Strategic Network Slicing Management in Radio Access Networks

Alessandro Lieto, *Member, IEEE*, Ilaria Malanchini, *Member, IEEE*, Silvio Mandelli, *Member, IEEE*, Eugenio Moro, *Member, IEEE*, and Antonio Capone, *Fellow, IEEE*

Abstract—*Network slicing* might radically change the relations among different actors of the telecommunications ecosystem, where new players, active in different markets, could benefit of tailored connectivity services based on different business strategies. We argue that for fully exploiting the opportunities offered by network slicing, dynamic sharing of resources is crucial not only for efficiency and cost savings, but also for enabling a resource negotiation that can unleash the potential of new business relations. We develop an automated mechanism that allows tenants to take strategic decisions to optimize the management of their slices based on their instantaneous demands and model their interaction as in marketplace. We integrate our solution, based on game theory, on a 3GPP calibrated system level simulator, where a slice-aware scheduler enforces the tenants' decisions at the Nash Equilibrium (NE). We compare our proposal with a static baseline, that assigns a fixed share of resources to each slice, and show that, by dynamically trading resources in the market, tenants achieve lower costs, and, therefore, higher profits. We provide an algorithmic implementation that guarantees the convergence to a single NE and test the computational complexity of our algorithm to an increasing number of slices in the system.

Index Terms—5G, resource allocation, dynamic sharing, network slicing, game theory, resource market

1 Introduction

With the strengthening of wireless technologies as a support for goods production and industrial processes, vertical industries have expressed new connectivity and communication needs that current wireless networks are not able to satisfy. It is expected that the upcoming more advanced releases of the standard 5G systems will be able to meet the new demand coming from different sectors, by offering network solutions tailored to the characteristics of the related vertical segment [1]. To make this possible, new mechanisms to open the way for a flexible sharing of the network infrastructure among multiple entities are necessary.

Network slicing is expected to be the key technology enabling the shift from traditional mobile communications, relying on a *one-size-fits-all* architecture, to a *service-oriented* network infrastructure, able to scale with a wide heterogeneity of vertical requirements [2]. Such technology allows multiple end-to-end (E2E) logical networks to be independently created, managed and deployed on top of a single physical infrastructure [3]. It is commonly believed that network slicing can potentially open novel business paradigms by making room for new players in the telco ecosystem [2], [4]. In addition, network slicing solutions may push in the direction of scenarios with few network infrastructure providers (i.e., the ones owning the infrastructure and providing the network resources) and several (or even many) slice tenants, which will manage network *slices* and buy resources to offer customized services to their end users [5].

On one side, the tenants are seeking dedicated and virtually isolated network solutions with full control on their Service Level Agreements (SLAs). On the other side, to exploit the business potential of network slicing [6], the infrastructure providers (InPs) aim to monetize at most their infrastructure by valuing the sharing of their networks. This conflict of interests is regulated by the trade-off between the degree of isolation of the network slices (which may result in dedicated physical resources) and the efficiency in the network resource utilization [7]. Although a dynamic sharing of resources exposes the tenants to the inherent risk of sharing, it may also bring benefits in terms of lower costs, thus more affordable performance. Indeed, dynamic sharing results in lower costs if compared to the static provisioning of resources for ad-hoc dedicated slicing solutions [6]. Moreover, the tenants may benefit from flexibility in their slice management, e.g., being able to scale up and down the resources where and when they need [8]. In particular, when applied to wireless resources, i.e., the spectrum, such flexibility would allow to cope with short-medium term fluctuations in their resource requirements, e.g., due to burst of traffic, user mobility, time-varying channel, which would be otherwise possible only through an high over-provisioning of resources.

By combining dynamic spectrum requirements with the need of satisfying heterogeneous and conflicting Quality of Service (QoS), we expect tenants to exhibit (rational)

- *Alessandro Lieto, Eugenio Moro and Antonio Capone are with the Dipartimento di Elettronica, Informazione e Bioingegneria Politecnico di Milano, 20133 Milano, MI, Italy.*
E-mail: {alessandro.lieto, eugenio.moro, antonio.capone}@polimi.it.
- *Ilaria Malanchini and Silvio Mandelli are with Bell Labs, Nokia, 70435 Stuttgart, Germany.*
E-mail: {ilaria.malanchini, silvio.mandelli}@nokia-bell-labs.com.

Manuscript received 14 Mar. 2020; revised 29 July 2020; accepted 9 Sept. 2020. Date of publication 18 Sept. 2020; date of current version 4 Mar. 2022.
(Corresponding author: Alessandro Lieto.)
Digital Object Identifier no. 10.1109/TMC.2020.3025027

strategic behaviors, i.e., taking selfish decisions in line with their private business models. Indeed we consider the profit-driven nature of slice tenants, where the optimal allocation of resources cannot be seen as a centralized decision taken by a unique entity. In this scenario, the diverse business interests of the stakeholders must be modeled by considering the presence of multiple real decision makers with conflicting objectives, i.e., slice tenants and their own profits, that compete on the resources to be allocated to their slices. Specifically, we introduce a dynamic marketplace, where the price of resources change over time and space and it is adapted according to supply and demand in a way similar to what happens in the energy market as well as more recently for cloud resources, e.g., Amazon AWS [9]. In this marketplace, the tenants take strategic decisions when purchasing resources in the market, which derive from the trade-off between the QoS they want to provide to their end users and the resulting price they are willing to pay to the infrastructure provider. Therefore, since resource optimization is performed and driven by techno-economic tenants' preferences, we model the resource allocation in network slicing not as a centralized problem, rather as a market game, that is regulated by the infrastructure provider. From an implementation point of view, this would correspond to independent software agent for each tenant, which monitors their techno-economic key performance indicators and optimize online their services according to their own private preferences.

In order to address the aforementioned issues, we generalize and validate the approach introduced in [10], which allows tenants to automatically trade real-time and in each cell the network resources in a shared marketplace. The proposed resource renegotiation mechanism adapts the resource share of each tenant in the medium-shot term, every few seconds or minutes. The decisions taken are then enforced in each cell by a slice-aware radio scheduler which, by enforcing SLAs on configurable time averages and not every Transmission Time Interval (TTI), achieves further pooling gains compared to resource partitioning [11]. Like in many economic models, the game-theoretical formulation of this marketplace defines the rules for the contention of the scarce wireless resources among the tenants.

1.1 Key Contributions

The key contributions of this paper are summarized in the following.

- We develop an *automated* negotiation mechanism, where tenants can trade radio resources in the medium-short term. Our approach can be related to the trading in stock markets, where fully-automated decisions are taken by properly designed software agents to capitalize short term events.
- We propose a heuristic that allows fast convergence to a NE also for a quite high number of players competing in the market. Customized resource overload control policies are also considered, geared towards the stability of the system at the achieved NE.
- We map the high-level descriptors of slices into specific network domain parameters, adapting the corresponding slice resource demand, based on (i) the slice high-level requirements, (ii) the load and user conditions for the given cell and time interval. In this paper, we focus on radio resources, given the additional challenge of dealing with the wireless dynamic channel conditions.
- We consider the tenants as independent decision makers. Accordingly, they do not delegate decisions to a central controller, to whom they would need to expose sensitive business information. The interactions among the tenants and the infrastructure provider for the negotiation of radio resources is modeled as in a marketplace, where the prices are automatically adjusted according to supply and demand.

1.2 Structure of the Paper

The paper is structured as follows. In Section 2, an overview of the state of the art is discussed. We analyze the prior works related to resource allocation in network slicing, as well as similar renegotiation approaches already applied in other industry markets. We base our work on the framework proposed in [10] and introduce our approach in Section 3, along with the system model adopted in this manuscript. In Section 4, we define the market game and analyze the theoretical properties of our formulation, while Section 5 describes the algorithmic implementation and analyzes the characteristics of the achieved Nash Equilibria. In Section 6, we test our algorithm on a 3GPP calibrated system level simulator and validate the proposed approach against state of the art solution based on static resource provisioning. We summarize our conclusions in Section 7.

2 Related Works

The new challenges driven by network slicing impose novel techniques to control the allocation of resources among multiple tenants, which account for both efficiency in resource management and algorithmic complexity [12]. Standardization bodies, like 3GPP, have already defined schemes and functional descriptions for sharing a common infrastructure among multiple network operators. The specification in [13], though, suggests a static partition of resources among multiple operators. While such approach results in quite-low complexity, it is quite known that dynamically sharing a pool of resources provides higher multiplexing gains and resource efficiency [14], [15], [16].

Following this direction, the management of resources in network slicing has been considered so far mainly as a centralized problem, where a single entity (e.g., the InP or a slice manager) defines the sharing policy to allocate the resources to multiple slices [17], [18]. Namely, the authors in [17] propose a priority-based scheme, that assigns to each tenant a priority weight, which is applied by the radio scheduler in order to allocate resources to end users. They show that the proposed scheme achieve higher fairness and QoS performance compared to baseline schemes. Along the same lines, the work in [18] proposes an auction model for resource allocation, that maximizes the infrastructure provider revenues, while guaranteeing a given priority level for each slice, depending on the business agreement in place. Other works, such as [5], [19], [20], focus on optimal resource allocation policies which minimize the risk of SLA violations, while maximizing resource efficiency and/or infrastructure provider revenues. Furthermore, a common approach that is usually considered for resource allocation in centralized system concerns the maximization of a social optimum function, e.g. defined as the sum of utility functions (or preferences) of the different tenants [21], [22]. However, all these centralized solutions do not give enough control to the tenants, excluding them to take any decision in the management of resources for their slices. In other words, the interest of the single tenant is not directly taken into account whenever its interest collides with the others', which may lead issues in convincing tenants to accept the solution.

To address the strategic behavior of tenants, many works based on game theory have been proposed in the literature. For example, [23], [24] propose a game-theoretic formulation where tenants can dynamically redistribute their fixed share of resources among different cells. The proposed formulation corresponds to a Fisher market, where players have a fixed budget and bid for resources, subject to budget limitation. Their formulation remains quite general, assuming the case of elastic users, without specifically addressing the diverse requirements that are expected to be served by the different slices. The same authors of [23] extend the prior work to the case of inelastic users, showing that, unless under proper dimensioning of the network and suitable admission control policies, a NE cannot be always guaranteed when strict QoS requirements must be enforced [25]. However, they still focus on a static scenario, where a fixed resource share is pre-agreed and determined somehow by a genie. This approach can be suitable for slicing systems run by few big tenants, while our proposal can be applied to any multi-tenants system, in which any tenant can buy resources only when and where needed, allowing also micro-slices to enter in this business.

All the aforementioned works do not consider the economic impact of the decisions taken. [26], [27] model the resource allocation for slices as a trading mechanism. Although the work in [26] models the trading as a Stackelberg techno-economic game, the utility functions adopted in their formulations do not describe effective and simple economical quantities, like profits and costs. The idea of techno-economic models based on game theory for dynamic resource allocation problems have been successfully adopted already in diverse fields. For example, the approach used in Google AdWords for online advertisement allocation in multiple-agent systems [28] implements automated bid strategies, which can be executed by a central controller, under the specification of sensitive parameters customized by the end-user [29]. Similarly, [30] proposes a mechanism design to optimally manage the energy consumption for smart grid applications in energy market. However, those two approaches assume utility functions to be known by a central controller, raising agents' privacy issues. Indeed, some information may not be necessarily available at the central controller: for example, the agents may not be willing to expose their own preferences and business evaluations both to a central entity or to the other agents. In this work, we present a distributed solution, where decisions are taken autonomously by each agent, without the need nor risk of exposing private sensitive information to any entity external to the tenant itself. A similar approach has been recently proposed in [31]. In this work, the authors define a trading mechanism for dynamic resource allocation for slices, where the business model of the tenants consists of two interfaces, one toward the InP, and the other toward their end users. Their resource trading mechanism is modeled as a Markov Decision Process (MDP) and solved by Q-learning algorithm. A distributed approach based on game theory is also proposed in [32]. Although the authors provide an algorithmic implementation with low complexity for applications in radio access networks, they do not assess the implications of the proposed formulation on the service performance of the slices, in particular in highly congested scenarios. In our formulation, we mainly focus on overloaded network conditions, where the renegotiation of resource shares among the slices can give helpful indications to handle critical conditions, for example to guide admission and overload control policies.

3 Slicing Management Framework

In [10] we introduce a Slicing Management Framework (SMF) that can be applied to any system that is in charge of the management of network resources in multi-tenant networks. The SMF allows to translate the high-level policies defined by the tenants (in the form of SLAs or slice templates [33]) to particular policies (e.g., resource requirements/consumption) to be enforced at different layers of the network. It automatically modifies and adapts resource and performance requirements to the actual needs of their slices. The need of a dynamic renegotiation of resources comes to accommodate the end-to-end performance of the slices together with an efficient allocation of wireless resources. The former are defined in terms of aggregate QoS requirements (i.e., expected throughput and latency) covering a large geographical area and lasting a variable time duration; the latter must be computed at a different granularity, i.e., at cell level and in very short-time, specifically in each Transmission Time Interval (TTI) to guarantee efficiency and multiplexing gain. Although we do not exclude the case of reserved resource guarantees to slices in each area, it is well-known that allocating and renegotiating resource shares dynamically (i.e., assigning them only when needed to satisfy the long-term performance requirements and user QoS targets) brings gains in terms of efficiency in the resource utilization as well as cost savings [6]. Such an automated and scalable system is needed to guarantee a flexible and efficient management of the network for a generic high number of slices, able to adapt itself in real-time without the need of human operation.

We design the SMF as a decision making process, as sketched in Fig. 1: by moving from top to bottom, not only the type of decisions changes, but also their complexity and effectiveness, i.e., the duration of the decisions taken and the geographical extent of those decisions. At the top of the hierarchy, long-term decisions are taken and those may last a variable time (e.g., weeks, days or hours) depending on the life-cycle of the individual slice. At this level, the tenants define, together with the InP, the high-level descriptors of their slices, in form of techno-economic Key Performance Indicators (KPIs). These KPIs describe the specific slice business model and may include, among others, macro-area/per-cell constraints on aggregate throughput/resource utilization, and/or information about the QoS requested by the

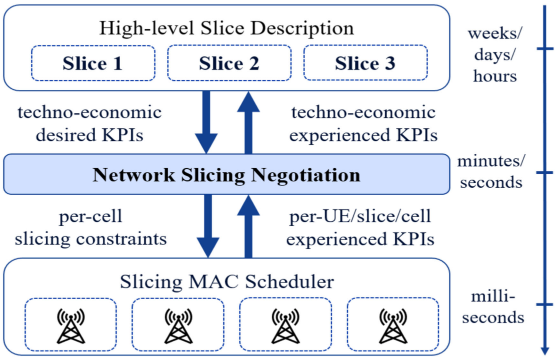


Fig. 1. Slicing management framework.

users of the slice. By definition, the aforementioned descriptors are valid in a long-time window and over the entire spatial coverage area of the slice. Even though these descriptors generalize the expected behavior of the slice, they cannot capture the dynamic evolution of the radio environment. For instance, operators and tenants may agree on the average resource share of a slice, given an average user distribution for a specific area and a user target QoS. However, this does not provide indications on the decisions to be taken in case of bursts of traffic for some slices. For this reason, we model the preferences of the tenants through *utility functions*, which describe the behavior of slices based on their target KPIs, reporting the degree of flexibility to adapt their requirements to the underlying network conditions.

We identify as the Network Slicing Negotiation (NSN) the entity responsible to convert the slice high-level descriptors into physical requirements/constraints for the MAC scheduler (e.g., resource share for each slice). The NSN allows to accommodate the requirements of the slices according to the actual needs of their users. Renegotiations of resource shares among tenants take place whenever network conditions change, e.g., due to handovers, change of users behavior or position. How these are handled is defined by techno-economic KPIs, assessing the profit of given achieved performance. To model the contention of resources, we consider a resource market that regulates the negotiation of the allocations among slices. Hereafter we assume that the tenants compete for all the available resources, however our considerations also hold in hybrid scenarios, where a part of the resources is statically assigned and the other part is contented among them.

Finally, given the outcome of the NSN in form of low-level KPIs (e.g., slice/user average throughput/resource allocation), radio resources are allocated at every TTI to single users by a Slicing MAC Scheduler (SMS). Previous works in the literature [11], [34] address the SMS problem. Namely, they propose solutions where resources are allocated to users relying on proportional fair metrics, twisted to take into account user/slice constraints received by the higher layers. Those constraints are satisfied by averaging the allocations in time windows of $500-1000$ ms, which, in turn, allows to achieve further pooling gains compared to resource partitioning [11].

In this work, we focus on the modeling and characterization of the NSN. In particular, the NSN must (i) process the high-level KPIs coming from the tenants and input its decision to the SMS of each cell, (ii) adapt those decisions based on SMS feedback, and (iii) report achieved performance to the topmost layer, where tenants may monitor and correct their long-term policies. According to the scheme in Fig. 1, the NSN adaptation and decision process take place in time windows of seconds to minutes.

3.1 System Model

We consider a simple yet meaningful network slicing setup, where a single infrastructure provider accommodates a set of slices, $\mathcal{S}$, each of them corresponding to a single tenant[1]. We assume that slices share a common pool of radio resources which are allocated to each slice by the radio scheduler of each base station. We denote by x_s the amount of radio resources allocated to slice s, normalized by the system bandwidth, i.e., $x_s \geq 0, \forall s \in \mathcal{S}$ and $\sum_{s\in\mathcal{S}} x_s \leq 1$. The resources x_s are then distributed and allocated to each user of that slice by the SMS scheduler. Let $\mathcal{K}_s$ be the set of active users of slice s, and $x_k \geq 0$ be the amount of resources assigned to the user k, according to some mapping $x_k = f(x_s; \mathcal{K}_s)$ such that $\sum_{k\in\mathcal{K}_s} x_k = x_s$. Moreover, let $\mathcal{K}$ be the set of all users connected to the network, such that a single user $k \in \mathcal{K}$ belongs to a unique slice, i.e., $\bigcap_{s\in\mathcal{S}} \mathcal{K}_s = \emptyset$ and $\bigcup_{s\in\mathcal{S}} \mathcal{K}_s = \mathcal{K}$. This assumptions holds just for ease of notation: indeed, the case of a user belonging to multiple slices, e.g., requiring different traffic profiles simultaneously, can be handled by virtually copying that user in multiple replicas, one for each slice.

The NSN aims at finding a stable resource allocation that satisfies all the requirements of the tenants. In order to do so, we map the techno-economic KPIs coming form the northbound interface in Fig. 1 in *utility functions*, which reflect the economic evaluation of a slice, i.e., the expected revenues of a slice for the allocated resources. We denote by $R_s(x_s)$ the revenue function of a tenant. We assume that the revenues of tenants increase with an increase in the quality of service experienced by their users. To this purpose, we consider an *acceptance probability function*, $A_k(r_k(x_k)) \in [0, 1]$, which models the level of satisfaction of a user, due to the experienced QoS, $r_k(x_k)$, which in turn depends on the obtained resources, x_k. Then, the revenue can be computed as

$$R_s(x_s) = \sum_{k\in\mathcal{K}_s} \chi_k \cdot A_k(r_k(x_k)), \tag{1}$$

where χ_k denotes the economic value of each user for the tenant.

Although the shape of revenue functions, i.e., user acceptance probability, can be privately defined and customized by each tenant, we assume that any revenue function satisfies the following properties:

$$\frac{dR_s}{dx_s} > 0, \tag{2a}$$

$$\lim_{x_s\to+\infty} \frac{dR_s}{dx_s} = 0. \tag{2b}$$

1. Note that the one-to-one correspondence between tenants and slices is made only for ease of notation and clarity. Throughout the paper, we will refer to both interchangeably.

Inequality (2a) implies that any increase in the resources allocated to a slice leads to larger revenues, since more resources generally result in better QoS and higher user satisfaction. However, it is also true that the improvement in performance will vanish when a certain QoS is already achieved. This represents the law of diminishing return adopted in many economic systems, captured by Eq. (2b).

As mentioned above, we are dealing with a scenario where the shared resources are limited, namely by the total bandwidth of each cell.

Assuming that tenants are selfish agents pursuing their own private interests, centralized solutions based on social welfare maximization may not necessarily favor them. We would rather consider that tenants are willing to compete against each other for the resources to be assigned to their slices. Hence, we introduce the definition of a market of resources, which regulates the interaction among the tenants for the competition on radio resources.

3.2 The Market of Resources

We recall the concept of *purchasing power* adopted in economics to model the market mechanism of our system. In economics, the purchasing power defines the amount of goods that can be bought by a unit of currency. Since in our system the goods are defined as radio resources, and they are scarce by nature, we assume that the purchasing power decreases with the resource availability in the cells. We define the relative normalized cell load as

$$l = \sum_{s \in \mathcal{S}} x_s \quad \in [0,1], \tag{3}$$

which defines the domain of the purchasing power, i.e., $pp : [0,1] \rightarrow \mathbb{R}^+$. We can compute the cost of a unit of resource by the following relationship

$$C(x,l) = \frac{x}{pp(l)}, \tag{4}$$

where $C(\cdot)$ is the cost of buying x resources for a generic slice s, given the actual load l. Notice that since $C(0,l) = 0$, the market has no entry barriers. At this stage, we do not discriminate between tenants and assume the purchasing power to be equally defined for all of them. In this way, all the tenants are subject to the same pricing policy. However, we do not exclude the case where the infrastructure provider may differentiate the pricing policies over different slices.

4 Problem Formulation and Properties of the Game

The analysis of the proposed market model is based on game theory, which is well suited to study the rational behaviors of autonomous entities.

Let $\Gamma = \langle \mathcal{S}, (X_s)_{s\in\mathcal{S}}, (u_s)_{s\in\mathcal{S}} \rangle$ be the *market game* in strategic form, where:

- $\mathcal{S} \doteq \{1, 2, \ldots, S\}$ is the finite set of players (i.e., the tenants/slices);
- $X_s = [0,1]$ is the pure strategy space (i.e., quantity of resources to buy) of the generic player $s \in \mathcal{S}$;
- $\mathbf{X} \doteq X_1 \times X_2 \times \ldots \times X_S$ is the Cartesian product of all player strategy spaces, and $\mathbf{x} \in \mathbf{X}$ denotes a strategy profile of the game;
- $u_s : \mathbf{X} \rightarrow \mathbb{R}$ is the payoff of player s, which is defined as:

$$u_s(\mathbf{x}) = u_s(x_s, \mathbf{x}_{-s}) = R_s(x_s) - C_s(x_s, \mathbf{x}_{-s}). \tag{5}$$

The expression in Eq. (5) highlights the dependency of the payoff of player s on its own strategy, x_s, as well as on the other players' strategies, $\mathbf{x}_{-s}$. One can notice that the payoff of the players consists of i) a player specific component, $R_s(x_s)$, that defines the economic value of the slice for the purchased resources and ii) a shared component,

$$C_s(x_s, \mathbf{x}_{-s}) = C\left(x_s, \sum_{t\in\mathcal{S}} x_t\right) = C(x_s, l), \tag{6}$$

that accounts for the cost of buying x_s resources, which is affected also by the decisions of the other players, $\mathbf{x}_{-s}$. Due to the intrinsic economic nature of the payoff functions, in the remainder of the paper we also refer to them as *profit functions* of the tenants. Let also define the *best response correspondence* of a player as the strategy (or strategies) that maximizes the payoff of a player given all other players' strategies. Formally, we define $BR_s : \mathbf{X}_{-s} \rightarrow X_s$ such that:

$$BR_s(\mathbf{x}_{-s}) = \arg\max_{x_s \in X_s} u_s(x_s, \mathbf{x}_{-s}), \quad \mathbf{x}_{-s} \in \mathbf{X}_{-s}, s \in \mathcal{S}. \tag{7}$$

We now elaborate some relevant properties characterizing the market game Γ.

4.1 Analysis of the Game

Since our game belongs to the class of aggregative games[2], we can derive the existence and convergence properties of the game Γ from the fundamental result in [35]. In order to prove the existence of a NE for the proposed market game, we consider the following two assumptions about the components of the payoff function of the players.

Assumption 1. *The cost function $C(x,l)$ is convex, continuous and twice differentiable in x and l.*

The convexity assumption models the behavior for increasing load in the system, where we expect that the cost for unit of resources increases accordingly.

Assumption 2. *The tenant revenue function $R_s(x_s)$ is continuous, increasing and twice differentiable in the strategy space X_s.*

Based on these assumptions, we first analyze some properties of the best response correspondences.

Lemma 1. *Given utility functions $u_s(\cdot)$ satisfying Assumption 1 and Assumption 2, the best response correspondences of the game Γ always admit a single-valued decreasing selection.*

Proof. Consider the generic best response of a player s, $BR_s(\mathbf{x}_{-s}) = BR_s(\sum_{t\in\mathcal{S}\setminus\{s\}} x_t)$. We consider the selection ϕ as the largest element in a set, i.e.

2. An *aggregative game* is a game in which every player's payoff is a function of the player's own strategy, x_s in game Γ, and the aggregate of all players' strategies, defined as l in game Γ.

$$\phi\left(BR_s\left(\sum_{t\in\mathcal{S}\setminus\{s\}} x_t\right)\right) = \sup\left\{BR_s\left(\sum_{t\in\mathcal{S}\setminus\{s\}} x_t\right)\right\}.$$

This selection is not empty, single-valued and decreasing. From the definition of best response in Eq. (7), given the continuity of the utility functions $u_s(\cdot)$ in the compact set X_s, we can claim that the function always admits a maximum value, and, therefore, the best response correspondence is not empty. Moreover, the $\sup\{\cdot\}$ operator returns a single value, hence ϕ is single-valued. Finally, we can prove that ϕ is decreasing by showing that

$$\frac{\partial}{\partial x_s \partial x_t}[u_s(x_s, \mathbf{x}_{-s})] < 0 \;\forall s, t \in \mathcal{S}\;, s \neq t.$$

The derivation is left to the reader. The reasoning behind it is that when any of the opponents' action increases, an increase in the own action leads to a smaller utility. Therefore, the best response to an increase in the opponents' action is always a decrease in the own action, proving that the best response of each players is decreasing in the variable $\sum_{t\in\mathcal{S}\setminus\{s\}} x_t$. □

The key message of Lemma 1 is that the best response of a player decreases to any increase in the aggregate of his opponents' strategies. We can now state the following :

Theorem 1. *For any utility function $u_s(\cdot)$ verifying Assumption 1 and Assumption 2, the game Γ always admits a (not necessarily unique) NE in pure strategies.*

Proof. We leverage the result in [35, Corollary 1] to prove the existence of a pure strategy NE. In order to apply [35, Corollary 1], we need to prove that the following conditions apply to the market game Γ:

- (i) The strategy sets are compact,
- (ii) The payoff functions are upper semi-continuous and continuous in the opponents' strategies,
- (iii) The best response correspondences admit a decreasing selection,
- (iv) The continuous aggregative functions $f_s(x_s, \sigma_s(\mathbf{x}_{-s})) = x_s + \sum_{t\in\mathcal{S}\setminus\{s\}} x_t,\ \forall s \in \mathcal{S}$, exhibit strictly increasing differences in x_s and $\mathbf{x}_{-s}$ (possibly after a strictly monotonic transformation).

The conditions (i), (ii) are verified by the definition of the strategy space $\mathbf{X}$ and Assumption 1 and Assumption 2, since upper semi-continuity of condition (ii) is guaranteed by the continuity of the functions $R_s(x_s)$ and $C_s(x_s, \mathbf{x}_{-s})$. Condition (iii) holds by Lemma 1. In order to prove condition (iv), we can use a monotonic transformation $h \circ f_s$, e.g., $h(z) = exp(z)$ as done in [35, Example 3] and verifying that $\frac{\partial}{\partial x_s \partial x_t}[h(f_s(x_s, x_t))] > 0$, which implies the strictly increasing differences property. Thus, we can apply [35, Corollary 1] and prove the existence of a NE of the game Γ. □

5 Algorithmic Approach and Nash Equilibria Analysis

To analyze the quality and quantity of the equilibria of the proposed market game, we now introduce specific revenue and cost functions, that will be used for the remainder of the paper. As done in [36], we define the per-user acceptance probability function introduced in Section 3 as:

$$A_k(r_k(x_k)) = 1 - q_s^{\left(\frac{r_k(x_k)}{r_s^0}\right)^{\mu_s}}, \quad \forall k \in K_s, \forall s \in \mathcal{S}, \tag{8}$$

where:

- μ_s captures the slice elasticity to QoS degradation, i.e., the higher μ_s, the more inelastic the service is to performance degradation,
- $r_k(x_k)$ is the achievable QoS of a user,
- r_s^0 is the maximal QoS value that characterizes the service offered by the specific slice,
- q_s is a parameter tuning the acceptance function value, i.e., $A_k(r_k(x_k) = r_s^0) = 1 - q_s$.

The per-slice tuple of elements (μ_s, r_s^0, q_s) represents the techno KPIs of the slice, i.e., the parameters that best describe the expected performance and behavior for the slice. Finally, we can compute the revenue function in Eq. (1), by weighting the acceptance values with the per-user economic parameter, χ_k.

In Section 3, we have defined the cost function through an inverse proportional relationship with the purchasing power, $C(x, l) \propto \frac{1}{pp(l)}$. In this work, we model the purchasing power as:

$$pp(l) = 1 - \frac{1}{1 + e^{-\alpha(l-l_0)}}, \tag{9}$$

that is a sigmoid, approaching to zero when the load l exceeds the available spectrum. The parameters α and l_0 characterize the steepness and the midpoint of the sigmoid, therefore shaping the overall sigmoidal function.

In the following, we still consider a generic setting of the slices for the only purpose of analyzing the properties of the equilibria and the convergence of the algorithm.

5.1 Convergence of the Best Response Dynamics

In algorithmic game theory, one of the most common tool to study the convergence to a NE is the application of the Best Response Dynamics (BRD) algorithm. BRD is an algorithm where each player, one at a time and in turn, updates his optimal strategy after observing the decisions of the other players, which, for the market game Γ, all converge to a single value, i.e., the aggregate load. The convergence to a NE is achieved when no player further moves from his previous selected strategy. In this sense, BRD is a *sequential improvement path* based algorithm, i.e., at each turn players move to a strictly preferred strategy, if it exists, or stay with the previous strategy if it does not. We implement the BRD algorithm for the game Γ by sampling the strategy space of the players, X_s, with sample rate $\Delta x > 0$. This choice comes from the discrete nature of wireless resources at scheduler level, where resources are grouped in Resource Blocks (RBs) and assigned to end users accordingly. Notice that the general properties of the game, described in Section 4, are not affected by the redefinition of the strategy space in a finite set. Moreover, we can prove that, for the finite version of the game Γ, it is always possible to converge to a pure strategy NE in a finite number of steps.

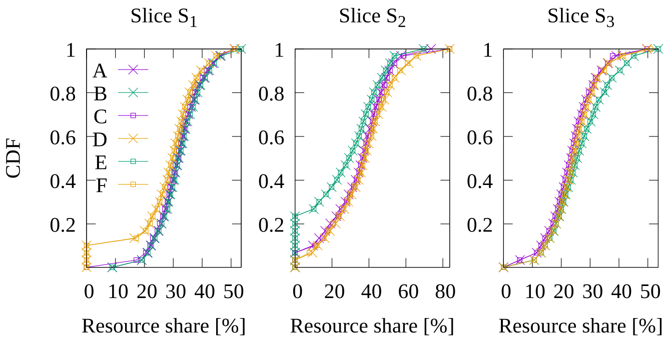


Fig. 2. Distribution of per-slice resource share at NE with BRD.

Theorem 2. *Consider the finite version of the market game* $\tilde{\Gamma} = \langle \mathcal{S}, (\tilde{X}_s)_{s\in\mathcal{S}}, (u_s)_{s\in\mathcal{S}} \rangle$, *with finite strategy space* $\tilde{X}_s$. *Under Assumption 1 and Assumption 2, the BRD algorithm always converges to a set of pure strategy Nash Equilibria of the game* $\tilde{\Gamma}$, *regardless of the initial condition.*

Proof. Similarly as done for Theorem 1, we can apply the result of [37] to prove the convergence of BRD. Indeed, considering the same conditions already verified in Theorem 1, we can apply [37, Theorem 3], for which any admissible sequential improvement path converges to a set of pure strategies Nash Equilibria. Being the BRD algorithm based on sequential improvement path, the thesis is easily proved[3]. □

5.2 Equilibria Characterization

We apply the BRD algorithm to analyze the Nash equilibria of the market game defined in Section 4, given the utility functions described above. Since the outcome of BRD is usually affected by the initialization strategies as well as the orders in which players play their best response, we consider different fixed playing orders for the execution of BRD, assuming all the permutations of orders obtainable from the set of three slices, $\mathcal{S} = \{S_1, S_2, S_3\}$. We label the different orders as A = $[S_1, S_2, S_3]$, B = $[S_1, S_3, S_2]$ ..., F = $[S_3, S_2, S_1]$. We assume that BRD starts from a zero allocation for all the players, i.e., $x_s^{(n=0)} = 0, \ \forall s \in \mathcal{S}$. We consider 200 simulation drops, with random user displacement throughout the coverage area of 21 cells. The resource renegotiation is performed on each cell for all the simulation drops and the results are presented as the resource allocations (i.e., percentage of resource share) obtained by each slice at the NE. In Fig. 2 we show the Cumulative Distribution Function (CDF) of the resource shares of the slices at the NE (collected from all the different simulation drops and cells), when BRD is executed following the aforementioned playing orders. Results show that the order of execution plays a fundamental role determining the strategy at the equilibrium, proving that the game Γ admits, in general, multiple equilibria[4]. For completeness, we have also considered the case in which both the playing order and initial strategies are random (rather than using a fixed initial strategy set to zero). Simulation results have shown that, depending on the random initial strategy and on the random playing order, we may end up in different sets of Nash equilibria, obtaining similar results to the ones we have discussed in Fig. 2. This confirms that both the playing order and the strategy initialization may lead, when random, to arbitrary equilibrium selection, with non predictable properties. Hereafter, we assume that the playing order of the game cannot be controlled nor fixed by a system designer and any implementation scheme in a real system should be independent from it. In contrast, we believe that it is reasonable to assume that tenants start playing their game by a zero allocation. Therefore, we introduce an alternative algorithm to ensure convergence to a single NE, independently from the orders in which the players update their strategies, while assuming an initial zero allocation.

3. The main result we obtain in [37] is that the game Γ can be classified as a *best-reply potential game* [38], for which it is always guaranteed the convergence of BRD in a finite number of steps. For further details about the convergence of improvement dynamics in finite games the reader can refer to [37].

4. Indeed, only by assuming concavity of the payoff functions one can guarantee the uniqueness of the equilibrium for aggregative games [39].

Algorithm 1. Double Step Smoothed Best Response

procedure DSSBR $\mathcal{S}, X$
  initialize $\mathbf{x}^{(0)}$
  $n = 0$
  while ($stop\_bucket = 0$ or $n \leq n_{\max}$) **do**
    update (10)
    for $s \in \mathcal{S}$ **do**
      $u_s^{(n)} \leftarrow u_s(x_s^{(n)}, \mathbf{x}_{-s}^{(n)})$
      $\hat{x}_s \leftarrow \arg\max_{x_s} u_s(x_s, \mathbf{x}_{-s}^{(n)})$
    $\mathbf{x} \leftarrow \mathbf{x}^{(n)} + \gamma_n(\hat{\mathbf{x}} - \mathbf{x}^{(n)})$
    if $\mathbf{x} = \mathbf{x}^{(n)}$ **then**
      $stop\_bucket = 1$
    else
      $n \leftarrow n + 1$
      $\mathbf{x}^{(n)} \leftarrow \mathbf{x}$ update the allocation
  Apply BRD with $\mathbf{x}^{(0)} = \mathbf{x}$

5.3 Double Step Smoothed Best Response Algorithm

We propose a Double Step Smoothed Best Response (DSSBR) algorithm to force the convergence to a single NE. Differently from the classic BRD algorithm, at the first stages the players move all together by playing simultaneously their best response strategies. In this way, the players react to the same external conditions, i.e., the same previous total load, $l = \sum_{s\in\mathcal{S}} x_s$. The first steps of the algorithm proceed in the following way. Given an initial position, $\mathbf{x}^{(0)}$, and a desired optimal position (the player best responses), $\hat{\mathbf{x}}$, the actual movement of the player is *smoothed* by a factor γ. In other words, if $\mathbf{\Delta}_x^{\text{req}} = \hat{\mathbf{x}} - \mathbf{x}^{(0)}$ defines the requested player movements from their previous positions, the actual movement of the players will be smoothed as $\mathbf{\Delta}_x^{\text{act}} = \gamma \cdot (\hat{\mathbf{x}} - \mathbf{x}^{(0)})$. Therefore, the players at time $n = 1$ will move in $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \mathbf{\Delta}_x^{act}$. The value of γ is updated at each iteration step of the algorithm (i.e., at each step n), according to the following rule:

$$\gamma[n] = \gamma_0 \cdot \frac{\log(n+1)}{(n+1)^\beta}, \ \gamma_0 > 0, \beta < 1, n \in \mathbb{N}, \qquad (10)$$

The idea behind Eq. (10) is to control the movements of the players during the first execution phases of the algorithm.

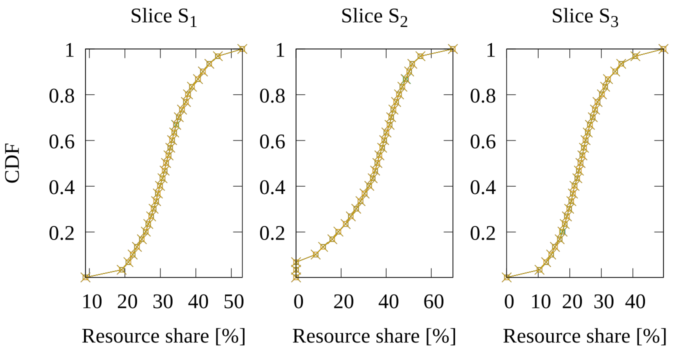


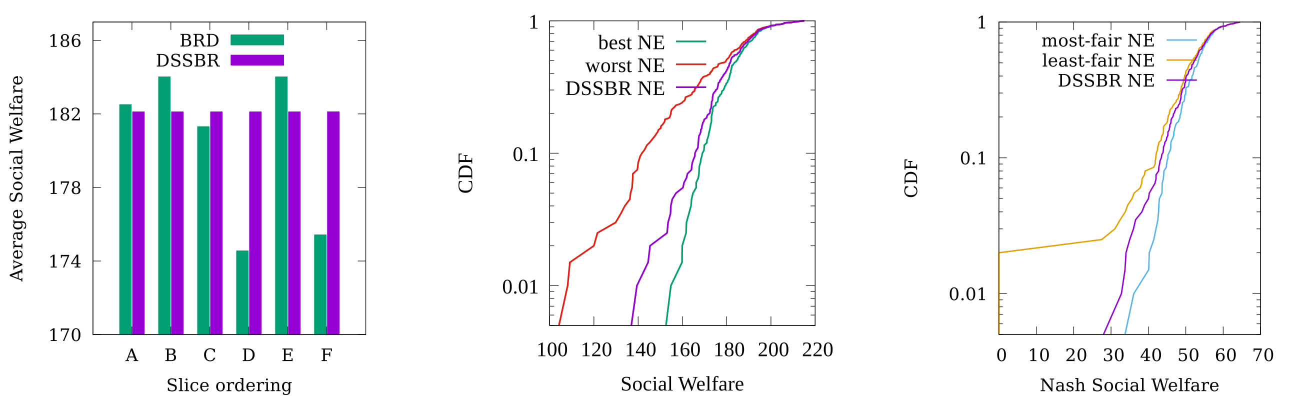
Fig. 3. Distribution of per-slice resource share at NE with DSSBR

Indeed, we may expect that, at the first steps of the algorithm, the players will move further from their initial allocation (e.g., n assuming they start from a zero allocation), causing a big increase in the overall load of the cell, l. Therefore, at the first steps of the algorithm, we let the players play their strategy and smoothly move towards their desired outcome, in order to avoid big jumps. However, due to the non-concavity of the payoff functions, we cannot guarantee the convergence to a NE when players play simultaneously [40]. Therefore, after a transient phase (i.e., maximum number of iterations, $n_{\max}$, or convergence of the first stage), we remove the smoothing factor and apply the classic BRD, where each player in turns best responds to the opponents' strategies. The full description of the algorithm can be found in Algorithm 1.

By applying this double step approach, we can still guarantee the convergence to a NE, given the convergence criterion of Theorem 2. We set the parameters of DSSBR in the update rule of Eq. (10) equal to $\beta = 0.5, \gamma_0 = 0.5$. In Fig. 3, we show the distribution of the allocations at the NE achieved by DSSBR, considering (for the second phase of DSSBR) the same simulation scenarios and execution orders already adopted for BRD. It shows that DSSBR always converges to the same equilibrium, making the equilibrium selection of the game independent of the order in which players execute their best responses. This is a fundamental result to leverage a *fair competition* in the market, where no player can experience any advantage due to the order in which the best responses are executed or due to the random initialization of the strategy. In Fig. 4, we also analyze the properties of the equilibria obtained by DSSBR and BRD. We define two metrics to compare the NE allocations, based on the payoff of the players: (i) the *Social Welfare*, $SW(\mathbf{x}) = \sum_{s\in\mathcal{S}} u_s(\mathbf{x})$, that measures the total welfare of the system, and (ii) the *Nash Social Welfare* [41], $NSW(\mathbf{x}) = \left(\prod_{s\in\mathcal{S}} u_s(\mathbf{x})\right)^{\frac{1}{|\mathcal{S}|}}$, that measures the social fairness among the players.

In Fig. 4a, we show the average social welfare that is achieved by the two algorithms, when averaging the social welfare values obtained at the NE over 200 randomly simulated scenarios. Since DSSBR always converges to the same equilibrium, the purple bars show the same value of SW for any playing order. Differently, the green bars show different values, based on the playing order, due to the different set of NE that are found by BRD. In Fig. 4b, we pick, for each random instance, the best and worst NE in terms of social welfare SW among the ones found with BRD for the 6 different playing orders and compare it with the single one obtained with DSSBR. We apply the same procedure in Fig. 4c to evaluate the Nash social welfare, NSW. By comparing the distributions in Fig. 4b, we see that the equilibria achieved by DSSBR are closer to the "best NE", i.e., the equilibria maximizing the social welfare of the system, rather than the "worst NE", i.e., the ones minimizing the social welfare. Furthermore, in Fig. 4c, one can see that DSSBR provides also a good trade-off in terms of fairness among the players. Indeed, some of the equilibria achieved by BRD can result in highly unfair allocations ("least-fair NE"), where one of the player achieves zero or quasi-zero payoff, whereas DSSBR avoids such situations. Therefore, DSSBR can perform close to the maximum social welfare equilibria, while preserving fairness among the players.

Notice that we do not make any general claim about the *optimality* of the equilibria at this stage. Indeed, we are just providing a comparison among a subset of the possible equilibria achieved by BRD and the unique ones found with DSSBR. In the next section, we provide a more detailed analysis on the optimality of the NE achieved by DSSBR and compare the solutions at the NE with the optimal solution that would be computed by a central controller.

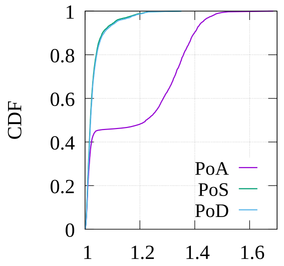


(a) Average social welfare of the system at NE with the proposed algorithms considering different player orderings

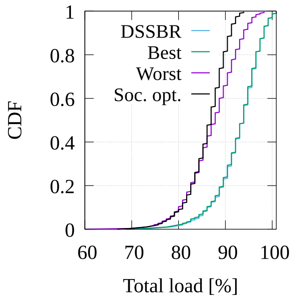
(b) Comparison of social welfare of the equilibria

(c) Comparison of social fairness of the equilibria

Fig. 4. Properties of the equilibria.



(a) PoA, PoS and PoD of the game



(b) Load distribution of the different approaches

Fig. 5. Quality of the nash equilibria of the game with respect to the social optimum solution.

5.4 Analysis on the Quality of the Equilibria

In this section, we compare the Nash equilibria of the market game Γ with the optimal solution that can be computed by solving a centralized optimization problem. We consider as objective function of the optimization problem the following social welfare function

$$Welf(\mathbf{x}) = \sum_{s\in\mathcal{S}} \log\left(u_s(\mathbf{x})\right), \tag{11}$$

where the sum of the log of the utility functions has the purpose to measure the fairness of the resource allocation, similarly to what is done for the well known proportional fair scheduling [42]. We claim that this reflects the objective of the InP, which is interested in providing a fair service to all the tenants and avoid situations where only some of them gets most of the resources.

Hence, we analyze the Price of Anarchy (PoA) and the Price of Stability (PoS) of the game [43] by computing the ratio between the optimal value which maximizes the objective function in Eq. (11) and the values at the best/worst NE (i.e., the NE that maximizes/minimizes the social welfare function, respectively). Note that, differently from the results of the previous section, the "best NE" (to compute the PoS) and the "worst NE" (to compute the PoA), as well as the optimal solution of Eq. (11), are found by solving the corresponding optimization problems as Mixed Integer Linear Programming (MILP) problems with the Gurobi solver [44][5]. To evaluate the quality of the NE obtained with the proposed DSSBR, we introduce a novel metric, referred to as *Price of DSSBR* (PoD), defined as

$$\text{PoD} = \frac{\max_{\mathbf{x}\in\mathbf{X}} Welf(\mathbf{x})}{Welf(\mathbf{x}_{\text{DSSBR}})}, \tag{12}$$

where we denote by $\mathbf{x}_{\text{DSSBR}}$ the solution at the NE achieved by DSSBR.

In Fig. 5, we compare the quality of the NE with respect to the social optimum by collecting the results over 1000 random instances, with random slice settings to analyze the quality of the equilibria under different slicing setups and network load conditions. In Fig. 5a, we show the values of PoA, PoS and PoD of the game. By comparing the values of PoS and PoD, we see that the NE achieved by DSSBR performs very close to the "best NE", therefore resulting in the most fair equilibria of the game for almost all the simulated instances. This result shows the benefits of the proposed algorithm, whose implementation favours the fairness among the players in the marketplace. Moreover, if we look at the distribution of the total load of the system in Fig. 5b, we see that the solutions at both the "best NE" and the one from DSSBR result in a higher resource utilization with respect to the social optimum solution. Hence, by triggering the competition in the marketplace, the InP can increase the utilization of network resources and, therefore, also its own profits.

5. Note that this approach is not practical in real systems due to the complexity of the optimization problems involved. It is therefore only used in this section for the specific analysis on the quality of the equilibria.

TABLE 1
Slice Parameters Setting

Tenant	cIoT	eMBB Pr.	eMBB Bs.
μ_s	8	4	2
$\lvert\mathcal{K}_s\rvert$	280	63	105
r_s^0	0.5 Mbps	4 Mbps	2 Mbps
q_s	0.001	0.001	0.001
χ_s	3	8	3

6 Numerical Results

In this section, our simulation setup consists of three different slices, $\mathcal{S}$ = {critical Internet of Things (cIoT), ehnanced Mobile BroadBand Premium (eMBB Pr.), enhanced Mobile BroadBand Basic (eMBB Bs.)}, which span different service characteristics and user behaviors, i.e., ranging from critical applications with low-rate requirements but high QoS guarantees (cIoT) to elastic traffic with medium-high rate requirements but adaptive QoS (eMBB slices), that we model by means of different values of the tuple (μ_s, r_s^0, q_s). In our experiments, the users of each slice are uniformly distributed throughout the coverage area of 21 cells, and their random position determines the diverse traffic demand on each cell. In this work, we consider full buffer users. We further consider, without loss of generality, that all the users within a slice have the same QoS requirements, which are defined in terms of throughput, r_s^0, and the same economic value, i.e., $\chi_k = \chi_s, \forall k \in \mathcal{K}_s$. In Table 1, we summarize the list of parameters that describe the slice settings. The experiments are performed by means of a downlink system level simulator which is 3GPP-calibrated [45] and abstracts the physical-layer effects through a link-to-system level interface. The interface applies an equivalent Signal-to-Interference-and-Noise Ratio (SINR), computed given the cell topology, the active user transmissions, and a vertically polarized antenna configuration. The radio environment and other relevant simulation parameters are taken from [11] and listed in Table 2. We assume that the strategy step size of each slice is equal to a minimum allocation of $\Delta x = 15$ kHz, that is the subcarrier spacing. We run our experiments over 50 randomly deployed instances and collect the results by aggregating and averaging them from all the simulated scenarios.

At each execution of the algorithm, we collect information from the MAC scheduler, namely the spectral efficiency for each user, η_k, $\forall k \in \mathcal{K}$. We take advantage of the

TABLE 2
Simulation Parameters

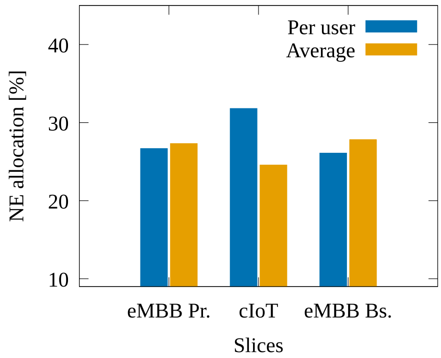
General Environment	3GPP 3D UMa Scenario [47]
Number of 120° Cells	21, Wraparound interference
Simulation Time / Drops	12 s
Number of Simulation Drops	50
CQI Model	Pilot-based, sub-band reports every 5 ms
Subcarrier Spacing	15 kHz
Bandwidth	10 MHz $\longrightarrow$ 48 PRBs
TTI	1 ms
gNB/UEs Antennas	2 Vertically polarized
User Mobility	3 km/h
Traffic	Full Buffer

knowledge of the spectral efficiency to estimate the achievable throughput of a user, according to the relationship $r_k(x_k) = x_k \cdot \eta_k$. We consider also the case in which per-user information is not available from the MAC scheduler. In this case, we rely on an average per-slice information, i.e., we assume to have access to the average spectral efficiency of users belonging to each slice, $\eta_k = \bar{\eta}_s, \quad \forall k \in \mathcal{K}_s$. Accordingly, the per-user achievable rate can be rewritten as $r_k(x_k) = x_k \cdot \bar{\eta}_s$. We estimate that each user, in average, will get a proportional share of the slice resources from the SMS scheduler, i.e., $x_k \simeq \frac{x_s}{|\mathcal{K}_s|}, \forall k \in \mathcal{K}_s$[6]. However, given that users do not experience improvement in their QoS when the reference throughput value, r_s^0, is already achieved, we limit the estimated resource share of a user to $x_k \leq \frac{r_s^0}{\eta_k}$, and redistribute the unused resources to the remaining users. Then, we can estimate the per-user acceptance values, $A_k(r_k(x_k))$. Notice that, with the per-user spectral efficiency we can estimate the achievable rate of each user more accurately than when using the per-slice average spectral efficiency, which may not necessarily reflect each user performance. For each cell, the NSN solves the game described in the previous sections, and forwards the achieved NE as new resource constraint to be enforced at the SMS scheduler. At each simulation instance, we periodically collect the information coming from the MAC scheduler of each cell every 6 seconds, after which we apply the DSSBR algorithm described in Section 5. Note that, in a realistic scenario, the algorithm could be performed only when a renegotiation is actually needed, e.g., triggered by predefined thresholds from the radio scheduler. At the end of the simulations, the achieved performances are collected, both in terms of actual per-user acceptance values and overall profits for the slices.

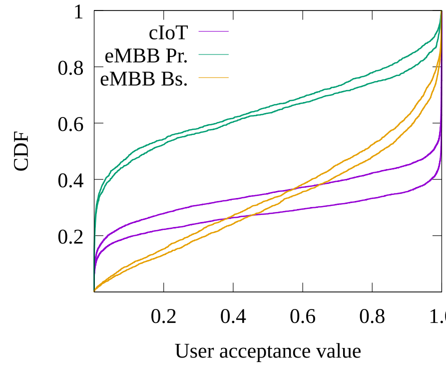
In what follows, we first characterize the features of our approach. We analyze the impact on the performance of the spectral efficiency information collected from the MAC scheduler and combine our solution with overload control policy to handle critical congested scenarios. Then, we compare the performance obtained by the NSN with a baseline solution, which assigns a static resource share for each slice.

6.1 Users and Slice Performance Evaluation

In this section, we analyze the performance of the system under the two use cases that we have mentioned before, i.e., considering i) per-user spectral efficiency, when we can



(a) NE allocation



(b) User performance: Per-user (solid), Average (dotted)

Fig. 6. Performance for different revenue functions.

estimate the spectral efficiency of each users, $\eta_k, \forall k \in \mathcal{K}$ and ii) per-slice average spectral efficiency over each cell, i.e., when only the average spectral efficiency of users within one slice can be estimated, $\eta_k = \overline{\eta}_s, \forall k \in \mathcal{K}_s$.

In Figs. 6a, 6b, we analyze the equilibria achieved by the slices and the CDF of user acceptance values, respectively, for the two cases, i.e., per-user η_k and average $\bar{\eta}_s$. We show that, due to the specific settings of our simulation scenario, the cIoT slice in the per-user configuration is the most demanding in terms of resources, as shown in Fig. 6a. This result comes from the lower flexibility of these users in QoS degradation, given by their higher μ_s value. However, for the cIoT slice, the per-slice average spectral efficiency case results in a decreasing demand of resources, which in turn determines a deterioration of performance for the users, as shown in Fig. 6b. This behavior can be explained by the quasi-step behavior of the acceptance function, that, in the per-user case, encourages to serve also users experiencing bad channel conditions, where more resources are needed. On the other side, we can see an opposite behavior for the remaining slices, which increase their resource demands. This can be explained by recalling the aggregative property of the game. As shown in Lemma 1, an increase/decrease in any player strategy determines an opposite reaction by at least one other player in the game[7]. This strategic behavior is then reflected in the per-user acceptance values in Fig. 6b. We see that the performance increases due to a slice resource share increment, and viceversa. Hereafter, we

6. This assumption holds true for proportional fair scheduling [46].

7. This property is usually referred as *comparative statics* in aggregative games [40].

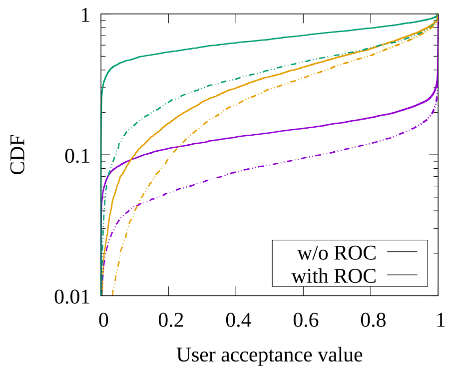
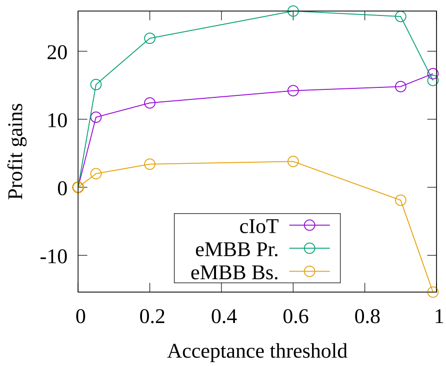
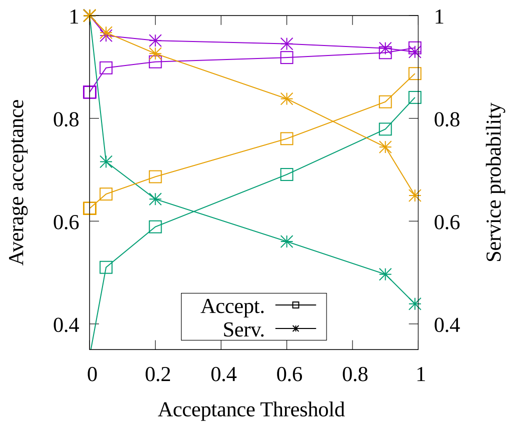


Fig. 7. Comparing user performance at NE with $A_s^{\min} = 0.2$ and without ROC policy implementation.



(a) Slice profits at NE



(b) Slice performances at NE

Fig. 8. Performance comparison with ROC implementation.

adopt the per-user revenue functions, which we expect to better fit the requirements of both tenants and end users.

6.2 Per-Slice Overload Control Policy

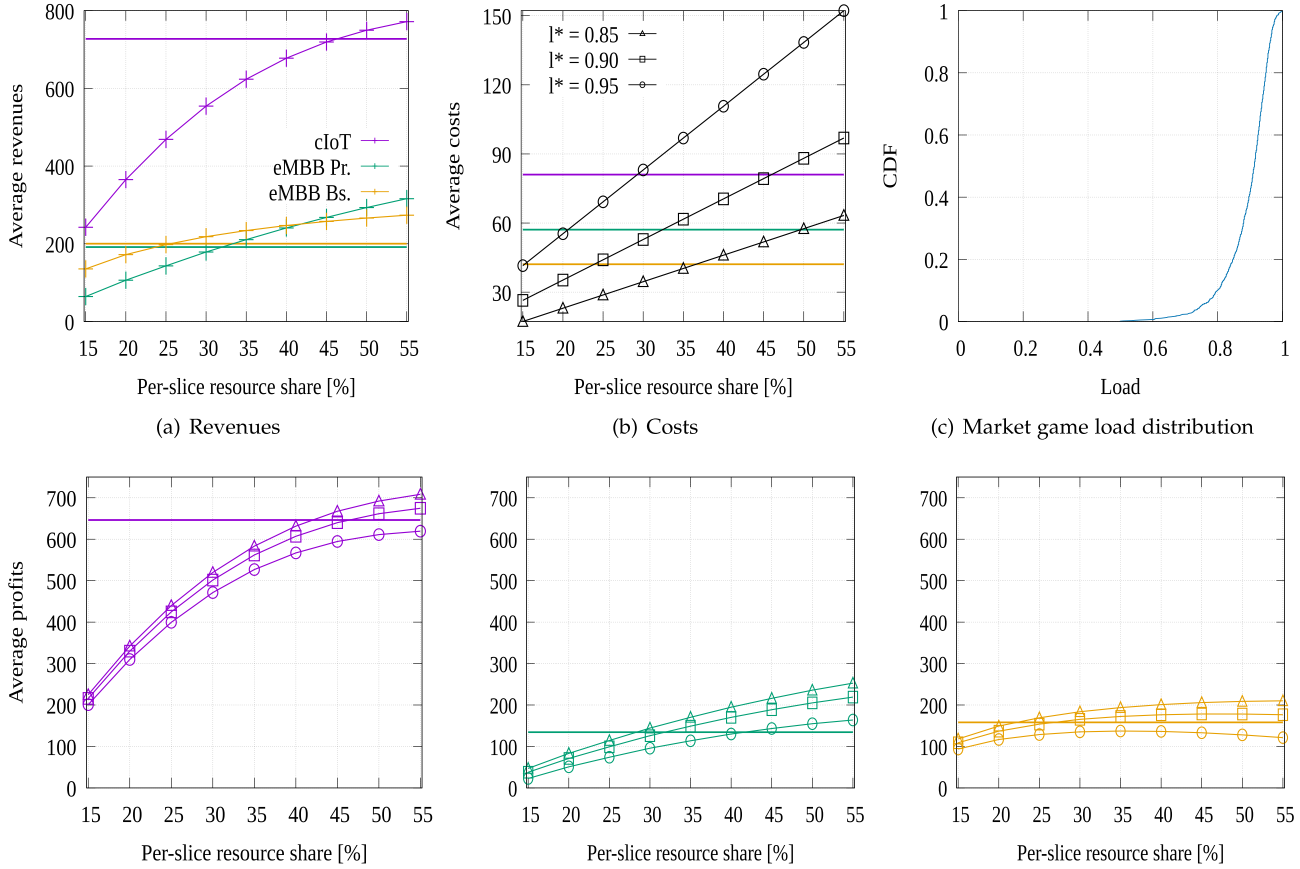
When looking at the performance of the slices in terms of their user acceptance values in Fig. 6b, one can notice that, at the NE, not all the users can be served with the expected QoS, leading to possible waste of resources that could be better managed by the radio scheduler. This mainly happens due to the particular conditions that we reproduce in our simulations. Indeed, in order to make appreciable the interactions among the tenants in the market and the outcome of the game, we force the system to work with very high demand, with a load that approaches the resource saturation. Our approach suggests then, even in case of dynamic renegotiation of resource shares, it is still necessary to rely on external tools that may control the incoming traffic in the network. Therefore, we implement a basic radio overload control (ROC) policy, that drops user connections whenever a user will not be able to meet a minimum QoS target. We define the ROC threshold in terms of minimum acceptance value, by imposing

$$\begin{cases} \text{if } A_k(r_k(x_k)) \geq A_s^{\min} & \text{keep user } k, \\ \text{if } A_k(r_k(x_k)) < A_s^{\min} & \text{drop user } k, \end{cases}$$

where the acceptance threshold, $A_s^{\min}$, can be independently tuned and defined for each slice. The graph in Fig. 7 shows that, as expected, by removing from the system users that were not able to achieve their QoS (e.g., due to critical channel conditions), it is possible to improve the performance of the remaining users. One can also notice that a few percentage of users is still not able to meet his QoS requirements. Although the difficulty of fulfilling the QoS requirements in case of sudden degradation of channel conditions for some users, our approach is still able to control the overall QoS of the slices. It also turns out that the overall profits of the slices increase. In Fig. 8, we show the impact of different ROC threshold values on the overall performance of slices and their users. We measure the gain in terms of difference in profits of the tested acceptance threshold values with respect to the case when no overload control is implemented, that corresponds to the acceptance threshold value equal to 0. By varying from low to high acceptance threshold values (up to the threshold $r_k(x_k) = r_s^0$), we recognize the different behavior of the slices, according to their service type. Indeed, the most critical slices exhibit a clear gain in profits, due to the steeper behavior of the acceptance of their users, while the eMBB Bs. slice, that is more prone to QoS degradation, does not show evident gains (rather the contrary) by rejecting users from the system. In Fig. 8b, we also show the trade-off between the improvement in performance of served users and their probability to be served. One can notice that increasing the acceptance threshold value the ratio of the number of served users decreases, but the performance of users increases. However, aggressive ROC policies with very high threshold values can have a negative impact on some slices, since too many users are being removed from the system (indeed the profits in Fig. 8a may decrease). Although the absolute values in Fig. 8 are just qualitative and strictly depend on the incoming traffic in the network, the general trade-off between threshold value and improvement in performance still holds for any load condition. Hereafter, we consider the acceptance threshold value to be equal to $A_s^{\min} = 0.2$ for each slice.

6.3 Dynamic Game Versus Static Allocations

In this section, we compare the results achieved by NSN with a baseline approach, where the resource share for each slice is not automatically renegotiated, but statically assigned by manual input and kept constant in each cell at each simulation instance. In order to compare the two approaches, we consider the same utility functions (i.e., same definition of revenues and costs) and same ROC



(a) Revenues (b) Costs (c) Market game load distribution

(d) cIoT profits (e) eMBB Pr. profits (f) eMBB Bs. profits

Fig. 9. Slice performance comparison between the dynamic market game and static allocation baseline. Different colors are used for different slices (purple for cIoT, green for eMBB Premium and orange for eMBB Basic). Dash-dotted lines represent the results from the market game and solid lines from the static allocations. Different markers are used for different static load combinations.

policy discussed previously. In Fig. 9 we present the slice performance in terms of revenues, costs and total profits for both the dynamic renegotiation and the static allocation approaches. We show the results of the static baseline in solid lines when varying the static resources share of the individual slices. Instead the dash-dotted lines represent the outcome of the game, which is shown as constant lines since it does not depend on the static resource share of the slices. As discussed in the previous sections of the papers, the utility function, i.e., the profit, of each tenant can be decomposed into two components: i) revenue function, $R_s(x_s)$, which only depends on the resources that are allocated to slice s, and ii) cost function, $C_s(x_s, l)$, which depends on the resources allocated to slice s and on the total amount of resources allocated in the network l, where $l = \sum_{s \in \mathcal{S}} x_s$. In particular, in order to compute the revenue function of a tenant, we just need to know the amount of resources allocated to its own slice, i.e., x_s, independently of l, i.e., the resources allocated to the remaining slices x_{-s}. This allows us to plot the revenue functions of the players in a single graph, as in Fig. 9a, where we see that to an increase in the allocated resources for a slice always corresponds an increase in its revenues. We also show that, due to the law of diminishing return, the increase in revenues is diminishing when enough resources are allocated to the slices. Conversely, in order to evaluate the cost function for a tenant we need to know i) the resources allocated to that tenant, i.e., x_s, and ii) the total amount of resources allocated in the network, i.e., l, but not the individual allocations of the other slices. Notice that, while in the market game the total load l, and therefore the costs, depends on the achieved NE, thus automatically adapted over time and in each cell, in the static baseline we need to fix one parameter per time, i.e., x_s or l, to compute their costs. Hereafter, we consider three different load values, i.e., $l^* = [0.85, 0.9, 0.95]$. For a given fixed total load, the purchasing power of the tenants is constant (and so the cost of a unit of resource) and, therefore, the costs for a tenant increase linearly with the amount of obtained resources (cf. Eq. (4)). This trend is confirmed in Fig. 9b, where the curves for the static baseline are colored in black since the cost is uniquely defined for each slice and no pricing differentiation is assumed, with different markers to represent the different fixed load values l^*. In contrast, the (average) costs for the dynamic renegotiation are shown in dash-dotted lines with the same color code as Fig. 9a. We intentionally leave room for free resources for the static baseline in order to reproduce similar load conditions obtained in the proposed market game, which are shown in Fig. 9c, where the total experienced load is greater than 0.8 for almost 90 percent of the simulated scenarios. Notice that the remaining resources (both in the dynamic and in the static case) are allocated to best-effort users, which do not take part to the renegotiation, but always get the

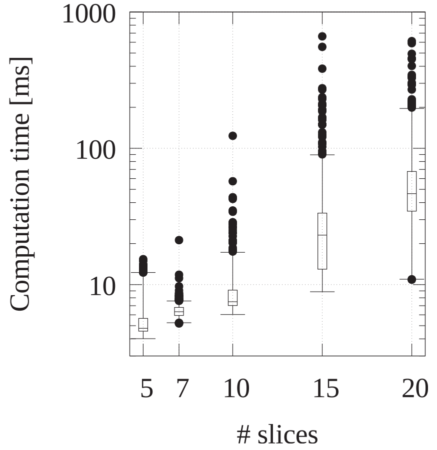


Fig. 10. Time convergence of DSSBR with different tenant set size.

unused resources. By combining the curves obtained in Figs. 9a, 9b, we obtain the overall profits of the three slices, that are shown in Figs. 9d, 9e, 9f, respectively. The solid curves with different markers show the profits achieved by the tenants when imposing a static resource share and assuming a given total network load (e.g., looking at the curve with the square markers in Fig. 8d, we could say that the cIoT slice gets a profit of 500 when its own resource share is 30 percent and the other two slices get 60 percent since we assume a 0.9 total load), with the same notation as in Fig. 9b.

It is trivial that, by looking at the graphs in Figs. 9d, 9e, 9f, the profits of the tenants are higher when purchasing the same amount of resources in less loaded scenarios, due to the lower costs.

If we consider the performance in high congested scenarios, namely $l^* = 0.95$, we can notice that the tenants of the cIoT and eMBB Bs. slice do not achieve higher profits than the ones obtained with the proposed market game, which suggests that any combination of static allocations summing up to $l^* = 0.95$ cannot improve our proposed solution. However, for lower total load values, although there are allocations that allow the tenants to individually improve their profits, one can varify that it does not exist an admissible static allocation for which every tenant can improve his performance with respect to the dynamic allocation strategy implemented in the market game. For example, let us consider the smallest static allocation value, for a fixed load $l^* = 0.85$, at which each slice gets higher profits than our proposal, or very close ones, i.e., when the triangle markers are above or close to the dash-dotted lines of Figs. 9d, 9e, 9f. This corresponds approximately to the allocation [40,30,25] percent for cIoT, eMBB Pr. and eMBB Bs. slice, respectively, which is not an admissible configuration given that the total load sums up to 0.95, violating the assumption of a total load equal to 0.85.

This example generalizes well for all the other possible combinations that one can observe in Figs. 9d, 9e, and 9f. Indeed, the improvement in performance for one slice in the fixed static allocation baseline can be achieved only at the cost of degrading the performance of at least one of the other slices. Conversely, by playing their Nash Equilibria strategies, we show that the dynamic allocations obtained in our market game result in higher profits for all slices.

TABLE 3
Algorithm Performance for Different Tenant Set Size

# Slices	Average # iterations
5	32
7	39
10	55
15	152
20	214

6.4 Algorithm complexity

Finally, we test the algorithm complexity of our solution and analyze the impact of the tenant set size on the computational complexity of the negotiation framework. We recall that, in the implementation of DSSBR, we consider a first step which provides a fair initialization of the strategies of the players and a second step that implements the classic BRD with the achieved strategy initialization. While the complexity of the first step can be considered deterministic, i.e., it can be controlled by the system designer since it serves only for the purpose of providing a fair initialization, the second step is not deterministic, since the convergence time may vary according to the settings of the game. From [48], we can derive the overall computational complexity for BRD in aggregative games for sufficient large number of players. In particular, the settings of our market game Γ can be translated in the conditions of [48, Theorem 1] which shows that the complexity of finding a NE with BRD scales with the number of players as $\mathcal{O}(NlogN)$, for big values of N, where $N = |\mathcal{S}|$, i.e., the number of players in the game. This gives us an upper bound for the implementation of the second step of the proposed DSSBR for high number of players. In Fig. 10, we show the convergence time of DSSBR when considering an increasing number of slices simultaneously active in the cells and competing for the resources in the market. The boxplot shows the median value, indicated by the horizontal bar in the box, and the 25th and 75th percentile as the extreme of the rectangular boxes, while the whiskers represent the 5th and 95th percentile, and the remaining dots the outliers. We can see that, although the median computation time in Fig. 10 increases exponentially (linear behavior on a logscale plot) with the number of tenants, in the 95 percent of the cases with 20 tenants we do not need more than 200 ms per cell to converge to the NE and we do not envision to run this algorithm more often than every few seconds. This result makes our approach in line with the horizon time that we have sketched in Fig. 1, where the resource renegotiation can be triggered on a time granularity in the order of seconds to minutes. Consider that such performance can be further improved if implementing such algorithm in an edge cloud platform. In Table 3, we also show the average number of iterations needed for the convergence to a NE, for different tenant set sizes. In this case, a single iteration refers to the number of executions of the best responses of the players, where in the first step of DSSBR we assume that one iteration is the simultaneous movement of all the players, while in the second step of DSSBR one iteration is the single movement of each player. The values in Table 3 clearly show that the average number of iterations do not increase exponentially in the number of players, but rather confirm the theoretical expectations about the computational complexity of the game.

7 Conclusion

We have proposed a Slicing Management Framework, that automatizes the interaction of tenants as in a marketplace and assigns the available radio resources to each slice. We have modeled the competition among the tenants through game theory and analyzed the properties of the game. We have also developed a heuristic approach to guarantee the convergence to a single NE, which provides good results in terms of fairness and social welfare in the tenants' allocation.

Our results have been validated through extensive experiments in a 3GPP-compliant system level simulator. We have observed that the experienced performance matches the different SLAs required by each of the three simulated tenants and adapt to different cell loads and users' presence. However, for high overloaded cells we have shown that admission and overload control have to be properly implemented. By integrating our renegotiation mechanism with basic overload control policies, we have achieved an increase in the performance of the slices, especially in terms of overall profits. We have compared the proposed renegotiation mechanism with a static baseline, which assigns a pre-agreed fixed resource share on each cell. We have shown that, for any feasible static allocation, our dynamic approach provides performance gains for all the slices. Finally, we have analyzed the computational complexity of the proposed solution, considering very populated scenarios, with up to 20 slices simultaneously bidding for resources on a cell. Empirical results have shown that the convergence time of our algorithmic implementation justifies the application of such negotiation mechanism in a very dynamic environment, like radio access networks, where fast adaptation of the resources must be guaranteed in the order of seconds or minutes.

References

[1] GSA, "5G network slicing for vertical industries," 2017. [Online]. Available: https://gsacom.com/paper/5g-network-slicing-vertical-industries/

[2] X. Zhou, R. Li, T. Chen, and H. Zhang, "Network slicing as a service: Enabling enterprises' own software-defined cellular networks," *IEEE Commun. Mag.*, vol. 54, no. 7, pp. 146–153, Jul. 2016.

[3] NGMN, "5G white paper," 2015. [Online]. Available: http://www.ngmn.de/5gwhite-paper.html

[4] S. E. Elayoubi, S. B. Jemaa, Z. Altman, and A. Galindo-Serrano, "5G RAN slicing for verticals: Enablers and challenges," *IEEE Commun. Magazine*, vol. 57, no. 1, pp. 28–34, Jan. 2019.

[5] D. Bega, M. Gramaglia, A. Banchs, V. Sciancalepore, K. Samdanis, and X. Costa-Perez, "Optimising 5G infrastructure markets: The business of network slicing," in *Proc. IEEE Conf. Comput. Commun.*, 2017, pp. 1–9.

[6] Nokia Bell Labs, "Unleashing the economic potential of network slicing," 2018. [Online]. Available: https://onestore.nokia.com/asset/202089

[7] C. Marquez, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, "How should I slice my network?: A multi-service empirical evaluation of resource sharing efficiency," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, 2018, pp. 191–206.

[8] 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, Jun. 2017.

[9] Amazon AWS, "Amazon elastic compute cloud documentation," 2020. Accessed: Jul. 2020. [Online]. Available: https://docs.aws.amazon.com/ec2/index.html

[10] A. Lieto, E. Moro, I. Malanchini, S. Mandelli, and A. Capone, "Strategies for network slicing negotiation in a dynamic resource market," in *Proc. IEEE 20th Int. Symp. World Wireless Mobile Multimedia Netw."*, 2019, pp. 1–9.

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

[12] S. Vassilaras *et al.*, "The algorithmic aspects of network slicing," *IEEE Commun. Magazine*, vol. 55, no. 8, pp. 112–119, Aug. 2017.

[13] 3GPP TS 23.251, V11.5.0, "Network sharing: Architecture and functional description," Mar. 2013.

[14] I. Malanchini, S. Valentin, and O. Aydin, "Wireless resource sharing for multiple operators: Generalization, fairness, and the value of prediction," *Comput. Netw.*, vol. 100, pp. 110–123, 2016.

[15] E. A. Jorswieck, L. Badia, T. Fahldieck, E. Karipidis, and J. Luo, "Spectrum sharing improves the network efficiency for cellular operators," *IEEE Commun. Mag.*, vol. 52, no. 3, pp. 129–136, Mar. 2014.

[16] J. M. Peha, "Sharing spectrum through spectrum policy reform and cognitive radio," *Proc. IEEE*, vol. 97, no. 4, pp. 708–719, Apr. 2009.

[17] Y. L. Lee, J. Loo, T. C. Chuah, and L. Wang, "Dynamic network slicing for multitenant heterogeneous cloud radio access networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2146–2161, Apr. 2018.

[18] M. Jiang, M. Condoluci, and T. Mahmoodi, "Network slicing in 5G: An auction-based model," in *Proc. IEEE Int. Conf. Commun.*, 2017, pp. 1–6.

[19] B. Khodapanah, A. Awada, I. Viering, A. N. Barreto, M. Simsek, and G. Fettweis, "Slice management in radio access network via iterative adaptation," in *Proc. IEEE Int. Conf. Commun.*, 2019, pp. 1–7.

[20] V. Sciancalepore, K. Samdanis, X. Costa-Perez, D. Bega, M. Gramaglia, and A. Banchs, "Mobile traffic forecasting for maximizing 5G network slicing resource utilization," in *Proc. IEEE IEEE Conf. Comput. Commun.*, 2017, pp. 1–9.

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

[22] M. Leconte, G. S. Paschos, P. Mertikopoulos, and U. C. Kozat, "A resource allocation framework for network slicing," in *Proc. IEEE Conf. Comput. Commun.*, 2018, pp. 2177–2185.

[23] P. Caballero, A. Banchs, G. de Veciana, and X. Costa-Pérez, "Network slicing games: Enabling customization in multi-tenant networks," in *Proc. IEEE Conf. Comput. Commun.*, 2017, pp. 1–9.

[24] L. Guijarro, J. Vidal, and V. Pla, "Competition in service provision between slice operators in 5G networks," *Electronics*, vol. 7, 2018, Art. no. 315.

[25] P. Caballero, A. Banchs, G. de Veciana, X. Costa-Pérez, and A. Azcorra, "Network slicing for guaranteed rate services: Admission control and resource allocation games," *IEEE Trans. Wireless Commun.*, vol. 17, no. 10, pp. 6419–6432, Oct. 2018.

[26] W. Xie, J. Zhu, C. Huang, M. Luo, and W. Chou, "Network virtualization with dynamic resource pooling and trading mechanism," in *Proc. IEEE Global Commun. Conf.*, 2014, pp. 1829–1835.

[27] O. U. Akguel, I. Malanchini, V. Suryaprakash, and A. Capone, "Service-aware network slice trading in a shared multi-tenant infrastructure," in *Proc. IEEE Global Commun. Conf.*, 2017, pp. 1–7.

[28] B. Edelman, M. Ostrovsky, and M. Schwarz, "Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords," *The Amer. Econ. Rev.*, vol. vol. 97(1), pp. 242–259, 2007.

[29] GoogleAds, "Automated bid strategy: Definition," Last visit: December, 2019. [Online]. Available: https://support.google.com/google-ads/answer/6325042?hl=en

[30] P. Samadi, H. Mohsenian-Rad, R. Schober, and V. W. Wong, "Advanced demand side management for the future smart grid using mechanism design," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1170–1180, Sep. 2012.

[31] Y. Kim, S. Kim, and H. Lim, "Reinforcement learning based resource management for network slicing," *Appl. Sci.*, vol. 23, 2019, Art. no. 2361.

[32] 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.

[33] 3GPP TS 28.531, V15.4.0, "Management and orchestration; Provisioning," Sept. 2019.

[34] B. Khodapanah, A. Awada, I. Viering, D. Oehmann, M. Simsek, and G. P. Fettweis, "Fulfillment of service level agreements via slice-aware radio resource management in 5G networks," in *Proc. IEEE 87th Veh. Technol. Conf.*, 2018, pp. 1–6.

[35] M. Jensen, "Aggregative games and best-reply potentials," *Econ. Theory*, vol. 43, pp. 45–66, 2010.

[36] L. Badia, M. Lindstrom, J. Zander, and M. Zorzi, "Demand and pricing effects on the radio resource allocation of multimedia communication systems," in *Proc. IEEE Global Telecommun. Conf.*, 2003, pp. 4116–4121.

[37] M. K. Jensen, Aggregative games and best-reply potentials, *Econ. Theory*, vol. 43.1, pp. 45–66, 2010.

[38] M. Voorneveld, "Best-response potential games," *Econ. Lett.*, vol. 66, pp. 289–295, 2000.

[39] L. Corchón, "Comparative statics for aggregative games. The strong concavity case," *Math. Soc. Sci.*, vol. 28, pp. 151–165., 1994.

[40] P. Dubey, O. Haimanko, and A. Zapechelnyuk, "Strategic complements and substitutes, and potential games," *Games Econ. Behav.*, vol. 54, pp. 77–94, 2006.

[41] M. Kaneko and K. Nakamura, "The Nash social welfare function," *Econometrica*, vol. 47, pp. 423–435, 1979.

[42] F. P. Kelly, A. K. Maulloo, and D. K. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *J. Oper. Res. Soc.*, vol. 49, no. 3, pp. 237–252, 1998.

[43] E. Koutsoupias and C. Papadimitriou, "Worst-case equilibria," *Comput. Sci. Rev.*, vol. 3, pp. 65–69, 2009.

[44] Gurobi, in *Gurobi Solver*. Accessed: Jun. 2020. [Online]. Available: http://www.gurobi.com

[45] 3GPP TR 38.802, V14.1.0, "Study on new radio access technology. Physical layer aspects." Jun. 2017.

[46] M. Andrews, S. Borst, S. Klein, H. Kroener, and S. Mandelli, "The effect of additive and multiplicative scheduler weight adjustments on 5G slicing dynamics," in *Proc. IEEE 2nd 5G World Forum*, 2019, pp. 34–39.

[47] 3GPP TR 36.873, V12.6.0, "Study on 3D channel model for LTE," Sept. 2017.

[48] Y. Babichenko, "Best-reply dynamics in large binary-choice anonymous games," *Games Econ. Behav.*, vol. 81, pp. 130–144, 2013.

Alessandro Lieto (Member, IEEE) received the BS and MS degree in telecommunication engineering from Università degli Studi di Napoli Federico II, in 2014, and Politenico di Milano, in 2017, respectively. He is currently working toward the PhD degree in information technology from Politecnico di Milano, where he joined the Advanced Network Technologies Laboratory (ANTLab). In 2018, he has spent time as a visiting PhD student at Nokia Bell Labs Stuttgart, working for the End-to-End Network Service Automation (ENSA) Laboratory. His current research interests include on resource allocation and algorithmic game theory applied to wireless networks.

Ilaria Malanchini (Member, IEEE) received the BS and MS degrees in telecommunications engineering from Politecnico di Milano, Italy, in 2005 and 2007, respectively, and the PhD degree in electrical engineering from Drexel University, Philadelphia, and Politecnico di Milano, in 2011. She is currently a senior research engineer at the E2E Network and Service Automation Lab and has been with Bell Labs Stuttgart since 2012. She was awarded the Meucci-Marconi Award and the Chorafas Foundation Prize for her Master and PhD thesis, respectively. She published more than 35 peer reviewed journal and conference papers and has more than ten granted or filed patents. Her research interests include optimization models, mathematical programming, game theory, and machine learning, with the application of these techniques to wireless network problems, such as wireless resource allocation, anticipatory network optimization, infrastructure and resource sharing, and network slicing.

Silvio Mandelli (Member, IEEE) received the BS and MS degrees in telecommunication engineering from the Politecnico di Milano, Italy, in 2010 and 2012, respectively, and the PhD degree in information technology from the Politecnico di Milano, in 2012. He is currently a research engineer with the Access and Device Lab at Nokia Bell Labs Stuttgart since 2016. He published more than 20 peer reviewed journal and conference papers and has more than 15 granted or filed patents. His research interests include on signal processing, machine and deep learning, statistics, positioning and tracking techniques, resource scheduling and game theory, with the application of these techniques to wireless and optical network, such as 5G-based localization, radio resource scheduling, network slicing, and optical phase noise compensation.

Eugenio Moro (Member, IEEE) received the bachelor's degree in information engineering from the Università del Salento, in 2016, and the MSc degree of telecommunications engineering from Politecnico di Milano, in 2019. He is currently working toward the PhD degree from the Department of Electronics, Information and Bioengineering, at Politecnico di Milano. His research interests include is telecommunication, with a focus on optimization techniques and game theory applied to wireless networks.

Antonio Capone (Fellow, IEEE) is currently a full professor with the Politecnico di Milano (Technical University of Milan), where he is also the director of the ANTLab. His expertise is on networking and his main research activities include radio resource management in wireless networks, traffic management in software defined networks, network planning, and optimization. He has more than 250 publications on these topics. He is an editor of the *IEEE Transactions on Mobile Computing*, *Computer Networks*, and *Computer Communications*. He contributes to major conferences on networking as a Technical Program Comittee member. He was an editor of the *ACM/IEEE Transactions on Networking* from 2010 to 2014 as well.