This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LWC.2018.2842189, IEEE Wireless Communications Letters

Slicing the Edge: Resource Allocation for RAN Network Slicing

Phuong Luu Vo, Minh N. H. Nguyen, Tuan Anh Le, Nguyen H. Tran, *Member, IEEE*

***Abstract*—Network slicing is considered to be a crucial feature of 5G cellular system. By dividing the network infrastructure into multiple logical segments, network slicing can support parallel services with different requirements. While technology developments focus on slicing the core networks, there are limited studies in network slicing for radio access networks (RAN). Hence, in this work, we study the RAN slicing and slice coordination, which is formulated as a bi-convex problem. Even though there are complicated couplings between the RAN resource allocation for each slice and the coordination of the slices that share the same resources, we design two algorithms addressing these couplings of this bi-convex problem. Simulation results validate the efficacy of our proposed algorithms.**

I. Introduction

Network slicing is emerging as one of the key elements to enable the fifth generation (5G) cellular networks. Specifically, with network slicing, the network infrastructure can be segmented into multiple logical network slices to support various services with different requirements. For example, one network slice is dedicated to augmented reality applications with ultra-reliable and low-latency communications, whereas another slice is designated for extremely high throughput video-on-demand services. With those benefits, network slicing has attracted interest from various mobile operators, e.g., SK Telecom and Ericsson demonstrated how network slices were created to support augmented reality in 2015 whereas Ericsson and NTT DoCoMo provided a dynamic network slicing proof-of-concept for 5G core networks in 2016 [1].

Recently, a number of comprehensive surveys have discussed different 5G network slicing architectures and its applications, including vehicular networks, information-centric networks, isolated wireless network slicing, etc., [2]–[5]. However, these works do not consider the problems of *slice allocation and coordination*, which are equally important to bring network slicing into fruition. Thus, in this work, we propose slicing algorithms that can perform radio access network (RAN) slicing (including base station (BS) bandwidth, caching, and backhaul capacties) and handle the coordination between these slices. While BS radio bandwidth is inarguably a critical resource of RAN, in 5G cellular system, caching and backhaul will play important roles because i) the backhaul link will be bottlenecked due to 5G network densification, and ii) caching at BS can reduce the traffic to the core network. However, cache storage and backhaul link are mostly ignored in RAN slicing. Furthermore, in order to support a wide range of segmented services, the coordination of all created slices to optimize the aggregated slice performance is also a critical issue, which is, unexpectedly, under-explored.

We see that none of the existing works consider the *RAN slicing of jointly BS radio bandwidth, caching, and backhaul* components. This creates a unique challenge: how to allocate and coordinate multiple RAN slices for tenants such that the aggregated load of all allocated slices and the backhaul delay are minimized, considering each physical resource has its own capacity constraint. This challenge imposes two interdependences: while the first one is between the BS radio bandwidth and caching slice allocation to minimize the aggregate BS slice load, the second one is between the caching and backhaul slice allocation to minimize the aggregate backhaul delay.

By tackling these challenges, our contributions can be summarized as follows:

- We first formulate a network slicing design as an optimization problem, which is further shown to be a bi-convex problem, that accounts for both RAN vertical slicing and slice coordination issues. To solve this problem, we propose two alternative minimization algorithms for this bi-convex problem. While the first is a centralized scheme that can be supported by 5G network virtualization and orchestration tools [2], the second one is a distributed approach where all slices coordinate to reach the solution.
- Even though our alternative minimization approaches have no global convergence guarantee, simulation results show that they can achieve a global optimal solution (compared with exhaustive search) in a simple scenario with two tenants. When the number of tenants is increased, the results show that our algorithms can converge to the same performance as that of IpOPt solver, validating the efficacy of our proposed approaches.

P. Vo is with School of Computer Science and Engineering, International University-VNUHCM, Vietnam (email: vtlphuong@hcmiu.edu.vn). This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.02-2015.36.

TA. Le is with Faculty of Engineering and Technology, Thu Dau Mot University, Vietnam (email: letuanh@tdmu.edu.vn).

M.N.H. Nguyen is with the Department of Computer Science and Engineering, Kyung Hee University, Korea (email: minhnhn@khu.ac.kr).

N.H. Tran is with the School of Information Technologies, The University of Sydney, Sydney, NSW 2006, Australia (email: nguyen.tran@sydney.edu.au).

II. System Model

A. Network Model

We consider a network slicing architecture where a set $\mathcal{N}$ of tenants (e.g., services) sharing a physical network resources, including a BS with a cache storage and capacity-constrained backhaul link as in Fig. 1. The BS radio bandwidth, cache storage, and backhaul link are denoted by W (MHz), S (contents), and B (Mbps), respectively.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LWC.2018.2842189, IEEE Wireless Communications Letters

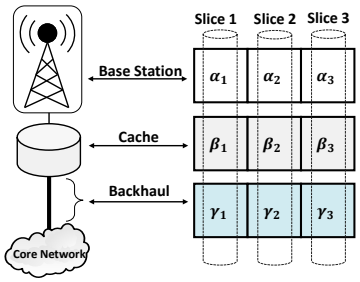

Fig. 1: A RAN network slicing model with three tenants.

These network resources will be "sliced" according to the demand of each tenant. Specifically, we define a slice allocated to a tenant n by a tuple $(\alpha_n, \beta_n, \gamma_n)$, which is a fraction of (W, S, B), respectively, such that $\sum_{n\in\mathcal{N}} \alpha_n = \sum_{n\in\mathcal{N}} \beta_n = \sum_{n\in\mathcal{N}} \gamma_n = 1$. We denote the vector $(\alpha_n)_{n\in\mathcal{N}}$ by $\boldsymbol{\alpha}$, similarly for $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$.

Each tenant n has its own set of users $\mathcal{U}_n$. Users of all tenants request the contents from a catalog with a set $\mathcal{F}$ of contents with equal size L[1]. Furthermore, for a tenant n with its user $u_n \in \mathcal{U}_n$, content access pattern of u_n is modeled by a distribution $p_{u_n,f}$ such that $\sum_{f\in\mathcal{F}} p_{u_n,f} = 1$, $\forall u_n \in \mathcal{U}_n$.

We consider a timeslot model where slicing decision is updated according to the demand of tenants' users (e.g., 10 minutes). In one timeslot, the tenant n's user request rate for contents is denoted by λ_{u_n}, $\forall u_n \in \mathcal{U}_n$.

B. Cache Slicing

According to [6], by using the reset time-to-live (TTL) cache model, the hit probability of content f of tenant n is

$$h_{n,f} = 1 - e^{-\sum_{u_n\in\mathcal{U}_n} \lambda_{u_n,f} T_{n,f}} \tag{1}$$

where $\lambda_{u_n,f} := p_{u_n,f}\lambda_{u_n}$, and $T_{n,f}$ is the characteristic time of content f of tenant n, which can be considered as timer in TTL caching. Because the expected number of contents of a tenant must not exceed its allocated cached capacity, we have

$$\sum\nolimits_{f\in\mathcal{F}} h_{n,f} = \beta_n S, \quad \forall n \in \mathcal{N}. \tag{2}$$

Since the number of contents is very large, the request rate to each content of tenant n can be small. Thus, we approximately have

$$h_{n,f} = \sum\nolimits_{u_n\in\mathcal{U}_n} \lambda_{u_n,f} T_{n,f}, \quad \forall n \in \mathcal{N}, \tag{3}$$

since $e^{-x} = 1 - x$ when x is small enough.

Tenants usually have no content discrimination; thus, all of their contents have the same characteristic time: $T_{n,f} = T_n$, $\forall f$. Therefore, from (2) and (3), we have

$$h_{n,f} = \frac{\sum_{u_n\in\mathcal{U}_n} \lambda_{u_n,f}}{\sum_{u_n\in\mathcal{U}_n} \lambda_{u_n}} \beta_n S, \quad \forall n \in \mathcal{N}. \tag{4}$$

There are two implications from (4): i) the hit probability of each content increases when the cache space increases, and ii) the hit probability is proportional to the request rate. Furthermore, the content with a higher request rate (i.e., more popular) has a higher hit probability.

[1]Heterogeneous content sizes can be divided into multiple content chunks with the same size.

C. BS Bandwidth Slicing

When the tenant n receives its BS slicing α_n, the BS bandwidth it receives is $\alpha_n W$. With this bandwidth, the downlink rate of a tenant n's user is as follows

$$c_{u_n} = \alpha_n W \log(1 + SNR_{u_n}), \quad \forall u_n \in \mathcal{U}_n, \tag{5}$$

where SNR_{u_n} is user u_n's received signal-to-noise ratio. The SNR_{u_n} can be given as

$$SNR_{u_n} = \frac{g_{u_n} P_{u_n}}{\sigma_{u_n}^2 + I_{u_n}}, \tag{6}$$

where g_{u_n} captures the channel gain (i.e., path loss, shadowing, etc.) from the BS to the location of user u_n, P_{u_n} is the BS transmission power, $\sigma_{u_n}^2$ denotes the background noise, and I_{u_n} denotes the other average interference sources (e.g., coming from neighboring BSs) seen by user u_n. We assume that all users of tenant n will be scheduled in a time-shared manner (e.g., TDMA) so there is no inter-user interference.

From (4), the hit rate and missed rate of the user u_n are defined respectively as follows

$$\lambda_{u_n}^{hit} = \sum\nolimits_f \lambda_{u_n,f} h_{n,f} = \sum\nolimits_f \lambda_{u_n,f} \frac{\sum_{u_n\in\mathcal{U}_n} \lambda_{u_n,f}}{\sum_{u_n\in\mathcal{U}_n} \lambda_{u_n}} \beta_n S \tag{7}$$

$$\lambda_{u_n}^{mis} = \lambda_{u_n} - \lambda_{u_n}^{hit}, \quad \forall u_n \in \mathcal{U}_n. \tag{8}$$

When a request to a content f of tenant n is hit (i.e., the content is found in the cache storage), this request can be served by the BS slice of tenant n immediately. Therefore, the load of the BS slice n is

$$\rho_n(\alpha_n, \beta_n) = \sum\nolimits_{u_n\in\mathcal{U}_n} \frac{\lambda_{u_n}^{hit}}{\mu_{u_n}}, \tag{9}$$

where $\mu_{u_n} = c_{u_n}/L$ is the service rate of user u_n's content.

D. Backhaul Capacity Slicing

When the requests are missed, they will be forwarded to the content server and downloaded through the backhaul link. Given the backhaul capacity slice $\gamma_n B$, the backhaul link's delay cost of the tenant n is modeled by using M/M/1 queueing delay [7] as follows

$$\Phi_n(\beta_n, \gamma_n) := \begin{cases} \frac{\lambda_n^{mis}}{\mu_n^{mis} - \lambda_n^{mis}}, & 0 \le \lambda_n^{mis} < \mu_n^{mis}; \\ \infty, & \text{otherwise}, \end{cases} \tag{10}$$

where $\lambda_n^{mis} := \sum_{u_n\in\mathcal{U}_n} \lambda_{u_n}^{mis}$, and $\mu_n^{mis} = \gamma_n B/L$ is the backhaul service rate for tenant n.

III. RAN Network Slicing

In this section, we first formulate the RAN slicing problem, then propose two algorithms for this problem.

A. Problem Formulation

The RAN network slicing is formulated as follows

$$\min_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}\ge 0} \quad \sum\nolimits_{n\in\mathcal{N}} \rho_n(\alpha_n, \beta_n) + \omega \sum\nolimits_{n\in\mathcal{N}} \Phi_n(\beta_n, \gamma_n) \tag{11}$$

$$\text{s.t.} \quad \sum\nolimits_{n\in\mathcal{N}} \alpha_n = \sum\nolimits_{n\in\mathcal{N}} \beta_n = \sum\nolimits_{n\in\mathcal{N}} \gamma_n = 1, \tag{12}$$

$$\rho_n(\alpha_n, \beta_n) \le 1 - \epsilon, \qquad \forall n \in \mathcal{N}. \tag{13}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LWC.2018.2842189, IEEE Wireless Communications Letters

In this problem, the objective is to minimize two components: i) the aggregated BS slice load for load-balancing, and ii) the backhaul link cost of all slices for delay minimization, where ω is a weight factor which characterizes the Pareto optimal values of these two objectives. This objective captures two interdependences i) between the BS radio bandwidth and caching slice allocation to minimize the aggregate slice load, and ii) between the caching and backhaul slice allocation to minimize the aggregate backhaul delay. Constraint (13) prevents an over-loaded tenant's BS slice with a pre-determined threshold ϵ.

We see that

$$\rho_n(\alpha_n, \beta_n) = \sum\nolimits_{u_n \in \mathcal{U}_n} \frac{\lambda_{u_n}^{hit}}{\mu_{u_n}} = \theta_n \frac{\beta_n}{\alpha_n}, \tag{14}$$

where $\theta_n := \sum_{u_n \in \mathcal{U}_n} \frac{\sum_{f \in \mathcal{F}} \lambda_{u_n,f} \frac{\sum_{u_n \in \mathcal{U}_n} \lambda_{u_n,f}}{\sum_{u_n \in \mathcal{U}_n} \lambda_{u_n}} S}{c_{u_n}/L}$. Thus, the objective (11) can be rewritten as follows

$$\sum\nolimits_{n \in \mathcal{N}} \theta_n \frac{\beta_n}{\alpha_n} + \omega \sum\nolimits_{n \in \mathcal{N}} \Phi_n(\beta_n, \gamma_n). \tag{15}$$

We observe that (11) is not a convex, but a bi-convex problem: Although it is not jointly convex of all variables, given β_n, it is convex with respect to both α_n and γ_n, $\forall n \in \mathcal{N}$, and vice versa. Thus, we next propose two alternative minimization algorithms solving this bi-convex problem.

B. Algorithms

The first algorithm is centralized. This is inspired by the emerging software-defined network orchestration tools which can maintain a global state information to centrally implement network slicing [2].

1) Centralized Approach: We adopt the Proximal Block Coordinate Descent (PBCD) method presented in [8] for solving a general multi-convex problem.

At each iteration indexed by k, given the previously achieved $\boldsymbol{\alpha}^{(k)}$ and $\boldsymbol{\gamma}^{(k)}$, a controller will solve the following convex problem to obtain $\boldsymbol{\beta}^{(k+1)}$ as follows

$$\min_{\boldsymbol{\beta} \geq 0} \quad \sum\nolimits_{n \in \mathcal{N}} \theta_n \frac{\beta_n}{\alpha_n^{(k)}} + \omega \sum\nolimits_{n \in \mathcal{N}} \Phi_n(\beta_n, \gamma_n^{(k)}) + \frac{\rho}{2} \|\beta - \beta^{(k)}\|^2 \tag{16}$$

$$\text{s.t.} \quad \sum\nolimits_{n \in \mathcal{N}} \beta_n = 1, \tag{17}$$

$$\theta_n \beta_n - (1-\epsilon)\alpha_n^{(k)} \leq 0, \quad \forall n \in \mathcal{N}. \tag{18}$$

Then, based on this $\boldsymbol{\beta}^{(k+1)}$, the controller updates $\boldsymbol{\alpha}^{(k+1)}, \boldsymbol{\gamma}^{(k+1)}$ by solving the following convex problem

$$\min_{\boldsymbol{\alpha}, \boldsymbol{\gamma} \geq 0} \quad \sum\nolimits_{n \in \mathcal{N}} \theta_n \frac{\beta_n^{(k+1)}}{\alpha_n} + \omega \sum\nolimits_{n \in \mathcal{N}} \Phi_n(\beta_n^{(k+1)}, \gamma_n) + \frac{\rho}{2} \|\alpha - \alpha^{(k)}\|^2 + \frac{\rho}{2} \|\gamma - \gamma^{(k)}\|^2 \tag{19}$$

$$\text{s.t.} \quad \sum\nolimits_{n \in \mathcal{N}} \alpha_n = \sum\nolimits_{n \in \mathcal{N}} \gamma_n = 1, \tag{20}$$

$$\theta_n \beta_n^{(k+1)} - (1-\epsilon)\alpha_n \leq 0, \quad \forall n \in \mathcal{N}, \tag{21}$$

The algorithm will alternatively solve (16) and (19) repeatedly until convergence [8].

For this approach, the controller, which can be the Network Element Manager on the shared BS as proposed in [3], has to collect all information of tenants. When the tenant private information, e.g., hit/miss rates, is unobtainable, a decentralized scheme is needed to implement the RAN network slicing problem (11).

2) Decentralized Approach: We next design a decentralized algorithm based on the Jacobi-Proximal Alternating Direction Method of Multipliers (JP-ADMM) method in [9]. Although there are several decentralized optimization algorithms such as dual decomposition and ADMM, we adopt this JP-ADMM method, which is a variant of ADMM, because it i) converges faster than dual decomposition method, and ii) exploits the parallelized structure of subproblem [9].

At each iteration k, given the announced dual variables $(u_1^{(k)}, u_2^{(k)}, u_3^{(k)})$ and aggregated values $\sum_{n \in \mathcal{N}} \beta_n^{(k)}$, $\sum_{n \in \mathcal{N}} \alpha_n^{(k)}$, $\sum_{n \in \mathcal{N}} \gamma_n^{(k)}$ of previous iteration, at the *first* step, each tenant n only needs to solve its own problem as follow: First, each tenant n updates $\beta_n^{(k+1)}$ by solving

$$\min_{\beta_n \geq 0} \quad \theta_n \frac{\beta_n}{\alpha_n^{(k)}} + \omega\, \Phi_n(\beta_n, \gamma_n^{(k)}) + \frac{\tau_n}{2}(\beta_n - \beta_n^{(k)})^2 + \frac{\rho}{2}\Big(\beta_n + \sum\nolimits_{i \neq n} \beta_i^{(k)} - 1 - u_1^{(k)}/\rho\Big)^2 \tag{22}$$

$$\text{s.t.} \quad \text{Contraint (18).}$$

Then, with this $\beta_n^{(k+1)}$, the tenant n updates $(\alpha_n^{(k+1)}, \gamma_n^{(k+1)})$ by solving

$$\min_{\alpha_n, \gamma_n \geq 0} \theta_n \frac{\beta_n^{(k+1)}}{\alpha_n} + \omega\, \Phi_n(\beta_n^{(k+1)}, \gamma_n) + \frac{\tau_n}{2}(\alpha_n - \alpha_n^{(k)})^2 + \frac{\rho}{2}\Big(\alpha_n + \sum\nolimits_{i \neq n} \alpha_i^{(k)} - 1 - \frac{u_2^{(k)}}{\rho}\Big)^2 + \frac{\tau_n}{2}(\gamma_n - \gamma_n^{(k)})^2 + \frac{\rho}{2}\Big(\gamma_n + \sum\nolimits_{i \neq n} \gamma_i^{(k)} - 1 - \frac{u_3^{(k)}}{\rho}\Big)^2$$

$$\text{s.t.} \quad \text{Contraint (21).}$$

At the *second* step, the dual variables can be updated by the controller as follows

$$u_i^{(k+1)} = u_i^{(k)} - \sigma\rho\Big(\sum\nolimits_{n=1}^{N} x_{i,n}^{(k+1)} - 1\Big), i = 1,2,3, \tag{23}$$

where $x_{i,n}^{(k+1)}$ denotes $\beta_n^{(k+1)}, \alpha_n^{(k+1)}, \gamma_n^{(k+1)}$ when $i = 1,2,3$, respectively. Then, dual variables and the aggregated values $\sum_{n \in \mathcal{N}} \beta_n^{(k+1)}$, $\sum_{n \in \mathcal{N}} \alpha_n^{(k+1)}$, and $\sum_{n \in \mathcal{N}} \gamma_n^{(k+1)}$ will be announced to all tenants. The signaling of the proposed schemes can be implemented via Type 5 interface as in [3]. When τ_n and σ are chosen according to [9], the above two steps are repeatedly performed until convergence.

IV. Case Studies

For simulation settings, we consider an area of 500m x 500m where the BS is at the center and user locations are uniformly distributed. We consider four tenants, which have the number of users follows a uniform distribution within $[10, 30]$, and users of each tenant have the same request rates. The BS transmission power P_{u_n} is 49 dBm and the bandwidth W is 20 MHz. Furthermore, we consider 50,000 contents with fixed size 100KB, the content popularity with Zipf distribution, and ω is set to 1. In the following, we consider two scenarios

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LWC.2018.2842189, IEEE Wireless Communications Letters

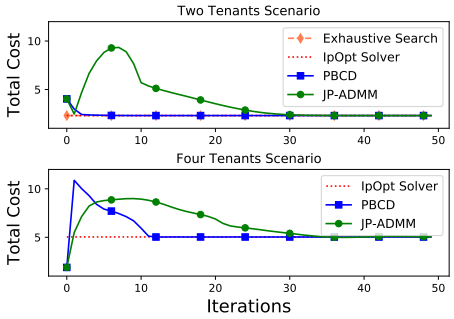

Fig. 2: Cost convergence.

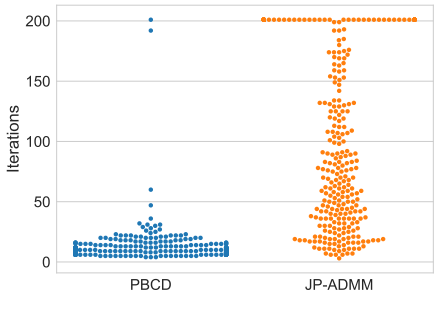

Fig. 3: Iteration convergence.

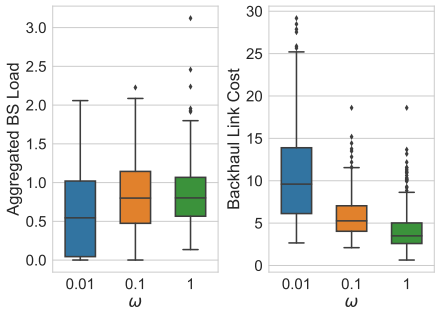

Fig. 4: Cost trade-off with varying ω.

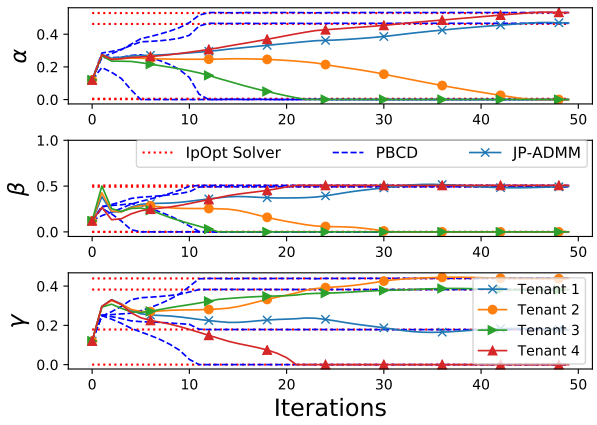

Fig. 5: Slicing allocation of four tenants.

of two tenants and four tenants, in which the cache capacity (number of files) and backhaul capacity (Mbps) are 220 and 800 for the first scenario, and 220 and 1200 for the second scenario, respectively. We use Julia to implement our proposed algorithms.

Even though our proposed algorithms does not theoretically guarantee the global optimum for the bi-convex problem (11), Fig. 2 shows that both algorithms can numerically converge to a tenant's global optimal cost, compared with an exhaustive search and IpOpt solver in a simple scenario of two tenants. In the second scenario with four tenants, since exhaustive search is impractical in this case, we compare the total cost convergence of the proposed schemes with that of IpOpt in the lower plot of Fig. 2, which shows that all have the same convergent value. Also, the convergence of BS, cache, and backhaul slicing is shown in Fig. 5. We see that Tenant 1 and 4 with higher traffic demands use most of the BS and cache capacities while the remaining tenants will get contents via backhaul link by using 38% and 44% backhaul capacity. Although Tenant 1 receives 46% and 49% of BS and cache capacities, respectively, it still needs an additional 17% of backhaul capacity for its high demand.

For statistical distribution of convergent iterations sampled from 300 simulations as shown in Fig. 3, we see that the PBCD has faster convergence than the JP-ADMM scheme. However, JP-ADMM surpasses PBCD in twofold: First, the size of the optimization problem in each iteration of PBCD is larger than that of JP-ADMM. Second, PBCD is a centralized approach in which the controller needs all the tenants' private information, whereas with distributed algorithm JP-ADMM, the controller only needs to observe the current physical resource load. Finally, in Fig 4, we study the trade-off between tenant load and backhaul cost in four-tenant scenario by tuning ω. This table shows that lower values of ω give lower tenant load and higher backhaul cost, and vice versa.

V. Conclusions

We studied the network slicing for multiple services using RAN resources such as BS, cache storage, and backhaul capacity. The considered RAN network slicing is formulated as a bi-convex problem that accounts for dependencies between resource allocation for each slice and coordination of slices sharing the same resources. We proposed two algorithms for solving the formulated bi-convex problem and simulation results show that the proposed algorithms can converge to the solution validated by other baselines.

References

[1] J. P. Tomas, "Understanding network slicing, a key technology for 5G," 2017. [Online]. Available: http://www.rcrwireless.com/20170106/internet-of-things/network-slicing-5g-tag23-tag99

[2] A. Nakao, P. Du, Y. Kiriha, F. Granelli, A. A. Gebremariam, T. Taleb, and M. Bagaa, "End-to-end Network Slicing for 5G Mobile Networks," *Journal of Information Processing*, vol. 25, pp. 153–163, 2017.

[3] G. Tseliou, K. Samdanis, F. Adelantado, X. C. Perez, and C. Verikoukis, "A capacity broker architecture and framework for multi-tenant support in LTE-A networks," in *IEEE ICC*. IEEE, may 2016, pp. 1–6.

[4] J. S. Panchal, R. D. Yates, and M. M. Buddhikot, "Mobile Network Resource Sharing Options: Performance Comparisons," *IEEE Transactions on Wireless Communications*, vol. 12, no. 9, pp. 4470–4482, sep 2013.

[5] M. Richart, J. Baliosian, J. Serrat, and J.-L. Gorricho, "Resource Slicing in Virtual Wireless Networks: A Survey," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 462–476, sep 2016.

[6] D. L. Fofack, N. Choungmo and Dehghan, M. and Towsley, D. and Badov, M. and Goeckel, "On the Performance of General Cache Networks," in *ACM ValueTools*, Bratislava, Slovakia, 2014, pp. 106–113.

[7] T. Han and N. Ansari, "Network Utility Aware Traffic Load Balancing in Backhaul-Constrained Cache-Enabled Small Cell Networks with Hybrid Power Supplies," *IEEE Transactions on Mobile Computing*, vol. 16, no. 10, pp. 2819–2832, Oct. 2017.

[8] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *SIAM Journal on imaging sciences*, vol. 6, no. 3, pp. 1758–1789, 2013.

[9] W. Deng, M.-J. Lai, Z. Peng, and W. Yin, "Parallel multi-block ADMM with o(1/k) convergence," *Journal of Scientific Computing*, vol. 71, no. 2, pp. 712–736, may 2017.