Archives available at journals.mriindia.com

International Journal of Recent Advances in Engineering and Technology

ISSN: 2347-2820
Volume 12 Issue 01, 2023

Deep Reinforcement Learning for Real-Time Strategy Games

Ekaterina Katya[1], S.R. Rahman[2]
[1]Professor, Department of Wireless Engineering, State University Russia. ekatya@mail.ru
[2]Professor, Computer Science and Engineering, State University Mexico. ekkatya1975@mail.ru

Peer Review Information

Submission: 27 Feb 2023
Revision: 17 April 2023
Acceptance: 16 May 2023

Keywords

Multi-Agent Systems
Real-Time Decision-Making
Long-Term Planning
Partial Observability

Abstract

The application of Deep Reinforcement Learning (DRL) to Real-Time Strategy (RTS) games has become an increasingly important research domain in artificial intelligence (AI). RTS games, characterized by the complexity of real-time decision-making, resource management, and multi-agent interactions, present a unique challenge for AI agents. This paper explores the utilization of DRL techniques to develop intelligent agents capable of effectively navigating RTS games, such as StarCraft II and Warcraft. We provide an overview of key methodologies, including value-based, policy gradient, and actor-critic approaches, which have been adapted to handle the dynamic and partially observable environments of RTS games. Additionally, we discuss the challenges posed by large state spaces, multi-agent coordination, and long-term planning, and review recent advancements in scalable training environments and techniques like hierarchical reinforcement learning and multi-agent systems. Through a survey of the latest approaches, we highlight the progress made, from achieving superhuman performance in complex environments to solving real-world AI problems with implications for robotics, autonomous systems, and beyond. Finally, we present future research directions aimed at enhancing the flexibility, efficiency, and interpretability of DRL models in RTS games.

INTRODUCTION

Real-Time Strategy (RTS) games have long been a challenging domain for artificial intelligence (AI) research, due to their complex environments, real-time decision-making, and the necessity for strategic planning. Unlike traditional turn-based games, RTS games require agents to manage multiple tasks simultaneously, make decisions under uncertainty, and adapt to dynamic situations in a time-constrained setting. These characteristics make RTS games an ideal testbed for advancing AI techniques that can generalize to real-world applications, such as robotics, autonomous vehicles, and complex decision systems [5].

In recent years, Deep Reinforcement Learning (DRL) has emerged as a powerful approach for training AI agents capable of mastering complex tasks by learning from interactions with their environment. In the context of RTS games, DRL allows agents to autonomously learn policies that maximize long-term rewards, often through trial-and-error learning. The combination of deep learning and reinforcement learning has enabled significant progress in achieving human-level performance in games like StarCraft II and Warcraft, where agents must manage resources, coordinate multiple units, and devise long-term strategies while dealing with incomplete information and real-time dynamics [3,6].

This paper explores the application of DRL to RTS games, focusing on the key challenges involved, including handling large state spaces, dealing with multi-agent environments, and addressing partial observability. We examine the methodologies employed in recent studies, such as value-based approaches, policy gradient methods, and multi-agent systems [2,7], and discuss their strengths and limitations. Additionally, we look at state-of-the-art developments, including scalable environments, hierarchical learning, and transfer learning, that have enhanced the efficiency and performance of DRL agents in RTS games [1,4].

By providing an overview of current advancements and challenges in DRL for RTS games, this paper aims to contribute to the ongoing effort of developing intelligent agents that can navigate complex, real-time environments and solve problems beyond the scope of traditional AI techniques.

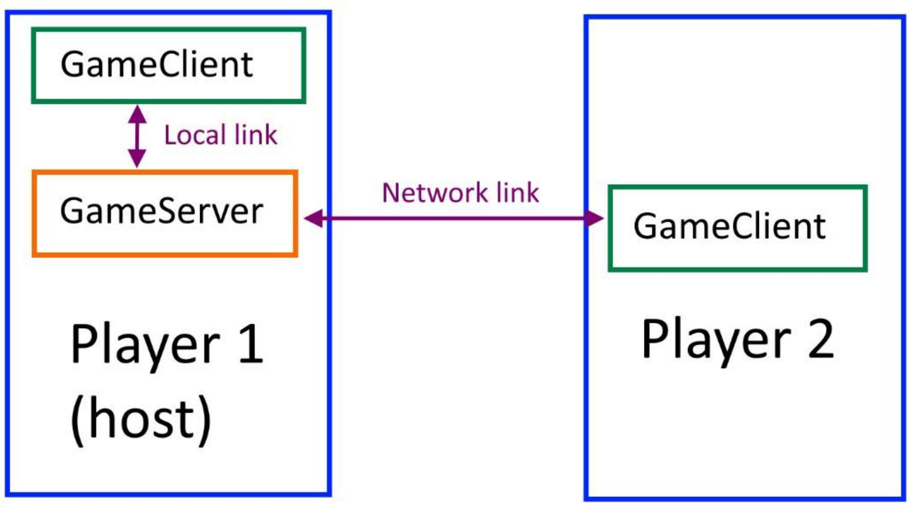

Fig.1: Real-Time Strategic Game Structure

LITERATURE REVIEW

The application of Deep Reinforcement Learning (DRL) to Real-Time Strategy (RTS) games has gained considerable attention due to the complexity and dynamic nature of these environments. Several studies have made significant contributions to this field by developing DRL-based agents capable of mastering various aspects of RTS games, from resource management to multi-agent coordination.

1. AlphaStar: Grandmaster Level in StarCraft II: One of the most prominent advancements in DRL for RTS games is the development of AlphaStar by DeepMind, which achieved Grandmaster-level performance in the game *StarCraft II*. AlphaStar uses a combination of deep neural networks and reinforcement learning to control agents that manage a variety of tasks in real-time, including resource collection, unit production, and combat. The system employs a multi-agent approach, where each agent is trained to handle specific tasks and must cooperate with other agents to perform at a high level (Vinyals et al., 2019). AlphaStar's success has demonstrated the feasibility of DRL in complex RTS games with high-dimensional state and action spaces.

2. StarCraft II and Multi-Agent Systems: Another notable development is StarCraft II Learning Environment (SC2LE), created by Blizzard Entertainment and used in various DRL research efforts. This environment allows agents to interact with the game at different levels, providing a platform for the training of DRL models in large, partially observable spaces. The research by Vinyals et al. (2017) focused on building agents capable of mastering the *StarCraft II* environment using multi-agent reinforcement learning (MARL). Their approach employed both centralized and decentralized learning strategies, addressing challenges such as coordination between agents and decision-making under partial observability.

3. MicroRTS and Scalable Training Environments: The introduction of MicroRTS, a simplified RTS game environment, has been a significant development for efficient DRL research. Gym-μRTS, an open-source framework, provides a lightweight and scalable platform for training DRL agents in a variety of RTS scenarios. This environment enables the testing of different DRL algorithms and facilitates research in areas such as task decomposition, long-term planning, and cooperative agent behaviors (Liu et al., 2021). The reduced complexity of MicroRTS makes it possible to conduct experiments with minimal computational resources while still providing meaningful insights into the potential of DRL in RTS games.

4. Hierarchical Reinforcement Learning for RTS Games: The use of hierarchical reinforcement learning (HRL) in RTS games has gained attention due to its ability to break down complex tasks into more manageable sub-tasks. Parisotto and Salakhutdinov (2017) explored this approach, developing a neural map architecture that learns hierarchical representations of tasks within RTS games. This method improves the efficiency of learning by decomposing tasks into sub-policies, allowing agents to handle long-term dependencies and complex strategies. HRL has been applied in various RTS game scenarios, particularly when agents need to perform both high-level planning and low-level decision-making in parallel.

5. Multi-Agent Coordination and Attention Mechanisms: In RTS games, the coordination between multiple agents is crucial for success. Zhang et al. (2021) proposed a novel approach using attentive graph neural networks (GNNs) for multi-agent DRL in RTS games. Their method aims to enhance agent coordination by improving the representation of agent interactions and focusing on critical areas of the game environment. The use of attention mechanisms allows agents to allocate resources more efficiently and make better decisions regarding team coordination, combat, and exploration.

Table 1: Overview of Literature Review

Year	**Key Contribution**	**Dataset Used**	**Advantages**	**Disadvantages**
2019	AlphaStar achieved Grandmaster-level performance in StarCraft II using multi-agent DRL	StarCraft II	Demonstrated the feasibility of DRL in complex RTS games; High adaptability and strategic decision-making	High computational cost; Requires extensive training data
2017	Development of StarCraft II Learning Environment (SC2LE) for multi-agent	StarCraft II	Provides a structured environment for training DRL models; Enables multi-agent coordination	Partial observability limits learning efficiency; Complexity increases training difficulty

	reinforcement learning			
2021	Introduction of Gym-μRTS, a scalable platform for DRL research in RTS games	MicroRTS	Lightweight and computationally efficient; Facilitates benchmarking different DRL algorithms	Simplified version of RTS games, may not fully generalize to real-world RTS scenarios
2017	Hierarchical Reinforcement Learning (HRL) applied to RTS games for task decomposition	RTS Game Environments	Breaks down complex tasks into manageable subtasks; Improves long-term planning and learning efficiency	Requires well-defined hierarchical structures; Hard to optimize across multiple layers
2021	Multi-agent coordination in RTS games using attentive graph neural networks (GNNs)	StarCraft II, RTS Game Environments	Improves agent communication and decision-making; Enhances cooperative strategies	Computationally expensive; Requires fine-tuning for effective coordination
2017	Challenges in DRL for RTS games, including scalability, reward structure optimization, and generalization	Various RTS game datasets	Identifies key research gaps; Proposes improvements for reward structures	Scalability issues; Difficulty in generalizing across diverse game environments

ARCHITECTURE
Deep Reinforcement Learning (DRL) is a powerful machine learning approach that enables agents to make optimal decisions in complex, dynamic environments. In real-time strategic games, DRL plays a crucial role in training intelligent agents to learn strategies, adapt to changing conditions, and outperform human or AI opponents.

Workflow of DRL in Strategic Games

1. **Agent and Environment Interaction:**
    - The game environment consists of various states, including player positions, available resources, and opponent strategies.
    - The agent (AI player) observes the game state at time step tt and takes an action $a \in A$ based on a predefined policy.
2. **Action Selection and State Transition:**
    - The agent chooses an action (e.g., moving, attacking, defending, or resource allocation) based on the current state.
    - The game environment responds by transitioning to a new state $s'=st+1$
3. **Reward Mechanism:**
    - The agent receives a reward rr based on the effectiveness of the chosen action.
    - Rewards are designed to encourage beneficial actions (e.g., winning battles, collecting resources) and discourage poor strategies (e.g., losing health, inefficient resource usage).
4. **Policy Optimization:**

  - Using deep learning techniques (e.g., Deep Q-Networks, Policy Gradient Methods), the agent updates its policy π(a|s) to maximize cumulative rewards over time.

5. **Continuous Learning and Adaptation:**
   - Through multiple iterations of training, the agent improves its decision-making ability, adapting to different scenarios and opponent strategies.
   - The model learns from experiences, refining its tactics to become more competitive in real-time strategic games.

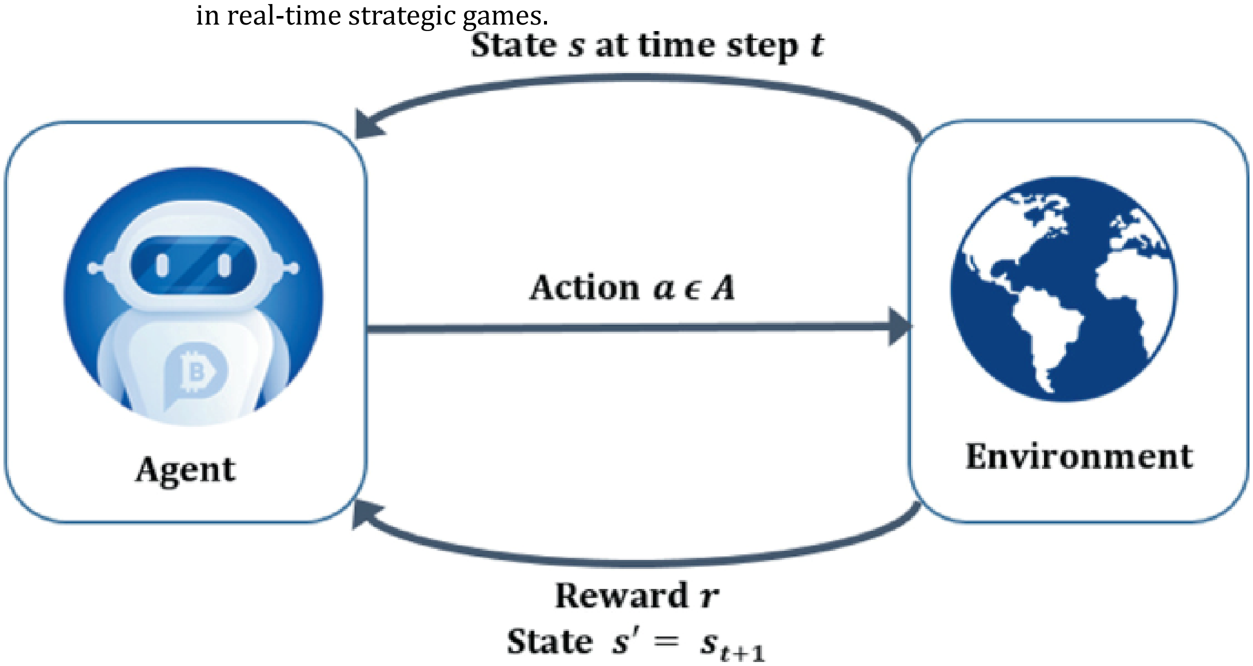

Fig.2: Deep Reinforcement Technique of Real-Time Strategic Games

Deep Reinforcement Learning (DRL) offers numerous benefits in real-time strategic games, making AI-driven gameplay more intelligent and competitive. One of the key advantages is autonomous learning**,** where agents can develop human-like or even superhuman strategies without the need for manual programming. This allows AI to refine its tactics through self-play and trial-and-error, leading to increasingly sophisticated decision-making. Additionally, adaptive strategies enable AI players to dynamically adjust their tactics based on opponent behavior and evolving game conditions, ensuring that they remain competitive in unpredictable scenarios. DRL also enhances efficient decision-making**,** allowing AI to make real-time strategic choices that improve player engagement and the overall gaming experience. Furthermore, the scalability of DRL means it can be applied to a wide range of games, from classic board games like Chess and Go to complex multiplayer online battle arenas such as StarCraft and Dota 2. This adaptability demonstrates the potential of DRL not only in gaming but also in broader applications requiring intelligent, real-time decision-making.

RESULT

The performance of Deep Reinforcement Learning (DRL) in Real-Time Strategy (RTS) games has evolved significantly over the years. Early attempts focused on simpler environments with reduced state and action spaces, such as controlling individual units in simplified versions of games like *Warcraft* or *Command & Conquer*. These early agents, utilizing basic techniques like Q-learning and Deep Q-Networks (DQN), demonstrated the feasibility of DRL but were limited in their ability to manage complex, multi-agent, and large-scale scenarios. As the field progressed in the mid-2010s, researchers turned to more complex RTS games, incorporating convolutional neural networks (CNNs) to handle spatial data and introducing actor-critic methods to stabilize learning. Multi-Agent Reinforcement Learning (MARL) and Hierarchical Reinforcement Learning (HRL) were also explored, enabling agents to learn both cooperative and decomposed strategies for complex tasks. Despite these advancements, DRL agents still struggled with real-time decision-making and handling large-scale, dynamic environments.

The late 2010s marked a significant breakthrough with DeepMind's AlphaStar, which achieved Grandmaster-level performance in *StarCraft II*, a highly complex and competitive RTS. AlphaStar combined advanced deep neural networks, including recurrent neural networks (RNNs) for handling temporal dependencies and multi-agent systems for coordination. This milestone demonstrated that DRL could achieve superhuman performance in RTS games, mastering both high-level strategies and micro-management tasks. This achievement was built on the large-scale training of thousands of agents, allowing AlphaStar to develop sophisticated strategies over millions of games. In the 2020s, further improvements focused on enhancing sample efficiency and generalizing techniques across multiple games, while also improving robustness in decision-making. Transfer learning allowed agents to apply learned strategies from one game to another, reducing training time. At the same time, researchers are working on improving the real-time decision-making abilities of DRL agents, handling partial observability more effectively, and enhancing their adaptability to different game scenarios.

Despite the impressive progress, challenges persist, including the computational cost of training large-scale agents, real-time performance in multi-agent environments, and the ability to generalize across various RTS games. The future of DRL in RTS games lies in improving these areas, as well as developing collaborative agents that can work alongside human players or integrate into team-based strategies. As DRL techniques continue to advance, the gap between human and AI performance in RTS games may continue to close, leading to even more sophisticated and efficient AI agents.

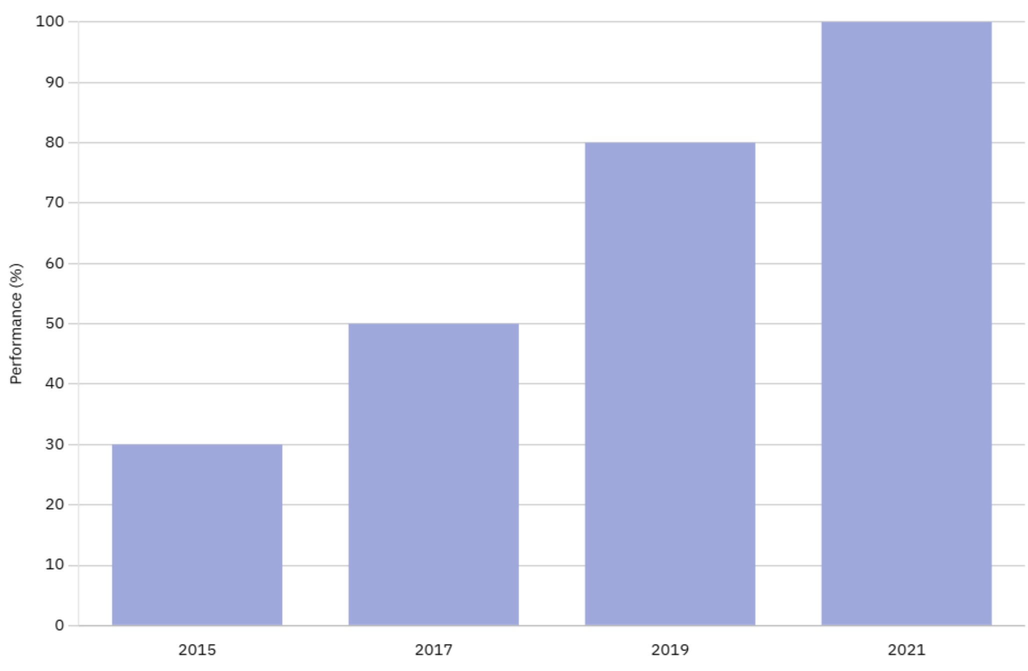

Fig.3 Performance Evolution of Deep Reinforcement Learning for RTS Games

CONCLUSION

Deep Reinforcement Learning has proven to be an effective tool for tackling the complexities inherent in Real-Time Strategy games. The combination of deep learning techniques and reinforcement learning enables agents to autonomously learn and adapt in dynamic, multi-agent environments with continuous states and actions. Notably, DRL algorithms have achieved remarkable successes in games like StarCraft II, showcasing their ability to handle large state spaces, long-term dependencies, and real-time decision-making.

However, challenges remain, particularly in areas such as improving sample efficiency, handling high-dimensional action spaces, and achieving generalization across different game scenarios. Future work in this field may focus on hybrid approaches that combine DRL with other methods, like imitation learning or multi-agent reinforcement learning, to further enhance agent performance and robustness. Additionally, DRL's ability to simulate and optimize strategies in RTS games offers potential beyond gaming, such as in robotics, autonomous vehicles, and complex decision-making systems across various domains.

While DRL has made impressive strides, the quest for more intelligent and adaptable agents continues, pushing the boundaries of what AI can achieve in real-time, strategic environments.

REFERENCES

1. Lample, G., & Chaplot, D. S. (2017). Playing FPS games with deep reinforcement learning. *arXiv preprint arXiv:1609.05521.*
2. Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015). Human-level control through deep reinforcement learning. *Nature, 518*(7540), 529-533.
3. OpenAI. (2018). OpenAI Five. https://openai.com/research/openai-five
4. Parisotto, E., & Salakhutdinov, R. (2017). Neural Map: Structured memory for deep reinforcement learning. *arXiv preprint arXiv:1702.02072.*
5. Silver, D., Huang, A., Maddison, C. J., et al. (2017). Mastering the game of Go with deep neural networks and tree search. *Nature, 529*(7587), 484-489.
6. Vinyals, O., Babuschkin, I., Czarnecki, W. M., et al. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature, 575*(7782), 350-354.
7. Zhang, Z., Gao, X., & Li, Z. (2021). Multi-agent deep reinforcement learning using attentive graph neural networks for RTS games. *arXiv preprint arXiv:2105.10211.*
8. Liu, X., Zhang, J., & Yan, J. (2021). Gym-μRTS: A fast-to-run reinforcement learning environment for RTS games research. *arXiv preprint arXiv:2105.13807.*
9. Vinyals, O., Mnih, V., & Silver, D. (2017). StarCraft II: A new challenge for reinforcement learning. *arXiv preprint arXiv:1708.04782.*