Enhancing Autonomous Space Exploration with Distributed Case-Based Reasoning and Learning (DCBRL) in Multi-Agent Systems

Ardianto Wibowo, Paulo Santos, Amer Baghdadi, Matthew Stephenson, Jean-Philippe Diguet

i-SAIRAS 2024

Enhancing Autonomous Space Exploration with Distributed Case-Based Reasoning and Learning (DCBRL) in Multi-Agent Systems

Ardianto Wibowo[1,2,3]*, Paulo E Santos[2,3], Amer Baghdadi[1], Matthew Stephenson[2,3], Jean-Philippe Diguet[3]
*lead presenter, ardianto.wibowo@imt-atlantique.fr
[1] IMT Atlantique, France
[2] Flinders University, Australia
[3] IRL CNRS, Crossing, Australia

ABSTRACT

Space exploration robots must operate autonomously due to the challenges posed by communication delays and power constraints, especially in dynamic and unpredictable extraterrestrial environments. Decentralized Multi-Agent Reinforcement Learning (MARL) offers a potential solution by enabling agents to operate without the need for continuous communication with a central controller, thus alleviating communication delay issues. However, traditional MARL approaches are not inherently optimized for power efficiency, and suffer from non-stationarity issues, which can destabilize the learning process. To address these challenges, we propose a preliminary version of an innovative solution that combines distributed Case-Based Reasoning (CBR) and MARL to form a Distributed Case-Based Reasoning and Learning (DCBRL) implemented in a decentralized way. DCBRL addresses the challenges of non-stationarity and dynamic environmental changes through a trust-based mechanism that allows agents to adapt quickly and share successful strategies. By leveraging QCBRL principles, the proposed system enables autonomous agents, such as planetary rovers or drones, to cooperate efficiently in unpredictable extraterrestrial environments, ensuring mission success despite communication delays.

1. INTRODUCTION

Space robotics face significant challenges due to communication delays and power constraints, leading to a reliance on ground control or tele-operation. Despite advances in autonomous capabilities over the past two decades, spacecraft still heavily depend on pre-scripted commands and human operators [1]. However, this approach becomes unsustainable in dynamic and unpredictable environments, particularly during physical interactions with planetary surfaces, as demonstrated by Mars operations. In such scenarios, autonomy becomes essential as environmental changes occur unpredictably, and the necessary response time is often shorter than the communication cycle with ground control [2].

One promising approach to address these challenges is decentralized Multi-Agent Reinforcement Learning (MARL), which enables multiple autonomous agents to coordinate and adapt their policies in dynamic environments without the need for a central controller [3]. Decentralized MARL allows agents to make a collective decisions based on local observations, which is crucial in environments where communication with ground control is delayed or intermittent. However, decentralized MARL systems also face the issue of non-stationarity, wherein agents

simultaneously adapt their policies, causing shifts in the environment's dynamics from each agent's perspective. This continuous adaptation violates the Markov assumption and destabilizes the learning process, making it difficult for agents to converge to stable policies [4].

The problem is often linked to overfitting, where agents become overly reliant on their previously learned policies, which may not generalize well to new or changing conditions. This is where Case-Based Reasoning (CBR) offers a potential solution, as it allows agents to leverage past successful experiences through a collection of Cases, without solely depending on learned parameters like Q-tables or neural networks. According to the Qualitative Case-Based Reasoning and Learning (QCBRL) algorithm in [5], CBR enables agents to leverage past experiences to inform decision-making, allowing them to adapt their policies to similar situations much quicker. Trust mechanisms can further enhance this process by ensuring that outdated or unreliable Cases are gradually discarded, making the system more robust to changes over time.

Inspired by the work in [5], we propose combining the advantages of Distributed CBR and MARL to address cooperative missions in dynamic environments within multi-agent systems, named Distributed Case-Based Reasoning and Learning (DCBRL). In this approach, each agent maintains its own case base and uses a trust-based system to update the relevance of stored cases. Successful cases receive higher trust-values, while unsuccessful or outdated cases are gradually discarded. Additionally, successful cases can be shared with other agents, enabling them to benefit from strategies proven effective in similar situations. This allows agents to focus on strategies effective under current environmental conditions. By balancing exploration-exploitation in both Q-learning and trusted cases in CBR, this hybrid approach enables agents to adapt dynamically while minimizing the computational overhead of continual exploration.

The DCBRL approach holds significant potential in space exploration scenarios where autonomous navigation, reliability and coordination between multiple agents are critical. By allowing agents—such as rovers or drones—to share successful strategies and experiences, like locating targets or navigating hazardous terrains, through DCBRL and local communication, these agents can continue to operate effectively even in environments with communication delays or interruptions. This capability is especially valuable in planetary exploration, where real-time communication with ground control is limited or non-existent. DCBRL thus enables space exploration agents to adapt quickly and work cooperatively to achieve mission objectives, even in the face of unpredictable and dynamic extraterrestrial environments.

2. RELATED WORKS

The advancements in autonomy across various mission phases emphasize the increasing necessity for autonomous systems in future space exploration. Despite significant progress, most spacecraft and planetary rovers or drones still rely heavily on ground control for decision-making [1], a model that is unsustainable for future missions, especially those involving interaction with unpredictable environments like planetary surfaces. Autonomy will play a critical role in addressing challenges related to communication delays, limited knowledge, and dynamic environmental conditions. Past successes in autonomous systems, such as in spacecraft navigation and surface mobility, underscore the need for more flexible and robust autonomous systems capable of adapting to space's uncertainties [2].

Several decentralized MARL approaches have been developed to tackle the dynamic environments. Hysteretic Q-learning introduces different learning rates for positive and negative updates, improving stability in cooperative tasks [6], [7]. However, this method suffers from slow adaptation and depends on careful tuning of learning rates. Lenient Q-learning, which encourages exploration by initially ignoring negative rewards, gradually becomes stricter [6], but it also faces complex tuning issues and struggles with adaptability in fast-changing environments.

An alternative approach to handling dynamic environments is the integration of CBR, a problem-solving method that follows four steps: RETRIEVE, REUSE, REVISE, and RETRAIN [8]. In the RETRIEVE step, the system searches for past cases that are similar to the current problem. During REUSE, the solution from the retrieved case is adapted to the new problem. The REVISE step tests and adjusts the solution to ensure its effectiveness. Finally, in RETAIN, the system stores the new solution as a case for future problems, continuously expanding its knowledge base. The integration can be realized in two ways: 1). RL supporting the CBR process [9], [10], [11] , or 2). CBR supporting the RL process [12]. A notable single-agent solution is QCBRL [5], which integrates Case-Based Reasoning (CBR) and Q-learning to improve adaptability in dynamic environments while reducing computational costs. However, QCBRL is designed for a strongly centralized Multi-Agent Reinforcement Learning (MARL) setup, where a single state vector represents the positions of all agents (joint observation), and a single action vector represents the actions of all agents (joint action). Due to the requirement of a central controller that communicates intensively with all agents, this approach is not suitable for distributed systems, particularly in scenarios like planetary exploration, where effective communication and coordination must be achieved with minimal or no centralized control.

Building on these insights, this research proposes a novel DCBRL solution implemented in a decentralized way. This approach extends the principles of QCBRL to tackle the challenges of decentralized multi-agent coordination, particularly in dynamic environments. DCBRL introduces a trust-based adaptation mechanism through distributed CBR, allowing agents to quickly adapt without continually updating Q-tables. Additionally, DCBRL optimizes communication by efficiently timing exchanges and facilitating indirect knowledge sharing.

3. METHODS

Due to space limitations, this paper does not delve into the theoretical backgrounds, which can be found in [3], [13] for MARL, and in [8], [14], [15] for DCBR. This section outlines the methodology behind the proposed DCBRL approach.

The overall process flow of DCBRL for each agent is illustrated in Figure 1. At each timestep, an agent observes its physical state o^p and, if available, receives communication actions $a^c{}_{others}$ from other agents. The communication actions $a^c{}_{others}$ contain cases broadcasted by other agents, based on their prior successful experiences. Based on the received observation data, the agent uses a greedy parameter to determine whether to select an action from the Case Base *(CB)* or to generate it through the Problem Solver component.

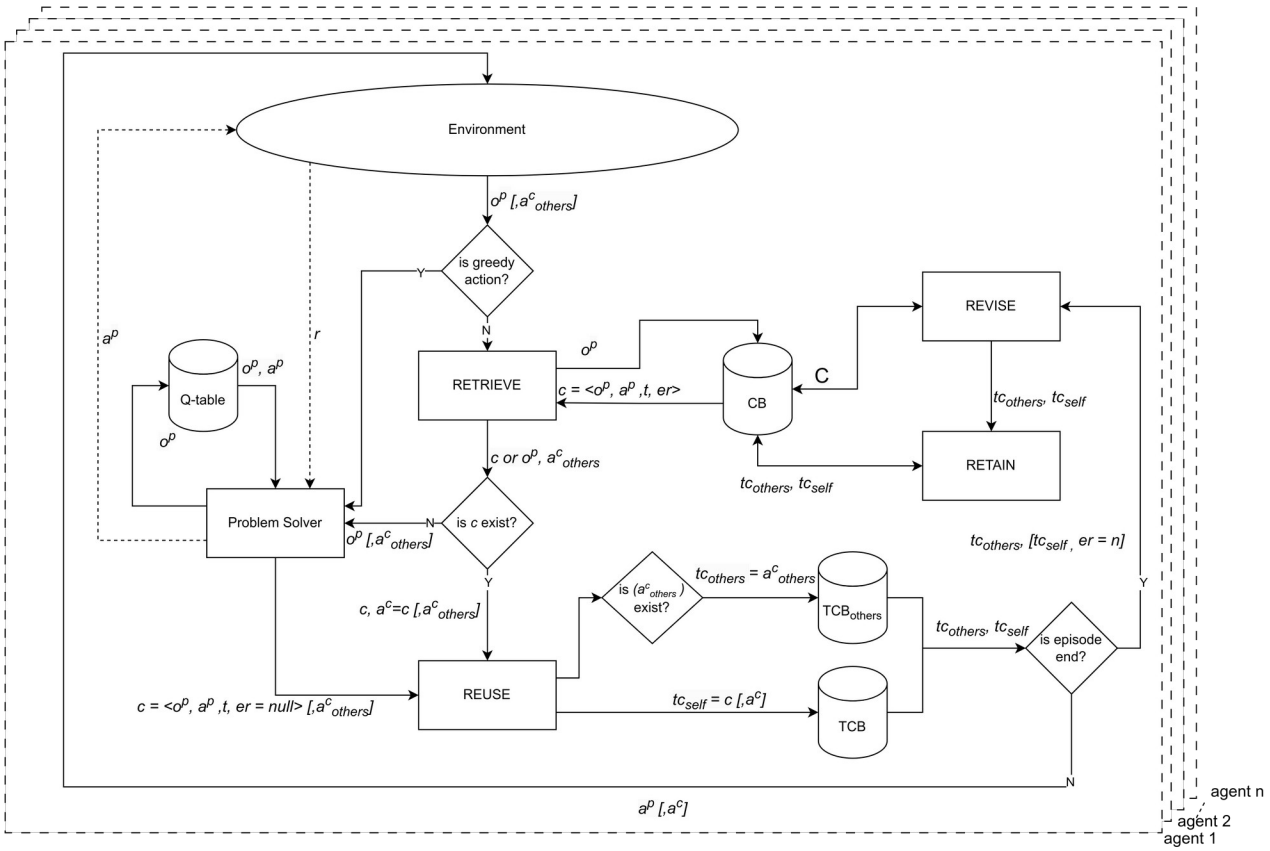


Figure 1: DCBRL algorithm process flow

If the action is selected from the CB, the agent executes the RETRIEVE process, searching for a matching case *c* in its local case base, based on the observed physical state o^p. If a matching case is found, the associated action from that case is reused in the REUSE step. If no matching case exists, the agent uses the *Problem Solver*, such as a Q-learning algorithm, to determine the appropriate physical action a^p based on its Q-table or on a random action. A new case is then generated as $c = <o^p, a^p, tv, er>$, where $tv=0.5$ is the initial trust-value and $er=null$ is the initial episode rewards. This *er* will be updated at the end of each successful episode, just before it is stored to the case base. The case is stored in the agent's Temporary Case Base *(TCB)*.

If communication actions a^c_{others} are available, they are stored in the agent's TCB_{others}. The agent complete the physical action a^p that will impact the environment, and, if the action was retrieved from a case in the *CB*, it also completes a communication action a^c that broadcast the current appropriate case *c* to other agents. This cycle continues until the episode is completed.

At the end of the episode, the agent performs the REVISE and RETAIN steps. During the REVISE step, the agent adjusts the trust-value tv of each case by incrementing or decrementing it based on the outcome of the episode. In the RETAIN step, the algorithm checks for trust-value removal, where cases with low trust-values are discarded from the *CB*.

5. EXPERIMENTAL RESULTS

This section presents the experimental results of the Multi-Agent Decentralized Case-Based Reinforcement Learning (MA-DCBRL) algorithm, compared against the following approaches:

1. *Multi-Agent Q-learning (MA-QL):* Implements the standard Q-learning mechanism to update the Q-table based on agent-environment interactions.
2. *Multi-Agent Lenient Q-learning (MA-LQL):* Utilizes the Lenient Q-learning approach [6], applying leniency to early negative rewards to encourage exploration and gradually tightening learning as agents improve.
3. *Multi-Agent Hysteretic Q-learning (MA-HQL):* Leverages Hysteretic Q-learning approach [6], where different learning rates are applied to positive and negative updates, promoting optimistic updates while handling suboptimal experiences more conservatively.

Initially, *MA-QL*, *MA-LQL*, and *MA-HQL* operated without any communication between agents [6]. However, in this experiment, we introduce communication, where each agent maintains its own Q-table but can also utilize shared Q-values communicated by other agents. For the consensus mechanism, all of the three comparable algorithms update the Q-table using a weighted combination of self-experience and other agents' experiences, as follows:

$$Q_{\text{updated}}(s,a) = (1-\alpha_1)\cdot Q_{\text{self}}(s,a) + \alpha_2 \cdot Q_{\text{shared}}(s,a)$$

Where:

- $Q_{self}(s,a)$ represents the Q-value for the state-action pair based on the agent's own experience.
- $Q_{shared}(s,a)$ is the Q-value shared by other agents for the same state-action pair.
- α_1 is the learning rate that controls the influence of the agent's own experience $Q_{self}(s,a)$.
- α_2 is the learning rate that governs the contribution of other agents' experience $Q_{shared}(s,a)$.

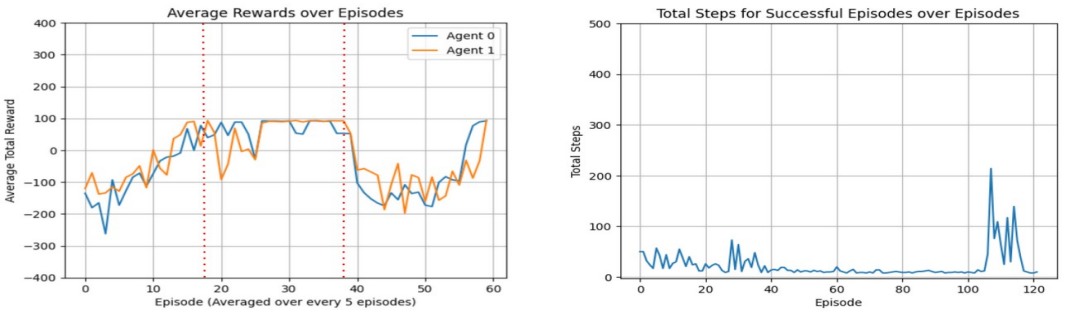
The experiments were conducted in a Multi-Agent Grid World 5x5 scenario, involving two agents cooperating to complete a navigation task. The agents are able to share knowledge with each other to increase their success rate in reaching the target, using the proportion of $\alpha_1 = 0.7$ and $\alpha_2 = 0.3$. An episode is considered successful when both agents reach the target.

Conversely, an episode is deemed a failure if any agent hits an obstacle. The environment is dynamic, meaning the positions of obstacles change after a fixed number of steps, forcing the agents to adapt their strategies.

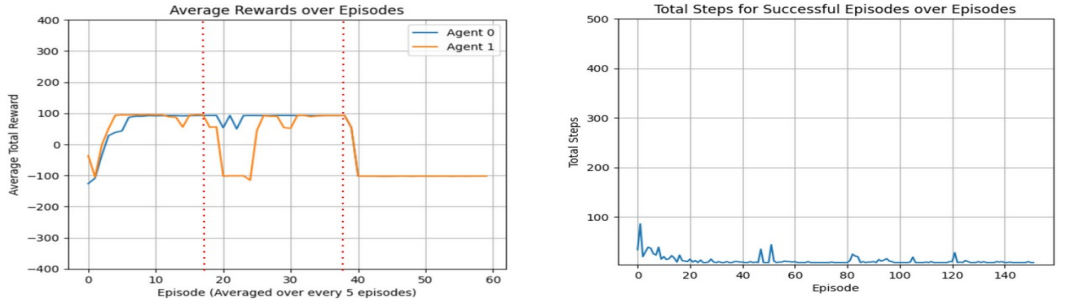
The reward function used in the experiments is defined as follows:

- A penalty of -100 for hitting an obstacle.
- A reward of +100 for reaching the target.
- A penalty of -1 for each normal movement step.

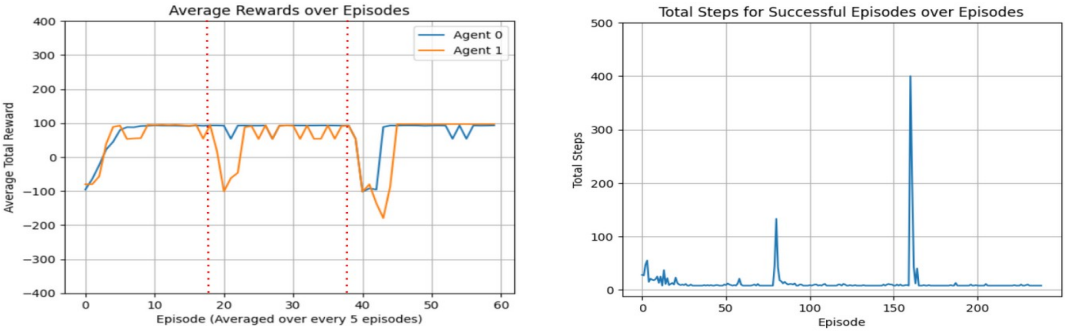
MA-HQL
Success rate:
40.80%



MA-LQL
Success rate:
50.83%



MA-QL
Success rate:
79.93%



DCBRL
Success rate:
90.30%

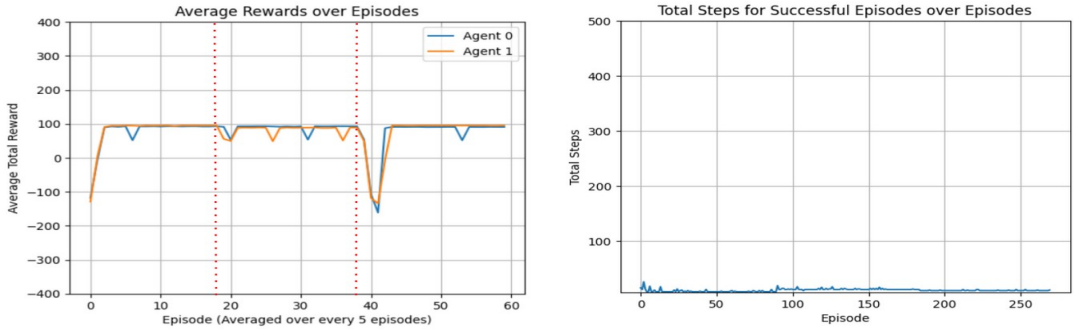


Figure 2: DCBRL algorithm comparison performance

In this experiment, the environment changes every episodes through the movement of three obstacles in a pre-defined pattern. The controlled changes ensure a fair comparison between the algorithms. The first movement of the obstacles is relatively minor, allowing the agents to adapt easily. However, the second movement is a more drastic change, as it shifts the obstacles into positions that close off previously discovered optimal paths, forcing the agents to discover new, optimal strategies.

Figure 2 presents a comparison of the three algorithms under different environmental conditions:

1. *Average Rewards:* on the left side, Figure 2 shows the average rewards obtained over the episodes in three different environmental conditions. Each environment change is marked by a thin dotted red line. It can be observed that the DCBRL algorithm consistently outperforms the other three algorithms. After each environmental change, the agents running the DCBRL algorithm are able to rapidly adjust their strategy, finding new successful trajectories. In contrast, the other algorithms show slower adaptation to the changing environment.
2. *Total Steps to Success:* on the right side, Figure 2 also shows the total steps required to complete a successful episode. The results indicate that the DCBRL algorithm requires the smallest number of steps to achieve success compared to the other three algorithms. This reflects the algorithm's ability to quickly adapt to changes in the environment and find shorter, more efficient paths to the target.
3. *Success Rate:* The DCBRL algorithm also achieves the highest success rate compared to the other three approaches. The success rate is calculated by dividing the number of successful episodes (where both agents reach the target) by the total number of episodes. The success rate of DCBRL remains high even in the presence of significant environmental changes, demonstrating the robustness of the algorithm.

6. DISCUSSION

While DCBRL shows significant potential, there are still several limitations that must be addressed in future work. First, the algorithm currently does not tackle the scalability challenges that arise when dealing with a large number of agents in MARL environments. Second, the focus on discrete observation spaces limits the algorithm's applicability in continuous environments, which are common in real-world scenarios. Finally, the algorithm does not prioritize minimizing risky actions, which is crucial for ensuring the safety of agents, especially during autonomous space missions. Addressing these limitations by improving scalability, adapting to continuous observation spaces, and incorporating safe action selection will be important areas for further research and development.

7. CONCLUSION

This paper presents the decentralized DCBRL algorithm for multi-agent systems, specifically designed to handle non-stationarity or dynamic environments, such as those encountered in planetary exploration. By integrating distributed Case-Based Reasoning (CBR) with Multi-Agent Reinforcement Learning (MARL), DCBRL enhances both knowledge sharing and adaptability among agents. Experimental results demonstrate that DCBRL outperforms several approaches like MA-HQL, MA-LQL, and MA-HQL with communication, positioning it as a promising solution for autonomous space missions.

ACKNOWLEDGEMENT

This research was co-funded by the Marie Skłodowska-Curie Actions (MSCA) through the AUFRANDE program.

REFERENCES

[1] R. B. Ashith Shyam *et al.*, "Autonomous Robots for Space: Trajectory Learning and Adaptation Using Imitation," *Front. Robot. AI*, vol. 8, p. 638849, May 2021.

[2] I. A. D. Nesnas, L. M. Fesq, and R. A. Volpe, "Autonomy for Space Robots: Past, Present, and Future," *Curr. Robot. Rep.*, vol. 2, no. 3, pp. 251–263, Sep. 2021.

[3] K. Zhang, Z. Yang, and T. Başar, "Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms," Apr. 28, 2021, *arXiv*: arXiv:1911.10635. Available: http://arxiv.org/abs/1911.10635

[4] L. Canese *et al.*, "Multi-Agent Reinforcement Learning: A Review of Challenges and Applications," *Appl. Sci.*, vol. 11, no. 11, p. 4948, May 2021.

[5] T. P. D. Homem, P. E. Santos, A. H. Reali Costa, R. A. Da Costa Bianchi, and R. Lopez De Mantaras, "Qualitative case-based reasoning and learning," *Artif. Intell.*, vol. 283, p. 103258, Jun. 2020.

[6] C. Amato, "An Introduction to Decentralized Training and Execution in Cooperative Multi-Agent Reinforcement Learning," Aug. 19, 2024, *arXiv*: arXiv:2405.06161. Available: http://arxiv.org/abs/2405.06161

[7] L. Matignon, G. J. Laurent, and N. Le Fort-Piat, "Hysteretic Q-learning : an algorithm for Decentralized Reinforcement Learning in Cooperative Multi-Agent Teams," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA: IEEE, Oct. 2007, pp. 64–69.

[8] A. Aamodt and E. Plaza, "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches," *AI Commun.*, vol. 7, no. 1, pp. 39–59, 1994.

[9] M. Kolbe, P. Reuss, J. M. Schoenborn, and K.-D. Althoff, "Conceptualization and Implementation of a Reinforcement Learning Approach Using a Case-Based Reasoning Agent in a FPS Scenario," in *Workshop on Knowledge Management*, Berlin, 2019.

[10] S. Wender and I. Watson, "Combining Case-Based Reasoning and Reinforcement Learning for Unit Navigation in Real-Time Strategy Game AI," in *Case-Based Reasoning Research and Development*, vol. 8765, L. Lamontagne and E. Plaza, Eds., in Lecture Notes in Computer Science, vol. 8765. , Cham: Springer International Publishing, 2014, pp. 511–525.

[11] Y. Huang, R. Wang, Z. Wei, and G. Wang, "An Intelligent Design Method Based on Case-based Reasoning and Reinforcement Learning," in *2023 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, Singapore, Singapore: IEEE, Dec. 2023, pp. 1583–1587.

[12] S. Esfandiari, B. Masoumi, M. Reza Meybodi, and A. Niazi, "Accelerated Method Based on Reinforcement Learning and Case Base Reasoning in Multi agent Systems," *Int. J. Comput. Appl.*, vol. 38, no. 4, pp. 25–31, Jan. 2012.

[13] S. V. Albrecht, F. Christianos, and L. Schäfer, "Multi-Agent Reinforcement Learning: Foundations and Modern Approaches".

[14] R. Ros, J. L. Arcos, R. Lopez De Mantaras, and M. Veloso, "A case-based approach for coordinated action selection in robot soccer," *Artif. Intell.*, vol. 173, no. 9–10, pp. 1014–1039, Jun. 2009.

[15] E. Plaza and L. Mcginty, "Distributed case-based reasoning," *Knowl. Eng. Rev.*, vol. 20, no. 3, pp. 261–265, Sep. 2005.