



**HAL**  
open science

# A GRASP-based approach for Dynamic and Reliable Virtual Network Embedding

Amine Rguez, Yassine Hadjadj-Aoul, Gerardo Rubino

## ► To cite this version:

Amine Rguez, Yassine Hadjadj-Aoul, Gerardo Rubino. A GRASP-based approach for Dynamic and Reliable Virtual Network Embedding. GIIS 2025 - Global Information Infrastructure and Networking Symposium, Feb 2025, Dubai, United Arab Emirates. pp.1-6, <10.1109/GIIS64151.2025.10922063>. <hal-05267740>

**HAL Id: hal-05267740**

**<https://inria.hal.science/hal-05267740v1>**

Submitted on 18 Sep 2025

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.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

# A GRASP-based approach for Dynamic and Reliable Virtual Network Embedding

Amine Rguez\*, Yassine Hadjadj-Aoul†, and Gerardo Rubino†

\*Univ Rennes, INRIA, France

†Univ Rennes, INRIA, CNRS, IRISA, France

**Abstract**—Ensuring reliable placements of network functions is critical in modern infrastructure, especially in Telco-cloud environments where high availability of services is paramount. In fact, the massive hardware infrastructure required for the emerging 6G and the multiplication of softwarized services to be deployed in the edge-cloud continuum have exacerbated failure concerns, underscoring the importance of addressing reliability challenges.

In this context, we propose a new approach for providing reliable placement of network slices dynamically. This approach leverages the Greedy Adaptive Search Procedure (GRASP), a metaheuristic algorithm that combines greedy randomized construction with local search optimization. We compared the results of our approach against some traditional ones in terms of (1) the acceptance rate when reliability serves as a hard constraint and (2) the placement reliability values when reliability is an optimization objective. Our results demonstrate that our GRASP-based algorithm achieves up to 33% improvement in terms of reliability compared to baseline approaches.

**Index Terms**—Network Reliability, Virtual Network Embedding, GRASP, Network Slicing.

## I. INTRODUCTION

In modern communication networks, the optimization of hardware resource utilization has become increasingly critical, especially with the recent augmentation of connected devices and the increased complexity of infrastructures, where the software plays today a central role, particularly with the recent advancements in virtualization and containerization technologies. In order to address this challenge, initiatives are being taken by both academic and industrial research communities to increasingly enhance infrastructure management through automation and continuous optimisation.

Optimizing infrastructure management is also leading to a growing interest to related topics, like the placement of virtual network functions, which is an essential technological brick for network automation. Ensuring the reliability of the placed network functions has become a major concern, as it is crucial to guarantee that deployed services operate continuously and without failure.

Several approaches in the literature have focused on the problem of minimizing the risk of failure. While existing approaches to virtual network embedding have predominantly focused on static methods or heuristic algorithms with limited considerations for reliability, this paper presents a novel GRASP-based approach designed specifically for dynamic and reliable virtual network embedding. Experimental results

demonstrate that our approach significantly outperforms traditional embedding techniques in terms of both reliability and scalability, making it particularly suitable for real-world applications in 5G and post-5G networks.

In this paper, we present our approach addressing reliability challenges in network slice placement, introducing a methodology consisting of a two-phase optimization process: (1) the initial optimization phase focuses on the optimal placement of Virtual Network Requests (VNRs) or network slices, which aims to maximize resource efficiency by optimizing both system resource (e.g. CPU), and network resources (e.g. bandwidth) utilization.

Following the initial placement, the reliability enhancement phase implements comprehensive optimization strategies for the deployed services, ensuring robust and dependable network operations.

Within the scope of this study, we concentrate specifically on link reliability with the underlying assumption that network links may experience varying degrees of failure, taking into account different failures' severity and their corresponding impact on network performance. The same methodology could be applied to address node failures within the network infrastructure.

The contributions of this paper are summarized as follows:

- We mathematically model the problem of preserving network resources and maximizing the number of accepted NFVs with highest reliability score.
- We introduce a solution based on the GRASP metaheuristic for reliability-aware NFV placement.
- Finally, we investigate the convergence of the GRASP algorithm for NFV placement and evaluate the performance of this method concerning the admission ratio and reliability scores. We also, compare its results against some existing approaches.

The remainder of the paper is organized as follows. Next section discusses the related work. Section III introduces the problem of resource allocation in the cloud and the proposed algorithm. We focus especially on the main functions of our approach. Then, in Section IV, the simulation results of the proposed method are discussed. We compare its performance against those obtained with an enhanced First-Fit heuristic. Finally, our conclusions are presented in Section V.

## II. RELATED WORK

The reliability of the placed network slices has been a major topic in the literature. In fact, many existing research works tried to optimize the quality of the placement by trying to place requests with minimal failure probability.

In [1], a multi-objective network slice embedding model based on reliability constraints was proposed. Although the proposed algorithm is innovative, it lacks a comprehensive comparison with alternative queue-aware algorithms. Similarly, in [2] the authors proposed using a genetic algorithm for a reliable placement of virtual network requests. While the use of a genetic algorithm for service placement is promising, the proposed solution is limited to linear services, which restricts its practical applicability.

Whereas, [3] discussed the reliability requirements of the network slicing process as well as the features and challenges of elastic demand scenarios. It also presented a reliability strategy based on link importance selection, encouraging the use of low saturated links that are part of possible solution. While the paper introduces reliability mechanisms, it would benefit from a comparative analysis with alternative reliability techniques, such as redundancy-based or resource-over provisioning methods. In [4] a workflow driven micro-service architecture in cloud environment is introduced. Also, a reliability optimization algorithm based on partial critic path is proposed. Although, the evaluation could be improved by incorporating more detailed performance metrics, such as mean time to failure (MTTF), service response time, and resource utilization.

Then, in [5] a dataflow-based micro service orchestration language that explicitly support execution equivalence. Although the concept of equivalence-enhanced orchestration is intriguing, the paper could better define what makes this approach unique compared to other reliability-enhancing strategies, such as redundancy or check-pointing. The reference [6] formulated an optimisation problem for dynamic reliability-aware service placement, as a Markov Decision Process to minimize the placement cost and maximize the number of admitted services. In this paper, the evaluation could be more rigorous by incorporating comparisons with baseline reliability approaches, such including a variety of performance metrics, like latency, throughput, and resource usage, would provide a clearer picture of the dynamic strategy's efficiency and trade-offs in performance.

In [7] and [8], the authors addressed the problem reliability-aware provisioning of service function chains with multi-path routing and Redundant Virtual Network Function decomposition. The problem was formulated as an MILP. Then, an hybrid shortest path-based heuristic algorithm is proposed to achieve a near-optimal solution for the first paper and an heuristic to minimize the end-to-end delay and maximize the reliability in the second. Although the approach is promising, the first paper could better highlight the unique aspects of combining function decomposition with multi-path routing for reliability. A clearer comparison to other SFC reliability

strategies, such as path redundancy or fail-over mechanisms, would clarify how this approach innovates within the SFC landscape. Then, in the approach presented in the second paper could further explore the trade-offs between reliability and delay. Since redundancy, often used to improve reliability, can increase delay, a deeper discussion on how these two metrics are balanced in real-world scenarios with varied workloads would be valuable. In [9], an ILP model is proposed to find the placement plan of the VNFs at the underlying physical machines inside the data centers that enhances the availability and the reliability of the NS. While the approach considers both reliability and availability, it would benefit from a more thorough discussion on the potential trade-offs involved, such as increased resource usage or added latency due to redundancy.

In another hand, the use of AI and especially Deep Learning and Deep Reinforcement Learning marked the tendency of research works in recent years. For example, [10] introduced an hybrid deep learning model based on convolution neural network (CNN) and long short term memory (LSTM). The CNN performs resource allocation and slice selection while the LSTM is responsible for statistical information regarding network slices. While the paper is timely, it could have added value by further contextualizing its approach within existing 5G slicing frameworks. A more explicit comparison to alternative approaches in 5G slicing (e.g., rule-based or traditional ML methods) would have highlighted the unique benefits and innovations of their hybrid deep learning approach.

Then, in [11], a Deep Q-Learning network approach for VNF placement is proposed. They model the problem of jointly minimizing the placement cost and maximizing the number of admitted services based on the reliability requirement. The architecture and learning process could be described in greater detail, especially regarding how the state space, action space, and reward structure are defined. A clearer explanation of how these components work together to guide the DRL agent's learning would enhance comprehension of the paper. In [12], a zero-touch (ZT) DRL-based Proactive Failure recovery framework for NFVs is presented. Where a novel ZT proactive failure recovery scheme to maximize the stateful SFC reliability and ensure network service consistency. While the relevance is clear, the paper could improve by situating its contributions within the broader context of existing literature on failure recovery in NFV and zero-touch networks. A more detailed comparison with previous approaches would help to establish the novelty of their model-free deep reinforcement learning (DRL) approach.

## III. BACKGROUND

### A. Reliability Definition

In this section, we provide some definitions of the reliability concept in Network Functions Virtualisation (NFV), according to the ETSI standard [13].

Reliability refers to the ability of virtualized network services and their underlying infrastructure to perform consistently over time, without failure, under predefined conditions.



Fig. 1. Serial composition of VNFs

It is defined as the probability that a placement remains functional after a time period of length  $t$ , typically denoted  $R(t)$ . A reference example is to assume that the time during which the system is functional follows an exponential distribution, whose parameter is here called failure rate. If we denote the latter by  $\lambda$ , we then have  $R(t) = e^{-\lambda t}$ . The probability that a component experiences at least one failure within the time period  $[0, t]$  is given by  $F(t) = 1 - R(t)$ . This is sometimes also called the unreliability at time  $t$ .

### B. Problem Definition

In an NFV environment, as defined above, the objective is to maximize the number of services (i.e. Virtual Network Requests (VNRs)), ensuring that their constraints are fulfilled, while maximizing their reliability and conserving network resources to enable the deployment of subsequent services. Each VNR typically consists of multiple Virtual Network Functions (VNFs).

To effectively estimate and manage the reliability of the overall system, one needs to estimate the reliability of its components. There are two main abstract patterns for this derivation, which will be briefly outlined below.

When combining components, two fundamental types of dependencies can occur: *parallel* and *serial*.

A serial (SER) dependency between two sub-components (for example VNF1 and VNF2) (see Figure 1) implies that both must function simultaneously for the composite system to operate. In other words, both must be operational at the same time for the system as a whole to remain available. Regarding the reliability of this composite system, we have  $R_{SER} = R_{VNF1}(t)R_{VNF2}(t)$ .

In contrast, the parallel (PAR) dependency (see Figure 2) models the situation where either of the two sub-components can be used. This model assumes that the two sub-components are completely redundant, meaning that the composite service can be delivered by either VNF1 or VNF2 with no impact on service quality. Specifically, it assumes that fail-over between the two does not cause any service disruption. With respect to the reliability, since the Virtual Network Request (VNR) fails during a time period  $[0, t]$  if both VNF1 and VNF2 fail during that interval, we have  $F_{PAR} = F_{VNF1}(t) \times F_{VNF2}(t)$ . In the general case with  $N$  sub-components, we have

$$R_{SER}(t) = \prod_{i=1}^N R_{VNF_i}(t), \quad R_{PAR}(t) = 1 - \prod_{i=1}^N (1 - R_{VNF_i}(t)).$$

When extending the concept of reliability to general network topologies, the analysis becomes more complex due to the interdependence of multiple components and paths. A general topology typically includes various paths and structures, such as serial, parallel, or combinations, forming more intricate

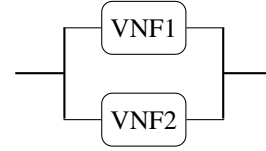


Fig. 2. Parallel composition of VNFs

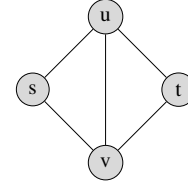


Fig. 3. Bridge network case

arrangements. In such cases, the reliability can be expressed as the probability that at least one valid path between a source and a destination remains functional over a time period  $t$ . A reference example is the *bridge* model depicted in Figure 3.

The exact evaluation of a network's reliability often involves solving a combinatorial problem. In the general case, this is a #P-complete problem, and it remains in this class even after strong specific conditions (e.g., the graph is planar and the nodes' degrees are less than or equal to 3). Advanced techniques, such as Monte Carlo simulations, are then typically employed to calculate the reliability of complex network topologies.

### C. Problem Formulation

We consider a system in which Virtual Network Requests (VNRs) arrive dynamically to be placed in a physical or virtual substrate network.

The Substrate Network, denoted as a graph  $G_S = (N_S, E_S)$ , consists of substrate nodes and substrate edges, represented by the sets  $N_S$  and  $E_S$ , respectively. Each substrate edge  $e_{ij}$  within  $E_S$  is associated with a reliability measure,  $R(e_{ij})$ . This probability value quantifies the likelihood that the edge will continue to function without failure, with values closer to 1 indicating higher reliability.

The Virtual Network Request is represented as a graph  $G_V = (N_V, E_V)$ , where  $N_V$  and  $E_V$  are the sets of virtual nodes and virtual edges, respectively.

To realize these virtual edges within the physical substrate, a mapping function  $P$  is defined from the set of virtual edges  $E_V$  to the set of paths of the substrate graph. Specifically,  $P(e_v)$  for any  $e_v \in E_V$  is a set of substrate edges that form a path within the substrate network. This mapping effectively embeds the virtual network demands into the physical substrate by designating substrate paths for each virtual edge.

This model captures the structure necessary for embedding virtual networks onto a physical substrate, integrating both the connectivity and reliability considerations of the physical network in supporting virtual network demands.

Observe that while it is natural to assume independence of the edges' states (up or down) in the substrate, the virtual links are in general dependent.

1) *Objective*: To calculate the reliability of each virtual link  $e_v \in E_V$  given the reliability value of each of the substrate edges that constitute the path corresponding to the virtual link.

2) *Link Reliability Calculation*: The reliability of a virtual edge  $e_v \in E_V$ , denoted  $R(e_v)$ , is determined by the reliability of the substrate path  $P(e_v)$  that it is mapped to.

3) *Reliability of a Virtual Link*: The reliability of a virtual link  $e_v$  is computed based on the reliability values of the substrate edges that form the path  $P(e_v)$ . Assuming the reliability values of the substrate edges are independent, the reliability of the virtual link is the product of the reliability of the substrate edges in the path:  $R(e_v) = \prod_{e_k \in P(e_v)} R(e_k)$ .

#### D. Placement Strategy

As in [9], here we define the placement strategy for our approach. An optimized placement of services should be aware of the properties of the substrate network and subject to the requirements of the generated services.

1) *Constraints*: We define the following constraints for the substrate network and the generated VNR to be mapped onto it.

- *Node resources constraints* : for the sake of simplicity, we only consider the CPU constraint. This means that the amount of CPU available on each substrate network node should be greater or equal than the required amount of CPU by the VNR.
- *Network resources constraints* : each edge of the network would have an available amount of bandwidth and a reliability value. Of course, each virtual link can occupy the required quantity of bandwidth on all the links of the selected path for its placement.
- *Duration* : each generated VNR would occupy the resources during its randomly generated service time.
- *Collocation constraint* : as we target the VNE problem, the VNFs of a two different VNRs can be collocated together on the same node.
- *Anti-Collocation constraint* : it forces the VNFs of the same VNR to be never collocated together on the same substrate node.

2) *Evaluation Metrics*: Here we detail the different metrics we used to evaluate our approach and also to compare it against some literature methods.

- *Success rate*: it is the number of accepted and placed VNRs divided by the total number of generated requests.

$$\text{Success Rate} = \frac{\text{Number of accepted and placed VNRs}}{\text{Total number of generated requests}}$$

- *Maximum Reliability Value*: if a VNR is placed on the substrate network, we try to maximize the the reliability value of the placement solution.

Let  $L$  represent the set of links involved in the placement of a given VNR. The total reliability  $R_{\text{VNR}}$  of the VNR

would be the product of the reliability values  $r_\ell$  for each link  $\ell \in L$ :  $R_{\text{VNR}} = \prod_{\ell \in L} r_\ell$ .

Here:

- $R_{\text{VNR}}$  is the reliability of the entire VNR;
- $r_\ell$  is the reliability of each link  $\ell$ ;
- $L$  is the set of links that the VNR is using.

In the process of reliability calculation for the VNE, we have to precise that redundant edges used in the placement for a given VNR are not included in the calculation. Specifically, if the same link or edge between two substrate edges is used across different paths in the solution, only one instance of the redundant edge is included in the overall reliability calculation for the placement.

## IV. A GRASP-BASED APPROACH FOR RELIABLE NETWORK SLICES

### A. Algorithm Overview

The Greedy Randomised Adaptive Search Algorithm (GRASP) is a metaheuristic that employs a greedy approach for exploring solutions. In fact, this algorithm firstly generates a solution, then, tries to optimize it locally.

### B. Proposed Algorithm Principles

Our approach is mainly based on two major steps :

- **Solution construction** : this first step aims to create an initial solution that matches the defined constraints. In our case, it is divided into two parts : filtering and scoring. The filtering step involves selecting the nodes with available resources, then finding the shortest path between the selected nodes with available bandwidth. The scoring step consists of selecting the solutions with a cost above a threshold  $\lambda$ . This forms the Restricted Candidate List (RCL) of solutions. Finally, a solution is randomly chosen from this set.
- **Local search** : it is the second step that will try to optimize the firstly constructed solution. It locally searches for a better solution based on some features. In our case, we tried to find a better placement for the VNF in the directly connected nodes with the original placement node. As we try to preserve network resources, we used revenue to the cost of the bandwidth on the substrate network links as an objective function, and also the failure probability or the link reliability constraint. Other strategies could be considered, such as searching even in the indirect vicinity. This solution turns out to be much more costly and not very effective.

## V. PERFORMANCE EVALUATION

### A. Evaluation Setup

We developed a discrete event-based simulator, using the Simpy library, for the placement of VNRs on a Substrate Network. Although we have tested our solution on several network topologies, the results have been fairly consistent. Therefore, in the following, we focus only on the BTEurope topology that contains 24 nodes, with heterogeneous CPU resources, and 49

---

**Algorithm 1** Solution Construction Function

---

**Require:**  $\lambda$  (Filtering Parameter),  $vnr$  (Virtual Network Request),  $N_{indiv}$  (Number of Individuals)  
**Ensure:**  $bestSolution$  (Best Solution),  $bestCost$  (Best Reliability),  $valid$  (Solution Validity)

```
1:  $vnf \leftarrow \text{SELECTRANDOMVNF}(vnr)$ 
2:  $filteredNodes \leftarrow \text{SUBNODESFILTER}(vnf)$ 
3: if  $filteredNodes$  is empty then
4:   return  $False, \emptyset, 0$ 
5: end if
6:  $candidateSol \leftarrow \emptyset$ 
7: while  $\text{LENGTH}(filteredNodes) \neq 0$  and  $N_{indiv} > 0$  do
8:    $valid, partialSol \leftarrow \text{CREATEPARTIALSOLUTION}(vnr, filteredNodes, vnf)$ 
9:   if  $valid$  then
10:     $candidateSol \leftarrow candidateSol \cup partialSol$ 
11:   end if
12:    $N_{indiv} \leftarrow N_{indiv} - 1$ 
13: end while
14: if  $candidateSol$  is empty then
15:   return  $False, \emptyset, 0$ 
16: end if
17:  $solCosts \leftarrow \emptyset$ 
18: for  $sol$  in  $candidateSols$  do
19:    $cost \leftarrow \text{CALCULATERELIABILITY}(sol)$ 
20:    $solCosts \leftarrow solCosts \cup cost$ 
21: end for
22:  $sortedSolutions \leftarrow \text{SORTBYCOST}(candidateSols, solCosts)$ 
23:  $bestCost \leftarrow solCosts[-1]$   $\triangleright$  Best Reliability
24:  $worstCost \leftarrow solCosts[0]$   $\triangleright$  Worst Reliability
25:  $RCL \leftarrow \emptyset$   $\triangleright$  Restricted Candidate List
26:  $f \leftarrow bestCost + \lambda \times (worstCost - bestCost)$ 
27: for  $index$  from 1 to  $\text{LENGTH}(candidateSols)$  do
28:   if  $solCosts[index] \geq f$  then
29:      $RCL \leftarrow RCL \cup candidateSols[index]$ 
30:   end if
31: end for
32:  $bestSolution \leftarrow \text{RANDOMSELECT}(RCL)$ 
   return  $True, bestSolution, bestCost$ 
```

---

edges with limited and heterogeneous available bandwidth and a randomly generated reliability value to ensure more realistic conditions for our simulations.

To simplify the simulations and the calculation of the reliability value for each VNR placement on the SubNet, we did our experiences with linear VNRs. That means that each generated VNR is a linear graph with  $n$  nodes and  $n - 1$  edges.

We consider an environment in which the services' requests (VNRs) arrive dynamically. Each VNR must have sufficient resources to be placed, while respecting its reliability constraints when needed. The VNR remains in the system for a random duration, after which it leaves, releasing the resources it has taken. The objective of the solution is to maximize the

---

**Algorithm 2** Local Search Algorithm

---

**Require:**  $sn$  (Substrate Network),  $vnr$  (Virtual Network Request),  $solution$  (Initial Solution)  
**Ensure:**  $bestSolution$  (Best Solution),  $bestReliability$  (Best Reliability Score)

```
1:  $bestSolution \leftarrow \text{COPY}(solution)$ 
2:  $bestReliability \leftarrow \text{CALCULATERELIABILITY}(sn, bestSolution)$ 
3:  $improved \leftarrow \text{True}$ 
4: while  $improved$  do
5:    $improved \leftarrow \text{False}$ 
6:   for  $neighborSolution$  in  $\text{GENERATENEIGHBORSOLUTIONS}(sn, vnr, bestSolution)$  do
7:      $neighborReliability \leftarrow \text{CALCULATERELIABILITY}(sn, neighborSolution)$ 
8:     if  $neighborReliability > bestReliability$  then
9:        $bestSolution \leftarrow neighborSolution$ 
10:       $bestReliability \leftarrow neighborReliability$ 
11:     end if
12:   end for
13:    $improved \leftarrow \text{True}$ 
14: end while
   return  $bestSolution, bestReliability$ 
```

---

reliability value during placement.

### B. Results

We evaluate the performance of our approach by first considering a model that aims to place the generated services in a manner that maximizes the reliability score. In this case, reliability serves as an objective function that the algorithm learns to optimize. In the second step, reliability is introduced as an additional placement constraint for the generated services.

For all the experiments, we used the following hyperparameters for the GRASP algorithm:

- $\lambda = 0.5$ : This parameter serves as a threshold for selecting candidate solutions.
- $Num\_individuals = 3$ : The number of individuals generated at each iteration.
- $Num\_iterations = 3$ : The total number of iterations performed by the algorithm.

1) *Normal network conditions*: In this case, we configured the resource parameters of the substrate network's nodes and links to accommodate more than two or three VNFs. Additionally, we adjusted the mean time between service arrivals to allow for a higher number of VNRs to be accepted for placement.

In standard network conditions, the results in figure 4 show that the GRASP algorithm clearly outperforms the First-Fit in terms of the obtained reliability score.

2) *Highly congested network*: For this case, we configured the parameters for the substrate network in order to reject more services and the nodes and links would be able to host one or two VNFs and virtual links for the best case scenarios. We

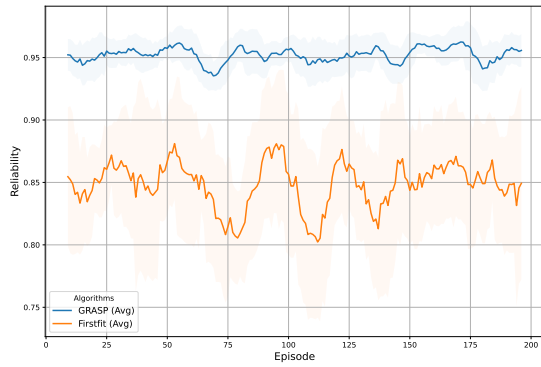


Fig. 4. Reliability values for GRASP vs First-Fit on normal network load conditions

also set the arrival of the services very low in order to reject more placements.

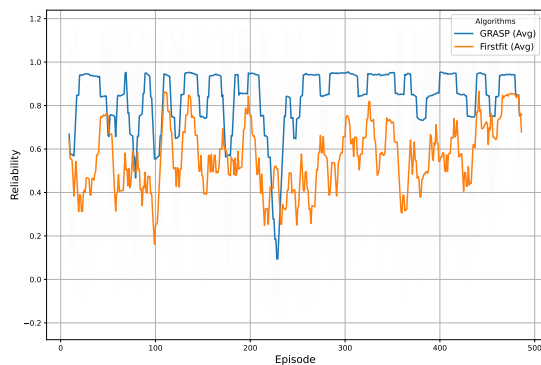


Fig. 5. Reliability values for GRASP vs First-Fit on a highly congested network

For a better assessment of our approach, we did more experiences with a highly congested network. In fact, Figure 5 shows that our algorithms always ensures better reliability scores.

3) *Acceptance Rate comparison:* Here we present the results of the acceptance rate for the placement of our requests in the substrate network. We can clearly see that our algorithm is much better than the First-fit algorithm in terms of accepted requests.

In figure 6, we can see that the acceptance rate of our algorithm is always better than the First-Fit. So, to summarize, our approach, not only gives better reliability scores for the placements, but also accepts more requests than the First-Fit.

## VI. CONCLUSIONS

In this paper, we propose a GRASP-based approach for dynamic and reliable virtual network embedding, addressing the key challenges of service deployment in modern telecom-cloud environments.

We proposed a two-step optimization process, involving an initial placement followed by a reliability enhancement, which allows to dynamically respond to incoming virtual network requests by combining a greedy random structure with

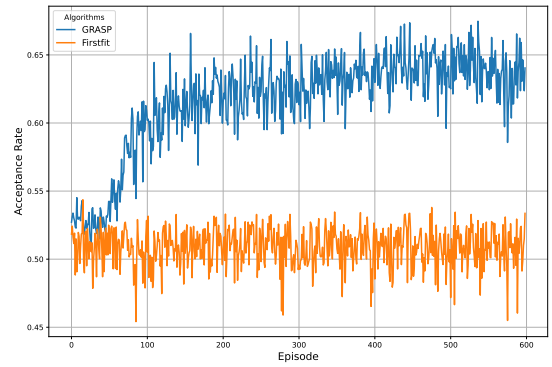


Fig. 6. Acceptance rate comparison for GRASP algorithm vs First-Fit

local search optimization. The algorithm effectively balances resource usage and reliability requirements.

Experimental results show that our approach significantly outperforms traditional methods such as First-fit, with a 33% improvement in reliability and a higher acceptance rate for dynamic service requests.

## REFERENCES

- [1] L. Tang, G. Zhao, C. Wang, P. Zhao, and Q. Chen, "Queue-aware reliable embedding algorithm for 5g network slicing," *Computer Networks*, vol. 146, pp. 138–150, 2018.
- [2] K. Abdellah, S. Ait-Chellouche, Y. Hadjadj-Aoul, and M. Bagaia, "Rapg: Reliability-aware service placement using genetic algorithm for deep edge computing," in *2023 IEEE CCNC*. IEEE, 2023, pp. 255–260.
- [3] R. L. Gomes, L. F. Bittencourt, and E. R. Madeira, "Reliability-aware network slicing in elastic demand scenarios," *IEEE Communications Magazine*, vol. 58, no. 10, pp. 29–34, 2020.
- [4] Z. Liu, H. Yu, G. Fan, and L. Chen, "Reliability modelling and optimization for microservice-based cloud application using multi-agent system," *IET Communications*, vol. 16, no. 10, pp. 1182–1199, 2022.
- [5] Z. Song and E. Tilevich, "Equivalence-enhanced microservice workflow orchestration to efficiently increase reliability," in *2019 IEEE International Conference on Web Services (ICWS)*. IEEE, 2019, pp. 426–433.
- [6] M. Karimzadeh-Farshbafan, V. Shah-Mansouri, and D. Niyato, "A dynamic reliability-aware service placement for network function virtualization (nfv)," *IEEE JSAC*, vol. 38, no. 2, pp. 318–333, 2019.
- [7] L. Qu, C. Assi, M. J. Khabbaz, and Y. Ye, "Reliability-aware service function chaining with function decomposition and multipath routing," *IEEE TNSM*, vol. 17, no. 2, pp. 835–848, 2019.
- [8] L. Qu, C. Assi, K. Shaban, and M. J. Khabbaz, "A reliability-aware network service chain provisioning with delay guarantees in nfv-enabled enterprise datacenter networks," *IEEE Transactions on Network and Service Management*, vol. 14, no. 3, pp. 554–568, 2017.
- [9] Y. Alahmad and A. Agarwal, "Vnf placement strategy for availability and reliability of network services in nfv," in *2019 International Conference on Software Defined Systems (SDS)*. IEEE, 2019, pp. 284–289.
- [10] S. Khan, S. Khan, Y. Ali, M. Khalid, Z. Ullah, and S. Mumtaz, "Highly accurate and reliable wireless network slicing in 5th generation networks: a hybrid deep learning approach," *Journal of Network and Systems Management*, vol. 30, no. 2, p. 29, 2022.
- [11] H. R. Khezri, P. A. Moghadam, M. K. Farshbafan, V. Shah-Mansouri, H. Kebriaei, and D. Niyato, "Deep reinforcement learning for dynamic reliability aware nfv-based service provisioning," in *2019 IEEE GLOBE-COM*. IEEE, 2019, pp. 1–6.
- [12] A. Shaghaghi, A. Zakeri, N. Mokari, M. R. Javan, M. Behdadfar, and E. A. Jorswieck, "Proactive and aoi-aware failure recovery for stateful nfv-enabled zero-touch 6g networks: Model-free drl approach," *IEEE TNSM*, vol. 19, no. 1, pp. 437–451, 2021.
- [13] European Telecommunications Standards Institute (ETSI), "ETSI GS NFV-REL 003 V1.1.1 - Network Functions Virtualisation (NFV); Reliability; Report on Availability and Reliability of Virtualised Network Functions," ETSI, Tech. Rep. GS NFV-REL 003 V1.1.1, 2016.