On the Optimal Deployment of Virtual Network Functions in Non-Terrestrial Segments

Antonio Petrosino, *Student Member, IEEE*, Giuseppe Piro, *Member, IEEE*, Luigi Alfredo Grieco, *Senior Member, IEEE*, and Gennaro Boggia, *Senior Member, IEEE*

***Abstract*—In the upcoming Sixth-Generation (6G) of mobile communication systems, space network entities will cooperate with conventional terrestrial networks to provide three-dimensional wireless connectivity around the World. By considering the resulting massive amount of data to be managed into non-terrestrial segments, it will be necessary to dynamically configure functionalities across space network entities. Preliminary contributions in this context focus on quasi-static scenarios, while ignoring the challenging issues (i.e., intermittent visibility, dynamic network configuration, and communication delays) introduced by satellites' movement and communication protocols enabling the integration of terrestrial and non-terrestrial networks. To bridge this gap, this work presents a network architecture with novel orchestration capabilities of services into non-terrestrial segments. In the proposed approach, the interaction between terrestrial and non-terrestrial entities and the cloud has been detailed across service request, configuration, and provisioning phases. Then, starting from a system model describing the network configuration and the resulting deployment delays of services, an optimization problem has been formulated to dynamically allocate Virtual Network Functions (VNFs) among LEO CubeSats over a looking-ahead horizon, based on service requests, computational capabilities of the involved CubeSats, visibility matrices, and expected deployment delay bounds. Finally, the proposed optimization problem has been solved through three heuristic strategies. Computer simulations have been carried out to demonstrate the ability of the developed strategies to achieve results close to the optimal solution and to ensure better performance against a benchmark scheme.**

***Index Terms*—6G, Non-Terrestrial Network, Virtual Network Functions, Network Optimization.**

I. Introduction

WHILE the Fifth-Generation (5G) of mobile communication systems are being deployed in many parts of the World, recent research interests are moving towards the Sixth-Generation (6G) of mobile networks [1]. At the time of this writing, there already exists a common consensus that 6G networks will target the very ambitious goal to realize a Ubiquitous Intelligent Mobile Society, based on scalable and effective fruition of connectivity and computing services on demand, and wherever needed [2]. To this end, Non-Terrestrial Network (NTN) entities will cooperate with conventional terrestrial networks to provide three-dimensional wireless connectivity, also covering deserts, forests, and oceans [3], [4].

Antonio Petrosino, Giuseppe Piro, Luigi Alfredo Grieco, and Gennaro Boggia are with the Department of Electrical and Information Engineering (DEI), Politecnico di Bari, Italy, and with Consorzio Nazionale Interuniversitario per le Telecomunicazioni (CNIT). Email: {antonio.petrosino, giuseppe.piro, alfredo.grieco, gennaro.boggia}@poliba.it,

The global coverage capabilities will enable a range of innovative 6G-oriented use cases and the potential for customized, on-demand services through the use of the spatial segment [5]–[7]. However, the on-demand deployment of customized services raises the need for the dynamic management of the NTN. Software-Defined Networking (SDN) and Network Function Virtualization (NFV) principles can be used to separate the data and control planes in NTNs and configure their functionalities according to service requests [8]. Nevertheless, the network dynamism provoked by satellites' movement invites considering the usage of effective orchestration frameworks willing to deploy specialized Virtualized Network Functions (VNFs) across satellites, on demand [9], [10]. Unfortunately, conventional strategies conceived for the VNF deployment at the edge of terrestrial networks, like [6]- [19], appear inadequate for this purpose, because of the motion of satellites and the resulting intermittent visibility they grant to the terminals on the Earth's surface. Still, the contributions explicitly focusing on NTNs, such as [20]–[37], just concentrate on quasi-static scenarios, thus ignoring the challenging issues (i.e., intermittent visibility and dynamic network configuration) introduced by satellites' movement and communication protocols enabling the integration of terrestrial space network elements.

Based on these premises, and starting from a very preliminary study presented by the same authors of this paper in a previous conference contribution [38], the work presented herein intends to extend the current state of the art by successfully addressing the aforementioned open research challenges and providing the following main scientific contributions:

1) Definition of a novel network architecture able to collect service requests and related quality of service constraints, implement a service orchestration function, and effectively deploy the corresponding VNFs across satellites over time. The resulting approach is general and may support any kind of service. Specifically, a new communication protocol has been conceived to enable the interaction among terrestrial and space network entities (distributed among User, Edge, and Cloud layers) during three different operating phases, namely service request, configuration, and provisioning;
2) Design of a system model able to catch the network configuration (i.e., groups of terminals on the Earth's surface, satellites' constellation, orbits and consecutive visibility time windows, allocation of VNFs over time, and so on) and quantify the deployment delay experi-

enced by the end-users;
3) Formulation of an optimization problem willing to dynamically allocate VNFs among satellites over a looking-ahead time horizon, based on service requests, computational capabilities of involved satellites, visibility matrices, and expected deployment delay bounds;
4) Development of three different strategies based on meta heuristic approaches (i.e., Tabu Search (TS) [39], Simulated Annealing (SA) [40], and Genetic Local Search (GLS) [41]), able to solve the aforementioned optimization problem.

Computer simulations have been carried out to demonstrate the effectiveness of the proposed approach, in terms of average deployment delay, resource consumption, and processing time, by varying parameter settings. The study demonstrates that the three heuristic approaches can produce outcomes that are similar to the optimal solution, as well as better performance with respect to a benchmark technique, namely Greedy Algorithm (GR) algorithm. Among the others, the SA-based strategy emerges as the best solution that can guarantee better performance in terms of service deployment delays, resource consumption, and processing time.

The remainder of the paper is organized as follows. Related works are discussed in Section II. Section III presents the proposed network and protocol architecture, the related system model, and the formulated optimization problem. The performance assessment is discussed in Section IV. Finally, conclusions are drawn in Section V.

II. Related Works

The dynamic provisioning of VNFs at the edge of the terrestrial networks has been widely investigated in the current scientific literature. Most noteworthy contributions addressed this topic by formulating optimization problems (quite often solved through heuristic techniques), willing to reduce energy consumption [11], improve resource allocation [12], or minimize the end-to-end delay [13]–[15]. Other works focused attention on the real-time provisioning of services via VNFs. In this context, the proposed approaches have been conceived to lower the migration cost of the VNFs [16], [17], maximize the security level of each VNF [18], reduce the energy consumption [19], or minimize the end-to-end delay [6].

The advent of 6G-oriented use cases brought a rapid increment of research efforts focusing on NTN [7]. Recent contributions also emphasize the use of computational resources onboard satellites for task offloading and advanced service provisioning to ground users [9]. However, the enticing solutions designed for the terrestrial network appear inadequate for the NTN infrastructure due to the motion of the satellite and the intermittent visibility offered by a satellite constellation. In this context, the contributions discussed in [20]–[23], [25], [26] addressed the task offloading theme over the NTN segment, but without explicitly mentioning the usage of VNFs. These solutions have been conceived in order to improve the load balance in the network, by separating services into different network slices [20], manage the allocation of communication [21] or computational (i.e., processing and storage) [22] resources in order to maximize the number of accomplished requests, as well as minimize the end-to-end delay [23], [24]. Likewise, the strategy presented in [25] aims to lower energy consumption by optimizing the satellite resources according to the forecast number of ground users to be served. Furthermore, the solution proposed in [26] jointly minimizes the energy consumption and the end-to-end delay by handling the offloading and the resource allocation strategy with a game theory and Lagrange multiplier based methods.

Other contributions, such as [27]–[32], [34]–[36] tackled such an important research topic by leveraging the virtualization and quick reconfiguration capabilities provided by the NFV and SDN paradigms, respectively. Most contributions in this context aim to maximize the number of the accomplished request in each time interval [27]–[33] by solving the proposed optimization problem with meta-heuristic solutions. Furthermore, other works minimize the bandwidth usage with a greedy approach solution [34], minimize the end-to-end delay [35]–[37], or increase the link reliability [42].

Without a doubt, the current state of the art, as summarized in Table I, provides preliminary (but very valuable) approaches to the task offloading problem in NTN. However, all of the reviewed solutions present some shortcomings that must be addressed in the ongoing research activities. To begin, they usually develop a system model by assuming a quasi-static scenario, willing to manage the task offloading process within the single upcoming time slot. Differently, the mobility of satellites belonging to a given constellation requires the definition of more complex and sophisticated schemes supporting, for example, a looking-ahead optimization. Furthermore, the communication protocol enabling the interaction and the configuration into NTN segments is quite often neglected. On the contrary, this work would extend the current state of the art by offering concrete answers to these challenging research issues.

III. The proposed approach

This Section describes the proposed network architecture and the related protocol interaction enabling the on-demand provisioning of customized services into NTN segments. Then, it presents a novel looking-ahead optimization problem, based on an accurate system model describing the expected delays for deploying services, according to the actual parameter settings.

A. The reference network architecture

This work focuses on a NTN-based architecture in accordance with the recommendations discussed beginning with Release 17 [43] and progressing to the Beyond-5G and 6G scenarios [44], in which satellites host specific VNFs capable of processing the huge amount of data sent by heterogeneous NTN terminals deployed on the Earth's surface [45].

The resulting network architecture is composed of User, Edge, and Cloud Layers, as depicted in Fig. 1.

TABLE I
REVIEW OF RELATED WORKS.

References	Satellite Architecture Design	NTN Network	End-to-end Delay Optimization	Real time On-Demand VNF Deployment	Satellite visibility Constraints for LEO constellation	Looking Ahead Optimization	Protocol Stack Design
[11]				✓			
[12]				✓			
[13]			✓	✓			
[14]			✓	✓			
[15]			✓	✓			
[6]			✓	✓			
[18]				✓			
[17]				✓			
[19]				✓			
[16]				✓			
[20]		✓			✓		
[22]	✓	✓					
[21]	✓	✓			✓		
[25]		✓			✓		
[26]		✓	✓				
[23]		✓	✓				
[35]	✓	✓	✓		✓		
[24]		✓	✓				
[32]	✓	✓		✓			
[28]		✓		✓			
[34]		✓		✓			
[30]		✓		✓	✓		
[31]	✓	✓		✓	✓		
[36]	✓	✓	✓	✓	✓		
[37]	✓	✓	✓	✓			
[42]	✓	✓		✓			
[33]		✓		✓			
[29]	✓	✓		✓	✓		
[27]	✓	✓		✓	✓		
This work	✓	✓	✓	✓	✓	✓	✓

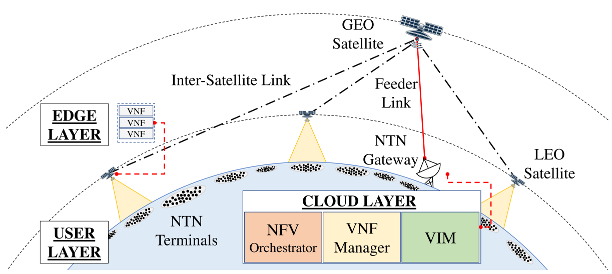


Fig. 1. The reference network architecture.

At the User Layer, NTN terminals sharing the same organization and geographic region and offering and/or offering the same service are grouped into the same cluster. In each cluster, the service provider deploys and configures a dedicated entity, namely master node, which is in charge of: i) realizing the need for a given cluster of NTN terminals to deliver data and use specific VNFs deployed onboard the satellite, ii) announcing the service request to the remote NFV Orchestrator hosted by the Cloud Layer, through space network elements belonging to the Edge Layer, iii) learning the outcome of the orchestration algorithm, and iv) announcing within the cluster of NTN terminals the presence of a satellite hosting a requested service. Without loss of generality, it is possible to assume that the interaction among NTN terminals and the master node is implemented through out-of-band communication, enabled (for example) by LoRaWAN or WiFi technologies.

The Edge Layer includes a constellation of Low Earth Orbit (LEO) CubeSats, Geostationary Orbit (GEO) satellites, and NTN gateways. Today, the constellation of LEO CubeSats is considered as the widely used low-cost satellite platform because of its low overall infrastructure deployment cost. Therefore, it concretely offers a cost-effective deployment of NTN segments, thus fostering connectivity also in remote areas of the Earth [46]. In the proposed network architecture, a LEO CubeSat represents the first contact point for the master node of a cluster, issuing a service request. Such a request must be delivered to the remote NFV Orchestrator in the Cloud Layer. Nevertheless, the constant movement of a LEO CubeSat makes intermittent the connection with the NTN terminals on the ground. Specifically, they can establish a connection with a given LEO CubeSat for a short period, namely visibility window, as depicted in Fig. 2. Furthermore, even the connection with the NTN gateways spread on the Earth is intermittent [47]. Indeed, to overcome this issue, the constellation of LEO CubeSats exploits inter-satellite links with a group of GEO satellites, granting connectivity even with the lack of a persistent feeder link with NTN gateways. As a result, the multi-hop connectivity established at the Edge Layer easily allows the LEO CubeSat to forward service requests to the Cloud Layer. Additionally, once configured (i.e., through the protocol architecture and optimization algorithm proposed in this work), LEO CubeSat can establish a connection with

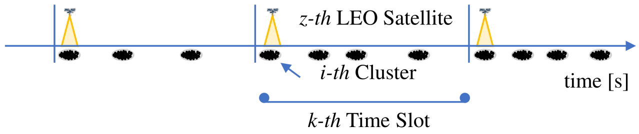


Fig. 2. Intermittent connectivity between terminals on the Earth and the LEO satellite constellation.

NTN terminals and provide them with the required services.

Finally, the Cloud Layer includes the NFV Orchestrator, the VNF Manager, and the Virtualised Infrastructure Manager (VIM). The NFV Orchestrator elaborates the VNFs deployment instruction borne by the VNF Manager. The VIM provides an interface through the SDN facilities that support the implementation of the optimization outcome by exploiting its comprehensive knowledge of the network topology. Thus, the Cloud Layer exploits these entities for the implementation of the optimization outcome by dynamically deploying VNFs onboard the LEO CubeSats [48].

TABLE II
LIST OF MAIN SYMBOLS USED IN THIS PAPER.

Symbol	Description
$\mathbf{\Psi}$	List of clusters, $L = \|\mathbf{\Psi}\|$
$\mathbf{\Sigma}$	List of LEO CubeSats, $S = \|\mathbf{\Sigma}\|$
$\mathbf{\Pi}$	List of available services
ψ_i	Master node of the i-th cluster
σ_z	z-th LEO CubeSat
$\pi_\mathbf{w}$	w-th service
$c(\sigma_z)$	Processing capability of the z-th LEO CubeSat
$m(\sigma_z)$	Memory capability of the z-th LEO CubeSat
$t(r_{k,f})$	Time slot in which the request has been generated
$\psi_i(r_{k,f})$	Master node of the i-th cluster that generated the request
$\pi_w(r_{k,f})$	Requested service
$\tau(r_{k,f})$	Provisioning upper bound delay
$\xi(r_{k,f})$	Processing load associated with request
$\zeta(r_{k,f})$	Memory load associated with the request
λ_i	Number of requests generated in a day for the i-th cluster
$\mathbf{V}(k)$	LEO CubeSat visibility matrix for the k-th time slot
$\mathbf{B}(k)$	Services deployment matrix for the k-th time slot
$\mathbf{R}(k)$	Set of pending requests at the k-th time slot
T_p	Time slot duration
$T(k)$	Time horizon at the k-th time slot

B. Protocol interaction

The protocol interaction between Terrestrial and non-Terrestrial entities belonging to User, Edge, and Cloud Layers is detailed in what follows across three consecutive phases, namely service request, configuration, and service provisioning (see Fig. 3).

Phase 1: Service Request. The master node realizes the need of a given cluster of NTN terminals to deliver data and use specific VNFs deployed onboard one of the available LEO CubeSat. As a consequence, the master node announces the service request to the first visible LEO CubeSat. Subsequently, the NTN gateway receives the forwarded request through the persistent feeder link ensured by the GEO satellite. Therefore, the NFV Orchestrator gathers all the received requests by the Edge Layer and collects them into a list of unaccomplished tasks.

Phase 2: Configuration. The Cloud Layer knows the set of unaccomplished tasks and their processing requirements, the network topology, and the computational resources available in LEO CubeSats. Based on these details, it implements an optimal allocation of VNFs (as discussed below), over the constellation of LEO CubeSats. Then, the NFV Orchestrator sends the optimization outcome to the VNF Manager, aiming at deploying the given VNFs in the constellation. Subsequently, the Virtualised Infrastructure Manager receives the deployment instruction forwarded by the VNF manager. Finally, the VNFs are deployed in the constellation through the feeder link ensured by the NTN gateways and the GEO satellites. The master node is finally informed about the deployment of the requested VNF onboard a specific LEO CubeSat.

Phase 3: Service Provisioning. At the beginning of the Service Provisioning phase, the master node notifies the cluster of NTN terminals about the presence of the given LEO CubeSat configured to receive and process (via dedicated VNFs) their data. As anticipated before, the communication between the master node and NTN terminals can be enabled through out-of-band communication, such as LoRaWAN or WiFi. Finally, the NTN terminals' communication can be supported by the requested services onboard that LEO CubeSat for the whole visibility window. Obviously, the set of unaccomplished tasks is updated by removing the services that are already deployed.

C. System model

Since the orbits of LEO CubeSats are independent of each other, the proposed system model considers a group of LEO CubeSats belonging to the same orbit of a constellation. This choice does not limit the generality of the presented methodology [47]. The main mathematical symbols used to formulate the system model are reported in Table II.

Indeed, let $\mathbf{\Sigma} = \{\sigma_1, \ldots, \sigma_S\}$ be the set of evenly spaced LEO CubeSats in the considered orbit and $S = \|\mathbf{\Sigma}\|$ be the number of the LEO CubeSats in the considered orbit. Likewise, the processing and memory capabilities of the z-th LEO CubeSat are expressed with $c(\sigma_z)$ and $m(\sigma_z)$, respectively. Similarly, let $\mathbf{\Psi} = \{\psi_1, \ldots, \psi_L\}$ be the list of the master nodes belonging to the clusters served by the LEO CubeSats of the considered orbit, with $L = \|\mathbf{\Psi}\|$ denoting the number of clusters. Finally, let $\mathbf{\Pi} = \{\pi_1, \ldots, \pi_W\}$ be the list of the different types of services provided by the Cloud Layer.

As well known, the amount of time required by a given LEO CubeSat to complete a revolution around the Earth depends on the height of its orbit. The motion of the LEO CubeSat, as Newton's form of Kepler's third law, that is: $T_o = 2\pi\sqrt{\frac{R^3}{GM_e}}$, where T_o represents the revolution time of the LEO CubeSat, R is the average radius of the orbit estimated as the distance from the center of the Earth, G is the gravitational constant, and M_e is the mass of the Earth [49]. Accordingly, a given same satellite can periodically communicate with a specific cluster of NTN terminals every T_o. Anyway, since the LEO CubeSats in the considered orbit are evenly spaced, the elapsed time between two consecutive visibility windows is stated as: $T_p = \frac{T_o}{S} = \frac{2\pi}{S}\sqrt{\frac{R^3}{G \cdot M_e}}$.

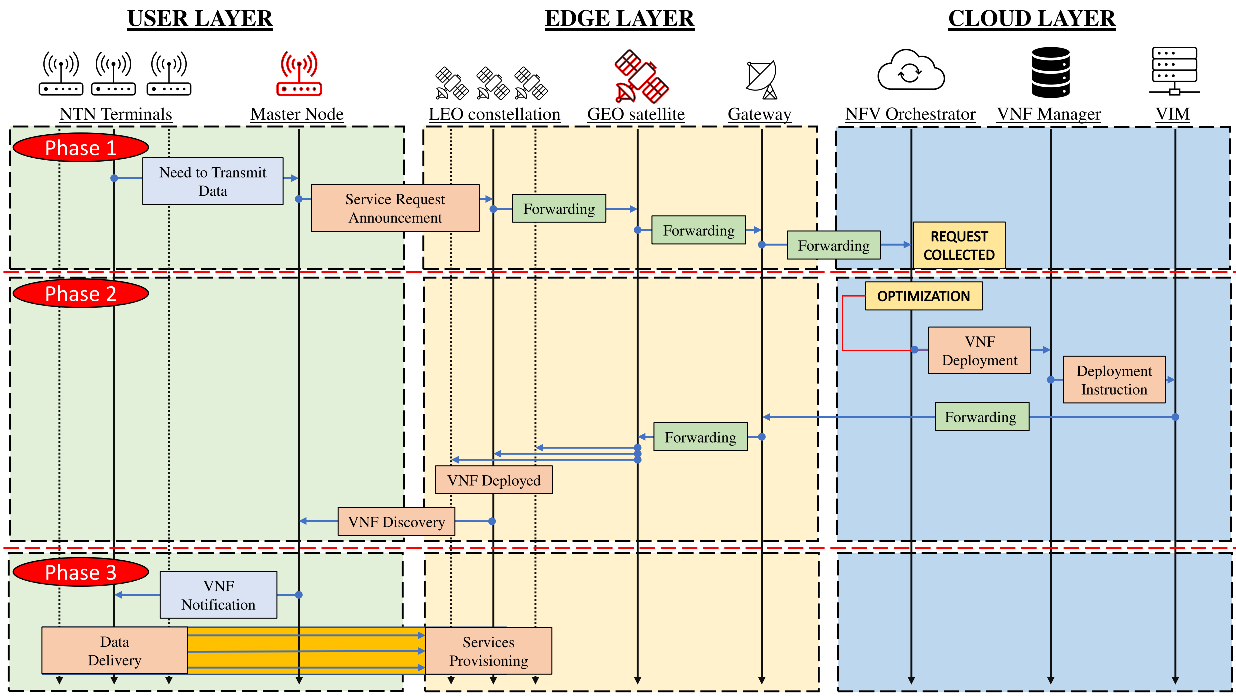


Fig. 3. Interaction among network entities.

Based on these premises, the proposed system model assumes to partition time into time slots, lasting T_p. Therefore, the consecutive steps belonging to the procedure described in Section III-A are implemented over consecutive time slots. Specifically, requests are collected by the NFV Orchestrator during the k-th time slot. Subsequently, the looking-ahead optimization problem (formally described in Section III-D) is solved within the $(k+1)$-th time slot. Finally, according to the solution of the optimization problem, the VNFs will be deployed onboard satellites in the upcoming time slots, i.e., starting from the $(k+2)$-th time slot and within a specified deadline.

The list of pending requests at the k-th time slot is denoted with $\mathbf{R}(k)$. Each request $r_{k,f} \in \mathbf{R}(k)$, where $F_k = \|\mathbf{R}(k)\|$ and $f = 1, \ldots, F_k$, (delivered to the NFV Orchestrator) includes the following information:

- the master node belonging to the cluster that generated the request, $\psi_i(r_{k,f})$,
- the requested services, $\pi_w(r_{k,f})$,
- the time slot in which the request has been generated, $t(r_{k,f})$,
- the upper bound delay for the provisioning of the requested services, $\tau(r_{k,f})$,
- the processing load associated with the request, $\xi(r_{k,f})$,
- the memory load associated with the request, $\zeta(r_{k,f})$.

Without loss of generality, it is assumed that $\tau(r_{k,f})$ is defined as a multiple of the time slot, that is $\tau(r_{k,f}) = nTp$.

As anticipated, a cluster can establish a communication with only one LEO CubeSat during a time slot and only for a short visibility window. Indeed, the proposed system model introduces the $\mathbf{V}(k)$ matrix to represent the reciprocal visibility between LEO CubeSats and clusters on the ground during the k-th time slot. Therefore, $v_{i,z}(k) = 1$, with $v_{i,z}(k) \in \mathbf{V}(k)$, if the z-th LEO CubeSat can communicate with the i-th cluster in the k-th time slot. Otherwise, $v_{i,z}(k) = 0$. The values of the $\mathbf{V}(k)$ matrix depend on the deploying position of both clusters and satellites in the network.

Furthermore, the services deployment matrix $\mathbf{B}(k) = \left(b_{w,z}(k)\right) \in \{0,1\}^{L \times S \times T(k)}$ contains boolean flags denoting if the z-th satellite hosts the w-th VNFs when $b_{w,z}(k) = 1$, with $b_{w,z}(k) \in \mathbf{B}(k)$. Otherwise, $b_{w,z}(k) = 0$. In this case, it depends on the outcome of the optimization problem, as defined in Section III-D.

D. Optimization problem

The goal of the optimization problem proposed in this work is to dynamically deploy VNFs onboard satellites while ensuring that:

- the deployment of the service must be performed within a strict deadline,
- for all the LEO CubeSats, the sum of the processing and memory requirements pertaining to the deployed VNFs, is never greater than their capabilities,
- the sum of all the experienced service provisioning delays is minimized.

As mentioned in Section II, prior works solely consider a quasi-static scenario and neglect the movement of LEO CubeSat throughout the optimization process. Differently, this work formulates a novel optimization problem willing to dynamically deploy VNFs across LEO CubeSats, over a looking-ahead horizon.

Given the list of pending requests, that is $\mathbf{R}(k)$, and the set of the request's upper bound delay, that is $\mathcal{T}(k) =$

$\{\tau(r_{k,1}),\ldots,\tau(r_{k,F_k})\}$, the proposed looking-ahead optimization algorithm considers an observation time interval $T(k)$ defined as:

$$T(k) = \max \mathcal{T}(k). \tag{1}$$

According to the proposed system model, $T(k)$ is a multiple of the duration of a single time slot T_p. In the k-th time slot, the partial service provisioning delay already accumulated by a generic request $r_{k,f}$ can be defined as the elapsed time between the generation and the current time slot, that is $k - t(r_{k,f})$. Such a delay may increase, slot by slot, till the actual VNFs provisioning onboard a given LEO CubeSat. Indeed, given the service deployment matrix for all the time slots available till the end of the observation time interval $T(k)$, the delay achieved by the service $r_{k,f}$ can be formally defined as:

$$\delta(k, r_{k,f}) = \sum_{\nu=1}^{T(k)} \sum_{\sigma_z \in \Sigma} \Big(b_{w,z}(\nu) \left[k + \nu - t(r_{k,f})\right] \Big), \tag{2}$$

while considering that $\sum_{\nu=1}^{T(k)} \sum_{\sigma_z \in \mathbf{\Sigma}} b_{w,z}(\nu) \geq 1,\ \forall r_{k,f} \in \mathbf{R}(k)$. Such an expression implicitly assumes that the request will be accomplished at least by a single LEO CubeSat within the time horizon $T(k)$.

It is important to note that messages exchanged among the involved entities experience communication delays. However, these delays can be safely considered much lower than the duration of the time slot (as discussed in detail in Section IV-D), which is used as the minimum time interval of interest for the conceived system model, the formulated optimization problem, and the resulting service provisioning. As a consequence, by assuming to observe the overall system on a time-slot basis, the impact of both network architecture and interaction flow is intrinsically taken into account.

The objective function to be minimized can be formally defined as the sum of all the partial delays related to each pending request, as reported in what follows:

$$\begin{aligned} U(k) &= \sum_{r_{k,f} \in \mathbf{R}(k)} \delta(k, r_{k,f}) = \\ &= \sum_{r_{k,f} \in \mathbf{R}(k)} \left[\sum_{\nu=1}^{T(k)} \sum_{\sigma_z \in \mathbf{\Sigma}} \Big(b_{w,z}(\nu) \left[k + \nu - t(r_{k,f})\right] \Big) \right]. \end{aligned} \tag{3}$$

Indeed, the dynamic allocation of VNFs across CubeSats, over a looking-ahead horizon, is formulated in this work through an Integer Linear Programming (ILP) problem:

$$P1 : \min_{\mathbf{R}\,\mathbf{\Sigma}} \sum_{r_{k,f} \in \mathbf{R}(k)} \left[\sum_{\nu=1}^{T(k)} \sum_{\sigma_z \in \mathbf{\Sigma}} \Big(b_{w,z}(\nu) \left[k + \nu - t(r_{k,f})\right] \Big) \right]$$

$$\text{s.t.} \sum_{r_{k,f} \in \mathbf{R}(k)} b_{w,z}(\nu)\, \xi(r_{k,f}) \leq c(\sigma_z), \forall \sigma_z, \nu \tag{4a}$$

$$\sum_{r_{k,f} \in \mathbf{R}(k)} b_{w,z}(\nu)\, \zeta(r_{k,f}) \leq m(\sigma_z), \forall \sigma_z, \nu \tag{4b}$$

$$\sum_{\sigma_z \in \mathbf{\Sigma}} \sum_{\nu = t(r_{k,f})+2}^{\tau(r_{k,f})} b_{w,z}(\nu)\, v_{i,z}(\nu) = 1, \forall r_{k,f} \tag{4c}$$

$$b_{w,z}(\nu) \leq v_{i,z}(\nu), \forall \pi_w, \sigma_z, \nu, \tag{4d}$$

where (4a) specifies that the sum of the computing requirements of the VNFs deployed on a given LEO CubeSat cannot exceed the CPU processing capabilities of the considered satellite. Furthermore, (4b) states that the total amount of memory used by the allocated VNFs on a given LEO CubeSat in any time slot must not exceed its memory capability. Moreover, (4c) ensures that a VNF requested by a pending request is deployed, by the NFV Orchestrator, no later than its deadline, beginning at least two time slots after the request is generated (allowing for collection and optimization). Finally, (4d) mandates the NFV Orchestrator to deploy the VNFs only onboard specific LEO CubeSat capable of communicating with the cluster within the deadline, specifically during its visibility window.

IV. Performance evaluation

This Section investigates the effectiveness of the proposed methodology in different scenarios and through computer simulations.

First of all, it is important to remark that the non-convex optimization problem formulated in Section III-D is NP-hard. Indeed, a brute force strategy can be hypothetically used to test all the binary combinations of the 3D matrix (i.e., $\mathbf{B}(k) = \big(b_{w,z}(k)\big) \in \{0,1\}^{L \times S \times T(k)}$), every time slot and for the overall observation time interval. Such an approach, however, is feasible only for simple scenarios (e.g., with few LEO CubeSats and few clusters on the ground). More in general, instead, the optimal solution cannot be retrieved in a polynomial time.

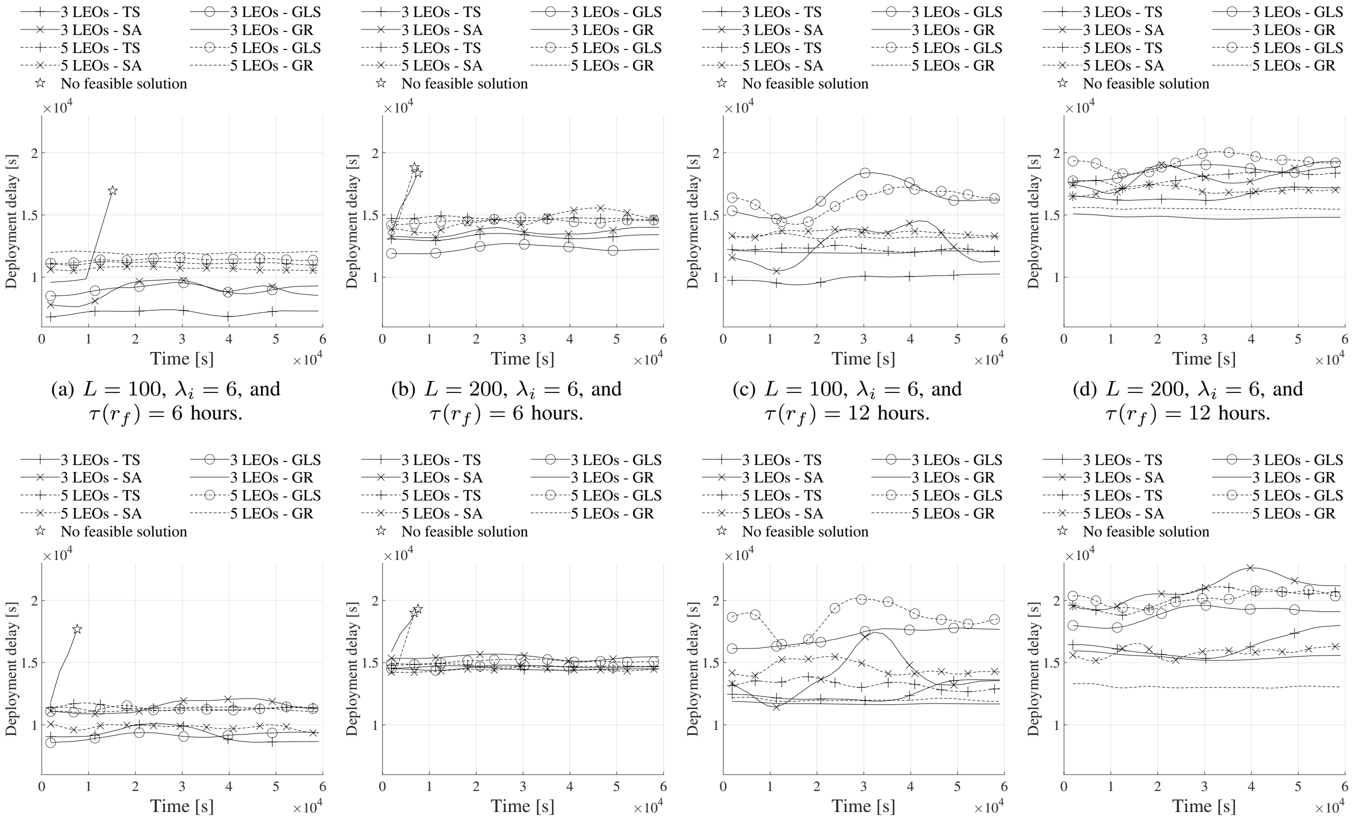
At the same time, conventional optimization frameworks like Gurobi, CVXR, or Casadi cannot be used in this context because they work with decision variables represented by 2D matrices, whereas the proposed approach deals with decision variables in the form of a 3D matrix.

To bridge this gap and solve the formulated optimization problem three different heuristic strategies, inspired by well-known meta-heuristic approaches (such as TS [39], SA [40], and GLS [41]) have been properly developed and tested through a custom Python tool. More details about the implemented algorithms can be found in Appendix A.

The behavior of the developed strategies has been also compared with respect to a benchmark scheme, namely GR algorithm. Specifically, it deploys the required VNFs on the first available LEO CubeSat with sufficient memory and processing power without utilizing any optimization methods.

A. The considered use case

The 3GPP, starting from Rel-15 [50], began exploring the potential of a new communication standard for NTNs, by examining various deployment scenarios and challenges. At the time of writing, Rel-17 [43], [51] is investigating key issues related to business roles, service, and network management when orchestrating services in the space segment [52]. The ease of deploying LEO CubeSats satellite constellations is driving the growth of tailored services for various companies [53], [54]. Indeed, since NTN segments are called to securely manage an ever-growing amount of data, connections,



(a) $L = 100$, $\lambda_i = 6$, and $\tau(r_f) = 6$ hours.
(b) $L = 200$, $\lambda_i = 6$, and $\tau(r_f) = 6$ hours.
(c) $L = 100$, $\lambda_i = 6$, and $\tau(r_f) = 12$ hours.
(d) $L = 200$, $\lambda_i = 6$, and $\tau(r_f) = 12$ hours.
(e) $L = 100$, $\lambda_i = 12$ and $\tau(r_f) = 6$ hours.
(f) $L = 200$, $\lambda_i = 12$ and $\tau(r_f) = 6$ hours.
(g) $L = 100$, $\lambda_i = 12$, and $\tau(r_f) = 12$ hours.
(h) $L = 200$, $\lambda_i = 12$, and $\tau(r_f) = 12$ hours.

Fig. 4. Deployment delay of security services.

and services, it is extremely important to envisage novel methodologies for the dynamic and optimal deployment of VNFs properly devoted to security functionalities (including authentication, authorization, firewall, intrusion detection systems, intrusion prevention systems, and so on). Of course, given the limited amount of resources on the LEO CubeSats, these security services should only be activated as needed to conserve onboard resources [55]. To this aim, security VNFs are considered to demonstrate the effectiveness of the proposed approach in a realistic use case. It's worth noting that our scenario does not restrict the use of the proposed approach in other contexts.

B. Parameter setting

The study has been conducted by varying the number of clusters deployed on the Earth's surface, the number of LEO CubeSats in the orbit, and the average number of services issued in a day by each cluster.

The altitude of LEO CubeSats is set to 500 km. The corresponding orbit period is equal to $T_o = 5676$s. Moreover, the number of LEO CubeSats, that is S, in the orbit, is set to 3 and 5. Indeed, based on the number of satellites per orbit, T_p is equal to:

$$T_p = \begin{cases} 1892\, s, & \text{if } S = 3, \\ 1135\, s, & \text{if } S = 5. \end{cases} \tag{5}$$

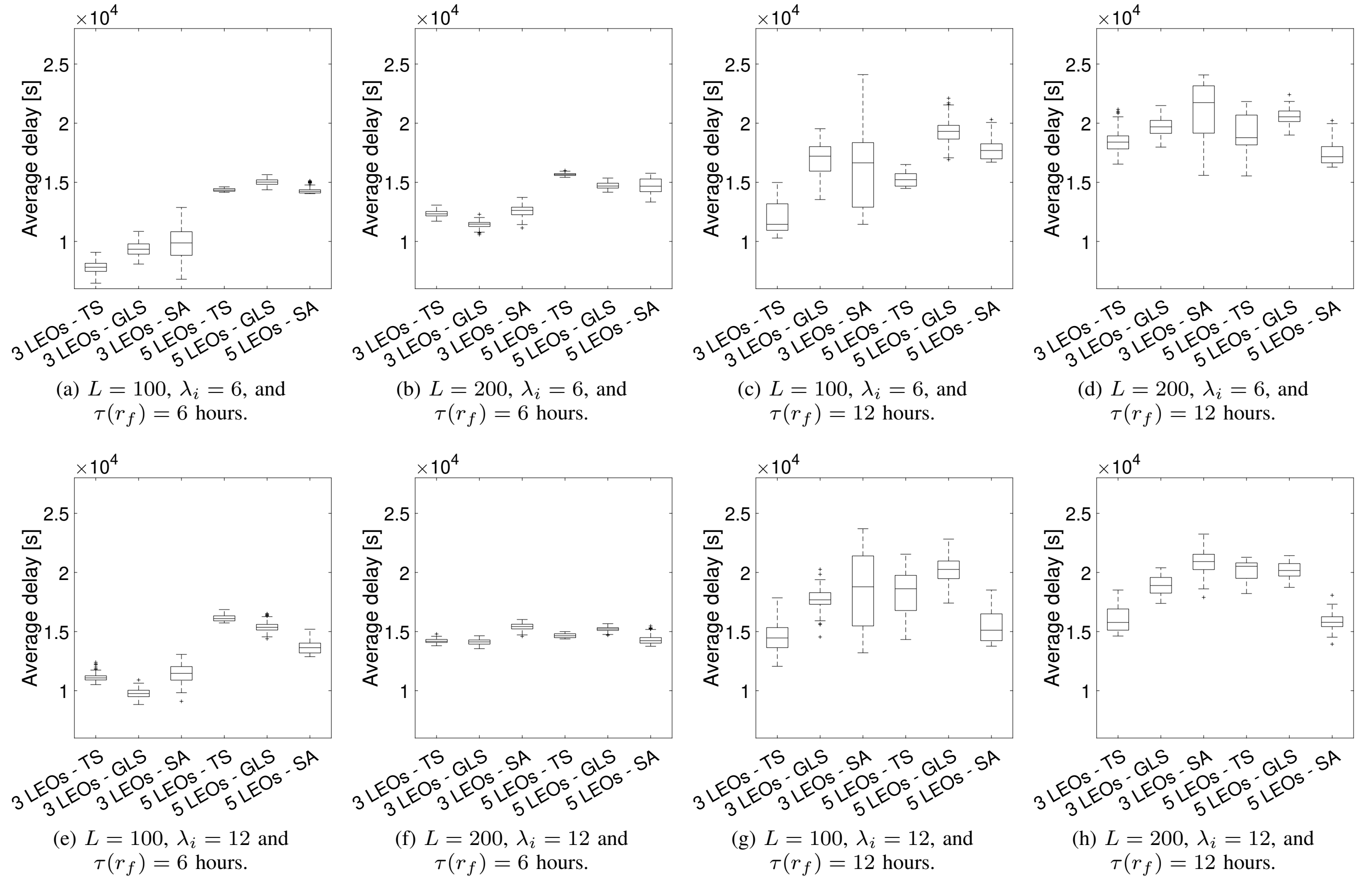
TABLE III
COMPUTATIONAL CAPABILITIES EXPOSED BY LEO CUBESATS FOR THE CONSIDERED SERVICES [56].

Symbol	Parameter	Value
$m(\sigma_z)$	Memory capability	64 GB
$c(\sigma_z)$	CPU processing capability	128 Gigacycles/s
β^c	Maximum aggregated throughput	92 kbps

Considering the time slot duration, the impact of processing and transmission time, in the order of milliseconds, can be negligible.

It is assumed that each LEO CubeSat is equipped with a PowerEdge R6515 with AMD EPYC 7702P, 2.00 GHz, 64 core, and 64 GB RAM [56], as indicated in Table III.

In the considered scenario, the clusters are uniformly distributed on the Earth's surface covered by the considered orbit. In particular, the number of clusters L varies between 100 and 200. Let λ_i be the average number of requests generated by the $i-$th cluster in a day, set to 6 and 12 events per day. Accordingly, the average number of requests received by the Cloud Layer in a day is equal to $\lambda = \lambda_i L$. Data generated by terminals belonging to each cluster can be processed by means of services summarized in Table IV. The service is chosen randomly for each request. The expected computational capabilities of VNFs strictly depend on the amount of data generated by each cluster. The proposed study considers a



(a) $L = 100$, $\lambda_i = 6$, and $\tau(r_f) = 6$ hours.
(b) $L = 200$, $\lambda_i = 6$, and $\tau(r_f) = 6$ hours.
(c) $L = 100$, $\lambda_i = 6$, and $\tau(r_f) = 12$ hours.
(d) $L = 200$, $\lambda_i = 6$, and $\tau(r_f) = 12$ hours.
(e) $L = 100$, $\lambda_i = 12$ and $\tau(r_f) = 6$ hours.
(f) $L = 200$, $\lambda_i = 12$ and $\tau(r_f) = 6$ hours.
(g) $L = 100$, $\lambda_i = 12$, and $\tau(r_f) = 12$ hours.
(h) $L = 200$, $\lambda_i = 12$, and $\tau(r_f) = 12$ hours.

Fig. 5. Confidence interval for the deployment delay of security services.

TABLE IV
SECURITY SERVICES REQUIREMENTS FOR VNFS IMPLEMENTATION.

Provider	Service	ξ^c [cycles/bit]	$\zeta(r_f)$ [GB]
Fortigate VM [57]	NGFW	9	4
	IPSec VPN	14.5	2
	Threat Prot.	11.3	4
Cisco ASAv [58]	Stateful IDS	4.2	4
	AES VPN	6.9	2
Juniper vSRX [59]	FW	2.3	2
	IPS	2.4	2
	APPMonitor	1.5	2
Others	Snort IDS/IPS [60]	9.5	2
	OpenVPN AES-NI [61]	31	4

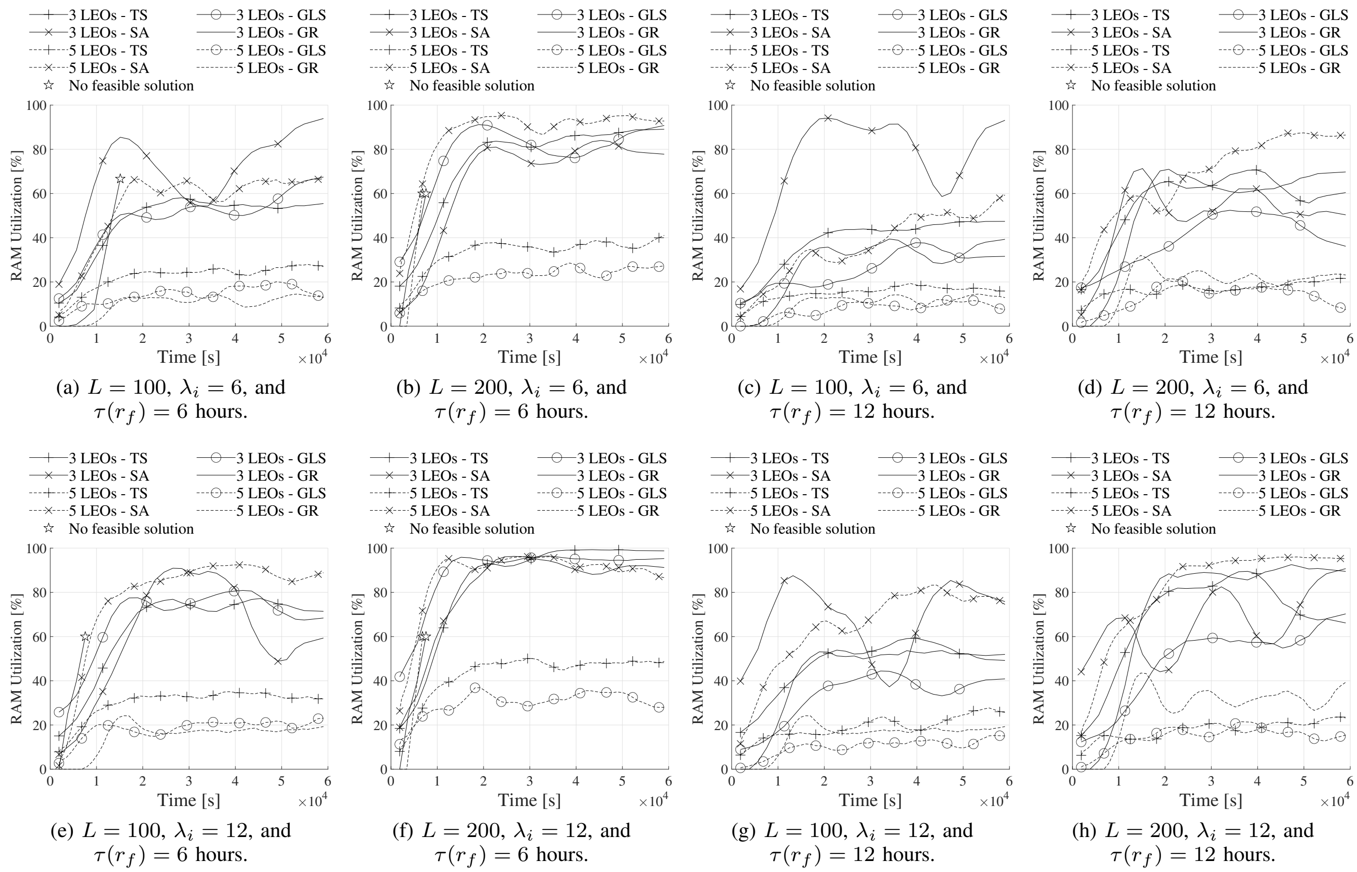
worst-case scenario, in which each cluster exploits the overall available satellite bandwidth during the related visibility time to send data, denoted with β^c bps. Therefore, given the CPU cycles required to process one bit, that is ξ^c cycles/bit, the overall processing requirement of a service is equal to $\xi(r_f) = \xi^c \cdot \beta^c$ cycles/s [6]. The conducted study assumes that the radio access technology used in the link between LEO CubeSat and clusters is the Narrow-Band IoT (NB-IoT). In this case, $\beta^c = 92$kbps [62].

C. KPIs

The evaluated Key Performance Indicators (KPIs) include:

- **deployment delay of services**: it represents the amount of time (expressed as a multiple of the time slot) needed to deploy a VNF requested by a group of NTN terminals on a specific LEO CubeSat. In general, and according to the design principles at the basis of the conceived approach, it is expected the deployment delay ranges from a minimum of two time slots (one slot is required to deliver the request to the Cloud Layer during the Phase 1 and the other one is required to elaborate the deployment instructions during the Phase 2) to the expected upper bound delay,
- **percentage of computational resources (including RAM and CPU) consumed by LEO CubeSats for hosting the deployed VNFs**: it represents the amount of RAM and the processing capabilities utilized onboard each LEO CubeSat in relation to the total amount of resources hosted in the space segment,
- **average processing time of each heuristic algorithm in solving the optimization problem**: it represents the amount of time (expressed in seconds) spent by the Cloud Layer in evaluating the optimal deployment of VNFs requested by the pending services.

Computer simulations consider an observation period of 60000s, embracing multiple visible time intervals. Moreover, the analysis of deployment delay of requested services and the percentage of computational resources consumed onboard



(a) $L = 100$, $\lambda_i = 6$, and $\tau(r_f) = 6$ hours.
(b) $L = 200$, $\lambda_i = 6$, and $\tau(r_f) = 6$ hours.
(c) $L = 100$, $\lambda_i = 6$, and $\tau(r_f) = 12$ hours.
(d) $L = 200$, $\lambda_i = 6$, and $\tau(r_f) = 12$ hours.
(e) $L = 100$, $\lambda_i = 12$, and $\tau(r_f) = 6$ hours.
(f) $L = 200$, $\lambda_i = 12$, and $\tau(r_f) = 6$ hours.
(g) $L = 100$, $\lambda_i = 12$, and $\tau(r_f) = 12$ hours.
(h) $L = 200$, $\lambda_i = 12$, and $\tau(r_f) = 12$ hours.

Fig. 6. RAM utilization.

satellite has been organized into two parts. The former focuses the attention on a specific simulation run (e.g., network realization) and illustrates the considered KPI as a function of time. The latter, instead, reports minimum, average, and maximum values, as well as the 25th and 75th percentile of the measured KPI, obtained by considering numerous realizations.

D. Deployment delay of security services

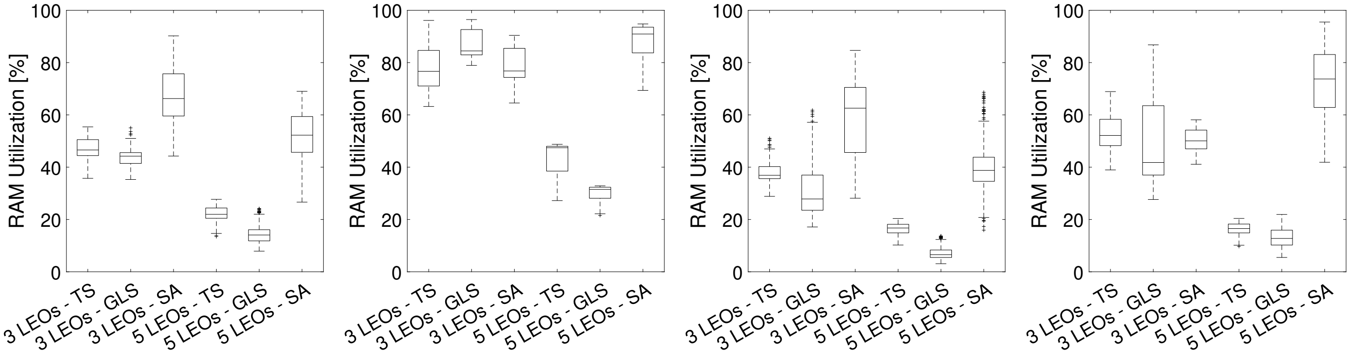
Fig. 4 shows the deployment delay of the deployed security services, experienced for a single specific test. Moreover, reported results have been obtained by averaging the deployment delays experienced by each service request across moving windows of 5 time slots each.

Looking at the behavior of the benchmark scheme, that is the GR algorithm, it is possible to observe that it cannot provide a feasible solution in several scenarios, including the one with 3 LEO CubeSats, 100 clusters on the ground, and an upper bound delay equal to $\tau(r_{k,f}) = 6$ hours, as well as the one with the same upper bound delay, 200 clusters on the ground, and 5 LEO CubeSats. All these negative results are achieved because the GR algorithm tends to overload LEO CubeSats thus being not able to accept further requests.

On the contrary, the proposed approach always guarantees feasible solutions, independently from the heuristic algorithm adopted to find a solution for the formulated optimization problem. More specifically, obtained results show that the deployment delay increases with the number of clusters. In fact, the more clusters interact with the NTN segment, the higher the number of pending requests to be handled by the NFV Orchestrator at each optimization round. In these conditions, the NFV Orchestrator may encounter a lack of available resources across the first visible LEO CubeSats. Therefore, it is obliged to delay the deploy the VNFs by considering other future time slots (according to the conceived looking-ahead time horizon logic). Indeed, a longer deadline introduces a higher extent of the solution space, bringing an increment of deployment delays. Furthermore, regardless of the number of LEO CubeSats in the orbit, the deployment delay remains almost the same in the majority of the investigated scenarios. It relies on the exploration strategy of the optimal solution in the solution space for each heuristic approach.

To provide a further insight, additional tests have been conducted to collect performance levels by considering multiple realizations. Indeed, Fig. 5 depicts minimum, average, and maximum values of the experienced deployment delays, together with both the 25th and 75th percentile. The analysis of the GR algorithm has been omitted in this case because the previous analysis already highlighted its inability to provide feasible solutions in the most of investigated scenarios.

Regarding the proposed approach, instead, all the obtained results confirm the analysis reported in Fig. 4. Moreover, it also shows that the greatest deviation from the average value (up to 30%) is registered in scenarios with 3 satellites and when the SA method is used. Conversely, with 5 satellites,

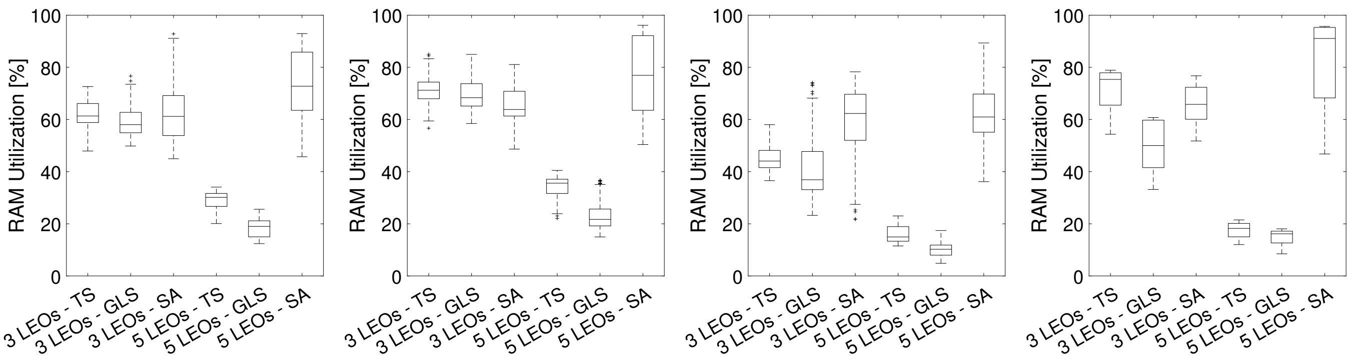


(a) $L = 100$, $\lambda_i = 6$, and $\tau(r_f) = 6$ hours.

(b) $L = 200$, $\lambda_i = 6$, and $\tau(r_f) = 6$ hours.

(c) $L = 100$, $\lambda_i = 6$, and $\tau(r_f) = 12$ hours.

(d) $L = 200$, $\lambda_i = 6$, and $\tau(r_f) = 12$ hours.



(e) $L = 100$, $\lambda_i = 12$ and $\tau(r_f) = 6$ hours.

(f) $L = 200$, $\lambda_i = 12$ and $\tau(r_f) = 6$ hours.

(g) $L = 100$, $\lambda_i = 12$, and $\tau(r_f) = 12$ hours.

(h) $L = 200$, $\lambda_i = 12$, and $\tau(r_f) = 12$ hours.

Fig. 7. Confidence interval for the RAM utilization.

the maximum deviation (up to 13%) is obtained with the TS method. In any case, obtained results always demonstrate the ability of the developed heuristic strategy to meet the expected quality of service constraint in all the considered scenarios.

To conclude, it has emerged that TS outperforms the other approaches in almost all scenarios with a low number of LEO CubeSats (i.e., 3) in the orbit, by reducing up to 25% the deployment delay of services. On the contrary, SA well suits scenarios with a higher number of LEO CubeSats (i.e., 5), for which it reduces up to 20% of the measured deployment delay.

E. Percentage of computational resources consumed by LEO CubeSats

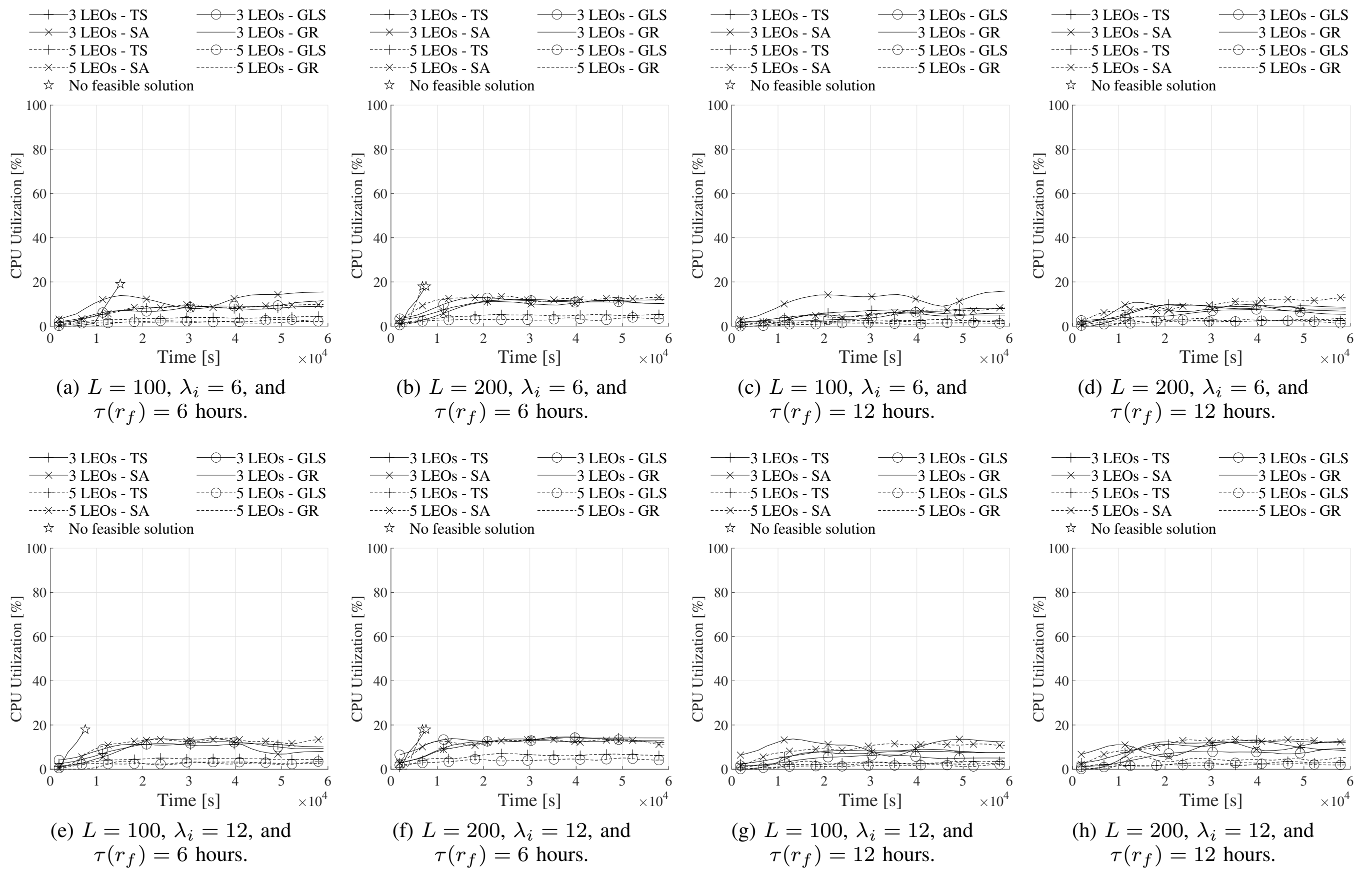
Fig. 6 shows the percentage of RAM consumed by the deployed VNFs over time. Also in this case, reported curves have been generated by considering a single specific test and by averaging the measured KPI among all the satellites and across moving windows of 5 time slots each. It is important to remark that Fig. 6 depicts a specific realization that is a time-varying process and jointly influenced by the random number generator adopted by the simulation tool, the statistical generation of service requests (see, for instance, the variable λ_i of the i-th cluster), and the chosen optimization strategy. Peaks in Fig. 6 are registered when the network is handling a larger number of requests.

In line with the previous comments, the GR algorithm does not present feasible solutions for all the investigated scenarios. In those (few) configurations where it provides an effective deployment of VNFs, the percentage of consumed RAM is comparable with respect to results registered by the strategies proposed in this paper.

As expected, the higher the number of served clusters, the higher the resulting memory consumption onboard the satellites. Furthermore, a wider deadline and higher number of LEO CubeSats in the considered orbit causes a higher extent of the set of feasible solutions to explore. As a result, the memory resources are allocated less efficiently by each heuristic algorithm. It is noteworthy to highlight that the SA-based strategy registers better allocation of memory resources in scenarios with a higher number of satellites.

On the other hand, Fig. 8 shows the percentage of the CPU usage. Due to the low data rate of the considered radio access technology (i.e., NB-IoT), the processing capability never represents a blocking condition for the deployment of the VNFs.

With reference to parallel tests, Fig. 7 and Fig. 9 depict minimum, average, and maximum values, as well as the 25th and 75th percentile, of both RAM and CPU usage, respectively. Here, Fig. 7 highlights that the SA-based strategy exhibits the most significant deviations from the mean value, both for the configurations involving 3 LEO CubeSats and 5 LEO CubeSats. Notably, within the multiple realizations, the largest deviation recorded is up to 50% and 60% from the mean value, respectively. Fig. 9, instead, illustrates that the CPU usage remains consistently below 20%, with a negligible



(a) $L = 100$, $\lambda_i = 6$, and $\tau(r_f) = 6$ hours.
(b) $L = 200$, $\lambda_i = 6$, and $\tau(r_f) = 6$ hours.
(c) $L = 100$, $\lambda_i = 6$, and $\tau(r_f) = 12$ hours.
(d) $L = 200$, $\lambda_i = 6$, and $\tau(r_f) = 12$ hours.
(e) $L = 100$, $\lambda_i = 12$, and $\tau(r_f) = 6$ hours.
(f) $L = 200$, $\lambda_i = 12$, and $\tau(r_f) = 6$ hours.
(g) $L = 100$, $\lambda_i = 12$, and $\tau(r_f) = 12$ hours.
(h) $L = 200$, $\lambda_i = 12$, and $\tau(r_f) = 12$ hours.

Fig. 8. CPU utilization.

deviation from the mean value. Note that, also in this case, the analysis of the GR algorithm has been omitted because of its inability to provide feasible solutions for all the considered scenarios.

F. Processing time

The processing time emphasizes the computational burden required by each heuristic algorithm to find a solution to the optimization problem. It has been evaluated with a computer Intel(R) Xeon(R) Bronze 3106 CPU with 16 cores at 1.70GHz and 92 GB of RAM. In this case, the number of clusters deployed on the ground is set to 100, 150, and 200. Results are shown in Fig. 10.

The evaluation of the processing time for the GR algorithm is negligible because it just defines the VNFs allocation without implementing any time-consuming task.

Particularly important is the processing time for the proposed approach. The developed heuristic strategies, in fact, are called to provide an optimal and feasible solution within a threshold represented by the time slot duration. Only in this way, they will be able to trigger the deployment of VNFs onboard LEO CubeSats in time.

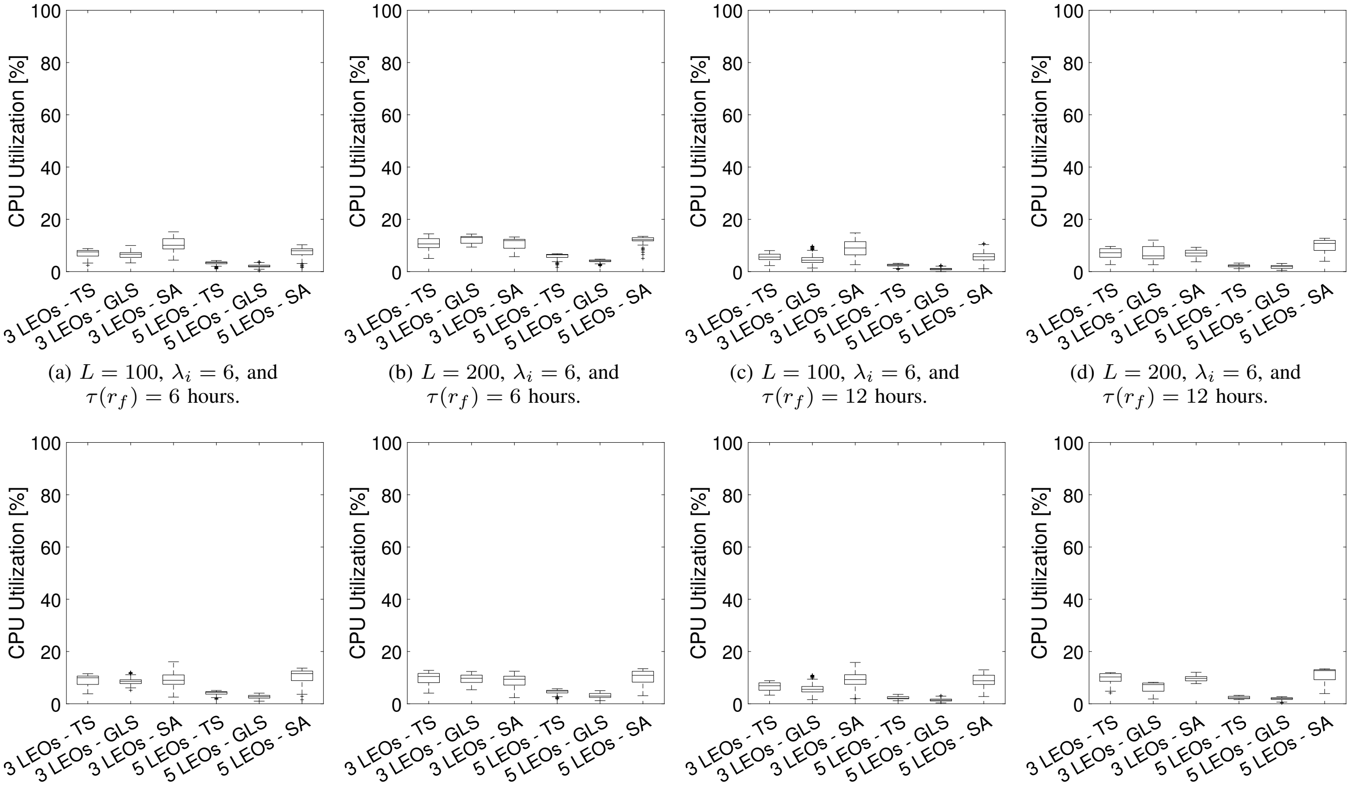
From the analysis of Fig. 10, it is possible to observe that only the TS-based strategy reaches a high value of processing time, very close to the threshold (i.e., the time slot duration). Indeed, it is the heaviest heuristic approach in terms of processing load for each considered scenario as demonstrated by the results. Nevertheless, by jointly taking into account all the considered KPIs, since the SA-based strategy is able to find an optimal solution in a lower time than other approaches, it represents the best choice for both processing time and deployment delay of services, by saving up 95% in processing time than TS.

G. Comparison with the optimal solution

In conclusion, to further demonstrate the effectiveness of the proposed approaches against the optimal solution, a comparison in smaller scenarios has been conducted. In detail, the number of clusters on the ground L is set to 5, the upper bound delay $\tau(r_f)$ is equal to 6 hours, λ_i is set to 1, and the number of LEO CubeSat S ranges from 2 to 5. Fig. 11 depicts how these heuristic schemes are able to produce results comparable to those expected by an optimal solution. Obtained results allow to trust the effectiveness of the heuristic strategies also in more complex scenarios.

V. Conclusion

This paper proposed a novel 6G-oriented architecture with advanced orchestration capabilities of security services into the Non-Terrestrial segment. Specifically, it provides these main scientific contributions: i) a definition of network architecture and protocol stack enabling the interaction among terrestrial and space network entities, ii) a definition of a system model



(a) $L = 100$, $\lambda_i = 6$, and $\tau(r_f) = 6$ hours.

(b) $L = 200$, $\lambda_i = 6$, and $\tau(r_f) = 6$ hours.

(c) $L = 100$, $\lambda_i = 6$, and $\tau(r_f) = 12$ hours.

(d) $L = 200$, $\lambda_i = 6$, and $\tau(r_f) = 12$ hours.

(e) $L = 100$, $\lambda_i = 12$ and $\tau(r_f) = 6$ hours.

(f) $L = 200$, $\lambda_i = 12$ and $\tau(r_f) = 6$ hours.

(g) $L = 100$, $\lambda_i = 12$, and $\tau(r_f) = 12$ hours.

(h) $L = 200$, $\lambda_i = 12$, and $\tau(r_f) = 12$ hours.

Fig. 9. Confidence interval for the CPU utilization.

describing the network configuration and the delays associated with the deployment of specific security services, and iii) formulation of optimization problem willing to dynamically allocate security VNFs among satellites over a looking-ahead horizon. Three alternative heuristic methods for the aforementioned optimization problem have been investigated through computer simulation to assess the overall performance of the proposed approach. Obtained results demonstrated the ability of the conceived approach to deploying the requested services within a strict deadline. Specifically, the SA-based solution demonstrates to outperform the other approaches in terms of service deployment delays, resource consumption, and processing time. Future research activities will investigate more complex scenarios envisaging a deep integration of terrestrial and NTN, embracing other space network elements (such as drones). Moreover, they will also evaluate the effectiveness of the proposed solution through real experimental testbeds.

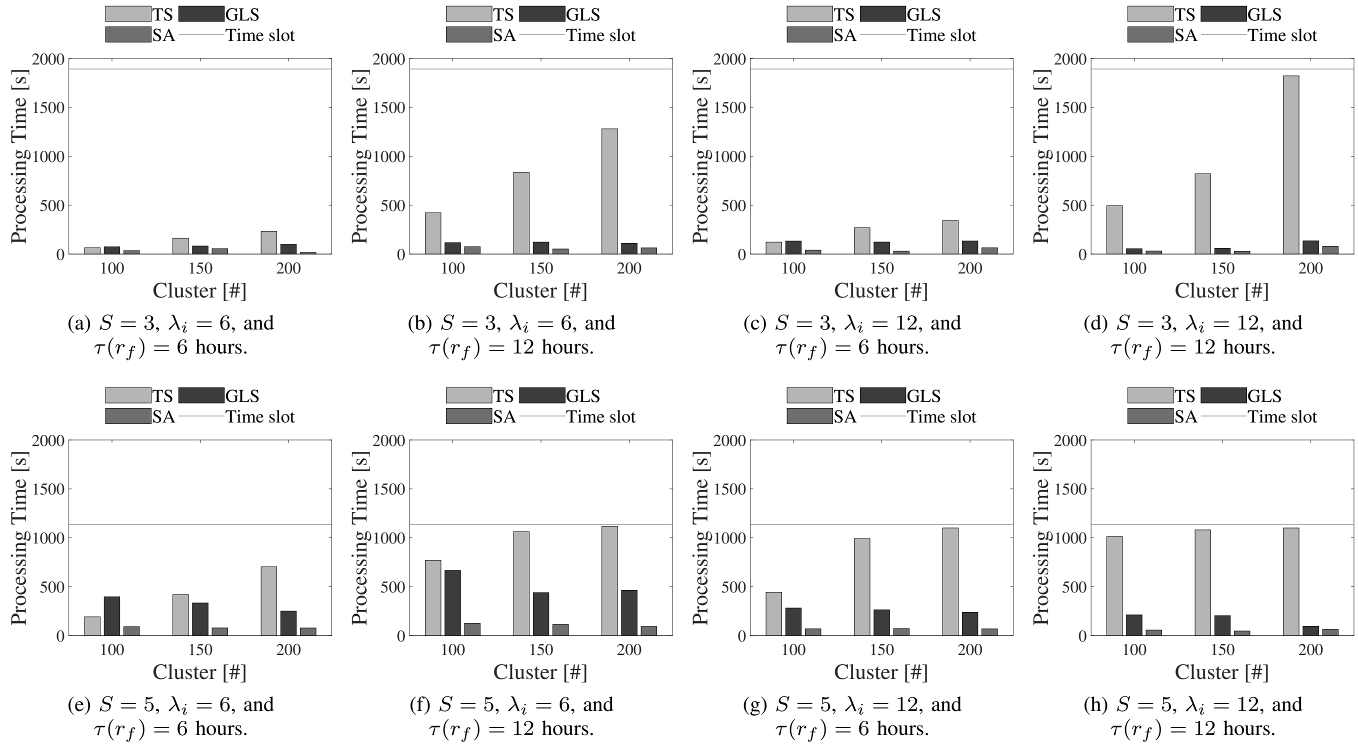
Appendix A
Heuristic Implementation Details

The design and the implementation of the three heuristics are based on the following common methods:

- As a first step, it is introduced a way to estimate the utility function starting from a feasible solution, by exploiting the 3D matrix as input.
- Then, it has been implemented a method that helps to explore the neighbourhood of a given solution (i.e., swap move) by applying a little variation from the initial one and by checking its feasibility. For example, this is done by assuming to deploy a given VNF on another LEO CubeSat of the orbit, as well as by assuming to make available that VNF in a different time slot. It is worth noting that the parameters of each heuristic approach have been determined experimentally throughout the simulations to ensure the optimal performance for each method.
- The iteration ends upon reaching a specific criterion, which varies depending on the algorithm, as detailed below.

More specifically, the technique based on the TS meta-heuristic approach starts with an initial random solution and proceeds through a sequence of swap moves that lead to a new solution inside the neighbourhood of the current one, with the utility function assuming a value smaller than the selected value. To avoid the trap of local minimum, TS permits ”worsening moves”. However, it is possible to risk sliding back into the local minimum quickly after. To cope with this issue, it is crucial to make the last moves in the search path ”forbidden”, so that the algorithm cannot retrace its steps and fall back into the local minimum. Specifically, its stopping condition is verified when the same solution (i.e., $\mathbf{B}(k)$) is elected as the best solution on two different iterations in a row since exploring the same neighbourhood more than one time is pointless.

The technique based on the SA meta-heuristic approach



(a) $S = 3$, $\lambda_i = 6$, and $\tau(r_f) = 6$ hours.
(b) $S = 3$, $\lambda_i = 6$, and $\tau(r_f) = 12$ hours.
(c) $S = 3$, $\lambda_i = 12$, and $\tau(r_f) = 6$ hours.
(d) $S = 3$, $\lambda_i = 12$, and $\tau(r_f) = 12$ hours.
(e) $S = 5$, $\lambda_i = 6$, and $\tau(r_f) = 6$ hours.
(f) $S = 5$, $\lambda_i = 6$, and $\tau(r_f) = 12$ hours.
(g) $S = 5$, $\lambda_i = 12$, and $\tau(r_f) = 6$ hours.
(h) $S = 5$, $\lambda_i = 12$, and $\tau(r_f) = 12$ hours.

Fig. 10. Processing time.

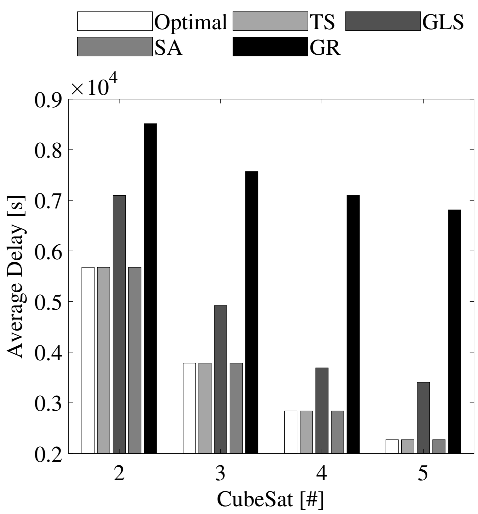


Fig. 11. Comparison with the optimal solution with $L = 5$ and $\tau(r_f) = 6$ hours.

begins with the generation of a random solution. After that, a new feasible one is generated by performing a swap move starting from the original. If the new solution's utility function is less than the initial one, it is accepted as the new solution. Otherwise, it can still be accepted with a decreasing likelihood as the search duration increases. If this probability is dropped too fast, the algorithm faces the risk of being stuck in a local minimum. On the other side, increasing the chance of adopting worsening solutions too slowly, lengthens the total search time. Specifically, to reach the equilibrium state in a sufficient time, the likelihood is imposed by the Metropolis Criterion, which is equal to $\min\{1, e^{-(\frac{U_2-U_1}{T})}\}$, where U_1 and U_2 express the value of the utility function of starting solution of the iteration and the newly generated one, respectively. T is the decreasing factor that drives the final equilibrium state. Finally, the maximum number of iterations is equal to 10000.

Furthermore, the technique based on the GLS meta-heuristic approach represents a search algorithm that generates solutions to problems by using strategies inspired by natural genetic populations. The basic idea is to create a population composed of random feasible solutions to a given problem. In particular, each element in the population can be utilized to generate new members of the population through crossover or mutation, specifically, by combining two alternative solutions to obtain another one. In detail, the optimal solution search starts with a population of 400 feasible solutions, which are then used to define a set of 200 parents. Finally, for each iteration, 80 new solutions are created by combining the others contained in the set of parents. The maximum number of iterations for this algorithm is set equal to 100.

ACKNOWLEDGMENT

This work was partially supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, with particular reference to the partnership on ”Telecommunications of the Future” (PE00000001 - program ”RESTART”, CUP: D93C22000910001) and the national center on ”Sustainable Mobility” (CN00000023 - program “MOST”, CUP: D93C22000410001). It was also supported by the PRIN project no. 2017NS9FEY entitled ”Realtime Control of 5G

Wireless Networks: Taming the Complexity of Future Transmission and Computation Challenges" funded by the Italian MUR, by "The house of emerging technologies of Matera (CTEMT)" project funded by the Italian MIMIT, and by the PON AGREED projects (ARS01 00254) funded by the Italian MUR.

REFERENCES

[1] 3gpp, "System architecture for the 5G System (5GS) - 3GPP TS 23.501 version 17.4.0 Release 17," 3rd Generation Partnership Project, Technical Specification (TS) 36.331, 05 2022, version 17.4.0.

[2] G. Araniti, A. Iera, S. Pizzi, and F. Rinaldi, "Toward 6G Non-Terrestrial Networks," *IEEE Network*, vol. 36, no. 1, pp. 113–120, 2022.

[3] M. Giordani and M. Zorzi, "Non-Terrestrial Networks in the 6G Era: Challenges and Opportunities," *IEEE Network*, vol. 35, no. 2, pp. 244–251, 2021.

[4] E. Yaacoub and M.-S. Alouini, "A Key 6G Challenge and Opportunity—Connecting the Base of the Pyramid: A Survey on Rural Connectivity," *Proceedings of the IEEE*, vol. 108, no. 4, pp. 533–582, 2020.

[5] H. Cui, J. Zhang, Y. Geng, Z. Xiao, T. Sun, N. Zhang, J. Liu, Q. Wu, and X. Cao, "Space-air-ground integrated network (SAGIN) for 6G: Requirements, architecture and challenges," *China Communications*, vol. 19, no. 2, pp. 90–108, 2022.

[6] R. Doriguzzi-Corin, S. Scott-Hayward, D. Siracusa, M. Savi, and E. Salvadori, "Dynamic and Application-Aware Provisioning of Chained Virtual Security Network Functions," *IEEE Transactions on Network and Service Management*, vol. 17, no. 1, p. 294–307, Mar. 2020.

[7] Q.-V. Pham, R. Ruby, F. Fang, D. C. Nguyen, Z. Yang, M. Le, Z. Ding, and W.-J. Hwang, "Aerial Computing: A New Computing Paradigm, Applications, and Challenges," *IEEE Internet of Things Journal*, pp. 1–1, 2022.

[8] M. Jia, L. Zhang, J. Wu, S. Meng, and Q. Guo, "Collaborative Satellite-Terrestrial Edge Computing Network for Everyone-Centric Customized Services," *IEEE Network*, pp. 1–21, 2022.

[9] S. Yuan, M. Peng, Y. Sun, and X. Liu, "Software defined intelligent satellite-terrestrial integrated networks: Insights and challenges," *Digital Communications and Networks*, 2022.

[10] M. Pattaranantakul, R. He, Z. Zhang, A. Meddahi, and P. Wang, "Leveraging Network Functions Virtualization Orchestrators to Achieve Software-Defined Access Control in the Clouds," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 1, pp. 372–383, 2021.

[11] V. Eramo, E. Miucci, M. Ammar, and F. G. Lavacca, "An Approach for Service Function Chain Routing and Virtual Function Network Instance Migration in Network Function Virtualization Architectures," *IEEE/ACM Transactions on Networking*, vol. 25, no. 4, pp. 2008–2025, 2017.

[12] S. Sahhaf, W. Tavernier, M. Rost, S. Schmid, D. Colle, M. Pickavet, and P. Demeester, "Network service chaining with optimized network function embedding supporting service decompositions," *Computer Networks*, vol. 93, pp. 492–505, 2015, cloud Networking and Communications II.

[13] Y. T. Woldeyohannes, A. Mohammadkhan, K. K. Ramakrishnan, and Y. Jiang, "ClusPR: Balancing Multiple Objectives at Scale for NFV Resource Allocation," *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1307–1321, 2018.

[14] H. Hawilo, M. Jammal, and A. Shami, "Network Function Virtualization-Aware Orchestrator for Service Function Chaining Placement in the Cloud," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 3, pp. 643–655, 2019.

[15] R. Ponmagal, S. Karthick, B. Dhiyanesh, S. Balakrishnan, and K. Venkatachalam, "Optimized virtual network function provisioning technique for mobile edge cloud computing," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 6, pp. 5807–5815, 2021.

[16] N. Moradi, A. Shameli-Sendi, and A. Khajouei, "A Scalable Stateful Approach for Virtual Security Functions Orchestration," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 6, pp. 1383–1394, 2021.

[17] A. Shameli-Sendi, Y. Jarraya, M. Pourzandi, and M. Cheriet, "Efficient Provisioning of Security Service Function Chaining Using Network Security Defense Patterns," *IEEE Transactions on Services Computing*, vol. 12, no. 4, pp. 534–549, 2019.

[18] Y. Liu, Y. Lu, W. Qiao, and X. Chen, "A Dynamic Composition Mechanism of Security Service Chaining Oriented to SDN/NFV-Enabled Networks," *IEEE Access*, vol. 6, pp. 53 918–53 929, 2018.

[19] S. Demirci, S. Sagiroglu, and M. Demirci, "Energy-efficient virtual security function placement in NFV-enabled networks," *Sustainable Computing: Informatics and Systems*, vol. 30, p. 100494, 2021.

[20] F. Wang, D. Jiang, S. Qi, C. Qiao, and H. Song, "Fine-Grained Resource Management for Edge Computing Satellite Networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[21] Y. Yin, C. Huang, D.-F. Wu, S. Huang, M. Ashraf, Q. Guo, and L. Zhang, "Deep Reinforcement Learning-Based Joint Satellite Scheduling and Resource Allocation in Satellite-Terrestrial Integrated Networks," *Wireless Communications and Mobile Computing*, vol. 2022, 2022.

[22] C. Wang, L. Liu, C. Jiang, S. Wang, P. Zhang, and S. Shen, "Incorporating Distributed DRL into Storage Resource Optimization of Space-Air-Ground Integrated Wireless Communication Network," *IEEE Journal of Selected Topics in Signal Processing*, pp. 1–1, 2021.

[23] G. Cui, X. Li, L. Xu, and W. Wang, "Latency and Energy Optimization for MEC Enhanced SAT-IoT Networks," *IEEE Access*, vol. 8, pp. 55 915–55 926, 2020.

[24] Y. Hao, Z. Song, Z. Zheng, Q. Zhang, and Z. Miao, "Joint Communication, Computing, and Caching Resource Allocation in LEO Satellite MEC Networks," *IEEE Access*, vol. 11, pp. 6708–6716, 2023.

[25] S. Pan, S. Cao, L. Yan, and H. Wang, "A Dynamic Predictive VM Resource Scaling Strategy in Satellite-Ground Computing Networks," in *The 5th International Conference on Computer Science and Application Engineering*, ser. Csae 2021. New York, NY, USA: Association for Computing Machinery, 2021.

[26] M. Tong, X. Wang, S. Li, and L. Peng, "Joint Offloading Decision and Resource Allocation in Mobile Edge Computing-Enabled Satellite-Terrestrial Network," *Symmetry*, vol. 14, no. 3, 2022.

[27] Z. Jia, M. Sheng, J. Li, Y. Zhu, W. Bai, and Z. Han, "Virtual Network Functions Orchestration in Software Defined LEO Small Satellite Networks," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[28] Z. Jia, M. Sheng, J. Li, R. Liu, K. Guo, Y. Wang, D. Chen, and R. Ding, "Joint Optimization of VNF Deployment and Routing in Software Defined Satellite Networks," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, 2018, pp. 1–5.

[29] Z. Jia, M. Sheng, J. Li, D. Zhou, and Z. Han, "VNF-Based Service Provision in Software Defined LEO Satellite Networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 9, pp. 6139–6153, 2021.

[30] I. Maity, T. X. Vu, S. Chatzinotas, and M. Minardi, "D-ViNE: Dynamic Virtual Network Embedding in Non-Terrestrial Networks," in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*. Ieee, 2022, pp. 166–171.

[31] X. Gao, R. Liu, and A. Kaushik, "Virtual Network Function Placement in Satellite Edge Computing with a Potential Game Approach," *IEEE Transactions on Network and Service Management*, pp. 1–1, 2022.

[32] G. Wang, S. Zhou, S. Zhang, Z. Niu, and X. Shen, "SFC-based service provisioning for reconfigurable space-air-ground integrated networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 7, pp. 1478–1489, 2020.

[33] H. Yang, W. Liu, X. Wang, and J. Li, "Group Sparse Space Information Network with Joint Virtual Network Function Deployment and Maximum Flow Routing Strategy," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2023.

[34] X. Gao, R. Liu, and A. Kaushik, "Service Chaining Placement Based on Satellite Mission Planning in Ground Station Networks," *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 3049–3063, 2021.

[35] W. Qiao, X. Ni, Y. Lu, X. Li, D. Zhao, and Y. Liu, "A Service Function Chain Deployment Scheme of the Software Defined Satellite Network," *Mobile Information Systems*, vol. 2022, 2022.

[36] X. Gao, R. Liu, A. Kaushik, and H. Zhang, "Dynamic Resource Allocation for Virtual Network Function Placement in Satellite Edge Clouds," *IEEE Transactions on Network Science and Engineering*, pp. 1–1, 2022.

[37] X. Qin, T. Ma, Z. Tang, X. Zhang, H. Zhou, and L. Zhao, "Service-Aware Resource Orchestration in Ultra-Dense LEO Satellite-Terrestrial Integrated 6G: A Service Function Chain Approach," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2023.

[38] A. Petrosino, G. Piro, L. A. Grieco, and G. Boggia, "An Optimal Allocation Framework of Security Virtual Network Functions in 6G Satellite Deployments," in *2022 IEEE 19th Annual Consumer Communications Networking Conference (CCNC)*, 2022, pp. 917–920.

[39] F. Glover, "Tabu search: A tutorial," *Interfaces*, vol. 20, no. 4, pp. 74–94, 1990.

[40] D. Bertsimas and J. Tsitsiklis, "Simulated annealing," *Statistical science*, vol. 8, no. 1, pp. 10–15, 1993.

[41] B. Freisleben and P. Merz, "A genetic local search algorithm for solving symmetric and asymmetric traveling salesman problems," in *Proceedings of IEEE International Conference on Evolutionary Computation*, 1996, pp. 616–621.

[42] Y. Hao, L. Zhang, L. Zheng, Z. Zhao, and Z. Li, "An Efficient VNF Deployment Scheme for Reliable Service Chain in Software Defined Satellite Networks," in *2022 37th Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 2022, pp. 1363–1366.

[43] 3gpp, "Study on architecture aspects for using satellite access in 5G," 3rd Generation Partnership Project, Technical Report (TR) 23.737, 2019.

[44] M. M. Azari, S. Solanki, S. Chatzinotas, O. Kodheli, H. Sallouha, A. Colpaert, J. F. M. Montoya, S. Pollin, A. Haqiqatnejad, A. Mostaani *et al.*, "Evolution of non-terrestrial networks from 5G to 6G: A survey," *IEEE communications surveys & tutorials*, 2022.

[45] A. Guidotti, A. Vanelli-Coralli, V. Schena, N. Chuberre, M. El Jaafari, J. Puttonen, and S. Cioni, "The path to 5G-Advanced and 6G non-terrestrial network systems," in *2022 11th Advanced Satellite Multimedia Systems Conference and the 17th Signal Processing for Space Communications Workshop (ASMS/SPSC)*. IEEE, 2022, pp. 1–8.

[46] T. Villela, C. A. Costa, A. M. Brandão, F. T. Bueno, and R. Leonardi, "Towards the thousandth CubeSat: A statistical overview," *International Journal of Aerospace Engineering*, vol. 2019, 2019.

[47] G. Sciddurlo, A. Petrosino, M. Quadrini, C. Roseti, D. Striccoli, F. Zampognaro, M. Luglio, S. Perticaroli, A. Mosca, F. Lombardi, I. Micheli, A. Ornatelli, V. Schena, A. Di Mezza, A. Mattioni, D. Morbidelli, G. Boggia, and G. Piro, "Looking at NB-IoT over LEO Satellite Systems: Design and Evaluation of a Service-Oriented Solution," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[48] N. Hyodo, T. Sato, R. Shinkuma, and E. Oki, "Virtual Network Function Placement for Service Chaining by Relaxing Visit Order and Non-Loop Constraints," *IEEE Access*, vol. 7, pp. 165 399–165 410, 2019.

[49] S. J. Ling, J. Sanny, and W. Moebs, "University Physics, Volume 1 (OpenStax)," 2016.

[50] 3gpp, "Study on New Radio (NR) to support non-terrestrial networks," 3rd Generation Partnership Project, Technical Report (TR) 38.811, 2017.

[51] ——, "Study on Narrow-Band Internet of Things (NB-IoT) / enhanced Machine Type Communication (eMTC) support for Non-Terrestrial Networks (NTN)," 3rd Generation Partnership Project, Technical Report (TR) 36.763, 2021.

[52] ——, "Study on management and orchestration aspects of integrated satellite components in a 5G network," 3rd Generation Partnership Project, Technical Report (TR) 28.808, 2021.

[53] X. Lin, S. Rommer, S. Euler, E. A. Yavuz, and R. S. Karlsson, "5G from Space: An Overview of 3GPP Non-Terrestrial Networks," *IEEE Communications Standards Magazine*, vol. 5, no. 4, pp. 147–153, 2021.

[54] Z. Lai, H. Li, Q. Wu, Q. Ni, M. Lv, J. Li, J. Wu, J. Liu, and Y. Li, "Futuristic 6G Pervasive On-Demand Services: Integrating Space Edge Computing With Terrestrial Networks," *IEEE Vehicular Technology Magazine*, pp. 2–11, 2022.

[55] I. Ahmad, J. Suomalainen, P. Porambage, A. Gurtov, J. Huusko, and M. Höyhtyä, "Security of Satellite-Terrestrial Communications: Challenges and Potential Solutions," *IEEE Access*, vol. 10, pp. 96 038–96 052, 2022.

[56] S. P. E. Corporation. (2021) Fourth Quarter 2021. [Online]. Available: https://www.spec.org/power_ssj2008/results/res2021q4/

[57] FortiGate. (2021) Virtual Appliances. [Online]. Available: https://www.fortinet.com/content/dam/fortinet/assets/data-sheets/fortigate-vm.pdf

[58] Cisco. (2021) Virtual Appliance. [Online]. Available: https://www.cisco.com/c/en/us/products/collateral/security/adaptive-security-virtual-appliance-asav/adapt-security-virtual-appliance-ds.html

[59] Juniper. (2021) Virtual Appliance. [Online]. Available: https://www.juniper.net/content/dam/www/assets/datasheets/us/en/security/vsrx-virtual-firewall-datasheet.pdf

[60] S. A. R. Shah and B. Issac, "Performance comparison of intrusion detection systems and application of machine learning to Snort system," *Future Generation Computer Systems*, vol. 80, pp. 157–170, 2018.

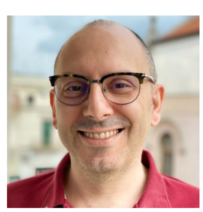
[61] D. Lacković and M. Tomić, "Performance analysis of virtualized VPN endpoints," in *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2017, pp. 466–471.

[62] H. Malik, H. Pervaiz, M. Mahtab Alam, Y. Le Moullec, A. Kuusik, and M. Ali Imran, "Radio Resource Management Scheme in NB-IoT Systems," *IEEE Access*, vol. 6, pp. 15 051–15 064, 2018.

Antonio Petrosino obtained his Bachelor of Engineering in Computer Science and Automation, with a focus on Computer System and Software Engineering in 2016 and his Master degree (with Hons.) in Telecommunication Engineering in 2020. His research interests include Satellite Communication, Software-Defined Networking, Network Function Virtualization, and Internet of Things. Since 2020, he has been a Ph.D student at the Department of Electrical and Information Engineering at Politecnico di Bari.

Giuseppe Piro is an Associate Professor at "Politecnico di Bari", Italy. He received the Ph.D. degree in Electronic Engineering, a first, and a second level degree (both with Hons.) in Telecommunications Engineering from "Politecnico di Bari", Italy, in 2006 and 2008, respectively. His main research interests include wireless networks, network simulation tools, 5G and beyond, network security, nano-scale communications, Internet of Things, and Software-Defined Networking. He serves as Associate Editor for Internet Technology Letter (Wiley), Wireless Communications and Mobile Computing (Hindawi), and Sensors (MDPI).

L. Alfredo Grieco is a full professor in telecommunications at Politecnico di Bari. His research interests include Internet of Things, Future Internet Architectures, and Nano-communications. He serves as Founder Editor in Chief of the Internet Technology Letters journal (Wiley) and as Associate Editor of the IEEE Transactions on Vehicular Technology journal (for which he has been awarded as top editor in 2012, 2017, and 2020).



Gennaro Boggia received the Dr.Eng. and Ph.D. degrees (with Hons.) in electronics engineering from the Politecnico di Bari, Bari, Italy, in July 1997 and March 2001, respectively. He is a Full Professor of Telecommunication with the Politecnico di Bari, Bari, Italy. His research interests include wireless networking, cellular communication, protocol stacks for industrial applications and smart grids, Internet measurements, and network performance evaluation.