WARSAW UNIVERSITY OF TECHNOLOGY Index 351733
FACULTY OF CIVIL ENGINEERING
COMMITTEE FOR CIVIL AND WATER ENGINEERING
POLISH ACADEMY OF SCIENCES ISSN 1230-2945
DOI: 10.24425/ace.2026.158596
ARCHIVES OF CIVIL ENGINEERING
Vol. LXXII ISSUE 2 2026
pp. 93–107

Research paper

Deep excavation wall design using reinforcement learning

Maksym Beichuk[1], Kostiantyn Zdor[2], Monika Mitew-Czajewska[3]

Abstract: This study investigates the application of reinforcement learning (RL) for obtaining near-optimal designs of diaphragm walls in geotechnical engineering. A physics-based numerical environment is developed to simulate soil–structure interaction, relying on a Winkler spring formulation with pressure-dependent soil springs to approximate the nonlinear response of the ground. This modelling framework allows the agent to evaluate candidate designs through physically meaningful structural responses rather than surrogate performance indicators. To reflect realistic engineering practice, a specifically designed action space and reward function are formulated, incorporating both discrete design decisions and continuous geometric parameters. During training, the agent iteratively proposes a design configuration, observes the response computed by the physical simulator, and updates its policy based on the resulting reward signal. Several common RL algorithms are investigated, including Proximal Policy Optimization (PPO), REINFORCE, and the Parameterized Deep Q-Network (P-DQN), enabling a comparative assessment of policy-based and hybrid value-based approaches for this task. The algorithms are evaluated in terms of learning stability, convergence behaviour, and the quality of the resulting design solutions. The results demonstrate the potential of RL-based methods to explore complex design spaces efficiently while respecting physical constraints, highlighting their suitability for supporting automated or decision-assisted design of diaphragm walls.

Keywords: design optimization, diaphragm walls, deep reinforcement learning, retaining wall, winkler spring model

[1]MSc. Eng., Warsaw University of Technology, Faculty of Civil Engineering, Al. Armii Ludowej 16, 00-637 Warsaw, Poland, e-mail: maksym.beichuk.dokt@pw.edu.pl, ORCID: 0009-0000-6506-9840

[2]Ph.D., National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Department of Digital Technologies in Energy, Prospect Beresteiskyi 37, 03056, Kyiv, Ukraine, e-mail: kostya9919moonlight@gmail.com, ORCID: 0009-0008-7640-1499

[3]DSc., PhD., Eng., Warsaw University of Technology, Faculty of Civil Engineering, Al. Armii Ludowej 16, 00-637 Warsaw, Poland, e-mail: monika.mitew@pw.edu.pl, ORCID: 0000-0002-2651-2026

1. Introduction

Recent advances in artificial intelligence have opened new opportunities for automating and optimizing engineering design processes. While machine learning (ML) techniques have been successfully applied to tasks ranging from predicting construction costs [1] and labour productivity [2] to optimizing building energy efficiency [3], the potential for autonomous design agents remains largely unexplored.

In particular, reinforcement learning (RL) has shown promise in applications where sequential decision-making is required, and feedback can be quantified through a reward function. Although RL has been explored in some structural contexts, such as obstacle avoidance in BIM environments [4] or structural topology optimization [5], this study investigates the potential of RL algorithms for geotechnical design problems, focusing specifically on the design of reinforced concrete walls (i.e. diaphragm walls) with struts.

In current engineering practice, evolutionary algorithms and metaheuristics are commonly used by researchers and practitioners to tackle complex design optimization tasks. These approaches, such as Particle Swarm Optimization (PSO) and Genetic Algorithms (GA), have been successfully applied to applications ranging from the structural design of prefabricated buildings [6] to the optimization of reinforced concrete frames [7].

In geotechnical engineering, Gandomi et al. [8] demonstrated that swarm intelligence techniques can successfully optimize retaining walls by minimizing material costs under strict stability constraints. However, despite their effectiveness, these approaches treat each scenario in isolation and do not leverage knowledge gained from previous successful designs to accelerate future optimization tasks.

In the experiment presented in this paper, reinforcement learning is applied to obtain a semi-optimal design for a problem configuration. While, in such isolated cases, RL does not offer a decisive advantage over metaheuristic approaches, its long-term potential is significantly greater. Unlike metaheuristic algorithms, an RL agent can be saved, pretrained, and reused for new problems. As the agent is exposed to many additional tasks, it gradually acquires a deeper understanding of how to propose effective designs based on geometry, material properties, and boundary conditions. Over time, this cumulative learning ability may provide a substantial advantage over traditional optimization methods.

2. Experiment

The objective is to train an agent to identify an optimal structural design for a diaphragm wall. The agent learns by repeatedly proposing a design, observing the response produced by the physical simulator, and updating its strategy based on the reward it receives. In this study, several modern reinforcement learning algorithms were evaluated, including Proximal Policy Optimization (PPO) [9], REINFORCE [10], and Parameterized Deep Q-Network (P-DQN) [11], to compare their performance on this geotechnical design task.

For simplicity, a uniform soil layer was assumed, and the influence of underground water was neglected. The range of soil parameters was deliberately constrained to simplify the implementation at the current stage of agent development. Specifically, the friction angle was limited to 30–35°, cohesion to 0-5 kPa, excavation depth to 5–15 m, oedometric modulus to 30–50 MPa, and surface load to 10–50 kPa. During training, the agents were presented with problems in which parameters were randomly selected within these bounds. This approach enabled performance evaluation across multiple design challenges, with the agent trained independently on each dataset to assess initial behavior. Time-dependent effects, such as excavation sequence or soil consolidation, were not considered at this stage. Additionally, it is assumed that the retaining wall is a reinforced concrete element and that only struts are considered as support elements. In terms of possible decisions or actions, each agent can modify the embedment length, add up to five struts (including the option of having no struts), specify the vertical position of each strut, select a strut profile from the CHS section library for each row, adjust strut spacing, and choose the wall thickness. To ensure the action space remains reasonable, boundaries were imposed: embedment length is limited to 2–30 m, strut spacing to 1–10 m, the maximum number of strut rows is five, and wall thickness ranges from 0.6–1.2 m. Because of the diverse action space, agents handle two types of action: continuous and categorical. Continuous actions include embedment length, strut spacing, and strut vertical positions, while all other decisions are treated as categorical, meaning the agent selects from predefined options.

2.1. Learning loop and agents' description

The first agent implemented in this study uses the REINFORCE algorithm. The agent operates in an episodic setting, and the learning loop proceeds as presented in Fig. 1. It is important to note that during training, the agent receives the same geotechnical parameters in every episode. Consequently, this setup does not test generalization but rather functions as a stochastic optimizer for a specific problem. In essence, this pipeline can be interpreted as a contextual bandit problem, where each episode corresponds to a single optimization trial conditioned on the current site state.

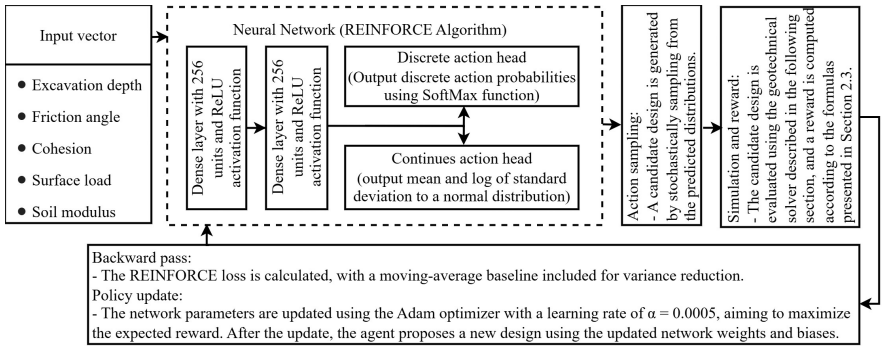

Fig. 1. REINFORCE learning loop

The second agent implemented in this study is based on the PPO algorithm. The agent operates in a batched setting, and the learning loop proceeds as presented in Fig. 2.

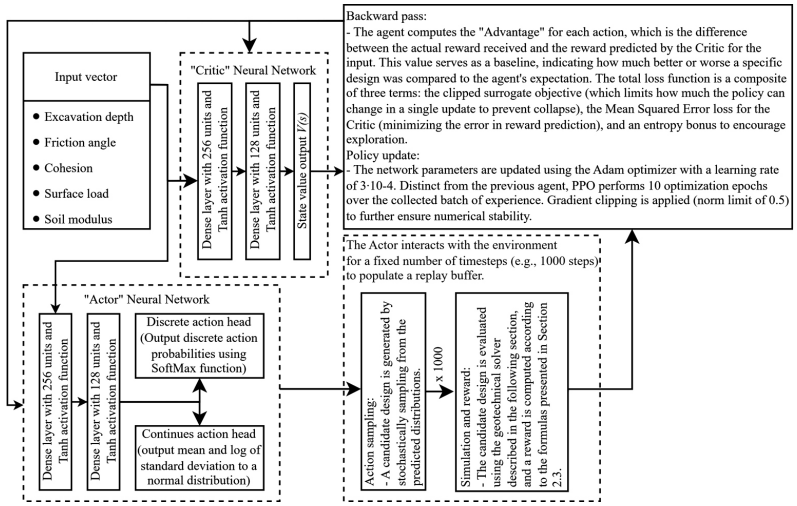

Fig. 2. PPO learning loop

The third agent implemented in the study utilizes the P-DQN algorithm and the learning loop proceeds as presented in Fig. 3.

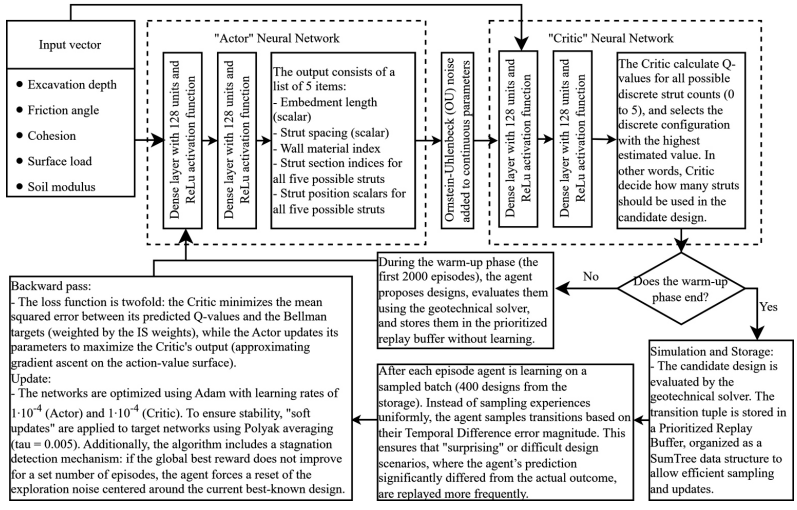

Fig. 3. P-DQN learning loop

It is worth noting that, unlike the REINFORCE agent, the PPO or P-DQN agent can be pre-trained on a diverse set of prior projects to learn general design patterns and strategies, after appropriate adjustments of the Critic network.

2.2. Environment

For this experiment, a geotechnical solver for deep excavation wall design was developed and was used as the physical simulator (i.e., the environment) in which the agents operated. A Winkler spring model [12] with pressure-dependent soil springs was selected as the environment for reinforcement learning because it is less computationally demanding than commercial software. This reduced complexity enables much faster evaluation of individual learning episodes. Rapid episode computation is essential for comparing different reinforcement learning agents efficiently, without long simulation times that would not be practical in a real design workflow. The wall is modelled as a Euler-Bernoulli beam with initial soil springs distributed evenly along both sides and analyzed using the finite element method. The global stiffness matrix is assembled from beam element stiffness matrices, with additional contributions from Winkler soil springs and axial strut springs acting on translational degrees of freedom only.

Before the main calculation loop, the lateral earth pressures are determined. These pressures are computed using the classic Rankine earth pressure theory. The at-rest earth pressure coefficient is obtained using Jaky's formula for normally consolidated soils, or the Mayne and Kulhawy formula for over-consolidated soils. The beam is divided into elements, and for each node the active, passive, and at-rest forces are calculated. Active and passive forces are treated as limiting values and are applied only when the corresponding mobilizing displacements occur. Otherwise, they do not contribute their full calculated magnitude. The stiffness of the soil springs is defined using the subgrade reaction modulus, which is estimated using the Schmitt method [13]. The soil-structure interaction is modeled using elastic-perfectly plastic soil springs distributed along the wall. The wall response is obtained through an iterative procedure in which spring yielding is checked and the global stiffness matrix is updated until convergence is achieved. The iterative procedure is presented in Fig. 4.

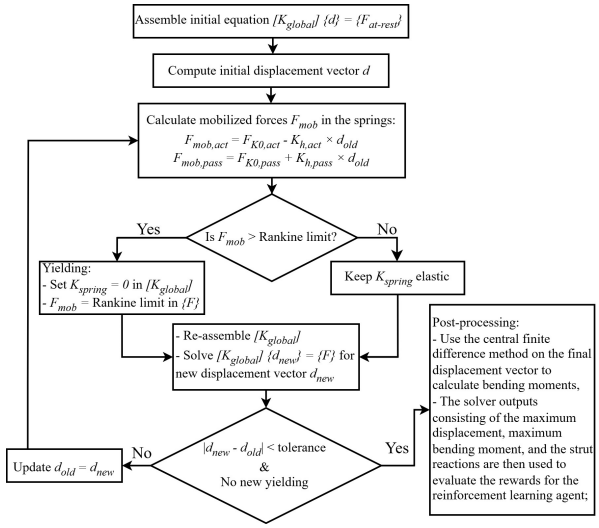

Fig. 4. Iterative loop scheme

2.3. Reward calculation

The goal was to formulate reward functions that reflect the actual objectives of structural engineers during the design process. These checks include the bending moment utilization of the wall, the utilization of the struts, verification of maximum deflection, and verification of the strut spacing and embedment length. The checks related to strut spacing and embedment length are included to represent economic considerations in practical design. For example, excessively small strut spacing, such as 1 m, is rarely economically justified. It was decided that, during the learning process, rewards should always be negative or zero, meaning that every design decision results in some level of penalty. Consequently, the agent's objective is to maximise the value of the reward function, i.e., to minimise the imposed penalties.

Figure 5 illustrates how the wall bending-moment utilization and the strut utilization components of the reward function are calculated based on their respective utilization ratios.

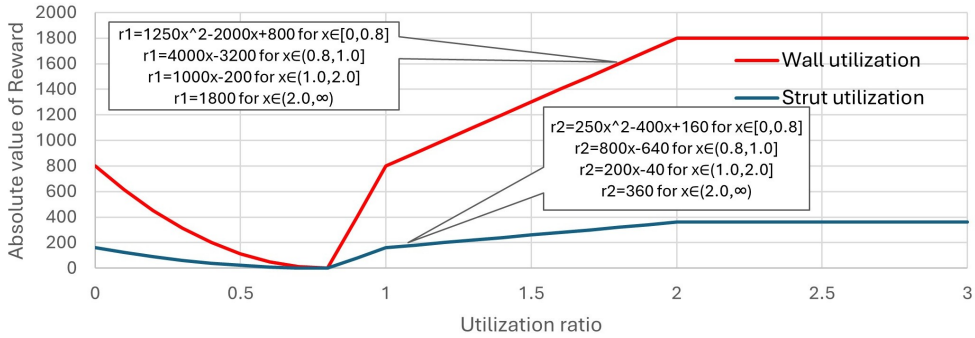

Fig. 5. The wall and struts utilization part of the reward function

The bending moment utilization of the wall is calculated as the ratio of the maximum bending moment to the wall's moment resistance capacity. The wall resistance capacity is determined by assuming a rectangular reinforced concrete section with a height of 1 m and a width specified by the agent, with a steel reinforcement ratio of 1.5%. The calculation follows the principles of Eurocode 2 for a singly reinforced section, assuming reinforcement only on the tension side. Strut utilization is calculated as the ratio of the reaction force to the strut's capacity. The strut capacity represents the maximum elastic compressive force that the strut can sustain before failure. Based on the cross-section selected by the agent, the compressive capacity is calculated as the product of the strut material yield strength and its cross-sectional area. A library of properties for circular tubes CHS has been used in the reinforcement learning loop. Figure 6 illustrates how the struts spacing and the wall embedment length components of the reward function are calculated based on their respective values.

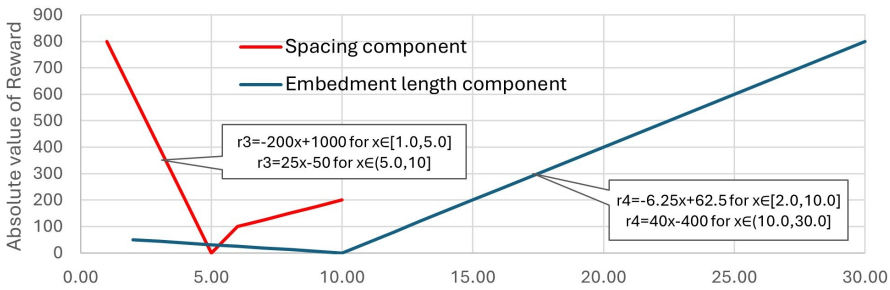

Fig. 6. Spacing part of the reward function

Figure 7 illustrates how the wall deflection component of the reward function is calculated based on respective values.

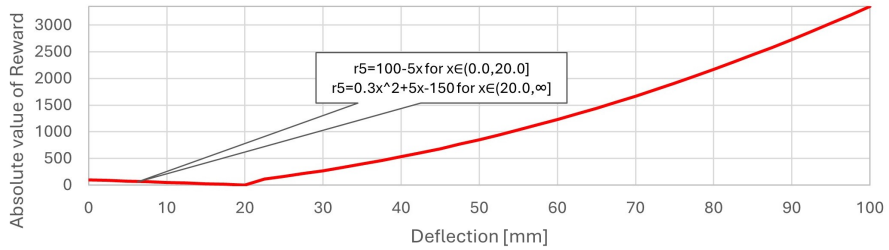

Fig. 7. Deflection part of the reward function

The final penalty is obtained by taking the negative of the sum of all individual components of the reward functions.

2.4. Results

Figure 8 shows the performance of the REINFORCE algorithm. Reward convergence over 10000 episodes. The solid black line represents the median reward across a set of independent training runs. The shaded regions depict the percentile intervals (10th–90th, 20th–80th, etc.), illustrating the distribution of performance. The darkest region represents the inter- quartile range (approximate), indicating where the highest density of trial outcomes is concentrated.

The agent demonstrates a rapid initial learning phase, with the median reward improving from approximately –5000 to –1000 within the first 2000 episodes. Subsequently, performance stabilizes, as evidenced by the plateauing of the median line and achieving approximately –700 reward at the end of the learning cycle. However, even in the later stages (episodes 4000–10000), the median reward displays persistent high-frequency oscillations and sharp fluctuations. This behavior indicates that the policy remains sensitive to stochastic elements in the environment and has not settled into a global optimum. Furthermore, because REINFORCE relies on Monte Carlo sampling, the policy is updated only once per episode and exclusively based on the reward from that episode. This leads to intrinsically high variance in the gradient estimates, which further contributes to the unstable learning dynamics observed in the results.

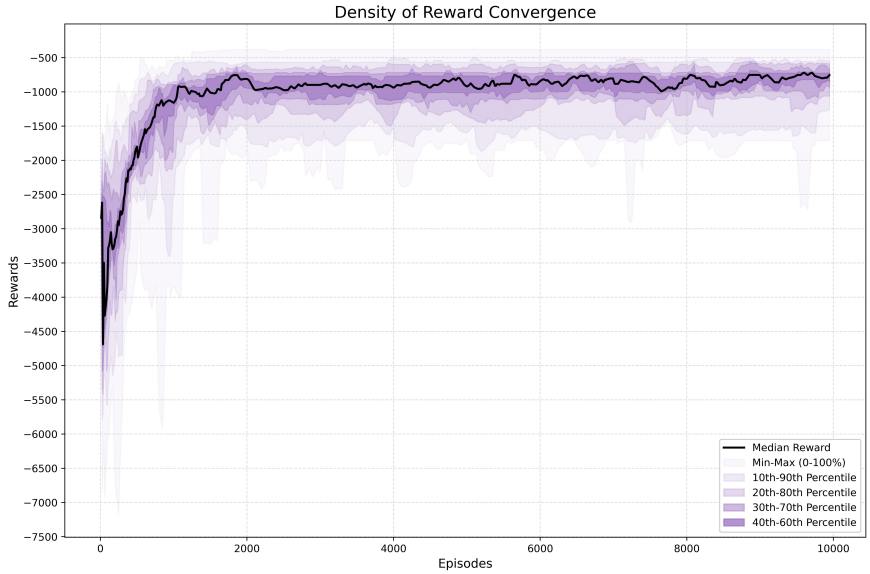

Fig. 8. REINFORCE learning convergence

Figure 9 illustrates the performance of the PPO algorithm over 50000 episodes. As in the previous case, the plot shows the statistical distribution of achieved rewards across multiple training runs.

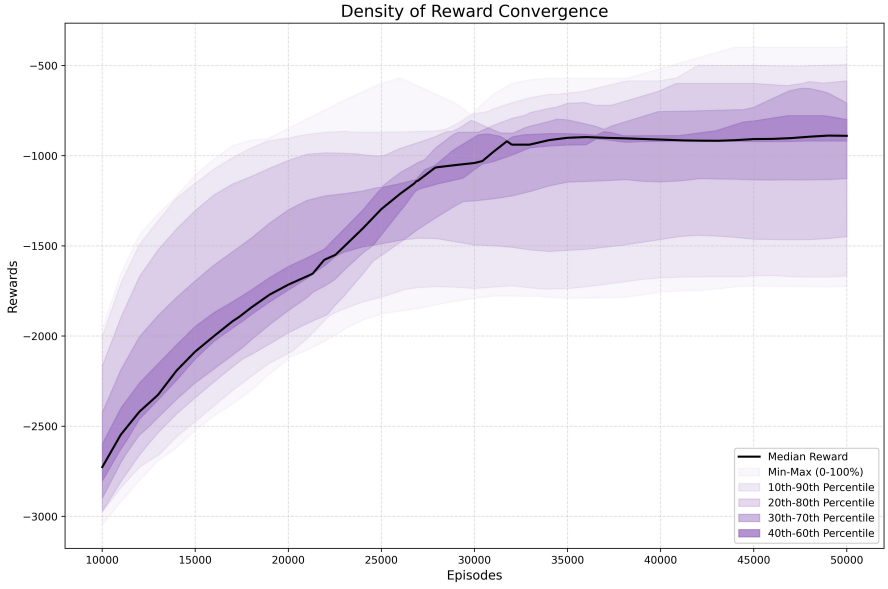

Fig. 9. PPO learning convergence

The agent exhibits a smooth and consistent learning phase, with the median reward increasing from approximately –3000 to around –1000 within the first 30000 episodes. After this point, the performance gradually stabilizes, as indicated by the flattening of the median curve, ultimately reaching values near –900 toward the end of the training process. In this experiment, the PPO policy is updated only once every 1000 episodes. Such infrequent updates produce very gradual changes to the policy parameters, resulting in a slow but steady trend towards improved performance. Consequently, the reward distribution narrows and the median reward progressively shift toward zero, reflecting the agent's increasingly stable and moderate improvements.

Figure 10 presents the performance of the P-DQN algorithm over more than 12000 training episodes.

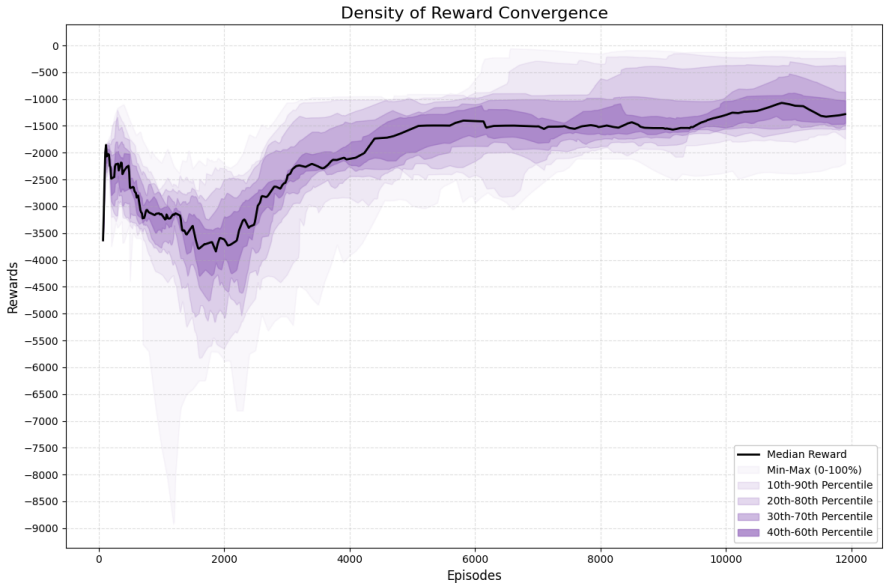

Fig. 10. P-DQN learning convergence

During the first 2000 episodes, the average reward is highly unstable and noisy, reflecting the exploration phase in which the agent attempts to understand the problem and tests a wide range of design configurations. After this initial period, the algorithm gradually begins to improve and converges toward more optimal solutions. By the end of the learning cycle, the median reward reaches approximately –1250.

Compared to the REINFORCE agent, the later stages of P-DQN training exhibit greater stability, with noticeably reduced variance in the reward distribution and a more consistent learning trend. However, this increased stability comes at the cost of a poorer median reward in the late episodes, indicating that the algorithm converges more reliably but to a slightly inferior solution. On the other hand, the median reward exhibits higher variance compared to the PPO agent.

In summary, all three reinforcement learning algorithms (PPO, REINFORCE, and P-DQN) demonstrate the ability to learn and progressively improve their performance in the retaining-wall design task. Among them, PPO shows the most stable and smooth learning behaviour, with low reward variance and consistent convergence. However, this stability comes at the cost of sample efficiency, because PPO requires a very large number of episodes before achieving high-quality results.

In contrast, both REINFORCE and P-DQN reach competitive or near-optimal designs much earlier in the training process, which makes them more suitable for real-world engineering applications where each simulation step is computationally expensive. Their drawback is a substantially higher variance in the learning process, especially during the early and middle stages of training. Between the two, P-DQN provides noticeably more stable late-stage performance and sometimes achieves significantly smaller penalties in some cases than REINFORCE, making it the more robust option.

The performance of each agent was tested on problems with different combinations of input parameters, as described previously. A more detailed description of agent performance for an exemplary set of task parameters is presented in Fig. 11, which shows the optimization trajectory of the P-DQN agent while solving the retaining wall problem with the following parameters: a uniform soil layer with a oedometric modulus of 30 MPa, a friction angle of 31.5°, zero cohesion, a surface load of 10 kPa, and an excavation depth of 10.0 m.

The optimization trajectory below presents the evolution of key design quantities, i.e. strut spacing, wall deflection, and wall utilization recorded throughout the training process. For clarity, only the episodes in which a new best reward was achieved are shown, illustrating how the agent progressively identified more efficient design configurations.

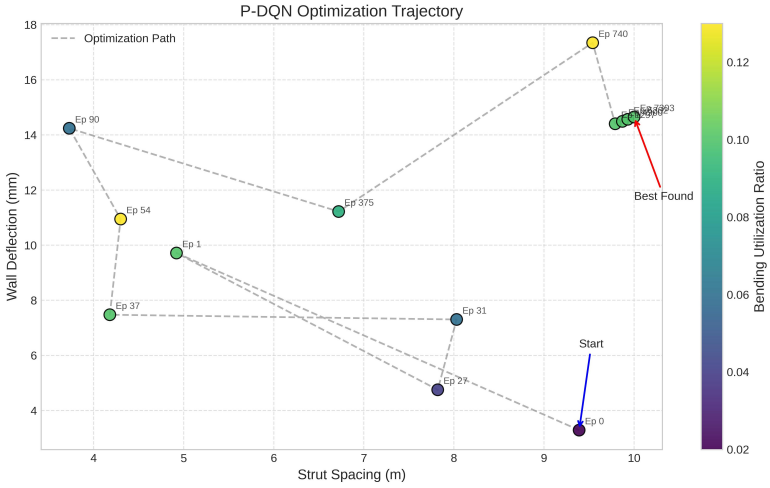

Fig. 11. P-DQN trajectory for exemplary problem

The best-performing design identified by the P-DQN algorithm for this problem is a retaining wall with an embedment length of 17.5 m and a single row of struts spaced at 9.9 m. The struts are positioned 3.2 m below the top of the wall, while the wall itself has a thickness of 0.80 m and utilizes a CHS457t10 strut profile.

This proposed configuration was subsequently validated using commercial FEM software to verify its structural feasibility and ensure that all components meet the required utilisation and safety criteria. According to the Plaxis 2D analysis, the maximum wall deflection is 12 mm, and the maximum bending moment is –401.4 kNm/m, corresponding to an approximate bending utilisation ratio of 0.13. The bending moment and horizontal displacement distributions generated by Plaxis 2D are presented in Fig. 12 and Fig. 13 respectively. The following settings were used in the program: Mohr–Coulomb soil model, wall (modelled as a plate element) with interfaces (Rinter is 0.67), fixed-end anchor and triangular mesh (minimum element area: 0.355 m^2). These results indicate that the implemented geotechnical solver tends to overpredict deflections and underpredict internal forces relative to the more advanced FEM model. However, this simplified geotechnical solver is sufficient for training the agents and for comparing their performance against one another.

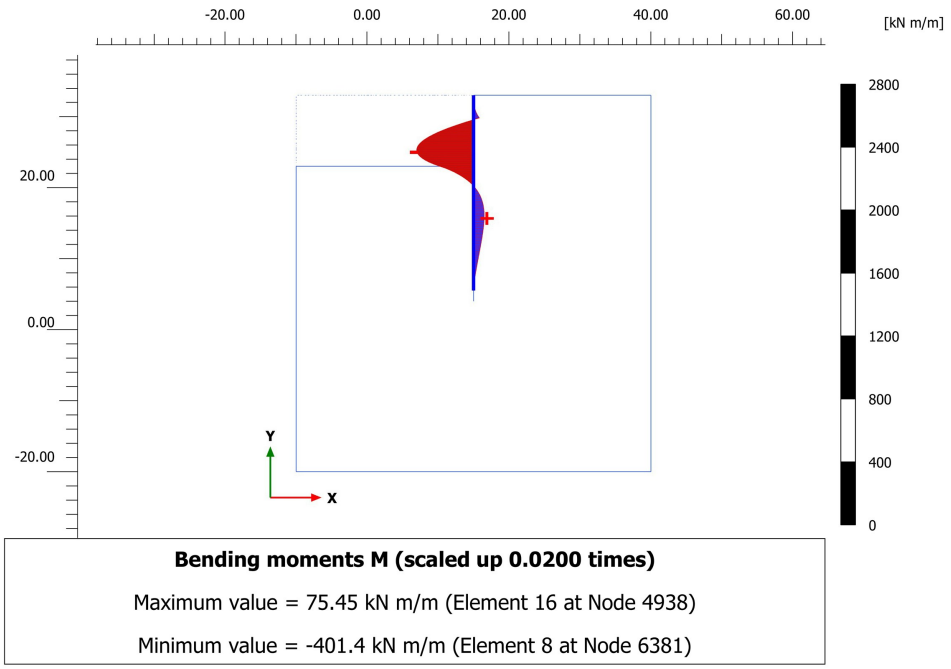

Fig. 12. Bending moments diagram for exemplary problem (PLAXIS 2D Output)

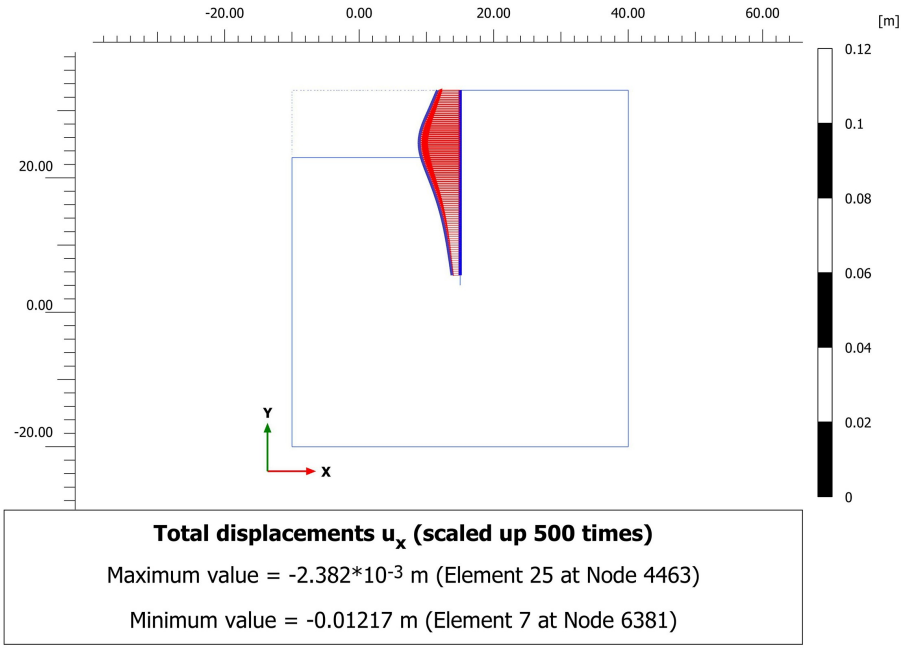

Fig. 13. Total displacements diagram for the exemplary problem (PLAXIS 2D Output)

3. Summary and conclusions

Several studies have investigated the application of ML methods to problems similar to those addressed in this study, such as retaining wall design. For example, [14] applied ML models to predict the factor of safety and reliability index of retaining walls under uncertain soil parameters. In contrast, the RL framework developed in the present study focuses on improving the design configuration through iterative modifications over successive training episodes. RL has also been investigated for structural optimization. In [15], RL combined with metaheuristic algorithms was used to optimize reinforced concrete retaining walls with respect to cost and environmental impact. In that study, the agent explores the design space by taking discrete actions to adjust wall dimensions. In the present study, the design process is formulated using modern deep reinforcement learning algorithms operating in a mixed discrete-continuous design space, allowing more flexible and detailed modification of deep excavation wall parameters.

This article presents the implementation of three RL algorithms for solving a geotechnical problem related to deep excavation design. A physics-based environment was developed using a Winkler spring model with pressure- dependent soil springs to idealize the behaviour of the real structure. In addition, a dedicated action space and reward function were proposed to reflect realistic design choices.

The objective of the agents is to identify an optimal solution using the smallest possible number of steps after receiving problem data. The algorithms were tested using a set of predefined input parameters chosen to represent realistic field conditions. The primary performance metrics were the evolution of the mean reward over episodes and the dispersion of reward values in the late-stage episodes after convergence to a plateau. It is important to note that achieving a reward equal to zero is practically impossible for certain cases with highly challenging problem parameters (e.g., a combination of a 15 m excavation depth, a 50 MPa surface load, and unfavourable soil conditions). Additionally, the agent's behaviour is highly sensitive to the reward function. The current formulation can sometimes produce uneconomical results (for example, suggesting a total wall height of 27.5 m for an excavation depth of 10 m with struts) and therefore requires further refinement. Therefore, the mean reward was evaluated over the entire dataset, which included both less and more favourable conditions, rather than for a single specific parameter set.

The experimental results indicate that medium reward closest to zero is achieved with REINFORCE algorithm (approximately –700), followed by PPO (approximately –900), and finally P-DQN (approximately –1250). However, only the P-DQN agent occasionally achieved rewards close to zero, reaching values near –200, typically in less challenging cases. The median performance of the P-DQN agent improved from –3800 to –1250, corresponding to an improvement factor of 3.04. The PPO agent improved from –2750 to –900, corresponding to improvement factor of 3.06. The REINFORCE agent demonstrated the largest improvement, with median performance increasing from –4750 to –700, corresponding to an improvement factor of 6.79. Although REINFORCE exhibits the highest improvement rate, the current agent architecture limits the possibility of introducing a pretraining strategy.

For future development, the proposed algorithms will be applied using commercial software such as PLAXIS 2D as the simulation environment. Despite the increased computational cost, this approach is expected to provide more robust results due to the use of a realistic numerical framework. Furthermore, the PPO and P-DQN agents will be pretrained using real-world project data, which constitutes a key direction of further work. This pretraining is intended to improve sample efficiency and accelerate convergence, enabling the agents to generalize previously unseen geotechnical problems.

References

[1] M.F. Hasan, O.H. Abdullah, and K.S. Albayati, "Estimate final cost of roads using support vector machine", *Archives of Civil Engineering*, vol. 68, no. 4, pp. 669-682, 2022, doi: 10.24425/ace.2022.143061.

[2] D.A. Nguyen, D.Q. Tran, T.N. Nguyen, and H.H. Tran, "Modeling labor productivity in high-rise building construction projects using neural networks", *Archives of Civil Engineering*, vol. 69, no. 1, pp. 675-692, 2023, doi: 10.24425/ace.2023.144195.

[3] C. Gu, "Optimize building energy efficiency design and evaluation with machine learning", *Archives of Civil Engineering*, vol. 71, no. 1, pp. 615-629, 2025, doi: 10.24425/ace.2025.153353.

[4] H. Chai and J. Guo, "Design optimization of obstacle avoidance of intelligent building steel bar by integrating reinforcement learning and BIM technology", *Archives of Civil Engineering*, vol. 70, no. 1, pp. 621-634, 2024, doi: 10.24425/ace.2024.148932.

[5] N.K. Brown, A.P. Garland, G.M. Fadel, and G. Li, "Deep reinforcement learning for engineering design through topology optimization of elementally discretized design domains", *Materials & Design*, vol. 218, art. no. 110672, 2022, doi: 10.1016/j.matdes.2022.110672.

[6] C. Li, "The application of ICPA optimization algorithm in multi-objective optimization structural design of prefabricated buildings", *Archives of Civil Engineering*, vol. 70, no. 4, pp. 57-70, 2024, doi: 10.24425/ace.2024.151879.

[7] S. Chutani and J. Singh, "Optimal design of RC frames using a modified hybrid PSOGSA algorithm", *Archives of Civil Engineering*, vol. 63, no. 4, pp. 123-134, 2017, doi: 10.1515/ace-2017-0044.

[8] A.H. Gandomi, A.R. Kashani, D.A. Roke, and M. Mousavi, "Optimization of retaining wall design using recent swarm intelligence techniques", *Engineering Structures*, vol. 103, pp. 72-84, 2015, doi: 10.1016/j.engstruct.2015.08.034.

[9] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms", *arXiv preprint arXiv:1707.06347*, 2017, doi: 10.48550/arXiv.1707.06347.

[10] R.J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning", *Machine Learning*, vol. 8, no. 3, pp. 229-256, 1992, doi: 10.1007/BF00992696.

[11] J. Xiong, et al., "Parametrized deep q-networks learning: Reinforcement learning with discrete-continuous hybrid action space", *arXiv preprint arXiv:1810.06394*, 2018, doi: 10.48550/arXiv.1810.06394.

[12] D. Lourenço, F. Schnaid, and M.M. Rocha, "Finite elements and Winkler model applied to retaining walls design", in *Proceedings of the 14th Pan- American Conference on Soil Mechanics and Geotechnical Engineering*. Toronto, Canada, 2011.

[13] P. Schmitt, "Estimating the coefficient of subgrade reaction for diaphragm wall and sheet pile wall design", *Revue Française de Géotechnique*, no. 71, pp. 3-10, 1995, doi: 10.1051/geotech/1995071003 (in French).

[14] P. Mishra, P. Samui, and E. Mahmoudi, "Probabilistic design of retaining wall using machine learning methods", *Applied Sciences*, vol. 11, no. 12, art. no. 5411, 2021, doi: 10.3390/app11125411.

[15] J. Lemus-Romani, D. Ossandón, R. Sepúlveda, N. Carrasco-Astudillo, V. Yepes, and J. García, "Optimizing retaining walls through reinforcement learning approaches and metaheuristic techniques", *Mathematics*, vol. 11, no. 9, art. no. 2104, 2023, doi: 10.3390/math11092104.

Projektowanie ścian obudowy głębokich wykopów z wykorzystaniem algorytmów uczenia ze wzmocnieniem

Słowa kluczowe: głębokie uczenie ze wzmocnieniem, model sprężyn Winklera, optymalizacja projektowania, ściana oporowa, ściany szczelinowe

Streszczenie:

Artykuł dotyczy zastosowania algorytmów uczenia ze wzmocnieniem, będących jedną z metod sztucznej inteligencji, w zagadnieniach optymalizacji projektowania ścian szczelinowych stanowiących obudowę głębokich wykopów. Celem badań było opracowanie podejścia umożliwiającego automatyczne poszukiwanie rozwiązań projektowych spełniających wymagania nośności i użytkowalności przy jednoczesnym racjonalnym doborze parametrów geometrycznych i materiałowych konstrukcji. Wybrano metody wykorzystujące sztuczną inteligencję ze względu na ich zdolność do kumulacji doświadczenia. W miarę eksponowania agenta na kolejne problemy projektowe, systematycznie nabywa on głębszą wiedzę na temat zależności między geometrią, właściwościami materiałów a warunkami brzegowymi, co umożliwia generowanie coraz bardziej efektywnych rozwiązań. W artykule analizowano skuteczność algorytmów uczenia ze wzmocnieniem przy założeniu braku wcześniejszego uczenia przed rozpoczęciem interakcji

z każdym kolejnym zadaniem, w celu oceny i opisu zachowania bazowego. W celu odzwierciedlenia rzeczywistej praktyki inżynierskiej zdefiniowano przestrzeń decyzyjną obejmującą zarówno decyzje dyskretne, takie jak dobór materiałów czy liczba elementów konstrukcyjnych, jak i parametry ciągłe związane z geometrią ściany. Funkcję celu sformułowano w taki sposób, aby penalizowała rozwiązania nie spełniające kryteriów projektowych. Proces uczenia polega na iteracyjnej aktualizacji polityki decyzyjnej algorytmu na podstawie informacji zwrotnej uzyskiwanej ze środowiska obliczeniowego. Środowisko obliczeniowe zostało opracowane z zastosowaniem modelu Winklera, z wykorzystaniem metody parć zależnych. W pracy przeanalizowano działanie wybranych algorytmów uczenia ze wzmocnieniem, w tym Proximal Policy Optimization (PPO), REINFORCE oraz Parameterized Deep Q-Network (P-DQN). Przeprowadzono ich porównanie pod względem stabilności procesu uczenia, szybkości zbieżności oraz jakości otrzymywanych rozwiązań projektowych. Uzyskane wyniki potwierdzają potencjał metod uczenia ze wzmocnieniem jako narzędzi wspomagających proces projektowania ścian szczelinowych, szczególnie w kontekście analizy złożonych problemów decyzyjnych oraz integracji obliczeń inżynierskich z metodami sztucznej inteligencji.

Received: 2026-01-11, Revised: 2026-03-09