# Optimizing Network Slicing in Distributed Large Scale Infrastructures: From Heuristics to Controlled Deep Reinforcement Learning

José Jurandir Alves Esteves[§], Amina Boubendir[‡], Fabrice Guillemin[*] and Pierre Sens[†]
[*]Orange Labs, France
[†]Sorbonne Université / CNRS / Inria, LIP6, France
[‡]Airbus Defence and Space, France
[§]Mercado Livre, Brazil

josejurandir.alves@mercadolivre.com, amina.boubendir@airbus.com, fabrice.guillemin@orange.com, pierre.sens@lip6.fr

***Abstract*—This paper summarizes the PhD thesis and the 10 associated publications on the optimization of network slice placement in large-scale distributed infrastructures by focusing on online heuristics and approaches based on Deep Reinforcement Learning (DRL). First, we rely on Integer Linear Programming (ILP) to propose a data model for on-Edge and on-network slice placement. Second, we leverage an approach called Power of Two Choices (P2C) to propose an online heuristic adapted to support placement on large-scale distributed infrastructures while incorporating Edge-specific constraints like latency. Finally, we investigate the use of Machine Learning (ML) methods, specifically DRL, to increase the scalability and automation of network slice placement by considering a multi-objective optimization approach to the problem. We will go through the extensive evaluation work that provide encouraging results about the advantages of the proposed approaches when used in realistic network scenarios.**

***Index Terms*—Network Functions Virtualization, Network Slicing, Placement, Large-scale infrastructures, Optimization, Heuristics, Automation, Deep Reinforcement Learning.**

## I. Introduction

### A. Thesis Context and Motivations

Today, telecom operators are going through an accelerated evolution of their technological scope with the appropriation of different technologies and paradigms. First, we have the Cloud, which has made it possible to move many applications and data to centralized Data Centers (DC), thus enabling the growth of a variety of services such as, for example, Over-The-Top (OTT) services. Telecom operators are also moving to the Cloud via Network Function Virtualization (NFV). NFV is network cloudfication and will allow distributed deployment of Virtualized Network Functions (VNFs) on a shared physical infrastructure. Combined with network virtualization, NFV has paved the way for Network Slicing, which allows the interconnection of different virtual networks deployed within a DC network. Network Slicing is included in the 5G specifications foreseeing a high number of new connected devices, new uses and an imperative for privacy and security. To cope with these different new requirements, telecos have started to deploy distributed DCs located at the Edge of the network. Closer to the users, these DCs are used to host sensitive data but also VNFs sensitive to latency. The core of my PhD thesis lies at the intersection of three research topics: Management and Orchestration; Distributed Infrastructures, Edge/MEC, large scale networks; and Artificial Intelligence (AI), Machine Learning (ML).

### B. Major Challenges and Problem Statement

We have focused on the optimization of network slice management and the challenge of ensuring optimized resource utilization and compliance with application and service requirements during the management of the life cycle of a slice. More specifically, we study the Network Slice Placement problem which can be formulated as an optimization problem and has been largely studied in the literature [1].

Indeed, we define Network Slice Placement Requests (NSPRs) that represent the resource and Quality-of-Service (QoS) requirements of a slice arriving sequentially for placement and the placement solution should decide on the optimal selection of servers hosted by the Physical Substrate Network (PSN) on which to deploy the slice's VNFs and the paths to direct traffic between these VNFs by mapping the associated Virtual Links (VLs). Therefore, the main challenges are: i) to accept as many placement requests as possible while meeting the service needs in terms of QoS requirements (latency, bandwidth), ii) consider the optimization of other criteria that may affect the acceptance rate, such as resource consumption (CPU, RAM) and network load.

We have considered challenging research questions that are at the intersections of the three topics. Firstly, how can we ensure QoS/QoE for Network Slice Users in a converged network-Edge/MEC?. Secondly, what are the more accurate optimization models and algorithms for slice placement in large-scale networks?. Thirdly, how to build automated and scalable algorithms to compute optimal placement decisions that are robust to changes in network behaviour?.

### C. Thesis Contributions

In the light of theses questions, we can summarize the thesis into 3 main contributions as follows:

1) an ILP-based Solution that deals with on-Edge and on-Network Slice Placement;
2) a Heuristic that relies on the principle of the Power of Two Choices (P2C) for Large Scale Network Slice Placement;
3) a Heuristically-assisted Deep Reinforcement Learning algorithim to automate the multi-objective Network Slice Placement.

This paper is a digest of the PhD thesis dissertation [2]. Sections II, III and IV present the 10 thesis publications structured into three major proposals. In Section V, we highlight the main conclusions of the work and give some perspectives.

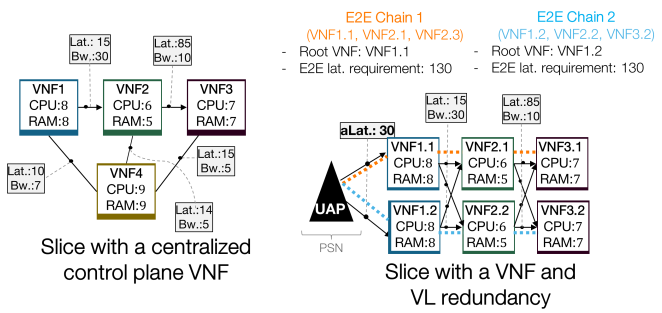



Fig. 1: Network Slice Placement Request (NSPR) model

## II. Enabling on-Edge and on-Network Slice Placement: an ILP-based Solution

We describe in this section the first contribution along with the main identified gaps it tackled and its evaluation results.

### A. Related Work Analysis: Main Identified Gaps

In spite of the numerous papers about placement in virtual networks, most of them frequently ignore the geographic dimension of placement decision (some exceptions are [3]–[6]). Existing studies often do not take into account neither the user's location when solving the problem nor user location implications, especially in the E2E latency calculation. These aspects are fundamental to enable on-Edge and on-Network slice placement and we specifically address this challenge.

### B. Contribution Overview

This contribution gathers three main proposals as follows:

1) Propose an E2E latency model that integrates the user location as an end-point of the Network Slice, and improves scalability by exploiting the possibility of grouping NSUs instead of considering them individually (unlike [5]) ;
2) Deal with complex Network Slice topologies, going beyond the currently studied Service Functions Chaining (SFC) (generalizing [6] and [5] hypotheses);
3) Set no restrictions on the placement location of two VNF of the same Network Slice (generalizing [4] hypothesis).

The proposed model is formalized mathematically using ILP and implemented in a latency-aware Network Slice Placement solution described in [7]. We assess the performance of the model to understand its pertinence and scalability in [8].

*1) NSPR modeling:* As illustrated by Fig. 1, the NSPR represents the view of the requirements of VNFs and VLs of the slices to place. Each placement request is associated with a specific group of network slice users. Fig. 1 also shows that multiple NSPR topologies are considering from SFC to more complex graph topologies. Another feature of our model is the E2E chain which corresponds to VNF sequences or VNF paths through which the traffic can pass.For each E2E Chain, we define a request in terms of E2E delay which corresponds to delays from the UAP associated with this placement request to the last VNF in the chain.

*2) PSN modeling:* As depicted in Fig. 2, the The PSN is composed by the infrastructure resources needed to support the VNF deployment and interconnection by VLs. The PSN is divided in 3 parts. First, the virtualized infrastructure with the DCs offering IT resources to run the VNFs. And we consider 3 types of DCs: Central Cloud Platforms (CCPs) as national DCs with big resource capacities, Core DCs (CDCs) as regional DCs with medium resource capacities, and Edge DCs (EDCs) as local DCs with small resources capacities. The second part of the PSN is the the Acess Network representing the user access points that can be a cellular antenna or a Wifi AP used by end-users to access the slices. And finally we have the Transport Network with the routers and transmission links needed to interconnect the different DCs and the user access points.

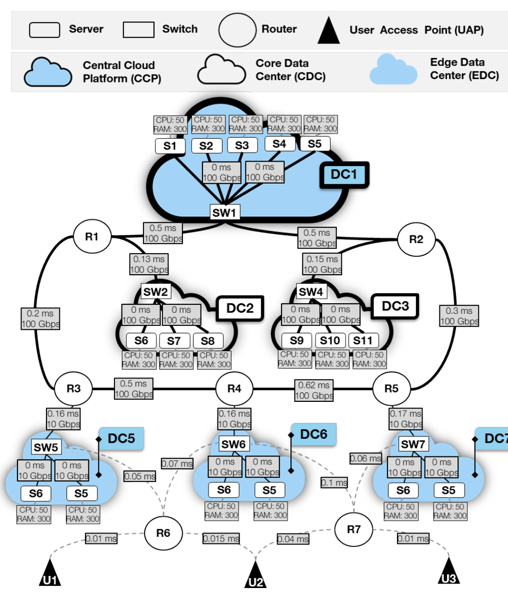
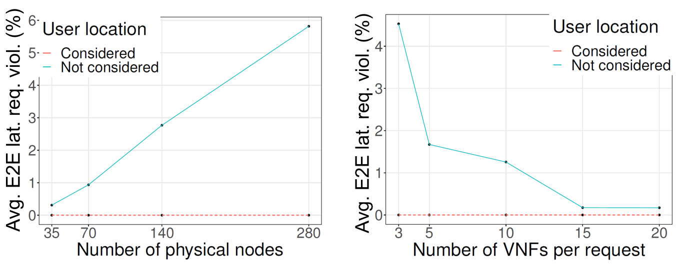



Fig. 2: Physical Substrate Network (PSN) model.

### C. Experiments and Evaluation results

The proposed ILP model was implemented using Julia language and solved using the default branch and bound algorithm of CPLEX. In order to understand the level of violations that can be obtained when user location is not considered, we compare the proposed location-based model with a location-agnostic model widely used in the state-of-the-art, that considers E2E latency but does not take user location into account [9]. Fig. 3(a) presents the evolution of the average E2E latency requirement violation according to the number of nodes on the PSN. In contrast to our formulation that always respects E2E latency requirements, a growing average E2E latency requirement violation is observed when we do not take into account user location.

Fig. 3(b) presents the evolution of the average of the E2E latency requirement violations according to the number of VNFs of the NSPRs. We also observe that the average violation decreases while the number of VNFs in the NSPRs increases. This happens because the more VNFs we have in the NSPRs, less the NSPRs are spread in the PSN, this reduces the E2E latency requirements violations. In fact, the CPU and RAM requirements of each VNF decrease when the number of VNFs per request increases. Hence, the tested models concentrate more VNFs inside the same machines to prevent from using link resources.




(a)

(b)

Fig. 3: E2E latency model relevance analysis

## III. Optimizing Large Scale Network Slice Placement: a Heuristic using P2C

We describe in this section the second contribution along with the main identified gaps it tackles and its evaluation results.

### A. Related Work Analysis: Main Identified Gaps

Numerous papers about placement in virtual networks use heuristic-based approaches to solve associated optimization problems (some examples are [10]–[14]). However, most of them do not jointly address the large-scale network aspect and the Edge-specific constraints and thus the direct impact on QoS metrics (notably, E2E latency).

### B. Contribution Overview

This contribution gathers two main proposals:

1) An original method of placing network slices through a heuristic based on the P2C algorithm [15] which is adapted to large-scale network scenarios and integrate both Edge-specific constraints related to user location and strict E2E latency requirements;
2) A policy for selecting servers for VNF placement that offloads Edge DCs (EDC) and improves Network Slice acceptance ratio.

We implemented the heuristic inside an Network Slice Placement solution described in [16] and compare it to ILP-based placement algorithms in [17].

This heuristic is based on the P2C principle [15], which states in the present context that considering two possible DCs chosen "randomly" instead of only one brings exponential improvement of the solution quality. It is a greedy algorithm such that for each VNF $\bar{b} \in V$:

1) Randomly select 2 candidate servers $s_1, s_2 \in S$;
2) Evaluate the resource consumption when placing $\bar{b}$ in $s_1$ and $s_2$ and place $\bar{b}$ on the best server;
3) Map the VLs $(\bar{a}, \bar{b}) \in E$ associated to $\bar{b}$.

### C. Experiments and Evaluation results

We have used Julia language to implemented two version of the proposed heuristic with two different policies for selecting candidate servers for placement P2C1 and P2C 2 for policies 1 and 2 respectively (see [17] for a complete description of the proposed candidate server selection policies). We compared them with to versions of the ILP introduced in Section II, ILP1 and ILP2 maximizing resource utilization and acceptance of slices respectively.

*1) Average execution time evaluation:* The average execution times in function of the number of servers in the PSN is given in Fig. 4. Starting from a PSN with 126 servers as described in [17], we generated new PSN settings by doubling the number of servers in each DC. The evaluation results confirmed our expectations showing that the average execution time grows much faster for the ILPs than for the heuristics. In the scenario with 16128 nodes the execution times are 9.8 and 12.5 seconds for the ILPs 1 and 2 respectively and 2.17 and 1.96 seconds for P2C 1 and 2 respectively.

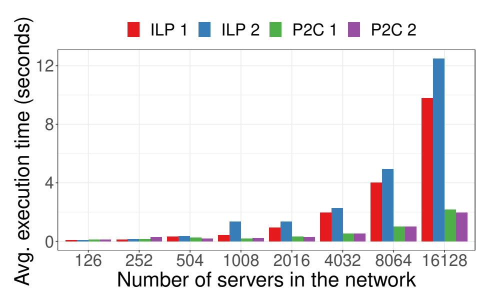
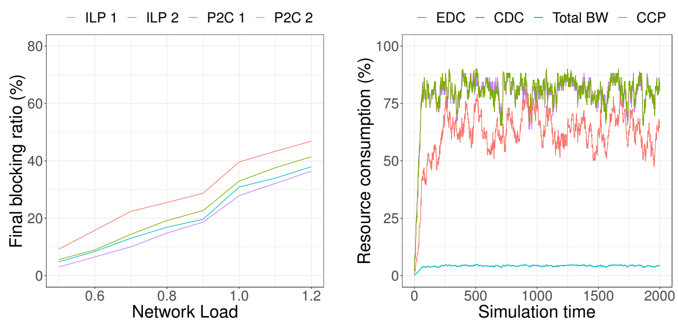



Fig. 4: Average execution time evaluation.

*2) Average final blocking ratio evaluation:* We evaluated the final blocking rates for different simulation scenarios, i.e., the percentage of NSPRs rejected along a given simulation duration in relation to the load submitted to the network, which can be calculated from the parameters of arrival and exit rates of the slices and the capacities and requirements in terms of resources. We have evaluated moderate loads up to overload phenomena. Fig. 5 present results for URLLC simulation scenario. Fig. 5(a) shows that the P2C2 heuristic has the best blocking rate. This happens because load balancing is essential in this scenario to avoid overloading the Edge DCs and this heuristic achieve better load balancing by actively offloading the Edge DCs. This behavior of the heuristic can be observed in Fig. 5(b). P2C 2, concentrates the load on the Cloud and Core DCs to offload the Edge DCs.




(a) Average final blocking ratio.

(b) P2C 2 Resource consumption.

Fig. 5: Evaluation results for URLLC simulation scenario

## IV. Automating Multi-objective Network Slice Placement using a Heuristically-assisted DRL

We describe in this section the third contribution along with the main identified gaps it tackled and its evaluation results.

### A. Related Work Analysis: Main Identified Gaps

From an operational perspective, heuristic approaches are more suitable than ILP as they yield faster placement results. The drawback of heuristic approaches is that they give sub-optimal solutions. To address this issue, ML offer a corpus of methods, such as DRL, which are able to overcome the convergence issues of ILP while being more accurate than heuristics. DRL has recently been used in the context of Network Slice Placement [18]–[20].

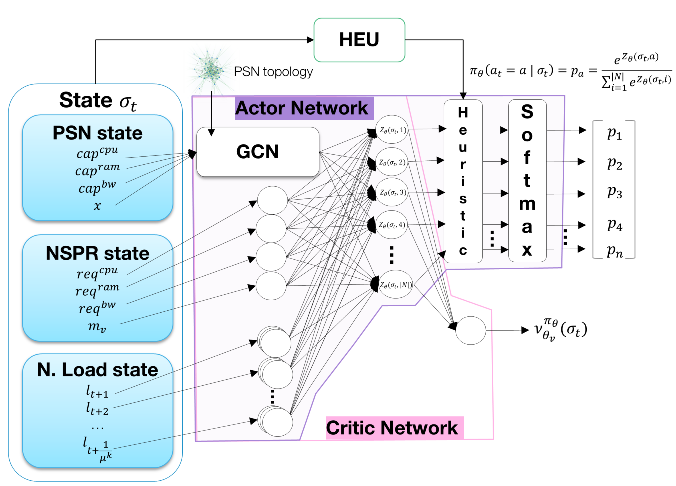



Fig. 6: Proposed framework for the learning algorithms.

However, from a practical point of view, ensuring that a DRL agent converges to an optimal policy is still a challenge. A first important drawback is that DRL agents act as self-controlled black boxes. In addition, there are a large number of hyper-parameters to fine-tune in order to ensure an adequate equilibrium between exploring solutions and exploiting the knowledge acquired via training. While there are techniques to improve the efficiency of the solution exploration process (e.g., $\epsilon$-greedy, entropy regularization), their use may also lead to situations of instability, where the algorithm may diverge from the optimal point.

Another issue with DRL is its use in non-stationary environments. As a matter of fact, when the environment is continually changing the rules, the algorithm has trouble in using the acquired knowledge to find optimal solutions. The usage of the DRL algorithm in a online fashion can then become impractical. Most of the existing works applying DRL to placement in virtual networks assume a stationary environment, i.e., with static network load. However, traffic conditions in networks are basically non-stationary with daily and weekly variations and subject to drastic changes (e.g., traffic storm due to an unpredictable event).

To overcome this unsuitable behaviour of DRL agents, based on the concept of Heuristically Accelerated Reinforcement Learning [21], we introduce the concept of HA-DRL and we apply it in a fully online learning scenario with time-varying network loads to show how this strategy can accelerate and stabilize the convergence of DRL techniques when applied to the Network Slice Placement.

### B. Contribution Overview

This contribution gathers three main proposals:

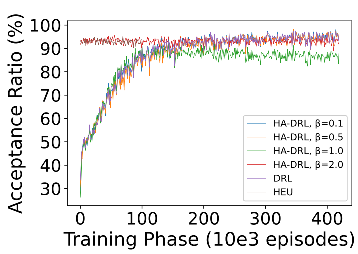
1) Combines Graph Convolutional Network (GCN)—to automatically extract PSN related features—and Advantage Actor Critic algorithm—for optimal policy learning—to solve multi-objective Network Slice Placement optimization problem;
2) Provides a network load model to network slice infrastructure conditions with time-varying network loads;
3) Reinforces the DRL learning process by using the P2C based heuristic we propose in [17] to control the DRL convergence.

We implemented the proposed algorithms inside an Network Slice Placement solution described in [22] and evaluated them in three different network load scenarios: static, in which the network load is represented by a constant function (see [23], [24]); cycle-stationary, in which the network load is represented by a cycle-stationary function (see [25]); and non-stationary, in which the network load is represented by a stair-stepped function (see [26]). Figure 6 represents the structure of the proposed HA-DRL framework. It is an extension of the Asynchronous Advantage Actor Critic (A3C) algorithm. We consider two non-controlled DRL algorithms, DRL, which observes only PSN and NSPR state and eDRL which also observe the netork load state to learn the network load variations. Both algorithms uses two Deep Neural Networks (DNNs) to learn the optimal policy $\pi_\theta$ and the optimal state-value $\nu_{\theta_v}^{\pi_\theta}$ function called Actor and Critic Networks, respectively. We propose to modify the Actor Network by adding a Heuristic Function layer that enhances the exploration process and accelerates the convergence of the algorithm by influencing the policy choice of actions. This layer benefits from external information provided by the heuristic introduced in Section III, referred as HEU. A detailed description of the two heuristically assisted algorithms, HA-DRL and HA-eDRL, derived from DRL and eDRL algorithms, can be found in [23], [25].

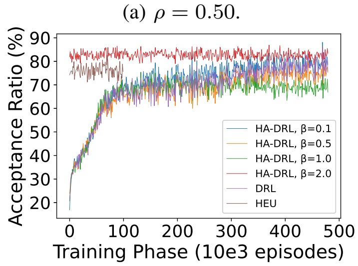
### C. Experiments & Evaluation results

The proposed solution has been implemented in Python language, using the Pytorch library for the ML components. We simulated online learning scenarios using the same PSN configurations of the previous contributions except that we do not consider the latency aspect as we focused rather on the resource usage and network load aspects.

*1) Static network load scenario evaluation results:* Fig. 7 presents evaluation results for the static network load scenario. In this evaluation we considered the DRL algorithm, the HEU algorithm and 4 versions of the HA-DRL algorithm, with 4 values for the $\beta$ parameter that controls the influence of the heuristic on the placement policy. Fig. 7(a) shows that all the algorithms end up having a similar performance after a certain number of training phases when the considered network load $\rho$ is low ($\rho = 0.5$). On the other hand, when the network load increases, we observe a more marked difference in performance between the algorithms. Fig. 7(a) shows that when $\rho = 0.9$ HA-DRL algorithm with $\beta = 2$ has a better performance than the other algorithms—including HEU— with a much faster convergence time than other DRL algorithms.




(a) $\rho = 0.50$.




(b) $\rho = 0.90$.

Fig. 7: Evaluation results for static network load scenario.

*2) Cycle-stationary network load scenario evaluation results:* Fig. 8 presents evaluation results for the cycle-stationary network load scenario. In this scenario, in addition to the DRL and HA-DRL algorithms, we also evaluated the eDRL and HA-eDRL algorithms which observe the network load state to try to learn the variations. Fig. 8 shows that with this short training time of 3 simulated days only the HA-DRL and HA-eDRL algorithms with $\beta = 2$ are able to converge. The other algorithms will actually need a much longer training time to learn in this cyclo-stationary network load scenario [25].

*3) Non-stationary network load scenario evaluation results:* The last step was therefore the evaluation on the non-stationary network load scenario, with unpredictable load variations. We considered here only the DRL and HA-DRL algorithms since the traffic breaks are not predictable and the network load state is not observable. Fig. 9 shows one of the evaluated network load disruption scenarios where we consider a jump in the network load from 40% to 100% at the training phase 108 marked here by the blue line. We can observe that only the HA-DRL algorithm with $\beta = 2$ has an almost optimal performance before the load disruption due to its fast convergence. Also, the load disruption affects all the algorithms, but the only one to succeed in maintaining a good performance is the HA-DRL with $\beta = 2$ due to the fact that it had been able to obtain a first convergence before the break. We could extend this analysis by also indicating that due to its fast convergence, HA-DRL with $\beta = 2$ is able to relearn faster on new network load conditions, which is very important in dynamic load scenarios.

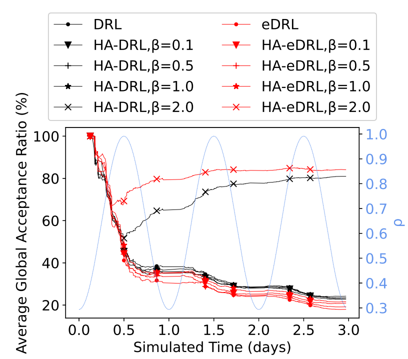



Fig. 8: Evaluation of cycle-stationary network load scenario

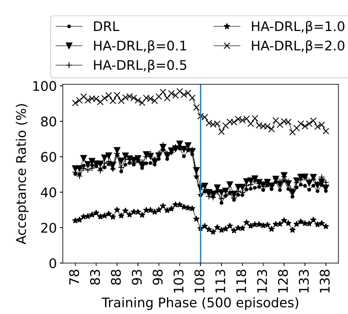



Fig. 9: Evaluation of non-stationary network load scenario.

## V. Conclusions & Perspectives

This paper summarizes the PhD dissertation and the associated 10 publications where key challenges for Network Slice Placement optimization in large scale networks have been studied. The first thesis contribution introduces ILP and offline optimization for slice placement considering an E2E latency model which takes into account user location and complex slice placement request topologies. Evaluation results showed that taking user location into account was essential to ensure strict E2E latency requirements and that we need more scalable solutions to support larger scales. The second thesis contribution was a heuristic for the optimization of slice placement in large-scale based on the P2C algorithm. We proposed an online optimization approach adapted to large-scale scenarios while integrating Edge-specific constraints (i.e., strict E2E latency and user location). Evaluations showed that the approach achieves good solutions in a short execution time and that the server selection policies proposed to improve load balancing which leads to a higher slice acceptance ratio. The third thesis contribution is an HA-DRL approach to optimize and automate placement, an online, multi-objective optimization approach that was proven scalable after several simulations considering three network load scenarios: static, cycle-stationary, and non-stationary. The study conducted in this part of the thesis has shown the limitations of DRL approaches for our problem due to a long convergence time. The HA-DRL approaches that we have proposed present a fast convergence and a higher robustness even in dynamic load scenarios and are thus adapted to real network scales.

As perspectives, we emphasize two points. First, to extend the proposed approaches to networks with heterogeneous technological domains within the PSN such as the RAN and devices. Second, move from a centralized DRL to a distributed multi-agent DRL for a greater scalability and to minimize the amount of transmitted monitoring traffic.

## VI. Acknowledgments

This PhD thesis has been conducted within Orange Labs and LIP6 department of Sorbonne University and was funded by the 5GPPP MON-B5G project (monb5g.eu). The experiments were conducted using Grid'5000 platform, a large scale testbed by Inria (grid5000.fr). The authors thank the Network and Service Management community for their valuable feedback at IEEE conferences like CNSM, IM, NetSoft, ICIN, and NoF.

## References

[1] A. Fischer, J. F. Botero, M. T. Beck, H. De Meer, and X. Hesselbach, "Virtual network embedding: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 1888–1906, 2013.

[2] J. J. Alves Esteves, "Optimization of network slice placement in distributed large scale infrastructures: From heuristics to controlled deep reinforcement learning," Ph.D. dissertation, Orange Labs and Sorbonne Université, 2021. [Online]. Available: https://hal.inria.fr/tel-03500387/

[3] N. M. K. Chowdhury, M. R. Rahman, and R. Boutaba, "Virtual network embedding with coordinated node and link mapping," in *Proc. IEEE INFOCOM 2009*. IEEE, 2009, pp. 783–791.

[4] L. Gong, H. Jiang, Y. Wang, and Z. Zhu, "Novel location-constrained virtual network embedding lc-vne algorithms towards integrated node and link mapping," *IEEE/ACM Trans. Netw.*, vol. 24, no. 6, 2016.

[5] D. Harutyunyan, N. Shahriar, R. Boutaba, and R. Riggio, "Latency-aware service function chain placement in 5g mobile networks," in *Proc. IEEE Conf. Netw. Softwarization (NetSoft)*, 2019.

[6] B. Spinnewyn, P. H. Isolani, C. Donato, J. F. Botero, and S. Latré, "Coordinated service composition and embedding of 5g location-constrained network functions," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 4, pp. 1488–1502, 2018.

[7] J. J. Alves Esteves, A. Boubendir, F. Guillemin, and P. Sens, "Optimized network slicing proof-of-concept with interactive gaming use case," in *Proc. 2020 23rd Conf. Innovation Clouds, Int. Netw. Workshops (ICIN)*, 2020, pp. 150–152.

[8] J. J. Alves Esteves, A. Boubendir, F. Guillemin, and P. Sens, "Location-based data model for optimized network slice placement," in *Proc. 6th IEEE Conf. Netw. Softwarization (NetSoft)*, 2020, pp. 404–412.

[9] A. Alleg, T. Ahmed, M. Mosbah, R. Riggio, and R. Boutaba, "Delay-aware vnf placement and chaining based on a flexible resource allocation approach," in *Proc. 2017 13th Int. Conf. Netw. Service Manag. (CNSM)*. IEEE, 2017, pp. 1–7.

[10] M. Mechtri, C. Ghribi, and D. Zeghlache, "Vnf placement and chaining in distributed cloud," in *Proc. 2016 IEEE 9th Int. Conf. Cloud Comput. (CLOUD)*. IEEE, 2016, pp. 376–383.

[11] G. Wang, G. Feng, W. Tan, S. Qin, R. Wen, and S. Sun, "Resource allocation for network slices in 5g with network resource pricing," in *Proc. GLOBECOM 2017-IEEE Global Commun. Conf.* IEEE, 2017, pp. 1–6.

[12] P. Vizarreta, M. Condoluci, C. M. Machuca, T. Mahmoodi, and W. Kellerer, "Qos-driven function placement reducing expenditures in nfv deployments," in *Proc. 2017 IEEE Int. Conf. Commun. (ICC)*. IEEE, 2017, pp. 1–7.

[13] M. A. T. Nejad, S. Parsaeefard, M. A. Maddah-Ali, T. Mahmoodi, and B. H. Khalaj, "vspace: Vnf simultaneous placement, admission control and embedding," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 542–557, 3 2018.

[14] L. Gao and G. N. Rouskas, "On congestion minimization for service chain routing problems," in *Proc. 2019 IEEE IEEE Int. Conf. Commun. (ICC)*. IEEE, 2019, pp. 1–6.

[15] M. Mitzenmacher, "The power of two choices in randomized load balancing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 12, no. 10, 2001.

[16] J. J. Alves Esteves, A. Boubendir, F. Guillemin, and P. Sens, "Edge-enabled optimized network slicing in large scale networks," in *Proc. 2020 11th Int. Conf. Netw. Future (NoF)*, 2020, pp. 129–131.

[17] J. J. Alves Esteves, A. Boubendir, F. Guillemin, and P. Sens, "Heuristic for edge-enabled network slicing optimization using the "power of two choices"," in *Proc. 2020 16th Int. Conf. Netw. Service Manag. (CNSM)*, 2020, pp. 1–9.

[18] Z. Yan, J. Ge, Y. Wu, L. Li, and T. Li, "Automatic virtual network embedding: A deep reinforcement learning approach with graph convolutional networks," *"IEEE J. Sel. Areas Commun."*, vol. 38, no. 6, pp. 1040–1057, 06 2020.

[19] M. Dolati, S. B. Hassanpour, M. Ghaderi, and A. Khonsari, "DeepViNE: Virtual network embedding with deep reinforcement learning," in *Proc. IEEE INFOCOM 2019 - IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, 2019, pp. 879–885.

[20] H. Yao, X. Chen, M. Li, P. Zhang, and L. Wang, "A novel reinforcement learning algorithm for virtual network embedding," *Neurocomputing*, vol. 284, pp. 1–9, 4 2018.

[21] R. A. Bianchi, C. H. Ribeiro, and A. H. R. Costa, "Heuristically accelerated reinforcement learning: Theoretical and experimental results." in *Proc. 20th Eur. Conf. Artif. Intell. (Frontiers in Artificial Intelligence and Applications, vol. 242).*, 2012, pp. 169–174.

[22] J. J. Alves Esteves, A. Boubendir, F. Guillemin, and P. Sens, "Drl-based slice placement under realistic network load conditions," in *Proc. 2021 17th Int. Conf. Netw. Service Manag. (CNSM)*, 2021, pp. 1–3. [Online]. Available: https://arxiv.org/pdf/2109.12857.pdf

[23] J. J. Alves Esteves, A. Boubendir, F. Guillemin, and P. Sens, "A heuristically assisted deep reinforcement learning approach for network slice placement," *IEEE Trans. Netw. Service Manag.*, pp. 1–1, 2021. [Online]. Available: https://arxiv.org/pdf/2105.06741.pdf

[24] J. J. Alves Esteves, A. Boubendir, F. Guillemin, and P. Sens, "Controlled deep reinforcement learning for optimized slice placement," in *Proc. 2021 IEEE Int. Mediterranean Conf. Commun. Netw. (MeditCom)*, 2021, pp. 1–3. [Online]. Available: https://arxiv.org/pdf/2108.01544.pdf

[25] J. J. Alves Esteves, A. Boubendir, F. Guillemin, and P. Sens, "Drl-based slice placement under non-stationary conditions," in *Proc. 2021 17th Int. Conf. Netw. Service Manag. (CNSM)*, 2021, pp. 1–9. [Online]. Available: https://arxiv.org/pdf/2108.02495.pdf

[26] J. J. Alves Esteves, A. Boubendir, F. Guillemin, and P. Sens, "On the robustness of controlled deep reinforcement learning for slice placement," *Journal of Network and Systems Management*, vol. 30, no. 3, pp. 1–24, 2022. [Online]. Available: https://arxiv.org/pdf/2108.02505.pdf