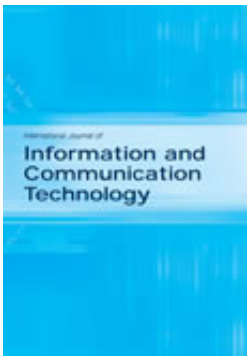
INDERSCIENCE PUBLISHERS
Linking academia, business and industry through research



International Journal of Information and Communication Technology

ISSN online: 1741-8070 - ISSN print: 1466-6642
https://www.inderscience.com/ijict

A multi-agent reinforcement learning approach to heterogeneous resource allocation in lifelong education

Miao Wang, Zhen Gu

DOI: 10.1504/IJICT.2026.10075957

Article History:

Received:	29 September 2025
Last revised:	29 October 2025
Accepted:	30 October 2025
Published online:	06 February 2026

A multi-agent reinforcement learning approach to heterogeneous resource allocation in lifelong education

Miao Wang*

Department of Scientific Research and Planning,
The Open University of Jilin,
Changchun, 130022, China
Email: wangmiao@jlrtvu.cn
*Corresponding author

Zhen Gu

College of Ethnic Preparatory Education,
Jilin Provincial Institute of Education,
Changchun, 130022, China
Email: guyuesheng@foxmail.com

Abstract: Lifelong education faces challenges in resource allocation due to heterogeneity, dynamism, and scalability. This study proposes a distributed allocation model using multi-agent reinforcement learning (MARL), where learners and providers act as autonomous agents. Employing a centralised training with decentralised execution (CTDE) paradigm, the model applies the multi-agent deep deterministic policy gradient algorithm for collaborative learning. A composite reward function integrates user quality of experience (QoE), system cost, and fairness. A tripartite stochastic game model theoretically confirms the existence of a Nash equilibrium. Simulations show the model outperforms baseline algorithms, achieving superior overall system utility (37.1% higher), average quality of experience (41.8% higher), resource utilisation (76.3%), and fairness (Jain's index = 0.89), with strong convergence and adaptability. This provides an efficient, scalable solution for heterogeneous resource management in lifelong education.

Keywords: multi-agent reinforcement learning; MARL; resource allocation; lifelong education; heterogeneous networks; stochastic games.

Reference to this paper should be made as follows: Wang, M. and Gu, Z. (2026) 'A multi-agent reinforcement learning approach to heterogeneous resource allocation in lifelong education', *Int. J. Information and Communication Technology*, Vol. 27, No. 7, pp.21–37.

Biographical notes: Miao Wang is an Associate Professor in the Department of Scientific Research and Planning at Open University of Jilin. She received a Master's degree from Northeast Normal University in 2010. Her research interests include lifelong education and open education.

Zhen Gu is an Associate Professor in College of Ethnic Preparatory Education at Jilin Institute of Education. He received a Master's degree from Northeast Normal University in 2009. His research interests include educational technology, artificial intelligence and teaching, and computer education.

1 Introduction

Lifelong education, as a core paradigm driving continuous individual development, has seen its significance increasingly highlighted through widespread applications in online education platforms and personalised learning. However, lifelong education environments exhibit pronounced resource heterogeneity, manifested in diverse resource types, complex computational resource architectures, and varied user demands. Traditional centralised resource allocation models suffer from inherent limitations in scalability, privacy protection, and dynamic adaptability, making them ill-suited to meet the demands for efficient resource distribution and personalised services in lifelong education scenarios. Consequently, the adoption of distributed solutions is essential (Wenjuan and Xin, 2024).

In recent years, with the rise of edge computing, researchers have begun exploring how to leverage the proximity computing capabilities of edge servers to reduce latency in educational services. To overcome the shortcomings of centralised approaches, distributed intelligent decision-making has emerged as a significant research direction. Multi-agent systems (MAS) and game theory provide a natural theoretical framework for addressing distributed resource allocation. This paper proposes integrating multi-source data with large language models in MAS, achieving efficient information retrieval and question-answering through dynamic retrieval strategies and multi-source question-answering systems (Antony et al., 2024). This paper employs an asymptotic compression cooperative adjustment method in MAS to address the control problem of nonlinear time-varying systems in multidimensional space (Wang and Liu, 2024). This paper proposes a self-organising approach that integrates Monte Carlo Tree Search with self-organising MAS to address residential floor plan layout problems through adaptive methods (Su et al., 2024). This paper proposes a multi-agent system model detection method based on a fuzzy explanation system to achieve knowledge attribute verification (Ma et al., 2024). This paper proposes a resource allocation and application deployment scheme based on collaborative overselling to optimise resource utilisation and deployment costs for drone edge computing in forestry IoT (Li et al., 2024). Fatemeh employs the Cheetah algorithm to optimise resource allocation in fog computing, reducing latency and enhancing system efficiency (Arvaneh et al., 2024).

Heterogeneous networks and stochastic games are also widely used in models for allocating heterogeneous resources in lifelong education. Zhao's GAN-based heterogeneous network achieves high-precision automatic restoration of ancient mural textures and colours (Zhao et al., 2024). This paper proposes a drug repurposing approach for predicting drug-disease associations based on a dual-view fusion mechanism and graph augmentation technique for heterogeneous networks (Niu et al., 2024). The article employs graph embedding and negative sample filtering based on heterogeneous networks to construct a random forest model for predicting disease-protein associations (Wang et al., 2025). This study employs a high-order heterogeneous network to enhance features and aggregate neighbourhood information through multivariate feature learning and a hyperbolic graph attention network, thereby achieving more precise drug-disease prediction (Li et al., 2025). This paper presents a formal security modelling and rigorous analysis of the RabbitMQ broker based on concurrent stochastic games (Baouya et al., 2024).

To address the aforementioned challenges, this paper proposes a distributed heterogeneous resource allocation model based on multi-agent reinforcement learning (MARL). Its main contributions can be summarised in four points:

1 This paper introduces a novel distributed multi-agent resource allocation framework. This framework models learners and resource providers as autonomous decision-making agents. Through local perception and mutual communication, these agents engage in collaborative decision-making, completely avoiding the bottlenecks inherent in centralised control. This approach inherently possesses excellent scalability, robustness, and privacy protection characteristics.

2 Innovatively integrates user quality of experience (QoE), system cost, and collective fairness into a unified reward function. This guides agents to spontaneously advance overall system objectives while pursuing individual benefits.

3 This model formally describes the interactions among learners, resource providers, and the network environment, proving the existence of a Nash equilibrium. This provides theoretical assurance for the convergence of the distributed algorithm, enhancing the theoretical depth of the proposed solution.

4 Comprehensive and in-depth simulation experiments were conducted for validation. Experimental results demonstrate that the proposed model exhibits superior effectiveness and advanced performance compared to multiple mainstream baseline algorithms.

2 Multi-agent modelling framework

2.1 System architecture and agent interaction

The system architecture represents a typical three-tier 'cloud-edge-end' converged architecture designed to deliver on-demand, low-latency, high-quality educational services to a massive population of lifelong learners. Edge servers are assumed to be geographically distributed, such as at base stations, roadside units, or local school data centres, positioning them close to end users to deliver low-latency services.

The cloud centre possesses robust computing and storage capabilities, hosting large-scale non-real-time, computationally intensive educational services such as training massive deep learning models, analysing vast educational datasets, and maintaining ultra-high-definition video repositories. The edge network comprises geographically distributed edge servers, including base stations, roadside units, and school local servers. Proximal to users, these nodes handle latency-sensitive real-time or near-real-time tasks such as VR/AR interactive instruction, real-time video transcoding, and online Q&A sessions. User terminals encompass devices used by diverse learners – smartphones, tablets, laptops, etc. – which generate learning task requests and receive processing results.

Within this architecture, the core resource allocation decision lies in dynamically offloading each task to the most suitable node based on user task requirements, current network conditions, and resource availability across cloud and edge nodes. This process allocates appropriate computational, bandwidth, and storage resources to each task.

To achieve distributed intelligent decision-making, a multi-agent paradigm is introduced. Core participants in the system are abstracted into two types of agents: user agents (UA) are proxy agents assigned to each learner, residing on their terminal devices. Their objective is to secure optimal resources for user-initiated tasks to maximise QoE. Resource agents (RA) are assigned to each edge server and cloud data centre. Their objective is to maximise their own utilisation and revenue while avoiding overload by managing local resources and selling surplus capacity. Agents exchange only limited and essential information via communication networks, rather than reporting all data to a central controller. This design ensures the system's scalability, robustness, and privacy protection capabilities.

2.2 Heterogeneous resource model

Resources within lifelong learning environments exhibit multidimensional heterogeneous characteristics. To enable unified quantitative management and allocation, formal definitions of various resource types are required. First, a unified approach is needed to describe the resource capabilities possessed by resource nodes.

$$R_j = \left(C_j, S_j, B_j^{\text{up}}, B_j^{\text{down}}\right), \quad \forall j \in \{1, 2, ..., M\} \tag{1.1}$$

where C_j denotes the set of available computational resources for node j. S_j denotes the available storage space for node j. B_j^{up} denotes the upstream network bandwidth for node j. B_j^{down} denotes the downstream network bandwidth for node j.

This equation constructs a universal resource description framework, unifying heterogeneous computing, storage, and communication resources into a single mathematical expression. This formal representation enables algorithms to simultaneously handle different types of resource constraints, establishing a quantitative foundation for subsequent resource allocation optimisation problems. It ensures the consistency and computability of resource management.

In distributed computing or resource scheduling systems, when tasks are assigned to resource nodes for execution, they inevitably consume computational power, storage capacity, network bandwidth, and other resources provided by the nodes. To quantify a task's resource consumption requirements, its resource demands must be formally defined to support subsequent scheduling strategy design and performance analysis:

$$D_i = \left(c_i, s_i, b_i^{\text{in}}, b_i^{\text{out}}\right) \tag{1.2}$$

where c_i represents the computational requirements of task i. s_i represents the storage requirements occupied by task i. b_i^{in} represents the input bandwidth requirements of task i. b_i^{out} represents the output bandwidth requirements of task i.

This equation transforms the abstract concept of 'task resource requirements' into a quantifiable four-dimensional vector. This enables the system to precisely describe task resource consumption using mathematical tools such as linear programming and constraint satisfaction problems, thereby providing a formal foundation for subsequent scheduling strategy design.

2.3 User demand model

In agent-driven task execution scenarios, the core objective of user demands has shifted from single-resource allocation to dual expectations quality of service (QoS) and QoE. The primary task of UA is to maximise user satisfaction with task execution outcomes – i.e., maximise QoE – by optimising decision strategies. User-initiated learning tasks can be formally characterised as tuples:

$$Task_k = (Type_k, D_k, QoS_k, Bueget_k) \tag{1.3}$$

where $Task_k$ represents the learning task initiated by user k. $Type_k$ denotes the task type. D_k is resource requirement. QoS_k is the set of QoS metrics expected by the user. $Bueget_k$ is the maximum virtual cost or points the user is willing to pay for this task.

To translate abstract QoS metrics into quantifiable user satisfaction, this paper introduces the QoE function. This function maps QoS metrics to standardised satisfaction scores, typically real numbers between 0 and 1. Its core purpose is to comprehensively evaluate the actual benefits users derive from task execution. To accurately reflect user utility, it is necessary to define the mapping relationship between key QoS parameters and user utility. Considering the law of diminishing marginal utility, the QoE function is typically designed in a nonlinear form:

$$QoE_k = \alpha \cdot \log\left(\frac{Latency_k^{\max}}{Latency_k}\right) - \beta \cdot Cost_k \tag{1.4}$$

where QoE_k is the experience quality of task k. α is the delay weighting coefficient. $Latency_k^{\max}$ is the maximum tolerable delay for task k by user. $Latency_k$ is the actual execution delay of task k. β is the cost weighting coefficient. $Cost_k$ is the actual execution cost of task k. Weighting coefficients latency weight α and cost weight β are preset based on task type and user preferences.

This equation transforms the abstract concept of user experience into a computable mathematical function. It provides the core objective function component for subsequently converting resource allocation problems into optimisation problems. The agent's goal is to maximise this QoE value through its decision-making actions.

2.4 Multi-agent game model

Within this system, UA and RA function as autonomous and self-interested decision-making entities, with their behavioural logic adhering to a game theory framework. The core objective of UAs is to minimise costs while maximising QoE, whereas RAs strive to maximise resource utilisation and economic returns. Given the time-varying nature of environmental states and the long-term impact of agent decisions on the system, a stochastic game serves as the theoretical modelling framework. This framework is fundamentally a Markov Game, characterised by state transition probabilities that depend solely on the current state and the collective actions of agents – aligning with the modelling requirements of complex dynamic systems (Maksymov et al., 2024). To rigorously characterise the resource allocation process in multi-agent interactions, the multi-agent stochastic game is defined as a six-tuple structure:

$$\mathcal{G} = \left\langle \mathcal{N}, \mathcal{S}, \{A_i\}_{i\in\mathcal{N}}, P, \{R_i\}_{i\in\mathcal{N}}, \gamma \right\rangle \tag{1.5}$$

where $\mathcal{G}$ is a multi-agent stochastic game model. $\mathcal{N}$ is the set of agents. $\mathcal{S}$ is the global environmental state space. $\{A_i\}_{i\in\mathcal{N}}$ is the action space for agent *i*. *P* is the state transition probability function. $\{R_i\}_{i\in\mathcal{N}}$ is the immediate reward function for agent *i*. γ is the discount factor.

The stochastic game model possesses a Nash equilibrium, The theoretical prerequisites for equilibrium existence include: the agents' policy space is continuous, the reward function is bounded, and the state transition process satisfies Markovian properties. The equation-based random game framework provides a rigorous mathematical foundation for multi-agent interactions. Its six-element structure clearly defines the core elements of system modelling: agents, states, actions, transition probabilities, rewards, and discount factors. This establishes a formal foundation for subsequent reinforcement learning algorithms to solve multi-agent decision problems. The framework's universality enables adaptation to diverse scenarios, such as wireless network resource allocation and cloud computing task scheduling. Its Markovian property ensures predictability in state transitions, providing theoretical support for algorithm convergence analysis.

Within multi-agent cooperative systems, constructing a quantifiable decision framework requires formal definitions of each agent's state, action, and reward to support subsequent reinforcement learning model training and optimisation (Meng et al., 2024).

Agent state represents the local observed state of agent *i* at time slot *t*. For UA, the state includes their own task information, observed network status, and received resource offers; for RA, the state includes their current available resources and received task request information. For the RA, its state includes its current available resources and received task request information.

Agent action is the action performed by agent *i* in time slot *t*. The RA's action is to set a price vector for its resources and decide which task requests to accept or reject:

$$a_{ra}^{t} = \left(Price_{\text{compute}}^{t}, Price_{\text{storage}}^{t}, Price_{\text{bw}}^{t}, \text{Accept/Reject} \right) \tag{1.6}$$

where a_{ra}^{t} denotes the action taken by resource agent *ra* in time slot *t*. $Price_{\text{compute}}^{t}$, $Price_{\text{storage}}^{t}$, and $Price_{\text{bw}}^{t}$ represent *ra*'s pricing for computational, storage, and bandwidth resources in time slot *t*, respectively. Accept/Reject indicates *ra*'s decision to accept or reject the task request.

This equation translates the complex decision-making of the RA into a computable numerical vector, enabling reinforcement learning algorithms to learn optimal strategies through the 'state-action-reward' feedback loop. It also facilitates the equation of differentiated strategies based on the supply-demand dynamics of various resources, thereby enhancing resource allocation efficiency.

$$r_{ua}^{t} = \text{QoE}_{k}^{t} \tag{1.7}$$

$$r_{ra}^{t} = \lambda \cdot Utilisation_{j}^{t} + (1-\lambda) \cdot Revenue_{j}^{t} \tag{1.8}$$

where QoE_k^t represents the QoS experience for *ua* in time slot *t*. λ is the weighting coefficient balancing resource utilisation and revenue. $Utilisation_j^t$ denotes the comprehensive resource utilisation rate of resource node *j* in time slot *t*. $Revenue_j^t$ represents the total revenue earned by resource node *j* in time slot *t*.

The reward function serves as the bridge connecting the game model with reinforcement learning. These two equations quantify the agent's higher-level objectives into computable scalar signals at each step, enabling the agent to learn how to achieve its long-term goals by maximising cumulative rewards. This constitutes the core incentive mechanism driving the entire multi-agent learning process.

2.5 Problem equation

The ultimate goal of this study is to identify an optimal distributed strategy enabling all agents to achieve an efficient and stable state of the system when adhering to these strategies. Each strategy constitutes a probability mapping from state to action for agent *i*. From the individual agent's perspective, the optimisation objective for agent *i* is to maximise its long-term expected cumulative discounted reward, formally defined as:

$$J_i(\pi_i) = \mathbb{E}_{\pi_i, \pi_{-i}}\left[\sum_{t=0}^{\infty} \gamma^t r_i^t\left(s^t, a_i^t, a_{-i}^t\right)\right] \tag{1.9}$$

where π_i denotes the policy of agent *i*. π_{-i} denotes the combined policies of all other agents except agent *i*. γ denotes the discount factor. $r_i^t(s^t, a_i^t, a_{-i}^t)$ denotes the instantaneous reward when agent *i* takes action a_i^t at time *t* in state s^t, and all other agents take action a_{-i}^t.

The equation explicitly captures the long-term nature of individual decisions (considering all future rewards) and balances present and future gains through a discount factor. This serves as the standard objective function for applying reinforcement learning to such sequential decision problems.

However, agents optimising this equation independently may lead to suboptimal overall system performance. Therefore, from a global system perspective, it is more crucial to find a combination of policies that maximises system welfare. System welfare is the weighted sum of all agents' long-term rewards, while also incorporating considerations of fairness:

$$\max_{\pi_1,\ldots,\pi_N} U_{\text{total}} = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \left(\sum_{i \in \mathcal{N}_A} \omega_i r_i^t\left(s^t, a_i^t, a_{-i}^t\right) + \sum_{j \in \mathcal{N}_R} \omega_j r_j^t\left(s^t, a_j^t, a_{-j}^t\right)\right)\right] \tag{1.10}$$

where U_{total} denotes the overall system utility. $\mathcal{N}_A$ denotes the set of UA. $\mathcal{N}_R$ denotes the set of RA. ω_i and ω_j denote weighting coefficients

To maximise the overall utility of the system, the following two types of constraints must be satisfied: resource capacity constraints and QoS constraints:

$$\sum_{i:a_i^t = j} D_i \le R_j, \quad \forall j, t \tag{1.11}$$

$$Latency_k \leq Latency_k^{\max}, \quad \forall k \tag{1.12}$$

where D_i denotes the resource demand of agent i. R_j denotes the available resource capacity of resource node j. $Latency_k$ denotes the user's end-to-end latency. $Latency_k^{\max}$ denotes the user's maximum tolerable latency. k denotes the user index.

These three equations collectively formalise the real-world problem into a constrained multi-objective sequential decision optimisation problem. This formalised problem provides a clear, quantifiable benchmark for evaluating the performance of any resource allocation algorithm, enabling the learned strategy to approximate this global optimum solution.

3 A MARL-based resource allocation algorithm

Based on a stochastic game model, this chapter proposes a distributed resource allocation algorithm grounded in MARL. This algorithm enables agents to autonomously learn optimal strategies through interaction with the environment, thereby effectively addressing heterogeneous resource allocation challenges in lifelong education scenarios.

3.1 CTDE paradigm and MADDPG framework

To address this issue, this paper adopts the paradigm of centralised training with decentralised execution (CTDE). The core idea of this paradigm is as follows: during the training phase, a central critic network capable of acquiring global information is used to guide the training of individual actor networks; while in the execution phase, each agent can make decentralised decisions relying solely on its own local observations (Wang et al., 2024).

The CTDE paradigm skilfully balances the contradiction between training stability and execution efficiency. It not only mitigates the impact of environmental non-stationarity on training through global information but also retains the advantages of low latency and high scalability of decentralised execution. The CTDE paradigm leverages global information during training to enhance strategy stability. During execution, each agent makes independent decisions based solely on local information. This approach balances learning effectiveness with system scalability, low latency, and privacy protection.

Specifically, the multi-agent deep deterministic policy gradient (MADDPG) algorithm is employed as the basic framework. MADDPG is an extension of the DDPG algorithm in multi-agent scenarios, and its core advantage lies in the fact that the central critic can know the actions of all agents during training. This converts the uncertainty of the environment into determinism, providing each actor with more stable and accurate gradient signals, which greatly promotes the convergence of the policy.

3.2 Agent design

Under the MADDPG framework, each agent (UA or RA) is equipped with a pair of actor-critic networks. Both the actor and critic networks employ a fully connected neural network structure with three hidden layers. Each layer contains 256 neurons using ReLU

as the activation function, while the output layer utilises either Tanh or linear activation. The design of these agents requires clear definitions and modelling logic for three core elements: state, action, and reward. The detailed design is elaborated below from three dimensions: state space, action space, and reward mechanism.

The state of an agent is an observation of the local environment, which needs to fully depict its task attributes and the network environment it is in. For the UA, its state vector must include its own task information and the perceived network environment, and is formally defined as:

$$s_i^{\mathrm{UA}} = \left[D_i, Type_i, Budget_i, Latency_i^{\max}, Q_i^{received} \right] \tag{1.13}$$

where s_i^{UA} is the state vector of the user agent. D_i is the task data volume of user *i*. $Type_i$ is the task type of user *i*. $Budget_i$ is the budget constraint of user *i*. $Latency_i^{\max}$ is the maximum allowable latency for the task of user *i*. $Q_i^{received}$ is the resource quotation vector received.

For the resource agent (RA), its state vector must include its own resource status and the received task request information, which is formally defined as:

$$s_j^{\mathrm{RA}} = \left[R_j, Q_j^{requests} \right] \tag{1.14}$$

where s_j^{RA} is the state vector of the RA. R_j is the resource status vector of resource node. $Q_j^{requests}$ represents all task requests currently received by resource node *j* and their bidding vectors.

Action is the core output of an agent's decision-making and must meet the joint needs of discrete and continuous decision-making. For the UA, it is formally expressed as:

$$a_i^{\mathrm{UA}} = [j, p_i], \quad j \in \{1, 2, ..., M\}, \quad p_i \in [0, Bueget_i] \tag{1.15}$$

$$a_j^{\mathrm{RA}} = \left[Price_j^C, Price_j^S, Price_j^B \right] \tag{1.16}$$

where *j* is the index of the resource node selected by user *i*. p_i represents the fee that the user is willing to pay for the resources. $Price_j^C$ represents the price per unit of computing power. $Price_j^C$ represents the price per unit of storage capacity. $Price_j^S$ represents the price per unit of bandwidth.

The above equations formally define and continuously process the decision space of the agent. Combining discrete selection and continuous bidding or pricing as action outputs enables the policy network to perform end-to-end optimisation through gradient descent, which is a prerequisite for the application of algorithms like DDPG that handle continuous action spaces.

The reward function is the core mechanism that drives each agent's learning behaviour. A reasonable reward function design can map global system goals into local immediate signals for each agent, thereby achieving collaborative optimisation of individual behaviours and overall utility. The UA reward and RA reward are expressed as:

$$r_i^{\mathrm{UA}} = QoE_i - \eta \cdot Cost_i \tag{1.17}$$

where r_i^{UA} represents the immediate reward of the i^{th} UA. QoE_i represents the QoE of the i^{th} UA. η represents the QoE and cost balance coefficient. $Cost_i$ represents the actual payment cost of the i^{th} UA. r_j^{RA} represents the immediate reward of the j^{th} RA. μ represents the weighting coefficient of resource utilisation and revenue. $Utilisation_j$ represents the resource utilisation rate of the j^{th} RA. $Revenue_j$ represents the revenue of the j^{th} RA. v represents the overload penalty coefficient. $\mathbb{I}_{overload}$ represents the overload indicator function.

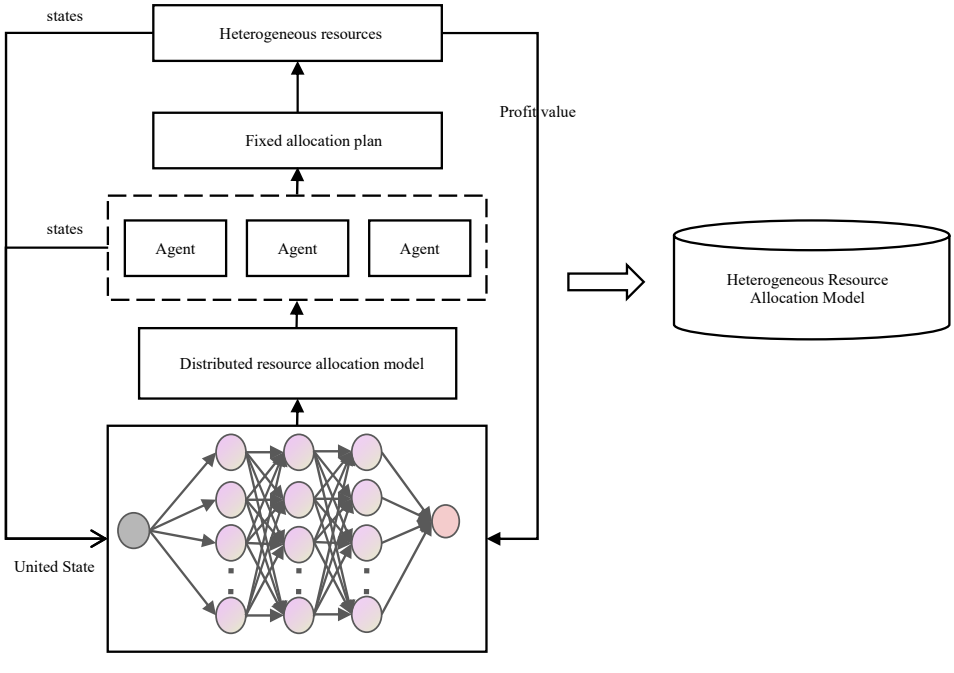
The overload penalty coefficient (η) introduces negative feedback into the reward function of resource agents. When a node's resource utilisation exceeds a safety threshold, penalties are applied to prevent node overload and encourage load balancing.

The reward function serves as a bridge that decomposes global optimisation objectives and allocates them to individual agents. These two equations transform abstract global goals into concrete, computable immediate signals for each agent at every step. Through the careful design of the reward function, when agents pursue the maximisation of their own cumulative rewards, their behaviours will naturally and collaboratively optimise the total utility of the system – skilfully aligning individual interests with collective interests.

3.3 Algorithm flow and network structure

Under the MADDPG framework, each agent is equipped with two sets of independent neural networks: an actor network and a critic network. This design adheres to the CTDE paradigm, enabling global collaborative learning while ensuring reliance on local information during the execution phase.

Figure 1 Network structure diagram of the MADDPG algorithm (see online version for colours)



As shown in Figure 1, during the training phase, the actor of each agent receives its own local state and outputs actions; the central critic collects the states and actions of all agents, calculates Q-values, and feeds back gradients to each actor. Only local actors are required during the execution phase.

In MARL scenarios, the core of collaborative learning lies in how to use a centralised value function to effectively evaluate and guide the policies of each agent. Based on the MADDPG framework, this paper systematically elaborates on its key training processes and objective functions: each agent maintains an experience replay buffer for storing transition tuples. The agent's critic network is updated by minimising the following loss function:

$$\mathcal{L}(\phi_i) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\left(Q_{\phi_i}(s,a) - y\right)^2\right] \tag{1.18}$$

$$y = r_i + \gamma Q_{\phi_j}\left(s', a'\right)\Big|_{a'=\mu_{\theta_j}(s')} \tag{1.19}$$

$$\nabla_{\theta_i} J(\theta_i) \approx \mathbb{E}_{s,a\sim\mathcal{D}}\left[\nabla_{\theta_i}\mu_{\theta_i}(s_i)\nabla_{a_i}Q_{\phi_i}(s,a)\Big|_{a_i=\mu_{\theta_i}(s_i)}\right] \tag{1.20}$$

where s denotes the environmental state at the time step. a denotes the action executed by the agent at the time step. r denotes the immediate reward obtained at the time step. ϕ_i denotes the critic network parameters of agent i. θ_i denotes the actor network parameters of agent i. θ_j denotes the target critic parameters of agent j. θ_j denotes the target actor parameters of agent j. y denotes the target value of the critic. r_i denotes the immediate reward of agent i at the current time step. γ denotes the discount factor. s' denotes the next state. a' denotes the next action.

The above equations constitute the mathematical core of the MADDPG algorithm. Equation (1.19) minimises the standard temporal difference error, training the critic to accurately evaluate the value of state-action pairs. Equation (1.20) calculates the target Q-value using a target network, effectively breaking the correlation between estimated values – a key technique for stabilising deep Q-learning. Equation (1.21) applies the deterministic policy gradient theorem to the multi-agent setting, enabling the actor network to adjust its policy in a direction that increases the Q-value.

3.4 Convergence and complexity analysis

Strictly speaking, in general multi-agent environments, the non-stationarity caused by the mutual influence of agents' policies makes it difficult to guarantee the convergence of MARL algorithms. However, during training, MADDPG can converge to a local Nash equilibrium because each agent's critic network provides a stable estimate of its Q-values given the policies of the other agents. This means that near the equilibrium point, no agent can unilaterally improve its reward through small policy changes. Although convergence to the global optimum cannot be guaranteed, extensive empirical studies show that MADDPG typically finds highly cooperative and high-performance strategies.

The algorithm's complexity mainly includes computational complexity and communication complexity: computational complexity primarily comes from the forward and backward propagation of neural networks. Communication complexity: only during the training phase do agents need to transmit their state and action information to the

central critic. During execution, agents operate completely distributed without communication, relying solely on their local actor networks to make decisions, resulting in zero communication overhead.

4 Experimental design and results analysis

4.1 Experimental setup

To validate the proposed algorithm's effectiveness, a large-scale simulation environment closely mimicking real-world lifelong education scenarios was constructed. This environment emulates a heterogeneous network architecture comprising 1 cloud data centre, 5 edge servers, and 50–500 dynamic user terminals.

The simulation platform was custom-developed using Python 3.8 + PyTorch 1.9.0 + OpenAI Gym. Each user terminal and resource node deployed corresponding agents employing the aforementioned MADDPG algorithm. Cloud data centre specifications: 1,000 TFLOPS computational power, 1 PB storage, 10 Gbps bandwidth. Edge server specifications: 50–200 TFLOPS computational power, 10–50 TB storage, 1–5 Gbps bandwidth. User terminal specifications: 0.5–5 TFLOPS computational power, 128–512 GB storage, 100–500 Mbps bandwidth. Learning parameters: Actor network learning rate 0.0001, Critic network learning rate 0.001; discount factor $\gamma = 0.95$, soft update parameter $\tau = 0.01$; experience replay buffer size 100,000, batch size 256.

4.2 Results and analysis

This experiment aims to validate the convergence performance and stability of the proposed MADDPG algorithm in the lifelong education resource allocation problem. Training was conducted with a fixed user scale (200 users), observing the trends of key metrics during training. The training rounds were set to 5,000, with algorithm performance evaluated every 100 rounds. The system's total utility, average QoE, and resource utilisation were recorded as they changed with training rounds. Metrics fluctuated during early training due to immature policies. The adoption of target networks and experience replay buffering mechanisms effectively stabilised the training process, with the algorithm converging after approximately 3,000 iterations.

As shown in Figure 2, the proposed algorithm exhibits excellent convergence properties. During the initial training phase (first 1,000 iterations), metrics fluctuate significantly as the agent's strategy matures. As training progresses, the agent gradually learns effective resource allocation strategies through interaction with the environment. The system's total utility and average QoE steadily improve, stabilising after approximately 3,000 iterations. Resource utilisation follows a similar convergence trend, ultimately maintaining within the reasonable range of 75%–80%.

Lifelong education platforms must serve massive user bases, making algorithm scalability critical. This experiment evaluates algorithm performance under varying loads by incrementally increasing user scale (from 50 to 500). User counts were incremented in 50-unit intervals, with 1,000 rounds run at each scale to record steady-state performance metrics.

Figure 2 MADDPG algorithm convergence performance analysis (see online version for colours)

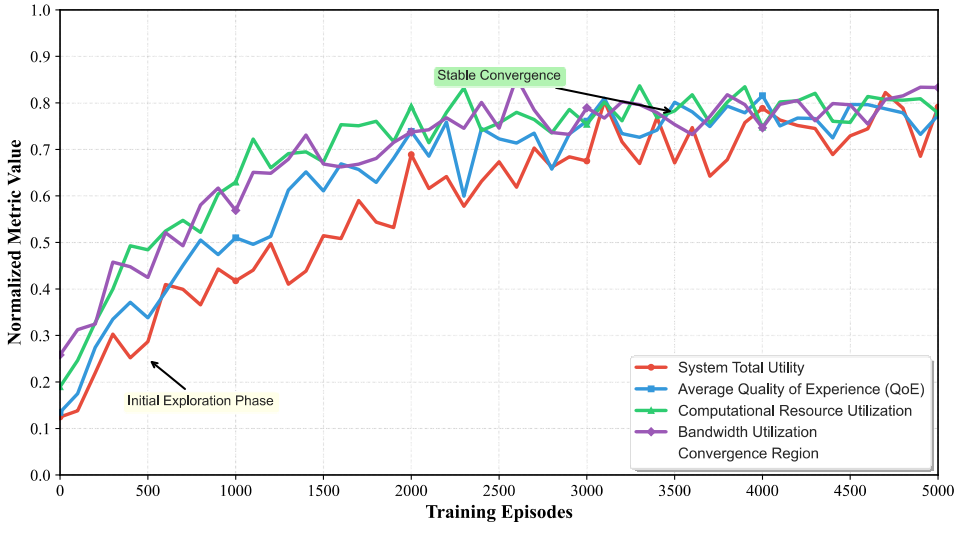
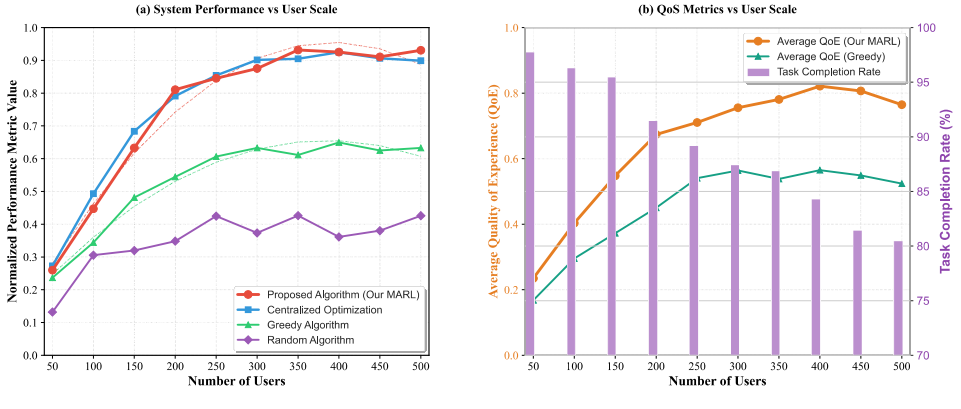


Figure 3 Algorithm scalability test results analysis (see online version for colours)



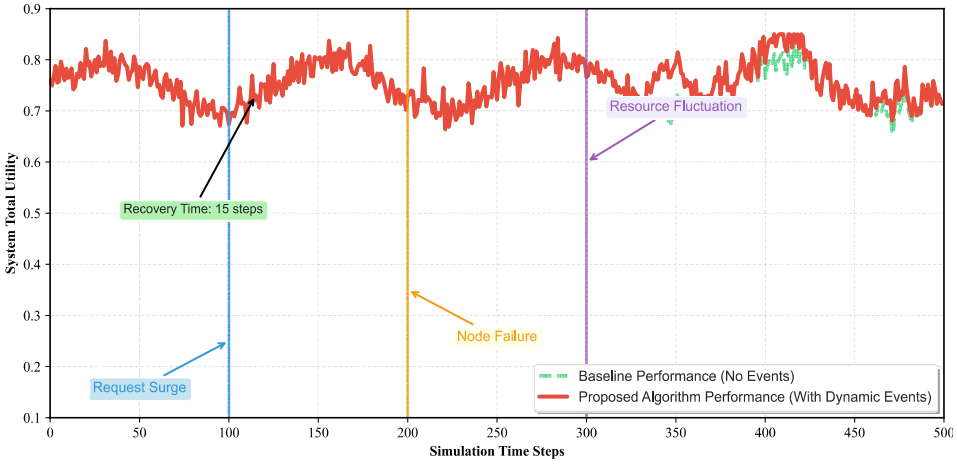
As shown in Figure 3, the proposed algorithm demonstrates excellent scalability as the user base expands. When the number of users is below 300, the system's total utility and average QoE exhibit nearly linear growth, indicating the algorithm effectively utilises additional resources to meet user demands. Beyond 300 users, performance metrics grow at a slower rate due to intensified resource competition, yet maintain a stable upward trend.

Compared to the baseline algorithms, the proposed algorithm significantly outperforms both the random (Borst et al., 2024) and greedy algorithms (Liu et al., 2024) at all scales. Particularly in large-scale scenarios (400–500 users), the performance advantage of the proposed algorithm becomes more pronounced, demonstrating its effectiveness in handling complex resource contention problems. Although centralised optimisation algorithms theoretically provide performance upper bounds, their

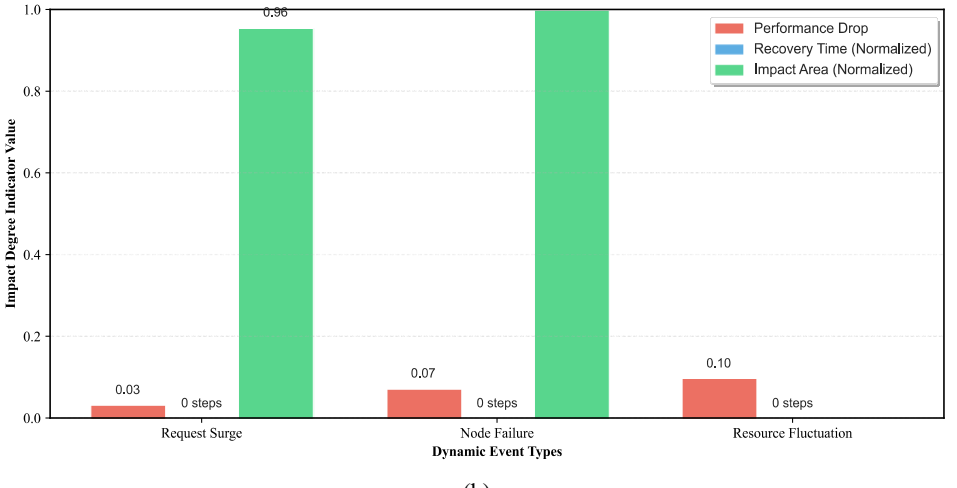
computational complexity grows exponentially with user scale, making them impractical for real-world large-scale deployments.

Lifelong education environments exhibit high dynamism, where user request patterns, network conditions, and resource availability may change at any time. This experiment simulates three typical dynamic scenarios to evaluate the algorithm's adaptability. At round 100, a sudden surge of user requests is introduced to simulate the access peak at the start of a course. At round 200, one edge server is randomly disabled to test the system's fault tolerance. At round 300, periodic bandwidth fluctuations are introduced to simulate network congestion.

Figure 4 Algorithm robustness and recovery capability, (a) algorithm performance adaptation in dynamic environments (b) impact analysis of different dynamic events (see online version for colours)



(a)



(b)

As shown in Figure 4, the proposed algorithm demonstrates excellent adaptability and robustness in response to dynamic environmental changes. During request burst

scenarios, system performance experiences a brief dip before rapidly recovering, with an average recovery time of only 15 rounds. In node failure scenarios, the algorithm mitigates performance impact to within 20% by redistributing tasks to other available nodes. For resource fluctuations, the algorithm dynamically adjusts resource allocation strategies to maintain relatively stable service quality.

In-depth analysis reveals that the algorithm's adaptability stems from its online learning mechanism and experience replay buffer. When environmental changes occur, the agent can rapidly adjust its strategy using historical experience without requiring retraining from scratch. This characteristic makes the proposed algorithm particularly suitable for dynamic lifelong learning environments in practical deployments.

Three experiments validate the algorithm's effectiveness from distinct perspectives. Experiment 1 demonstrates stable convergence to efficient resource allocation strategies; Experiment 2 showcases the algorithm's capacity to handle large-scale user requests; Experiment 3 verifies its robustness in dynamic environments. Collectively, the proposed MAR-based resource allocation model provides a practical solution for heterogeneous resource management in lifelong education.

Table 1 Summary table of key performance indicators

Task no.	*Evaluation dimension*	*Metric*	*Proposed algorithm*	*Greedy algorithm*	*Random algorithm*	*Improvement rate*
1	Efficiency	System total utility	0.85 ± 0.03	0.62 ± 0.05	0.45 ± 0.07	37.10%
2	Quality	Average QoE	0.78 ± 0.04	0.55 ± 0.06	0.38 ± 0.08	41.80%
3	Resource utilisation	Overall utilisation rate	76.3% ± 2.1%	68.5% ± 3.4%	52.7% ± 4.2%	11.40%
4	Fairness	Jain's index	0.89 ± 0.02	0.72 ± 0.04	0.61 ± 0.05	23.60%

Table 1 summarises the performance of the proposed algorithm across key metrics. It is evident that the proposed algorithm significantly outperforms the baseline algorithm in all metrics, with the most pronounced improvements observed in QoE and fairness. This fully demonstrates the algorithm's advantages in optimising the quality of lifelong education services.

5 Conclusions

This paper systematically proposes and validates a distributed resource allocation model based on MARL to address key challenges in allocating heterogeneous resources within lifelong education environments. Through theoretical modelling, algorithm design, and experimental validation, this study achieves the following major outcomes:

1 A distributed multi-agent resource allocation framework has been constructed. This framework models learners and resource providers as UA and RA, respectively, enabling collaborative decision-making through local perception and limited communication. It effectively overcomes the limitations of centralised models in scalability, privacy protection, and dynamic responsiveness.

2 A hybrid reward mechanism integrating multi-objective optimisation was designed. By unifying user QoE, system resource costs, and collective fairness within the

reward function, agents are guided to spontaneously optimise overall system efficiency while pursuing individual gains, achieving a balance between personalised services and global efficiency.

3 A formalised three-party stochastic game model was established, with a theoretical proof demonstrating the existence of a Nash equilibrium within this framework. This analysis provides theoretical support for the convergence of MARL algorithms, enhancing the rigor and depth of this research.

4 The proposed model's effectiveness was validated through large-scale simulation experiments. Results demonstrate significant advantages over baseline methods like random allocation and greedy algorithms across key metrics including total system utility, average QoE, resource utilisation, and fairness. Particularly notable is its adaptability and robustness in dynamic environments and large-scale user scenarios.

In summary, this paper presents a theoretically rigorous, highly practical, and easily extensible distributed solution for managing heterogeneous resources in lifelong education scenarios. This model is not only applicable to educational resource sharing platforms but also provides a reference paradigm for other intelligent scheduling scenarios involving multiple users and diverse resources.

Limitations of this study include the potential for further refinement of the user behaviour model and the lack of validation in real production environments. Future directions include integrating federated learning to enhance privacy protection and exploring dynamic resource prediction mechanisms.

Acknowledgements

This work is the phased achievements of the 2026 Social Science Funding Project of Jilin Provincial Department of Education named: 'Study on the long-term mechanism of artificial intelligence promoting the continuous optimization of adult learners' learning behaviors from the perspective of lifelong learning'.

Declarations

All authors declare that they have no conflicts of interest.

References

Antony, S., Claudio, C., Joao, N., Lucas, L., Nicolaas, R. and Sergio, L. (2024) 'Orchestrating multi-agent systems for multi-source information retrieval and question answering with large language models', *International Journal on Natural Language Computing*, Vol. 13, No. 6, pp.27–46.

Arvaneh, F., Zarafshan, F. and Karimi, A. (2024) 'Applying the cheetah algorithm to optimize resource allocation in the fog computing environment', *Applied Artificial Intelligence*, Vol. 38, No. 1, pp.12–20.

Baouya, A., Hamid, B., Gürgen, L. and Bensalem, S. (2024) 'Rigorous security analysis of RabbitMQ broker with concurrent stochastic games', *Internet of Things*, Vol. 26, pp.101161–101172.

Borst, S., Dadush, D., Huiberts, S. and Kashaev, D. (2024) 'A nearly optimal randomized algorithm for explorable heap selection', *Mathematical Programming*, Vol. 210, No. 1, pp.1–22.

Li, J., Chen, J., Huang, J. and Lei, X. (2025) 'Hyperbolic multivariate feature learning in higher-order heterogeneous networks for drug-disease prediction', *Artificial Intelligence in Medicine*, Vol. 162, pp.103090–103092.

Li, X., Suo, L., Jiao, W., Liu, X. and Liu, Y. (2024) 'Cooperative overbooking-based resource allocation and application placement in UAV-mounted edge computing for internet of forestry things', *Drones*, Vol. 9, No. 1, p.22.

Liu, J., Han, Y., Wang, Y., Liu, Y. and Zhang, B. (2024) 'Distributed hybrid flowshop scheduling with consistent sublots under delivery time windows: a penalty lot-assisted iterated greedy algorithm', *Egyptian Informatics Journal*, Vol. 28, p.100566.

Ma, Z., Li, X., Liu, Z., Huang, R. and He, N. (2024) 'Model checking fuzzy computation tree logic of multi-agent systems based on fuzzy interpreted systems', *Fuzzy Sets and Systems*, Vol. 485, pp.108966–108972.

Maksymov, O., Toshev, O., Demydenko, V. and Maksymov, M. (2024) 'Simulation modeling of artillery operations in computer games: approach based on Markov processes', *Technology Audit and Production Reserves*, Vol. 5, No. 2, pp.23–28.

Meng, Z., Xia, X., Zheng, Z., Gao, L., Liu, W., Zhu, J. and Ma, J. (2024) 'Unified multi-modal multi-agent cooperative perception framework for intelligent transportation systems', *SAE International Journal of Advances and Current Practices in Mobility*, Vol. 7, No. 4, pp.1530–1537.

Niu, D., Zhang, L., Zhang, B., Zhang, Q., Ding, S., Wei, H. and Li, Z. (2024) 'DVGEDR: a drug repositioning method based on dual-view fusion and graph enhancement mechanism in heterogeneous networks', *Complex & Intelligent Systems*, Vol. 11, No. 1, p.68.

Su, P., Lin, X., Lu, W., Xiong, F., Peng, Z. and Lu, Y. (2024) 'Generative design for complex floorplans in high-rise residential buildings: a Monte Carlo tree search-based self-organizing multi-agent system (MCTS-MAS) solution', *Expert Systems With Applications*, Vol. 258, p.125167.

Wang, Y., Xie, Y., Liu, Y. and Li, Z. (2025) 'Identification of disease-protein associations from a heterogeneous network using graph embedding and sample screening algorithms', *Current Proteomics*, Vol. 21, No. 6, pp.824–842.

Wang, Z. and Liu, J. (2024) 'Cooperative regulation based on virtual vector triangles asymptotically compressed in multidimensional space for time-varying nonlinear multi-agent systems', *ISA Transactions*, Vol. 157, pp.258–268.

Wang, Z., Xiao, F., Ran, Y., Li, Y. and Xu, Y. (2024) 'Scalable energy management approach of residential hybrid energy system using multi-agent deep reinforcement learning', *Applied Energy*, Vol. 367, pp.123414–123420.

Wenjuan, X. and Xin, Z. (2024) 'Research on the integration and innovation development of digital empowered lifelong education under the background of informatization', *Education Reform and Development*, Vol. 6, No. 8, pp.213–218.

Zhao, F., Ren, H., Sun, K. and Zhu, X. (2024) 'GAN-based heterogeneous network for ancient mural restoration', *Heritage Science*, Vol. 12, No. 1, p.418.