J) and we include them to the aggregated RBAM (see bottom-left RBAM);
5) We select the row b_n (among the rows above b_0) that shares the highest number of links with b_0 (row 3 in Step D and row 1 in Step E), and we move it above b_0 (row 3 in Step D is already above b_0, while in Step E we need to swap rows 1 and 2);
6) If all the entries in b_n are paired, we go to 7) (such as in Steps D and F); otherwise, we find K unpaired entries and we generate a partial swap of b_n and the upper rows (Steps G and I) such that i) the number of links remains the same[1]; 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. Accordingly, the solution generated by the partial swap and the initial optimal solution are equivalent and share the same number of links.*;
7) We set $b_0 = b_n$ and go to 3).

Upon termination, the algorithm creates an aggregated RBAM (bottom-left RBAM) that is then transformed into a reshaped one (bottom-right RBAM) by replicating the columns of the aggregated RBAM exactly $K - 1$ times. As shown in Fig. 17, all the entries in the reshaped RBAM $\boldsymbol{\sigma}$ are paired and the total number of links is equal to the original optimal RBAM (bottom-center RBAM). It is easy to note that both RBAMs generate the same number of linked RBs, *i.e.,* the aggregation mapping generates an aggregated RBAM that is optimal for the RSEP. In fact, both the original and the reshaped RBAMs have 20 linked RBs.

REFERENCES

[1] Ericsson. (Jun. 2020). *Ericsson Mobility Report*. [Online]. Available: https://www.ericsson.com/49da93/assets/local/mobility-report/documents/2020/june2020-ericsson-mobility-report.pdf

[2] NIST. *Spectrum Crunch*. Accessed: Apr. 17, 2021. [Online]. Available: https://www.nist.gov/topics/Advanced-communications/spectrum-crunch

[3] L. Bonati, M. Polese, S. D'Oro, S. Basagni, and T. Melodia, "Open, programmable, and virtualized 5G networks: State-of-the-art and the road ahead," *Comput. Netw.*, vol. 182, Dec. 2020, Art. no. 107516.

[4] L. Bonati *et al.*, "CellOS: Zero-touch softwarized open cellular networks," *Comput. Netw.*, vol. 180, Oct. 2020, Art. no. 107380.

[5] A. Nakao *et al.*, "End-to-end network slicing for 5G mobile networks," *J. Inf. Process.*, vol. 25, no. 5, pp. 153–163, Dec. 2017.

[6] R. Kokku, R. Mahindra, H. Zhang, and S. Rangarajan, "CellSlice: Cellular wireless resource slicing for active RAN sharing," in *Proc. IEEE COMSNETS*, Jan. 2013, pp. 1–10.

[7] X. Foukas, M. K. Marina, and K. Kontovasilis, "Orion: RAN slicing for a flexible and cost-effective multi-service mobile network architecture," in *Proc. ACM MobiCom*, 2017, pp. 127–140.

[8] P. Rost *et al.*, "Network slicing to enable scalability and flexibility in 5G mobile networks," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 72–79, May 2017.

[1]Note that the initial unpaired solution is optimal and maximizes the number of links in $\boldsymbol{\sigma}$. Accordingly, any partial swap can produce a number of links that is at most as high as that of the initial optimal solution.

[9] V. Mancuso, P. Castagno, M. Sereno, and M. A. Marsan, "Slicing cell resources: The case of HTC and MTC coexistence," in *Proc. IEEE INFOCOM*, Apr. 2019, pp. 667–675.

[10] S. D'Oro, F. Restuccia, A. Talamonti, and T. Melodia, "The slice is served: Enforcing radio access network slicing in virtualized 5G systems," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2019, pp. 442–450.

[11] S. Mandelli, M. Andrews, S. Borst, and S. Klein, "Satisfying network slicing constraints via 5G MAC scheduling," in *Proc. IEEE INFOCOM*, Apr. 2019, pp. 2332–2340.

[12] G. Garcia-Aviles, M. Gramaglia, P. Serrano, and A. Banchs, "POSENS: A practical open source solution for end-to-end network slicing," *IEEE Wireless Commun.*, vol. 25, no. 5, pp. 30–37, Oct. 2018.

[13] D. Bega, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, "DeepCog: Cognitive network management in sliced 5G networks with deep learning," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2019, pp. 280–288.

[14] N. Van Huynh, D. Thai Hoang, D. N. Nguyen, and E. Dutkiewicz, "Optimal and fast real-time resource slicing with deep dueling neural networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1455–1470, Jun. 2019.

[15] B. Han, V. Sciancalepore, D. Feng, X. Costa-Perez, and H. D. Schotten, "A utility-driven multi-queue admission control solution for network slicing," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2019, pp. 55–63.

[16] S. D'Oro, L. Galluccio, P. Mertikopoulos, G. Morabito, and S. Palazzo, "Auction-based resource allocation in OpenFlow multi-tenant networks," *Comput. Netw.*, vol. 115, pp. 29–41, Mar. 2017.

[17] S. D'Oro, L. Bonati, F. Restuccia, M. Polese, M. Zorzi, and T. Melodia, "Sl-EDGE: Network slicing at the edge," *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (Mobihoc)*, 2020, pp. 1–10.

[18] S. D'Oro, F. Restuccia, and T. Melodia, "Toward operator-to-waveform 5G radio access network slicing," *IEEE Commun. Mag.*, vol. 58, no. 4, pp. 18–23, Apr. 2020.

[19] E. Hossain, M. Rasti, H. Tabassum, and A. Abdelnasser, "Evolution toward 5G multi-tier cellular wireless networks: An interference management perspective," *IEEE Wireless Commun.*, vol. 21, no. 3, pp. 118–127, Jun. 2014.

[20] 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 Trans. Wireless Commun.*, vol. 17, no. 5, pp. 3226–3241, May 2018.

[21] V. Jungnickel *et al.*, "The role of small cells, coordinated multipoint, and massive MIMO in 5G," *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 44–51, May 2014.

[22] R. Irmer *et al.*, "Coordinated multipoint: Concepts, performance, and field trial results," *IEEE Commun. Mag.*, vol. 49, no. 2, pp. 102–111, Feb. 2011.

[23] J. Lee *et al.*, "Coordinated multipoint transmission and reception in LTE-advanced systems," *IEEE Commun. Mag.*, vol. 50, no. 11, pp. 44–50, Nov. 2012.

[24] D. Boviz and Y. E. Mghazli, "Fronthaul for 5G: Low bit-rate design enabling joint transmission and reception," in *Proc. IEEE GLOBECOM Workshops*, Dec. 2016, pp. 1–6.

[25] W. Nam, D. Bai, J. Lee, and I. Kang, "Advanced interference management for 5G cellular networks," *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 52–60, May 2014.

[26] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2429–2453, 3rd Quart., 2018.

[27] A. Kaloxylos, "A survey and an analysis of network slicing in 5G networks," *IEEE Commun. Standards Mag.*, vol. 2, no. 1, pp. 60–65, Mar. 2018.

[28] K. Samdanis, S. Wright, A. Banchs, A. Capone, M. Ulema, and K. Obana, "5G network slicing—Part 1: Concepts, principles, and architectures," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 70–71, May 2017.

[29] Q. Zhang, F. Liu, and C. Zeng, "Adaptive interference-aware VNF placement for service-customized 5G network slices," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2019, pp. 2449–2457.

[30] C.-Y. Chang, N. Nikaein, and T. Spyropoulos, "Radio access network resource slicing for flexible service execution," in *Proc. IEEE Conf. Comput. Commun. Workshops*, Apr. 2018, pp. 668–673.

[31] O. Sallent, J. Perez-Romero, R. Ferrus, and R. Agusti, "On radio access network slicing from a radio resource management perspective," *IEEE Wireless Commun.*, vol. 24, no. 5, pp. 166–174, Oct. 2017.

[32] R. Ferrus, O. Sallent, J. P. Romero, and R. Agusti, "On 5G radio access network slicing: Radio interface protocol features and configuration," *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 184–192, May 2018.

[33] P. Zhao, H. Tian, S. Fan, and A. Paulraj, "Information prediction and dynamic programming-based RAN slicing for mobile edge computing," *IEEE Wireless Commun. Lett.*, vol. 7, no. 4, pp. 614–617, Aug. 2018.

[34] P. Caballero, A. Banchs, G. de Veciana, and X. Costa-Pérez, "Multi-tenant radio access network slicing: Statistical multiplexing of spatial loads," *IEEE/ACM Trans. Netw.*, vol. 25, no. 5, pp. 3044–3058, Dec. 2017.

[35] A. A. Gebremariam, M. Chowdhury, M. Usman, A. Goldsmith, and F. Granelli, "SoftSLICE: Policy-based dynamic spectrum slicing in 5G cellular networks," in *Proc. IEEE ICC*, May 2018, pp. 1–6.

[36] 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 Commun. Lett.*, vol. 7, no. 6, pp. 970–973, Dec. 2018.

[37] P. Caballero, A. Banchs, G. de Veciana, and X. Costa-Pĕrez, "Network slicing games: Enabling customization in multi-tenant networks," in *Proc. IEEE Conf. Comput. Commun.*, May 2017, pp. 1–5.

[38] Y. Jia, H. Tian, S. Fan, P. Zhao, and K. Zhao, "Bankruptcy game based resource allocation algorithm for 5G cloud-RAN slicing," in *Proc. IEEE WCNC*, Apr. 2018, pp. 1–6.

[39] O. Narmanlioglu and E. Zeydan, "Learning in SDN-based multi-tenant cellular networks: A game-theoretic perspective," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage. (IM)*, May 2017, pp. 929–934.

[40] S. D'Oro, F. Restuccia, T. Melodia, and S. Palazzo, "Low-complexity distributed radio access network slicing: Algorithms and experimental results," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2815–2828, Dec. 2018.

[41] V. Sciancalepore, X. Costa-Perez, and A. Banchs, "RL-NSB: Reinforcement learning-based 5G network slice broker," *IEEE/ACM Trans. Netw.*, vol. 27, no. 4, pp. 1543–1557, Aug. 2019.

[42] A. Devlic, A. Hamidian, D. Liang, M. Eriksson, A. Consoli, and J. Lundstedt, "NESMO: Network slicing management and orchestration framework," in *Proc. IEEE ICC Workshops*, May 2017, pp. 1202–1208.

[43] A. Ksentini and N. Nikaein, "Toward enforcing network slicing on RAN: Flexibility and resources abstraction," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 102–108, Dec. 2017.

[44] 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. 33137–33147, 2018.

[45] R. Mahindra, M. A. Khojastepour, H. Zhang, and S. Rangarajan, "Radio access network sharing in cellular networks," in *Proc. IEEE ICNP*, Oct. 2013, pp. 1–10.

[46] E. Dahlman, S. Parkvall, and J. Skold, *4G: LTE/LTE-Advanced for Mobile Broadband*. New York, NY, USA: Academic, 2013.

[47] M. Zambianco and G. Verticale, "Interference minimization in 5G physical-layer network slicing," *IEEE Trans. Commun.*, vol. 68, no. 7, pp. 4554–4564, Jul. 2020.

[48] P. M. Pardalos and S. A. Vavasis, "Quadratic programming with one negative eigenvalue is NP-hard," *J. Global Optim.*, vol. 1, no. 1, pp. 15–22, 1991.

[49] S. Sahni, "Computationally related problems," *SIAM J. Comput.*, vol. 3, no. 4, pp. 262–279, Dec. 1974.

[50] H. S. Ryoo and N. V. Sahinidis, "Global optimization of nonconvex NLPs and MINLPs with applications in process design," *Comput. Chem. Eng.*, vol. 19, no. 5, pp. 551–566, May 1995.

[51] F. Giannessi and F. Tardella, *Connections Between Nonlinear Programming and Discrete Optimization* New York, NY, USA: Springer, 1999, pp. 149–188.

[52] C. A. Floudas and V. Visweswaran, "Quadratic optimization," in *Handbook Global Optimization*. New York, NY, USA: Springer, 1995, pp. 217–269.

[53] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

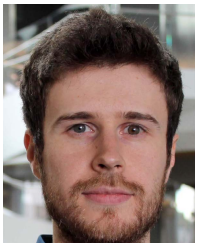
[54] X. Huang, G. Xue, R. Yu, and S. Leng, "Joint scheduling and beamforming coordination in cloud radio access networks with QoS guarantees," *IEEE Trans. Veh. Technol.*, vol. 65, no. 7, pp. 5449–5460, Jul. 2016.

[55] L. Bertizzolo *et al.*, "Arena: A 64-antenna SDR-based ceiling grid testing platform for sub-6 GHz 5G-and-Beyond radio spectrum research," *Comput. Netw.*, vol. 181, Nov. 2020, Art. no. 107436.

[56] I. Gomez-Miguelez, A. Garcia-Saavedra, P. Sutton, P. Serrano, C. Cano, and D. Leith, "SRSLTE: An open-source platform for LTE evolution and experimentation," in *Proc. ACM WiNTECH*, Oct. 2016, pp. 32–36

[57] N. Nikaein, M. K. Marina, S. Manickam, A. Dawson, R. Knopp, and C. Bonnet, "OpenAirInterface: A flexible platform for 5G research," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, pp. 33–38, Oct. 2014.

Salvatore D'Oro (Member, IEEE) received the Ph.D. degree from the University of Catania in 2015. He is currently a Research Assistant Professor with the Institute for the Wireless Internet of Things (WIoT), Northeastern University, USA. His research interests include game theory, optimization, learning, and their applications to telecommunication networks with a specific focus on 5G systems and beyond. He serves on the Technical Program Committee (TPC) for several international conferences such as IEEE INFOCOM, IEEE CSCN, and IEEE ICC. He is also a reviewer of major IEEE and ACM journals and conferences. He is an Associate Editor of the *Computer Communications* journal (Elsevier).



Leonardo Bonati (Student Member, IEEE) received the B.S. degree in information engineering and the M.S. degree in telecommunication engineering from the University of Padova, Italy, in 2014 and 2016, respectively. He is currently pursuing the Ph.D. degree in computer engineering with Northeastern University, MA, USA. His research interests include on 5G and beyond cellular networks, network slicing, and software-defined networking for wireless networks.

Francesco Restuccia (Member, IEEE) received the B.S. and M.S. degrees (Hons.) in computer science and engineering from the University of Pisa, Italy, in 2009 and 2011, respectively, and the Ph.D. degree in computer science from the Missouri University of Science and Technology, Rolla, MO, USA, in 2016. He is currently an Assistant Professor of electrical and computer engineering with Northeastern University, USA, with affiliations in the Roux Institute and the Institute for the Wireless Internet of Things, Northeastern University. His research interests include modeling, analysis, and experimental evaluation of wireless networked systems, with applications to pervasive computing and the Internet of Things. He is a member of the ACM. He regularly serves as a TPC Member for conferences such as IEEE INFOCOM and ACM MobiHoc. He is a reviewer of several ACM and IEEE conferences and journals. He was a recipient of the 2019 ISSNAF Mario Gerla Award for Young Investigators in Computer Science.

Tommaso Melodia (Fellow, IEEE) received the Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology in 2007. He is currently the William Lincoln Smith Chair Professor of the Department of Electrical and Computer Engineering, Northeastern University, Boston. He is also the Founding Director of the Institute for the Wireless Internet of Things. He is also the Director of Research for the Platforms for Advanced Wireless Research (PAWR) Project Office, a $100M public-private partnership to establish four city-scale platforms for wireless research to advance the U.S. wireless ecosystem in years to come. His research on modeling, optimization, and experimental evaluation of the Internet of Things and wireless networked systems has been funded by the National Science Foundation, the Air Force Research Laboratory, the Office of Naval Research, DARPA, and the Army Research Laboratory. He is a Senior Member of the ACM. He was a recipient of the National Science Foundation CAREER Award. He has served as the Technical Program Committee Chair for IEEE Infocom 2018; and the General Chair for IEEE SECON 2019, ACM Nanocom 2019, and ACM WUWnet 2014. He has served as an Associate Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON MOBILE COMPUTING, and *Computer Networks* (Elsevier).