Svedberg Open
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Recursive Strategy Refinement Algorithms in Multi-Agent Collaborative Environments

C.K. Rajashri[1*], P.S.G. Aruna Sri[2], K. Priyadharshini[3], Dr.U. Harita[4], Ashu Nayak[5]

[1*]Assistant Professor, Department of Mathematics, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, Chennai, Tamil Nadu, India. E-mail: rajashrick@maher.ac.in

[2]Professor, Department of Internet of Things, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India. E-mail: arunasri_2012@kluniversity.in

[3]Assistant Professor, Department of Mathematics, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, Chennai, Tamil Nadu, India. E-mail: kpriyamba@maher.ac.in

[4]Assistant Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Guntur, Andhra Pradesh, India. E-mail: uharita@gmail.com, https://orcid.org/0000-0002-7809-1067

[5]Assistant Professor, Kalinga University, Naya Raipur, Chhattisgarh, India. E-mail: ku.ashunayak@kalingauniversity.ac.in, https://orcid.org/0009-0002-8371-7324

*Corresponding author: Email: rajashrick@maher.ac.in

Abstract

A multi-agent collaboration scenario necessitates coordination, effective task assignment, and strategic adjustment in order to attain maximum efficiency in a multi-agent environment. Existing solutions such as fixed heuristics and reinforcement learning often fail in scalability, convergence speed, and adaptability when employed in dynamic, diverse multi-agent systems. In this research, a Recursive Strategy Refinement model is suggested in which the agent's strategy can be adjusted recursively depending on feedback from their previous actions, thereby enhancing coordination efficiency, task performance, and resource management in multi-agent collaboration scenarios. Evaluation was done on simulated experiments in the OpenAI Multi-Agent Particle Environment and Kaggle Drone Swarm Coordination datasets. The recursive update was done using a gradient-based policy refinement approach that facilitated selective information exchange between heterogeneous agents and adaptive behavior improvement. The suggested model showed an average TCR of 92.6%, CE of 88.4%, and RU of 91.3%, surpassing traditional algorithms such as a static heuristic model where TCR is 72.4%, CE 68.3%, and RU 79.5%. The convergence rate was optimized from 35 iterations in static heuristic models to 15 iterations, compared to 28 iterations in non-recursive reinforcement learning algorithms. The convergence pattern revealed fast policy convergence, especially the TCR, which demonstrates statistically significant differences between both methods. Recursive strategy improvement facilitates efficient multi-agent coordination by converging fast, minimizing conflicts, and optimizing resource allocation.

Keywords: Multi-Agent Systems, Recursive Strategy Refinement, Task Completion Rate, Coordination Efficiency, Convergence Time, Resource Utilization, Adaptive Policy

1. Introduction

The study of multi-agent collaborative environments is one of the fundamental areas of artificial intelligence, whereby multiple agents can collaborate among themselves to perform complicated operations which cannot easily be performed by any one agent on its own [1][2]. Multi-agent collaborative environments find extensive application in robotics, self-driving cars, sensor networks, smart grids, and logistics management, among other areas, where coordination and resource allocation become crucial components in achieving the desired objectives within the system [3][4]. All individual agents act independently based on their perception and decision-making ability, but collectively contribute to the overall performance of the system. In order to gain insight into agent collaborations, communication, negotiations, and joint decisions of the agents must be well understood. This introduction to the topic forms the basis of the subsequent discussion on how agents can improve upon each other's strategies through iterations.

Dynamic environments are characteristic of real-world situations when multiple intelligent entities act simultaneously; they feature ongoing shifts in task characteristics, goals, and constraints [5][6]. Multi-agent systems are likely to encounter uncertainty caused by incomplete knowledge of environment state and unpredictable behaviors of other agents, as well as resource fluctuations [7][8]. Heuristic approaches are known to yield unsatisfactory results as they cannot ensure high efficiency in dynamic environments characterized by the lack of constancy and stability. The rationale for the current research is the need for dynamic solutions that enable learning from past experience and improving strategies in order to deal successfully with changes. Strategy refinement appears to be an effective technique as it allows for improvement through feedbacks.

Although significant improvements have been made in multi-agent systems, some critical issues still need addressing [10][12]. The first major issue faced by most multi-agent systems is that of scalability. Most multi-agent systems do not function effectively in cases where there are large numbers of agents or when the tasks are highly interdependent [19]. In addition, many multi-agent systems lack an element of adaptability that helps the system adjust in accordance with changing conditions [14]. Another issue with multi-agent systems is the problem of efficiency of convergence [16][18]. Multi-agent systems require that the agents keep refining their policies to ensure effective convergence. This study focuses on how to implement a multi-agent system that enables recursive policy refinement.

Research Objectives

1. To develop a recursive approach for refining the strategies of agents in collaborative multi-agent systems where the agents can recursively refine their strategies in response to feedback from their past actions.
2. To improve the coordination, task allocation, and overall system performance by utilizing the feedback-based strategy refinement approach.
3. To assess the influence of the proposed framework on several important performance metrics, such as efficiency, level of coordination, and resource utilization in comparison with static or heuristic solutions.
4. To test the generalizability of the proposed framework under various heterogeneous multi-agent systems characterized by a different number of agents and complexity of tasks.

The paper is structured as follows: In section II, a literature review of existing methods and research gaps is performed. The proposed methodological framework, its architecture, and mathematical modeling are described in section III. In section IV, the experiments' design, datasets, and performance metrics are described. In section V, experimental results and performance analysis are discussed. Lastly, in section VI, conclusions and future research directions are outlined.

2. Related Work

Cooperative Multi-Agent Systems have been an active area of research within the field of Artificial Intelligence, where the main concerns have been coordinated behavior, cooperative problem solving, and joint decision making among intelligent entities [13]. Examples of conventional techniques are distributed constraint optimization, task assignment through markets, and behavior-based coordinated actions that allow agents to communicate and collaborate towards their objectives [15] [17]. However, these techniques generally operate in rather static environments, hence rendering them unsuitable for use in more dynamic contexts. Studies related to this topic have confirmed that appropriate communication and decision-making strategies are of paramount importance to ensure the success of multi-agent systems [11] [20][22]. Furthermore, network structures, latency, and heterogeneity of the entities may also impact the coordination process.

Refinement of strategies refers to continuous improvements in the policies of agents so that they achieve optimal performances for tasks[24]. Some examples of strategy refinement methods include reinforcement learning techniques, policy gradient techniques, and iterated best response techniques. These strategies enable agents to improve based on their experiences and adjust to changes in the environment. Nevertheless, some of the current approaches used in refining strategies suffer from issues of long training time, centralization, or lack of ability to converge, especially in situations where there are complicated interdependencies among the agents. In addition, the problem of exploitation versus exploration may arise in strategy refinement.

Recursive algorithms offer a systematic structure within which agents may repeatedly refine their policies based on experience gained through feedback from their past behaviors [9][21]. Examples of such applications include recursive utility functions, dynamic programming approaches to coordination problems, and reinforcement learning techniques involving recursive updates to policy parameters. Though promising, such techniques are not yet integrated with adaptive learning processes or cannot be applied in cases where agents are different with respect to capabilities, goals, or levels of knowledge[23]. It has been recently shown that the performance of recursive techniques can be further improved by combining them with predictions or uncertainty estimation, but practical implementation is still lacking.

Although substantial progress has been made, current frameworks still lack scalability, flexibility in dealing with dynamic environments, and compatibility with learning-based coordination. Existing research on multi-agent systems mostly considers either homogeneous agents or static scenarios without a suitable method for recursive strategy improvement in a multi-agent scenario[25]. The motivation behind the suggested framework stems from the aforementioned deficiencies and the necessity for recursive strategy improvement methods that are flexible enough to be implemented in heterogeneous multi-agent environments. In addition, there is a lack of comparative results between recursive refinement and other methods, such as heuristics and learning, which will be discussed in this paper.

3. Proposed Recursive Strategy Refinement Framework

Framework Overview

This framework allows a number of agents to cooperate with one another in a changing environment via a process of strategy improvement that is performed recursively. Each individual agent has its own policy, which is periodically adjusted as per previous actions made and as per observations made concerning other agents. The agents have communication channels that they use selectively, and this makes it possible for them to exchange pertinent information concerning their work, available resources, and environmental dynamics without incurring high communication costs. It is also very appropriate for heterogeneous agents, which have varying capabilities and aims, and could be used in areas like autonomous robotics and logistic systems.

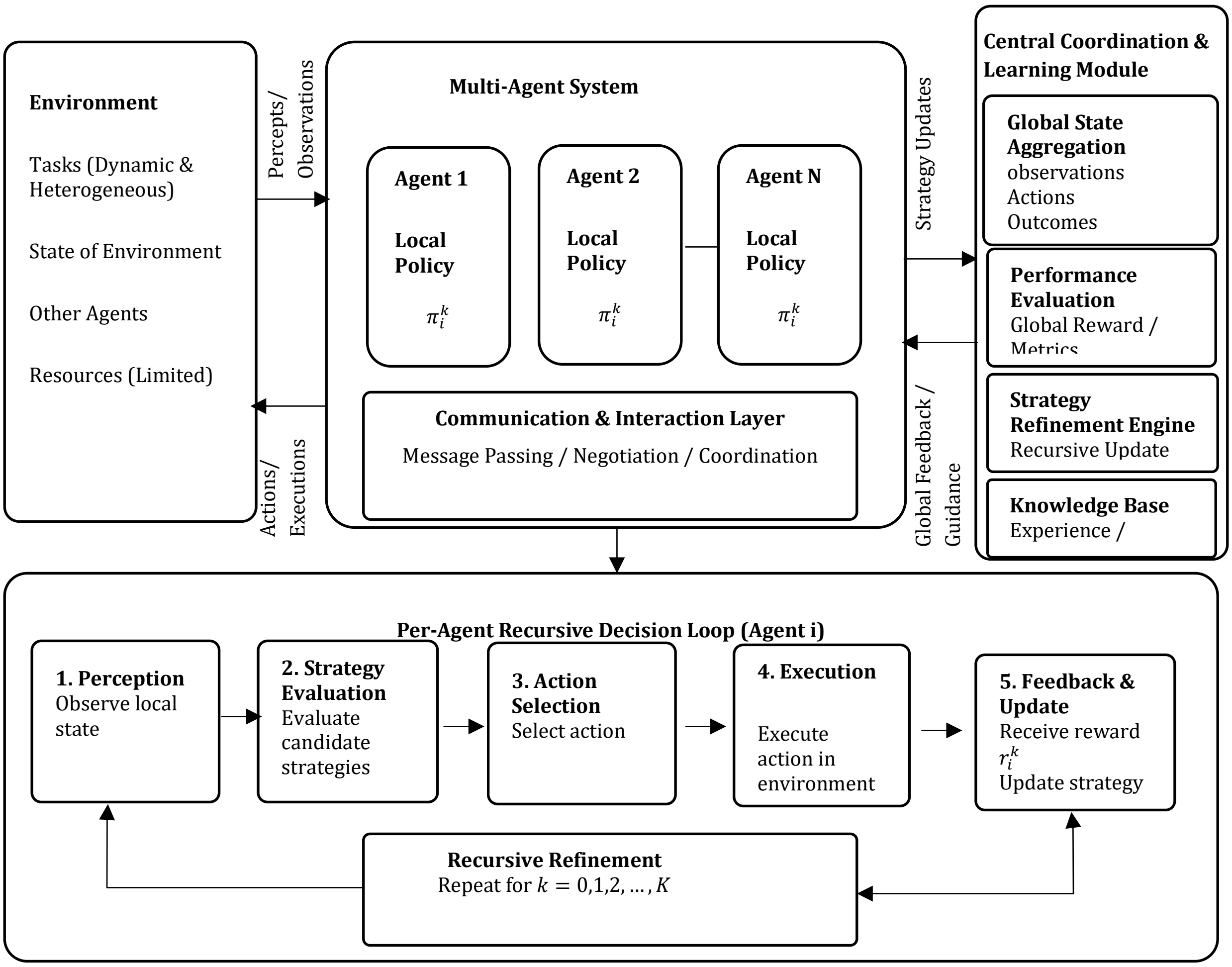

Figure 1: Architecture of the Proposed Recursive Strategy Refinement Framework

The architecture of the framework used for multi-agents collaboration system is shown in Figure 1. Four major layers are shown including the Agents Layer in which the agents independently sense the environment and perform actions according to their own strategies, the Feedback Layer where results are assessed and utilities are calculated, the Recursive Update Layer that does strategy improvement with the aid of policy updates driven by the feedback, and finally, the Communication Layer through which selected data communication is facilitated to improve coordination among agents. The second part of the figure emphasizes the agent-based iterative decision process, which comprises perception, strategy assessment, action selection, execution, and policy update.

Agent Interaction and Decision-Making Loops

Actions are performed by the agents in discrete time instants depending on the current policy and local observation. Having completed the action, agents assess the result using the utility function, which is the measure of success in the task performed, efficiency of the interaction, and effectiveness of resource utilization. Further, agents make a recursive process of updating the strategy based on the received feedback and, if necessary, on the observation of their neighbors' behavior. This decision cycle provides for an iterative improvement of the strategy by resolving possible conflicts and optimizing task allocation.

Algorithm 1: Recursive Strategy Refinement for Multi-Agent Collaboration

Input:

- Nagents with initial policies $\pi_i^0, i = 1,2,\ldots,N$
- Environment state $s_0$and task set T

- Learning rate α
- Maximum iterations K
- Utility function $U_i(\pi_i, \pi_{-i}, s)$

Output:

- Refined agent policies π_i^*
- Optimized task allocation and improved collaborative performance

Steps:

1. Initialize agents' policies π_i^0 and environment state s_0
2. For $k = 0$ to $K - 1$ do:
   a. For each agent $i = 1$ to N do:
      i. Observe local environment o_i^k and other agents' actions π_{-i}^k
      ii. Evaluate candidate strategies using utility:
         $U_i^k = U_i(\pi_i^k, \pi_{-i}^k, s_k)$
      iii. Select action a_i^k according to policy π_i^k
      iv. Execute action a_i^k in environment, observe reward r_i^k and new state s_{k+1}
      v. Update policy recursively:
         $\pi_i^{k+1} = \pi_i^k + \alpha \cdot \nabla_{\pi_i} U_i^k$
      vi. Communicate selective feedback to relevant agents
   b. End For
   c. Aggregate global feedback and update environment state s_{k+1}
3. End For
4. Return final agent policies π_i^* and system-level task performance metrics

In Algorithm 1, multiple agents can continuously improve their joint capabilities through iteration. Each agent monitors its immediate environment as well as the behavior of other agents, assesses the possible strategies with the use of a utility function which takes into consideration factors like success, coordination, and resource utilization, and makes its decisions based on such assessment. Agents then implement these decisions and receive feedback, after which they recursively optimize their utility functions by applying a gradient approach. Through selective communication, agents exchange necessary information without putting excessive demands on the communication network. This iterative procedure is repeated until agents find their best strategies and resolve any issues within the system.

Recursive Strategy Update Formulation

The recursive strategy refinement process can be expressed mathematically as:

$$\pi_i^{(t+1)} = \pi_i^{(t)} + \alpha \cdot \nabla_{\pi_i} U_i\left(\pi_i^{(t)}, \pi_{-i}^{(t)}, s_t\right) \quad (1)$$

Where in equation (1), $\pi_i^{(t)}$ is the policy of agent i at time step t, $\pi_{-i}^{(t)}$ represents the policies of all other agents, s_t is the system state, U_i is the utility function, and α is the learning rate controlling the magnitude of strategy refinement.

The utility function itself is defined as a combination of task performance, coordination quality, and resource efficiency:

$$U_i = w_1 \cdot R_i^{task} + w_2 \cdot R_i^{coord} + w_3 \cdot R_i^{res} \quad (2)$$

Where in equation (2), R_i^{task} measures task completion success, R_i^{coord} evaluates alignment with other agents' actions, R_i^{res} reflects efficient resource usage, and w_1, w_2, w_3 are weighting factors reflecting the relative importance of each metric.

4. Experimental Setup and Dataset

Experimental Environment

The recursive strategy refinement approach was tested through simulations within a multi-agent environment that models varying and heterogeneous tasks. This environment is made up of autonomous agents tasked with varying responsibilities and dependent on each other. These simulations were conducted through the use of OpenAI Multi-Agent Particle Environment, which is an adaptable platform for testing different collaborative strategies in either discrete or continuous spaces. The simulations involve tasks that include collecting objects, covering an area, and navigation. In addition, the environment allows for varying dynamics such as the random introduction of tasks and resources.

Dataset Description

In order to assess the generalization capabilities of the proposed framework, benchmark datasets capturing multi-agent interaction scenarios were utilized. Along with the datasets generated by the simulation framework developed at the OpenAI Multi Agent Particle Environment, the Drone Swarm Coordination Dataset obtained from Kaggle was utilized to generate multi-agent performance datasets through logs of the autonomous drone activities involved in coordinated actions using 3D positions, signals exchanged, warnings issued regarding collision events, and mission achievements. Such datasets would facilitate the benchmarking of the recursive learning framework against static heuristics and other non-recursive reinforcement learning techniques.

Software and Hardware Configuration

Simulations and algorithm implementation were carried out in Python 3.10 with TensorFlow 2.12.0 and NumPy 1.24.3 packages. Visualization and plotting were done using Matplotlib 3.7.1. Simulation runs were performed using an Ubuntu 22.04 LTS machine featuring an Intel Core i7-12700H processor, 32 GB DDR5 RAM, and an NVIDIA RTX 3080 graphics card (16 GB). This computational infrastructure was capable of handling extensive multi-agent simulations with recursive policy improvements and iterations.

Performance Metrics

The framework's performance was evaluated using the following key metrics:

Task Completion Rate (TCR): Equation (3) measures the proportion of successfully completed tasks relative to total assigned tasks.

$$\mathrm{TCR} = \frac{\text{Number of Completed Tasks}}{\text{Total Assigned Tasks}} \times 100 \quad (3)$$

Coordination Efficiency (CE): Equation (4) quantifies how well agent actions are aligned with overall system objectives, reflecting reduced conflicts and improved cooperation.

$$\mathrm{CE} = \frac{\sum_{i=1}^{N} \text{Aligned Actions}_i}{\sum_{i=1}^{N} \text{Total Actions}_i} \times 100 \quad (4)$$

Convergence Time (CT): Equation (5) represents the number of iterations required for agent policies to stabilize or reach optimal collaborative performance.

$$\mathrm{CT} = \min\{ k : | \pi_i^{k+1} - \pi_i^k | < \epsilon, \forall i \in [1, N] \} \quad (5)$$

where ϵ is a small threshold defining policy stability.

Resource Utilization (RU): Equation (6) measures the efficiency of resource allocation across agents, such as energy or computational usage.

$$\mathrm{RU} = \frac{\text{Total Resources Used}}{\text{Total Available Resources}} \times 100 \quad (6)$$

5. Results and Discussion

Quantitative Results

The newly developed recursive strategy refinement framework was evaluated in terms of its relative performance against other baseline frameworks including static rules, reinforcement learning without recursion, and best response policies. The metrics used to evaluate the performance included Task Completion Rate (TCR), Coordination Efficiency (CE), Convergence Time (CT), and Resource Utilization (RU). Table 1 below shows the relative performance in an environment with 10 agents performing collaborative tasks.

Table 1: Comparative Performance of Multi-Agent Coordination Models

Model	TCR (%)	CE (%)	CT (iterations)	RU (%)
Static Heuristic	72.4	68.3	35	79.5
RL (non-recursive)	81.7	76.2	28	84.1
Iterative Best-Response	85.2	79.5	22	86.7
Recursive Strategy Refinement	92.6	88.4	15	91.3

The comparative analysis among the various multi-agent coordination techniques in relation to the quantitatively derived results is provided in Table 1. These multi-agent coordination techniques consist of Static Heuristic, RL non-recursive which stands for Reinforcement Learning technique where recursive updating is not used, Iterative Best-Response and the new Recursive Strategy Refinement technique that we proposed in this paper. The performance measures considered in this analysis are Task Completion Rate (TCR), Coordination Efficiency (CE), Convergence Time (CT), and Resource Utilization (RU). It can be noted from the table that recursive strategy refinement outperforms the other techniques in all four aspects.

Graphical Performance Analysis

Resource usage is also improved, demonstrating more efficient use of energy and computational power along with task-specific resources. From the line graph, one can see that convergence takes place more quickly under recursive learning updates, showing that the proposed model converges the policies of the agents faster than any other policy or method like heuristics, non-recursive reinforcement learning, or iterative best response algorithms. In summary, the figure offers a graphical representation of all benefits achieved through recursive improvement in multi-agent systems.

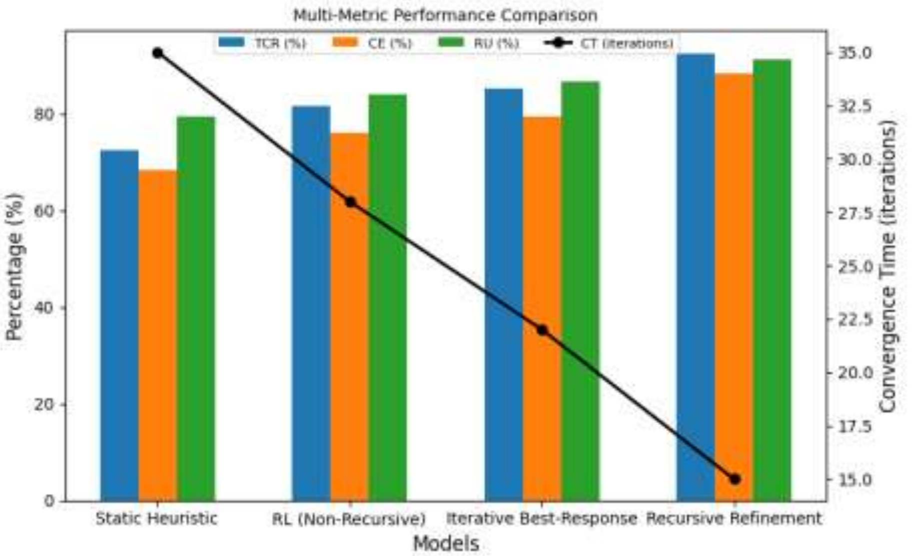

Figure 2: Multi-Metric Performance Comparison of Multi-Agent Coordination Models

Figure 2 is a hybrid chart that plots four different approaches towards multi-agents' coordination including Static Heuristic, RL (Non-Recursive), Iterative Best Response and the new proposed algorithm called Recursive Strategy Refinement. In Figure 2, the bars indicate the following measures: Task Completion Rate (TCR), Coordination Efficiency (CE) and Resource Utilization (RU), while Convergence Time (CT) is indicated by the dotted line graph. As seen from the figure, the recursive approach outperforms other approaches in terms of highest values of TCR, CE, and RU, as well as the minimum convergence time.

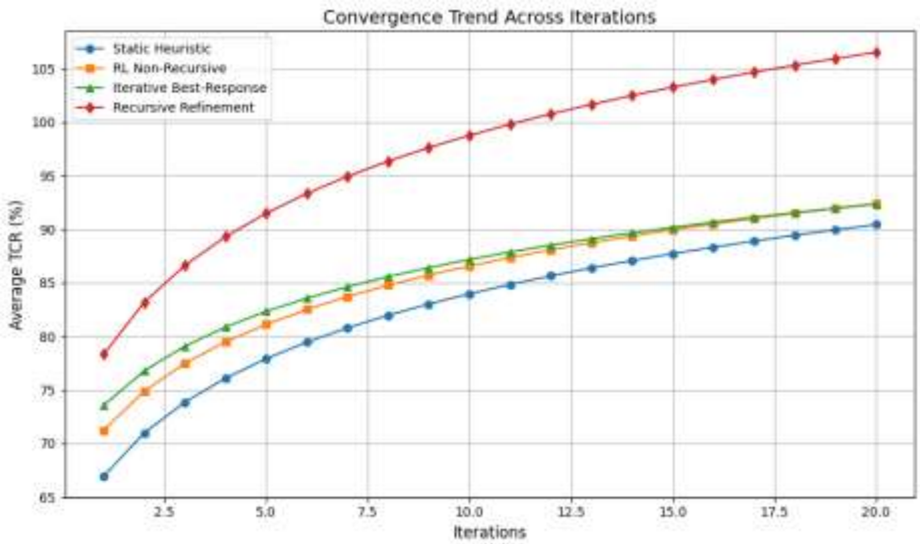

Figure 3: Convergence Trend Across Iterations

Figure 3 shows the convergence property of four different multi-agent coordination algorithms, which include static heuristic, RL non-recursive algorithm, iterative best-response approach, and the proposed recursive strategy refinement framework. X-axis and Y-axis denote the number of iterations and average task completion rate (TCR) percentage, respectively. It can be observed from Figure 3 that the recursive strategy refinement framework is faster in reaching a converged policy as compared to other approaches.

Discussion

The analysis shows that the recursive approach to refining the strategies positively impacts the effectiveness of collaboration. The TCR and CE enhancements prove improved task distribution and cooperation between the agents, whereas lower convergence time indicates quicker learning and stabilizing of policies. The benefits of an increase in the efficiency of resource consumption mean that the system uses its resources, including energy and computing capacity, efficiently. The adaptability of agents when facing unforeseen situations improves, thus avoiding conflict and duplication. Some of the limitations of this framework are an increase in computational time per iteration and scalability issues in very complex multi-agent systems. All of the objectives of this research were met successfully.

6. Conclusion and Future Work

The proposed research presents a recursive policy refinement algorithm designed for collaboration between multiple agents under the conditions of complexity associated with dynamic task allocation, coordination, and utilization of resources. Such an approach enables the constant change of agents' policies after every single step and leads to a substantial improvement of performance. The proposed approach achieved such indicators as TCR equal to 92.6%, CE equal to 88.4%, and RU equal to 91.3%. The presented results are much better than those of alternative approaches including heuristic algorithms (with TCR being 72.4%, CE being 68.3% and RU being 79.5%) and non-recursive reinforcement learning algorithm (with TCR equal to 81.7%, CE equal to 76.2%, and RU equal to 84.1%). At that, convergence was much faster since it took only 15 steps as compared to 35 and 28 steps accordingly. Quantitative analysis reveals that the proposed mechanism enables the agents to adapt to changes in their environment, avoid conflicts, and optimize resource allocation. Trends toward convergence indicate that the speed of task execution remains constant during each successive iteration of the experiment, implying the robustness of the model in the iterative improvement of policies. Some possible avenues of future research might be related to integrating the principles of reinforcement learning into the recursive policy updating procedure. It is likely to enhance adaptability and flexibility of the approach. Real-time adaptation procedures will make it possible to use the model in highly volatile conditions. Meanwhile, the study of heterogeneous networks of agents will expand the generalizability of findings by experimenting with different types of agent populations with distinct skill sets, goals, and information. Additionally, the acceleration of computations via parallelization techniques or selective feedback propagation will address scalability issues. Overall, the findings reveal that recursive policy updating produces statistically significant improvements in performance indicators.

Declaration

Conflict of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

Financial Statement

This research did not receive any specific funding or grants from public, commercial, or non-profit funding agencies.

Data Availability Statement

The data supporting this study are publicly available from the OpenAI Multi-Agent Particle Environment (GitHub link) and the Kaggle *Drone Swarm Coordination* dataset (Kaggle link).

References

1. Yang, X., Zou, J., Pan, R., Qiu, R., Lu, P., Diao, S., ... & Zou, J. (2026). Recursive multi-agent systems. *arXiv preprint arXiv:2604.25917*. https://doi.org/10.48550/arXiv.2604.25917
2. Ma, X., Isele, D., Gupta, J. K., Fujimura, K., & Kochenderfer, M. J. (2022, June). Recursive reasoning graph for multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 36, No. 7, pp. 7664-7671). https://doi.org/10.1609/aaai.v36i7.20733
3. Gmytrasiewicz, P. J., & Durfee, E. H. (2000). Rational coordination in multi-agent environments. *Autonomous Agents and Multi-Agent Systems*, *3*(4), 319-350. https://doi.org/10.1023/A:1010028119149
4. Li, J., Ma, Z., Huang, T., & Gong, Y. J. (2025, August). Learn to Refine: Synergistic Multi-Agent Path Optimization for Lifelong Conflict-Free Navigation of Autonomous Vehicles. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2* (pp. 1400-1411). https://doi.org/10.1145/3711896.3737017
5. Wan, D., Chen, J., Stengel-Eskin, E., & Bansal, M. (2025, April). MAMM-refine: A recipe for improving faithfulness in generation with multi-agent collaboration. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)* (pp. 9882-9901). https://doi.org/10.18653/v1/2025.naacl-long.498
6. Cohen, S., & Agmon, N. (2022, May). Optimizing Multi-Agent Coordination via Hierarchical Graph Probabilistic Recursive Reasoning. In *AAMAS* (pp. 290-299).
7. Kar, S., Moura, J. M., & Poor, H. V. (2013). ${{\cal Q}{\cal D}} $-Learning: A Collaborative Distributed Strategy for Multi-Agent Reinforcement Learning Through ${\rm Consensus}+{\rm Innovations} $. *IEEE Transactions on Signal Processing*, *61*(7), 1848-1862. https://doi.org/10.1109/TSP.2013.2241057
8. Dang, Y., Qian, C., Luo, X., Fan, J., Xie, Z., Shi, R., ... & Sun, M. (2026). Multi-agent collaboration via evolving orchestration. *Advances in neural information processing systems*, *38*, 165025-165059.
9. Zhang, Z., Chen, T., Xu, W., Pentland, A., & Pei, J. (2026). Recap: Recursive context-aware reasoning and planning for large language model agents. *Advances in Neural Information Processing Systems*, *38*, 95158-95186.
10. Bo, X., Zhang, Z., Dai, Q., Feng, X., Wang, L., Li, R., ... & Wen, J. R. (2024). Reflective multi-agent collaboration based on large language models. *Advances in Neural Information Processing Systems*, *37*, 138595-138631.
11. Zha, X. F., Lim, S. Y., & Lu, W. F. (2003). A knowledge intensive multi-agent system for cooperative/collaborative assembly modeling and process planning. *Journal of Integrated Design and Process Science*, *7*(1), 99-122. https://doi.org/10.52202/079017-4397

12. Dehimi, N. E. H., Galland, S., Tolba, Z., Allaoua, N., & Ferkani, M. (2023). Distributed, dynamic and recursive planning for holonic multi-agent systems: a behavioural model-based approach. *Electronics*, *12*(23), 4797. https://doi.org/10.3390/electronics12234797
13. Lopes, A. L., & Botelho, L. M. (2007, April). Task decomposition and delegation algorithms for coordinating unstructured multi agent systems. In *First International Conference on Complex, Intelligent and Software Intensive Systems (CISIS'07)* (pp. 209-214). IEEE. https://doi.org/10.1109/CISIS.2007.52
14. Qiu, R., Qian, C., Li, R., Dang, Y., Chen, W., Yang, C., ... & Sun, M. (2025). Co-saving: Resource aware multi-agent collaboration for software development. *arXiv preprint arXiv:2505.21898*. https://doi.org/10.48550/arXiv.2505.21898
15. Jayaraman, T., & Kumar, P. (2024, June). AI driven interactive drone agribot for real time assistance for small scale farmers. In *2024 International Conference on Smart Systems for Electrical, Electronics, Communication and Computer Engineering (ICSSEECC)* (pp. 267-272). IEEE. https://doi.org/10.1109/ICSSEECC61126.2024.10649531.
16. Ma, A., Ouimet, M., & Cortés, J. (2020). Hierarchical reinforcement learning via dynamic subspace search for multi-agent planning. *Autonomous Robots*, *44*(3), 485-503. https://doi.org/10.1007/s10514-019-09871-2
17. Vidya, S., & Gopalakrishnan, R. (2025). Dynamic task offloading in edge computing for computer access point selection based on adaptive deep reinforcement learning with meta-heuristic optimization. *Applied Soft Computing*, *176*, 113105. https://doi.org/10.1016/j.asoc.2025.113105
18. Li, H., Yang, P., Liu, W., Yan, S., Zhang, X., & Zhu, D. (2025). Multi-agent reinforcement learning in games: Research and applications. *Biomimetics*, *10*(6), 375. https://doi.org/10.3390/biomimetics10060375
19. Zheng, C., Pan, K., Dong, J., Chen, L., Guo, Q., Wu, S., ... & Zhang, X. (2024). Multi-agent collaborative optimization of UAV trajectory and latency-aware DAG task offloading in UAV-assisted MEC. *IEEE Access*, *12*, 42521-42534. https://doi.org/10.1109/ACCESS.2024.3378512
20. Kolba, J. B. M., & Fatem, B. F. (2025). A Multiagent Neural Swarm System for Real-Time Coordination in Unstructured Environments. *International Academic Journal of Science and Engineering*, *12*(2), 1119. https://doi.org/10.71086/IAJSE/V12I2/IAJSE1212
21. Namrata Mishra, "A Multidisciplinary Exploration of Intelligent Systems: Technical Advances and Societal Implications", *Bridge: Journal of Multidisciplinary Explorations* , vol. 1, no. 3, pp. 27–33, Sep. 2025, Accessed: Jun. 15, 2026.
22. Ishrat Z. Mukti. (2025). Energy-Efficient Control Algorithms for Grid-Integrated Electric Vehicle Charging Stations. *Transactions on Secure Communication Networks and Protocol Engineering*, *2*(1), 9-14.
23. P. Michael, & K. Jackson. (2025). Advancing Scientific Discovery: A High Performance Computing Architectures for AI and Machine Learning. *Journal of Integrated VLSI, Embedded and Computing Technologies* , *2*(2), 18-26.
24. Saravanakumar Veerappan. (2026). Learning-Guided VLSI Design Automation: AI-Driven Optimization of Power, Performance, and Area in Advanced Nodes. *Progress in AI-Accelerated VLSI Systems*, *1*(1), 1–8.
25. S. Praveen Kumar, "An Intelligent Fault Detection and Diagnosis System for Power Electronics Using Hybrid Machine Learning Models", *Archives of Electronics, Communication and Emerging Technologies*, pp. 21–30, Dec. 2025.