Toward Radio Access Network Slicing Enforcement in Multi-Cell 5G System

Imane Oussakel, Philippe Owezarski, Pascal Berthou*, Laurent Houssin*
LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France
**LAAS-CNRS, Université de Toulouse, UPS, Toulouse, France*
{imane.oussakel, philippe.owezarski, pascal.berthou, laurent.houssin}@laas.fr

***Abstract*—The proliferation of sophisticated applications and services comes with diverse performance requirements. The 5G cellular network is advocated to support this diversity through an end-to-end network slicing. Even though the slicing is not a novel concept, its implementation in the RAN still remains challenging. In this article, we aim to enforce the real time 5G slicing from radio resources perspective in a multi-cell system. For that, two exact optimization models are proposed. Due to their high convergence time, heuristics are developed and evaluated with the optimal models. Results are promising, as two heuristics are highly enforcing the real time RAN slicing.**

***Index Terms*—Radio Access Network (RAN), RAN Slicing, 5G, Optimization, Resource allocation.**

I. Introduction

The tremendous growth of services and/or applications demand is increasing over the years with diverse QoS (Quality of Service) requirements. 3GPP and other organizations aim to support this variety of services requirements through 5G system with a service-based architecture.

Network slicing is considered as one of the pillars to enable such architecture where each Mobile Network Operator (MNO) shares its physical infrastructure with several tenants as slices, such as automotive and health-care industries. Thus, each slice is considered as a business service with certain QoS requirements. The fulfillment of the envisioned network slicing approach involves high flexibility and programmability of 5G network. To that end, virtualization and softwarization based solutions have been nominated, mainly NFV (Network Function Virtualization) and SDN (Software Defined Network) based solutions. The former allows flexibility of NFs via virtualization, and the latter separates the control from the user data functions with a centralized controller. Several prototypes based on NFV and SDN have been proposed to address the Core (CN) [16] and Radio Access Networks (RAN) [7] slicing. The SDN implementation at the RAN part is referred by SD-RAN (Software Defined RAN).

Nevertheless, the enforcement of RAN slicing still attracts the academy and industries researchers attention, as maintaining slices isolation with efficient use of radio resources is a challenging task. In fact, the scarcity of radio resources turns infeasible the resources over-provisioning already used in the CN. Hence, the wireless resource allocation needs to meet the service requirements for each slice regardless the channel conditions or network congestion, while efficiently using the scarce available resources. Moreover, the 5G system framework is subdivided into three layers, infrastructure, network and service layers. Isolation must be sustained over the different system layers. Particularly, the outage performance of one slice, i.e. congestion, attack or QoS degradation, should not impact negatively the other available slices in the network.

Further, it is based on the slice performance requirements, the traffic demand and the channel/network conditions, the amount of the slice required resources should be decided. This is known by resource slicing policy. The implementation of such policy (i.e. resources allocation) has to respect the RAN slicing requirements formulated and summarized as follows:

- Orthogonality (resource isolation): it must be guaranteed between slices. Each radio resource, in terms of time and frequency, must be allocated to only one slice to avoid inter-slices interference, thus, ensuring the slice isolation at the radio resources level.
- Satisfaction: each slice has to be allocated the amount of assigned resources based on the slicing policy, i.e. for a given slice that has been assigned 25 radio resources, it should receive approximately the amount of 25 radio resources, without excess. This way the slice demand is satisfied and each slice fully uses its resources.
- Scalability: the MNO should be able to scale up/down the slice allocated resources with respect to the network conditions and slice demand variation. Moreover, as the slices are created dynamically and on-demand, the radio resource model should allow the MNO to serve new slices requests. This can be achieved through the reuse of the unallocated resources during the allocation window.
- Cooperation enabling: the 5G advanced radio techniques such as IBSPC (inter-base station power control) and CoMP (coordinated multi-point) involve a tight cooperation between the base stations (i.e. gNBs in case of 5G) to achieve their objective [13]. As the RAN slicing imposes the slices resources orthogonality, the activation of the appropriate technology is based on the slice performance requirements and SLA. The slices radio resources allocation should therefore ease the deployment of these advanced technologies for each tenant.

The achievement of these requirements is considered as a

This work is funded by Continental Digital Service France (CDSF) in the framework of the eHorizon project.

RAN slicing enforcement problem. In this article we focus on its resolution in the context of 5G system at radio resources level as it still remains under-explored. For that, resources allocation strategies are proposed as to achieve the aforementioned RAN slicing enforcement requirements.

Further, the remaining of this paper is organized as follows. The RAN slicing enforcement problem is covered in section II with the major 5G elements used in this article. The related work is reviewed in section III. Section IV exhibits the system design and the proposed models. The developed models are uncovered to converge slowly as evaluated in section V. They then limit the RAN slicing enforcement for real time scenarios. Therefore, we propose in Section VI three heuristics that enforce the real time slicing. Then, a comparison of the three algorithms performance with respect to the optimal values given by mathematical models is conducted in section VII. Finally, section VIII concludes this paper with open issues and future work.

II. RAN SLICING ENFORCEMENT PROBLEM

This section covers the important 5G terminologies required for the exhibition of the RAN slicing problem. Mainly, the 5G radio resources structure is presented. Then, the RAN slicing enforcement problem is explained and the required allocations strategies to achieve the above-mentioned requirements are highlighted.

A. 5G background

In 5G system, the physical layer is more flexible with respect to the previous generations. Recall that radio resources in 4G are uniformly distributed over a time-frequency grid, i.e. the later is decomposed into resource blocks (RB) of 1 ms over 12 sub-carriers spaced by 15 kHz.

In order to fulfill the variety of services requirements, increase the network reliability and adapt to frequency range, 5G introduces different radio frames numerologies for sub-6 GHz, and above-6 GHz bands. In this article, we focus on the sub-6 GHz bands. The same developed approaches can be easily applied for bands above-6 GHz. Table I exhibits the different numerologies for sub-6 GHz bands. Each given numerology μ[1] defines the time-frequency resource size in one Transmission Time interval (TTI), TTI=1ms. That is, a numerology μ refers to the sub-carrier spacing (SCS) in frequency domain and the slot duration in time domain. For instance, as depicted in fig. 1, for $\mu = 1$ the radio resource size is fixed to 0.5 ms over 12 sub-carriers spaced by 30 KHz. In general, the SCS scales by $2^{\mu} * 15kHz$ and the slot duration decreases with higher numerology (μ). Such flexibility is essentially introduced as to achieve the diverse services requirements. For example, it is preferable to transmit latency sensitive services in shorter time interval with larger sub-carrier spacing, e.g. $\mu = 3$.

[1]3GPP, TR 38.802, TR 38.804: Study on new radio access technology Physical layer aspects, Study on new radio access technology Radio interface protocol aspects. https://www.3gpp.org/

μ	SCS (kHz)	Slot duration (ms)
0	15	1
1	30	0.5
2	60	0.25

TABLE I: 5G Radio frames numerologies for sub-6 GHz bands

In order to support the coexistence of the multiple numerologies on same carrier, the resources are structured in the so-called tiles [5]. The tile is the smallest subset of frequency and time resources allocated to a particular slice/service with same numerology μ. Hence, for sub-6 GHz three tiles structures are tailored as shown in fig. 1. For instance, the tile structure for $\mu = 0$ is 1 ms over 12 subcarriers spaced by 15 KHz. Further, multiplexing over time and frequency is required for the transmission of the different numerologies, e.g. over time, 3GPP imposes symbol alignment between tiles to insure orthogonality.

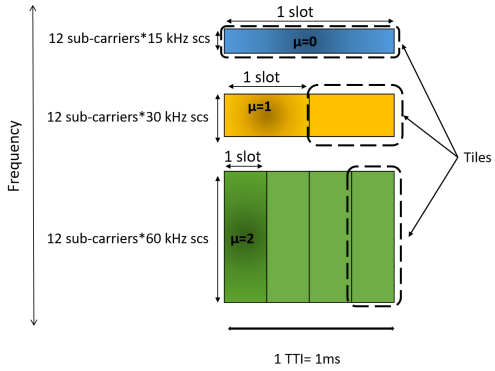


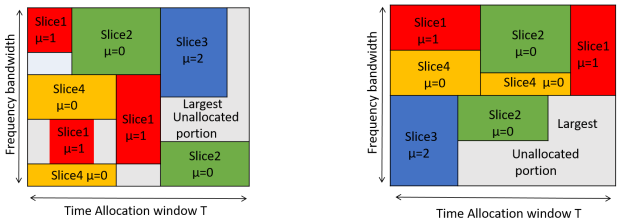
Fig. 1: Sub-6 GHz numerologies

B. Problem Formulation

The enforcement of 5G RAN slicing approach rises many requirements, as indicated in section I, mainly scalability, orthogonality, slices satisfaction and easing the inter-base stations cooperation. In the following, resources refer to the radio resources.

To tackle the scalability requirement, the RAN should be more flexible about the radio resources allocation. With that vision, the RAN slicing enforcement algorithms should allocate resources in a way leaving the largest unallocated portion of resources, instead of sparse unallocated resources. This objective is illustrated in fig. 2. Two time-frequency resource grids are schematized as to explain the difference between an optimal (b) and sub-optimal (a) resource allocation for slices requests in 5G context. Each slice demands a different amount of resources with specific numerology, i.e. a number of tiles. Even though both allocations (a) and (b) satisfy the four slices requests during the allocation window T, it is clear that the allocation strategy in (a) is sub-optimal compared to (b). In fact, the old fashioned resource allocation strategies as (a) might lead to inefficient resource utilization

as they don't consider the different tiles structures during the allocation process. 5G proposes a variety of tiles structures and such old allocation strategies induce a quite small sparse unallocated resources over the resource grid, as represented in fig. 2 (a). Those small leaved resources are considered as wasted as they don't fit any tile structure. Only, the small continuous unallocated portion of resources is then reused by the MNO. In contrast, the allocation strategy in (b) results in a large unallocated portion of resources. Hence, it allows the MNO to further reuse the unallocated resources in an efficient manner, as different tiles structures could fit in this portion, e.g. the sudden delay-critical requests could also be considered by the MNO. Thus, it enables the scalability requirement and increases the resource utilization efficiency.



(a) Sub-optimal allocation (small continuous unallocated area)

(b) Optimal allocation (large continuous unallocated area)

Fig. 2: Optimal and sub-optimal resource allocation

On the other hand, 5G strategies rely on a tight cooperation and coordination among different BSs in the network. Therefore, the RAN slicing enforcement algorithms should also guarantee the allocation of the same radio resources over time and frequency to the same slices among the adjacent BSs, i.e. BSs close enough to interfere. Such allocation eases not only the deployment of 5G advanced techniques such as MIMO and beamforming, but also the transmission schemes for inter-slice interference mitigation. Thus, it improves the overall network performance. Through experimentation, D'Oro et al [4] proved the ability of such allocation to double the network throughput in some cases compared to a random allocation. Fig. 3 illustrates the above point. It schematizes three radio resource allocations on three adjacent BSs (BS1, BS2 and BS3), close enough to interfere. On scheme (a), the resources are allocated to each slice on a given BS independently of the adjacent BSs allocation. Hence, Even-though the same slices exist on the two BSs, they are allocated different time-frequency resources portion. For instance, while slice 3 is allocated the left lower resources portion on BS1, the same portion on BS2 and BS3 is allocated to slice 2. Therefore, an inter-slice interference is observed between slice 2 and slice 3. Contrarily, on fig. 3 (b), each slice owner manages the same time-frequency portion of resources on all BSs. Hence, inter-slice interference is absent. Also, the slice owners have more flexibility to mitigate intra-slice interference and enable the advanced 5G techniques. Further, clearly the allocation strategy in (b) is optimal for the RAN slicing enforcement in 5G compared to the traditional allocation approach in (a), where each BS allocates its resources independently from the adjacent BSs. That is, the RAN slicing enforcement requires a coordinated resources allocation over adjacent BSs.

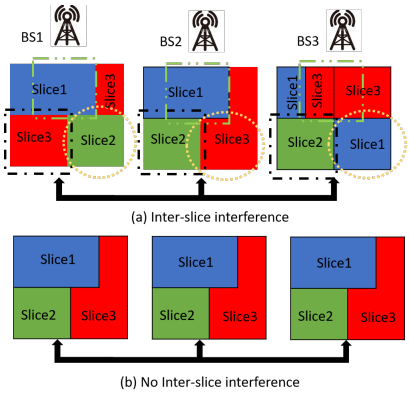


Fig. 3: Resource allocation for inter-slice interference mitigation

We argue that the combination of both allocation strategies allows the realization of the aforementioned requirements for the RAN slicing enforcement. For that, we investigate the optimization of such strategies in the upcoming parts.

III. Related work

In the context of RAN slicing, resource management and orchestration have received significant interest from the research community. Many frameworks [3], [6], [7] have been proposed to deal with the high level wireless resource orchestration and management. While the proposed approaches are effective in resource control and orchestration, they might lack effectiveness for fine grained control scenarios, where performing and enabling advanced 5G transmission techniques are required. Also, a major challenge with these frameworks is the efficient resource allocation while preserving the radio resources isolation.

Recently, researches have converged to enforce the RAN slicing from a resource allocation aspect. To overcome the static resources segmentation limitations, shared allocation strategies are proposed [20], [15], [1]. For instance, B.Han et al. [1] propose the use of Genetic algorithm to optimize resource management between heterogeneous slices with maximized long-term network utility. Although the proposed methods gain in terms of multiplexing, they lack of programmability and resources isolation aspects, that allow each tenant to manage its resources independently. By introducing resources virtualization, Chang et al. [2] formulate the problem as a knapsack problem. An algorithm is proposed to maximize the number of accepted slices with an efficient 5G resource partitioning. Nevertheless, all of the above-mentioned contributions consider a network with only one BS, which limits the deployment of their approaches in a multi-BS network, where each tenant requires a different amount of resources on each BS, based on the channel condition and the number of

connected users. Moreover, the inter-cell interference is not addressed, i.e. cooperation enabling requirement.

Few work has been done in multi-BS system. Netshare [14] and AppRAN [10] frameworks are based on a centralized controller that decides the amount of resources to be allocated for each tenant on each BS. Then, the slice resources allocation is executed on each BS. Thus, the systems ensure isolation at best from packet-level. Also, they didn't take the RAN slicing cooperation enabling requirement in their approaches.

On the other hand, the contribution in [17] sheds light on four approaches for the radio resources management from multi-cell multi tenant perspectives. Although, the fine grained resource management is covered to mitigate inter-slice interference, they didn't propose any algorithm to enforce their approach. Authors in [21] formalized a multi-objective function that minimizes the inter-slice interference in 5G dense networks with isolation assurance. Nevertheless, the Scalability and satisfaction requirements are not taken into consideration.

D'oro et al. [4] proposed an algorithm to enforce the RAN slicing policies with interference mitigation. This is enabled through guarantying that the same (or similar in time/frequency) resource blocks (RB) are assigned to the same slices when BSs are close enough to interfere among themselves. Although, their approach is efficient from interference mitigation perspective in 4G networks, it might be ineffective in resource utilization in the 5G system with the presence of different numerologies. Also, their work targets only orthogonality, satisfaction and cooperation enabling requirements. Thus, the scalability requirement is not considered. Moreover, contrarily to [4] that performs the allocation over two interfering BSs basis, our work aims to an allocation from a multi-cell perspective, i.e. proceeding toward real 5G network deployment.

Overall, the work discussed in this article differs from existing work in that the RAN slicing enforcement is tackled in a multi-cell multi-slice perspective adapted to 5G system. We propose a new formulation of the RAN slicing enforcement problem that handles the slicing requirements in terms of orthogonality, scalability, satisfaction and cooperation enabling.

IV. System design and model

Considering the RAN slicing requirements to achieve the 5G objectives, we proceed to the system design of this work. It highlights the 5G RAN vision where the RAN is controlled in a centralized manner. This is crucial, as a cooperation between BSs is required for a global resources allocation. Moreover, the presence of the flexible resources structures involves a fine grained resource management. Therefore, the resource grid decomposition is exhibited. Further, we investigate the possibility of deploying the already discussed optimal allocation strategies. System models are then depicted.

A. System design

Let consider a set of BSs covering a geographical zone. The 5G base station (BS) is named gNB. The gNBs cluster is controlled by a centralized SD-RAN controller, as illustrated in fig. 4, noted R. This is essentially due to the high cooperation level required between the gNBs. The SD-RAN controls the RAN traffic, e.g. it receives the slices demand on each gNB (5G Base station) and all the RAN signaling information. We assume that SD-RAN copes with the scheduling and radio resources allocation over the specific zone. With the advanced implementation of intelligence in the radio part, the estimation of the slice demand traffic is possible [18]. On the other hand, several researches have been interested in the slicing profile generation, i.e. the slice demand and resources assignment [7], [9]. Therefore, the slicing profile is considered as an input argument for our system.

Furthermore, we propose to take advantage of the RAN intelligence in the 5G and the upcoming cellular networks to build a proactive allocation system for slices resources. For instance, with the pre-knowledge of the slices demand over the gNBs set, the SD-RAN proposes a resource allocation for the upcoming 10 ms, which corresponds to the frame duration in current cellular networks. This has the advantage to minimize the signaling exchange between the SD-RAN controller and the gNBs over the cluster.

On the other side, with the different tiles structures proposed by the 5G, an efficient resource management involves a fine grained access to the resources. Therefore, we put forward a new scheme for the resources grid decomposition as explained in the following.

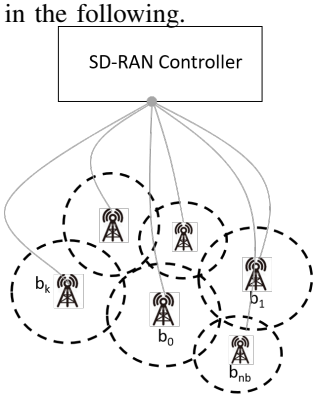


Fig. 4: System design

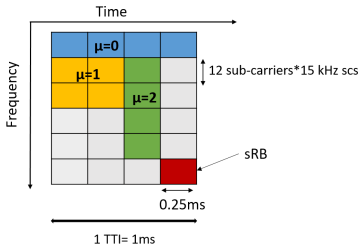


Fig. 5: Resource grid decomposition

1) Radio resources grid decomposition:
With the diverse services flows and the increasing demand of cellular traffic, the 3GPP emphasizes the importance of treating the 5G radio resources differently from the earlier standards. For that, it introduces different numerologies on each frequency band. Each numerology is efficient for a specific service flow, particularly, the $\mu = 2$ is much required for services with low latency. In the same perspective, we push this flexibility a step forward, and propose to handle the radio resources at small time and frequency granularities.

To that end, each gNB is entitled by its resource grid. With the variety of numerologies, we consider a resource grid decomposed into the smallest granularity in time and frequency. For instance, for the sub-6 Ghz bands, where μ can take values in $\{0, 1, 2\}$, the smallest resource block (sRB) is of size 180 kHz*0.25 ms. Fig. 5 illustrates a decomposition

for a small resource grid of 1 ms over 1.4Mhz. Three tiles structures are considered. Namely, the tile structure for a given slice with $\mu = 1$ is a square of 2*2 sRBs.

The proposed resource grid decomposition allows a fine grained manipulation of the available resources, as they are shaped based on the slices numerologies requirements. Moreover, this decomposition results in an efficient control and management of the scarce radio resources. Also, it eases the way for the tight cooperation required by the 5G advanced techniques.

2) Objective formulation: Given the SD-RAN controller of a given zone, each gNB is characterized by its decomposed radio resource grid. For a specific allocation window, each slice is assigned an amount of tiles on each gNB over the RAN and the SD-RAN proposes an allocation for the slices tiles taking into account both of the following objectives:

The first objective aims to maximize the placement of the tiles in the resource grid with respect to the BSs set, i.e. the maximization of the number of allocated tiles in the same or similar position (time/frequency), for each slice, over the gNBs set. This is because of the tight cooperation and coordination involved over the RAN for the 5G advanced techniques deployment as explained in fig.3.

Moreover, while allocating the slices tiles, an efficient radio resources utilization in each gNB is required. The latter could be achieved by an allocation that minimizes the sparse wasted unallocated resources. Or from another vision, maximizes the largest continuous unallocated resources space of each resource grid. Such allocation allows the MNO to scale up/down slices demand and also accept new slices requests (i.e. scalability) through reusing the largest unallocated resources portion. Thus, the objective is implicitly multi-objective.

This multi-objective allocation strategy, combining space and position optimization, assures an enforcement of the RAN slicing. In order to reach the optimal solution of this multi-objective problem, we propose to attain the optimal solution for each objective separately. Then, three heuristics are depicted for simultaneous resolution.

B. System Model

Let denote $B = \{b_1, ..., b_{n_b}\}$ the cluster/set of n_b gNBs covering a geographical area. Notice that the gNB might offer a macro as well as small cell coverage. They are controlled by a centralized SD-RAN (Software Defined RAN) controller R, as illustrated in fig. 4. The multiple gNBs are adjacent to cover efficiently the geographical area. Such adjacency is highly vulnerable to interference.

Let us consider that R receives n_s slices requests to be served simultaneously during the allocation window T, $S = \{s_0, s_1, ..., s_{n_s}\}$. Based on the slices requirements on each BS (gNB), R generates the slicing profile $\Gamma = (\gamma^{\mu}_{s_i,k})_{s_i \in S, k \in B}$, with $\gamma^{\mu}_{s_i,k}$ is the amount of tiles to be allocated to slice s_i in BS b_k with numerology μ during T. Each slice s_i is supposed to have the same numerology μ over B, but requests a different amount of resources on each gNB, i.e. different slices have different numerologies μ over B. As the generation of the slicing profile Γ has been already investigated by many researchers, it is considered as an input argument in our system model taking the resources grid size as the upper limit. Therefore, once it is generated in our work, it is primary to test its feasibility before the allocation process. In other words, a verification step of the possibility to allocate all the assigned slices resources in the appropriate gNB resource grid. In the following, we propose an exact method with an underlying constraint programming (CP) approach to test the slicing profile feasibility and tiles placement objective. This is because the CP eases the resolution of discrete problems through high level constraint propagation and controlled search behaviors [19]. A constraint problem is stated as a set of variables, where each variable has a finite domain of values, and a set of relations on subsets of these variables.

*1) **Slicing Profile Feasibility Model (SPFM)**:* Let $g_k = (r_{k,x,y})_{0 \leq x \leq N_r, 1 \leq y \leq T}$ be the matrix representing the resource grid of gNB b_k, $k \in \{0, .., n_b\}$, with T and N_r representing the number of temporal slots (0.25 ms) and frequency channels of 180 kHz respectively, i.e. $r_{k,x,y}$ symbolizes the sRB in gNB b_k in position (x,y).[2] The resource grid size is therefore $A = N_r * T$.

A slicing profile Γ is considered as feasible, if all the tiles assigned to a group of slices on a given gNB b_k can be allocated over g_k without any overlapping, for all $k \in \{0, .., n_b\}$.

Let $\zeta_{s_i} = \{\tau_j \quad for \quad j \in \{0, .., \gamma^{\mu}_{s_i,k}\} \forall b_k \in B\}$ be the set of tiles requested by slice s_i over B. Each tile has a form of a rectangle based on the slice numerology (see fig. 5). From that, we represent each tile τ_j of slice S_i in gNB b_k by two interval variables $X_{b_k,s_i,j}$ and $Y_{b_k,s_i,j}$. They refer to the tile allocation over frequency and time axis respectively. The length of the intervals is fixed as to reproduce the rectangle form of the tile. Particularly, if the tile τ_j corresponds to a slice resource with $\mu = 2$, the length of $X_{b_k,s_i,j}$ and $Y_{b_k,s_i,j}$ are fixed to 4 sRBs and 1 sRB respectively. Therefore, a non overlapping between two tiles τ_j and τ_h on a given g_k refers to their non overlapping over X and Y axis, i.e. $X_{b_k,s_i,j} \cap X_{b_k,s_i,h} = 0$ and $Y_{b_k,s_i,j} \cap Y_{b_k,s_i,h} = 0$.

Let α^{j}_{x,b_k}, $\alpha^{t^j_i}_{y,b_k}$ be the variables referring to the starting point of the two intervals $X_{b_k,s_i,j}$ and $Y_{b_k,s_i,j}$ respectively. And β^{j}_{x,b_k}, β^{j}_{y,b_k} point out their ends. The SPFM can be therefore formulated using CP approach as follows:

$$\alpha^{j}_{x,b_k} \leq N_r \qquad \forall b_k \in B \quad \forall j \in \zeta_{s_i} \quad \forall s_i \in S \quad (1)$$

$$\alpha^{j}_{y,b_k} \leq T \qquad \forall b_k \in B \quad \forall j \in \zeta_{s_i} \quad \forall s_i \in S \quad (2)$$

$$\begin{aligned} &(\alpha^{j}_{x,b_k} \geq \beta^{r}_{x,b_k} \vee \alpha^{r}_{x,b_k} \geq \beta^{j}_{x,b_k}) \wedge \\ &(\alpha^{j}_{y,b_k} \geq \beta^{r}_{y,b_k} \vee \alpha^{r}_{y,b_k} \geq \beta^{j}_{y,b_k}) \quad \forall r, j \in \zeta_{s_i} \quad \forall b_k \in B \end{aligned} \quad (3)$$

The constraints 1 and 2 limit the allocation bounds of each tile over both X and Y axis respectively. Then, the second constraint 3 ensures the allocation of the required slices tiles on each gNB without any overlapping between two tiles. This

[2]In this work, the sub-6 GHz band is treated, but it can be extended easily to the above-6 GHz band, where the sRBs will be of size (1440 kHz*62.5μs).

way the model is feasible when all the slices demands in a given gNB are allocated to the appropriate resource grid.

For feasibility model (SPFM) implementation, the IBM CPOptimizer constraint programming solver IBM ILOG (CPO)[3] is used. It provides a high level scheduling constraints. The allocation of tiles taking into consideration the constraint 3 can be directed by the $SetSearchPhase$ function. It guides the search for positions with $SearchPhase$ over X-axis and Y-axis for each tile with respect to non-overlapping constraint. Let VX_k denotes all the interval variables over X-axis representing the tiles assigned for the allocation on b_k, and VY_k the ones over Y-axis. The use of $searchPhase$ is therefore written as:

$$SetSearchPhases(searchPhase(VX_k), searchPhase(VY_k)) \quad \forall b_k \in B \tag{4}$$

Once the SPFM is verified and the slicing profile is feasible, a RAN slicing enforcement policy ψ is required to fulfill the requirements depicted in section I. It should lead to an optimal radio resources allocation over the b_{n_b} gNBs.

As stated earlier, the problem is treated as a Multi-Objective Optimisation Problem (MOOP). One objective carries the maximization of slice' tiles placement in the same frequency-time position in the resource grid of the adjacent BSs. The other objective deals with the maximisation of the largest unallocated continuous portion of radio resources on each gNB. In the following, the radio resources placement objective is modeled with an exact optimization method. And, the approach followed to carry the largest continuous unallocated space during the resources allocation is explained.

*2) **Enforcement of Slice Resources Placement (ESRP)**:* The policy ψ has the objective to maximize the tiles placement of a given slice in the same position over the set of gNBs, B. For that we introduce the notion of tied tile.

Definition 1 (Tied tile): A given tile τ_j is tied to a slice s_i over a gNBs set B if and only if the tile τ_j is placed in the same position over all the gNBs in the cluster B, i.e. τ_j has the same frequency and time position on each g_k, $\forall b_k \in B$

Each tile τ_j of slice S_i in gNB B_k is represented by two interval variables $X_{b_k,s_i,j}$ and $Y_{b_k,s_i,j}$ as explained in IV-B1. With α^j_{x,b_k}, α^j_{y,b_k} are the variables indicating the starting point of the two intervals $X_{b_k,s_i,j}$ and $Y_{b_k,s_i,j}$ respectively, and β^j_{x,b_k}, β^j_{y,b_k} their ends. With that, a tile is tied if and only if $\alpha^j_{p,b_k} = \alpha^j_{p,b_{k'}}$ and $\beta^j_{p,b_k} = \beta^j_{p,b_k}$ $\forall b_k \in B$, $p \in \{x, y\}$.

In other words, a tile of a given slice is tied if all its sRBs are allocated in the same position over the set of involved gNBs, i.e. gNBs where tile τ_j is present, as the slice demand varies over the gNBs. Consequently, we introduce the concept of tied sRB:

Definition 2 (Tied sRB): A given sRB $r_{k,x,y}$ is tied to a slice s_i over B if and only if the sRB is allocated to the same slice over each gNB in B, i.e. $(x_k, y_k) = (x_{k'}, y_{k'})$ $\forall b \in B$.

[3]CPLEX Optimization studio 12.9 www.cplex.com

Even though the allocation is performed per tile, it is clear that the maximization of the total amount of tied tiles for all the slices turns out to the maximization of the total amount of tied sRBs for all the slices, i.e. a tile is composed of 4 contiguous sRBs. Accordingly, we model mathematically the system as to maximize the total amount of tied sRBs. We have conceived two ways to model this objective. Both model versions are depicted in the following. It is in fact interesting to compare their scores as to find the best optimal model. Notably, the convergence time that is considered as a pivotal performance metric for each model implementation.

- **ESRP model version 1 (ESRP-v1)**
  For a given tile τ_j of s_i, let denote θj the amount of its tied sRBs over B. As each tile τ_j is symbolized by two interval variables on each g_k, $X_{b_k,s_i,j}$ and $Y_{b_k,s_i,j}$, θ_j corresponds to the overlap length between both intervals over all the involved gNBs. It can be formulated as follows:
  $$\theta_j = \Pi_{p\in\{x,y\}}(\Psi^j_p - \Upsilon^j_p)$$
  Where $\Psi^j_p = \min_{\forall b_k \in B_j} \beta^j_{p,b_k}$ and $\Upsilon^j_p = \max_{\forall b \in B_j} \alpha^j_{p,b_k}$, $p \in \{x, y\}$, identify the start and end position of the overlap between the rectangles over the involved gNBs over both frequency and time axis. It is worth noting that the overlap score between two intervals I_1 and I_2 is given by the CPO function $OverlapLength$, i.e. $OverlapLength(I_1, I_2)$. Such function returns the length of overlap between two intervals I_1 and I_2. By way of illustration, let us consider the allocation over two gNBs of tile τ_j with $\mu = 2$ as shown in fig. 6. Let suppose that both tiles have the same y-axis position. The amount of tied sRBs is exactly the surface of the overlap between the τ_j in gNB b_0 and τ_j in gNB b_1. The starting point of this surface over each axis can be computed by $\max(\alpha^j_{p,b_0}, \alpha^j_{p,b_1})$ and the end position by $\min(\beta^j_{p,b_0}, \beta^j_{p,b_1})$. On the X-axis, they are equal to α^j_{x,b_0} and β^j_{x,b_1} respectively. Thus, it represents two sRBs, i.e. $\theta_j = 2$. That is, two sRBs are tied between the two gNBs. Therefore, the total tied sRBs for a given slice s_i over B is given by:
  $$\Theta_{s_i} = \sum_{j\in\zeta_{s_i}} \theta_j$$
  Further, the total tied sRBs over B can be expressed as the summation of the total tied sRB of each slice over B: $\chi = \sum_{s_i\in S} \Theta_{s_i}$
  The objective is then formulated as to find the slicing enforcement policy ψ that maximizes χ, Ψ is the set of all possible policies. It is developed with a constraint programming (CP) approach as the SPFM (section IV-B1) resolution. In the CP implementation, the constraints are explicitly stated to shape the aimed solution, i.e. in this case the maximization of χ.

$$\max_{\psi\in\Psi}(\chi) \tag{ESRP-v1}$$

subject to

$$\alpha^j_{x,b_k} \leq N_r \quad \forall b_k \in B \quad \forall j \in \zeta_{s_i} \quad \forall s_i \in S \tag{5a}$$
$$\alpha^j_{y,b_k} \leq T \quad \forall b_k \in B \quad \forall j \in \zeta_{s_i} \quad \forall s_i \in S \tag{5b}$$
$$\sum_{j \in \zeta} \theta_j \leq \gamma^{\mu}_{s_i,k} \quad \forall k \in B \quad \forall s_i \in S \tag{5c}$$
$$\begin{aligned} &\alpha^j_{x,b_k} \geq \beta^r_{x,b_k} \vee \alpha^r_{x,b_k} \geq \beta^j_{x,b_k} \wedge \\ &\alpha^j_{y,b_k} \geq \beta^r_{y,b_k} \vee \alpha^r_{y,b_k} \geq \beta^j_{y,b_k} \forall r, j \in \zeta_{s_i} \forall b_k \in B \end{aligned} \tag{5d}$$
$$\Psi^j_p \geq \Upsilon^j_p \quad \forall p \in \{x, y\} \quad \forall \tau_j \in \zeta_{s_i} \quad \forall s_i \in S \tag{5e}$$

The constraints (5a) and (5b) ensure that all the allocated tiles are inside the resource grid, i.e. the allocation doesn't outpace the gNB grid limits on both frequency (5a) and time (5b) axis. The second constraint (5c) guarantees that each slice receives at maximum its required amount of tiles over each gNB. Then, the constraint (5d) addresses the non overlapping between tiles on the same gNB, i.e. each sRB is allocated at maximum to one slice. Hence, the slices orthogonality is achieved (i.e. resource isolation). Then, the last constraint (5e) assures the non negativity of each tied sRB surface.

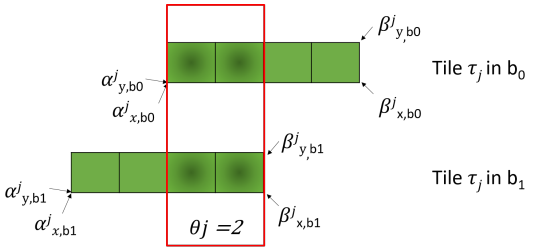


Fig. 6: Illustration of tied sRBs of a tile between two gNBs

- **ESRP model version 2 (ESRP-v2)**
  In this ESRP version, in order to get the tied sRBs of a given tile τ_j over a gNBs set, we propose to add its overlap over both axis X and Y taking into account the numerology type. In other words, let denote ξ_j the total tied sRBs of a tile τ_j over a set of gNBs. It is expressed by:

$$\xi_j = [\sum_{p \in \{x,y\}} (\Psi^j_p - \Upsilon^j_p)] - \delta$$

  With $\Psi^j_p = \min_{\forall b_k \in B_j} \beta^j_{p,b_k}$ and $\Upsilon^j_p = \max_{\forall b \in B_j} \alpha^j_{p,b_k}$, $p \in \{x, y\}$. δ is a binary variable, $\delta \in \{0, 1\}$, it equals 1 when the tile numerology is $\mu = 0$ or $\mu = 2$, and 0 otherwise. In fact, ξ_j refers to the tied sRBs surface computed without multiplication.
  Further, for a given slice, the total tied sRBs (TTR) over B can be expressed by: $\Xi_{s_i} = \sum_{j \in \zeta_{s_i}} \xi_j$
  With ζ_{s_i} is the set of tiles requested by s_i over B. Thus, the total amount of tied sRBs over all B is formulated by:

$$\Phi = \sum_{s_i \in S} \Xi_{s_i}$$

The slicing enforcement policy that maximizes Φ is formulated as follows with CP resolution approach:

$$\max_{\psi \in \Psi}(\Phi) \tag{ESRP-v2}$$

subject to

$$\alpha^j_{x,b_k} \leq N_r \quad \forall b_k \in B \quad \forall j \in \zeta_{s_i} \quad \forall s_i \in S \tag{6a}$$
$$\alpha^j_{y,b_k} \leq T \quad \forall b_k \in B \quad \forall j \in \zeta_{s_i} \quad \forall s_i \in S \tag{6b}$$
$$\sum_{j \in \zeta} \xi_j \leq \gamma^{\mu}_{s_i,k} \quad \forall k \in B \quad \forall s_i \in S \tag{6c}$$
$$\begin{aligned} &\alpha^j_{x,b_k} \geq \beta^r_{x,b_k} \vee \alpha^r_{x,b_k} \geq \beta^j_{x,b_k} \wedge \\ &\alpha^j_{y,b_k} \geq \beta^r_{y,b_k} \vee \alpha^r_{y,b_k} \geq \beta^j_{y,b_k} \forall r, j \in \zeta_{s_i} \forall b_k \in B \end{aligned} \tag{6d}$$
$$\Psi^j_p \geq \Upsilon^j_p \quad \forall p \in \{x, y\} \quad \forall \tau_j \in \zeta_{s_i} \quad \forall s_i \in S \tag{6e}$$
$$\Psi^j_x - \Upsilon^j_x \geq 0 \Leftrightarrow \Psi^j_y - \Upsilon^j_y \geq 0 \forall \tau_j \in \zeta_{s_i}, \forall s_i \in S \tag{6f}$$

The constraints (6a) and (6b) ensure the allocation inside the gNB grid limits. The constraint 6c assures that each slice receives at maximum its required amount of tiles over each gNB. Furthermore, the orthogonality between slices is ensured by the constraint (6d), i.e. each sRB is allocated to only one slice. Then, the non negativity of each overlap either over X or Y axis is verified by the constraint (6e). Further, the last constraint assures that the model looks for the overlapping over X axis as well as over Y axis. This way, the tied sRB surface is realized.

*3) **Largest Continuous Unallocated Space (LCUS)**:* The second objective targets the maximization of the continuous unallocated space on each gNB resource grid g_k, $k \in B$. For that, the problem is tackled as a two-dimensional rectangle bin packing (2DBP) optimization problem. In such problem, given a sequence of rectangular objects with specific height and width, the objective is to place the maximum of these objects inside a minimum bins of fixed size. With constraint of no-overlapping between the rectangles. The NP-Hardness of this problem is proven by a reduction from the 2-partition problem [8].

Let project the 2D bin packing to the context of the resource allocation with LCUS objective. In our case, the rectangular objects to pack in the bins are the slices tiles with their specific numerologies μ. Each tile τ_j has a form of a rectangle based on the slice numerology. Particularly, the tiles of a given slice with numerology $\mu = 2$ have a rectangular form of width and height equal to 4 and 1 respectively. Each gNB decomposed resource grid g_k is represented by a bin. It is supposed that the bins have the same size over all the gNBs set B, i.e. $\forall k \in B \quad size(g_k) = A$. Only one bin is available for the packing for each gNB. Its size is exactly the size of the resource grid in terms of time and frequency resources, i.e. allocation time over the carrier bandwidth. This can be considered as Knapsack use case problem of 2DBP.

The Knapsack problem is argued to be NP-hard. The achievement of the LCUS objective is then also NP-hard. Over decades, several algorithms are proposed in the literature

to approximate the optimal solution for 2DBP. They include the Skyline algorithm proposed in [12]. It starts by placing the first rectangle object in the bottom left (BL). Then, each new rectangle object is left-aligned on top of the skyline level that results in the top side of the object lying at the bottom-most position of the bin. The topmost edges of already packed objects is tracked as illustrated by the red line in fig. 7. The example shows the packing of 6 tiles with $\mu = 1$ and 5 tiles with $\mu = 0$ using the skyline algorithm. The algorithm then maintains the list of these horizons or "skyline" edges. The later grows linearly in the number of the packed rectangle objects. And for each rectangle packing top of a hole, it is possible and easy to compute the free rectangle that would be lost after packing. Thus, it is stored and evaluated for an aforementioned use. Such approach is referred as a waste map (WM) improvement for the skyline (BL) heuristic.

The authors tested a benchmark of 2DRP heuristics and variants of the skyline algorithms. They proved that skyline-BL-WM outperforms all the best tested online packers, in terms of packing efficiency as well as the run-time performance, when packing to one bin at a time. As the algorithm packs the objects in a way to minimize the wasted space between the packed objects, it results in letting the largest unallocated space. Therefore, the skyline-BL-WM heuristic is chosen to approximate the LCUS solution. Mainly for two reasons:

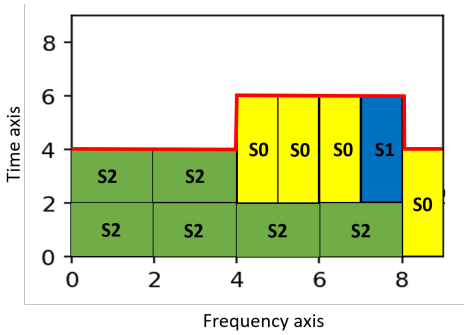


Fig. 7: Illustration of Skyline algorithm packing 3 slices tiles

- The algorithm is highly performing in both time convergence and packing efficiency on one bin. This corresponds to the use case of this work, i.e. each g_k is represented by one bin and the allocation is allowed only in this bin.
- The algorithm approach seeks to pack the objects (tiles) as to have the lowest skyline (contour). This is advantageous, as we are seeking to let the maximum of unallocated space over time axis. Hence the bottom could be chosen as the frequency axis. The skyline is then aligned over time as shown on fig. 7. This way, the allocation of different tiles numerologies could occupy the continuous unallocated space, e.g. serving the sudden delay-critical requests. Thus, the scalability requirement is assured.

The Skyline heuristic approximates the LCUS solution. Thus, there is a need to evaluate its performance. For that, an optimal score is necessary. In this work, the naive method that encounters the LCUS topmost upper bound is used, as depicted in the following. It is therefore considered as the optimal LCUS solution.

+LCUS topmost upper bound (LTUB): On a given resource grid g_k, $k \in B$, the topmost LCUS upper bound can be achieved when all the tiles of all the slices are allocated without overlapping and where no space left in between, i.e. non existence of wasted space between allocated tiles. The size of each resource grid can be computed, as stated before, by $A = N_r * T$, with N_r is the number of frequency channels and T the allocation window. A refers also to the total number of available sRBs on each g_k. Given the slices demand $\gamma^{\mu}_{s_i,k}$ in terms of tiles, the total required tiles on each b_k can be computed by: $\rho_k = \sum_{s_i \in S} \gamma^{\mu}_{s_i,k}$. Therefore, the total allocated sRBs on each g_k equals $4 * \rho_k$, as each tile contains 4 sRBs. From that, the highest upper bound of LCUS, noted $LCUS_k$, can be quantified by $LTUB_k = A - 4 * \rho_k$. The topmost upper bound over all B is then: $LTUB = \sum_{k \in B} LTUB_k$.

V. Models evaluation

In the previous section, the two objectives are modeled separately. Two mathematical models are proposed for the ESRP objective. Their evaluation is necessary, as well as the comparison of their performance. The second objective is treated as a 2DBP optimization problem. With the NP-hardness of such resolution, the skyline heuristic is used for this resolution. In order to evaluate its performance in term of LCUS, the LTUB is taken as the optimal solution. In this section, the evaluation of both objectives solutions is conducted. The metrics of this evaluation are: the total tied sRBS (TTR) for ESRP versions, the LCUS for skyline and the convergence time for all algorithms (i.e. ESRP-v1, ESRP-v2, Skyline).

A. Performance metrics computation

For the model evaluation, three metrics are used:

*1) **Convergence Time (CT)**:* For CPO models, the CT is computed with time function, and time limit is fixed to 600 s. For skyline, python time module is used.

*2) **Total tied sRBs (TTR)**:* The objective behind the implementation of both ESRP models is to maximize the total tied sRBs during an allocation of slices set over a gNBs set. Therefore, the basic evaluation metric is the achieved total tied sRBs, noted TTR. If the model doesn't reach the optimal TTR score during the 600 s, the compilation is stopped and the upper bound score is saved as optimal score. Otherwise, the objective score is retained. In both cases, TTR is extracted directly as an output from the optimization model.

*3) **Largest Continuous Unallocated Space (LCUS)**:* In order to count the largest continuous unallocated space (LCUS) after each allocation, we derive a binary matrix from the resource grids after the allocation completion, i.e. each allocated sRB to a given slice corresponds to an element matrix with value equals 1 and the unallocated sRBs (elements) worth 0.

Then, we apply the Connected Component Labeling (CCL) with the Depth First Search (DFS) method [11] on each binary matrix. A connected component in a matrix is the subset of matrix elements with same value, where each element is reachable by the other elements. Thus, in our case the LCUS is exactly the maximum subset of zeros among each matrix, i.e. continuous unallocated sRBs.

B. ESRP evaluation: Slice Resources Placement Problem

The resource grid size is fixed, i.e. $N_r = 27$, $T = 40$. For each test, the number of slices requests and the number of adjacent BSs are fixed. The slicing profile Γ is randomly generated for each simulation run as to have at maximum 80% of the grid usage. 100 independent simulation runs are realized for each given test. Each test has a fixed size of slices and gNBs sets. Six B set sizes and seven slices sets have been tested. Thus, a total of 4200 simulation run is performed in this work. For each simulation run the model feasibility (SPFM) is tested. In case of its feasibility, the same instances are used for ESRP models and skyline. Otherwise, a new slicing profile is generated. The results are then averaged over all the simulations runs for each test.

*1) **Total Tied sRBs (TTR)**:* Fig. 8 and fig. 9 illustrate the optimal/upper bound TTR score with both models as a function of B size when serving 3 slices and as a function of S size for a system with 3 gNBs respectively. Same TTR score is achieved by both models over the different B and S sizes (two models curves are superposed). The score is increasing with B size growth. Thus, larger B sets produce higher TTR scores. Regarding S variation, a slight variation of TTR score is observed (fig. 9). This reflects the insensitivity of both models to the slices set growth with respect to TTR.

It is worth noting that out of the total feasible simulation runs (4198), the successful optimal TTR is achieved only 15,55% and 14,7% with ESRP-v1 and ESRP-v2 respectively. Most of the optimal scores are reached for the small systems composed of 2 gNBs. Therefore, both models are unable to achieve the optimal TTR score for bigger systems within 600s.

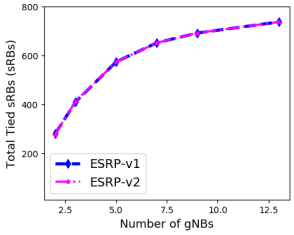


Fig. 8: ESRP TTR(sRBs) as a function of B.

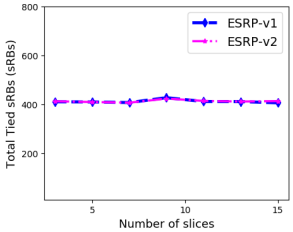


Fig. 9: ESRP TTR(sRBs) as a function of S.

*2) **Convergence Time (CT)**:* Fig. 10 and fig. 11 plot the CT in seconds as function of the B size with different slices sets and S size with different B sets respectively. For all case studies, the ESRP (ESRP-v1 and ESRP-v2) models have quite similar CT, i.e. their curves are superposed. For small systems the models converge approximately in the order of hundreds of seconds. Then, the CT increases gradually with the B or S set size growth. For larger systems, the models CT reach quickly 600 s that corresponds to the prefixed time limit.

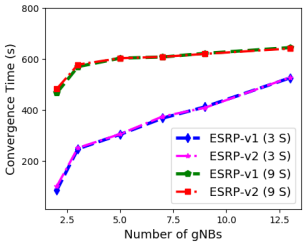


Fig. 10: ESRP models CT (s) as a function of B serving 3 and 9 slices.

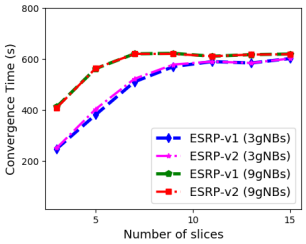


Fig. 11: ESRP models CT (s) as a function of S with 3 gNBs and 9 gNBs.

C. LCUS evaluation: Unallocated Space Problem

In this part, the achieved LCUS with skyline is compared with the upper bound LCUS, LTUB with respect to S and B sizes.

*1) **Largest Unallocated Space (LCUS)**:* Fig. 12 shows the variation of LCUS by skyline and LTUB as a function of B size in a system serving 3 slices. Over the different B sizes, the skyline heuristic reaches the topmost upper bound LCUS (LTUB). The two curves representing the LTUB and skyline score are similar. This reflects the capability of skyline to allocate efficiently the slices tiles without any space waste in between when the slicing profile is feasible. The LCUS is increasing with B size. This is expected, as the LCUS is the sum of $LCUS_k$, $k \in B$, over B. Moreover, the LCUS varies slightly with S size variation as shown on fig. 13. With different S sets, the skyline always attain the LTUB. From that, it is clear that LCUS skyline is insensitive to the number of served slices.

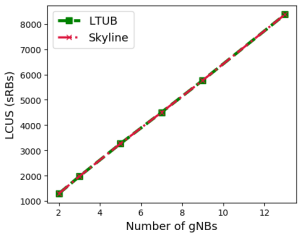


Fig. 12: LCUS (sRBs) for both Skyline and LTUB as a function of B set size.

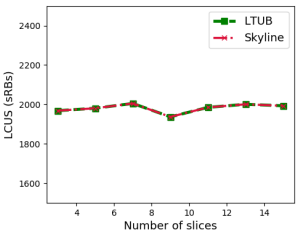


Fig. 13: LCUS (sRBs) for both Skyline and LTUB as a function of S set size.

*2) **Convergence Time (CT)**:* The skyline CT as a function of B size and S size is shown on fig. 14. The skyline converges in the order of hundreds of milliseconds. The CT increases for higher B sets (fig. 14 (a)). Nonetheless, it doesn't outpace 0.45 s. A small CT difference is remarked when serving 3 and 9 slices. It is zoomed out on fig. 14 (b). For lower gNBs set, the CT doesn't exceed 0.1 ms for different slices set sizes. Then, for larger B size, the CT is between 0.16 s and 0.45 s for the various S set size. Therefore, the skyline is interesting with respect to CT, as it can be implemented for SD-RAN real time allocations.

D. Discussion

Overall, both ESRP models converge slowly in the order of hundreds of seconds. This is with the fixed time limit before

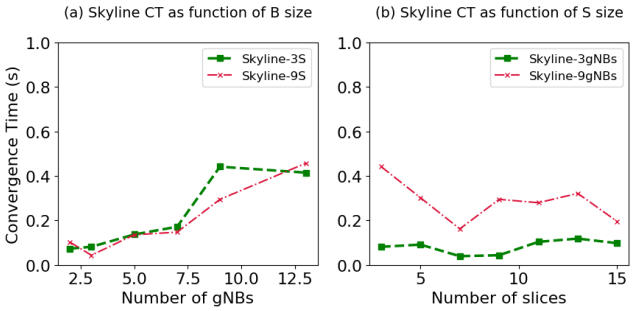


Fig. 14: CT for Skyline as a function of B set size (a) and S set size (b).

the simulations run, i.e. 600 s. Hence, the ESRP versions might converge at higher time scales. Also, the percentage of reaching the objective (i.e. optimal TTR) within this time limit is feeble.

It restricts their implementation for real time slicing. There is therefore a need for heuristics to achieve the cooperation enabling requirement rapidly.

On the other hand, the NP-hardness of the LCUS objective drove us to use the heuristic solution, especially the skyline heuristic. The later has demonstrated good performance in terms of largest continuous unallocated space that attains the topmost upper bound (LTUB) as well as a convergence time in less than 0.45 s for various B and S sets. This makes the skyline suitable for real time deployments of RAN slicing without enabling cooperation requirement.

With the aim of real time RAN slicing enforcement, an heuristic based solution seems to be the best choice, as the ESRP models converge slowly in the order of hundreds of seconds without taking into account the scalability requirement. Accordingly, we propose three heuristics to support the RAN slicing requirements.

VI. Proposed heuristics

The aim of this work targets the real time RAN slicing enforcement. For that, an allocation strategy combining the maximisation of the total tied sRBs over a given set of gNBs as well as the largest continuous unallocated space is argued to reach such aim. The space maximization objective is uncovered to be NP-hard. And, the developed ESRP models converge slowly. Thus, it limits the real time slicing enforcement. Therefore, we propose to tackle this multi-objective problem with heuristic based approach.

Given the muti-objective criterion, a compromise between both objectives is unavoidable. The slice owner might have the possibility to allocate its tied resources to the users at the cells boards highly affected by interference. Thus, it can enable the cooperation techniques on only these resources. Certainly, a slice with higher TTR is much more beneficial, as the slice owner would have more flexibility on its resources. On the other hand, the LCUS enables the scalability. Thus, the MNO can serve more slices. Moreover, an improvement of the current served slices QoS can be achieved by scaling up their resources. In addition, it allows a high level of spectral efficiency. Accordingly, The LCUS is prioritized in the heuristic development.

The aforementioned used skyline heuristic reaches the optimal LCUS in a small time scale, i.e. maximum 0.45 s. This is attractive from the real time implementation perspective. For such reason, the proposed heuristics use the skyline as an underlying allocation technique and three heuristics are developed as depicted in this section. With skyline approach, the LCUS is guaranteed while the TTR is targeted at best to find the optimal slicing enforcement policy.

A. Heuristic 1: Highest Slice First HSF

Each slice is assigned a different amount of resources on each BS. Thus, the slices requiring higher amount of resources over B are expected to generate high number of tied sRBs. Considering such fact, the total required resources over B is computed for each slice based on the slicing profile $\Gamma = (\gamma^{\mu}_{s_i,k})_{s_i \in S, k \in B}$, ($\gamma^{\mu}_{s_i,k}$ is the amount of tiles to be allocated to slice s_i in BS b_k with numerology μ during T) as follows: $\lambda_{s_i} = \sum_{k \in B} \gamma^{\mu}_{s_i,k}$.

The algorithm first tries to set the bigger slices at the same position in the interfering BS. Therefore, the slices are sorted in a decreasing order based on λ_{s_i}. Hence, the slice with higher amount of tiles is first served and the lower last. We denote such method as the `Highest Slice First, HSF`. It is represented in Algorithm 1 and performs as follows:

i) compute the total required resources of all the slices over B, $\Lambda = (\lambda_{s_i})_{s_i \in S}$.
ii) sort the slices in decreasing order based on λ_{s_i} and generate the set S^o with ordered slices.
iii) insert the first object of the first slice in S^o in the bottom left of the bins.
iv) allocate the current slice object in the bottom-most position leaving the largest unallocated space over time in each bin and minimizing the wasted space between objects of same bin.
v) keep inserting the objects of the current slice on all the bins with respect to iv until the required objects of the current slice are allocated on all the bins.
vi) repeat iv and v as to allocate the slices in sequential order as in S^o, until all the objects of each slice on each BS are allocated.

The first instructions of the total required resources computation and slices sorting run in $O(n_b * n_s + n_s log(n_s))$. Let denote ρ the total required tiles of all slices over all the gNBs, i.e. $\rho = \sum_{s_i \in S} \lambda_{s_i}$. The heuristic core code run in $O(n_b * \rho^2)$. This is because the packing time on each gNB is ρ^2. Consequently, the HSF converges with a time complexity of $O(n_b * \rho^2)$.

B. Heuristic 2: Iterative Minimum Allocation IMA

HSF allocates in sequential order all the tiles of a given slice over all the involved gNBs, starting with the slice requesting the highest total number of tiles over B.

Given that the slices request different amount of tiles over a set of gNBs, it is clear that the maximum tied sRBs between

Algorithm 1 Heuristic1- HSF

Input: B, S, Γ
Output: HSF sRBs allocation $G^{HSF} = (g_k^{HSF})_{k \in B}$
set $g_k^{HSF} = (\alpha_{k,f,t}^{s_i,\mu})_{f,t} = 0 \; k \in B \quad s_i \in S$
Compute $\Lambda = (\lambda_{s_i})_{s_i \in S}$
$S^o \leftarrow$ sort S in decreasing order based on Λ
for each BS $b_k \in B$ **do**
    for slice $s_i \in S^o$ **do**
        while $\gamma_{s_i,k} \neq 0$ **do**
            allocate s_i tile subsequent sRBs with LCUS account.
            update g_k^{HSF}
            remove the allocated object from $\gamma_{s_i,k}$
end

a subset of gNBs equals the minimum required resources over the same subset. For instance, let us consider a subset of three gNBs serving one slice s_1 with numerology $\mu = 0$, s_1 requests 3 tiles on b_1, 2 tiles on b_2 and 5 tiles b_3. The maximum tied sRBs over $B = \{b_1, b_2, b_3\}$ for s_1 equals 2 tiles, i.e. 8 sRBs as each tile is composed of 4 sRBs. Hence, to maximize the total tied sRBs over B we need to ensure the allocation of this minimum of s_1 8 sRBs over B. In a general case with multiple slices, the maximization of the total tied sRBs over a subset of gNBs implies the assurance at best that the minimum required sRBs for each slice is allocated in the same time/frequency position over the gNBs subset. Thus, with the aim to maximize the total tied sRBs over B, we propose an iterative allocation of the non null minimum required tiles of each slice over the involved BSs. We refer to such approach as an Iterative Minimum Allocation (IMA) approach.

Let denote m_{s_i} the non null minimum required objects for slice s_i over B. It is computed as follows: $m_{s_i} = \min_{\substack{k \in B \\ \gamma_k^{s_i} \neq 0}} \gamma_k^{s_i}$.

The IMA procedure works as follows:

i) compute the total required resources of all the slices over B, $\Lambda = (\lambda_{s_i})_{s_i \in S}$.
ii) sort the slices in decreasing order based on λ_{s_i} and generate the set S^o with ordered slices.
iii) Compute the minimum required objects for all slices, $s_i \in S^o$ over all B, $M = (m_{s_i})_{s_i \in S^o}$.
iv) allocate m_{s_i} sequentially with leaving the LCUS over each bin.
v) update $\gamma_{s_i,k}$ by subtracting m_{s_i}.
vi) repeat iii, iv and v for all $s_i \in S^o$. If $\gamma_{s_i,k} = 0$. Remove s_i from b_k.
vii) If $\Gamma = 0$, stop. Otherwise, repeat vi until all the slices are assigned the required tiles over B.

C. Heuristic 3: Highest Minimum First HMF

With IMA, the slices are sorted in decreasing order based on the total required resources over the involving BSs. As the key to reach the maximum TTR is the allocation of the minimum required sRBs on the same position over B for each slice, we propose then in HMF to sort the slices in decreasing order based on the minimum required resources over the gNBs set. Therefore, the minimum is computed at each iteration and the algorithm proceeds as follows:

i) Compute the minimum required tiles for all slices, $s_i \in S^o$ over all B, $M = (m_{s_i})_{s_i \in S}$.
ii) sort the slices in decreasing order based on $m_{s_i s_i \in S}$ and generate the set S^o with ordered slices.
iii) allocate m_{s_i} sequentially with leaving the LCUS over each bin.
iv) update $\gamma_{s_i,k}$ by subtracting m_{s_i}.
v) repeat the steps from i until iv for all $s_i \in S^o$. If $\gamma_{s_i,k} = 0$. Remove s_i from b_k.
vi) If $\Gamma = 0$, stop. Otherwise, repeat v until all the slices are assigned the required tiles over B.

HMF and IMA are implemented with time complexity of $O(n_b * \rho^2)$.

Algorithm 2 Heuristic2-IMA

Input: B, S, Γ
Output: IMA sRBs allocation $G^{IMA} = (g_k^{IMA})_{k \in B}$
set $g_k^{IMA} = (\alpha_{k,f,t}^{s_i,\mu})_{f,t} = 0 \; k \in B \quad s_i \in S$
Compute $\Lambda = (\lambda_{s_i})_{s_i \in S}$
$S^o \leftarrow$ sort S in decreasing order based on Λ
while $\Gamma \neq 0$ **do**
    Compute $M = (m_{s_i})_{s_i \in S^o}$
    for `each BS` $b_k \in B$ **do**
        for `each slice` $s_i \in S^o$ **do**
            add m_{s_i} to the allocation with LCUS
            Update g_k^{IMA} by allocating m_{s_i} tiles subsequent sRBs
            remove the allocated objects from $\gamma_{s_i,k}$
        Update Γ by removing the allocated m_{s_i} objects from $\gamma_{s_i,k}$
        if $\gamma_{s_i,k} = 0$ **then**
            remove s_i request in b_k
end

D. TTR computation

Once the allocation is performed, an evaluation of the total amount of tied sRBs, the largest continuous unallocated space in a grid and convergence time is prominent. The LCUS and CT are computed as explained in section V-A. With ESRP models, the TTR was exactly the objective score. With the heuristics, the computation of TTR is required once the allocation finishes. For that, we propose an algorithm to count the total tied sRBs over the gNBs.

An sRB is considered as tied if it is allocated to same slice over the involved gNBs. In fact, each slice requests a different amount of resources on each gNB, the maximum tied sRBs between a given subset of gNBs is then equal to the minimum required resources over the same subset. An example includes, a slice s_1 that requires 2 tiles (8 sRBs) on gNB b_1 and 1 tile (4 sRBs) on gNB b_2. If the allocation is optimal, we will have at maximum 1 tied tile for s_1 over b_1 and b_2, i.e. 4 tied sRBs.

Algorithm 3 Heuristic3- HMF

Input: B, S, Γ
Output: HMF sRBs allocation $G^{IMA} = (g_k^{IMA})_{k \in B}$
set $g_k^{IMA} = (\alpha_{k,f,t}^{s_i,\mu})_{f,t} = 0 \; k \in B \quad s_i \in S$
while $\Gamma \neq 0$ **do**
    Compute $M = (m_{s_i})_{s_i \in S^o}$
    $S^o \leftarrow$ sort S in decreasing order based on M
    for `each BS` $b_k \in B$ **do**
        for `each slice` $s_i \in S^o$ **do**
            add m_{s_i} to the allocation with LCUS
            Update g_k^{IMA} by allocating m_{s_i} tiles subsequent sRBs
            remove the allocated objects from $\gamma_{s_i,k}$
        Update Γ by removing the allocated m_{s_i} objects from $\gamma_{s_i,k}$
        if $\gamma_{s_i,k} = 0$ **then**
            remove s_i request in b_k
end

From that, given the resource grid with complete allocation, $G_c = g_k^c$, we propose to count the total tied sRBs as summarized in algorithm 4 and explained in the following for each slices$s_i \in S$.

i) compute the total required sRBs for each slice, $Max(s_i) = \sum_{k \in B} \gamma_{s_i,k}^{\mu}$.
ii) select the gNBs where the slice requests the resources, noted B_{s_i}
iii) compute the minimum required resources over B_{s_i}, i.e. $Min(s_i) = \min_{k \in B_{s_i}} \gamma_{s_i,k}^{\mu}$.
iv) compute the tied sRBs from $Min(s_i)$ without redundancy.
v) update $Max(s_i)$ and repeat from step 2.
vi) repeat from 2 until the total required sRBs is reached or there is no minimum sRBs between any gNBs to be tied.

Algorithm 4 Total tied sRBs (TTR)

Input: G_c, S, B, Γ.

Output: Total tied sRBs over B.
for `each slice` $s_i \in S$ **do**
    Compute $Max(s_i) = \sum_{k \in B} \gamma_{s_i,k}^{\mu}$
    Compute the $Min(s_i) = \min_{k \in B} \gamma_{s_i,k}^{\mu}$
    while $Max(s_i) \geq 0$ and $Min(s_i) \neq 0$ **do**
        $B_{s_i} \leftarrow \{b_k, k \in B$ where $\gamma_{s_i,k}^{\mu} \neq 0\}$
        $Min(s_i) = \min_{k \in B_{s_i}} \gamma_{s_i,k}^{\mu}$
        count θ_j from $Min(s_i)$ and check the non redundancy.
        update $Max(s_i) \leftarrow Max(s_i) - 4 * Min(s_i)$
    $\Theta_{s_i} = \sum_{j \in \zeta_{s_i}} \theta_j$
$\chi = \sum_{s_i \in S} \Theta_{s_i}$

This algorithm is validated based on ESRP models. For all the instances of ESRP simulations, the achieved optimal TTR by ESRP is compared to the one computed by algorithm 4.

VII. Heuristics Evaluation

In this section, the performance evaluation of the three heuristics is fulfilled. Mainly, a comparison based on CT, TTR and LCUS is conducted. For that, the heuristics achieved TTR is compared with the optimal TTR given by ESRP models. In section V, it is shown that quite similar results are given by both ESRP versions and that ESRP-v1 outperforms smoothly ESRP-v2. Only ESRP-v1 scores are then considered in this part. Regarding the LCUS, although the skyline has reached the optimal LCUS for the different B and S sizes, the comparison between the three heuristics LCUS is performed with respect to the topmost upper bound score (LTUB). Furthermore, for all evaluations, the impact of the gNBs number and the served slices is also investigated. The simulation environment and process is exactly as described in part V-B. Similar instances are used for both ESRP models and heuristics for each simulation run. The results are then averaged over all the simulations runs for each test.

A. TTR Analysis

Fig. 15 illustrates the TTR optimality gap achieved by the 3 algorithms as a function of B size. The optimality gap (OG) is obtained from the difference between the optimal score given by ESRP-v1 and the achieved score by a given algorithm divided by the optimal score. It refers to the gap between the reached score and the optimal one. IMA and HMF have quite similar results over the various B set sizes, i.e. their curves are superposing. They reach the optimality for lower B set sizes when serving different S sizes (see fig. 15), i.e. OG=0%. The HSF is at 30% from optimality in similar cases. Then, the three heuristics scores decrease proportionally to B size augmentation. Both algorithms outperform HSF for B sizes lower than 9 gNBs.

As for the impact of S size on TTR, fig. 16 shows the OG as function of S for two system sizes: 3 and 5 gNBs. The OG increases when moving from a system serving 3 slices to the one with 7 slices. Then, the OG is quite stable around 22% with 3 gNBs and 47% with 5 gNBs with both IMA and HMF. It can be concluded that HMF and IMA become insensitive to larger S set size starting from 7 slices. It is mainly the B set size that has an impact on the TTR score. The HSF has higher OG compared to HMF and IMA. Its OG score ranges between 47% and 74% with a system of 5 gNBs.

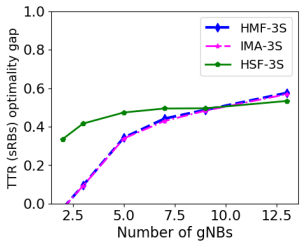


Fig. 15: TTR (sRBs) Optimality gap as a function of B size serving 3 slices.

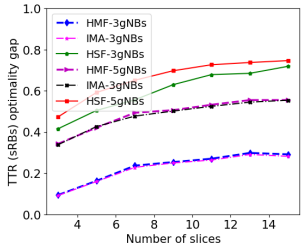


Fig. 16: TTR (sRBs) Optimality gap as a function of S for different B size.

Thus, IMA and HMF always outperform HSF in the case of 3 gNBs as well as 5 gNBs.Overall, such OG still remain acceptable as the three heuristics assure at best the cooperation enabling requirement while guarantying the three other requirements.

B. Convergence Time (CT) Analysis

Fig. 17 shows the convergence time of the three heuristics in seconds as function of B serving 3 slices. The three heuristics converge quickly at a time scale of hundreds of milliseconds. IMA and HMF have similar CT and slight difference is remarked with HSF.

The CT increases proportionally with B size growth. It expands gradually in the order of milliseconds. The CT is less than 10 ms for small B set size.

Regarding the S size impact on CT, fig. 18 plots the three heuristics CT as a function of S size with a system of 3 gNBs. The three heuristics converge in similar time granularities. The CT varies in small interval size and it is independent of the S size. Notably, for a system with 3 gNBs, the IMA CT ranges between 0.04 s and 0.11 s.

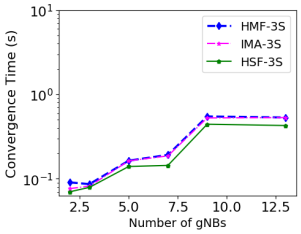


Fig. 17: CT (s) evaluation over varying B set size.

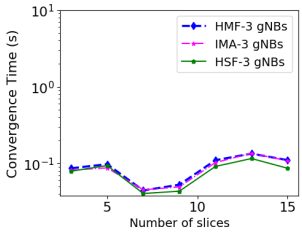


Fig. 18: CT (s) evaluation over varying S size.

C. Largest Continuous unallocated Space (LCUS) Analysis

Fig. 19 highlights the impact of S size on LCUS score in the system. The LCUS is computed over B as described in V-A3. The three heuristics achieve the optimal LCUS score given by LTUB. This is expected, as the skyline is used for the allocation. With S variation, the LCUS also varies in a small interval. This variation is independent of the S size evolution.

As for B size impact, fig. 20 plots the LCUS over each B set size. All heuristics achieve the optimal LCUS score over the different B set sizes, i.e. their curves are superposing. It is observed that larger B sets allows higher gain in terms of LCUS by all approaches. The LCUS score is then proportional to the B set size and insensitive to S size variation.

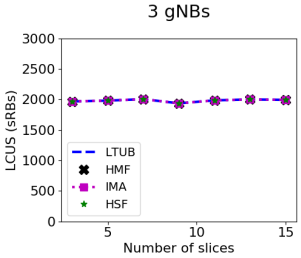


Fig. 19: LCUS (sRBs) as a function of S for a system with 3 gNBs.

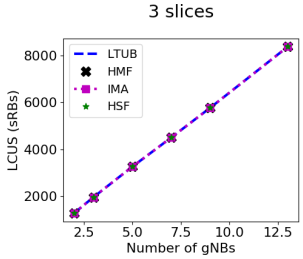


Fig. 20: LCUS (sRBs) as a function of B serving 3 slices.

D. Discussion

In order to enforce the real time RAN slicing, we proposed three heuristics, i.e. IMA, HMF and HSF that prioritize the LCUS while the TTR is achieved at best. An evaluation of the three heuristics is fulfilled based on CT, TTR and the LCUS. Contrarily to the ESRP models, the three heuristics converge at time scale of milliseconds. For lower B and S sizes, the CT is in the order of 10 ms. The CT increases smoothly with larger B set size. But, it doesn't outpace 0.7 s, 0.65 s, and 0.47s when tested with HMF, IMA and HSF respectively in a large B set of 13 gNBs serving 15 slices simultaneously. From that, these heuristics demonstrate their capability for real time deployment within the 5G SD-RAN.

Further, the HMF, HSF and IMA heuristics are compared to the optimal/upper bound solution given by ESRP-v1 for the TTR score. For the LCUS, the heuristics are compared with the upper bound LCUS (LTUB). For a small set of gNBs and slices, the IMA and HMF are highly enforcing the RAN slicing for real time system deployment by reaching the optimality for TTR and LCUS in very small time scale (in the order of 10 ms). This could be the case of macro cells deployment, as well as a small group of other cells type covering a specific geographical zone. Moreover, higher S size doesn't have a big impact on both IMA and HMF with respect to TTR, which is advantageous for the 5G RAN enforcement. Both heuristics performance degrade with larger B sets. But, the worst case study of 13 gNBs serving 15 slices, at least 32% of the optimal TTR is reached by both heuristics. This score might be higher as the ESRP doesn't converge within the time limit for this instances, and then the comparison is conducted with the upper bound TTR. Nevertheless, this score is still advantageous as the heuristics prioritize the LCUS at the expense of TTR. In fact, the highest B size produce the highest LCUS when applying both heuristics. It corresponds exactly to the optimal LCUS marked by the LCUS upper bound. Thus, they lead to an allocation without resources waste. This prioritization is intended because of the crucial task of efficient resource allocation required by the MNO.

In summary, although the ESRP models give the optimal allocation with higher total tied sRBs, their high convergence time and non assurance of resources efficient usage make their real time deployment questionable. The IMA, HMF and HSF heuristics achieve a good results in terms of TTR, CT and largest space for lower B sets. This proves the possibility of their real time deployment for such cases. The growth of B size allows a larger continuous unallocated space at the expense of TTR with all the developed heuristics. Even though this priority prospect, the TTR is assured at best by HSF, IMA and HMF. The HMF and IMA are outperforming the HSF. Thus, the slice owner could use the tied resources to enable the advanced transmission schemes for the critical transmissions. Moreover, all heuristics highly enforce the RAN slicing with respect to the four requirements. In fact, the orthogonality, satisfaction and scalability are guaranteed, while the enabling requirement is assured at best.

VIII. Conclusion

The RAN slicing comes with challenging requirements such as resources isolation, slices satisfaction, scalability and the cooperation enabling. In this work, we aimed to enforce it from resource perspective in the 5G context. For that, we have formulated the problem as a multi-objective optimization to allocate efficiently the slices resources with respect to the diverse RAN slicing requirements. The first objective addresses the scalability of the RAN slicing through the maximization of the largest continuous unallocated space on each gNB resource grid. Then, the second objective handles the cooperation enabling requirements by means of resource allocation in similar position over frequency and time for a given slice over the set of gNBs. The second objective involves a tight management of resources. Therefore, a resource grid decomposition is proposed as to have a fine grained resources monitoring. Both slices resources isolation and satisfaction are guaranteed by means of constraint in each objective.

With the multi-objective criterion, the optimal solution for each objective is targeted. Two mathematical models are developed for the first objective, whereas the second objective is tackled as a 2D bin packing optimization problem. An heuristic is then used to approximate rapidly the optimal score, as the problem is known to be NP-hard. The optimal models converge slowly, which limits their deployment for real time use cases. Nevertheless, they could be advantageous for the SD-RAN large scale decisions.

Therefore, three heuristics are implemented with the aim to enforce the allocation strategy for the RAN slicing. The scalability is prioritized with these heuristics at the expense of the enabling cooperation requirement. All the algorithms are evaluated in terms of convergence time, total tied resources and largest continuous unallocated space.

Contrarily to the optimal models, the developed heuristics, i.e. IMA, HMF and HSF achieve good results in different case studies. Especially, for lower set of gNBs, the IMA and HMF reach the optimal scores for both tied resources and largest unallocated continuous space with a very low convergence time in the order of 10 ms. In such case the four RAN slicing requirements are guaranteed. Moreover, all the tested algorithms show insensitivity to the number of served slices during the allocation window. Such results encourage the real time deployment test for the three approaches.

In this work, we have been mainly interested in the allocation over a cluster of gNBs controlled by one SD-RAN. Our future work concerns the RAN slicing enforcement in multi-SD-RAN multi-cell deployment. In such case, a coordination or a cooperation between SD-RANs should be investigated for large resources allocation with respect to slicing requirements.

References

[1] L. J. B. Han and H. D. Schotten. "slice as an evolutionary service: Ge-netic optimization for inter-slice resource management in5g networks,"ieee access, vol. 6, no. 1. *p*, 33:137, 2018.

[2] C.-Y. Chang, N. Nikaein, and T. Spyropoulos. "radio access network resource slicing for flexible service execution. In *" in IEEE INFOCOM 2018- IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE pp. 668–673, 2018.

[3] A. Devlic, A. Hamidian, D. Liang, M. Eriksson, A. Consoli, and J. Lundstedt. Nesmo: Network slicing management and orchestration framework. In *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, page 1202–1208. IEEE, May 2017.

[4] S. D'Oro, F. Restuccia, A. Talamonti, and T. Melodia. The slice is served: Enforcing radio access network slicing in virtualized 5g systems. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, page 442–450. IEEE, Apr 2019.

[5] S. E. Elayoubi, S. B. Jemaa, Z. Altman, and A. Galindo-Serrano. 5g ran slicing for verticals: Enablers and challenges. *IEEE Communications Magazine*, 57(1):28–34, Jan 2019.

[6] R. Ferrus, O. Sallent, J. Perez-Romero, and R. Agusti. On 5g radio access network slicing: Radio interface protocol features and configuration. *IEEE Communications Magazine*, 56(5):184–192, May 2018.

[7] X. Foukas, M. K. Marina, and K. Kontovasilis. Orion: Ran slicing for a flexible and cost-effective multi-service mobile network architecture. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking - MobiCom '17*, page 127–140. ACM Press, 2017.

[8] M. R. Garey and D. S. Johnson. *Computers and intractability: a guide to the theory of NP-completeness*. A series of books in the mathematical sciences. Freeman, 27. print edition, 2009.

[9] A. A. Gebremariam, M. Chowdhury, M. Usman, A. Goldsmith, and F. Granelli. Softslice: Policy-based dynamic spectrum slicing in 5g cellular networks. In *2018 IEEE International Conference on Communications (ICC)*, page 1–6. IEEE, May 2018.

[10] J. He and W. Song. Appran: Application-oriented radio access network sharing in mobile networks. In *2015 IEEE International Conference on Communications (ICC)*, page 3788–3794. IEEE, Jun 2015.

[11] L. He, X. Ren, Q. Gao, X. Zhao, B. Yao, and Y. Chao. The connected-component labeling problem: A review of state-of-the-art algorithms. *Pattern Recognition*, 70:25–43, Oct 2017.

[12] J. Jylänki. A thousand ways to pack the bin – a practical approach to two-dimensional rectangle bin packing, 2010.

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

[14] R. Mahindra, M. A. Khojastepour, H. Zhang, and S. Rangarajan. Radio access network sharing in cellular networks. In *2013 21st IEEE International Conference on Network Protocols (ICNP)*, page 1–10, Oct 2013.

[15] S. Mandelli, M. Andrews, S. Borst, and S. Klein. Satisfying network slicing constraints via 5g mac scheduling. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, page 2332–2340. IEEE, Apr 2019.

[16] Z. A. Qazi, M. Walls, A. Panda, V. Sekar, S. Ratnasamy, and S. Shenker. A high performance packet core for next generation cellular networks. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication - SIGCOMM '17*, page 348–361. ACM Press, 2017.

[17] O. Sallent, J. Perez-Romero, R. Ferrus, and R. Agusti. On radio access network slicing from a radio resource management perspective. *IEEE Wireless Communications*, 24(5):166–174, Oct 2017.

[18] 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 *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, page 1–9. IEEE, May 2017.

[19] M. Wallace. Practical applications of constraint programming. *Constraints*, 1(1–2):139–168, Sep 1996.

[20] M. Yan, G. Feng, J. Zhou, Y. Sun, and Y.-C. Liang. Intelligent resource scheduling for 5g radio access network slicing. *IEEE Transactions on Vehicular Technology*, 68(8):7691–7703, Aug 2019.

[21] M. Zambianco and G. Verticale. Interference minimization in 5g physical-layer network slicing. *IEEE Transactions on Communications*, 68(7):4554–4564, 2020.