

An Intelligent Zero-Touch Management and Orchestration in 6G: A Green Hierarchical Reinforcement Learning Approach

Golshan Famitafreshi¹, John Vardakas^{1,2}, Kostas Ramantas¹, Christos Verikoukis³

¹Iquadrat Informática, S.L., Barcelona, Spain

²Dept. of Informatics, UoWM, Kastoria, Greece

¹{g.famitafreshi, jvardakas, kramantas}@iquadrat.com

³Industrial Systems Institute Athena Research Center and University of Patras, Greece

³cveri@isi.gr

Abstract— Fully autonomous, zero-touch systems emphasizing on energy efficiency, high reliability, and ultra-low latency will be possible with the introduction of 6G networks. But with more devices and services, energy usage is expected to rise, necessitating sustainable solutions. A Decision Engine (DE) based on Hierarchical Reinforcement Learning (HRL) is presented in this research to improve the deployment of Service Function Chains (SFCs) based on Cloud-Native Functions (CNF) in dynamic contexts. The goal of the framework is to lower energy consumption while improving scalability and flexibility in the cloud, far-edge, and edge domains. By simulating actual 6G situations, we demonstrate that the HRL-based DE improves resource allocation, reduces latency by 80%, and considerably reduces energy usage by 60% compared to the flat RL. By assisting in self-optimizing network management, our method presents a viable route to intelligent, sustainable 6G networks.

Index Terms—Zero Touch Management, Energy efficiency, Hierarchical Reinforcement Learning, Decision Engine, 6G network

I. INTRODUCTION

The emergence of 6G indicates a significant shift towards fully autonomous, zero-touch networks designed to meet the needs of future applications beyond the service classes like Massive Machine-Type Communication (mMTC), Enhanced Mobile Broadband (eMBB), and Ultra Reliable Low Latency Communication (URLLC). At the same time, network energy consumption is anticipated to rise sharply due to the quick expansion of connected devices and intelligent services. For this reason, 6G has to employ green, energy-conscious designs that align with the UN's Sustainable Development Goals [1] to address these issues by fusing technological automation with sustainability. Intent-Based Networking (IBN) and Reinforcement Learning (RL) make intelligent, self-managed, and ecologically sensitive network operations possible [2].

Given the complexity of diverse settings with dynamic traffic and resource limits, one of the main challenges in 6G is the efficient deployment of Cloud-Native Function (CNF)-based Service Function Chains (SFCs) [3]. Conventional optimization techniques are unable to adjust to changes in real-time, resulting in inefficient use of resources and increased energy usage. Hierarchical Reinforcement Learning (HRL)

delivers a promising technique that enables the adoption of placement strategies that are both scalable and adaptable by decomposing complex decisions into smaller, easier-to-manage tasks. However, other issues also need to be resolved to integrate HRL into 6G-native orchestration frameworks, especially when it comes to making energy-efficient choices without sacrificing performance.

In this regard, we propose a Decision Engine (DE) based on HRL that is intended to maximize the positioning and scalability of CNFs in 6G networks. By adding energy awareness as a primary optimization objective, our framework expands the capabilities of the DE and aims to minimize energy consumption and computing resource utilization while guaranteeing effective service orchestration across heterogeneous domains (edge, far-edge, and cloud). Real-time, completely automated network management with no decision overhead is made possible by the DE's ability to learn multi-level rules that adjust to changing service needs thanks to HRL. The suggested approach, which combines AI-driven decision-making with green networking concepts, is assessed using simulations that mimic actual 6G settings. This strategy offers a viable route towards intelligent, sustainable 6G infrastructures by balancing performance and energy efficiency.

The main contributions of this paper are as follows:

- The creation and use of a DE based on HRL that is specifically suited for energy-efficient orchestration of 6G networks.
- The incorporation of an energy model to facilitate environmentally friendly decision-making.
- A thorough simulation setting that assesses the DE under actual network topologies, workloads, and energy limitations. Furthermore, a performance study demonstrated how well the HRL-based DE reduces energy use without sacrificing service quality.

This study advances the development of 6G networks that are self-optimizing, sustainable, and prepared for the future by bringing intelligent automation and environmental responsibility into balance.

The remainder of this paper is structured as follows. In Section II, we highlight the relevant studies in the literature. Then, the core concepts of this study and the problem statement are explained in Section III. The conducted methodology of this work and simulation setup are described in detail in Sections IV and V, respectively. Section VI is devoted to the performance evaluation of the proposed approach. Finally, in Section VII, we provide final remarks and future work.

II. RELATED WORK

Virtual Network Function (VNF) placement and migration in the concept of 5G and Beyond (5GB) communication have undergone a lot of investigation and research studies [4]. However, none of these methods proposed a multi-scale zero-touch measurement approach, which is capable of reducing the energy consumption of the system. The nature of the proposed methods in the literature is not dynamic enough to meet the requirements of the dynamic and live migration of the CNFs, which is required for the 6G paradigm.

Over the last decade, several sets of techniques were described in the literature that have addressed VNF-based SFC (or even CNF) placement and migration based on mathematical optimization models, one of the earliest studies that utilizes the Integer Linear Programming (ILP) for SFC deployment is defined in [5]. In this work, the authors compare their proposed model with a heuristic algorithm to find the conflict between two algorithms, thus improving the proposed work. Similar to this work by authors in [6]–[8], a mathematical model is defined first to ensure the initial service placement and then service migration across the multiple domains, the second paper describes an optimal placement of the services while meeting the latency requirement. In contrast, the third one defines an ILP model to evaluate the experimental scenario of a VNF placement scenario and then compare it with a heuristic algorithm. However, these works do not provide an optimization method to reduce the energy consumption of domains or the energy consumption of the entire system. Nevertheless, compared to previous works, a few studies proposed energy-efficient methods for the VNF placement through the mathematical model. For instance, in [9], the authors proposed a mathematical model that is able to reduce the bandwidth, energy, and replacement cost. In addition, the goal of the study in [10] was to minimize the energy consumption of the deployed chains by formulating the service deployment problem as an ILP model. The proposed model used a polynomial technique that was both quick and scalable. Nevertheless, the suggested technique was created for situations with static flow.

Another set of techniques used in the literature is based on a wide range of Machine Learning (ML) algorithms from supervised [11], and unsupervised learning [12] to Reinforcement Learning (RL) algorithms. Recently, among other methods, RL techniques have attracted more attention among researchers in the wireless and especially cellular communication networks area since it is able to provide reliable solutions to complex decision-making problems by interacting the agents with the environment. Proposing a high level of dynamicity and making

the technology capable of new features. For instance, the authors in [13] propose a flexible Q-learning algorithm to achieve energy efficiency during the VNF-based SCF migration while meeting the latency requirement simultaneously. The authors demonstrated a considerable enhancement over the greedy heuristic algorithm. Another study proposed in [14] introduced a Multi-agent Deep RL (MADRL) algorithm known as Monitor and Successive Decision Framework (MSDF) designed to reduce network cost in multiple SFC migrations under dynamic traffic, outperforming traditional heuristic algorithms.

As stated before, moving from the 5G to the 6G era will introduce different challenges in terms of network infrastructure and technologies that meet the requirements of futuristic applications beyond the features of the 5G. Rather than extreme URLLC, extreme mMTC, and extreme eMBB, the evaluation of the applications includes AI native communication, which introduces complex network features to the 6G paradigm [2]. It is necessary to introduce flexible and scalable optimization techniques to satisfy the upcoming requirement of the 6G and support the systems with dynamic resource allocation, scaling, service placement, and migration. For this reason, deploying a CNFs-based SFC, capable of achieving optimal performance and low latency in highly distributed and dynamic environments while meeting quality service restrictions, can be beneficial. As mentioned earlier, RL algorithms, specifically HRL methods, are capable of providing the required level of flexibility and reliability in decision-making for complex problems. To the best of our knowledge, no HRL-based DE method has ever been introduced for CNFs-based SFC placement before. Thus, this work benefits from the characteristics that the HRL technique can provide, which is particularly suitable for complex and dynamic environments. The proposed HRL method dynamically deploys the services in the most proper technological domain to reduce the usage of computational resources along with the energy consumption of each domain.

III. PROBLEM STATEMENT

This section outlines the current issue in the 6G environment that the paper addresses while providing a brief overview of the core ideas that motivate the research and are crucial to comprehending the approach used in this work.

One major challenge in 6G is the efficient placement of CNF-based SFCs. The complexity of the heterogeneous 6G environment and the requirement of futuristic applications, which span multiple technological domains with contrasting resource constraints, make this problem far more complicated than in previous network generations. Traditional optimization techniques often struggle to adapt in real-time or require too much computational power, making them impractical for fully automated networks.

The research aims to develop an HRL-based zero-touch management framework for CNF-based SFC placement in 6G. HRL simplifies learning into simpler sub-tasks and improves policy optimization, making it suitable for zero-touch CNF-based SFC placement. However, incorporating HRL into 6G-

native orchestration frameworks requires real-time learning strategies, adaptability across domains, and energy-efficient decision-making. The framework aims to reduce energy consumption and computational resources while ensuring efficient resource allocation across different domains, paving the way for fully automated 6G network management.

Several fundamental ideas in RL form the basis of this approach. The core of the modeling lies in the *Markov Decision Process (MDP)*, which demonstrates the interaction between defined technological domains and updates the sequential decision-making in which two levels of agents interact with the environment. This framework models the service placement problem, which can be simplified as the following tuple:

$$(\Omega, \Psi, \Pi_\alpha, \Lambda_\alpha) \quad (1)$$

where Ω is the representative of the group of states, which includes the variables that define the environment or observation (in this paper, it corresponds to a set of computational resources -CPU, RAM, and storage- aligned with network metrics such as delay, available energy, and Bandwidth). Ω is updated at each step of the execution of the algorithms. Ψ is the group of actions the agents execute (low-level agent executes inside a domain, and high-level agents execute between domains), which is responsible for the dynamic placement of the services. $\Pi_\alpha(\Omega_{t+1}|\Omega_t, \alpha)$ is the probability transition function depicting the probability of action α (which belongs to Ψ) being taken in the state t to reach state $t+1$ in the environment. Finally, Λ_α is the reward function, calculated after the execution of action α and updated at each step of the process to provide feedback for the decision-making in the next state. It is important to take into consideration that in the proposed RL-based algorithm, since two levels of agents are defined, two reward functions and two levels of actions are required.

In general, *Q-learning* is a model-free (does not require prior knowledge of the environment), off-policy RL algorithm used to train an agent to make optimal decisions in an environment. It is based on the concept of learning the value of actions in different states to maximize cumulative rewards over time. Q-learning serves by learning a function known as the Q-value function ($Q(s, a)$), which estimates the expected cumulative reward when taking action a in a state s , followed by the Bellman equation update rule.

Stochastic policies specified in the maximum entropy framework are optimized using the off-policy actor-critic DRL method known as *Soft Actor-Critic (SAC)*. The algorithm's foundation is a policy iteration formulation that exchanges between phases for policy improvement and policy evaluation. In the former, the policy is updated towards the exponential of the updated Q-function, whereas in the latter, a parameterized soft Q-function is updated to match the value of the parameterized policy in accordance with the maximum entropy aim [15].

Hierarchical Intentional-Unintentional SAC (HIU-SAC) is a hierarchical RL method for tackling a complex task that can be broken down into simpler subtasks to facilitate the

learning and exploration phase. It utilizes composable policies, represented by Gaussian distributions, for each subtask, and a compound policy, a higher-level policy, which combines and orchestrates these policies to execute the overall objective. By leveraging activation vectors, it manages the combination of tasks and facilitates the fulfillment of the complicated task through the defined sub-tasks. The algorithm learns these policies concurrently using an off-policy approach, improving learning efficiency and making it well-suited for challenging decision-making problems [16].

Furthermore, according to the high-level architecture of the proposed DE, which is based on the ETSI GS ZSM [17] four main phases of a closed-loop function are outlined: monitoring, analysis, decision-making, and execution. The initial phase is collecting raw data, which is then analyzed to provide knowledge and guide choices. To get the intended outcomes, strategies are developed and implemented throughout the decision-making stage. These stages guarantee a thorough analysis of both past and present facts, improving decision models and directing execution tactics. In order to facilitate adaptive model modifications and decision-making for external entities or other closed loops, the decision-making stage is made to interact with external components and other closed-loop systems. The paper [18] dives into the depth of this architectural framework.

Based on the aforementioned information to provide fine-grained control at both the strategic and operational levels, this paper is the first to characterize 6G service orchestration as a hierarchical, time-dependent, multi-domain RL problem.

IV. METHODOLOGY

This section applies an HRL-based optimization algorithm that is derived from MDP to meet the proposed objective of this paper.

The *Intelligent Service Placement Algorithm* evaluates the available network and computing resources to optimize the placement of service requests (S_j) across several technical domains (D_i). Initially, the algorithm creates service requests with particular resource requirements and sets the maximum resource capabilities for each domain (C_{s_j} and N_{s_j}). Service requests are prioritized based on predefined precedence levels (URLLC, mMTC, and eMBB) to mimic real-world efficient service placement. Then, during each iteration, the algorithm evaluates the resources of the available domains and whether each domain has sufficient resources to accommodate the incoming requests. If a domain satisfies the service requirements, a request is made, and the existing resources (Available C_{d_i} , Available N_{d_i}) are updated appropriately. If not, the algorithm makes an effort to assign the service to the next domain (D_{i+1}), guaranteeing the best possible use of the resources at hand while reducing service rejection. The process continues until all requests are either placed successfully or marked as rejected due to resource constraints.

Beyond optimizing placement efficiency, the algorithm also contributes to energy efficiency and green networking by minimizing unnecessary resource usage and optimizing workload

distribution. It ensures that resources are allocated in a way that reduces energy consumption by avoiding overloading specific domains and leveraging underutilized ones. By dynamically adjusting placement decisions based on available domains' resources, the algorithm helps lower power consumption in the defined technological domains and provision a balanced trade-off between performance, power efficiency, and sustainability. Furthermore, evaluating the suggested algorithm will simultaneously present the efficiency of the proposed algorithm in terms of energy and computational resources.

Algorithm 1 Intelligent Service Placement Algorithm.

```

1: Initialization: Maximum  $C_{d_i}$  and  $N_{d_i}$  and hyperparameters
2: Service generation  $S_j$ 
3: Input:  $C_{s_j}$  and  $N_{s_j}$ 
4: Output: Available  $C_{d_i}$ , Available  $N_{d_i}$  and  $S_{rej}$ 
5: for each iteration do
6:   for  $S_j$  do
7:     Prioritize the service request based on their level of
       priority
8:   end for
9:   for  $D_i$  do
10:    if  $C_{s_j} \leq C_{d_i}$  and  $N_{s_j} \leq N_{d_i}$  then
11:      Place  $S_j$  on  $D_i$  and update  $C_{d_i} - C_{s_j} = \text{Available}$ 
         $C_{d_i}$ ,  $N_{d_i} - N_{s_j} = \text{Available } N_{d_i}$ 
12:    else
13:      Place  $S_j$  on  $D_{i+1}$  and update  $C_{d_{i+1}} - C_{s_j} = \text{Available}$ 
         $C_{d_{i+1}}$ ,  $N_{d_{i+1}} - N_{s_j} = \text{Available } N_{d_{i+1}}$ 
14:    end if
15:  end for
16: end for
17: Return  $S_{rej}$  and Available  $C_{d_i}$ , Available  $N_{d_i}$ 
18: end procedure

```

In Algorithm 1, each service is characterized based on the following required factors:

$$S_j = (B_j, L_j, E_j, T_j, R_j, \Pi_j) \quad (2)$$

where the required bandwidth, service duration, latency, energy consumption, computational resources, and priority level are noted as $B_j, T_j, L_j, E_j, R_j, \Pi_j$, respectively. In addition, each domain provides a dedicated set of computational resources and network resources, which are demonstrated as follows.

$$\begin{aligned} C_i &= (Memory_i, CPU_i, RAM_i) \\ N_i &= (B_i, L_i, E_i) \end{aligned} \quad (3)$$

in which the required bandwidth (B_i), latency (L_i), and energy consumption (E_i) are defined as N_i network resources, and computational resources include memory, CPU, and RAM.

V. SIMULATION SETUP

In this section, the important evaluation metrics' models will be explained. Then, the system model and the experimental environment will be described in detail. The following simulation step-up allows us to deploy the proposed HRL algorithm under the defined conditions. Section VI will evaluate and discuss the assessment of the algorithm's performance.

A. Energy Model

Various components contribute to the overall energy consumption of 5GB networks. However, accurately calculating total energy consumption is complex due to the interplay of static and dynamic factors, as well as fluctuations in network traffic patterns. This section briefly summarizes total energy consumption in an idealized scenario. In this context, the total energy consumption of a 5GB network is the sum of its static and dynamic components, expressed as follows.

$$E_T = \sum_{i=1}^n E_{static,i} + E_{dynamic,i}, \quad (4)$$

where

$$\begin{aligned} E_{static} &= T \times (P_{BS} + O), \\ E_{dynamic} &= T \times (U_{CPU} \times P_{CPU} + U_{BW} \times P_{BW} \\ &\quad + L_{CPU} \times P_x), \end{aligned}$$

In this case, n domains are considered, in which P_{BS} , P_{CPU} , P_{tx} , P_{BW} denote the power consumption of the base station, CPU, transmission state, and each unit of bandwidth, respectively. U_{CPU} , L_{CPU} , and U_{BW} represent the CPU utilization, CPU load, and bandwidth utilization, which depend on the actual traffic load within the domain. Lastly, O accounts for the power consumption associated with the network infrastructure, while T representing the total operational time.

B. Delay Model

As mentioned in the Introduction, service placement latency is a critical factor in 5G and 6G communications, distinguishing it from traditional packet transmission delay is important. In this case, the total service latency is defined as the sum of different factors, including processing capabilities, resource availability, and migration overhead, and can be conveyed as follows.

$$L_{service,T} = L_{processing} + L_{I/O} + L_{queue} + L_{migration}, \quad (5)$$

where $L_{processing}$ is noted as the processing delay, which is the time taken to allocate computing resources to the service placement. $L_{I/O}$ denotes the time required to transfer the service-related data (code, dependencies, configurations) between different hardware components, storage devices, or network layers. L_{queue} is the time that each service is waiting before execution due to scheduling (service prioritization) or resource constraints. $L_{migration}$ is the duration that a running service in one domain is transferred to another domain to optimize the response time while preventing service rejection. It is worth mentioning that in this scenario, the migration delay has not been taken into account since the focus of the paper is on the placement of the SFC.

C. Traffic Pattern

In this study, three different traffic distribution models are defined to model the most proper traffic pattern for each service type: square, Poisson, and exponential, which map to URLLC, mMTC, and eMBB services.

In general, the square wave pattern imitates the constant bit rate (CBR) and on-off transmission. The Poisson distribution matches mMTC well, as it represents random event arrivals. Finally, the exponential pattern models the sudden bursts in data transmission, such as adaptive video streaming or large file downloads, where traffic grows rapidly and stabilizes.

D. System Model

We consider a 5GB network comprising multiple technological domains (e.g., edge and cloud), which are geographically distributed across different locations. Each domain contains a random number of interconnected servers, forming a local infrastructure. User terminals connect to these domain servers to request network services, which are represented in the form of SFCs.

Fig. 1 illustrates a domain-wise network graph. Within each domain (intra-domain), low-level (local) agents assess whether the requested computational and communication resources are available. If the required resources are sufficient, the service is successfully placed in the selected domain (depicted by the green flow). However, if the resources are insufficient, the service request fails (red flow), and a high-level (global) agent oversees the selection of an alternative domain for placement. It is important to note that service requests are prioritized based on their urgency before allocation. In addition, for the matter of simplification in this set of simulations, three domains are taken into consideration.

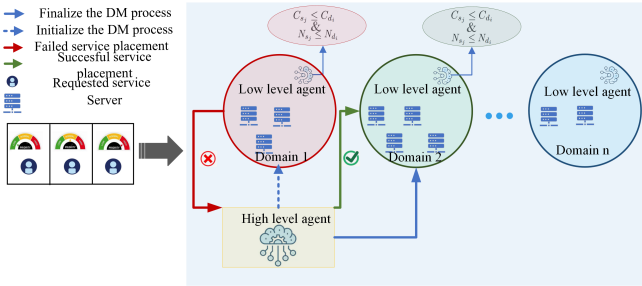


Fig. 1: Architecture of the HRL-based DE.

The simulation of the described HRL-based DE was conducted using Python language and a custom OpenAI Gym environment, and HRL agents were built using PyTorch. The environment defines the following components to map the defined optimization problem to the MDP.

- **State:** According to the HRL algorithm, the state space includes the computational and communicational metrics of the domain (see. Eq. 3), which is updated in each iteration of the algorithm procedure.
- **Action:** Since two levels of agents are making decisions, two levels of actions are defined: local Action and Global Action. The local Action is a binary value that demonstrates whether the comparison conducted addresses the requirement or not. Then, the global Action, which the global agent performs, is to select the most proper domain for the required service process.

- **Reward:** Similar to the Action, two levels of rewards are conducted at two levels of the algorithm; however, the optimization function of the HRL algorithm is defined in the high level of the algorithm as follows.

$$\text{Reward} = \sum_{i=0}^n \alpha \cdot \frac{1 - E_T}{\omega_1 \cdot L_{\text{service},T}}, \quad (6)$$

where n is the total number of the domains, E_T is the total energy consumption of the network, ω_1 is the weight of $L_{\text{service},T}$ in the reward function, and α determines how much priority is given to maximizing the available energy and minimizing latency. A detailed explanation of the functionality of the implemented HRL algorithm can be found in [18].

VI. PERFORMANCE EVALUATION

In this section, the impact of the proposed HRL-based DE on the performance of the defined network is assessed through simulations. The baselines to evaluate the proposed algorithm are Q-learning-based single-agent [19] and the SAC-based algorithm [20], which are adapted to the defined network environment.

Fig. 2a highlights the amount of CPU usage percentage under the deployment of the three defined algorithms for each domain. As it is demonstrated, Q-learning and SAC show higher CPU usage, while HIU-SAC demonstrates slightly lower usage, which suggests better computational efficiency (around 60% and 20% compared to the Q-learning and SAC algorithms, respectively). The Q-learning spikes higher, possibly due to its less adaptive nature in handling complex environments.

HIU-SAC achieves the lowest energy consumption by using a hierarchical decision-making structure (190 mJ), while Q-learning consumes the most due to an inefficient exploration phase (445 mJ). By showing 390 mJ energy consumption, SAC falls in between, balancing learning efficiency with computational cost better than Q-learning (see Fig. 2b).

Q-learning's high latency (average of 21 ms) is linked to its increased energy consumption and CPU usage, which is caused by its inefficient decision-making process, and therefore, it reveals less throughput. In contrast, HIU-SAC achieves lower latency (4 ms) and energy use through a more streamlined, hierarchical approach. These results show an 80% reduction in latency and around a 65% increase in throughput (Fig. 2d and Fig. 2c).

Service rejection is a critical performance metric in 5GB networks, representing the proportion of service requests that are denied due to resource limitations, policy restrictions, or network congestion. Fig. 3 demonstrates the effectiveness of the HIU-SAC algorithm in managing network resources, highlighting a significant reduction in service rejection (58%) compared to traditional approaches such as Q-learning and SAC. This highlights the benefits of employing a more advanced and adaptive resource management strategy.

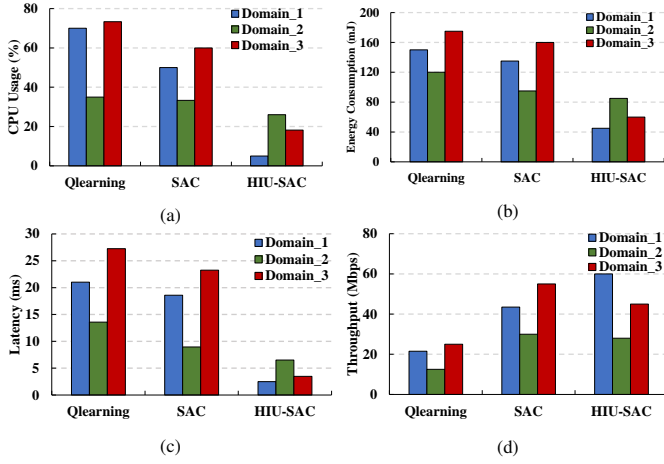


Fig. 2: (a) The average CPU usage for different algorithms per domain. (b) The total energy consumption for different algorithms per domain. (c) The average latency for different algorithms per domain. (d) The average throughput for different algorithms per domain.

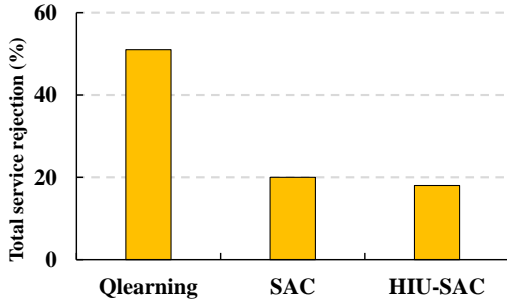


Fig. 3: Total service rejection for different algorithms.

VII. CONCLUSION AND FUTURE WORK

In this work, we assessed the effectiveness of a DE based on HRL for automated and zero-touch SFC placement, with an emphasis on increasing energy efficiency and lowering operating overhead. The results of our approach demonstrated a considerable reduction in energy consumption (60%) and latency (80%) while improving the resource allocation optimization compared to the flat RL. However, it is important to mention that the proposed framework did not take into account the fairness of service placement across various domains. By including fairness criteria in the decision-making process, future research will attempt to overcome this constraint and guarantee equal resource allocation across several domains. Our study will also expand to create an Intent-Based Network (IBN) solution that dynamically adjusts to shifting network conditions. A strong solution for live migration services in the 6G context will be introduced as part of this, facilitating smooth transitions and enhancing service continuity.

ACKNOWLEDGMENT

This work has received funding from the European Union under the ADROIT6G project (Grant agreement ID: 101095363).

REFERENCES

- [1] J. R. Bhat, and S. A. Alqahtani. 6G ecosystem: Current status and future perspective. *IEEE Access*, 9:43134–43167, 2021.
- [2] D. Brodimas, and K. Trantzas, and B. Agko, and G. Ch. Tziavas, and Ch. Tranoris, and S. Denazis, and A. Birbas. Towards intent-based network management for the 6g system adopting multimodal generative ai. In *Proc. EuCNC/6G Summit 2024*.
- [3] S. B. Chetty, and A. Nag, and A. Al-Tahmeesschi, and Q. Wang, and B. Canberk, and J. Marquez-Barja, and H. Ahmadi. Optimized Resource Allocation for Cloud-Native 6G Networks: Zero-Touch ML Models in Microservices-based VNF Deployments. *IEEE Network*, 2024.
- [4] W. Attaoui, and E. Sabir, and H. Elbiaze, and M. Guizani. Vnf and cnf placement in 5g: Recent advances and future trends. *IEEE Transactions on Network and Service Management*, 20(4):4698–4733, 2023.
- [5] D. Zhao, and J. Ren, and R. Lin, and Sh. Xu, and V. Chang. On orchestrating service function chains in 5G mobile network. *IEEE Access*, 7:39402–39416, 2019.
- [6] B. E. Mada, and M. Bagaa, and T. Tale, and H. Flinck. Latency-aware service placement and live migrations in 5G and beyond mobile systems. In *Proc. ICC 2024*.
- [7] P. K. Thiruvassagam, and A. Chakraborty, and A. Mathew, and C. S. R. Murthy. Reliable placement of service function chains and virtual monitoring functions with minimal cost in softwareized 5G networks. *IEEE Transactions on Network and Service Management*, 18(2):1491–1507, 2021.
- [8] Sarrigiannis, Ioannis and Antonopoulos, Angelos and Ramantas, Kostas and Efthymiopoulou, Maria and Contreras, Luis M and Verikoukis, Christos. Cost-aware placement and enhanced lifecycle management of service function chains in a multidomain 5G architecture. *IEEE Transactions on Network and Service Management*, 19(4):5006–5020, 2022.
- [9] M. A. Abdelaal, and G. A. Ebrahim, and W. R. Anis. Efficient placement of service function chains in cloud computing environments. *Electronics*, 10(3):323, 2021.
- [10] B. Farkiani, and B. Bakhshi, and S. A. Mirhassani. A fast near-optimal approach for energy-aware SFC deployment. *IEEE Transactions on Network and Service Management*, 16(4):1360–1373, 2019.
- [11] L. Le, and B. P. Lin, and L. Tung, and D. Sinh. SDN/NFV, Machine Learning, and Big Data Driven Network Slicing for 5G. In *Proc. IEEE 5GWF 2018*.
- [12] J. Chen, and J. Chen, and R. Hu, and H. Zhang. ClusVNFI: A hierarchical clustering-based approach for solving VNFI dilemma in NFV orchestration. *IEEE Access*, 7:173257–173272, 2019.
- [13] J. Yao, and M. Chen. A flexible deployment scheme for virtual network function based on reinforcement learning. In *Proc. IEEE ICC 2020*, pages 1505–1510. IEEE, 2020.
- [14] R. Chen, and H. Lu, and Y. Lu, and J. Liu. MSDF: A deep reinforcement learning framework for service function chain migration. In *Proc. IEEE WCNC 2020*.
- [15] T. Haarnoja, and A. Zhou, and P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr, 2018.
- [16] D. Esteban, and L. Roza, and D. G. Caldwell. Hierarchical reinforcement learning for concurrent discovery of compound and composable policies. In *Proc. IEEE/RSJ IROS 2019*.
- [17] ETSI, G. Zero-touch network and service management (ZSM); closed loop automation; Part 1: enablers [J]. *Group Specification (GS) ETSI GS ZSM*, 2021.
- [18] G. Famitafreshi, and M. Trigka, D. Selis, and J. Vardakas, and Ch. Verikoukis. An Innovative Multi-Scale Strategy-Based Decision Engine for Zero-Touch Management and Orchestration in 6G. In *Proc. IEEE CAMAD 2024*.
- [19] Z. Zhang, and L. Ma, and K. K. Leung, and L. Tassiulas, and J. Tucker. Q-placement: Reinforcement-learning-based service placement in software-defined networks. In *Proc. IEEE ICDCS 2018*.
- [20] M. Bansal, and I. Chana, and S. Clarke. UrbanEnQoSPlace: A deep reinforcement learning model for service placement of real-time smart city IoT applications. *IEEE Transactions on Services Computing*, 16(4):3043–3060, 2022.