



**HAL**  
open science

## NSPlacer: A Tool for Evaluating Service Placement Algorithms in Post-5G Networks

Masoud Taghavian, Yassine Hadjadj-Aoul, Géraldine Texier, Philippe Bertin

► **To cite this version:**

Masoud Taghavian, Yassine Hadjadj-Aoul, Géraldine Texier, Philippe Bertin. NSPlacer: A Tool for Evaluating Service Placement Algorithms in Post-5G Networks. CloudNet 2022 - IEEE 11th International Conference on Cloud Networking, Nov 2022, Paris, France. pp.252-256, 10.1109/CloudNet55617.2022.9978830 . hal-03911372v1

**HAL Id: hal-03911372**

**<https://inria.hal.science/hal-03911372v1>**

Submitted on 22 Dec 2022 (v1), last revised 7 Feb 2023 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# *NSPlacer*: A Tool for Evaluating Placement Algorithms in Network Function Virtualization

Masoud Taghavian<sup>\*†</sup>, Yassine Hadjadj-Aoul<sup>\*‡</sup>, Géraldine Texier<sup>\*†</sup>, Philippe Bertin<sup>\*§</sup>

<sup>\*</sup>IRT-BCOM, France; [firstname.lastname@b-com.com](mailto:firstname.lastname@b-com.com)

<sup>†</sup>IMT Atlantique/IRISA/Adopnet, France; [firstname.lastname@imt-atlantique.fr](mailto:firstname.lastname@imt-atlantique.fr)

<sup>‡</sup>University of Rennes, Inria, CNRS, IRISA, France; [firstname.lastname@irisa.fr](mailto:firstname.lastname@irisa.fr)

<sup>§</sup>Orange, France; [firstname.lastname@orange.com](mailto:firstname.lastname@orange.com)

**Abstract**—Network Function Virtualization (NFV) has been a remarkable shift from dedicated network devices towards reusable Virtual Network Functions (VNF). A network service is defined as a set of VNFs connected via Virtual Links (VL), and it is realized by assigning its elements to their corresponding substrate network nodes and links. Efficient placement of the network services is one of the most important challenges introduced by NFV. We propose a demonstration of the “Network Service Placer (*NSPlacer*)” tool, as an implementation of several popular state-of-the-art placement algorithms, to pave the way to study the placement problem for the researches in this field of interest. By being able to adjust various parameters, related to the substrate network, the service and the placement algorithm, our online tool provides near-optimal and optimal results to serve as a comparison with many placement solutions.

**Index Terms**—Network function virtualization, service placement, online tool.

## I. INTRODUCTION

Using cloud computing technologies in Network Function Virtualisation (NFV) to virtualize and orchestrate network functions are inevitable in modern networks. Placement is one of the most important steps in NFV, which addresses the allocation of network resources required by the services, and it is proven to be NP-Hard in terms of complexity [3].

In [7], we addressed the online placement problem, based on Branch and Bound (BB), allowing to apply various Artificial Intelligence (AI) search strategies (especially A\*). We demonstrated how to obtain near-optimal and optimal results with a high degree of the scalability. BB caught our attention since the large number of constraints inherent to the placement problem can be beneficial on BB (the more constraints we have, the more we can prune the tree, leading to a faster search). The effectiveness of our approach is confirmed by the exhaustive evaluations made by our *NSPlacer* tool.

In this paper, we present *NSPlacer* (accessible online on <https://nsplacer.labs.b-com.com/>), which we designed and implemented to evaluate and compare several placement algorithms, including our approach in [7].

Placement algorithms have been studied in the literature for several years. In online placement, we look

for a suitable placement (utilizing the minimum possible amount of resources) for a single service request as soon as it arrives over a timeout. The studied placement algorithms range from the ILP formulations like [2] to evolutionary algorithms like [4], and heuristic methods. Using sophisticated AI and ML techniques like [5] have been gaining attention only recently, but they seem highly promising. Our work is classified in this category discovering the potential of the AI search strategies.

Since placement is encountered in many related fields, *NSPlacer* can facilitate the study of these fields of interest for the research community. Allowing to adjust many parameters, whether related to the substrate network, the service specifications, or the placement algorithm, our online tool provides near-optimal and optimal results to serve as a comparison with many placement solutions.

## II. PROPOSED SOLUTION

To use our *NSPlacer* tool, it is worth to explain some of the fundamentals and the general idea of our approach.

A Substrate Network (SN) is a physical network represented by a graph of the nodes and the links, with a capacity of resources and the Quality of Service (QoS) metrics. Similarly, a Service Graph (SG) is a graph of Virtual Network Functions (VNFs) and Virtual Links (VLs), requiring some resources with QoS considerations. The placement problem deals with finding the mappings from the VNFs and the VLs of a service into the network nodes and the paths (the sequence of links), respectively.

BB is one of the most popular algorithm design paradigms used for solving problems with exponential complexity. BB allows to perform a complete search, while avoiding progression on unfeasible branches by pruning the tree (Fig. 1). We use the term *state* to refer to the nodes of the search tree to better differentiate them from the nodes of the network. On the search for a complete placement of a service, we traverse the search tree, according to an order which is determined by a search strategy. Our ultimate goal is to investigate the influence of applying various search strategies to maximize the number of placed services. Accordingly, we propose several search strategies. A\* *Bandwidth Optimized (ABO)*: A\* is amongst the most

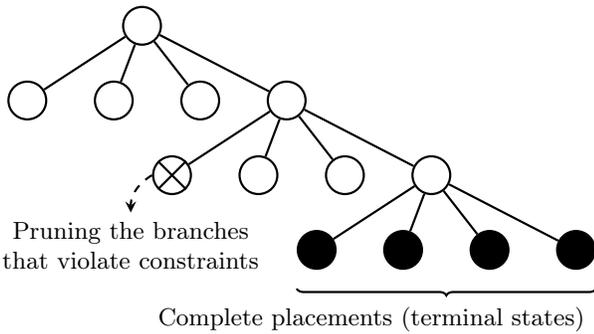


Figure 1. A sample BB search tree, placing a service with 3 VNFs over a network with 4 nodes

popular AI informed search algorithms due to its completeness, optimality, and efficiency [6]. ABO is one of our search strategies based on A\* algorithm to perform a complete and optimal search by concentrating on the bandwidth usage optimization. The proof of the optimality of the ABO is presented in [7]. *Fair A\* Bandwidth Optimized (FABO)*: Along with the optimality, increasing the chances of accepting the new service requests by avoiding network defragmentation is another important consideration. FABO is our other search strategy, which is defined on top of ABO trying to address this issue by performing a fair distribution of the load over the network links. *DFS Bandwidth Optimized (DBO)*: DBO is our other search strategy based on a Depth-First Search (DFS)-traversal of the search tree in order to quickly reach a near-optimal result. *Best-Fit (BF) and Enhanced Best-Fit (EBF)*: BF is one of the most well-known heuristic for service placement. By adapting BF in our BB, we could obtain our other search strategy called EBF, which improves the results of BF remarkably. *Parallel*: Parallel does not define a specific traversal, but it executes FABO, ABO and DBO in parallel, and integrates them to obtain the best results.

### III. EXECUTION AND PARAMETERS

The evaluations performed by our tool starts with initializing the entire nodes and links of the network with the maximum available resources, then trying to place services iteratively. At each iteration, a service is created according to the specified input parameters. Next, it tries to place the service over the network using a selected strategy within the timeout. If the strategy succeeds to find a placement, it is applied to the network by updating the remaining resources, before starting a new iteration. Otherwise, it terminates the evaluation and reports the results.

The input parameters are classified in 3 sections (Fig. 2). *SN*: Our tool allows users to upload and visualize topologies from *Zoo Topology* dataset [1], or by creating their own custom topologies (based on Zoo Topology XML file format) (Fig. 3). *SG*: We propose different types of service typologies, including Daisy Chain, Ring and Star. The *size* specifies the number of VNFs in the SG. Each

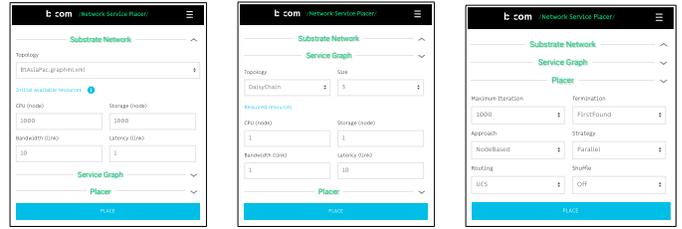


Figure 2. The input parameters

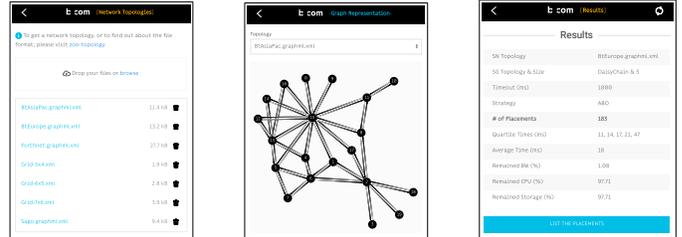


Figure 3. Network selector, network representation, and results

VNF requires several units of resources from a network node, specified by the *CPU* and the *storage*. Likewise, *bandwidth* and *latency* are defined for VNs. *Placer*: In an online placement, we place one service as soon as it arrives, imposing a *Timeout*. The *Termination* parameter specifies whether we should search for better placements until the timeout is reached or terminate as soon as we find a placement. The *Approach* represents two possible formulations in BB. The *Strategy* chooses a search strategy, which the placer follows.

*The Results*: At the end of the evaluation, the results are reported (Fig. 3), including the number of placed services, and the details of the performed placements.

### IV. CONCLUSIONS AND FUTURE WORK

We presented online *NSPlacer* tool, as an implementation of several placement algorithms, in order to facilitate evaluations for research purposes. As a future direction, we are interested in the edge networks, where the location of the service depends on the location of the service consumers.

### REFERENCES

- [1] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan. Zoo topology. <http://www.topology-zoo.org/>.
- [2] M. C. Luizelli, L. R. e. Bays, L. S. Buriol, M. P. Barcellos, and L. P. Gaspar. Piecing together the nfv provisioning puzzle: Efficient placement and chaining of virtual network functions. In *2015 IFIP/IEEE Int. Symp. on Int. Network Management (IM)*, pages 98–106. IEEE, 2015.
- [3] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba. Network function virtualization: State-of-the-art and research challenges. *IEEE Communications surveys & tutorials*, 18(1):236–262, 2015.
- [4] T. Pham, J.-M. Sanner, C. Morin, and Y. Hadjadj-Aoul. Virtual network function-forwarding graph embedding: A genetic algorithm approach. *Int. J. of Communication Systems*, 33(10):e4098, 2020. e4098 0.1002/dac.4098.

- [5] P. Quang, Y. Hadjadj-Aoul, and A. Outtagarts. A deep reinforcement learning approach for vnf forwarding graph embedding. *IEEE Transactions on Network and Service Management*, 16(4):1318–1331, 2019.
- [6] S. Russell and P. Norvig. *Artificial intelligence: a modern approach*. Prentice Hall, 2002.
- [7] M. Taghavian, Y. Hadjadj-Aoul, G. Texier, and P. Bertin. An approach to network service placement using intelligent search strategies over branch-and-bound. In *2021 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2021.