Towards cost estimation of DRL for task scheduling and placement

Julio Corona[1][0009−0001−7898−7883], Mário Antunes[1,2][0000−0002−6504−9441], José Quevedo[1][0000−0002−8112−6726], and Rui L. Aguiar[1,2][0000−0003−0107−6253]

[1] Instituto de Telecomunicações, University of Aveiro, Aveiro, Portugal
`[jcamejo, mario.antunes, quevedo, ruilaa]@av.it.pt`
[2] DETI, University of Aveiro, Aveiro, Portugal
`[mario.antunes, ruilaa]@ua.pt`

Abstract. This paper explores the effectiveness of Deep Reinforcement Learning (DRL) in 5G scheduling and placements tasks. The study implements multiple DRL models with established architectures and compares their energy consumption and performance during training and execution against traditional models. The results show an improvement in the task response times while using DRL models. However, it also shows that DRL's decision-making process leads to higher energy consumption.

Keywords: DRL · costs · scheduling · placement · energy consumption.

1 Introduction

With an estimated 1.2 million active Internet of Everything (IoE) devices expected by 2030 [3], precise task scheduling and placement are crucial for optimizing network infrastructure utilization and ensuring prompt responses [9]. However, the efficient allocation of computing resources considering the heterogeneity of processing hardware and network channels remains a challenge [19,11].

Dynamic solutions based on Machine Learning (ML) techniques, particularly Deep Reinforcement Learning (DRL), are proposed for task scheduling and placement in the field of Artificial Intelligence (AI) [14,5]. However, previous DRL research for this problem domain has largely overlooked the computational cost of training models and executing actions.

This work aims to address this research gap by quantifying the energy cost of DRL in the domain. The key contributions are as follows:

- A realistic environment is provided, simulating real-world conditions and capturing the inherent challenges and dynamics of the problem. This enhances the reliability and relevance of the findings.
- Different strategies from previous works are summarized and adapted to the proposed scenario based on model objectives and specifications.
- A comprehensive performance evaluation of DRL algorithms for task scheduling and placement is conducted, considering various approaches in terms of performance and energy consumption.

The paper is structured as follows: In section 2, related work, strategies, and reviewed DRL algorithms are discussed. section 3 outlines the network architecture and problem formulation for each strategy. The performance evaluation of the algorithms is presented in section 4, and conclusions are drawn in section 5.

2 Related work

This section presents various scheduling and placement techniques developed for fog cloud environments.

In [8], an intelligent resource allocation and task scheduling algorithm based on Proximal Policy Optimization (PPO) is presented, considering task priorities based on deadlines and vehicle service availability factors to minimize waiting time and delays in dynamic vehicular fog networks. [16] proposes a Deep Q-Learning (DQL)-based approach for dynamic task scheduling and resource management in Software Defined Networking (SDN) networks, aiming to minimize network latency and ensure energy efficiency. In [7], three similar solutions (Deep Q-Network (DQN), Deep Deterministic Policy Gradient (DDPG), and Soft Actor-Crítico (SAC)) are introduced, focusing on Quality of Service (QoS) factors such as end-to-end latency, power consumption, and meeting deadlines. A scalable approach to parallel task scheduling is presented in [20], using a customized variant of Advantage Actor-Critic (A3C) to assign tasks to distributed nodes and reduce execution times. In [6], a DRL framework combines a Convolutional Neural Network (CNN) to solve job shop scheduling problems, with manufacturing states represented as multi-channel images and fed into the network; a Double Deep Q-Network (DDQN)-based agent selects jobs to dispatch, minimizing execution times. Researchers in [15] propose a novel approach using DDPG to handle load balancing in cloud computing, particularly in traditional application migration scenarios, dynamically allocating tasks to virtual machines to optimize service-level agreement (SLA) compliance and minimize average task response time. The paper [10] introduces a task scheduling framework based on DQN that adapts to dynamic workload fluctuations in complex cloud data centers, ensuring lower energy consumption and faster response times. In [2], a DQN-based approach for real-time job scheduling in hybrid cloud environments is presented, optimizing monetary costs associated with job execution while ensuring QoS and minimizing response time.

3 System architecture and adapted strategies

We developed a fog cloud computing environment by extending FogWorkflowSim [13]. The fog layer consists of N processing nodes and a gateway device for task scheduling and placement. To ensure the existence of at least one node with sufficient resources for any task, a small cloud is integrated. The gateway device dynamically selects tasks and processing nodes based on priority levels, MI (million of instructions), RAM and storage requirements of each task.

3.1 Task scheduling and placement strategies

Based on the objectives and specifications of the models observed in the state of the art, we implemented the strategies: Placement (TP), Scheduling and Placement (TSP), and Batch Placement (TBP) in the gateway. To cover all possibilities, we also implemented the Scheduling (TS) strategy. The objective in all of them is to minimize tasks' response time. Table 1 summarizes the algorithm used for each decision by strategy.

Table 1: Explored strategies

Strategy	**Scheduling**	**Placement**
Scheduling	PPO	First server with enough resources
Placement	FIFO	DQN
Scheduling and Placement	DQN	
Batch Placement	DDPG	

4 Results evaluation

We used a public available dataset [1] This dataset comprises approximately 1.3 million tasks, encompassing various requirements such as the requested CPU, RAM, and storage resources for each task, along with estimated start time and estimated finish time.

Since FogWorkflowSim, like many others, operates with a virtual clock, to simulate the energy consumption of the gateway, all scheduling/placement models were implemented in a separate service hosted on a real computer. The MIPS were estimated using the 7z application, similarly to [4]. This allowed us to simulate the decision as a single task on a dedicated device within the simulator, referred to as the Gateway.

In this simulation, the parameters were carefully chosen to closely resemble real-world scenarios, drawing inspiration from values reported in other works such as [12,17,18]. The configuration of each fog node was randomly selected. The experiments were executed with 20 and 40 nodes.

The average response time of the tasks was generally consistent across all cases. Figure 1 shows that the TSP and TP with PPO achieved slightly higher results compared to others when using 40 and 20 fog nodes, respectively, with the first dataset. The results from the second dataset closely resembled those from the first dataset, but the TS strategy with PPO achieved the lowest response times with 40 nodes, closely followed by other strategies. However, with 20 nodes, the best result was obtained by the TS strategy with Round Robin. Notably, the TP strategy performed poorly, yielding response times approximately 35% higher with both datasets and 20 fog nodes.

The energy consumption was almost the same with both datasets. As is shown in Figure 2, all baseline algorithms showed negligible energy consumption, while DRL showed much higher energy consumption.

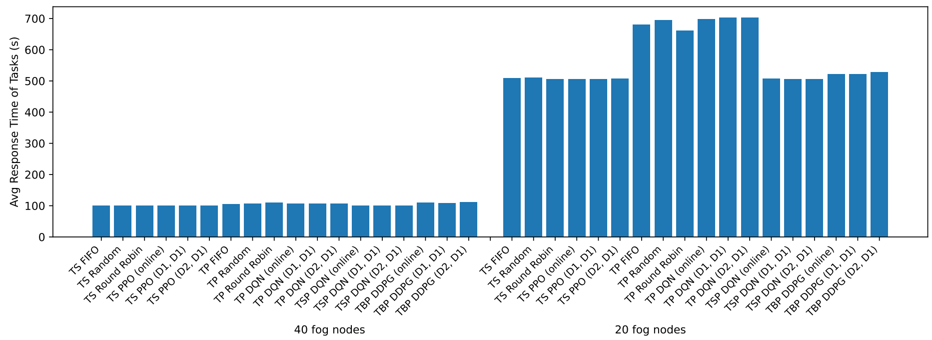

Fig. 1: Average Response Time of Tasks with the first dataset

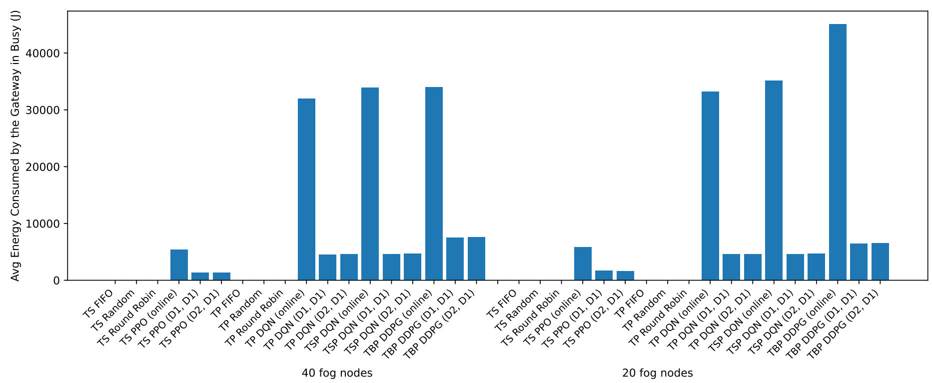

Fig. 2: Average Energy Consumed by the Gateway in Busy with the first dataset

5 Conclusions

We adapted several programming and placement strategies found in the literature to a scenario as close to reality as possible. The results showed very few variations in the performance of the strategies. TSP and TP with DQN achieved slightly improved performance, although they were occasionally surpassed by the TS strategy using Random and Round Robin. Regarding energy consumption, the baseline algorithms showed negligible values. In contrast, the DRL agents exhibited significantly higher energy consumption.

Acknowledgements. This work is supported by the European Union / Next Generation EU, through Programa de Recuperação e Resiliência (PRR) [Project Nr. 29: Route 25];

References

1. Ali Rezaee, S.A.: Jobs (DAG workflow) and tasks dataset with near 50k job instances and 1.3 Millions of tasks. (Sep 2020). https://doi.org/10.5281/zenodo.4667690
2. Cheng, L., Kalapgar, A., Jain, A., Wang, Y., Qin, Y., Li, Y., Liu, C.: Cost-aware real-time job scheduling for hybrid cloud using deep reinforcement learning. Neural Computing and Applications **34**(21), 18579–18593 (2022)
3. Chowdhury, A., Karmakar, G., Kamruzzaman, J.: The co-evolution of cloud and iot applications: Recent and future trends. In: Handbook of Research on the IoT, Cloud Computing, and Wireless Network Optimization, pp. 213–234. IGI Global (2019)
4. Domingues, P., Araujo, F., Silva, L.: Evaluating the performance and intrusiveness of virtual machines for desktop grid computing. In: 2009 IEEE International Symposium on Parallel & Distributed Processing. pp. 1–8. IEEE (2009)
5. Gondhi, N.K., Gupta, A.: Survey on machine learning based scheduling in cloud computing. In: Proceedings of the 2017 International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence. pp. 57–61 (2017)
6. Han, B.A., Yang, J.J.: Research on adaptive job shop scheduling problems based on dueling double dqn. Ieee Access **8**, 186474–186495 (2020)
7. Jain, V., Kumar, B.: Qos-aware task offloading in fog environment using multi-agent deep reinforcement learning. Journal of Network and Systems Management **31**(1), 7 (2023)
8. Jamil, B., Ijaz, H., Shojafar, M., Munir, K.: Irats: A drl-based intelligent priority and deadline-aware online resource allocation and task scheduling algorithm in a vehicular fog network. Ad Hoc Networks p. 103090 (2023)
9. Jamil, B., Ijaz, H., Shojafar, M., Munir, K., Buyya, R.: Resource allocation and task scheduling in fog computing and internet of everything environments: A taxonomy, review, and future directions. ACM Computing Surveys (CSUR) **54**(11s), 1–38 (2022)
10. Kang, K.X., Ding, D., Xie, H.M., Yin, Q., Zeng, J.: Adaptive drl-based task scheduling for energy-efficient cloud computing. IEEE Transactions on Network and Service Management (2021)
11. Kaur, N., Kumar, A., Kumar, R.: A systematic review on task scheduling in fog computing: Taxonomy, tools, challenges, and future directions. Concurrency and Computation: Practice and Experience **33**(21), e6432 (2021)
12. Khan, A.A., Zakarya, M., Khan, R.: Energy-aware dynamic resource management in elastic cloud datacenters. Simulation modelling practice and theory **92**, 82–99 (2019)
13. Liu, X., Fan, L., Xu, J., Li, X., Gong, L., Grundy, J., Yang, Y.: Fogworkflowsim: An automated simulation toolkit for workflow performance evaluation in fog computing. In: 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE). pp. 1114–1117. IEEE (2019)
14. Lohi, S.A., Tiwari, N.: Assessment of suitability of metaheuristics and machine learning for task scheduling process: A review of aptness in heavy task environments. Smart Trends in Computing and Communications: Proceedings of SmartCom 2020 pp. 419–425 (2021)
15. Ran, L., Shi, X., Shang, M.: Slas-aware online task scheduling based on deep reinforcement learning method in cloud environment. In: 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS). pp. 1518–1525. IEEE (2019)
16. Sellami, B., Hakiri, A., Yahia, S.B., Berthou, P.: Deep reinforcement learning for energy-efficient task scheduling in sdn-based iot network. In: 2020 IEEE 19th International Symposium on Network Computing and Applications (NCA). pp. 1–4. IEEE (2020)

17. Silva, D.M.A.d., Asaamoning, G., Orrillo, H., Sofia, R.C., Mendes, P.M.: An analysis of fog computing data placement algorithms. In: Proceedings of the 16th EAI international conference on mobile and ubiquitous systems: computing, networking and services. pp. 527–534 (2019)
18. Sorensen, N.: Industry benchmarks performance (2011)
19. Yang, X., Rahmani, N.: Task scheduling mechanisms in fog computing: review, trends, and perspectives. Kybernetes (2020)
20. Zhang, L., Qi, Q., Wang, J., Sun, H., Liao, J.: Multi-task deep reinforcement learning for scalable parallel task scheduling. In: 2019 IEEE International Conference on Big Data (Big Data). pp. 2992–3001. IEEE (2019)