20th IMEKO TC10 Conference
"Measurement for Diagnostics, Optimization and Control to Support Competitiveness and Innovation"
Lisbon, Portugal, September 28–29, 2026

Parallelized Reinforcement Learning Environment for Industrial Production Optimization

Zsolt János Viharos[1,2], Bence Pálvölgyi[1]

[1] Research Laboratory on Engineering and Management Intelligence of the HUN-REN Institute for Computer Science and Control (SZTAKI), Center of Excellence of the Hungarian Academy of Sciences (MTA),
Budapest, 1111, Hungary,
viharos.zsolt@sztaki.hu, palvolgyi.bence@sztaki.hu
[2]Faculty of Economics and Business of the John von Neumann University,
Kecskemét, 6000, Hungary,

Abstract **– Reinforcement Learning (RL) has shown strong potential in optimizing production and manufacturing systems, where complex decision-making processes can be modeled as sequential control problems. However, training RL agents in such environments is often computationally expensive and sensitive to stochasticity, leading to unstable learning dynamics and long convergence times. This paper investigates the application of parallelized RL training in a production environment, focusing on its impact on training stability and computational efficiency. By employing vectorized environments and running multiple simulations concurrently, the agent is exposed to a more diverse set of state transitions, reducing variance in policy updates and improving robustness. At the same time, parallelization enables significant reductions in wall-clock training time, providing practical speedup without altering the underlying learning algorithm or the environment. Experimental results on a simulated production system demonstrate that increasing the number of parallel environments leads to more stable learning curves and faster convergence, although diminishing returns appear beyond a certain level of parallelism. The findings highlight the importance of parallel training in scalable RL deployments and provide guidelines for balancing computational resources with learning performance.**

Keywords – IMEKO, TC10, Diagnostics, Optimisation, Control, Reinforcement Learning, Production Systems, Parallelization, Industry, Innovation and Infrastructure.*

I. INTRODUCTION

Direct optimization of real-world industrial production systems is typically impractical due to cost, safety constraints, and limited accessibility for iterative experimentation. For these reasons, accurate simulations are built to mimic the real-world environment, Digital Twins (DT), as Markov Decision Processes (MDP). Within this context, Reinforcement Learning (RL), has emerged as a viable approach for sequential decision-making in manufacturing and production environments. RL-based methods have been increasingly adopted for tasks such as scheduling, routing, and resource allocation as shown by Kegyes et al. [1].

However, as production environments grow in complexity, incorporating multiple interacting components, high-dimensional state representations, and constrained action spaces, the computational burden of training RL agents increases significantly. Traditional approaches to hyperparameter tuning and performance evaluation, which rely on repeated sequential training runs, become computationally expensive and time-consuming. Moreover, stochasticity in both the environment and the learning process introduces high variance, often leading to unstable convergence behavior.

To address these challenges, parallelized training using vectorized simulation environments has gained attention as a practical solution [2]. By executing multiple environment instances concurrently, the learning process benefits from a more diverse set of state transitions within each update cycle, which reduces variance in policy updates and improves robustness, with little computer overhead (Fig. 1). This way, in addition, parallelization reduces wall-clock training time, making extensive experimentation more feasible.

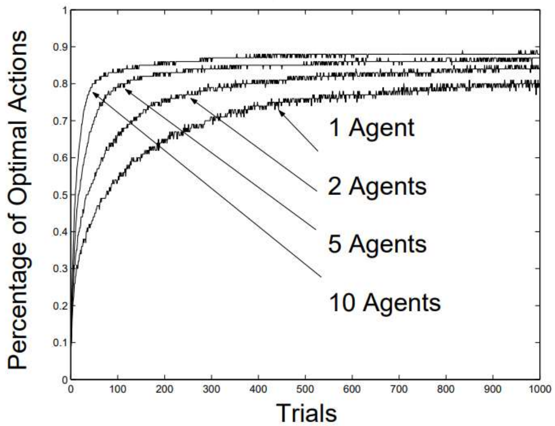

Fig. 1. Result comparison of multiple agents (Parallel Reinforcement Learning)

Computer resources are finite, so increasing the level of parallelism introduces synchronization overhead and hardware constraints, leading to diminishing returns beyond a certain scale.

This paper investigates the role of parallelized RL training in an industrial production setting, with a focus on its impact on convergence stability, robustness, and computational efficiency.

II. RELATED RESULTS IN THE LITERATURE

Efficient implementation of machine learning algorithms has been extensively studied, with particular emphasis on vectorization, memory optimization, and parallel execution. The book "Vectorization: A Practical Guide to Efficient Implementation of Machine Learning Algorithms" [3] provides a comprehensive overview of these techniques, covering topics such as data-level parallelism, memory allocation strategies, and multithreading. These principles form the foundation of many modern high-performance learning systems, including Reinforcement Learning (RL) frameworks that rely on vectorized environments to process multiple samples simultaneously.

At the system level, Daghaghi et al. [4] demonstrates how hardware-aware optimization can significantly improve training efficiency. The SLIDE framework, implemented in C++, leverages sparse hash-table-based backpropagation and exploits AVX (Advanced Vector Extensions) for vectorized computation, achieving up to sevenfold speedup on the same hardware. This highlights that substantial performance gains can be achieved through careful alignment of algorithms with underlying hardware architectures.

García et al. [5] presents an implementation designed specifically for scalability in high-performance computing environments. By utilizing both shared-memory and distributed-memory parallelism, the approach reduces memory requirements and enables processing of significantly larger datasets. Such design considerations are directly relevant for RL applications involving complex environments and large-scale simulation data.

Focusing on industrial RL applications, Cordeido et al. [6] emphasizes the challenges posed by slow simulation speeds and stochastic process dynamics. The study introduces parallelization strategies alongside runtime-based evaluation metrics, demonstrating that scalable implementations are essential for practical deployment in industrial settings where sequential training would be computationally infeasible.

Weishaupt et al. [7] provides empirical validation of vectorized training in an RL context. Using a collision-free path planning task with a seven-degree-of-freedom Franka Research 3 robotic manipulator, the authors show that collecting experience from multiple environments in parallel reduces wall-clock training time, improves training stability, and leads to more generalized policies. This supports the notion that parallelized sampling not only accelerates learning but also enhances policy robustness.

Overall, these works collectively demonstrate that parallelization—ranging from low-level vectorization to high-level environment parallelism—is a key enabler for scaling machine learning methods. In RL-based industrial optimization, vectorized environments offer a particularly effective approach, balancing implementation complexity with tangible improvements in training speed and stability.

III. DESCRIPTION OF THE METHOD

The proposed approach is developed in collaboration with the industrial partner Opel, ensuring that the modeled system reflects realistic production constraints and operational characteristics. The environment represents a scalable industrial process with multiple interacting components, where decision-making is influenced by stochastic events and asynchronous system behavior. A central challenge in such settings is the non-deterministic nature of the process, which can lead to softlock situations, states in which the system remains operational but becomes effectively unproductive due to cyclic agent behavior.

The environment is implemented using Gymnasium, providing a standardized interface for Reinforcement Learning (RL). Parallelized training is realized Stable Baselines3, where vectorized environments are used to execute multiple independent simulation instances concurrently. A single agent is trained across N parallel environments, enabling the collection of diverse experience within each update step while maintaining a centralized learning paradigm.

To evaluate the effectiveness of parallelization beyond short training horizons, experiments are conducted using extended benchmarks of x-hour duration. This setup allows the analysis of long-term learning behavior, including the impact of initialization overhead and early training instability relative to sustained performance improvements. By observing agent behavior over prolonged simulated operation, the method provides insight into both convergence characteristics and robustness against degradation into softlock states.

IV. RESULTS AND DISCUSSIONS

The results (Fig. 2) demonstrate a clear efficiency gain from parallelization up to a moderate number of environments, followed by diminishing returns at higher scales. Increasing the number of environments from 1 to 2 yields a substantial improvement in efficiency across all measured configurations, with peak performance observed around 2–4 environments depending on the setup. In this region, the system achieves near or even above nominal efficiency relative to the 100% baseline, indicating highly effective utilization of parallel computation and reduced idle time. However, beyond this point, efficiency begins to decline steadily as the number of environments increases to 8, 16, and 32. This degradation can be attributed to synchronization overhead, increased communication costs, and hardware limitations, which offset the benefits of additional parallelism. Notably, longer training configurations (e.g., 10-step and 16-step setups) maintain higher efficiency across larger environment counts compared to shorter runs, suggesting that increased computational workload per update helps amortize parallelization overhead. Overall, the results confirm that parallelization significantly improves training performance, but only up to an optimal scale, beyond which efficiency decreases.

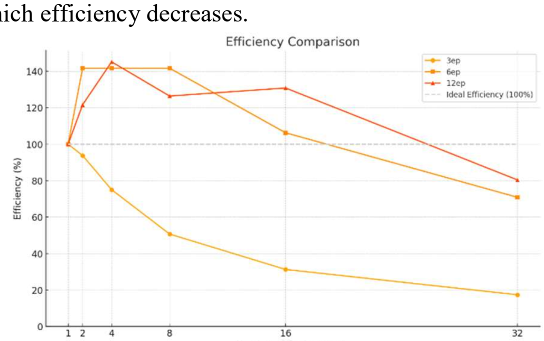

Fig. 2. Efficiency of Agent on multiple environments

V. CONCLUSIONS AND OUTLOOK

In this section the results of the paper are to be summarized and outlook for further research can be appointed shortly here.

VI. ACKNOWLEDGMENTS

The research was supported by the European Commission (EC) through the DiGreeS project under grant No. 101178079 and by AI EDIH Hungary project under EC grant No. 101120929.

Special thank has to be expressed for Jenő Csanaki, Krisztián Meskó, Zsolt Nagy at the Opel Szentgotthárd Kft. for their continuous collaboration.

On behalf of Project "Comprehensive testing of machine-learning algorithms" we thank for the usage of HUN-REN Cloud () that significantly helped us achieving the results published in this paper.

REFERENCES

[1] **Kegyes, T., Süle, Z., & Abonyi, J.**: The applicability of reinforcement learning methods in the development of Industry 4.0 applications. Complexity, 2021, Article 7179374. 10.1155/2021/7179374

[2] **R. Matthew Kretchmar**: Parallel Reinforcement Learning, 2002, Mathematics and Computer Science, Denison University Granville, OH 43023, USA

[3] **Edward, D. C.**: Vectorization: A Practical Guide to Efficient Implementation of Machine Learning Algorithms.

[4] **Shabnam, D., Nicholas, M., Mengnan, Z., Yong, W., Sameh, G., Charlie, T., Anshumali, S.**: Accelerating SLIDE Deep Learning on Modern CPUs: Vectorization, Quantizations, Memory Optimizations, and More, Proceedings of Machine Learning and Systems 3, 2021, p. 156--166.

[5] **García, J. M..**: Parallelization of a graph-based multi-view machine learning algorithm, TRABALLO FIN DEGRAO GRAOENENXEÑARÍAINFORMÁTICA MENCIÓNENCOMPUTACIÓN, 2023.

[6] **Guilherme, P. C., Eduardo, M. F. B. J., Michell T. F. S.**: Optimization of code for scalability of deep reinforcement learning agents in slow and stochastic industrial process patterns, XI INTERNATIONAL SYMPOSIUM ON INNOVATION AND TECHNOLOGY, Quantum Technologies: The information revolution that will change the future, 2025.

[7] **S. Weishaupt, R. Husmann and H. Aschemann.**: Improving Generalization and Training Speed of Deep Reinforcement Learning-Based Robotic Path Planning With Vectorized Environments, IECON 2025 – 51st Annual Conference of the IEEE Industrial Electronics Society, Madrid, Spain, 2025, pp. 1-7