Automatic enhancement of vascular configuration for self-healing concrete through reinforcement learning approach

Wan, Zhi; Xu, Yading; Chang, Ze; Liang, Minfei; Šavija, Branko

DOI
10.1016/j.conbuildmat.2023.134592

Publication date
2024

Document Version
Final published version

Published in
Construction and Building Materials

Citation (APA)
Wan, Z., Xu, Y., Chang, Z., Liang, M., & Šavija, B. (2024). Automatic enhancement of vascular configuration for self-healing concrete through reinforcement learning approach. *Construction and Building Materials*, *411*, Article 134592. https://doi.org/10.1016/j.conbuildmat.2023.134592

Construction and Building Materials 411 (2024) 134592

Contents lists available at ScienceDirect

Construction and Building Materials

journal homepage: www.elsevier.com/locate/conbuildmat

Automatic enhancement of vascular configuration for self-healing concrete through reinforcement learning approach

Zhi Wan [a], Yading Xu [a,*], Ze Chang [a,b], Minfei Liang [a], Branko Šavija [a]

[a] Faculty of Civil Engineering and Geosciences, Delft University of Technology, 2628CN Delft, the Netherlands
[b] Department of Mechanical Engineering, Eindhoven University of Technology, 5600MB Eindhoven, the Netherlands

ARTICLE INFO

Keywords:
Concrete
Self-healing
Reinforcement learning
Optimization
Numerical simulation

ABSTRACT

Vascular self-healing concrete (SHC) has great potential to mitigate the environmental impact of the construction industry by increasing the durability of structures. Designing concrete with high initial mechanical properties by searching a specific arrangement of vascular structure is of great importance. Herein, an automatic optimization method is proposed to arrange vascular configuration for minimizing the adverse influence of vascular system through a reinforcement learning (RL) approach. A case study is carried out to optimize a concrete beam with 3 pores (representing a vascular network) positioned in the beam midspan within a design space of 40 possibilities. The optimization is performed by the interaction between RL agent and Abaqus simulation environment with the change of target properties as a reward signal. The results illustrates that the RL approach is able to automatically enhance the vascular arrangement of SHC given the fact that the 3-pore structures that have the maximum target mechanical property (i.e., peak load or fracture energy) are accessed for all of the independent runs. The RL optimization method is capable of identifying the structure with high fracture energy in the new optimization task for 4-pore concrete structure.

1. Introduction

Self-healing concrete (SHC) shows great possibility in mitigating the environmental impact of the construction industry by increasing the durability of structures and thereby reducing the need for repair or replacement [1]. Based on the previous research [2], vascular-based healing is one of the most important approaches to self-healing. Compared with the intrinsic and capsule-based self-healing system, incorporating a vascular system into matrix enables an ongoing supply of the healing agents [3]. Except for healing agents and vascular materials [4,5], the performance of SHC embedded with vascular networks is significantly influenced by the vascular configuration.

A number of studies on design the vascular network of self-healing materials has been investigated based on fluid flow [6–9]. In some cases, however, the healing agent is pressurized to the cracked region and the fluid flow property of vascular networks are less important. Instead, the vascular configuration of SHC is designed based on mechanical properties [10–15]. Among others, some researchers designed vascular system with large tube coverage considering that a dense distribution of vascular system raises the likelihood that a crack intersects the network and triggering the healing process. On the other hand, the existence of vascular networks may cause stress concentration in the host matrix and compromise the mechanical properties [16–18]. Therefore, it is of necessity to balance the tradeoff between large coverage (for better healing capacity) and strength reduction.

Except for theoretical analysis and trail-and-error approaches [19], heuristic optimization methods such as evolutionary algorithms and machine learning (ML) have been employed to design the vascular structures under certain configurations. For example, the vascular arrangement in self-cooling polymeric materials could be optimized using genetic algorithm (GA), where the influencing factors like network redundancy are taken into account [20]. More importantly, ML has emerged as a promising way to analyze and optimize materials [21,22]. Compared with analytical and numerical models, ML models speed up the prediction of target properties for the similar new structures. Besides, the well-trained model could be employed for inverse design [23]. For vascular self-healing concrete, generative deep neural network (GDNN) is adopted to arrange the vascular system [24]. Although a structure with better target property has been found after the optimization process, there are still several issues present in the optimization process: (1) a large dataset is needed to train the ML model (i.e., map concrete structure to the target property), causing huge computational

* Corresponding author.
E-mail address: Y.Xu-5@tudelft.nl (Y. Xu).

https://doi.org/10.1016/j.conbuildmat.2023.134592
Received 14 October 2023; Received in revised form 8 December 2023; Accepted 12 December 2023
Available online 22 December 2023

burden; (2) the ML-recommended structures must be verified by simulation software to obtain the best structure; (3) the ML model must be re-trained with a new dataset when the optimizing structure or the target property are changed. As a result, the large dataset for training deep neural network (DNN) and the ML-recommended structure verification are time-intensive, thereby compromising the optimization effectiveness.

In recent years, reinforcement learning (RL) has shown significant success in solving complex problems such as game playing [25,26] and robotics [27]. Compared with gradient based ML methods, an RL agent can select the sequential actions that maximize the future rewards by iteratively interacting with their operation environment [28]. Furthermore, the algorithm does not rely on prior knowledge nor the large amount of initialization samples. In engineering, deep reinforcement learning has been applied in scheduling of precast concrete production [29], design of reinforced concrete structures [30] and material optimization [31,32]. In the studies, most of the designed structures are simplified with discretized design domains since the interactive environments are created either by solving governing equations or satisfying the designing provisions. As a result, the optimized structures are still far from the practical engineering. It is noted that reinforcement learning outperforms the conventional optimization algorithms especially when the environment is complicated. To fully realize the optimization potential of reinforcement learning, interaction environment could be established by using numerical simulation to solve some practical problems. However, the optimization performance of RL is greatly influenced by the updating strategy and the hyperparameters. Furthermore, the number of interactions tends to be large, making it time-consuming to compute the reward with the numerical environment.

In this study, we aim to automatically optimize vascular configuration of vascular self-healing concrete through the interaction between an RL agent and a created numerical environment. Three-point bending test (3PBT) of concrete beam with different vascular arrangement is modelled with Abaqus/Explicit to create the numerical environment. Similar to previous research [33], the optimization objectives are defined as peak load or fracture energy of the concrete beams. To investigate the optimization effectiveness of different updating strategies, 3 pores out of the 40 positions are arranged as the optimization constraint. Afterwards, the optimization for the 4-pore structure towards high fracture energy is carried out using the selected updating strategy to design the vascular configuration. There are mainly three novelties in this research: (1) The vascular configuration optimization is transformed to a Markov decision process (MDP) and two updating strategies are recommended to optimize the vascular configuration. The framework of MDP could be easily used to optimize structures with different number of pores. (2) Numerical environment is established using Abaqus software, which could solve optimization problems close to practical engineering. The connection between RL agent and numerical environment realizes the automatic enhancement of structures. (3) Except for the maximum number of interaction steps, another termination criterion is set to accelerate the optimization process by using the historical highest value as the threshold value. The used termination criterion helps improve the optimization performance with a relatively small number of interactions.

This study is organized as follows: the concrete damage plasticity model and model calibration are described in Section 2.1 and data representation is introduced in Section 2.2. The formation of Markov decision process (MDP) is shown in Section 3.1 and the deep Q-learning approach to optimize vascular structures is described in Section 3.2. Optimization 84 of 3-pore concrete structure with two updating strategies (3→2→3 and 3→4→3) is compared and analyzed, and the result and discussion are provided in Section 4. Finally, a new optimization of 4-pore concrete structure is carried out (Section 5).

2. Numerical simulations

To optimize the vascular configuration using machine learning (ML) method, an accurate numerical model is necessary to generate the dataset or create the interaction environment. In addition to accuracy, computational time and mapping relationship need to be taken into consideration when selecting the numerical models. In this study, the mapping relationship is defined as the sensitivity of target mechanical property to the change of vascular structure. The mapping relationship could be reflected by the prediction accuracy of the trained neural network. In our previous work [15], we simulated the fracture response of 3D-printed ABS vascular self-healing concrete. However, the 3D model with ABS vascular system is extremely time-consuming, making it not suitable for data-driven optimization (see Appendix). In addition, considering that hollow channels could act as the vascular system for self-healing, a 3D model with hollow channel or 2D model with pores could be also employed to generate the dataset. For the 3D model with hollow channels, the fracture response showed not to be sensitive to the change of vascular structure, making it difficult to establish the mapping relationship between the vascular configuration and the mechanical properties. More details can be found in Appendix. Therefore, a 2D model with pores is the most suitable model for the vascular configuration optimization due to the favorable computation time and mapping relationship. We do, of course, acknowledge that this is a simplification of an actual vascular system, and that in practice a vascular network would preferably be oriented perpendicular to the crack plane (as attempted in the 3D model discussed in the Appendix).

In this study, 3-point bending tests on the notched concrete structures are numerically simulated. The pores in the design region act as the channels for transporting the self-healing agents. The radius of pores is set as 1.5 mm for the verification of designed structures with experiments in the future. To create a relatively large design space, the pores could be positioned in the compressive zone, and are not positioned in pairs. Based on the load-displacement curve, different mechanical properties are obtained and set as the optimization targets to calculate the reward signals. The schematic of the concrete structure with 4 pores is shown in Fig. 1.

2.1. Model calibration

2.1.1. Concrete damage plasticity model

Concrete damage plasticity model (CDPM) is often used to describe cementitious materials [34]. For CDPM, tension stiffening and compression hardening need to be defined and the corresponding stress-strain relations are shown in Eqs. (1) and (2), individually.

$$\sigma_t = (1-d_t)E_0(\varepsilon_t - \tilde{\varepsilon}_t^{pl}) \tag{1}$$

$$\sigma_c = (1-d_c)E_0(\varepsilon_c - \tilde{\varepsilon}_c^{pl}) \tag{2}$$

where σ_t, σ_c are the tensile stress and compressive stress respectively; d_t, d_c are tensile damage variable and compressive damage variable ranging from 0 (undamaged) to 1 (total loss of strength). E_0 is the initial (undamaged) elastic stiffness of the material; ε_t, ε_c are the total strains; $\tilde{\varepsilon}_t^{pl}$ and $\tilde{\varepsilon}_c^{pl}$ are the equivalent plastic strains. For simplification, the stiffness degradation is not considered and the damage variables are set to 0 in this study. As a result, the equivalent plastic strains are equal to crack strains. As shown in Fig. 2, the crack strains ($\tilde{\varepsilon}_t^{ck}$, $\tilde{\varepsilon}_c^{ck}$) are defined as the total strain minus the elastic strain corresponding to the undamaged materials (Eqs. (3) and (4)).

Where σ_t (σ_c) is the tensile (compressive) stress; d_t (d_c) is tensile (compressive) damage variable. E_0 is elastic modulus; ε_t and ε_c are the total strains; $\tilde{\varepsilon}_t^{pl}$ and $\tilde{\varepsilon}_c^{pl}$ are the equivalent plastic strains. In this study, d_t and d_c are set to 0. $\tilde{\varepsilon}_t^{ck}$ and $\tilde{\varepsilon}_c^{ck}$ are appeared in Eqs. (3) and (4) (see

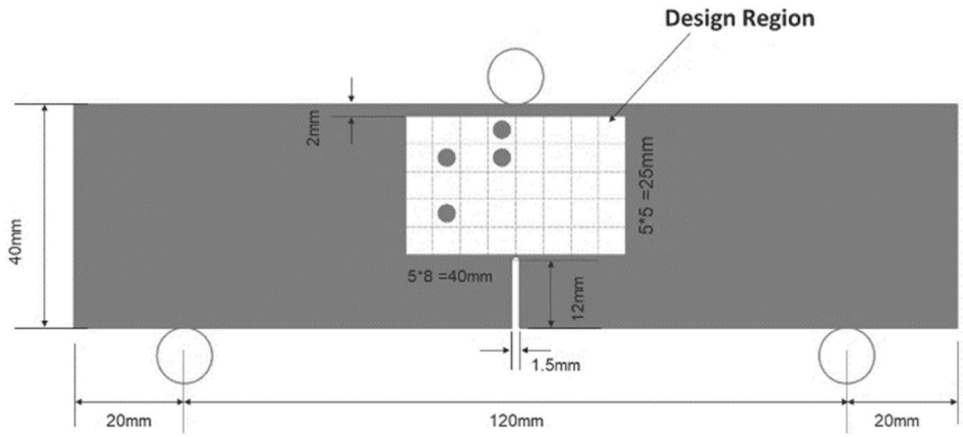


Fig. 1. Schematic of concrete structure under 3-point bending.

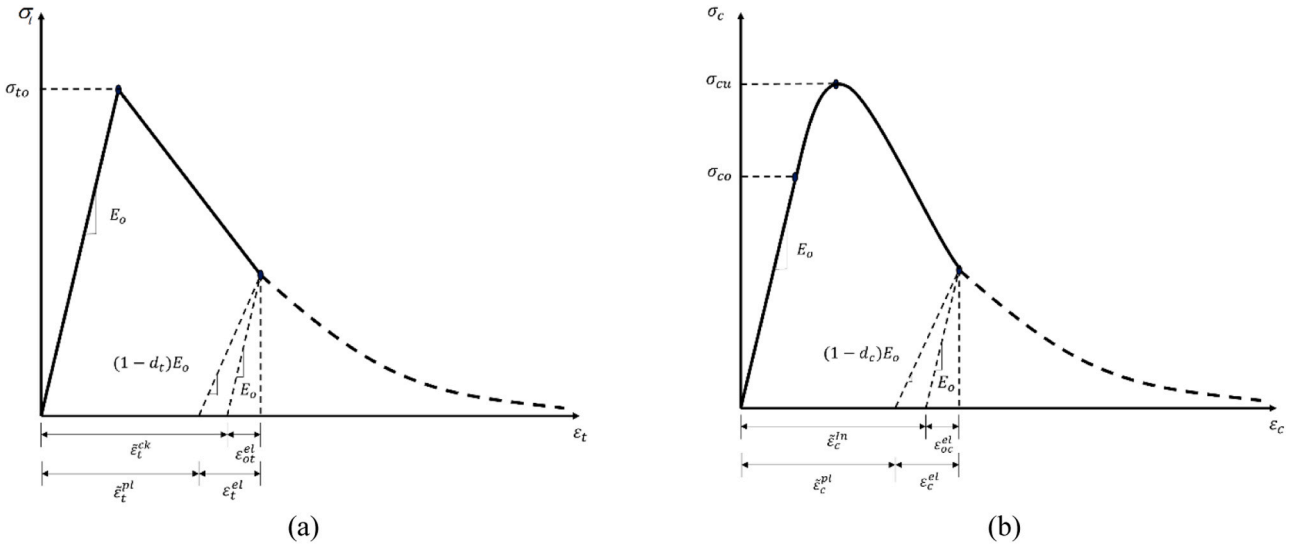


Fig. 2. Uniaxial tension/compression behavior for CDPM. (a) Tension; (b) Compression.

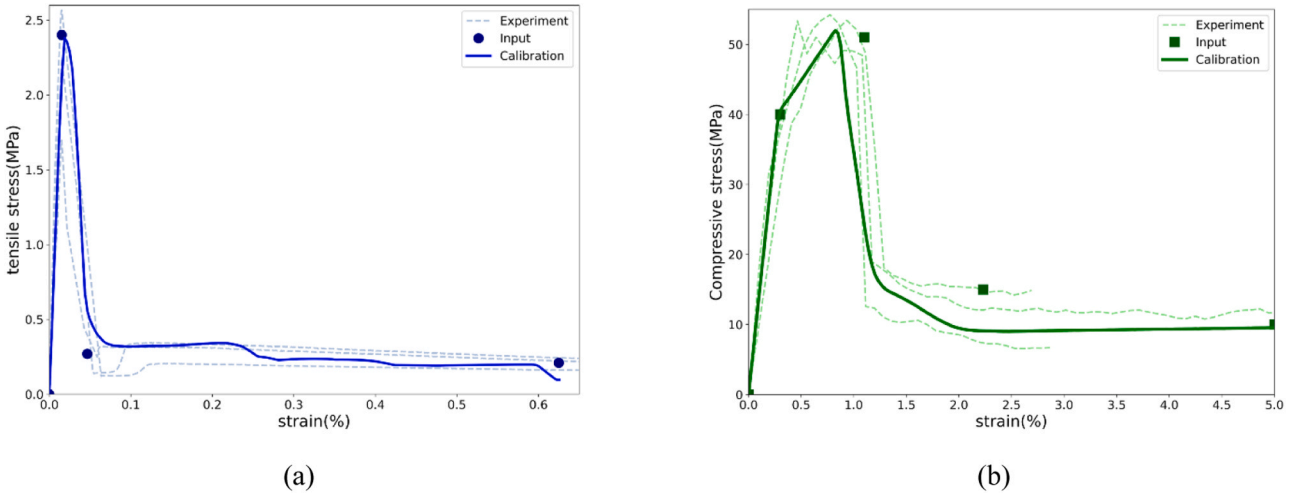


Fig. 3. Calibration result for CDPM. (a) Tension; (b) Compression.

Fig. 2).

$$\tilde{\varepsilon}_t^{ck} = \varepsilon_t - \varepsilon_{0t}^{el} \tag{3}$$

$$\tilde{\varepsilon}_c^{ck} = \varepsilon_c - \varepsilon_{0c}^{el} \tag{4}$$

2.1.2. Parameter determination

A cement mortar mix was prepared using CEM-III B and water in a 0.4:1 ratio by weight, sand (0.125–0.250 mm) in a ratio of 0.458:1 by weight. To prevent brittle failure during 3-point bending, 0.1 % of PE fibres are added to the cementitious matrix by volume. The input parameters for CDPM refer to our previous research [35].

Uniaxial tensile and compressive tests were carried out to calibrate the model parameters related to cementitious matrix. According to the obtained results (Fig. 3), the calibrated parameters of CDPM for the used mortar are listed in Table 1. The mesh size is set as 0.5 mm to ensure the simulation accuracy during the calibration process. Note that there is a slight difference between the calibrated curve and the input since the stiffness degradation is not considered in this study.

When simulating 3-point bending test, the displacement of concrete structure gradually increases to 0.4 mm in a time period of 20 s. The mesh size in the midspan is chosen as 0.5 mm, which is kept same with the mesh size in the calibration process. To save the computation times, the mesh size progressively transforms to 4 mm, then it stays 4 mm for the remaining 40 mm (Fig. 4). A structure without pores is first created and meshed to act as the prototype. The mesh of the prototype is symmetric to eliminate the impact of mesh size on numerical analysis. To avoid the influence of the mesh on the simulated response, structures with pores are generated based on the meshed prototype. As a result, the mesh of the structure is identical in all simulations, except for the existence of pores.

2.2. Data representation

2.2.1. Structure representation

To enable the automatic optimization process, an input file for numerical simulation should be automatically generated based on the updated state after taking an action. Although the concrete structure seems complicated, the structures could be characterized by the difference in the design region (Fig. 1).

In the simulation environment, the 40 positions are numbered from 0 to 39 in accordance with the Python programming convention. The total design space for structures with 3 and 4 pores is 9880 (C_{40}^3) and 91,390 (C_{40}^4) respectively. The 3-pore structure is sequentially encoded from combination 0 ([0, 1, 2]) to combination 9879 ([37–39]) and the same goes for 4 pores (combination 0 for [0, 1, 2, 3]; and combination 91,389 for [36–39]). Based on the updated state after taking an action, a structure number ranging from 0 to C_{40}^N-1 (N = 2, 3, 4) is passed to the Abaqus environment. Afterwards, the corresponding 3 or 4 positions are set as pores when generating the input file for numerical simulation.

In the RL environment, the state of concrete structure is represented with a 5 × 8 matrix. In particular, a location is encoded as 1 if there is a pore, and 0 otherwise. Subsequently, the 5 × 8 matrix is flattened into a 40-dimensional vector as the input state. An action is performed by converting the state from 0 to 1 (matrix to pore) or vice versa. An example of the input representation is shown in Fig. 5a.

Table 1
Input parameters for fiber reinforced mortar.

(a) Compressive CDPM parameters	
Yield stress (MPa)	**Inelastic strain (%)**
40	0
53	0.008
15	0.022
10	0.047
(b) Tensile CDMP parameters	
Yield stress (MPa)	**Displacement**
2.4	0
0.15	0.05
0.1	0.976

2.2.2. Target representation

The target mechanical property is initially defined as peak load to investigate the RL approach as well as to tune the hyperparameters of the Q-networks. Afterwards, structures are optimized for high fracture energy. In this study, a healing agent is assumed to be stored/transported in the pores. When crack hits the pores, the healing agent could flow into the crack from the pores. In other words, the structure with pores hit by the crack is regarded to be healable. At the same time, the influence of pores on the initial mechanical properties should be minimized. According to our previous research [24], more pores should be hit by the crack and therefore triggering the healing process and this results in higher fracture energy. In this study, fracture energy is defined as Eq. (5). An example of the optimization targets is shown in Fig. 5b.

$$Fracture \quad energy = \int_0^{0.4} Fds \tag{5}$$

Similar to the input generation, the output file should be automatically post-processed to obtain the target properties. In addition, the target value should be passed to the RL agent to calculate the reward corresponding to the state and action pair. Note that the Python running environment is different for RL agent training (Tensorflow framework), Abaqus running (system environment) and pre-/post- process (Abaqus Python). Therefore, the environment should be changed during the interaction process.

3. Reinforcement learning algorithm

3.1. Markov decision process (MDP) and Q-learning

Markov Decision Process (MDP) is a mathematically idealized form of RL problems [28]. The learner (decision maker) is called the agent. The agent takes an action based on the perceived state s_t and the state transfers to a new state s_{t+1}. To guide the action, a numerical value r_t is given as reward. The interaction between the agent and the environment continues until the perceived state is the terminal state s_T, where T is the final time step. MDPs are meant to be straightforward framing of the problem of learning from interaction to achieve a goal. The agent-environment interaction in a Markov decision process is shown in Fig. 6.

The aim of RL is to maximize the expected discounted return in an interaction episode, which is called the reward hypothesis. The expected discounted return can be expressed as shown in Eq. (6).

$$G_t = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{T-t-1} R_T = \sum_{k=0}^{T-t-1} \gamma^k R_{t+k+1} \tag{6}$$

where γ is a parameter called the discount rate.

Among the RL algorithms, Q-learning is a widely-used off-policy control method due to its capacity to converge to optimal policy even if acting sub-optimally [28]. Watkins was the first to introduce Q-learning, in which the value of a state-action pair is represented by Q(s,a), and the value is based on Eq. (7).

$$Q(s_t, A_t) \leftarrow Q(s_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, A_t)] \tag{7}$$

where α is the learning rate.

The learned action-value function $Q(s,a)$ directly approximate q_*, the optimal action-value function. In most cases, a q-table is sufficient to store the action-value pair when using Q-learning. However, the linear form cannot take into account any interactions between features. In this study, a neural network is employed for the nonlinear function

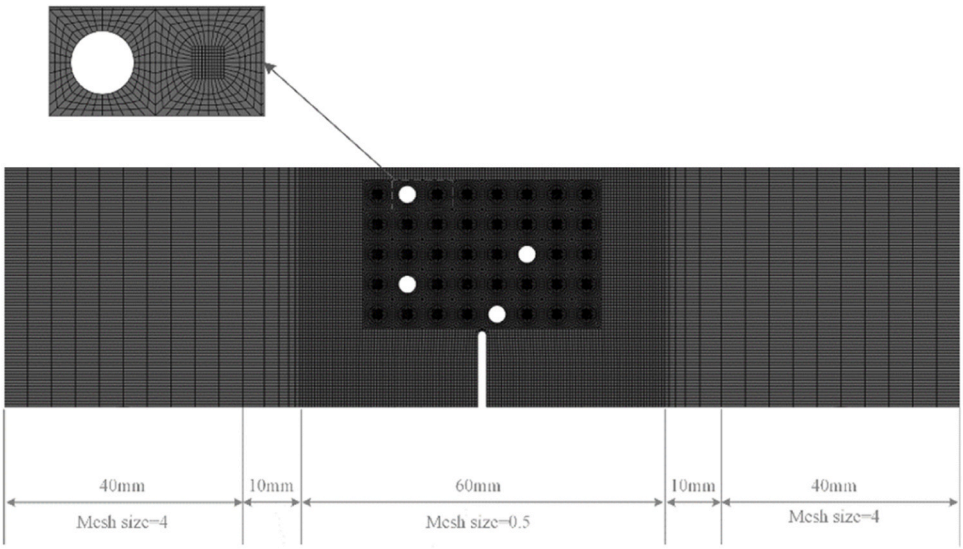


Fig. 4. Mesh size of the 4-pore structure under 3-point bending.

approximation. In other words, Q-network is adopted to describe the state-action value [29,36]. The input is the state s and the output is the parameterized Q function $Q_\theta(s,a)$, where θ are the weights of the neural network. To promote the convergence of the network, experience replay and two separated networks are used [37].

3.2. Deep Q-learning approach to optimize vascular structures

The optimization target is to design a concrete structure with high flexural strength or fracture energy under the predefined constraint. The update strategy is as follows: (1) A concrete structure with 3 pores is randomly generated as the start state s_0; (2) One agent (Agent 1) performs an action to decrease the number of pores from 3 to 2 (state transfers from s_t to $s_t^{'}$); (3) The other agent (Agent 2) performs an action to increase the number of pores from 2 to 3 with (state transfers from $s_t^{'}$ to s_{t+1}); (4) State s_t is updated to state s_{t+1} in step (3) until reaching the terminal state s_T.

The optimization of a 3-pore structure could also be performed via the update strategy of changing 3 pores to 4 pores, then changing from 4 pores to 3 pores. More importantly, note that structure with other number of pores could be optimized without significantly changing the programming code. Therefore, structures with 3 pores are first investigated with the RL framework. At the beginning of each episode, we initialize the concrete structure. To optimize the concrete structure with less computational burden, the interaction process does not start from a certain structure. Instead, the starting state is randomly generated at the beginning of each episode. Therefore, 3 unique integers in the range between 0 and 39 are randomly chosen. The 3 selected positions are encoded as 1′s (pores) and the remaining 37 positions are encoded as 0′s (matrix) to form a 40-dimensional vector, which is used as the start state. The optimization effectiveness via the update strategy of 3→2→3 and 3→4→3 is compared. The two updating strategies for 3-pore structure optimization are shown in Fig. 7.

Based on [38], an MDP is defined by seven elements, i.e., (1) set of state S; (2) Set of action A; (3) Transition function $P(s^{'}|s,a)$; (4) Reward function $R(s^{'},a,s)$; (5) start state S_0; (6) Discount factor γ; (7) Horizon H. According to previous studies [32], the discount factor is set to be 0.95. The other elements are introduced in the following section.

3.2.1. Set of state S and start state s_0

The entire design space is all of the possible structures when there are 3 pores out of the 40 positions in the middle span. Theoretically, there are 9880 (C_{40}^3) possibilities at most, and each state is represented by a 40-dimensional vector with 0′s and 1′s. For update strategy 1, the transition state is a 2-pore structure, where there are 780 (C_{40}^2) possibilities. The initialized state is called the start state (s_0).

Similarly, for the updating strategy of 3→4→3, the sets of state and transition state are 9880 (C_{40}^3, 3-pore structure) and 91,390 (C_{40}^4, 4-pore structure) respectively. The start state is also randomly generated at the beginning of each episode.

3.2.2. Action function and transition function

Based on the current state s_t, the agent will take actions to maximize the discounted future reward. Herein, actions are taken to convert the 'state' of one position: turn a position from pore to matrix (1→0) or from matrix to pore (0→1). Considering that the agent can only take one action in one interaction with the environment, two agents are created to take two separate actions in a row to maintain the pore number as 3. Taking the updating strategy 2 as example: Agent 1 first takes an action to turn one position from matrix to pore (0→1). To ensure that the pore number changes from 3 to 4, the action should be taken among the 37 matrix positions. Afterwards, Agent 2 takes another action $a^{'}$ to turn one position from pore to matrix (1→0). Similarly, the action $a^{'}$ acts on the 4 positions which are defined as pores. The transition functions are shown in Eq. (8)–(11) respectively.

For updating strategy 1 (3→2→3):

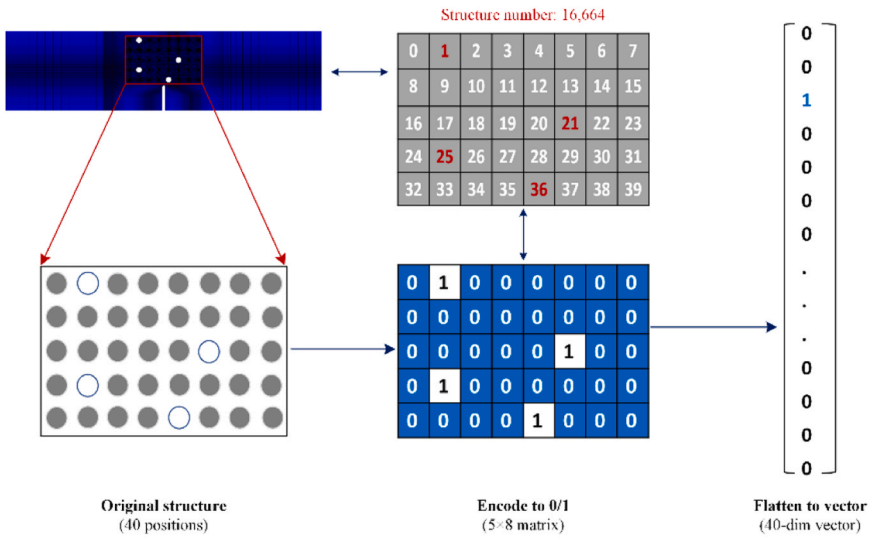
$$P\left(s_t^{'}|s_t,a^{'}\right)=\begin{cases}\frac{1}{3} & (for\quad pore\quad positions)\\ 0(for\quad other\quad positions)\end{cases} \tag{8}$$

$$P\left(s_{t+1}|s_t^{'},a^{'}\right)=\begin{cases}\frac{1}{38} & (matrix\quad positions)\\ 0(other\quad positions)\end{cases} \tag{9}$$

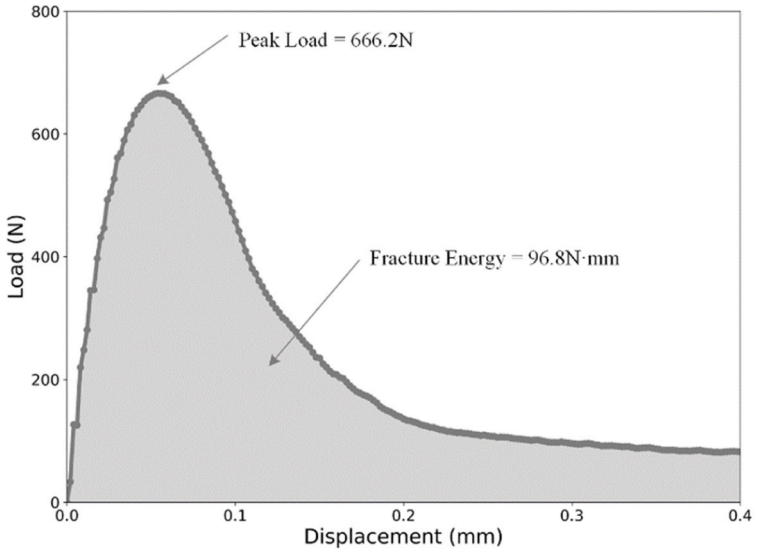
For updating strategy 2 (3→4→3):

$$P\left(s_t^{'}|s_t,a^{'}\right)=\begin{cases}\frac{1}{37} & (matrix\quad positions)\\ 0(other\quad positions)\end{cases} \tag{10}$$

$$P\left(s_{t+1}|s_t^{'},a^{'}\right)=\begin{cases}\frac{1}{4} & (pore\quad positions)\\ 0(other\quad positions)\end{cases} \tag{11}$$



(a)



(b)

Fig. 5. Representation of structure and target. (a) Structure representation; (b) Target representation.

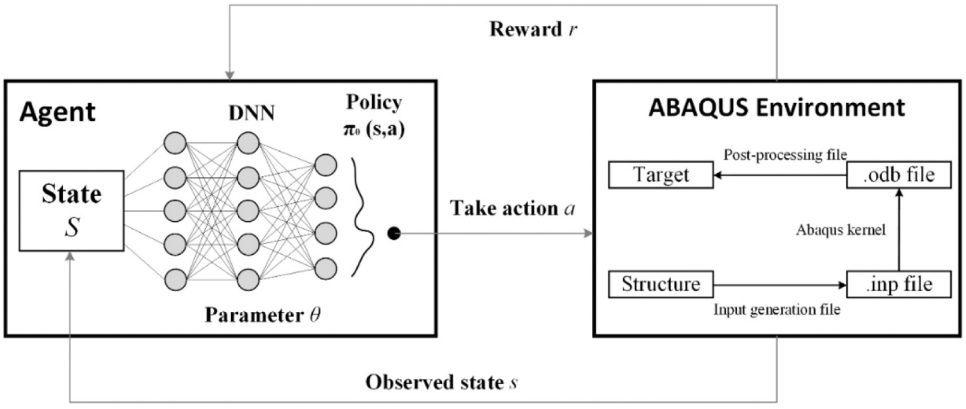


Fig. 6. Agent-environment interaction in MDP.

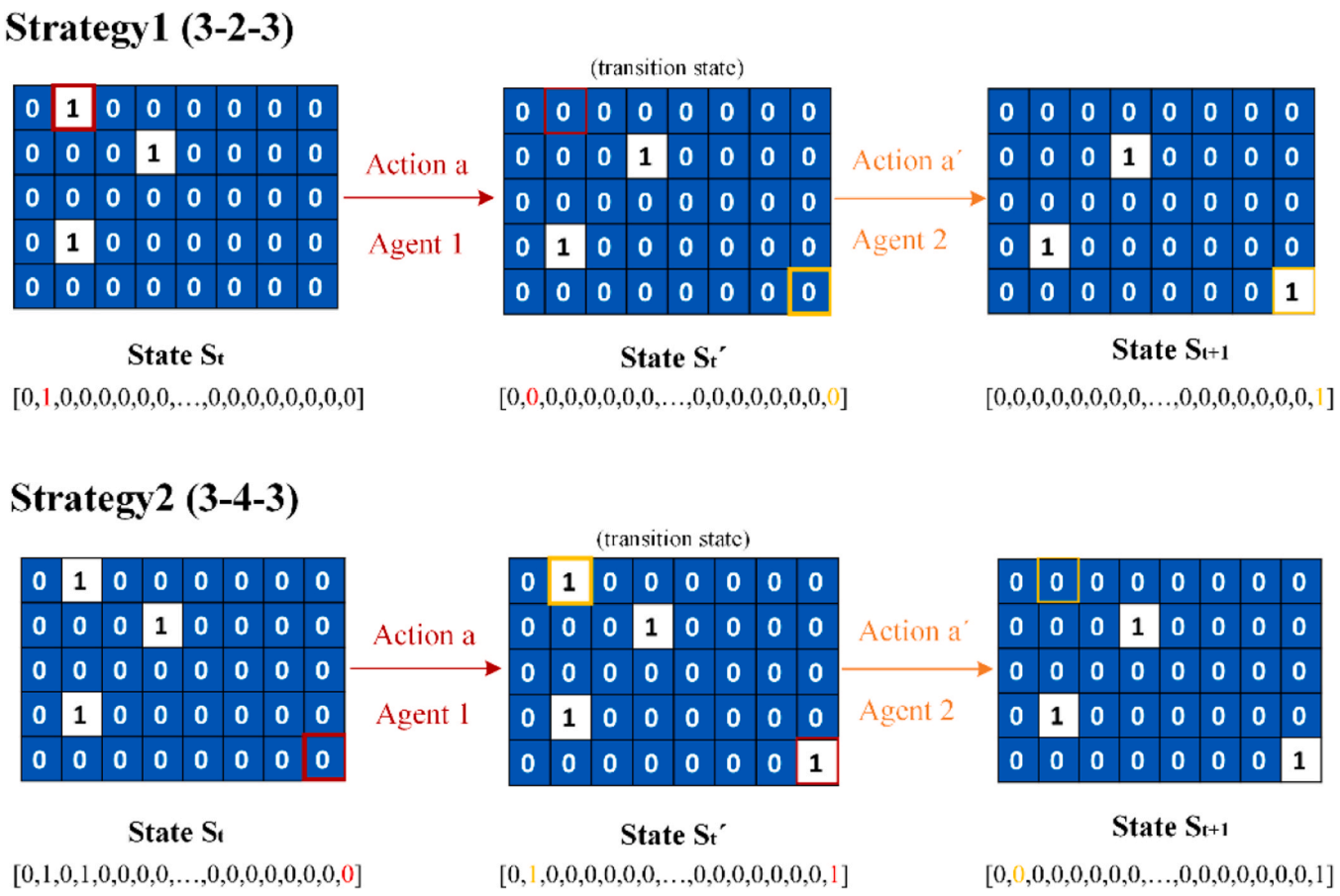


Fig. 7. A schematic representation of the updating process for optimization.

3.2.3. Reward function

The feedback the agent receives from the environment in response to its action is referred to as a reward. In this study, the reward function is based on the target properties. To optimize structures towards the higher peak load (i.e., higher strength), the change of the peak load before and after taking actions is defined as the reward function. Two actions are taken in a row by the two agents and the reward function when state transfers from s_t to s_t', s_t' to s_{t+1} are shown in Eqs. (12) and (13) respectively.

$$R_1\left(s_t^{'}|s_t,a\right) = PL_t^{'} - PL_t \tag{12}$$

$$R_1\left(s_{t+1}|s_t^{'},a^{'}\right) = PL_{t+1} - PL_t^{'} \tag{13}$$

Where PL_t, $PL_t^{'}$ and PL_{t+1} are the peak load of the concrete structure at state s_t (3 pores, before action a), state $s_t^{'}$ (2/4 pores, after action a) and state s_{t+1} (3 pores, after action $a^{'}$) respectively.

Similarly, the reward function is defined as the change of fracture energy before and after taking actions in order to optimize the concrete structure for higher self-healing capacity. The reward functions are shown in Eqs. (14) and (15) respectively.

$$R_2\left(s_t^{'}|s_t,a\right) = T_t^{'} - T_t \tag{14}$$

$$R_2\left(s_{t+1}|s_t^{'},a^{'}\right) = T_{t+1} - T_t^{'} \tag{15}$$

Where T_t, $T_t^{'}$ and $T_{t+1}$are the defined fracture energy of the concrete structure at state s_t (3 pores, before action a), state $s_t^{'}$ (2/4 pores, after action a) and state s_{t+1} (3 pores, after action $a^{'}$) respectively.

3.2.4. Horizon

Horizon defines how long the agent interacts with the environment. In this study, the concrete structure optimization is mapped into a finite MDP and is an episodic task. Therefore, we should determine when to stop the interaction. Here, two termination criterions are set to stop the interaction process: (1) the maximum number of interaction steps; (2) the target value equals or exceeds a threshold value.

For an episodic task, it is necessary to set a maximum number of steps for each episode. Considering the computational time for the visited structures, the number of interaction step is set as 200 after hyper-parameter tuning. However, the maximum number of steps is not sufficient since the RL agent is likely to miss the better structures during the interaction process. In addition, the starting state is randomly generated and the initial guess may be with high peak load or fracture energy. As a result, the interaction should be terminated in advance if a concrete structure with high target occurs. The historical highest value is set as the threshold value to judge whether an equally good or better structure has been encountered during the training process. The threshold value is initialized as 0 before the interaction, and it is updated with the highest peak load/ fracture energy during the interaction. To increase the exploration of agent, the historical maximum value is multiplied by a factor (<1). Since the computational errors for peak load and fracture energy are different, the factors for peak load and fracture energy are tuned separately.

3.2.5. Agent

The agent learns to choose the subsequent actions based on the current state in an attempt to achieve maximum reward. Two agents are created to make two sequential actions in a row to maintain a constant number of pores. Deep Q-network (DQN) is employed to map the state to the action-state pair. The number of neurons of the input and the output

layer is set to 40 since the concrete structure is represented with a 40-dimensional vector (Section 2.2.1). There are 2 hidden layers with 1024 and 512 hidden neurons, respectively. For the agent takes action to change 2-pore structures to 3-pore structures, one hidden layer is created with 1024 neurons considering the limited state ($C_{40}^{3} = 780$) after hyperparameter tuning. The output layer has 40 neurons so that any of the 40 locations can be selected. Similar to [32], the loss function and activation function are chosen as Huber loss and ReLu function for their favourable performance. Compared with other loss functions, Huber loss is more robust. As a commonly-used activation function, ReLu function gets rid of the vanishing gradient problem. Huber loss and ReLu function could be represented with Eqs. (16) and (17) respectively. The used DQN is schematically presented in Fig. 8.

$$L_{\delta}(a) = \begin{cases} \frac{1}{2}(y - f(x))^2 & for \quad |y - f(x)| \le \delta \\ \delta|y - f(x)| - \frac{1}{2}\delta^2 & otherwise \end{cases} \tag{16}$$

$$f(x) = \max(0, x) \tag{17}$$

where δ is the threshold at which to change between delta-scaled L1 and L2 loss.

During the learning process, the weights (θ) of the deep Q-network are updated instead of directly updating Q-value. The updating logics are given in Eqs. (18) and (19).

$$target(s^{'}) = R(s, a, s^{'}) + \gamma \max_{a^{'}} Q_{\theta_k}(s^{'}, a^{'}) \tag{18}$$

$$\theta_{k+1} \leftarrow \theta_k - \alpha \nabla_{\theta} [\frac{1}{2}(Q_{\theta}(s, a) - target(s^{'}))^2]|_{\theta=\theta_k} \tag{19}$$

Considering that function approximation with neural network faces possible instabilities or even divergence [39], two heuristics, i.e., experience replay and target Q-network, are employed to fix this problem. They could help to disconnect the data correlation and increase the learning efficiency. The replay buffer is 64, and the update target frequency is 10, and the batch-size is 16. Tensorflow framework is adopted for coding and the training process.

4. 4. Results and discussion

4.1. 3-pore concrete structure optimization for maximum peak load

To test the feasibility of vascular structure optimization using the RL approach, SHC with 3 pores are first enhanced aiming to mitigate the influence of pores on strength. The optimization process is independently run for multiple times to investigate the average behavior of the learning algorithm.

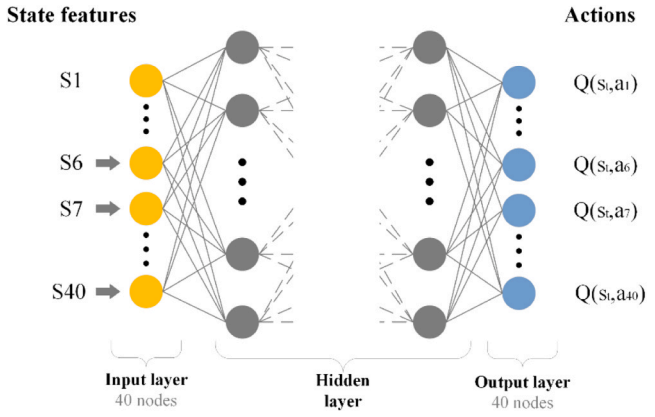


Fig. 8. Schematic representation of DQN.

4.1.1. The optimized structure and frequency

Considering that the goal is to find a concrete structure with the highest peak load, the historical maximum peak load is recorded during the interaction process. Both updating strategies are employed and compared in terms of the percentage of the best structures. The results are shown in Fig. 9a and b. In addition, the displacement-load curves of the optimized structures are also shown in Fig. 9c and d. It is noted that the subfigure in the upper right of Fig. 9c and d represent the corresponding structures, where the yellow circles are pores and the teal regions are cracks. This also applies to Fig. 13 in Section 4.2.3.

For both updating strategies, there are two optimal structures with peak loads of 785.3 N and 784.0 N (See Fig. 9a and b). During the 20 independent runs (10 runs for each updating strategy), the final two structures are encountered before the episode of 100 in most runs. In the whole design space, the maximum peak load of a 3-pore structure determined by the Abaqus model is 785.3 N, which has been identified in 16 out of the 20 optimization runs. The remaining 4 runs identify a structure with a peak load of 784.0 N, which is ranked as the second highest in the whole design space. As shown in Fig. 9b, the percentage of runs which successfully identified the best structure (peak load = 785.3 N) through updating strategy 1 (3→4→3) is higher than that of updating strategy 2.

When looking into the corresponding optimal structures, it is found that those two structures are symmetric (see Fig. 9c and d). The main crack does not hit pores in neither of the two structures, which explains why these structures have a high peak load. Note, however, that this is not optimal for self-healing: in these cases, no vascular networks would have been activated, and no self-healing would have been possible. In theory, the peak load of symmetric structures should be identical; the slight difference between the two structures (0.16 %) is a result of a simulation error. Therefore, it could be concluded that the optimization algorithm successfully identified the structure with the highest peak load in all 20 runs.

4.1.2. Number of visited structures during interaction

The total number of visited structures can reflect the number of interactions as well as the convergence of Q-networks. More importantly, more computation time is used if more unique structures are visited. Therefore, the number of visited 3-pore structures is recorded (Fig. 10).

As shown in Fig. 10a, the total number of visited 3-pore structures is dramatically different between the two updating strategies. Except for the random initialization of the start state, this may be also caused by the high probability of exploration since the exploration rate decays from 1 with a decay rate of 0.995 in each episode. Exploration enables the RL agent to get rid of suboptimal actions. The average numbers of total visited 3-pore structures with the updating strategy 1 and 2 are 4355 and 3078, which are much less than 20,000 (maximum number of visited 3-pore structures). In other words, the interaction process in most of the runs stops before reaching the maximum number of steps (i.e., 200 steps). This can also be verified by Fig. 10b, where the number of visited 3-pore structures in most episodes is less than 100. As a result, the small number of totally visited structure decreases the training time of RL agent.

Compared with the time spent on training Q-networks, the simulation time (i.e., running Abaqus models) accounts for most of the training time in this study. The simulation time increases only when new structures are encountered since the target properties of previous structures could be directly accessed. For a reinforcement learning task, it is common that one structure (including the best structure) is visited multiple times during the interaction process. Therefore, the number of unique visited structures is also recorded to look into the computation burden. According to Fig. 10(a), the number of visited unique 3-pore structures is much smaller than the total number of visited structures. The average numbers of visited unique structure with the two update strategies are 2143 and 1666 respectively. Therefore, the RL agent manages to find the best structure by visiting 22 % of the complete

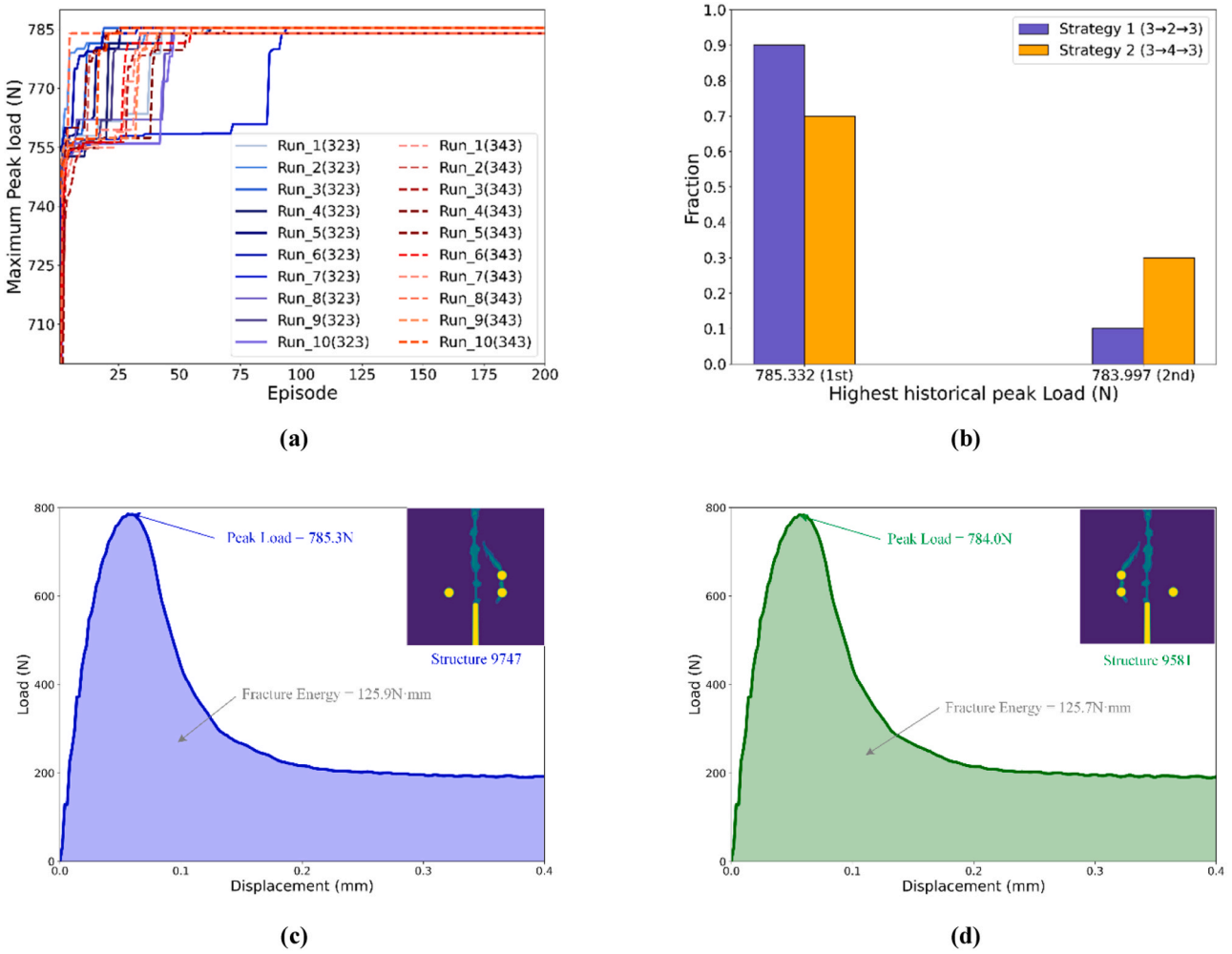


Fig. 9. Optimization result of 3-pore structure towards high peak load. (a) historical best structure during the interaction process; (b) Percentage of the best historical structures; (c) Structure with a peak load of 785.3 N; (d) Structure with a peak load of 784.0 N.

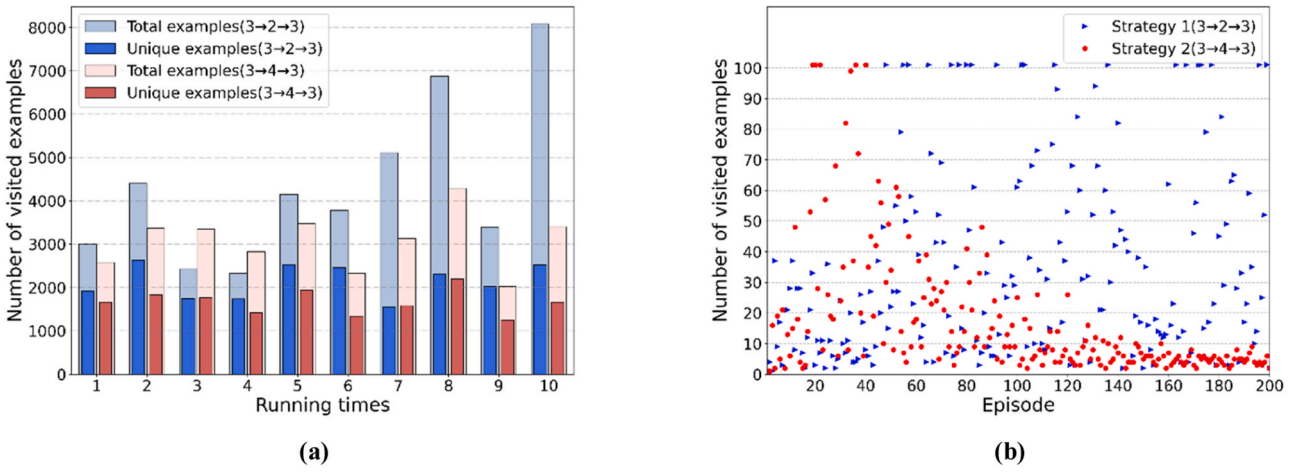


Fig. 10. Number of visited unique structures in (a) different running times; and (b) Number of visited examples in Run 8.

dataset. Compared with the updating strategy 1, the other strategy is more efficient since less computation power is needed. The advantage of an RL agent will be more pronounced as the design space increases.

4.1.3. Change of average reward

The increase in expected reward with experience is another important parameter to evaluate the performance of the learning algorithm. Due to the mechanisms of exploration (i.e., random selection of action) during the interaction process, the reward in a single run could significantly fluctuate. Therefore, 10 independent runs are repeated to measure the average behavior of the two updating strategies and the result is shown in Fig. 11.

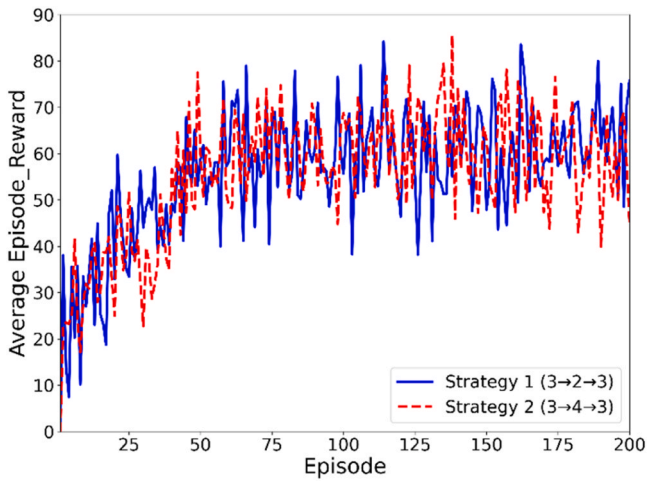


Fig. 11. Average reward of structure optimization for high peak load over 30 runs.

From Fig. 11, it can be seen that the average reward still fluctuates due to the noisy reward, which is set as the change of the peak load of 3-pore structure before/after the actions. For the two updating strategies, the average reward gradually improves before episode 70. Afterwards, the average reward levels off at the reward of about 60 N. Compared with the first terminal condition, the second terminal condition greatly influences the average reward. The target value of the historically best structure is relatively low at the beginning, where an equally good or better structure is encountered easily and the interaction ends up with a large reward at the beginning. However, the threshold value increases with an increasing number of visited structures. Although the RL agent is trained to take right actions to maximize the reward, it is possible that the interaction cannot find an equally good or better structure within 200 steps. As a result, the average reward gradually increases and then remains steady.

4.1.4. Loss function of the two Q-networks

To evaluate the performance of two RL agents, it is necessary to investigate the change of loss functions as training progresses. The loss functions of the two Q-networks in one run is shown in Fig. 12.

For updating strategy 1, the loss functions of the two Q-networks decrease as increase of episode even though they fluctuate during the training process (see Fig. 12a). Due to the limited number of visited examples (restricted by terminal criterion 2) at the beginning, the loss functions remain 0 for both two agents. This phenomenon can also be observed in updating strategy 2 (see Fig. 12b). Compared with Agent 1, the loss function of Agent 2 starts increasing at the last stage of training process. The possible reason is visiting the repetitive examples.

As to updating strategy 2, the loss functions of the two agents show similar trends during the training process. The loss functions gradually decrease before episode 50, then keep fluctuating until episode 130. A clear decrease can be seen around the episode 150. At the last stage of the training process, the loss functions of the two agents increase again. Compared with updating strategy 1, the two Q-networks are deeper and therefore more likely to suffer from overfitting with repetitive examples. Note, however, that the loss functions do not influence the search for the best structures according to the results in Section 4.1.1.

4.2. 3-pore concrete structure optimization for high fracture energy

4.2.1. Optimized structure and frequency

In Section 4.1 it is shown that the vascular concrete can be optimised for maximum peak load (i.e., strength). However, as stated, this is not optimal for the self-healing capacity. To design a vascular structure with more pores hit by the crack, in this section fracture energy is set as the target property. The results are shown in Fig. 13.

As shown in Fig. 13, two 3-pore structures with the fracture energy of 136.0 N·mm and 134.6 N·mm are found in the 20 runs. The performance of updating strategy 2 is better, since 70 % of the runs end with the structure with higher fracture energy. According to the whole dataset, the highest fracture energy of 3-pore structure from numerical simulation is 136.0 N·mm. However, considering that the two best structures are symmetric, it could be concluded that the best structure is obtained in all of the runs. Compared with peak load, the computation error of fracture energy for the two symmetric structures is larger (1.03 %). This is caused by the truncation error when calculating fracture energy (displacement = 0.4 mm). Compared with peak load optimization, the final two structures are encountered after the episode of 100 in most runs. Therefore, it is necessary to set the maximum step to be 200 when optimizing concrete structure towards higher fracture energy.

4.2.2. Number of visited structures during interaction

Similarly, the numbers of visited structures in the 20 runs are recorded to study the training time as well as the convergence. The results are shown in Fig. 14a. In addition, the number of visited structures in one episode is shown in Fig. 14b.

Compared with the peak load, the number of total visited structures

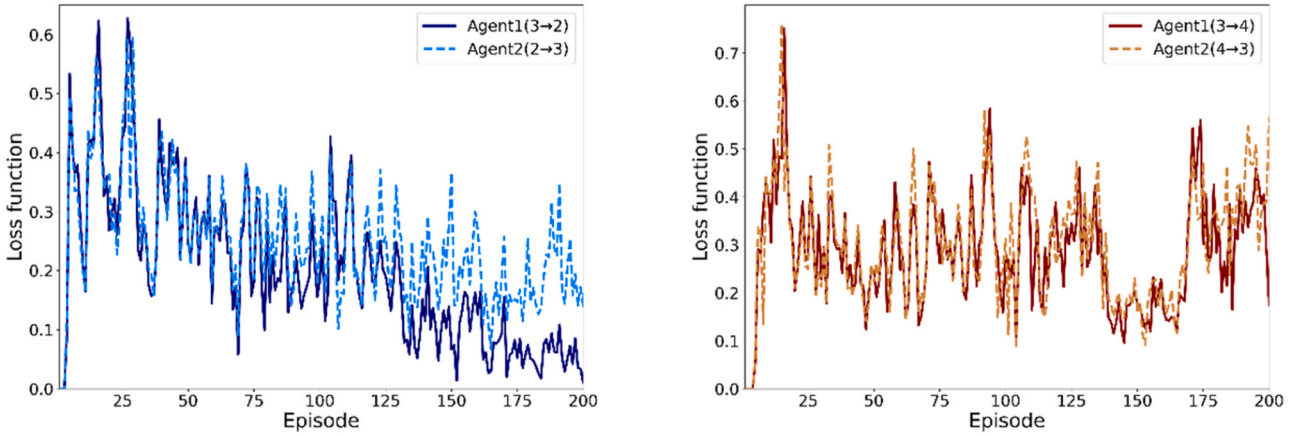


Fig. 12. Loss functions of two Q-networks of structure optimization for high peak load with (a) updating strategy 1; (b) updating strategy 2.

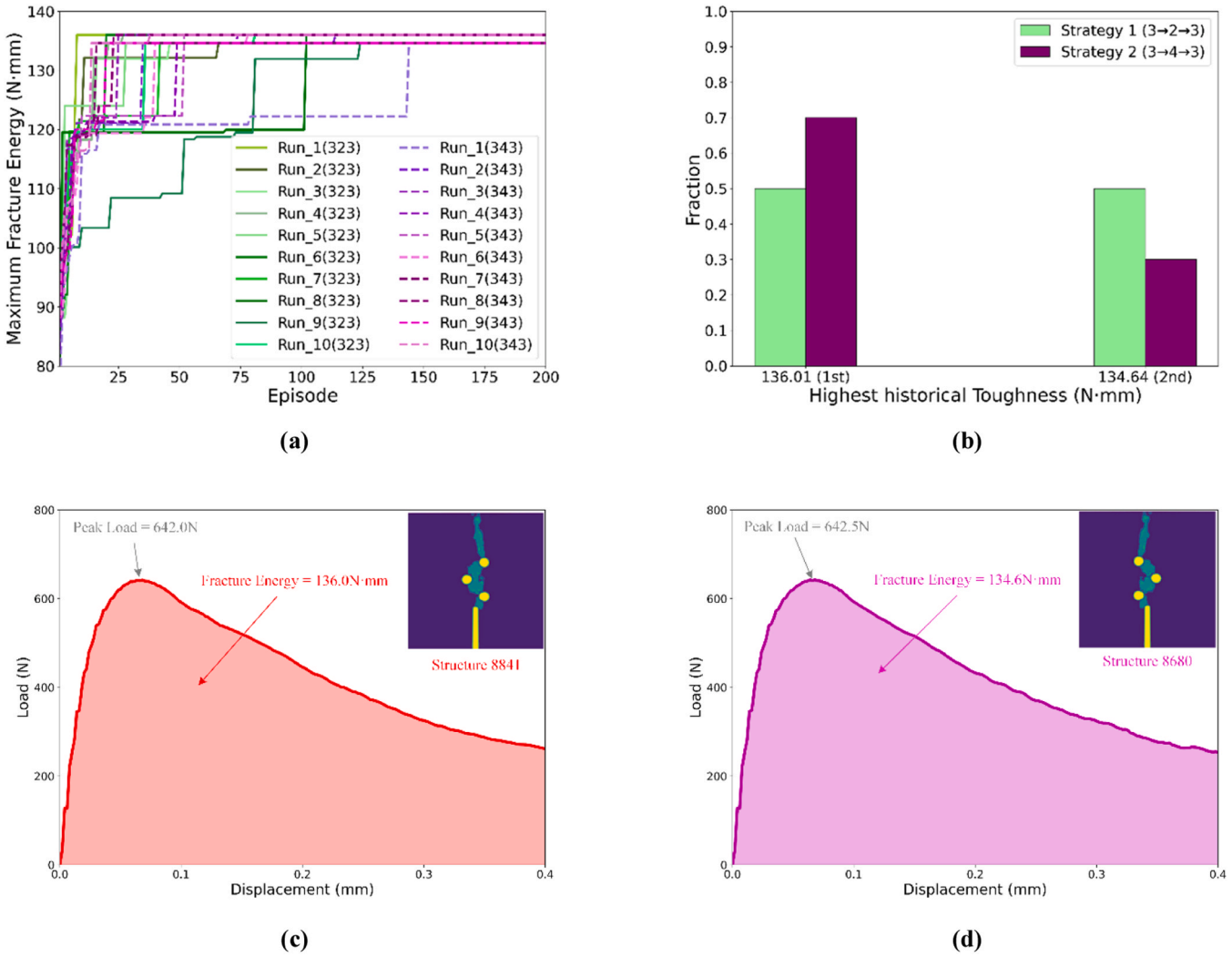


Fig. 13. Optimization result of 3-pore structure towards higher fracture energy. (a) historical best structure during the interaction process; (b) Frequency of the best historical structures; (c) Structure with fracture energy of 136.0 N·mm; (d) Structure with fracture energy of 134.6 N·mm.

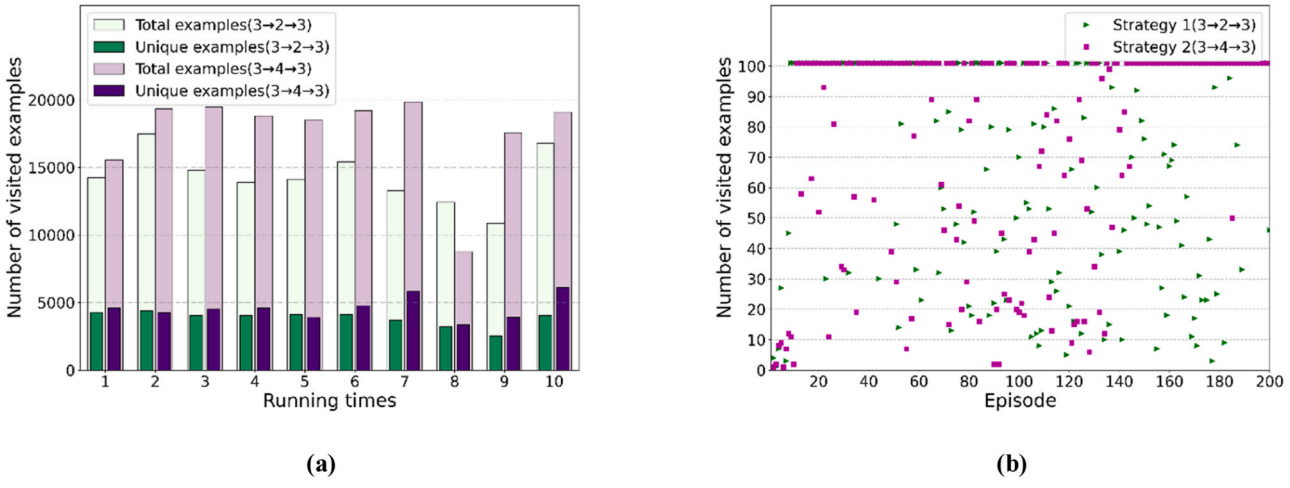


Fig. 14. Number of visited structure in (a) different running times; and (b) one episode (running 1).

is much larger, manifesting that it is more difficult for the RL agent to converge when optimizing concrete for high fracture energy. Furthermore, the large number of visited unique structures requires more computational time during the interaction process.

The performance of updating strategy 1 is better than that of updating strategy 2. For updating strategy 1 (3→2→3), the numbers of

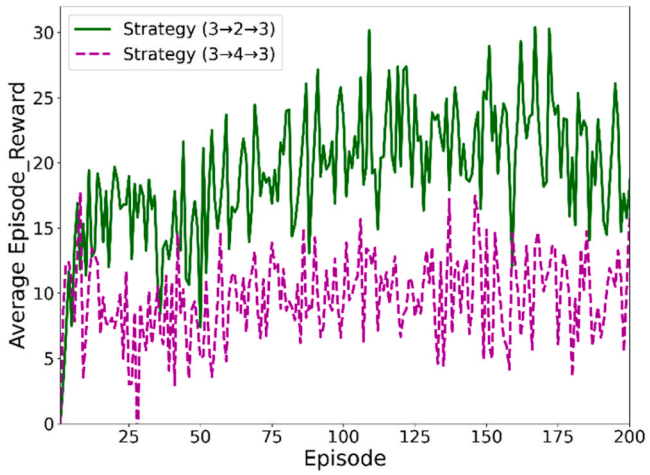


Fig. 15. Average reward of structure optimization for higher fracture energy over 20 runs.

total visited and unique 3-pore structures are 14,329 and 3843, respectively, which are lower than that of updating strategy 2 (17,609 and 4570, respectively). From Fig. 14b, it can be seen that the interactions in most of episodes is terminated by the maximum step criterion when using update strategy 2.

4.2.3. *Change of average episode-reward*

The average reward of the 20 independent runs for the two updating strategies is shown in Fig. 15.

From Fig. 15, the average reward also fluctuates for the two update strategies. The average reward of updating strategy 1 is significantly higher than that of updating strategy 2. For updating strategy 1, the average reward gradually improves at the beginning and then remains steady with an average reward of about 25 N·mm after episode 100. However, the average reward of updating strategy 2 sees a significant increase at the beginning, but it increases and remains unchanged in the following training process. The average reward of strategy 2 is less than 10 N·mm. Therefore, updating strategy 1 is recommended for optimizing the 4-pore structures for high self-healing capacity in the following part.

4.2.4. *Loss function of the two Q-networks*

Similarly, the loss functions of two Q-networks in one run is shown in Fig. 16.

According to Fig. 16, loss functions of the two Q-networks show different trends. The loss functions of Q-network turning matrix to pore (agent 2 for strategy 1, agent 1 for strategy 2) increases as the episodes pass. Compared with updating strategy 1, the loss function of agent 2 (4→3) dramatically decreases at the first 50 episodes and then gradually decreases in the remaining episodes. The loss function of agent 1 in updating strategy 1 first increases and then gradually deceases. The loss functions also prove that it is more difficult for the RL agent to converge when optimizing concrete for high fracture energy compared with optimization for high peak load.

5. Application to new optimization

Based on the results above, vascular configuration optimization towards good fracture energy through RL approach is performed for 4-pore concrete structures. The set of state is 91,380, which would make it time consuming to optimize by brute force. Considering that the performance of updating strategy 1 is better than updating strategy 2 in Section 4.2, the updating strategy for 4-pore structure optimization is chosen as 4→3→4. The two Q-networks consist of 2 hidden layers with 1024 and 512 neurons in each layer. The result of 4-pore structure optimization for fracture energy is shown in Fig. 17.

As shown in Fig. 17, the optimized 4-pore vascular structure has a fracture energy of 173.5 N·mm, which is higher than the highest fracture energy of 3-pores. When looking at the historical best structure in the optimization process (Fig. 17(a)), great improvement of fracture energy occurs at the episode 27, where 3 of the 4 pores are the same with the 3-pore structure with the highest fracture energy. In the following episodes, the 4th pore changes at the episodes of 51 and 136, and ends up with the position of 29. Compared with other structures, the 4th pore of the final optimized structure is the closest to the other 3 pores. From Fig. 17(b), it can be seen that damages occur around this pore. As a result, the fracture energy of the final structure is higher than the other structures. The number of the total visited structures and unique visited structures are 18,902 and 12,392 respectively. Considering that there are 91,390 combinations of 4-pore structures, the interaction visits 13.6 % of the complete dataset. Therefore, it is feasible to optimize the vascular structure of concrete using the RL approach.

Compared with generative deep neural network (GDNN), the recommended optimization is capable of automatically improve the structures without a pretrained ML model. In addition, the model in this study has better "generalization" property since it could be easily used for different pores by updating the simulation environment in Abaqus software. However, unlike supervised learning, it is difficult for reinforcement learning to make predictions for given similar inputs. In

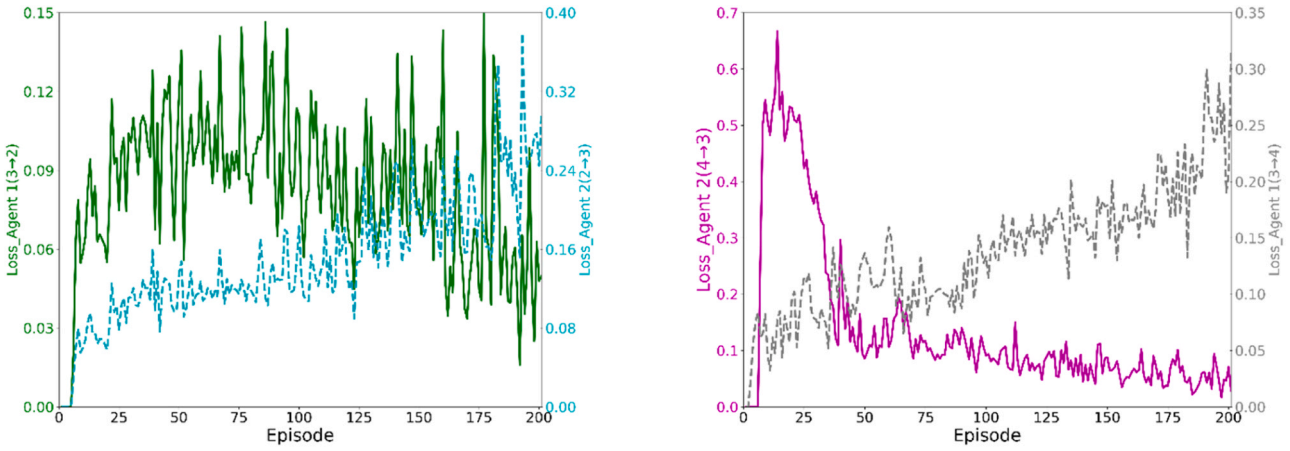


Fig. 16. Loss functions of two Q-networks of structure optimization for higher fracture energy. (a) updating strategy 1; (b) updating strategy 2.

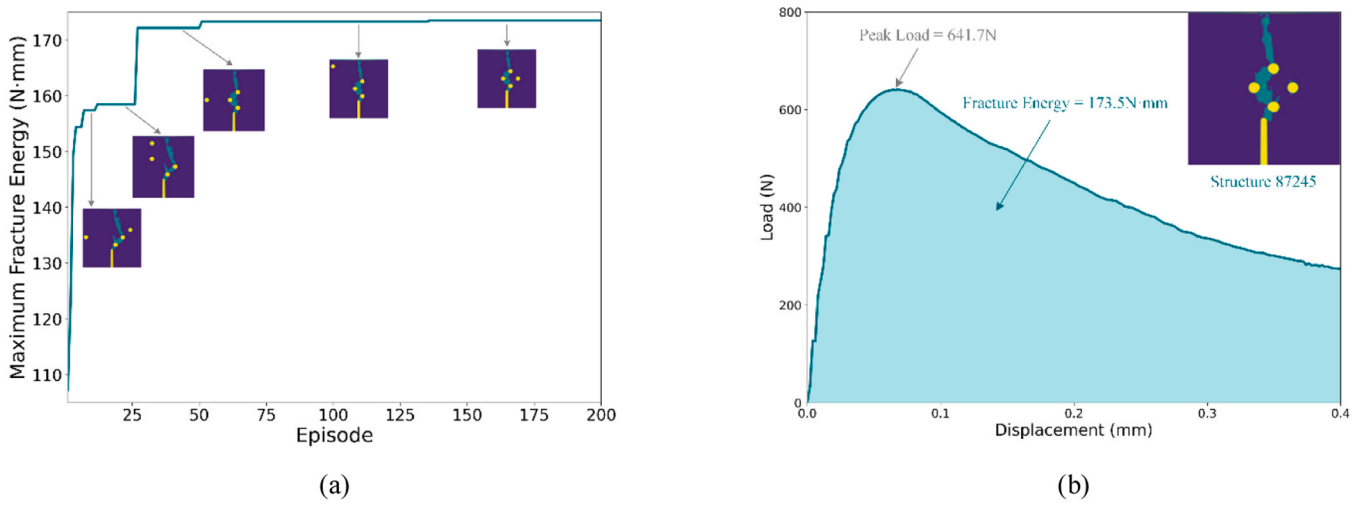


Fig. 17. Optimization result of 4-pore structure towards higher fracture energy. (a) historical best structure during the interaction process; (b) structure with highest fracture energy.

practice, it is of necessity to establish prediction models based on existing dataset (especially from experiments). Besides, an interactive environment is indispensable for reinforcement learning so as to calculate the corresponding reward after taking an action.

In future research, the optimization of vascular configuration should be extended towards more realistic scenarios, e.g., considering different loading conditions and the influence of structural parameters such as presence of steel reinforcement. In the current optimization approach, the vascular network was simplified as pores in the middle span of the 2D beam to reduce the computational burden. Clearly, the vascular system perpendicular to the crack is more appropriate for self-healing in the real structures. Instead of simple vascular system (i.e., four channels) in this study, a more complicated vascular system without vascular wall may help establishing the mapping relationship between the vascular configuration and the mechanical response without significantly increasing the computational time. Besides, the optimization of vascular self-healing concrete reinforced by rebars could be more complicated since the influence of vascular configuration on mechanical response will be much smaller compared with the influence from rebar. Furthermore, the vascular network is optimized when the specimens were under 3-point bending. Clearly, these cases are simple, and the methods must be extended before they could be useful for practice.

6. Conclusions

In this work, an automatic optimization method is proposed to arrange the vascular configuration of SHC through RL approach. A Markov Decision Process (MDP) is first formed and its elements are defined. To assess the viability of utilizing the method for vascular arrangement, SHC with 3 pores is first enhanced for higher peak load or fracture energy with two update strategies. Subsequently, vascular structure of a 4-pore concrete is optimized for good self-healing capacity by setting fracture energy as the target property. The main conclusions are as follows:

(1) The proposed method is capable of automatically optimizing the vascular structure of concrete towards different target properties through the interaction between RL agent and Abaqus simulation environment. The optimization process is influenced by the design constraint, target properties and updating strategies.
(2) Considering the symmetry of structures, the structure with highest peak load of 3-pore concrete structures is accessed in all 20 independent runs through two updating strategies. The change trends of average reward for the two updating strategies are similar during the training process. However, updating strategy 2 outperforms strategy 1 since the average number of unique visited structure is less, which saves computational time.
(3) The 3-pore structure with the highest fracture energy is also visited in all 20 independent runs. However, the loss functions of RL agents taking action to increase the pore number dramatically increase as the episode increases. For both updating strategies, the numbers of total visited and unique visited structures are much larger than those of the peak load optimization of 3-pore structures. Therefore, it is more difficult for the RL agent to converge when optimizing concrete for high fracture energy.
(4) When optimizing a 3-pore structure towards high fracture energy, the optimization performance of updating strategy 1 is better since the number of unique visited structures is smaller and it is time-efficient. Besides, the average reward of strategy 1 is higher.
(5) The RL optimization method is able to identify the structure with high fracture energy in the new optimization task for 4-pore concrete structure. As such, the method can be a powerful tool to automatically optimize structure towards higher target property.

CRediT authorship contribution statement

Zhi Wan: Writing – original draft, Visualization, Investigation, Data curation, Conceptualization. **Branko Šavija:** Writing – review & editing, Project administration. **Minfei Liang:** Writing – review & editing. **Ze Chang:** Writing – review & editing. **Yading Xu:** Writing – review & editing, Software, Data curation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

Zhi Wan, Ze Chang and Minfei Liang would like to acknowledge the financial support of the China Scholarship Council (CSC) under the grant

agreement No. 201906220205, No. 201806060129 and No. 202007000027. Yading Xu and Branko Šavija acknowledge the financial support of the European Research Council (ERC) within the framework of the ERC Starting Grant Project "Auxetic Cementitious Composites by 3D printing (ACC-3D)", Grant Agreement Number 101041342. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

Appendix A. Supporting information

Supplementary data associated with this article can be found in the online version at doi:10.1016/j.conbuildmat.2023.134592.

References

[1] Y. Shields, N. De Belie, A. Jefferson, K. Van Tittelboom, A review of vascular networks for self-healing applications, Smart Mater. Struct. 30 (2021), 063001, https://doi.org/10.1088/1361-665x/abf41d.
[2] B.J. Blaiszik, S.L.B. Kramer, S.C. Olugebefola, J.S. Moore, N.R. Sottos, S.R. White, Self-healing polymers and composites, Annu. Rev. Mater. Res. 40 (2010) 179–211, https://doi.org/10.1146/annurev-matsci-070909-104532.
[3] Y. Shields, N. De Belie, A. Jefferson, K. Van Tittelboom, A review of vascular networks for self-healing applications, Smart Mater. Struct. 30 (2021), 063001, https://doi.org/10.1088/1361-665X/ABF41D.
[4] Y. Shields, T. Van Mullem, N. De Belie, K. Van Tittelboom, An investigation of suitable healing agents for vascular-based self-healing in cementitious materials, Sustain 13 (2021) 12948, https://doi.org/10.3390/SU132312948.
[5] C.J. Hansen, S.R. White, N.R. Sottos, J.A. Lewis, Accelerated self-healing via ternary interpenetrating microvascular networks, Adv. Funct. Mater. 21 (2011) 4320–4326, https://doi.org/10.1002/adfm.201101553.
[6] A. Bejan, S. Lorente, K.M. Wang, Networks of channels for self-healing composite materials, J. Appl. Phys. 100 (2006), https://doi.org/10.1063/1.2218768.
[7] K.M. Wang, S. Lorente, A. Bejan, Vascularization with grids of channels: multiple scales, loops and body shapes, J. Phys. D. Appl. Phys. 40 (2007) 4740–4749, https://doi.org/10.1088/0022-3727/40/15/057.
[8] O. Yenigun, E. Cetkin, Experimental and numerical investigation of constructal vascular channels for self-cooling: parallel channels, tree-shaped and hybrid designs, Int J. Heat. Mass Transf. 103 (2016) 1155–1165, https://doi.org/10.1016/j.ijheatmasstransfer.2016.08.074.
[9] E. Cetkin, S. Lorente, A. Bejan, Hybrid grid and tree structures for cooling and mechanical strength, J. Appl. Phys. 110 (2011), https://doi.org/10.1063/1.3626062.
[10] A.R. Hamilton, N.R. Sottos, S.R. White, Self-healing of internal damage in synthetic vascular materials, Adv. Mater. 22 (2010) 5159–5163, https://doi.org/10.1002/adma.201002561.
[11] C.J. Hansen, W. Wu, K.S. Toohey, N.R. Sottos, S.R. White, J.A. Lewis, Self-healing materials with interpenetrating microvascular networks, Adv. Mater. 21 (2009) 4143–4147, https://doi.org/10.1002/adma.200900588.
[12] K.S. Toohey, N.R. Sottos, J.A. Lewis, J.S. Moore, S.R. White, Self-healing materials with microvascular networks, Nat. Mater. 6 (2007) 581–585, https://doi.org/10.1038/nmat1934.
[13] E. Tsangouri, C. Van Loo, Y. Shields, N. De Belie, K. Van Tittelboom, D.G. Aggelis, Reservoir-vascular tubes network for self-healing concrete: performance analysis by acoustic emission, digital image correlation and ultrasound velocity, Appl. Sci. 12 (2022) 4821, https://doi.org/10.3390/APP12104821.
[14] Minnebo, P. Thierens, G. De Valck, G. Van Tittelboom, K. Belie, N. De, D. Van Hemelrijck, et al., A novel design of autonomously healed concrete: towards a vascular healing network, Materials (Basel) 10 (2017) 49, https://doi.org/10.3390/ma10010049.
[15] Z. Wan, Y. Xu, Y. Zhang, S. He, B. Šavija, Mechanical properties and healing efficiency of 3D-printed ABS vascular based self-healing cementitious composite: experiments and modelling, Eng. Fract. Mech. 267 (2022), 108471, https://doi.org/10.1016/J.ENGFRACMECH.2022.108471.
[16] S. Soghrati, P.R. Thakre, S.R. White, N.R. Sottos, P.H. Geubelle, Computational modeling and design of actively-cooled microvascular materials, Int J. Heat Mass Transf. 55 (2012) 5309–5321, https://doi.org/10.1016/J.IJHEATMASSTRANSFER.2012.05.041.
[17] A.M. Aragón, R. Saksena, B.D. Kozola, P.H. Geubelle, K.T. Christensen, S.R. White, Multi-physics optimization of three-dimensional microvascular polymeric components, J. Comput. Phys. 233 (2013) 132–147, https://doi.org/10.1016/J.JCP.2012.07.036.
[18] A.R. Hamilton, N.R. Sottos, S.R. White, Local strain concentrations in a microvascular network, Proc. Soc. Exp. Mech. Inc. 67 (2010) 255–263, https://doi.org/10.1007/s11340-009-9299-5.
[19] Z. Li, L.R. de Souza, C. Litina, A.E. Markaki, A. Al-Tabbaa, A novel biomimetic design of a 3D vascular structure for self-healing in cementitious materials using Murray's law, Mater. Des. 190 (2020), 108572, https://doi.org/10.1016/j.matdes.2020.108572.
[20] A.M. Aragón, J.K. Wayer, P.H. Geubelle, D.E. Goldberg, S.R. White, Design of microvascular flow networks using multi-objective genetic algorithms, Comput. Methods Appl. Mech. Eng. 197 (2008) 4399–4410, https://doi.org/10.1016/j.cma.2008.05.025.
[21] Z. Chang, Z. Wan, Y. Xu, E. Schlangen, B. Šavija, Convolutional neural network for predicting crack pattern and stress-crack width curve of air-void structure in 3D printed concrete, Eng. Fract. Mech. 271 (2022), 108624, https://doi.org/10.1016/J.ENGFRACMECH.2022.108624.
[22] Z. Wan, Z. Chang, Y. Xu, Y. Huang, B. Šavija, Inverse design of digital materials using corrected generative deep neural network and generative deep convolutional neural network, Adv. Intell. Syst. (2022), 2200333, https://doi.org/10.1002/AISY.202200333.
[23] C.T. Chen, G.X. Gu, Generative deep neural networks for inverse materials design using backpropagation and active learning, Adv. Sci. 7 (2020), 1902607, https://doi.org/10.1002/advs.201902607.
[24] Z. Wan, Z. Chang, Y. Xu, B. Šavija, Optimization of vascular structure of self-healing concrete using deep neural network (DNN), Constr. Build. Mater. 364 (2023), 129955, https://doi.org/10.1016/J.CONBUILDMAT.2022.129955.
[25] R.R. Torrado, P. Bontrager, J. Togelius, J. Liu, D. Perez-Liebana, Deep reinforcement learning for general video game AI, IEEE Conf. Comput. Intell. Games, CIG (2018), https://doi.org/10.1109/CIG.2018.8490422.
[26] Mnih V., Kavukcuoglu K., Silver D., Graves A., Antonoglou I., Wierstra D., et al. Playing Atari with Deep Reinforcement Learning n.d.
[27] W. Zhang, J. Gai, Z. Zhang, L. Tang, Q. Liao, Y. Ding, Double-DQN based path smoothing and tracking control method for robotic vehicle navigation, Comput. Electron Agric. 166 (2019), 104985, https://doi.org/10.1016/J.COMPAG.2019.104985.
[28] R.S. Sutton, A.G. Barto, Reinforcement Learning: an Introduction, MIT Press, 2018.
[29] T. Kim, Y.W. Kim, D. Lee, M. Kim, Reinforcement learning approach to scheduling of precast concrete production, J. Clean. Prod. 336 (2022), 130419, https://doi.org/10.1016/J.JCLEPRO.2022.130419.
[30] Jeong Hongki Jo J.-H., Hongki Jo C. Deep reinforcement learning for automated design of reinforced concrete structures 2021. https://doi.org/10.1111/mice.12773.
[31] N.K. Brown, A.P. Garland, G.M. Fadel, G. Li, Deep reinforcement learning for engineering design through topology optimization of elementally discretized design domains, Mater. Des. 218 (2022), 110672, https://doi.org/10.1016/J.MATDES.2022.110672.
[32] F. Sui, R. Guo, Z. Zhang, G.X. Gu, L. Lin, Deep reinforcement learning for digital materials design, ACS Mater. Lett. (2021) 1433–1439, https://doi.org/10.1021/acsmaterialslett.1c00390.
[33] C. Qiu, S. Du, J. Yang, A deep learning approach for efficient topology optimization based on the element removal strategy, Mater. Des. 212 (2021), 110179, https://doi.org/10.1016/J.MATDES.2021.110179.
[34] S. V.Chaudhari, M. A. Chakrabarti, Modeling of concrete for nonlinear analysis using finite element code ABAQUS, Int J. Comput. Appl. 44 (2012) 14–18, https://doi.org/10.5120/6274-8437.
[35] Y. Xu, H. Zhang, Y. Gan, B. Šavija, Cementitious composites reinforced with 3D printed functionally graded polymeric lattice structures: experiments and modelling, Addit. Manuf. 39 (2021), 101887, https://doi.org/10.1016/j.addma.2021.101887.
[36] J. ZHAO, J. HUO, Coordination mechanism combining supply chain optimization and rule in exchange, Asia-Pac. J. Oper. Res. 30 (2013) 1350015, https://doi.org/10.1142/S0217595913500152.
[37] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, et al., Human-level control through deep reinforcement learning, Nature 518 (2015) 529–533, https://doi.org/10.1038/nature14236.
[38] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, P. Abbeel, Overcoming exploration in reinforcement learning with demonstrations, IEEE Int. Conf. Robot. Autom. 2018 (2018) 6292–6299, https://doi.org/10.1109/ICRA.2018.8463162.
[39] J.N. Tsitsiklis, B. Van Roy, An analysis of temporal-difference learning with function approximation, IEEE Trans. Autom. Cont. 42 (1997) 674–690, https://doi.org/10.1109/9.580874.